

DESENVOLVIMENTO DE CÓDIGO PARA SIMULADOR DE FORMULAÇÃO DE SISTEMAS PARA PRODUÇÃO DE EMULSÕES COM HLD CONHECIDO, PARA UMA E DUAS VARIÁVEIS DESCONHECIDAS. CÁLCULO DE PARÂMETROS DE FORMULAÇÃO DE INTERESSE.

Palavras-Chave: COALESCÊNCIA, INVERSÃO DE FASES, EMULSÕES

Autores:

Juliana Mayumi Millani Doi FEQ-Unicamp

Prof. Dr. Antonio Carlos Bannwart (orientador) FEM-Unicamp

Rodrigo Berini Tamaki FEQ-Unicamp

Dr. Carlos Eduardo Perles CEPETRO-Unicamp

Dra. Vanessa Cristina Bizotto Guersoni CEPETRO-Unicamp

INTRODUÇÃO:

O projeto de pesquisa tem como objetivo desenvolver um código em Python para o cálculo de um parâmetro termodinâmico conhecido como HLD (*Hydrophilic/Lipophilic Difference*) para o desenvolvimento de formulação de emulsões para estudos sobre o ponto de inversão de emulsões petróleo/água.

O desenvolvimento de formulações é uma etapa essencial na investigação da influência da composição das fases aquosa e óleo no fenômeno de coalescência de emulsões do tipo água em óleo (A/O) ou óleo em água (O/A), visto que as propriedades das emulsões, como viscosidade, presença de eletrólitos na fase aquosa, entre outros, e fenômenos como a inversão de fases, se alteram com a formulação do sistema. A formação de emulsões durante a produção pode dificultar o bombeamento do petróleo e até causar corrosões nas tubulações. Assim, com esta ferramenta, o estudo das formulações de emulsões é facilitado, já que é possível calcular parâmetros desconhecidos de um sistema químico com base na entrada de alguns parâmetros conhecidos, como o valor de HLD de interesse e pelo menos duas variáveis independentes, sendo o código responsável por calcular uma variável independente ou possíveis combinações de duas variáveis independentes necessárias para que a formulação apresente o valor requisitado de HLD.

Neste trabalho, o código foi escrito para o cálculo de até duas variáveis desconhecidas das equações de HLD, para surfactantes iônicos e não-iônicos, Equações 1 e 2, respectivamente, visando a formulação de sistemas com valores de HLD previamente definidos.

$$HLD = \ln S - kEACN + Cc + f(A) - c_T(T - T_{ref}) \quad (1)$$

$$HLD = bS - kEACN + Cc + f(A) + c_T(T - T_{ref}) \quad (2)$$

$$f(A) = a * A \quad (3)$$

Onde: “S” é a salinidade do sistema (g/100 mL de água); “EACN” é o número de carbonos efetivo do alcão; “Cc” é a curva característica do surfactante; “k” e “cT” são constantes relacionadas ao(s) surfactante(s) utilizado(s), “T” é a temperatura do sistema (°C), “Tref” é dado por 25°C e “f(A)” é uma função do álcool, representada pela equação 3, em que “a” é uma constante referente ao álcool presente no sistema e “A” é sua concentração em peso (%). (SALAGER, 2006)

O código do simulador foi desenvolvido na forma de linguagem de texto, será capaz de agilizar o processo de formulação de sistemas de emulsões, além de fornecer respostas mais precisas que o tradicional processo de cálculo por tentativa e erro através de planilhas. Este projeto de pesquisa foi desenvolvido dentro de uma das linhas de Pesquisa do Energy Production Innovation Center (EPIC), uma parceria entre EQUINOR, FAPESP e UNICAMP.

METODOLOGIA:

Como primeira estratégia, foi considerado o uso da ferramenta *Solver* do Excel para a realização dos cálculos das diferentes variáveis das equações do HLD, já que por meio de um chute inicial, seria possível a obtenção dos resultados. Para o cálculo de uma variável, a ferramenta se mostrou precisa, porém já para duas variáveis, sua convergência foi dificultada, visto que dependendo dos chutes iniciais que eram feitos para o valor da variável, obtinha-se respostas diferentes, não se mostrando uma forma consistente de se realizar os cálculos. Dessa maneira, a utilização da ferramenta *Solver* foi descartada já nas primeiras etapas.

Com isso, optou-se pelo uso da linguagem de programação em Python, para a construção do código do programa, uma vez que ele possui bibliotecas que facilitam a realização de cálculos e a plotagem de gráficos, como o *math* e o *matplotlib*, respectivamente. Vale ressaltar que o gráfico será necessário apenas para a parte do cálculo de duas variáveis, pois será ele que permitirá a visualização dos resultados obtidos.

Ademais, ressalta-se que uma base de dados disponibilizada por Salager (2001) foi utilizada para a montagem de tabelas de consulta para a utilização do usuário, as quais apresentam constantes de diferentes surfactantes e óleos necessários para os cálculos.

RESULTADOS E DISCUSSÃO:

Tendo em vista a complexidade dos cálculos para duas variáveis, inicialmente o código foi desenvolvido para o cálculo de apenas uma variável. Sua construção se baseia nos vários comandos que o Python possui, sendo os principais utilizados os comandos condicionais (*if*, *elif*, *else*), os de repetição (*while*, *for*) e os de funções (*def*). Um exemplo disso pode ser visualizado na figura 1, em que a função “alcohol()” é responsável por definir o valor da função de álcool $f(A)$, caso a variável a ser calculada seja diferente da concentração de álcool no sistema (A), ou definir o valor da constante do álcool (a), caso se queira calcular sua concentração.

```
505 def alcohol():
506     if SeleccionadorVariavel() != "A":
507         if textoalcohol != "Sem presença de álcool":
508             a = tipo_de_alcohol()
509             A = alcohol_concentraçao()
510             f_A = a * A
511             return f_A
512         elif textoalcohol == "Sem presença de álcool":
513             f_A = 0
514             return f_A
515     if SeleccionadorVariavel() == "A":
516         a = tipo_de_alcohol()
517         return a
```

Figura 1: Exemplificação de alguns comandos utilizados no desenvolvimento do código.

A partir disso, ao se rodar o código, automaticamente as tabelas de consultas com constantes de diversos tipos de surfactantes e de valores de número de carbono efetivo (EACN) em óleos de surfactantes poderão ser visualizadas, como é mostrado nas figuras 2 e 3, respectivamente, já que foram incluídas a ele com o comando *print*, responsável por exibir as mensagens na tela.

Surfactante	sigma	k	Ct
iso-C9 Benzeno Sulfonato de Sódio:	-1.6	0.16	-0.01
iso-C12 Benzeno Sulfonato de Sódio:	-0.6	0.16	-0.01
iso-C12 o-xilenosulfonato de Sódio:	0.5	0.16	-0.01
iso-C12 p-benzoato de sódio:	0.6	0.16	-0.01
C12 carboxilato de sódio:	-2.4	0.1	-0.01
C12 Sulfato de Sódio:	-3.0	0.1	-0.01
C12 amônio, cloreto (ph3):	-0.8	0.17	-0.02
C12 trimetilamônio, cloreto:	-3.6	0.19	-0.02
C16 trimetilamônio, cloreto:	-1.7	0.19	-0.02
iso-C8 fenoletoxilado:	6.2	0.15	0.06
iso-C9 fenoletoxilado:	6.5	0.15	0.06
iso-C12 fenoletoxilado:	7.3	0.15	0.06
iso-C18 fenoletoxilado:	9.5	0.15	0.06
C10 álcool etoxilado:	5.3	0.15	
C12 álcool etoxilado:	5.9	0.15	
iso-C13 álcool etoxilado:	4.3	0.15	0.06
C14 álcool etoxilado:	6.6	0.15	

Figura 2: Tabela de surfactantes e suas respectivas constantes.

Óleo	EACN
Óleo de parafina:	20 à 25
Óleo de soja:	18
Triglicerídeo C8-C10:	15
Diglicerídeo C10-C12:	9
Alcano linear de n carbonos:	n
Querosene:	9 à 11
Limonene:	9
Pinene:	7.500000
Etil oleato:	6.500000
Cicloexano:	3.500000
Tetracloroetileno:	3
p-Xileno:	2
Tolueno:	1
Benzeno:	0
Tetraclorometano:	0
1,1,1-Tricloroetano:	-2
Tricloroetileno:	-4
1,2-Diclorobenzeno:	-5
1,2-Dicloroetano:	-12
Diclorometano:	-14
Clorofórmio:	-14
1,1,2,2-Tetracloroetano:	-22

Figura 3: Tabela de óleos e seus respectivos valores de EACN.

Ademais, junto às tabelas haverá uma mensagem para a seleção da variável a ser calculada, como é vista a partir da figura 4. Com isso, deve-se incluir o *input* com a variável desejada, e com isso, mais mensagens surgirão, como o tipo de surfactante, para se definir a equação a ser utilizada no cálculo, o valor de HLD requerido para a formulação, o sal presente na formulação, a presença (ou não) de álcool e de misturas de óleos e surfactantes, de forma a se adicionar ao programa informações sobre a formulação requerida, possibilitando a realização do cálculo. Assim, pode ser observado nas figuras abaixo, um exemplo feito para o cálculo da temperatura, em que na formulação considerada o tipo de surfactante utilizado é um etoxilato, o sal é dado pelo NaCl, há uma mistura de óleos e surfactantes no sistema, além da presença de álcool. É importante destacar que quando se há misturas de óleos e/ou de surfactantes e as variáveis a serem

```

Digite uma única variável da lista a seguir para ser encontrada:
S
EACN
Cc
A
T

Escolha: 1
Informe o tipo de surfactante: iônico, etoxilato ou APGs: etoxilato
Informe o valor de HLD a ser atingido, HLD: 6

Escolha o sal utilizado no sistema
1 | NaCl
2 | CaCl2
3 | KCl
4 | Sem presença de sal
Escolha: 1
Informe o valor da salinidade, S (g/100 mL): 20
  
```

```

Há uma mistura de óleos no sistema?
1 - Sim
2 - Não
Escolha: 1
Informe o valor do EACN1 da mistura: 10
Informe a fração molar x1 da mistura: 0.4
Informe o valor do EACN2 da mistura: 15
Informe a fração molar x2 da mistura: 0.6

Há uma mistura de surfactantes no sistema?
1 - Sim
2 - Não
Escolha: 1
Informe o valor do Cc1 do primeiro surfactante: 1.4
Informe a massa molar "MW1" do primeiro surfactante (g/mol): 1300
Informe a fração molar x1 do primeiro surfactante: 0.4
Informe o valor do Cc2 do segundo surfactante: 0.4
Informe a massa molar "MW2" do segundo surfactante (g/mol): 1300
Informe a fração molar x2 do segundo surfactante: 0.6

Há álcool no sistema?
1 - Sim
2 - Não
Escolha: 1
  
```

Figuras 4 e 5: Execução do código para 1 variável.

encontradas não são nem o EACN do óleo e nem a curva característica (Cc) do surfactante, o código precisará ser informado sobre valores de suas respectivas frações molares e massas molares (no caso de surfactantes), como também é mostrado nas figuras.

Posteriormente, o código para o cálculo de duas variáveis foi desenvolvido baseando-se no código de uma variável. Dessa maneira, tem-se como a principal diferença entre os dois sendo que para o cálculo da combinação de duas variáveis, foi necessária a criação de laços de repetição, a partir dos comandos *for* e *while*, para que por meio de uma limitação de valores para a variável 1, a cada passo, ou seja, acréscimo de valor ao limite mínimo, fossem calculados os respectivos valores resultantes para a variável 2, até com que se alcançasse o limite máximo da primeira variável, calculando-se uma última combinação. Vale ressaltar que cada combinação calculada é guardada em forma de listas, as quais são utilizadas para a plotagem dos resultados na forma de gráfico para a visualização do usuário. Um exemplo para essa parte comentada pode ser visualizado por meio da figura 6.

```
if variavel_1 == "T" or variavel_2=="S":
    #limites para a temperatura
    lim_i = int(input('Informe o limite inferior para a temperatura T (°C): '))
    lim_s = int(input('Informe o limite superior para a temperatura T (°C): '))
    listaT = []
    #HLD=math.log(S)-k*EACN+Cc+f_A+Ct*(T-Tref)

    if variavel_2=="S" or variavel_1=="S":
        EACN=númerodecarbonoefetivo()
        Cc=númerodesurfactantes()
        f_A=alcool()
        listaS=[]
        for T in range(lim_i, lim_s, 1):
            S = funcao_S()
            listaT.append(T)
            listaS.append(S)
        plt.plot(listaT,listaS,"bo")
        plt.title('Salinidade x Temperatura')
        plt.xlabel("Temperatura (°C)")
        plt.ylabel("Salinidade (g/100 mL)")
        plt.show()
```

Figura 6: Exemplo das limitações feitas com a utilização de laços e listas e plotagem de gráficos.

Dessa forma, caso fosse desejado calcular além da temperatura, mas também a salinidade, para determinados *inputs*, obteríamos um gráfico, como mostra a figura 7, com todas as possíveis combinações de resultados, que foram limitados entre valores mínimos e máximos de temperatura.

Essa limitação mencionada foi feita priorizando as variáveis que menos influenciavam nos resultados. Ou seja, por exemplo, para a combinação de temperatura e salinidade, foi decidido se limitar a temperatura, pois pequenas variações de salinidade causam altos impactos na formulação, porém, a salinidade em detrimento com a curva característica do surfactante, a salinidade é a variável plausível de ser limitada, já que o sistema pode ter mais de um surfactante, tornando seu cálculo mais extenso. Dessa maneira, as variáveis priorizadas para serem limitadas seguiram a seguinte ordem: temperatura > salinidade > concentração de álcool > EACN.

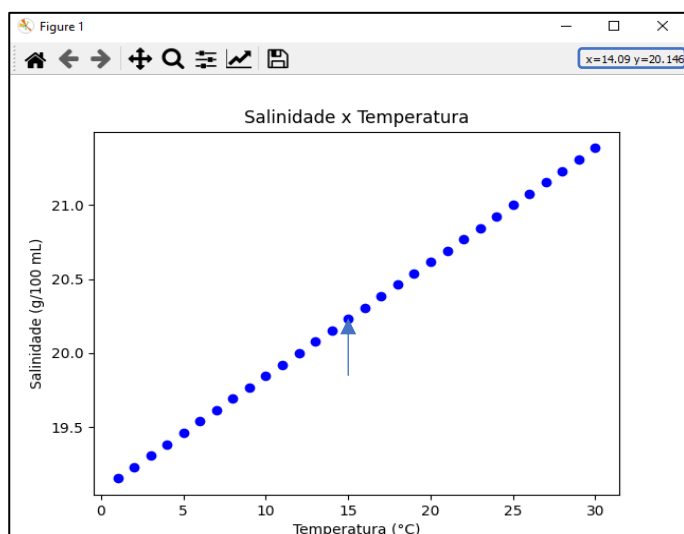


Figura 7: Resultados do cálculo para as variáveis de temperatura e salinidade apresentados através de um gráfico.

Optou-se pela apresentação da relação entre as variáveis na forma gráfica em detrimento da apresentação em tabela devido à facilidade de visualização dos resultados, bastando apenas posicionar a seta do *mouse* em cima dos pontos que as coordenadas (x,y) são fornecidas na direita superior do menu, com possibilidade ampliação dos pontos do gráfico.

Por fim, o programa desenvolvido foi convertido para um arquivo executável, o que

dispensa a instalação do Python e de suas respectivas bibliotecas necessárias para o funcionamento do código nos computadores dos usuários.

CONCLUSÕES:

Tendo-se em vista as considerações feitas, o programa cumpriu com seus objetivos, de modo a facilitar os cálculos das variáveis das equações de HLD, que antes eram feitas na base da tentativa e erros através de planilhas, de modo a auxiliar no desenvolvimento e no estudo de formulações com parâmetro HLD determinado. Destaca-se a possibilidade de melhorias no programa como a obtenção das funções dos pontos obtidos graficamente, como, por exemplo, por meio de regressões lineares, e a opção de se adicionar novas constantes de surfactantes e valores de EACN nas tabelas. Por conta da pandemia do Covid-19, o tema inicial de projeto foi alterado e por conta disso, o desenvolvimento do programa foi feito em um tempo reduzido. Apesar disso, os objetivos foram cumpridos.

BIBLIOGRAFIA

ABBOT, S. **Surfactant Science: Principles and Practice**. 2015, p. 13 – 67.

ANDREOLLI, I. **Introdução à Elevação e Escoamento Monofásico e Multifásico de Petróleo**. Rio de Janeiro, 1st Editio ed. [s.l.] Editora Interciência, 2016.

SALAGER, JEAN-LOUIS & ANTÓN, R.E. & ANDÉREZ, J.M. & AUBRY, JEAN-MARIE. **Formulation des micro-émulsions par la méthode HLD**. Techniques de l'Ingénieur. 1-20. 2001

SALAGER, J. L. **Emulsion Phase Inversion Phenomena**. In: SJÖBLOM, J. Emulsion and Emulsion Stability. 2nd ed. Boca Raton: CRC Press, 2006.

SALAGER, J.L.; FORGIARINI, A.M.; BULLON, J. “**How to Attain Ultralow Interfacial Tension and Three-Phase Behavior with Surfactant Formulation for Enhanced Oil Recovery: A Review. Part 1. Optimum Formulation for Simple Surfactant–Oil–Water Ternary Systems**”, J. Surfact. Deterg. 2013 (16), p. 449-472.