



Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Engenharia Química
Programa de Pós-Graduação em Engenharia Química



TESE DE DOUTORADO

Modelagem e Simulação da Solubilidade de Sais em Sistemas Aquosos com Monoetilenoglicol

José Augusto Furtado de Oliveira

Orientador: Prof. Dr. Osvaldo Chiavone-Filho

Coorientador: Prof. Dr. Afonso Avelino Dantas Neto

Natal

Junho/2014

José Augusto Furtado de Oliveira

Modelagem e Simulação da Solubilidade de Sais em Sistemas Aquosos com Monoetilenoglicol

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Química (PPGEQ), da Universidade Federal do Rio Grande do Norte (UFRN), como parte dos requisitos necessários para a obtenção do título de Doutor em Engenharia Química.

Universidade Federal do Rio Grande do Norte

Centro de Tecnologia

Departamento de Engenharia Química

Programa de Pós-Graduação em Engenharia Química

Orientador: Prof. Dr. Osvaldo Chiavone-Filho

Coorientador: Prof. Dr. Afonso Avelino Dantas Neto

Natal

Junho/2014

Catálogo da Publicação na Fonte.
UFRN / CT / DEQ / PPGEQ
Biblioteca Setorial "Professor Horácio Nicolas Solimo".

Oliveira, José Augusto Furtado de.

Modelagem e Simulação da Solubilidade de Sais em Sistemas Aquosos com Monoetilenoglicol / José Augusto Furtado de Oliveira. - Natal : Junho/2014.
169 p.; il. (algumas color.); 30 cm.

Orientador: Prof. Dr. Osvaldo Chiavone-Filho

Coorientador: Prof. Dr. Afonso Avelino Dantas Neto

Tese de Doutorado - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Departamento de Engenharia Química. Programa de Pós-Graduação em Engenharia Química.

1. Solubilidade - Tese. 2. Monoetilenoglicol (MEG) - Tese. 3. Sal - Tese. 4. Eletrólitos - Tese. 5. Termodinâmica molecular - Tese. 6. Desidratação do gás natural - Tese. 7. Regeneração do MEG - Tese. I. Chiavone-Filho, Osvaldo. II. Dantas Neto, Afonso Avelino. III. Universidade Federal do Rio Grande do Norte Centro de Tecnologia

Departamento de Engenharia Química

Programa de Pós-Graduação em Engenharia Química. IV. Título.

RN/UF/BSEQ

CDU 02:141:005.7

José Augusto Furtado de Oliveira

Modelagem e Simulação da Solubilidade de Sais em Sistemas Aquosos com Monoetilenoglicol

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Química (PPGEQ), da Universidade Federal do Rio Grande do Norte (UFRN), como parte dos requisitos necessários para a obtenção do título de Doutor em Engenharia Química.

Trabalho aprovado. Natal, 02 de Junho de 2014:

Prof. Dr. Osvaldo Chiovone-Filho
Orientador - UFRN

**Prof. Dr. Afonso Avelino Dantas
Neto**
Coorientador - UFRN

Prof. Dr. Edson Luiz Foletto
Membro Externo - UFSM

Prof. Dr. André Luís Novais Mota
Membro Externo - UFRSA

**Prof. Dr. Humberto Neves Maia de
Oliveira**
Membro Interno - UFRN

Dr^a. Dannielle Janainne da Silva
Membro Interno - UFRN

*Dedico este trabalho à minha esposa **Marizete de Melo Oliveira** e filhos,
Jéssica de Melo Oliveira e **Samuel Furtado de Oliveira**,
cuja simples existência faz tudo valer a pena!*

Agradecimentos

Ao Senhor Jesus, em primeiro lugar, por **TUDO!** Sem Ele nada seria possível!

À minha esposa Marizete e meus Filhos, Jéssica e Samuel, pelo **AMOR!**

À minha mãe, Eudézia, pela **EDUCAÇÃO** e **SUPORTE**, além do natural para uma mãe, que foram-me dados!

À Agência Nacional de Petróleo Gás Natural e Biocombustíveis (ANP) pelo **APOIO FINANCEIRO**.

Ao Centro de Pesquisas da Petrobras (CENPES) pela formação do grupo de **PESQUISA** do MEG no Laboratório de Fotoquímica e Equilíbrio de Fases (FOTEQ).

Ao Prof. Osvaldo Chiavone Filho pela **ORIENTAÇÃO** e **DEDICAÇÃO** no desenvolvimento desta tese de doutorado.

Ao Prof. Afonso Avelino Dantas Neto pelo **APOIO** do Programa de Recursos Humanos ANP-14 (PRH ANP-14).

Ao Prof. Eduardo Lins de Barros Neto pelo **INCENTIVO!**

Ao Prof. Humberto Neves Maia de Oliveira pelo **CAFEZINHO!**

Ao Prof. José Romualdo Dantas Vidal e Maria Carlenise Paiva de Alencar Moura pelo **ESTÍMULO** e **AMIZADE!**.

À Maria José de Sousa Brunet pela **PACIÊNCIA** e **RAPIDEZ** nas diversas solicitações de vital importância deste projeto.

Ao aluno de iniciação científica Bruno Dezan Samorini pela **EFICIÊNCIA**.

A TODO o pessoal do Núcleo de Ensino e Pesquisa em Petróleo e Gás (NUPEG) e do laboratório de Fotoquímica e Equilíbrio de Fases (FOTEQ) pelo **COLEGUISMO**.

E a todos que, embora não citados aqui, **COLABORARAM** de forma direta ou indireta para o desenvolvimento desta tese de doutorado. **MUITO OBRIGADO!**

*“Tudo tem o seu tempo determinado, e há
tempo para todo o propósito debaixo do céu.
(Bíblia ACF, Eclesiastes 3:1)*

Resumo

OLIVEIRA, José Augusto Furtado de - Modelagem e Simulação da Solubilidade de Sais em Sistemas Aquosos com Monoetilenoglicol. Tese de Doutorado, Universidade Federal do Rio Grande do Norte (UFRN), Programa de Pós-Graduação em Engenharia Química (PPGEQ), Natal, Brasil. Área de Concentração: Engenharia de Processos.

Orientador: Prof. Dr. Osvaldo Chiavone-Filho

Coorientador: Prof. Dr. Afonso Avelino Dantas Neto

O processamento primário do gás natural em plataformas como a do Campo de Mexilhão (PMXL-1) na Bacia de Santos, onde o Monoetileno glicol (MEG) tem sido utilizado como inibidor da formação de hidratos, apresenta problemas operacionais causados por incrustações de sal na unidade de recuperação do MEG. Pesquisa bibliográfica e análise dos dados de solubilidade de sal em misturas de solventes, a saber: água e MEG, revelam que estudos experimentais estão disponíveis para um número relativamente restrito das espécies iônicas presentes na água produzida, tais como NaCl e KCl. O objetivo deste trabalho foi desenvolver um método de cálculo para a solubilidade de um sal em misturas de solventes, neste caso, para o NaCl ou KCl em misturas aquosas contendo MEG. O método de cálculo estendeu o modelo de Pitzer, com a abordagem de Lorimer, para sistemas aquosos contendo um sal e um outro solvente (MEG). A linguagem Python no ambiente de desenvolvimento integrado (IDE) Eclipse foi usada na criação de aplicativos computacionais. Os resultados indicam a viabilidade da modelagem proposta para uma série de dados de solubilidade do sal (NaCl ou KCl) em misturas de água e MEG. Além disso, o aplicativo desenvolvido em Python tem demonstrado ser adequado para estimação de parâmetros e de simulação.

Palavras-chave: solubilidade. eletrólitos. monoetilenoglicol. mistura de solventes. processamento de gás natural. Python.

Abstract

OLIVEIRA, José Augusto Furtado de - Modeling and Simulation of the Solubility of Salts in Aqueous Systems with Monoethyleneglycol. Doctoral Thesis, Federal University of Rio Grande do Norte (UFRN), Program of Postgraduation in Chemical Engineering (PPGEQ), Natal, Brazil. Concentration Area: Process Engineering.

Supervisor: Prof. Dr. Osvaldo Chiavone-Filho

Joint Supervisor: Prof. Dr. Afonso Avelino Dantas Neto

Primary processing of natural gas platforms as Mexilhão Field (PMXL-1) in the Santos Basin, where monoethylene glycol (MEG) has been used to inhibit the formation of hydrates, present operational problems caused by salt scale in the recovery unit of MEG. Bibliographic search and data analysis of salt solubility in mixed solvents, namely water and MEG, indicate that experimental reports are available to a relatively restricted number of ionic species present in the produced water, such as NaCl and KCl. The aim of this study was to develop a method for calculating of salt solubilities in mixed solvent mixtures, in explantion, NaCl or KCl in aqueous mixtures of MEG. The method of calculating extend the Pitzer model, with the approach Lorimer, for aqueous systems containing a salt and another solvent (MEG). Python language in the Integrated Development Environment (IDE) Eclipse was used in the creation of the computational applications. The results indicate the feasibility of the proposed calculation method for a systematic series of salt (NaCl or KCl) solubility data in aqueous mixtures of MEG at various temperatures. Moreover, the application of the developed tool in Python has proven to be suitable for parameter estimation and simulation purposes.

Keywords: solubility. electrolytes. monoethyleneglycol. mixture of solvents. natural gas processing. Python.

Lista de ilustrações

Figura 1 – Localização do Campo de Mexilhão.	31
Figura 2 – Desenho esquemático do desenvolvimento do Campo de Mexilhão. . . .	32
Figura 3 – Atividade de produção de gás e condensado projetada para o Campo de Mexilhão.	33
Figura 4 – Remoção de bloco de hidratos de tubulação.	34
Figura 5 – Estrutura dos hidrato de metano e etano.	34
Figura 6 – Mecanismos de inibição de hidratos.	35
Figura 7 – Processo Geral de recuperação do MEG.	36
Figura 8 – Unidade de MEG - Campo de Mexilhão.	37
Figura 9 – Sistema de Recuperação - Evaporador à vácuo / <i>Downcomer</i>	38
Figura 10 – Estratégia de abordagem do trabalho desta tese de doutorado.	41
Figura 11 – Esquema de formação de etilenoglicóis.	43
Figura 12 – Representação do Equilíbrio de um eletrólito em mistura de solventes. .	46
Figura 13 – Conceito de solução ideal hipotética.	47
Figura 14 – Fluxograma de obtenção de propriedades termodinâmicas a partir da energia livre de Gibbs.	51
Figura 15 – Fluxograma da metodologia de desenvolvimento e teste do modelo de solubilidade de sais em misturas de solventes.	59
Figura 16 – Programa desenvolvido : Seleção do sistema a ser estudo.	71
Figura 17 – Programa desenvolvido : Relatório gerado.	72
Figura 18 – Programa desenvolvido : Opções para geração de gráficos binários. . .	72
Figura 19 – Programa desenvolvido : Opções para geração de gráficos ternários. .	73
Figura 20 – Programa desenvolvido : Gráficos.	73
Figura 21 – Molalidade do NaCl em H ₂ O versus Temperatura.	75
Figura 22 – Molalidade do NaCl em MEG versus Temperatura.	76
Figura 23 – Isotermas do potencial químico padrão de excesso do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão). .	78
Figura 24 – Isotermas do potencial químico padrão de excesso do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição). .	79
Figura 25 – Isotermas da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão).	80
Figura 26 – Isotermas da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição).	81
Figura 27 – Isotermas do logaritmo de excesso da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão). .	82

Figura 28 – Isotermas do logaritmo de excesso da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição). . .	83
Figura 29 – Isotermas do logaritmo do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão). . .	84
Figura 30 – Isotermas do logaritmo do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição). . .	85
Figura 31 – Isotermas do logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão).	86
Figura 32 – Isotermas do logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição).	87
Figura 33 – Molalidade do KCl em H ₂ O versus Temperatura.	89
Figura 34 – Molalidade do KCl em MEG versus Temperatura.	90
Figura 35 – Isotermas do potencial químico padrão de excesso do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão). . .	92
Figura 36 – Isotermas do potencial químico padrão de excesso do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição). . . .	93
Figura 37 – Isotermas da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão).	94
Figura 38 – Isoterma da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição).	95
Figura 39 – Isotermas do logaritmo de excesso da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão). . .	96
Figura 40 – Isotermas do logaritmo de excesso da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição). . . .	97
Figura 41 – Isotermas do logaritmo do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão). . .	98
Figura 42 – Isotermas do logaritmo do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição). . . .	99
Figura 43 – Isotermas do logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão).	100
Figura 44 – Isotermas do logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição).	101
Figura 45 – Validação do Modelo de Pitzer implementado (γ).	102
Figura 46 – Validação do Modelo de Pitzer implementado (ϕ).	102
Figura 47 – Comparação do Modelo implementado com o Modelo ideal (Equação 2.35). .	103

Lista de tabelas

Tabela 1 – Banco de dados de solubilidade de sal em mistura de solventes.	42
Tabela 2 – Banco de dados de solubilidade de sal em água.	42
Tabela 3 – Características da água de formação do reservatório do campo de Mexilhão.	44
Tabela 4 – Parâmetros da Equação 4.1 obtidos por regressão para o Sistema H ₂ O+NaCl.	74
Tabela 5 – Parâmetros da Equação 4.2 obtidos por regressão para o Sistema MEG+NaCl.	75
Tabela 6 – Parâmetros da Equação 4.3 obtidos por regressão para o Sistema H ₂ O+MEG+NaCl.	77
Tabela 7 – Parâmetros da Equação 4.4 obtidos por regressão para o Sistema H ₂ O+KCl.	88
Tabela 8 – Parâmetros da Equação 4.5 obtidos por regressão para o Sistema MEG+KCl.	89
Tabela 9 – Parâmetros da Equação 4.6 obtidos por regressão para o Sistema H ₂ O+MEG+KCl.	91
Tabela 10 – Comparação do Modelo implementado com o Modelo ideal (Equação 2.35).	103

Lista de abreviaturas e siglas

PPGEQ	Pós-Graduação em Engenharia Química
UFRN	Universidade Federal do Rio Grande do Norte
UFSM	Universidade Federal de Santa Maria
UFERSA	Universidade Federal Rural do Semi-Árido
ANP	Agência Nacional de Petróleo Gás Natural e Biocombustíveis
CENPES	Centro de Pesquisas da Petrobras
FOTEQ	Laboratório de Fotoquímica e Equilíbrio de Fases
PRH	Programa de Recursos Humanos
NUPEG	Núcleo de Ensino e Pesquisa em Petróleo e Gás
MEG	Monoetileno glicol
NaCl	Cloreto de sódio
KCl	Cloreto de potássio
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
PMXL-1	Plataforma número 1 do Campo de Mexilhão na Bacia de Santos

Lista de símbolos

b_i	molalidade de i
m_{solv}	massa do solvente
n_i	número de moles de i
M_i	massa molecular de i
c_i	molaridade de i
V	volume da solução
x_i	fração molar de i (fase líquida)
n	número de moles total na solução
μ_i	potencial químico de i
G	energia livre de Gibbs
T	temperatura
P	pressão
$n_{j \neq i}$	todos os números de moles exceto o referente a i
R	constante universal do gases ideais
$\mu_i^\bullet$	potencial químico de i em um estado padrão apropriadamente escolhido
a_i	atividade de i
γ_i	coeficiente de atividade de i
ξ_i	uma conveniente escala de concentração
μ_i^0	potencial químico racional simétrico de i
$\mu_{i,x}^\nabla$	potencial químico racional assimétrico
$\gamma_{i,x}^*$	coeficiente de atividade racional assimétrico
γ_i^∞	coeficiente de atividade em diluição infinita
b_0	molalidade unitária

$M_{\nu+} X_{\nu-}$	eletrólito eletricamente neutro
ν_+	número de cátions
z_+	carga do cátion
ν_-	número de ânions
z_-	carga do ânion
MX	eletrólito eletricamente neutro (abreviado)
ν	número total de íons
y_i	fração molar de i (fase vapor)
P_i^{sat}	pressão de vapor do componente i puro
H_i	constante de Henry para o componente i
κ^{-1}	comprimento de Debye-Hückel
ε_0	permissividade no vácuo
ε_r	permissividade relativa, usualmente conhecida como constante dielétrica
ρ_{solv}	densidade do solvente
N_A	número de Avogrado
e	carga do elétron
I	força iônica
b	um parâmetro do modelo de Pitzer ($b = 1,2 \text{ kg}^{\frac{1}{2}} \cdot \text{mol}^{-\frac{1}{2}}$)
$B_{\pm}$	parâmetro empírico da extensão de Guggenheim
M^{E}	uma propriedade de excesso
M^{id}	uma propriedade ideal
$\bar{M}_i$	uma propriedade parcial molar de i
$\bar{M}_i^{\text{id}}$	uma propriedade parcial molar ideal de i
G^{E}	energia livre de Gibbs de excesso
V^{E}	volume de excesso
S^{E}	a entropia de excesso

$\bar{G}^E$	energia livre de Gibbs parcial molar de excesso de i
$\hat{f}_i$	fugacidade de i na solução, ou mistura
$f_i^\bullet$	fugacidade de i no estado padrão
f_i^0	fugacidade racional simétrica de i
$H_{i,x}$	constante de Henry racional
$H_{i,b}$	constante de Henry molal
$\left[\ln b_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}}\right]^{\text{id}}$	logaritmo ideal da molalidade do sal MX em água e MEG
$x_{\text{H}_2\text{O}}^{\{\text{H}_2\text{O}+\text{MEG}\}}$	fração molar da água livre de sal
$x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}}$	fração molar do MEG livre de sal
$\ln b_{\text{MX}}^{\{\text{H}_2\text{O}\}}$	logaritmo ideal da molalidade do sal MX em água
$\ln b_{\text{MX}}^{\{\text{MEG}\}}$	logaritmo ideal da molalidade do sal MX em MEG
$\left[\ln b_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}}\right]^E$	logaritmo de excesso da molalidade do sal MX em água e MEG
$b_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}}$	molalidade do sal MX em água e MEG
$b_{\text{MX}}^{\{\text{H}_2\text{O}\}}$	molalidade do sal MX em água
k_S	constante de Setchenov
ϕ	coeficiente osmótico
$a_{\text{H}_2\text{O}}$	atividade da água
$M_{\text{H}_2\text{O}}$	massa molecular da água
$\beta_{MX}^{(0)}$	parâmetro do modelo de Pitzer
$\beta_{MX}^{(1)}$	parâmetro do modelo de Pitzer
Λ_{MMX}	parâmetro do modelo de Pitzer
Λ_{MXX}	parâmetro do modelo de Pitzer
ν	coeficiente estequiométrico do sal
ν_M	coeficiente estequiométrico do cátion
ν_X	coeficiente estequiométrico do ânion
z_M	carga do cátion
z_X	carga do ânion

Sumário

1	INTRODUÇÃO	31
1.1	Atividade de produção no Campo de Mexilhão ¹	31
1.2	Formação de hidratos em tubulações	33
1.3	Estrutura dos hidratos	34
1.4	Inibição da formação de hidratos	35
1.5	Uso do MEG como inibidor de hidratos	36
1.6	Problemas no processo de recuperação do MEG	36
2	REVISÃO BIBLIOGRÁFICA	41
2.1	Dados Experimentais	41
2.2	O monoetilenoglicol	43
2.3	Seleção dos sistemas	43
2.4	Equilíbrio de Fases para Sistemas com Eletrólitos	44
2.4.1	Escalas de concentração	44
2.4.2	Potencial químico	45
2.4.3	Estado padrão	46
2.4.4	Coeficiente de atividade médio iônico e molalidade média iônica	47
2.4.5	Leis limitantes	48
2.4.6	Energia livre de Gibbs para soluções eletrolíticas reais	49
2.4.7	Fugacidade	50
2.5	Modelos Termodinâmicos	51
2.5.1	Ideal	52
2.5.2	Setchenov	52
2.5.3	Pitzer	53
2.6	Outros Modelos Termodinâmicos	54
2.7	Ferramentas Computacionais	56
3	METODOLOGIA	59
3.1	Molalidade e fração molar	60
3.1.1	Cálculo da molalidade do sal na mistura de solventes	60
3.1.2	Cálculo da fração molar do MEG livre de sal na mistura de solventes	60
3.2	Constantes dielétricas	60
3.2.1	Cálculo das constantes dielétricas dos solventes puros	60
3.2.2	Cálculo das constantes dielétricas da mistura de solventes	60
3.3	Densidades	61

¹ As informações presentes nesta seção foram obtidas de [Petrobras \(2007\)](#)

3.3.1	Cálculo das densidades dos solventes puros	61
3.3.2	Cálculo das densidades da mistura de solventes	61
3.4	Coefficiente de atividade	62
3.4.1	Cálculo do coeficiente de atividade do sal pelo Modelo de Pitzer	62
3.4.2	Parâmetros do modelo de Pitzer em água	62
3.4.2.1	Cálculo dos parâmetros do modelo de Pitzer para o NaCl	62
3.4.2.2	Cálculo dos parâmetros do modelo de Pitzer para o KCl	63
3.4.3	Constante de Debye-Hückel	63
3.4.3.1	Cálculo da constante de Debye-Hückel para a água	63
3.4.3.2	Cálculo da constante de Debye-Hückel para o MEG e para a mistura de solventes	63
3.4.4	Cálculo do coeficiente de atividade do sal em MEG	64
3.4.5	Parâmetros do modelo de Pitzer em mistura de solventes	64
3.5	Algoritmos	64
3.5.1	Cálculo das propriedades de excesso do sal em mistura de solventes	65
3.5.2	Cálculo das demais propriedades	67
4	RESULTADOS E DISCUSSÕES	71
4.1	Como os resultados foram calculados através do programa desenvolvido	71
4.2	Sistemas com NaCl	74
4.2.1	Sistemas Binários	74
4.2.1.1	Sistema H ₂ O+NaCl	74
4.2.1.2	Sistema MEG+NaCl	75
4.2.2	Sistemas Ternários (H ₂ O+MEG+NaCl)	77
4.2.2.1	Potencial químico padrão de excesso do NaCl em mistura de solventes (Regressão)	77
4.2.2.2	Potencial químico padrão de excesso do NaCl em mistura de solventes (Predição)	79
4.2.2.3	Molalidade do NaCl em mistura de solventes (Regressão)	80
4.2.2.4	Molalidade do NaCl em mistura de solventes (Predição)	81
4.2.2.5	Logaritmo de excesso da molalidade do NaCl em mistura de solventes (Regressão)	82
4.2.2.6	Logaritmo de excesso da molalidade do NaCl em mistura de solventes (Predição)	83
4.2.2.7	Logaritmo do coeficiente de atividade do NaCl em mistura de solventes (Regressão)	84
4.2.2.8	Logaritmo do coeficiente de atividade do NaCl em mistura de solventes (Predição)	85
4.2.2.9	Logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes (Regressão)	86
4.2.2.10	Logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes (Predição)	87
4.3	Sistemas com KCl	88
4.3.1	Sistemas Binários	88
4.3.1.1	Sistema H ₂ O+KCl	88
4.3.1.2	Sistema MEG+KCl	89
4.3.2	Sistemas Ternários	91

4.3.2.1	Potencial químico padrão de excesso do KCl em mistura de solventes (Regressão)	91
4.3.2.2	Potencial químico padrão de excesso do KCl em mistura de solventes (Predição)	93
4.3.2.3	Molalidade do KCl em mistura de solventes (Regressão)	94
4.3.2.4	Molalidade do KCl em mistura de solventes (Predição)	95
4.3.2.5	Logaritmo de excesso da molalidade do KCl em mistura de solventes (Regressão)	96
4.3.2.6	Logaritmo de excesso da molalidade do KCl em mistura de solventes (Predição)	97
4.3.2.7	Logaritmo do coeficiente de atividade do KCl em mistura de solventes (Regressão)	98
4.3.2.8	Logaritmo do coeficiente de atividade do KCl em mistura de solventes (Predição)	99
4.3.2.9	Logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes (Regressão)	100
4.3.2.10	Logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes (Predição)	101
4.4	Validação do Modelo de Pitzer implementado	102
4.5	Comparação do Modelo implementado com outros modelos	103
4.5.1	Modelo Ideal	103
4.5.2	Modelo UNIQUAC+DH	104
5	CONCLUSÕES	107
	Referências	109
	APÊNDICES	117
	APÊNDICE A – ARTIGO	119
A.1	DE CORRELAÇÃO DE DADOS DE EQUILÍBRIO LÍQUIDO VA- POR PELO MODELO UNIQUAC ORIGINAL PARA SISTEMAS AQUOSOS COM SAL E ÁLCOOL	119
	APÊNDICE B – PROGRAMA COMPUTACIONAL VLE	127
B.1	Interface gráfica do programa VLE	127
B.2	Código fonte do programa VLE desenvolvido para a correlação de dados de Equilíbrio Líquido-Vapor pelo modelo UNIQUAC Original para sistemas aquosos com sal e álcool	129
	APÊNDICE C – PROGRAMA COMPUTACIONAL JAFOSMS	135
C.1	Interface gráfica do programa JAFOSMS	135
C.2	Código fonte do programa JAFOSMS desenvolvido para o cálculo da solubilidade de sal em solução aquosa com MEG	151

CAPÍTULO 1

INTRODUÇÃO

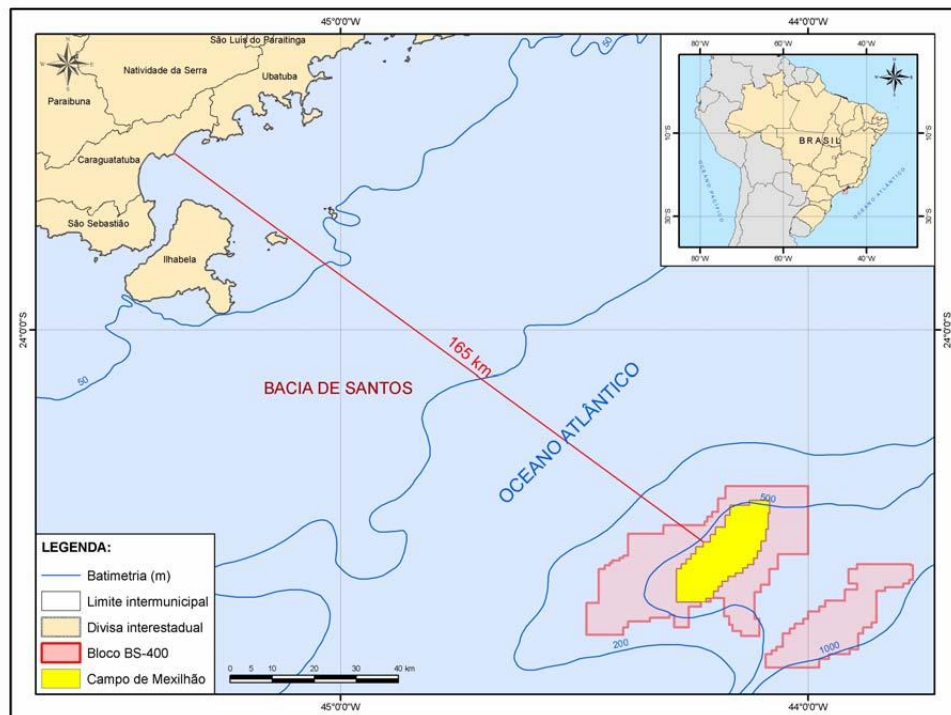
1 INTRODUÇÃO

Na indústria de produção do gás natural, de uma maneira geral, evita-se a formação de hidratos nas tubulações por meio da injeção de MEG na cabeça do poço, principalmente, de monoetilenoglicol (MEG). Embora considere-se que a injeção do MEG resolva os problemas com hidratos na produção, ocorre que nas etapas de recuperação e regeneração do MEG injetado pode ocorrer a formação de precipitados, sendo portanto necessário o conhecimento da solubilidade dos sais presentes na água de produção em função da concentração do MEG para evitar-se estes indesejáveis precipitados.

1.1 Atividade de produção no Campo de Mexilhão¹

Na atividade de produção de gás natural no Brasil tem grande destaque a plataforma PMXL-1 do Campo de Mexilhão que está localizado na Bacia de Santos (Figura 1), sendo a maior unidade fixa da Petrobras e uma das maiores do mundo. A sua operação iniciou-se em 2011. Nela a atividade de produção e escoamento de gás e condensado contempla o

Figura 1 – Localização do Campo de Mexilhão.

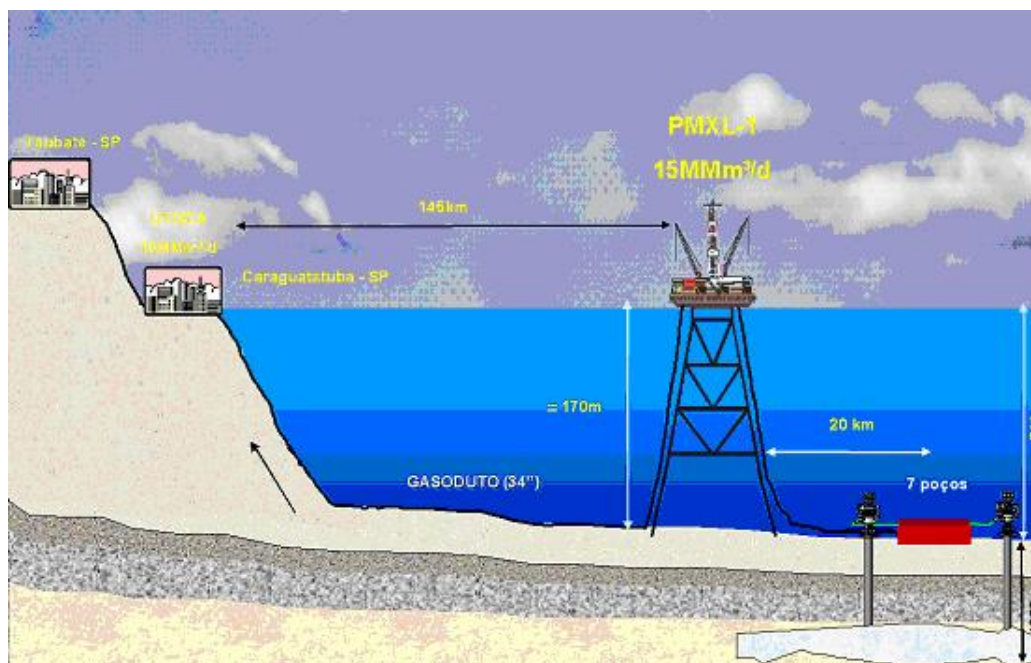


Fonte: Petrobras (2007, p. 5)

¹ As informações presentes nesta seção foram obtidas de Petrobras (2007)

emprego de poços produtores, um sistema de coleta constituído por linhas e estruturas submarinas, uma plataforma marítima e um sistema de escoamento da produção para a costa por meio de duto submarino (Figura 2).

Figura 2 – Desenho esquemático do desenvolvimento do Campo de Mexilhão.



Fonte: Petrobras (2007, p. 24)

A plataforma de produção encontra-se fixada, por meio de jaqueta, ao leito submarino a uma profundidade de 172 metros. Ela está interligada aos poços produtores possuindo uma capacidade de processamento estimada em 15 milhões de m^3/dia de gás e de 3,2 mil m^3/dia para o condensado.

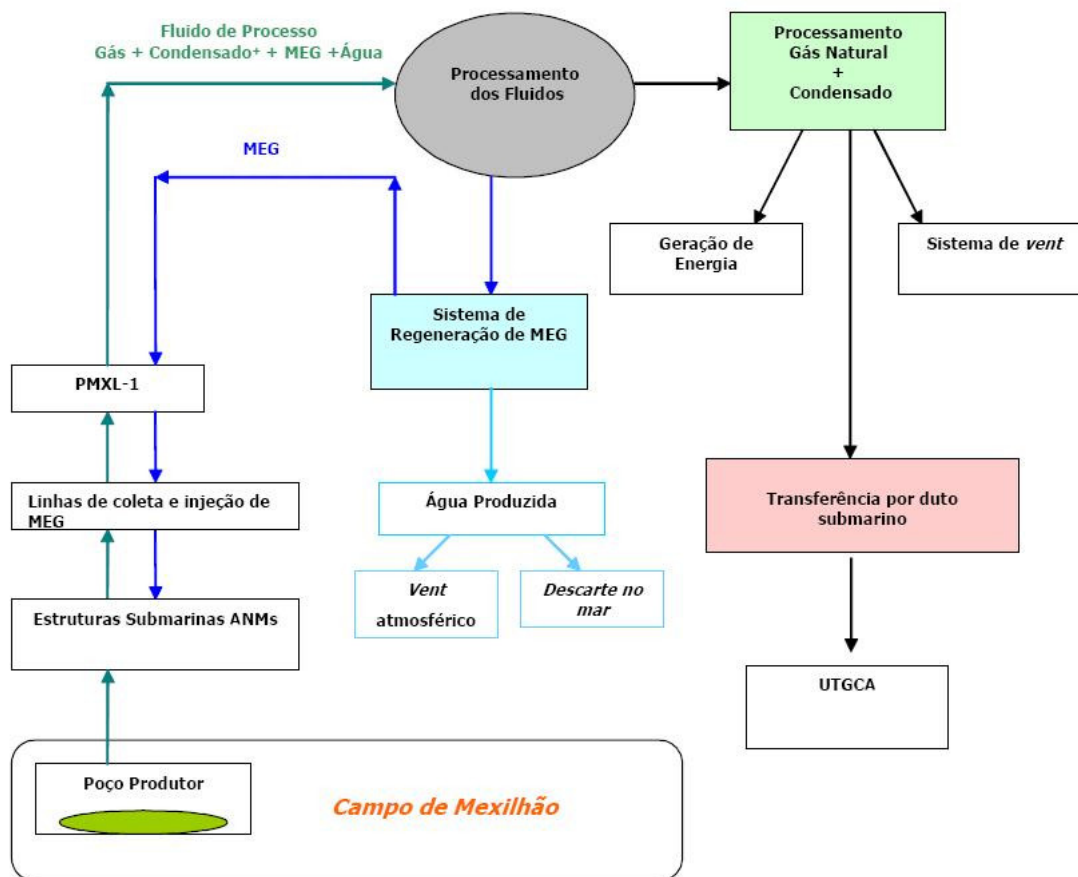
O fluido proveniente do reservatório é coletado através do sistema submarino diretamente para a plataforma. Este é rico em gás, além de condensado e pequena quantidade de água. Na plataforma realiza-se o processamento da produção.

As principais etapas envolvendo o fluido produzido são:

- Coleta do fluido produzido (gás, condensado e água) no poço;
- Separação do gás, condensado e água na plataforma;
- Tratamento do gás e do condensado na plataforma;
- Processamento de parte do gás para suporte ao processo de produção (geração de energia);
- Circulação do MEG pelo sistema submarino de coleta, evitando-se assim entupimentos por formação de hidratos no interior das tubulações, e posterior regeneração do MEG na plataforma.

A atividade de produção de gás e condensado do Campo de Mexilhão está representada pela Figura 3.

Figura 3 – Atividade de produção de gás e condensado projetada para o Campo de Mexilhão.



Fonte: Petrobras (2007, p. 25)

1.2 Formação de hidratos em tubulações

Em condições termodinâmicas favoráveis, de temperatura e pressão, poderá haver a formação de hidratos em tubulações. Estes poderão provocar o bloqueio do fluido em linhas, dutos e equipamentos e, conseqüentemente, interrupção na produção, entre outros problemas.

A obstrução de tubulações de produção pode ser um problema grave com enormes prejuízos e, portanto, deve ser evitado a todo custo.

A interrupção para remoção manual dos blocos de hidratos normalmente demanda diminuição ou até interrupção na produção, além de normalmente exigir procedimentos com operações de risco para a plataforma. O processo de remoção manual e a dimensão do problema pode ser compreendida melhor por meio da visualização das dimensões dos

blocos de hidratos removidos do interior de uma tubulação mostrados na Figura 4(a) e na Figura 4(b).

Figura 4 – Remoção de bloco de hidratos de tubulação.



(a) Sendo removido.



(b) Após a remoção.

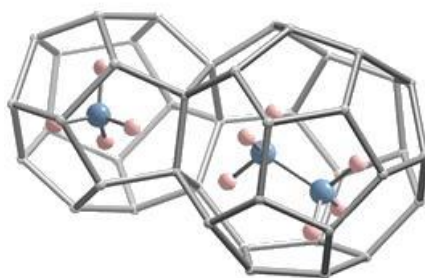
Fonte: Moura (2007, p. 11)

1.3 Estrutura dos hidratos

Conforme Baptista (2007), a atração entre os pólos de cargas opostas de duas moléculas de água distintas origina ligações conhecidas como pontes de hidrogênio. Estas ligações são as mais fortes dentre as ligações intermoleculares. Necessita-se de aproximadamente 5 kcal/mol de energia para romper uma ponte de hidrogênio enquanto que para uma ligação de van der Waals essa energia é de 0,3 kcal/mol. As pontes de hidrogênio contribuem ativamente na energia de dissociação da estrutura dos hidratos.

Na Figura 5 representa-se como uma molécula de hidrocarboneto encontra-se aprisionada por meio de rede de pontes de hidrogênio formadas com outras moléculas de água.

Figura 5 – Estrutura dos hidrato de metano e etano.



Fonte: Statoil

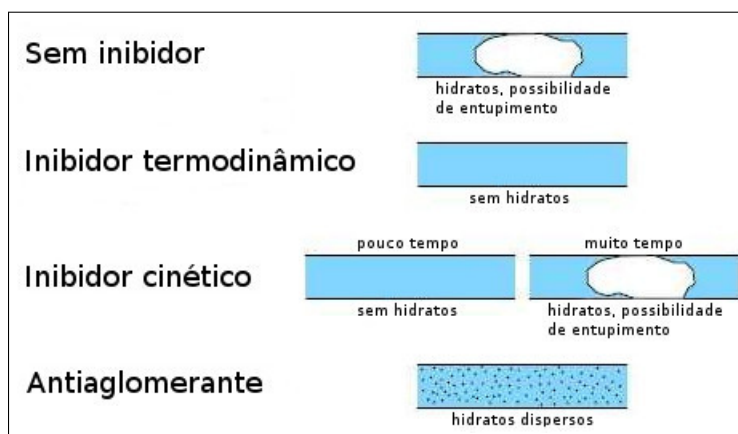
1.4 Inibição da formação de hidratos

Conforme (ANDRADE, 2009) os inibidores da formação de hidratos podem ser classificados em três tipos:

- Inibidores termodinâmicos:** são geralmente sais inorgânicos (NaCl , CaCl_2 , KCl), alcoóis (metanol) e glicóis (MEG) que objetivam reduzir a atividade da água, quantidade de água livre na mistura, e com isso deslocar a curva de equilíbrio de fases, desfavorecendo a formação de hidratos;
- Inibidores cinéticos:** são polímeros solúveis em água e que reagem sinergicamente com glicóis e alcoóis de alto peso molecular. Esses aditivos são capazes de retardar o início da nucleação e diminuem a taxa de crescimento de cristais de hidratos;
- Antiaglomerantes:** são basicamente polímeros e tensoativos, os quais, de modo similar aos inibidores cinéticos, quando utilizados em pequenas quantidades já surtem os efeitos desejados. Sua principal função é retardar a aglomeração de cristais e facilitar o transporte dos núcleos já formados. Não impedem a formação de cristais de hidratos, no entanto, ajudam bastante no transporte dos cristais formados.

Estes inibidores e seus mecanismos estão ilustrados na Figura 6.

Figura 6 – Mecanismos de inibição de hidratos.



Fonte: Andrade (2009, p. 40), adaptada.

As características de alguns destes inibidores de formação de hidratos são:

- Sais inorgânicos:** podem causar corrosão e incrustação;
- Metanol:** alta toxicidade, difícil recuperação, não adequado para injeção contínua;
- Glicóis:** pouco voláteis, fácil recuperação e se concentram na fase aquosa.

1.5 Uso do MEG como inibidor de hidratos

O uso do MEG como inibidor de hidratos, por injeção na cabeça do poço, apresenta vantagens, como por exemplo:

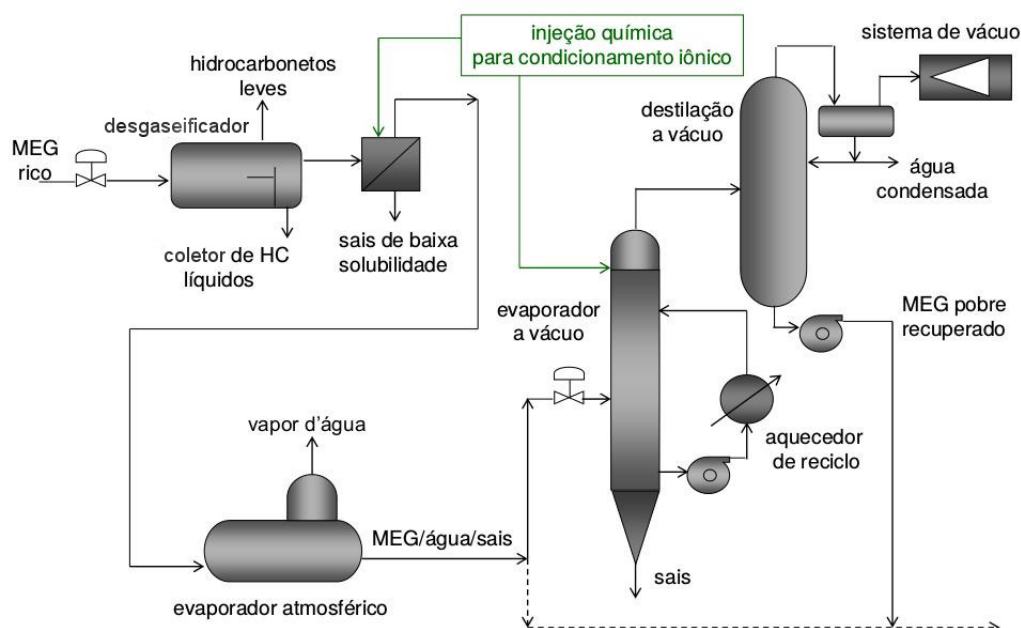
- a) *Menor viscosidade* $\Rightarrow$ maior facilidade de escoamento;
- b) *Fácil recuperação* $\Rightarrow$ menos solúvel em hidrocarbonetos líquidos;
- c) Atua na redução da temperatura de formação de hidratos.

1.6 Problemas no processo de recuperação do MEG

Em plataformas de Gás Natural, o MEG tem sido largamente utilizado como inibidor da formação de hidratos sem grandes problemas. Por outro lado, no processo de recuperação do MEG podem ocorrer problemas causados por incrustações formadas pela precipitação de sais e, sendo assim, faz-se necessário o conhecimento das condições de formação, quantidade e natureza dos respectivos precipitados para poder-se evitá-los.

A [Figura 7](#) representa o processo de recuperação do MEG. O processo, normalmente, consiste de um pré-tratamento do MEG rico (pré-aquecimento do fluido e resfriamento do MEG), vaporização, destilação a vácuo e remoção de sais e outras impurezas. Ou seja, o MEG contaminado com sais e outras impurezas é conduzido a um vaso separador que, operando sob vácuo, vaporiza instantaneamente a mistura, deixando, ao sair, grande parte dos resíduos sólidos que estavam suspensos ou dissolvidos no líquido, sendo estes

Figura 7 – Processo Geral de recuperação do MEG.



Fonte: [Zouain \(2014, p. 59\)](#), adaptada.

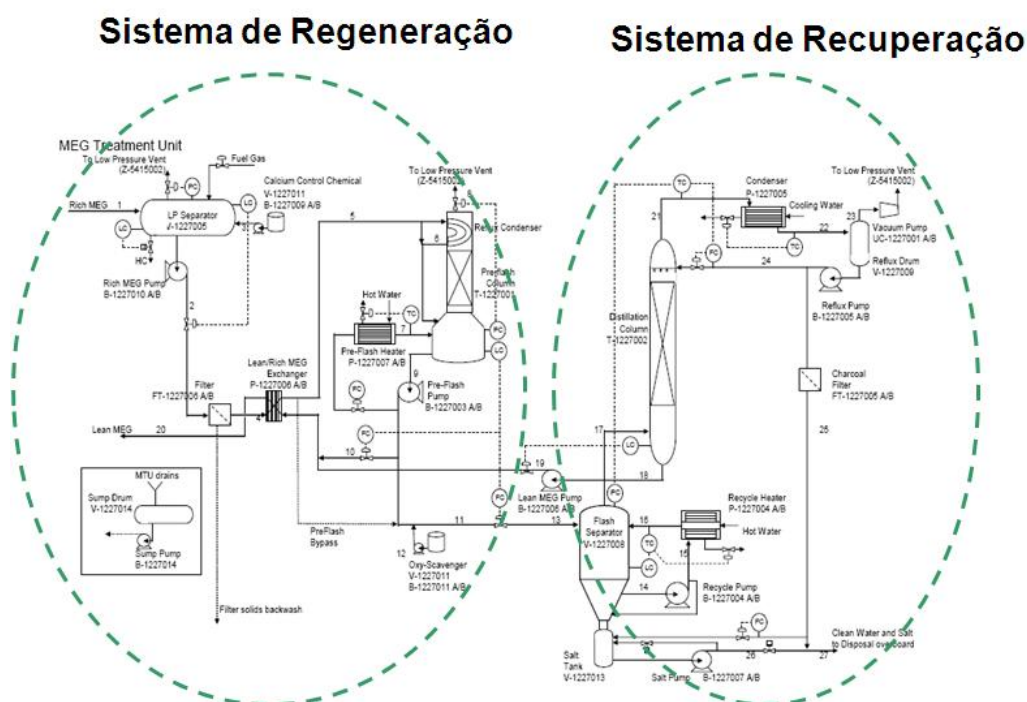
removidos por decantação ou centrifugação. O fluxo líquido, devido à vaporização e ao fato da temperatura de ebulição do MEG ser mais alta que a da água, tem uma redução na quantidade de água presente. Estas etapas, ou seja: aquecimento, vaporização, resfriamento e remoção dos sólidos (decantação, centrifugação), podem ser repetidas visando a maior remoção dos sólidos e diminuição da água na corrente de MEG. A etapa final consiste em passar a corrente líquida com MEG por uma coluna de destilação à vácuo que recupera o MEG limpo e livre de sal.

Com o objetivo meramente ilustrativo, é apresentada a [Figura 8](#) que representa o fluxograma da unidade de regeneração e recuperação do MEG na plataforma PMXL-1. Bem como a [Figura 9](#) que ilustra um protótipo de evaporador à vácuo para soluções salinas de MEG. Este esquema representa um processo novo de regeneração do MEG que se baseia na separação, por densidade, da salmoura do MEG concentrado para reuso na desidratação do MEG.

Como é notório, o conhecimento da solubilidade dos sais nas misturas aquosas com MEG nos equipamentos da unidade de MEG é informação crucial para o monitoramento das formações de incrustação de sais durante o processo.

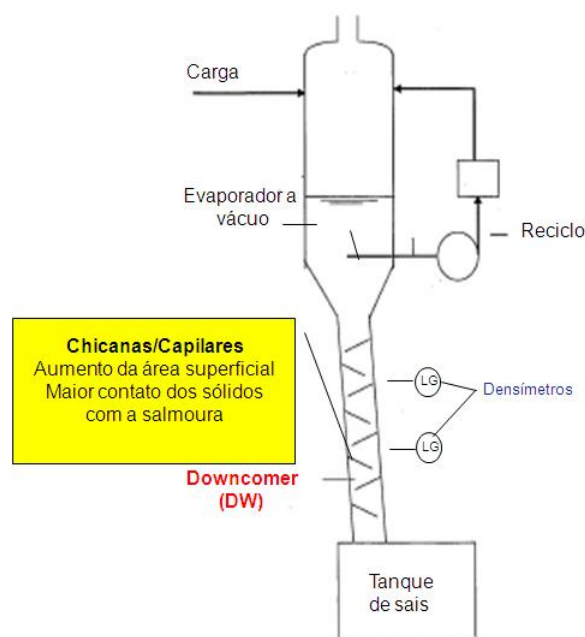
Dados de solubilidade, densidade, pH e condutividade tem sido obtidos da literatura e em especial no laboratório de Fotoquímica e Equilíbrio de Fases (FOTEQ) da Universidade Federal do Rio Grande do Norte (UFRN). Os dados gerados pelo FOTEQ tem por meta

Figura 8 – Unidade de MEG - Campo de Mexilhão.



Fonte: Nazzer e Keogh (2006)

Figura 9 – Sistema de Recuperação - Evaporador à vácuo / *Downcomer*.



Fonte: Nazzer e Keogh (2006)

possibilitar um melhor monitoramento do processo de regeneração do MEG, visando suprir a escassez dessas informações para sistemas aquosos com MEG e sais. Este fato também resulta na demanda de modelos para estes sistemas aquosos com eletrólitos e solvente orgânico, tema científico desta tese de doutorado. Neste sentido, também deve ser ressaltada a importância do projeto no que diz respeito à demanda da indústria do petróleo, em particular o campo de Mexilhão na Bacia de Santos que se destaca por sua significativa reserva.

Esta tese de doutorado visa abordar, através de modelagem termodinâmica, o comportamento dos sistemas aquosos contendo sais e MEG nas condições do processo de Recuperação do MEG. Portanto, a precipitação de sais neste processo, as chamadas incrustações, provocam dificuldades operacionais de maneira similar à formação de hidratos.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

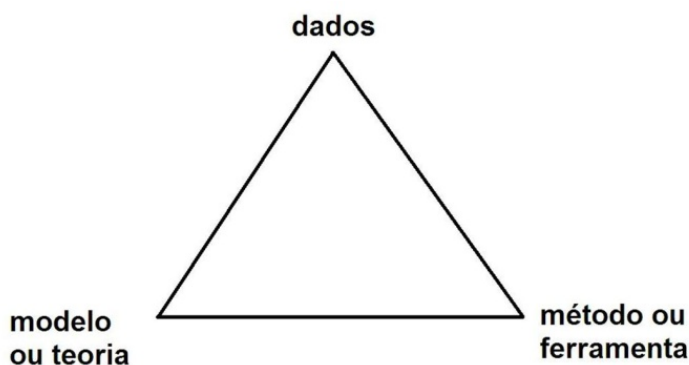
2 REVISÃO BIBLIOGRÁFICA

Estabeleceu-se uma estratégia de abordagem desta tese através da classificação de três tipos de trabalhos, a saber:

- a) dados experimentais;
- b) modelos ou teorias;
- c) métodos e ferramentas de cálculo.

A [Figura 10](#) ilustra essa estratégia. Observa-se a que os dados experimentais estão em nível de destaque (mais elevados), isto reforça a importância da qualidade destes e a prioridade que deve ser-lhes dada.

Figura 10 – Estratégia de abordagem do trabalho desta tese de doutorado.



Fonte: [CERE \(2009\)](#), adaptada.

2.1 Dados Experimentais

A [Tabela 1](#) apresenta um resumo contendo vários dados de solubilidade de sais em misturas de solventes coletados da literatura. Estes estão agrupados numa série sistemática de misturas, seguidas por suas isotermas, número de pontos experimentais e a referência consultada. Os sistemas em destaque (negrito) foram utilizados nesta tese.

Foi observado uma escassez de trabalhos com dados de solubilidade de sal em meio aquoso com MEG. Este fato destaca a importância do trabalho de [Figueiredo et al. \(2014\)](#) desenvolvido pelo grupo de pesquisa do FOTEQ, sendo suas seis isotermas e os 76 pontos experimentais de grande relevância nesta tese.

Outros trabalhos experimentais de sal em meio aquoso com MEG que reportam outras propriedades que não a solubilidade foram encontrados na literatura. [Tsierkezos e](#)

Tabela 1 – Banco de dados de solubilidade de sal em mistura de solventes.

Sistema	Temperatura (K)	Número de Pontos Experimentais	Referência
Água+Etanol+KCl	298,15 323,15 348,15	6 6 6	Chiavone-Filho e Rasmussen (1993)
	298,15 323,15 348,15	9 9 9	Pinho e Macedo (2005)
Água+Etanol+NaBr	298,15 323,15 348,15	9 9 9	Pinho e Macedo (2005)
Água+Etanol+ NaCl	298,15 323,15 348,15	9 9 7	Pinho e Macedo (1996)
Água+Glicerol+Ba(NO ₃) ₂	298,15	6	Kraus, Raridon e Baldwin (1964)
Água+Glicerol+NaCl	298,15	6	Kraus, Raridon e Baldwin (1964)
Água+Glicerol+KCl	298,15	6	Kraus, Raridon e Baldwin (1964)
Água+MEG+Ba(NO ₃) ₂	298,15	7	Kraus, Raridon e Baldwin (1964)
Água+MEG+K ₂ SO ₄	303,15	9	Trimble (1931)
Água+MEG+KBr	303,15	6	Trimble (1931)
Água+MEG+KCl	298,15 323,15 348,15	7 7 7	Chiavone-Filho e Rasmussen (1993)
	298,15	1	Isbin e Kobe (1945)
	298,15	7	Kraus, Raridon e Baldwin (1964)
	303,15	6	Trimble (1931)
	298,15	11	Zhou et al. (2010)
Água+MEG+KI	303,15	6	Trimble (1931)
Água+MEG+NaCl	323,15	6	Baldwin, Raridon e Kraus (1969)
	293,15 318,15 348,15 363,15 383,15 403,15	14 12 12 12 13 13	Figueiredo et al. (2014)
	298,15	1	Isbin e Kobe (1945)
	298,15	7	Kraus, Raridon e Baldwin (1964)
	303,15	6	Trimble (1931)
	298,15	11	Zhou et al. (2010)
Água+Metanol+KCl	298,15 323,15	9 9	Pinho e Macedo (1996)
Água+Metanol+NaBr	298,15 323,15	9 9	Pinho e Macedo (2005)
Água+Metanol+NaCl	298,15 323,15	9 9	Pinho e Macedo (1996)

Fonte: o autor

Molinou (1998) estudaram propriedades termodinâmicas como volume de excesso, viscosidade e refração da água com etilenoglicol à diferentes temperaturas. Braun, Persson e Karlsson (2001) estudaram soluções ternárias de MEG+2-amino-2-metil-1-propanol+água em diferentes concentrações obtendo dados de viscosidade e densidade. Sandengen e Kasa (2006) estudaram as soluções aquosas de MEG na presença de sais com respeito à condutividade e densidade, no sentido de gerarem curvas de referência para determinar as concentrações a partir dessas duas propriedades para sistemas ortogonais.

Uma compilação de dados experimentais de solubilidade de sais é o banco de dados em Linke e Seidell (1958), Linke e Seidell (1965). Nesta compilação também observa-se a carência de dados experimentais para o MEG. Dados de solubilidade dos sais KCl e NaCl em água juntamente com outros de Lide e Haynes (2009) foram utilizados e são resumidos na Tabela 2.

Tabela 2 – Banco de dados de solubilidade de sal em água.

Sistema	Número de Pontos Experimentais	Referência
Água+KCl	12	Lide e Haynes (2009)
	33	Linke e Seidell (1965)
Água+NaCl	12	Lide e Haynes (2009)
	25	Linke e Seidell (1965)

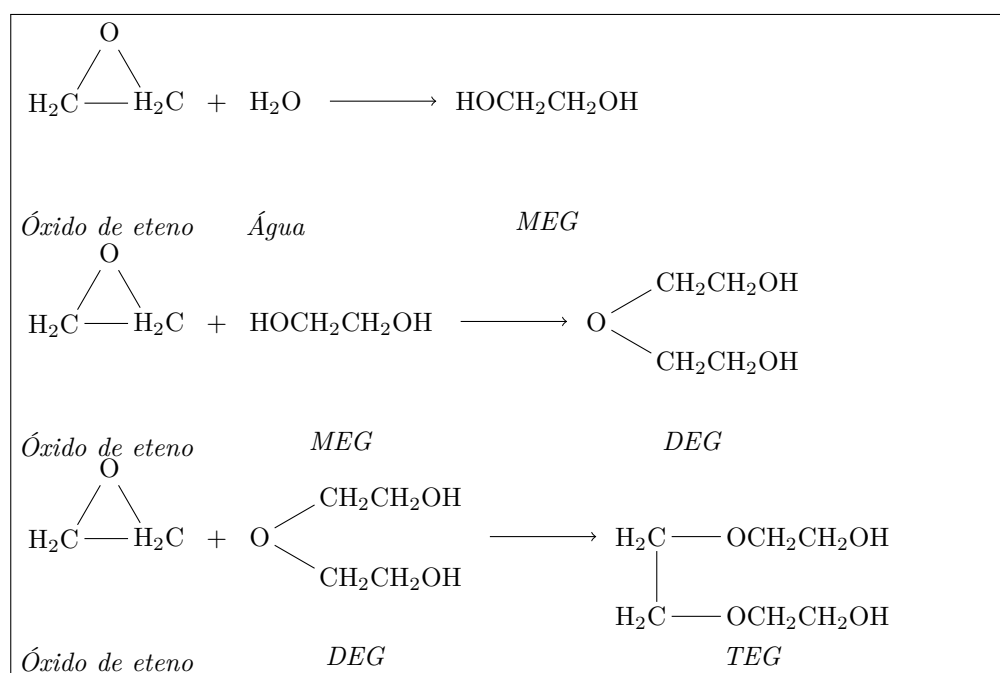
Fonte: o autor

2.2 O monoetilenoglicol

O monoetilenoglicol, conhecido simplesmente como etilenoglicol, ou MEG, ou pelo nome IUPAC: etano-1,2-diol, ou 1,2-etanodiol, é um álcool com dois grupos-OH (um diol), sendo largamente utilizado como um anticongelante automotivo. Puro é um composto inodoro, incolor, líquido xaroposo com um sabor doce. O MEG é tóxico, e sua ingestão é considerada uma emergência médica. Sua densidade a 20 °C é 1.1132 g/cm³, com massa molecular 62.068 g/mol, ponto de fusão -12.9 °C e Ponto de ebulição 197.3 °C (WIKIPEDIA, 2014).

O MEG é o mais simples dos etilenoglicóis. Normalmente é obtido pela reação de água com óxido de eteno. Reagindo-se MEG com óxido de eteno, produz-se o dietilenoglicol (DEG) e a partir deste, adicionando-se mais óxido de eteno, chega-se ao trietilenoglicol (TEG) (ANDRADE, 2009). O esquema das reações de obtenção destes compostos e sua estrutura molecular são mostrados na Figura 11.

Figura 11 – Esquema de formação de etilenoglicóis.



Fonte: Andrade (2009, p. 41), adaptada.

2.3 Seleção dos sistemas

A Tabela 3 apresenta as características da água de formação representativa do campo de Mexilhão, coletada no poço 3-MXL-3-SPS. Considerando que a água de produção possui composição semelhante a de formação, dentre os componentes presentes, optou-se por selecionar os seguintes componentes ou sistemas, com respeito ao comportamento da

solubilidade de sais e outras propriedades, para modelagem e simulação: MEG+H₂O+NaCl e MEG+H₂O+KCl.

Tabela 3 – Características da água de formação do reservatório do campo de Mexilhão.

Parâmetro	Valor
Na ⁺	52.000 mg/L
K ⁺	3.400 mg/L
Mg ²⁺	67 mg/L
Ca ²⁺	31.000 mg/L
Ba ²⁺	670 mg/L
Sr ²⁺	4.300 mg/L
Ferro total	1,9 mg/L
Cl ⁻	139.500 mg/L
Salinidade (NaCl)	230.175 mg/L
SO ₄ ⁴⁻	<10 mg/L
Br ⁻	1.540 mg/L
Alcalinidade Total (HCO ₃ ⁻)	-
pH	8,1

Fonte: [Petrobras \(2007, p. 110\)](#)

Em paralelo ao trabalho de modelagem e simulação de solubilidade de sais para sistemas aquosos com MEG, tema desta de tese de doutorado, uma equipe do FOTEQ tem realizado medidas de propriedades termodinâmicas de interesse como condutividade, densidade, pH e dados de equilíbrio líquido-vapor (ELV).

Dados de ELV de sistemas aquosos com MEG e alcoóis (metanol e etanol) na presença de eletrólitos fortes, como o cloreto de sódio, também foram estudados neste trabalho, em termos de modelagem e simulação, gerando um artigo ([OLIVEIRA et al., 2014](#)). Este artigo é incluído no [Apêndice A](#) e o programa desenvolvido para ele é apresentado no [Apêndice B](#).

2.4 Equilíbrio de Fases para Sistemas com Eletrólitos

2.4.1 Escalas de concentração

Geralmente a concentração de uma solução eletrolítica é expressa em molalidade, molaridade ou fração molar.

A molalidade é definida como o número de moles do soluto por quilo de solvente:

$$b_i \equiv \frac{n_i}{m_{\text{solv}}} = \frac{n_i}{n_{\text{solv}} M_{\text{solv}}} 1000 \quad (2.1)$$

onde b_i é a molalidade de i , m_{solv} é a massa do solvente, n_i é o número de moles de i e M_i é a massa molecular de i .

A molaridade é definida como o número de moles do soluto por litro de solução:

$$c_i \equiv \frac{n_i}{V} \quad (2.2)$$

onde c_i é a molaridade de i e V é o volume da solução.

A fração molar é definida como a razão entre o número de moles do soluto pelo número de moles total na solução:

$$x_i \equiv \frac{n_i}{n} \quad (2.3)$$

onde x_i é a fração molar de i e n é o número de moles total na solução.

Como a molaridade é dependente da temperatura, esta não é preferida como unidade de concentração em soluções eletrolíticas. Daqui para a frente será utilizada, quando for necessário maior clareza, os subscritos b , m ou x , respectivamente para molalidade, molaridade ou fração molar, como indicação da escala de concentração utilizada.

2.4.2 Potencial químico

O potencial químico é definido como:

$$\mu_i \equiv \left[\frac{\partial (nG)}{\partial n_i} \right]_{T,P,n_{j \neq i}} \quad (2.4)$$

onde μ_i é o potencial químico de i , G é a energia livre de Gibbs, T é a temperatura, P é a pressão e $n_{j \neq i}$ designa todos os números de moles exceto o referente a i .

O potencial químico relaciona-se a atividade ou concentração e ao coeficiente de atividade da seguinte forma:

$$\mu_i = \mu_i^\bullet + RT \ln a_i = \mu_i^\bullet + RT \ln (\gamma_i \xi_i) \quad (2.5)$$

onde R é a constante universal dos gases ideais, $\mu_i^\bullet$ é o potencial químico de i em um estado padrão apropriadamente escolhido, a_i é a atividade de i , γ_i é o coeficiente de atividade de i e ξ_i é uma conveniente escala de concentração.

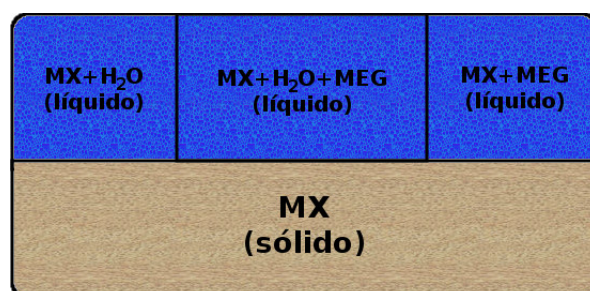
O potencial químico constitui uma grandeza de suma importância na abordagem de equilíbrio de fases. Aqui, mostraremos um exemplo ilustrado pela [Figura 12](#), onde a igualdade dos potenciais de um eletrólito no equilíbrio sólido-líquido, contendo uma mistura de solventes, como água e MEG, pode ser descrita em termos das seguintes equações:

$$\mu_{MX}^s = \mu_{MX}^{\{H_2O\}} \quad (2.6)$$

$$\mu_{MX}^s = \mu_{MX}^{\{MEG\}} \quad (2.7)$$

$$\mu_{MX}^s = \mu_{MX}^{\{H_2O+MEG\}} \quad (2.8)$$

Figura 12 – Representação do Equilíbrio de um eletrólito em mistura de solventes.



Fonte: o autor.

2.4.3 Estado padrão

Um equívoco comum é dizer que o estado padrão do potencial químico para o soluto é o soluto à temperatura e pressão do sistema e à diluição infinita. Isso não é correto, pois, à diluição infinita, o potencial químico do soluto é negativamente infinito (PRAUSNITZ; LICHTENTHALER; AZEVEDO, 1999). O estado padrão do potencial químico para um soluto precisa estar à uma concentração fixa (diferente de zero), conhecida como concentração unitária. Esta é usada por ser o seu logaritmo igual a zero.

O estado padrão do potencial químico não precisa necessariamente ser realizável, mas precisa ser bem definido.

Para soluções concentradas, ou líquidos puros, a fração molar é uma escala de concentração mais conveniente, pois para soluções concentradas os valores tanto para molaridade quanto para a molalidade tendem a serem enormes, e no caso de líquidos puros atingem o infinito. Uma convenção de estado padrão comumente usada é a convenção por *coeficiente de atividade racional simétrico* onde normaliza-se de forma que os coeficientes de atividade são unitários para componentes puros, assim sendo:

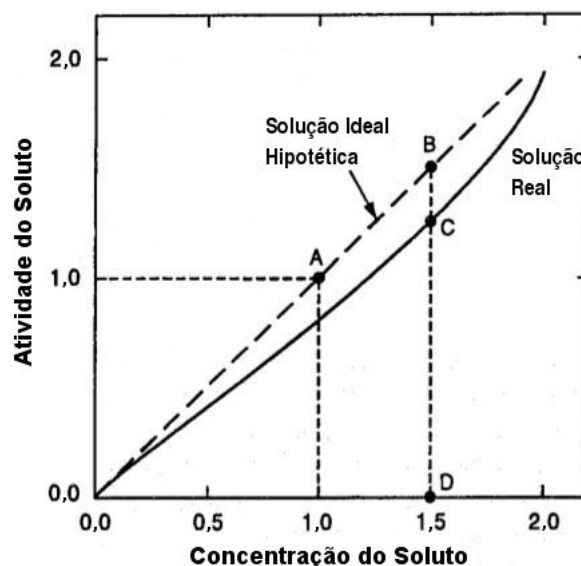
$$\mu_i = \mu_i^0 + RT \ln(\gamma_i x_i) \quad (2.9)$$

onde μ_i^0 é o potencial químico racional simétrico de i .

A Figura 13 ilustra o conceito de solução ideal hipotética. A atividade no estado padrão é dada pelo ponto A e corresponde a concentração unitária (ξ_i). A atividade do soluto para uma concentração de 1,5 é dada pelo ponto C, enquanto que na solução ideal hipotética é dada por B. Na solução ideal a atividade é igual a concentração porque, na solução ideal, o coeficiente de atividade é unitário para todas as concentrações de soluto. Assim, o coeficiente de atividade em soluções reais, $\gamma_i = a_i/\xi_i$ é dado pela razão entre os segmentos $\overline{CD}$ e $\overline{BD}$.

Como as propriedades dos íons não podem ser medidos (reportados) independentemente dos outros íons em soluções, usa-se normalmente a convenção assimétrica baseada

Figura 13 – Conceito de solução ideal hipotética.



Fonte: Prausnitz, Lichtenthaler e Azevedo (1999, p. 511), adaptada.

na fração molar ou na molalidade (KONTOGEORGIS; FOLAS, 2010). Na convenção por *coeficiente de atividade racional assimétrico* normaliza-se de forma que os coeficientes de atividade são unitários à diluição infinita, ou seja, em solvente puro:

$$\mu_i = \mu_{i,x}^\nabla + RT \ln (\gamma_{i,x}^* x_i) \quad \text{com} \quad \gamma_{i,x}^* = \gamma_i / \gamma_i^\infty \quad (2.10)$$

onde $\mu_{i,x}^\nabla$ é o potencial químico racional assimétrico, $\gamma_{i,x}^*$ é o coeficiente de atividade racional assimétrico e γ_i^∞ é o coeficiente de atividade em diluição infinita. Na convenção por *coeficiente de atividade molal assimétrico* normaliza-se da mesma forma, e usando o conceito de solução ideal hipotética à molalidade unitária, $b_0 = 1$, tem-se:

$$\mu_i = \mu_{i,b}^\nabla + RT \ln \left(\frac{\gamma_{i,b}^* b_i}{b_0} \right) = \mu_{i,b}^\nabla + RT \ln (\gamma_{i,b}^* b_i) \quad \text{com} \quad \gamma_{i,b}^* = \gamma_{i,x}^* x_{\text{solv}} \quad (2.11)$$

onde $\mu_{i,b}^\nabla$ é o potencial químico molal assimétrico, $\gamma_{i,b}^*$ é o coeficiente de atividade molal assimétrico.

2.4.4 Coeficiente de atividade médio iônico e molalidade média iônica

Representa-se a dissociação de um eletrólito eletricamente neutro $M_{\nu_+} X_{\nu_-}$ em ν_+ cátions (cada qual com carga z_+) e ν_- ânions (cada qual com carga z_-) da seguinte forma:



onde a eletroneutralidade requer que $\nu_+ z_+ + \nu_- z_- = 0$. Por comodidade, abreviaremos $M_{\nu_+} X_{\nu_-}$ por MX.

De Prausnitz, Lichtenthaler e Azevedo (1999), tem-se as definições de coeficiente de atividade médio iônico (Equação 2.13) e molalidade média iônica (Equação 2.14):

$$\gamma_{\pm} \equiv (\gamma_+^{\nu_+} \gamma_-^{\nu_-})^{\frac{1}{\nu}} \quad (2.13)$$

$$b_{\pm} \equiv (b_+^{\nu_+} b_-^{\nu_-})^{\frac{1}{\nu}} \quad (2.14)$$

onde $\nu = \nu_+ + \nu_-$.

2.4.5 Leis limitantes

Por lei limitante compreende-se aquela cuja validade (aplicação) encontra-se na região de diluição infinita.

Numa solução ideal, o cálculo das propriedades de mistura requer a composição da mistura e as propriedades dos componentes puros. A solução ideal de um sistema não eletrolítico de acordo com a Lei de Raoult para o equilíbrio líquido-vapor é:

$$y_i P = x_i P_i^{\text{sat}} \quad (2.15)$$

onde o subscrito i se refere ao componente i e, portanto, y_i e x_i são as frações molares deste nas fases gasosa e líquida, respectivamente, e P_i^{sat} é a pressão de vapor do componente i puro.

Em soluções reais diluídas, com respeito ao soluto, o solvente tende a seguir a Lei de Raoult. Uma solução diluída ideal é a solução onde o solvente segue a Lei de Raoult e o soluto segue a Lei de Henry:

$$y_i P = x_i H_i \quad (2.16)$$

onde H_i é a constante de Henry para o componente i .

Para soluções eletrolíticas as interações entre os íons e o solvente, através de suas cargas, tende a provocar desvios da idealidade, podendo ser relevantes, mesmo a baixas concentrações. Uma vez que os íons não são voláteis a pressão atmosférica e temperatura ambiente, existe a necessidade de uma teoria específica para descrever o comportamento de soluções eletrolíticas. A partir da hipótese de que cada íon esteja numa atmosfera de íons de carga oposta (*blindagem*), Debye e Hückel (1923) formularam uma teoria para soluções eletrolíticas. A medida do comprimento desta *blindagem* recebe o nome de comprimento de Debye-Hückel (κ^{-1}), definido como:

$$\kappa^{-1} \equiv \left(\frac{\varepsilon_0 \varepsilon_r R T}{2 \rho_{\text{solv}} N_A^2 e^2 I} \right)^{\frac{1}{2}} \quad (2.17)$$

onde ε_0 é a permissividade no vácuo, ε_r é permissividade relativa, usualmente conhecida como constante dielétrica, ρ_{solv} é a densidade do solvente, N_A é o número de Avogadro, e é a carga do elétron e I é a força iônica, definida como:

$$I \equiv \frac{1}{2} \sum_i b_i z_i^2 \quad (2.18)$$

Debye e Hückel (1923) derivaram uma expressão simples para o coeficiente de atividade γ_i com carga z_i e força iônica I :

$$\ln \gamma_{i,c} = -z_i^2 \frac{e^2 N_A}{8\pi\epsilon_0\epsilon_r RT} \kappa \quad (2.19)$$

Para soluções aquosas diluídas próximo a temperatura ambiente, não existe diferença significativa entre molalidade e molaridade (PRAUSNITZ; LICHTENTHALER; AZEVEDO, 1999). Assim sendo, trocando as unidades do coeficiente de atividade e da força iônica na Equação 2.19, temos:

$$\ln \gamma_{i,b} = -A_\gamma z_i^2 I^{\frac{1}{2}} \quad (2.20)$$

onde A_γ é:

$$A_\gamma = \left(\frac{e^2}{\epsilon_0\epsilon_r RT} \right)^{\frac{3}{2}} \frac{N_A^2}{8\pi} (2\rho_{\text{solv}})^{\frac{1}{2}} \quad (2.21)$$

Para o eletrólito $M_{\nu+}X_{\nu-}$, usando Equação 2.13, e desprezando o subscrito b , temos a Lei de Debye-Hückel:

$$\ln \gamma_{\pm} = -A_\gamma |z_+ z_-| I^{\frac{1}{2}} \quad (2.22)$$

O comportamento descrito pela Lei de Raoult (Equação 2.15) e pela Lei de Henry (Equação 2.16) tende a ser o comportamento de soluções reais conforme estas tendam a ser diluídas. Da mesma forma, a Lei de Debye-Hückel providencia uma representação satisfatória do comportamento dos coeficientes de atividade em soluções diluídas, sendo limitada para soluções aquosas com força iônica até 0,01 mol/kg. A Lei de Debye-Hückel Extendida (GUGGENHEIM, 1935) aumentou o limite de aplicação no cálculo de coeficiente de atividade para soluções aquosas com força iônica até 0,1 mol/kg:

$$\ln \gamma_{\pm} = -\frac{1}{\nu} \sum_i \nu_i z_i^2 \frac{AI^{\frac{1}{2}}}{1 + \ell I^{\frac{1}{2}}} + B_{\pm} I \quad (2.23)$$

onde ℓ é um parâmetro dependente do tamanho dos íons, usualmente considerado constante ($\ell = 1,2 \text{ kg}^{\frac{1}{2}} \cdot \text{mol}^{-\frac{1}{2}}$) e $B_{\pm}$ parâmetro empírico da extensão de Guggenheim.

2.4.6 Energia livre de Gibbs para soluções eletrolíticas reais

Soluções reais podem ser descritas a partir de seus desvios em relação à idealidade por propriedades de excesso. Para uma propriedade M , temos:

$$M^E \equiv M - M^{\text{id}} \quad (2.24)$$

onde M^E representa uma propriedade de excesso e M^{id} representa uma propriedade ideal.

Para uma propriedade parcial molar de excesso de uma espécie i ($\bar{M}_i^E$) pode ser definida de forma análoga:

$$\bar{M}_i^E \equiv \bar{M}_i - \bar{M}_i^{\text{id}} \quad (2.25)$$

onde $\bar{M}_i$ representa uma propriedade parcial molar de i e $\bar{M}_i^{\text{id}}$ representa uma propriedade parcial molar ideal de i .

A propriedade parcial molar da energia livre de Gibbs, que é função das variáveis operacionais T e P , vale o teorema de Euler ou a característica de aditividade (PRAUSNITZ; LICHTENTHALER; AZEVEDO, 1999).

Para um sistema aberto, multicomponente e com uma única fase, tem-se:

$$d(nG) = nVdP - nSdT + \sum_i \bar{G}_i dn_i \quad (2.26)$$

onde G é a energia livre de Gibbs, S é a entropia e $\bar{G}$ é a energia livre de Gibbs parcial molar de i .

Dividindo nG por RT e diferenciando em relação à T , tem-se:

$$d\left(\frac{nG}{RT}\right) = \frac{1}{RT}d(nG) - \frac{nG}{RT^2}dT \quad (2.27)$$

Dividindo a Equação 2.26 por RT e somando com a equação Equação 2.27, tem-se:

$$d\left(\frac{nG}{RT}\right) = \frac{nV}{RT}dP - \left(\frac{nS}{RT} + \frac{nG}{RT^2}\right)dT + \sum_i \frac{\bar{G}_i}{RT}dn_i \quad (2.28)$$

onde H é a energia livre de Helmholtz.

Como $G = H - TS$, tem-se, substituindo na Equação 2.27, uma relação fundamental para G bastante útil em termodinâmica de soluções, ou seja:

$$d\left(\frac{nG}{RT}\right) = \frac{nV}{RT}dP - \frac{nH}{RT^2}dT + \sum_i \frac{\bar{G}_i}{RT}dn_i \quad (2.29)$$

A partir da energia livre de Gibbs pode-se gerar outras propriedades termodinâmicas, isto é ilustrado na Figura 14.

Para soluções aquosas com eletrólitos, como, por exemplo, um sal MX completamente dissociado em água, à pressão e temperatura constantes, a partir da Equação 2.29 e da Equação 2.4, pode-se expressar a energia livre de Gibbs como:

$$G = \sum_i n_i \mu_i = n_{\text{H}_2\text{O}} \mu_{\text{H}_2\text{O}} + n_{\text{MX}} \sum_i \nu_i \mu_i \quad (2.30)$$

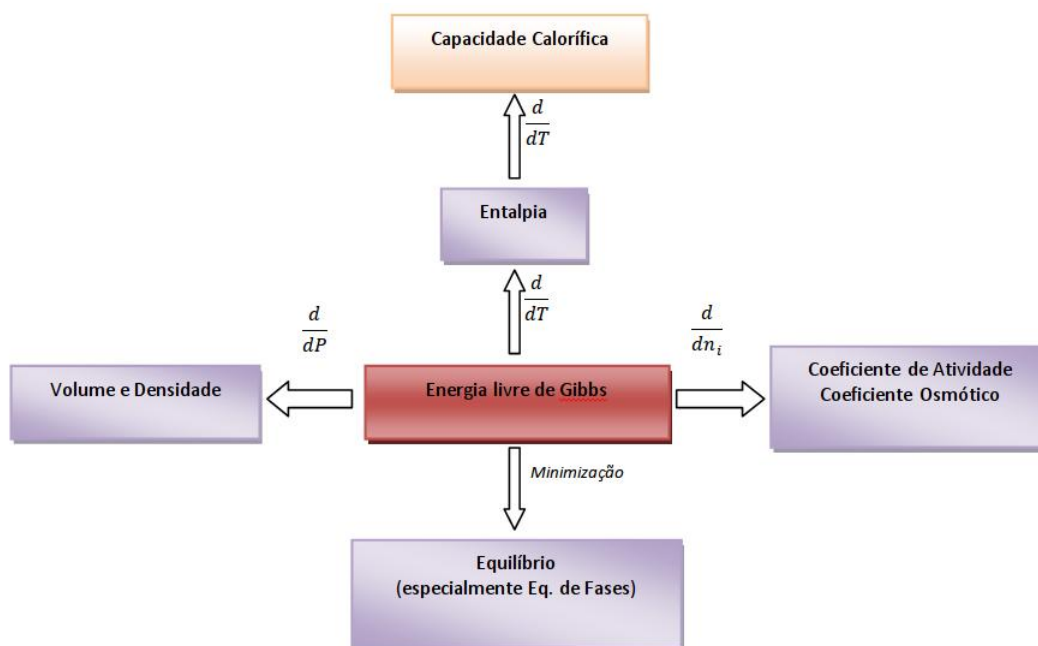
2.4.7 Fugacidade

Da termodinâmica clássica, à temperatura constante, temos:

$$d\mu_i \equiv RT d(\ln \hat{f}_i) \Rightarrow \mu_i = \mu_i^\bullet + RT d \ln \frac{\hat{f}_i}{f_i^\bullet} \quad (2.31)$$

onde $\hat{f}_i$ é a fugacidade de i na solução, ou mistura, e $f_i^\bullet$ é a fugacidade de i no estado padrão.

Figura 14 – Fluxograma de obtenção de propriedades termodinâmicas a partir da energia livre de Gibbs.



Fonte: o autor.

A fugacidade de um componente i pode ser expressa, usando convenção simétrica e escala de fração molar, como:

$$\hat{f}_i = x_i \gamma_i f_i^0 \quad (2.32)$$

onde f_i^0 é a fugacidade racional simétrica de i .

A fugacidade de um componente i pode ser expressa, usando convenção assimétrica e escala de fração molar, como:

$$\hat{f}_i = x_i \gamma_i^* H_{i,x} \quad (2.33)$$

onde $H_{i,x}$ é a constante de Henry racional.

A fugacidade de um componente i pode também ser expressa, usando convenção assimétrica e escala de molalidade (b), como:

$$\hat{f}_i = b_i \gamma_{i,b}^* H_{i,b} \quad (2.34)$$

onde $H_{i,b}$ é a constante de Henry molal.

2.5 Modelos Termodinâmicos

Nesta seção são descritos os modelos termodinâmicos aplicados para a descrição dos dados de solubilidade de sais em misturas de solventes, seguindo uma ordem de complexidade, ou seja, partindo-se do modelo ideal, passando por uma correlação empírica

(Setschenov) e por último aplicando a correção das não-idealidades através da equação de Pitzer adaptada para mistura de solventes.

2.5.1 Ideal

A definição do logaritmo de excesso da solubilidade (LORIMER, 1993) permite uma avaliação simplificada do comportamento das soluções aquosas com MEG e o sal em estudo. Isto é, considerando solução ideal não é necessário usar-se modelo de energia livre de excesso. Primeiramente, deve-se definir o logaritmo ideal da solubilidade que foi escolhida como uma regra linear da composição para uma mistura binária de solventes:

$$\left[\ln b_{MX}^{\{H_2O+MEG\}} \right]^{id} \equiv x_{H_2O}^{\{H_2O+MEG\}} \ln b_{MX}^{\{H_2O\}} + x_{MEG}^{\{H_2O+MEG\}} \ln b_{MX}^{\{MEG\}} \quad (2.35)$$

onde $\left[\ln b_{MX}^{\{H_2O+MEG\}} \right]^{id}$ é o logaritmo ideal da molalidade do sal MX em água e MEG, $x_{H_2O}^{\{H_2O+MEG\}}$ é a fração molar da água livre de sal, $x_{MEG}^{\{H_2O+MEG\}}$ é a fração molar do MEG livre de sal, $\ln b_{MX}^{\{H_2O\}}$ é o logaritmo ideal da molalidade do sal MX em água e $\ln b_{MX}^{\{MEG\}}$ é o logaritmo ideal da molalidade do sal MX em MEG.

A definição do logaritmo de excesso da solubilidade foi feita conforme a Equação 2.24:

$$\left[\ln b_{MX}^{\{H_2O+MEG\}} \right]^E \equiv \ln b_{MX}^{\{H_2O+MEG\}} - \left(\ln b_{MX}^{\{H_2O+MEG\}} \right)^{id} \quad (2.36)$$

onde $\left[\ln b_{MX}^{\{H_2O+MEG\}} \right]^E$ é o logaritmo de excesso da molalidade do sal MX em água e MEG.

Essa abordagem ideal pode representar surpreendentemente e de maneira simplificada a influência do solvente na solubilidade do sal. Ela requer o conhecimento das solubilidades do sal nos solventes puros e descreve nestes extremos os valores de maneira plena. Vale ressaltar nesse momento que as soluções de água e MEG formam soluções ideais (CHIAVONE-FILHO; PROUST; RASMUSSEN, 1993) para o comportamento líquido-vapor.

2.5.2 Setchenov

O efeito linear do solvente sobre a solubilidade do sal numa escala logarítmica pode ser mensurado pela equação de Setchenov (Equação 2.37), que expressa o efeito *salting-out* (LEE, 1997), que é justamente o oposto do que ocorre no processo de regeneração do MEG, mas acompanha o mesmo fenômeno físico.

$$\ln b_{MX}^{\{H_2O+MEG\}} = \ln b_{MX}^{\{H_2O\}} + k_S x_{MEG}^{\{H_2O+MEG\}} \quad (2.37)$$

Onde $b_{MX}^{\{H_2O+MEG\}}$ é a molalidade do sal MX em água e MEG, $b_{MX}^{\{H_2O\}}$ é a molalidade do sal MX em água e k_S é a constante de Setchenov. Aplicando uma correlação equivalente entre o produto de solubilidade de um componente em água e o mesmo em uma mistura de solventes (água+MEG) usando essa equação de Setchenov, pode-se estimar o produto de solubilidade nesse novo meio solvente (OLIVEIRA et al., 2010).

2.5.3 Pitzer

Seguindo a abordagem de [Lorimer \(1993\)](#), que usa a teoria de McMillan-Mayer, é possível corrigir as não-idealidades da fase líquida com uma equação de coeficiente de atividade, como o modelo de Pitzer. O modelo de Pitzer foi desenvolvido para o solvente água e portanto a aplicação dele para mistura de solventes requer uma estratégia para estimar os parâmetros solvente orgânico-sal.

Na termodinâmica de soluções eletrolíticas é comum expressar os modelos de correção das não idealidades pelo coeficiente osmótico (solvente) ou pelo coeficiente de atividade (soluto). A definição do coeficiente osmótico é dada pela [Equação 2.38](#).

$$\phi \equiv \frac{-\ln a_{\text{H}_2\text{O}}}{M_{\text{H}_2\text{O}} \sum_i b_i} \quad (2.38)$$

onde ϕ é o coeficiente osmótico, $a_{\text{H}_2\text{O}}$ é a atividade da água e $M_{\text{H}_2\text{O}}$ é a massa molecular da água.

Para o sal MX, o modelo de Pitzer para o coeficiente osmótico e para o coeficiente de atividade, é dado pelas equações (23) e (24), respectivamente.

$$\phi = 1 + |z_M z_X| f_{\text{MX}}^\phi + b \frac{2\nu_M \nu_X}{\nu} B_{\text{MX}}^\phi + b^2 \frac{2(\nu_M \nu_X)^{3/2}}{\nu} C_{\text{MX}}^\phi \quad (2.39)$$

$$\ln \gamma_{\text{MX}} = |z_M z_X| f_{\text{MX}}^\gamma + b \frac{2\nu_M \nu_X}{\nu} B_{\text{MX}}^\gamma + b^2 \frac{2(\nu_M \nu_X)^{3/2}}{\nu} C_{\text{MX}}^\gamma \quad (2.40)$$

onde:

$$f_{\text{MX}}^\phi = -A_\phi \frac{I^{1/2}}{1 + \mathfrak{b} I^{1/2}} \quad \text{e} \quad f_{\text{MX}}^\gamma = -A_\phi \left[\frac{2}{\mathfrak{b}} \ln(1 + \mathfrak{b} I^{1/2}) + \frac{I^{1/2}}{1 + b I^{1/2}} \right] \quad (2.41)$$

$$A_\phi = \frac{1}{3} \left(\frac{2\pi N_0 \rho_w}{1000} \right)^{1/2} \left(\frac{e^2}{DkT} \right)^{3/2} \quad (2.42)$$

$$I = \frac{|z_M z_X|}{2} \frac{b}{b_0} \nu \quad \text{e} \quad \mathfrak{b} = 1, 2 \text{ kg}^{0,5} \text{mol}^{-0,5} \text{ (para soluções aquosas a 25 deg C)} \quad (2.43)$$

$$\nu = \nu_M + \nu_X \quad \text{e} \quad b_0 = 1 \text{ mol/kg}_{\text{H}_2\text{O}} \quad (2.44)$$

$$B_{\text{MX}}^\phi = \beta_{\text{MX}}^{(0)} + \beta_{\text{MX}}^{(1)} e^{-\alpha I^{1/2}} \quad \text{e} \quad B_{\text{MX}}^\gamma = 2\beta_{\text{MX}}^{(0)} + \frac{2\beta_{\text{MX}}^{(1)}}{\alpha^2 I} \left[1 - \left(1 + \alpha I^{1/2} - \frac{1}{2} \alpha^2 I \right) e^{-\alpha I^{1/2}} \right] \quad (2.45)$$

$$\alpha = 2, 0 \text{ kg}^{0,5} \text{mol}^{-0,5} \quad (2.46)$$

$$C_{MX}^{\phi} = \frac{3}{(\nu_M \nu_X)^{1/2}} (\nu_M \Lambda_{MMX} + \nu_X \Lambda_{MXX}) \quad \text{e} \quad C_{MX}^{\gamma} = \frac{3}{2} C_{MX}^{\phi} \quad (2.47)$$

sendo $\beta_{MX}^{(0)}$, $\beta_{MX}^{(1)}$, Λ_{MMX} , Λ_{MXX} são parâmetros do modelos e ν , ν_M e ν_X referem-se aos coeficientes estequiométricos do sal, do cátion e do ânion, respectivamente e, z_M e z_X referem-se carga do cátion e do ânion, respectivamente.

Para expressar o efeito térmicos, os parâmetros do modelo devem ser considerados dependentes da temperatura (JR. et al., 1986). Lorimer aplicou o modelo de Pitzer para misturas de solventes determinando o parâmetro β como função da constante dielétrica e da composição de solvente livre de sal. Essa modelagem foi aplicada neste trabalho e será descrita no Capítulo 3.

2.6 Outros Modelos Termodinâmicos

Várias teses de doutorado no tema de modelagem termodinâmica de solubilidade para sistemas aquosos com eletrólitos pode ser encontradas (CARDOSO, 1988; CHIAVONE-FILHO, 1993; NICOLAISEN, 1994; THOMSEN, 1997). Nestes trabalhos, desenvolvidos na Universidade Técnica da Dinamarca, pode-se observar o potencial de aplicação dos modelos de coeficiente de atividade para sistemas aquosos com sais e MEG. Esta abordagem, entretanto, requer informações experimentais para a estimação dos parâmetros dos respectivos modelos e, por conta da escassez citada, há necessidade de desenvolvimento de modelos para as misturas específicas, como é o caso do sistema aquoso com MEG e sais.

Renon (1996) e Anderko, Wang e Rafal (2002) apresentam resumos de diferentes modelos utilizados para soluções eletrolíticas com mistura de solventes, com proposta de classificação, tipo de equilíbrio, estados de referência e algumas hipóteses simplificadoras adotadas na modelagem. Observa-se que novos modelos são obtidos através de combinações de modelos anteriores para representar contribuições de interações entre espécies, de longe, médio (interações específicas) e curto alcance.

Liu e Watanasiri (1996) e Chen e Song (2004) utilizaram o modelo NTRL para representar as interações de curto alcance e o modelo de Pitzer-Debye-Hückel (PDK) para as interações de longo alcance.

Pinho e Macedo (1996), Iliuta, Thomsen e Rasmussen (2000), Wang, Anderko e Young (2002), Thomsen, Iliuta e Rasmussen (2004), Wang et al. (2006), Kosinski et al. (2007), (FOSBOL; THOMSEN; STENBY, 2009) utilizaram o modelo UNIQUAC para representar as interações de curto alcance e o modelo de Pitzer-Debye-Hückel (PDK) para as interações de longo alcance. Observa-se em alguns dos trabalhos citados, o uso da equação de estado cúbica de Soave-Redlich-Kwong (SRK) para VLE e a descrição de interações específicas entre íons realizadas por uma expansão de virial.

Grenthe e Plyasunov (1997) fizeram uma comparação entre o modelo de Interação Específica entre Íons de Brønsted-Guggenheim-Scatchard (SIT) e o modelo de Pitzer-Debye-Hückel (PDK).

Dahl e Macedo (1992) utilizaram o modelo de Contribuição de Grupos UNIFAC juntamente com a equação de estado cúbica de Soave-Redlich-Kwong (SRK) para VLE e LLE de misturas com eletrólitos fortes.

Thomsen (1997) desenvolveu um modelo UNIQUAC estendido para soluções aquosas com eletrólitos baseado nas seguintes propriedades de equilíbrio e calorimétricas: atividade, coeficiente osmótico, grau de dissociação, densidade, calor de diluição e de solução e capacidade calorífica. A partir destes novos parâmetros foram elaborados diagramas de fases e simulações de processos, seguidas de otimização. O trabalho de Thomsen é uma importante referência, mas não se aplica diretamente à mistura de solventes, foco da presente proposta de tese de doutorado.

Lorimer (1993) apresenta uma modelagem não-convencional para sistemas eletrolíticos com mistura de solventes, baseado no desenvolvimento de soluções aquosas. Os modelos derivados de Pitzer são os convencionais para soluções aquosas eletrolíticas Pytkowicz, 1979; (JR. et al., 1986). Estes modelos se encontram em versões mais atualizadas de simuladores de processos como Aspen HYSYS®. Entretanto, parâmetros de íons e moléculas orgânicas ou grupos funcionais são raros e assim sendo comprometem as simulações de sistemas aquosos eletrolíticos com misturas de solventes.

Com respeito aos métodos de cálculo de equilíbrio de fases e fundamentação, livros textos da área serão utilizados no projeto (SMITH; NESS; ABBOTT, 2005; WALAS, 1985; PITZER, 1995; PRAUSNITZ; LICHTENTHALER; AZEVEDO, 1999; SANDLER, 1993).

Os modelos de coeficiente de atividade desenvolvidos para soluções não-eletrolíticas, como UNIQUAC (ABRAMS; PRAUSNITZ, 1975), têm sido extensivamente aplicados. Estes modelos se caracterizam por uma forte flexibilidade matemática permitindo correlação de dados de equilíbrio para misturas complexas e também possuem fundamentação teórica.

O modelo UNIQUAC+DH foi aplicado para a modelagem de solubilidade de sais em mistura de solventes para os sistemas estudados (CHIAVONE-FILHO; RASMUSSEN, 2000). Contudo com base na literatura é mais adequado uso de modelos normalizados com a convenção assimétrica para sistemas com eletrólitos e portanto o modelo de Pitzer (1973) e modelagem de solubilidade de excesso apresentada por Lorimer (1993), que aplica a abordagem de McMillan-Mayer se apresenta mais coerente do ponto de vista de termodinâmica molecular e projeto de processos.

2.7 Ferramentas Computacionais

O uso de ferramentas computacionais para a resolução de problemas em equilíbrio de fases de sistemas com eletrólitos se faz necessário devido à quantidade enorme de cálculos e complexidade destes. A utilização destas ferramentas acompanhou o desenvolvimento das linguagens de computação, onde, hoje, cada qual apresenta suas vantagens:

- a) O FORTRAN apresenta uma grande quantidade de bibliotecas matemáticas já consolidadas, como por exemplo, as bibliotecas numéricas e estatísticas IMSL;
- b) O C++ apresenta-se como uma linguagem muito utilizada no desenvolvimento final de programas onde a programação de alto nível simplificada e rápida permite excelentes resultados no nível de interação com o usuário;
- c) Deve-se destacar os programas desenvolvidos para atividades específicas de simulação de processos como, por exemplo, o conhecido Aspen HYSYS®, que apresenta um conjunto de ferramentas, inclusive gráficas que permitem reproduzir por simulação muitos dos processos encontrados na indústria de petróleo e gás natural;
- d) Pode-se destacar também programas com uma grande gama de pacotes matemáticos e de engenharia, como por exemplo; o MATLAB, ou o seu clone livre: o SCILAB;
- e) A linguagem de programação Python destaca-se pelo fato de como atualmente as máquinas (hardwares) estarem cada vez mais rápidos, implica que a necessidade de otimização do código para a máquina (como em linguagens como FORTRAN e C++) perde cada vez mais espaço em relação à otimização do código para o programador. É o caso da linguagem de programação Python, cuja otimização em relação ao programador permite até a mudança dinâmica da programação, ou seja, em tempo de execução. O destaque do Python em relação a outras linguagens também é evidente pela sua característica de ser multiplataforma (Windows, Linux, MacOS), possuir licença livre (tendência mundial), apresentar bibliotecas numéricas avançadas, como o NumPy, científicas, como o SciPy, gráficas, como o Matplotlib, de excepcional qualidade, além do desenvolvimento em programação visual, por meio do PySide, ser simples e totalmente multiplataforma.

Para a implementação e testes dos modelos neste trabalho de tese de doutorado adotou-se a linguagem Python, juntamente com as já citadas bibliotecas: PySide, NumPy, SciPy, Matplotlib e utiliza-se o ambiente de programação Eclipse.

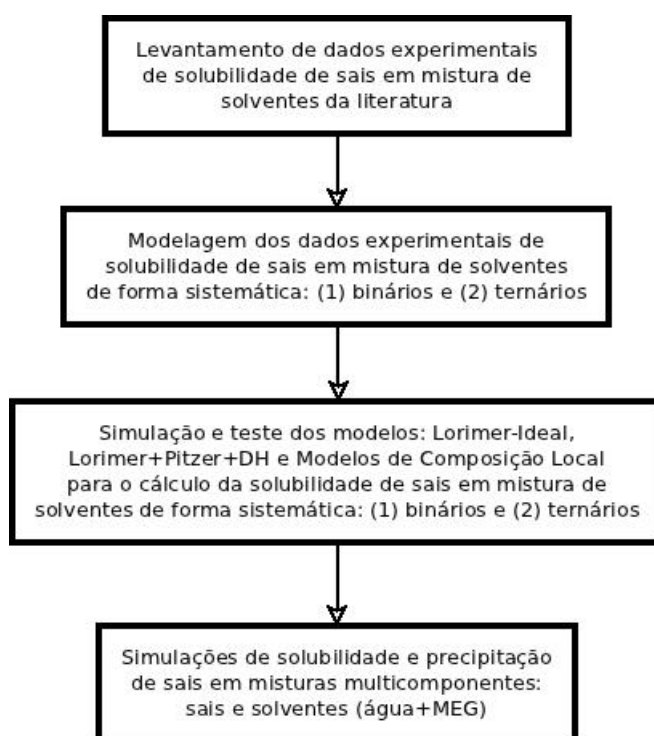
CAPÍTULO 3

METODOLOGIA

3 METODOLOGIA

A metodologia de trabalho aplicada nesta tese está resumida na [Figura 15](#).

Figura 15 – Fluxograma da metodologia de desenvolvimento e teste do modelo de solubilidade de sais em misturas de solventes.



Fonte: o autor

Como pode ser observado na [Figura 15](#), de posse do modelo de coeficiente de atividade, pode-se desenvolver um método de cálculo para calcular a solubilidade de uma espécie salina e também identificar se uma mistura forma ou não precipitados. É importante destacar que soluções eletrolíticas apresentam forte afastamento da idealidade e, por conseguinte, demandam correção produzida por um modelo de coeficiente de atividade. Por outro lado com a abordagem de Lorimer de solubilidade de excesso, esse afastamento fica reduzido.

3.1 Molalidade e fração molar

3.1.1 Cálculo da molalidade do sal na mistura de solventes

A molalidade do sal na mistura de solventes ($b_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}}$) foi calculada em função das propriedades e unidades dos dados experimentais consultados:

$$b_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}} = \frac{w_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}+\text{MX}\}}}{\left(1 - w_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}+\text{MX}\}}\right) M_{\text{MX}}} 1000 \quad (3.1)$$

$$b_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}} = \frac{m_{\text{MX}}}{m_{\text{H}_2\text{O}+\text{MEG}} M_{\text{MX}}} 1000 \quad (3.2)$$

3.1.2 Cálculo da fração molar do MEG livre de sal na mistura de solventes

A fração molar do MEG livre de sal na mistura de solventes ($x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}}$) foi calculada em função das propriedades e unidades dos dados experimentais consultados:

$$x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}} = \frac{\frac{m_{\text{MEG}}}{M_{\text{MEG}}}}{\frac{m_{\text{H}_2\text{O}}}{M_{\text{H}_2\text{O}}} + \frac{m_{\text{MEG}}}{M_{\text{MEG}}}} \quad (3.3)$$

$$x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}} = \frac{\frac{1 - w_{\text{H}_2\text{O}}^{\{\text{H}_2\text{O}+\text{MEG}\}}}{M_{\text{H}_2\text{O}}}}{\frac{w_{\text{H}_2\text{O}}^{\{\text{H}_2\text{O}+\text{MEG}\}}}{M_{\text{H}_2\text{O}}} + \frac{1 - w_{\text{H}_2\text{O}}^{\{\text{H}_2\text{O}+\text{MEG}\}}}{M_{\text{MEG}}}} \quad (3.4)$$

$$x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}} = \frac{\frac{w_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}}}{M_{\text{MEG}}}}{\frac{1 - w_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}}}{M_{\text{H}_2\text{O}}} + \frac{w_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}}}{M_{\text{MEG}}}} \quad (3.5)$$

$$x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}} = \frac{\frac{w_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}+\text{MX}\}}}{M_{\text{MEG}}}}{\frac{1 - w_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}+\text{MX}\}}}{M_{\text{H}_2\text{O}}} - \frac{w_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}+\text{MX}\}}}{M_{\text{MX}}} + \frac{w_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}+\text{MX}\}}}{M_{\text{MEG}}}} \quad (3.6)$$

3.2 Constantes dielétricas

3.2.1 Cálculo das constantes dielétricas dos solventes puros

As constantes dielétricas dos solventes puros ($\varepsilon_{\text{r,H}_2\text{O}}$ e $\varepsilon_{\text{r,MEG}}$) foram calculadas em função da temperatura (T) pelas seguintes equações (ÅKERLÖF, 1932):

$$\varepsilon_{\text{r,H}_2\text{O}} = 10^{1,9051 - 0,00205(T - 293,15)} \quad (3.7)$$

$$\varepsilon_{\text{r,MEG}} = 10^{1,5872 - 0,00224(T - 293,15)} \quad (3.8)$$

3.2.2 Cálculo das constantes dielétricas da mistura de solventes

A constante dielétrica da mistura de solventes ($\varepsilon_{\text{r,H}_2\text{O}+\text{MEG}}$) foi calculada em função da temperatura (T), da fração molar do MEG na mistura de solventes livre de sal ($x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}}$) e das Constantes Dielétricas dos solventes puros na mesma temperatura

(ε_{r,H_2O} e $\varepsilon_{r,MEG}$) pela equação (JOUYBAN; SOLTANPOUR; CHAN, 2004):

$$\varepsilon_{r,H_2O+MEG} = \exp \left\{ x_{H_2O}^{\{H_2O+MEG\}} \ln \varepsilon_{r,H_2O} + x_{MEG}^{\{H_2O+MEG\}} \ln \varepsilon_{r,MEG} + \frac{x_{H_2O}^{\{H_2O+MEG\}} x_{MEG}^{\{H_2O+MEG\}}}{T} \left[153,6 + 57,3 \left(x_{H_2O}^{\{H_2O+MEG\}} - x_{MEG}^{\{H_2O+MEG\}} \right) \right] \right\} \quad (3.9)$$

sendo $x_{H_2O}^{\{H_2O+MEG\}} = 1 - x_{MEG}^{\{H_2O+MEG\}}$.

3.3 Densidades

3.3.1 Cálculo das densidades dos solventes puros

As densidades dos solventes puros (ρ_{H_2O} e ρ_{MEG}) foram calculadas em função da temperatura (T) usando as massas molares dos solventes puros (M_{H_2O} e M_{MEG}) (LIDE; HAYNES, 2009) pelas equações (GREEN; PERRY, 2007):

$$\rho_{H_2O} = \left(17,863 + 58,606\tau^{0,35} - 95,396\tau^{\frac{2}{3}} + 213,89\tau - 141,26\tau^{\frac{4}{3}} \right) M_{H_2O} \quad (3.10)$$

$$\rho_{MEG} = \frac{1,315}{0,25125^{1+\left(1-\frac{T}{720}\right)^{0,21868}}} M_{MEG} \quad (3.11)$$

sendo $\tau = 1 - \frac{T}{647,096}$.

3.3.2 Cálculo das densidades da mistura de solventes

A densidade da mistura de solventes (ρ_{H_2O+MEG}) foi calculada em função da Densidade dos solventes puros (ρ_{H_2O} e ρ_{MEG}) e da fração molar do MEG na mistura de solventes livre de sal ($x_{MEG}^{\{H_2O+MEG\}}$) usando as massas molares dos solventes puros (M_{H_2O} e M_{MEG}) (LIDE; HAYNES, 2009), assumindo regra de mistura ideal para o seu inverso (volume específico), pela equação:

$$\rho_{H_2O+MEG} = \rho_{H_2O+MEG} = \frac{1}{w_{H_2O}^{\{H_2O+MEG\}}/\rho_{H_2O} + w_{MEG}^{\{H_2O+MEG\}}/\rho_{MEG}} \quad (3.12)$$

sendo $x_{H_2O}^{\{H_2O+MEG\}} = 1 - x_{MEG}^{\{H_2O+MEG\}}$, $w_{MEG}^{\{H_2O+MEG\}} = \frac{x_{MEG}^{\{H_2O+MEG\}} M_{MEG}}{x_{H_2O}^{\{H_2O+MEG\}} M_{H_2O} + x_{MEG}^{\{H_2O+MEG\}} M_{MEG}}$ e $w_{H_2O}^{\{H_2O+MEG\}} = 1 - w_{MEG}^{\{H_2O+MEG\}}$.

3.4 Coeficiente de atividade

3.4.1 Cálculo do coeficiente de atividade do sal pelo Modelo de Pitzer

O modelo de Pitzer (PITZER, 1973) foi empregado para o cálculo do coeficiente de atividade do sal (γ_{MX}):

$$I_b = \frac{b_M z_M^2 + b_X z_X^2}{2} \quad (3.13)$$

$$X = \frac{2}{b} \ln \left(1 + b I_b^{0,5} \right) + \frac{I_b^{0,5}}{1 + b I_b^{0,5}} \quad (3.14)$$

$$f_{MX}^\gamma = -A_\phi X \quad (3.15)$$

$$Y = \frac{1 - e^{-\alpha I_b^{0,5}} (1 + \alpha I_b^{0,5} - 0,5 \alpha^2 I_b)}{\alpha^2 I_b} \quad (3.16)$$

$$B_{MX}^\gamma = 2 \left(\beta_{MX}^{(0)} + \beta_{MX}^{(1)} Y \right) \quad (3.17)$$

$$C_{MX}^\gamma = 1,5 C_{MX}^\phi \quad (3.18)$$

$$\ln \gamma_{MX} = |z_M z_X| f_{MX}^\gamma + \frac{2b_{MX} \nu_M \nu_X}{\nu_{MX}} \left[B_{MX}^\gamma + b_{MX} (\nu_M \nu_X)^{0,5} C_{MX}^\gamma \right] \quad (3.19)$$

onde: $b_M = \nu_M b_{MX}$, $b_X = \nu_X b_{MX}$, $\nu_{MX} = \nu_M + \nu_X$, b_M , b_M e b_X são as molalidades do sal, do cátion e do ânion, respectivamente; ν_{MX} , ν_M e ν_X são os coeficientes estequiométricos do sal, do cátion e do ânion, respectivamente; z_M , z_X são as cargas do cátion e do ânion, respectivamente; I_b é a força iônica; f_{MX}^γ é a função de Debye-Hückel do sal; B_{MX}^γ e C_{MX}^γ são formas adotadas para o segundo e terceiro coeficientes do virial; A_ϕ é a constante de Debye-Hückel; X e Y são funções para simplificação da representação do modelo; C_{MX}^ϕ é semelhante a C_{MX}^γ , exceto que é utilizado na equação do Modelo de Pitzer para o coeficiente osmótico ao invés do de atividade; b , α são parâmetros cujos valores otimizados (PITZER, 1973) são $b = 1,2$ e $\alpha = 2,0$, respectivamente; e, $\beta_{MX}^{(0)}$ e $\beta_{MX}^{(1)}$ também são parâmetros do modelo.

3.4.2 Parâmetros do modelo de Pitzer em água

3.4.2.1 Cálculo dos parâmetros do modelo de Pitzer para o NaCl

Os valores de $\beta_{MX}^{(0)}$, $\beta_{MX}^{(1)}$ e C_{MX}^γ para o NaCl em água como função da temperatura (T) são dados pelas equações (SILVESTER; PITZER, 1977):

$$\begin{aligned} \beta_{NaCl}^{(0)\{H_2O\}}(T) = & 0,0765 - 777,03 \left(\frac{1}{T} - \frac{1}{298,15} \right) - 4,4706 \ln \left(\frac{T}{298,15} \right) + \\ & + 0,008946 (T - 298,15) - 0,0000033158 (T^2 - 298,15^2) \end{aligned} \quad (3.20)$$

$$\begin{aligned} \beta_{NaCl}^{(1)\{H_2O\}}(T) = & 0,2664 + 0,000061608 (T - 298,15) + \\ & + 0,0000010715 (T^2 - 298,15^2) \end{aligned} \quad (3.21)$$

$$C_{\text{NaCl}}^{\phi\{\text{H}_2\text{O}\}}(T) = 0,00127 + 33,317 \left(\frac{1}{T} - \frac{1}{298,15} \right) + \\ + 0,09421 \ln \left(\frac{T}{298,15} \right) - 0,00004655 (T - 298,15) \quad (3.22)$$

3.4.2.2 Cálculo dos parâmetros do modelo de Pitzer para o KCl

Os valores de $\beta^{(0)}$, $\beta^{(1)}$ e C_{MX}^γ para o KCl em água como função da temperatura (T) são dados pelas equações (ARCHER, 1999):

$$\beta_{\text{KCl}}^{(0)\{\text{H}_2\text{O}\}}(T) = 0,413229483398493 - 0,000870121476114027 (T - 298,15) + \\ + 0,00000101413736179231 (T - 298,15)^2 + \\ - \frac{1,99822538522801}{T - 225} - \frac{99,8120581680816}{T} \quad (3.23)$$

água

$$\beta_{\text{KCl}}^{(1)\{\text{H}_2\text{O}\}}(T) = 0,206691413598171 + 0,00102544606022162 (T - 298,15) + \\ - \frac{1883,49608000903}{(T - 225)^3} \quad (3.24)$$

$$C_{\text{KCl}}^{\phi\{\text{H}_2\text{O}\}}(T) = -0,00133515934994478 + \frac{0,234117693834228}{T - 225} + \\ - \frac{0,75896583546707}{T} \quad (3.25)$$

3.4.3 Constante de Debye-Hückel

3.4.3.1 Cálculo da constante de Debye-Hückel para a água

A constante de Debye-Hückel para água ($A_{\phi,\text{H}_2\text{O}}$) foi calculada em função da temperatura (T) pela equação (CHEN et al., 1982):

$$A_{\phi,\text{H}_2\text{O}}(T) = -61,44534e^{\frac{T-273,15}{273,15}} + 2,864468e^{2\left(\frac{T-273,15}{273,15}\right)} + 183,5379 \ln \left(\frac{T}{273,15} \right) + \\ - 0,6820223 (T - 273,15) + 0,0007875695 (T^2 - 273,15^2) + \\ + 58,95788 \left(\frac{273,15}{T} \right) \quad (3.26)$$

3.4.3.2 Cálculo da constante de Debye-Hückel para o MEG e para a mistura de solventes

As constantes de Debye-Hückel para o MEG e para a mistura de solventes ($A_{\phi,\text{MEG}}$ e $A_{\phi,\text{H}_2\text{O}+\text{MEG}}$, respectivamente) foram calculadas a partir das constante de Debye-Hückel, densidade e constante dielétrica da água ($A_{\phi,\text{H}_2\text{O}}$, $\rho_{\text{H}_2\text{O}}$ e $\varepsilon_{\text{r,H}_2\text{O}}$, respectivamente), e das densidade e constante dielétrica para o MEG ou para a mistura de solventes (ρ_{MEG} e $\varepsilon_{\text{r,MEG}}$, respectivamente, ou $\rho_{\text{H}_2\text{O}+\text{MEG}}$ e $\varepsilon_{\text{r,H}_2\text{O}+\text{MEG}}$, respectivamente), tudo calculado a mesma temperatura (T) por equação obtida a partir da razão entre as equações de definição

da constante de Debye-Hückel para a água e para um outro em solvente (ou mistura de solventes):

$$A_{\phi,\text{solvente}} = \left(\frac{\rho_{\text{solvente}}}{\rho_{\text{H}_2\text{O}}} \right)^{0,5} \left(\frac{\varepsilon_{r,\text{H}_2\text{O}}}{\varepsilon_{r,\text{solvente}}} \right)^{1,5} A_{\phi,\text{H}_2\text{O}} \quad (3.27)$$

3.4.4 Cálculo do coeficiente de atividade do sal em MEG

O coeficiente de atividade do sal em MEG ($\gamma_{\text{MX}}^{\{\text{MEG}\}}$) foi calculado por meio da equação abaixo (LORIMER, 1993) como função da temperatura (T), do coeficiente de atividade do sal em água ($\gamma_{\text{MX}}^{\{\text{H}_2\text{O}\}}$) das molalidades do sal em água e MEG ($b_{\text{MX}}^{\{\text{H}_2\text{O}\}}$ e $b_{\text{MX}}^{\{\text{MEG}\}}$, respectivamente) e da Energia Livre de Gibbs de Transferência da água para o MEG ($\Delta_{\text{tr}}G_{\text{H}_2\text{O} \rightarrow \text{MEG}}^0$), sendo os valores para esta última, tanto para o NaCl quanto para o KCl, iguais à $\Delta_{\text{tr}}G_{\text{H}_2\text{O} \rightarrow \text{MEG}}^0 = 7000$ (MARCUS, 1985). O valor da constante universal dos gases utilizado é $R = 8,314472$ (LIDE; HAYNES, 2009).

$$\ln \gamma_{\text{MX}}^{\{\text{MEG}\}} = \ln \gamma_{\text{MX}}^{\{\text{H}_2\text{O}\}} + \ln \left(\frac{b_{\text{MX}}^{\{\text{H}_2\text{O}\}}}{b_{\text{MX}}^{\{\text{MEG}\}}} \right) - \frac{\Delta_{\text{tr}}G_{\text{H}_2\text{O} \rightarrow \text{MEG}}^0}{\nu_{\text{MX}} RT} \quad (3.28)$$

3.4.5 Parâmetros do modelo de Pitzer em mistura de solventes

Usando a metodologia de Lorimer (LORIMER, 1993) que considera constantes os parâmetros $\beta_{\text{MX}}^{(0)}$ e C_{MX}^ϕ , em relação aos solventes, e que o parâmetro $\beta_{\text{MX}}^{(1)}$, em relação aos solventes, apresenta dependência linear com a constante dielétrica, isto em escala logarítmica, podemos estender o modelo de Pitzer para mistura de solventes:

$$\beta_{\text{MX}}^{(0)\{\text{solvente}\}} = \beta_{\text{MX}}^{(0)\{\text{H}_2\text{O}\}} \quad (3.29)$$

$$\beta_{\text{MX}}^{(1)\{\text{solvente}\}} = \beta_0 \exp(A \ln \varepsilon_{r,\text{solvente}} + B) \quad (3.30)$$

$$C_{\text{MX}}^{\phi\{\text{solvente}\}} = C_{\text{MX}}^{\phi\{\text{H}_2\text{O}\}} \quad (3.31)$$

onde $\beta_0 = 1$, $A = \frac{\ln(\beta_{\text{MX}}^{(1)\{\text{H}_2\text{O}\}}/\beta_{\text{MX}}^{(1)\{\text{MEG}\}})}{\ln(\varepsilon_{r,\text{H}_2\text{O}}/\varepsilon_{r,\text{MEG}})}$ e $B = \ln \frac{\beta_{\text{MX}}^{(1)\{\text{H}_2\text{O}\}}}{\beta_0} - A \ln \varepsilon_{r,\text{H}_2\text{O}}$.

3.5 Algoritmos

Através do conjunto de equações apresentados anteriormente podemos calcular as propriedades de excesso: molalidade ($\left[\ln \left(\frac{b_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}}}{b_0} \right) \right]^E$), coeficiente de atividade ($\left[\ln \left(\gamma_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}} \right) \right]^E$) e potencial químico ($\left[\mu_{\text{MX}}^{0\{\text{H}_2\text{O}+\text{MEG}\}} \right]^E$), para tanto usamos a definição de propriedade de excesso (Equação 2.24) e a regra de mistura ideal:

$$M^{\text{id}} = Mx + M(1-x) \quad (3.32)$$

onde M representa a propriedade molar e x a fração molar.

3.5.1 Cálculo das propriedades de excesso do sal em mistura de solventes

Calcula-se as propriedades de excesso da seguinte forma:

- a) Entra-se $b_{MX}^{\{H_2O+MEG\}}$, T e $x_{MEG}^{\{H_2O+MEG\}}$;
- b) Calcula-se:
 - $A_{\phi,H_2O} = A_{\phi,H_2O}(T)$ (Equação 3.26);
 - $b_{MX}^{\{H_2O\}} = b_{MX}^{\{H_2O\}}(T)$;
 - $\beta_{MX}^{(0)\{H_2O\}} = \beta_{MX}^{(0)\{H_2O\}}(T)$ (Equação 3.20 e Equação 3.23);
 - $\beta_{MX}^{(1)\{MEG\}} = \beta_{MX}^{(1)\{MEG\}}(T)$ (Equação 3.21 e Equação 3.24);
 - $C_{MX}^{\phi\{H_2O\}} = C_{MX}^{\phi\{H_2O\}}(T)$ (Equação 3.22 e Equação 3.25);
- c) Calcula-se:
 - $I_b^{\{H_2O\}} = I_b^{\{H_2O\}}(b_{MX}^{\{H_2O\}})$ (Equação 3.13);
 - $C_{MX}^{\gamma\{H_2O\}} = C_{MX}^{\gamma\{H_2O\}}(C_{MX}^{\phi\{H_2O\}})$ (Equação 3.18);
- d) Calcula-se:
 - $X^{\{H_2O\}} = X^{\{H_2O\}}(I_b^{\{H_2O\}})$ (Equação 3.14);
 - $Y^{\{H_2O\}} = Y^{\{H_2O\}}(I_b^{\{H_2O\}})$ (Equação 3.16);
- e) Calcula-se:
 - $f_{MX}^{\gamma\{H_2O\}} = f_{MX}^{\gamma\{H_2O\}}(A_{\phi,H_2O}; X^{\{H_2O\}})$ (Equação 3.15);
 - $B_{MX}^{\gamma\{H_2O\}} = B_{MX}^{\gamma\{H_2O\}}(\beta_{MX}^{(0)\{H_2O\}}; \beta_{MX}^{(1)\{H_2O\}}; Y^{\{H_2O\}})$ (Equação 3.17);
- f) Calcula-se o logaritmo do coeficiente de atividade do sal em água:

$$\ln \gamma_{MX}^{\{H_2O\}} = \ln \gamma_{MX}^{\{H_2O\}}(b_{MX}^{\{H_2O\}}; f_{MX}^{\gamma\{H_2O\}}; B_{MX}^{\gamma\{H_2O\}}; C_{MX}^{\gamma\{H_2O\}})$$
 (Equação 3.19);
- g) Calcula-se:

$$b_{MX}^{\{MEG\}} = b_{MX}^{\{MEG\}}(T)$$
- h) Calcula-se o logaritmo do coeficiente de atividade do sal em MEG:

$$\ln \gamma_{MX}^{\{MEG\}} = \ln \gamma_{MX}^{\{MEG\}}(\ln \gamma_{MX}^{\{H_2O\}}; b_{MX}^{\{H_2O\}}; b_{MX}^{\{MEG\}}; T)$$
 (Equação 3.28);
- i) Calcula-se:
 - $\rho_{H_2O} = \rho_{H_2O}(T)$ (Equação 3.10);
 - $\rho_{MEG} = \rho_{MEG}(T)$ (Equação 3.11);
 - $\varepsilon_{r,H_2O} = \varepsilon_{r,H_2O}(T)$ (Equação 3.7);
 - $\varepsilon_{r,MEG} = \varepsilon_{r,MEG}(T)$ (Equação 3.8);
 - $I_b^{\{MEG\}} = I_b^{\{MEG\}}(b_{MX}^{\{MEG\}})$ (Equação 3.13);
 - $\beta_{MX}^{(0)\{MEG\}} = \beta_{MX}^{(0)\{MEG\}}(\beta_{MX}^{(0)\{H_2O\}})$ (Equação 3.29);
 - $C_{MX}^{\phi\{MEG\}} = C_{MX}^{\phi\{MEG\}}(C_{MX}^{\phi\{H_2O\}})$ (Equação 3.31);
- j) Calcula-se:

- $A_{\phi, \text{MEG}} = A_{\phi, \text{MEG}}(A_{\phi, \text{MEG}}; \rho_{\text{H}_2\text{O}}; \rho_{\text{MEG}}; \varepsilon_{\text{r}, \text{H}_2\text{O}}; \varepsilon_{\text{r}, \text{MEG}})$ (Equação 3.27);
- $C_{\text{MX}}^{\gamma\{\text{MEG}\}} = C_{\text{MX}}^{\gamma\{\text{MEG}\}}(C_{\text{MX}}^{\phi\{\text{MEG}\}})$ (Equação 3.18);

k) Calcula-se:

- $X^{\{\text{MEG}\}} = X^{\{\text{MEG}\}}(I_b^{\{\text{MEG}\}})$ (Equação 3.14);
- $Y^{\{\text{MEG}\}} = Y^{\{\text{MEG}\}}(I_b^{\{\text{MEG}\}})$ (Equação 3.16);

l) Calcula-se:

- $f_{\text{MX}}^{\gamma\{\text{MEG}\}} = f_{\text{MX}}^{\gamma\{\text{MEG}\}}(A_{\phi, \text{MEG}}; X^{\{\text{MEG}\}})$ (Equação 3.15);

m) Resolve-se:

$$\ln \gamma_{\text{MX}}^{\{\text{MEG}\}} - \ln \gamma_{\text{MX}}(b_{\text{MX}}^{\{\text{MEG}\}}; f_{\text{MX}}^{\gamma\{\text{MEG}\}}; B_{\text{MX}}^{\gamma\{\text{MEG}\}}; C_{\text{MX}}^{\gamma\{\text{MEG}\}}) = 0 \text{ (Equação 3.19)}$$

para $\beta_{\text{MX}}^{(1)\{\text{MEG}\}}$, sendo:

$$B_{\text{MX}}^{\gamma\{\text{MEG}\}} = B_{\text{MX}}^{\gamma\{\text{MEG}\}}(\beta_{\text{MX}}^{(0)\{\text{MEG}\}}; \beta_{\text{MX}}^{(1)\{\text{MEG}\}}; Y^{\{\text{MEG}\}}) \text{ (Equação 3.17);}$$

n) Calcula-se:

- $\rho_{\text{H}_2\text{O}+\text{MEG}} = \rho_{\text{H}_2\text{O}+\text{MEG}}(\rho_{\text{H}_2\text{O}}; \rho_{\text{MEG}}; x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.12)
- $\varepsilon_{\text{r}, \text{H}_2\text{O}+\text{MEG}} = \varepsilon_{\text{r}, \text{H}_2\text{O}+\text{MEG}}(\varepsilon_{\text{H}_2\text{O}}; \varepsilon_{\text{MEG}}; x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.9);

o) Calcula-se:

- $A_{\phi, \text{H}_2\text{O}+\text{MEG}} = A_{\phi, \text{H}_2\text{O}+\text{MEG}}(A_{\phi, \text{H}_2\text{O}}; \rho_{\text{H}_2\text{O}}; \rho_{\text{H}_2\text{O}+\text{MEG}}; \varepsilon_{\text{r}, \text{H}_2\text{O}}; \varepsilon_{\text{r}, \text{H}_2\text{O}+\text{MEG}})$ (Equação 3.27);
- $I_b^{\{\text{H}_2\text{O}+\text{MEG}\}} = I_b^{\{\text{H}_2\text{O}+\text{MEG}\}}(b_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.13);
- $\beta_{\text{MX}}^{(0)\{\text{H}_2\text{O}+\text{MEG}\}} = \beta_{\text{MX}}^{(0)\{\text{H}_2\text{O}+\text{MEG}\}}(\beta_{\text{MX}}^{(0)\{\text{H}_2\text{O}\}})$ (Equação 3.29);
- $\beta_{\text{MX}}^{(1)\{\text{H}_2\text{O}+\text{MEG}\}} = \beta_{\text{MX}}^{(1)\{\text{H}_2\text{O}+\text{MEG}\}}(\beta_{\text{MX}}^{(0)\{\text{H}_2\text{O}\}}; \beta_{\text{MX}}^{(0)\{\text{MEG}; \varepsilon_{\text{r}, \text{H}_2\text{O}}; \varepsilon_{\text{r}, \text{MEG}}\}})$ (Equação 3.30);
- $C_{\text{MX}}^{\phi\{\text{H}_2\text{O}+\text{MEG}\}} = C_{\text{MX}}^{\phi\{\text{H}_2\text{O}+\text{MEG}\}}(C_{\text{MX}}^{\phi\{\text{H}_2\text{O}\}})$ (Equação 3.31);

p) Calcula-se:

- $X^{\{\text{H}_2\text{O}+\text{MEG}\}} = X^{\{\text{H}_2\text{O}+\text{MEG}\}}(I_b^{\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.14);
- $Y^{\{\text{H}_2\text{O}+\text{MEG}\}} = Y^{\{\text{H}_2\text{O}+\text{MEG}\}}(I_b^{\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.16);
- $C_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}} = C_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}}(C_{\text{MX}}^{\phi\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.18);

q) Calcula-se:

- $f_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}} = f_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}}(A_{\phi, \text{H}_2\text{O}+\text{MEG}}; X^{\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.15);
- $B_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}} = B_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}}(\beta_{\text{MX}}^{(0)\{\text{H}_2\text{O}+\text{MEG}\}}; \beta_{\text{MX}}^{(1)\{\text{H}_2\text{O}+\text{MEG}\}}; Y^{\{\text{H}_2\text{O}+\text{MEG}\}})$ (Equação 3.17);

r) Calcula-se o logaritmo do coeficiente de atividade do sal na mistura de solventes:

$$\ln \gamma_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}} = \ln \gamma_{\text{MX}}(b_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}}; f_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}}; B_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}}; C_{\text{MX}}^{\gamma\{\text{H}_2\text{O}+\text{MEG}\}});$$

s) Calcula-se o logaritmo ideal da molalidade do sal na mistura de solventes:

$$[\ln b_{\text{MX}}^{\{\text{H}_2\text{O}+\text{MEG}\}}]_{\text{id}} = \ln b_{\text{MX}}^{\{\text{H}_2\text{O}\}} x_{\text{H}_2\text{O}}^{\{\text{H}_2\text{O}+\text{MEG}\}} + \ln b_{\text{MX}}^{\{\text{MEG}\}} x_{\text{MEG}}^{\{\text{H}_2\text{O}+\text{MEG}\}};$$

- t) Calcula-se o logaritmo de excesso da molalidade do sal na mistura de solventes:
- $$\left[\ln b_{MX}^{\{H_2O+MEG\}} \right]^E = \ln b_{MX}^{\{H_2O+MEG\}} - \left[\ln b_{MX}^{\{H_2O+MEG\}} \right]^{id};$$
- u) Calcula-se o logaritmo ideal do coeficiente de atividade do sal na mistura de solventes:
- $$\left[\ln \gamma_{MX}^{\{H_2O+MEG\}} \right]^{id} = \ln \gamma_{MX}^{\{H_2O\}} x_{H_2O}^{\{H_2O+MEG\}} + \ln \gamma_{MX}^{\{MEG\}} x_{MEG}^{\{H_2O+MEG\}};$$
- v) Calcula-se o logaritmo de excesso do coeficiente de atividade do sal na mistura de solventes:
- $$\left[\ln \gamma_{MX}^{\{H_2O+MEG\}} \right]^E = \ln \gamma_{MX}^{\{H_2O+MEG\}} - \left[\ln \gamma_{MX}^{\{H_2O+MEG\}} \right]^{id};$$
- w) Calcula-se o potencial químico padrão de excesso do coeficiente de atividade do sal na mistura de solventes:
- $$\frac{\left[\mu_{MX}^{0\{H_2O+MEG\}} \right]^E}{\nu RT} = - \left[\ln b_{MX}^{\{H_2O+MEG\}} \right]^E - \left[\ln \gamma_{MX}^{\{H_2O+MEG\}} \right]^E$$

Após esta sequencia de cálculos, faz-se o ajuste dos resultados para o potência de excesso do sal na misturas de solventes em função da fração molar do MEG livre de sal e da temperatura, conforme a [Equação 3.33](#):

$$\frac{\left[\mu_{MX}^{0\{H_2O+MEG\}} \right]^E}{nRT} = x_{MEG}^{\{H_2O+MEG\}} \left(1 - x_{MEG}^{\{H_2O+MEG\}} \right) \left[\theta_0 + \theta_1 T + (\theta_2 + \theta_3 T) x_{MEG}^{\{H_2O+MEG\}} + (\theta_4 + \theta_5 T) x_{MEG}^{\{H_2O+MEG\}^2} \right] \quad (3.33)$$

3.5.2 Cálculo das demais propriedades

A partir da [Equação 3.33](#) e de iteração com o procedimento descrito na seção anterior obtemos as outras propriedades, a saber:

- molalidade do sal em mistura de solventes;
- logaritmo do coeficiente de atividade do sal em mistura de solventes;
- logaritmo de excesso da molalidade do sal em mistura de solventes;
- logaritmo de excesso do coeficiente de atividade do sal em mistura de solventes.

CAPÍTULO 4

RESULTADOS E DISCUSSÕES

4 RESULTADOS E DISCUSSÕES

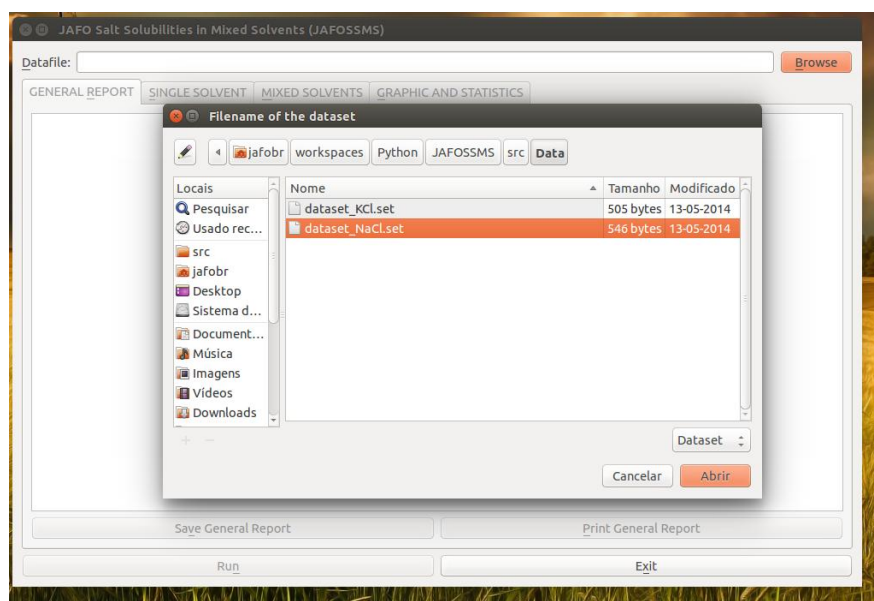
Os resultados e discussões são apresentados primeiramente pelo sal (NaCl ou KCl); seguido do sistema (binário ou ternário), onde para o binário são apresentados primeiro resultados para o sal com água e depois para o sal com MEG, e para o ternário são apresentados sempre primeiro o caso que usa os valores do potencial químico, sendo primeiro a regressão e em seguida a predição.

Antes de apresenta-se a maneira como estes resultados foram gerados através do programa desenvolvido.

4.1 Como os resultados foram calculados através do programa desenvolvido

Primeiramente escolhe-se o arquivo referente ao sistema a ser estudado por meio do botão "*Browse*" (Figura 16) ou caixa de texto "*Datafile*". Em seguida, executa-se os cálculos de regressão por meio da tecla "*Run*".

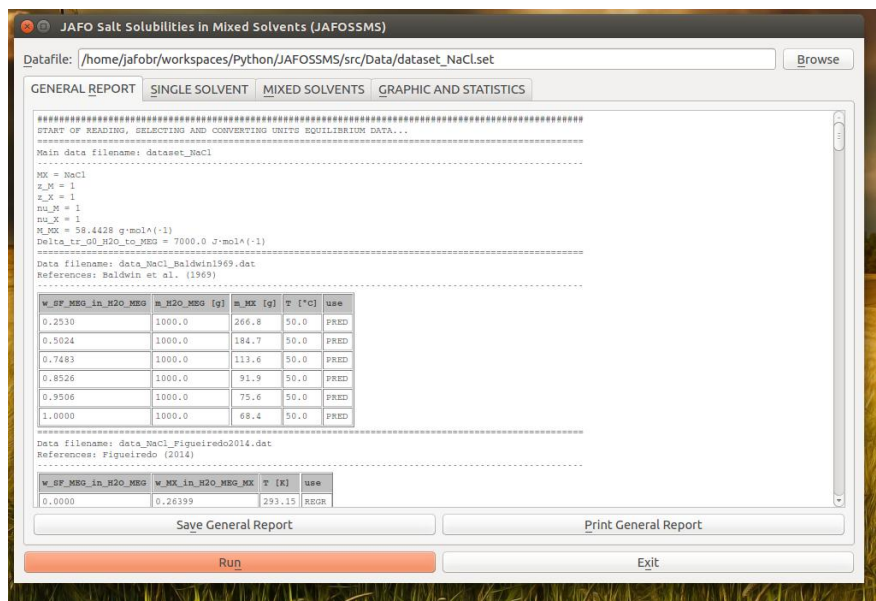
Figura 16 – Programa desenvolvido : Seleção do sistema a ser estudo.



Fonte: o autor.

Terminado os cálculos o programa retorna para o usuário com um relatório, na aba "*GENERAL REPORT*", de tudo que foi calculado e utilizado nos cálculos, inclusive toda a parte estatística (Figura 17). Este relatório pode ser copiado, salvo ou impresso.

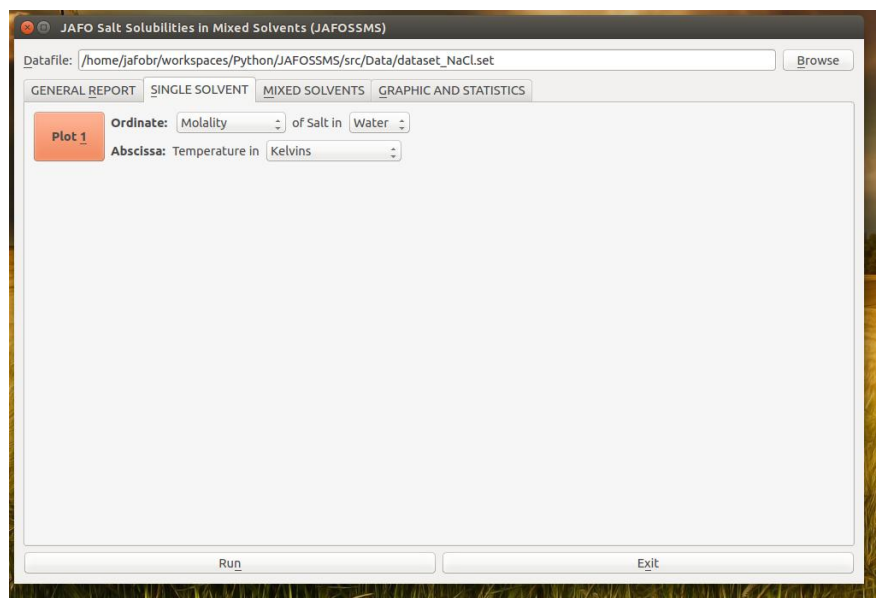
Figura 17 – Programa desenvolvido : Relatório gerado.



Fonte: o autor.

O programa oferece por meio da aba "*SINGLE SOLVENT*" botões de opções e caixas de seleção com opções para gerar gráficos binários (Figura 18).

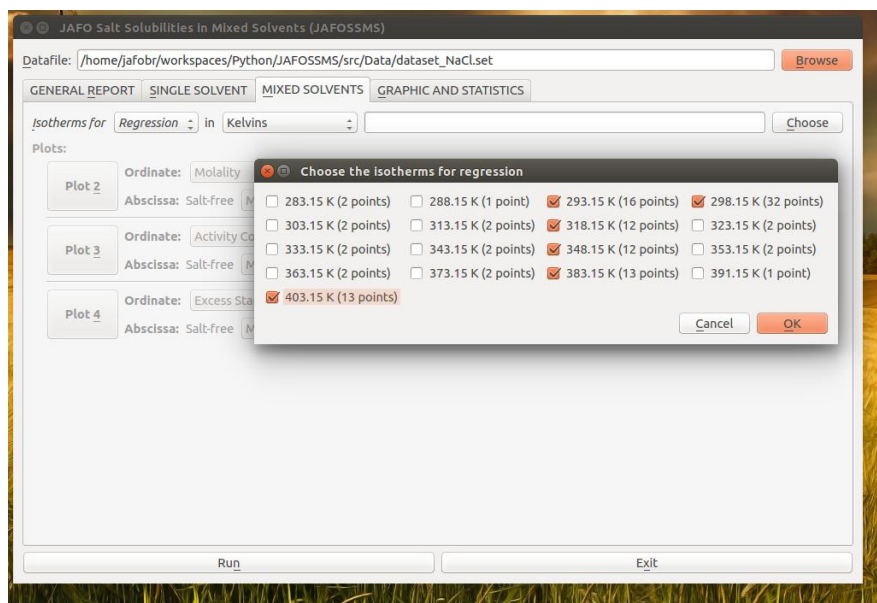
Figura 18 – Programa desenvolvido : Opções para geração de gráficos binários.



Fonte: o autor.

O programa oferece por meio da aba "*MIXED SOLVENTS*" botões de opções e caixas de seleção com opções para gerar os gráficos ternários (Figura 19), possuindo este a opção para escolhas de isotermas e caixas de seleção tanto para regressão quanto para predição.

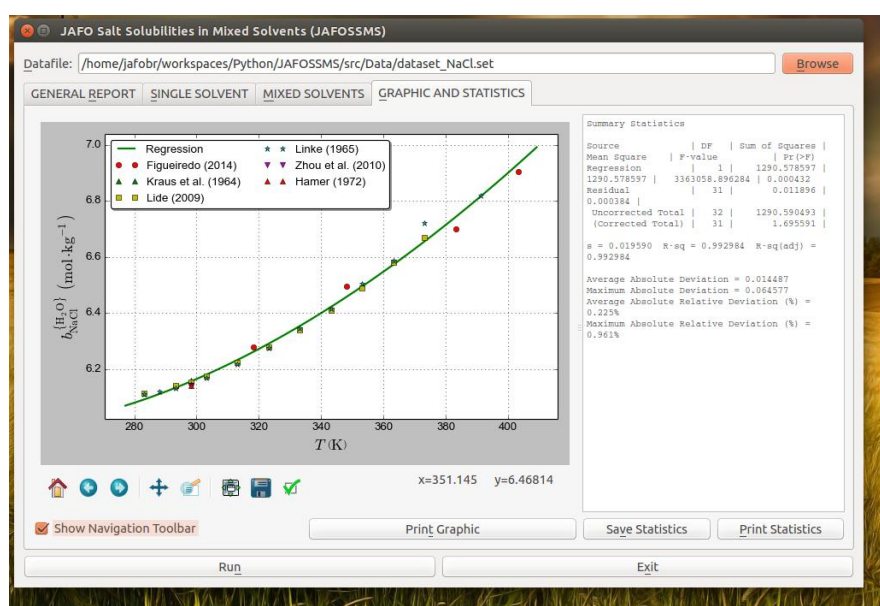
Figura 19 – Programa desenvolvido : Opções para geração de gráficos ternários.



Fonte: o autor.

Por fim, de acordo com a seleção, um gráfico e um pequeno resumo estatístico são gerados na aba "GRAPHIC AND STATISTICS" (Figura 20) em uma interface que permite ampliar regiões no gráfico, mudar os eixos, salvar a imagem, etc., além de poder-se copiar, salvar ou imprimir tanto o gráfico quanto o resumo estatístico.

Figura 20 – Programa desenvolvido : Gráficos.



Fonte: o autor.

4.2 Sistemas com NaCl

4.2.1 Sistemas Binários

4.2.1.1 Sistema H₂O+NaCl

Dados de solubilidade de NaCl em água foram ajustados por regressão não-linear usando-se a função:

$$b_{\text{NaCl}}^{\{\text{H}_2\text{O}\}} = \exp(\theta_0 + \theta_1 T + \theta_2 \ln T) \quad (4.1)$$

A Tabela 4 apresenta os valores obtidos para os parâmetros θ_i .

Tabela 4 – Parâmetros da Equação 4.1 obtidos por regressão para o Sistema H₂O+NaCl.

θ_i	Valor
θ_0	$5,169456 \pm 0,580512$
θ_1	$0,003319 \pm 0,000360$
θ_2	$-0,762002 \pm 0,120679$

Fonte: o autor

O resultado estatístico apresentado pelo programa para esta regressão é:

Non-Linear Least Squares Summary Statistics

Function: $y = T$, *theta: $\exp(\text{theta}[0] + \text{theta}[1]*T + \text{theta}[2]*\log(T))$

Source	DF	Sum of Squares	Mean Square	F-value	Pr(>F)
Regression	3	1252.870515	417.623505	1002896.737440	0.000000
Residual	28	0.011660	0.000416		
Uncorrected Total	31	1252.882174			
(Corrected Total)	30	1.651879			

alpha = 0.050000 t-critical = 2.048407

s = 0.020406 R-sq = 0.992942 R-sq(adj) = 0.992437

Parameter	Estimate	Asymptotic Std. Error	t-value	Pr(> t)	Asymptotic 95% Confidence Interval
					Lower Upper
P[0]	5.169456	0.580512	8.904999	0.000000	3.980332 6.358580
P[1]	0.003319	0.000360	9.218842	0.000000	0.002582 0.004057
P[2]	-0.762002	0.120679	-6.314290	0.000001	-1.009202 -0.514802

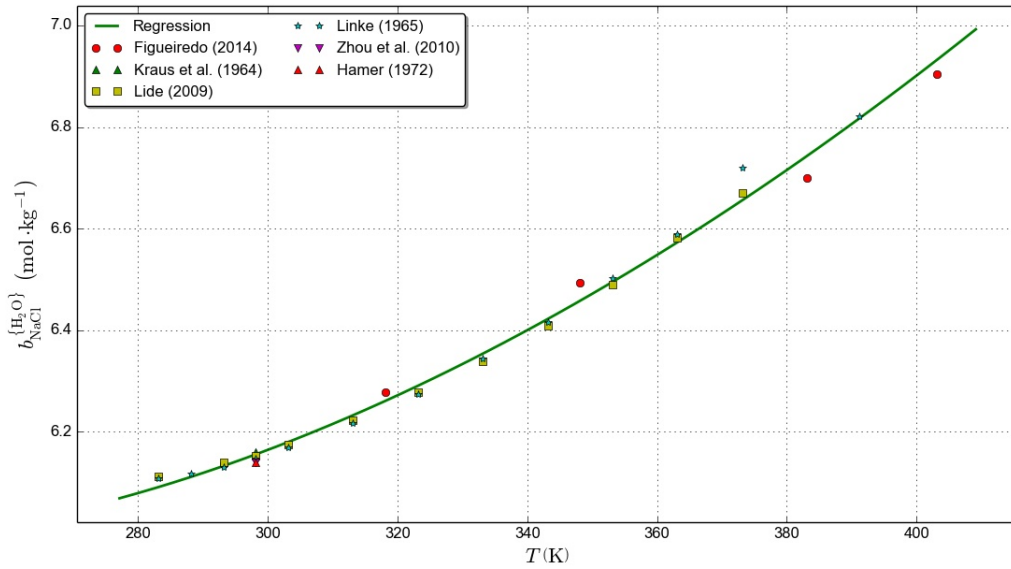
Parameter | Correlation Matrix

	P[0]	P[1]	P[2]
P[0]	1.000000	0.998459	-0.999955
P[1]	0.998459	1.000000	-0.998939
P[2]	-0.999955	-0.998939	1.000000

O ajuste foi bastante satisfatório apresentando um coeficiente de correlação de 0,992942 (R^2). Isto pode ser confirmado por meio da Figura 21, onde observa-se a forte tendência dos dados a aderirem a curva.

Observa-se na [Figura 21](#) que a solubilidade do NaCl aumenta com a temperatura de forma ligeiramente acima da linear em toda a faixa de temperatura correlacionada.

Figura 21 – Molalidade do NaCl em H₂O versus Temperatura.



Fonte: o autor.

4.2.1.2 Sistema MEG+NaCl

Dados de solubilidade de NaCl em MEG foram ajustados por regressão não-linear usando-se a função:

$$b_{\text{NaCl}}^{\{\text{MEG}\}} = \exp(\theta_0 + \theta_1 T) \quad (4.2)$$

A [Tabela 5](#) apresenta os valores obtidos para os parâmetros θ_i .

Tabela 5 – Parâmetros da [Equação 4.2](#) obtidos por regressão para o Sistema MEG+NaCl.

θ_i	Valor
θ_0	$0,413063 \pm 0,025925$
θ_1	$-0,000722 \pm 0,000079$

Fonte: o autor

O resultado estatístico apresentado pelo programa para esta regressão é:

Non-Linear Least Squares Summary Statistics

Function: $y = T$, *theta: $\exp(\text{theta}[0] + \text{theta}[1] * T)$

Source	DF	Sum of Squares	Mean Square	F-value	Pr(>F)
Regression	2	11.363925	5.681962	52079.033045	0.000019
Residual	6	0.000655	0.000109		

```
Uncorrected Total | 8 | 11.364579 |
(Corrected Total) | 7 | 0.010122 |
```

```
alpha = 0.050000 t-critical = 2.446912
```

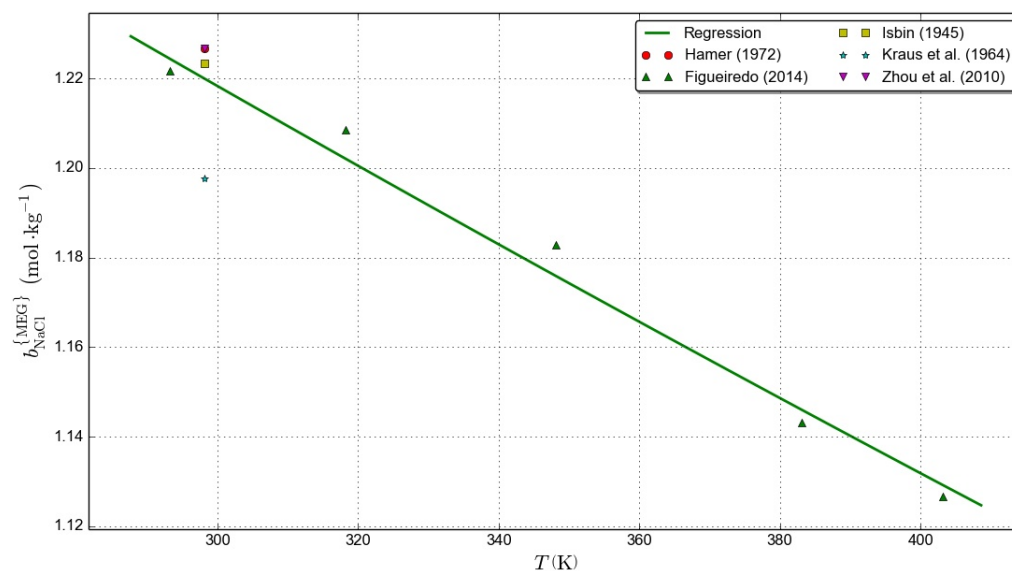
```
s = 0.010445 R-sq = 0.935327 R-sq(adj) = 0.924548
```

Parameter	Estimate	Asymptotic Std. Error	t-value	Pr(> t)	Asymptotic 95% Confidence Interval
					Lower Upper
P[0]	0.413063	0.025925	15.933301	0.000004	0.349628 0.476498
P[1]	-0.000722	0.000079	-9.197600	0.000093	-0.000915 -0.000530

Parameter	Correlation Matrix	
	P[0]	P[1]
P[0]	1.000000	-0.992832
P[1]	-0.992832	1.000000

O ajuste foi bastante satisfatório apresentando um coeficiente de correlação de 0,935327 (R^2). Isto pode ser confirmado por meio da Figura 22, onde observa-se a tendência dos dados a aderirem a curva.

Figura 22 – Molalidade do NaCl em MEG versus Temperatura.



Fonte: o autor.

Observa-se na Figura 22 que a solubilidade do NaCl em MEG diminui com a temperatura de forma visualmente linear em toda a faixa de temperatura correlacionada.

4.2.2 Sistemas Ternários (H₂O+MEG+NaCl)

4.2.2.1 Potencial químico padrão de excesso do NaCl em mistura de solventes (Regressão)

Dados de solubilidade de NaCl em H₂O+MEG foram empregados no cálculo de valores de potencial químico padrão de excesso. Em seguida, estes foram ajustados por regressão não-linear usando-se a função:

$$\frac{[\mu_{\text{NaCl}}^{\text{H}_2\text{O}+\text{MEG}}]^E}{nRT} = x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}} \left(1 - x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}}\right) \left[\theta_0 + \theta_1 T + (\theta_2 + \theta_3 T) x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}} + (\theta_4 + \theta_5 T) x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}}^2\right] \quad (4.3)$$

A [Tabela 6](#) apresenta os valores obtidos para os parâmetros θ_i .

Tabela 6 – Parâmetros da [Equação 4.3](#) obtidos por regressão para o Sistema H₂O+MEG+NaCl.

θ_i	Valor
θ_0	8.085783 ± 0.338373
θ_1	-0.007834 ± 0.000996
θ_2	-17.962668 ± 1.590260
θ_3	0.037503 ± 0.004678
θ_4	13.113160 ± 1.674601
θ_5	-0.016159 ± 0.004914

Fonte: o autor

O resultado estatístico apresentado pelo programa para esta regressão é:

Non-Linear Least Squares Summary Statistics

Function: y = (<function f3 at 0x4ac06e0>, 'function')

Source	DF	Sum of Squares	Mean Square	F-value	Pr(>F)
Regression	6	45.517278	7.586213	49024.181541	0.000000
Residual	101	0.015629	0.000155		
Uncorrected Total	107	45.532908			
(Corrected Total)	106	22.215424			

alpha = 0.050000 t-critical = 1.983731

s = 0.012440 R-sq = 0.999296 R-sq(adj) = 0.999262

Parameter	Estimate	Asymptotic Std. Error	t-value	Pr(> t)	Asymptotic 95% Confidence Interval
					Lower Upper
P[0]	8.085783	0.338373	23.896055	0.000000	7.414542 8.757025
P[1]	-0.007834	0.000996	-7.864318	0.000000	-0.009810 -0.005858
P[2]	-17.962668	1.590260	-11.295425	0.000000	-21.117317 -14.808019
P[3]	0.037503	0.004678	8.016076	0.000000	0.028222 0.046784
P[4]	13.113160	1.674601	7.830619	0.000000	9.791202 16.435117
P[5]	-0.016159	0.004914	-3.288321	0.001388	-0.025907 -0.006411

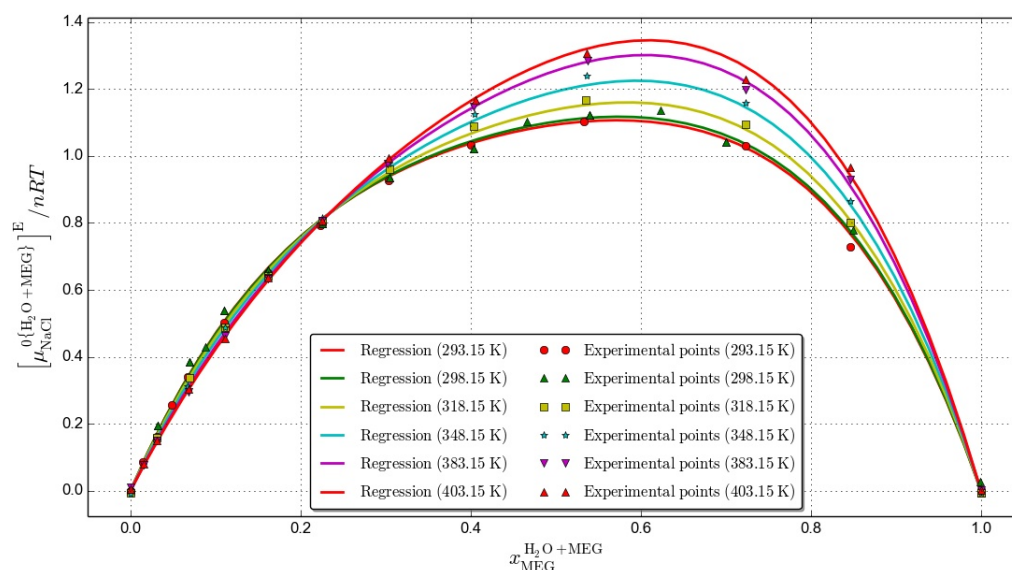
Parameter | Correlation Matrix

	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]
P[0]	1.000000	-0.992475	-0.938074	0.931542	0.850491	-0.846009
P[1]	-0.992475	1.000000	0.930903	-0.938490	-0.843257	0.851565
P[2]	-0.938074	0.930903	1.000000	-0.992489	-0.972733	0.966759
P[3]	0.931542	-0.938490	-0.992489	1.000000	0.964276	-0.972887
P[4]	0.850491	-0.843257	-0.972733	0.964276	1.000000	-0.992523
P[5]	-0.846009	0.851565	0.966759	-0.972887	-0.992523	1.000000

O ajuste foi bastante satisfatório apresentando um coeficiente de correlação de 0,999296 (R^2). Isto pode ser confirmado por meio da Figura 23, onde observa-se a excelente aderência dos valores de potencial químico padrão de excesso calculados a partir dos dados de solubilidade e metodologia empregados.

Observa-se na Figura 23 que o potencial químico padrão de excesso do NaCl na mistura de solventes, nas diversas isotermas, cresce desde zero até um valor máximo correspondente a fração molar de MEG livre de solvente próximo de 0,6 e depois decresce até o valor zero. Observa-se, também, que as isotermas se cruzam no valor de 0,22 da fração molar de MEG livre de solvente. Para valores de fração molar de MEG livre de solvente menores que este valor (0,22) o potencial químico padrão de excesso do NaCl na mistura de solventes diminui com o aumento da temperatura e para a faixa restante fração molar de MEG livre de solvente, tem um comportamento inverso.

Figura 23 – Isotermas do potencial químico padrão de excesso do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão).

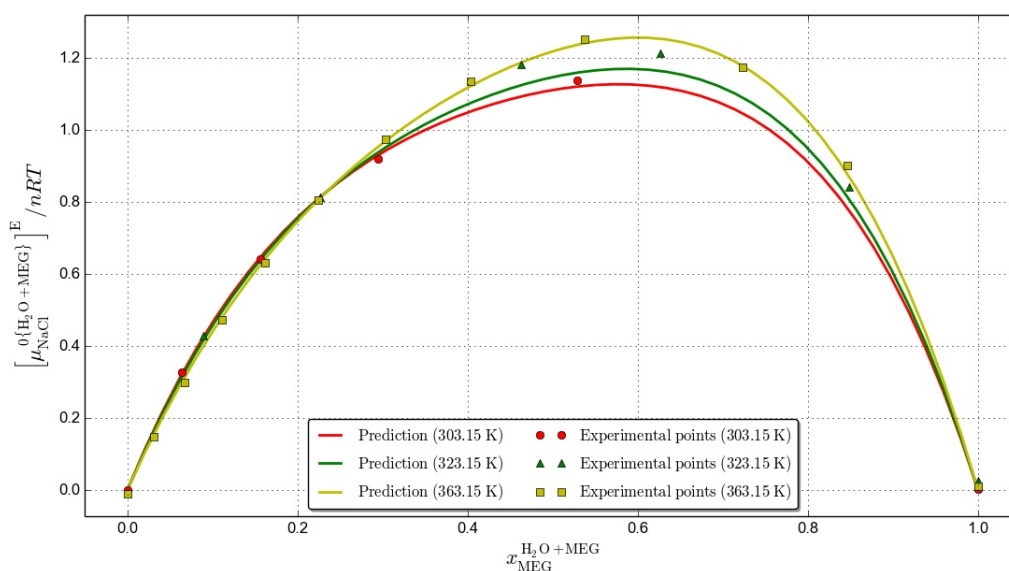


Fonte: o autor.

4.2.2.2 Potencial químico padrão de excesso do NaCl em mistura de solventes (Predição)

A Figura 24 representa as isotermas do potencial químico padrão de excesso do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl (Predição). Nesta figura observa-se novamente a excelente aderência dos valores de potencial químico padrão de excesso calculados a partir dos dados de solubilidade e metodologia empregados. Também observa-se o fenômeno de inversão em torno do mesmo valor citado na regressão, ou seja, 0,22 da fração molar de MEG livre de solvente.

Figura 24 – Isotermas do potencial químico padrão de excesso do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição).

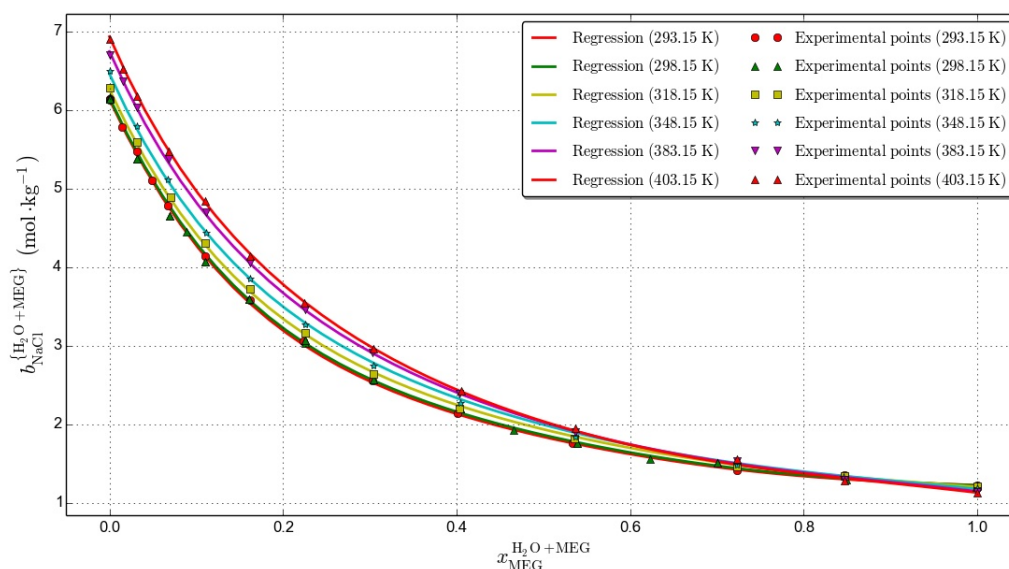


Fonte: o autor.

4.2.2.3 Molalidade do NaCl em mistura de solventes (Regressão)

A Figura 25 representa as isotermas da molalidade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl. Nesta figura observa-se a excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Também observa-se o fenômeno de inversão em torno do valor de 0,82 para a fração molar de MEG livre de solvente. Neste caso para valores de fração molar de MEG livre de solvente menores que 0,82 temos que a molalidade aumenta com a temperatura, e após o valor de 0,82 inverte-se o comportamento. Observa-se a diminuição da solubilidade com o aumento da concentração do MEG. A figura

Figura 25 – Isotermas da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão).

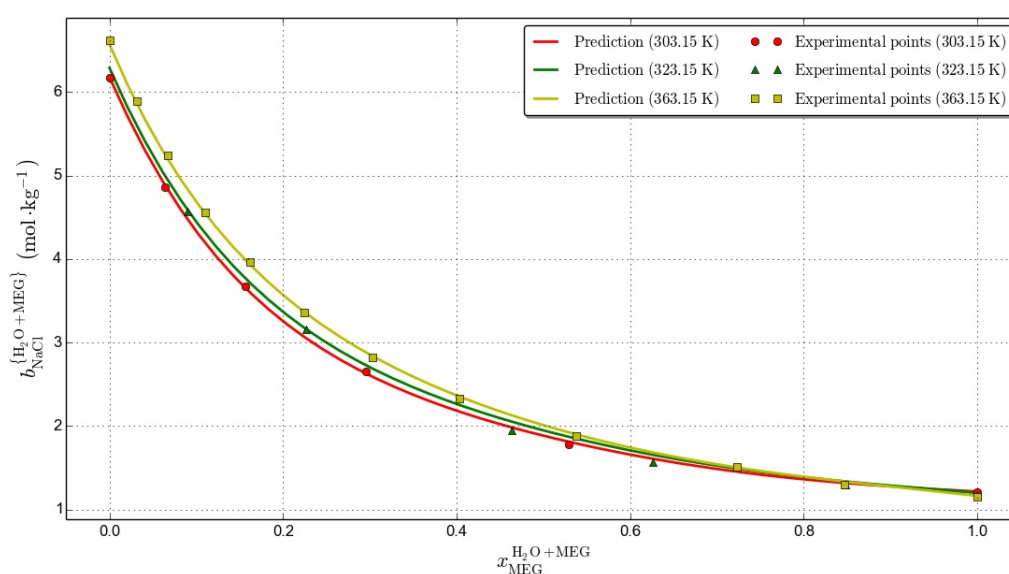


Fonte: o autor.

4.2.2.4 Molalidade do NaCl em mistura de solventes (Predição)

A Figura 26 representa as isotermas da molalidade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl (Predição). Nesta figura observa-se novamente a excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Também observa-se o fenômeno de inversão em torno do mesmo valor citado na regressão, ou seja, 0,82 da fração molar de MEG livre de solvente. Observa-se, também, a diminuição da solubilidade com o aumento da concentração do MEG.

Figura 26 – Isotermas da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição).

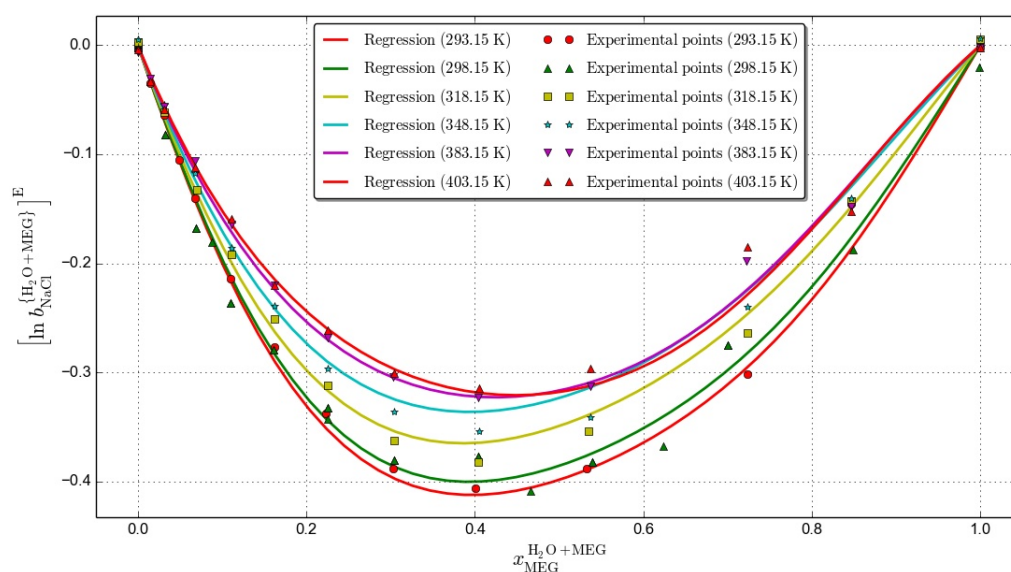


Fonte: o autor.

4.2.2.5 Logaritmo de excesso da molalidade do NaCl em mistura de solventes (Regressão)

A Figura 27 representa as isotermas do logaritmo de excesso da molalidade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl. Nesta figura observa-se uma boa aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se que os valores do logaritmo de excesso da molalidade do NaCl em mistura de solventes são negativos e atingem um valor mínimo para o valor aproximado de 0,40 para a fração molar de MEG livre de solvente. Neste caso para valores de fração molar de MEG livre de solvente menores que 0,40 temos que o logaritmo de excesso da molalidade do NaCl em mistura de solventes diminui com o aumento da concentração de MEG, e após o valor de 0,40 inverte-se o comportamento.

Figura 27 – Isotermas do logaritmo de excesso da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão).

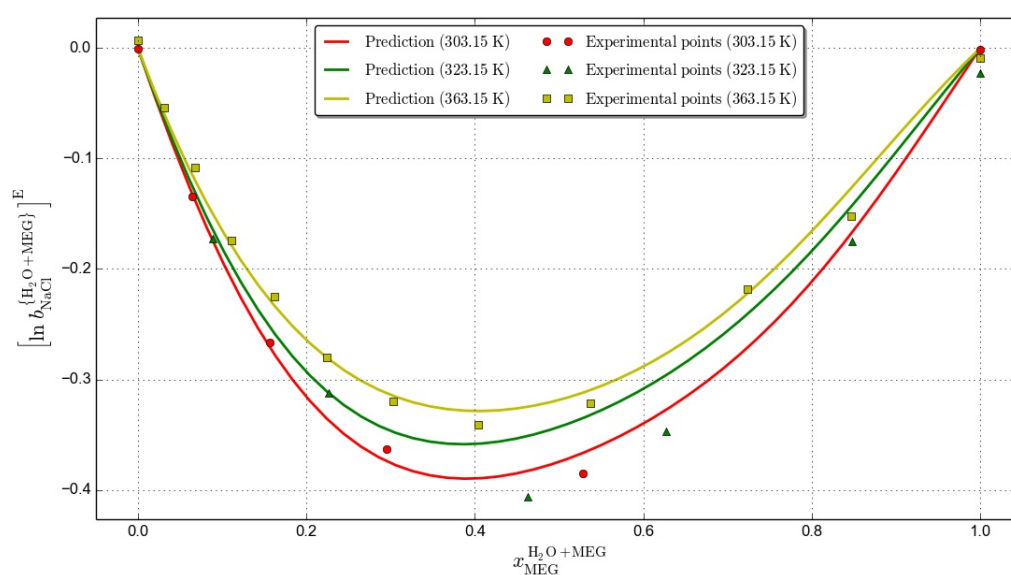


Fonte: o autor.

4.2.2.6 Logaritmo de excesso da molalidade do NaCl em mistura de solventes (Predição)

A Figura 28 representa as isotermas do logaritmo de excesso da molalidade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl (Predição). Nesta figura observa-se uma razoável aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se comportamento análogo ao da Figura 27.

Figura 28 – Isotermas do logaritmo de excesso da molalidade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição).

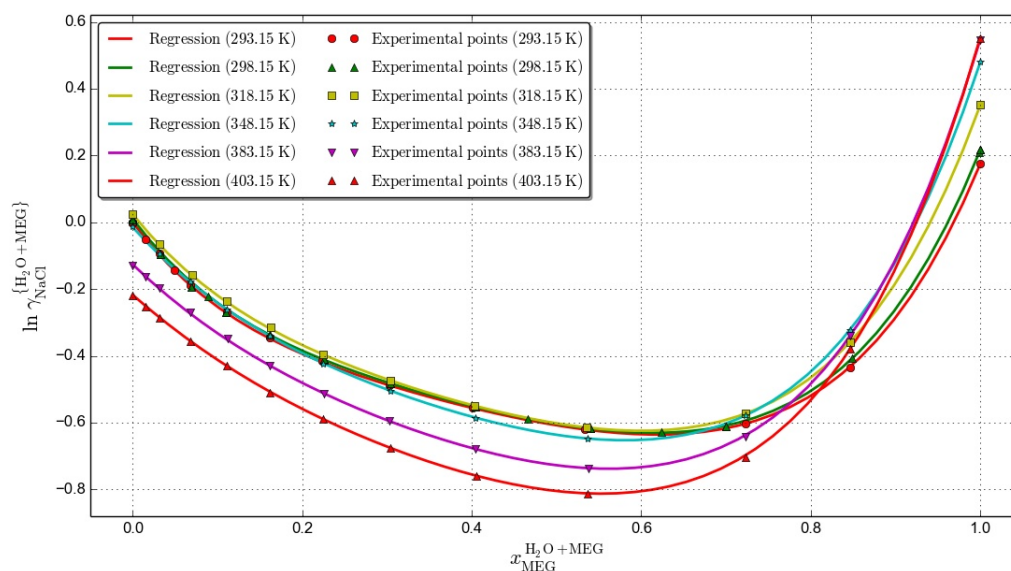


Fonte: o autor.

4.2.2.7 Logaritmo do coeficiente de atividade do NaCl em mistura de solventes (Regressão)

A Figura 29 representa as isotermas do logaritmo do coeficiente de atividade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl. Nesta figura observa-se a excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Também observa-se o fenômeno de inversão em torno do valor de 0,8 para a fração molar de MEG livre de solvente. Neste caso para valores de fração molar de MEG livre de solvente menores que 0,82 temos que o logaritmo do coeficiente de atividade do NaCl em mistura de solventes aumenta com a temperatura, e após o valor de 0,8 inverte-se o comportamento.

Figura 29 – Isotermas do logaritmo do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão).

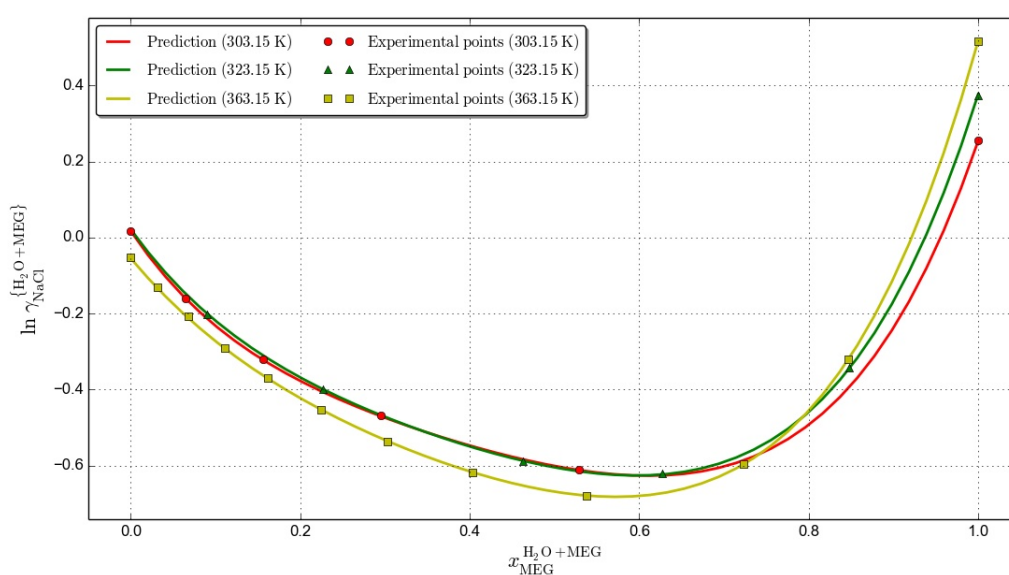


Fonte: o autor.

4.2.2.8 Logaritmo do coeficiente de atividade do NaCl em mistura de solventes (Predição)

A Figura 30 representa as isotermas do logaritmo de excesso da molalidade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl (Predição). Nesta figura observa-se uma excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se comportamento análogo ao da Figura 29.

Figura 30 – Isotermas do logaritmo do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição).

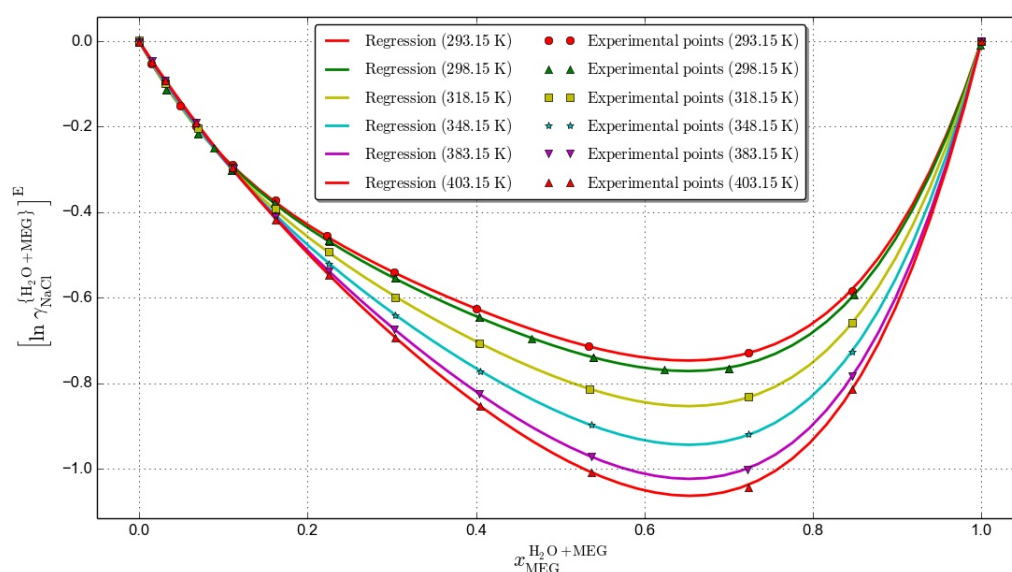


Fonte: o autor.

4.2.2.9 Logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes (Regressão)

A Figura 31 representa as isotermas do logaritmo do coeficiente de atividade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl. Nesta figura observa-se uma excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se que os valores do logaritmo de excesso da molalidade do NaCl em mistura de solventes são negativos e atingem um valor mínimo para o valor aproximado de 0,64 para a fração molar de MEG livre de solvente. Neste caso para valores de fração molar de MEG livre de solvente menores que 0,64 temos que o logaritmo de excesso da molalidade do NaCl em mistura de solventes diminui com o aumento da concentração de MEG, e após o valor de 0,64 inverte-se o comportamento.

Figura 31 – Isotermas do logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Regressão).

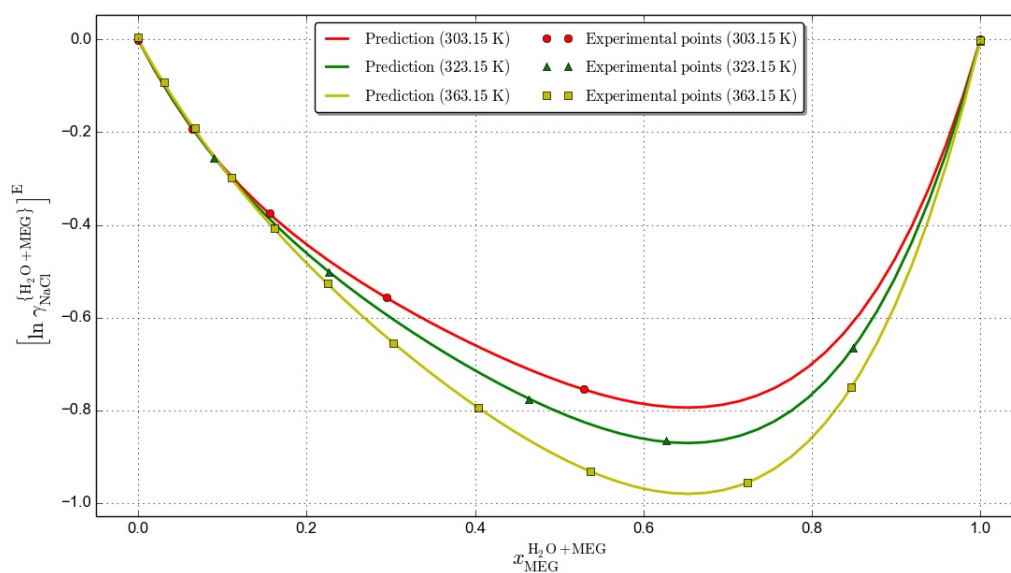


Fonte: o autor.

4.2.2.10 Logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes (Predição)

A Figura 32 representa as isotermas do logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes em função da concentração de MEG livre de NaCl (Predição). Nesta figura observa-se uma razoável aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se comportamento análogo ao da Figura 31.

Figura 32 – Isotermas do logaritmo de excesso do coeficiente de atividade do NaCl em mistura de solventes versus concentração de MEG livre de NaCl (Predição).



Fonte: o autor.

4.3 Sistemas com KCl

4.3.1 Sistemas Binários

4.3.1.1 Sistema H₂O+KCl

Dados de solubilidade de KCl em água foram ajustados por regressão não-linear usando-se a função:

$$b_{\text{KCl}}^{\{\text{H}_2\text{O}\}}(T) = \exp(\theta_0 + \theta_1 T + \theta_2 \ln T) \quad (4.4)$$

A [Tabela 7](#) apresenta os valores obtidos para os parâmetros θ_i .

Tabela 7 – Parâmetros da [Equação 4.4](#) obtidos por regressão para o Sistema H₂O+KCl.

θ_i	Valor
θ_0	$-28,334183 \pm 1,467987$
θ_1	$-0,011414 \pm 0,000942$
θ_2	$5,845133 \pm 0,306710$

Fonte: o autor

O resultado estatístico apresentado pelo programa para esta regressão é:

REGRESSION FOR H2O:

Non-Linear Least Squares Summary Statistics

Function: $y = T, *theta: \exp(theta[0]+theta[1]*T+theta[2]*\log(T))$

Source	DF	Sum of Squares	Mean Square	F-value	Pr(>F)
Regression	3	759.879336	253.293112	383817.645485	0.000000
Residual	22	0.014518	0.000660		
Uncorrected Total	25	759.893855			
(Corrected Total)	24	21.660921			

alpha = 0.050000 t-critical = 2.073873

s = 0.025689 R-sq = 0.999330 R-sq(adj) = 0.999269

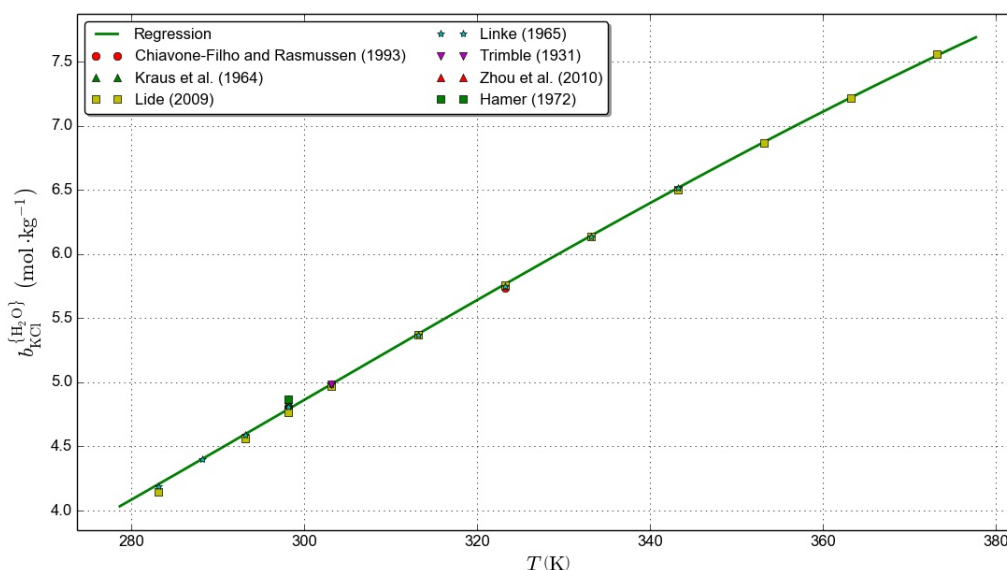
Parameter	Estimate	Asymptotic Std. Error	t-value	Pr(> t)	Asymptotic 95% Confidence Interval
					Lower Upper
P[0]	-28.334183	1.467987	-19.301388	0.000000	-31.378602 -25.289765
P[1]	-0.011414	0.000942	-12.112830	0.000000	-0.013368 -0.009460
P[2]	5.845133	0.306710	19.057522	0.000000	5.209056 6.481211

Parameter	Correlation Matrix		
	P[0]	P[1]	P[2]
P[0]	1.000000	0.998987	-0.999970
P[1]	0.998987	1.000000	-0.999306
P[2]	-0.999970	-0.999306	1.000000

O ajuste foi bastante satisfatório apresentando um coeficiente de correlação de 0,999330 (R^2). Isto pode ser confirmado por meio da [Figura 33](#), onde observa-se a fortíssima tendência dos dados a aderirem a curva.

Observa-se na [Figura 33](#) que a solubilidade do KCl diminui com a temperatura de forma ligeiramente abaixo da linear em toda a faixa de temperatura correlacionada.

Figura 33 – Molalidade do KCl em H₂O versus Temperatura.



Fonte: o autor.

4.3.1.2 Sistema MEG+KCl

Dados de solubilidade de KCl em MEG foram ajustados por regressão não-linear usando-se a função:

$$b_{\text{KCl}}^{\{\text{MEG}\}}(T) = \exp(\theta_0 + \theta_1 T) \quad (4.5)$$

A [Tabela 8](#) apresenta os valores obtidos para os parâmetros θ_i .

Tabela 8 – Parâmetros da [Equação 4.5](#) obtidos por regressão para o Sistema MEG+KCl.

θ_i	Valor
θ_0	$-1,872213 \pm 0,165790$
θ_1	$0,005120 \pm 0,000528$

Fonte: o autor

O resultado estatístico apresentado pelo programa para esta regressão é:

REGRESSION FOR MEG:

Non-Linear Least Squares Summary Statistics

Function: $y = T$, *theta: $\exp(\theta[0] + \theta[1] * T)$

Source	DF	Sum of Squares	Mean Square	F-value	Pr(>F)
Regression	2	4.013748	2.006874	4516.198037	0.000221
Residual	5	0.002222	0.000444		
Uncorrected Total	7	4.015970			
(Corrected Total)	6	0.039801			

 $\alpha = 0.050000$ t-critical = 2.570582 $s = 0.021080$ R-sq = 0.944175 R-sq(adj) = 0.933011

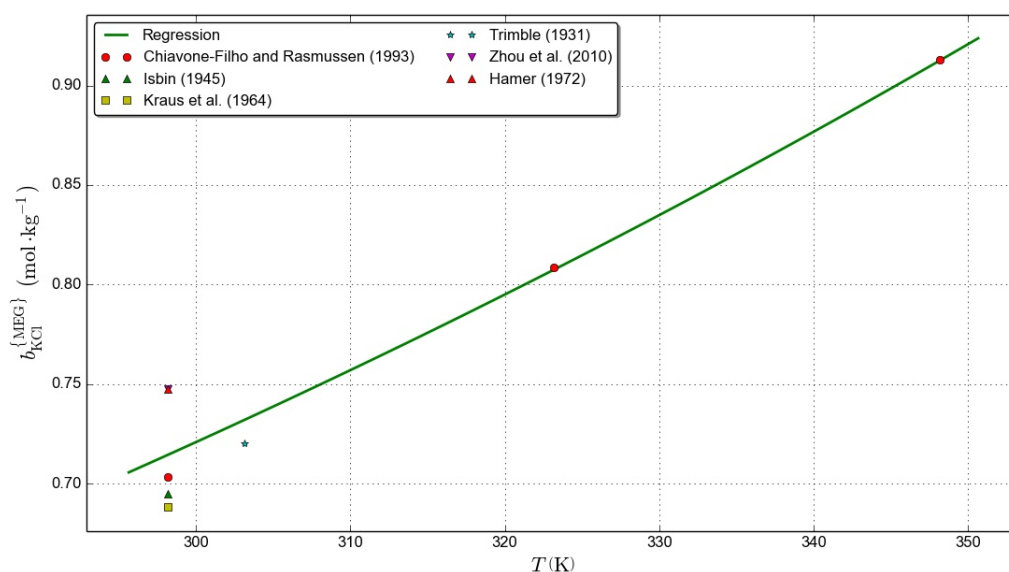
Parameter	Estimate	Asymptotic Std. Error	t-value	Pr(> t)	Asymptotic 95% Confidence Interval
					Lower Upper
P[0]	-1.872213	0.165790	-11.292652	0.000095	-2.298391 -1.446036
P[1]	0.005120	0.000528	9.693384	0.000198	0.003762 0.006478

Parameter	Correlation Matrix
	P[0] P[1]
P[0]	1.000000 -0.997984
P[1]	-0.997984 1.000000

O ajuste foi bastante satisfatório apresentando um coeficiente de correlação de 0,944175 (R^2). Isto pode ser confirmado por meio da [Figura 34](#), onde observa-se a tendência dos dados a aderirem a curva.

Observa-se na [Figura 34](#) que a solubilidade do KCl em MEG aumenta com a temperatura de forma visualmente linear em toda a faixa de temperatura correlacionada.

Figura 34 – Molalidade do KCl em MEG versus Temperatura.



Fonte: o autor.

4.3.2 Sistemas Ternários

4.3.2.1 Potencial químico padrão de excesso do KCl em mistura de solventes (Regressão)

Dados de solubilidade de KCl em H₂O+MEG foram empregados no cálculo de valores de potencial químico padrão de excesso. Em seguida, estes foram ajustados por regressão não-linear usando-se a função:

$$\frac{[\mu_{\text{KCl}}^{\text{H}_2\text{O}+\text{MEG}}]^E}{nRT} = x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}} \left(1 - x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}}\right) \left[\theta_0 + \theta_1 T + (\theta_2 + \theta_3 T) x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}} + (\theta_4 + \theta_5 T) x_{\text{MEG}}^{\text{H}_2\text{O}+\text{MEG}}^2\right] \quad (4.6)$$

A [Tabela 9](#) apresenta os valores obtidos para os parâmetros θ_i .

Tabela 9 – Parâmetros da [Equação 4.6](#) obtidos por regressão para o Sistema H₂O+MEG+KCl.

θ_i	Valor
θ_0	1.433040 ± 3.651707
θ_1	-0.010232 ± 0.012044
θ_2	-20.275677 ± 16.940293
θ_3	0.050808 ± 0.055854
θ_4	12.410073 ± 17.367966
θ_5	-0.018566 ± 0.057203

Fonte: o autor

O resultado estatístico apresentado pelo programa para esta regressão é:

Non-Linear Least Squares Summary Statistics

Function: y = (<function f3 at 0x4ac0500>, 'function')

Source	DF	Sum of Squares	Mean Square	F-value	Pr(>F)
Regression	6	14.219218	2.369870	5092.169462	0.000000
Residual	54	0.025131	0.000465		
Uncorrected Total	60	14.244349			
(Corrected Total)	59	8.470604			

alpha = 0.050000 t-critical = 2.004879

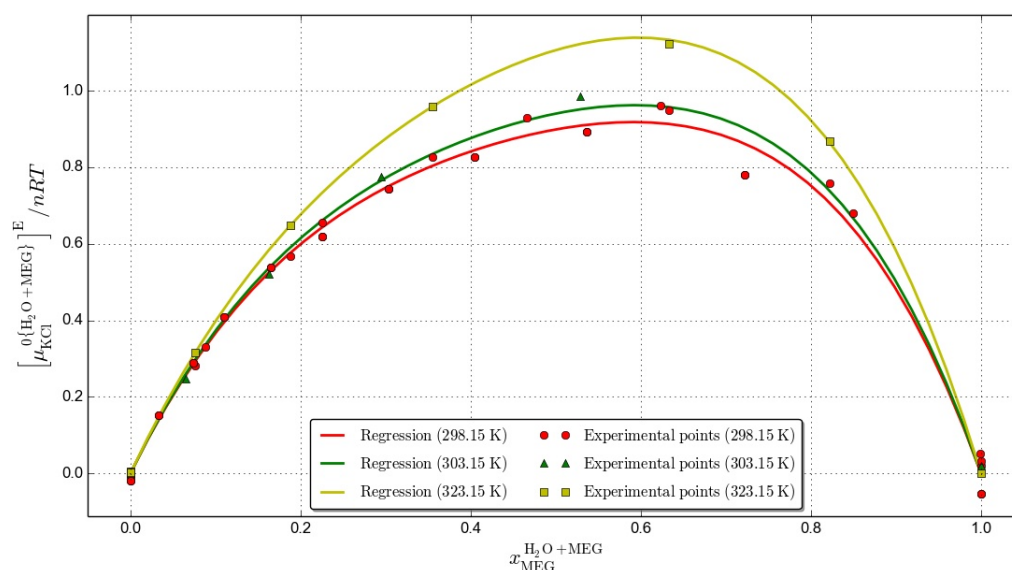
s = 0.021573 R-sq = 0.997033 R-sq(adj) = 0.996758

Parameter	Estimate	Asymptotic Std. Error	t-value	Pr(> t)	Asymptotic 95% Confidence Interval
					Lower Upper
P[0]	1.433040	3.651707	0.392430	0.696285	-5.888191 8.754272
P[1]	0.010232	0.012044	0.849516	0.399345	-0.013915 0.034378
P[2]	-20.275677	16.940293	-1.196891	0.236577	-54.238920 13.687565
P[3]	0.050808	0.055854	0.909646	0.367052	-0.061173 0.162788
P[4]	12.410073	17.367966	0.714538	0.477972	-22.410602 47.230748
P[5]	-0.018566	0.057203	-0.324566	0.746764	-0.133251 0.096119

Parameter	Correlation Matrix					
	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]
P[0]	1.000000	-0.999554	-0.942158	0.941940	0.866598	-0.867038
P[1]	-0.999554	1.000000	0.941633	-0.942268	-0.865826	0.867072
P[2]	-0.942158	0.941633	1.000000	-0.999538	-0.978231	0.978362
P[3]	0.941940	-0.942268	-0.999538	1.000000	0.977314	-0.978381
P[4]	0.866598	-0.865826	-0.978231	0.977314	1.000000	-0.999501
P[5]	-0.867038	0.867072	0.978362	-0.978381	-0.999501	1.000000

O ajuste foi bastante satisfatório apresentando um coeficiente de correlação de 0,997033 (R^2), muito embora as estimativas para os erros assintóticos (ASE) para os parâmetros tenham valores que "mascaram" ou "encobrem" os valores dos parâmetros. Isto pode ser confirmado por meio da [Figura 35](#), onde observa-se uma boa aderência dos valores de potencial químico padrão de excesso calculados a partir dos dados de solubilidade e metodologia empregados.

Figura 35 – Isotermas do potencial químico padrão de excesso do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão).



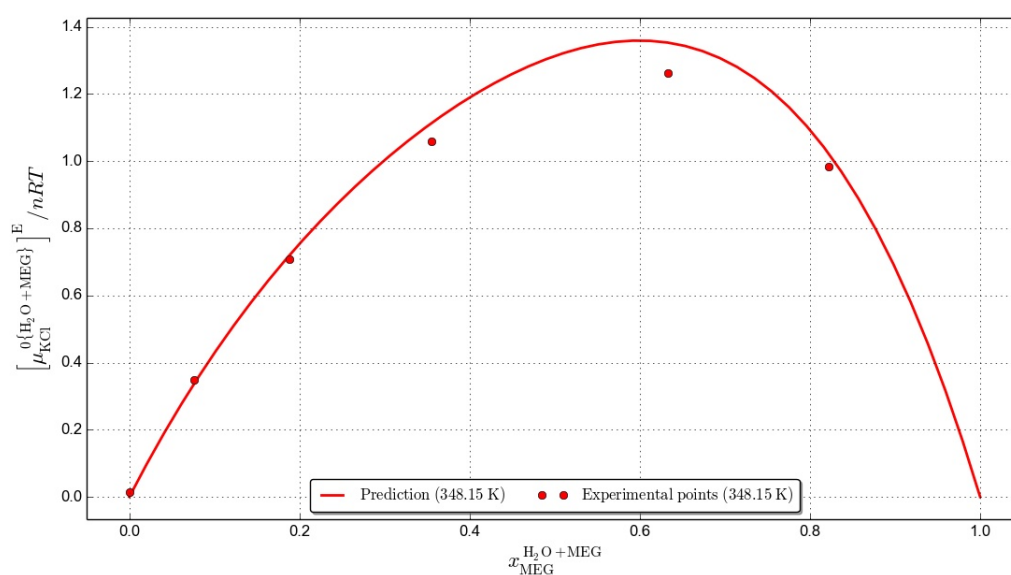
Fonte: o autor.

Observa-se na [Figura 35](#) que o potencial químico padrão de excesso do KCl na mistura de solventes, nas diversas isotermas, semelhante ao comportamento para o NaCl, cresce desde zero até um valor máximo correspondente a fração molar de MEG livre de solvente próximo de 0,6 e depois decresce até o valor zero. Para o KCl não foi observado o comportamento inverso descrito para no caso do NaCl.

4.3.2.2 Potencial químico padrão de excesso do KCl em mistura de solventes (Predição)

A Figura 36 representa uma isoterma do potencial químico padrão de excesso do KCl em mistura de solventes em função da concentração de MEG livre de KCl. Nesta figura observa-se novamente uma boa aderência dos valores de potencial químico padrão de excesso calculados a partir dos dados de solubilidade e metodologia empregados. Novamente não é observado o comportamento inverso descrito para no caso do NaCl.

Figura 36 – Isotermas do potencial químico padrão de excesso do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição).

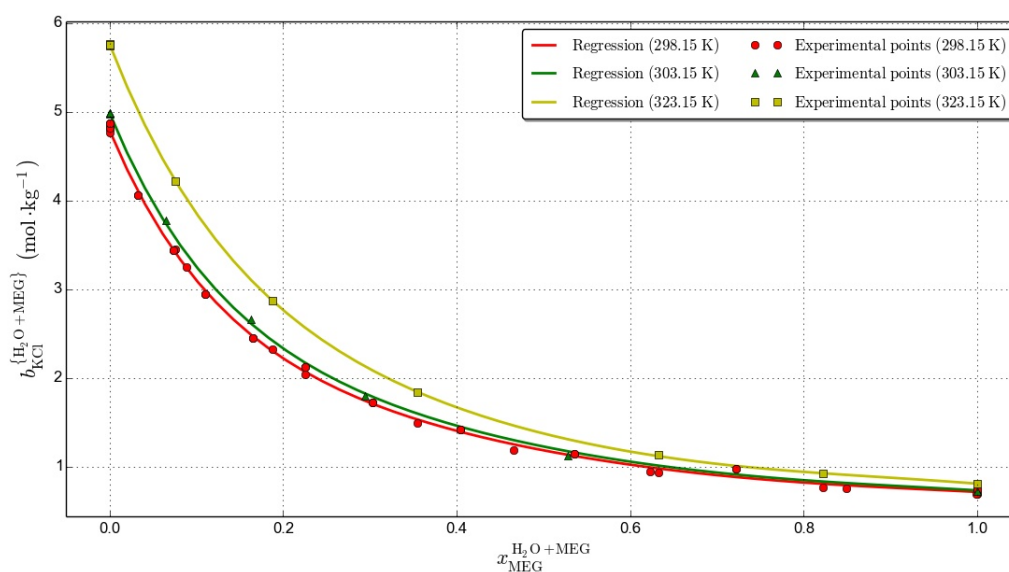


Fonte: o autor.

4.3.2.3 Molalidade do KCl em mistura de solventes (Regressão)

A Figura 37 representa as isotermas da molalidade do KCl em mistura de solventes em função da concentração de MEG livre de KCl. Nesta figura observa-se a excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Para o KCl não foi observado o comportamento inverso descrito para no caso do NaCl. Observa-se a diminuição da solubilidade com o aumento da concentração do MEG. A figura

Figura 37 – Isotermas da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão).

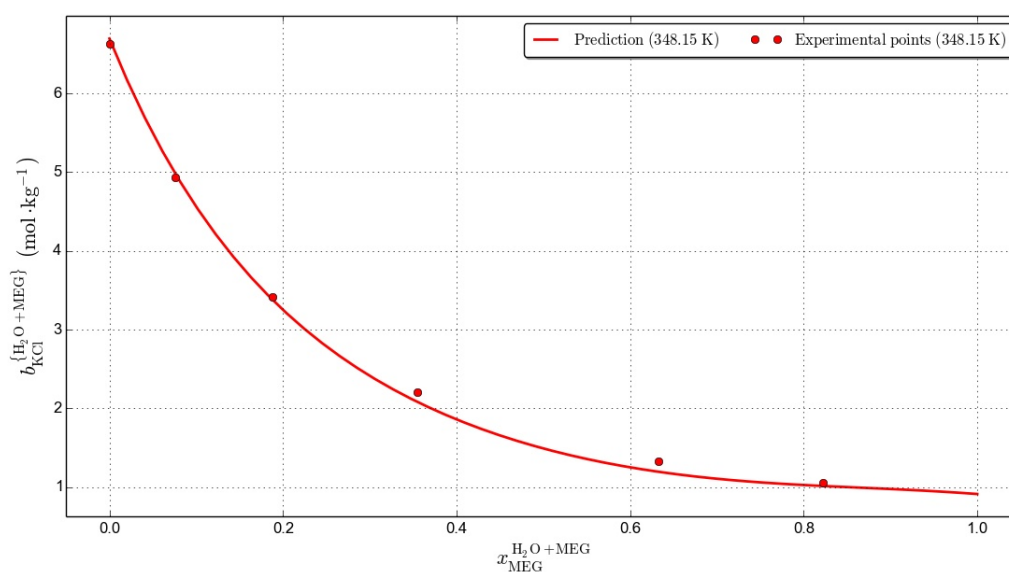


Fonte: o autor.

4.3.2.4 Molalidade do KCl em mistura de solventes (Predição)

A Figura 38 representa a isoterma da molalidade do KCl em mistura de solventes em função da concentração de MEG livre de KCl (Predição). Nesta figura observa-se novamente a excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Também, para o KCl, não foi observado o comportamento inverso descrito para no caso do NaCl. Observa-se, também, a diminuição da solubilidade com o aumento da concentração do MEG.

Figura 38 – Isotherma da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição).

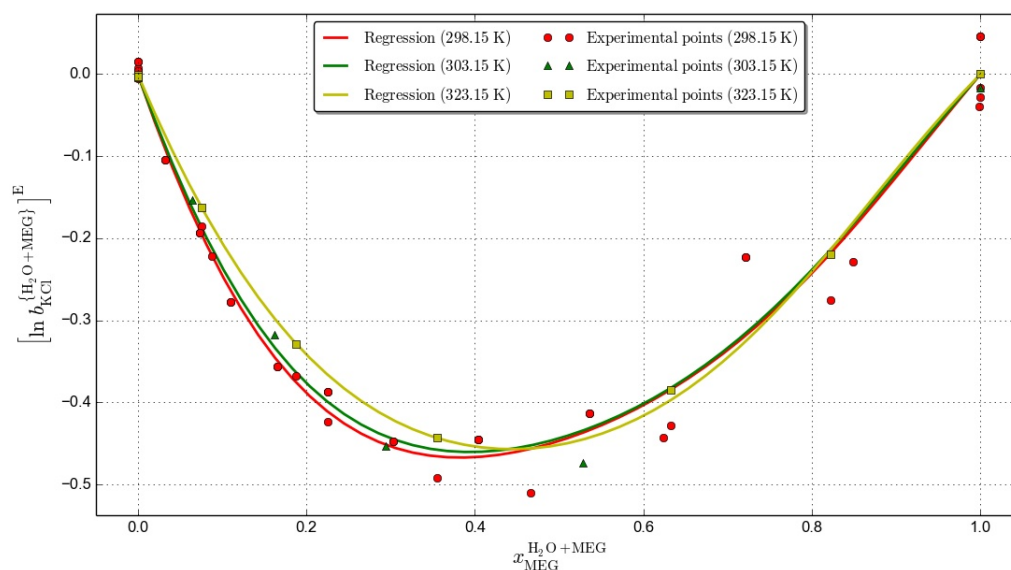


Fonte: o autor.

4.3.2.5 Logaritmo de excesso da molalidade do KCl em mistura de solventes (Regressão)

A Figura 39 representa as isotermas do logaritmo de excesso da molalidade do KCl em mistura de solventes em função da concentração de MEG livre de KCl. Nesta figura observa-se uma boa aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se que os valores do logaritmo de excesso da molalidade do KCl em mistura de solventes são negativos e atingem um valor mínimo para o valor aproximado de 0,40 para a fração molar de MEG livre de solvente. Neste caso para valores de fração molar de MEG livre de solvente menores que 0,40 temos que o logaritmo de excesso da molalidade do KCl em mistura de solventes diminui com o aumento da concentração de MEG, e após o valor de 0,40 inverte-se o comportamento.

Figura 39 – Isotermas do logaritmo de excesso da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão).

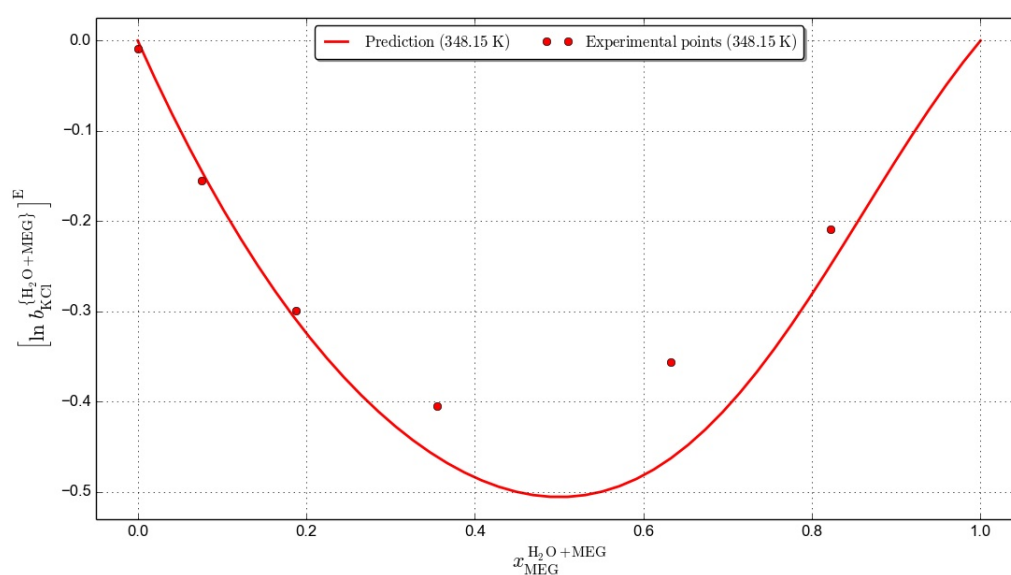


Fonte: o autor.

4.3.2.6 Logaritmo de excesso da molalidade do KCl em mistura de solventes (Predição)

A Figura 40 representa as isotermas do logaritmo de excesso da molalidade do KCl em mistura de solventes em função da concentração de MEG livre de KCl (Predição). Nesta figura observa-se uma razoável aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se comportamento análogo ao da Figura 39.

Figura 40 – Isotermas do logaritmo de excesso da molalidade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição).

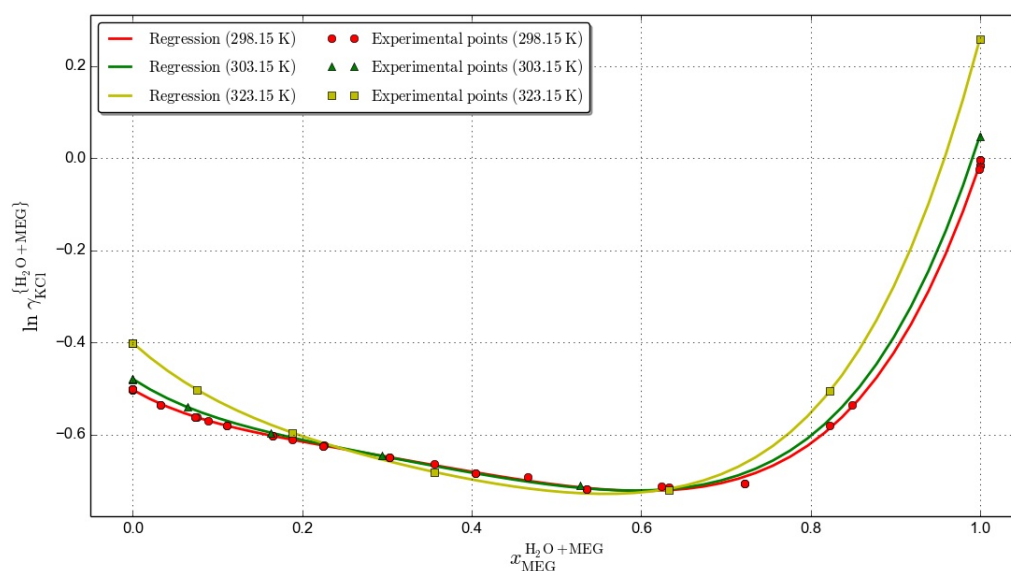


Fonte: o autor.

4.3.2.7 Logaritmo do coeficiente de atividade do KCl em mistura de solventes (Regressão)

A Figura 41 representa as isotermas do logaritmo do coeficiente de atividade do KCl em mistura de solventes em função da concentração de MEG livre de KCl. Nesta figura observa-se a excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Também observa-se o fenômeno de inversão em torno do valor de 0,64 para a fração molar de MEG livre de solvente. Neste caso para valores de fração molar de MEG livre de solvente menores que 0,64 temos que o logaritmo do coeficiente de atividade do KCl em mistura de solventes aumenta com a temperatura, e após o valor de 0,64 inverte-se o comportamento.

Figura 41 – Isotermas do logaritmo do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão).

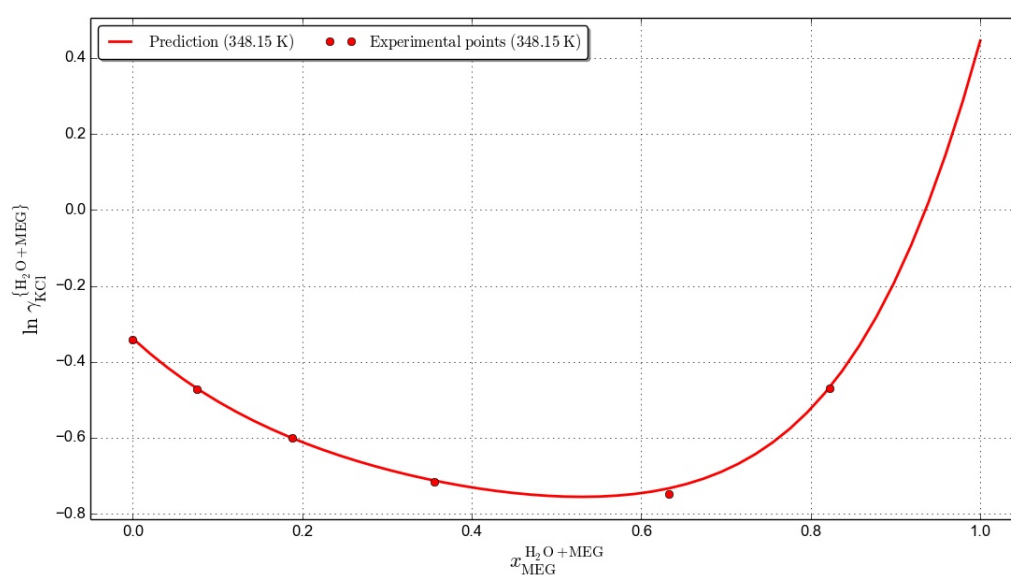


Fonte: o autor.

4.3.2.8 Logaritmo do coeficiente de atividade do KCl em mistura de solventes (Predição)

A Figura 42 representa uma isoterma do logaritmo de excesso da molalidade do KCl em mistura de solventes em função da concentração de MEG livre de KCl (Predição). Nesta figura observa-se uma excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se comportamento análogo ao da Figura 41.

Figura 42 – Isotermas do logaritmo do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição).

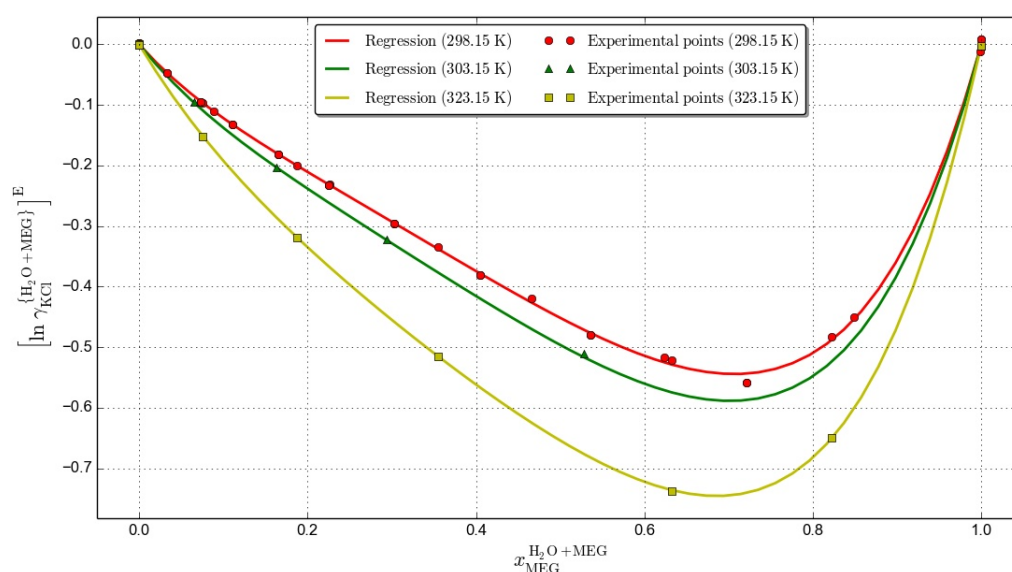


Fonte: o autor.

4.3.2.9 Logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes (Regressão)

A Figura 43 representa as isotermas do logaritmo do coeficiente de atividade do KCl em mistura de solventes em função da concentração de MEG livre de KCl. Nesta figura observa-se uma excelente aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se que os valores do logaritmo de excesso da molalidade do KCl em mistura de solventes são negativos e atingem um valor mínimo para o valor aproximado de 0,64 para a fração molar de MEG livre de solvente. Neste caso para valores de fração molar de MEG livre de solvente menores que 0,64 temos que o logaritmo de excesso da molalidade do KCl em mistura de solventes diminui com o aumento da concentração de MEG, e após o valor de 0,64 inverte-se o comportamento.

Figura 43 – Isotermas do logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Regressão).

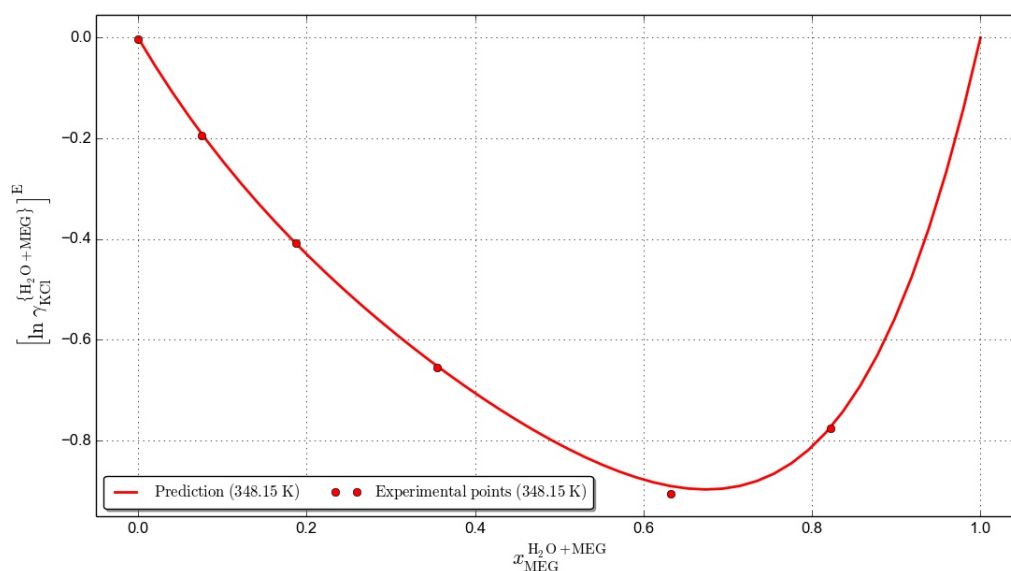


Fonte: o autor.

4.3.2.10 Logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes (Predição)

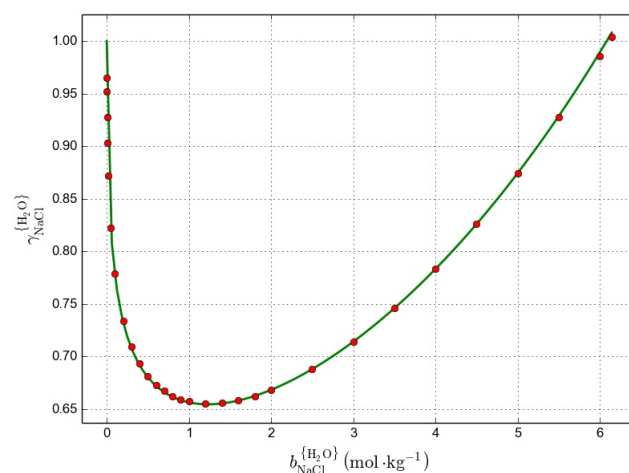
A Figura 44 representa as isotermas do logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes em função da concentração de MEG livre de KCl (Predição). Nesta figura observa-se uma razoável aderência dos valores obtidos por meio da função obtida para o potencial químico padrão de excesso e metodologia empregados. Observa-se comportamento análogo ao da Figura 43.

Figura 44 – Isotermas do logaritmo de excesso do coeficiente de atividade do KCl em mistura de solventes versus concentração de MEG livre de KCl (Predição).



Fonte: o autor.

Figura 45 – Validação do Modelo de Pitzer implementado (γ).



Fonte: o autor.

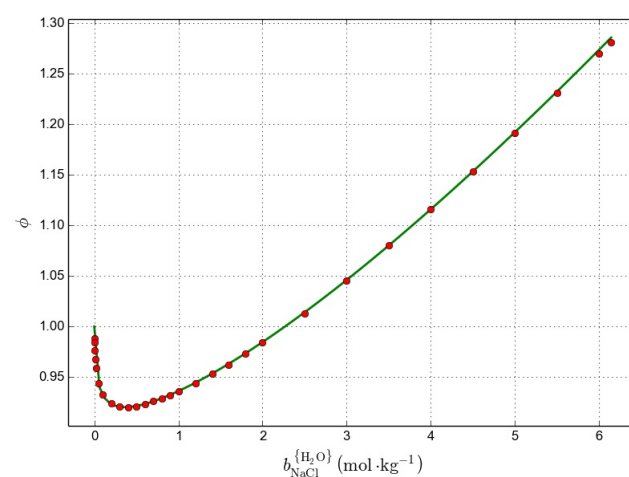
4.4 Validação do Modelo de Pitzer implementado

A rotina de cálculo para o Modelo de Pitzer implementado foi devidamente validada com base nos dados de [Hamer e Wu \(1972\)](#), tanto para o coeficiente de atividade quanto para o coeficiente osmótico.

Os resultados para o coeficiente de atividade estão representados na [Figura 45](#) e os resultados para o coeficiente osmótico estão representados na [Figura 46](#).

Observa-se em ambas as figuras a requerida aderência com os dados experimentais.

Figura 46 – Validação do Modelo de Pitzer implementado (ϕ).



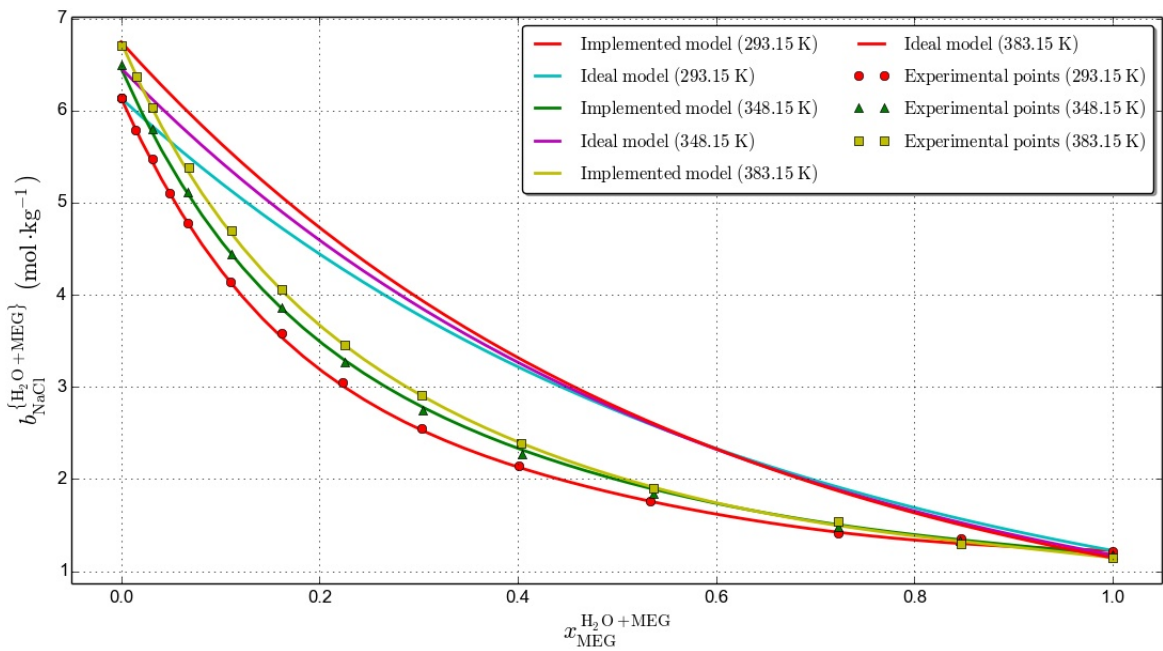
Fonte: o autor.

4.5 Comparação do Modelo implementado com outros modelos

4.5.1 Modelo Ideal

É importante verificar a diferença entre o modelo implementado e a abordagem ideal (Equação 2.35). Isto foi realizado de maneira qualitativa com base nos desvios em relação à solubilidade. O gráfico gerado é apresentado na Figura 47.

Figura 47 – Comparação do Modelo implementado com o Modelo ideal (Equação 2.35).



Fonte: o autor.

Observa-se na Figura 47 a excelente representatividade dos dados experimentais pelo modelo implementado e o afastamento deste em relação ao modelo ideal. Isto também é confirmado pela Figura 47 que apresenta os desvios dos dados experimentais em relação ao modelos implementado e ideal.

Tabela 10 – Comparação do Modelo implementado com o Modelo ideal (Equação 2.35).

Modelo	Desvio medio absoluto	Desvio medio relativo percentual
Modelo Ideal	0.581431	20.422%
Modelo Implementado	0.019541	0.744%

Fonte: o autor

4.5.2 Modelo UNIQUAC+DH

Uma comparação quantitativa foi feita entre o modelo implementado e o modelo apresentado por (CHIAVONE-FILHO; RASMUSSEN, 2000) onde os cálculos resultarão em precisões equivalentes, dentro das incertezas experimentais. No entanto, deve-se ressaltar que o modelo implementado possui por ser parametrizado possui a vantagem de necessitar de uma menor quantidade de informações para poder-se gerar resultados.

CAPÍTULO 5

CONCLUSÕES

5 CONCLUSÕES

Uma metodologia para descrever a solubilidade de sais em mistura de solventes foi desenvolvida partindo-se da equação de Pitzer, a qual foi extendida para misturas de solventes pela metodologia adotada por Lorimer. Esta metodologia mostrou-se adequada na descrição dos sistemas apresentados.

Os resultados obtidos demonstraram precisão dos cálculos dentro das incertezas experimentais para os sais KCl e NaCl em misturas com água e MEG. Modelos e métodos de cálculo de propriedades de sais em misturas de solventes não são tão frequentes. Portanto a metodologia aqui desenvolvida é uma contribuição importante.

A extensão do modelo de Pitzer para misturas de solventes permite que lacunas existentes em bancos de dados experimentais disponíveis sejam preenchidas. Além da parte termodinâmica do equilíbrio, foi desenvolvida uma ferramenta computacional, adequada à realização dos cálculos da solubilidade de sais em misturas aquosas contendo MEG, ou outro solvente, nos modos de correlação e predição, com um código muito flexível. Embora o desenvolvimento do código tenha sido realizado em Linux usando a linguagem Python no ambiente Eclipse, é portátil para outros sistemas operacionais como Windows e OS X.

Sugestões para trabalhos futuros são:

- a) utilização desta metodologia de cálculo em sistemas com os mesmos solventes e com outras espécies de sal;
- b) utilização desta metodologia de cálculo em sistemas com os mesmos solventes e com mais de um sal;
- c) utilização desta metodologia de cálculo em sistemas com outros solventes e sais;
- d) estudar outros sais presentes na água de produção do campo de Mexilhão;
- e) aplicar o modelo diretamente na modelagem e simulação de processos, como na unidade de regeneração do MEG.

Referências

ABRAMS, D. S.; PRAUSNITZ, J. M. Statistical thermodynamics of liquid mixtures: A new expression for the excess gibbs energy of partly or completely miscible systems. *AIChE Journal*, American Institute of Chemical Engineers, v. 21, n. 1, p. 116–128, 1975. ISSN 1547-5905. Disponível em: <<http://dx.doi.org/10.1002/aic.690210115>>. Citado na página 55.

ANDERKO, A.; WANG, P.; RAFAL, M. Electrolyte solutions: from thermodynamic and transport property models to the simulation of industrial processes. *Fluid Phase Equilibria*, v. 194–197, n. 0, p. 123–142, mar. 2002. ISSN 0378-3812. <ce:title>Proceedings of the Ninth International Conference on Properties and Phase Equilibria for Product and Process Design</ce:title>. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0378381201006458>>. Citado na página 54.

ANDRADE, A. R. Monografia apresentada ao Programa de Pós-Graduação Lato Sensu em Engenharia Mecânica do Departamento de Engenharia Mecânica da PUC-Rio, *Hidratos no Âmbito da Perfuração: Histórico, Mecanismos de Inibição e Técnicas para Análises de Hidrato*. 2009. Disponível em: <<http://www.ppgem.ct.utfpr.edu.br/lacit/publicacoes/monografias/monografia%20Alex%20R.%20de%20Andrade%20-%20Vers%20Final.pdf>>. Citado 2 vezes nas páginas 35 e 43.

ARCHER, D. G. Thermodynamic properties of the kcl + h₂O system. *Journal of Physical and Chemical Reference Data*, NIST, v. 28, n. 1, p. 1–17, 1999. Disponível em: <<http://link.aip.org/link/?JPR/28/1/1>>. Citado na página 63.

BALDWIN, W. H.; RARIDON, R. J.; KRAUS, K. A. Properties of organic-water mixtures. x. activity coefficients of sodium chloride at saturation in water mixtures of polyglycols and polyglycol ethers at 50.deg. *The Journal of Physical Chemistry*, v. 73, n. 10, p. 3417–3420, 1969. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/j100844a045>>. Citado na página 42.

BAPTISTA, J. Trabalho de Conclusão de Curso, *Análise da Formação de Hidratos em Atividades de Perfuração de Campos Petrolíferos*. 2007. Disponível em: <http://www.ppgem.ct.utfpr.edu.br/lacit/publicacoes/projetos_finais/PF%20-%20An%20E1lise%20da%20Forma%20E7%20de%20Hidratos%20em%20atividades%20de%20Perfura%20E7%20de%20Campos%20Petrol%20feros%202007.pdf>. Citado na página 34.

BRAUN, N. O.; PERSSON, U. A.; KARLSSON, H. T. Densities and viscosities of mono(ethylene glycol) + 2-amino-2-methyl-1-propanol + water. *Journal of Chemical & Engineering Data*, v. 46, n. 4, p. 805–808, jul. 2001. ISSN 0021-9568. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/je010004z>>. Citado na página 42.

CARDOSO, M. J. E. M. Ph. D. Thesis, Institut for Kemiteknik, Danmarks Tekniske Hojskole, Bygning 229, DK-2800 Lyngby, Denmark, *Calculation of Phase and Chemical Equilibria in Eletrolyte Systems*. 1988. Citado na página 54.

CERE. *CERE*. 2009. Disponível em: <<http://www.cere.dtu.dk>>. Citado na página 41.

CHEN, C.-C. et al. Local composition model for excess gibbs energy of electrolyte systems. part i: Single solvent, single completely dissociated electrolyte systems. *AIChE Journal*, American Institute of Chemical Engineers, v. 28, n. 4, p. 588–596, jul. 1982. ISSN 1547-5905. Disponível em: <<http://doi.wiley.com/10.1002/aic.690280410>>. Citado na página 63.

CHEN, C.-C.; SONG, Y. Generalized electrolyte-nrtl model for mixed-solvent electrolyte systems. *AIChE Journal*, American Institute of Chemical Engineers, v. 50, n. 8, p. 1928–1941, ago. 2004. ISSN 1547-5905. Disponível em: <<http://doi.wiley.com/10.1002/aic.10151>>. Citado na página 54.

CHIAVONE-FILHO, O. Ph. D. Thesis. Department of Chemical Engineering. Technical University of Denmark; DK-2800 Lyngby, Denmark, *Phase Behavior of Aqueous Glycol Ether Mixtures: (1) Vapor-Liquid Equilibria (2) Salt Solubility*. 1993. Citado na página 54.

CHIAVONE-FILHO, O.; PROUST, P.; RASMUSSEN, P. Vapor-liquid equilibria for glycol ether + water systems. *Journal of Chemical & Engineering Data*, v. 38, n. 1, p. 128–131, jan. 1993. ISSN 0021-9568. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/je00009a031>>. Citado na página 52.

CHIAVONE-FILHO, O.; RASMUSSEN, P. Solubilities of salts in mixed solvents. *Journal of Chemical & Engineering Data*, v. 38, n. 3, p. 367–369, jul. 1993. ISSN 0021-9568. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/je00011a009>>. Citado na página 42.

CHIAVONE-FILHO, O.; RASMUSSEN, P. Modeling of salt solubilities in mixed solvents. *Brazilian Journal of Chemical Engineering*, scielo, v. 17, p. 117–131, 06 2000. ISSN 0104-6632. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-66322000000200001&nrm=iso>. Citado 2 vezes nas páginas 55 e 104.

DAHL, S.; MACEDO, E. A. The mhv2 model: A unifac-based equation of state model for vapor-liquid and liquid-liquid equilibria of mixtures with strong electrolytes. *Industrial & Engineering Chemistry Research*, v. 31, n. 4, p. 1195–1201, abr. 1992. ISSN 0888-5885. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ie00004a033>>. Citado na página 55.

DEBYE, P.; HÜCKEL, E. The theory of electrolytes. i. lowering of freezing point and related phenomena. *Physikalische Zeitschrift*, v. 24, p. 185–206, 1923. Citado 2 vezes nas páginas 48 e 49.

FIGUEIREDO, C. S. et al. Fluid Phase Equilibria (em Publicação), *Salt solubility data for sodium chloride and mono ethylene glycol aqueous systems from 293.15 to 403.15 K*. 2014. Citado 2 vezes nas páginas 41 e 42.

FOSBOL, P. L.; THOMSEN, K.; STENBY, E. H. Modeling of the mixed solvent electrolyte system $\text{co}_2\text{-na}_2\text{co}_3\text{-nahco}_3\text{-monoethylene glycol-water}$. *Industrial & Engineering Chemistry Research*, v. 48, n. 9, p. 4565–4578, maio 2009. ISSN 0888-5885. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ie801168e>>. Citado na página 54.

GREEN, D. W.; PERRY, R. H. (Ed.). *Perry's Chemical Engineers' Handbook*. 8. ed. Nova Iorque: McGraw-Hill, 2007. 2400 p. ISBN 978-0-07-142294-9. Disponível em: <<http://www.amazon.com/Perrys-Chemical-Engineers-Handbook-Edition/dp/0071422943%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0071422943>>.

Citado na página 61.

GRENTHE, I.; PLYASUNOV, A. On the use of semiempirical electrolyte theories for modeling of solution chemical data. *Pure and Applied Chemistry*, v. 69, n. 5, p. 951–958, 1997. ISSN 0033-4545. Disponível em: <<http://iupac.org/publications/pac/69/5/0951/>>.

Citado na página 55.

GUGGENHEIM, E. L. the specific thermodynamic properties of aqueous solutions of strong electrolytes. *Philosophical Magazine Series 7*, v. 19, n. 127, p. 588–643, 1935.

Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/14786443508561403>>.

Citado na página 49.

HAMER, W. J.; WU, Y.-C. Osmotic coefficients and mean activity coefficients of uni-univalent electrolytes in water at 25 °C. *Journal of Physical and Chemical Reference Data*, NIST, v. 1, n. 4, p. 1047–1100, 1972. ISSN 0047-2689. Disponível em: <<http://link.aip.org/link/?JPR/1/1047/1>>. Citado na página 102.

ILIUTA, M. C.; THOMSEN, K.; RASMUSSEN, P. Extended uniquac model for correlation and prediction of vapour-liquid-solid equilibria in aqueous salt systems containing non-electrolytes. part a. methanol-water-salt systems. *Chemical Engineering Science*, v. 55, n. 14, p. 2673–2686, abr. 2000. ISSN 0009-2509. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0009250999005345>>. Citado na página 54.

ISBIN, H. S.; KOBE, K. A. The solubility of some salts in ethylenediamine, monoethanolamine and ethylene glycol. *Journal of the American Chemical Society*, v. 67, n. 3, p. 464–465, 1945. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ja01219a029>>. Citado na página 42.

JOUYBAN, A.; SOLTANPOUR, S.; CHAN, H.-K. A simple relationship between dielectric constant of mixed solvents with solvent composition and temperature. *International Journal of Pharmaceutics*, v. 269, n. 2, p. 353–360, jan. 2004. ISSN 0378-5173. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0378517303005118>>. Citado na página 61.

JR., J. F. Z. et al. *Handbook of Aqueous Electrolyte Thermodynamics: Theory & Application*. 1. ed. Nova Iorque: Wiley-AIChE, 1986. 878 p. ISBN 978-0-8169-0350-4. Disponível em: <<http://www.amazon.com/Handbook-Aqueous-Electrolyte-Thermodynamics-Application/dp/0816903506%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0816903506>>. Citado 2 vezes nas páginas 54 e 55.

KONTOGEORGIS, G. M.; FOLAS, G. K. *Thermodynamic Models for Industrial Applications: From Classical and Advanced Mixing Rules to Association Theories*. Wiley, 2010. ISBN 978-0-470-69726-9. Disponível em: <<http://www.amazon.com/Thermodynamic-Models-Industrial-Applications-Association/dp/0470697261%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0470697261>>.

3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0470697261>.

Citado na página 47.

KOSINSKI, J. J. et al. Modeling acid-base equilibria and phase behavior in mixed-solvent electrolyte systems. *Fluid Phase Equilibria*, v. 256, n. 1–2, p. 34–41, ago. 2007. ISSN 0378-3812. <ce:title>16th Symposium on Thermophysical Properties</ce:title> <xocs:full-name>Proceedings of the Sixteenth Symposium on Thermophysical Properties held jointly with Nineteenth International Conference on Chemical Thermodynamics and Sixty-First Calorimetry Conference</xocs:full-name>. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0378381206004973>>. Citado na página 54.

KRAUS, K. A.; RARIDON, R. J.; BALDWIN, W. H. Properties of organic-water mixtures. i. activity coefficients of sodium chloride, potassium chloride, and barium nitrate in saturated water mixtures of glycol, glycerol, and their acetates. model solutions for hyperfiltration membranes. *Journal of the American Chemical Society*, v. 86, n. 13, p. 2571–2576, abr. 1964. ISSN 0022-3654. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ja01067a010>>. Citado na página 42.

LEE, L. L. A molecular theory of setchenov's salting-out principle and applications in mixed-solvent electrolyte solutions. *Fluid Phase Equilibria*, v. 131, n. 1–2, p. 67–82, maio 1997. ISSN 0378-3812. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0378381296031986>>. Citado na página 52.

LIDE, D. R.; HAYNES, W. M. M. (Ed.). *CRC Handbook of Chemistry and Physics, 90th Edition (CRC Handbook of Chemistry & Physics)*. 90. ed. Boca Raton: CRC Press, 2009. 2804 p. ISBN 978-1-4200-9084-0. Disponível em: <<http://www.amazon.com/Handbook-Chemistry-Physics-90th-Edition/dp/1420090844%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D1420090844>>. Citado 3 vezes nas páginas 42, 61 e 64.

LINKE, W. F.; SEIDELL, A. *Solubilities Inorganic and Metal-Organic Compounds A-Ir: A Compilation of Solubility Data from the Periodical Literature, Volume 1*. 4. ed. Princeton: Van Nostrand, 1958. 1491 p. ISBN 978-0-8357-4124-8. Citado na página 42.

LINKE, W. F.; SEIDELL, A. *Solubilities Inorganic and Metal-Organic Compounds K-Z: A Compilation of Solubility Data from the Periodical Literature, Volume II*. 4. ed. Washington: American Chemical Society, 1965. 1914 p. ISBN 978-0-608-04425-5. Citado na página 42.

LIU, Y.; WATANASIRI, S. Representation of liquid-liquid equilibrium of mixed-solvent electrolyte systems using the extended electrolyte nrtl model. *Fluid Phase Equilibria*, v. 116, n. 1–2, p. 193–200, mar. 1996. ISSN 0378-3812. <ce:title>Proceedings of the Seventh International Conference on Fluid Properties and Phase Equilibria for Chemical Process Design</ce:title>. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/0378381295028870>>. Citado na página 54.

LORIMER, J. W. Thermodynamics of solubility in mixed solvent systems. *Pure and Applied Chemistry*, v. 65, n. 2, p. 183–191, 1993. ISSN 0033-4545. Disponível em:

<<http://iupac.org/publications/pac/65/2/0183/>>. Citado 4 vezes nas páginas 52, 53, 55 e 64.

MARCUS, Y. *Ion Solvation*. 1. ed. Chichester: Wiley-Interscience, 1985. 314 p. ISBN 978-0-471-90756-5. Disponível em: <<http://www.amazon.com/Ion-Solvation-Yitzhak-Marcus/dp/0471907561%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0471907561>>. Citado na página 64.

MOURA, N. R. d. *Desafios Tecnológicos para o Gás Natural*. 2007. Disponível em: <http://sites.petrobras.com.br/minisite/premiotecnologia/pdf/TecnologiaGas_DesafiosTecnologicos-GasNatural.pdf>. Acesso em: 02 maio 2014. Citado na página 34.

NAZZER, C. A.; KEOGH, J. Paper presented at Offshore Technology Conference, Houston, TX, USA, *Advances in Glycol Reclamation Technology*. 2006. Disponível em: <<http://e-book.lib.sjtu.edu.cn/otc-2006/pdfs/otc18010.pdf>>. Citado 2 vezes nas páginas 37 e 38.

NICOLAISEN, H. Ph. D. Thesis. Department of Chemical Engineering. Technical University of Denmark; DK-2800 Lyngby, Denmark, *Phase equilibria in aqueous electrolyte solutions*. 1994. Citado na página 54.

OLIVEIRA, J. A. F. et al. Latin American Applied Research (em Publicação), *Levenberg-Marquardt method applied to the determination of vapor-liquid equilibrium model parameters*. 2014. Citado na página 44.

OLIVEIRA, J. A. F. et al. *Modelagem e Simulação da Solubilidade de Sais em Mistura de Solventes em Sistemas Aquosos com dois Sais Fortes e Monoetileno Glicol*. 2010. Citado na página 52.

PETROBRAS. *Atividade de Produção de Gás e Condensado no Campo de Mexilhão, Bacia de Santos : EIA – Estudo de Impacto Ambiental*. 2007. Disponível em: <<http://licenciamento.ibama.gov.br/Petroleo/Sistema%20de%20Producao%20de%20Gas%20Natural%20e%20Condensados%20-%20Campo%20de%20Mexilhao%20-%20Bacia%20de%20Santos/EIA%20Mexilh%C3%A3o/>>. Acesso em: 02 maio 2014. Citado 5 vezes nas páginas 25, 31, 32, 33 e 44.

PINHO, S. P.; MACEDO, E. A. Representation of salt solubility in mixed solvents: A comparison of thermodynamic models. *Fluid Phase Equilibria*, v. 116, n. 1–2, p. 209–216, mar. 1996. ISSN 0378-3812. <ce:title>Proceedings of the Seventh International Conference on Fluid Properties and Phase Equilibria for Chemical Process Design</ce:title>. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/0378381295028897>>. Citado 2 vezes nas páginas 42 e 54.

PINHO, S. P.; MACEDO, E. A. Solubility of nacl, nabr, and kcl in water, methanol, ethanol, and their mixed solvents. *Journal of Chemical & Engineering Data*, v. 50, n. 1, p. 29–32, jan. 2005. ISSN 0021-9568. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/je049922y>>. Citado na página 42.

PITZER, K. S. Thermodynamics of electrolytes. i. theoretical basis and general equations. *The Journal of Physical Chemistry*, v. 77, n. 2, p. 268–277, jan. 1973. ISSN 0022-3654. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/j100621a026>>. Citado 2 vezes nas páginas 55 e 62.

- PITZER, K. S. *Thermodynamics*. 3. ed. Nova Iorque: McGraw-Hill, 1995. 626 p. ISBN 978-0-07-050221-5. Disponível em: <<http://www.amazon.com/Thermodynamics-Mcgraw-Series-Advanced-Chemistry/dp/0070502218%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0070502218>>. Citado na página 55.
- PRAUSNITZ, J. M.; LICHTENTHALER, R. N.; AZEVEDO, E. G. de. *Molecular Thermodynamics of Fluid-Phase Equilibria*. 3. ed. Upper Sanddle River: Prentice-Hall PTR, 1999. 864 p. ISBN 978-0-13-977745-5. Disponível em: <<http://www.amazon.com/Molecular-Thermodynamics-Fluid-Phase-Equilibria-Edition/dp/0139777458%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0139777458>>. Citado 6 vezes nas páginas 46, 47, 48, 49, 50 e 55.
- RENON, H. Models for excess properties of electrolyte solutions: molecular bases and classification, needs and trends for new developments. *Fluid Phase Equilibria*, v. 116, n. 1–2, p. 217–224, mar. 1996. ISSN 0378-3812. <ce:title>Proceedings of the Seventh International Conference on Fluid Properties and Phase Equilibria for Chemical Process Design</ce:title>. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/0378381295028900>>. Citado na página 54.
- SANDENG, K.; KAASA, B. Estimation of monoethylene glycol (meg) content in water + meg + nacl + nahco₃ solutions. *Journal of Chemical & Engineering Data*, v. 51, n. 2, p. 443–447, mar. 2006. ISSN 0021-9568. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/je0503711>>. Citado na página 42.
- SANDLER, S. I. (Ed.). *Models for Thermodynamic and Phase Equilibria Calculations*. 1. ed. Nova Iorque: Marcel Dekker, 1993. 704 p. ISBN 978-0-8247-9130-8. Disponível em: <<http://www.amazon.com/Thermodynamic-Equilibria-Calculations-Chemical-Industries/dp/0824791304%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0824791304>>. Citado na página 55.
- SILVESTER, L. F.; PITZER, K. S. Thermodynamics of electrolytes. 8. high-temperature properties, including enthalpy and heat capacity, with application to sodium chloride. *The Journal of Physical Chemistry*, v. 81, n. 19, p. 1822–1828, 1977. ISSN 0022-3654. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/j100534a007>>. Citado na página 62.
- SMITH, J. M.; NESS, H. C. van; ABBOTT, M. M. *Introduction to Chemical Engineering Thermodynamics*. 7. ed. Nova Iorque: McGraw-Hill, 2005. 840 p. ISBN 978-0-07-310445-4. Disponível em: <<http://www.amazon.com/Introduction-Chemical-Engineering-Thermodynamics-Mcgraw-Hill/dp/0073104450%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0073104450>>. Citado na página 55.
- STATOIL. *Gas Hydrates as a resource : Technology Readiness Level*. Disponível em: <<http://nfpweb.org/oneday2011/Gas%20hydrates,%20Technology%20Readiness%20Level.pdf>>. Acesso em: 02 maio 2014. Citado na página 34.

THOMSEN, K. Ph. D. Thesis. Department of Chemical Engineering. Technical University of Denmark; DK-2800 Lyngby, Denmark, *Aqueous electrolytes: model parameters and process simulation*. 1997. Citado 2 vezes nas páginas 54 e 55.

THOMSEN, K.; ILIUTA, M. C.; RASMUSSEN, P. Extended uniquac model for correlation and prediction of vapor-liquid-liquid-solid equilibria in aqueous salt systems containing non-electrolytes. part b. alcohol (ethanol, propanols, butanols)-water-salt systems. *Chemical Engineering Science*, v. 59, n. 17, p. 3631 – 3647, set. 2004. ISSN 0009-2509. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S000925090400329X>>. Citado na página 54.

TRIMBLE, H. M. Solubilities of salts in ethylene glycol and in its mixtures with water. *Industrial & Engineering Chemistry*, v. 23, n. 2, p. 165–167, fev. 1931. ISSN 0019-7866. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ie50254a016>>. Citado na página 42.

TSIERKEZOS, N. G.; MOLINOU, I. E. Thermodynamic properties of water + ethylene glycol at 283.15, 293.15, 303.15, and 313.15 K. *Journal of Chemical & Engineering Data*, v. 43, n. 6, p. 989–993, nov. 1998. ISSN 0021-9568. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/je9800914>>. Citado na página 42.

WALAS, S. M. *Phase Equilibria in Chemical Engineering*. Butterworth, US, 1985. ISBN 0409951625. Disponível em: <<http://www.amazon.com/Phase-Equilibria-Chemical-Engineering-Stanley/dp/0409951625%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0409951625>>. Citado na página 55.

WANG, P. et al. Modeling phase equilibria and speciation in mixed-solvent electrolyte systems: II. liquid-liquid equilibria and properties of associating electrolyte solutions. *Journal of Molecular Liquids*, v. 125, n. 1, p. 37 – 44, mar. 2006. ISSN 0167-7322. <ce:title>Selected Papers on Molecular Liquids</ce:title> <xocs:full-name>18th ICCT</xocs:full-name>. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0167732205002060>>. Citado na página 54.

WANG, P.; ANDERKO, A.; YOUNG, R. D. A speciation-based model for mixed-solvent electrolyte systems. *Fluid Phase Equilibria*, v. 203, n. 1–2, p. 141 – 176, dez. 2002. ISSN 0378-3812. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0378381202001784>>. Citado na página 54.

WIKIPEDIA. *Etilenoglicol* — *Wikipedia, A Enciclopédia Livre*. 2014. Disponível em: <<http://pt.wikipedia.org/wiki/Etilenoglicol>>. Acesso em: 02 maio 2014. Citado na página 43.

ZHOU, Y. et al. Compositions, densities, and refractive indices for the ternary systems ethylene glycol + nacl + h₂o, ethylene glycol + kcl + h₂o, ethylene glycol + rbcl + h₂o, and ethylene glycol + cscl + h₂o at 298.15 K. *Journal of Chemical & Engineering Data*, v. 55, n. 3, p. 1289–1294, 2010. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/je900630n>>. Citado na página 42.

ZOUAIN, R. N. A. *Sistemas de Produção, Refino e Transporte de Petróleo - Promimp - Engenheiro de Automação*. 2014. Disponível em: <http://www.producao.ufrgs.br/arquivos/disciplinas/492_aula_1_5.pdf>. Acesso em: 23 maio 2014. Citado na página 36.

ÅKERLÖF, G. Dielectric constants of some organic solvent-water mixtures at various temperatures. *Journal of the American Chemical Society*, v. 54, n. 11, p. 4125–4139, 1932. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ja01350a001>>. Citado na página 60.

Apêndices

APÊNDICE A – ARTIGO

A.1 DE CORRELAÇÃO DE DADOS DE EQUILÍBRIO LÍQUIDO VAPOR PELO MODELO UNIQUAC ORIGINAL PARA SISTEMAS AQUOSOS COM SAL E ÁLCOOL

LEVENBERG-MARQUARDT METHOD APPLIED TO THE DETERMINATION OF VAPOR-LIQUID EQUILIBRIUM MODEL PARAMETERS

J.A.F. OLIVEIRA[†], M.M.L. DUARTE[†], E.L. FOLETTTO^{‡*} and O. CHIAVONE-FILHO[†]

[†] Chemical Eng. Department, Federal University of Rio Grande do Norte, Natal, 59066-800, Brazil

[‡] Chemical Eng. Department, Federal University of Santa Maria, Santa Maria, 97105-900, Brazil. * efoletto@gmail.com

Abstract— In order to correlate and optimize experimental data either from the laboratory or industry, one needs a robust method of data regression. Among the non-linear parameter estimation methods it may be pointed out of Levenberg, which applies the conversion of an arbitrary matrix into a positive definite one. Later, Marquardt applied the same procedure, calculating λ parameter in an iterative form. The Levenberg-Marquardt algorithm is described and two routine for correlating vapor-liquid equilibrium data for pure component and mixtures, based on this efficient method, have been applied. The routines have been written with an interface very accessible for both users and programmers, using Python language. The flexibility of the developed programs for introducing the desired details is quite interesting for both process simulators and modeling properties. Furthermore, for mixtures with electrolytes, it was obtained a coherent and compatible relation for the structural parameters of the salt species, with the aid of the method and the graphical interface designed.

Keywords— Nonlinear Regression, Levenberg-Marquardt Method, Vapor-Liquid Equilibria, Mixed Solvents Electrolyte Systems.

I. INTRODUCTION

The phenomena and properties involved in the chemical processes present, usually, non-linear behavior. Therefore, it is needed a robust method of non-linear regression for providing the required correlation of the experimental data collected, either from the laboratory or from the industry.

Bard (1974) pointed out three methods of parameter estimation, i.e., least squares, maximum likelihood and Bayesian. The method of least squares is the oldest and most widely used estimation procedure. It can be applied directly to a deterministic model, without any knowledge being taken of the probability distribution of the observations. This feature satisfies quite well the propose of correlating most of the desired thermodynamic properties and process variables.

For solving these unconstrained problems, the Levenberg-Marquardt method has proven very reliable results. In this work the Levenberg-Marquardt algorithm is described in a simple form.

This parameter estimation method has been reproduced and applied for the correlation of vapor-liquid equilibrium (VLE) data using activity coefficient and equation of state approaches (Lazzús, 2010; Camacho-

Camacho *et al.*, 2007). For the first case, an evaluation of the thermodynamic consistency of the data can be provided, since activity coefficient models fulfill the exact definition of partial molar excess Gibbs energy.

Systematic evaluation of thermodynamic properties of solutions of sodium chloride (NaCl) in water, e.g., boiling point and osmotic coefficient, with an extensive experimental database, was provided by Clarke and Glew (1985). VLE calculation for mixed solvents electrolyte systems have been applied using functional group activity coefficient model, i.e., UNIFAC (Kikic *et al.*, 1991). It was taken into account the electrostatic term with Debye-Huckel term, normalized according to the theory of Mc-Millan-Mayer. However, the estimated parameters present scattering and large order of magnitude. Loehe and Donohue (1997) described a survey of the literature models to determine thermodynamic properties of aqueous systems with strong electrolytes. Thomsen *et al.* (2004) have presented a systematic modeling work with mixed electrolytes and mixed solvents systems for solid-liquid-vapor equilibrium data, using UNIQUAC also. Haghtalab and Peyvandi (2009) have proposed a new version of UNIQUAC model for electrolyte solutions that requires just two fitted interaction parameters per binary aqueous system. The structural parameters of the ionic species were evaluated considering solvation for the determination of the radii.

This study aimed to describe the Levenberg-Marquardt computational procedure for estimation of vapor-liquid equilibrium model parameters. Validations of the method are present for pure component vapor pressure and binary VLE data. Application to strong electrolytes in mixed solvents systems has also been performed, using UNIQUAC model. It is proposed a new form to determine structural UNIQUAC parameters of the ions that is compatible with the solvent species parameters.

II. METHODS

A. Levenberg-Marquardt Method

The method presented by Levenberg (1944) utilizes a conversion of an arbitrary matrix into a positive definite one. This method rests on the observation that if $\underline{P}$ is any positive definite matrix, then $\underline{A} + \lambda \underline{P}$ is positive definite for λ sufficiently large, no matter what $\underline{A}$. Marquardt (1963) suggested an iterative algorithm for the selection of λ .

The importance of working with a definite positive matrix is to satisfy the necessary and sufficient conditions of minimization in the data regression, where the least square of the residuals (e_i) is to be found.

B. Levenberg-Marquardt Algorithm

In this section the Levenberg-Marquardt algorithm applied for a nonlinear parameter estimation is described in four phases.

Initialization:

- (i) Enter $\rho_{\min}$ (a minimum value of the step length ρ , e.g. 10^{-7}), ε (a positive small number, e.g. 10^{-7}), $\underline{y}$ (vector of the dependent variables), $\underline{X}$ (matrix of the independent variables) and $\underline{\theta}_1$ (vector of the initial guess of the parameters to be estimated). Set $J = 1$ and $\lambda = 0.01$ (Marquardt parameter);
 - (ii) Calculate $\underline{e}$ (vector of the residuals) and $\Phi_1 = \Phi(\underline{\theta}_1)$ (objective function), where $e_i = y_i - f_i(\underline{X}, \underline{\theta})$ and $\Phi = \sum_i e_i^2$;
 - (iii) Calculate $\underline{q}$ (gradient of Φ) and $\underline{A}$ (approximation of the Hessian matrix of the function Φ), where $q_i = \frac{\partial \Phi_1}{\partial \theta_i} = 2 \sum_j e_j \frac{\partial e_j}{\partial \theta_i}$ and $A_{ij} = 2 \sum_k \left(\frac{\partial e_k}{\partial \theta_i} \right) \left(\frac{\partial e_k}{\partial \theta_j} \right)$;
 - (iv) Calculate $\underline{P} = \begin{cases} i \neq j \Rightarrow P_{ij} = 0 \\ i = j, A_{ij} = 0 \Rightarrow P_{ij} = 1 \\ i = j, A_{ij} \neq 0 \Rightarrow P_{ij} = A_{ij} \end{cases}$;
 - (v) Calculate $\underline{A} + \lambda \underline{P}$, reminding that these values may be directly evaluated by: $A_{ij} + \lambda P_{ij} = \begin{cases} i \neq j \Rightarrow A_{ij} + \lambda P_{ij} = A_{ij} \\ i = j, A_{ij} = 0 \Rightarrow A_{ij} + \lambda P_{ij} = \lambda \\ i = j, A_{ij} \neq 0 \Rightarrow A_{ij} + \lambda P_{ij} = A_{ij}(1 + \lambda) \end{cases}$;
 - (vi) Calculate $\underline{v} = -(\underline{A} + \lambda \underline{P})^{-1} \underline{q}$ (vector in the direction of the proposed step), $\underline{\theta}^{(1)} = \underline{\theta}_1 + \underline{v}$ and $\Phi^{(1)} = \Phi(\underline{\theta}^{(1)})$, where $(\underline{A} + \lambda \underline{P})^{-1}$ and $\underline{v}$ are evaluated by the Cholesky method, see, e.g., Fox (1964), which is specific for the inversion of a definite positive matrix;
 - (vii) If $\Phi^{(1)} < \Phi_1$ accept $\underline{\theta}_2 = \underline{\theta}^{(1)}$, replace λ by $\max(0.1\lambda, \varepsilon)$ and go to the step (xxi). Or else, continue from the step (viii);
 - (viii) Compute $\rho^{(1)} = 2^{-J}$, $\Psi^{(1)} = \Phi(\underline{\theta}_1 + \rho^{(1)} \underline{v})$ and $\gamma = \underline{q}^T \underline{v}$. With these values $\rho^* = \gamma \rho^{(1)2} / [2(\gamma \rho^{(1)} + \Phi - \Psi^{(1)})]$ is calculated;
 - (ix) If $\Psi^{(1)} < \Phi_1$, then continue from the step (x). Else, go to the step (xvi);
- Extrapolation of ρ :**
- (x) Set J equal to the higher integer number that do not overcome the half of J ;
 - (xi) If $\rho^* \leq 0$ set $\rho^* = 2^{1-J}$;

- (xii) If $\rho^* > 1.1\rho^{(1)}$ continue on the step (xiii). Else, set $\rho = \rho^{(1)}$ and go to the step (xx);
- (xiii) Compute $\Psi^{(2)} = \Phi(\underline{\theta}_1 + \rho^* \underline{v})$;
- (xiv) If $\Psi^{(2)} < \Psi^{(1)}$ continue on the step (xv). Else, set $\rho = \rho^{(1)}$ and go to the step (xx);
- (xv) Set $\rho = \rho^*$ and go to the step (xx);

Interpolation of ρ :

- (xvi) Set $\rho = \max[0.25\rho^{(1)}; \min(0.75\rho^{(1)}; \rho^*)]$;
- (xvii) If $\rho \leq \rho_{\min}$ go to the step (xx). Else, continue on the step (xviii);
- (xviii) Compute $\Psi^{(2)} = \Phi(\underline{\theta}_1 + \rho \underline{v})$ and set $J = J + 1$;
- (xix) If $\Psi^{(2)} < \Phi_1$ go to the step (xx). Else, set $\rho^{(1)} = \rho$

and $\rho^* = \frac{\gamma \rho^{(1)2}}{2(\gamma \rho^{(1)} + \Phi - \Psi^{(2)})}$, and return to the step (xvi);

- (xx) Calculate $\underline{\theta}_2 = \underline{\theta}_1 + \rho \underline{v}$;

Termination:

- (xxi) The criterion of termination is tested, using $\underline{\theta}_1$ and $\underline{\theta}_2$, as an example suggested by Marquardt (1963), is given according to the following equation:

$$|\theta_{i+1,j} - \theta_{i,j}| \leq \varepsilon_{j,j} \quad j = 1, 2, \dots, l$$

where $\varepsilon_j = 10^{-4}(\theta_{i,j} + 10^{-3})$.

If the criterion is satisfied for all parameters, iterations are terminated and $\underline{\theta}_2$ is accepted as the solution $\underline{\theta}^*$. Otherwise, set $\lambda = 10\lambda$ and $\underline{\theta}_1 = \underline{\theta}_2$, and return to the step (ii).

C. Vapor-Liquid Equilibrium Data Regression

In order to demonstrate the applicability of the implemented Levenberg-Marquardt method, two programs for correlating vapor-liquid equilibrium data have been developed.

Antoine Constants Estimation

The first program allows data correlation of vapor pressure and temperature for pure components, using the Antoine equation. In Eq. (1) we have that P^{sat} is the vapor pressure, T is the saturation temperature, and A , B and C are the adjusted Antoine constants. This form of the Antoine equation is usual in the literature; however the user may, easily, apply a modified version of the correlation both in terms of the logarithm base and units.

$$\log_{10} P^{\text{sat}} (\text{mmHg}) = A - \frac{B}{T(^{\circ}\text{C}) + C} \quad (1)$$

To perform the data regression, the following objective function (OF) is to be minimized, which is the defined by the absolute residual in terms of vapor pressure.

$$OF = \text{MIN} = \sum_i (P_{\text{exptl}} - P_{\text{calcd}})^2 \quad (2)$$

The program requires initial guesses of A , B and C pa-

rameters to be estimated, name of the substance, number of data points and list of the experimental pairs of vapor pressure and saturation temperature. Indication of the units and the literature source from which the data has been retrieved is also requested. An output is provided where information about the iterations performed during the execution is registered, allowing the detection and identification of an eventual error. After convergence, the values of the estimated constants are displayed together with respective interval confidences, standard deviations and correlation matrix of the parameters. The program also provides the relative and absolute deviations in terms of vapor pressure and the estimation of the normal boiling point, at 760 mmHg, using the obtained correlation. Results for propyl acetate have been compared with Boublik *et al.* (1984) and validated the implemented program. The Antoine correlation for the propyl acetate has presented relative deviation (ΔP) of 0.04% and estimated the following parameters with corresponding confidence intervals: $A = 7.03008 \pm 0.03391$, $B = 1290.200 \pm 19.505$ and $C = 209.466 \pm 2.161$.

Reduction and consistency test of vapor-liquid equilibrium data for binary systems

The second program developed is to accomplish the correlation of vapor-liquid equilibrium data for mixtures, using activity coefficient models presented in the literature (Prausnitz *et al.*, 1999), i.e. Margules 2-suffixes, Margules 3-suffixes, Van Laar, Wilson and NRTL (Non-Random Two Liquids). Therefore, data regression consists of the corresponding parameter estimation of the activity coefficient model. In the case that the VLE data set is complete, i.e. presents the values of temperature, total pressure, liquid and vapor compositions, the correlation tests also the thermodynamic consistency of the experimental data set. This is due to the fact that the activity coefficient models (γ_i) respect the exact thermodynamic definition of partial molar excess Gibbs energy.

$$\ln \gamma_i \equiv \left[\frac{\partial (nG^E/RT)}{\partial n_i} \right]_{T,P,n_{i \neq j}} = \frac{\bar{G}_i^E}{RT} \quad (3)$$

Thus, the Euler theorem may be applied, i.e. the additive property, together with the fundamental equation for G^E , resulting in the so-called Gibbs-Duhem relation, see Prausnitz *et al.* (1999). This is the base of the thermodynamic consistency test of an experimental phase equilibrium data set. We may find in the literature, in general, two tests for thermodynamic consistency of VLE data, i.e. area test and deviation test. In our case, we have applied the deviation test which is in fact a data correlation followed by the analysis of the deviations (see Fredenslund *et al.*, 1977). We may describe, easily, the VLE data correlation of a binary system by writing the total pressure equation (Barker, 1953), and neglecting the nonidealities of the vapor phase, which is a reasonable simplification for low pressures and nonassociating components.

$$P_{\text{calcd}} = x_1 \gamma_1 P_1^{\text{sat}} + (1 - x_1) \gamma_2 P_2^{\text{sat}} \quad (4)$$

Table 1. Correlation and consistency test of the vapor-liquid equilibrium data for the *n*-heptane(1) + ethyl butyrate(2) at 100 °C (Kojima and Tochigi, 1979).

Model	A	B	ΔP (%)	Δy_i^{**}
Margules	0.5563		1.36	0.0057
2- suffixes				
Margules	0.7486	0.4059	0.87	0.0097
3- suffixes				
Van Laar	0.4664	0.5769	0.76	0.0096
Wilson	0.3928	1.1612	1.02	0.0089
NRTL*	1.2171	-0.4020	0.98	0.0091

* C is the random parameter and has been set to 0.3, as default value.

** The values of the absolute average deviation are lower than 0.01 ($\Delta y_i < 0.01$), indicating that the data are considered to be consistent, according to Fredenslund *et al.* (1977).

It may be pointed out that the activity coefficients depend on the temperature, liquid mole fraction and their parameters, which are adjusted from phase equilibrium data reduction. To give an illustration, Eq. (5) shows the simple Margules 2-suffixes gamma model, for both components.

$$\ln \gamma_1 = Ax_2^2; \quad \ln \gamma_2 = Ax_1^2 \quad (5)$$

Therefore, the data reduction for the mixture is similar to the Antoine correlation, and may also be represented by Eq. (2), which defines the objective function in terms of the pressure. For the mixture the differences are that the saturation pressure is replaced by the total pressure and the parameters come from the activity coefficient model instead of the Antoine equation. After the data regression, the vapor composition may be evaluated, according to Eq. (6).

$$y_{1,\text{calcd}} = \frac{\gamma_1 x_1 P_1^{\text{sat}}}{x_1 \gamma_1 P_1^{\text{sat}} + (1 - x_1) \gamma_2 P_2^{\text{sat}}} \quad (6)$$

The vapor-liquid equilibrium data set for the *n*-heptane(1) + ethyl butyrate(2) system at 100 °C presented by Kojima and Tochigi (1979) has been used to test the nonlinear regression method implemented. The developed program allows parameter estimation of the five activity coefficient models listed at Table 1. It may be observed that all models have fitted satisfactorily the experimental data. Furthermore, the deviations presented indicate that the experimental data is consistent. Concerning the operation of the program for correlating vapor-liquid equilibrium data of mixtures, it may be pointed out that all information relative to the system is given by the input file. The model is chosen during the execution of the program, together with the initial guess of the gamma coefficients and the Marquardt parameter. An output file is also created. Similar to the Antoine program, this output file contains information related to the iterations realized, during the execution of the Levenberg-Marquardt method.

After convergence, the values of the estimated parameters together with confidence intervals are presented together with the correlation matrix and standard deviations. Furthermore, the dependent variables are properly compared, and for the case of isothermic data, the percent average pressure deviation and the absolute average mole fraction of the vapor phase deviation are

displayed, from which a thermodynamic consistence evaluation of the equilibrium data is prompt provided.

Vapor-liquid equilibrium correlation for mixtures with electrolytes

The Levenberg-Marquardt method was also applied for two aqueous ternary mixtures with methanol and NaCl and ethanol and NaCl with the second program. The corresponding binary systems have also been correlated. The UNIQUAC activity coefficient model (Abrams and Prausnitz, 1975) was applied in its original form and considering NaCl as one molecular species, i.e., neglecting the dissociation.

Table 2 reports structural parameters used for the studied species. It should be pointed out that structural parameters of the salt species, i.e., surface area and volume parameters were determined from the values of the water and a molar mass ratio. Two goals have been achieved using this approach. One was to provide data fitting capability with a reasonable order of magnitude of the structural parameters. Secondly, compatibility with the interaction parameters estimated for non-electrolyte binary mixtures are encountered. This is also an important feature for predictability of the model and group contribution applications like UNIFAC (Kikic *et al.*, 1991). Values of r and q for ionic species have been fitted by Macedo *et al.* (1990). Another approach encountered in the literature is the increment of the parameters from the ionic radii by considering solvation. However, the use of these two methods demands estimation of new solvent-solvent parameters and thereby compatibility requirement is not fulfilled. By the other hand, using the values of r and q calculated from the values of water and a ratio of the molar masses, as presented in Table 2, it was obtained the desired compatibility and simultaneously good correlation capability, eliminating the problem of low structural values due to the size of the ion for UNIQUAC equation.

Table 3 presents the estimated UNIQUAC interaction parameters for the species studied. It should be pointed out that the non-electrolyte binary systems have been correlated initial and separately and then with these predetermined values the salt-solvent constants were estimated from the corresponding experimental data. All calculations have presented absolute average deviation in terms of vapor mole fraction lower than 0.01 and relative average deviation in terms of pressure less than 2%. These deviations indicate the consistency of the experimental data and also the quality of the parameters that gather many experimental points and different sources.

Figures 1 and 2 illustrate the quality of the correlation obtained for the aqueous ternary systems with methanol and NaCl at 314.6 K and with ethanol and NaCl at 700 mmHg, respectively.

Table 2: Structural parameters for UNIQUAC model.

Species i	Methanol	Ethanol	Water	NaCl
r_i	1.4311	2.1055	0.9200	2.9845 ^a
q_i	1.4320	1.9720	1.4000	4.5417 ^b

$$^a r_i = r_{\text{water}} (M_i / M_{\text{water}}) \quad ; \quad ^b q_i = q_{\text{water}} (M_i / M_{\text{water}})$$

Table 3: Estimated UNIQUAC interaction parameters (a_{ij}) in kelvin for the studied systems with methanol (MeOH), ethanol (EtOH), water (H₂O) and sodium chloride (NaCl)

i \ j	MeOH	EtOH	H ₂ O	NaCl
MeOH	0.00	432.85 ^a	-103.32 ^b	-932.37 ^d
EtOH	-261.57 ^a	0.00	-12.47 ^c	-878.07 ^d
H ₂ O	150.07 ^b	166.12 ^c	0.00	-925.33 ^d
NaCl	5817.00 ^d	4722.55 ^d	53.67 ^d	0.00

^aParameters estimated from methanol+ethanol VLE data (Niesen *et al.*, 1986; Kurihara *et al.*, 1993).

^bParameters estimated from methanol+water VLE data (Soujanya *et al.*, 2010; Yao *et al.*, 1999; Kurihara *et al.*, 1993).

^cParameters estimated from ethanol+water VLE data (Pemberton and Mash, 1978; Niesen *et al.*, 1986; Kurihara *et al.*, 1993, 1995; Arce *et al.*, 1996; Navarro-Espinosa *et al.*, 2010; Lai *et al.* 2014).

^dParameters estimated from water+NaCl, methanol+water+NaCl and ethanol+water+NaCl VLE data (Clarke and Glew, 1985; Gmehling, 1997; Yang and Lee, 1998; Jödecke *et al.*, 2005), fixing MeOH+EtOH, MeOH+H₂O and EtOH+H₂O parameters determined previously^{a,b,c}.

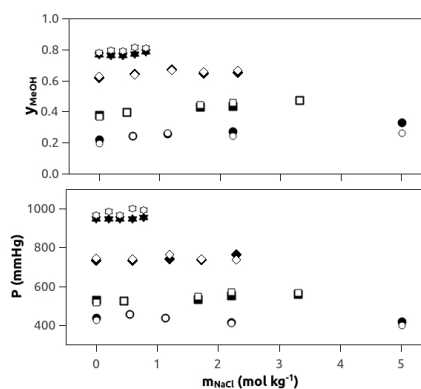


Fig. 1. Vapor-liquid equilibrium diagrams for methanol(1)+water(2)+NaCl(3) at 314.6 K; solid symbols experimental (Jödecke *et al.*, 2005) and open symbols: UNIQUAC this work: ● and ○, $x'_1=0.0328$; ▲ and △, $x'_1=0.0789$; ■ and □, $x'_1=0.216$; ▼ and ▽, $x'_1=0.466$.

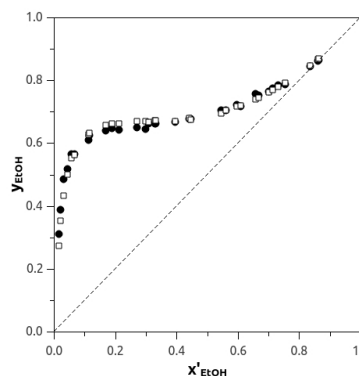


Fig. 2. VLE y-x diagram for the system ethanol(1)+water(2)+NaCl(3) at 700 mmHg; 32 experimental points with salt mole fraction in the range of 0.001-0.1; ● experimental data (Gmehling, 1997); ○ calculated values with UNIQUAC (this work).

III. CONCLUSIONS

The coherence of the developed programs has been verified via comparison with the parameters estimated for the same data set, confirming the reliability of the Levenberg-Marquardt routines implemented.

The results of the correlation program for mixtures are also efficient and analysis of the deviations allows the evaluation of the thermodynamic consistence of the experimental data. This qualification of the data may be useful for their application, i.e. for the development of models, and design and operation of processes.

The developed tool of calculation may be applied in problems of simulation and optimization of processes, using inclusive the visual recourses, which facilitates the interaction with the users. Furthermore, new correlation programs may also be developed to describe other thermodynamic and physical-chemistry properties, since Levenberg-Marquardt has been properly implemented.

These new salt species structural parameters demonstrated also to be flexible for the data correlation and coherent to respect to order of magnitude. The electrostatic contribution has not been considered for VLE calculations. The original form of the UNIQUAC model demonstrated to be feasible with the molecular approach for strong electrolyte species in mixed solvents systems.

NOMENCLATURE

a_{ij}	UNIQUAC interaction parameters
A, B, C	Antoine constants or activity coefficient parameters
$\underline{\underline{A}}$	approximation of the Hessian matrix of the function Φ
calcd	calculated value
$\underline{e}$	vector of the residuals
e_i	residue of point i , $e_i = y_i - f_i(\underline{X}, \underline{\theta})$
exptl	experimental value
$G^E, \overline{G}_i^E$	molar excess Gibbs energy and partial molar excess Gibbs energy of species i
M	number of experimental data points or atomic mass or molecular mass
m_i	molality of species i
n, n_i	total number of moles and number of moles of species i
NRTL	<i>Non Random Two-Liquid</i>
OF	objective function
P	total pressure
$\underline{P}$	any positive definite matrix
P_i^{sat}	vapor pressure of species i
$\underline{q}$	gradient of Φ or UNIQUAC surface area parameter
r, q	volume and surface area parameter
R	universal ideal gas constant
T	temperature
VLE	<i>Vapor-Liquid Equilibrium</i>
$\underline{X}$	matrix of the independent variables
x_i, y_i	mole fraction of the component i in the liq-

uid and vapor phase

x_i	salt free mole fraction of the component i in the liquid phase
$\underline{y}$	vector of the dependent variables
	mole fraction of the component i in the vapor phase

Greek letters

ΔP	percent average deviation, defined as $100/M \sum_{i=1}^M (P_{\text{exptl},i} - P_{\text{calcd},i}) / P_{\text{exptl},i}$
Δy_1	absolute average deviation, defined as $1/M \sum_{i=1}^M y_{1,\text{exptl},i} - y_{1,\text{calcd},i} $
ε	a positive small number
γ_i	activity coefficient of the component i
$\underline{\theta}$	vector of the parameters
$\underline{\theta}_1$	vector of the parameters, initial guess
$\underline{\theta}_i$	vector of the parameters at iteration i
ρ	value of the step length
λ	Marquadt parameter
$\underline{v}$	vector in the direction of the proposed step
Φ	objective function
Φ_1	objective function at the iteration 1
$\Psi^{(1)}$	objective function at $\underline{\theta}_1 + \rho^{(1)} \underline{v}$

REFERENCES

- Abrams, D.S. and J.M. Prausnitz, "Statistical thermodynamics of liquid mixtures: a new expression for the excess Gibbs energy of partly or completely miscible systems," *AIChE J.*, **21**, 116-128 (1975).
- Arce, A., J. Martínez-Ageitos and A. Soto, "VLE for water + ethanol + 1-octanol mixtures. Experimental measurements and correlations," *Fluid Phase Equil.*, **122**, 117-129 (1996).
- Bard, Y., *Nonlinear Parameter Estimation*, Academic Press, New York (1974).
- Barker, J.A., "Determination of Activity Coefficients from Pressure Total Measurements," *Austral. J. Chem.*, **6**, 207-210 (1953).
- Boublik, T., V. Fried and E. Hála, *The Vapor Pressure of Pure Substances*, Elsevier, Amsterdam (1984).
- Camacho-Camacho, L.E., L.A. Galicia-Luna, O. Elizalde-Solis and Z. Martinez-Ramirez, "New isothermal vapor-liquid equilibria for the CO₂ + n-nonane, and CO₂ + n-undecane systems," *Fluid Phase Equilibria*, **259**, 45-50 (2007).
- Clarke, E.C. and D.N. Glew, "Evaluation of the Thermodynamic Functions for Aqueous Sodium Chloride from Equilibrium and Calorimetric Measurements below 154 °C," *J. Phys. Chem. Ref. Data*, **14**, 2, 489-610 (1985).
- Fox, L., *An Introduction to Numerical Linear Algebra*, Oxford Univ. Press (Clarendon), London and New

- York (1964).
- Fredenslund, A., J. Gmehling and P. Rasmussen, *Vapor-Liquid Equilibria Using UNIFAC*, Elsevier, Amsterdam (1977).
- Gmehling, J., *Dortmund Data Bank (DDB) – electrolytes (ELE)*. Oldenburg, Germany: DDBST GmbH (1997).
- Haghtalab, A. and K. Peyvandi, “Electrolyte-UNIQUAC-NRF model for the correlation of the mean activity coefficient of electrolyte solutions,” *Fluid Phase Equilibria*, **281**, 163–171 (2009).
- Jödecke, M., A.P.S. Kamps and G. Maurer, “Experimental Investigation of the Influence of NaCl on the Vapor-Liquid Equilibrium of CH₃OH + H₂O,” *J. Chem. Eng. Data*, **50**, 138–141 (2005).
- Kikic, I., M. Fermeglia and P. Rasmussen, “UNIFAC Prediction of Vapor-Liquid Equilibria in Mixed Solvent-Salt Systems,” *Chemical Eng. Science*, **46**, 2775–2780 (1991).
- Kojima, K. and K. Tochigi, *Prediction of Vapor-Liquid Equilibria by the ASOG*, Elsevier, Amsterdam (1979).
- Kurihara, K. M. Nakamichi and K. Kojima, “Isobaric vapor-liquid equilibria for methanol + ethanol + water and the three constituent binary systems,” *J. Chem. Eng. Data*, **38**, 446–449 (1993).
- Kurihara, K., T. Minoura, K. Takeda and K. Kojima, “Isothermal Vapor-Liquid Equilibria for Methanol + Ethanol+Water, Methanol+Water, and Ethanol + Water,” *J. Chem. Eng. Data*, **40**, 679–684 (1995).
- Lai, H.S., Y. Lin and C.-H. Tu, “Isobaric (vapor + liquid) equilibria for the ternary system of (ethanol + water + 1,3-propanediol) and three constituent binary systems at P = 101.3 kPa,” *J. Chem. Thermodyn.*, **68**, 13–19 (2014).
- Lazzús, J.A., “Optimization of activity coefficient models to describe vapor–liquid equilibrium of (alcohol + water) mixtures using a particle swarm algorithm,” *Comp. and Mathem. with Applicat.*, **60**, 2260–2269 (2010).
- Levenberg, K., “A method for the solution of certain nonlinear problems in least squares,” *Quart. Appl. Math.*, **2**, 164–168 (1944).
- Loeche, J.R. and M.D. Donohue, “Recent advances in modeling thermodynamic properties of aqueous strong electrolyte systems,” *AIChE J.*, **43**, 1, 180–195 (1997).
- Macedo, E.A., P. Skovborg and P. Rasmussen, “Calculation of phase equilibria for solutions of strong electrolytes in solvent-water mixtures,” *Chem. Eng. Sci.*, **45**, 875–882 (1990).
- Marquardt, D.W., “An algorithm for least squares estimation of nonlinear parameters,” *SIAM J.*, **11**, 431–441 (1963).
- Navarro-Espinosa, I.R., C.A. Cardona and J.A. López, “Experimental measurements of vapor–liquid equilibria at low pressure: Systems containing alcohols, esters and organic acids,” *Fluid Phase Equil.*, **287**, 141–145 (2010).
- Niesen, V., A. Palavra, A.J. Kidnay and V.F. Yesavage, “An apparatus for vapor-liquid equilibrium at elevated temperatures and pressures and selected results for the water-ethanol and methanol-ethanol systems,” *Fluid Phase Equil.*, **31**, 283–298 (1986).
- Pemberton, R.C. and C.J. Mash, “Thermodynamic properties of aqueous non-electrolyte mixtures II. Vapour pressures and excess Gibbs energies for water + ethanol at 303.15 to 363.15 K determined by an accurate static method,” *J. Chem. Thermodynamics*, **10**, 867–888 (1978).
- Prausnitz, J.M., R.N. Lichtenthaler and E.G. Azevêdo, *Molecular Thermodynamics of Fluid Phase Equilibria*, Prentice-Hall (1999).
- Soujanya, J., B. Satyavathi and T.E. Vittal Prasad, “Experimental (vapour + liquid) equilibrium data of (methanol + water + glycerol) and (methanol + glycerol) systems at atmospheric and sub-atmospheric pressures,” *J. Chem. Thermodyn.*, **42**, 621–624 (2010).
- Thomsen, K., M.C. Iliuta and P. Rasmussen, “Extended UNIQUAC model for correlation and prediction of vapor–liquid–liquid–solid equilibria in aqueous salt systems containing non-electrolytes. Part B. Alcohol (ethanol, propanols, butanols)–water–salt systems,” *Chem. Eng. Sci.*, **59**, 3631–3647 (2004).
- Yang, S. and C.S. Lee, “Vapor-Liquid Equilibria of Water + Methanol in the Presence of Mixed Salts,” *J. Chem. Eng. Data*, **43**, 558–561 (1998).
- Yao, J., H. Li and S. Han, “Vapor–liquid equilibrium data for methanol–water–NaCl at 45°C,” *Fluid Phase Equil.*, **162**, 253–260 (1999).

APÊNDICE B – PROGRAMA COMPUTACIONAL VLE

B.1 Interface gráfica do programa VLE

Form1.ui

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <ui version="4.0">
3    <class>Form</class>
4    <widget class="QWidget" name="Form">
5      <property name="geometry">
6        <rect>
7          <x>0</x>
8          <y>0</y>
9          <width>800</width>
10         <height>600</height>
11        </rect>
12      </property>
13      <property name="windowTitle">
14        <string>VLE Regression</string>
15      </property>
16      <layout class="QVBoxLayout" name="verticalLayout">
17        <item>
18          <widget class="QLabel" name="label_Description">
19            <property name="text">
20              <string>Regression of VLE data to calculate binary parameters (Aij) of the UNIQUAC
21              model</string>
22            </property>
23          </widget>
24        </item>
25        <item>
26          <layout class="QHBoxLayout" name="horizontalLayout_1">
27            <item>
28              <widget class="QLabel" name="label_Datafile">
29                <property name="text">
30                  <string>Datafile:</string>
31                </property>
32                <property name="buddy">
33                  <cstring>lineEdit_Datafile</cstring>
34                </property>
35              </widget>
36            </item>
37            <item>
38              <widget class="QLineEdit" name="lineEdit_Datafile"/>
39            </item>
40            <item>
41              <widget class="QPushButton" name="pushButton_Browse">
42                <property name="text">
43                  <string>&Browse</string>
44                </property>
45              </widget>
46            </item>
47          </layout>
48        </item>
49        <item>
50          <layout class="QHBoxLayout" name="horizontalLayout_2">
51            <item>
52              <widget class="QCheckBox" name="checkBox_FixedAij">
53                <property name="text">
54                  <string>Use &Fixed Aij</string>
55                </property>
56              </widget>
57            </item>
58            <item>
59              <widget class="QLineEdit" name="lineEdit_FixedAij">
60                <property name="enabled">
61                  <bool>false</bool>

```

```

62     </widget>
63 </item>
64 </layout>
65 </item>
66 <item>
67 <layout class="QHBoxLayout" name="horizontalLayout_3">
68 <item>
69 <widget class="QCheckBox" name="checkBox_TRange">
70 <property name="text">
71 <string>&Use only experimental data in the temperature range (oC):</string>
72 </property>
73 </widget>
74 </item>
75 <item>
76 <widget class="QLabel" name="label_TMin">
77 <property name="enabled">
78 <bool>>false</bool>
79 </property>
80 <property name="text">
81 <string>Mi&nimum =</string>
82 </property>
83 <property name="buddy">
84 <cstring>lineEdit_TMin</cstring>
85 </property>
86 </widget>
87 </item>
88 <item>
89 <widget class="QLineEdit" name="lineEdit_TMin">
90 <property name="enabled">
91 <bool>>false</bool>
92 </property>
93 </widget>
94 </item>
95 <item>
96 <widget class="QLabel" name="label_TMax">
97 <property name="enabled">
98 <bool>>false</bool>
99 </property>
100 <property name="text">
101 <string>Ma&ximum =</string>
102 </property>
103 <property name="buddy">
104 <cstring>lineEdit_TMax</cstring>
105 </property>
106 </widget>
107 </item>
108 <item>
109 <widget class="QLineEdit" name="lineEdit_TMax">
110 <property name="enabled">
111 <bool>>false</bool>
112 </property>
113 </widget>
114 </item>
115 </layout>
116 </item>
117 <item>
118 <widget class="QTextEdit" name="textEdit"/>
119 </item>
120 <item>
121 <layout class="QHBoxLayout" name="horizontalLayout_4">
122 <item>
123 <widget class="QPushButton" name="pushButton_Run">
124 <property name="text">
125 <string>&Run</string>
126 </property>
127 </widget>
128 </item>
129 <item>
130 <spacer name="horizontalSpacer">
131 <property name="orientation">
132 <enum>Qt::Horizontal</enum>
133 </property>
134 <property name="sizeHint" stdset="0">
135 <size>
136 <width>40</width>
137 <height>20</height>
138 </size>
139 </property>
140 </spacer>
141 </item>

```

```

142     <item>
143         <widget class="QPushButton" name="pushButton_Exit">
144             <property name="text">
145                 <string>&Exit</string>
146             </property>
147         </widget>
148     </item>
149 </layout>
150 </item>
151 </layout>
152 </widget>
153 <tabstops>
154     <tabstop>lineEdit_Datafile</tabstop>
155     <tabstop>pushButton_Browse</tabstop>
156     <tabstop>checkBox_FixedAij</tabstop>
157     <tabstop>lineEdit_FixedAij</tabstop>
158     <tabstop>checkBox_TRange</tabstop>
159     <tabstop>lineEdit_TMin</tabstop>
160     <tabstop>lineEdit_TMax</tabstop>
161     <tabstop>textEdit</tabstop>
162     <tabstop>pushButton_Run</tabstop>
163     <tabstop>pushButton_Exit</tabstop>
164 </tabstops>
165 <resources/>
166 <connections/>
167 </ui>

```

B.2 Código fonte do programa VLE desenvolvido para a correlação de dados de Equilíbrio Líquido-Vapor pelo modelo UNIQUAC Original para sistemas aquosos com sal e álcool

UNIQUAC.py

```

1  #!/usr/bin/env python
2  # -*- coding: utf8 -*-
3  from __future__ import division
4  from PySide.QtGui import QApplication, QDialog, QFileDialog
5  from PySide.QtCore import QFile, QTextStream
6  import sys
7  from numpy import *
8  import matplotlib.pyplot as plt
9
10 import ui_Form
11 import ThermodynamicModels
12 from numpy.f2py.auxfuncs import isarray
13
14 class Form(QDialog, ui_Form.Ui_Form):
15
16     def __init__(self):
17         super(Form, self).__init__()
18         self.setupUi(self)
19         self.textEdit.setStyleSheet('font: 8pt "Courier";')
20         self.checkBox_FixedAij.stateChanged.connect(self.toggleCheckBox_FixedAij)
21         self.checkBox_TRange.stateChanged.connect(self.toggleCheckBox_TRange)
22         self.pushButton_Browse.clicked.connect(self.browse)
23         self.pushButton_Run.clicked.connect(self.runCalculations)
24         self.pushButton_Exit.clicked.connect(self.exitProgram)
25         self.show()
26
27     def toggleCheckBox_FixedAij(self):
28         flag = self.checkBox_FixedAij.isChecked()
29         self.lineEdit_FixedAij.setEnabled(flag)
30
31     def toggleCheckBox_TRange(self):
32         flag = self.checkBox_TRange.isChecked()
33         self.label_TMin.setEnabled(flag)
34         self.label_TMax.setEnabled(flag)
35         self.lineEdit_TMin.setEnabled(flag)
36         self.lineEdit_TMax.setEnabled(flag)
37

```

```

38 def browse(self):
39     self.textEdit.clear()
40     fName, _ = QFileDialog.getOpenFileName(self, 'Filename of the dataset', '.', 'Dataset
41         (*.dat);; All files (*.*)')
42     if fName: self.lineEdit_Datafile.setText(fName)
43     return
44
45 def runCalculations(self):
46     fName = self.lineEdit_Datafile.text()
47     if fName.strip() == '': return
48     if not QFile(fName).exists(): return
49     self.nl_regression(fName)
50
51 def exitProgram(self):
52     self.close()
53
54 def printText(self, text):
55     self.textEdit.insertPlainText(text.decode('utf-8'))
56
57 def nl_regression(self, fName):
58     import scipy.stats as stats
59     import scipy.optimize as optimize
60     self.textEdit.clear()
61     def readData(stream): return stream.readLine().split('#')[0].split()
62     f = QFile(fName)
63     f.open(QFile.ReadOnly | QFile.Truncate | QFile.Text)
64     stream = QTextStream(f)
65     self.printText('Datafile: {} \n \n'.format(fName))
66     Comp = readData(stream); NComp = len(Comp)
67     self.printText('VLE System: ')
68     for i in range(NComp-1):
69         self.printText('({}) {} \n'.format(i+1, Comp[i]))
70     self.printText('({}) {} \n'.format(NComp, Comp[-1]))
71     EqAntoine, r, q = [], [], []
72     for i in range(NComp):
73         EqAntoine_i = []
74         NEqs = int(readData(stream)[0])
75         if NEqs==0: EqAntoine.append(0.)
76         else:
77             for j in range(NEqs): EqAntoine_i.append(map(float, readData(stream)))
78             EqAntoine.append(EqAntoine_i)
79     for eq in EqAntoine:
80         if eq!=0.:
81             if len(eq)>1:
82                 T0=None
83                 for i in range(len(eq)-1):
84                     T1,T2,T3,T4=eq[i][-2],eq[i][-1],eq[i+1][-2],eq[i+1][-1]
85                     if i>0: eq[i-1][-1]=T0
86                     T0=(T2*(T4-T3)+T3*(T2-T1))/((T4-T3)+(T2-T1))
87                     eq[-2][-1]=T0; eq[-1][-2]=T0
88     for i in range(NComp):
89         line = map(float, readData(stream))
90         r.append(line[0]); q.append(line[1])
91     r = array(r); q = array(q)
92     x, y, PExp, TExp = [], [], [], []
93     for i in range(NComp): x.append([]); y.append([])
94     self.printText('Reference of data: \n')
95     NRef, Refs = 1, []
96     while not stream.atEnd():
97         notUse = int(readData(stream)[0])
98         ref = stream.readLine().split('#')[0].strip()
99         SysOrder = map(lambda x: int(x)-1, readData(stream))
100         NPoints = SysOrder.pop()+1
101         SysType = len(SysOrder)
102         NOutOfRange = 0
103         for _ in range(NPoints):
104             line = map(float, readData(stream))
105             if notUse: continue
106             if self.checkBox_TRange.isChecked():
107                 if line[-1]<float(self.lineEdit_TMin.text()) or
108                    line[-1]>float(self.lineEdit_TMax.text()):
109                     NOutOfRange += 1
110                     continue
111             for i in range(SysType-1):
112                 x[SysOrder[i]].append(line[i])
113                 y[SysOrder[i]].append(line[i+SysType-1])
114             x[SysOrder[SysType-1]].append(1.-sum(line[0:SysType-1]))
115             y[SysOrder[SysType-1]].append(1.-sum(line[SysType-1:2*SysType-2]))
116             for i in set(range(NComp))-set(SysOrder):
117                 x[i].append(0.)

```

```

116         y[i].append(0.)
117         PExp.append(line[-2])
118         TExp.append(line[-1])
119         Refs.append(NRef)
120     if notUse: txtNoUsed = ' >>> NOT USED <<<'
121     else: txtNoUsed = ''
122     if SysType < 1:
123         self.printText('\n=> ERROR: SysType < 2')
124     elif SysType == 2:
125         self.printText('{: < 35}'.format('({}) Binary VLE System'
126         '{:}'.format(NRef, tuple(array(SysOrder)+1))))
127     elif SysType == 3:
128         self.printText('{: < 35}'.format('({}) Ternary VLE System'
129         '{:}'.format(NRef, tuple(array(SysOrder)+1))))
130     elif SysType == 4:
131         self.printText('{: < 35}'.format('({}) Quaternary VLE System'
132         '{:}'.format(NRef, tuple(array(SysOrder)+1))))
133     elif SysType > 4:
134         self.printText('{: < 35}'.format('({}) n-ary VLE System'
135         '{:}'.format(SysType, tuple(NRef, array(SysOrder)+1))))
136     self.printText('{:4g} -{:4g} (out of T range) ={:4g} points => {}{}\n'
137     .format(NPoints, NOutOfRange, NPoints-NOutOfRange, ref, txtNoUsed))
138     NRef += 1
139     f.close()
140     x = array(x).T; y = array(y).T; PExp = array(PExp); TExp = array(TExp); NRef = array(NRef)
141     PSat = []
142     for T in TExp:
143         PSat_i = []
144         for i in range(NComp):
145             if type(EqAntoine[i]) == float: PSat_i.append(0.)
146             else:
147                 eqUsed = False
148                 for eq in EqAntoine[i]:
149                     A,B,C,T1,T2 = eq
150                     if T1 <= T < T2:
151                         PSat_i.append(10**(A-B/(C+T)))
152                         eqUsed = True; break
153                 if eqUsed: continue
154                 self.printText('\n=> ERROR: Impossible to calculate PSat')
155                 self.printText('without Antoine Equation ({}) for T = {} C.'
156                 .format(Comp[i], T))
157             return
158         PSat.append(PSat_i)
159     PSat = array(PSat)
160     TExp += 273.15 # C => K
161     theta_fixed_i, theta_fixed = [], []
162     def f_LCtoI(l,c,n):
163         a = l*n+c; return a-a/(n+1)-1
164     def f_ItoLC(i,n):
165         l = i/(n-1); c = i%(n-1)
166         if c >= l: c += 1
167         return l, c
168     if self.checkBox_FixedAij.isChecked():
169         for value in self.lineEdit_FixedAij.text().split(','):
170             lin, col, num = map(eval, value.split())
171             theta_fixed_i.append(f_LCtoI(lin-1, col-1, NComp))
172             theta_fixed.append(num)
173     theta_i = sorted(set(range(NComp*(NComp-1)))-set(theta_fixed_i))
174     def f(X,*theta):
175         for i,num in zip(theta_i, theta): a[f_ItoLC(i,NComp)] = num
176         gamma = ThermodynamicModels.UNIQUAC(r,q,a,X,TExp)
177         return array(map(sum,X*gamma*PSat))
178     def plus_minus(a,b): return a-b,a+b
179     theta0 = (random.rand(NComp*(NComp-1))-len(theta_fixed))*2000-1000.).astype(int).astype(float)
180     N,P = x.shape[0], theta0.shape[0]
181     a = zeros((NComp,NComp))
182     for i,num in zip(theta_fixed_i, theta_fixed): a[f_ItoLC(i,NComp)] = num
183     self.printText('\nAij = [ ]')
184     idx = 0
185     for i in range(NComp):
186         for j in range(NComp):
187             if i==j: self.printText('{:~10}'.format(0))
188             else:
189                 k = f_LCtoI(i,j,NComp)
190                 if k in theta_fixed_i:
191                     self.printText('{:~10}'.format('{:7.2f}'.format(theta_fixed[theta_fixed_i.index(k)])))
192                 else:
193                     self.printText('{:~10}'.format('theta[{}]'.format(idx))); idx += 1
194             if i < NComp-1: self.printText(']\n\nInitial Guess: \nAij = [ ]')
195     self.printText(']\n\nInitial Guess: \nAij = [ ]')
196     l=0

```

```

193     for i in range(NComp):
194         for j in range(NComp):
195             if i==j: self.printText('{:^10}'.format(0))
196             else:
197                 k = f_LCtoI(i,j,NComp)
198                 if k in theta_fixed_i:
199                     self.printText('{:^10}'.format('{:7.2f}'.format(theta_fixed[theta_fixed_i.index(k)])))
200                 else:
201                     self.printText('{:^10}'.format('{:7.2f}'.format(theta0[l])))
202                     l += 1
203             if i < NComp-1: self.printText(']\n\n')
204     self.printText(']\n\n')
205     try:
206         theta_optimum,cov_theta_optimum,_,errmsg,_ = optimize.curve_fit(f,x,PExp,theta0,
207                                 full_output=1)
208     except:
209         self.printText('\n=>NOT CONVERGED: {} \n'.format(sys.exc_info()[0]))
210         self.printText('\n=>TRY RUNNING AGAIN OR ADD/REMOVE ELV SYSTEMS !!!')
211         return
212     self.printText('\n=>CONVERGED: {} \n\nSolution: \nAij= {}'.format(errmsg))
213     for i in range(NComp):
214         for j in range(NComp): self.printText('{:^10}'.format('{:7.2f}'.format(a[i,j])))
215         if i < NComp-1: self.printText(']\n\n')
216     self.printText(']\n\n')
217     if type(cov_theta_optimum) == float:
218         self.printText('\n=>COVARIANCE MATRIX IS INFINITE! \n')
219         return
220     PExp_optimum = f(x,*theta_optimum)
221     # Total Sum of Squares (TSS)
222     TSS_corrected = ((PExp-PExp.mean())**2).sum(); TSS_uncorrected = (PExp**2).sum()
223     # Residual Sum of Squares (RSS) or Sum of Squared Residuals (SSR) or
224     # Sum of Squared Errors of Prediction (SSE)
225     RSS = ((PExp-PExp_optimum)**2).sum()
226     # Explained Sum of Squares (ESS)
227     ESS_uncorrected = TSS_uncorrected-RSS
228     # Mean Total Sum of Squares
229     MST_corrected = TSS_corrected/(N-1)
230     # Mean Residual Sum of Squares
231     MSR = RSS/(N-P)
232     # Mean Explained Sum of Squares
233     MSE_uncorrected = ESS_uncorrected/P
234     alpha = 0.05
235     F_critical_uncorrected = stats.f.isf(alpha,N-P,P)
236     F_value_uncorrected = MSE_uncorrected/MSR
237     Pr_F_value_uncorrected = stats.f.sf(F_value_uncorrected,N-P,P)
238     s = sqrt(MSR)
239     R2 = 1-RSS/TSS_corrected; R2_bar = 1-MSR/MST_corrected
240     ASE = sqrt(cov_theta_optimum.diagonal())
241     corr_theta_optimum = (cov_theta_optimum/ASE).T/ASE
242     t_critical = stats.t.isf(alpha/2,N-P)
243     t_value = theta_optimum/ASE
244     Pr_tvalue = stats.t.sf(abs(t_value),N-P)*2
245     CI_theta = plus_minus(theta_optimum,t_critical*ASE)
246     self.printText('\nNon-Linear Least Squares Summary Statistics \n\n')
247     self.printText('Function: {} \n\n'.format('UNIQUAC'))
248     self.printText('Source {} \n\n'.format(''))
249     self.printText('Regression {} \n\n'.format(''))
250     self.printText('Residual {} \n\n'.format(''))
251     self.printText('Uncorrected Total {} \n\n'.format(''))
252     self.printText('Corrected Total {} \n\n'.format(''))
253     self.printText('alpha {} \n\n'.format(''))
254     self.printText('s {} \n\n'.format(''))
255     self.printText('Parameter Estimate {} \n\n'.format(''))
256     self.printText('Asymptotic {} \n\n'.format(''))
257     self.printText('Std. Error {} \n\n'.format(''))
258     self.printText('Confidence Interval {} \n\n'.format(''))
259     self.printText('Lower {} \n\n'.format(''))
260     self.printText('Upper {} \n\n'.format(''))
261     for i in range(P):
262         self.printText('{:0:10s} |{:1:16.6f}|{:2:20.6f}|{:3:12.6f}|{:4:8.6f}|{:5:23.6f}|{:6:23.6f}\n'.format('theta[{:0}]'.format(i),theta_optimum[i],ASE[i],t_value[i],Pr_tvalue[i],CI_theta[0][i],CI_theta[1][i]))
263     for i in range(P):
264         self.printText('{:0:11s}'.format('theta[{:0}]'.format(i)))
265     for i in range(P):
266         self.printText('\n{:0:10s}'.format('theta[{:0}]'.format(i)))
267     for j in range(P):

```

```

268         self.printText(' |_{0:9.6f}'.format(corr_theta_optimum[i][j]))
269     Gamma = ThermodynamicModels.UNIQUAC(r,q,a,x,TExp)
270     PCalc = array(map(sum,x*Gamma*PSat))
271     dP = abs(PExp-PCalc)
272     yCalc = ((x*Gamma*PSat).T/array(map(sum,x*ThermodynamicModels.UNIQUAC(r,q,a,x,TExp)*PSat))).T
273     dy = abs(y-yCalc)
274     sum_dyRel, yNull = [],[]
275     for i in range(NComp-1): sum_dyRel.append(0.); yNull.append(0)
276     for i in range(N):
277         for j in range(NComp-1):
278             if y[i,j] == 0.: yNull[j] += 1
279             else: sum_dyRel[j] += dy[i,j]/y[i,j]
280     for i in range(NComp-1):
281         self.printText('\n\nMaximum deviation of y_{:8.6f}\n'.format(i+1,max(dy[:,i])))
282         self.printText('Average absolute deviation of y_{:8.6f}\n'.format(i+1,sum(dy[:,i])/N))
283         self.printText('Average relative deviation of y_{:5.2%}\n\nReference_{:5.2%}\n'.format(i+1,yNull[i], sum_dyRel[i]/N))
284     self.printText('\n\nMaximum deviation of Pressure_{:7.2f}\n'.format(max(dP)))
285     self.printText('Average absolute deviation of Pressure_{:7.2f}\n'.format(sum(dP)/N))
286     self.printText('Average relative deviation of Pressure_{:5.2%}\n\nReference_{:5.2%}\n'.format(sum(dP/PExp)/N))
287     for i in range(NComp-1):
288         self.printText(' |_{:8.6f}'.format(i+1))
289     for i in range(NComp-1):
290         self.printText(' |_{:8.6f}Exp_{:8.6f}Calc_{:8.6f}abs(dy_{:8.6f})\n'.format(i+1))
291     self.printText(' |_{:8.6f}PExp_{:8.6f}PCalc_{:8.6f}abs(dP)_{:8.6f}(mmHg)_{:8.6f}TE_{:8.6f}(C)\n')
292     for i in range(N):
293         self.printText(' |_{:8.6f}'.format(Refs[i]))
294         for j in range(NComp-1): self.printText(' |_{:8.6f}'.format(x[i,j]))
295         for j in range(NComp-1):
296             self.printText(' |_{:8.6f}y_{:8.6f}yCalc_{:8.6f}dy_{:8.6f}\n')
297             self.printText(' |_{:8.6f}y_{:8.6f}yCalc_{:8.6f}dy_{:8.6f}\n')
298             self.printText(' |_{:8.6f}y_{:8.6f}yCalc_{:8.6f}dy_{:8.6f}\n')
299             self.printText(' |_{:8.6f}y_{:8.6f}yCalc_{:8.6f}dy_{:8.6f}\n')
300     self.printText('\n\nGamma_{:8.6f}\n')
301     for i in range(NComp-1):
302         self.printText('Gamma_{:8.6f}\n')
303     for i in range(NComp):
304         self.printText('PSat_{:8.6f}\n')
305     self.printText('PE_{:8.6f}T_{:8.6f}\n')
306     for i in range(N):
307         for j in range(NComp):
308             self.printText(' |_{:8.6f}\n'.format(Gamma[i,j]))
309         for j in range(NComp):
310             self.printText(' |_{:8.6f}\n'.format(PSat[i,j]))
311         self.printText(' |_{:8.6f}PE_{:8.6f}T_{:8.6f}\n'.format(PE[i],T[i]-173.15))
312     return
313
314
315 def main():
316     app = QApplication(sys.argv)
317     _ = Form()
318     sys.exit(app.exec_())
319
320
321 if __name__ == '__main__':
322     main()

```


APÊNDICE C – PROGRAMA COMPUTACIONAL JAFOSMS

C.1 Interface gráfica do programa JAFOSMS

Form2.ui

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <ui version="4.0">
3    <class>Form</class>
4    <widget class="QWidget" name="Form">
5      <property name="geometry">
6        <rect>
7          <x>0</x>
8          <y>0</y>
9          <width>752</width>
10         <height>485</height>
11        </rect>
12      </property>
13      <property name="sizePolicy">
14        <sizepolicy hsize="Minimum" vsize="Minimum">
15          <horstretch>0</horstretch>
16          <verstretch>0</verstretch>
17        </sizepolicy>
18      </property>
19      <property name="windowTitle">
20        <string>JAFO Salt Solubilities in Mixed Solvents (JAFOSMS)</string>
21      </property>
22      <layout class="QVBoxLayout" name="verticalLayout_6">
23        <item>
24          <layout class="QHBoxLayout" name="horizontalLayout_Datafile">
25            <item>
26              <widget class="QLabel" name="label_Datafile">
27                <property name="font">
28                  <font>
29                    <family>Ubuntu</family>
30                    <pointsize>11</pointsize>
31                  </font>
32                </property>
33                <property name="text">
34                  <string>& Datafile:</string>
35                </property>
36                <property name="buddy">
37                  <cstring>lineEdit_Datafile</cstring>
38                </property>
39              </widget>
40            </item>
41            <item>
42              <widget class="QLineEdit" name="lineEdit_Datafile">
43                <property name="font">
44                  <font>
45                    <family>Ubuntu</family>
46                    <pointsize>11</pointsize>
47                  </font>
48                </property>
49              </widget>
50            </item>
51            <item>
52              <widget class="QPushButton" name="pushButton_Datafile_Browse">
53                <property name="font">
54                  <font>
55                    <family>Ubuntu</family>
56                    <pointsize>11</pointsize>
57                  </font>
58                </property>
59                <property name="text">
60                  <string>& Browse</string>
61                </property>

```

```

62     </widget>
63 </item>
64 </layout>
65 </item>
66 <item>
67 <widget class="QTabWidget" name="tabWidget_1">
68 <property name="enabled">
69 <bool>>false</bool>
70 </property>
71 <property name="font">
72 <font>
73 <family>Ubuntu</family>
74 </font>
75 </property>
76 <property name="currentIndex">
77 <number>0</number>
78 </property>
79 <widget class="QWidget" name="tab_GeneralReport">
80 <attribute name="title">
81 <string>GENERAL &REPORT</string>
82 </attribute>
83 <layout class="QVBoxLayout" name="verticalLayout_4">
84 <item>
85 <widget class="QTextEdit" name="textEdit_GeneralReport"/>
86 </item>
87 <item>
88 <layout class="QHBoxLayout" name="horizontalLayout_Save_Print_GeneralReport">
89 <item>
90 <widget class="QPushButton" name="pushButton_SaveGeneralReport">
91 <property name="text">
92 <string>Sa&ve General Report</string>
93 </property>
94 </widget>
95 </item>
96 <item>
97 <widget class="QPushButton" name="pushButton_PrintGeneralReport">
98 <property name="text">
99 <string>&Print General Report</string>
100 </property>
101 </widget>
102 </item>
103 </layout>
104 </item>
105 </layout>
106 </widget>
107 <widget class="QWidget" name="tab_SingleSolvent">
108 <attribute name="title">
109 <string>&SINGLE SOLVENT</string>
110 </attribute>
111 <layout class="QVBoxLayout" name="verticalLayout_2">
112 <item>
113 <layout class="QGridLayout" name="gridLayout_Plot1">
114 <item row="0" column="1">
115 <layout class="QHBoxLayout" name="horizontalLayout_Plot1_Ordinate">
116 <item>
117 <widget class="QLabel" name="label_Plot1_Ordinate_1">
118 <property name="font">
119 <font>
120 <weight>75</weight>
121 <bold>true</bold>
122 </font>
123 </property>
124 <property name="text">
125 <string>Ordinate: </string>
126 </property>
127 </widget>
128 </item>
129 <item>
130 <widget class="QComboBox" name="comboBox_Plot1_Ordinate_Concentration">
131 <property name="font">
132 <font>
133 <pointsize>11</pointsize>
134 </font>
135 </property>
136 <item>
137 <property name="text">
138 <string>Molality</string>
139 </property>
140 </item>
141 <item>

```

```

142         <property name="text">
143             <string>Molarity</string>
144         </property>
145     </item>
146     <item>
147         <property name="text">
148             <string>Molar Fraction</string>
149         </property>
150     </item>
151     <item>
152         <property name="text">
153             <string>Mass Fraction</string>
154         </property>
155     </item>
156 </widget>
157 </item>
158 <item>
159     <widget class="QLabel" name="label_Plot1_Ordinate_2">
160         <property name="font">
161             <font>
162                 <pointsize>11</pointsize>
163             </font>
164         </property>
165         <property name="text">
166             <string>of Salt in</string>
167         </property>
168     </widget>
169 </item>
170 <item>
171     <widget class="QComboBox" name="comboBox_Plot1_Ordinate_Solvent">
172         <property name="font">
173             <font>
174                 <pointsize>11</pointsize>
175             </font>
176         </property>
177     <item>
178         <property name="text">
179             <string>Water</string>
180         </property>
181     </item>
182     <item>
183         <property name="text">
184             <string>MEK</string>
185         </property>
186     </item>
187 </widget>
188 </item>
189 <item>
190     <spacer name="horizontalSpacer_Plot1_Ordinate">
191         <property name="font">
192             <font>
193                 <pointsize>11</pointsize>
194             </font>
195         </property>
196         <property name="orientation">
197             <enum>Qt::Horizontal</enum>
198         </property>
199         <property name="sizeHint" stdset="0">
200             <size>
201                 <width>40</width>
202                 <height>20</height>
203             </size>
204         </property>
205     </spacer>
206 </item>
207 </layout>
208 </item>
209 <item row="0" column="0" rowspan="2">
210     <widget class="QPushButton" name="pushButton_Plot1">
211         <property name="sizePolicy">
212             <sizepolicy hsize="Minimum" vsize="Expanding">
213                 <horstretch>0</horstretch>
214                 <verstretch>0</verstretch>
215             </sizepolicy>
216         </property>
217         <property name="font">
218             <font>
219                 <pointsize>11</pointsize>
220                 <weight>75</weight>
221                 <bold>true</bold>

```

```

222         </font>
223     </property>
224     <property name="text">
225         <string>Plot &1</string>
226     </property>
227 </widget>
228 </item>
229 <item row="1" column="1">
230     <layout class="QHBoxLayout" name="horizontalLayout_Plot1_Abscissa">
231         <item>
232             <widget class="QLabel" name="label_Plot1_Abscissa_1">
233                 <property name="font">
234                     <font>
235                         <weight>75</weight>
236                         <bold>true</bold>
237                     </font>
238                 </property>
239                 <property name="text">
240                     <string>Abscissa:</string>
241                 </property>
242             </widget>
243         </item>
244         <item>
245             <widget class="QLabel" name="label_Plot1_Abscissa_2">
246                 <property name="font">
247                     <font>
248                         <pointsize>11</pointsize>
249                     </font>
250                 </property>
251                 <property name="text">
252                     <string>Temperature in</string>
253                 </property>
254             </widget>
255         </item>
256         <item>
257             <widget class="QComboBox" name="comboBox_Plot1_Abscissa_Temperature">
258                 <property name="font">
259                     <font>
260                         <pointsize>11</pointsize>
261                     </font>
262                 </property>
263                 <item>
264                     <property name="text">
265                         <string>Kelvins</string>
266                     </property>
267                 </item>
268                 <item>
269                     <property name="text">
270                         <string>Degrees Celsius</string>
271                     </property>
272                 </item>
273                 <item>
274                     <property name="text">
275                         <string>Degrees Rankine</string>
276                     </property>
277                 </item>
278                 <item>
279                     <property name="text">
280                         <string>Degrees Fahrenheit</string>
281                     </property>
282                 </item>
283             </widget>
284         </item>
285         <item>
286             <spacer name="horizontalSpacer_Plot1_Abscissa">
287                 <property name="orientation">
288                     <enum>Qt::Horizontal</enum>
289                 </property>
290                 <property name="sizeHint" stdset="0">
291                     <size>
292                         <width>40</width>
293                         <height>20</height>
294                     </size>
295                 </property>
296             </spacer>
297         </item>
298     </layout>
299 </item>
300 </layout>
301 </item>

```

```

302     <item>
303         <spacer name="verticalSpacer_SingleSolvent">
304             <property name="orientation">
305                 <enum>Qt::Vertical</enum>
306             </property>
307             <property name="sizeHint" stdset="0">
308                 <size>
309                     <width>20</width>
310                     <height>40</height>
311                 </size>
312             </property>
313         </spacer>
314     </item>
315 </layout>
316 </widget>
317 <widget class="QWidget" name="tab_MixedSolvents">
318     <attribute name="title">
319         <string>&MIXED SOLVENTS</string>
320     </attribute>
321     <layout class="QVBoxLayout" name="verticalLayout">
322         <item>
323             <layout class="QHBoxLayout" name="horizontalLayout_Isotherms">
324                 <item>
325                     <widget class="QLabel" name="label_Isotherms_1">
326                         <property name="font">
327                             <font>
328                                 <family>Ubuntu</family>
329                                 <pointsize>11</pointsize>
330                                 <italic>true</italic>
331                             </font>
332                         </property>
333                         <property name="text">
334                             <string>&Isotherms for</string>
335                         </property>
336                         <property name="buddy">
337                             <cstring>lineEdit_Isotherms</cstring>
338                         </property>
339                     </widget>
340                 </item>
341                 <item>
342                     <widget class="QComboBox" name="comboBox_Isotherms_Mode">
343                         <property name="font">
344                             <font>
345                                 <family>Ubuntu</family>
346                                 <italic>true</italic>
347                             </font>
348                         </property>
349                         <item>
350                             <property name="text">
351                                 <string>Regression</string>
352                             </property>
353                         </item>
354                         <item>
355                             <property name="text">
356                                 <string>Prediction</string>
357                             </property>
358                         </item>
359                     </widget>
360                 </item>
361                 <item>
362                     <widget class="QLabel" name="label_Isotherms_2">
363                         <property name="font">
364                             <font>
365                                 <family>Ubuntu</family>
366                             </font>
367                         </property>
368                         <property name="text">
369                             <string>in</string>
370                         </property>
371                     </widget>
372                 </item>
373                 <item>
374                     <widget class="QComboBox" name="comboBox_Isotherms_Temperature">
375                         <property name="font">
376                             <font>
377                                 <family>Ubuntu</family>
378                                 <pointsize>11</pointsize>
379                             </font>
380                         </property>
381                     </item>

```

```

382     <property name="text">
383     <string>Kelvins</string>
384     </property>
385 </item>
386 <item>
387     <property name="text">
388     <string>Degrees Celsius</string>
389     </property>
390 </item>
391 <item>
392     <property name="text">
393     <string>Degrees Rankine</string>
394     </property>
395 </item>
396 <item>
397     <property name="text">
398     <string>Degrees Fahrenheit</string>
399     </property>
400 </item>
401 </widget>
402 </item>
403 <item>
404     <widget class="QLineEdit" name="lineEdit_Isotherms">
405     <property name="font">
406     <font>
407     <family>Ubuntu</family>
408     <pointsize>11</pointsize>
409     </font>
410     </property>
411 </widget>
412 </item>
413 <item>
414     <widget class="QPushButton" name="pushButton_Isotherms_Choose">
415     <property name="font">
416     <font>
417     <family>Ubuntu</family>
418     <pointsize>11</pointsize>
419     </font>
420     </property>
421     <property name="text">
422     <string>& Choose</string>
423     </property>
424 </widget>
425 </item>
426 </layout>
427 </item>
428 <item>
429     <widget class="QGroupBox" name="groupBox_Plots">
430     <property name="enabled">
431     <bool>false</bool>
432     </property>
433     <property name="font">
434     <font>
435     <family>Ubuntu</family>
436     </font>
437     </property>
438     <property name="title">
439     <string>Plots:</string>
440     </property>
441     <layout class="QVBoxLayout" name="verticalLayout_3">
442     <item>
443     <layout class="QGridLayout" name="gridLayout_Plot2">
444     <item row="0" column="0" rowspan="2">
445     <widget class="QPushButton" name="pushButton_Plot2">
446     <property name="sizePolicy">
447     <sizepolicy hsize="Minimum" vsize="Expanding">
448     <horstretch>0</horstretch>
449     <verstretch>0</verstretch>
450     </sizepolicy>
451     </property>
452     <property name="font">
453     <font>
454     <family>Ubuntu</family>
455     <pointsize>11</pointsize>
456     <weight>75</weight>
457     <bold>true</bold>
458     </font>
459     </property>
460     <property name="text">
461     <string>Plot &2</string>

```

```

462     </property>
463 </widget>
464 </item>
465 <item row="1" column="1">
466     <layout class="QHBoxLayout" name="horizontalLayout_Plot2_Abscissa">
467         <item>
468             <widget class="QLabel" name="label_Plot2_Abscissa_1">
469                 <property name="font">
470                     <font>
471                         <family>Ubuntu</family>
472                         <weight>75</weight>
473                         <bold>true</bold>
474                     </font>
475                 </property>
476                 <property name="text">
477                     <string>Abscissa:</string>
478                 </property>
479             </widget>
480         </item>
481         <item>
482             <widget class="QLabel" name="label_Plot2_Abscissa_2">
483                 <property name="font">
484                     <font>
485                         <family>Ubuntu</family>
486                         <pointsize>11</pointsize>
487                     </font>
488                 </property>
489                 <property name="text">
490                     <string>Salt-free</string>
491                 </property>
492             </widget>
493         </item>
494         <item>
495             <widget class="QComboBox" name="comboBox_Plot2_Abscissa_Concentration">
496                 <property name="font">
497                     <font>
498                         <family>Ubuntu</family>
499                         <pointsize>11</pointsize>
500                     </font>
501                 </property>
502                 <item>
503                     <property name="text">
504                         <string>Molar</string>
505                     </property>
506                 </item>
507                 <item>
508                     <property name="text">
509                         <string>Massic</string>
510                     </property>
511                 </item>
512             </widget>
513         </item>
514         <item>
515             <widget class="QLabel" name="label_Plot2_Abscissa_3">
516                 <property name="font">
517                     <font>
518                         <family>Ubuntu</family>
519                         <pointsize>11</pointsize>
520                     </font>
521                 </property>
522                 <property name="text">
523                     <string>Fraction of</string>
524                 </property>
525             </widget>
526         </item>
527         <item>
528             <widget class="QComboBox" name="comboBox_Plot2_Abscissa_Solvent">
529                 <property name="font">
530                     <font>
531                         <family>Ubuntu</family>
532                         <pointsize>11</pointsize>
533                     </font>
534                 </property>
535                 <item>
536                     <property name="text">
537                         <string>MEG</string>
538                     </property>
539                 </item>
540             <item>
541                 <property name="text">

```

```

542         <string>Water</string>
543     </property>
544 </item>
545 </widget>
546 </item>
547 <item>
548     <spacer name="horizontalSpacer_Plot2_Abscissa">
549         <property name="font">
550             <font>
551                 <family>Ubuntu</family>
552             </font>
553         </property>
554         <property name="orientation">
555             <enum>Qt::Horizontal</enum>
556         </property>
557         <property name="sizeHint" stdset="0">
558             <size>
559                 <width>40</width>
560                 <height>20</height>
561             </size>
562         </property>
563     </spacer>
564 </item>
565 </layout>
566 </item>
567 <item row="0" column="1">
568     <layout class="QHBoxLayout" name="horizontalLayout_Plot2_Ordinate">
569         <item>
570             <widget class="QLabel" name="label_Plot2_Ordinate_1">
571                 <property name="font">
572                     <font>
573                         <family>Ubuntu</family>
574                         <weight>75</weight>
575                         <bold>true</bold>
576                     </font>
577                 </property>
578                 <property name="text">
579                     <string>Ordinate: </string>
580                 </property>
581             </widget>
582         </item>
583         <item>
584             <widget class="QComboBox" name="comboBox_Plot2_Ordinate_Concentration">
585                 <property name="font">
586                     <font>
587                         <family>Ubuntu</family>
588                         <pointsize>11</pointsize>
589                     </font>
590                 </property>
591                 <item>
592                     <property name="text">
593                         <string>Molality</string>
594                     </property>
595                 </item>
596                 <item>
597                     <property name="text">
598                         <string>Molarity</string>
599                     </property>
600                 </item>
601                 <item>
602                     <property name="text">
603                         <string>Molar Fraction</string>
604                     </property>
605                 </item>
606                 <item>
607                     <property name="text">
608                         <string>Mass Fraction</string>
609                     </property>
610                 </item>
611             </widget>
612         </item>
613         <item>
614             <widget class="QLabel" name="label_Plot2_Ordinate_2">
615                 <property name="font">
616                     <font>
617                         <family>Ubuntu</family>
618                         <pointsize>11</pointsize>
619                     </font>
620                 </property>
621                 <property name="text">

```



```

622         <string>of Salt in Mixed Solvents</string>
623     </property>
624 </widget>
625 </item>
626 <item>
627     <spacer name="horizontalSpacer_Plot2_Ordinate">
628         <property name="font">
629             <font>
630                 <family>Ubuntu</family>
631             </font>
632         </property>
633         <property name="orientation">
634             <enum>Qt::Horizontal</enum>
635         </property>
636         <property name="sizeHint" stdset="0">
637             <size>
638                 <width>40</width>
639                 <height>20</height>
640             </size>
641         </property>
642     </spacer>
643 </item>
644 </layout>
645 </item>
646 </layout>
647 </item>
648 <item>
649     <widget class="Line" name="line_Plots_1">
650         <property name="font">
651             <font>
652                 <family>Ubuntu</family>
653             </font>
654         </property>
655         <property name="orientation">
656             <enum>Qt::Horizontal</enum>
657         </property>
658     </widget>
659 </item>
660 <item>
661     <layout class="QGridLayout" name="gridLayout_Plot3">
662         <item row="0" column="1">
663             <layout class="QHBoxLayout" name="horizontalLayout_Plot3_Ordinate">
664                 <item>
665                     <widget class="QLabel" name="label_Plot3_Ordinate_1">
666                         <property name="font">
667                             <font>
668                                 <family>Ubuntu</family>
669                                 <weight>75</weight>
670                                 <bold>true</bold>
671                             </font>
672                         </property>
673                         <property name="text">
674                             <string>Ordinate: </string>
675                         </property>
676                     </widget>
677                 </item>
678                 <item>
679                     <widget class="QComboBox" name="comboBox_Plot3_Ordinate_Function">
680                         <property name="font">
681                             <font>
682                                 <family>Ubuntu</family>
683                                 <pointsize>11</pointsize>
684                             </font>
685                         </property>
686                     </item>
687                     <property name="text">
688                         <string>Activity Coefficient</string>
689                     </property>
690                 </item>
691                 <item>
692                     <property name="text">
693                         <string>Logarithmic Activity Coefficient</string>
694                     </property>
695                 </item>
696             </widget>
697         </item>
698         <item>
699             <widget class="QLabel" name="label_Plot3_Ordinate_2">
700                 <property name="font">
701                     <font>

```

```

702         <family>Ubuntu</family>
703         <pointsizes>11</pointsizes>
704     </font>
705 </property>
706 <property name="text">
707     <string>of Salt in Mixed Solvents</string>
708 </property>
709 </widget>
710 </item>
711 <item>
712     <spacer name="horizontalSpacer_Plot3_Ordinate">
713         <property name="font">
714             <font>
715                 <family>Ubuntu</family>
716                 <pointsizes>11</pointsizes>
717             </font>
718         </property>
719         <property name="orientation">
720             <enum>Qt::Horizontal</enum>
721         </property>
722         <property name="sizeHint" stdset="0">
723             <size>
724                 <width>40</width>
725                 <height>20</height>
726             </size>
727         </property>
728     </spacer>
729 </item>
730 </layout>
731 </item>
732 <item row="0" column="0" rowspan="2">
733     <widget class="QPushButton" name="pushButton_Plot3">
734         <property name="sizePolicy">
735             <sizepolicy hsizepolicy="Minimum" vsizetype="Expanding">
736                 <horstretch>0</horstretch>
737                 <verstretch>0</verstretch>
738             </sizepolicy>
739         </property>
740         <property name="font">
741             <font>
742                 <family>Ubuntu</family>
743                 <pointsizes>11</pointsizes>
744                 <weight>75</weight>
745                 <bold>true</bold>
746             </font>
747         </property>
748         <property name="text">
749             <string>Plot &3</string>
750         </property>
751     </widget>
752 </item>
753 <item row="1" column="1">
754     <layout class="QHBoxLayout" name="horizontalLayout_Plot3_Abscissa">
755         <item>
756             <widget class="QLabel" name="label_Plot3_Abscissa_1">
757                 <property name="font">
758                     <font>
759                         <family>Ubuntu</family>
760                         <weight>75</weight>
761                         <bold>true</bold>
762                     </font>
763                 </property>
764                 <property name="text">
765                     <string>Abscissa:</string>
766                 </property>
767             </widget>
768         </item>
769         <item>
770             <widget class="QLabel" name="label_Plot3_Abscissa_2">
771                 <property name="font">
772                     <font>
773                         <family>Ubuntu</family>
774                         <pointsizes>11</pointsizes>
775                     </font>
776                 </property>
777                 <property name="text">
778                     <string>Salt-free</string>
779                 </property>
780             </widget>
781         </item>

```

```

782     <item>
783     <widget class="QComboBox" name="comboBox_Plot3_Abscissa_Concentration">
784     <property name="font">
785     <font>
786     <family>Ubuntu</family>
787     <pointsize>11</pointsize>
788     </font>
789     </property>
790     <item>
791     <property name="text">
792     <string>Molar</string>
793     </property>
794     </item>
795     <item>
796     <property name="text">
797     <string>Massic</string>
798     </property>
799     </item>
800     </widget>
801 </item>
802 <item>
803 <widget class="QLabel" name="label_Plot3_Abscissa_3">
804 <property name="font">
805 <font>
806 <family>Ubuntu</family>
807 <pointsize>11</pointsize>
808 </font>
809 </property>
810 <property name="text">
811 <string>Fraction of</string>
812 </property>
813 </widget>
814 </item>
815 <item>
816 <widget class="QComboBox" name="comboBox_Plot3_Abscissa_Solvent">
817 <property name="font">
818 <font>
819 <family>Ubuntu</family>
820 <pointsize>11</pointsize>
821 </font>
822 </property>
823 <item>
824 <property name="text">
825 <string>MEG</string>
826 </property>
827 </item>
828 <item>
829 <property name="text">
830 <string>Water</string>
831 </property>
832 </item>
833 </widget>
834 </item>
835 <item>
836 <spacer name="horizontalSpacer_Plot3_Abscissa">
837 <property name="font">
838 <font>
839 <family>Ubuntu</family>
840 </font>
841 </property>
842 <property name="orientation">
843 <enum>Qt::Horizontal</enum>
844 </property>
845 <property name="sizeHint" stdset="0">
846 <size>
847 <width>40</width>
848 <height>20</height>
849 </size>
850 </property>
851 </spacer>
852 </item>
853 </layout>
854 </item>
855 </layout>
856 </item>
857 <item>
858 <widget class="Line" name="line_Plots_2">
859 <property name="font">
860 <font>
861 <family>Ubuntu</family>

```

```

862     </font>
863     </property>
864     <property name="orientation">
865         <enum>Qt::Horizontal</enum>
866     </property>
867 </widget>
868 </item>
869 <item>
870     <layout class="QGridLayout" name="gridLayout_Plot4">
871         <item row="0" column="1">
872             <layout class="QHBoxLayout" name="horizontalLayout_Plot4_Ordinate">
873                 <item>
874                     <widget class="QLabel" name="label_Plot4_Ordinate_1">
875                         <property name="font">
876                             <font>
877                                 <family>Ubuntu</family>
878                                 <weight>75</weight>
879                                 <bold>true</bold>
880                             </font>
881                         </property>
882                         <property name="text">
883                             <string>Ordinate: </string>
884                         </property>
885                     </widget>
886                 </item>
887                 <item>
888                     <widget class="QComboBox" name="comboBox_Plot4_Ordinate_ExcessFunction">
889                         <property name="font">
890                             <font>
891                                 <family>Ubuntu</family>
892                                 <pointsize>11</pointsize>
893                             </font>
894                         </property>
895                         <item>
896                             <property name="text">
897                                 <string>Excess Standard Chemical Potential</string>
898                             </property>
899                         </item>
900                         <item>
901                             <property name="text">
902                                 <string>Excess Logarithmic Activity Coefficient</string>
903                             </property>
904                         </item>
905                         <item>
906                             <property name="text">
907                                 <string>Excess Logarithmic Molality</string>
908                             </property>
909                         </item>
910                     </widget>
911                 </item>
912                 <item>
913                     <widget class="QLabel" name="label_Plot4_Ordinate_2">
914                         <property name="font">
915                             <font>
916                                 <family>Ubuntu</family>
917                                 <pointsize>11</pointsize>
918                             </font>
919                         </property>
920                         <property name="text">
921                             <string>of Salt in Mixed Solvents</string>
922                         </property>
923                     </widget>
924                 </item>
925                 <item>
926                     <spacer name="horizontalSpacer_Plot4_Ordinate">
927                         <property name="font">
928                             <font>
929                                 <family>Ubuntu</family>
930                                 <pointsize>11</pointsize>
931                             </font>
932                         </property>
933                         <property name="orientation">
934                             <enum>Qt::Horizontal</enum>
935                         </property>
936                         <property name="sizeHint" stdset="0">
937                             <size>
938                                 <width>40</width>
939                                 <height>20</height>
940                             </size>
941                         </property>

```

```

942         </spacer>
943     </item>
944 </layout>
945 </item>
946 <item row="0" column="0" rowspan="2">
947     <widget class="QPushButton" name="pushButton_Plot4">
948         <property name="sizePolicy">
949             <sizepolicy hsize="Minimum" vsize="Expanding">
950                 <horstretch>0</horstretch>
951                 <verstretch>0</verstretch>
952             </sizepolicy>
953         </property>
954         <property name="font">
955             <font>
956                 <family>Ubuntu</family>
957                 <pointsize>11</pointsize>
958                 <weight>75</weight>
959                 <bold>true</bold>
960             </font>
961         </property>
962         <property name="text">
963             <string>Plot &4</string>
964         </property>
965     </widget>
966 </item>
967 <item row="1" column="1">
968     <layout class="QHBoxLayout" name="horizontalLayout_Plot4_Abscissa">
969         <item>
970             <widget class="QLabel" name="label_Plot4_Abscissa_1">
971                 <property name="font">
972                     <font>
973                         <family>Ubuntu</family>
974                         <weight>75</weight>
975                         <bold>true</bold>
976                     </font>
977                 </property>
978                 <property name="text">
979                     <string>Abscissa:</string>
980                 </property>
981             </widget>
982         </item>
983         <item>
984             <widget class="QLabel" name="label_Plot4_Abscissa_2">
985                 <property name="font">
986                     <font>
987                         <family>Ubuntu</family>
988                         <pointsize>11</pointsize>
989                     </font>
990                 </property>
991                 <property name="text">
992                     <string>Salt-free</string>
993                 </property>
994             </widget>
995         </item>
996         <item>
997             <widget class="QComboBox" name="comboBox_Plot4_Abscissa_Concentration">
998                 <property name="font">
999                     <font>
1000                         <family>Ubuntu</family>
1001                         <pointsize>11</pointsize>
1002                     </font>
1003                 </property>
1004                 <item>
1005                     <property name="text">
1006                         <string>Molar</string>
1007                     </property>
1008                 </item>
1009                 <item>
1010                     <property name="text">
1011                         <string>Massic</string>
1012                     </property>
1013                 </item>
1014             </widget>
1015         </item>
1016         <item>
1017             <widget class="QLabel" name="label_Plot4_Abscissa_3">
1018                 <property name="font">
1019                     <font>
1020                         <family>Ubuntu</family>
1021                         <pointsize>11</pointsize>

```

```

1022         </font>
1023     </property>
1024     <property name="text">
1025         <string>Fraction of</string>
1026     </property>
1027 </widget>
1028 </item>
1029 <item>
1030     <widget class="QComboBox" name="comboBox_Plot4_Abscissa_Solvent">
1031         <property name="font">
1032             <font>
1033                 <family>Ubuntu</family>
1034                 <pointsize>11</pointsize>
1035             </font>
1036         </property>
1037         <item>
1038             <property name="text">
1039                 <string>MEC</string>
1040             </property>
1041         </item>
1042         <item>
1043             <property name="text">
1044                 <string>Water</string>
1045             </property>
1046         </item>
1047     </widget>
1048 </item>
1049 <item>
1050     <spacer name="horizontalSpacer_Plot4_Abscissa">
1051         <property name="font">
1052             <font>
1053                 <family>Ubuntu</family>
1054             </font>
1055         </property>
1056         <property name="orientation">
1057             <enum>Qt::Horizontal</enum>
1058         </property>
1059         <property name="sizeHint" stdset="0">
1060             <size>
1061                 <width>40</width>
1062                 <height>20</height>
1063             </size>
1064         </property>
1065     </spacer>
1066 </item>
1067 </layout>
1068 </item>
1069 </layout>
1070 </item>
1071 </layout>
1072 </widget>
1073 </item>
1074 <item>
1075     <spacer name="verticalSpacer_MixedSolvents">
1076         <property name="font">
1077             <font>
1078                 <family>Ubuntu</family>
1079             </font>
1080         </property>
1081         <property name="orientation">
1082             <enum>Qt::Vertical</enum>
1083         </property>
1084         <property name="sizeHint" stdset="0">
1085             <size>
1086                 <width>20</width>
1087                 <height>40</height>
1088             </size>
1089         </property>
1090     </spacer>
1091 </item>
1092 </layout>
1093 </widget>
1094 <widget class="QWidget" name="tab_GraphicAndStatistics">
1095     <attribute name="title">
1096         <string>&GRAPHIC AND STATISTICS</string>
1097     </attribute>
1098     <layout class="QVBoxLayout" name="verticalLayout_8">
1099         <item>
1100             <widget class="QSplitter" name="splitter_GraphicAndStatistics">
1101                 <property name="orientation">

```

```

1102     <enum>Qt::Horizontal</enum>
1103 </property>
1104 <widget class="QWidget" name="layoutWidget">
1105     <layout class="QVBoxLayout" name="verticalLayout_Graphic">
1106         <item>
1107             <widget class="MplWidget" name="mplWidget_Graphic" native="true">
1108                 <property name="enabled">
1109                     <bool>>false</bool>
1110                 </property>
1111                 <property name="sizePolicy">
1112                     <sizepolicy hsize="Expanding" vsize="Expanding">
1113                         <horstretch>0</horstretch>
1114                         <verstretch>0</verstretch>
1115                     </sizepolicy>
1116                 </property>
1117             </widget>
1118         </item>
1119         <item>
1120             <layout class="QHBoxLayout" name="horizontalLayout_Graphic">
1121                 <item>
1122                     <widget class="QCheckBox" name="checkBox_Graphic">
1123                         <property name="enabled">
1124                             <bool>>false</bool>
1125                         </property>
1126                         <property name="text">
1127                             <string>Show Navigation Toolbar</string>
1128                         </property>
1129                         <property name="checked">
1130                             <bool>>true</bool>
1131                         </property>
1132                     </widget>
1133                 </item>
1134                 <item>
1135                     <widget class="QPushButton" name="pushButton_PrintGraphic">
1136                         <property name="enabled">
1137                             <bool>>false</bool>
1138                         </property>
1139                         <property name="text">
1140                             <string>Print Graphic</string>
1141                         </property>
1142                     </widget>
1143                 </item>
1144             </layout>
1145         </item>
1146     </layout>
1147 </widget>
1148 <widget class="QWidget" name="layoutWidget">
1149     <layout class="QVBoxLayout" name="verticalLayout_Statistic">
1150         <item>
1151             <widget class="QTextEdit" name="textEdit_Statistics">
1152                 <property name="enabled">
1153                     <bool>>false</bool>
1154                 </property>
1155             </widget>
1156         </item>
1157         <item>
1158             <layout class="QHBoxLayout" name="horizontalLayout_Save_Print_Statistics">
1159                 <item>
1160                     <widget class="QPushButton" name="pushButton_SaveStatistics">
1161                         <property name="enabled">
1162                             <bool>>false</bool>
1163                         </property>
1164                         <property name="text">
1165                             <string>Save Statistics</string>
1166                         </property>
1167                     </widget>
1168                 </item>
1169                 <item>
1170                     <widget class="QPushButton" name="pushButton_PrintStatistics">
1171                         <property name="enabled">
1172                             <bool>>false</bool>
1173                         </property>
1174                         <property name="text">
1175                             <string>Print Statistics</string>
1176                         </property>
1177                     </widget>
1178                 </item>
1179             </layout>
1180         </item>
1181     </layout>

```

```

1182     </widget>
1183 </widget>
1184 </item>
1185 </layout>
1186 </widget>
1187 </widget>
1188 </item>
1189 <item>
1190 <layout class="QHBoxLayout" name="horizontalLayout_Run_Exit">
1191 <item>
1192 <widget class="QPushButton" name="pushButton_Run">
1193 <property name="enabled">
1194 <bool>false</bool>
1195 </property>
1196 <property name="font">
1197 <font>
1198 <family>Ubuntu</family>
1199 <pointsize>11</pointsize>
1200 </font>
1201 </property>
1202 <property name="text">
1203 <string>Run</string>
1204 </property>
1205 </widget>
1206 </item>
1207 <item>
1208 <widget class="QPushButton" name="pushButton_Exit">
1209 <property name="font">
1210 <font>
1211 <family>Ubuntu</family>
1212 <pointsize>11</pointsize>
1213 </font>
1214 </property>
1215 <property name="text">
1216 <string>Exit</string>
1217 </property>
1218 </widget>
1219 </item>
1220 </layout>
1221 </item>
1222 </layout>
1223 </widget>
1224 <customwidgets>
1225 <customwidget>
1226 <class>MplWidget</class>
1227 <extends>QWidget</extends>
1228 <header>mplwidget.h</header>
1229 <container>1</container>
1230 </customwidget>
1231 </customwidgets>
1232 <tabstops>
1233 <tabstop>lineEdit_Datafile</tabstop>
1234 <tabstop>pushButton_Datafile_Browse</tabstop>
1235 <tabstop>tabWidget_1</tabstop>
1236 <tabstop>textEdit_GeneralReport</tabstop>
1237 <tabstop>pushButton_SaveGeneralReport</tabstop>
1238 <tabstop>pushButton_PrintGeneralReport</tabstop>
1239 <tabstop>pushButton_Plot1</tabstop>
1240 <tabstop>comboBox_Plot1_Ordinate_Concentration</tabstop>
1241 <tabstop>comboBox_Plot1_Ordinate_Solvent</tabstop>
1242 <tabstop>comboBox_Plot1_Abscissa_Temperature</tabstop>
1243 <tabstop>comboBox_Isotherms_Mode</tabstop>
1244 <tabstop>comboBox_Isotherms_Temperature</tabstop>
1245 <tabstop>lineEdit_Isotherms</tabstop>
1246 <tabstop>pushButton_Isotherms_Choose</tabstop>
1247 <tabstop>pushButton_Plot2</tabstop>
1248 <tabstop>comboBox_Plot2_Ordinate_Concentration</tabstop>
1249 <tabstop>comboBox_Plot2_Abscissa_Concentration</tabstop>
1250 <tabstop>comboBox_Plot2_Abscissa_Solvent</tabstop>
1251 <tabstop>pushButton_Plot3</tabstop>
1252 <tabstop>comboBox_Plot3_Ordinate_Function</tabstop>
1253 <tabstop>comboBox_Plot3_Abscissa_Concentration</tabstop>
1254 <tabstop>comboBox_Plot3_Abscissa_Solvent</tabstop>
1255 <tabstop>pushButton_Plot4</tabstop>
1256 <tabstop>comboBox_Plot4_Ordinate_ExcessFunction</tabstop>
1257 <tabstop>comboBox_Plot4_Abscissa_Concentration</tabstop>
1258 <tabstop>comboBox_Plot4_Abscissa_Solvent</tabstop>
1259 <tabstop>checkBox_Graphic</tabstop>
1260 <tabstop>pushButton_PrintGraphic</tabstop>
1261 <tabstop>textEdit_Statistics</tabstop>

```



```

1262 <tabstop>pushButton_SaveStatistics</tabstop>
1263 <tabstop>pushButton_PrintStatistics</tabstop>
1264 <tabstop>pushButton_Run</tabstop>
1265 <tabstop>pushButton_Exit</tabstop>
1266 </tabstops>
1267 <resources/>
1268 <connections/>
1269 </ui>

```

C.2 Código fonte do programa JAFOSSMS desenvolvido para o cálculo da solubilidade de sal em solução aquosa com MEG

JAFOSSMS.py

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  from __future__ import division
4  from PySide.QtGui import (QAbstractPrintDialog, QApplication, QCheckBox, QDialog,
5  QDialogButtonBox, QFileDialog, QFont, QPrinter, QPrintDialog, QGridLayout, QRegExpValidator,
6  QTextCharFormat, QTextDocument, QTextTableFormat)
7  from PySide.QtCore import (QFile, QFileInfo, QRegExp, Qt, QTextStream, SLOT)
8  from numpy import (abs, append, array, asarray, concatenate, exp, linspace, log, random, sqrt,
9  zeros_like)
10 from scipy.optimize import fsolve
11 import sys
12 import ui_Form
13
14 CHARS_MINUS, CHARS_EQUAL, CHARS_NUMBER = '-'*100, '='*100, '#'*100
15 COLORS_AND_MARKERS = [
16     'ro', 'g^', 'ys', 'c*', 'mv', 'r^', 'gs', 'y*', 'cv', 'mo', 'rs', 'g*', 'yv', 'co', 'm^',
17     'r*', 'gv', 'yo', 'c^', 'ms', 'rv', 'go', 'y^', 'cs', 'm*'
18 ]
19
20 # =====
21 # LIDE e HAYNES, 2009
22 # =====
23 M_H2O = 18.015 # g.mol-1
24 M_MEG = 62.068 # g.mol-1
25 R = 8.314472 # J.K-1.mol-1
26 # =====
27 # Temperature conversion
28 # =====
29 def K_to_C(T): # Kelvins to degrees Celsius
30     return T-273.15
31 def K_to_R(T): # Kelvins to degrees Rankine
32     return T*1.8
33 def K_to_F(T): # Kelvins to degrees Fahrenheit
34     return T*1.8-459.67
35 def C_to_K(T): # degrees Celsius to Kelvins
36     return T+273.15
37 def R_to_K(T): # degrees Rankine to Kelvins
38     return T/1.8
39 def F_to_K(T): # degrees Fahrenheit to Kelvins
40     return (T+459.67)/1.8
41 # =====
42
43 class chooseIsotherms(QDialog):
44     def __init__(self, lineEdit, comboBox_Mode, comboBox_Temperature):
45         super(chooseIsotherms, self).__init__()
46         if comboBox_Temperature.currentIndex() == 0:
47             f_T = lambda _: _
48             str_T = 'K'
49         elif comboBox_Temperature.currentIndex() == 1:
50             f_T = K_to_C
51             str_T = 'oC'
52         elif comboBox_Temperature.currentIndex() == 2:
53             f_T = K_to_R
54             str_T = 'oR'
55         elif comboBox_Temperature.currentIndex() == 3:
56             f_T = K_to_F
57             str_T = 'oF'
58         if comboBox_Mode.currentIndex() == 0:
59             T = list(f_T(T_reg))

```

```

60         self.setWindowTitle('Choose the isotherms for regression')
61     elif comboBox_Mode.currentIndex() == 1:
62         T = list(f_T(T_prd))
63         self.setWindowTitle('Choose the isotherms for prediction')
64     else:
65         self.reject()
66     self.lineEdit = lineEdit
67     T_set = set(map(eval, lineEdit.text().split()))
68     self.T_sorted = sorted(set(T) | T_set)
69     layout = QGridLayout()
70     self.checkboxes = []
71     for col in range(4):
72         row = 0
73         for t in self.T_sorted[col::4]:
74             n = T.count(t)
75             if n < 1:
76                 cb = QCheckBox('{}{}{}'.format(t, str_T))
77             elif n == 1:
78                 cb = QCheckBox('{}{}{}{}'.format(t, str_T))
79             elif n > 1:
80                 cb = QCheckBox('{}{}{}{}'.format(t, str_T, n))
81             layout.addWidget(cb, row, col)
82             if t in T_set:
83                 cb.setChecked(True)
84             row += 1
85             self.checkboxes.append(cb)
86     buttonBox = QDialogButtonBox(QDialogButtonBox.Cancel | QDialogButtonBox.Ok)
87     layout.addWidget(buttonBox, row+1, 0, 1, 4)
88     buttonBox.accepted.connect(self.accept)
89     buttonBox.rejected.connect(self.rejected)
90     self.setLayout(layout)
91
92     def accept(self):
93         T = [eval(cb.text().split()[0]) for cb in self.checkboxes if cb.isChecked()]
94         self.lineEdit.setText(' '.join(map(str, sorted(T))))
95         QDialog.accept(self)
96
97     def rejected(self):
98         QDialog.rejected(self)
99
100 class JAFOSMS(QDialog, ui_Form.Ui_Form):
101     def __init__(self):
102         super(JAFOSMS, self).__init__()
103         self.setupUi(self)
104         self.comboBox_Isotherms_Temperature_index = 0
105         self.textDocument_Statistics = QTextDocument()
106         self.textDocument_GeneralReport = QTextDocument()
107         self.textEdit_Statistics.setDocument(self.textDocument_Statistics)
108         self.textEdit_GeneralReport.setDocument(self.textDocument_GeneralReport)
109         self.textEdit_Statistics.setStyleSheet('font: 8pt "Courier";')
110         self.textEdit_GeneralReport.setStyleSheet('font: 8pt "Courier";')
111         self.textCursor_Statistics = self.textEdit_Statistics.textCursor()
112         self.textCursor_GeneralReport = self.textEdit_GeneralReport.textCursor()
113         re = '(((+|-)?((\d*[\.\d+])|(\d+[\.\d*])|([eE][+|-]?(\d+)?\s)*))'
114         editor = self.lineEdit_Isotherms
115         validator = QRegExpValidator(QRegExp(re), editor)
116         self.lineEdit_Isotherms.setValidator(validator)
117         self.pushButton_Datafile_Browse.clicked.connect(self.browse)
118         self.pushButton_Plot1.clicked.connect(self.plot1)
119         self.pushButton_Plot2.clicked.connect(self.plot2)
120         self.pushButton_Plot3.clicked.connect(self.plot3)
121         self.pushButton_Plot4.clicked.connect(self.plot4)
122         self.pushButton_Run.clicked.connect(self.runCalculations)
123         self.pushButton_Exit.clicked.connect(qApp, SLOT("quit()"))
124         self.pushButton_SaveGeneralReport.clicked.connect(self.saveGeneralReport)
125         self.pushButton_PrintGeneralReport.clicked.connect(self.printGeneralReport)
126         self.pushButton_PrintGraphic.clicked.connect(self.printGraphic)
127         self.pushButton_SaveStatistics.clicked.connect(self.saveStatistics)
128         self.pushButton_PrintStatistics.clicked.connect(self.printStatistics)
129         self.checkBox_Graphic.stateChanged.connect(self.showNavigationToolbar)
130         self.comboBox_Plot1_Abscissa_Temperature.currentIndexChanged.connect(
131             self.clearGraphicsAndStatistics)
132         self.comboBox_Plot1_Ordinate_Concentration.currentIndexChanged.connect(
133             self.clearGraphicsAndStatistics)
134         self.comboBox_Plot1_Ordinate_Solvent.currentIndexChanged.connect(
135             self.clearGraphicsAndStatistics)
136         self.comboBox_Isotherms_Mode.currentIndexChanged.connect(
137             self.clearGraphicsAndStatistics)
138         self.comboBox_Isotherms_Temperature.currentIndexChanged[int].connect(
139             self.convertTemperature)

```

```

140     self.lineEdit_Isotherms.lostFocus.connect(self.validate)
141     self.pushButton_Isotherms_Choose.clicked.connect(self.chooseIsotherms)
142     self.comboBox_Plot2_Abscissa_Concentration.currentIndexChanged.connect(
143         self.clearGraphicsAndStatistics)
144     self.comboBox_Plot2_Abscissa_Solvent.currentIndexChanged.connect(
145         self.clearGraphicsAndStatistics)
146     self.comboBox_Plot2_Ordinate_Concentration.currentIndexChanged.connect(
147         self.clearGraphicsAndStatistics)
148     self.comboBox_Plot3_Abscissa_Concentration.currentIndexChanged.connect(
149         self.clearGraphicsAndStatistics)
150     self.comboBox_Plot3_Abscissa_Solvent.currentIndexChanged.connect(
151         self.clearGraphicsAndStatistics)
152     self.comboBox_Plot3_Ordinate_Function.currentIndexChanged.connect(
153         self.clearGraphicsAndStatistics)
154     self.comboBox_Plot4_Abscissa_Concentration.currentIndexChanged.connect(
155         self.clearGraphicsAndStatistics)
156     self.comboBox_Plot4_Abscissa_Solvent.currentIndexChanged.connect(
157         self.clearGraphicsAndStatistics)
158     self.comboBox_Plot4_Ordinate_ExcessFunction.currentIndexChanged.connect(
159         self.clearGraphicsAndStatistics)
160     self.show()
161
162     def chooseIsotherms(self):
163         dialog = chooseIsotherms(self.lineEdit_Isotherms, self.comboBox_Isotherms_Mode,
164             self.comboBox_Isotherms_Temperature)
165         dialog.exec_()
166         self.validate()
167
168     def validate(self):
169         self.convertTemperature(self.comboBox_Isotherms_Temperature_index)
170
171     def convertTemperature(self, index):
172         T = array([t for t in self.T_to_K() if t >= -273.15])
173         if index == 1:
174             T = K_to_C(T)
175         elif index == 2:
176             T = K_to_R(T)
177         elif index == 3:
178             T = K_to_F(T)
179         self.comboBox_Isotherms_Temperature_index = index
180         self.lineEdit_Isotherms.setText(' '.join(map(str, sorted(T))))
181         if self.lineEdit_Isotherms.text():
182             self.groupBox_Plots.setEnabled(True)
183         else:
184             self.groupBox_Plots.setEnabled(False)
185         self.clearGraphicsAndStatistics()
186
187     def T_to_K(self):
188         T = array(sorted(set(map(eval, self.lineEdit_Isotherms.text().split()))))
189         if self.comboBox_Isotherms_Temperature_index == 1:
190             T = C_to_K(T)
191         elif self.comboBox_Isotherms_Temperature_index == 2:
192             T = R_to_K(T)
193         elif self.comboBox_Isotherms_Temperature_index == 3:
194             T = F_to_K(T)
195         return T
196
197     def showNavigationToolbar(self):
198         if self.checkBox_Graphic.isChecked():
199             self.mplWidget_Graphic.navbar.show()
200         else:
201             self.mplWidget_Graphic.navbar.hide()
202
203     def clearGraphicsAndStatistics(self):
204         self.mplWidget_Graphic.canvas.ax.clear()
205         self.mplWidget_Graphic.canvas.draw()
206         self.mplWidget_Graphic.setEnabled(False)
207         self.checkBox_Graphic.setEnabled(False)
208         self.pushButton_PrintGraphic.setEnabled(False)
209         self.textEdit_Statistics.setEnabled(False)
210         self.textEdit_Statistics.setText('')
211         self.pushButton_SaveStatistics.setEnabled(False)
212         self.pushButton_PrintStatistics.setEnabled(False)
213
214     def browse(self):
215         file_path, __ = QFileDialog.getOpenFileName(self,
216             'Filename of the dataset', '.', 'Dataset (*.set)')
217         if file_path:
218             self.lineEdit_Datafile.setText(file_path)
219             self.tabWidget_1.setEnabled(False)

```

```

220         self.groupBox_Plots.setEnabled(False)
221         self.mplWidget_Graphic.canvas.ax.clear()
222         self.mplWidget_Graphic.setEnabled(False)
223         self.textEdit_Statistics.setText('')
224         self.textEdit_GeneralReport.setText('')
225         self.tabWidget_1.setCurrentIndex(0)
226         self.pushButton_Run.setEnabled(True)
227
228     def saveGeneralReport(self):
229         fName, _ = QFileDialog.getSaveFileName(self, 'Saving_General_Report', '.',
230         'HTML_file (*.html);; All_files (*.*)')
231         if fName:
232             data = QFile(fName)
233             if data.open(QFile.WriteOnly | QFile.Text):
234                 QTextStream(data) << self.textDocument_GeneralReport.toHtml(encoding="utf-8")
235
236     def printGeneralReport(self):
237         printer = QPrinter()
238         dialog = QPrintDialog(printer, self)
239         dialog.setWindowTitle('Printing_General_Report')
240         if self.textEdit_GeneralReport.textCursor().hasSelection():
241             printer.setPrintRange(QPrinter.Selection)
242         if dialog.exec_() != QDialog.Accepted:
243             return
244         printer.setPrintRange(QPrinter.Selection)
245         print printer.printRange()
246         self.textEdit_GeneralReport.print_(printer)
247
248     def printGraphic(self):
249         pass
250
251     def saveStatistics(self):
252         fName, _ = QFileDialog.getSaveFileName(self, 'Saving_Statistics', '.',
253         'HTML_file (*.html);; All_files (*.*)')
254         if fName:
255             data = QFile(fName)
256             if data.open(QFile.WriteOnly | QFile.Text):
257                 QTextStream(data) << self.textDocument_Statistics.toHtml(encoding="utf-8")
258
259     def printStatistics(self):
260         printer = QPrinter()
261         dialog = QPrintDialog(printer, self)
262         dialog.setWindowTitle('Printing_Statistics')
263         if self.textEdit_Statistics.textCursor().hasSelection():
264             dialog.addEnabledOption(QAbstractPrintDialog.PrintSelection)
265         if dialog.exec_() != QDialog.Accepted:
266             return
267         self.textEdit_Statistics.print_(printer)
268
269
270 # =====
271 # AKERLOF, 1932
272 # =====
273     def epsilon_r_H2O(self, T):
274         return 10.**(1.9051-0.00205*(T-293.15))
275     def epsilon_r_MEG(self, T):
276         return 10.**(1.5872-0.00224*(T-293.15))
277 # =====
278 # JOUYBAN, SOLTANPOUR and CHAN, 2004
279 # =====
280     def epsilon_r_H2O_MEG(self, epsilon_r_H2O, epsilon_r_MEG, x_SF_MEG_in_H2O_MEG, T):
281         x_SF_H2O_in_H2O_MEG = 1.-x_SF_MEG_in_H2O_MEG
282         return exp(x_SF_H2O_in_H2O_MEG*log(epsilon_r_H2O)+x_SF_MEG_in_H2O_MEG*log(epsilon_r_MEG)+
283         x_SF_H2O_in_H2O_MEG*x_SF_MEG_in_H2O_MEG/T*(153.6+57.3*(x_SF_H2O_in_H2O_MEG-x_SF_MEG_in_H2O_MEG)))
284 # =====
285 # GREEN and PERRY, 2007
286 # =====
287     def rho_H2O(self, T):
288         tau = 1.-T/647.096
289         return (17.863+58.606*tau**0.35-95.396*tau**(2./3.)+ 213.89*tau-141.26*tau**(4./3.))*M_H2O
290     def rho_MEG(self, T):
291         return (1.315/(0.25125**((1.+(1.-T/720.))*0.21868)))*M_MEG
292 # =====
293 # Rule of Mixture Ideal with Inverse Density
294 # =====
295     def rho_H2O_MEG(self, rho_H2O, rho_MEG, x_SF_MEG_in_H2O_MEG):
296         x_SF_H2O_in_H2O_MEG = 1.-x_SF_MEG_in_H2O_MEG
297         w_SF_MEG_in_H2O_MEG =
298             x_SF_MEG_in_H2O_MEG*M_MEG/(x_SF_H2O_in_H2O_MEG*M_H2O+x_SF_MEG_in_H2O_MEG*M_MEG)
299         return 1./self.p12_ideal(1./rho_H2O, 1./rho_MEG, w_SF_MEG_in_H2O_MEG)

```

```

299 # =====
300 # PITZER, 1973
301 # =====
302 def I_b(self, b_MX):
303     b_M, b_X = nu_M*b_MX, nu_X*b_MX
304     return (b_M*z_M**2.+b_X*z_X**2.)/2.
305 def X_phi(self, I_b, b_=1.2):
306     sqrt_I_b = sqrt(I_b)
307     return sqrt_I_b/(1.+b_*sqrt_I_b)
308 def X_gamma(self, I_b, b_=1.2):
309     sqrt_I_b = sqrt(I_b)
310     return 2./b_*log(1.+b_*sqrt_I_b)+self.X_phi(I_b)
311 def f_phi_MX(self, A_phi, X_phi):
312     return -A_phi*X_phi
313 def f_gamma_MX(self, A_phi, X_gamma):
314     return -A_phi*X_gamma
315 def Y_phi(self, I_b, alpha=2.):
316     sqrt_I_b = sqrt(I_b)
317     return exp(-alpha*sqrt_I_b)
318 def Y_gamma(self, I_b, alpha=2.):
319     sqrt_I_b = sqrt(I_b)
320     return (1.-(1.+alpha*sqrt_I_b-0.5*alpha**2.*I_b)*self.Y_phi(I_b))/(alpha**2.*I_b)
321 def B_phi_MX(self, beta0_MX, beta1_MX, Y_phi):
322     return beta0_MX+beta1_MX*Y_phi
323 def B_gamma_MX(self, beta0_MX, beta1_MX, Y_gamma):
324     return 2.*(beta0_MX+beta1_MX*Y_gamma)
325 def C_gamma_MX(self, C_phi_MX):
326     return 1.5*C_phi_MX
327 def phi(self, b_MX, f_phi_MX, B_phi_MX, C_phi_MX):
328     return 1+abs(z_M*z_X)*f_phi_MX+2.*b_MX*nu_M*nu_X/nu_MX*(B_phi_MX+b_MX*(nu_M*nu_X)**0.5*
329         C_phi_MX)
330 def ln_gamma_MX(self, b_MX, f_gamma_MX, B_gamma_MX, C_gamma_MX):
331     return abs(z_M*z_X)*f_gamma_MX+2.*b_MX*nu_M*nu_X/nu_MX*(B_gamma_MX+b_MX*(nu_M*nu_X)**0.5*
332         C_gamma_MX)
333 # =====
334 # SILVESTER and PITZER, 1977 (NaCl); ARCHER, 1999 (KCl)
335 # =====
336 def f_Pitzer_parameters(self, i, T):
337     Tr = 298.15 # K
338     if MX == 'NaCl':
339         q = [[ 0.0765, 0.2664, 0.00127 ],
340             [ -777.03, 0., 33.317 ],
341             [ -4.4706, 0., 0.09421 ],
342             [ 0.008946, 6.1608e-5, -4.655e-5],
343             [ -3.3158e-6, 1.0715e-6, 0. ]]
344         q = array(q).T
345         return (q[i,0]+q[i,1]*(1/T-1/Tr)+q[i,2]*log(T/Tr)+q[i,3]*(T-Tr)+
346             q[i,4]*(T**2-Tr**2))
347     elif MX == 'KCl':
348         b = [[ 0.413229483398493, 0.206691413598171, -0.00133515934994478],
349             [ -0.0870121476114027, 0.102544606022162, 0. ],
350             [ 0.101413736179231, 0., 0. ],
351             [ -0.0199822538522801, 0., 0.00234117693834228],
352             [ -0.0998120581680816, 0., -0.00075896583546707],
353             [ 0., -0.00188349608000903, 0. ]]
354         b = array(b).T
355         T0 = 1.0 # K
356         return (b[i,0]+1e-2*b[i,1]*(T-Tr)/T0+1e-5*b[i,2]*((T-Tr)/T0)**2+1e2*b[i,3]*T0/(T-225)+
357             1e3*b[i,4]*(T0/T)+1e6*b[i,5]*(T0/(T-225))**3)
358     return None
359 def beta0_MX_H2O(self, T):
360     return self.f_Pitzer_parameters(0, T)
361 def beta1_MX_H2O(self, T):
362     return self.f_Pitzer_parameters(1, T)
363 def C_phi_MX_H2O(self, T):
364     return self.f_Pitzer_parameters(2, T)
365 # =====
366 # CHEN, BRITT, et al., 1982
367 # =====
368 def A_phi_H2O(self, T):
369     return (-61.44534*exp((T-273.15)/273.15)+2.864468*exp(2.*(T-273.15)/273.15)+183.5379*
370         log(T/273.15)-0.6820223*(T-273.15)+0.0007875695*(T**2.-273.15**2.))+58.95788*(273.15/T))
371 # =====
372 # Debye-Huckel constant in other solvent
373 # =====
374 def A_phi_solvent(self, A_phi_H2O, rho_H2O, epsilon_r_H2O, rho_solvent, epsilon_r_solvent):
375     return ((rho_solvent/rho_H2O)**0.5*(epsilon_r_H2O/epsilon_r_solvent)**1.5*A_phi_H2O)
376 # =====
377 # LORIMER, 1993
378 # =====

```

```

379 def ln_gamma_MX_in_MEG(self, ln_gamma_MX_in_H2O, b_MX_in_H2O, b_MX_in_MEG, T):
380     return (ln_gamma_MX_in_H2O+log(b_MX_in_H2O/b_MX_in_MEG)-Delta_tr_G0_H2O_to_MEG/(nu_MX*R*T))
381 def beta0_MX_H2O_MEG(self, beta0_MX_H2O):
382     return beta0_MX_H2O
383 def betal_MX_H2O_MEG(self, betal_MX_H2O, betal_MX_MEG, epsilon_r_H2O, epsilon_r_MEG,
384     epsilon_r_H2O_MEG):
385     beta0 = 1.
386     A = log(betal_MX_H2O/betal_MX_MEG)/log(epsilon_r_H2O/epsilon_r_MEG)
387     B = log(betal_MX_H2O/beta0)-A*log(epsilon_r_H2O)
388     return beta0*exp(A*log(epsilon_r_H2O_MEG)+B)
389 def C_phi_MX_H2O_MEG(self, C_phi_MX_H2O):
390     return C_phi_MX_H2O
391 # =====
392 # Ideal property
393 # =====
394 def p12_ideal(self, p1, p2, x2):
395     x1 = 1.-x2
396     return x1*p1+x2*p2
397 # =====
398 # Excess Property
399 # =====
400 def p12_excess(self, p12, p12_ideal):
401     return p12-p12_ideal
402 # =====
403 # Normalization of salt-free mole fraction of MEG
404 # =====
405 def x_SF_MEG_in_H2O_MEG(self, i, c_H2O, c_MEG=None):
406     if i == 1: # m_H2O (g), m_MEG (g)
407         pass
408     elif i == 2: # w_SF_H2O_in_H2O_MEG
409         c_MEG = 1.-c_H2O
410     elif i == 3: # w_SF_MEG_in_H2O_MEG
411         c_H2O, c_MEG = 1.-c_H2O, c_H2O
412     return c_MEG/M_MEG/(c_H2O/M_H2O+c_MEG/M_MEG)
413 # =====
414 # Normalization of molality of the salt in the solvent mixture
415 # =====
416 def b_MX_in_H2O_MEG(self, i, c_MX, c_H2O_MEG=None):
417     if i == 1: # m_MX (g), m_H2O_MEG (g)
418         pass
419     elif i == 2: # w_MX_in_H2O_MEG_MX
420         c_H2O_MEG = 1.-c_MX
421     return c_MX/M_MX/c_H2O_MEG*1000.
422 # =====
423 # Concentration conversion
424 # =====
425 def b_MX_in_H2O_MEG_to_c_MX_in_H2O_MEG(self, b_MX_in_H2O_MEG, rho_H2O_MEG):
426     # Molality to Molarity
427     return b_MX_in_H2O_MEG*rho_H2O_MEG/1000
428 def b_MX_in_H2O_MEG_to_x_MX_in_H2O_MEG(self, b_MX_in_H2O_MEG, x_SF_MEG_in_H2O_MEG):
429     # Molality to Molar Fraction
430     return b_MX_in_H2O_MEG/(1000/((1-x_SF_MEG_in_H2O_MEG)*M_H2O+x_SF_MEG_in_H2O_MEG*M_MEG)+
431     b_MX_in_H2O_MEG)
432 def b_MX_in_H2O_MEG_to_w_MX_in_H2O_MEG(self, b_MX_in_H2O_MEG, _):
433     # Molality to Massic Fraction
434     return b_MX_in_H2O_MEG/(1000/M_MX+b_MX_in_H2O_MEG)
435 # =====
436 # Salt-free Concentration conversion
437 # =====
438 def x_SF_MEG_in_H2O_MEG_to_w_SF_MEG_in_H2O_MEG(self, x_SF_MEG_in_H2O_MEG):
439     # Molar Fraction to Mass Fraction
440     return x_SF_MEG_in_H2O_MEG/((1-x_SF_MEG_in_H2O_MEG)*M_H2O/M_MEG+x_SF_MEG_in_H2O_MEG)
441 # =====
442 def printText(self, text, flag=True):
443     if flag:
444         self.textCursor_GeneralReport.insertText(text.decode('utf-8'))
445     else:
446         self.textCursor_Statistics.insertText(text.decode('utf-8'))
447
448 def printVars(self, *args):
449     names, values, units = (args[i::3] for i in range(3))
450     for name, value, unit in zip(names, values, units):
451         if unit:
452             self.printText('{ }={ }\n'.format(name, value, unit))
453         else:
454             self.printText('{ }={ }\n'.format(name, value))
455
456 def printVectors(self, *args):
457     from numpy import float64
458     def f1(x):

```

```

459         return len(str(x))
460     def f2(x):
461         return str(x).find('.')
462     def f3(x):
463         return f1(x)-f2(x)
464     names, vectors, units = (args[i::3] for i in range(3))
465     tableformat = QTextTableFormat()
466     tableformat.setCellPadding(3)
467     tableformat.setCellSpacing(1)
468     tableformat.setHeaderRowCount(1)
469     headerformat = QTextCharFormat()
470     headerformat.setFontWeight(QFont.Bold)
471     brush = headerformat.background()
472     brush.setColor(Qt.lightGray)
473     brush.setStyle(Qt.SolidPattern)
474     headerformat.setBackground(brush)
475     nrows, ncols = len(vectors[0]), len(vectors)
476     table = self.textEdit_GeneralReport.textCursor().insertTable(nrows+1, ncols, tableformat)
477     for col in range(ncols):
478         cell = table.cellAt(0, col)
479         cell.setFormat(headerformat)
480         cellCursor = cell.firstCursorPosition()
481         cellCursor.insertText(names[col])
482         if units[col]:
483             cellCursor.insertText(' '+unicode(units[col])+' ')
484         vector = vectors[col]
485         if type(vector[0]) == float64:
486             a = max(map(f2, vector))
487             b = max(map(f3, vector))
488             format_ = '{:{}.{}f}'.format(a+b, b-1)
489         else:
490             format_ = '{} '
491         for row in range(nrows):
492             cell = table.cellAt(row+1, col)
493             cellCursor = cell.firstCursorPosition()
494             cellCursor.insertText(format_.format(vector[row]))
495
496     def readDataFile(self, filename):
497         f = QFile(filename)
498         if f.open(QFile.ReadOnly | QFile.Truncate | QFile.Text):
499             lines = QTextStream(f).readAll().splitlines()
500             f.close()
501             values, vectors, list_of_types = [], [], ['str', 'int', 'float']
502             def processVectors():
503                 for vector in vectors:
504                     vector0 = vector[0]
505                     if vector0 in list_of_types:
506                         values.append(map(eval, vector[3:]))
507                         if vector[2]:
508                             values.append(vector[2])
509             for line in lines:
510                 args = [arg.strip() for arg in line.split(';')]
511                 type_ = args[0]
512                 if type_:
513                     if vectors:
514                         processVectors()
515                         vectors = []
516                     if type_ in list_of_types:
517                         values.append(eval(type_)(args[2]))
518                         if args[3:4]:
519                             values.append(args[3])
520                     elif type_ == 'vectors':
521                         vectors.extend([i for i in args[1:]])
522                 else:
523                     for i, v in enumerate(args[1:]):
524                         vectors[i].append(v)
525             if vectors:
526                 processVectors()
527             return values
528
529     def readDataset(self):
530         global MX, z_M, z_X, nu_M, nu_X, M_MX, Delta_tr_G0_H2O_to_MEG
531         (MX, z_M, z_X, nu_M, nu_X, M_MX, units_M_MX, Delta_tr_G0_H2O_to_MEG,
532          units_Delta_tr_G0_H2O_to_MEG, data_filenames) = \
533             self.readDataFile(self.lineEdit_Datafile.text())
534         global nu_MX
535         nu_MX = nu_M+nu_X
536         self.printText('Main data filename: {} \n'.format(QFileInfo(
537             self.lineEdit_Datafile.text()).fileName()+CHARS_MINUS+' \n'))

```

```

539 self.printVars('MX', MX, '', 'z_M', z_M, '', 'z_X', z_X, '', 'nu_M',
540 nu_M, '', 'nu_X', nu_X, '', 'M_MX', M_MX, units_M_MX,
541 'Delta_tr_G0_H2O_to_MEG', Delta_tr_G0_H2O_to_MEG,
542 units_Delta_tr_G0_H2O_to_MEG)
543 path = QFileInfo(self.lineEdit_Datafile.text()).path()
544 global bibref
545 global idx_set_reg, x_SF_MEG_in_H2O_MEG_reg, b_MX_in_H2O_MEG_reg, T_reg
546 global idx_set_prd, x_SF_MEG_in_H2O_MEG_prd, b_MX_in_H2O_MEG_prd, T_prd
547 global b_MX_in_H2O_chk, phi_MX_in_H2O_chk, gamma_MX_in_H2O_chk
548 bibref, idx_set, idx_set_reg, idx_set_prd = [], 0, [], []
549 (x_SF_MEG_in_H2O_MEG_reg, x_SF_MEG_in_H2O_MEG_prd, b_MX_in_H2O_MEG_reg,
550 b_MX_in_H2O_MEG_prd, T_reg, T_prd) = ([], [], [], [], [], [])
551 for data_filename in data_filenames:
552 data = self.readDataFile(path+'/' + data_filename)
553 bibref.append(data[0])
554 self.printText(CHARS_EQUAL+'\nData_{}_filename:{}_{}\nReferences:{}_{}\n'
555 '\n'.format(data_filename, bibref[-1])+CHARS_MINUS+'\n')
556 DataType = data[1]
557 data = map(lambda x: asarray(x) if type(x) != str else x, data[2:])
558 if DataType == 0:
559 # b_MX_in_H2O (mol/Kg), phi_MX_in_H2O, gamma_MX_in_H2O
560 (b_MX_in_H2O_chk, units_b_MX_in_H2O_chk, phi_MX_in_H2O_chk,
561 gamma_MX_in_H2O_chk) = data
562 self.printVectors('b_MX_in_H2O_chk', b_MX_in_H2O_chk,
563 units_b_MX_in_H2O_chk, 'phi_MX_in_H2O_chk', phi_MX_in_H2O_chk,
564 '', 'gamma_MX_in_H2O', gamma_MX_in_H2O_chk, '')
565 elif DataType == 1:
566 # w_SF_MEG_in_H2O_MEG, m_H2O_MEG (g), m_MX (g), T (oC)
567 (w_SF_MEG_in_H2O_MEG, m_H2O_MEG, units_m_H2O_MEG, m_MX,
568 units_m_MX, T, units_T, use) = data
569 self.printVectors('w_SF_MEG_in_H2O_MEG', w_SF_MEG_in_H2O_MEG, '',
570 'm_H2O_MEG', m_H2O_MEG, units_m_H2O_MEG, 'm_MX', m_MX,
571 units_m_MX, 'T', T, units_T, 'use', use, '')
572 x_SF_MEG_in_H2O_MEG = self.x_SF_MEG_in_H2O_MEG(3, w_SF_MEG_in_H2O_MEG)
573 b_MX_in_H2O_MEG = self.b_MX_in_H2O_MEG(1, m_MX, m_H2O_MEG)
574 T += 273.15
575 elif DataType == 2:
576 # w_SF_MEG_in_H2O_MEG, w_MX_in_H2O_MEG_MX, T (oC)
577 (w_SF_MEG_in_H2O_MEG, w_MX_in_H2O_MEG_MX, T, units_T, use) = data
578 self.printVectors('w_SF_MEG_in_H2O_MEG', w_SF_MEG_in_H2O_MEG, '',
579 'w_MX_in_H2O_MEG_MX', w_MX_in_H2O_MEG_MX, '', 'T', T,
580 units_T, 'use', use, '')
581 x_SF_MEG_in_H2O_MEG = self.x_SF_MEG_in_H2O_MEG(3, w_SF_MEG_in_H2O_MEG)
582 b_MX_in_H2O_MEG = self.b_MX_in_H2O_MEG(2, w_MX_in_H2O_MEG_MX)
583 T += 273.15
584 elif DataType == 3:
585 # w_MX_in_H2O_MEG_MX, m_H2O (g), m_MEG (g), T (oC)
586 (w_MX_in_H2O_MEG_MX, m_H2O, units_m_H2O, m_MEG, units_m_MEG, T,
587 units_T, use) = data
588 self.printVectors('w_MX_in_H2O_MEG_MX', w_MX_in_H2O_MEG_MX,
589 '', 'm_H2O', m_H2O, units_m_H2O, 'm_MEG', m_MEG,
590 units_m_MEG, 'T', T, units_T, 'use', use, '')
591 x_SF_MEG_in_H2O_MEG = self.x_SF_MEG_in_H2O_MEG(1, m_H2O, m_MEG)
592 b_MX_in_H2O_MEG = self.b_MX_in_H2O_MEG(2, w_MX_in_H2O_MEG_MX)
593 T += 273.15
594 elif DataType == 4:
595 # w_SF_H2O_in_H2O_MEG, w_MX_in_H2O_MEG_MX, T (K)
596 (w_SF_H2O_in_H2O_MEG, w_MX_in_H2O_MEG_MX, T, units_T, use) = data
597 self.printVectors('w_SF_H2O_in_H2O_MEG', w_SF_H2O_in_H2O_MEG, '',
598 'w_MX_in_H2O_MEG_MX', w_MX_in_H2O_MEG_MX, '', 'T', T,
599 units_T, 'use', use, '')
600 x_SF_MEG_in_H2O_MEG = self.x_SF_MEG_in_H2O_MEG(2, w_SF_H2O_in_H2O_MEG)
601 b_MX_in_H2O_MEG = self.b_MX_in_H2O_MEG(2, w_MX_in_H2O_MEG_MX)
602 elif DataType == 5:
603 # w_SF_MEG_in_H2O_MEG, w_MX_in_H2O_MEG_MX, T (K)
604 (w_SF_MEG_in_H2O_MEG, w_MX_in_H2O_MEG_MX, T, units_T, use) = data
605 self.printVectors('w_SF_MEG_in_H2O_MEG', w_SF_MEG_in_H2O_MEG, '',
606 'w_MX_in_H2O_MEG_MX', w_MX_in_H2O_MEG_MX, '', 'T', T,
607 units_T, 'use', use, '')
608 x_SF_MEG_in_H2O_MEG = self.x_SF_MEG_in_H2O_MEG(3, w_SF_MEG_in_H2O_MEG)
609 b_MX_in_H2O_MEG = self.b_MX_in_H2O_MEG(2, w_MX_in_H2O_MEG_MX)
610 for x, b, t, u in zip(x_SF_MEG_in_H2O_MEG, b_MX_in_H2O_MEG, T, use):
611 if u == "REGR":
612 idx_set_reg.append(idx_set)
613 x_SF_MEG_in_H2O_MEG_reg = append(x_SF_MEG_in_H2O_MEG_reg, x)
614 b_MX_in_H2O_MEG_reg = append(b_MX_in_H2O_MEG_reg, b)
615 T_reg = append(T_reg, t)
616 elif u == "PRED":
617 idx_set_prd.append(idx_set)
618 x_SF_MEG_in_H2O_MEG_prd = append(x_SF_MEG_in_H2O_MEG_prd, x)

```



```

619         b_MX_in_H2O_MEG_prd = append(b_MX_in_H2O_MEG_prd, b)
620         T_prd = append(T_prd, t)
621         idx_set += 1
622     return
623
624
625 def nl_regression(self, f_str, X, y, theta0):
626     import scipy.stats as stats
627     import scipy.optimize as optimize
628     if type(f_str) == tuple:
629         f, fstr = f_str
630     else:
631         fstr = 'lambda'+f_str
632         f = eval(fstr)
633     def plus_minus(a, b):
634         return a-b, a+b
635     theta_optimum, cov_theta_optimum = optimize.curve_fit(f, X, y, theta0)
636     N, P = X.shape[-1], theta_optimum.shape[-1]
637     y_optimum = f(X, *theta_optimum)
638     # Total Sum of Squares (TSS)
639     TSS_corrected, TSS_uncorrected = ((y-y.mean())**2).sum(), (y**2).sum()
640     # Residual Sum of Squares (RSS) or Sum of Squared Residuals (SSR) or
641     # Sum of Squared Errors of Prediction (SSE)
642     RSS = ((y-y_optimum)**2).sum()
643     # Explained Sum of Squares (ESS)
644     ESS_uncorrected = TSS_uncorrected-RSS
645     # Mean Total Sum of Squares
646     MST_corrected = TSS_corrected/(N-1)
647     # Mean Residual Sum of Squares
648     MSR = RSS/(N-P)
649     # Mean Explained Sum of Squares
650     MSE_uncorrected = ESS_uncorrected/P
651     alpha = 0.05
652     F_value_uncorrected = MSE_uncorrected/MSR
653     Pr_F_value_uncorrected = stats.f.sf(F_value_uncorrected, N-P, P)
654     s, R2, R2_bar = sqrt(MSR), 1-RSS/TSS_corrected, 1-MSR/MST_corrected
655     ASE = sqrt(cov_theta_optimum.diagonal())
656     corr_theta_optimum = (cov_theta_optimum/ASE).T/ASE
657     t_value, t_critical = theta_optimum/ASE, stats.t.isf(alpha/2., N-P)
658     Pr_tvalue = stats.t.sf(abs(t_value), N-P)*2.
659     CI_theta = plus_minus(theta_optimum, t_critical*ASE)
660     self.printText('Non-Linear Least Squares Summary Statistics\n\n')
661     self.printText('Function: y={0}\n\n'.format(f_str))
662     self.printText('Source: Sum of Squares, DF, Mean Squa',
663                    're, F-value, Pr(>F)\n')
664     self.printText('Regression, {0:4d}, {0:14.6f}, {0:14.6f}, {0:1',
665                    '6.6f}, {0:8.6f}\n'.format(P, ESS_uncorrected, MSE_uncorrected,
666                    F_value_uncorrected, Pr_F_value_uncorrected))
667     self.printText('Residual, {0:4d}, {0:14.6f}, {0:14.6f}, {0:1',
668                    '6.6f}, {0:8.6f}\n'.format(N-P, RSS, MSR))
669     self.printText('Uncorrected Total, {0:4d}, {0:14.6f}, {0:14.6f}, {0:1',
670                    '6.6f}, {0:8.6f}\n'.format(N, TSS_uncorrected))
671     self.printText('Corrected Total, {0:4d}, {0:14.6f}, {0:14.6f}, {0:1',
672                    '6.6f}, {0:8.6f}\n'.format(N-1, TSS_corrected))
673     self.printText('alpha={0:8.6f}, t-critical={0:8.6f}\n\n'.format(alpha,
674                    t_critical))
675     self.printText('s={0:8.6f}, R-sq={0:14.6f}, R-sq(adj)={0:14.6f}, R-sq',
676                    'adj={0:14.6f}\n\n'.format(s, R2, R2_bar))
677     # Average Absolute Deviation
678     AAD = abs(y-y_optimum).sum()/N
679     # Maximum Absolute Deviation
680     MAD = max(abs(y-y_optimum))
681     # Average Absolute Relative Deviation
682     AARD = abs((y-y_optimum)/y).sum()/N
683     # Maximum Absolute Relative Deviation
684     MARD = max(abs((y-y_optimum)/y))
685     self.printVectors('y', y, 'y_optimum', y_optimum, '|y-y_optimum|', abs(y-y_optimum), '|(y-y_optimum)/y| (%)', abs((y-y_optimum)/y*100))
686     self.printText('Average Absolute Deviation, {0:0.6f}\n'.format(AAD))
687     self.printText('Maximum Absolute Deviation, {0:0.6f}\n'.format(MAD))
688     self.printText('Average Absolute Relative Deviation, {0:0.3%}\n'.format(AARD))
689     self.printText('Maximum Absolute Relative Deviation, {0:0.3%}\n\n'.format(MARD))
690     self.printText('Parameter Estimate, Asymptotic, t-value',
691                    '|Pr(>|t|)| Asymptotic, {0:2.0%}\n'.format(1-alpha))
692     self.printText('Standard Error, Confidence Interval\n')
693     self.printText('Lower, Upper\n')
694     for i in range(P):
695         self.printText('{0:10s} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
696                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
697                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
698                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
699                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
700                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
701                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
702                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
703                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
704                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
705                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
706                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
707                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
708                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
709                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
710                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
711                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
712                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
713                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
714                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
715                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
716                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
717                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
718                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
719                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
720                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
721                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
722                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
723                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
724                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
725                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
726                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
727                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
728                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
729                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
730                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
731                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
732                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
733                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
734                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
735                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
736                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
737                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
738                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
739                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
740                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
741                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
742                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
743                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
744                        ' {0:10.6f} {0:12.6f} {0:12.6f} {0:10.6f} {0:8.6f}'
745                        ' {0:10.6f} {0:12.6f} {0:
```

```

698         '{:12.6f}\n'.format('P['+str(i)+']',
699         theta_optimum[i], ASE[i], t_value[i], Pr_tvalue[i],
700         CI_theta[0][i], CI_theta[1][i]))
701     self.printText('\nParameter\CorrelationMatrix\n')
702     for i in range(P):
703         self.printText('{:11s}'.format('P['+str(i)+']'))
704     for i in range(P):
705         self.printText('\n{:10s}'.format('P['+str(i)+']'))
706         for j in range(P):
707             self.printText('{:9.6f}'.format(corr_theta_optimum[i][j]))
708     self.printText('\n')
709     return f, theta_optimum
710
711 def calculateStatistics(self, ycalc, yexp):
712     import scipy.stats as stats
713     N, P = yexp.shape[0], 1
714     # Total Sum of Squares (TSS)
715     TSS_corrected, TSS_uncorrected = ((yexp-yexp.mean())**2).sum(), (yexp**2).sum()
716     # Residual Sum of Squares (RSS) or Sum of Squared Residuals (SSR) or
717     # Sum of Squared Errors of Prediction (SSE)
718     RSS = ((yexp-ycalc)**2).sum()
719     # Explained Sum of Squares (ESS)
720     ESS_uncorrected = TSS_uncorrected-RSS
721     # Mean Total Sum of Squares
722     MST_corrected = TSS_corrected/(N-1)
723     # Mean Residual Sum of Squares
724     MSR = RSS/(N-P)
725     # Mean Explained Sum of Squares
726     MSE_uncorrected = ESS_uncorrected/P
727     F_value_uncorrected = MSE_uncorrected/MSR
728     Pr_F_value_uncorrected = stats.f.sf(F_value_uncorrected, N-P, P)
729     s, R2, R2_bar = sqrt(MSR), 1-RSS/TSS_corrected, 1-MSR/MST_corrected
730     # Average Absolute Deviation
731     AAD = abs(yexp-ycalc).sum()/N
732     # Maximum Absolute Deviation
733     MAD = max(abs(yexp-ycalc))
734     # Average Absolute Relative Deviation
735     AARD = abs((yexp-ycalc)/yexp).sum()/N
736     # Maximum Absolute Relative Deviation
737     MARD = max(abs((yexp-ycalc)/yexp))
738     self.printText('Summary Statistics\n', False)
739     self.printText('Source\DF\Sum of Squares\Mean Squa
740     're\value\Pr(>F)\n', False)
741     self.printText('Regression\n{:4d}\{:14.6f}\{:14.6f}\{:1
742     '6.6f}\{:8.6f}\n'.format(P, ESS_uncorrected, MSE_uncorrected,
743     F_value_uncorrected, Pr_F_value_uncorrected), False)
744     self.printText('Residual\n{:4d}\{:14.6f}\{:14.6f}\n'
745     .format(N-P, RSS, MSR), False)
746     self.printText('Uncorrected Total\n{:4d}\{:14.6f}\n'.format(N,
747     TSS_uncorrected), False)
748     self.printText('Corrected Total\n{:4d}\{:14.6f}\n'.format(N-
749     1, TSS_corrected), False)
750     self.printText('s=\{:8.6f}\R-sq=\{:8.6f}\R-sq(adj)=\{:8.6f}\n'
751     '\n'.format(s, R2, R2_bar), False)
752     self.printText('Average Absolute Deviation=\{:0.6f}\n'.format(AAD), False)
753     self.printText('Maximum Absolute Deviation=\{:0.6f}\n'.format(MAD), False)
754     self.printText('Average Absolute Relative Deviation(%)=\{:0.3%}\n'.format(AARD), False)
755     self.printText('Maximum Absolute Relative Deviation(%)=\{:0.3%}\n'.format(MARD), False)
756
757 #
758 # Function that calculates:
759 # mu0_MX_in_H2O_MEG_E_DividedBy_nRT, ln_gamma_MX_in_H2O_MEG_E
760 # and ln_b_MX_in_H2O_MEG_E
761 # from:
762 # b_MX_in_H2O_MEG, b_MX_in_H2O, b_MX_in_MEG, x_SF_MEG_in_H2O_MEG,
763 # A_phi_H2O, beta0_MX_H2O, beta1_MX_H2O, C_phi_MX_H2O, rho_H2O,
764 # rho_MEG, epsilon_r_H2O, epsilon_r_MEG and T
765 #
766 def f_excess(self, b_MX_in_H2O_MEG, b_MX_in_H2O, b_MX_in_MEG, x_SF_MEG_in_H2O_MEG,
767 A_phi_H2O, beta0_MX_H2O, beta1_MX_H2O, C_phi_MX_H2O, rho_H2O,
768 rho_MEG, epsilon_r_H2O, epsilon_r_MEG, T):
769     ln_gamma_MX_in_H2O = self.ln_gamma_MX(
770         b_MX_in_H2O,
771         self.f_gamma_MX(A_phi_H2O, self.X_gamma(self.I_b(b_MX_in_H2O))),
772         self.B_gamma_MX(beta0_MX_H2O, beta1_MX_H2O, self.Y_gamma(self.I_b(
773             b_MX_in_H2O))),
774         self.C_gamma_MX(C_phi_MX_H2O))
775     ln_gamma_MX_in_MEG = self.ln_gamma_MX_in_MEG(ln_gamma_MX_in_H2O,
776         b_MX_in_H2O, b_MX_in_MEG, T)
777     def ff(beta1_MX_MEG):

```

```

778     beta0_MX_MEG, C_phi_MX_MEG = beta0_MX_H2O, C_phi_MX_H2O
779     return ln_gamma_MX_in_MEG - self.ln_gamma_MX(
780         b_MX_in_MEG,
781         self.f_gamma_MX(
782             self.A_phi_solvent(A_phi_H2O, rho_H2O, epsilon_r_H2O,
783                 rho_MEG, epsilon_r_MEG),
784             self.X_gamma(self.I_b(b_MX_in_MEG))),
785             self.B_gamma_MX(beta0_MX_MEG, beta1_MX_MEG, self.Y_gamma(
786                 self.I_b(b_MX_in_MEG))),
787             self.C_gamma_MX(C_phi_MX_MEG))
788     beta1_MX_MEG = fsolve(ff, beta1_MX_H2O)
789     epsilon_r_H2O_MEG = self.epsilon_r_H2O_MEG(epsilon_r_H2O,
790         epsilon_r_MEG, x_SF_MEG_in_H2O_MEG, T)
791     beta0_MX_H2O_MEG = self.beta0_MX_H2O_MEG(beta0_MX_H2O)
792     beta1_MX_H2O_MEG = self.beta1_MX_H2O_MEG(beta1_MX_H2O, beta1_MX_MEG,
793         epsilon_r_H2O, epsilon_r_MEG, epsilon_r_H2O_MEG)
794     C_phi_MX_H2O_MEG = self.C_phi_MX_H2O_MEG(C_phi_MX_H2O)
795     ln_gamma_MX_in_H2O_MEG = self.ln_gamma_MX(
796         b_MX_in_H2O_MEG,
797         self.f_gamma_MX(
798             self.A_phi_solvent(
799                 A_phi_H2O,
800                 rho_H2O,
801                 epsilon_r_H2O,
802                 self.rho_H2O_MEG(rho_H2O, rho_MEG, x_SF_MEG_in_H2O_MEG),
803                 epsilon_r_H2O_MEG),
804             self.X_gamma(self.I_b(b_MX_in_H2O_MEG))),
805             self.B_gamma_MX(beta0_MX_H2O_MEG, beta1_MX_H2O_MEG, self.Y_gamma(self.I_b(
806                 b_MX_in_H2O_MEG))),
807             self.C_gamma_MX(C_phi_MX_H2O_MEG))
808     b0 = 1.0
809     ln_b_MX_in_H2O_MEG_id = self.p12_ideal(log(b_MX_in_H2O/b0), log(
810         b_MX_in_MEG/b0), x_SF_MEG_in_H2O_MEG)
811     ln_b_MX_in_H2O_MEG_E = self.p12_excess(log(b_MX_in_H2O_MEG/b0),
812         ln_b_MX_in_H2O_MEG_id)
813     ln_gamma_MX_in_H2O_MEG_id = self.p12_ideal(ln_gamma_MX_in_H2O,
814         ln_gamma_MX_in_MEG, x_SF_MEG_in_H2O_MEG)
815     ln_gamma_MX_in_H2O_MEG_E = self.p12_excess(ln_gamma_MX_in_H2O_MEG,
816         ln_gamma_MX_in_H2O_MEG_id)
817     mu0_MX_in_H2O_MEG_E_DividedBy_nRT = (-ln_b_MX_in_H2O_MEG_E -
818         ln_gamma_MX_in_H2O_MEG_E)
819     return (mu0_MX_in_H2O_MEG_E_DividedBy_nRT, ln_gamma_MX_in_H2O_MEG_E,
820         ln_b_MX_in_H2O_MEG_E, ln_gamma_MX_in_H2O_MEG)
821
822 # =====
823 # Function that calculates:
824 # b_MX_in_H2O_MEG
825 # from:
826 # mu0_MX_in_H2O_MEG_E_DividedBy_nRT, b_MX_in_H2O, b_MX_in_MEG, x_SF_MEG_in_H2O_MEG and T
827 # =====
828 def fsolve_b_MX_in_H2O_MEG(self, b_MX_in_H2O_MEG, mu0_MX_in_H2O_MEG_E_DividedBy_nRT,
829     b_MX_in_H2O, b_MX_in_MEG, x_SF_MEG_in_H2O_MEG, T):
830     return mu0_MX_in_H2O_MEG_E_DividedBy_nRT - self.f_excess(
831         b_MX_in_H2O_MEG,
832         b_MX_in_H2O,
833         b_MX_in_MEG,
834         x_SF_MEG_in_H2O_MEG,
835         self.A_phi_H2O(T),
836         self.beta0_MX_H2O(T),
837         self.beta1_MX_H2O(T),
838         self.C_phi_MX_H2O(T),
839         self.rho_H2O(T),
840         self.rho_MEG(T),
841         self.epsilon_r_H2O(T),
842         self.epsilon_r_MEG(T),
843         T)[0]
844
845 def runCalculations(self):
846     QApplication.setOverrideCursor(Qt.WaitCursor)
847     self.textEdit_GeneralReport.clear()
848     self.printText(CHARS_NUMBER+'nSTART_OF_READING, SELECTING AND CONVERT'
849         'ING UNITS, EQUILIBRIUM DATA...n'+CHARS_EQUAL+'n')
850     self.readDataset()
851
852     self.printText(CHARS_EQUAL+'nEQUILIBRIUM DATA SELECTED FOR REGRESSION'
853         'n'+CHARS_EQUAL+'n')
854     self.printVectors('x_SF_MEG_in_H2O_MEG', x_SF_MEG_in_H2O_MEG_reg, '',
855         'b_MX_in_H2O_MEG', b_MX_in_H2O_MEG_reg, 'mol.kg-1', 'T', T_reg,
856         'K', 'Reference', [bibref[i] for i in idx_set_reg], '')
857     self.printText(CHARS_MINUS+'nTOTAL POINTS: {0}n'.format(len(
858         x_SF_MEG_in_H2O_MEG_reg)))

```

```

858
859     if x_SF_MEG_in_H2O_MEG_prd != []:
860         self.printText(CHARS_EQUAL+'\\nEQUILIBRIUM_DATA_SELECTED_FOR_PREDIC'
861             'TION\\n'+CHARS_EQUAL+'\\n')
862     self.printVectors('x_SF_MEG_in_H2O_MEG', x_SF_MEG_in_H2O_MEG_prd, '',
863         'b_MX_in_H2O_MEG', b_MX_in_H2O_MEG_prd, 'mol.kg(-1)', 'T', T_prd,
864         'K', 'Reference', [bibref[i] for i in idx_set_prd], '')
865     self.printText(CHARS_MINUS+'\\nTOTAL_POINTS: {0}\\n'.format(len(
866         x_SF_MEG_in_H2O_MEG_prd)))
867
868     self.printText(CHARS_EQUAL+'\\nEND_OF_READING,\\nSELECTING_AND_CONVERTING'
869         '\\nUNITS,EQUILIBRIUM_DATA\\n'+CHARS_NUMBER+'\\n'*3)
870
871     self.printText(CHARS_NUMBER+'\\nSELECTING_EQUILIBRIUM_DATA_AND_REGRESSI'
872         'ON_THEIRS_FOR_SOLUTIONS_WITH_SINGLE_SOLVENT...\\n'+CHARS_EQUAL+
873         '\\n')
874     global b_MX_in_H2O, T_MX_in_H2O, idx_MX_in_H2O
875     global b_MX_in_MEG, T_MX_in_MEG, idx_MX_in_MEG
876     (b_MX_in_H2O, T_MX_in_H2O, idx_MX_in_H2O, b_MX_in_MEG, T_MX_in_MEG,
877         idx_MX_in_MEG) = [], [], [], [], [], []
878     for b, x, T, i in zip(b_MX_in_H2O_MEG_reg, x_SF_MEG_in_H2O_MEG_reg,
879         T_reg, idx_set_reg):
880         if x < 0.01:
881             array(b_MX_in_H2O.append(b))
882             array(T_MX_in_H2O.append(T))
883             array(idx_MX_in_H2O.append(i))
884         elif x > 0.99:
885             array(b_MX_in_MEG.append(b))
886             array(T_MX_in_MEG.append(T))
887             array(idx_MX_in_MEG.append(i))
888     b_MX_in_H2O = array(b_MX_in_H2O)
889     T_MX_in_H2O = array(T_MX_in_H2O)
890     idx_MX_in_H2O = array(idx_MX_in_H2O)
891     b_MX_in_MEG = array(b_MX_in_MEG)
892     T_MX_in_MEG = array(T_MX_in_MEG)
893     idx_MX_in_MEG = array(idx_MX_in_MEG)
894
895     self.printText('EQUILIBRIUM_DATA_SELECTED_FOR_H2O\\n'+CHARS_MINUS+'\\n')
896     self.printVectors('b_MX_in_H2O', b_MX_in_H2O, 'mol.kg(-1)', 'T',
897         T_MX_in_H2O, 'K', 'Reference', [bibref[i] for i in idx_MX_in_H2O],
898         '')
899     fstr_b_MX_in_H2O = 'T,\\n*theta: exp(theta[0]+theta[1]*T+theta[2]*log(T))'
900     self.printText(CHARS_EQUAL+'\\nREGRESSION_FOR_H2O:\\n'+CHARS_MINUS+'\\n')
901     global f_b_MX_in_H2O, theta_b_MX_in_H2O
902     f_b_MX_in_H2O, theta_b_MX_in_H2O = self.n1_regression(fstr_b_MX_in_H2O,
903         T_MX_in_H2O, b_MX_in_H2O, [1,.001,1])
904
905     self.printText(CHARS_EQUAL+'\\nEQUILIBRIUM_DATA_SELECTED_FOR_MEG\\n'+
906         CHARS_MINUS+'\\n')
907     self.printVectors('b_MX_in_MEG', b_MX_in_MEG, 'mol.kg(-1)', 'T',
908         T_MX_in_MEG, 'K', 'Reference', [bibref[i] for i in idx_MX_in_MEG],
909         '')
910     fstr_b_MX_in_MEG = 'T,\\n*theta: exp(theta[0]+theta[1]*T)'
911     self.printText(CHARS_EQUAL+'\\nREGRESSION_FOR_MEG:\\n'+CHARS_MINUS+'\\n')
912     global f_b_MX_in_MEG, theta_b_MX_in_MEG
913     f_b_MX_in_MEG, theta_b_MX_in_MEG = self.n1_regression(fstr_b_MX_in_MEG,
914         T_MX_in_MEG, b_MX_in_MEG, [1,.001])
915
916     self.printText(CHARS_EQUAL+'\\nEQUILIBRIUM_DATA_SELECTED_AND_REGRESSION'
917         '\\nS_FINISHED\\n'+CHARS_NUMBER+'\\n'*3)
918
919     global mu0_MX_in_H2O_MEG_E_DividedBy_nRT_reg
920     global ln_gamma_MX_in_H2O_MEG_E_reg, ln_b_MX_in_H2O_MEG_E_reg
921     global ln_gamma_MX_in_H2O_MEG_reg
922     (mu0_MX_in_H2O_MEG_E_DividedBy_nRT_reg, ln_gamma_MX_in_H2O_MEG_E_reg,
923         ln_b_MX_in_H2O_MEG_E_reg, ln_gamma_MX_in_H2O_MEG_reg) =
924         self.f_excess(b_MX_in_H2O_MEG_reg,
925             f_b_MX_in_H2O(T_reg, *theta_b_MX_in_H2O),
926             f_b_MX_in_MEG(T_reg, *theta_b_MX_in_MEG),
927             x_SF_MEG_in_H2O_MEG_reg,
928             self.A_phi_H2O(T_reg),
929             self.betal_MX_H2O(T_reg),
930             self.C_phi_MX_H2O(T_reg),
931             self.rho_H2O(T_reg),
932             self.rho_MEG(T_reg),
933             self.epsilon_r_H2O(T_reg),
934             self.epsilon_r_MEG(T_reg),
935             T_reg)
936

```

```

937 global mu0_MX_in_H2O_MEG_E_DividedBy_nRT_prd
938 global ln_gamma_MX_in_H2O_MEG_E_prd, ln_b_MX_in_H2O_MEG_E_prd
939 global ln_gamma_MX_in_H2O_MEG_prd
940 (mu0_MX_in_H2O_MEG_E_DividedBy_nRT_prd, ln_gamma_MX_in_H2O_MEG_E_prd,
941 ln_b_MX_in_H2O_MEG_E_prd, ln_gamma_MX_in_H2O_MEG_prd) =
942     self.f_excess(b_MX_in_H2O_MEG_prd,
943 f_b_MX_in_H2O(T_prd, *theta_b_MX_in_H2O),
944 f_b_MX_in_MEG(T_prd, *theta_b_MX_in_MEG),
945 x_SF_MEG_in_H2O_MEG_prd,
946 self.A_phi_H2O(T_prd),
947 self.beta0_MX_H2O(T_prd),
948 self.beta1_MX_H2O(T_prd),
949 self.C_phi_MX_H2O(T_prd),
950 self.rho_H2O(T_prd),
951 self.rho_MEG(T_prd),
952 self.epsilon_r_H2O(T_prd),
953 self.epsilon_r_MEG(T_prd),
954 T_prd)
955
956 self.printText(CHARS_NUMBER+' \nStarting regression Set of Experiments.'
957 '...\n'+CHARS_NUMBER+' \n'*2)
958
959 def f3(X,*a):
960     x,t,f=X[0],X[1],0
961     for i in range(0,len(a),2):
962         f += (a[i]+a[i+1]*t)*x**(i/2)
963     f *= x*(1-x)
964     return f
965
966 theta0 = random.rand(6)*100-200
967 global f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT
968 global theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT
969 (f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT,
970 theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT) = self.n1_regression((
971 f3, 'function'), array([x_SF_MEG_in_H2O_MEG_reg,
972 T_reg]), mu0_MX_in_H2O_MEG_E_DividedBy_nRT_reg, theta0)
973 self.printText(CHARS_NUMBER+' \nFinishing regression Set of Experiments'
974 '...\n'+CHARS_NUMBER+' \n')
975 self.tabWidget_1.setEnabled(True)
976 QApplication.restoreOverrideCursor()
977
978 def selectPoints(self, x_all, T_all, T):
979     return [x_all[i] for i in range(len(T_all)) if abs(T_all[i]-T) < 0.05]
980
981 def limits_xy(self, x1, x2):
982     x_min, x_max = min(min(x1), min(x2)), max(max(x1), max(x2))
983     Delta_x = abs(x_max-x_min)/20.
984     return [x_min-Delta_x, x_max+Delta_x]
985
986 def minmaxPlot(self, x, n=20):
987     xmin, xmax = min(x), max(x)
988     dx = abs(xmax-xmin)/n
989     return xmin-dx, xmax+dx
990
991 def plot1(self):
992     QApplication.setOverrideCursor(Qt.WaitCursor)
993     self.textEdit_Statistics.setText('')
994     # Choice Solvent
995     if self.comboBox_Plot1_Ordinate_Solvent.currentIndex() == 0:
996         ylabel, T_MX_in_solvent, b_MX_in_solvent = 'H_2O', T_MX_in_H2O, b_MX_in_H2O
997         idx_MX_in_solvent, f_rho, f_x_SF = idx_MX_in_H2O, self.rho_H2O, lambda _: 0.0
998         f_b_MX_in_solvent, theta_b_MX_in_solvent = f_b_MX_in_H2O, theta_b_MX_in_H2O
999     elif self.comboBox_Plot1_Ordinate_Solvent.currentIndex() == 1:
1000         ylabel, T_MX_in_solvent, b_MX_in_solvent = 'MEG', T_MX_in_MEG, b_MX_in_MEG
1001         idx_MX_in_solvent, f_rho, f_x_SF = idx_MX_in_MEG, self.rho_MEG, lambda _: 1.0
1002         f_b_MX_in_solvent, theta_b_MX_in_solvent = f_b_MX_in_MEG, theta_b_MX_in_MEG
1003     # Choice Temperature
1004     if self.comboBox_Plot1_Abscissa_Temperature.currentIndex() == 0:
1005         f_T = lambda _: _
1006         xlabel = r'$T_{\left(\rm{K}\right)}$'
1007     elif self.comboBox_Plot1_Abscissa_Temperature.currentIndex() == 1:
1008         f_T = K_to_C
1009         xlabel = r'$T_{\left(^{\circ}\rm{C}\right)}$'
1010     elif self.comboBox_Plot1_Abscissa_Temperature.currentIndex() == 2:
1011         f_T = K_to_R
1012         xlabel = r'$T_{\left(^{\circ}\rm{R}\right)}$'
1013     elif self.comboBox_Plot1_Abscissa_Temperature.currentIndex() == 3:
1014         f_T = K_to_F
1015         xlabel = r'$T_{\left(^{\circ}\rm{F}\right)}$'
1016     # Choice Concentration
1017     if self.comboBox_Plot1_Ordinate_Concentration.currentIndex() == 0:
1018         f_concentration1 = lambda *_: _[0]

```

```

1016         f_concentration2 = lambda _: _
1017         ylabel = r'$b^{\{\rm{'+ylabel+r'\}}}_{\rm{'+MX+r'\}}\left(\rm{mol\dot{kg}^{-1}}\right)$'
1018     elif self.comboBox_Plot1_Ordinate_Concentration.currentIndex() == 1:
1019         f_concentration1 = self.b_MX_in_H2O_MEG_to_c_MX_in_H2O_MEG
1020         f_concentration2 = f_rho
1021         ylabel = r'$c^{\{\rm{'+ylabel+r'\}}}_{\rm{'+MX+r'\}}\left(\rm{mol\dot{kg}^{-1}}\right)$'
1022     elif self.comboBox_Plot1_Ordinate_Concentration.currentIndex() == 2:
1023         f_concentration1 = self.b_MX_in_H2O_MEG_to_x_MX_in_H2O_MEG
1024         f_concentration2 = f_x_SF
1025         ylabel = r'$x^{\{\rm{'+ylabel+r'\}}}_{\rm{'+MX+r'\}}$'
1026     elif self.comboBox_Plot1_Ordinate_Concentration.currentIndex() == 3:
1027         f_concentration1 = self.b_MX_in_H2O_MEG_to_w_MX_in_H2O_MEG
1028         f_concentration2 = lambda _: _
1029         ylabel = r'$w^{\{\rm{'+ylabel+r'\}}}_{\rm{'+MX+r'\}}$'
1030     f_concentration = lambda b, T: f_concentration1(b, f_concentration2(T))
1031     ax = self.mplWidget_Graphic.canvas.ax
1032     ax.clear()
1033     xmin, xmax = self.minmaxPlot(T_MX_in_solvent)
1034     if xmin < 0:
1035         xmin = 0
1036     T = linspace(xmin, xmax, 50)
1037     xcurve = f_T(T)
1038     ycurve = f_concentration(f_b_MX_in_solvent(T, *theta_b_MX_in_solvent), T)
1039     xmin, xmax, ymin, ymax = min(xcurve), max(xcurve), min(ycurve), max(ycurve)
1040     ax.plot(xcurve, ycurve, 'g-', lw=2)
1041     l = list(set(idx_MX_in_solvent))
1042     Ycurve, Ypoints = array([]), array([])
1043     for idx in range(len(l)):
1044         T = array([T for T, i in zip(T_MX_in_solvent, idx_MX_in_solvent) if i == l[idx]])
1045         xpoints = f_T(T)
1046         ypoints = f_concentration(array([b for b, i in zip(b_MX_in_solvent, idx_MX_in_solvent)
1047             if i == l[idx]]), T)
1048         xmin, xmax, ymin, ymax = (min(xmin, min(xpoints)), max(xmax, max(xpoints)),
1049             min(ymin, min(ypoints)), max(ymax, max(ypoints)))
1050         ax.plot(xpoints, ypoints, COLORS_AND_MARKERS[idx])
1051         ycurve = f_concentration(f_b_MX_in_solvent(T, *theta_b_MX_in_solvent), T)
1052         Ycurve, Ypoints = concatenate((Ycurve, ycurve), concatenate((Ypoints, ypoints)))
1053     self.calculateStatistics(Ycurve, Ypoints)
1054     ax.set_xlabel(xlabel, fontsize=16)
1055     ax.set_ylabel(ylabel, fontsize=16)
1056     lstLegend = [bibref[i] for i in l]
1057     lstLegend.insert(0, 'Regression')
1058     ax.legend(lstLegend, loc=0, fontsize=12, ncol=2, fancybox=True, shadow=True)
1059     ax.grid(True)
1060     ax.set_xlim(self.minmaxPlot([xmin, xmax]))
1061     ax.set_ylim(self.minmaxPlot([ymin, ymax]))
1062     self.mplWidget_Graphic.canvas.draw()
1063     self.mplWidget_Graphic.setEnabled(True)
1064     self.checkBox_Graphic.setEnabled(True)
1065     self.pushButton_PrintGraphic.setEnabled(True)
1066     self.textEdit_Statistics.setEnabled(True)
1067     self.pushButton_SaveStatistics.setEnabled(True)
1068     self.pushButton_PrintStatistics.setEnabled(True)
1069     self.tabWidget_1.setCurrentIndex(3)
1070     QApplication.restoreOverrideCursor()
1071
1072     def plot2(self):
1073         QApplication.setOverrideCursor(Qt.WaitCursor)
1074         T_choice = self.T_to_K()
1075         if len(T_choice) < 1:
1076             QApplication.restoreOverrideCursor()
1077             return
1078         self.textEdit_Statistics.setText('')
1079         # Choice Concentration
1080         if self.comboBox_Plot2_Ordinate_Concentration.currentIndex() == 0:
1081             f_concentration1 = lambda *: _[0]
1082             f_concentration2 = lambda *: _[0]
1083             ylabel = r'$b^{\{\rm{H_2O+MEG}\}}_{\rm{'+MX+r'\}}\left(\rm{mol\dot{kg}^{-1}}\right)$'
1084         elif self.comboBox_Plot2_Ordinate_Concentration.currentIndex() == 1:
1085             f_concentration1 = self.b_MX_in_H2O_MEG_to_c_MX_in_H2O_MEG
1086             f_concentration2 = lambda x_SF, T: self.rho_H2O_MEG(self.rho_H2O(T), self.rho_MEG(T),
1087                 x_SF)
1088             ylabel = r'$c^{\{\rm{H_2O+MEG}\}}_{\rm{'+MX+r'\}}\left(\rm{mol\dot{kg}^{-1}}\right)$'
1089         elif self.comboBox_Plot2_Ordinate_Concentration.currentIndex() == 2:
1090             f_concentration1 = self.b_MX_in_H2O_MEG_to_x_MX_in_H2O_MEG
1091             f_concentration2 = lambda *: _[0]
1092             ylabel = r'$x^{\{\rm{H_2O+MEG}\}}_{\rm{'+MX+r'\}}$'
1093         elif self.comboBox_Plot2_Ordinate_Concentration.currentIndex() == 3:
1094             f_concentration1 = self.b_MX_in_H2O_MEG_to_w_MX_in_H2O_MEG
1095             f_concentration2 = lambda *: _[0]

```

```

1095         ylabel = r'$w^{\{\rm{H_2O+MEG}\}}_{\rm{\{+MX+r\}}}$'
1096     f_concentration = lambda b, x_SF, T: f_concentration1(b, f_concentration2(x_SF, T))
1097     # Choice Solvent
1098     if self.comboBox_Plot2_Abscissa_Solvent.currentIndex() == 0:
1099         xlabel, f_x_SF = 'MEG', lambda _: _
1100     elif self.comboBox_Plot2_Abscissa_Solvent.currentIndex() == 1:
1101         xlabel, f_x_SF = 'H_2O', lambda _: 1-_
1102     # Choice Salt-free Concentration
1103     if self.comboBox_Plot2_Abscissa_Concentration.currentIndex() == 0:
1104         f_SF_concentration1 = f_x_SF
1105         f_SF_concentration2 = lambda _: _
1106         xlabel = r'$x^{\{\rm{H_2O+MEG}\}}_{\rm{\{+xlabel+r\}}}$'
1107     elif self.comboBox_Plot2_Abscissa_Concentration.currentIndex() == 1:
1108         f_SF_concentration1 = f_x_SF
1109         f_SF_concentration2 = self.x_SF_MEG_in_H2O_MEG_to_w_SF_MEG_in_H2O_MEG
1110         xlabel = r'$w^{\{\rm{H_2O+MEG}\}}_{\rm{\{+xlabel+r\}}}$'
1111     f_SF_concentration = lambda x_SF: f_SF_concentration1(f_SF_concentration2(x_SF))
1112     # Choice Mode
1113     if self.comboBox_Isotherms_Mode.currentIndex() == 0:
1114         leg_mode, x_all, y_all, T_all = (r'$\rm{Regression}$\_\_\_(' , x_SF_MEG_in_H2O_MEG_reg,
1115         b_MX_in_H2O_MEG_reg, T_reg)
1116     elif self.comboBox_Isotherms_Mode.currentIndex() == 1:
1117         leg_mode, x_all, y_all, T_all = (r'$\rm{Prediction}$\_\_\_(' , x_SF_MEG_in_H2O_MEG_prd,
1118         b_MX_in_H2O_MEG_prd, T_prd)
1119     # Choice Temperature
1120     if self.comboBox_Isotherms_Temperature.currentIndex() == 0:
1121         f_T = lambda _: _
1122         str_T = r'$\_\_\_\rm{K}$'
1123     elif self.comboBox_Isotherms_Temperature.currentIndex() == 1:
1124         f_T = K_to_C
1125         str_T = r'$\_\_\_\rm{C}$'
1126     elif self.comboBox_Isotherms_Temperature.currentIndex() == 2:
1127         f_T = K_to_R
1128         str_T = r'$\_\_\_\rm{R}$'
1129     elif self.comboBox_Isotherms_Temperature.currentIndex() == 3:
1130         f_T = K_to_F
1131         str_T = r'$\_\_\_\rm{F}$'
1132     ax = self.mplWidget_Graphic.canvas.ax
1133     ax.clear()
1134     lstLegend = []
1135     xmin, xmax, ymin, ymax = 0.0, 1.0, None, None
1136     x_SF = linspace(xmin, xmax, 50)
1137     for i, t in enumerate(T_choice):
1138         T = zeros_like(x_SF)+t
1139         mu0_MX_in_H2O_MEG_E_DividedBy_nRT = f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT(
1140             [x_SF, T], *theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT)
1141         b_MX_in_H2O = f_b_MX_in_H2O(T, *theta_b_MX_in_H2O)
1142         b_MX_in_MEG = f_b_MX_in_MEG(T, *theta_b_MX_in_MEG)
1143         b_MX_in_H2O_MEG = fsolve(self.fsolve_b_MX_in_H2O_MEG, b_MX_in_H2O*(1-x_SF)+
1144             b_MX_in_MEG*x_SF, args=(mu0_MX_in_H2O_MEG_E_DividedBy_nRT, b_MX_in_H2O,
1145             b_MX_in_MEG, x_SF, T))
1146         xcurve = f_SF_concentration(x_SF)
1147         ycurve = f_concentration(b_MX_in_H2O_MEG, x_SF, T)
1148         if ymin:
1149             ymin = min(ymin, min(ycurve))
1150         else:
1151             ymin = min(ycurve)
1152         if ymax:
1153             ymax = max(ymax, max(ycurve))
1154         else:
1155             ymax = max(ycurve)
1156         ax.plot(xcurve, ycurve, COLORS_AND_MARKERS[i][0]+'-',lw=2)
1157         lstLegend.append(leg_mode+str(f_T(t))+str_T+r')$')
1158     Ycurve, Ypoints = array([]), array([])
1159     for i, t in enumerate(T_choice):
1160         x_SF = array(self.selectPoints(x_all, T_all, t))
1161         if len(x_SF) > 0:
1162             xpoints = f_SF_concentration(x_SF)
1163             ypoints = f_concentration(array(self.selectPoints(y_all, T_all, t)), x_SF, t)
1164             ymin, ymax = min(ymin, min(ypoints)), max(ymax, max(ypoints))
1165             ax.plot(xpoints, ypoints, COLORS_AND_MARKERS[i])
1166             lstLegend.append(r'$\rm{Experimental}$\_\_\_\rm{points}$\_\_\_(' +str(f_T(t))+str_T+r')$')
1167         # Statistics
1168         T = zeros_like(x_SF)+t
1169         mu0_MX_in_H2O_MEG_E_DividedBy_nRT = f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT(
1170             [x_SF, T], *theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT)
1171         b_MX_in_H2O = f_b_MX_in_H2O(T, *theta_b_MX_in_H2O)
1172         b_MX_in_MEG = f_b_MX_in_MEG(T, *theta_b_MX_in_MEG)
1173         b_MX_in_H2O_MEG = fsolve(self.fsolve_b_MX_in_H2O_MEG, b_MX_in_H2O*(1-x_SF)+
1174             b_MX_in_MEG*x_SF, args=(mu0_MX_in_H2O_MEG_E_DividedBy_nRT, b_MX_in_H2O,

```

```

1175         b_MX_in_MEG, x_SF, T))
1176         ycurve = f_concentration(b_MX_in_H2O_MEG, x_SF, T)
1177         Ycurve, Ypoints = concatenate((Ycurve, ycurve)), concatenate((Ypoints, ypoints))
1178     self.calculateStatistics(Ycurve, Ypoints)
1179     ax.set_xlabel(xlabel, fontsize=16)
1180     ax.set_ylabel(ylabel, fontsize=16)
1181     ax.legend(lstLegend, loc=0, fontsize=12, ncol=2, fancybox=True, shadow=True)
1182     ax.grid(True)
1183     ax.set_xlim(self.minmaxPlot([xmin, xmax]))
1184     ax.set_ylim(self.minmaxPlot([ymin, ymax]))
1185     self.mplWidget_Graphic.canvas.draw()
1186     self.mplWidget_Graphic.setEnabled(True)
1187     self.checkBox_Graphic.setEnabled(True)
1188     self.pushButton_PrintGraphic.setEnabled(True)
1189     self.textEdit_Statistics.setEnabled(True)
1190     self.pushButton_SaveStatistics.setEnabled(True)
1191     self.pushButton_PrintStatistics.setEnabled(True)
1192     self.tabWidget_1.setCurrentIndex(3)
1193     QApplication.restoreOverrideCursor()
1194
1195     def plot3(self):
1196         QApplication.setOverrideCursor(Qt.WaitCursor)
1197         T_choice = self.T_to_K()
1198         if len(T_choice) < 1:
1199             QApplication.restoreOverrideCursor()
1200             return
1201         self.textEdit_Statistics.setText('')
1202         # Choice Function
1203         if self.comboBox_Plot3_Ordinate_Function.currentIndex() == 0:
1204             f_function = lambda _: exp(_)
1205             ylabel = r'$\gamma^{\{\rm{H_2O+MEG}\}}_{\rm{+MX+r}}$'
1206         elif self.comboBox_Plot3_Ordinate_Function.currentIndex() == 1:
1207             f_function = lambda _: _
1208             ylabel = r'$\ln \gamma^{\{\rm{H_2O+MEG}\}}_{\rm{+MX+r}}$'
1209         # Choice Solvent
1210         if self.comboBox_Plot3_Abscissa_Solvent.currentIndex() == 0:
1211             xlabel, f_x_SF = 'MEG', lambda _: _
1212         elif self.comboBox_Plot3_Abscissa_Solvent.currentIndex() == 1:
1213             xlabel, f_x_SF = 'H_2O', lambda _: 1-_
1214         # Choice Salt-free Concentration
1215         if self.comboBox_Plot3_Abscissa_Concentration.currentIndex() == 0:
1216             f_SF_concentration1 = f_x_SF
1217             f_SF_concentration2 = lambda _: _
1218             xlabel = r'$x^{\{\rm{H_2O+MEG}\}}_{\rm{+xlabel+r}}$'
1219         elif self.comboBox_Plot3_Abscissa_Concentration.currentIndex() == 1:
1220             f_SF_concentration1 = f_x_SF
1221             f_SF_concentration2 = self.x_SF_MEG_in_H2O_MEG_to_w_SF_MEG_in_H2O_MEG
1222             xlabel = r'$w^{\{\rm{H_2O+MEG}\}}_{\rm{+xlabel+r}}$'
1223         f_SF_concentration = lambda x_SF: f_SF_concentration1(f_SF_concentration2(x_SF))
1224         # Choice Mode
1225         if self.comboBox_Isotherms_Mode.currentIndex() == 0:
1226             leg_mode, x_all, y_all, T_all = (r'$\rm{Regression}_{\rm{+x_SF_MEG_in_H2O_MEG_reg,}}$
1227             ln_gamma_MX_in_H2O_MEG_reg, T_reg)
1228         elif self.comboBox_Isotherms_Mode.currentIndex() == 1:
1229             leg_mode, x_all, y_all, T_all = (r'$\rm{Prediction}_{\rm{+x_SF_MEG_in_H2O_MEG_prd,}}$
1230             ln_gamma_MX_in_H2O_MEG_prd, T_prd)
1231         # Choice Temperature
1232         if self.comboBox_Isotherms_Temperature.currentIndex() == 0:
1233             f_T = lambda _: _
1234             str_T = r'$\rm{K}$'
1235         elif self.comboBox_Isotherms_Temperature.currentIndex() == 1:
1236             f_T = K_to_C
1237             str_T = r'$\rm{C}$'
1238         elif self.comboBox_Isotherms_Temperature.currentIndex() == 2:
1239             f_T = K_to_R
1240             str_T = r'$\rm{R}$'
1241         elif self.comboBox_Isotherms_Temperature.currentIndex() == 3:
1242             f_T = K_to_F
1243             str_T = r'$\rm{F}$'
1244         ax = self.mplWidget_Graphic.canvas.ax
1245         ax.clear()
1246         lstLegend = []
1247         xmin, xmax, ymin, ymax = 0.0, 1.0, None, None
1248         x_SF = linspace(xmin, xmax, 50)
1249         for i, t in enumerate(T_choice):
1250             T = zeros_like(x_SF)+t
1251             mu0_MX_in_H2O_MEG_E_DividedBy_nRT = f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT(
1252             [x_SF, T], *theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT)
1253             b_MX_in_H2O = f_b_MX_in_H2O(T, *theta_b_MX_in_H2O)
1254             b_MX_in_MEG = f_b_MX_in_MEG(T, *theta_b_MX_in_MEG)

```



```

1255     b_MX_in_H2O_MEG = fsolve(self.fsolve_b_MX_in_H2O_MEG, b_MX_in_H2O*(1-x_SF)+
1256         b_MX_in_MEG*x_SF, args=(mu0_MX_in_H2O_MEG_E_DividedBy_nRT, b_MX_in_H2O,
1257         b_MX_in_MEG, x_SF, T))
1258     A_phi_H2O = self.A_phi_H2O(T)
1259     beta0_MX_H2O = self.beta0_MX_H2O(T)
1260     beta1_MX_H2O = self.beta1_MX_H2O(T)
1261     C_phi_MX_H2O = self.C_phi_MX_H2O(T)
1262     F_excess = self.f_excess(b_MX_in_H2O_MEG, b_MX_in_H2O, b_MX_in_MEG,
1263         x_SF, A_phi_H2O, beta0_MX_H2O, beta1_MX_H2O,
1264         C_phi_MX_H2O, self.rho_H2O(T), self.rho_MEG(T),
1265         self.epsilon_r_H2O(T), self.epsilon_r_MEG(T), T)
1266     ln_gamma_MX_in_H2O = self.ln_gamma_MX(
1267         b_MX_in_H2O,
1268         self.f_gamma_MX(A_phi_H2O, self.X_gamma(self.I_b(b_MX_in_H2O))),
1269         self.B_gamma_MX(beta0_MX_H2O, beta1_MX_H2O, self.Y_gamma(self.I_b(
1270             b_MX_in_H2O))),
1271         self.C_gamma_MX(C_phi_MX_H2O))
1272     ln_gamma_MX_in_MEG = self.ln_gamma_MX_in_MEG(ln_gamma_MX_in_H2O,
1273         b_MX_in_H2O, b_MX_in_MEG, T)
1274     xcurve = f_SF_concentration(x_SF)
1275     ycurve = f_function(F_excess[1]+self.p12_ideal(ln_gamma_MX_in_H2O, ln_gamma_MX_in_MEG,
1276         x_SF))
1277     if ymin:
1278         ymin = min(ymin, min(ycurve))
1279     else:
1280         ymin = min(ycurve)
1281     if ymax:
1282         ymax = max(ymax, max(ycurve))
1283     else:
1284         ymax = max(ycurve)
1285     ax.plot(xcurve, ycurve, COLORS_AND_MARKERS[i][0]+'-', lw=2)
1286     lstLegend.append((leg_mode+str(f_T(t))+str_T+r')$'))
1287     Ycurve, Ypoints = array([], array([])
1288     for i, t in enumerate(T_choice):
1289         x_SF = array(self.selectPoints(x_all, T_all, t))
1290         if len(x_SF) > 0:
1291             xpoints = f_SF_concentration(x_SF)
1292             ypoints = f_function(array(self.selectPoints(y_all, T_all, t)))
1293             ymin, ymax = min(ymin, min(ypoints)), max(ymax, max(ypoints))
1294             ax.plot(xpoints, ypoints, COLORS_AND_MARKERS[i])
1295             lstLegend.append(r'$\rm{Experimental\_points\_}(' +str(f_T(t))+str_T+r')$')
1296             # Statistics
1297             T = zeros_like(x_SF)+t
1298             mu0_MX_in_H2O_MEG_E_DividedBy_nRT = f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT(
1299                 [x_SF, T], *theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT)
1300             b_MX_in_H2O = f_b_MX_in_H2O(T, *theta_b_MX_in_H2O)
1301             b_MX_in_MEG = f_b_MX_in_MEG(T, *theta_b_MX_in_MEG)
1302             b_MX_in_H2O_MEG = fsolve(self.fsolve_b_MX_in_H2O_MEG, b_MX_in_H2O*(1-x_SF)+
1303                 b_MX_in_MEG*x_SF, args=(mu0_MX_in_H2O_MEG_E_DividedBy_nRT, b_MX_in_H2O,
1304                 b_MX_in_MEG, x_SF, T))
1305             A_phi_H2O = self.A_phi_H2O(T)
1306             beta0_MX_H2O = self.beta0_MX_H2O(T)
1307             beta1_MX_H2O = self.beta1_MX_H2O(T)
1308             C_phi_MX_H2O = self.C_phi_MX_H2O(T)
1309             F_excess = self.f_excess(b_MX_in_H2O_MEG, b_MX_in_H2O, b_MX_in_MEG,
1310                 x_SF, A_phi_H2O, beta0_MX_H2O, beta1_MX_H2O,
1311                 C_phi_MX_H2O, self.rho_H2O(T), self.rho_MEG(T),
1312                 self.epsilon_r_H2O(T), self.epsilon_r_MEG(T), T)
1313             ln_gamma_MX_in_H2O = self.ln_gamma_MX(
1314                 b_MX_in_H2O,
1315                 self.f_gamma_MX(A_phi_H2O, self.X_gamma(self.I_b(b_MX_in_H2O))),
1316                 self.B_gamma_MX(beta0_MX_H2O, beta1_MX_H2O, self.Y_gamma(self.I_b(
1317                     b_MX_in_H2O))),
1318                 self.C_gamma_MX(C_phi_MX_H2O))
1319             ln_gamma_MX_in_MEG = self.ln_gamma_MX_in_MEG(ln_gamma_MX_in_H2O,
1320                 b_MX_in_H2O, b_MX_in_MEG, T)
1321             ycurve = f_function(F_excess[1]+self.p12_ideal(ln_gamma_MX_in_H2O,
1322                 ln_gamma_MX_in_MEG, x_SF))
1323             Ycurve, Ypoints = concatenate((Ycurve, ycurve)), concatenate((Ypoints, ypoints))
1324     self.calculateStatistics(Ycurve, Ypoints)
1325     ax.set_xlabel(xlabel, fontsize=16)
1326     ax.set_ylabel(ylabel, fontsize=16)
1327     ax.legend(lstLegend, loc=0, fontsize=12, ncol=2, fancybox=True, shadow=True)
1328     ax.grid(True)
1329     ax.set_xlim(self.minmaxPlot([xmin, xmax]))
1330     ax.set_ylim(self.minmaxPlot([ymin, ymax]))
1331     self.mplWidget_Graphic.canvas.draw()
1332     self.mplWidget_Graphic.setEnabled(True)
1333     self.checkBox_Graphic.setEnabled(True)
1334     self.pushButton_PrintGraphic.setEnabled(True)

```

```

1334 self.textEdit_Statistics.setEnabled(True)
1335 self.pushButton_SaveStatistics.setEnabled(True)
1336 self.pushButton_PrintStatistics.setEnabled(True)
1337 self.tabWidget_1.setCurrentIndex(3)
1338 QApplication.restoreOverrideCursor()
1339
1340 def plot4(self):
1341     QApplication.setOverrideCursor(Qt.WaitCursor)
1342     T_choice = self.T_to_K()
1343     if len(T_choice) < 1:
1344         QApplication.restoreOverrideCursor()
1345         return
1346     self.textEdit_Statistics.setText('')
1347     # Choice Excess Function
1348     if self.comboBox_Plot4_Ordinate_ExcessFunction.currentIndex() == 0:
1349         y_all = [mu0_MX_in_H2O_MEG_E_DividedBy_nRT_reg, mu0_MX_in_H2O_MEG_E_DividedBy_nRT_prd]
1350         ylabel = r'$\left[\mu^0\{\rm{H_2O+MEG}}\}_{\rm{+MX+r}}\right]^{\rm{E}}_{\rm{RT}}$'
1351     elif self.comboBox_Plot4_Ordinate_ExcessFunction.currentIndex() == 1:
1352         y_all = [ln_gamma_MX_in_H2O_MEG_E_reg, ln_gamma_MX_in_H2O_MEG_E_prd]
1353         ylabel = r'$\left[\ln\gamma^{\rm{H_2O+MEG}}_{\rm{+MX+r}}\right]^{\rm{E}}$'
1354     elif self.comboBox_Plot4_Ordinate_ExcessFunction.currentIndex() == 2:
1355         y_all = [ln_b_MX_in_H2O_MEG_E_reg, ln_b_MX_in_H2O_MEG_E_prd]
1356         ylabel = r'$\left[\ln b^{\rm{H_2O+MEG}}_{\rm{+MX+r}}\right]^{\rm{E}}$'
1357     # Choice Solvent
1358     if self.comboBox_Plot4_Abscissa_Solvent.currentIndex() == 0:
1359         xlabel, f_x_SF = 'MEG', lambda _: _
1360     elif self.comboBox_Plot4_Abscissa_Solvent.currentIndex() == 1:
1361         xlabel, f_x_SF = 'H_2O', lambda _: 1-_
1362     # Choice Salt-free Concentration
1363     if self.comboBox_Plot4_Abscissa_Concentration.currentIndex() == 0:
1364         f_SF_concentration1 = f_x_SF
1365         f_SF_concentration2 = lambda _: _
1366         xlabel = r'$x^{\rm{H_2O+MEG}}_{\rm{+xlabel+r}}$'
1367     elif self.comboBox_Plot4_Abscissa_Concentration.currentIndex() == 1:
1368         f_SF_concentration1 = f_x_SF
1369         f_SF_concentration2 = self.x_SF_MEG_in_H2O_MEG_to_w_SF_MEG_in_H2O_MEG
1370         xlabel = r'$w^{\rm{H_2O+MEG}}_{\rm{+xlabel+r}}$'
1371     f_SF_concentration = lambda x_SF: f_SF_concentration1(f_SF_concentration2(x_SF))
1372     # Choice Mode
1373     if self.comboBox_Isotherms_Mode.currentIndex() == 0:
1374         leg_mode, x_all, y_all, T_all = (r'$\rm{Regression}$', x_SF_MEG_in_H2O_MEG_reg,
1375         y_all[0], T_reg)
1376     elif self.comboBox_Isotherms_Mode.currentIndex() == 1:
1377         leg_mode, x_all, y_all, T_all = (r'$\rm{Prediction}$', x_SF_MEG_in_H2O_MEG_prd,
1378         y_all[1], T_prd)
1379     # Choice Temperature
1380     if self.comboBox_Isotherms_Temperature.currentIndex() == 0:
1381         f_T = lambda _: _
1382         str_T = r'$\rm{K}$'
1383     elif self.comboBox_Isotherms_Temperature.currentIndex() == 1:
1384         f_T = K_to_C
1385         str_T = r'$\circ\rm{C}$'
1386     elif self.comboBox_Isotherms_Temperature.currentIndex() == 2:
1387         f_T = K_to_R
1388         str_T = r'$\circ\rm{R}$'
1389     elif self.comboBox_Isotherms_Temperature.currentIndex() == 3:
1390         f_T = K_to_F
1391         str_T = r'$\circ\rm{F}$'
1392     ax = self.mplWidget_Graphic.canvas.ax
1393     ax.clear()
1394     lstLegend = []
1395     xmin, xmax, ymin, ymax = 0.0, 1.0, None, None
1396     x_SF = linspace(xmin, xmax, 50)
1397     for i, t in enumerate(T_choice):
1398         T = zeros_like(x_SF)+t
1399         mu0_MX_in_H2O_MEG_E_DividedBy_nRT = f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT(
1400         [x_SF, T], *theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT)
1401         b_MX_in_H2O = f_b_MX_in_H2O(T, *theta_b_MX_in_H2O)
1402         b_MX_in_MEG = f_b_MX_in_MEG(T, *theta_b_MX_in_MEG)
1403         b_MX_in_H2O_MEG = fsolve(self.fsolve_b_MX_in_H2O_MEG, b_MX_in_H2O*(1-x_SF)+
1404         b_MX_in_MEG*x_SF, args=(mu0_MX_in_H2O_MEG_E_DividedBy_nRT, b_MX_in_H2O,
1405         b_MX_in_MEG, x_SF, T))
1406         A_phi_H2O = self.A_phi_H2O(T)
1407         beta0_MX_H2O = self.beta0_MX_H2O(T)
1408         beta1_MX_H2O = self.beta1_MX_H2O(T)
1409         C_phi_MX_H2O = self.C_phi_MX_H2O(T)
1410         F_excess = self.f_excess(b_MX_in_H2O_MEG, b_MX_in_H2O, b_MX_in_MEG,
1411         x_SF, A_phi_H2O, beta0_MX_H2O, beta1_MX_H2O,
1412         C_phi_MX_H2O, self.rho_H2O(T), self.rho_MEG(T),
1413         self.epsilon_r_H2O(T), self.epsilon_r_MEG(T), T)

```

```

1414 xcurve = f_SF_concentration(x_SF)
1415 ycurve = F_excess[self.comboBox_Plot4_Ordinate_ExcessFunction.currentIndex()]
1416 if ymin:
1417     ymin = min(ymin, min(ycurve))
1418 else:
1419     ymin = min(ycurve)
1420 if ymax:
1421     ymax = max(ymax, max(ycurve))
1422 else:
1423     ymax = max(ycurve)
1424 ax.plot(xcurve, ycurve, COLORS_AND_MARKERS[i][0]+'-',lw=2)
1425 lstLegend.append(leg_mode+str(f_T(t))+str_T+r')$')
1426 Ycurve, Ypoints = array([]), array([])
1427 for i, t in enumerate(T_choice):
1428     x_SF = array(self.selectPoints(x_all, T_all, t))
1429     if len(x_SF) > 0:
1430         xpoints = f_SF_concentration(x_SF)
1431         ypoints = array(self.selectPoints(y_all, T_all, t))
1432         ymin, ymax = min(ymin, min(ypoints)), max(ymax, max(ypoints))
1433         ax.plot(xpoints, ypoints, COLORS_AND_MARKERS[i])
1434         lstLegend.append(r'$\rm{Experimental\_points\_}\{'+str(f_T(t))+str_T+r'\}$')
1435         # Statistics
1436         T = zeros_like(x_SF)+t
1437         mu0_MX_in_H2O_MEG_E_DividedBy_nRT = f_mu0_MX_in_H2O_MEG_E_DividedBy_nRT(
1438             [x_SF, T], *theta_mu0_MX_in_H2O_MEG_E_DividedBy_nRT)
1439         b_MX_in_H2O = f_b_MX_in_H2O(T, *theta_b_MX_in_H2O)
1440         b_MX_in_MEG = f_b_MX_in_MEG(T, *theta_b_MX_in_MEG)
1441         b_MX_in_H2O_MEG = fsolve(self.fsolve_b_MX_in_H2O_MEG, b_MX_in_H2O*(1-x_SF)+
1442             b_MX_in_MEG*x_SF, args=(mu0_MX_in_H2O_MEG_E_DividedBy_nRT, b_MX_in_H2O,
1443             b_MX_in_MEG, x_SF, T))
1444         A_phi_H2O = self.A_phi_H2O(T)
1445         beta0_MX_H2O = self.beta0_MX_H2O(T)
1446         beta1_MX_H2O = self.beta1_MX_H2O(T)
1447         C_phi_MX_H2O = self.C_phi_MX_H2O(T)
1448         F_excess = self.f_excess(b_MX_in_H2O_MEG, b_MX_in_H2O, b_MX_in_MEG,
1449             x_SF, A_phi_H2O, beta0_MX_H2O, beta1_MX_H2O,
1450             C_phi_MX_H2O, self.rho_H2O(T), self.rho_MEG(T),
1451             self.epsilon_r_H2O(T), self.epsilon_r_MEG(T), T)
1452         ycurve = F_excess[self.comboBox_Plot4_Ordinate_ExcessFunction.currentIndex()]
1453         Ycurve, Ypoints = concatenate((Ycurve, ycurve)), concatenate((Ypoints, ypoints))
1454     self.calculateStatistics(Ycurve, Ypoints)
1455     ax.set_xlabel(xlabel, fontsize=16)
1456     ax.set_ylabel(ylabel, fontsize=16)
1457     ax.legend(lstLegend, loc=0, fontsize=12, ncol=2, fancybox=True, shadow=True)
1458     ax.grid(True)
1459     ax.set_xlim(self.minmaxPlot([xmin, xmax]))
1460     ax.set_ylim(self.minmaxPlot([ymin, ymax]))
1461     self.mplWidget_Graphic.canvas.draw()
1462     self.mplWidget_Graphic.setEnabled(True)
1463     self.checkBox_Graphic.setEnabled(True)
1464     self.pushButton_PrintGraphic.setEnabled(True)
1465     self.textEdit_Statistics.setEnabled(True)
1466     self.pushButton_SaveStatistics.setEnabled(True)
1467     self.pushButton_PrintStatistics.setEnabled(True)
1468     self.tabWidget_1.setCurrentIndex(3)
1469     QApplication.restoreOverrideCursor()
1470
1471 def main():
1472     app = QApplication(sys.argv)
1473     __ = JAFOSMS()
1474     sys.exit(app.exec_())
1475
1476 if __name__ == '__main__':
1477     main()

```