Statistical Thermodynamic Isotherm-Based Model for Activity Coefficients in Complex Aqueous Solutions with Atmospheric Aerosol Applications

A THESIS
SUBMITTED TO THE FACULTY OF
UNIVERSITY OF MINNESOTA
BY

Peter Brian Ohm

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE IN MECHANICAL ENGINEERING

Cari S. Dutcher, Advisor

May, 2015

Acknowledgements

I would like to acknowledge my friends and family and thank them for their praise and support. I would like to thank my advisor, Cari Dutcher, for taking me as a student and seeing my project through to the end. I would like to thank all of my fellow Dutcher lab group members for providing an enriching and entertaining work and social environment. Finally, I would like to thank the faculty and staff at the University of Minnesota for their support in my research endeavors.

Abstract

Aqueous aerosol particles are nearly ubiquitous in the atmosphere and yet there remain large uncertainties in their formation processes and ambient properties. The uncertainty is in part due to the complex nature of the individual particle microenvironment, which can involve a myriad of chemical components and multiple phases. The calculation of gas-liquid-solid equilibrium partitioning of the water, electrolyte, and soluble organic components is critical to accurate determination of atmospheric chemistry properties and processes such as new particle formation and activation to cloud condensation nuclei. Previously, a transformative model for capturing thermodynamic properties of multicomponent aqueous solutions over the entire concentration range (Dutcher et al. J. Phys. Chem 2011, 2012, 2013) was developed using statistical mechanics and multilayer adsorption isotherms. That model needed only a few adsorption energy values to represent the solution thermodynamics of each solute. In the current work, we posit that the adsorption energies are due to dipole-dipole electrostatic forces in solute-solvent and solvent-solvent interactions. This hypothesis was tested in aqueous solutions on (a) thirty-seven 1:1 electrolytes, over a range of cation sizes, from H^+ to tetrabutylammonium, for common anions including Cl^-, Br^-, I^-, NO_3^-, OH^-, ClO_4^-, and (b) twenty water soluble organic molecules including alcohols and polyols. For both electrolytes and organic solutions, the energies of adsorption can be calculated with the dipole moments of the solvent, molecular size of the solvent and solute, and the solvent-solvent and solvent-solute intermolecular bond lengths. Many of these physical properties are available in the literature, with the exception of the solute-solvent intermolecular bond lengths. For those,

predictive correlations developed here enable estimation of solute and solvent solution activities for which there are little or no activity data. The model was successfully validated using thirty-seven 1:1 electrolytes and twenty non-dissociating organic solutions (Ohm et al. J. Phys. Chem. 2015). However, careful attention is needed for weakly dissociating semi-volatile organic acids. Dicarboxylic acids such as malonic and glutaric acid are treated here as a mixture of non-dissociated organic species (HA) and dissociated organic species ($H^+ + A^-$). It was found that the apparent dissociation was greater than that predicted by known dissociation constants alone, emphasizing the effect of dissociation on activity coefficient predictions. To avoid additional parameterization from the mixture approach, an expression was used to relate the Debye-Hückel hard-core collision diameter to the adjustable solute-solvent intermolecular distance. This work results in predictive correlations for estimation of solute and solvent solution activities for which there are little or no activity data.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Atmospheric aerosol particles, such as those that form clouds, are one of the major contributing factors to our climate's behavior, yet remain the largest source of uncertainty in climate modeling.[1] Atmospheric aerosol particles affect our climate in two ways. First, aerosol particles can reflect or absorb solar radiation, directly altering the amount of solar radiation that reaches the earth's surface.[2,3] Second, atmospheric aerosol particles can alter cloud properties causing clouds to scatter more incoming solar radiation back into space, indirectly causing less solar radiation to reach the earth's surface.[4] These effects have yet to be fully understood and are a major source of uncertainty in climate predictions. Part of the challenge in understanding aerosol particles is their complexity. Aerosol particles range in size by orders of magnitude, from nanometers to hundreds of micrometers, and can contain multiple liquid and sold phases filled with hundreds or thousands of unique compounds. The structure, composition and size of atmospheric aerosols are governed by the thermodynamic properties of the chemical species present in the particle. Of these thermodynamic properties, such as surface tension, chemical activity is one of the most fundamental, and can be used to derive other important thermodynamic properties, such as the surface tension of an aerosol or the relative humidity. Determining chemical activity is critical in determining the size distributions and compositions of atmospheric aerosols.

Chemical activity is the measure of the chemical effectiveness of a component in a solution. The activity of the solvent, a_j, is defined as:

$$\mu_j = \mu_j^o + kT \ln a_j \tag{1.1}$$

where μ_j is the chemical potential, μ_j^o is the chemical potential at a given reference state, k is Boltzmann's constant, and T is temperature. The ratio between the chemical activity and the concentration is known as the activity coefficient, $\gamma_j = a_j / x_j$, where here concentration is given as mole fraction of species j. The activity coefficient indicates how much the real solution deviates from an ideal solution. For example, if the activity coefficient is greater than unity, the solution behaves as if there is more of that species than the concentration would suggest. However, if the activity coefficient is equal to unity, then the activity exactly equals the concentration, an ideal relationship known as Raoult's Law.

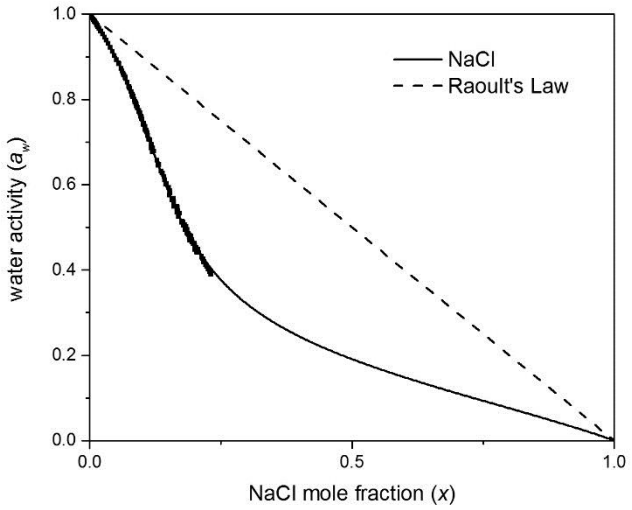

Figure 1.1. Roult's Law. Plot showing the relationship between the water activity and the concentration of solute. For an aqueous NaCl solution, the relationship deviates quite far from the ideal case shown by Raoult's Law. The solid line is the model prediction from Ohm et al.[5] The experimental data for NaCl from Archer, Tang et al., Cohen et al., and Chan et al.[6–9]

In addition to the activity coefficient, the osmotic coefficient is defined by

$$\phi = \frac{-\ln a_w}{M_w m_j v_j} \tag{1.2}$$

where a_w is the water activity of the aqueous solution, M_w is the molar mass of water, m_j is the molarity of solute j in solution, and v_j is the number of moles of ions into which one mole of solute disassociates ($v_j = 1$ for organics). The osmotic coefficient provides additional information on the thermodynamic properties of species in solution. In addition, the osmotic coefficient allows for more sensitive parameterization of activity coefficient models.

Activity coefficients are used in aerosol science to determine properties of the particles, including particle size distribution and particle optical properties. In addition, the thermodynamic properties of the particle dictate the chemical uptake and phase partitioning in the solid-liquid (aqueous) – liquid (organic) and vapor phases. The size of a particle is governed by the thermodynamics surrounding particle growth. Chemical thermodynamic properties, such as activity coefficient, can alter the formation and growth processes of aerosol particles. The chemical composition of the particle informs particle size and growth, determining the overall particle size distribution of aerosol particles. Particle size is an important factor in determining the optical properties of clouds.[10] The water activity of these aerosols can be used to determine the surface tension of these aerosol particles,[11] and the particle growth and phase morphology can be determined from surface tension. Hence, having an accurate model for chemical activity at atmospherically relevant conditions is important for climate and cloud modeling.

Another important application of activity coefficients arises in the food industry. For food processing and preservation processes water activity is an important factor in controlling the rate of deterioration of a food. Water activity affects the caking and clumping of powders and the textural properties of foods. Micro-organisms that commonly cause food spoilage are, in general, inhibited in food with a water activity below 0.6. Water activity can be used to determine a food's shelf life stability, predicting which microorganisms will be potential sources of spoilage and infection.[12]

However, multiple aspects can complicate the modeling of activity. The presence of electrolytes in solution introduces multiple ions from the disassociation of the electrolyte. The electrostatic forces and ionic interactions that arise from the presence of ions add complexity to the solution. One of the first models for activity coefficients in electrolyte solutions was the Debye-Hückel equation.[13] The basic principle behind the Debye-Hückel equation is that the nonideal behavior of an electrolyte solution is comprised of an electrostatic contribution and a short-range nonelectrolyte contribution.[14] While the Debye-Hückel equation generally gives reasonable results for low concentrations (<0.001 molarity), the theory fails due to oversimplifications for solutions with high electrolyte concentrations, ions with higher charge, and unsymmetrical electrolytes. Despite the Debye-Hückel equation's shortcomings, the theory is still often used as a fundamental basis for other activity coefficient models. The isotherm model derived in this paper uses the Debye-Hückel theory as a basis for determining the long-range electrostatic interactions. Other activity coefficient models are outlined below.

The Universal quasichemical Functional-group Activity Coefficient (UNIFAC) method uses defined functional groups on the molecules in the mixture to calculate the activity coefficient.[15] The UNIFAC model was originally developed to handle organic mixtures, though it has since been extended to handle electrolytes through the addition of an electrolyte part based on the Debye-Hückel theory and a mixture part based on a viral equation for solvent-ion interactions. The Non-Random Two Liquid (NRTL) equation, another activity coefficient model, is based on the hypothesis that the local concentration around a molecule is different from that of the bulk concentration.[16] The NRTL model and the UNIFAC model are examples of local composition models. Local composition models incorrectly assume that the local composition around molecule *i* is independent of the local composition around molecule *j*, and thus are inaccurate under moderate to high concentrations of solute.

The Statistical Associating Fluid Theory (SAFT) model and its derivatives represent molecules as hard chains of spherical segments, from which the excess Helmholtz energy can be calculated.[17] When extended to electrolyte solutions, SAFT models typically contain four terms: a segment term that accounts for the nonideality of the reference fluid (LJ fluid) of nonbonded chain segments, a chain term that accounts for covalent bonding , an association term, and a term for ionic interaction.[18] The ion interaction term accounts for the long-range electrostatic interactions. SAFT model for electrolytes address specific volume properties with reasonable accuracy, but often contain many fit parameters,[18] or are widely inaccurate for electrolytes, requiring reparameterization.[19]

Models such as the Pitzer model and the Pitzer-Simonson-Clegg (PSC) model work by calculating the excess Gibbs energy.[20–22] The PSC uses a Debye-Hückel term for long-range interactions and Margules expansion for short-range interactions. The model is useful for systems of high electrolyte concentrations including fused salts when data is available for validations. However, the model is also a highly parameterized model, and yields unphysical results for less soluble systems.

The LIFAC model is based on the interactions between the different species as well as solvent-solvent, solvent-solute, and solute-solute interactions as predicted by statistical thermodynamics. Electrolyte solutions are treated as nonelectrolyte solutions with charge interactions. In addition, LIFAC uses the Debye-Hückel theory and UNIQUAC to handle long-range and short-range interactions.[23] Expanding on the LIFAC model, the AIOMFAC model includes a modified UNIFAC model in addition to a Pitzer-like ion-interaction model and an organic-inorganic mixing term similar to the one in LIFAC.[24] This model successfully predicts solutions of mixed organic and inorganic solutes, a rare feature among activity coefficient models. However, AIOMFAC fails to predict activity coefficients for solutions at high concentrations.

Existing models for predicting chemical activity, such as those mentioned above, have been successful at predicting and modeling activity coefficients at low solute concentrations. These models, however, often fail to accurately predict activity coefficients at high solute concentrations, or require a large number of empirical parameters, which in turn requires a large amount of reliable data in order to estimate these parameters. Atmospheric aerosols can have supersaturated concentrations of solute, and the is a lack of

activity coefficient data at atmospheric conditions makes many models undesirable choices when dealing with atmospheric aerosols.

Sorption isotherm models describe the adsorption of particles at a solid lattice. In electrolyte solutions, hydration shells form around ions, similar to multilayer sorption. Stokes and Robinson first noted the parallel between multilayer sorption and the formation of hydrogen shells around ions in aqueous solutions due to intermolecular forces.[25]

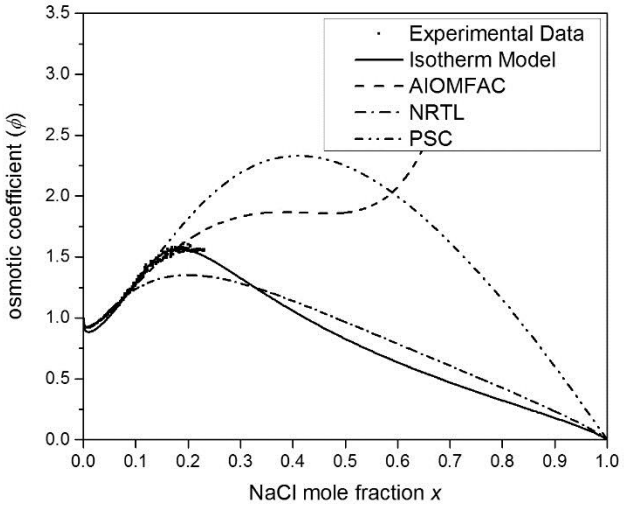

Figure 1.2. Osmotic Coefficient Prediction Comparisons for Aqueous NaCl Solutions. Osmotic coefficient versus NaCl mole fraction for various activity coefficient models. Experimental data from Archer, Tang et al., Cohen et al., and Chan et al.[6–9]. Isotherm model,[5] AIOMFAC,[24] NRTL,[16] PSC[20–22]. The AIOMFAC, NRTL and PSC models struggle to either match the available data at higher concentrations or fail to limit to an osmotic coefficient of zero at a NaCl mole fraction of one.

The isotherm based activity coefficient model, derived by Dutcher et al.[26–28] and rederived here in Chapter 2, requires only a few parameters to accurately predict activity coefficients across the entire concentration range. The work presented here aims to both decrease the number of parameters needed for the model, as well as to give physical interpretation to the fit parameters required.

This thesis covers the derivation, implementation and interpretation of an isotherm based activity coefficient model. Chapter 2 begins with a review of the statistical mechanical derivation for the isotherm based model for activity coefficients, based on work by Dutcher et al.[26–28] Following the derivation, sample MATLAB code is provided along with an explanation of how the model was scripted and solved using MATLAB. Chapter 3 introduces coulombic electrostatic interactions into the isotherm based model, as well as providing a method for the interpretation and prediction of the energy interaction parameters present in the model. The model treatment developed in Chapter 3 lead to a significant reduction in the number of model parameters required. Chapter 4 specifically addresses organic acids by introducing a method to account for the partial disassociation of the weak acid. Finally, Chapter 5 provides a summary of the results discussed in this Thesis as well as a discussion of future work.

Chapter 2

Multilayer Adsorption Isotherm Derivation and Numerical Implementation

Multilayer adsorption isotherm models successfully describe a wide array of sorption processes. In electrolyte solutions hydration shells form around ions, similar to multilayer sorption. Stokes and Robinson first noted the parallel between multilayer sorption and the formation of hydration shells around ions in aqueous solutions due to intermolecular forces.[25] The concept behind using an adsorption isotherm model applied to a solution is to treat the solute as a lattice onto which the solvent water molecules adsorb, forming hydration shells. Sorption isotherms, such as the Brunauer-Emmett-Teller (BET) or the Guggenheim-Anderson-de Boer (GAB), are able to successfully model the relation between water activities and solute concentrations for high solute concentrations ($a_w < 0.7$).[29–31] The following derivation, based on work done by Dutcher et al.,[26–28] extends adsorption isotherms to electrolyte solutions across the entire range of $0 < a_w < 1$ using a unified treatment. The model works by combining the adsorption multilayer approach to capture short-range electrostatic interaction with the Pitzer modified Debye-Hückel expression for the long-range electrostatic interactions.

2.1 Statistical Mechanics

Consider a system containing N_w water molecules and N_j molecules of solute j. Each solute molecule has r_j sorption sites, $r_j N_j$ total sorption sites, and n_j sorption layers

surrounding the solute, consisting of $(n_j - 1)$ monolayers and a single outer multilayer. In the first monolayer, $X_{j,1}$ water molecules occupy $r_j N_j$ sorption sites. Only one water molecule is allowed per sorption site, leaving $(r_j N_j - X_{j,1})$ sites unoccupied. The number of distinguishable ways the $X_{j,1}$ water molecules can sorb onto the $r_j N_j$ sorption sites of solute j is given by

$$\Omega_{j,1} = \frac{(r_j N_j)!}{(r_j N_j - X_{j,1})!\,(X_{j,1})!} \tag{2.1}$$

For the subsequent monolayers, the $X_{j,i}$ water molecules occupying the ith monolayer are sorbed on top of the $X_{j,i-1}$ water molecules sorbed in the $(i-1)$th monolayer. With only one water molecule allowed per sorption site per layer there are $(X_{j,i-1} - X_{j,i})$ unoccupied sorption sites in the ith monolayer of solute j. The number of distinguishable ways to arrange $X_{j,i}$ water molecules on $X_{j,i-1}$ sites is given by

$$\Omega_{j,i} = \frac{(X_{j,i-1})!}{(X_{j,i-1} - X_{j,i})!\,(X_{j,i})!} \tag{2.2}$$

where $2 \leq i \leq n_j - 1$.

Finally, for the n_jth layer, the multilayer, there is no limit to the number of molecules that can sorb to a single site. The number of distinguishable ways to arrange the $X_{j,n}$ water molecules of the multilayer onto the $X_{j,n-1}$ sorption sites of the previous monolayer is given by

$$\Omega_{j,n} = \frac{(X_{j,n-1} + X_{j,n} - 1)!}{(X_{j,n-1} - 1)!\,(X_{j,n})!} \approx \frac{(X_{j,n-1} + X_{j,n})!}{(X_{j,n-1})!\,(X_{j,n})!} \tag{2.3}$$

where both $X_{A,n-1} \gg 1$ and $X_{A,n} \gg 1$.

For an aqueous system containing N_j molecules of solute j and an arbitrary number solute species, ie. $j = A, B, \ldots Z$, the number of distinguishable ways in which the solute molecules can mix is given by

$$\Omega_{A,B,\ldots Z} = \frac{\left(\sum_{j=A}^{Z} N_j\right)!}{\prod_{j=A}^{Z}\left(N_j!\right)} \tag{2.4}$$

The total number of distinguishable arrangements for water molecules to sorb to an arbitrary number of solute species is given by

$$\Omega = \Omega_{\mathrm{A,B,\ldots Z}} \prod_{\mathrm{j=A}}^{\mathrm{Z}} \prod_{\mathrm{i=1}}^{\mathrm{n_j}} \Omega_{\mathrm{j,i}} \tag{2.5}$$

Combining equations (2.1) through (2.5), gives

$$\Omega = \frac{\left(\sum_{j=A}^{Z} N_j\right)!}{\prod_{j=A}^{Z}\left(N_j!\right)} \prod_{j=A}^{Z} \left(\frac{\left(r_j N_j\right)!}{\left(r_j N_j - X_{j,1}\right)! \, X_{j,1}!} \frac{\left(X_{j,n-1} + X_{j,n}\right)!}{X_{j,n-1}! \, X_{j,1}!} \prod_{i=2}^{n-1} \frac{\left(X_{j,i-1}\right)!}{\left(X_{j,i-1} - X_{j,i}\right)! \, X_{j,i}!} \right) \tag{2.6}$$

Using Stirlng's approximation, $\ln(N!) \approx N \ln(N) - N$, and the statistical thermodynamic definition of entropy, $S = k \ln(\Omega)$, a statistical thermodynamic entropy can be given by

$$S = k \sum_{j=A}^{Z} \left[N_j \ln \left(\left(\frac{\sum_{k=A}^{Z} N_k}{N_j} \right) \left(\frac{r_j N_j}{r_j N_j - X_{j,1}} \right)^{r_j} \right) + X_{j,1} \ln \left(\frac{r_j N_j - X_{j,1}}{X_{j,1} - X_{j,2}} \right) \right.$$
$$+ \sum_{i=2}^{n-2} \left(X_{j,i} \ln \left(\frac{X_{j,i-1} - X_{j,i}}{X_{j,i} - X_{j,i+1}} \right) \right)$$
$$+ X_{j,n-1} \ln \left(\frac{\left(X_{j,n-2} - X_{j,n-1}\right)\left(X_{j,n} - X_{j,n-1}\right)}{\left(X_{j,i-1}\right)^2} \right) \tag{2.7}$$
$$\left. + X_{j,n} \ln \left(\frac{X_{j,n} - X_{j,n-1}}{X_{j,n}} \right) \right]$$

where k is Boltzmann's constant.

The goal is to maximize the system's entropy given by equation (2.7). To do this some constraints on the system need to be defined. First, a constraint for the total number of water molecules is needed. Each water molecule is statistically associated with only one solute, either in a monolayer or the multilayer. The total number of water molecules is then given by

$$N_w = \sum_{j=A}^{Z} \sum_{i=1}^{n} X_{j,i} \tag{2.8}$$

The maximization of entropy in the system is constrained by the energy of the system. The total change in energy, E, of the system due to the sorption of water molecules from a bulk free liquid state is given by

$$E = -\sum_{j=A}^{Z} \sum_{i=1}^{n} X_{j,i}\, \varepsilon_{j,i}{}^{*} \tag{2.9}$$

where $\varepsilon_{j,i}{}^{*} = E_{j,i}{}^{*} - E_L$ for the ith layer of solute j. In the multilayer adsorption of a gas onto a solid substrate, E_L is the energy of condensation from the gas to the liquid state, and $E_{j,i}{}^{*}$ is the energy of sorption of the gas in the ith layer. For multilayer sorption in solutions, E_L is the energy of the free water in the bulk and $E_{j,i}{}^{*}$ is the energy of sorbed water.

In the derivation by Anderson (J. Am. Chem. Soc. 1946, 68, 686–691),[32] the sorption energy of the multilayer, $E_{j,n}{}^{*}$, differs from the heat of liquefaction, E_L, by an amount d_j for each solute j. Thus $E_{j,n}{}^{*} = E_L + d_j$ and $\varepsilon_{j,n}{}^{*} = d_j$. Following Anderson, the additional energy of the multilayer adsorption, d_j, is incorporated into the terms for monolayer adsorption energy, $E_{j,i}{}^{*}$,so that $E_{j,i}{}^{*} = E_{j,i} + d_j$ and $\varepsilon_{j,i}{}^{*} = \varepsilon_{j,i} + d_j$. Substituting these equations into equation AI results in

$$E = -\sum_{j=A}^{Z}\sum_{i=1}^{n-1} X_{j,i}\left(\varepsilon_{j,i} + d_j\right) - \sum_{j=A}^{Z} X_{j,n} d_j \tag{2.10}$$

The goal is to maximize the total statistical thermodynamic entropy, given by equation (2.7), using equation (2.10) as a constraint on the energy and the Lagrange method of undetermined multipliers

$$\left(\frac{\partial \ln \Omega}{\partial X_{j,i}}\right)_{N,X} - \frac{1}{kT}\left(\frac{\partial E}{\partial X_{j,i}}\right)_{N,X} = 0 \tag{2.11}$$

where N and X are explained below.

The entropy for the monolayers $(i = 1,2,\ldots n-1)$ of each solute j is maximized using equation (2.11), where the following variables are fixed: the total number of particles (N) of each species, the total number of sorbed water molecules (including the multilayer)

associated with each individual solute, and the number of sorbed water molecules (X) for each monolayer of each solute, except for those in layer i of solute j. The results are expressed in terms of energy parameters $C_{j,i}$ for sorption for each monolayer i of solute j. For the first monolayer $(i = 1)$ the energy parameter is given by

$$\frac{\left(X_{j,n} - X_{j,n-1}\right)\left(X_{j,1} - X_{j,2}\right)}{\left(X_{j,n}\right)\left(r_j N_j - X_{j,1}\right)} = \exp\left(\frac{\varepsilon_{j,1}}{kT}\right) = C_{j,1} \tag{2.12}$$

For the intermediate monolayers $(2 \leq i \leq n-2)$ the energy parameter is given by

$$\frac{\left(X_{j,n} - X_{j,n-1}\right)\left(X_{j,i} - X_{j,i+1}\right)}{\left(X_{j,n}\right)\left(X_{j,i-1} - X_{j,i}\right)} = \exp\left(\frac{\varepsilon_{j,i}}{kT}\right) = C_{j,i} \tag{2.13}$$

And finally for the $(n-1)$th layer, the last monolayer before the multilayer, the energy parameter is given by

$$\frac{\left(X_{j,n-1}\right)^2}{\left(X_{j,n}\right)\left(X_{j,n-2} - X_{j,n-1}\right)} = exp\left(\frac{\varepsilon_{j,n-1}}{kT}\right) = C_{j,n-1} \tag{2.14}$$

The entropy of the multilayer $(i = n_j)$ of each solute j is also maximized using the Lagrange method of undetermined multipliers. The total number of particles (N) of each species and the number of sorbed water molecules (X) for each monolayer $(i < n_j)$ are fixed for each solute j. The result is a relationship between the energy parameter K_j for sorption in the multilayer for each solute

$$\frac{1}{K_j}\left(\frac{X_{j,n}}{X_{j,n} + X_{j,n-1}}\right) = \frac{1}{K_k}\left(\frac{X_{k,n}}{X_{k,n-1} + X_{k,n}}\right) \tag{2.15}$$

where $K_j \equiv \exp\left(d_j/kT\right)$, and j and k are any pair of solutes. This ratio for a specific solute, j, can be expressed in terms of all the solutes in the system by multiplying equation (2.15)

by unity (ie. $\left(1+\sum_{k'} X_{k',n}/X_{k,n}\right)/\left(1+\sum_{k'} X_{k',n}/X_{k,n}\right)$, where $k' \neq k$) and applying the equalities of equation (2.15) for each solute $k = A, B, \ldots, Z$

$$\frac{1}{K_j}\left(\frac{X_{j,n}}{X_{j,n}+X_{j,n-1}}\right) = \frac{\left(\sum_{k=A}^{Z} X_{k,n}\right)}{\sum_{k=A}^{Z} K_k\left(X_{k,n-1}+X_{k,n}\right)} \tag{2.16}$$

Substituting equations (2.12) to (2.16) into equation (2.7) yields the most probable distribution, Ω^*, of the number of adsorbed water molecules in all layers for all solutes

$$\begin{aligned}\ln \Omega^* = \sum_{j=A}^{Z}\left[N_j \ln\left(\left(\frac{\sum_{k=A}^{Z} N_k}{N_j}\right)\left(\frac{r_j N_j}{r_j N_j - X_{j,1}}\right)^{r_j}\right)\right] - \sum_{j=A}^{Z}\sum_{i=1}^{n-1} X_{j,i}\left(\frac{\varepsilon_{j,i}}{kT}\right) \\ + \sum_{j=A}^{Z}\sum_{i=1}^{n} X_{j,i} \ln\left(\left(X_{j,n-1}+X_{j,n}\right)/X_{j,n}\right)\end{aligned} \tag{2.17}$$

The Gibbs energy is related to the entropy and enthalpy of the system via

$$G/kT \approx E/kT - \ln \Omega^* \tag{2.18}$$

where the approximation comes from the assumption that the pressure is constant. Using equations (2.10), (2.16), and (2.17), the Gibbs free energy for a system with an arbitrary number of solutes and adsorption layers is

$$\begin{aligned}G/kT = \sum_{j=A}^{Z}\left[N_j\left(ln\left(x_j^*\right) + r_j\, ln\left(1-\frac{X_{j,1}}{r_j N_j}\right)\right)\right] \\ + N_w\, ln\left(\frac{\sum_{j=A}^{Z} X_{j,n}}{\sum_{j=A}^{Z} K_j\left(X_{j,n-1}+X_{j,n}\right)}\right)\end{aligned} \tag{2.19}$$

where x_j^* is defined as the dry mole fraction of solute j, so that $x_j^* = N_j/\sum_k N_k$. Differentiating equation (2.19) with respect to the total water content of the system, N_w,

and applying the partial derivatives of the energy equalities from equations (2.12), (2.13), (2.14), to each $\partial X_{j,i}/\partial N_w$ yields the water activity of the system

$$a_w = \frac{\sum_{j=A}^{Z} X_{j,n}}{\sum_{j=A}^{Z} K_j\left(X_{j,n-1} + X_{j,n}\right)} \tag{2.20}$$

The water activity of the system can be written in terms of the amounts of water adsorbed in the multilayer and outermost monolayer of any individual solute, j, using equation (2.16).

$$a_w = \frac{1}{K_j}\left(\frac{X_{j,n}}{X_{j,n-1} + X_{j,n}}\right) \tag{2.21}$$

The expression for the activity of solute j can be similarly obtained by partial differentiation of the Gibbs free energy, equation (2.19), with respect to N_j

$$a_j = x_j^*\left(\frac{r_j N_j - X_{j,1}}{r_j N_j}\right)^{r_j} \tag{2.22}$$

2.2 Solute Concentration

Combining the expressions for the energy parameters, $C_{j,i}$, given by equations (2.12), (2.13), (2.14), and the expression for water activity, equation (2.21), the number of water molecules in the monolayer of solute j can be written in terms of the number of water molecules in the multilayer, $X_{j,n}$

$$X_{j,i} = X_{j,n}\left(1 - a_w K_j\right)\left(\left(1 - a_w K_j\right)\left(\frac{\sum_{h=i}^{n-2} {a_w}^h {K_j}^h \prod_{k=1}^{h} C_{j,k}}{{a_w}^n {K_j}^n \prod_{k=1}^{n-1} C_{j,k}}\right) + \frac{1}{a_w K_j}\right) \tag{2.23}$$

Substituting equation (2.23) into equation (2.12) and (2.21), the molality of solute j (defined as $m_j = N_j/(M_w N_w)$) is given by

$$m_j = \frac{\left(1 - a_w K_j\right)}{\left(M_w r_j C_{j,1} a_w K_j\right)} \frac{X_{j,n}}{N_w} \times \left[\left(\left(1 - a_w K_j\right) \left(\frac{\sum_{h=1}^{n-2} {a_w}^h {K_j}^h \prod_{k=1}^{h} C_{j,k}}{{a_w}^n {K_j}^n \prod_{k=1}^{n-1} C_{j,k}} \right) + \frac{1}{a_w K_j} \right) \times \left(1 + C_{j,1} a_w K_j\right) - \left(\left(1 - a_w K_j\right) \left(\frac{\sum_{h=2}^{n-2} {a_w}^h {K_j}^h \prod_{k=1}^{h} C_{j,k}}{{a_w}^n {K_j}^n \prod_{k=1}^{n-1} C_{j,k}} \right) + \frac{1}{a_w K_j} \right) \right] \tag{2.24}$$

Summing equation (2.23) over the monolayers and multilayer, the total number of water molecules associated with solute j, written in terms of the number of water molecules in the multilayer of solute j is given by

$$\sum_{m=1}^{n} X_{j,m} = X_{j,n} \left(\sum_{m=1}^{n-1} \left[\left(1 - a_w K_j\right) \left(\left(1 - a_w K_j\right) \left(\frac{\sum_{h=m}^{n-2} {a_w}^h {K_j}^h \prod_{k=1}^{h} C_{j,k}}{{a_w}^n {K_j}^n \prod_{k=1}^{n-1} C_{j,k}} \right) + \frac{1}{a_w K_j} \right) \right] + 1 \right) \tag{2.25}$$

Using equation (2.25) to multiply equation (2.24) by unity results in

$$m_j = \left(\frac{1 - a_w K_j}{M_w r_j a_w K_j} \right) \times \left(\begin{array}{c} \frac{1 - \sum_{i=1}^{n-1} \left({a_w}^i {K_j}^i \left(1 - C_{j,i}\right) \prod_{k=1}^{i-1} C_{j,k} \right)}{\left(1 - a_w K_j\right)^2 \sum_{m=1}^{n-2} \left(m {a_w}^{m-1} {K_j}^{m-1} \prod_{k=1}^{m} C_{j,k} \right)} \\ + \left((n-1) - (n-2) a_w K_j \right) {a_w}^{n-2} {K_j}^{n-2} \prod_{i=1}^{n-1} C_{j,i} \end{array} \right) \times \left(\frac{\sum_{m=1}^{n} X_{j,m}}{N_w} \right) \tag{2.26}$$

If the solution only contains a single solute, the constraint on the number of water molecules, equation (2.8), becomes

$$N_w = \sum_{i=1}^{n} X_{A,i} \tag{2.27}$$

Using equation (2.26) and equation (2.27), the molality for a single solute solution, m_j^o, can be written as

$$m_j^o = \left(\frac{1 - a_w K_j}{M_w r_j a_w K_j}\right) \times \left(\begin{array}{c} \dfrac{1 - \sum_{i=1}^{n-1} \left({a_w}^i {K_j}^i \left(1 - C_{j,i}\right) \prod_{k=1}^{i-1} C_{j,k} \right)}{\left(1 - a_w K_j\right)^2 \sum_{m=1}^{n-2} \left(m {a_w}^{m-1} {K_j}^{m-1} \prod_{k=1}^{m} C_{j,k} \right)} \\ + \left((n-1) - (n-2) a_w K_j \right) {a_w}^{n-2} {K_j}^{n-2} \prod_{i=1}^{n-1} C_{j,i} \end{array} \right) \tag{2.28}$$

Combining the expression for the molality of solute j in a single solute solution, equation (2.28), with the expression for the molality of solute j in a mixture, equation (2.26), results in

$$\frac{m_j}{m_j^o} = \frac{\sum_{i=1}^{n} X_{j,i}}{N_w} \tag{2.29}$$

For a solution containing an arbitrary number of solutes, using equation (2.29) and equation (2.8), the molalities of the solutes in both pure (single solute) aqueous and mixtures at the same water activity have the following relationship

$$\sum_{j=A}^{Z} \frac{m_j}{m_j^o} = 1 \tag{2.30}$$

2.3 Solute Activity

Substituting the expression for water activity, equation (2.21), into the expression for the energy parameter of the first monolayer, equation (2.12), and rearranging equation (2.12) for the total number of j adsorption sites, $r_j N_j$, gives

$$r_j N_j = \frac{\left(X_{j,1} - X_{j,2}\right) + C_{j,1} a_w K_j X_{j,1}}{C_{j,1} a_w K_j} \tag{2.31}$$

Substituting the total number of j adsorption sites from equation (2.31) into the expression for solute activity in equation (2.22) results in

$$a_j = x_j^* \left(\frac{\left(X_{j,1} - X_{j,2}\right)}{\left(X_{j,1} - X_{j,2}\right) + C_{j,1} a_w K_j X_{j,1}} \right)^{r_j} \tag{2.32}$$

Using equation (2.23) to obtain expressions for $X_{j,1}$ and $X_{j,2}$ the equation for solute activity, equation (2.32), can be written as

$$a_j = x_j^* \left(\frac{1 - a_w K_j}{1 - \sum_{i=1}^{n-1} \left({a_w}^i {K_j}^i \left(1 - C_{j,i}\right) \prod_{k=1}^{i-1} C_{j,k} \right)} \right)^{r_j} \tag{2.33}$$

2.4 Long Range Interaction

To more accurately model the solute activity in dilute solutions, the mole fraction based Debye-Hückel equation for excess Gibbs energy from Pitzer[20–22,33] has been included in the Gibbs energy given by equation (2.19). The extra term helps to account for long-range interactions due to electrostatic screening of the ions in solution. This screening is more dominant at low solute concentrations compared to the close range interactions of hydrations shells. With the addition of this new term the equation for Gibbs free energy becomes

$$G/kT = \sum_{j=A}^{Z}\left[N_j\left(\ln\left(x_j^*\right) + r_j \ln\left(1 - \frac{X_{j,1}}{r_j N_j}\right)\right)\right]$$

$$+ N_w \ln\left(\frac{\sum_{j=A}^{Z} X_{j,n}}{\sum_{j=A}^{Z} K_j\left(X_{j,n-1} + X_{j,n}\right)}\right) \qquad (2.34)$$

$$- 2A_x\left(N_w + \sum_j v_j N_j\right)\frac{\sum_j v_j N_j \left|z_{j-}\right| z_{j+} Y_j}{\sum_j v_j N_j + N_w}$$

where A_x is the Debye-Hückel coefficient on a mole fraction basis, equal to 2.917 at 298.15K,[34] z_{j+} is the charge on the cation of solute j, z_{j-} is the charge on the anion of solute j, and v_j is the stoichiometric coefficient of solute j, equal to the number of ions into which one solute molecule dissociates. For non-electrolytes v_j is equal to unity, except in the specific cases discussed in chapter 4. The function Y_j is given by

$$Y_j = \frac{1}{\rho_j}\ln\left(\frac{1 + \rho_j {I_x}^{1/2}}{1 + \rho_j \left(I_{x,j}^{ref}\right)^{1/2}}\right) \qquad (2.35)$$

where I_x is the mole fraction ionic strength of the solution, and $I_{x,j}^{ref}$ is the ionic strength of a pure solution of solute j at the chosen reference state. The value of ρ_j can be related to the hard-core collision diameters of the solute ions, as shown in equation 34 of Pitzer and Simonson,[20] but it is usually treated empirically or set to a constant value for all solutes.

The ionic strength, in terms of the moles of solute present, and alternatively their molalities, is given by

$$I_x = \frac{1}{2}\frac{\sum_j v_j N_j \left|z_{j-}\right| z_{j+}}{\sum_j v_j N_j + N_w} = \frac{1}{2}\frac{\sum_j v_j m_j \left|z_{j-}\right| z_{j+}}{\sum_j v_j m_j + {M_w}^{-1}} \qquad (2.36)$$

where $j+$ and $j-$ refer to the cation and anion of the electrolyte, respectively, and the summations are over all solutes present in solution. Using a pure fused salt reference state, $I_{x,j}^{ref}$ can be written as

$$I_{x,j}^{ref} = \lim_{N_w \to 0} I_{x,j} = \frac{|z_{j-}|z_{j+}}{2} \tag{2.37}$$

Differentiating the Gibbs free energy, equation (2.34), with respect to the total water content of the system, N_w, and applying the partial derivatives of the energy equalities from equations (2.12), (2.13), (2.14), to each $\partial X_{j,i}/\partial N_w$ yields

$$a_w = K_w^{DH} \frac{\sum_{j=A}^{Z} X_{j,n_j}}{\sum_{j=A}^{Z} K_j \left(X_{j,n_j-1} + X_{j,n_j}\right)} \tag{2.38}$$

where the Debye-Hückel contribution K_w^{DH}, distinguished from the energy parameters K_j by the superscript "DH", is given by

$$K_w^{DH} = \exp\left(\frac{A_x {I_x}^{1/2} \sum_j \left(v_j N_j |z_{j-}|z_{j+}/\left(1+\rho_j {I_x}^{1/2}\right)\right)}{\sum_j v_j N_j + N_w}\right) \tag{2.39}$$

Using equation (2.16) the expression for water activity, equation (2.20), can be rewritten as

$$a_w = \frac{\bar{a}_w K_w^{DH}}{K_j} \tag{2.40}$$

where the sorption contribution, $\bar{a}_w$, is given by

$$\bar{a}_w = \frac{X_{j,n}}{X_{j,n-1} + X_{j,n}} \tag{2.41}$$

Similarly an expression for the solute activity can be calculated by taking the derivative of equation (2.34) with respect to the total solute content, N_j, yielding

$$a_j = \bar{a}_j K_j^{DH} \tag{2.42}$$

where the sorption contribution, $\bar{a}_j$, is given by

$$\bar{a}_j = x_j^* \left(\frac{r_j N_j - X_{j,1}}{r_j N_j} \right)^{r_j} \tag{2.43}$$

The Debye-Hückel contribution for electrolyte solutes, K_j^{DH}, is given by

$$K_j^{DH} = \left(\exp\left[-z_{j+}\left|z_{j-}\right| A_x \left\{ \frac{2}{\rho_j} \ln\left(\frac{1+\rho_j {I_x}^{1/2}}{1+\rho_j \left(I_{x,j}^{ref}\right)^{1/2}} \right) + \left(\frac{1-2I_x/\left|z_{j-}\right|z_{j+}}{\sum_k (v_k N_k) + N_w} \right) \sum_k \frac{N_k \left|z_{k-}\right| z_{k+} v_k}{2{I_x}^{1/2}\left(1+\rho_j {I_x}^{1/2}\right)} \right\} \right] \right)^{v_j} \tag{2.44}$$

Treating equation (2.43) the same way as equation (2.22) in section 2.3 results in the following equation for the sorption contribution

$$\bar{a}_j = x_j^* \left(\frac{1-\bar{a}_w}{1-\sum_{i=1}^{n_j-1}\left((\bar{a}_w)^i\left(1-C_{j,i}\right)\prod_{k=1}^{i-1} C_{j,k}\right)} \right)^{r_j} \tag{2.45}$$

For a single solute solution, the solute activity, $\bar{a}_j^o$, would be given by

$$\bar{a}_j^o = \left(\frac{1-\bar{a}_w}{1-\sum_{i=1}^{n_j-1}\left((\bar{a}_w)^i\left(1-C_{j,i}\right)\prod_{k=1}^{i-1} C_{j,k}\right)} \right)^{r_j} \tag{2.46}$$

Combining equation (2.42), (2.45) and (2.46) results in

$$a_j = x_j^* K_j^{DH} \bar{a}_j^o \tag{2.47}$$

2.5 Solute Concentration with Debye-Hückel Term for Pure Solutions

Following the same method as section 2.2, but applied to equation (2.41), the concentration of solutes in an aqueous solution with the consideration of the Debye-Hückel contribution can be derived. For a solution that contains only one solute j, it is found that the molality of solute j at the $\bar{a}_w$ of the binary water-solute j solution, is given by

$$m_j^o = \left(\frac{1-\bar{a}_w}{M_w r_j \bar{a}_w}\right) \times \left(\begin{array}{c} \dfrac{1-\sum_{i=1}^{n_j-1}\left((\bar{a}_w)^i\left(1-C_{j,i}\right)\prod_{k=1}^{i-1}C_{j,k}\right)}{(1-\bar{a}_w)^2\sum_{p=1}^{n_j-2}\left(p(\bar{a}_w)^{p-1}\prod_{k=1}^{p}C_{j,k}\right)} \\ +\left(\left(n_j-1\right)-\left(n_j-2\right)\bar{a}_w\right)(\bar{a}_w)^{n_j-2}\prod_{i=1}^{n_j-1}C_{j,i} \end{array} \right) \tag{2.48}$$

where $\bar{a}_w$ is given by equation (2.40).

2.6 Limit of Dilute Solutions

As a_w approaches unity, both m_j^o and $\bar{a}_j^o$ must approach zero. These limits applied, to equations (2.46) and equation (2.48), require that K_j is equal to unity when a_w is equal to unity. Therefore, the constant K_j is equal to unity in all cases. This is consistent with previous work,[26–28] where the K_j was treated as an adjustable parameter whose fitted value was found to be near unity.

In addition, when the water activity is equal to unity, the osmotic coefficient, ϕ, must also equal unity. The osmotic coefficient is related to the water activity and the molality of solute j by

$$\phi = \frac{-\ln a_w}{M_w \sum_j m_j v_j} \tag{2.49}$$

Substituting the expression for the molality of solute j, equation (2.48), into equation (2.49) and using L-Hôspital's rule to evaluate the limit results in r_j equal to v_j at a_w equal to unity. Assuming r_j, the number of sorption sites on solute j, is independent of concentration then r_j is equal to v_j for all concentrations and solute compositions.

2.7 Solution mixtures

Following the same method as section 2.2, but applied to equation (2.41), the concentration of solutes in an aqueous solution with the consideration of the Debye-Hückel contribution can be derived. In addition, the $r_j = v_j$ assumption can be included in the derivation of solute concentration. The molality of solute j in a pure aqueous solution at the $\bar{a}_w$ of the mixture, given by equation (2.40) with $K_j = 1$, is given by

$$\bar{m}_j^o = \left(\begin{array}{c} \left(\frac{1-\bar{a}_w}{M_w v_j \bar{a}_w}\right)\left(1 - \sum_{i=1}^{n_j-1}\left((\bar{a}_w)^i\left(1 - C_{j,i}\right)\prod_{k=1}^{i-1} C_{j,k}\right)\right) \\ \hline (1-\bar{a}_w)^2 \sum_{p=1}^{n_j-2}\left(p(\bar{a}_w)^{p-1}\prod_{k=1}^{p} C_{j,k}\right) \\ +\left(\left(n_j - 1\right) - \left(n_j - 2\right)\bar{a}_w\right)(\bar{a}_w)^{n_j-2}\prod_{i=1}^{n_j-1} C_{j,i} \end{array} \right) \tag{2.50}$$

For a mixture containing an arbitrary number of solutes the equivalent of equation (2.30) is

$$\sum_j \frac{m_j}{\bar{m}_j^o} = 1 \tag{2.51}$$

where the summation is over all solutes in solution.

2.8 Model Fitting Methods

The model derived above was written into MATLAB scripts for solving and parameterization. Since the expression for solute molality, equation (2.50), is itself a function of solute molality (from the K_w^{DH} term), an iterative method is needed to solve numerically. To numerically solve equation (2.50) the Newton-Raphson method was applied.

Given a function $f(x)$ and its derivative $f'(x)$, we begin with an initial guess, x_0, as a root of the function $f(x)$. A better approximation for the root of $f(x)$ is then given by $x_1 = x_0 - f(x_0)/f'(x_0)$. This process can be repeated as $x_{n+1} = x_n - f(x_n)/f'(x_n)$ until the estimate for the root is sufficiently accurate. Written in MATLAB, the Newton-Raphson method used to determine molality, *x*, appears as follows (lines 28-36 of mjSolve_matrix.m in appendix)

```
f = @(x) x – molality_function(x);
df = @(x) (f(x+dx)-f(x-dx))./(2*dx); % dx is step size, make small
x = initial_guess;
for i = 1:num_iteration
        x = x – f(x)./df(x);
end
```

The function "*molality_function*" solves for an arbitrary reference molality that results from rewriting equation (2.51) as

$$\left(\sum_j \frac{\alpha_j}{\bar{m}_j^o}\right)^{-1} = m_{ref} \tag{2.52}$$

where $\bar{m}_j^o$ is calculated using equation (2.50), and α_j gives the molar ratios of the solutes in solution such that $m_j = \alpha_j m_{ref}$. Equation (2.52) written in MATLAB as our "*molality_function*" appears in lines 19-26 of mjSolve_matrix.m in the Appendix.

For model parameterization the built in MATLAB non-liner solver, nlinfit, was used to optimize the parameter values. The number of sorption shells, *n*, being an integer, was not fit using nlinfit. Instead the value for *n* was incremented from 2 to 10 and a full fit was performed at each integer value of *n*. Here we have provided the mathematical framework as well as examples of the MATLAB code used in the numerical implementation of the isotherm model. Additional code can be found in the appendix of this document. The following chapters will cover the implementation, parameterization and results of the model.

Chapter 3

Isotherm-Based Thermodynamic Model for Electrolyte and Nonelectrolyte Solutions Incorporating Coulombic Electrostatic Interactions[1]

3.1 Introduction

Accurate calculation of gas/liquid/solid equilibrium is crucially important to environmental and industrial processes. In atmospheric chemistry, for example, the water uptake of aerosol particles as a function of temperature and relative humidity is central to new particle formation, the behavior of cloud condensation nuclei, cloud formation, visibility, air quality, and climate. Atmospheric aerosol particles, like most real-world chemical environments, are composed of highly complex chemical mixtures that confound accurate thermodynamic modeling, especially at low relative humidity (i.e., high solute concentration). Recently, by combining the advantages of adsorption isotherm models with traditional solution chemistry models, a unified thermodynamic treatment valid over the entire concentration range was derived and shown to work for a wide range of solutes.[26–28]

Models based on adsorption isotherms (which may loosely be regarded as a form of hydration model) have been used successfully to represent solvent activities of very

[1] Adapted with permission from Ohm, P. B.; Asato, C.; Wexler, A. S.; Dutcher, C. S. Isotherm-Based Thermodynamic Model for Electrolyte and Nonelectrolyte Solutions Incorporating Long- and Short-Range Electrostatic Interactions. *J. Phys. Chem. A* **2015**, *119* (13), 3244–3252.[5] Copyright 2015 American Chemical Society.

concentrated solutions and gels (Stokes and Robinson, 1948)[25], and activity coefficients and volumetric properties of concentrated salt-water systems that are liquid up to the pure salt melt (Abraham and Abraham, 1997, 2002; Ally and Braunstein, 1996).[35–37] The analogy is between the adsorption of gases on solids in systems of low vapor density and the sorption of solvent molecules on solutes in systems at low solvent activity. By allowing for multiple monolayer adsorption between solvent and solute, the validity of these isotherm-based models was extended to high water activity values, then extended further to infinite dilution by adding a "Debye-Hückel" term to represent long range electrostatic interactions.[28-3] The resulting model accurately predicts solute and solvent activities in both single solute (binary) solutions and multicomponent mixtures over the full concentration range. While employing no mixing parameters, the model reduces to the commonly used Zdanovskii-Stokes-Robinson (ZSR)[38,39] empirical mixing equation at high concentration but predicts mixture osmotic coefficients better at low concentrations.[39]

Ultimately, however, the need in chemical equilibrium modeling is for fully predictive models. In fields where the chemical composition is highly complex, such as in atmospheric aerosols that contain many electrolytes and organics, data are not available for many of the relevant compounds. Prior models required fit parameters for each dissolved species and often also ternary species-species interaction parameters leading to an intractable number of adjustable parameters in complex mixtures. For example, the widely used Pitzer model is known to have limitations due to the number of parameters necessary and the parameters' dependence on the accuracy of experimental data.[40] In prior work, a power law expression that related the energies to each other was employed to

substantially reduce the number of required parameters (Dutcher et al., 2012, 2013).[1,3] Here, we test that hypothesis that these energies are due to shorter-range dipole-dipole electrostatic forces at the molecular level, thereby reducing the number of fit parameters even further.

3.2 Theoretical Development

Consider an aqueous solution of solvent (water) activity a_w^o containing a single solute *j*, where *j* can be either an electrolyte or a non-electrolyte. Previously,[26–28] using statistical mechanics of lattice adsorption isotherms in combination with long range electrostatic interactions, equations for the solution molality, m_j^o, and solute activity, a_j^o, as functions of solvent activity, a_w^o,were derived and shown to be valid over the entire concentration range:

$$m_j^o = \frac{\left(\frac{1-\bar{a}_w^o}{M_w v_j \bar{a}_w^o}\right)\left(1-\sum_{i=1}^{n_j-1}\left((\bar{a}_w^o)^i\left(1-C_{j,i}\right)\prod_{k=1}^{i-1}C_{j,k}\right)\right)}{(1-\bar{a}_w^o)^2\sum_{p=1}^{n_j-2}\left(p(\bar{a}_w^o)^{p-1}\prod_{k=1}^{p}C_{j,k}\right)+\left(\left(n_j-1\right)-\left(n_j-2\right)\bar{a}_w^o\right)(\bar{a}_w^o)^{n_j-2}\prod_{k=1}^{n_j-1}C_{j,k}} \tag{3.1}$$

$$\bar{a}_j^o = \left((1-\bar{a}_w^o)\Big/\left(1-\sum_{i-1}^{n_j-1}\left((\bar{a}_w^o)^i\left(1-C_{j,i}\right)\prod_{k=1}^{i-1}C_{j,k}\right)\right)\right)^{v_j} \tag{3.2}$$

where M_w (kg mol^{-1}) is the molecular weight of the solvent (water), $_j$ is the stoichiometric coefficient of solute *j*, n_j is the number of sorption layers surrounding solute *j*, and superscript '*o*' indicates that the quantities are those of single solute (binary) aqueous solutions. The variables $\bar{a}_w^o$ or $\bar{a}_j^o$ are the solvent and solute activities normalized

by the long-range Debye-Hückel term, $K_w^{DH,o}$ and $K_j^{DH,o}$, respectively. For a non-electrolyte j, $K_w^{DH,o}$ and $K_j^{DH,o} = 1$. For an electrolyte j, $K_w^{DH,o}$ and $K_j^{DH,o}$ are given by:

$$K_w^{DH,o} = \exp\left(\frac{2A_x\left(I_x^o\right)^{3/2}}{1+\rho_j\left(I_x^o\right)^{1/2}}\right) \tag{3.3}$$

$$K_j^{DH,o} = \left(\exp\left[-z_{j+}|z_{j-}|A_x\left\{\left(I_x^o\right)^{\frac{1}{2}}\frac{\left(1-\frac{2I_x^o}{z_{j+}|z_{j-}|}\right)}{\left(1+\rho_j\left(I_x^o\right)^{1/2}\right)} + \frac{2}{\rho_j}\ln\left(\frac{1+\rho_j\left(I_x^o\right)^{1/2}}{1+\rho_j\left(I_{x,j}^{ref}\right)^{1/2}}\right)\right\}\right]\right)^{v_j} \tag{3.4}$$

where I_x^o is the ionic strength of the solution, $I_{x,j}^{ref}$ is the ionic strength of a pure aqueous solution of j at the chosen reference state (pure liquid solute), z_i is the charge on ion i (cation j_+, or anion j_-, of solute j), A_x is the Debye-Hückel coefficient on a mole fraction basis (equal to 2.917 at 298.15 K).[34] The parameter ρ_j has been related to the hard-core collision diameter of solute ions,[20] but in practice is treated either as a constant for all solutes,[21,22,34] or treated as an adjustable parameter.[28] Here, the parameter ρ_j is treated as an adjustable parameter that in this work will be related to properties of the ions in solution.

For multi-component systems, the binary expressions for solute molality and activity coefficients (equations (3.1) – (3.4)) can be used directly in the isotherm-based, zero-parameter mixing models developed previously (eqns. 25 and 26 of Dutcher et al. 2013).[28] The solute concentrations in mixtures conform to a modified Zdanovskii-Stokes-Robinson mixing rule, and solute activity coefficients to a modified McKay-Perring

relation, when the effects of the long-range terms (equations (3.3) and (3.4)) for each binary component are included. No additional mixing parameters are used, and the results for the mixtures show satisfactory accuracy over the entire concentration range. Ionic strength for a multi-component system is given by:

$$I_x = \frac{1}{2}\frac{\sum_j N_j |z_{j-}| z_{j+} v_j}{\sum_j v_j N_j + N_w} = \frac{1}{2}\frac{\sum_j m_j |z_{j-}| z_{j+} v_j}{\sum_j v_j m_j + {M_w}^{-1}} \quad (3.5)$$

where N_w is the number of water molecules and N_j is the number of molecules of solute *j*. Finally, the short-range sorption energy parameter, $C_{j,i}$, in equations (3.1) and (3.2), for the lattice sorption of a solvent molecule to a site in layer *i* of solute *j*, is given by:

$$C_{j,i} = \exp(\Delta\varepsilon_{j,i}/kT) \quad (3.6)$$

where *k* is Boltzmann's constant, $k = 1.38\times10^{-23}\ JK^{-1}$, *T* is temperature, and $\Delta\varepsilon_{j,i}$ is the energy change accompanying adsorption of solvent to monolayer *i* of solute *j*. The energy change is expressed as:

$$\Delta\varepsilon_{j,i} = E_{j,i} - E_L \quad (3.7)$$

where E_L is the energy of the free solvent in the bulk and $E_{j,i}$ is the energy of the bound solvent in layer *i*. Prior to the inclusion of the long-range electrostatics, $C_{j,i}$ was a fit parameter, with values that appeared independent of layer number (c.f., Table 5 in ref [28]). After the inclusion of long-range electrostatics, an empirical relationship between the $C_{j,i}$ values as a function of layer emerged, where the value in layers 2 to $n-1$ monotonically approached unity. An empirical model was then used: *n* and $C_{j,1}$ (the energy parameter for layer 1) were fit directly and $C_{j,i}$ values were obtained by fitting P_j in the equation: $C_{j,i} = (n_j/i)^{P_j}$ in layers 2 to $n-1$ (c.f., Table 4 in ref [28]).

Here, we estimate the values of $\Delta\varepsilon_{j,i}$ by assuming that electrostatics characterize solute-solvent and solvent-solvent interactions in solution. That is, all solute and solvent molecules are modeled as dipoles, or 'apparent' dipoles, even electrolytes. Solute ion parameters cannot simultaneously be adjusted to predict the aqueous electrolyte solution properties for all of the complementary pairs. In other words, the behavior of an ion in solution depends on the nature of its counterion. This is, in part, due to ion association effects such as contact ion pairing at high concentrations, and solvent-separated ion pairing at low and moderate concentrations, that have been observed experimentally[41,42,43] (e.g., see Figure 6 of ref [12] for a $H^+ - Cl^-$ water ternary cluster). Thus for electrolytes, extensive literature has established that activities in solution relate to electrolytes as the fundamental solute unit, not to ions.

The dipole-dipole interaction energy is:

$$E = \mu_1\mu_2 D^2/4\pi\varepsilon_0 r^3 \tag{3.8}$$

where ε_0 is the permittivity of space, and $4\pi\varepsilon_o = 1.113 \times 10^{-10}\ C^2N^{-1}m^{-2}$, D is a unit of conversion (Debye), $D = 3.33564 \times 10^{-30}\ Cm$, and μ_j is the dipole moment of species j. We assume that induced dipole moments are negligible.

Here we consider the effects of induced dipoles on the change in energy, $\Delta\varepsilon_{j,i}$, due to the sorption of a solvent molecule to a hydration layer, i, of a solute, j. An induced dipole moment, μ^*, is equal to the electric field, E, times the polarizability of a molecule, a ($\mu^* = Ea$). The parallel in this work is found when considering the presence of local non-neutrality (local electric field) near a point charge ($E = q/r$) or dipole (field = $\mu^* D/r^2$). The resultant induced dipole of molecule 2 is then described by either $\mu_2^* = q_1 C\alpha/r_{1,2}^2$ (point

charge, 1) or $\mu_2^* = \mu_1 C\alpha / r_{1,2}^3$ (dipole, 1). The Coulombic energy for a third induced dipole molecule, 3, near the point charge – dipole or dipole – dipole is given by $\mu_2^* \mu_3 D^2 / 4\pi\varepsilon_0 r_{2,3}^3$, yielding:

$$E = q_1 C\alpha\mu_3 D^2 / 4\pi\varepsilon_0 r_{1,2}^2 r_{2,3}^3 \text{ (point charge)} \tag{3.9}$$

$$E = \mu_1 C\alpha\mu_3 D^2 / 4\pi\varepsilon_0 r_{1,2}^3 r_{2,3}^3 \text{ (dipole)} \tag{3.10}$$

In the same manner, additional relationships can be found for all molecules present.

To calculate $\Delta\varepsilon_{j,i}$ we look at the energy difference between two states A and B, where A represents the presence of solute in the lattice while B represents the absence of solute. Consider an example of a non-electrolyte species with a solvent molecule, w_3, in a third monolayer. In state A, there are interactions between the solute molecule *j* and the solvent molecule w_3, as well as the interactions between w3 and the two intervening solvent molecules, w1 and w2 in monolayers 1 and 2, respectively. In State B, the solute is replaced with a solvent molecule w0 and all the interactions between it and the solvent molecules w1, w2 and w3 are as a result altered. Allowing for all absolute (no superscript) and induced ('*' superscript) dipole interactions and intermolecular differences, the Coulombic electrostatic difference of state B from state A is:

$$\begin{aligned}
&\Delta\varepsilon_{j,3} = \mu_j(\mu_w + \mu_{w3}^*)D^2/4\pi\varepsilon_0(r_{jw} + 2r_{ww} + r_{jw1}^* + r_{w1w2}^* + r_{w2w3}^*)^3 + \\
&(\mu_w + \mu_{w1}^*)(\mu_w + \mu_{w3}^*)D^2/4\pi\varepsilon_0(2r_{ww} + r_{w1w2}^* + r_{w2w3}^*)^3 + \\
&(\mu_w + \mu_{w2}^*)(\mu_w + \mu_{w3}^*)D^2/4\pi\varepsilon_0(r_{ww} + r_{w2w3}^*)^3 - \mu_w^2 D^2/4\pi\varepsilon_0(3r_{ww})^3 - \\
&\mu_w^2 D^2/4\pi\varepsilon_0(2r_{ww})^3 - \mu_w^2 D^2/4\pi\varepsilon_0(r_{ww})^3
\end{aligned} \tag{3.11}$$

where $D_w = 2.83 \times 10^{-10} m = D_{covalent} + L_{Hbond} = (0.6498 + 2.1711) \times 10^{-10} m$, $\mu_w = 2.9$,[44] $\mu_{w1}^* = \mu_j DC\alpha/(r_{jw} + r_{jw1}^*)^3$, and $\mu_{w2}^* = \mu_j DC\alpha/(r_{jw} + r_{jw1}^*)^3$.

Note that we allow each water molecule to have an absolute dipole and induced dipole due to the presence of the other dipoles, and the intermolecular distance to the first monolayer varies for each species, according to the size of the sorbent species and the strength of the intermolecular attraction. However, since $r_{xy} >> r_{xy}^*$ and $\mu_{xy} \gg \mu_{xy}^*$, to leading order, the induced moments can be neglected and the change of energy reduces to:

$$\Delta\varepsilon_{j,3} = \mu_j \mu_w D^2 / 4\pi\varepsilon_0 \left(r_{jw} - 2r_{ww}\right)^3 - \mu_w^2 D^2 / 4\pi\varepsilon_0 (3r_{ww})^3 \quad (3.12)$$

For an arbitrary layer, the leading order energy of sorption parameter in an isotherm model in layer *i* of solute *j*, as defined in Equation (3.6), is:

$$C_{j,i} = exp\left[\left(\frac{\mu_j \mu_w D^2}{4\pi\varepsilon_0 \left(r_{jw} + (i-1)r_{ww}\right)^3} - \frac{\mu_w \mu_w D^2}{4\pi\varepsilon_0 (i \cdot r_{ww})^3}\right)/(k_B T)\right] \quad (3.13)$$

3.2.1. Non-electrolytes. Consider a single solute solution, where the solute *j* and solvent *w* (i.e. water) are modeled as dipoles. Three kinds of dipole – dipole interactions are present: solute – solute, solute – solvent, and solvent – solvent. In what follows, we will neglect secondary effects such as induced dipole moments from multi-body effects, e.g., (solute with bound solvent) – (solvent) (see the Appendix for more discussion). For non-electrolytes, the unknowns in Equations (3.1)-(3.7) are n_j and $\Delta\varepsilon_{j,i} = E_{j,i} - E_L$, where $\Delta\varepsilon_{j,i}$ is the *change* of energy between two states A and B, where A represents the presence of solute in the lattice while B represents the absence of solute. In other words, A represents solvent sorption to *solute* while B represents sorption of solvent to *solvent*. These solvent-

solute and solvent-solvent interactions depend on the dipole of each species and the distance between these dipoles which is related to the size of the solute and solvent molecules. Thus Equations (3.7) and (3.8) can be combined to obtain an expression for the change in energy between states A and B:

$$\Delta\varepsilon_{j,i} = \frac{\mu_j \mu_w D^2}{4\pi\varepsilon_0 \left(r_{jw} + (i-1)r_{ww}\right)^3} - \frac{\mu_w \mu_w D^2}{4\pi\varepsilon_0 (i \cdot r_{ww})^3} \tag{3.14}$$

where $r_{ww} = 2.82 \times 10^{-10} m$ and $\mu_w = 2.9$.[44] Thus, for non-electrolytes, the remaining unknowns to the model (equations (3.1) – (3.4), (3.14)) are the number of layers, n_j, the solute dipole moment, μ_j, and the intermolecular distance between solute *j* and solvent *w*, r_{jw}. Further parameter reduction is possible by relating μ_j to the interspatial distance, discussed later.

3.2.2. Electrolytes. In the isotherm model in Dutcher et al. 2013, it was shown that the leading contributions to the thermodynamic model predictions could be broken down into a Debye-Hückel contribution that dominates at low concentrations, and an adsorption contribution that dominates at moderate to high concentrations (cf. part c's in Figs 2-7 in Dutcher et al. 2013).[28] Here, we assume that the screening of individual ions is accounted for by the Debye-Hückel term and that the sorbing species in the isotherm portion of the model is a solvent-separated ion pair state, a ternary cation-water-anion cluster. So, a 1:1 electrolyte *j* is cast as a single compound with an apparent dipole moment, μ_j, calculated by:

$$\mu_j = qe\left((r_{j+} + r_{j-})/2 + r_{jj}\right) \tag{3.15}$$

where q is the charge, which in the case of 1:1 electrolytes has a value of 1, e is the elementary charge, $e = 1.60218 \times 10^{-19} C$, r_{j+} and r_{j-} are the effective radii of the cation

and anion, respectively, and r_{jj} is an interspatial distance that includes the additional distance due to the solvent separation in the adsorbent structure. We assume that this interspatial distance is related to the intermolecular distance r_{jw} through the following relationship:

$$r_{jw} = r_{jj} + \frac{r_{j+} + r_{j-}}{2} + \frac{r_{ww}}{2} \tag{3.16}$$

With these assumptions, effective dipoles can be calculated for electrolytes (Equation (3.15)), and Equations (3.15) and (3.16) yield a relationship for the energy change that is identical to Equation (3.14) for organics. Thus, for electrolytes, the remaining unknowns in the model (Equations (3.1) – (3.4)), (3.14)) are the number of layers, n_{j}, the intermolecular distance, r_{jw}, and the Debye - Hückel parameter, ρ. Further parameter reduction is possible by relating ρ to the interspatial distance, discussed below.

3.3 Model Applications and Parameterization

Employing equations (2.1)-(2.4) and (3.14) above, the initial model parameters are n_{j} , r_{jw}, and μ_j for non-electrolytes and n_{j}, r_{jw}, and ρ for electrolytes. These sets will be reduced further in later sections of this work. These parameter values were fit using MATLAB's built in nonlinear regression model, nlinfit, and molality activity data from the literature. For robust fitting, the nlinfit function uses an iterative reweighting least squares algorithm.[45,46] At each iteration, the robust weights are recalculated and outlying points are down weighted. Since robust weighting was used (except when noted in Tables 1-3), no weights were explicitly assigned. The maximum number of nlinfit iterations used was set to 1000 and the termination tolerance on the residual sum of squares as well as on the

estimated coefficients was set to $1x10^{-300}$ which is on the order of the smallest non-zero number for MATLAB. A script looped through sorption layers ranging from n_j=1 to 10. Due to the implicit nature of Equation (3.1), a minimization algorithm was run to obtain molalities. The 'goodness of fit' is calculated using a normalized mean square error, defined by $\frac{1}{n_p}\sum_{i=1}^{n_p}\left(\frac{m_{model,i}-m_{data,i}}{m_{model,i}}\right)^2$, where n_p is the number of data points. Tables 1 through 3 provide the data sources, number of data points, and concentration ranges for each solute.

3.3.1 Non-electrolytes. Model predictions for osmotic coefficients of glycerol, ethanol, methanol, sucrose, and 1,4-butanediol are given in Figure 3.1. Figure 3.1a shows the model predictions from fitting both μ_j and r_{jw} simultaneously; Table 1 shows the corresponding parameter values. Figure 3.1b shows the relationship between these two parameters, which is well described by the cubic polynomial given in Equation (3.17), where r_{jw} is given in Angstroms. The fit for Equation (3.17) has an R-squared value of 0.9745. Figure 3.1c shows the model predictions from fitting r_{jw} and using Equation (3.17) to estimate μ_j; Table 2 provides the corresponding fit parameter values.

$$\mu_j = (r_{jw}/2.02\text{Å})^3 \tag{3.17}$$

3.3.2 Electrolytes. Model predictions for osmotic coefficients for families of chlorides, nitrates and sodium electrolytes are given in Figure 3.2, 3 and 4, respectively.

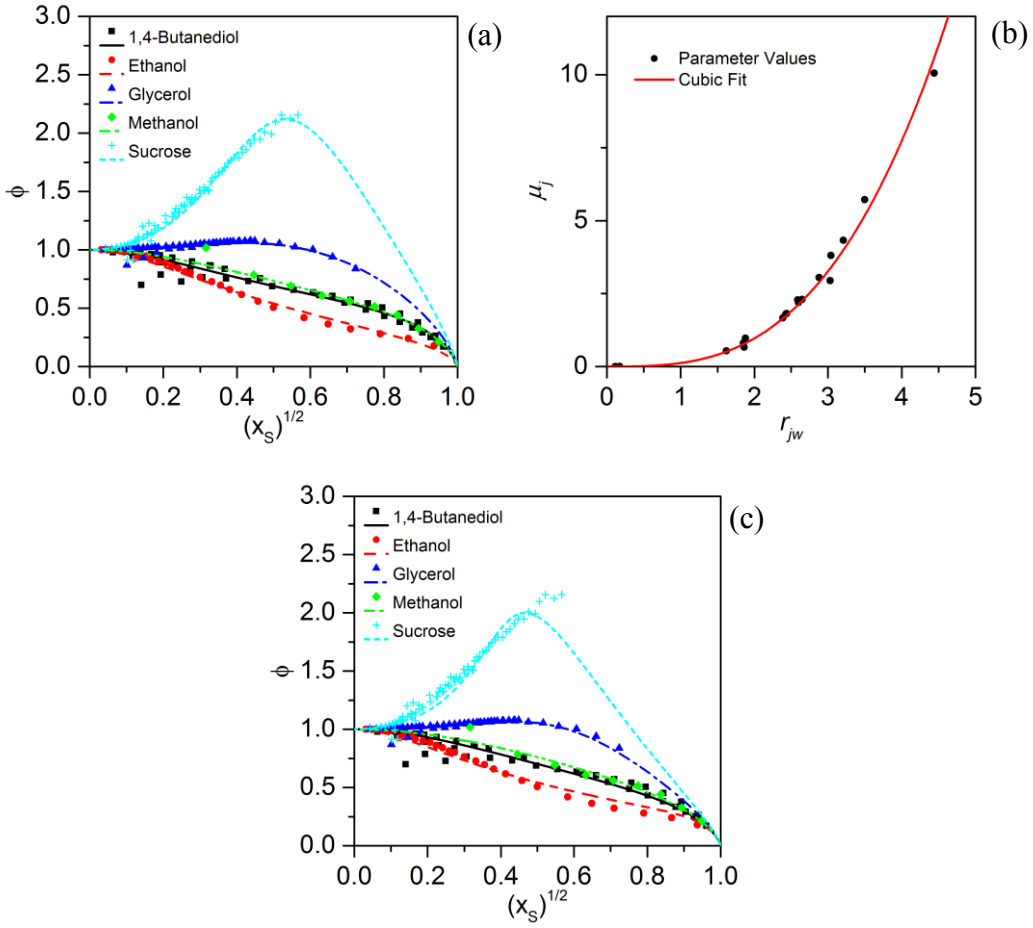

Figure 3.1. Coulombic Model Fit for Organic Solutions. (a) Measured and calculated osmotic coefficients, ϕ, of aqueous organic solutes plotted versus the square root of the solute mole fraction. Solute mole fraction is defined as $x_s=m/(m+1/M_w)$, where m is molality of the organic solute and M_w is the molar mass of water in kg/mol. (a) Solid lines show current model with both μ_j and r_{jw} fit as parameters. Symbols: blue triangles, Glycerol, Scatchard et al.,[47] Ninni et al.;[48] red circles, Ethanol, Strey et al.;[49] Green diamonds, Methanol, Zhu et al.;[50] light blue crosses, Sucrose, Scatchard et al.,[47] Bubnik et al. (data theifed from Baeza et al. 2010);[51] Black squares, 1,4-Butanediol, Marcolli and Peter,[52] Paez et al.;[53] (b) shows the relation between the r_{jw} and μ_j fit parametes for all organic species fit in this paper (Equation (3.17)). (c) Current model with μ_j constrained to the relation shown in (b).

The model predictions result from fitting ρ and r_{jw} simultaneously, and using Equations (3.15) and (3.16) to calculate μ_j. The parameter ρ is from the Debye-Hückel treatment of the aqueous solvent and nominally relates to the hard core collision radius of the solute. The parameter ρ first appears in Pitzer's modification of the Debye-Hückel

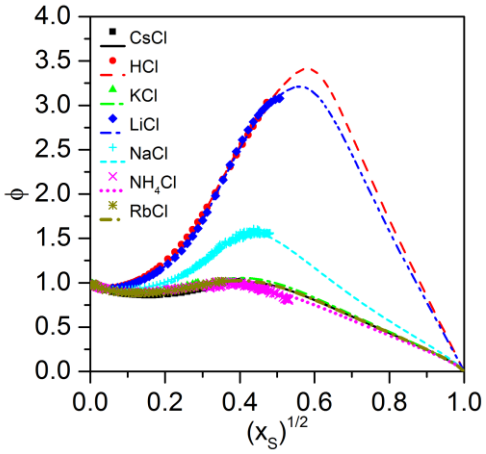

Figure 3.2. Coulcombic Model Fit for Chloride Solutions. Measured and calculated osmotic coefficients, ϕ, for aqueous chloride solutions plotted versus the square root of the solute mole fraction. Solute mole fraction is defined as $x_s=m/(m+1/M_w)$, where m is molality of the electrolyte and M_w is the molar mass of water in kg/mol. Solid lines show the current model. Symbols: Red circles, HCl, Guendouzi et al. and Hamer and Wu; [54,55] Blue diamonds, LiCl, Guendouzi et al. and Hamer and Wu;[54,55] Light blue crosses, NaCl, Archer, Tang et al., Cohen et al., Chan et al.;[6–9] Green triangles, KCl, Hamer and Wu;[55] Gold '*', RbCl, Hamer and Wu;[55] Black squares, CsCl, Hamer and Wu;[55] Magenta 'x', NH_4Cl, Guendouzi, Hamer and Wu;[54,55]

model as a fit parameter,[56,57] which was later related to the distance of closest approach, a [20] or the Stokes radii [58,59] of the corresponding anion and cation. However, due to the lack of an accurate model for a, in practice, ρ is treated empirically, as it is here. As with non-electrolyte solutes, the parameter r_{jw} represents the average center to center distance of a solute j and a nearest neighboring water molecule. Table 3 shows the fit parameters ρ and r_{jw} for all electrolyte species.

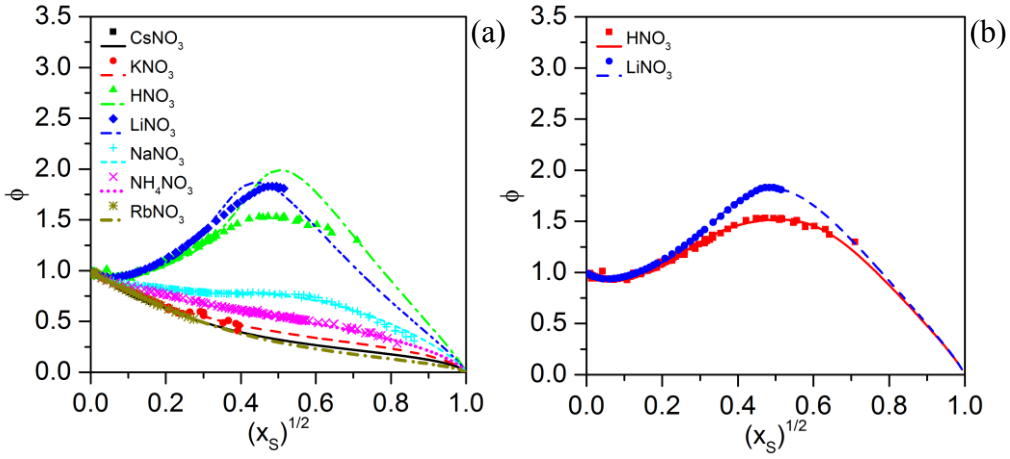

Figure 3.3. Coulombic Model Fit for Nitrate Solutions. (a) Measured and calculated osmotic coefficients, ϕ, of aqueous nitrate solutions plotted versus the square root of the solute mole fraction. Solute mole fraction is defined as in Figure 3.2. Solid lines show the current model. Symbols: Green triangles, HNO_3 , Hamer and Wu;[55] Blue diamonds, $LiNO_3$, Hamer and Wu;[55] Light blue crosses, $NaNO_3$, Data sources in Clegg et al.(1997);[60] Red circles, KNO_3 , Hamer and Wu, Kelly et al.;[55,61] Gold '*', $RbNO_3$, Hamer and Wu;[55] Black squares, $CsNO_3$, Hamer and Wu;[55] Magenta 'x', NH_4NO_3 , Wishaw and Stokes, Chan et al. Kirginstev and Lukyanov;[62–64] (b) Species for which the power law fit is capable of providing a much better fit than the Coulombic fit.

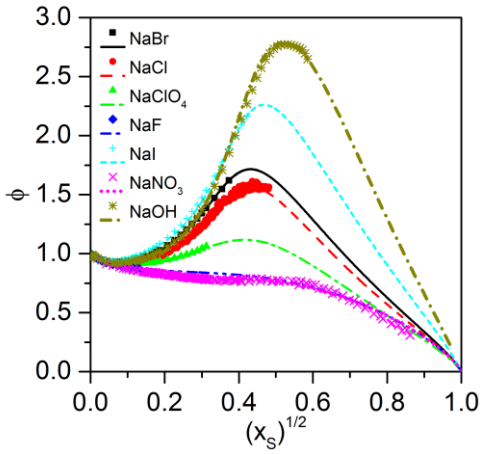

Figure 3.4. Coulombic Model Fit for Sodium (Na^+) Solutions. Measured and calculated osmotic coefficients , ϕ, of aqueous sodium electrolyte solutions plotted versus the square root of the solute mole fraction. Solute mole fraction is defined as in Figure 3.2. Solid line shows the current model. Symbols: Blue diamonds, NaF, Hamer and Wu;[55] Gold '*', NaOH, calculated values using the equation of Hamer and Wu;[55] Magenta 'x', $NaNO_3$, Clegg et al.(1997);[60] Red circles, NaCl, Archer, Tang et al., Cohen et al., Chan et al.;[6–9] Black squares, NaBr, Hamer and Wu;[55] Light blue crosses, NaI, Hamer and Wu;[55] Green triangles, $NaClO_4$, Hamer and Wu;[55]

There is excellent agreement between the model predictions and the available data for all solutes except $LiNO_3$ and HNO_3. The model is able to capture the low concentration characteristics of HNO_3 and $LiNO_3$ but is unable to predict the behavior at higher concentrations. However, $LiNO_3$ and HNO_3 can both be accurately modeled using the power law fit for the C parameters employed in prior work,[28] as shown in Figure 3.3b.

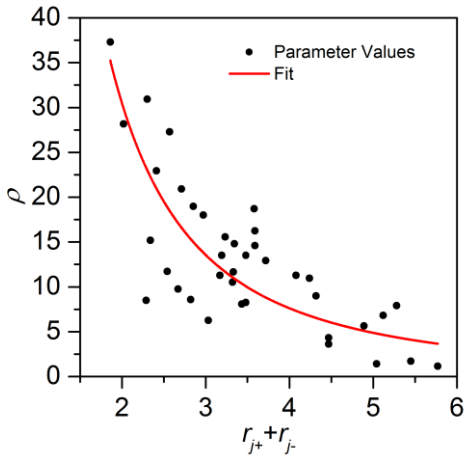

Figure 3.5. Electrolyte Fit Parameter Relationship. Fit parameter ρ plotted against the sum of the anion and cation radii for the electrolytes listed in Table 3. Solid line is Equation (3.18).

The poor fit for HNO_3 and $LiNO_3$ may be due to the planar structure of the nitrate ion, which may allow for a closer approach for small cations than an equivalently sized elemental anion. Another source of the anomalous nature may be that electrolyte solutions containing lithium are known to form clusters,[65] which are not being adequately captured by the current model approach.

The two fit parameters used here for the electrolyte solutions are correlated, though to a lesser extent than observed with the non-electrolyte solutions. Figure 3.5 Figure 3.5shows a plot of ρ versus $r_{j+} + r_{j-}$, which follows the relationship:

$$\rho = [11.04\text{Å}/(r_{j+} + r_{j-})]^2. \quad (3.18)$$

Although the fit is not nearly as good as for non-electrolytes (see Figure 1b), the Debye-Hückel term only plays a role for dilute solutions so errors in its value do not substantially alter the overall fit to the data. For example, for NH_4NO_3 solutions, the MSE is 8.0E-3 and 2.7E-4 for the ρ constrained and unconstrained fits, respectively.

3.4 Summary

In prior work, we used statistical mechanics to develop a solution thermodynamic model valid over the full range of concentrations. That model required at least three parameters for each solute. In this work we used a model of electrostatic interactions between solute and solvent molecules in solution to reduce the number of parameters required in the model and to relate the value of these parameters to properties of the solute and solvent molecules. The improved treatment and reduction of the parameters in the isotherm model yield accurate predictions of activity and osmotic coefficients for a wide range of electrolytes and non-electrolytes in water, including thirty-seven binary electrolyte systems, over a range of cation sizes from H^+ to tetrabutylammonium, for all of the common anions including Cl^-, Br^-, I^-, NO_3^-, OH^-, ClO_4^- and twenty water soluble non-electrolytes of mostly alcohols and polyols systems. The parameterization itself can also be used to estimate unknown physicochemical properties, such as intermolecular spacing, although we emphasize that caution should be used to not over-interpret the

parameterization. For example, future work is needed to better understand the role or physical understanding of the '*n*' parameter, as discussed in Chapter 5.

To conclude, this work represents a significant advancement towards a fully predictive quantitative structure property relationship for activity coefficient modeling. For the non-electrolytes, the physical properties necessary to calculate the energies of adsorption are the dipole moments of the solvent and the solute and the intermolecular solute-solvent and solvent-solvent bond lengths. For electrolyte solutions, the ion charge types, ion-solvent bond lengths and solvent-solvent bond lengths are the parameters necessary for calculating the energy of adsorption. The majority of these physical properties, with the exception of intermolecular distances for many solutes, are available in the literature. For the remaining properties, predictive correlations developed here for non-electrolytes (equation (3.17)) and electrolytes (equation (3.18)) allow for a remarkable reduction in the number of adjustable parameters. In fact, for systems where estimates for intermolecular spacing for a solute and solvent molecular are known, the model is fully predictive. For solute concentration and activity predictions of multi-component systems, the reduced parameter binary model predictions found here can be applied directly to the zero-parameter mixing model developed previously.[28]

Table 3-1. The Two Parameter Fit for Aqueous Organics at 298.15 K[a]

Species	*n*	μ_j	r_{jw} [Å]	MSE[b]	Min a_w	Min *m* *[kg mol^{-1}]*	Max *m* *[kg mol^{-1}]*	n_p	Ref.
1,2-Butanediol[c]	8	1.669	2.39	1.14E-02	1.85E-02	0.230	6.79E+02	52	52,53
1,3-Butanediol[c]	9	2.304	2.65	3.64E-03	9.80E-03	0.229	6.79E+02	48	52,53,66
1,4-Butanediol[c]	8	1.798	2.43	4.86E-03	1.54E-02	0.117	6.79E+02	54	52,53,66
2,3-Butanediol[c]	3	1.818	2.44	3.62E-04	1.28E-02	0.128	5.87E+02	42	52,53
1,2,3,4-Butanetetrol	9	442.8	22.57	2.15E-05	9.60E-01	0.051	1.11E+00	19	66,67
1,2,4-Butanetriol	3	5.716	3.50	3.04E-03	1.43E-02	0.753	3.11E+02	22	52,66
Ethanediol	3	4.324	3.21	2.52E-03	2.30E-03	0.119	9.41E+02	23	52,67
Ethanol	2	2.939	3.03	3.45E-03	7.76E-02	0.056	3.97E+02	31	49
Glycerol	12	3.804	3.04	1.35E-04	3.11E-03	0.185	5.39E+02	106	47,48,52,68
IPA	2	0.6580	1.86	1.65E-01	4.80E-02	0.431	8.81E+02	15	50
Methanol	4	2.192	2.60	3.95E-03	2.18E-02	0.100	4.97E+02	10	50
1,2-Pentanediol	3	5.5E-4[d]	0.17	3.87E-01	3.96E-02	0.834	3.79E+02	14	52
1,4-Pentanediol	11	0.7989	1.85	5.70E-03	4.16E-02	1.058	3.07E+02	14	52
1,5-Pentanediol	4	2.0E-4[d]	0.12	6.06E-03	1.88E-02	1.225	5.28E+02	14	52
2,4-Pentanediol	7	0.5386	1.62	4.29E-03	3.57E-02	1.035	3.56E+02	14	52
1,2-Propanediol	6	3.044	2.88	1.20E-03	6.89E-03	2.482	6.42E+02	13	52
1,3-Propanediol	5	2.278	2.59	2.99E-03	5.18E-03	2.561	6.98E+02	13	52
Sorbitol	9	10.05	4.44	5.26E-03	2.49E-03	0.200	1.89E+03	133	48,69,70
Sucrose	15	14.09	4.58	1.38E-03	1.30E-01	0.100	2.62E+01	52	47,51
Urea	10	0.9652	1.88	2.76E-06	5.68E-01	0.100	2.00E+01	41	47

[a]The values for r_{jw} and μ_j were fit. The number of sorption layers *n* was also fit, but separately from r_{jw} and μ_j. [b]MSE is a normalized mean-square error, equal to $\frac{1}{n_p}\sum_{i=1}^{n_p}\left(\frac{m_{model,i}-m_{data,i}}{m_{model,i}}\right)^2$, where n_p is the number of data points. [c]Fit with non-robust weighting (e.g., unweighted fit). [d]Unrealistic value.

Table 3-2. The Single Parameter Fit for Aqueous Organics at 298.15 K[a]

Species	n	μ_j [b]	r_{jw} [Å]	MSE[c]	Reference
1,2-Butanediol[d]	8	1.912	2.51	1.15E-02	52,53
1,3-Butanediol[d]	6	2.807	2.85	4.46E-03	52,53,66
1,4-Butanediol[d]	8	2.566	2.77	6.70E-03	52,53,66
2,3-Butanediol[d]	3	3.162	2.97	1.47E-03	52,53
1,2,3,4-Butanetetrol	12	5.942	3.66	2.39E-05	66,67
1,2,4-Butanetriol	3	16.90	5.19	6.45E-03	52,66
Ethanediol	3	12.00	4.63	7.52E-03	52,67
Ethanol	3	0.2653	1.30	5.80E-03	49
Glycerol	3	15.28	5.02	2.38E-03	47,48,52,68
IPA	3	0.0140	0.49	2.00E-01	50
Methanol	3	3.462	3.06	3.83E-03	50
1,2-Pentanediol	10	4.69E-13[e]	1.57E-04[e]	4.68E-02	52
1,4-Pentanediol	8	1.427	2.28	1.36E-02	52
1,5-Pentanediol	10	0.7127	1.81	2.17E-02	52
2,4-Pentanediol	12	1.215	2.16	8.67E-03	52
1,2-Propanediol	3	7.082	3.88	5.23E-03	52
1,3-Propanediol	3	6.706	3.81	1.40E-02	52
Sorbitol	12	7.260	3.92	7.89E-03	48,69,70
Sucrose	7	26.70	6.05	4.19E-03	47,51
Urea	3	3.046	2.93	8.24E-03	47

[a]The number of sorption shell layers *n* was also fit, but separately from the intermolecular distance, r_{jw}. [b]The dipole moment was calculated using equation (3.17) and was not a fit parameter. [c]MSE is a normalized mean-square error, equal to $\frac{1}{n_p}\sum_{i=1}^{n_p}\left(\frac{m_{model,i}-m_{data,i}}{m_{model,i}}\right)^2$, where n_p is the number of data points. [d]Fit with non-robust weighting. [e]Unrealistic value

Table 3-3. The Two Parameter Fit for Aqueous 1:1 Electrolyte Solutions at 298.15 K[a]

Species	n	ρ	r_{jj}[b] [Å]	cation r [Å][c]	anion r [Å][c]	MSE[d]	Min a_w	Max *m* *[kg mol^{-1}]*	n_p	Ref
HNO_3	3	37.287	2.27	0.21	1.65	1.57E-02	0.072	5.65E+01	150	[55]
HCl	7	28.172	1.47	0.21	1.81	2.63E-04	0.174	1.60E+01	39	[54,55]
HBr	13	22.852	2.22	0.21	1.96	2.13E-04	0.639	6.00E+00	40	[71,72]
LiOH	3	8.4789	2.76	0.76	1.53	3.41E-05	0.879	4.00E+00	25	[55]
$LiNO_3$[e]	4	22.925	2.28	0.76	1.65	2.40E-03	0.271	2.00E+01	43	[55]
LiCl	6	27.275	1.25	0.76	1.81	3.95E-04	0.119	1.92E+01	43	[54,55]
NaF	2	15.188	2.46	1.01	1.33	3.56E-07	0.969	1.00E+00	17	[55]
NaOH	4	11.706	1.50	1.01	1.53	1.00E-03	0.063	2.90E+01	52	[55]
$NaNO_3$	2	9.7517	2.26	1.01	1.65	4.93E-04	0.152	1.60E+02	162	[60]
NaCl	4	8.5709	2.24	1.01	1.81	5.63E-04	0.391	1.67E+01	179	[6–9]
NaBr	4	18.011	2.11	1.01	1.96	7.38E-06	0.592	9.00E+00	32	[55]
NaI	4	30.924	1.59	1.01	2.20	2.73E-04	0.392	1.20E+01	35	[55]
$NaClO_4$[e]	3	15.566	2.24	1.01	2.22	1.29E-05	0.795	6.00E+00	29	[55]
KF[e]	3	20.921	1.82	1.38	1.33	4.03E-04	0.292	1.75E+01	41	[55]
KNO_3	2	6.2577	2.55	1.38	1.65	3.78E-04	0.843	1.03E+01	37	[55,61]
KCl	3	13.497	2.32	1.38	1.81	9.67E-06	0.836	5.00E+00	28	[55]
KBr	3	14.802	2.21	1.38	1.96	3.59E-06	0.816	5.50E+00	28	[55]
KI	3	18.712	2.05	1.38	2.20	2.53E-06	0.846	4.50E+00	26	[55]
RbF	6	18.959	3.25	1.52	1.33	9.37E-06	0.873	3.50E+00	24	[55]
$RbNO_3$	4	11.265	3.74	1.52	1.65	1.17E-06	0.919	4.50E+00	26	[55]
RbCl	3	11.654	2.26	1.52	1.81	1.25E-05	0.752	7.80E+00	32	[55]
RbBr	2	13.501	1.36	1.52	1.96	1.41E-04	0.845	5.00E+00	27	[55]
RbI	2	12.926	1.05	1.52	2.20	2.51E-07	0.844	5.00E+00	27	[55]
$CsNO_3$	2	10.516	2.71	1.67	1.65	2.46E-06	0.963	1.50E+00	19	[55]
CsCl	3	8.2649	2.15	1.67	1.81	4.97E-05	0.668	1.10E+01	34	[55]
NH_4NO_3	2	8.9672	2.16	1.78	1.65	2.70E-04	0.320	1.11E+02	94	[62–64]
NH_4Cl	3	14.586	2.23	1.78	1.81	2.21E-04	0.527	2.18E+01	132	[54]
$DMANO_3$	2	11.292	1.66	2.42	1.65	9.60E-06	0.845	6.00E+00	23	[73]
DMACl	3	10.967	1.71	2.42	1.81	8.84E-05	0.503	1.70E+01	58	[74]
$(CH_3)_4NBr$	3	4.3046	1.62	2.51	1.96	1.86E-05	0.846	5.50E+00	36	[75,76]
$(CH_3)_4NCl$	3	8.9861	1.29	2.51	1.81	9.26E-04	0.331	1.90E+01	36	[75]
$(C_2H_5)_4NCl$	5	5.6277	0.94	3.08	1.81	2.87E-05	0.555	9.00E+00	26	[75]
$(C_2H_5)_4NBr$	4	1.3976	0.96	3.08	1.96	1.80E-03	0.552	1.20E+01	29	[75]
$(C_2H_5)_4NI$	2	7.8866	2.17	3.08	2.20	2.32E-04	0.966	1.90E+00	19	[75]
$TMANO_3$	2	6.8027	0.91	3.47	1.65	2.21E-05	0.784	8.50E+00	28	[73]
$(C_3H_7)_4NBr$	5	1.7025	1.26	3.49	1.96	7.54E-04	0.707	9.00E+00	26	[75]
$(C_4H_9)_4NBr$	2	1.1356	0.45	3.81	1.96	6.23E-04	0.520	2.70E+01	54	[75,76]

[a]The values for r_{jj} and ρ were fit. The number of sorption layers *n* was also fit, but separately from r_{jj} and ρ. [b]r_{jj} is related to r_{jw} through Eqn. 11. [c]Most ionic radii are from Shannon;[77] $(CH_3)_4N^+$, $(C_2H_5)_4N^+$, $(C_3H_7)_4N^+$, and $(C_4H_9)_4N^+$ were from Masterton et al.;[78] OH^-, NO_3^-, and NH_4^+ were from Marcus.[79] [d]MSE is the mean-square error, equal to $\frac{1}{n_p}\sum_{i=1}^{n_p}\left(\frac{m_{model,i}-m_{data,i}}{m_{model,i}}\right)^2$, where n_p is the number of data points. [e]Fit with non-robust weighting (e.g., unweighted fit).

Chapter 4

Treatment of Organic Acids with Consideration of Partial Disassociation

4.1 Introduction.

Organic acids make up an important fraction of atmospheric aerosol composition. In the atmosphere, the total amount of organic carbon has been estimated to account for anywhere between 10-65% of aerosol mass.[80] Dicarboxylic acids represent a major component of the total organic carbon mass in the atmosphere. Dicarboxylic acids and other organic acids are integral to many of the processes of atmospheric aerosol formation and growth as well as the properties expressed by the aerosols. Dicarboxylic acids in particular are believed to play an important role as cloud condensation nuclei (CCN) in the atmosphere, potentially being more important than sulfate, a main component of CCN, in the initial growth of very small particles. Other studies have shown that dicarboxylic acids can reduce the surface tension and hygroscopic property of CCN, affecting cloud formation and optical properties.[81]

The previous chapter detailed the introduction of a Coulombic interaction for determining the energy parameters for the sorption of water onto a solute molecule. However, the Coulombic model for organics is unable to accurately predict the properties of some organic acids such as malonic or glutaric acid, thus additional assumptions need to be considered. It is hypothesized that the model for organics is unable to reproduce the effects caused by partial dissociation in solutions of weak acids when treated as a neutral,

non-disassociating molecular species. Modeling the organic acid as a dissociated electrolyte is one method for accounting for this dissociation but is a non-physical representation of a weak organic acid in solution, which are known to not fully dissociate. An alternative solution to this problem is to model the organic acids as a mixture of non-dissociated organic species (HA) and dissociated organic species ($H^+ + A^-$).

This chapter discusses different methods for treating the energy C parameters for organic acids: acetic acid, butyric acid, citric acid, glutaric acid, malic acid, malonic acid, and succinic acid. In section 4.2 the organic is treated as a neutral species, identical to the treatment of the non-electrolytes in chapter 3. Section 4.3 models the organic as a partially disassociating species. As derived in Chapter 2, the C parameters are related to the energy change from the sorption of a water molecule from the bulk water onto the respective hydration shell

$$C_{j,i} = \exp\left(\frac{\varepsilon_{j,i}}{kT}\right) \tag{4.1}$$

where $\varepsilon_{j,i}$ is the difference in energy between a sorbed water and a free water molecule in the bulk.

4.2 Initial Organic Acid Fitting: Purely Empirical Parameterization, Power Law and Coulombic Relations for Organic Acids as Neutral Species

A model for predicting activity coefficients of aqueous solutions was given in Chapters 2 and 3. The isotherm based model was originally implemented using two different approaches to the determination of the energy C parameters.[26–28] The first approach was to treat each C value as a separate fit parameter, fitting all of the C values

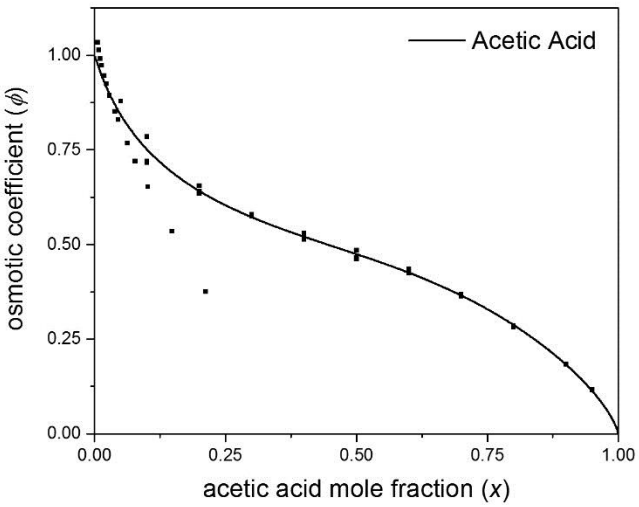

Figure 4.1. Acetic Acid All C Parameter Fit. Osmotic coefficient versus solute mole fraction for acetic acid. The model prediction treats each individual energy C parameter as an adjustable fit parameter (C_1=0.272865, C_2=2.95093, C_3=0.367909, mse=0.0061). Data from Hansen et al., Pirouzi et al., Sebastiani and Lacquaniti.[82–84]

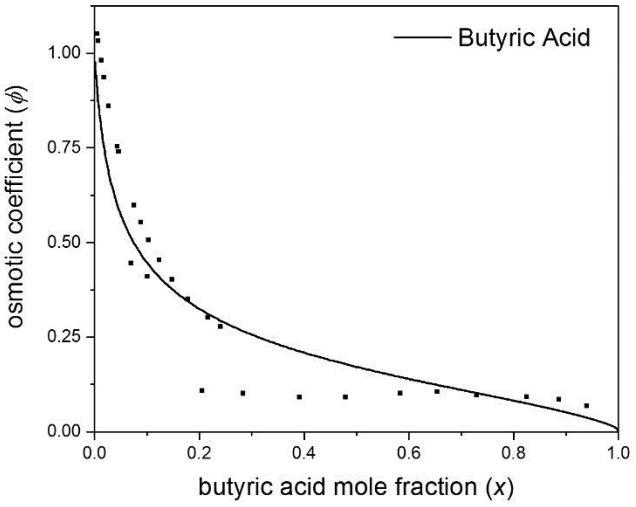

Figure 4.2. Butyric Acid All C Parameter Fit. Osmotic coefficient versus solute mole fraction for butyric acid. The model prediction treats each individual energy C parameter as an adjustable fit parameter (C_1=0.037642, C_2=0.000311, C_3=0.509333, C_4=6277.3, mse=0.1141). Data from Hansen et al., Pirouzi et al.[82,83]

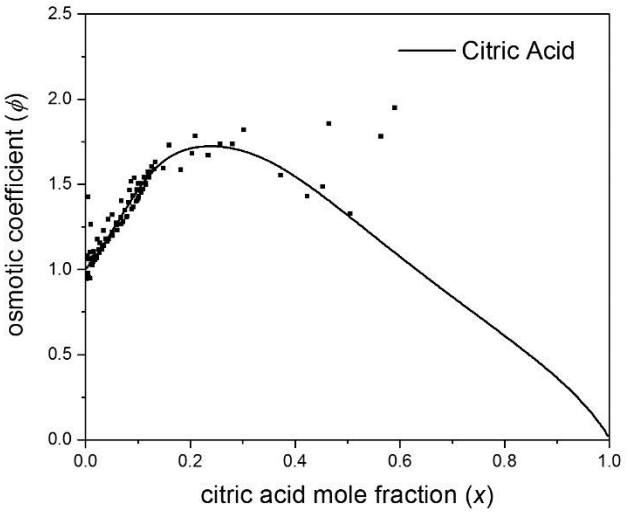

Figure 4.3. Citric Acid All C Parameter Fit. Osmotic coefficient versus solute mole fraction for citric acid. The model prediction treats each individual energy C parameter as an adjustable fit parameter (C_1=3.19653, C_2=0.005227, C_3=1355.52, C_4=0.237212, C_5=0.007644, C_6=469.941, mse=0.0255). Data from Peng et al.[85]

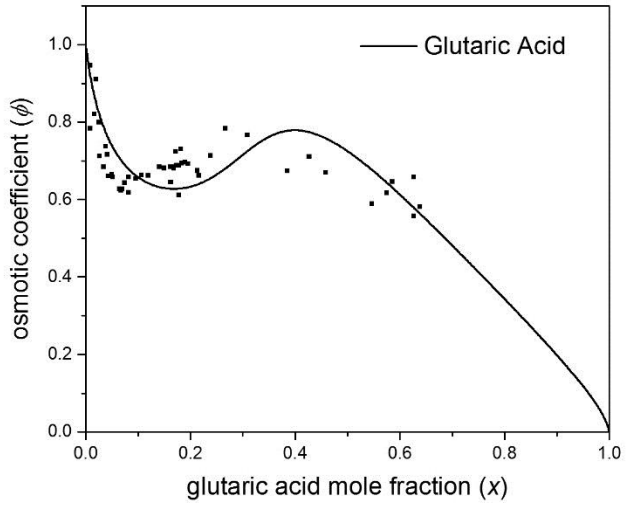

Figure 4.4. Glutaric Acid All C Parameter Fit. Osmotic coefficient versus solute mole fraction for glutaric acid. The model prediction treats each individual energy C parameter as an adjustable fit parameter (C_1=0.389634, C_2=0.000352, C_3=27452.3, C_4=0.092128, mse=0.0087). Data from Davies and Thomas, Marcolli et al., Peng et al.[86–88]

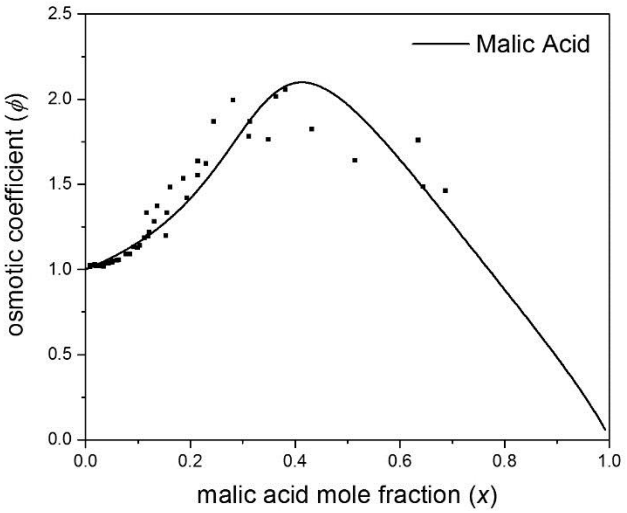

Figure 4.5. Malic Acid All C Parameter Fit. Osmotic coefficient versus solute mole fraction for malic acid. The model prediction treats each individual energy C parameter as an adjustable fit parameter (C_1=8.59186, C_2=4.41927, mse=0.0051). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

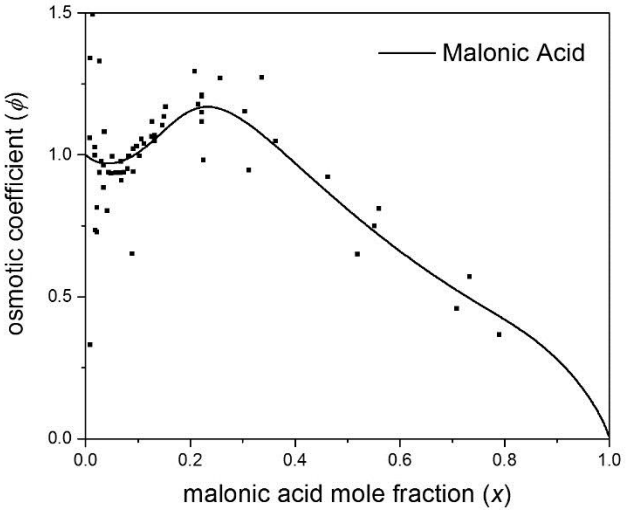

Figure 4.6. Malonic Acid All C Parameter Fit. Osmotic coefficient versus solute mole fraction for malonic acid. The model prediction treats each individual energy C parameter as an adjustable fit parameter (C_1=1.5325, C_2=0.012417, C_3=70.5216, C_4=0.3208, C_5=18.7443, C_6=0.215659, mse=0.0390). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

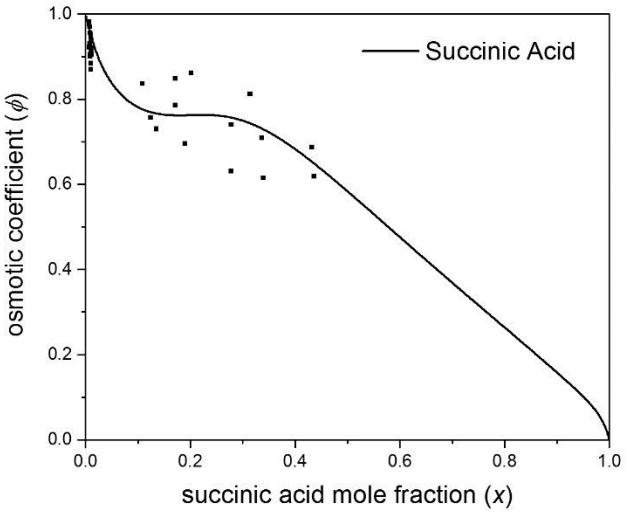

Figure 4.7. Succinic Acid All C Parameter Fit. Osmotic coefficient versus solute mole fraction for succinic acid. The model prediction treats each individual energy C parameter as an adjustable fit parameter (C_1=0.349163, C_2=0.001443, C_3=0.959903, C_4=5382.21, C_5=0.145737, mse=0.0044). Data from Davies and Thomas, Maffia and Meirelles, Peng et al., Robinson et al.[86,88–90]

individually. Applying the method of fitting all of the C parameters to organic acids had moderate success, as can be seen in Figure 4.1 through Figure 4.7. As shown in Figure 4.1, Figure 4.2, Figure 4.3, Figure 4.5, and Figure 4.7 the all C parameters fits for Acetic, Butyric, Citric, Malic, and Succinic acid each accurately fit the available data. The "all C parameters" fit for malonic acid, Figure 4.6, has good agreement with the available data, capturing both the infinite concentration and infinite dilution trends as well as the peak in osmotic coefficient that occurs around 0.3 mole fraction solute. Likewise, the "all C parameters" fit for glutaric acid shown in Figure 4.4, like the malonic acid fit, is able to capture the infinite concentration and infinite dilution limiting case trends. However, the "all C parameters" fit for glutaric acid struggles to replicate the peak in osmotic coefficient that occurs around 0.3 mole fraction solute. The fit tends to shift the peak towards higher

concentrations of solute, though does still manage to replicate it. The success of the "all C parameters" fit is likely attributable to the number of adjustable parameters, equal to the number of "C parameters" plus the 'n' parameter for the number of sorption layers.

Following the "all C parameters" fit, a power law relationship was added to calculate the C parameters, reducing the number of empirical parameters in the energy calculations to three: $C_{j,1}$, P_j, and n_j. The first C value ($C_{j,1}$) was fit and the following C parameters were calculated using the following relationship:

$$C_{j,i} = \left(i/n_j\right)^{P_j} \tag{4.2}$$

where P_j is a fit parameter and $i = 2,3, \ldots \left(n_j - 1\right)$. The idea behind applying a power law is to cause the energy C parameters to decay towards one as the solvent molecule moves further from the solute particle (as i increases). The reason to have the C parameters decay to one is that the further from the solute particle the solvent is, the less it should be affected by the solute particle, meaning that the C parameter goes to unity (equation (4.1)). If P_j is equal to zero, then all C parameters are equal to unity, meaning that there is no energy change from the sorption of water molecules. In this case, the isotherm model reduces to the idealized Raoult's Law. If P_j is less than zero, the $C_{j,i}$ parameters monotonically decrease to approach one as the monolayer location moves toward that of the multilayer. Having a P_j less than zero implies that the energy of a solvent molecule bound to a monolayer is greater than the energy of free solvent in the bulk. Likewise, if P_j is greater than zero, the $C_{j,i}$ parameters monotonically increase to approach one as the monolayer location moves toward that of the multilayer. Having P_j greater than zero implies that the

energy of a solvent molecule bound to a monolayer is less than the energy of free solvent in the bulk.

Compared to fitting each C parameter separately the power law fits reduce the number of fit parameters to three parameters for non-electrolytes ($C_{j,1}$, P_j, n_j). The fits using the power law calculation of C parameters are shown in Figure 4.8 through Figure 4.14. The power law fits for acetic acid, butyric acid, citric acid, malic acid, and succinic acid, shown in Figure 4.8, Figure 4.9, Figure 4.10, Figure 4.12 and Figure 4.14, respectively, accurately capture the trends in the osmotic coefficient data across the entire concentration range. In contrast, the power law fit for malonic acid, Figure 4.13, is unable to capture the trends shown in the osmotic coefficient data. The fit does predict the correct limiting conditions at infinite concentration or infinite dilution. The power law fit for malonic acid fails to replicate the trends over the rest of the concentration range, such as the dip in osmotic coefficient at low concentrations, or the peak in osmotic coefficient that occurs around 0.3 mole fraction solute.

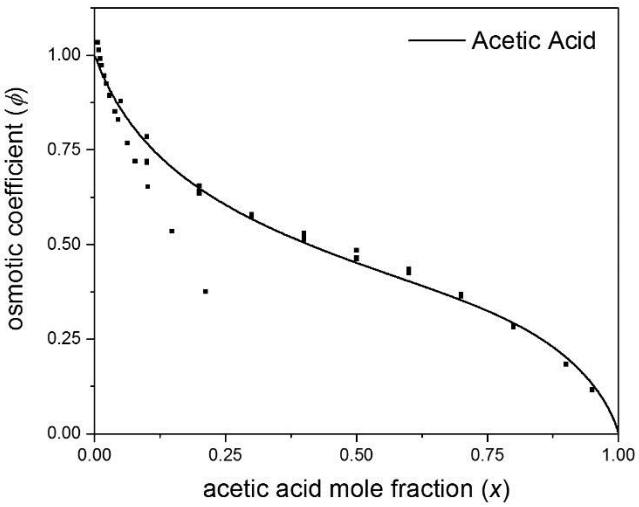

Figure 4.8. Acetic Acid Power Law C Parameter Fit. Osmotic coefficient versus solute mole fraction for acetic acid. The model fits the first C parameter, $C_{j,1}$, and calculates the rest of the C parameters using a power law relationship (n=4, C_1=0.605872, P=0.658973, mse=0.0085). Data from Hansen et al., Pirouzi et al., Sebastiani and Lacquaniti.[82–84]

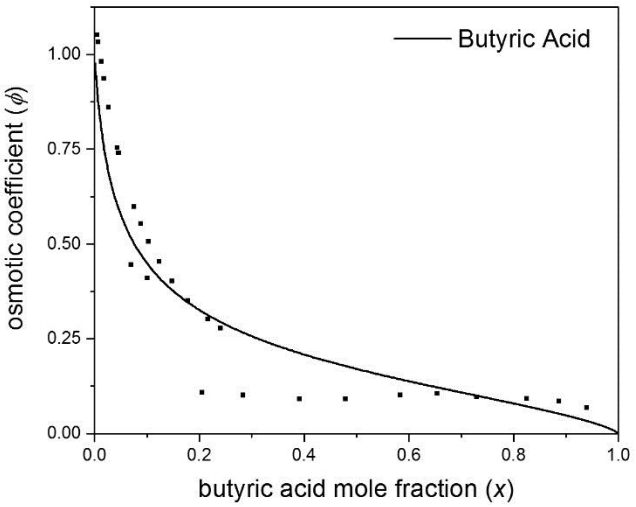

Figure 4.9. Butyric Acid Power Law C Parameter Fit. Osmotic coefficient versus solute mole fraction for butyric acid. The model fits the first C parameter, $C_{j,1}$, and calculates the rest of the C parameters using a power law relationship (n=5, C_1=2.07E-10, P=-11.5117, mse=0.1332). Data from Hansen et al., Pirouzi et al.[82,83]

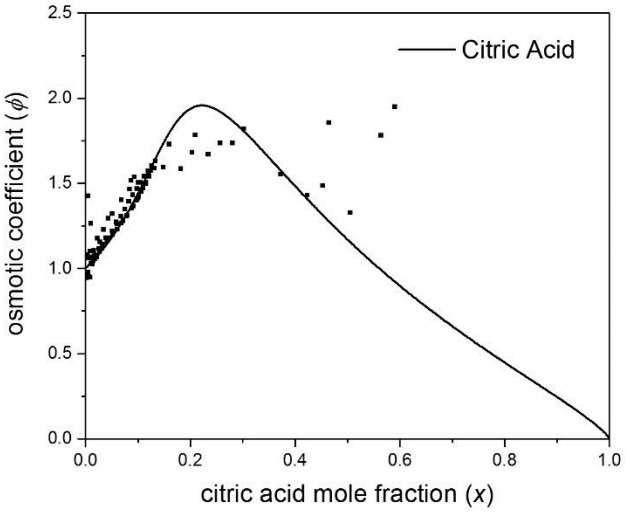

Figure 4.10. Citric Acid Power Law C Parameter Fit. Osmotic coefficient versus solute mole fraction for citric acid. The model fits the first C parameter, $C_{j,1}$, and calculates the rest of the C parameters using a power law relationship (n=6, C_1=0.402068, P=-1.4413, mse=0.0501). Data from Peng et al.[85]

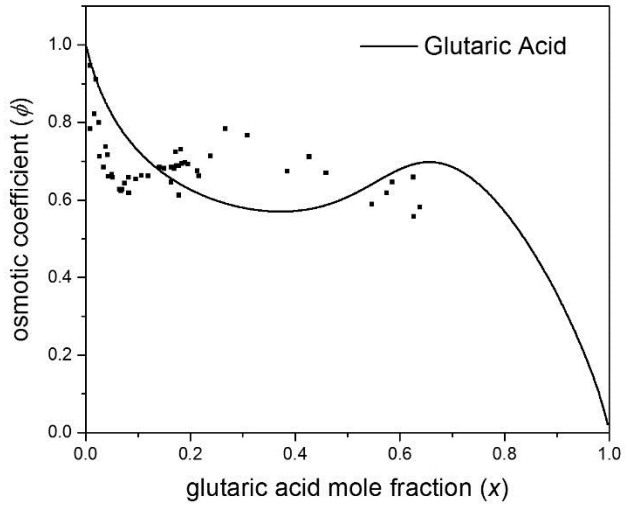

Figure 4.11. Glutaric Acid Power Law C Parameter Fit. Osmotic coefficient versus solute mole fraction for glutaric acid. The model fits the first C parameter, $C_{j,1}$, and calculates the rest of the C parameters using a power law relationship (n=4, C_1=3.00493, P=1.60346, mse=0.0202). Data from Davies and Thomas, Marcolli et al., Peng et al.[86–88]

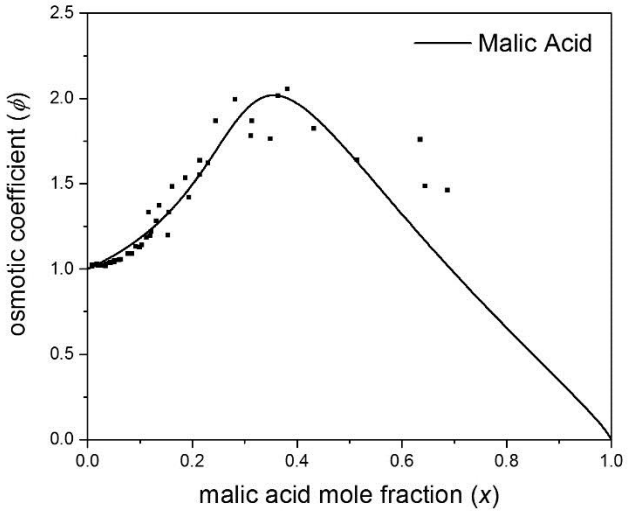

Figure 4.12. Malic Acid Power Law C Parameter Fit. Osmotic coefficient versus solute mole fraction for malic acid. The model fits the first C parameter, $C_{j,1}$, and calculates the rest of the C parameters using a power law relationship (n=3, C_1=0.95762, P=-7.37204, mse=0.0123). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

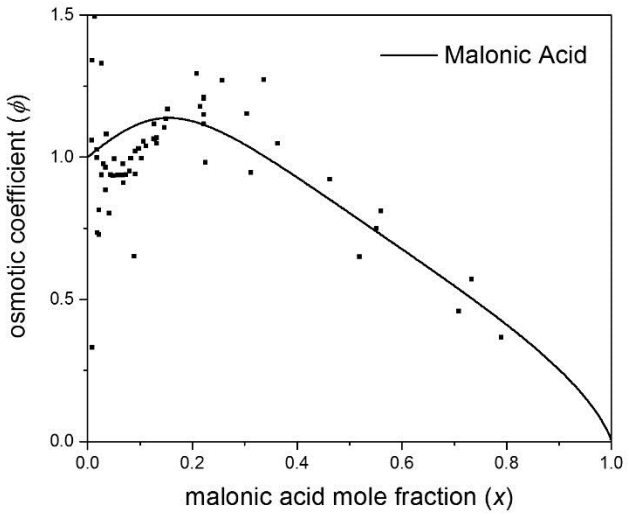

Figure 4.13. Malonic Acid Power Law C Parameter Fit. Osmotic coefficient versus solute mole fraction for malonic acid. The model fits the first C parameter, $C_{j,1}$, and calculates the rest of the C parameters using a power law relationship (n=8, C_1=0.909057, P=-0.19923, mse=0.0402). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

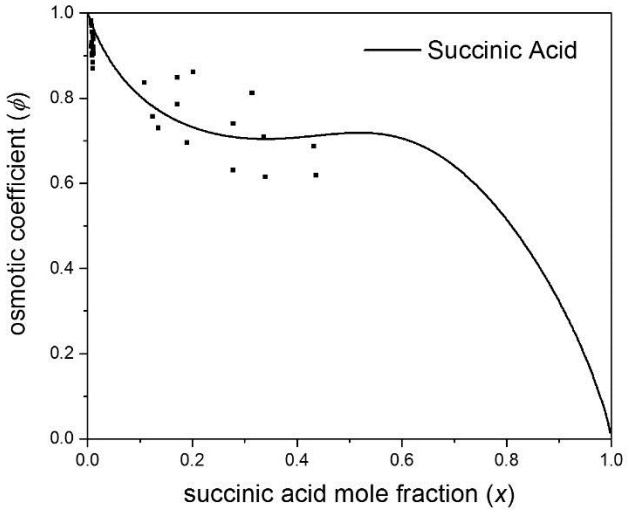

Figure 4.14. Succinic Acid Power Law C Parameter Fit. Osmotic coefficient versus solute mole fraction for succinic acid. The model fits the first C parameter, $C_{j,1}$, and calculates the rest of the C parameters using a power law relationship (n=7, C_1=2.10517, P=0.294431, mse=0.0055). Data from Davies and Thomas, Maffia and Meirelles, Peng et al., Robinson et al.[86,88–90]

Likewise, the power law fit for glutaric acid, Figure 4.11, is notably poor. The peak in the fit is shifted much farther over into the high solute concentration range than the peak in the data. The power law approach is also unable to match the steep initial decrease in osmotic coefficient that occurs at low solute concentrations. Use of the power law fit for glutaric acid, like the power law fit for malonic acid, results in the correct limiting values. The treatment of both glutaric and malonic acid need additional consideration, and are insufficiently represented by a power law expression alone.

Finally, the coulombic method for determining the energy C parameters as derived in Chapter 3 can be applied to the organic acid systems, treating the organic acids as neutral species with no disassociation. Results are shown in Figure 4.15 -Figure 4.21.

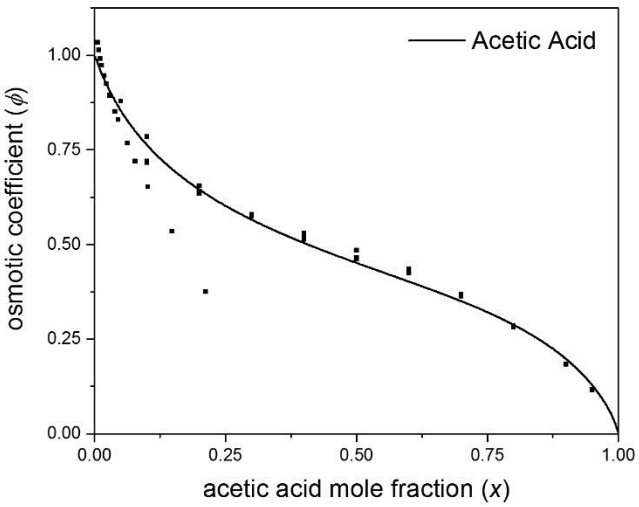

Figure 4.15. Acetic Acid Coulombic C Parameter Fit. Osmotic coefficient versus solute mole fraction for acetic acid. The model calculates the energy C parameters using coulombic interactions (n=6, μ_j=1.43626, r_{jw}=2.28Å, mse=0.0080). Data from Hansen et al., Pirouzi et al., Sebastiani and Lacquaniti.[82–84]

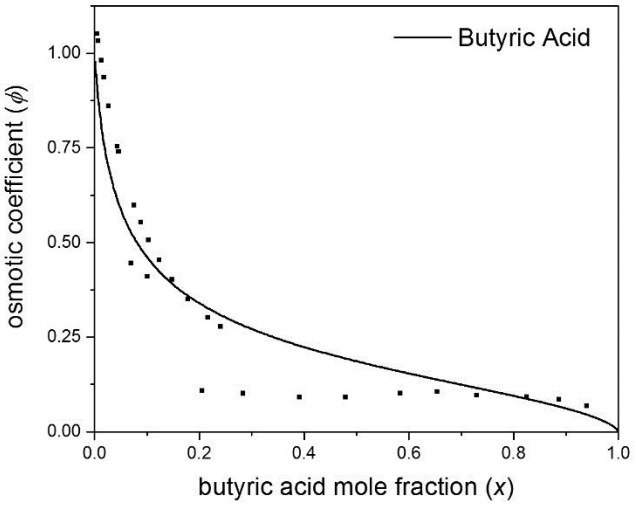

Figure 4.16. Butyric Acid Coulombic C Parameter Fit. Osmotic coefficient versus solute mole fraction for butyric acid. The model calculates the energy C parameters using coulombic interactions (n=3, μ_j=157.781, r_{jw}=13.5Å, mse=0.0931). Data from Hansen et al., Pirouzi et al.[82,83]

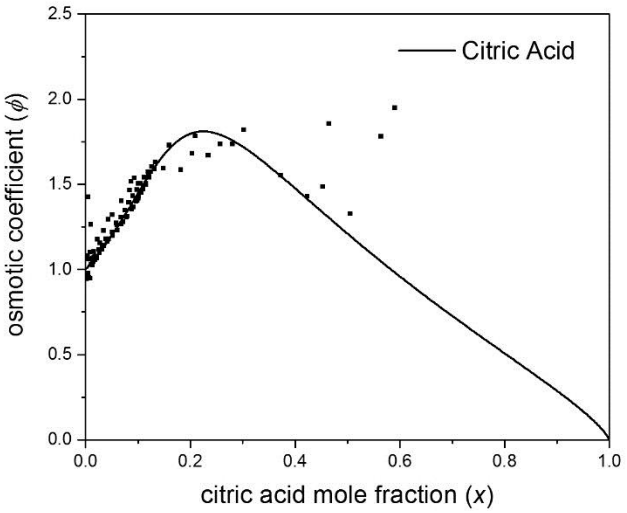

Figure 4.17. Citric Acid Coulombic C Parameter Fit. Osmotic coefficient versus solute mole fraction for citric acid. The model calculates the energy C parameters using coulombic interactions (n=9, μ_j=16.1368, r_{jw}=5.00Å, mse=0.0423). Data from Peng et al.[85]

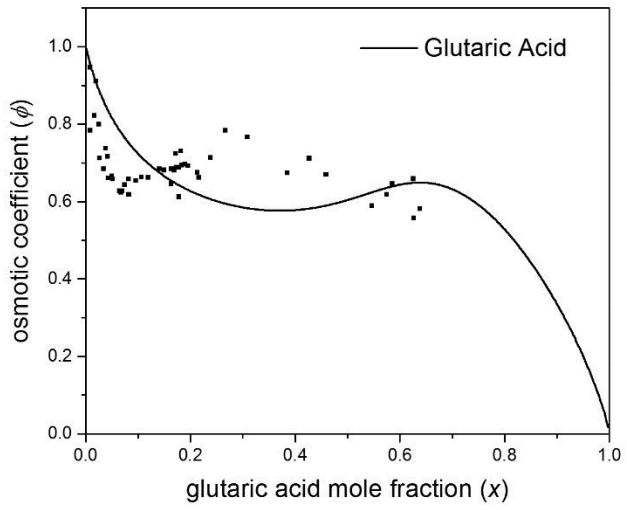

Figure 4.18. Glutaric Acid Coulombic C Parameter Fit. Osmotic coefficient versus solute mole fraction for glutaric acid. The model calculates the energy C parameters using coulombic interactions (n=7, μ_j=0.159749, r_{jw}=1.04Å, mse=0.0178). Data from Davies and Thomas, Marcolli et al., Peng et al.[86–88]

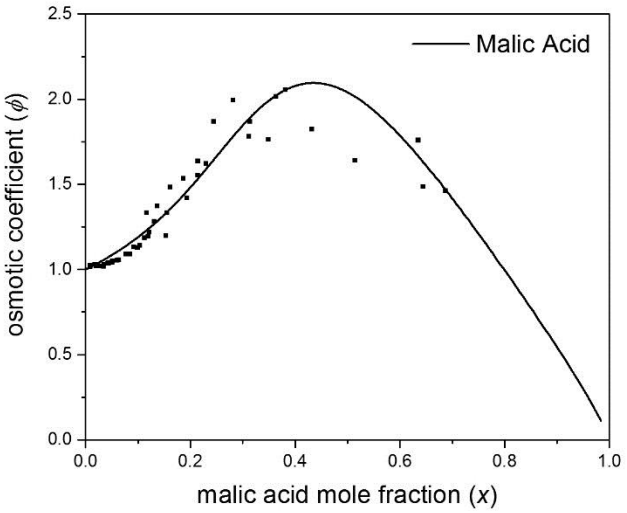

Figure 4.19. Malic Acid Coulombic C Parameter Fit. Osmotic coefficient versus solute mole fraction for malic acid. The model calculates the energy C parameters using coulombic interactions (n=4, μ_j=7.67324, r_{jw}=3.57Å, mse=0.0038). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

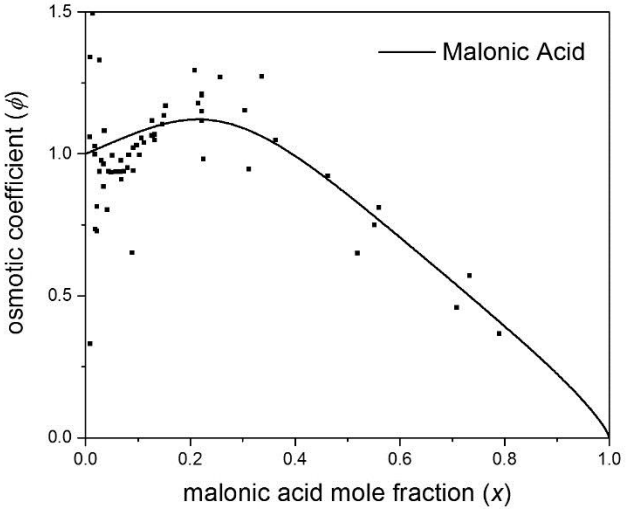

Figure 4.20. Malonic Acid Coulombic C Parameter Fit. Osmotic coefficient versus solute mole fraction for malonic acid. The model calculates the energy C parameters using coulombic interactions (n=3, μ_j=40.2079, r_{jw}=7.09Å, mse=0.0390). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

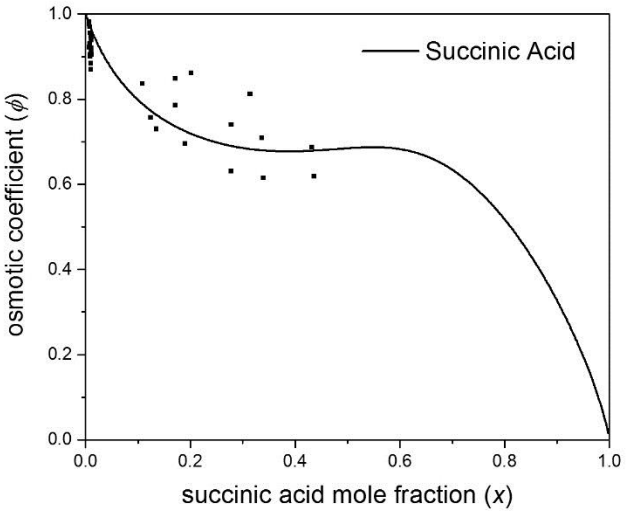

Figure 4.21. Succinic Acid Coulombic C Parameter Fit. Osmotic coefficient versus solute mole fraction for succinic acid. The model calculates the energy C parameters using coulombic interactions (n=10, μ_j=0.6911, r_{jw}=1.70Å, mse=0.0055). Data from Davies and Thomas, Maffia and Meirelles, Peng et al., Robinson et al.[86,88–90]

The use of the Coulombic fit for malonic acid, like the power law fit, is unable to capture the trends shown in the osmotic coefficient data. While it predicts the correct limiting values, the fit fails to replicate the dip and peak of the osmotic coefficient data. Likewise, the coulombic fit for glutaric acid is similar to the power law fit for glutaric acid. The Coulombic fit is unable to match the quickly decreasing osmotic coefficient at low mole fraction solute, and shifts the osmotic coefficient peak into a higher mole fraction range ignoring the osmotic coefficient peak shown in the data.

The Coulombic fits for acetic acid, butyric acid, citric acid, malic acid, and succinic acid all fit the data in the same manner as their respective power law fits.

Although the fits using the coulombic relation are similar to the fits using the power law, the coulombic fits lend themselves to physical interpretation from being based on a physical interaction, instead of a purely empirical fit.

The reduced parameter treatment of the C parameters using approaches from Chapter 2 (Power law fits) or Chapter 3 (Coulombic fits) has been successful for most electrolytes and organic species, but fail to capture the trends shown in malonic and glutaric acid. The less sophisticated approach of parameterization using each individual C value as a fit parameter could be used, but with loss of physical interpretation. Organic acids, such as malonic acid and glutaric acid, are known to disassociate into ions, similar to an electrolyte though only partial dissociation. Instead, a treatment incorporating the partial disassociation of organic acids can be used in the context of the Coulombic model.

4.3 Organic Acid Model Adjustments: Incorporating organic acid dissociation

In Chapter 3, the model parameters needed to model organics are reduced to two parameters (r_{jw},n). In addition the model for electrolytes has been reduced to three parameters (r_{jj},ρ,n). If a partially disassociating organic species is modeled as a mixture between a dissociated organic species (modeled as an electrolyte) and a non-dissociated organic species (modeled as an organic), an additional parameter, α, indicating the degree to which the organic acid has dissociated, is required. This initially gives the model a total of six parameters; three fit parameters ($r_{jw,org}$, $r_{jw,acid}$, ρ) and three adjustable parameters (n_{org}, n_{acid}, α). The required model equations, derived in chapter 2, are as follow:

$$\bar{m}_j^o = \frac{\left(\frac{1-\bar{a}_w}{M_w v_j \bar{a}_w}\right)\left(1-\sum_{i=1}^{n_j-1}\left((\bar{a}_w)^i\left(1-C_{j,i}\right)\prod_{k=1}^{i-1}C_{j,k}\right)\right)}{(1-\bar{a}_w)^2\sum_{p=1}^{n_j-2}\left(p(\bar{a}_w)^{p-1}\prod_{k=1}^{p}C_{j,k}\right)+\left(\left(n_j-1\right)-\left(n_j-2\right)\bar{a}_w\right)(\bar{a}_w)^{n_j-2}\prod_{k=1}^{n_j-1}C_{j,k}} \tag{4.3}$$

$$\sum_j \frac{m_j}{\bar{m}_j^o} = 1 \tag{4.4}$$

$$a_w = \bar{a}_w K_w^{DH} \tag{4.5}$$

$$K_w^{DH} = \exp\left(\frac{A_x {I_x}^{1/2}\sum_j\left(v_j N_j\left|z_{j-}\right|z_{j+}/\left(1+\rho_j {I_x}^{1/2}\right)\right)}{\sum_j v_j N_j + N_w}\right) \tag{4.6}$$

$$C_{j,i} = \exp\left(\left(\frac{\mu_j \mu_w D^2}{4\pi\varepsilon_0\left(r_{jw}+(i-1)r_{ww}\right)^3}-\frac{\mu_w \mu_w D^2}{4\pi\varepsilon_0(i\cdot r_{ww})^3}\right)/kT\right) \tag{4.7}$$

$$\mu_{acid} = qe\left(r_{jw}\right)/D \tag{4.8}$$

$$\mu_{org} = \left(r_{jw}/2\text{Å}\right)^3 \tag{4.9}$$

where, as before, m_j is the molality of the species j in solution and $\bar{m}_j^o$ is the molality of the solute j in a pure aqueous solution at the $\bar{a}_w$ of the mixture. Note that equation (4.8) differs from equation (3.15) for electrolytes presented in chapter 3 due to the fact that the ionic radii are unknown for organic acids, thus the full value of r_{jw} is used in place of r_{jj}. Writing out equation (2.51) for a partially dissociated organic acid gives the following

$$m_{org}\left(\frac{1}{\bar{m}_{org}^o}+\frac{\alpha}{\bar{m}_{acid}^o}\right) = 1 \tag{4.10}$$

where $\alpha = m_{acid}/m_{org}$ is the concentration ratio of dissociated acid to non-dissociated acid.

It is hypothesized that the known dissociation constants, K_a, can be used in the determination of the dissociation ratio, α. The dissociation constant, K_a is given by

$$\frac{[H^+][A^-]}{[HA]} = K_a \tag{4.11}$$

Testing this hypothesis with the assumption that the only source of hydrogen ions is from the dissociation of the organic acid gives us $[H^+] = [A^-]$, and that the concentration of non-dissociated acid, $[HA]$, is equal to the provided data concentration minus the concentration of dissociated acid. Substituting these equations for $[H^+]$ and $[HA]$ into to equation (4.11) results in a polynomial in terms of $[A^-]$ which can be solved for each data point, giving a concentration dependent dissociation ratio.

Many of the organic acids explored in this paper have multiple dissociation constants. For simplicity the model has assumed that each organic acid molecule can only dissociate one, effectively making a 1:1 electrolyte. This assumption is justified because the second dissociation constant for the organic acids is smaller than the first dissociation constant, meaning not many solute particles will dissociate twice.

The issue that arose using this method was that the resulting concentration of dissociated organic acid was too low to have any beneficial impact on the model fit. The resulting fits suffered from the same issues that the coulombic fits faced, and looked similar to their corresponding coulombic fit. *This may indicate that the apparent dissociation is greater than the actual dissociation, or that the effects of dissociation on an aqueous solution is more than just the sum of its parts*.

For simplicity, a static ratio between the disassociated and non-disassociated organic acid concentrations was used initially at all solute concentrations. Static disassociation ratios below 1:0.1 for non-disassociated to disassociated organic species exhibited the same insignificant contribution issues that the concentration dependent

disassociation ratio faced. For this study two different disassociation ratios were chosen to test, 1:0.5 and 1:0.1.

The partial disassociation model fit for malonic acid, Figure 4.27, is able to accurately capture the trends of the data across the entire available data range. The model also captures the expected limiting conditions at high solute concentrations. For malonic acid, the static disassociation ratio of 1:0.5 resulted in the best fit, and is plotted in Figure 4.27. The static disassociation ratio of 1:0.1 also resulted in a fit that accurately reproduced the available data.

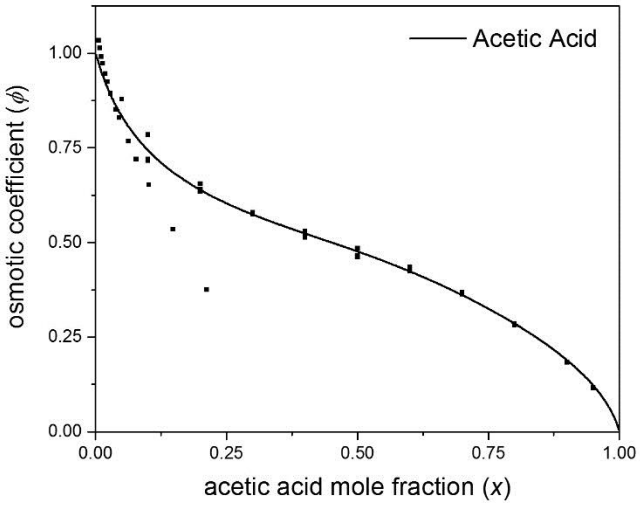

Figure 4.22. Acetic Acid Constant Disassociation Ratio Fit. Osmotic coefficient versus solute mole fraction for acetic acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=4, n_{org}=4, ρ_{acid}=11143, $r_{jw,acid}$=6.77Å, $r_{jw,org}$=1.60Å, α=0.1, mse=0.0060). Data from Hansen et al., Pirouzi et al., Sebastiani and Lacquaniti.[82–84]

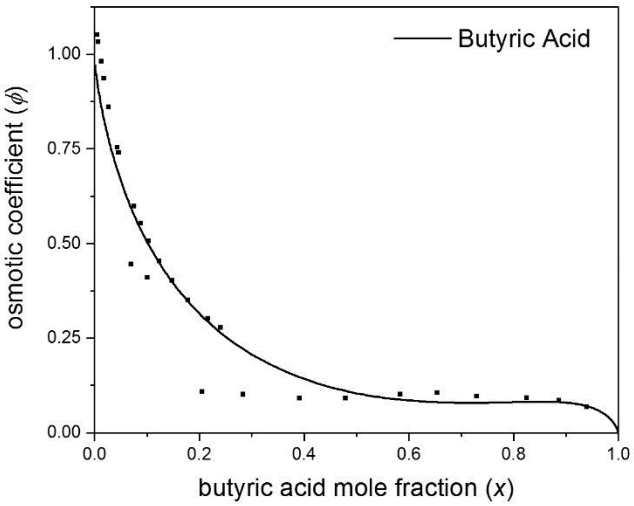

Figure 4.23. Butyric Acid Constant Disassociation Ratio Fit. Osmotic coefficient versus solute mole fraction for butyric acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=8, n_{org}=6, ρ_{acid}=0.351687, $r_{jw,acid}$=8.81Å, $r_{jw,org}$=2.22Å, α=0.5, mse=0.0544). Data from Hansen et al., Pirouzi et al.[82,83]

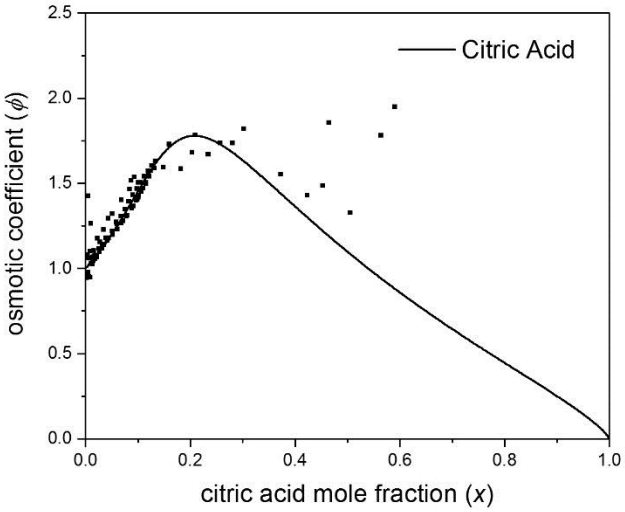

Figure 4.24. Citric Acid Constant Disassociation Ratio Fit. Osmotic coefficient versus solute mole fraction for citric acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=8, n_{org}=6, ρ_{acid}=4335.72, $r_{jw,acid}$=7.16Å, $r_{jw,org}$=5.99Å, α=0.1, mse=0.04833). Data from Peng et al.[85]

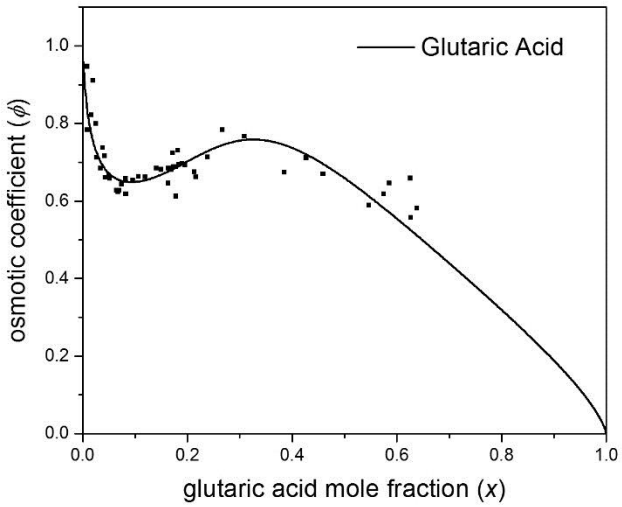

Figure 4.25. Glutaric Acid Constant Disassociation Ratio Fit. Osmotic coefficient versus solute mole fraction for glutaric acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=7, n_{org}=8, ρ_{acid}=18.5807, $r_{jw,acid}$=10.2Å, $r_{jw,org}$=5.08Å, α=0.5, mse=0.0142). Data from Davies and Thomas, Marcolli et al., Peng et al.[86–88]

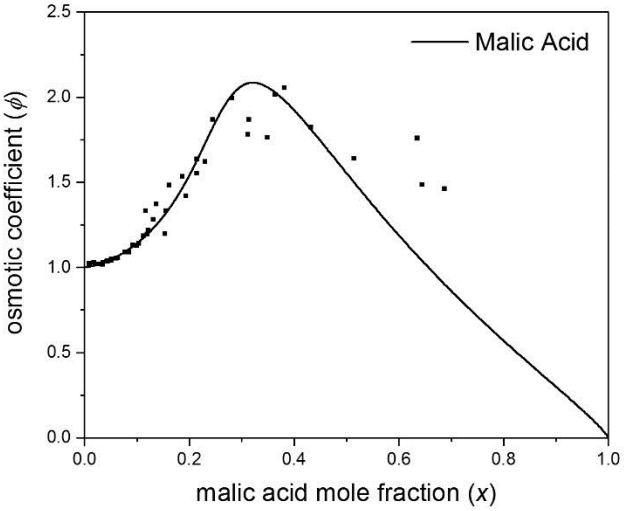

Figure 4.26. Malic Acid Constant Disassociation Ratio Fit. Osmotic coefficient versus solute mole fraction for malic acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=5, n_{org}=4, ρ_{acid}=628.583, $r_{jw,acid}$=8.48Å, $r_{jw,org}$=8.99Å, α=0.1, mse=0.0318). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

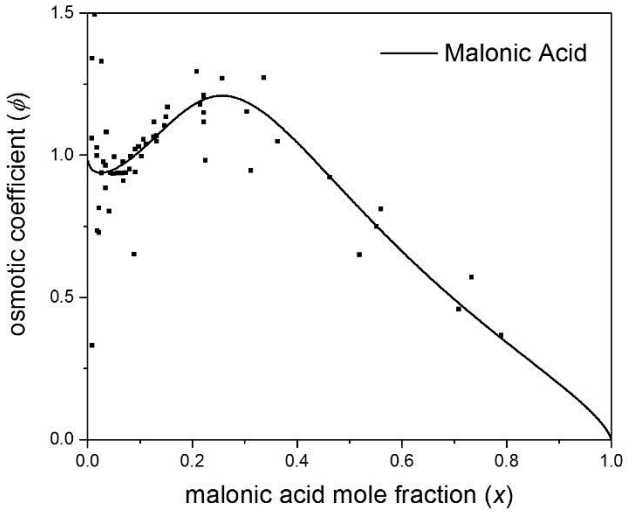

Figure 4.27. Malonic Acid Constant Disassociation Ratio Fit. Osmotic coefficient versus solute mole fraction for malonic acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=8, n_{org}=3, ρ_{acid}=0.5258, $r_{jw,acid}$=6.27Å, $r_{jw,org}$=3.67Å, α=0.5, mse=0.0528). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

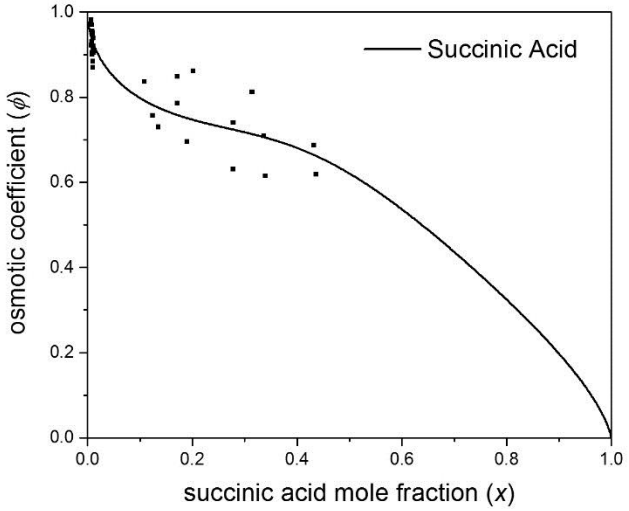

Figure 4.28. Succinic Acid Constant Disassociation Ratio Fit. Osmotic coefficient versus solute mole fraction for succinic acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=3, n_{org}=3, ρ_{acid}=0.982048, $r_{jw,acid}$=6.25Å, $r_{jw,org}$=2.80Å, α=0.5, mse=0.0048). Data from Davies and Thomas, Maffia and Meirelles, Peng et al., Robinson et al.[86,88–90]

The partial disassociation model fit for glutaric acid, Figure 4.25, also accurately reproduced the available data and accurately predicts the correct limiting conditions at high solute concentrations. However, for glutaric acid, the static disassociation ratio of 1:0.1 is too low, and the fit fails in a manner that is similar to the coulombic fit for glutaric acid. Using the higher disassociation ratio of 1:0.5 the results in a more accurate prediction.

The partial disassociation model fits for acetic acid, butyric acid, citric acid, malic acid, and succinic acid are in agreement with available data across the entire concentration range, much like their equivalent coulombic fits. For these organic acids both disassociation ratios, 1:0.1 and 1:0.5, resulted in fits that were in agreement with the available data. The inclusion of the disassociation contribution was slightly detrimental to the fits for citric acid, and thus a lower disassociation ratio, 1:0.01, was used. Given that acetic acid, butyric acid, citric acid, malic acid, and succinic acid are all successfully modeled using the simpler coulombic fit without disassociation it makes sense that choosing a lower disassociation ratio, such as with citric acid, results in a model fit that is still accurate. Overall, a ratio of 1:0.5 to 1:0.1 seemed to work well for the [non-dissociated]:[dissociated] ratio. The largest issue with this method is the lack of a physical meaning behind the constant dissociation ratio.

It was found that across multiple fitting approaches there was a general trend for $r_{jw,acid}$ to be approximately twice $r_{jw,org}$. This relationship can be seen in the parameterization of glutaric and malonic acid provided in the captions of Figure 4.25 and Figure 4.27. Applying this trend as a constraint (i.e. constraining $r_{jw,acid} = 2r_{jw,org}$) as a method to reduce the number of parameters was successful for species such as malonic

acid and glutaric acid. This constraint reduces the number of fit parameters by allowing us to only fit one r_{jw} value and calculate the remaining r_{jw}.

With $r_{jw,acid}$ constrained to twice $r_{jw,org}$ the disassociation fit for glutaric acid, Figure 4.29, is similar to the model prediction without the constraint. The model agrees with the available data and with the expected limiting conditions. The glutaric acid fits struggle with disassociation ratios lower than 1:0.5 similar to the fits without the constraint on r_{jw}.

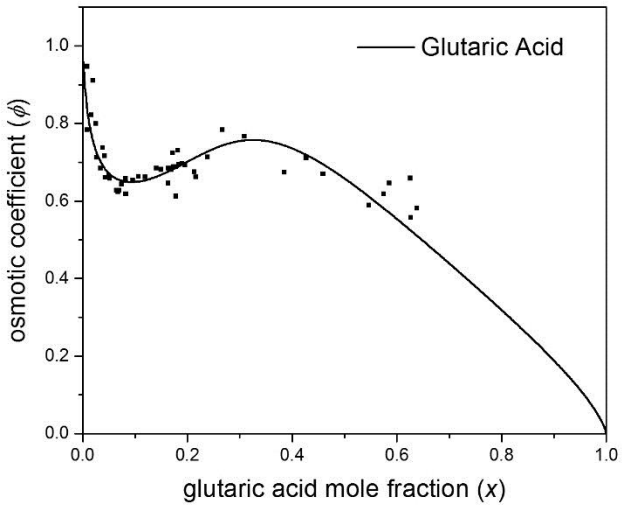

Figure 4.29. Glutaric Acid Constrained r_{jw} Fit. Osmotic coefficient versus solute mole fraction for glutaric acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=7, n_{org}=8, ρ_{acid}=18.498, $r_{jw,org}$=5.08Å, α=0.5, mse=0.0143). Data from Davies and Thomas, Marcolli et al., Peng et al.[86–88]

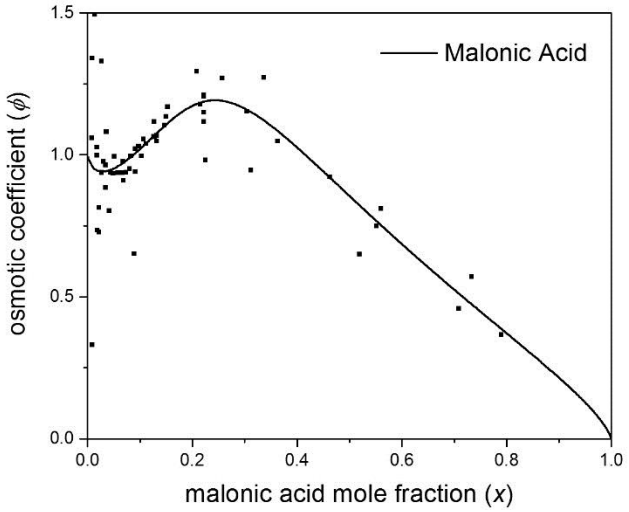

Figure 4.30. Malonic Acid Constrained r_{jw} Fit. Osmotic coefficient versus solute mole fraction for malonic acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=8, n_{org}=6, ρ_{acid}=1.19E-07, $r_{jw,org}$=5.05Å, α=0.1, mse=0.0432). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

With $r_{jw,acid}$ constrained to twice $r_{jw,org}$ the disassociation fit for Malonic Acid, Figure 4.30, remains remarkably similar to that of the fit without the constraint. The model agrees with the data across the entire concentration range as well as the limiting conditions. For the model parameterization a disassociation ratio close to 1:0.1 is better than 1:0.5 Pushing the disassociation ratio lower than 1:0.1 results in the disassociated organic no longer having a large impact indicated by a prediction similar to the Coulombic fit in section 4.2.

4.4 Single Parameter Model for Atmospherically Relevant Organic Acid Model

Up until this point the parameter ρ has been treated as either a constant[21,22,91] or as a fit parameter, as was done throughout chapter 3. A theoretical derivation of ρ performed

by Pitzer and Simonson[20] gives us the following definition of ρ in terms of the hard-core collision diameter, a, of the solute:

$$\rho = a(2e^2 N_A d_1 / M_1 \varepsilon_0 DkT)^{1/2} \tag{4.12}$$

where a is the hard-core collision diameter, e is the electronic charge, k is Boltzmann's constant, T is the Temperature, ε_0 is the permittivity of free space, N_A is Avogadro's number, d_1 is the density of the solvent, M_1 is the molecular weight of the solvent, D is the dielectric constant (relative permittivity) of the solvent.

Given that the hard-core collision diameter of a solute ion is difficult to measure, the non-dissociated organic radius, $r_{jw,org}$, was used as the hard-core collision diameter for the solute ion. Using $r_{jw,org}$ as the hard-core collision diameter in equation (4.12) and forcing $r_{jw,acid}$ to be twice $r_{jw,org}$, we are, for a given n_{org}, n_{acid} and α, able to obtain a model that has a single fit parameter, $r_{jw,org}$.

Remarkably, even with the adjustments to the model making it a "single parameter" model, the fits for malonic and glutaric acid remain very similar to their fits when ρ was used as a fit parameter, or even when the values for $r_{jw,acid}$ and $r_{jw,org}$ were uncoupled. Using the approach discussed here, the reduction to a single parameter has had little effect on the predictions or MSE of the model when compared to the MSE of the partial disassociation model using three fit parameters.

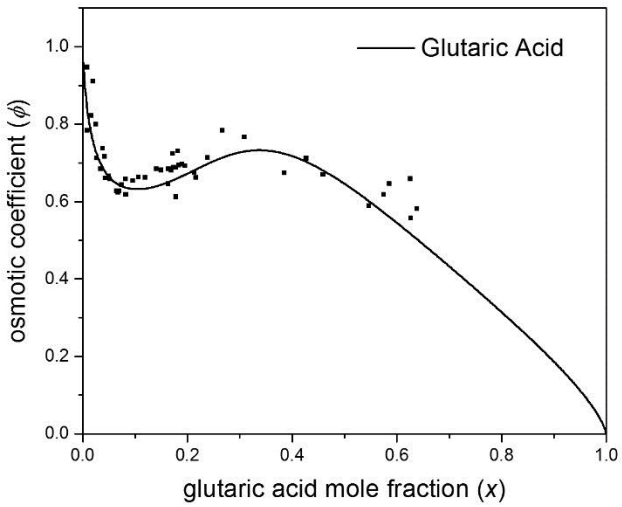

Figure 4.31. Glutaric Acid Single Parameter Disassociation Fit. Osmotic coefficient versus solute mole fraction for glutaric acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=8, n_{org}=7, $r_{jw,org}$=5.10Å, α=0.5, mse=0.0151). Data from Davies and Thomas, Marcolli et al., Peng et al.[86–88]

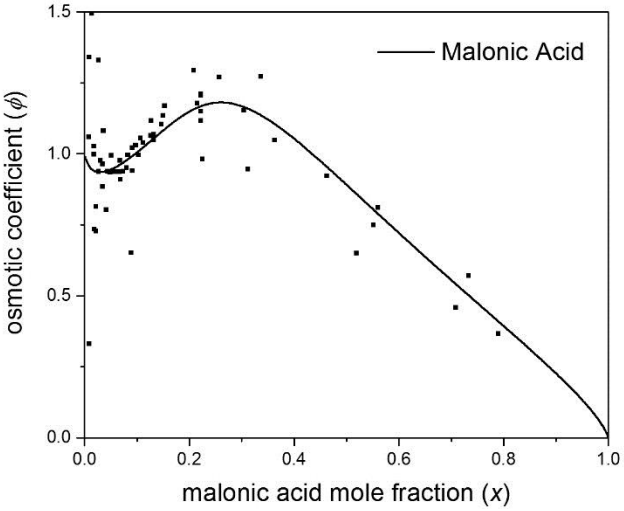

Figure 4.32. Malonic Acid Single Parameter Disassociation Fit. Osmotic coefficient versus solute mole fraction for malonic acid. The model calculates the energy C parameters using coulombic interactions. In addition, the organic acid is allowed to disassociate. (n_{acid}=8, n_{org}=5, $r_{jw,org}$=5.12Å, α=0.1, mse=0.0434). Data from Davies and Thomas, Maffia and Meirelles, Marcolli et al., Peng et al.[86–89]

4.5 Summary

By accounting for partial dissociation of organic acids the model has been able to successfully capture the trends of organic acids with only a small increase in the number of parameters needed for the model. The partial disassociation approach also implemented a theoretical equation for ρ, where previous models have treated it as a constant or an empirical fit parameter. The partial disassociation model, like the original coulombic model has reduced the empirical dependence of the parameters by providing physical interpretations of the fit parameters allowing checks to see if the predicted parameter values are of a reasonable magnitude.

Outside of glutaric acid and malonic acid, the other dicarboxylic and carboxylic acids modeled are capable of being modeled by the coulombic interaction without the need to account for disassociation. The fits for glutaric and malonic acid are greatly improved with the inclusion of partial disassociation. Despite the added complexity of including partial disassociation the number of fit parameters does not increase greatly. Compared to the coulombic model without partial disassociation the only parameters that are added are an additional n value and a term for the degree of partial disassociation.

Chapter 5

Conclusion

Throughout this document, a model for activity coefficient has been derived from first principles that is accurate across the entire concentration range, from pure solvent to the theoretical pure liquid solute. The model is applicable to both electrolytes and non-electrolytes, as well as mixtures of both through the use of the zero parameter mixing law given by equation (2.51). Chapter 2 details the development of the model using a unified treatment. Here multilayer adsorption isotherms were used to resemble the interactions between the sorbate (solvent- water) and sorbent (solute- electrolytes or organics) species. Toward interpretation and reduction of the adsorption isotherm model parameter, in Chapter 3 the model was applied to multiple electrolytes and non-electrolytes, obtaining model parameters for each species. The development of relationships between model parameters allowed for a further reduction in the number of parameters. The reduction in parameters resulted in a model for non-electrolytes that requires only two parameter, n_j and r_{jw}, and a model for electrolytes that requires three parameter, n_j, r_{jj} or r_{jw}, and ρ.

In Chapter 3, the treatment of electrolytes using coulombic interactions is in good agreement with available data for all solutes except $LiNO_3$ and HNO_3. The model accurately captures the low concentration behavior of $LiNO_3$ and HNO_3 but is unable to predict the behavior at higher concentrations. However, $LiNO_3$ and HNO_3 can both be accurately modeled using the power law fit for the C parameters employed in previous work as well as the initial treatment in Chapter 4. The poor fit for $LiNO_3$ and HNO_3 may

be due to the planar structure of the nitrate ion, which may allow for a closer approach for small cation than an equivalently sized elemental anion. Another source of the anomalous nature may be that electrolyte solutions containing lithium are known to form clusters,[65] which are not being adequately captured by the current model.

The use of electrostatic coulombic interactions to model the interactions between solvent and solute molecules yield accurate predictions of activity and osmotic coefficients for a wide range of electrolyte and nonelectrolytes in water, as shown in Chapter 3. However, the model presented in Chapter 3 had difficulties modeling some organic acid, such as malonic acid or glutaric acid. In Chapter 4, the model is applied to organic acids by incorporating partial disassociation in addition to the coulombic interaction used in Chapter 3. The initial application to organic acids using partial disassociation required significantly more parameters than coulombic fits used in Chapter 3. However, through the use of known parameter definitions, such as equation (4.12), and through observed parameter correlations, relationships between multiple fit parameters were developed, reducing the number of required parameters. Finally, the resulting model is able to accurately predict the trends present in organic acids without requiring a large number of parameters. For a given disassociation ratio, α, and number of sorption layers, n_{acid} and n_{org}, only one adjustable parameter, $r_{jw,org}$, is needed.

The electrostatic interaction implementation provides a physical interpretation for the adjustable parameters, meaning that the parameterization itself can now give insight into unknown physiochemical properties such as intermolecular spacing, r_{jw}. However, caution should be used to prevent over interpretation of the parameterization, especially

with the values associated with the inclusion of partial dissociation. The partial dissociation parameterization was done at a degree of dissociation that is at least an order of magnitude larger than experimental values for the average degree of dissociation across the entire concentration range. The use of a degree of dissociation for the organic acids that was much larger than the experimentally determined may, as stated before, be an indication that the partial disassociation of an organic acid in solution has a larger effect on the water activity than simply being a mixture of non-disassociated organic and dissociated organic. Additional interaction parameters may be necessary to fully and accurately model the partial dissociation of organic acids. Finally, to simplify the model, it was assumed that the dicarboxylic acids only disassociated once instead of twice. It is expected that errors caused by this simplification will be minimal, since the disassociated portion of the organic is a small part of the total solution.

The reduction of necessary parameters that occurs in Chapter 3 and over the course of Chapter 4 means that the model is less dependent on having large sets of experimental data. Since data for activity or osmotic coefficient is scarce, especially at high solute concentrations or for multicomponent systems, moving away from a dependence on large sets of experimental data allows for exploration into modeling molecular species that previously lacked sufficient experimental data.

5.1 Future Work: Solute Association

In addition to the partial dissociation of species such as organic acids, there is also association that can occur amongst species like sucrose.[92] Sucrose, as noted in chapter 3,

has an abnormally high n value compared to other organic species fit in the same manner. This may be caused by sucrose associating with itself in solution, effectively having the opposite effect on total molality than dissociation.

By making slight adjustments to the model for partial *dissociation*, it may be possible to account for particle *association* amongst species that often associate with themselves in solution. A preliminary attempt at this has switched the degree of disassociation term into a degree of association term, given by

$$\alpha_{association} = \frac{m_{associated}}{m_{non-associated}} \tag{5.1}$$

In the case of partial dissociation, the total molality increased as the species disassociated. For association, the total molality will now decrease as the species associates.

The model treats an associated pair of molecules as a single organic molecule. The model is be set up to treat the organic as a mixture of single non-associated molecules and pair associated molecules. Fitting this model in the same way the model for partial disassociation was fit obtains results that agree well with available sucrose data. The method of introducing association results in a fit that is comparable to the coulombic fit presented in chapter 3, though this might be heavily attributed to the increase in the number of adjustable parameters when fitting sucrose using this model to assume association. Figure 5.1 provides a proof of concept of the use of the association approach.

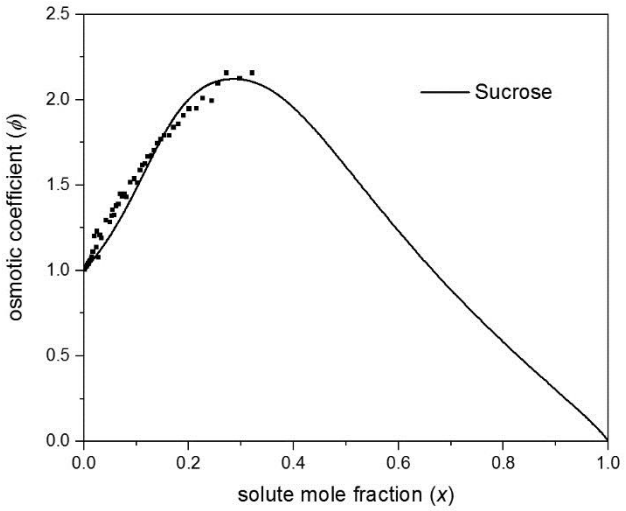

Figure 5.1. Sucrose with Self-Association. Osmotic coefficient plotted versus solute mole fraction for sucrose using a model that accounts for association between sucrose molecules. ($n_{association}$=8, n_{org}=7, $r_{jw,association}$=10.9Å, $r_{jw,org}$=4.71Å, α=0.5, mse=0.0127). Data from Scatchard et al, and Bubnik et al. (data from Baeza et al. 2010).[47,51]

Computational future work needs to go into a better analysis of how to handle the molality of the system when dealing with association. A better way to differentiate between the associated pair molecules and the solo molecules when fitting the model, as the current mathematical treatment has the two types of molecules as almost interchangeable, outside of the initial guess values for parameterization.

Further investigation into trends involving the number of sorption shells, *n*, is needed. For both electrolyte and non-electrolyte solutions there are instances of high '*n*' values, nominally indicating a large number of solvation layers. However, what is more likely is that larger '*n*' values are simply compensating for solute-solute association effects (e.g. dimerization for non-electrolytes, associations for electrolytes). Despite this caveat, large *n* values may still be within the realm of possibility. For instance, measurements have

shown some very long range effects from sulfate ions on water well outside the first solvation ring.[93] A deeper look into how varying the number of sorption shells, n, affects the model parameterization may provide better insight into the interpretation of an adsorption layer, a concept stemming from the traditional gas adsorption isotherm, in the context of solutions. In general, more consideration into the interpretation of the traditional gas adsorption isotherm constructs applied to solutions is needed. In the traditional gas adsorption isotherm "*r*" is the number of adsorption sites, "*n*" is the number of adsorption layers, and X_{ij} is the number of water molecules in layer i surrounding solute *j*. Instead, in solution, these parameters may be more along the lines of "*r*" being the number of sources or point charges or induced dipoles that induce a local non-Gaussian distribution of solvent molecules in some structural, dynamic, or electric manner,[94] "*n*" should be recast in terms of some radial distance from the source, and X_{ij} is a relative fractional number of solvent molecules effected by the point source at a given distance.

In conclusion, this work represents a significant advancement toward a fully predictive quantitative structure property relationship for activity coefficient modeling. For non-electrolytes, the physical properties necessary to calculate the energies of adsorption are the intermolecular solute-solvent bond length and the solvent-solvent bond length. For electrolyte solutions, the ion charge types, ion-solvent bond lengths, and solvent-solvent bond lengths are the parameters necessary for calculating the energy of adsorption. The majority of the physical properties, with the exception of the intermolecular distances for many solutes, are available in the literature.

Bibliography

(1) Forster, P.; Ramaswamy, V.; Artaxo, P.; Berntsen, T.; Betts, R.; Fahey, D. W.; Haywood, J.; Lean, J.; Lowe, D. C.; Myhre, G.; et al. 2007: Changes in Atmospheric Constituents and in Radiative Forcing. In *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*; Cambridge University Press: Cambridge ; New York, 2007; pp 131–234.

(2) Seinfeld, J. H.; Pandis, S. N. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*; Wiley: New York, 1998.

(3) Ramanathan, V. Aerosols, Climate, and the Hydrological Cycle. *Science* **2001**, *294* (5549), 2119–2124.

(4) Grabowski, W. W. Indirect Impact of Atmospheric Aerosols in Idealized Simulations of Convective–Radiative Quasi Equilibrium. *J. Clim.* **2006**, *19* (18), 4664–4682.

(5) Ohm, P. B.; Asato, C.; Wexler, A. S.; Dutcher, C. S. Isotherm-Based Thermodynamic Model for Electrolyte and Nonelectrolyte Solutions Incorporating Long- and Short-Range Electrostatic Interactions. *J. Phys. Chem. A* **2015**, *119* (13), 3244–3252.

(6) Archer, D. G. Thermodynamic Properties of the NaCl+H2O System. II. Thermodynamic Properties of NaCl(aq), NaCl·2H2(cr), and Phase Equilibria. *J. Phys. Chem. Ref. Data* **1992**, *21* (4), 793.

(7) Tang, I. N.; Munkelwitz, H. R.; Wang, N. Water Activity Measurements with Single Suspended Droplets: The NaCl-H2O and KCl-H2O Systems. *J. Colloid Interface Sci.* **1986**, *114* (2), 409–415.

(8) Cohen, M. D.; Flagan, R. C.; Seinfeld, J. H. Studies of Concentrated Electrolyte-Solutions Using the Electrodynamic Balance .1. Water Activities for Single-Electrolyte Solutions. *J. Phys. Chem.* **1987**, *91* (17), 4563–4574.

(9) Chan, C. K.; Liang, Z.; Zheng, J.; Clegg, S. L.; Brimblecombe, P. Thermodynamic Properties of Aqueous Aerosols to High Supersaturation: I—Measurements of Water Activity of the System Na+−Cl−−NO−3−SO2−4−H2O at ~ 298.15 K. *Aerosol Sci. Technol.* **1997**, *27* (3), 324–344.

(10) Pilat, M. J.; Ensor, D. S. Plume Opacity and Particulate Mass Concentration. *Atmospheric Environ. 1967* **1970**, *4* (2), 163–173.

(11) Wexler, A. S.; Dutcher, C. S. Statistical Mechanics of Multilayer Sorption: Surface Tension. *J. Phys. Chem. Lett.* **2013**, *4* (10), 1723–1726.

(12) *Water Activity in Foods*; Barbosa-Cnovas, G. V., Fontana, A. J., Schmidt, S. J., Labuza, T. P., Eds.; Blackwell Publishing Ltd: Oxford, UK, 2007.

(13) Debye, P.; Hückel, E. Zur Theorie der Elektrolyte. I. Gefrierpunktserniedrigung und verwandte Erscheinungen [The theory of electrolytes. I. Lowering of freezing point and related phenomena]. *Phys. Z.* **1923**, *24*, 185–206.

(14) *Activity Coefficients in Electrolyte Solutions*, 2nd ed.; Pitzer, K. S., Ed.; CRC Press: Boca Raton, 1991.

(15) Fredenslund, A.; Jones, R. L.; Prausnitz, J. M. Group-Contribution Estimation of Activity Coefficients in Nonideal Liquid Mixtures. *AIChE J.* **1975**, *21* (6), 1086–1099.
(16) Renon, H.; Prausnitz, J. M. Local Compositions in Thermodynamic Excess Functions for Liquid Mixtures. *AIChE J.* **1968**, *14* (1), 135–144.
(17) Chapman, W. G.; Gubbins, K. E.; Jackson, G.; Radosz, M. New Reference Equation of State for Associating Liquids. *Ind. Eng. Chem. Res.* **1990**, *29* (8), 1709–1721.
(18) Tan, S. P.; Adidharma, H.; Radosz, M. Statistical Associating Fluid Theory Coupled with Restricted Primitive Model To Represent Aqueous Strong Electrolytes. *Ind. Eng. Chem. Res.* **2005**, *44* (12), 4442–4452.
(19) Gil-Villegas, A.; Galindo, A.; Jackson, G. A Statistical Associating Fluid Theory for Electrolyte Solutions (SAFT-VRE). *Mol. Phys.* **2001**, *99* (6), 531–546.
(20) Pitzer, K. S.; Simonson, J. M. Thermodynamics of Multicomponent, Miscible, Ionic Systems: Theory and Equations. *J. Phys. Chem.* **1986**, *90* (13), 3005–3009.
(21) Clegg, S. L.; Pitzer, K. S. Thermodynamics of Multicomponent, Miscible, Ionic Solutions: Generalized Equations for Symmetrical Electrolytes. *J. Phys. Chem.* **1992**, *96* (8), 3513–3520.
(22) Clegg, S. L.; Pitzer, K. S.; Brimblecombe, P. Thermodynamics of Multicomponent, Miscible, Ionic Solutions. Mixtures Including Unsymmetrical Electrolytes. *J. Phys. Chem.* **1992**, *96* (23), 9470–9479.
(23) Li, J.; Polka, H.-M.; Gmehling, J. A gE Model for Single and Mixed Solvent Electrolyte Systems. *Fluid Phase Equilibria* **1994**, *94*, 89–114.
(24) Zuend, A.; Marcolli, C.; Luo, B. P.; Peter, T. A Thermodynamic Model of Mixed Organic-Inorganic Aerosols to Predict Activity Coefficients. *Atmospheric Chem. Phys.* **2008**, *8* (16), 4559–4593.
(25) Stokes, R. H.; Robinson, R. A. Ionic Hydration and Activity in Electrolyte Solutions. *J. Am. Chem. Soc.* **1948**, *70* (5), 1870–1878.
(26) Dutcher, C. S.; Ge, X.; Wexler, A. S.; Clegg, S. L. Statistical Mechanics of Multilayer Sorption: Extension of the Brunauer–Emmett–Teller (BET) and Guggenheim–Anderson–de Boer (GAB) Adsorption Isotherms. *J. Phys. Chem. C* **2011**, *115* (33), 16474–16487.
(27) Dutcher, C. S.; Ge, X.; Wexler, A. S.; Clegg, S. L. Statistical Mechanics of Multilayer Sorption: 2. Systems Containing Multiple Solutes. *J. Phys. Chem. C* **2012**, *116* (2), 1850–1864.
(28) Dutcher, C. S.; Ge, X.; Wexler, A. S.; Clegg, S. L. An Isotherm-Based Thermodynamic Model of Multicomponent Aqueous Solutions, Applicable Over the Entire Concentration Range. *J. Phys. Chem. A* **2013**, *117* (15), 3198–3213.
(29) Brunauer, S.; Emmett, P. H.; Teller, E. Adsorption of Gases in Multimolecular Layers. *J. Am. Chem. Soc.* **1938**, *60* (2), 309–319.
(30) Anderson, R. B.; Hall, W. K. Modifications of the Brunauer, Emmett and Teller Equation II [1]. *J. Am. Chem. Soc.* **1948**, *70* (5), 1727–1734.
(31) De Boer, J. H. *The Dynamical Character of Adsorption*; Clarendon Press: Oxford, UK, 1968.

(32) Anderson, R. B. Modifications of the Brunauer, Emmett and Teller Equation [1]. *J. Am. Chem. Soc.* **1946**, *68* (4), 686–691.
(33) Robinson, R. A.; Stokes, R. H. *Electrolyte Solutions*, 2nd rev. ed.; Dover Publications: Mineola, NY, 2002.
(34) Archer, D. G.; Wang, P. The Dielectric Constant of Water and Debye-Huckel Limiting Law Slopes. *J. Phys. Chem. Ref. Data* **1990**, *19*, 371–411.
(35) Abraham, M.-C.; Abraham, M. The Adsorption Theory of Electrolytes and the Volumetric Properties of Some Nitrate-Water Systems. From Fused Salts to Dilute Solutions. *Monatshefte Chem. Chem. Mon.* **1997**, *128* (8-9), 805–826.
(36) Abraham, M.; Abraham, M.-C. Electrolyte and Water Activities in Very Concentrated Solutions. *Electrochimica Acta* **2000**, *46* (1), 137–142.
(37) Ally, M. .; Braunstein, J. Activity Coefficients in Concentrated Electrolytes: A Comparison of the Brunauer-Emmett-Teller (BET) Model with Experimental Values. *Fluid Phase Equilibria* **1996**, *120* (1-2), 131–141.
(38) Zdanovskii, A. B. *Tr Solyanoi Lab Akad Nauk SSSR* **1936**.
(39) Stokes, R. H.; Robinson, R. A. Interactions in Aqueous Nonelectrolyte Solutions. I. Solute-Solvent Equilibria. *J. Phys. Chem.* **1966**, *70* (7), 2126–2131.
(40) Rowland, D.; Königsberger, E.; Hefter, G.; May, P. M. Aqueous Electrolyte Solution Modelling: Some Limitations of the Pitzer Equations. *Appl. Geochem.* **2014**.
(41) Fulton, J. L.; Balasubramanian, M. Structure of Hydronium (H3O+)/chloride (Cl-) Contact Ion Pairs in Aqueous Hydrochloric Acid Solution: A Zundel-like Local Configuration. *J Am Chem Soc* **2010**, *132* (36), 12597–12604.
(42) Baer, M. D.; Fulton, J. L.; Balasubramanian, M.; Schenter, G. K.; Mundy, C. J. Persistent Ion Pairing in Aqueous Hydrochloric Acid. *J Phys Chem B* **2014**.
(43) Fulton, J. L.; Balasubramanian, M. Structure of Hydronium (H3O+)/chloride (Cl-) Contact Ion Pairs in Aqueous Hydrochloric Acid Solution: A Zundel-like Local Configuration. *J Am Chem Soc* **2010**, *132* (36), 12597–12604.
(44) Kemp, D. D.; Gordon, M. S. An Interpretation of the Enhancement of the Water Dipole Moment Due to the Presence of Other Water Molecules. *J. Phys. Chem. A* **2008**, *112* (22), 4885–4894.
(45) Holland, P. W.; Welsch, R. E. Robust Regression Using Iteratively Reweighted Least-Squares. *Commun. Stat. - Theory Methods* **1977**, *6* (9), 813–827.
(46) DuMouchel, W.; O'Brien, F. Integrating a Robust Option into a Multiple Regression Computing Environment. *Comput. Sci. Stat. Proc. 21st Symp. Interface* **1989**.
(47) Scatchard, G.; Hamer, W. J.; Wood, S. E. Isotonic Solutions. I. The Chemical Potential of Water in Aqueous Solutions of Sodium Chloride, Potassium Chloride, Sulfuric Acid, Sucrose, Urea and Glycerol at 25°1. *J. Am. Chem. Soc.* **1938**, *60* (12), 3061–3070.
(48) Ninni, L.; Camargo, M. S.; Meirelles, A. J. A. Water Activity in Polyol Systems. *J. Chem. Eng. Data* **2000**, *45* (4), 654–660.
(49) Strey, R.; Viisanen, Y.; Aratono, M.; Kratohvil, J. P.; Yin, Q.; Friberg, S. E. On the Necessity of Using Activities in the Gibbs Equation. *J. Phys. Chem. B* **1999**, *103* (43), 9112–9116.

(50) Zhu, H. J.; Yuen, C. M.; Grant, D. J. W. Influence of Water Activity in Organic Solvent plus Water Mixtures on the Nature of the Crystallizing Drug Phase .1. Theophylline. *Int. J. Pharm.* **1996**, *135* (1-2), 151–160.
(51) Bubnik, Z.; Kadlec, P. Program for Calculation of Properties of Sucrose and Its Solutions. *Listy Cukrov. Reparske* **1995**, *111* (6), 185–185.
(52) Marcolli, C.; Peter, T. Water Activity in Polyol/water Systems: New UNIFAC Parameterization. *Atmospheric Chem. Phys.* **2005**, *5*, 1545–1555.
(53) Paez, M. S.; Portacio, A. A.; Sanchez, L. C. Estimation of the Activity Coefficients in Liquid Mixtures by Using Volumetric Data. *J. Solut. Chem.* **2012**, *41* (8), 1282–1289.
(54) Guendouzi, M. E.; Dinane, A.; Mounir, A. Water Activities, Osmotic and Activity Coefficients in Aqueous Chloride Solutions atT= 298.15 K by the Hygrometric Method. *J. Chem. Thermodyn.* **2001**, *33* (9), 1059–1072.
(55) Hamer, W. J.; Wu, Y.-C. Osmotic Coefficients and Mean Activity Coefficients of Uni-Univalent Electrolytes in Water at 25°C. *J. Phys. Chem. Ref. Data* **1972**, *1* (4), 1047.
(56) Pitzer, K. S. Thermodynamics of Electrolytes. I. Theoretical Basis and General Equations. *J Phys Chem* **1973**, *77* (Copyright (C) 2012 American Chemical Society (ACS). All Rights Reserved.), 268–277.
(57) Pitzer, K. S.; Mayorga, G. Thermodynamics of Electrolytes. II. Activity and Osmotic Coefficients for Strong Electrolytes with One or Both Ions Univalent. *J. Phys. Chem.* **1973**, *77* (19), 2300–2308.
(58) Czap, A.; Neuman, N. I.; Swaddle, T. W. Electrochemistry and Homogeneous Self-Exchange Kinetics of the Aqueous 12-Tungstoaluminate(5−/6−) Couple. *Inorg. Chem.* **2006**, *45* (23), 9518–9530.
(59) Prue, J. E.; Sherrington, P. J. Test of the Fuoss-Onsager Conductance Equation and the Determination of Ion Sizes in Dimethyl-Formamide. *Trans. Faraday Soc.* **1961**, *57*, 1795–1808.
(60) Clegg, S. L.; Brimblecombe, P.; Liang, Z.; Chan, C. K. Thermodynamic Properties of Aqueous Aerosols to High Supersaturation: II—A Model of the System Na+−Cl−−NO−3−SO2−4−H2O at 298.15 K. *Aerosol Sci. Technol.* **1997**, *27* (3), 345–366.
(61) Kelly, J. T.; Wexler, A. S.; Chan, C. K.; Chan, M. N. Aerosol Thermodynamics of Potassium Salts, Double Salts, and Water Content near the Eutectic. *Atmos. Environ.* **2008**, *42* (16), 3717–3728.
(62) Wishaw, B. F.; Stokes, R. H. The Osmotic and Activity Coefficients of Aqueous Solutions of Ammonium Chloride and Ammonium Nitrate at 25. *Trans. Faraday Soc.* **1953**, *49*, 27.
(63) Chan, C. K.; Flagan, R. C.; Seinfeld, J. H. Water Activities of NH4NO3/(NH4)2SO4 Solutions. *Atmospheric Environ. Part Gen. Top.* **1992**, *26* (9), 1661–1673.
(64) Kirgintsev, A. N.; Lukyanov, A. V. *Russ J Phys Chem* **1965**, *39*, 653–655.
(65) Thomas, A. S.; Elcock, A. H. Molecular Dynamics Simulations of Hydrophobic Associations in Aqueous Salt Solutions Indicate a Connection between Water

Hydrogen Bonding and the Hofmeister Effect. *J. Am. Chem. Soc.* **2007**, *129* (48), 14887–14898.
(66) Romero, C. M.; Páez, M. S. Isopiestic Determination of Osmotic and Activity Coefficients of Aqueous Solutions of Aliphatic Polyols at 298.15K. *Fluid Phase Equilibria* **2006**, *240* (2), 140–143.
(67) Kiyosawa, K. Partial Molar Volumes and Activity-Coefficients of the Water in Aqueous Polyol Solutions and the Osmotic Pressures of These Solutions. *J. Solut. Chem.* **1992**, *21* (4), 333–344.
(68) Rudakov, A. M.; Sergievskii, V. V. Activities of the Components of Glycerol-Water Binary Solutions at 298.15 K. *Russ. J. Phys. Chem.* **2006**, *80* (11), 1804–1808.
(69) Comensana, J. F.; Correa, A.; Sereno, A. M. Water Activity in Sorbitol or Xylitol + Water and Sorbitol or Xylitol + Sodium Chloride + Water Systems at 20C and 35C. *J. Chem. Eng. Data* **2001**, *46* (3), 716–719.
(70) Bonner, O. D. Osmotic and Activity Coefficients of Sodium Chloride-Sorbitol and Potassium Chloride-Sorbitol Solutions at 25C. *J. Solut. Chem.* **1982**, *11* (5), 315–324.
(71) Macaskill, J. B.; Bates, R. G. Osmotic Coefficients and Activity Coefficients of Aqueous Hydrobromic Acid Solutions at 25C. *J. Solut. Chem.* **1983**, *12* (9), 607–619.
(72) Partanen, J. I.; Makkonen, E. K.; Vahteristo, K. P. Re-Evaluation of Activity Coefficients in Dilute Aqueous Hydrobromic and Hydriodic Acid Solutions at Temperatures from 0 to 60 °C. *J. Solut. Chem.* **2013**, *42* (1), 190–210.
(73) Bonner, O. D. Osmotic and Activity Coefficients of Methyl-Substituted Ammonium Nitrates at 298.15 K. *J. Chem. Eng. Data* **1981**, *26* (2), 148–149.
(74) Bonner, O. D. Osmotic and Activity Coefficients of Methyl-Substituted Ammonium Chlorides. *J. Chem. Soc. Faraday Trans. 1 Phys. Chem. Condens. Phases* **1981**, *77* (10), 2515.
(75) Lindenbaum, S.; Boyd, G. E. Osmotic and Activity Coefficients for the Symmetrical Tetraalkyl Ammonium Halides in Aqueous Solution at 25° [1]. *J. Phys. Chem.* **1964**, *68* (4), 911–917.
(76) Wen, W.-Y.; Chen, C.-M. L. Osmotic and Activity Coefficients of Solutions of Diammonium Decahydrodecaborate, Disodium Dodecahydrododecaborate in Water, and of Four Tetraalkylammonium Halides in Water and in Water-d2 at 25.deg. *J. Chem. Eng. Data* **1975**, *20* (4), 384–387.
(77) Shannon, R. D. Revised Effective Ionic Radii and Systematic Studies of Interatomic Distances in Halides and Chalcogenides. *Acta Crystallogr. Sect. A* **1976**, *32* (5), 751–767.
(78) Masterton, W. L.; Bolocofsky, D.; Lee, T. P. Ionic Radii from Scaled Particle Theory of the Salt Effect. *J. Phys. Chem.* **1971**, *75* (18), 2809–2815.
(79) Marcus, Y. Ionic Radii in Aqueous Solutions. *Chem. Rev.* **1988**, *88* (8), 1475–1498.
(80) Cruz, C. N.; Pandis, S. N. A Study of the Ability of Pure Secondary Organic Aerosol to Act as Cloud Condensation Nuclei. *Atmos. Environ.* **1997**, *31* (15), 2205–2214.

(81) Yao, X.; Lau, A. P. S.; Fang, M.; Chan, C. K.; Hu, M. Size Distributions and Formation of Ionic Species in Atmospheric Particulate Pollutants in Beijing, China: 2—dicarboxylic Acids. *Atmos. Environ.* **2003**, *37* (21), 3001–3007.
(82) Hansen, R. S.; Miller, F. A.; Christian, S. D. Activity Coefficients of Components in.the.Systems Water–Acetic Acid, Water–Propionic Acid and Water–n-Butyric Acid at 25°. *J. Phys. Chem.* **1955**, *59* (5), 391–395.
(83) Pirouzi, A.; Nosrati, M.; Haghtalab, A.; Vasheghani-Farahani, E. Experiment and Correlation of Osmotic Coefficient for Aqueous Solution of Carboxylic Acids Using NRTL Nonrandom Factor Model. *Fluid Phase Equilibria* **2012**, *327*, 38–44.
(84) Sebastiani, E.; Lacquaniti, L. Acetic Acid—water System Thermodynamical Correlation of Vapor—liquid Equilibrium Data. *Chem. Eng. Sci.* **1967**, *22* (9), 1155–1162.
(85) Peng, C.; Chow, A. H. L.; Chan, C. K. Hygroscopic Study of Glucose, Citric Acid, and Sorbitol Using an Electrodynamic Balance: Comparison with UNIFAC Predictions. *Aerosol Sci. Technol.* **2001**, *35* (3), 753–758.
(86) Davies, M.; Thomas, D. K. Lsopiestic Studies of Aqueous. Dicarboxylic Acid Solutions. *J. Phys. Chem.* **1956**, *60* (1), 41–44.
(87) Marcolli, C.; Luo, B.; Peter, T. Mixing of the Organic Aerosol Fractions: Liquids as the Thermodynamically Stable Phases. *J. Phys. Chem. A* **2004**, *108* (12), 2216–2224.
(88) Peng, C.; Chan, M. N.; Chan, C. K. The Hygroscopic Properties of Dicarboxylic and Multifunctional Acids: Measurements and UNIFAC Predictions. *Environ. Sci. Technol.* **2001**, *35* (22), 4495–4501.
(89) Maffia, M. C.; Meirelles, A. J. A. Water Activity and pH in Aqueous Polycarboxylic Acid Systems. *J. Chem. Eng. Data* **2001**, *46* (3), 582–587.
(90) Robinson, R. A.; Smith, P. K.; Smith, E. R. B. The Osmotic Coefficients of Some Organic Compounds in Relation to Their Chemical Constitution. *Trans. Faraday Soc.* **1942**, *38*, 63.
(91) Chen, C.-C.; Britt, H. I.; Boston, J. F.; Evans, L. B. Local Composition Model for Excess Gibbs Energy of Electrolyte Systems. Part I: Single Solvent, Single Completely Dissociated Electrolyte Systems. *AIChE J.* **1982**, *28* (4), 588–596.
(92) Mathlouthi, M. X-Ray Diffraction Study of the Molecular Association in Aqueous Solutions of D-Fructose, D-Glucose, and Sucrose. *Carbohydr. Res.* **1981**, *91* (2), 113–123.
(93) O'Brien, J. T.; Prell, J. S.; Bush, M. F.; Williams, E. R. Sulfate Ion Patterns Water at Long Distance. *J. Am. Chem. Soc.* **2010**, *132* (24), 8248 – +.
(94) Ding, Y.; Hassanali, A., A.; Parrinello, M. Anomalous Water Diffusion in Salt Solutions. *PNAS* **2014**, *111* (9), 3310–3315.

Appendix

Sample MATLAB scripts used for model parameterization and processing.

ajSolve.m

A script for function ajSolve which solves for the activity of solute *j* using equation (3.2).

```
1   function [ ajFS,gammaFS,aj,gamma ] = ajSolve( aw,m,v,zz,rho,Cj,nj,j )
2   %AJSOLVE Solves for the activity of substance j.
3   %   m = molality of solutes (array) [m(1),m(2),...,m(j),...,m(N)]
4   %   v = Stoichiometric coeff (array) [v(1),v(2),...,v(j),...,v(N)]
5   %   zz = abs( z- * z+ ) (array)
6   %   rho = Parameter rho (array) [rho(1),...,rho(j),...,rho(N)]
7   %   aw = water activity
8   %   Cj = C_jk (array) for species j
9   %   nj = # sorption layers for species j
10  %   j = index for species j
11
12  Mw = 0.0180152; % Molecular weight of water (kg/mol)
13  Ax = 2.917; % Debye-Huckel coeff. mol frac. @298.15K
14  Ir = zz(j)/2; % Reference Ionic strength, fused salt
15  Ix = 1/2*sum(m.*zz.*v)/(sum(v.*m)+1/Mw); % Ionic strength
16
17  %% Calculate awbar
18  Kw = exp(Ax*(Ix^0.5)*sum(m.*zz.*v./(1+rho.*Ix^0.5))/(sum(v.*m)+1/Mw));
19  awbar = aw/Kw;
20  %% Calculate ajbar
21  ajden = 0;
22  for i = 1:(nj-1),
```

```
23      ajden=ajden+(awbar^i)*(1-Cj(i))*prod(Cj(1:(i-1)));
24    end
25    ajbar = ((1-awbar)/(1-ajden))^v(j);
26    %% Debye Huckel Contribution j
27    if zz(j) == 0,
28      Kj=exp(Ax*Ix^0.5*sum(m.*zz.*v./(1+rho.*Ix^0.5))/(sum(v.*m)+1/Mw));
29    else
30      Kj=exp(-zz(j)*Ax*(2/rho(j)*log((1+rho(j)*Ix^0.5)/(1+rho(j)*Ir^0.5))...
31          +((1-2*Ix/zz(j))/(sum(v.*m)+1/Mw))*...
32          sum(m.*zz.*v./(2*Ix^0.5*(1+rho*Ix^0.5)))))^v(j);
33    end
34    %% Fused Salt or pure liquid solute reference state
35    ajFS = ajbar*Kj*(m(j)/sum(m));
36    gammaFS = ajFS/m(j);
37
38    %% Infinite dilution reference state
39    % If returing wrong values it is because ajbar and Kj are now raised to the
40    % power of v. Though I believe I have corrected for this.
41    if zz(j) == 0, % Nonelectrolyte (Organic)
42      gamma=(ajbar*Kj*(1/(Mw*sum(m.*v)))*prod(Cj(1:(nj-1))));
43    else % Electrolyte
44      gamma=((ajbar*Kj)^(1/v(j))*(1/(Mw*sum(m.*v)))*prod(Cj(1:(nj-1))))/...
45          exp(-zz(j)*Ax*(2/rho(j)*log(1/(1+rho(j)*Ir^0.5))));
46    end
47    aj = gamma*m(j);
48
49    end
```

mjSolve_matrix.m

A script for the function mjSolve_matrix which solves for the molality of solute *j* using equation (2.50) and equation (2.52). The mjSolve_matrix function solves for and returns an arbitrary reference molality. The reference molality, m_{ref}, is related to the solute molality by $m_j=\alpha_j m_{ref}$, where α_j is the molar ratio of the solutes in solution. For the sample code below the input variable "RO" is α_j. For a single solute RO=[1]. For a mixture that is equal parts species A and species B, RO=[1,1]. For a mixture that is two parts species A, one part species B, and three parts species C, RO=[2,1,3]. The input variable "m_data" is mainly for parameterization purposes and should be set equal to unity. The input variable "aw" is the water activity at which you want to solve for solute molality. The script is written to be able to handle a matrix input for "aw" so you can run the script with a single value input for water activity (*aw=[0.5]*), or with a matrix input (*aw=[0.1:0.1:0.9]*).

```
% Peter Ohm, University of Minnesota 6/6/14
function [OUT] = mjSolve_matrix(rho,C_JK,v,n,zz,aw,RO,N,m_data)
%MJSOLVE Solves for a referance molality.
%   As opposed to solving for the molality directly this funciton solves
%   for a referance molality that is related to the  molalities of each
%   component by some user defined ratio, usually the molar ratios of the
%   solutes.
%
%   rho = Parameter rho (array) [rho(1),rho(2)]
%   C_JK = Cell arrray {[C_jk(1)],[C_jk(2)]}
%   v = Stoichiometric coeff (array) [v(1),v(2)]
%   n = number of sorption shells (array) [n(1),n(2)]
%   zz = absolute charge product abs( z- * z+ ) (array)
%   aw = water activity
%   RO = rato of species species1:species2 [species1,species2]
%   N = number fo solutes present


    function [DEN]=molality_function(m)
    %RMBARSUM Sums RO(j)/mjbar
    DEN = 0;
    for j=1:N,
       DEN=DEN+RO(j)./CalcMjbar(rho,C_JK{j},v,n(j),zz,aw,m,RO,N,j);
    end
    DEN = 1./(DEN);
    end

f = @(x) x - molality_function(x);
dx = 0.0000001;
df = @(x) (f(x+dx)-f(x-dx))./(2*dx);
xold = m_data;
```

```
for i = 1:20
    xnew = xold - f(xold)./df(xold);
    xold = abs(xnew);
end
OUT = xold;

% % Below is an alternative method for solving for molality.
% m1=1;m2=0;iteration=0;
% m1 = m_data;
% asdf = 1;
% while(asdf>0.0000000001)%0.0000000000001)
%     m2=m1;
%     m1=1./RMbarSum(m1);
%     assert(iteration<10000,...
%         'Function does not converge, or took too long to converge')
%
%     iteration=iteration+1;
%     asdf = max(abs(m1-m2));
% end
% OUT=m1;
end
function [OUT]=CalcMjbar(rho,C_jk,v,nj,zz,aw,m,RO,N,j)
%CALCMJBAR Calculate the molality of solute j
%   See EQN.27 Dutcher et al. 2013
%   Pass C_jk as the jth element of cell C_JK
Mw=0.0180152; % Molecular weight water (kg/mol)
Ax=2.917; % Debye-Huckel coeff. mol frac @ 298.15K
% Ix = (1/2)* sum(m.*RO.*zz.*v)/(sum(v.*m.*RO)+1/Mw); % Ionic strength
IxTop = 0; IxBottom = 0;
for ii = 1:N
    IxTop = IxTop + RO(ii).*zz(ii).*v(ii);
    IxBottom = IxBottom + RO(ii).*v(ii);
end
Ix = (1/2)* (m.*IxTop)./(m.*IxBottom+1/Mw);
% Calculate the Kw debye-huckel contribution
KwTop = 0;
for ii = 1:N
    KwTop = KwTop + RO(ii).*zz(ii).*v(ii)./(1+rho(ii)*Ix.^0.5);
end
Kw = exp(Ax.*(Ix.^0.5).*(m.*KwTop)./(m.*IxBottom+1/Mw));

awbar=aw./(Kw);
% Calculate molality Using EQN.27 from Dutcher et al. 2013
NumorSum=0;
```

```
76  for i=1:(nj-1),
77     NumorSum=NumorSum+(awbar.^i).*(1-C_jk(i)).*prod(C_jk(1:i-1));
78  end
79  DenomSum=0;
80  for i=1:(nj-2),
81     DenomSum=DenomSum+i*(awbar.^(i-1)).*prod(C_jk(1:i));
82  end
83  Denom=(1-awbar).^2.*(DenomSum)+...
84     (nj-1-(nj-2).*awbar).*awbar.^(nj-2).*prod(C_jk(1:nj-1));
85
86  OUT=((1-awbar)./(Mw.*v(j).*awbar)).*(1-NumorSum)./Denom;
87
88  end
```

osmFitAllC.m

A script for the function osmFitAllC which uses the built-in MATLAB function "nlinfit" to obtain fit values for the adjustable model parameters in an "all C parameter fit". The input parameters "m_data", "aw_data", and "osm_data" are experimental data for solute molality, water activity, and osmotic coefficient. The input parameter zz is the absolute value of the product of ion charges (equal to zero for non-electrolytes). The parameter "v" is stoichiometric coefficient. The input parameter "n" is the number of sorption layers. This function works for both electrolytes and non-electrolytes. For non-electrolytes the fit parameter "rho" is not used.

```
1   function [ rho, C, mse,P0rho ] = osmFitAllC( ...
2       m_data, aw_data, osm_data,zz,v,n,robustoptn )
3   %OSMFITALLC Use nlinfit to calculate values for adjustable fit parameters
4   %   This script fits each C parameter individually.
5   tic
6
7   %% Define constants
8   % Molar mass of water kg/mole
9   Mw = 0.0180152;
10   %Debye-Huckel coefficient (Archer, Wang 1990)
11   Ax = 2.917;
12
13   %% Fitting Parameter Assignment
14   % the values 1 to n-1 are the C values
15   Prho = 1;
16
17   %%
18
19      function [OUTm] = CBETm(P,aw)
20      %CBETm Input water activity, Output molality
21      Cjk = P(2:(n));
22
23      mref = mjSolve_matrix(abs(P(Prho)),{Cjk},v,n,zz,aw,1,1,m_data);
24      OUTm = mref .* 1;
25      end
26      function [OUTosm] = CBETosm(P,aw)
27      %CBETosm input water activity, Output Molality
28      m = CBETm(P,aw);
29      OUTosm = -log(aw)./(Mw*(v.*m));
30      end
31  %%
32
33  P0 = ones(1,n);
34  P0(Prho) = 13;
```

```
35
36
37  options = statset('MaxIter',1000,'Robust',robustoptn,'TolFun',1e-30,...
38      'TolX',1e-30,'FunValCheck','off','Display','iter');
39  [Pfit,r,J,cov,mse] = nlinfit(aw_data,osm_data,@CBETosm,P0,options);
40
41
42  rho = Pfit(Prho);
43  C = Pfit(2:(n));
44  P0rho = P0(Prho);
45
46
47  end
```

osmFitCoul_diss.m

A script for the function osmFitCoul_diss which uses the built-in MATALB function 'nlinfit' to determine the fit values for the adjustable parameters in the model that accounts for partial disassociation of organic acids. The input parameters "m_data", "aw_data", and "osm_data" are experimental data for solute molality, water activity, and osmotic coefficient. The input parameters "nA" and "nO" are the number of sorption layers on the disassociating acid and the non-disassociating acid respectively.

```
1   function [ rhoAcid,rjwAcid,rjwOrg,mujAcid,mujOrg,CjkA,CjkO,mse,P0 ] = ...
2       osmFitCoul_diss(aw_data,osm_data,nA,nO,robustoptn,soluteratio,m_data )
3   %OSMFITCOUL_DISS Parameter fitting for dissociating species
4   tic
5   %% Parameter Assignment
6   % number is what element of the fit parameter array each fit parameter will
7   % be.
8   PrhoA = 1;
9   PrjwA = 2;
10   PrjwO = 3;
11
12   %% Initial variable initialization
13   kb = 1.38*10^-23;   % Boltzmann
14   T = 298.15;         % Temperature
15   pieps4 = 1.113*10^-10;%=4*pi*eps0
16   Mw = 0.0180152;     % Water molar mass
17   D = 3.33564*10^-30; %
18   muw = 2.9;          % Water dipole moment
19   echarge = 1.60218*10^-19; % electron charge
20   rww = 2.1711 * 10^-10 + 0.6489 * 10 ^-10; % water-water distance
21
22   %% Initial Calculations
23   vA = 2;
24   vO = 1;
25   zzA = 1;
26   zzO = 0;
27
28   %% Intermdeiate functions
29       function [OUT] = CalCO(rjw,n)
30       % Calculate energy C parameters for nondisassociating species using the
31       % coulombic relationship as well as the muj rjw relationship from Ohm
32       % et al. 2015.
33       rjw = abs(rjw);
34       muj = 0.1208*(rjw*10^10)^3;
35       C = zeros(1,n-1);
36       for i = 1:(n-1)
```

```
37          C(1,i) = exp( (muj*muw*D^2/(pieps4*(rjw + (i-1)*rww).^3) - ...
38              (muw*muw*D^2)/(pieps4*(i*rww)^3))/ (kb*T) );
39      end
40      OUT = C;
41      end
42
43      function [OUT] = CalCA(rjw,n)
44      % Calculate energy C parameters for disassociating species using the
45      % coulombic relationship
46      rjw = abs(rjw);
47      muj = echarge*(rjw)/D;
48      C = zeros(1,n-1);
49      for i = 1:(n-1)
50          C(1,i) = exp( (muj*muw*D^2/(pieps4*(rjw + (i-1)*rww).^3) - ...
51              (muw*muw*D^2)/(pieps4*(i*rww)^3))/ (kb*T) );
52      end
53      OUT = C;
54      end
55  %%
56      function [OUT] = CBETm(P,aw)
57      % Function for molality in terms of water activity.
58      if length(P)==2
59          CA = CalCA(P(PrjwA),nA);
60          CO = CalCO(P(PrjwA)/2,nO);
61      else
62          CA = CalCA(P(PrjwA),nA);
63          CO = CalCO(P(PrjwO),nO);
64      end
65
66      mref = mjSolve_matrix([0,abs(P(PrhoA))], {CO,CA}, [vO,vA], [nO,nA],...
67          [zzO,zzA], aw, soluteratio, 2,m_data);
68      OUT = [mref.*soluteratio(1),mref.*soluteratio(2)];
69      end
70  %%
71      function [OUT] = CBETosm(P,aw)
72      % Function of osmotic coefficient in terms of water activity
73      m = CBETm(P,aw);
74      OUT = -log(aw)./(Mw*( vO*m(:,1) + vA*m(:,2)));
75      end
76
77  %%
78
79  % Initial parameter guess P0 = [rhoAcid,rjwAcid,rjwOrg]
80
```

```
% P0 = [0.687176,6.4*10^-10,8.6*10^-11];
P0 = [5,8*10^-10,4*10^-10];
% P0 = [5,2*10^-10,1*10^-10];

options = statset('MaxIter',1000,'Robust',robustoptn,'TolFun',1e-30,...
    'TolX',1e-30,'FunValCheck','off','Display','iter');
% Use nlinfit to optimize fit parameters.
[Pfit,r,J,cov,mse] = nlinfit(aw_data,osm_data,@CBETosm,P0,options);

% Assign the fit values to the corresponding variable.
    if length(P0)==2
        rhoAcid = abs(Pfit(PrhoA));
        rjwAcid = abs(Pfit(PrjwA));
        rjwOrg  = abs(Pfit(PrjwA)/2);
        CjkA = CalCA(rjwAcid,nA);
        CjkO = CalCO(rjwOrg,nO);
    else
        rhoAcid = abs(Pfit(PrhoA));
        rjwAcid = abs(Pfit(PrjwA));
         rjwOrg  = abs(Pfit(PrjwO));
         CjkA = CalCA(rjwAcid,nA);
         CjkO = CalCO(rjwOrg,nO);
     end


mujAcid = echarge*(rjwAcid)/D;
mujOrg  = 0.1208*(rjwOrg*10^10)^3;

toc


end
```

osmFitCoulEle.m

A script for the function osmFitCoulEle which uses the built-in MATALB function 'nlinfit' to determine the fit values for the adjustable parameters in the Coulombic model for electrolytes. The input parameters "m_data", "aw_data", and "osm_data" are experimental data for solute molality, water activity, and osmotic coefficient. The input parameters "r1" and "r2" are the ionic radii of the cation and anion.

```
1    function [ rho, rjw, Cjk, mse, P0 ] = osmFitCoulEle(
m_data,aw_data,osm_data,zz,v,n,r1,r2 )
2    %OSMFITCOUL Fitting function for electrolytes
3    %   Fitting function for electrolytes using a coulombic relation for the
4    %   definition of the energy C parameters.
5    %   The fit parameters are rho and rjw.
6    tic
7    %% Constants
8    kb = 1.38*10^-23;   % Boltzmann Constant
9    T = 298.15;         % Temperature
10   pieps4 = 1.113*10^-10; % 4 * pi * epsilon0
11   Mw = 0.0180152;     % Molar mass of water kg/mol
12   Ax = 2.917;         % Debye-Huckel coefficient from Dutcher et al. 2013
13   D = 3.33564*10^-30;
14   echarge = 1.60218*10^-19; % Charge of electron
15
16   muw = 2.9;          % Dipole moment of water
17   rww = 2.1711*10^-10 + 0.6489*10^-10; % Size of water (m)
18
19   %% Index value for fit parameters
20   Prho = 1;
21   Prjw = 2;
22
23   %% Calculate Ionic Strength
24   Ix = 0.5.*m_data.*zz.*v./(v*m_data+1/Mw);
25   %% Define needed intermediate funcitons
26       function [OUT] = Kw(rho)
27       % Calculate the debye-huckel term for water.
28       rho = abs(rho);
29       OUT = exp((2*Ax*Ix.^1.5)./(1+rho*Ix.^(0.5)));
30       end
31
32       function [OUT] = calC(rjw)
33       % Calculate the energy C parameters
34       rjw = abs(rjw);
35       muj = echarge*(r1/2+r2/2+rjw)/D;
36       C = zeros(1,n-1);
```

```
37      for i = 1:(n-1)
38          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40          C(1,i) = exp( (muj*muw*D^2/...
41              (pieps4*(rjw+(i-1)*rww+(r1+r2)/2+rww/2)^3)-...
42              (muw*muw*D^2)/(pieps4*(i*rww)^3))/(kb*T) );
43      end
44      OUT = C;
45      end
46  %%
47      function [OUT] = CBETm(P,aw)
48      % Function to calculate molality
49      C = calC(P(Prjw));
50      awbar = aw./Kw(P(Prho));
51
52      % Following is Eq. 19 from Dutcher et al. 2013
53      %Numerator
54      mNum = 0;
55      for i = 1:(n-1)
56          mNum = mNum + (awbar.^i).*(1-C(i)).*prod(C(1:(i-1)));
57      end
58      mNum = 1 - mNum;
59
60      %Denomiator
61      sum1 = 0;
62      for i = 1:(n-2)
63          sum1 = sum1 + i*awbar.^(i-1).*prod(C(1:i));
64      end
65      mDen = (1-awbar).^2.*sum1 + ...
66          ((n-1)-(n-2).*awbar).*awbar.^(n-2).*prod(C(1:(n-1)));
67      % output of molality
68      OUT = (1-awbar).*mNum./(Mw.*v.*awbar.*mDen);
69      end
70  %%
71      function [OUT] = CBETosm(P,aw)
72      % Equation for osmotic coefficient in terms of aw and molality
73      OUT = (-log(aw)./(Mw*v*CBETm(P,aw)));
74      end
75  %%
76
77  % Initial fit parameter guess
78  % P0 = [10,(r1+r2)/2];
79  P0 = [1,4*10^-10];
80
```

```
options = statset('MaxIter',1000,'Robust','on', 'TolFun',1e-300,...
    'TolX',1e-300,'FunValCheck','off','Display','iter');

% Use of matlab's nlinfit nonlinear solver to estimate parameters
[Pfit,r,J,cov,mse] = nlinfit(aw_data,osm_data,@CBETosm,P0,options);
%%
% Return Parameters as function output
rho = abs(Pfit(Prho));
rjw = abs(Pfit(Prjw));
Cjk = calC(rjw);


rho = 0.969076;
rjw = 6.18991*10^-10;
Cjk = calC(rjw);

toc
end
```

osmFitCoulOrg.m

A script for the function osmFitCoulOrg which uses the built-in MATALB function 'nlinfit' to determine the fit values for the adjustable parameters in the Coulombic model for organics. The input parameters "m_data", "aw_data", and "osm_data" are experimental data for solute molality, water activity, and osmotic coefficient.

```
1   function [ muj, rjw, Cjk, mse, P0 ] = osmFitCoulOrg( m_data,aw_data,osm_data,v,n)
2   %OSMFITCOUL Fitting function for Organics
3   %   Fitting function for organics using a coulombic relation for the
4   %   definition of the energy C parameters.
5   %   The fit parameters are muj and rjw.
6   tic
7   %% Constants
8   kb = 1.38*10^-23;   % Boltzmann Constant
9   T = 298.15;         % Temperature
10  pieps4 = 1.113*10^-10; % 4 * pi * epsilon0
11  Mw = 0.0180152;     % Molar mass of water kg/mol
12  Ax = 2.917;         % Debye-Huckel coefficient from Dutcher et al. 2013
13  D = 3.33564*10^-30;
14  echarge = 1.60218*10^-19; % Charge of electron
15
16  muw = 2.9;          % Dipole moment of water
17  rww = 2.1711*10^-10 + 0.6489*10^-10; % Size of water (m)
18
19  %% Index value for fit parameters
20  Pmuj = 1;
21  Prjw = 2;
22
23  % Prjw = 1;
24
25  %%
26      function [OUT] = calC(rjw,muj)
27      % Function for calculating C parameters
28      rjw = abs(rjw);
29  %     muj = 0.1208*(rjw*10^10)^3;
30      C = zeros(1,n-1);
31      for i = 1:(n-1)
32          C(1,i)= exp( (muj*muw*D^2/(pieps4*(rjw + (i-1)*rww)^3)-...
33              (muw*muw*D^2)/(pieps4*(i*rww)^3))/ (kb*T) );
34      end
35      OUT = C;
36      end
37  %%
38      function [OUT] = CBETm(P,aw)
```

```
    C = calC(P(Prjw),P(Pmuj));
%     C = calC(P(Prjw),1);
    awbar = aw;
    % The following is eq. 19 from Dutcher et al. 2013
    %Numerator
    mNum = 0;
    for i = 1:(n-1)
        mNum = mNum + (awbar.^i).*(1-C(i)).*prod(C(1:(i-1)));
    end
    mNum = 1 - mNum;
    %Denomiator
    sum1 = 0;
    for i = 1:(n-2)
        sum1 = sum1 + i*awbar.^(i-1).*prod(C(1:i));
    end
    mDen = (1-awbar).^2.*sum1 + ...
        ((n-1)-(n-2).*awbar).*awbar.^(n-2).*prod(C(1:(n-1)));
    % output of molality
    OUT = (1-awbar).*mNum./(Mw.*v.*awbar.*mDen);
    end
%%
    function [OUT] = CBETosm(P,aw)
    % Function for osmotic coefficient in terms of aw and molality (CBETm)
    OUT = (-log(aw)./(Mw*v*CBETm(P,aw)));
    end
%%

% Initial parameter guess
P0 = [2,5e-10];
% P0 = [10,9e-10];
P0 = [16,5e-10];
% P0 = [80e-10];

options = statset('MaxIter',1000,'Robust','on', 'TolFun',1e-5000,...
    'TolX',1e-300,'FunValCheck','off','Display','iter');

% Use nlinfit to estimate parameters
[Pfit,r,J,cov,mse] = nlinfit(aw_data,osm_data,@CBETosm,P0,options);


%%
% Return Parameters as function output
muj = Pfit(Pmuj);
rjw = abs(Pfit(Prjw));
```

```
83  Cjk = calC(rjw,muj);
84
85  %
86  % rjw = abs(Pfit(Prjw));
87  % muj = 0.1208*(rjw*10^10)^3;
88  % Cjk = calC(rjw,muj);
89
90  end
```

osmFitPowerLaw.m

A script for the function osmFitCoulPowerLaw which uses the built-in MATALB function 'nlinfit' to determine the fit values for the adjustable parameters in the power law model. The input parameters "m_data", "aw_data", and "osm_data" are experimental data for solute molality, water activity, and osmotic coefficient. The input parameter zz is the absolute value of the product of ion charges (equal to zero for non-electrolytes). The parameter "v" is stoichiometric coefficient. The input parameter "n" is the number of sorption layers.

This function works for both electrolytes and non-electrolytes. For non-electrolytes the fit parameter "rho" is unused.

```
1   function [rho,C,p, mse, P0 ] = osmFitPowerLaw( ...
2       m_data, aw_data, osm_data,zz,v,n, robustoptn )
3   %OSMFITPOWERLAW Summary of this function goes here
4   %   Detailed explanation goes here
5   tic
6
7   %% Define constants
8   % Molar mass of water kg/mole
9   Mw = 0.0180152;
10   %Debye-Huckel coefficient (Archer, Wang 1990)
11   Ax = 2.917;
12
13   %% Fitting Parameter Assignment
14   Pp = 1;
15   PC1 = 2;
16   Prho = 3;
17
18   Cjk = ones(1,(n-1));
19   %%
20
21       function [OUTm] = CBETm(P,aw)
22       %CBETm Input water activity, Output molality
23       Cjk(1) = abs(P(PC1));
24       for i = 2:(n-1)
25           Cjk(i) = (i/n)^(P(Pp));
26       end
27       mref = mjSolve_matrix(abs(P(Prho)),{Cjk},v,n,zz,aw,1,1,m_data);
28       OUTm = mref .* 1;
29       end
30       function [OUTosm] = CBETosm(P,aw)
31       %CBETosm input water activity, Output Molality
32       m = CBETm(P,aw);
33       OUTosm = -log(aw)./(Mw*(v.*m));
```

```
34      end
35  %%
36
37  P0 = [-1,4,13];
38
39  options = statset('MaxIter',1000,'Robust',robustoptn,'TolFun',1e-30,...
40      'TolX',1e-30,'FunValCheck','off','Display','iter');
41  [Pfit,r,J,cov,mse] = nlinfit(aw_data,osm_data,@CBETosm,P0,options);
42
43
44  rho = Pfit(Prho);
45  C1 = abs(Pfit(PC1));
46  p = Pfit(Pp);
47  % C = zeros(1:(n-1));
48  C(1) = C1;
49  for ii = 2:(n-1)
50      C(ii) = (ii/n)^(p);
51  end
52
53  end
```

Acid_fit_script.m

Script for setting up and running the fitting function osmFitCoul_diss for use with partially disassociating organic acids. The required additional scripts to run Acid_fit_script.m are mjSolve_matrix.m and osmFitCoul_diss.m as well as a function to calculate the mse of the fit.

```
clear
% Acid_fit_script.m
% Peter Ohm
% University of Minnesota

%% Fit information
% type of fit
typeoffit = 'Acid dissociation'; % Name of fit type
robustoptn = 'on'; % Turn robust fitting 'on' or 'off'

 % Variable initialization
 Mw = 0.0180152;

 %% Load data from datafile
 % Name of species to be fit. Should match datafile name. If the species is
 % NaCl then the data file should be NaCl.m
 % The values that are stored in the datafile should be molality (x) and
 % osmotic coefficient (Q)

 %\\\\\\\\\\\\\\\\\\\\\\\\Species\\\\\\\\\\\\\\\\\\\\\\\\%
 species = 'MalonicAcid';
 %\\\\\\\\\\\\\\\\\\\\\\\Parameter n\\\\\\\\\\\\\\\\\\\\\\\%
 % Number of sorption layers
 nO = 6; % n sorption shells for nondisassociating component
 nA = 6; % n sorption shells for disassociating component
 n = [nO,nA];

 % Absolute value Charge Product
 zzO = 0; % nondisassociating component is organic, no charge
 zzA = 1; % disassociating component is treated as 1:1 electrolyte
 zz = [zzO,zzA];

 % Stoichiometric Coeff
 vO = 1; % nondisassociating stays as one organic molecule
 vA = 2; % disassociating gives two "ions"
 v = [vO,vA];

 % Location of .m file
```

```
dataloc = 'F:\MATLAB\ActivityCoeff\Data_Organics\';
datafile = strcat(dataloc,species,'.m');
run(datafile)       % Q = Osmotic coefficient % x = molality

m_data = x; % Store molality data
osm_data = Q; % Store osmotic coeff data
% Calculate water activity from data. No disassociation is assumed
aw_data = exp(-Q.*1.*Mw.*x);
% Calculate mole fraction of solute from data. No disassociation is assumed
xmf_data = x./(x+1/Mw);

%% Determine solute ratio (degree of disassociation)
soluteratio = [1,0.5];

%% Model Fit ////////////////////////////////////////////////////////////////////////////
% Note: the values of v and zz are not inputs to the osmFitCoul_diss
% function and must be changed manually in the osmFitCoul_diss.m file.
[ rhoAcid, rjwAcid, rjwOrg, mujAcid, mujOrg, CjkA, CjkO, mse, P0 ] = ...
    osmFitCoul_diss(aw_data,osm_data,nA,nO,robustoptn,soluteratio,m_data);

% rho for nondisassociating is zero, rho for disassociating from fit
rho = [0,rhoAcid];
% Store C parameters as a cell array.
CJK = {CjkO,CjkA};

%% Generate model prediction

% Array of water activity across which prediction will be made
aw = 0.1:0.001:0.999;
% initialize matrix sizes
m = zeros(size(aw)); % Molality reference
mtot = zeros(size(aw)); % total molality of solution

% for ii = 1:length(aw)
%     m(ii) = mjSolve_matrix(rho,CJK,v,n,zz,aw(ii),soluteratio,length(n),1);
%     for jj = 1:length(n)
%         mtot(ii) = mtot(ii) + m(ii)*soluteratio(jj)*v(jj);
%     end
% end

m = mjSolve_matrix(rho,CJK,v,n,zz,aw,soluteratio,length(n),1);
for jj = 1:length(n)
    mtot = mtot + m.*soluteratio(jj).*v(jj);
end
```

```
83
84  msecalc = mseCalc_dicarb(osm_data,m_data,v,zz,n,CJK{1},CJK{2},rho,...
85      soluteratio);
86
87  %% Figure Generation
88  fig = figure;
89  set(fig,'Position',[400 100 1200 800]);
90
91  subplot(1,3,1);%%%%%%%%%%%%%%%%%%%%%%%%  Left Plot
92  plt = plot(xmf,osm,xmf_data,osm_data,'.');
93  set(plt(1),'LineWidth', 2);
94  set(plt(2),'MarkerSize',10);
95  axis([0 max(xmf_data) (min(osm_data)-0.1) (max(osm_data)+0.1)])
96  title(sprintf(strcat(species,sprintf(', nO=%d, nA=%d',nO,nA))))
97  xlabel('molefraction')
98  ylabel('osmotic coeff \phi')
99
100 subplot(1,3,2);%%%%%%%%%%%%%%%%%%%%%%%%  Mid Plot
101 plt = plot(xmf,osm,xmf_data,osm_data,'.');
102 set(plt(1),'LineWidth', 2)
103 title(sprintf(strcat(species,sprintf(', nO=%d, nA=%d',nO,nA))))
104 xlabel('molefraction')
105 ylabel('osmotic coeff \phi')
106
107 subplot(1,3,3);%%%%%%%%%%%%%%%%%%%%%%%%  Right Plot
108 title(sprintf(strcat(typeoffit,', Robust:',robustoptn)));
109 txstr(1) = {sprintf('Ratio Org:Acid   %g:%g',soluteratio)};
110 txstr(2) = {sprintf('nOrg = %d'     ,nO)};
111 txstr(3) = {sprintf('mujOrg = %g'   ,mujOrg)};
112 txstr(4) = {sprintf('rjwOrg = %g'   ,rjwOrg)};
113 txstr(5) = {sprintf('nAcid = %d'    ,nA)};
114 txstr(6) = {sprintf('mujAcid = %g'  ,mujAcid)};
115 txstr(7) = {sprintf('rjwAcid = %g'  ,rjwAcid)};
116 txstr(8) = {sprintf('rhoAcid = %g'  ,rhoAcid)};
117 txstr(9) = {sprintf('P0 = [%g, %g, %g]',P0(1),P0(2),P0(3))};
118 txstr(10) = {sprintf('msenorm = %g',msecalc)};
119 txstr(11) = {sprintf('msematlab = %g',mse)};
120
121 text(0.05,0.5,txstr);
122
123 %Save Figure Results
124 dirname = strcat('./Results/',species);
125 nid = sprintf('_nO%d_nA%d',nO,nA);
126 if isequal(exist(strcat('./Results/',species),'dir'),7)
```

```
127     fprintf(strcat('./Results/',species,'\n'))
128   else
129     mkdir(strcat('.\Results\',species))
130     fprintf(strcat('./Results/',species,'\n'))
131   end
132   figname = strcat(dirname,'/',species,nid,'.fig');
133   saveas(fig,figname)
134   figname = strcat(dirname,'/',species,nid,'.jpg');
135   set(fig,'PaperPositionMode','auto')
136   print(fig,'-djpeg',figname)%saveas(fig,figname)
137   %%
```

Fit_AllC_PowerLaw.m

Script for setting up and running the fitting function osmFitAllC and osmFitPowerLaw for use with the "all C" and "power law" models. The required additional scripts to run Fit_AllC_PowerLaw.m are mjSolve_matrix.m, osmFitPowerLaw.m and osmFitAllC.m as well as a function to calculate the mse of the fit.

```
clear
% Fit_AllC_PowerLaw.m
% Peter Ohm
% University of Minnesota

%% Constants
% Molar mass of water kg/mol
Mw = 0.0180152;

typeoffit1 = 'Power Law';
typeoffit2 = 'All C';
robustoptn = 'on';
%% Load data from datafile
% Name of species to be fit. Should match datafile name. If the species is
% NaCl then the data file should be NaCl.m
% The values that are stored in the datafile should be molality (x) and
% osmotic coefficient (Q)

%\\\\\\\\\\\\\\\\\\\\\\\\Species\\\\\\\\\\\\\\\\\\\\\\\\%
species = 'CitricAcid';
%\\\\\\\\\\\\\\\\\\\\\\\\Parameter n\\\\\\\\\\\\\\\\\\\\\\\\%
n = 7; % Number of sorption layers

% Charge Product
zz = 0;
% Stoichiometric Coeff
v = 1;

% Location of .m datafile
dataloc = 'F:\MATLAB\ActivityCoeff\Data_Organics\';
% dataloc = 'F:\MATLAB\ActivityCoeff\Data_Single_Salts\';
datafile = strcat(dataloc,species,'.m');
run(datafile)       % Q = Osmotic coefficient % x = molality

m_data = x;
osm_data = Q;

% Calculate water activity from data
```

```
39  aw_data = exp(-Q.*v.*Mw.*x);
40  % Calculate mole fraction from data
41  xmf_data = x./(x+1/Mw);
42
43  osm_data = osm_data(xmf_data<0.5);
44  m_data = m_data(xmf_data<0.5);
45  aw_data = aw_data(xmf_data<0.5);
46  xmf_data = xmf_data(xmf_data<0.5);
47
48  %% Fit data to model
49  % osmFitPowerLaw used the built in function nlinfit
50  % rho is the rho parameter
51  % Cjk are the energy parameters of the sorption layers
52  % P0 is the initial guess for the nlinfit function
53  % mse is the mean square error returned by nlinfit
54
55  [rho,Cjk,P,mse,P0] = osmFitPowerLaw(m_data,aw_data,osm_data,zz,v,n,...
56     robustoptn);
57
58  mse_norm = mseCalc(osm_data,m_data,v,zz,n,Cjk,rho);
59
60
61  %% Generate model prediction
62
63  aw_model = 0.01:0.001:0.999;
64  m_model = ones(size(aw_model));
65
66  m_model = mjSolve_matrix(rho,{Cjk},v,n,zz,aw_model,1,1,m_model);
67
68  xmf_model = m_model./(m_model+1/Mw);
69  osm_model = -log(aw_model)./(Mw*m_model*v);
70
71  xmf_model_pl = xmf_model;
72  osm_model_pl = osm_model;
73
74  %% Figure Generation
75  fig = figure;
76  set(fig,'Position',[400 0 1200 800]);
77
78  subplot(2,3,4);%%%%%%%%%%%%%%%%%%%%%%%%  Left Plot
79  plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
80  set(plt(1),'LineWidth', 2);
81  set(plt(2),'MarkerSize',10);
82  axis([0 max(xmf_data) (min(osm_data)-0.1) (max(osm_data)+0.1)])
```

```
83  title(sprintf(strcat(species,sprintf(', n=%d',n))))
84  xlabel('molefraction')
85  ylabel('osmotic coeff \phi')
86
87  subplot(2,3,5);%%%%%%%%%%%%%%%%%%%%%%%%%%%%   Mid Plot
88  plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
89  set(plt(1),'LineWidth', 2)
90  title(sprintf(strcat(species,sprintf(', n=%d',n))))
91  xlabel('molefraction')
92  ylabel('osmotic coeff \phi')
93
94  subplot(2,3,6);%%%%%%%%%%%%%%%%%%%%%%%%%%%%    Right Plot
95  title(sprintf(strcat(typeoffit1,', Robust:',robustoptn)));
96  txstr(1) = {sprintf('n = %d'     ,n)};
97  txstr(2) = {sprintf('P = %g'   ,P)};
98  txstr(3) = {sprintf('C1 = %g'   ,Cjk(1))};
99  txstr(4) = {sprintf('rho = %g', rho)};
100  txstr(5) = {sprintf('P0 = [%g, %g, %g]',P0(1),P0(2),P0(3))};
101  txstr(7) = {sprintf('msenorm = %g',mse_norm)};
102  txstr(8) = {sprintf('msematlab = %g',mse)};
103
104  text(0.05,0.5,txstr);
105
106
107  % Save Figure Results
108  dirname = strcat('./Results/',species);
109  nid = sprintf('_n%d',n);
110  if isequal(exist(strcat('./Results/',species),'dir'),7)
111     fprintf(strcat('./Results/',species,'\n'))
112  else
113     mkdir(strcat('.\Results\',species))
114     fprintf(strcat('./Results/',species,'\n'))
115  end
116  figname = strcat(dirname,'/',species,nid,'.fig');
117  saveas(fig,figname)
118  figname = strcat(dirname,'/',species,nid,'.jpg');
119  set(fig,'PaperPositionMode','auto')
120  print(fig,'-djpeg',figname)%saveas(fig,figname)
121
122  %%//////////////////////////////////////////////////////////////////////////
123  %%//////////////////////////////////////////////////////////////////////////
124  %%//////////////////////////////////////////////////////////////////////////
125  %% All C fit
126  %%//////////////////////////////////////////////////////////////////////////
```

```
%%//////////////////////////////////////////////////////////////////////////

%% Fit data to model
% osmFitAllC used the built in function nlinfit
% rho is the rho parameter
% Cjk are the energy parameters of the sorption layers
% P0 is the initial guess for the nlinfit function
% mse is the mean square error returned by nlinfit

[rho,Cjk,mse,P0rho]=osmFitAllC(m_data,aw_data,osm_data,zz,v,n,robustoptn);

mse_norm = mseCalc(osm_data,m_data,v,zz,n,Cjk,rho);


%% Generate model prediction

aw_model = 0.01:0.001:0.999;
m_model = ones(size(aw_model));

m_model = mjSolve_matrix(rho,{Cjk},v,n,zz,aw_model,1,1,m_model);

xmf_model = m_model./(m_model+1/Mw);
osm_model = -log(aw_model)./(Mw*m_model*v);

xmf_model_ac = xmf_model;
osm_model_ac = osm_model;

%% Figure Generation
% fig = figure;
% set(fig,'Position',[400 100 1200 800]);

subplot(2,3,1);%%%%%%%%%%%%%%%%%%%%%%%%  Left Plot
plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
set(plt(1),'LineWidth', 2);
set(plt(2),'MarkerSize',10);
axis([0 max(xmf_data) (min(osm_data)-0.1) (max(osm_data)+0.1)])
title(sprintf(strcat(species,sprintf(', n=%d',n))))
xlabel('molefraction')
ylabel('osmotic coeff \phi')

subplot(2,3,2);%%%%%%%%%%%%%%%%%%%%%%%%   Mid Plot
plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
set(plt(1),'LineWidth', 2)
title(sprintf(strcat(species,sprintf(', n=%d',n))))
```

```
xlabel('molefraction')
ylabel('osmotic coeff \phi')

subplot(2,3,3);%%%%%%%%%%%%%%%%%%%%%%%    Right Plot
title(sprintf(strcat(typeoffit2,', Robust:',robustoptn)));
clear txstr
txstr(1) = {sprintf('n = %d'     ,n)};
txstr(2) = {sprintf('msenorm = %g',mse_norm)};
txstr(3) = {sprintf('msematlab = %g',mse)};
txstr(4) = {sprintf('rho = %g',rho)};
for i = 1:(n-1)
   txstr(i+4) = {sprintf('C%d = %g',i,Cjk(i))};
end

txstr(n+5) = {sprintf('P0 for rho = %g',P0rho)};

text(0.05,0.5,txstr);


% Save Figure Results
dirname = strcat('./Results/',species);
nid = sprintf('_n%d',n);
if isequal(exist(strcat('./Results/',species),'dir'),7)
   fprintf(strcat('./Results/',species,'\n'))
else
   mkdir(strcat('.\Results\',species))
   fprintf(strcat('./Results/',species,'\n'))
end
figname = strcat(dirname,'/',species,nid,'.fig');
saveas(fig,figname)
figname = strcat(dirname,'/',species,nid,'.jpg');
set(fig,'PaperPositionMode','auto')
print(fig,'-djpeg',figname)%saveas(fig,figname)

% %% Create Data File
% fid = fopen(strcat(species,'_data.dat'),'w');
% for ii = 1:length(xmf_data)
%     fprintf(fid,'%f\t%f\n',xmf_data(ii),osm_data(ii));
% end
% fclose(fid);
% fid = fopen(strcat(species,'_AllC.dat'),'w');
% for ii = 1:length(xmf_model_ac)
%     fprintf(fid,'%f\t%f\n',xmf_model_ac(ii),osm_model_ac(ii));
% end
```

```
215  % fclose(fid);
216  % fid = fopen(strcat(species,'_PowerLaw.dat'),'w');
217  % for ii = 1:length(xmf_model_pl)
218  %     fprintf(fid,'%f\t%f\n',xmf_model_pl(ii),osm_model_pl(ii));
219  % end
220  % fclose(fid);
```

FitElectrolyte.m

Script for setting up and running the fitting function osmFitCoulEle for use with the Coulombic model and electrolytes. The required additional scripts to run FitElectrolyte.m are mjSolve_matrix.m and osmFitCoulEle.m as well as a function to calculate the mse of the fit.

```
clear
% electrolyteFit.m
% Peter Ohm
% University of Minnesota
% For Fitting of Electrolytes using the coulombic model
%% Constants
Mw = 0.0180152; % Molar mass of water kg/mol
%% Load data from datafile
% Name of species to be fit. Should match datafile name. If the species is
 % NaCl then the data file should be NaCl.m
 % The values that are stored in the datafile should be molality (x) and
 % osmotic coefficient (Q)

 %\\\\\\\\\\\\\\\\\\\\\\\\Species\\\\\\\\\\\\\\\\\\\\\\\\%
 species = 'NaCl';
 %\\\\\\\\\\\\\\\\\\\\\\Parameter n\\\\\\\\\\\\\\\\\\\\\\%
 n = 3;

 % Radius Cation
 r1 = 0.21 * 10^-10;
 % Radius Anion
 r2 = 1.96 * 10^-10;

 r1 = 0; r2 = 0;

 % Charge Product
 zz = 1;
 % Stoichiometric Coeff
 v = 2;

 % Location of .m datafile
 dataloc = 'F:\MATLAB\ActivityCoeff\Data_Single_Salts_More\';
 datafile = strcat(dataloc,species,'.m');
 run(datafile)       % Q = Osmotic coefficient % x = molality

 m_data = x;
 osm_data = Q;

```

```
% Calculate water activity from data
aw_data = exp(-Q.*v.*Mw.*x);
% Calculate mole fraction from data
xmf_data = x./(x+1/Mw);

%% Fit data to model
% osmFitCoul used the built in function nlinfit
% rho is the rho parameter
% rjw is the interspacial distance
% Cjk are the energy parameters of the sorption layers
% P0 is the initial guess for the nlinfit function
% mse is the mean square error returned by nlinfit

[rho,rjw,Cjk,mse,P0]=osmFitCoulEle(m_data,aw_data,osm_data,zz,v,n,r1,r2);

%% Post fit processing

% Generate the model prediciton
aw_model = 0.01:0.001:0.999;
m_model = ones(size(aw_model));
m_model = mjSolve_matirx(rho,{Cjk},v,n,zz,aw_model,1,1,m_model);
xmf_model = m_model./(m_model+1/Mw);
osm_model = -log(aw_model)./(Mw*m_model*v);

% Calculate a normalized mean square errors
mse_norm = mseCalc(osm_data,m_data,v,zz,n,Cjk,rho);


%% Plot model prediciton versus
fig = figure;
set(fig,'Position',[400 50 1200 400]);

subplot(1,3,1);%%%%%%%%%%%%%%%%%%%%%  Left Plot
plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
set(plt(1),'LineWidth', 2);
set(plt(2),'MarkerSize',10);
axis([0 max(xmf_data) (min(Q)-0.1) (max(Q))+0.1])
title(sprintf(strcat(species,sprintf(', n=%d, mse=%g',n,mse))))
xlabel('molefraction')
ylabel('osmotic coeff \phi')

subplot(1,3,2);%%%%%%%%%%%%%%%%%%%%%   Mid Plot
plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
set(plt(1),'LineWidth', 2)
```

```
83  title(sprintf(strcat(species,sprintf(', n=%d, mse=%g',n,mse))))
84  xlabel('molefraction')
85  ylabel('osmotic coeff \phi')
86
87  subplot(1,3,3);%%%%%%%%%%%%%%%%%%%%    Right Plot
88  txstr(1) = {sprintf('rho = %g',rho)};
89  txstr(2) = {sprintf('rjj = %g',rjw)};
90  txstr(3) = {sprintf('n = %g',n)};
91  txstr(4) = {sprintf('r1 = %g',r1)};
92  txstr(5) = {sprintf('r2 = %g',r2)};
93  txstr(6) = {sprintf('P0 = [%g, %g]',P0(1),P0(2))};
94  txstr(7) = {sprintf('mseNorm = %g',mse_norm)};
95  text(0.01,0.5,txstr);
96
97  %% File Management, save plots
98
99  dirname = strcat('./Results/',species);
100  nid = sprintf('_n%d',n);
101  if isequal(exist(strcat('./Results/',species),'dir'),7)
102      fprintf(strcat('./Results/',species,'\n'))
103  else
104      mkdir(strcat('.\Results\',species))
105      fprintf(strcat('./Results/',species,'\n'))
106  end
107  figname = strcat(dirname,'/',species,nid,'.fig');
108  saveas(fig,figname)
109  figname = strcat(dirname,'/',species,nid,'.jpg');
110  set(fig,'PaperPositionMode','auto')
111  print(fig,'-djpeg',figname)%saveas(fig,figname)
```

FitOrganic.m

Script for setting up and running the fitting function osmFitCoulOrg for use with the Coulombic model and non-electrolytes. The required additional scripts to run FitOrganic.m are mjSolve_matrix.m and osmFitCoulOrg.m as well as a function to calculate the mse of the fit.

```
clear
% organicFit.m
% Peter Ohm
% University of Minnesota
% For fitting organics using the coulombic model
%% Constants
Mw = 0.0180152; % Molar mass of water mol/kg
%% Load data from datafile
% Name of species to be fit. Should match datafile name. If the species is
 % NaCl then the data file should be NaCl.m
 % The values that are stored in the datafile should be molality (x) and
 % osmotic coefficient (Q)

 %\\\\\\\\\\\\\\\\\\\\\\\\Species\\\\\\\\\\\\\\\\\\\\\\\\%
 species = 'CitricAcid';
 %\\\\\\\\\\\\\\\\\\\\\\Parameter n\\\\\\\\\\\\\\\\\\\\\\%
 n =9;

 % Charge Product
 zz = 0; % Organics are neutral
 % Stoichiometric Coeff
 v = 1; % Organics don't dissociate

 % Location of .m datafile
 dataloc = 'F:\MATLAB\ActivityCoeff\Data_Organics\';
 datafile = strcat(dataloc,species,'.m');
 run(datafile)       % Q = Osmotic coefficient % x = molality

 m_data = x;
 osm_data = Q;

 % Calculate water activity from data
 aw_data = exp(-Q.*v.*Mw.*x);
 % Calculate mole fraction from data
 xmf_data = x./(x+1/Mw);

 %% Fit data to model
 % osmFitCoul used the built in function nlinfit
```

```
% rho is the rho parameter
% rjw is the interspacial distance
% Cjk are the energy parameters of the sorption layers
% P0 is the initial guess for the nlinfit function
% mse is the mean square error returned by nlinfit

[muj,rjw,Cjk,mse,P0]=osmFitCoulOrg(m_data,aw_data,osm_data,v,n);

%% Post fit processing

% rho only appears with electrolyte solutions. In this code it is set to
% zero in order to use the same equations that are used for electrolytes.
rho = 0;
% Generate the model prediciton
aw_model = 0.01:0.001:0.999;
m_model = ones(size(aw_model));
m_model = mjSolve_matirx(rho,{Cjk},v,n,zz,aw_model,1,1,m_model);
xmf_model = m_model./(m_model+1/Mw);
osm_model = -log(aw_model)./(Mw*m_model*v);

% Calculate a normalized mean square errors
mse_norm = mseCalc(osm_data,m_data,v,zz,n,Cjk,rho);


%% Plot model prediciton versus raw data
fig = figure;
set(fig,'Position',[400 50 1200 400]);

subplot(1,3,1);%%%%%%%%%%%%%%%%%%%%  Left Plot
plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
set(plt(1),'LineWidth', 2);
set(plt(2),'MarkerSize',10);
axis([0 max(xmf_data) (min(Q)-0.1) (max(Q))+0.1])
title(sprintf(strcat(species,sprintf(', n=%d, mse=%g',n,mse))))
xlabel('molefraction')
ylabel('osmotic coeff \phi')

subplot(1,3,2);%%%%%%%%%%%%%%%%%%%%   Mid Plot
plt = plot(xmf_model,osm_model,xmf_data,osm_data,'.');
set(plt(1),'LineWidth', 2)
title(sprintf(strcat(species,sprintf(', n=%d, mse=%g',n,mse))))
xlabel('molefraction')
ylabel('osmotic coeff \phi')

```

```
subplot(1,3,3);%%%%%%%%%%%%%%%%%%%%%%%%    Right Plot
txstr(1) = {sprintf('muj = %g',muj)};
txstr(2) = {sprintf('rjw = %g',rjw)};
txstr(3) = {sprintf('n = %g',n)};
% if length(P0)~=2
%     P0(2) = 0;
% end
txstr(6) = {sprintf('P0 = [%g, %g]',P0(1),P0(2))};
txstr(7) = {sprintf('mseNorm = %g',mse_norm)};
text(0.01,0.5,txstr);

%% File Management, save plots

dirname = strcat('./Results/',species);
nid = sprintf('_n%d',n);
if isequal(exist(strcat('./Results/',species),'dir'),7)
   fprintf(strcat('./Results/',species,'\n'))
 else
    mkdir(strcat('.\Results\',species))
    fprintf(strcat('./Results/',species,'\n'))
 end
 figname = strcat(dirname,'/',species,nid,'.fig');
 saveas(fig,figname)
 figname = strcat(dirname,'/',species,nid,'.jpg');
 set(fig,'PaperPositionMode','auto')
 print(fig,'-djpeg',figname)%saveas(fig,figname)



%% Create Data File
% fid = fopen(strcat(species,'_data.dat'),'w');
% for ii = 1:length(xmf_data)
%     fprintf(fid,'%f\t%f\n',xmf_data(ii),osm_data(ii));
% end
% fclose(fid);
% fid = fopen(strcat(species,'_Coul.dat'),'w');
% for ii = 1:length(xmf_model)
%     fprintf(fid,'%f\t%f\n',xmf_model(ii),osm_model(ii));
% end
% fclose(fid);
```

SolveActivity.m

Script showing how to use the function mjSolve_matrix and ajSolve to calculate solute activities. Additional scripts needed are mjSolve_matrix.m and ajSolve.m.

```
% Peter Ohm, University of Minnesota 4/3/2015
%
% This scirpt is an example of how to use the function mjsolve_matirx.m and
% ajSolve to solve for the activities of solutes in solution.
%
% For each solute in the solution you will need the following parameters:
%
% n - the number of sorption shells/hydration layers
% zz - abs((cation charge)*(anion charge)), the absolute value of the
%   product of the ion charges. zz = 0 for nonelectrolytes. zz = 1 for 1:1
%   electrolytes like NaCl. zz = 2 for 2:1 electrolytes like CaCl2.
% v - the number of ions the solute disassociates into. v = 1 for
%   nonelectrolytes. v = 2 for species like NaCl. v = 3 for species like
%   CaCl2.
% rho - the parameter rho specific to the solute. For nonelectrolytes the
%   rho parameter is not used, set rho = 0.
% C - the energy parameters for each (n-1) monolayers. These should be
%   stored as an array/matrix C = [C1,C2,C3,C4...];
%
% in addition, you will also need the molar ratio between each solute in
% the solution. i.e. something like:
%   ratio = [mA/mA, mB/mA, mC/mA,...]
clear
%% Add in the parameters for each species (species A, B, C,..)
N = 2; % N is the number of solute species in solution
%   These parameters are explained above.
nA = 3; %   n value for species A
nB = 4; %   n value for species B
n = [nA,nB]; % store n values as array.

zzA = 1;
zzB = 2;
zz = [zzA,zzB];

vA = 2;
vB = 3;
v = [vA,vB];

rhoA = 13;
rhoB = 13;
```

```
rho = [rhoA,rhoB];

CA = [10,4];
CB = [6,4,2];
C = {CA,CB}; % stored as cell array. Access elements as C{i}

ratio = [1,1]; % molar ratio between solutes. [1,1] is half A and half B.

%% water activity value
aw = 0.5;

% mguess is a guess for the value of mref. If no guess, let mguess = 1. If
% you are using mjsolve to fit data for model parameterization, let mguess
% = data molality.
mguess = 1;

%% mjSolve_matrix
% the funciton mjSolve_matrix returns a reference molality that is related
% to the solute molality through the ratio parameter. ie mA = mref*ratio(1)

% If the molalities of the solutes are already known then they can be put
% in directly and the mjSolve_matirx step can be skipped.

mref = mjSolve_matrix(rho,C,v,n,zz,aw,ratio,N,mguess);

mA = mref*ratio(1); % molality of solute A
mB = mref*ratio(2); % molality of solute B

m(1) = mA;
m(2) = mB;

%% ajSolve
% Solves for both the solute activity and activity coefficient at a fused
% salt (ajFS,gammaFS) reference state and an infinite dilution reference
% state (aj,gamma)
ajFS = zeros(1,N); gammaFS = zeros(1,N); aj = ajFS; gamma = gammaFS;
for j = 1:N
    [ajFS(j),gammaFS(j),aj(j),gamma(j)]=ajSolve(aw,m,v,zz,rho,C{j},n(j),j);
end
```

SolveMolality.m

Script showing how to use the function mjSolve_matrix to calculate solute molalities. Additional scripts needed are mjSolve_matrix.m.

```
% Peter Ohm, University of Minnesota 4/3/2015
%
% This scirpt is an example of how to use the function mjsolve_matirx.m to
% solve for the molalities of solutes in solution.
%
% For each solute in the solution you will need the following parameters:
%
% n - the number of sorption shells/hydration layers
% zz - abs((cation charge)*(anion charge)), the absolute value of the
%   product of the ion charges. zz = 0 for nonelectrolytes. zz = 1 for 1:1
%   electrolytes like NaCl. zz = 2 for 2:1 electrolytes like CaCl2.
% v - the number of ions the solute disassociates into. v = 1 for
%   nonelectrolytes. v = 2 for species like NaCl. v = 3 for species like
%   CaCl2.
% rho - the parameter rho specific to the solute. For nonelectrolytes the
%   rho parameter is not used, set rho = 0.
% C - the energy parameters for each (n-1) monolayers. These should be
%   stored as an array/matrix C = [C1,C2,C3,C4...];
%
% in addition, you will also need the molar ratio between each solute in
% the solution. i.e. something like:
%   ratio = [mA/mA, mB/mA, mC/mA,...]

%% Add in the parameters for each species (species A, B, C,..)
N = 2; % N is the number of solute species in solution
%   These parameters are explained above.
nA = 3; %   n value for species A
nB = 4; %   n value for species B
n = [nA,nB]; % store n values as array.

zzA = 1;
zzB = 2;
zz = [zzA,zzB];

vA = 2;
vB = 3;
v = [vA,vB];

rhoA = 13;
rhoB = 13;
```

```
41  rho = [rhoA,rhoB];
42
43  CA = [10,4];
44  CB = [6,4,2];
45  C = {CA,CB}; % stored as cell array. Access elements as C{i}
46
47  ratio = [1,1]; % molar ratio between solutes. [1,1] is half A and half B.
48
49  %% water activity value
50  aw = 0.5;
51
52  % mguess is a guess for the value of mref. If no guess, let mguess = 1. If
53  % you are using mjsolve to fit data for model parameterization, let mguess
54  % = data molality.
55  mguess = 1;
56
57  %% mjSolve_matrix
58  % the funciton mjSolve_matrix returns a reference molality that is related
59  % to the solute molality through the ratio parameter. ie mA = mref*ratio(1)
60
61  mref = mjSolve_matrix(rho,C,v,n,zz,aw,ratio,N,mguess);
62
63  mA = mref*ratio(1); % molality of solute A
64  mB = mref*ratio(2); % molality of solute B
```