

## DESENVOLVIMENTO DO MODELO DE MISTURADOR PARA O SIMULADOR DE PROCESSOS DWSIM UTILIZANDO A LINGUAGEM PYTHON

Nathally Mayumi Imamura (PIC/UEM), Oswaldo Curty da Motta Lima (Orientador),  
e-mail: ocmlima@uem.br

Universidade Estadual de Maringá / Centro de Tecnologia / Maringá, PR.

**Engenharia Química. Operações industriais e equipamentos para eng. Química.**

**Palavras-chave:** Modelagem, DWSIM, Misturador.

### Resumo:

As disciplinas de operações unitárias são de grande importância na graduação de engenharia química, dado que as operações unitárias são etapas básicas de um processo industrial. Sendo assim, é essencial que um engenheiro químico estude e compreenda este assunto. Simuladores como Aspen Plus e Aspen Hysys apresentam cálculos para tais operações, e podem ajudar o aluno a compreender o assunto; entretanto, suas licenças de uso possuem alto custo. Para contornar esta situação, pode-se utilizar o simulador DWSIM, gratuito e de código aberto. Este programa não apresenta tantas operações unitárias quanto os mencionados anteriormente, mas permite que o usuário crie seus próprios códigos de programação para realizar as operações desejadas. Tendo isto em vista, este projeto teve por objetivo o desenvolvimento de códigos de programação em Python para a operação unitária Misturador no software de simulação DWSIM.

### Introdução

Na Engenharia Química, a área de simulação de processos é muito importante, pois permite prever o comportamento de parâmetros e analisar dados antes que um processo ocorra de fato. Deste modo, é possível evitar possíveis erros, diminuir custos e otimizar processos (MARCUCCI, 2020). Como exemplo de simuladores de processos cotidianamente utilizados na Engenharia Química, podemos citar os softwares Aspen Plus e Aspen HYSYS, da AspenTech.

Devido ao fato destes softwares serem considerados dos mais sofisticados do mercado, eles possuem um custo de licença de uso consideravelmente alto, sendo mais utilizados por grandes empresas. (VERÍSSIMO, 2020)

Um dos simuladores de processos alternativos é o DWSIM, um simulador de processos químicos sequencial modular *open-source* (código aberto), brasileiro e compatível com o protocolo CAPE-OPEN, que tem como intuito permitir que, tanto

estudantes, quanto engenheiros químicos ativos, sejam capazes de simular processos químicos por meio de cálculos, modelos termodinâmicos e modelos de operações unitárias (VERÍSSIMO, 2020). O software possui uma interface gráfica intuitiva com diversas ferramentas e operações à disposição do usuário. Entretanto, quando comparado com simuladores utilizados no mercado de trabalho, o número de modelos e operações unitárias presentes no mesmo se mostra limitado, o que acaba dificultando a simulação de alguns tipos de processos industriais.

Contudo, para contornar essa dificuldade do programa, foi criada uma ferramenta dentro do próprio DWSIM, conhecida como Python Script (ou ainda *Custom Unit Operation*), que tem por finalidade a possibilidade ao usuário de criar uma operação unitária customizada no software, por meio de programação na linguagem Python. Qualquer pessoa pode compartilhar algum modelo que tenha desenvolvido para o software, sendo estes escritos em Python ou em Visual Basic no FOSSEE (*Free/Libre and Open Source Software for Education*). Esses modelos podem, posteriormente, ser utilizados por quaisquer usuários do DWSIM. (FOSSEE, 2022)

Portanto, este trabalho teve como objetivo desenvolver a operação unitária Misturador para o DWSIM em linguagem Python. O misturador é empregado quando se deseja misturar duas ou mais fases inicialmente separadas, de modo a tornar um meio heterogêneo em homogêneo e permitindo a transferência de calor e massa entre as correntes de alimentação do processo.

## Materiais e Métodos

Para os cálculos referentes ao Misturador, foi necessário realizar o balanço de massa global, de energia, de pressão e de componentes, expressos pelas equações abaixo

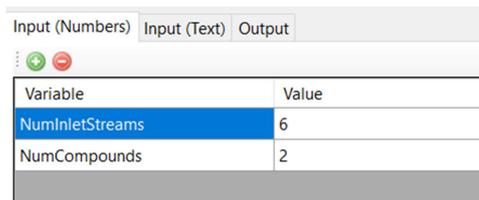
$$m = \sum_{i=1}^n m_i \quad (1) \quad H = \frac{\sum m_i h_i}{m} \quad (2) \quad P = \sum_{i=1}^n P_i \quad (3) \quad x_j = \frac{\sum_{i=1}^n x_i^j m_i}{m} \quad (04)$$

em que  $m$  é a massa da corrente de saída e  $m_i$  é a massa de cada corrente de entrada;  $H$  é a entalpia da corrente de saída e  $h_i$  é a entalpia específica de cada corrente de entrada;  $P$  é a pressão da corrente de saída e  $P_i$  é a pressão de cada corrente de entrada;  $x_j$  é a fração do componente  $j$  na corrente de saída e  $x_i^j$  é a fração do componente  $j$  em cada corrente de entrada.

## Resultados e Discussão

Ao utilizar essa operação, é fornecida ao usuário a possibilidade de inserir o número de correntes de entrada no processo (arquivado na variável *NumInletStreams*), além da quantidade de componentes que essas correntes terão (arquivado na variável *NumCompounds*), como representado pela Figura 1, cujas propriedades seriam

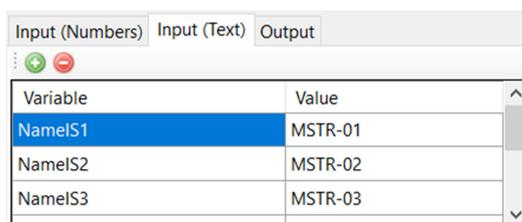
regidas pelo pacote termodinâmico "Raoult's Law", que considera sistema ideal, disponibilizado pelo software.



Variable	Value
NumInletStreams	6
NumCompounds	2

Figura 1: Aba de alterações de variáveis numéricas na interface do DWSIM.

Além disso, foi necessário criar variáveis que recebessem do usuário o nome de cada corrente de entrada, visto que os cálculos seriam realizados referenciando esses nomes.



Variable	Value
NameIS1	MSTR-01
NameIS2	MSTR-02
NameIS3	MSTR-03

Figura 2: Aba de alterações de variáveis textuais na interface do DWSIM.

O código foi escrito baseado na variável que recebia o número de correntes de entrada, ou seja, por meio das estruturas condicionais "if" e "elif", seriam rodadas diferentes seções do código.

Para o caso em que a variável *NumInletStreams* fosse igual a dois, primeiramente, teve-se que criar duas correntes (variáveis) de entrada, utilizando a função *Flowsheet.GetFlowsheetSimulationObject()*, e, em seguida, armazenar a entalpia específica, a vazão mássica, a pressão e a fração molar das correntes nas variáveis previamente criadas, utilizando as funções *.GetMassEnthalpy()*, *.GetMassFlow()*, *.GetPressure()* e *.GetOverallComposition()*.

Posteriormente, foi necessário realizar o balanço de massa global, de energia e de pressão, utilizando as equações 01, 02, 03 e, posteriormente, criar um vetor para armazenar as frações molares da corrente de saída. Em seguida, foi realizado o balanço de componentes, respeitando a equação 04, e criada a corrente de saída, tal como feita para a de entrada. Por fim, foram estabelecidas as propriedades calculadas dentro da variável da corrente de saída, utilizando as funções *.SetMassFlow()*, *.SetMassEnthalpy()*, *.SetPressure()*, *.SetOverallComposition()*.

O mesmo procedimento foi realizado para as diferentes quantidades de correntes de entrada, sendo que o DWSIM admite um máximo de seis correntes. O fluxograma do processo considerando seis correntes de entrada está representado pela Figura 3.

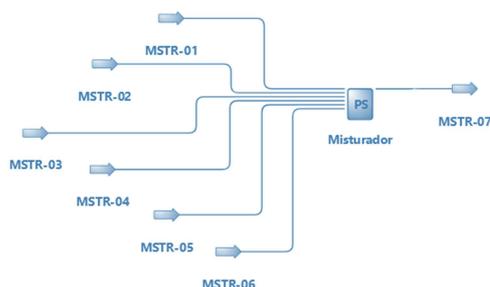


Figura 3 – Fluxograma da operação unitária Misturador na interface do DWSIM.

## Conclusões

O código desenvolvido para este projeto, embora apresente apenas um método de cálculo para a respectiva operação unitária, mostra-se suficientemente eficiente para resolução de problemas. O uso do mesmo requer algum conhecimento do software e das especificações do código, o que pode ser um empecilho inicial para sua execução. Contudo, ele pode ser utilizado, uma vez que necessita apenas da manipulação das variáveis para resolver os cálculos. Tendo em vista que os objetivos principais deste projeto eram desenvolver código de programação na linguagem Python para a operação unitária Misturador no software de simulação DWSIM, pode-se concluir que os objetivos foram alcançados.

## Referências

FREE/LIBRE AND OPEN SOURCE SOFTWARE FOR EDUCATION (FOSSEE). **DWSIM Chemical Process Simulator**, 2022. Disponível em: <<https://dwsim.fossee.in>>. Acesso em: 16 ago. 2022.

MARCUCCI, S. **Simulação de Processos: A importância da modelagem na análise de viabilidade técnica**, Propeq, 2020. Disponível em: <<https://propeq.com/simulacao-de-processos/#:~:text=A%20simulação%20e%20a%20modelagem,reduzindo%20custos%20e%20agilizando%20projetos.>>. Acesso em: 20 jul. 2022.

VERÍSSIMO, J. **10 Softwares Gratuitos Essenciais para Engenharia Química**, LinkedIn, 2020. Disponível em: <<https://www.linkedin.com/pulse/10-sofwares-gratuitos-essenciais-para-engenharia-jefferson-ver%C3%ADssimo/?originalSubdomain=pt>>. Acesso em: 20 jul. 2022.