



RAFAEL SILVA BARBOSA

**ANÁLISE DA DISTRIBUIÇÃO DE TEMPO DE
RESIDÊNCIA POR MEIO DE FERRAMENTAS
COMPUTACIONAIS**

**LAVRAS – MG
2024**

RAFAEL SILVA BARBOSA

**ANÁLISE DA DISTRIBUIÇÃO DE TEMPO DE RESIDÊNCIA POR MEIO
DE FERRAMENTAS COMPUTACIONAIS**

Monografia apresentada à Universidade Federal
de Lavras, como parte das exigências do Curso
de Engenharia Química, para a obtenção do
título de Bacharel.

Prof.^a Dr.^a Natália Maira Braga Oliveira
Orientadora

**LAVRAS – MG
2024**

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Barbosa, Rafael Silva.

Análise da distribuição de tempo de residência por meio de
ferramentas computacionais / Rafael Silva Barbosa. - 2024.

62 p. : il.

Orientador(a): Natália Maira Braga Oliveira.

TCC (graduação) - Universidade Federal de Lavras, 2024.

Bibliografia.

1. Engenharia das reações. 2. Simulação de processos. 3. DTR.
I. Oliveira, Natália Maira Braga. II. Título.

RAFAEL SILVA BARBOSA

**ANÁLISE DA DISTRIBUIÇÃO DE TEMPO DE RESIDÊNCIA POR MEIO
DE FERRAMENTAS COMPUTACIONAIS**

**ANALYSIS OF RESIDENCE TIME DISTRIBUTION USING
COMPUTATIONAL TOOLS**

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Engenharia Química, para a obtenção do título de Bacharel.

APROVADA em 17 de maio de 2024.

Prof.^a Dr.^a Natália Maira Braga Oliveira UFLA

Prof. Dr. Nathan Sombra Evangelista UFLA

Prof. Dr. João Moreira Neto UFLA

Prof.^a Dr.^a Natália Maira Braga Oliveira
Orientadora

**LAVRAS – MG
2024**

RESUMO

Atualmente, a simulação computacional é um recurso de extrema importância na formação e atuação dos engenheiros, fornecendo praticidade, agilidade, precisão e exatidão nos resultados, além de um conhecimento amplo do processo. Nesse contexto, o objetivo deste trabalho foi desenvolver um acervo de códigos gratuitos, na linguagem de programação Python, para cálculos voltados a projeto de reatores químicos, mais especificamente relativos à Distribuição de Tempo de Residência (DTR), com foco no ensino e na atuação profissional de engenheiros químicos, sendo implementado na *web* para facilitar o acesso e a utilização por parte dos usuários. Foi desenvolvido um conjunto de códigos que simulam a DTR a partir de dados de entrada de concentração de traçador pelo tempo de experimento. O acervo obtido é capaz de resolver problemas de perturbações em pulso e degrau, prevendo conversões por modelos sem parâmetros ajustáveis, como o modelo de segregação e o de mistura máxima, e com parâmetros ajustáveis, como o modelo de tanques em série (TES), de CSTR com desvio e espaço morto associado e de dois CSTR interconectados. Posteriormente, foi criada a interface e inteligência da página *web*, utilizando o *framework* Django do Python. Os algoritmos obtidos geram gráficos que permitem ao usuário visualizar as curvas de concentração de traçador, de DTR e da função de distribuição cumulativa do tempo de residência em função do tempo, sendo possível comparar os dados experimentais, dos reatores reais, com os correspondentes, relativos a reatores ideais, analisando a influência de parâmetros relevantes para operação dos equipamentos e permitindo identificar as causas dos desvios de idealidade. Por fim, conclui-se que o programa obtido é capaz de solucionar problemas complexos, de alto esforço numérico, sendo distribuído gratuitamente na internet, de modo que o usuário possa definir os parâmetros do processo, o que dispensa a instalação de *softwares* e permite avaliar diferentes condições de operação.

Palavras-chave: engenharia das reações; simulação de processos; DTR; Python; Django; ferramenta *web*.

SUMÁRIO

1.	INTRODUÇÃO.....	6
2.	REFERENCIAL TEÓRICO	9
2.1	Função Distribuição de Tempo de Residência (DTR)	9
2.2	Perturbação em pulso.....	10
2.3	Perturbação em degrau.....	12
2.4	Momentos da DTR.....	14
2.5	Modelos sem parâmetros ajustáveis	15
2.5.1	Segregação	15
2.5.2	Mistura máxima.....	16
2.6	Modelos com parâmetros ajustáveis	19
2.6.1	Tanques em série (TES)	20
2.6.2	Reator tanque modelado usando desvio e espaço morto.....	22
2.6.3	Reator tanque modelado como dois CSTR interconectados.....	24
2.7	DTR em reatores ideais.....	28
2.7.1	Reatores tubular (PFR) e batelada (BR) ideais.....	28
2.7.2	Reator de mistura ideal (CSTR).....	29
2.7.3	Reator com escoamento laminar (LFR).....	30
2.8	Diagnóstico usando a DTR.....	32
2.8.1	Operações dos reatores tanques	32
2.8.2	Operações dos reatores tubulares	33
3.	METODOLOGIA.....	35
3.1	Python.....	35
3.2	<i>Framework</i> Django	36
3.3	Regressão polinomial.....	38
4.	RESULTADOS E DISCUSSÃO	39
4.1	Cálculos para perturbação em pulso	39
4.2	Cálculos para perturbação em degrau	40
4.3	Modelos para reatores não ideais usando a DTR	42
4.3.1	Segregação e mistura máxima	43
4.3.2	TES.....	45
4.3.3	Tanque com desvio e espaço morto.....	47
4.3.4	Tanques interconectados.....	49

4.4 Ferramenta <i>web</i>	51
5. CONCLUSÃO.....	54
REFERÊNCIAS.....	56
APÊNDICE A – Enunciados dos problemas.....	58

1. INTRODUÇÃO

A indústria química é responsável por uma parcela significativa da economia global, com faturamento líquido no Brasil de US\$ 167,4 bilhões no ano de 2023, segundo a Associação Brasileira da Indústria Química (ABIQUIM, 2023) e tem seu valor global estimado em US\$ 6,2 trilhões com taxa de crescimento anual de 2% (MARKETS AND MARKETS, 2024). Esta indústria é responsável pela fabricação de produtos de consumo, petroquímicos, químicos básicos, agricultura, saúde e tecnologia em larga escala, e, para isso, utiliza de diversas ferramentas e equipamentos especializados. Dentre eles, os reatores ocupam uma posição crucial na modificação de substâncias químicas de forma eficiente e controlada.

Alguns eventos recentes têm causado impactos significativos na indústria química e seu futuro. A pandemia de COVID-19 impôs desafios pela perturbação na logística de suprimentos e mudanças da demanda, como a queda em indústrias não essenciais, por exemplo, na construção civil e no setor automotivo e a inflação de demanda nas áreas farmacêutica e de desinfetantes (ABEDSOLTAN, 2023). Outra questão em destaque é a crescente preocupação com a sustentabilidade e a adesão de grandes companhias a essa iniciativa, aderindo a novos processos e rotas químicas e buscando otimizar a logística para garantir uma redução dos impactos ambientais, como a pegada de carbono (COATINGS WORLD, 2022; YANG et al., 2022). Essas alterações impactam o fluxo de produção e a operação dos equipamentos, dentre os quais, têm-se os reatores, o que torna as ferramentas de análise de reatores essenciais para otimizar o processo (FLORIAN et al., 2020).

Nesse contexto, a análise da Distribuição de Tempo de Residência (DTR) é uma técnica fundamental para compreender e fornecer diagnóstico de problemas em reatores químicos, viabilizando assim a otimização de processos (FOGLER, 2022a; LEVENSPIEL, 2000). Portanto, é evidente a importância da didática no ensino dessa técnica para a formação dos engenheiros químicos.

Debate-se a eficácia do ensino utilizando simulação computacional e se esta pode ser comparada com aulas práticas. A aula prática é um método bem estabelecido para auxiliar na fixação dos conteúdos, mas é limitada pelo equipamento experimental disponível. Já a metodologia de ensino assistida por *softwares* supera essa limitação, proporcionando uma

visualização dos processos complexos da indústria química sem os custos dos experimentos e sem as restrições do uso de equipamentos experimentais. No entanto, esse método também apresenta desafios, como, por exemplo, a dificuldade de adaptação inicial ao programa. Fato é que tanto a simulação por computador quanto o experimento podem ser vistos como métodos instrucionais complementares, pois ambos contribuem para o conhecimento dos processos e para o desenvolvimento do pensamento crítico (RAMÍREZ et al., 2020; SUTHAR; JOSHIPURA, 2020).

Em adição a isso, a discussão sobre o ensino assistido por *softwares* e a utilização de aplicativos *mobile* vem crescendo (SILVA; VASCONCELOS, 2021). Essa metodologia traz a vantagem da rápida visualização e integração dos conceitos aprendidos, bem como a aproximação dos estudantes com a vivência profissional (SUTHAR; JOSHIPURA, 2023). Além disso, vale ressaltar a importância de *softwares* gratuitos para o ensino, quebrando a barreira econômica, incentivando a aprendizagem prática, além do incentivo à inovação nas universidades (LEITE et al., 2014).

Considerando o uso de recursos computacionais, tem-se a linguagem de programação Python, que é altamente requisitada e presente em ambientes acadêmicos e profissionais pela sua sintaxe e estrutura simples, com materiais e bibliotecas de fácil acesso. Esta ferramenta é muito útil para a criação de códigos abertos para o ensino, pois permite que qualquer pessoa com conhecimento em programação contribua para o desenvolvimento do projeto, mantendo-o atualizado e diversificado.

Na área de Engenharia Química, é possível encontrar algumas ferramentas gratuitas em Python, as quais podem ser utilizadas no ensino. Um exemplo é o SPyCE, um *framework* educacional com alguns exemplos utilizando Python para modelagem de problemas em engenharia química e biotecnologia (CACCAVALE, 2023). Além disso, há bibliotecas e módulos em Python que contribuem para simulação de reações químicas e cálculos termodinâmicos, como ChemPy (DAHLGREN, 2018), ASE (Atomic Simulation Environment) (ASE DEVELOPERS, 2024) e Cantera (CANTERA DEVELOPERS, 2024). Para simulação de fluidos, o PyFR é uma opção (PYFR TEAM, 2024), enquanto para problemas de transferência de calor, o FiPy se destaca (NIST, 2024); já o PyGAMD é para simulação da dinâmica molecular de polímeros (GAO et al., 2024).

Cabe mencionar que H. Scott Fogler criou um projeto *online*, vinculado ao seu livro "Elementos de Engenharia das Reações Químicas", para apoiar o ensino de Engenharia Química. Este projeto consiste em um *site* que oferece materiais adicionais e outras referências para cada capítulo do livro, com exemplos resolvidos em Polymath, Python, Matlab e Wolfram. No projeto, é possível encontrar problemas relacionados a temas como: cinética química, estequiometria, modelagem de reatores químicos ideais e não-ideais, distribuição de tempo de residência e reações catalíticas (FOGLER, 2022a; FOGLER, 2022b).

Entretanto, destaca-se a baixa disponibilidade de programas gratuitos e com tradução para o português capazes de simular a Distribuição de Tempo de Residência, um conteúdo essencial na abordagem dos reatores químicos. Durante o período de pesquisa deste trabalho, não foi identificada nenhuma ferramenta, além do *site* do Fogler (2022b) em inglês, capaz de tratar dados de DTR. Diante desse cenário, este trabalho buscou criar uma ferramenta que superasse a limitada disponibilidade desses recursos para os estudantes, ao mesmo tempo em que ressalta a importância do estudo da DTR.

De modo que o presente trabalho teve como objetivo desenvolver um acervo de códigos gratuito em Python capaz de solucionar problemas envolvendo cálculos de Distribuição de Tempo de Residência (DTR), aumentando a acessibilidade desse tipo de *software* a estudantes e engenheiros químicos, disponibilizando o recurso na *web*.

2. REFERENCIAL TEÓRICO

A modelagem dos reatores químicos geralmente é feita supondo um comportamento ideal, para a simplificação das equações. No entanto, na realidade, esse comportamento nem sempre se sustenta, o que afeta a exatidão do modelo. Dessa forma, se faz necessária a utilização de técnicas para estimar a proximidade dos reatores reais da idealidade, e melhorar as previsões de cálculo desses equipamentos.

Nesse contexto, a Distribuição de Tempo de Residência é uma ferramenta utilizada para caracterizar reatores não ideais, usada principalmente para prever a conversão e concentração de saída das espécies reacionais de maneira mais exata, em reatores que estão sendo projetados, ou para diagnosticar possíveis problemas em reatores em operação, como volume morto, mistura incompleta, desvio de fluxo e caminhos preferencias.

As deduções matemáticas das equações da DTR, seus momentos e dos modelos, apresentadas a seguir, foram desenvolvidas com base no Fogler (2022a).

2.1 Função Distribuição de Tempo de Residência (DTR)

O tempo que as moléculas permanecem no reator é denominado de tempo de residência. Em um reator tubular empistonado ideal, todas as moléculas em todo o raio da saída têm o mesmo tempo de residência, ou seja, fatores que alterariam essa dinâmica como a incrustação do tubo e os caminhos preferenciais são negligenciados. Ao pensar em um reator batelada ideal, é considerado que todos os átomos permanecem o mesmo tempo no reator, dessa forma, todas as moléculas têm o mesmo tempo de residência e os tempos de alimentação e retirada do conteúdo do reator são desprezados.

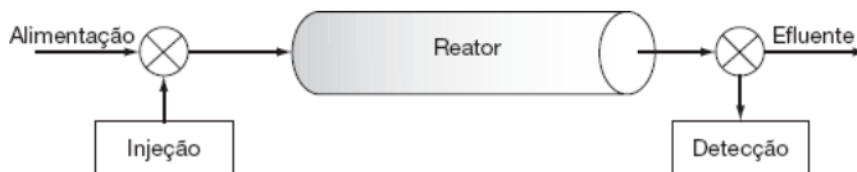
No entanto, em reatores reais, algumas moléculas podem sair antes ou depois do tempo de residência considerado para os reatores ideais.

Além disso, a qualidade da mistura e o volume morto são dois fatores que também afetam o tempo de residência. Em um CSTR, a mistura é considerada perfeita e não há volume morto, o que diz que as moléculas passam por todo o volume disponível até saírem do reator e, em qualquer ponto desse reator, as condições operacionais são as mesmas. Em um reator tanque de mistura não ideal, existem regiões nas extremidades nas quais a mistura

não é tão efetiva e os reagentes ficam estagnados, o que caracteriza o volume morto e diminui o tempo de residência, ao passo que existem partículas na entrada que são arrastadas diretamente para a saída, permanecendo pouco tempo no reator.

A determinação da distribuição de tempo de residência (DTR) tem o intuito de identificar essas variações no tempo de residência, a presença de volume morto e caminhos preferenciais. Assim, a técnica é realizada pela injeção de um traçador, uma substância inerte, facilmente detectável e com propriedades similares a dos reagentes. A injeção ocorre num tempo de referência, $t = 0$, e a concentração do traçador é monitorada na saída do reator, como ilustrado na Figura 1.

Figura 1 - Esquema experimental da injeção de traçador em um reator.



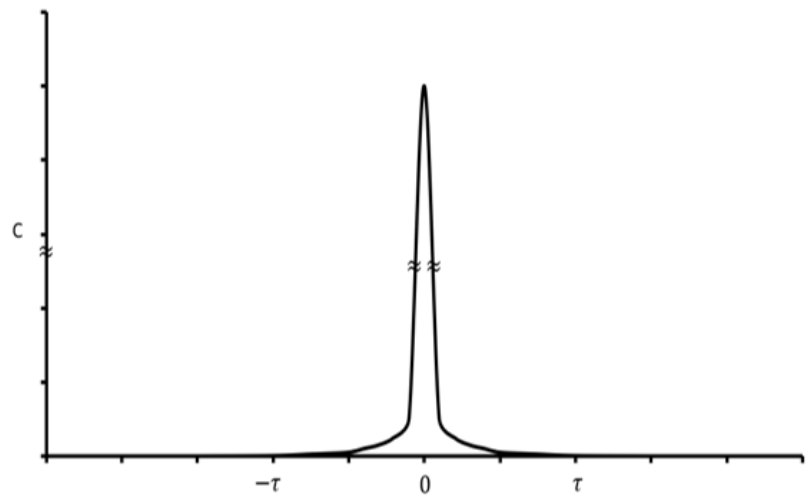
Fonte: Fogler (2022a).

As perturbações em pulso e em degrau são os dois métodos mais utilizados para a determinação da Distribuição do Tempo de Residência (DTR). A perturbação em pulso apresenta um ensaio experimental mais desafiador, mas com cálculos manuais mais simples. Por outro lado, o experimento de perturbação em degrau pode ser considerado de mais fácil execução, porém apresenta uma maior dificuldade nos cálculos sem auxílio de *softwares*.

2.2 Perturbação em pulso

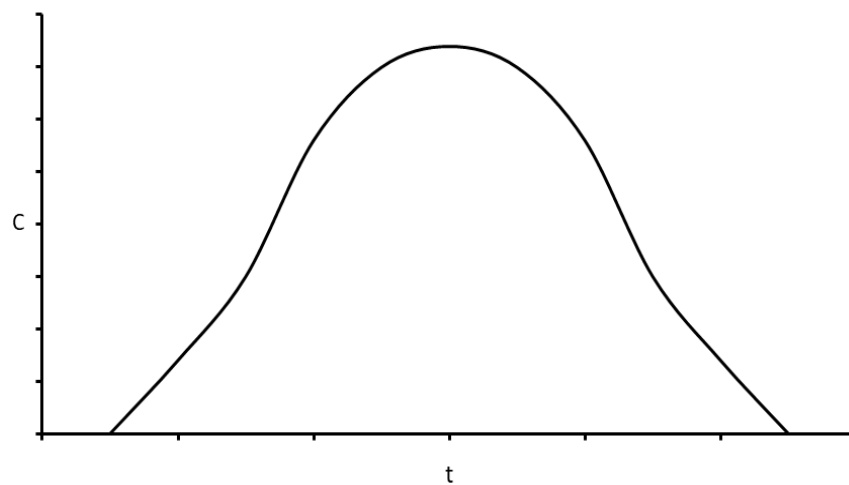
Na perturbação em pulso, o traçador é injetado de forma instantânea na alimentação do reator no tempo $t = 0$. A concentração de traçador na saída é mensurada ao longo do tempo. A Figura 2 e Figura 3 mostram os gráficos de concentração por tempo na entrada e saída, respectivamente, esperada de uma perturbação em pulso.

Figura 2 - Concentração de entrada para perturbação pulso.



Fonte: Adaptado de Fogler (2022a).

Figura 3 - Concentração de saída para perturbação pulso.



Fonte: Adaptado de Fogler (2022a).

Todo o traçador na entrada eventualmente saíra do reator. Dessa forma, a quantidade de traçador injetada (N_0) pode ser calculada pela área abaixo da curva C entre o tempo zero e infinito, na Figura 3, como mostra a Equação 1, em que v é a vazão volumétrica, geralmente constante, e $C(t)$ é a função de concentração de traçador ao longo do tempo.

$$N_0 = \int_0^{\infty} vC(t) dt \quad (1)$$

A função de distribuição de tempo de residência, $E(t)$, descreve por quanto tempo diferentes elementos de fluido permanecem no reator. De modo que, para uma injeção do tipo pulso, $E(t)$ é definida de acordo com a Equação 2.

$$E(t) = \frac{vC(t)}{N_0} = \frac{C(t)}{\int_0^{\infty} C(t) dt} \quad (2)$$

A função de distribuição cumulativa, $F(t)$, mostra a quantidade percentual de traçador que saiu do reator entre um tempo 0 até t e pode ser encontrada através da integral da Equação 3.

$$F(t) = \int_0^t E(t) dt \quad (3)$$

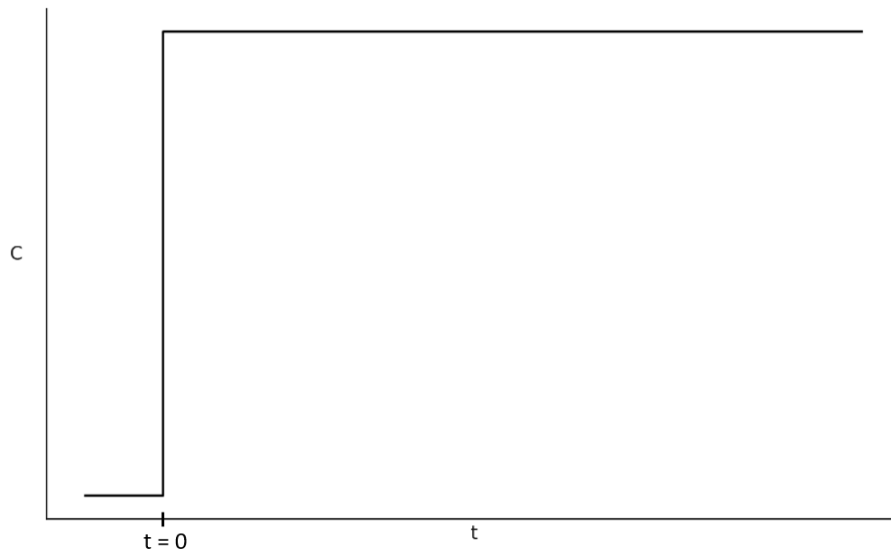
Analisando a função de distribuição cumulativa (Equação 3) de um tempo zero a infinito, todo o traçador injetado terá saído do reator, em outras palavras, 100% do traçador já passou pela saída e $F(t) = 1$.

Dentre as dificuldades relacionadas à perturbação em pulso, se destacam a impossibilidade da injeção instantânea do traçador, pois na prática ocorre uma injeção num curto intervalo de tempo, e problemas de ajuste de $E(t)$ a um polinômio, especialmente quando a curva de concentração por tempo apresenta uma cauda muito longa.

2.3 Perturbação em degrau

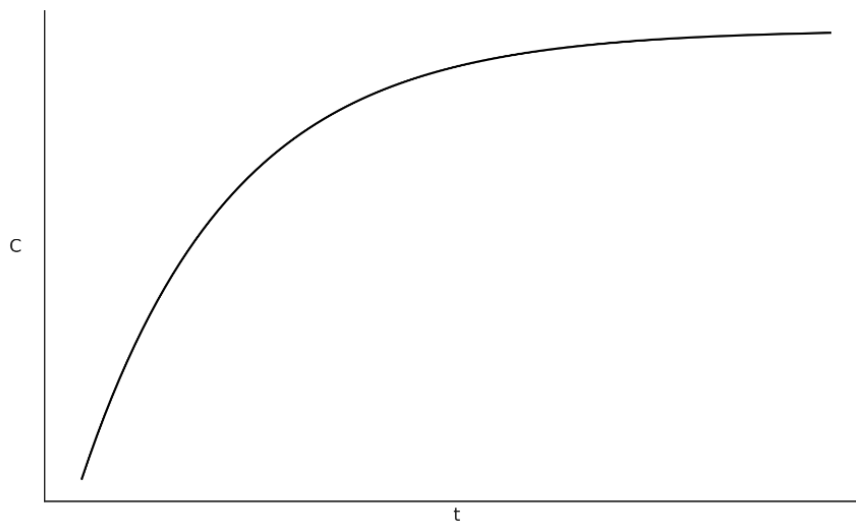
Na perturbação em degrau, inicia-se a injeção constante do traçador no tempo $t = 0$, com uma concentração C_0 , que não varia durante o experimento, e a concentração de saída é avaliada até que seja igual à concentração de entrada no reator. A Figura 4 e Figura 5 demonstram a concentração de entrada e saída para a perturbação em degrau.

Figura 4 - Concentração de entrada para perturbação degrau.



Fonte: Adaptado de Fogler (2022a).

Figura 5 - Concentração de saída para perturbação degrau.



Fonte: Adaptado de Fogler (2022a).

Na perturbação degrau, a concentração inicial, C_0 , é constante e a razão entre a concentração de saída e a concentração inicial do reator fornece a função de distribuição cumulativa, $F(t)$, como mostra a Equação 4. Ou seja, a concentração de traçador na saída em relação a de entrada corresponde ao percentual de traçador que permaneceu um tempo inferior a t no reator.

$$F(t) = \left[\frac{C_{saída}}{C_0} \right]_{degrau} \quad (4)$$

Aplicando a Equação 4 com o definido na Equação 3, pode-se calcular $E(t)$ através da Equação 5 para este tipo de perturbação.

$$E(t) = \frac{dF(t)}{dt} = \frac{d}{dt} \left[\frac{C_{saída}}{C_0} \right]_{degrau} \quad (5)$$

Como apontado por Fogler (2022a), comparada à perturbação em pulso, a degrau se apresenta como um ensaio experimental mais simples. No entanto, manter a concentração de alimentação constante e a alta quantidade de traçador requerida são pontos negativos desta técnica. Além disso, a função $E(t)$ é encontrada através da derivada da concentração, o que representa uma dificuldade de cálculo manual e pode acarretar erros. Apesar disso, considerando o auxílio de *softwares*, os dados obtidos pela perturbação em degrau apresentam uma regressão mais simples, favorecendo o tratamento computacional de dados obtidos por este método.

2.4 Momentos da DTR

Usualmente, a DTR é analisada em três momentos, o tempo de residência médio, a variância e a distorção (*skewness*).

O primeiro momento é o tempo de residência médio (t_m), que fornece o tempo médio que as moléculas permanecem no reator, similar ao tempo de residência (τ) do reator. Essa propriedade pode ser calculada segundo a Equação 6.

$$t_m = \int_0^{\infty} t \cdot E(t) dt \quad (6)$$

Para sistema fechado, sem dispersão e com vazão volumétrica (v) constante, $t_m = \tau$, em que o tempo de residência (τ) é definido na Equação 7.

$$\tau = \frac{V}{v} \quad (7)$$

O volume útil do reator é determinado pela Equação 8.

$$V = v \cdot t_m \quad (8)$$

O segundo momento da DTR é a variância (σ^2), ou o quadrado do desvio padrão, que indica a dispersão da distribuição em relação a t_m e pode ser calculada pela Equação 9.

$$\sigma^2 = \int_0^{\infty} (t - t_m)^2 \cdot E(t) dt \quad (9)$$

E o terceiro momento é a distorção (*skewness*) (s^3) que mede se a inclinação da distribuição está antes (valor negativo) ou posterior (valor positivo) a t_m . A Equação 10 demonstra o cálculo para determinar s^3 .

$$s^3 = \frac{1}{\sigma^{3/2}} \int_0^{\infty} (t - t_m)^3 \cdot E(t) dt \quad (10)$$

2.5 Modelos sem parâmetros ajustáveis

Há dois modelos sem parâmetros ajustáveis, os quais são apresentados a seguir, sendo os modelos de segregação e mistura máxima. Ambos apresentam cálculos mais simples e, por serem baseados em condições extremas de mistura, representam valores limites da previsão de conversão. Portanto, de acordo com Fogler (2022a), pode-se afirmar que a conversão do reator real estará entre as conversões previstas por esses dois modelos.

2.5.1 Segregação

O modelo de segregação é assim chamado pois considera que os reatores são formados por várias regiões de micromistura com idades diferentes. Dessa forma, uma

mistura ideal e perfeita seria inviável, já que elementos de idades diferentes não se misturam, ou seja, permanecem segregados.

No CSTR, há a hipótese de mistura perfeita desses elementos, já no PFR pressupõe-se um fluxo empistonado, no entanto, na realidade existem regiões do reator mais estagnadas, com longos períodos de residência, e outras com caminhos preferenciais, cujas moléculas apresentam menor tempo de residência. Dessa forma, o conteúdo do reator é dividido em pequenos reatores (glóbulos), cada um com tempo de residência diferente, que se comportam como reatores batelada.

O cálculo da conversão para esse modelo é feito através da média de conversão de todos os glóbulos, considerando o tempo de permanência no reator. A Equação 11 demonstra a análise infinitesimal, ao passo que a Equação 12 apresenta a integração, resultando na conversão média do reator prevista pelo modelo de segregação.

$$d\bar{X} = X(t) \cdot E(t) dt \quad (11)$$

$$\bar{X} = \int_0^{\infty} X(t) \cdot E(t) dt \quad (12)$$

Como a função $X(t)$ representa a conversão de cada glóbulo, ela é obtida pela equação de projeto de um reator batelada (Equação 13). Isso porque que cada glóbulo, em essência, pode ser considerado um reator batelada.

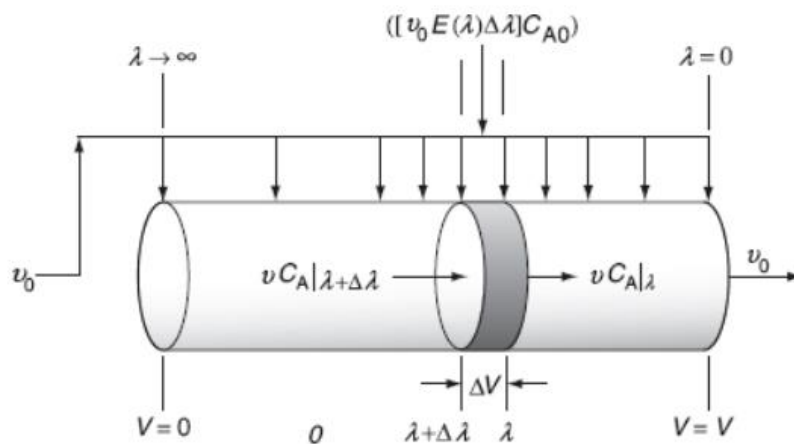
$$\frac{dX}{dt} = -\frac{r_A V}{N_{A0}} \quad (13)$$

2.5.2 Mistura máxima

No modelo de mistura máxima, examinamos uma situação extrema em que a mistura ocorre no momento mais precoce possível, ou seja, assim que o fluido entra no reator, suas partículas são instantaneamente misturadas ao conteúdo do reator. Esse cenário representa o oposto do modelo de segregação, no qual a mistura ocorre no momento mais tardio possível.

Para exemplificar o desenvolvimento do modelo de mistura máxima, é usado um reator tubular com escoamento empistonado e alimentação nas laterais, como mostra a Figura 6. A alimentação lateral é importante para considerar moléculas com um tempo longo no reator e moléculas que formaram caminhos preferenciais. Para essa avaliação, é estipulado um parâmetro λ , que mede a expectativa de vida das moléculas no reator. Dessa forma, a alimentação no início do reator tem a maior expectativa de vida possível ($\lambda \rightarrow \infty$) e a alimentação no final do reator tem $\lambda = 0$.

Figura 6 - Exemplificação do modelo de mistura máxima.



Fonte: Adaptado de Fogler (2022a).

Na Figura 6, é destacada uma porção de volume ΔV que representa volume de controle, em que o fluido que entra nessa porção tem expectativa de vida entre λ e $\lambda + \Delta \lambda$ e os termos $v C_A |_{\lambda + \Delta \lambda}$ e $v C_A |_{\lambda}$ representam a vazão molar que entra e sai nessa porção de volume. O parâmetro v_0 expressa a vazão volumétrica de entrada total, enquanto v representa a soma das vazões de entrada até aquela posição no reator. Por fim, o termo $([v_0 E(\lambda) \Delta \lambda] C_{A0})$ mostra a quantidade do componente A que entra no volume de controle.

Considerando esta modelagem, pode-se desenvolver o equacionamento para encontrar a conversão ao longo do reator prevista pelo modelo de mistura máxima, utilizando as funções $E(\lambda)$ e $F(\lambda)$, como apresentado a seguir.

A vazão volumétrica do fluido entrando no reator de modo a ter expectativa de vida entre λ e $\lambda + \Delta \lambda$ é descrita pela Equação 14.

$$v(\lambda) = \sum_{\lambda}^{\infty} v_0 E(\lambda) \Delta\lambda \quad (14)$$

Rearranjando a Equação 14, tomando limite quando $\Delta\lambda \rightarrow 0$ e integrando com limites $v(\lambda) = 0$ em $\lambda \rightarrow \infty$ e $v(\lambda) = v(\lambda)$ em $\lambda = \lambda$, chega-se na Equação 15.

$$v(\lambda) = v_0 \int_{\lambda}^{\infty} E(\lambda) d\lambda = v_0 [1 - F(\lambda)] \quad (15)$$

A taxa de geração do componente limitante A em ΔV é expressa pela Equação 16.

$$r_A \Delta V = r_A v(\lambda) \Delta\lambda = r_A v_0 [1 - F(\lambda)] \Delta\lambda \quad (16)$$

Considerando as entradas ao longo do reator tubular (Figura 6) e a geração de A (Equação 16), pode-se fazer o balanço molar de A entre λ e $\lambda + \Delta\lambda$, para operação em estado estacionário, como demonstrado na Equação 17, em que o primeiro termo representa a entrada em $\lambda + \Delta\lambda$, o segundo a entrada pelas laterais, o terceiro a saída em λ e o quarto a quantidade gerada ou consumida pela reação.

$$v_0 [1 - F(\lambda)] C_A|_{\lambda+\Delta\lambda} + v_0 C_{A0} E(\lambda) \Delta\lambda + v_0 [1 - F(\lambda)] C_A|_{\lambda} + r_A v_0 [1 - F(\lambda)] \Delta\lambda = 0 \quad (17)$$

Dividindo a Equação 17 por $(v_0 \Delta\lambda)$ e aplicando limite quando $\Delta\lambda \rightarrow 0$, obtém-se a Equação 18.

$$\frac{d\{[1 - F(\lambda)]C_A\}}{d\lambda} + C_{A0}E(\lambda) + r_A[1 - F(\lambda)] = 0 \quad (18)$$

Aplicando a derivada no primeiro termo da Equação 18, tem-se a Equação 19; a partir da qual, isolando a derivada da concentração de A em relação à λ , obtém-se a Equação 20.

$$\left\{ [1 - F(\lambda)] \frac{dC_A}{d\lambda} - C_A E(\lambda) \right\} + C_{A0} E(\lambda) + r_A [1 - F(\lambda)] = 0 \quad (19)$$

$$\frac{dC_A}{d\lambda} = -r_A + (C_A - C_{A0}) \cdot \frac{E(\lambda)}{1 - F(\lambda)} \quad (20)$$

Com base na Equação 20, considerando a definição de conversão do reagente limitante, tem-se a Equação 21.

$$\frac{dX}{d\lambda} = \frac{r_A}{C_{A0}} + X \cdot \frac{E(\lambda)}{1 - F(\lambda)} \quad (21)$$

Quando λ se aproxima do infinito, $C_A = C_{A0}$ ou $X = 0$, representando a entrada do reator. Por outro lado, quando $\lambda = 0$, a conversão final é encontrada. De modo que essas duas condições de contorno devem ser aplicadas à Equação 21 para encontrar a conversão prevista pelo modelo de mistura máxima.

2.6 Modelos com parâmetros ajustáveis

Os modelos com parâmetros ajustáveis acrescentam complexidade ao cálculo. No entanto, a conversão prevista por eles é mais próxima do valor real. Isso porque a escolha de um modelo com parâmetros ajustáveis, adequada ao contexto em estudo, tende a aumentar a exatidão dos cálculos de conversão.

Neste trabalho, são apresentados três modelos com parâmetros ajustáveis: reator tubular modelado como uma sequência de CSTR em série (TES), com um parâmetro ajustável; reator tanque modelado considerando desvio de fluxo e espaço morto; e reator tanque modelado como dois CSTR interconectados; sendo esses dois últimos com dois parâmetros ajustáveis.

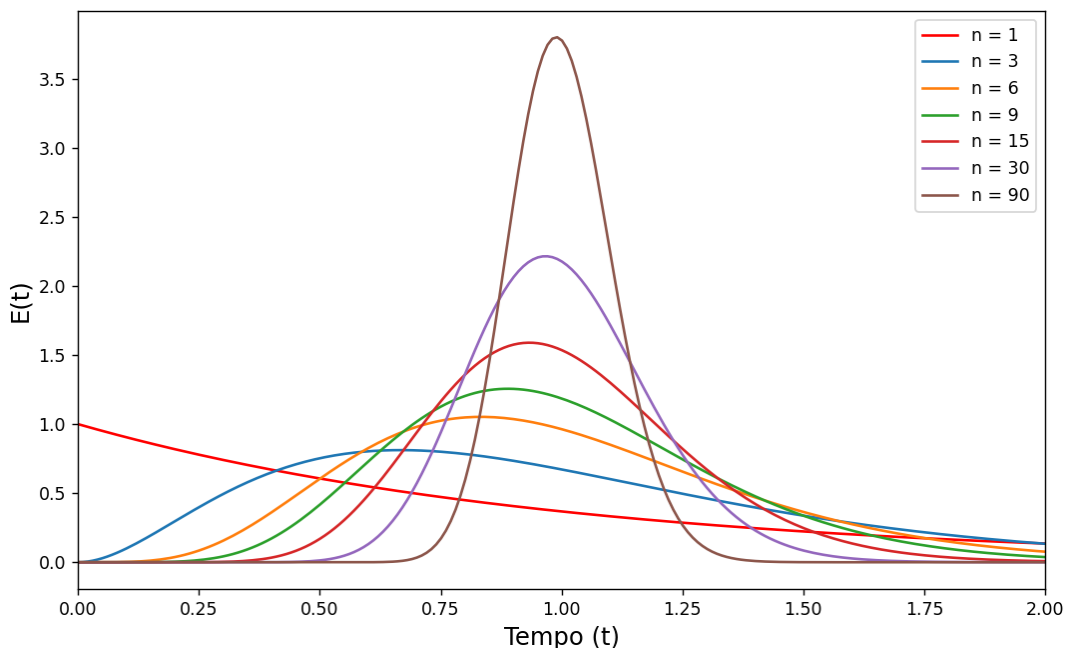
2.6.1 Tanques em série (TES)

O modelo de tanques em série descreve o comportamento de um reator não ideal modelado a partir de uma quantidade n de reatores ideais conectados em série. O modelo calcula a conversão com base em um único parâmetro ajustável, o número de reatores tanques ideais em série (n).

O modelo é eficaz para prever o comportamento de reatores tubulares que não seguem o fluxo pistão ideal, bem como de reatores com mistura imperfeita. Dividindo o reator tubular em reatores tanque em série, é possível capturar as variações de tempo de residência e os efeitos da dispersão ao longo do comprimento do reator. Além disso, a combinação de tanques em série pode simular a mistura parcial e as zonas mortas que podem ocorrer em reatores.

À medida que o número de reatores em série aumenta, a curva $E(t)$ se aproxima do comportamento de um PFR, como demonstrado na Figura 7.

Figura 7 – Impacto do número de reatores em série na curva $E(t)$.



Fonte: Adaptado de Gao, Muzzio e Ierapetritou (2012).

O balanço molar para n reatores em série resulta na Equação 22.

$$C_n = C_0 \frac{t^{n-1}}{(n-1)! \tau_i^n} e^{-t/\tau_i} \quad (22)$$

Sendo que τ_i é calculado pela Equação 23.

$$\tau_i = \frac{\tau}{n} \quad (23)$$

Substituindo a Equação 22 na definição de $E(t)$ para perturbação pulso, a Equação 2, obtém-se a Equação 24.

$$E(t) = \frac{C_n(t)}{\int_0^{\infty} C_n(t) dt} = \frac{t^{n-1}}{(n-1)! \tau_i^n} e^{-t/\tau_i} \quad (24)$$

Segundo este modelo, o número de reatores tanque em série é calculado pela Equação 25.

$$n = \frac{\tau^2}{var} \quad (25)$$

O tempo de residência (τ) e a variância (var) são dados pelas Equações 26 e 27, respectivamente, as quais são equivalentes aos momentos da DTR representados pelas Equações 6 e 9.

$$\tau = \int_0^{\infty} t \cdot E(t) dt \quad (26)$$

$$var = \int_0^{\infty} (t - \tau)^2 \cdot E(t) dt \quad (27)$$

O cálculo da conversão prevista pelo modelo TES para uma reação de primeira ordem é dado pela Equação 28.

$$X_{TES} = 1 - \frac{1}{(1 + \tau_i k)^n} \quad (28)$$

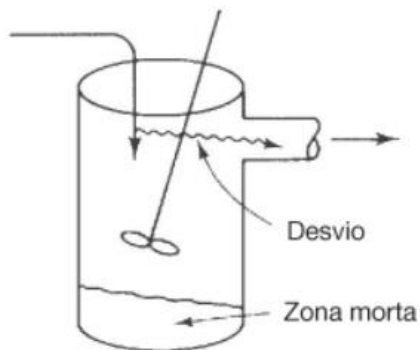
Cabe destacar que, para uma reação de primeira ordem, as conversões previstas pelo modelo TES, segregação e mistura máxima são as mesmas, conforme apontado por Fogler (2022a).

2.6.2 Reator tanque modelado usando desvio e espaço morto

O modelo propõe que o reator tanque real pode ser modelado como dois reatores separados, um CSTR com volume bem misturado (V_s) e uma zona de volume morto (V_m) e um desvio de vazão volumétrica (v_d).

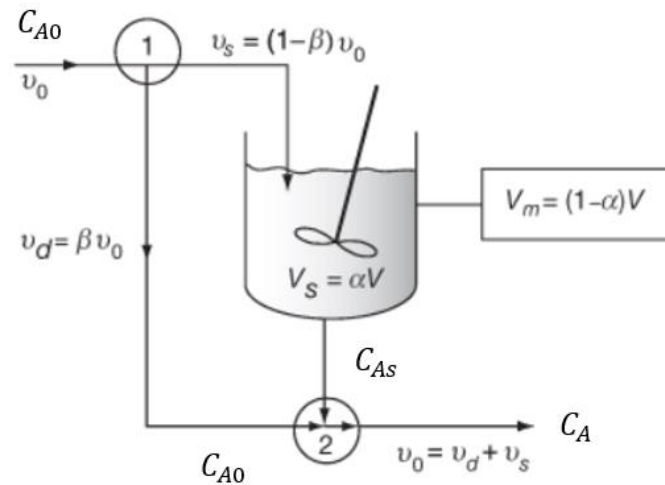
As Figura 8 e Figura 9 mostram os esquemas do sistema real e do sistema modelo, respectivamente.

Figura 8 - Sistema real do CSTR com volume morto.



Fonte: Adaptado de Fogler (2022a).

Figura 9 - Sistema modelo do CSTR com volume morto.



Fonte: Adaptado de Fogler (2022a).

Pela Figura 9, percebe-se que parte da corrente de entrada é desviada no ponto 1 e se encontra com a corrente de saída do reator no ponto 2. Além disso, não ocorre reação em parte do volume do reator, representado por V_m . As constantes α e β são calculadas pelas Equações 29 e 30, respectivamente.

$$\alpha = \frac{V_s}{V} \quad (29)$$

$$\beta = \frac{v_d}{v_0} \quad (30)$$

A constante α representa a porcentagem de volume útil do reator, ou seja, o volume em que a reação ocorre. Já a constante β mostra a porcentagem da vazão volumétrica que é desviada no reator. Dessa forma, esse modelo apresenta dois parâmetros ajustáveis, α e β .

Aplicando o modelo para uma reação de primeira ordem, como a descrita pela Equação 31, é possível encontrar a conversão em função dos parâmetros do modelo, através do balanço molar no volume bem misturado.



A Equação 32 descreve a concentração da espécie A em função de α e β para uma reação de primeira ordem.

$$\frac{C_A}{C_{A0}} = 1 - X = \beta + \frac{(1+\beta)^2}{(1-\beta)+\alpha\tau k} \quad (32)$$

Para a reação de segunda ordem, a concentração final do reagente depende de C_{As} , concentração de saída do reator antes do ponto de mistura 2, apontado na Figura 9. De modo que o cálculo das concentrações C_A e C_{As} seguem as Equações 33 e 34, respectivamente.

$$C_A = \beta C_{A0} + (1 - \beta) C_{As} \quad (33)$$

$$C_{As} = \frac{-1 + \sqrt{1 + 4\tau_s k C_{A0}}}{2\tau_s k} \quad (34)$$

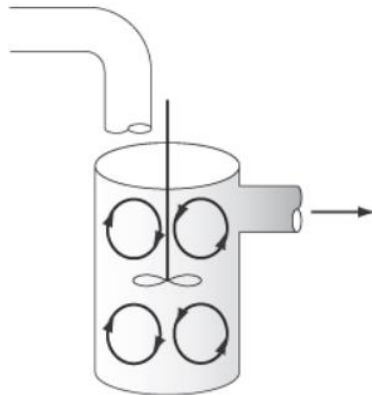
Aplicando o balanço molar para o traçador no ponto 2 da Figura 9 para perturbação degrau, é obtida a Equação 35 que relaciona a concentração de traçador (C_T) com o tempo (t).

$$\frac{C_T}{C_{T0}} = 1 - (1 - \beta) \exp \left[-\frac{1-\beta}{\alpha} \left(\frac{t}{\tau} \right) \right] \quad (35)$$

2.6.3 Reator tanque modelado como dois CSTR interconectados

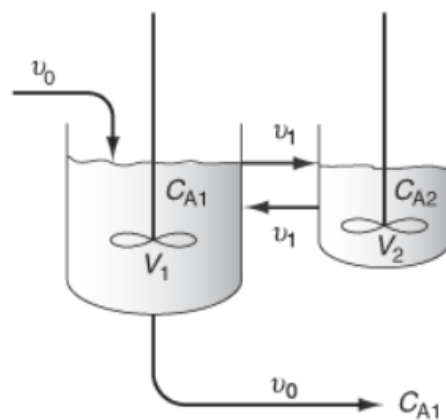
No modelo com dois CSTR interconectados, o reator real é modelado como dois reatores tanque ideais interconectados, um altamente agitado e outro pouco agitado. A Figura 10 mostra o sistema real enquanto a Figura 11 exemplifica o sistema modelado correspondente. O modelo apresenta dois parâmetros ajustáveis, a porcentagem do volume altamente agitado (α) e a fração do escoamento que passa pelo reator pouco agitado (β).

Figura 10 - Sistema real do reator tanque modelado como dois CSTR interconectados.



Fonte: Adaptado de Fogler (2022a).

Figura 11 - Sistema modelo do reator tanque modelado como dois CSTR interconectados.



Fonte: Adaptado de Fogler (2022a).

Considerando o reator muito agitado como o reator 1 e o pouco agitado como o reator 2 na Figura 11, define-se o tempo de residência pela Equação 36 e os parâmetros α e β pelas Equações 37 e 38, respectivamente.

$$\tau = \frac{V}{v_0} = \frac{V_1 + V_2}{v_0} \quad (36)$$

$$\alpha = \frac{V_1}{V} \quad (37)$$

$$\beta = \frac{v_1}{v_0} \quad (38)$$

A partir do balanço molar nos dois reatores para uma reação de primeira ordem, como a expressa na Equação 31, obtém-se a Equação 39, que apresenta a concentração de saída da espécie A em função dos parâmetros do modelo.

$$C_{A1} = \frac{C_{A0}}{1 + \beta + \alpha\tau k - \beta^2 / [\beta + (1 - \alpha)\tau k]} \quad (39)$$

De modo que a conversão para o modelo de dois CSTR interconectados é apresentada pela Equação 40.

$$X = 1 - \frac{C_{A1}}{C_{A0}} = \frac{(\beta + \alpha\tau k)[\beta + (1 - \alpha)\tau k] - \beta^2}{(1 + \beta + \alpha\tau k)[\beta + (1 - \alpha)\tau k] - \beta^2} \quad (40)$$

Para determinação dos parâmetros do modelo a partir do uso de um traçador com perturbação em pulso são utilizadas as Equação 41 e 42, que partem do balanço molar do traçador nos reatores 1 e 2 da Figura 11, respectivamente.

$$\tau\alpha \frac{dC_{T1}}{dt} = \beta C_{T2} - (1 + \beta)C_{T1} \quad (41)$$

$$\tau(1 - \alpha) \frac{dC_{T2}}{dt} = \beta C_{T1} - \beta C_{T2} \quad (42)$$

A solução dessas equações diferenciais (Equações 41 e 42) resulta na Equação 43.

$$\left(\frac{C_{T1}}{C_{T10}} \right)_{pulso} = \frac{(\alpha m_1 + \beta + 1) e^{\frac{m_2 t}{\tau}} - (\alpha m_2 + \beta + 1) e^{\frac{m_1 t}{\tau}}}{\alpha(m_1 - m_2)} \quad (43)$$

Em que os parâmetros m_1 e m_2 são calculados pela Equação 44.

$$m_1, m_2 = \left[\frac{1-\alpha+\beta}{2\alpha(1-\alpha)} \right] \left[-1 \pm \sqrt{1 - \frac{4\alpha\beta(1-\alpha)}{(1-\alpha+\beta)^2}} \right] \quad (44)$$

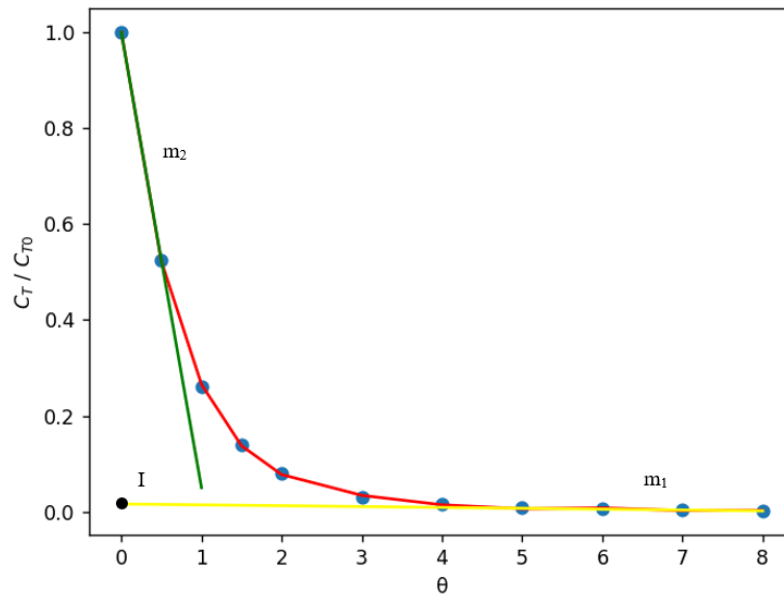
É possível calcular α pelo balanço do traçador, a partir da área abaixo da curva $C_T(\theta)/C_{T0}$, como mostra a Equação 45, sendo θ um adimensional de tempo, dado por $\theta = t/\tau$.

$$\alpha = \frac{1}{C_{T0}} \int_0^\infty C_T(\theta) d\theta = \int_0^\infty \frac{C_T(\theta)}{C_{T0}} d\theta \quad (45)$$

Já o parâmetro β é encontrado pela Equação 46, obtida pela extrapolação da curva correspondente à representada pela Equação 43, para um longo tempo atrás, $\theta = 0$, com o primeiro termo contendo m_2 no expoente sendo insignificante, como representado na Figura 12.

$$\text{Interseção} = I = -\frac{\alpha m_2 + \beta + 1}{\alpha(m_1 - m_2)} \quad (46)$$

Figura 12 - Curva $C_T(\theta)/C_{T0}$ com regressões nos primeiros e últimos pontos.



Fonte: Do autor (2024).

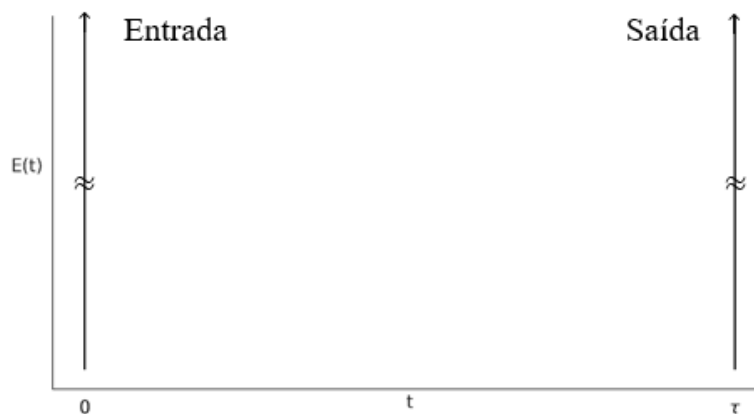
Ou seja, como demonstrado na Figura 12, a obtenção dos valores de m_1 , m_2 e a interseção pode ser realizada pelas regressões lineares nos dois primeiros e dois últimos pontos da curva $C_T(\theta)/C_{T0}$.

2.7 DTR em reatores ideais

2.7.1 Reatores tubular (PFR) e batelada (BR) ideais

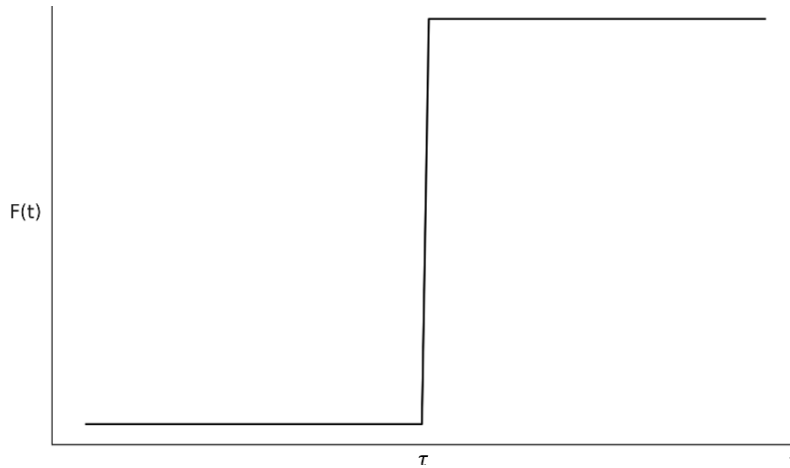
No reator tubular e batelada ideais, todos os elementos de fluido possuem o mesmo tempo de residência. A Figura 13 e Figura 14 mostram as funções $E(t)$ e $F(t)$, respectivamente.

Figura 13 - Função $E(t)$ para o reator tubular ideal.



Fonte: Adaptado de Fogler (2022a).

Figura 14 - Função $F(t)$ para o reator tubular ideal.



Fonte: Adaptado de Fogler (2022a).

Pelas Figuras 13 e 14, percebe-se que todos os elementos têm o mesmo tempo de residência de τ , de modo que $t_m = \tau$ e $\sigma^2 = 0$.

2.7.2 Reator de mistura ideal (CSTR)

No reator do tipo CSTR, sendo um tanque ideal, a mistura é perfeita, portanto a concentração no reator não varia. Aplicando o balanço para um traçador inerte, obtém-se a Equação 47, que descreve a concentração ao longo do tempo no CSTR.

$$C(t) = C_0 \cdot e^{-t/\tau} \quad (47)$$

A partir da Equação 47, pode-se encontrar as funções $E(t)$ e $F(t)$, a partir de suas definições, sendo expressas nas Equações 48 e 49, respectivamente.

$$E(t) = \frac{C(t)}{\int_0^{\infty} C(t) dt} = \frac{C_0 e^{-t/\tau}}{\int_0^{\infty} C_0 e^{-t/\tau} dt} = \frac{e^{-t/\tau}}{\tau} \quad (48)$$

$$F(t) = \int_0^t E(t) dt = \int_0^t \frac{e^{-t/\tau}}{\tau} dt = 1 - e^{-t/\tau} \quad (49)$$

Por fim, é calculado o tempo de residência médio (t_m) e a variância (σ^2) do CSTR, sendo os cálculos mostrados nas Equações 50 e 51, respectivamente. Parâmetros esses que são usados de comparação com os reatores não ideais.

$$t_m = \int_0^{\infty} t \cdot E(t) dt = \int_0^{\infty} \frac{t}{\tau} \cdot e^{-t/\tau} dt = \tau \quad (50)$$

$$\sigma^2 = \int_0^{\infty} \frac{(t - \tau)^2}{\tau} e^{-t/\tau} dt = \tau^2 \int_0^{\infty} (x - 1)^2 e^{-x} dx = \tau^2 \quad (51)$$

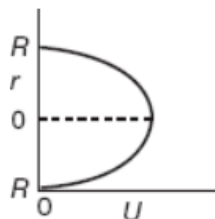
Dessa forma, pelas Equações 50 e 51, conclui-se que para o CSTR, $t_m = \tau$ e $\sigma^2 = \tau^2$.

2.7.3 Reator com escoamento laminar (LFR)

Reatores tubulares que utilizam fluidos viscosos não correspondem ao modelo ideal do PFR, porque apresentam escoamento laminar. Nesse tipo de escoamento, as moléculas perto das paredes do tubo se movem mais lentamente do que as moléculas no centro do tubo. Isso contrasta com o fluxo em pistão, no qual todas as moléculas mantêm uma velocidade uniforme ao longo da seção transversal do tubo (LEVENSPIEL, 2000).

Para o escoamento laminar em um reator tubular, o perfil de velocidade é parabólico, como demonstra a Figura 15. Nessa condição, as moléculas do centro têm tempo de residência menor que aquelas próximas a parede.

Figura 15 - Perfil parabólico para o escoamento laminar no LFR.



Fonte: Adaptado de Fogler (2022a).

O perfil de velocidade parabólico para um tubo de raio R é expresso matematicamente segundo a Equação 52.

$$U(r) = U_{m\acute{a}x} \left[1 - \left(\frac{r}{R} \right)^2 \right] = 2U_{m\acute{e}dio} \left[1 - \left(\frac{r}{R} \right)^2 \right] = \frac{2v_0}{\pi R^2} \left[1 - \left(\frac{r}{R} \right)^2 \right] \quad (52)$$

Sendo que, $U_{m\acute{a}x}$ é a velocidade no centro do tubo e $U_{m\acute{e}dio}$ é a velocidade média, que pode ser calculada pela razão entre a vazão volumétrica e a área da seção transversal.

Considerando a variação de velocidade com raio e o tempo necessário para o desenvolvimento do perfil parabólico, obtém-se a Equação 53 que expressa a distribuição dos tempos de residência.

$$E(t) = \frac{\tau^2}{2t^3} \quad (53)$$

O tempo mínimo que o fluido permanece no reator é descrito pela Equação 54.

$$t = \frac{L}{U_{m\acute{a}x}} = \frac{L}{2U_{m\acute{e}dia}} \left(\frac{\pi R^2}{\pi R^2} \right) = \frac{V}{2v_0} = \frac{\tau}{2} \quad (54)$$

Dessa forma, antes de $t < \tau/2$ a função de Distribuição de Tempos de Residência tem valor nulo, ou seja, $E(t) = 0$. Nos tempos posteriores a $\tau/2$, a $E(t)$ se comporta de acordo com a Equação 53.

A função distribuição cumulativa para o LFR é dada pela Equação 55.

$$F(t) = \int_0^t E(t) dt = 0 + \int_{\tau/2}^t E(t) dt = 1 - \frac{\tau}{4t^2} \quad (55)$$

Por fim, é possível definir o tempo de residência médio (t_m) para o LFR pela Equação 56.

$$t_m = \int_{\tau/2}^{\infty} t \cdot E(t) dt = \tau \quad (56)$$

2.8 Diagnóstico usando a DTR

O estudo da DTR apresenta duas funções importantes, pode ser utilizada para identificar problemas de operação em reatores existentes ou pode ser usada para modelar um reator real como combinações de reatores ideais, assim como mencionado na seção 3.6.

A seguir serão discutidas as não idealidades que impactam na operação do reator e podem ser visualizadas nas curvas $E(t)$ e $F(t)$ para os reatores CSTR e PFR.

2.8.1 Operações dos reatores tanques

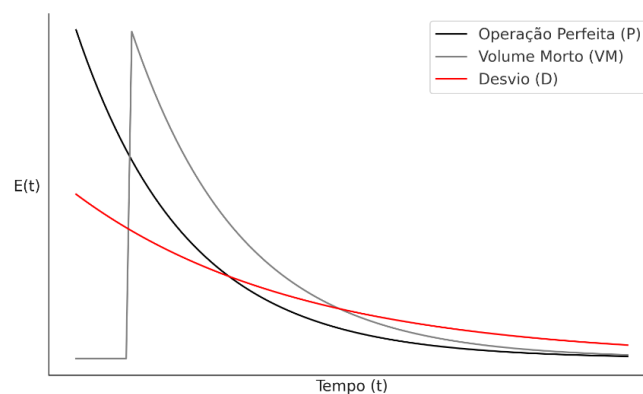
Para a operação perfeita (P), o tempo de residência pode ser expresso pela Equação 7.

Na operação com desvio de fluxo (D), parte da vazão de entrada tem o tempo médio de residência nula, ou seja, as moléculas saem imediatamente do reator.

O reator operando com volume morto (VM), apresenta uma região onde o fluido está estagnado e não há reação. Dessa forma, considera-se que o reator apresenta um volume menor que o volume da operação perfeita, o que diminui o tempo médio de residência de todas as moléculas que passam pelo equipamento.

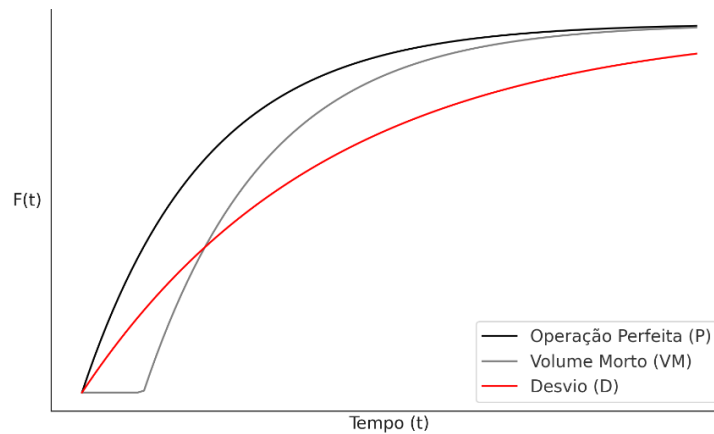
Nas Figuras 16 e 17 é possível observar os diferentes comportamentos provocados pelas não idealidades nas funções $E(t)$ e $F(t)$ no reator tanque. Na Figura 18 estão ilustrados outros comportamentos divergindo do ideal na curva $C(t)$.

Figura 16 - Impacto dos tipos de operação no CSTR na função $E(t)$.



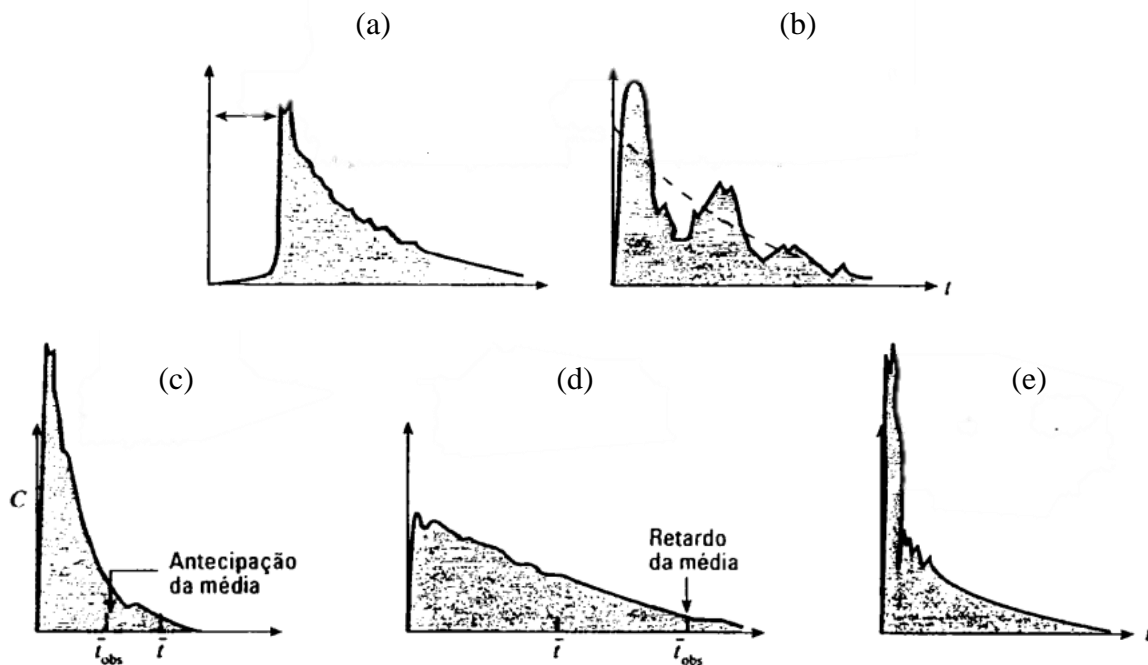
Fonte: Adaptado de Fogler (2022a).

Figura 17 - Impacto dos tipos de operação no CSTR na função $F(t)$.



Fonte: Adaptado de Fogler (2022a).

Figura 18 – Comportamentos divergindo do ideal no reator tanque: , sendo (a) volume morto, (b) circulação interna lenta (mistura inadequada), (c) fluido estagnado, (d) traçador não inerte e (e) desvio de fluxo.



Fonte: Adaptado de Levenspiel (2000).

2.8.2 Operações dos reatores tubulares

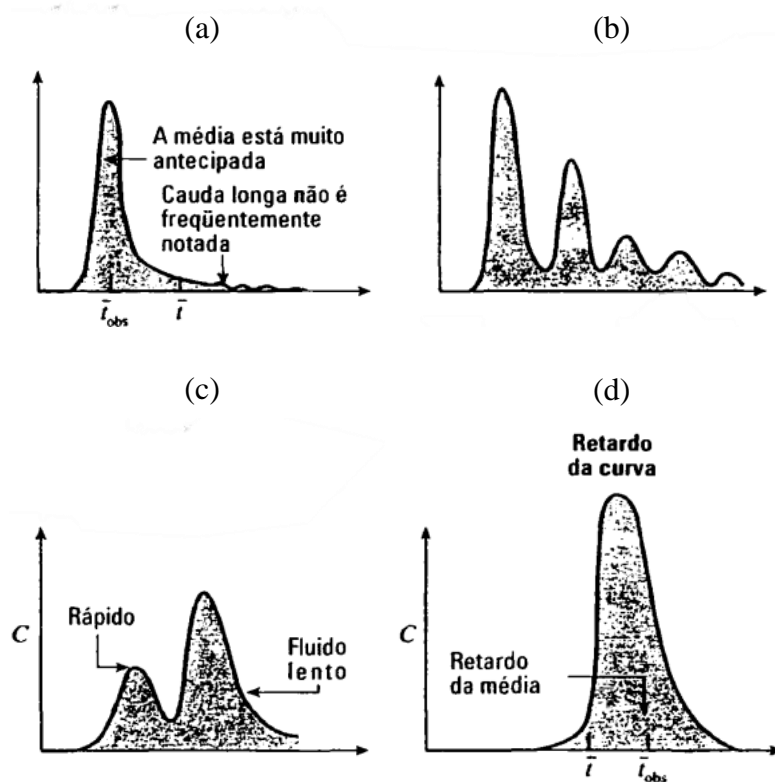
O tempo espacial de um PFR é descrito pela Equação 7.

Ao considerar caminhos preferenciais, são criados dois picos na curva de concentração de traçador por tempo. O primeiro pico expressa as moléculas desviadas, que saem mais rápido, e o segundo pico indica o fluido lento.

A operação com volume morto pode ocorrer por zonas de circulação na entrada do reator, criando uma região de estagnação, o que diminui o tempo de residência das moléculas.

Na Figura 19, é possível observar os diferentes comportamentos provocados pelas não idealidades na função $C(t)$ no reator tubular.

Figura 19 – Comportamentos divergindo do ideal no reator tubular: (a) regiões de estagnação, (b) forte circulação interna, (c) caminhos preferenciais e (d) traçador não inerte.



Fonte: Adaptado de Levenspiel (2000).

3. METODOLOGIA

O trabalho se propôs a desenvolver e analisar a distribuição de tempo de residência considerando duas técnicas de determinação experimental diferentes, perturbações dos tipos pulso e degrau. Para construção dos códigos, utilizou-se como referência o livro Elementos de Engenharia das Reações Químicas de H. Scott Fogler (2022a).

Os algoritmos foram desenvolvidos na linguagem de programação Python. Inicialmente, o programa recebe os dados de concentração e tempo do usuário a partir disso, é realizada uma regressão polinomial. Em seguida, os dados são manipulados para achar as funções DTR, $E(t)$, e DTR cumulativa, $F(t)$, tempo de residência médio (t_m), variância (σ^2) e distorção (S^3). Além disso, o programa recolhe dados cinéticos e condições de alimentação do reator fornecidos pelo usuário como a concentração inicial dos reagentes (C_{A0}), velocidade específica de reação (k), lei de velocidade da reação, volume do reator (V) e vazão volumétrica (v_0), a fim possibilitar o cálculo de conversão (X) prevista pelo modelo escolhido.

A regressão polinomial é realizada através da biblioteca *numpy* do Python, cuja utilização será detalhada no tópico 3.3. Nos códigos, é possível escolher o grau do polinômio gerado, quanto maior o número de dados, melhor será a capacidade preditiva do modelo. Para a resolução das integrais de forma numérica, foi utilizada a biblioteca *scipy*, que é responsável pelo cálculo iterativo.

3.1 Python

Python é uma linguagem de programação que apresenta bibliotecas com códigos e rotinas pré-programadas, que adicionam funções úteis e reduzem o tamanho dos programas. (PYTHON SOFTWARE FOUNDATION, 2024). Na implementação dos códigos em questão, foram usadas as bibliotecas: *numpy*, *scipy* e *matplotlib*.

A biblioteca *numpy* é amplamente utilizada e traz vários recursos matemáticos para complementar a sintaxe matemática presente no Python, nos códigos essa biblioteca foi importada com o codinome “*np*”. A biblioteca *scipy* fornece várias ferramentas para

resolução de EDO, integral e cálculos iterativos. Já a função *pyplot* da biblioteca *matplotlib* está sendo importada pelo codinome *plt* e é responsável pela apresentação dos gráficos.

A seguir estão descritos os comandos principais do Python e suas bibliotecas utilizados neste trabalho.

import: importa bibliotecas com funções prontas no Python;

def: define uma função;

return: retorna o valor de uma função;

np.array: função da biblioteca *numpy* que cria um *array* (arranjo de dados) dada uma sequência de valores;

np.linspace: função da biblioteca *numpy* que cria um *array* de valores igualmente espaçados em um intervalo especificado;

np.polyfit: função responsável pela regressão, a partir dos dados de *x*, *y* e grau de regressão respectivamente, retorna uma lista com os coeficientes da regressão polinomial;

np.poly1d: função que transforma os coeficientes advindos da função *np.polyfit* em uma função polinomial do tipo $y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$;

.integ(): realiza a integração analítica de um objeto gerado pela função *np.poly1d*;

.deriv(): realiza a derivada analítica de um objeto gerado pela função *np.poly1d*;

odeint: função da biblioteca *scipy* que resolve uma equação diferencial ordinária usando a função *Isolda* da biblioteca *odepack* da linguagem de programação FORTRAN.

3.2 Framework Django

Django é um conjunto de ferramentas que auxiliam no desenvolvimento *web*, ou seja, um conjunto de códigos e estruturas prontas para facilitar os passos iniciais de criação de um *site* e sua manutenção. Essa ferramenta se destaca por organizar o armazenamento das informações de um *site* por meio de um banco de dados, além de gerenciar as páginas *web* com segurança e ainda apresenta alta flexibilidade e facilidade de mudança na estrutura do *site* (DJANGO PROJECT, 2024).

Quando importado no Python, possibilita a integração dos códigos de resolução de cálculos com a interface do *site*, estabelecendo a conexão entre o usuário e os parâmetros de entrada para o código.

Neste trabalho, o Django foi utilizado majoritariamente na gestão de informações e envio de dados, principalmente na geração dos formulários, que recebem os valores inseridos pelo usuário e os enviam para o código Python e o caminho inverso, em que a resposta deve ser levada até o *site* para que os elementos gráficos e tabelas sejam exibidas. Essa função desempenhada pelo *framework* Django é ilustrada na Figura 20.

Figura 20 – Conexão entre o *site* e o Python através do *framework* Django.



Fonte: Do autor (2024).

Ao iniciar um projeto Django, vários arquivos são criados automaticamente dentro da pasta do projeto, dentre eles se destacam *manage.py*, *settings.py*, *urls.py*, *models.py*, *views.py*, além dos arquivos de *template*. O arquivo *manage.py* permite realizar várias tarefas úteis, como iniciar o servidor e realizar alterações no banco de dados. Em *settings.py* se encontram as configurações de segurança, pacotes externos e bibliotecas a serem implementadas e outras configurações gerais. O *urls.py* mapeia os caminhos para quais arquivos devem ser acessados de acordo com o link da página. *models.py* é responsável pelo banco de dados e definição de variáveis. *views.py* é responsável pela visualização da página e processa as solicitações do usuário e retorna as devidas respostas. Por fim os arquivos *template*, que são responsáveis pela estrutura e personalização de cada página com arquivos do tipo *.html*, *.css* e *.js*.

3.3 Regressão polinomial

No estudo de Distribuição de Tempos de Residência, os dados de concentração de traçador em função do tempo, obtidos experimentalmente, são manipulados para obter outras funções e parâmetros. Para simplificação das integrais e derivadas que surgem nos cálculos, foi escolhida a regressão polinomial para representar tais dados de concentração, indicando o coeficiente de determinação (R^2) obtido no ajuste dos dados experimentais.

A regressão polinomial foi realizada utilizando a função *polyfit()* da biblioteca numérica *numpy* do Python, que usa o método dos mínimos quadrados para definir os parâmetros. O grau do polinômio varia de acordo com cada problema e, em alguns códigos, pode ser definido pelo usuário. Considerando n o grau do polinômio, é importante garantir que o número de dados fornecidos seja no mínimo $n + 1$. Caso o usuário insira um conjunto de valores menor que o grau do polinômio, o programa retornará uma mensagem de erro.

Para a derivação e integração não definida da função polinomial obtida, foram utilizados os métodos *.deriv()* e *.integ()* do Python, que realizam as operações de forma analítica.

4. RESULTADOS E DISCUSSÃO

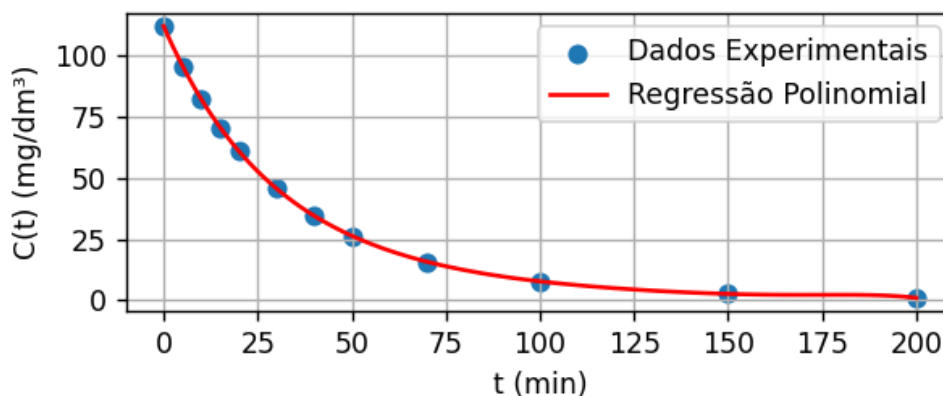
4.1 Cálculos para perturbação em pulso

Para perturbação em pulso, as Equações 2 e 3 são utilizadas para os cálculos de $E(t)$ e $F(t)$, respectivamente, através da manipulação da regressão polinomial. A princípio o ajuste polinomial apresenta grau elevado, como de sétima ordem, sendo robusto o suficiente para representar diferentes dados com uma única regressão. Um ponto negativo dessa abordagem é a dificuldade no ajuste para conjuntos de dados pequenos, o que pode levar a um grande erro nos cálculos.

Outro ponto a se considerar é a diferença no comportamento das curvas de concentração por tempo na perturbação pulso entre o reator tanque e tubular. No entanto, nenhuma modificação foi necessária, já que todos os cálculos partiram da integração da função polinomial, portanto o mesmo código consegue resolver ambas as situações.

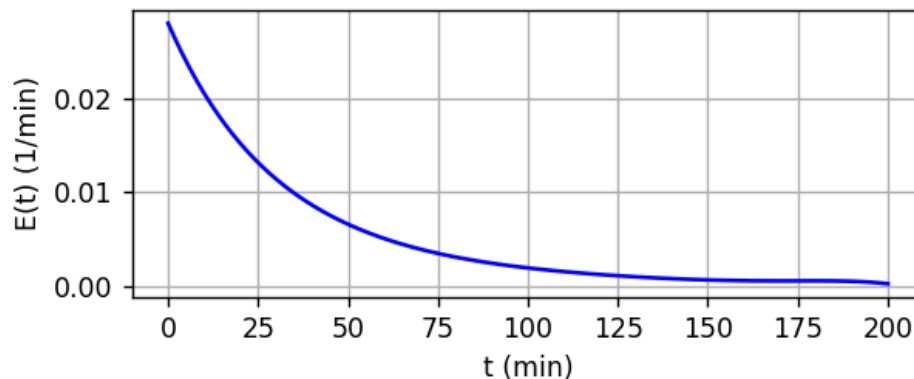
Após a submissão dos valores iniciais ao código, este retorna a conversão do reator, prevista pelos modelos de segregação e de mistura máxima, tempo de residência médio, variância e distorção da DTR. Por fim, o código exhibe as curvas de concentração, $C(t)$, Distribuição de Tempo de Residência, $E(t)$, e função de distribuição cumulativa, $F(t)$, como mostras as Figuras 21, 22 e 23, respectivamente, para os dados do Exemplo 17.4 do livro Elementos de Engenharia das Reações Químicas de H. Scott Fogler (2022a), conforme enunciado apresentado no Apêndice A.

Figura 21 - Curva de concentração por tempo para perturbação pulso em um reator tanque.



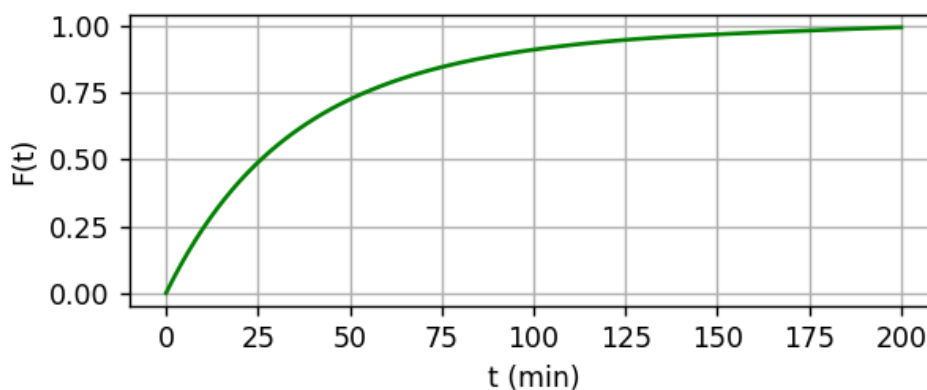
Fonte: Do autor (2024).

Figura 22 - Função de Distribuição de Tempo de Residência para perturbação pulso em um reator tanque.



Fonte: Do autor (2024).

Figura 23 - Função de distribuição cumulativa para perturbação pulso em um reator tanque.



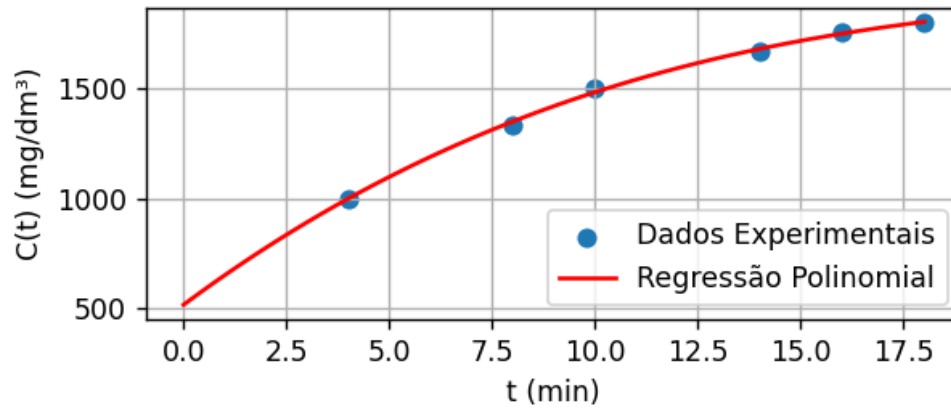
Fonte: Do autor (2024).

4.2 Cálculos para perturbação em degrau

Os cálculos para perturbação em degrau foram realizados utilizando as Equações 4 e 5 para determinação das curvas $F(t)$ e $E(t)$, respectivamente, ao passo que a curva de concentração $C(t)$ é obtida a partir da regressão dos dados obtidos experimentalmente.

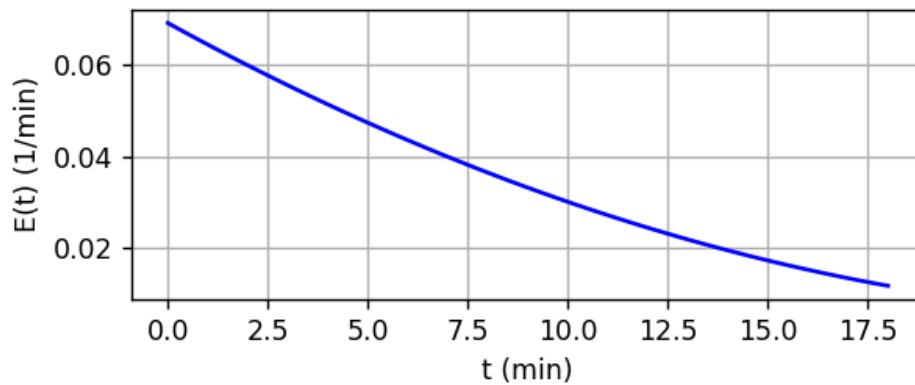
As Figuras 24, 25 e 26 demonstram as curvas de $C(t)$, $E(t)$ e $F(t)$, respectivamente, para uma perturbação do tipo degrau, com dados retirados do Exemplo 18.5 do livro Elementos de Engenharia das Reações Químicas de H. Scott Fogler (2022a), conforme enunciado apresentado no Apêndice A.

Figura 24 - Curva de concentração por tempo para perturbação degrau em um reator tanque.



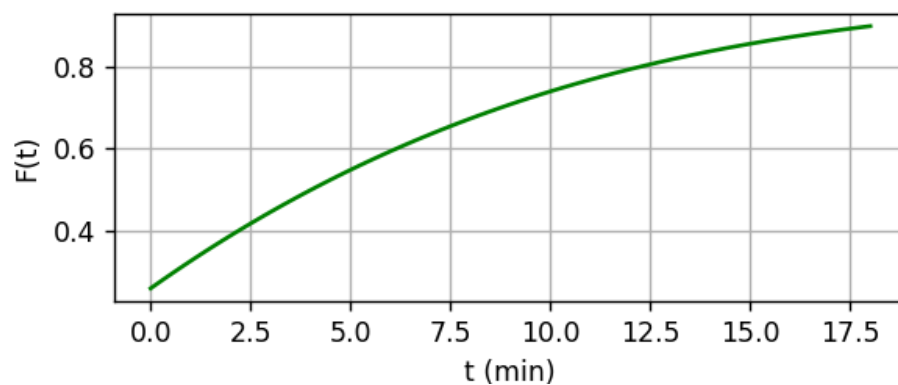
Fonte: Do autor (2024).

Figura 25 - Função de Distribuição de Tempos de Residência para perturbação degrau em um reator tanque.



Fonte: Do autor (2024).

Figura 26 - Função de distribuição cumulativa para perturbação degrau em um reator tanque.



Fonte: Do autor (2024).

4.3 Modelos para reatores não ideais usando a DTR

A modelagem para previsão de conversão em reatores não ideais pode apresentar ou não parâmetros ajustáveis. Nos modelos sem parâmetros ajustáveis, a conversão é calculada diretamente através das curvas $E(t)$ e $F(t)$. Para os modelos com um parâmetro ajustável apresentados na literatura (FOGLER, 2022a) são utilizados o tempo de residência médio (t_m) e a variância (σ^2), calculados a partir da DTR, para encontrar o parâmetro ajustável e, posteriormente, calcular a conversão. Já os modelos de dois parâmetros ajustáveis usam combinações de reatores ideais para modelagem do reator real, sendo a DTR usada para calcular os parâmetros do modelo. Em todos os casos, a lei de velocidade da reação, bem como os parâmetros cinéticos e as condições de alimentação do reator, devem ser fornecidos para a predição da conversão.

A princípio, os dados cinéticos e dados de concentração de traçador no tempo são fornecidos pelo usuário. Posteriormente, a curva de concentração de traçador é gerada a partir da regressão não linear, discutida em maiores detalhes no tópico 3.3. Obtém-se então o coeficiente de determinação (R^2), que indica a qualidade do ajuste polinomial, sendo tanto melhor quanto mais próximo da unidade for seu valor. Para os exemplos abordados neste trabalho, obteve-se valores de R^2 de no mínimo 0,98.

Em seguida, as curvas de $E(t)$ e $F(t)$ são calculadas respectivamente pelas Equações 2 e 3 se a perturbação é do tipo pulso, ou pelas Equações 4 e 5 se a perturbação é do tipo degrau. Já o tempo de residência médio (t_m), a variância (σ^2) e a inclinação (S^3) são calculados pelas Equações 6, 9 e 10, respectivamente.

Os códigos desenvolvidos neste trabalho apresentam respostas de conversão previstas pelos modelos. Tais respostas são compatíveis às dos respectivos exemplos do Fogler (2022a), considerando o mesmo número de casas decimais, as quais estão apresentadas no Apêndice A para fins de comparação. De modo que, há uma validação dos códigos desenvolvidos, apesar das diferenças nos métodos de resolução adotados.

A seguir são apresentados os cálculos para conversão de alguns modelos, com o desenvolvimento dos códigos correspondentes.

4.3.1 Segregação e mistura máxima

A construção dos códigos para os modelos de segregação e mistura máxima é baseado no Exemplo 17.4 (Apêndice A) do livro de Fogler (2022a), sendo nomeado de “DTR1 - Modelos de Segregação e Mistura Máxima”. Os parâmetros e dados de concentração do exemplo são exibidos ao abrir o programa, para que o usuário possa testar sua funcionalidade. No entanto, é possível inserir outros valores conforme a necessidade do usuário.

Ambos os modelos, de segregação e de mistura máxima, não apresentam parâmetros ajustáveis, portanto a conversão pode ser calculada diretamente pelos dados da DTR. O tempo de residência médio (t_m), a variância (σ^2) e a inclinação (S^3) são calculados para comparar o reator real com o ideal correspondente.

A conversão é obtida pela resolução iterativa da Equação 11 para o modelo de segregação e Equação 21 para o modelo de mistura máxima, considerando a reação de segunda ordem. O cálculo iterativo é realizado pela função *odeint()* da biblioteca *scipy*, que tem o objetivo de resolver equações diferenciais ordinárias.

A Figura 27 mostra o código de definição de variáveis e resolução das equações para obter as curvas da DTR, seus momentos e a conversão prevista pelos modelos de segregação e mistura máxima.

Figura 27 - Código do Exemplo DTR1: modelos de segregação e mistura máxima.

```

1 def dtr1(request):
2     # Exemplo 17.4
3     # 2A->B   ra=-kCa²
4
5     k = float(request.POST.get("k")) #dm³/(mol min)
6     CA0 = float(request.POST.get("CA0")) #mol/dm³
7     V = float(request.POST.get("V")) #dm³
8     v0 = float(request.POST.get("v0")) #dm³/min
9     N0 = float(request.POST.get("N0")) #g
10    tau= V/v0
11
12    t = np.array(json.loads(request.POST.get("t"))) #min
13    dadosC = np.array(json.loads(request.POST.get("C(t)"))) #mg/dm³
14
15    grau_de_regressao = float(request.POST.get("npolinomio"))
16
17    #Erros
18    if len(dadosC)!=len(t):
19        return {"erro":'Os vetores C(t) e t devem ter o mesmo número de dados'}
20
21    if grau_de_regressao+1>len(dadosC):
22        return {"erro":'O número de dados deve ser no mínimo n+1, sendo "n" o grau da regressão'}
23
24    regressaoC = np.polyfit(t,dadosC,grau_de_regressao)
25    C = np.poly1d(regressaoC)
26    E = C/(C.integ()(t[-1])-C.integ()(t[0]))
27    F = E.integ()
28
29    #Calculo de R²
30    y=dadosC
31    y_pred=C(t)
32    sq_res = np.sum((y - y_pred) ** 2)
33    sq_tot = np.sum((y - np.mean(y)) ** 2)
34    r_quadrado = 1 - (sq_res / sq_tot)
35
36    def segregado(CI,t):
37        X = CI[0]
38        dXdt = k*CA0*(1-X)**2
39        dXsegdt = X*E(t)
40        return [dXdt,dXsegdt]
41
42    def MM(CI,lamb):
43        X = CI[0]
44        dXdLamb = E(lamb)*X/(1-F(lamb)) -k*CA0*(1-X)**2
45        return dXdLamb
46
47    tODE = np.linspace(0,t[-1],len(t)*2)
48    CI = [0,0]
49    x = odeint(segregado,CI,tODE)
50
51    CI = 0
52    lamb = np.linspace(t[-1],0,len(t)*2)
53    xMM = odeint(MM,CI,lamb)
54
55    tm= (np.poly1d([1,0])*E).integ()
56    tm= tm(t[-1])-tm(t[0]) #min
57
58    variancia = ((np.poly1d([1,0])-tm)**2*E).integ()
59    variancia = variancia(t[-1])-variancia(t[0]) #min²
60
61    inclinacao = ((np.poly1d([1,0])-tm)**3)*E.integ()
62    inclinacao = 1/variancia**(3/2)*(inclinacao(t[-1]) - inclinacao(t[0])) #min³
63    C = list(C(tODE))
64    E = list(E(tODE))
65    F = list(F(tODE))
66
67    #X²-(2+1/tau*k*CA0)X+1 = 0 #Eq CSTR 2ª ordem
68    raizes = np.roots([1,-(2+1/(tau*k*CA0)),1])
69    raiz_valida = [raiz for raiz in raizes if 0 <= raiz <= 1]
70    XCSTR = raiz_valida[0]
71    XPFR = k*CA0*tau/(1+k*CA0*tau)
72
73    return {"t":np.round(tODE,2), "C":C, "E":E, "F":F, "DadosExperimentaisC":dadosC.tolist(),"DadosExperimentais":t.tolist(),
74            "tm":[round(tm,2)], "variancia":["{:.2e}".format(variancia)],"inclinacao":["{:.2e}".format(inclinacao,2)],
75            "XCSTR":[round(XCSTR,4)],"XPFR":[round(XPFR,4)], "Xseg":[round(x[-1,1],4)], "XMM":[round(xMM[-1,0],4)],
76            "r_quadrado": [round(r_quadrado,4)]}

```

Fonte: Do autor (2024).

4.3.2 TES

O modelo de tanques em série (TES) tem um parâmetro ajustável, n , que indica o número de reatores ideais em série necessário para representar o reator real.

Para o desenvolvimento do modelo TES foi utilizada a Equações 24 e regressão polinomial para se obter as curvas $E(t)$. Para encontrar os valores de tempo de residência médio (t_m), variância (σ^2) e inclinação (S^3) foi realizada a regressão polinomial dos dados de concentração de traçador ao longo do tempo, com o intuito de padronizar o método de obtenção desses parâmetros.

Para o cálculo da conversão, segue-se alguns passos. Primeiro, é obtida a curva $C(t)$ através da regressão polinomial; em seguida, encontra-se a função $E(t)$, através da Equação 2, e, assim, através das Equações 23, 24 e 25, obtém-se uma estimativa inicial para n . Com o valor de n , têm-se novos valores para a função $E(t)$, obtidos pela Equação 22 e, subsequentemente, um novo valor para n , que representará o número de reatores ideais ligados em série previsto pelo modelo.

A Figura 28 mostra o código de definição de variáveis e resolução das equações para obter as curvas da DTR, seus momentos, o parâmetro ajustável e a conversão prevista pelo modelo de tanques em séries (TES), considerando os dados do Exemplo 18.2 do livro de Fogler (2022a), cujo enunciado está descrito no Apêndice A e está nomeado como “DTR3 - Modelo de tanques em série”.

Figura 28 - Código do Exemplo DTR3: modelo de tanques em série.

```

1 def dtr3(request):
2     k = float(request.POST.get("k")) #dm³/(mol min)
3     CA0 = float(request.POST.get("CA0")) #mol/dm³
4     V = float(request.POST.get("V")) #dm³
5     v0 = float(request.POST.get("v0")) #dm³/min
6
7     t = np.array(json.loads(request.POST.get("t"))) #min
8     dadosC = np.array(json.loads(request.POST.get("C(t)")))#mg/dm³
9
10    grau_de_regressao = float(request.POST.get("npolinomio"))
11
12    #Erros
13    if len(dadosC)!=len(t):
14        return {"erro":'Os vetores C(t) e t devem ter o mesmo número de dados'}
15
16    if grau_de_regressao+1>=len(dadosC):
17        return {"erro":'O número de dados deve ser no mínimo n+1, sendo "n" o grau da regressão'}
18
19    regressaoC = np.polyfit(t,dadosC,grau_de_regressao)
20    C = np.poly1d(regressaoC)
21    E = C/(C.integ()(t[-1])-C.integ()(t[0]))
22
23    tau = (np.poly1d([1,0])*E).integ()
24    tau = tau(t[-1])-tau(t[0]) #min
25    variancia = ((np.poly1d([1,0])-tau)**2*E).integ()
26    variancia = variancia(t[-1])-variancia(t[0]) #min²
27
28    n = (tau**2/variancia)
29    tauí = tau/n
30
31    #Conversão TES 1ª ordem
32    XTES = 1 - 1/(1+tauí*k)**n
33
34    regressaoC = np.polyfit(t,dadosC,grau_de_regressao)
35    C = np.poly1d(regressaoC)
36    E = C/(C.integ()(t[-1])-C.integ()(t[0]))
37    F = E.integ()
38
39    #Calculo de R²
40    y=dadosC
41    y_pred=C(t)
42    sq_res = np.sum((y - y_pred) ** 2)
43    sq_tot = np.sum((y - np.mean(y)) ** 2)
44    r_quadrado = 1 - (sq_res / sq_tot)
45
46    tm= (np.poly1d([1,0])*E).integ()
47    tm= tm(t[-1])-tm(t[0]) #min
48
49    variancia = ((np.poly1d([1,0]) - tm)**2*E).integ()
50    variancia = variancia(t[-1])-variancia(t[0]) #min²
51    inclinacao = ((np.poly1d([1,0])-tm)**3*E).integ()
52    inclinacao = 1/variancia**(3/2)*(inclinacao(t[-1])-inclinacao(t[0])) #min³
53
54    tODE = np.linspace(0,t[-1],len(t)*2)
55    C = list(C(tODE))
56    E = list(E(tODE))
57    F = list(F(tODE))
58
59    return {"t":np.round(tODE,3), "C":C, "E":E, "F":F, "DadosExperimentaisC":dadosC.tolist(),"DadosExperimentais":t.tolist(),
60            "tm":[round(tau,2)],"variancia":["{: .2e}"].format(variancia),"inclinacao":["{: .2e}"].format(inclinacao,2),
61            "XTES":[round(XTES,4)], "n":[n], "r_quadrado": [round(r_quadrado,4)]}

```

Fonte: Do autor (2024).

4.3.3 Tanque com desvio e espaço morto

O modelo de desvio de fluxo e espaço morto apresenta dois parâmetros ajustáveis, α e β , definidos nas Equações 29 e 30, respectivamente. Os dados iniciais, bem como os parâmetros do problema, foram retirados do Exemplo 18.5 do livro de Fogler (2022a).

Esses parâmetros são calculados através da linearização da equação proveniente do balanço de massa do traçador (Equação 35), como mostrado na Equação 57.

$$\ln\left(\frac{C_{T0}}{C_{T0}-C_T}\right) = \ln\left(\frac{1}{1-\beta}\right) + \left(\frac{1-\beta}{\alpha}\right) \frac{t}{\tau} \quad (57)$$

Assim, pela Equação 57, a reta gerada por $\ln\left(\frac{C_{T0}}{C_{T0}-C_T}\right)$ versus t apresenta inclinação de $(1-\beta)/(\tau\alpha)$ e a interseção em $\ln\left[\frac{1}{1-\beta}\right]$, sendo possível calcular α e β utilizando a regressão linear através da biblioteca *scipy*, por meio da função *linregress*, que fornece os valores de inclinação e interseção.

A conversão prevista pelo modelo é calculada para uma reação de segunda ordem, utilizando as Equações 33 e 34.

A Figura 29 mostra o código de definição de variáveis e resolução das equações para obter as curvas da DTR, seus momentos, os parâmetros ajustáveis e a conversão prevista pelo modelo de desvio e espaço morto, considerando os dados do Exemplo 18.5 do livro de Fogler (2022a), cujo enunciado está descrito no Apêndice A e está nomeado como “DTR2 - Modelo de espaço morto e desvio”.

Figura 29 - Código do Exemplo DTR2: modelo de desvio e espaço morto.

```

1 def dtr2(request):
2     k = float(request.POST.get("k")) #m³/(kmol min)
3     CA0 = float(request.POST.get("CA0")) #kmol/m³
4     CT0 = float(request.POST.get("CT0")) #mg/dm³
5     tau = float(request.POST.get("tau")) #min
6
7     # CSTR degrau
8     t = np.array(json.loads(request.POST.get("t"))) #min
9     dadosC = np.array(json.loads(request.POST.get("C(t)"))) #mg/dm³
10
11     lndadosC = []
12     for C in dadosC:
13         lnC = np.log(CT0/(CT0-C))
14         lndadosC.append(lnC)
15
16     a, b, r_value, p_value, std_err = linregress(t/tau, lndadosC)
17     beta= 1 - np.exp(-b)
18     alfa = (1-b)/a
19     tauS = alfa*tau/(1-beta)
20
21     #ra = - k* CAs**2 - Segunda ordem
22     CAs = (-1 + (1+4*tauS*k*CA0)**(1/2))/(2*tauS*k)
23     CA = beta*CA0 + (1-b)*CAs
24     Xmodelo = (CA0-CA)/CA0
25
26     # REATOR IDEAL
27     # X/(1-X)^2 = k*tau*CA0
28     const = k*tau*CA0
29     # const*X^2-(2*const+1)*X + const
30     X = ((2*const+1) - ((2*const+1)**2-4*const*const)**0.5)/2/const #baskara negativo
31     if X<0 or X>1:
32         X = ((2*const+1) + ((2*const+1)**2-4*const*const)**0.5)/2/const #baskara positivo
33
34     grau_de_regressao = float(request.POST.get("npolinomio"))
35     regressaoC = np.polyfit(t,dadosC,grau_de_regressao)
36     C = np.poly1d(regressaoC)
37     F = C/CT0
38     E = F.deriv()
39
40     tm= (np.poly1d([1,0])*E).integ()
41     tm= tm(t[-1])-tm(t[0]) #min
42
43     variancia = (np.poly1d([1,0])*E - tm*E)**2
44     variancia = variancia.integ()
45     variancia = variancia(t[-1])-variancia(t[0]) #min²
46
47     inclinacao = ((np.poly1d([1,0])-tm)**3*E).integ()
48     inclinacao = 1/variancia**(3/2)*(inclinacao(t[-1])-inclinacao(t[0])) #min³
49
50     tODE = np.linspace(0,t[-1],100)
51     C = list(C(tODE))
52     E = list(E(tODE))
53     F = list(F(tODE))
54
55     return {"t":np.round(tODE,1), "C":C, "E":E, "F":F, "tm":[round(tm,2)],
56           "variancia":["{: .2e}"].format(variancia),"inclinacao":["{: .2e}"].format(inclinacao,2)],
57           "Xideal":round(X,4), "Xmodelo":round(Xmodelo,4), "alfa":round(alfa,2)],
58           "beta":round(beta,2), "tauS":round(tauS,2)}

```

Fonte: Do autor (2024).

4.3.4 Tanques interconectados

O modelo de tanques interconectados modela um CSTR real como dois CSTR interconectados, um altamente agitado e o outro com menor agitação. Neste modelo há dois parâmetros ajustáveis, α que representa a fração de volume altamente agitado e β que diz sobre a fração do escoamento total que circula entre as regiões de alta e baixa agitação.

O código recebe os dados de concentração de traçador ao longo do tempo e os parâmetros volume (V), vazão volumétrica (v_0) e velocidade específica de reação (k).

Para o cálculo dos parâmetros ajustáveis, calcula-se $\theta = t/\tau$. A partir disso, é realizada a regressão polinomial de sexto grau, no caso do exemplo avaliado, para encontrar a função $C(\theta)$ e, em seguida, calcular parâmetro α pela Equação 45.

Posteriormente, o gráfico da Figura 12 é gerado e obtém-se os valores de m_1 , m_2 e I . E, por meio da Equação 46, é possível encontrar o segundo parâmetro ajustável, β .

O valor da conversão prevista pelo modelo é encontrado pela Equação 40, substituindo os parâmetros do modelo e parâmetros cinéticos. O código também exibe os valores de conversão do CSTR e PFR para comparação.

A Figura 30 mostra o código de definição de variáveis e resolução das equações para obter as curvas da DTR, seus momentos, os parâmetros ajustáveis e a conversão prevista pelo modelo de tanques interconectados, comparando com as conversões de um CSTR e um PFR.

A construção do código foi baseada nos dados do Exemplo 18.6 do livro de Fogler (2022a), cujo enunciado está descrito no Apêndice A e está nomeado como “DTR4 - Modelo de tanques interconectados”.

Figura 30 – Código do Exemplo DTR4: modelo de tanques interconectados.

```

1 def dtr4(request):
2     V = 1000 #dm³
3     v0 = 25 #dm³/min
4     tau = V/v0
5     k = 0.03 #1/min
6
7     t = np.array([0, 20, 40, 60, 80, 120, 160, 200, 240, 280, 320]) #min
8     dadosC = np.array([2000, 1050, 520, 280, 160, 61, 29, 16.4, 10, 6.4, 4]) #mg/dm³
9     teta=t/tau
10    CC0=dadosC/dadosC[0]
11
12    regressaoC = np.polyfit(teta,dadosC,6)
13    regressaoC = np.poly1d(regressaoC)
14    CT0 = dadosC[0]
15
16    alfa = (regressaoC.integ()(teta[-1])-regressaoC.integ()(teta[0]))/CT0
17    coeficientes = np.polyfit([teta[-3],teta[-2]],[CC0[-3],CC0[-2]],1)
18
19    m1 = coeficientes[0]
20    interceptacao1 = coeficientes[1]
21    coeficientes = np.polyfit([teta[0],teta[1]],[CC0[0],CC0[1]],1)
22    m2 = coeficientes[0]
23    interceptacao2 = coeficientes[1]
24    beta = interceptacao2*alfa*(m1-m2)+alfa*m2+1
25    XCSTR = tau*k/(1+tau*k)
26    XPFR = 1-np.exp(-tau*k)
27    numeradorX = (beta + alfa * tau * k) * (beta + (1 - alfa) * tau * k) - beta**2
28    denominadorX = ((1 + beta + alfa * tau * k) * (beta + (1 - alfa) * tau * k) - beta**2)
29    X = numeradorX / denominadorX
30
31    grau_de_regressao = 6
32    regressaoC = np.polyfit(t,dadosC,grau_de_regressao)
33    C = np.poly1d(regressaoC)
34    E = C/(C.integ()(t[-1])-C.integ()(t[0]))
35    F = E.integ()
36
37    tm= (np.poly1d([1,0])*E).integ()
38    tm= tm(t[-1])-tm(t[0]) #min
39
40    variancia = ((np.poly1d([1,0]) - tm)**2*E).integ()
41    variancia = variancia(t[-1])-variancia(t[0]) #min²
42    inclinacao = ((np.poly1d([1,0])-tm)**3*E).integ()
43    inclinacao = 1/variancia**(3/2)*(inclinacao(t[-1])-inclinacao(t[0])) #min³
44
45    tODE = np.linspace(0,t[-1],100)
46    C = list(C(tODE))
47    E = list(E(tODE))
48    F = list(F(tODE))
49
50    return {"t":np.round(tODE,1), "C":C, "E":E, "F":F, "tm":[round(tm,2)],
51           "variancia":["{: .2e}"].format(variancia),"inclinacao":["{: .2e}"].format(inclinacao,2)],
52           "X":[round(X,4)], "XCSTR":[round(XCSTR,4)], "XPFR":[round(XPFR,4)],
53           "m1":[round(m1,4)], "m2":[round(m2,4)], "Intersecao": [round(interceptacao2,4)]}
54

```

Fonte: Do autor (2024).

4.4 Ferramenta web

O projeto integrou todos os códigos em uma interface na *web*, a qual é capaz de receber os dados do usuário e gerar um gráfico e uma pequena tabela com os resultados. O *site* foi confeccionado utilizando o *Framework* Django, que faz a comunicação entre as páginas da internet e os códigos na linguagem de programação Python.

Na Figura 31 está a demonstração do *site*, atualmente com endereço: <https://projetandoreatores.fly.dev/exemplos/dtr1> (BARBOSA; EVANGELISTA; OLIVEIRA, 2024). Ao entrar na página, o enunciado do problema é apresentado, além da barra de navegação lateral, a partir da qual pode-se navegar entre os demais exemplos.

Figura 31 - Apresentação da ferramenta *web*.

The screenshot shows a web application titled "Projetando Reatores". The main content area is titled "DTR1 - Modelos de Segregação e Mistura Máxima". The text describes a problem involving a DTR reactor with pulse perturbation, based on Example 17.4 from the book "Elementos de Engenharia das Reações Químicas" by H. Scott Fogler. The reaction is $2A \rightarrow B$ with rate $r_A = -kC_A^2$. The problem asks for the limits of conversion for different degrees of micromixing. The output concentration is given by the polynomial $C(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + \dots + a_nt^n$. A sidebar on the left lists various reactor types and topics like "PFR", "Reações Simples", "Reações Paralelas", "Reações Complexas", "Reações Complexas e Reversíveis", "CSTR", "Ponto de Operação", "Múltiplos Pontos de Operação", "Pontos de Operação com Reação Reversível", "Batelada", "Efeitos Térmicos", "Reações Exotérmicas Descontroladas", and "Reações Múltiplas em".

Fonte: Do autor (2024).

Em seguida, após o enunciado, encontram-se os formulários, nos quais são inseridos os dados pelo usuário. Em cada formulário há o nome da variável, o campo a ser preenchido e a unidade de medida esperada para aquela variável. Além disso, abaixo dos formulários há a lista de símbolos que descreve as siglas e abreviações utilizadas na página. Os formulários podem ser visualizados na Figura 32.

Figura 32 - Formulários e customização de dados pelo usuário da ferramenta *web*.

$C(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + \dots + a_nt^n$

t: [0, 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200] min

C(t): [112, 95.8, 82.2, 70.6, 60.9, 45.6, 34.5, 26.3, 15.7, 7.67, 2.55, 0.90] mg/dm³

k: 0,01 dm³/(mol min)

C_{A0}: 8 mol/dm³

V: 1000 dm³

v₀: 25 dm³/min

N₀: 100 g

n_{polinomio}: 7

Lista de Símbolos [?](#) Enviar

Fonte: Do autor (2024).

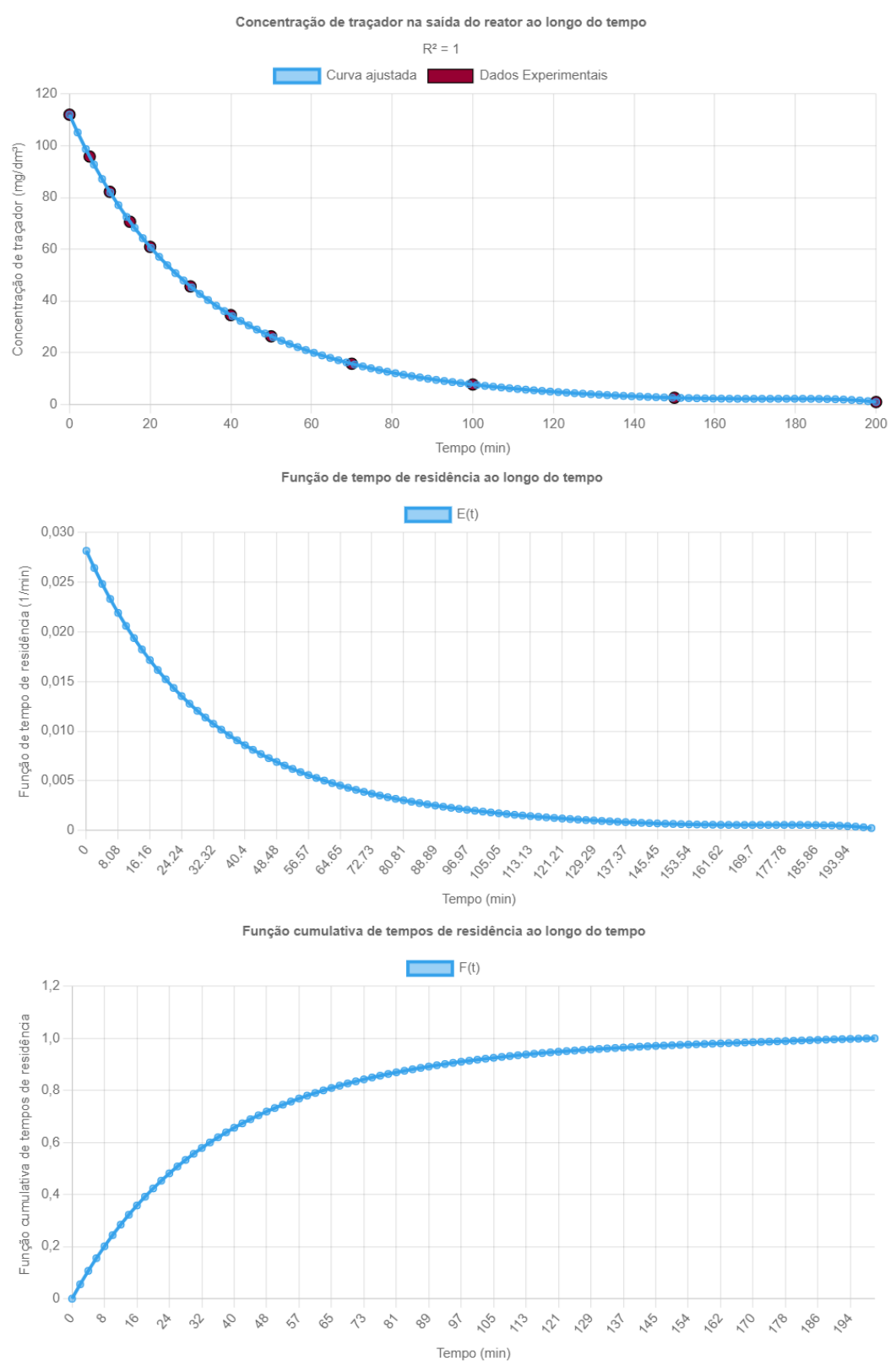
Na Figura 33, é possível ver os resultados. Sendo 1, 2 e 3 obtidos para diferentes valores de entrada fornecidos pelo usuário, permitindo assim comparar diferentes cenários de projeto. Já na Figura 34 encontram-se os gráficos gerados pela ferramenta, sendo possível comparar os dados experimentais com a curva gerada pela regressão polinomial e, conseqüentemente, avaliar visualmente se o grau da regressão adotado é adequado para o conjunto de dados em questão.

Figura 33 – Resultados da ferramenta *web*.

Resultado 1	Resultado 2	Resultado 3
tm = 38.5 min	tm = 39.65 min	tm = 38.5 min
$\sigma^2 = 1.06e+01 \text{ min}^2$	$\sigma^2 = 1.13e+01 \text{ min}^2$	$\sigma^2 = 1.06e+01 \text{ min}^2$
$S^3 = 1.23e+05 \text{ min}^3$	$S^3 = 1.09e+05 \text{ min}^3$	$S^3 = 1.23e+05 \text{ min}^3$
X = 0.9412	X = 0.9412	X = 0.9412
Xseg = 0.6124	Xseg = 0.6146	Xseg = 0.6124
XMM = 0.5623	XMM = 0.5638	XMM = 0.5623

Fonte: Do autor (2024).

Figura 34 - Respostas gráficas da ferramenta *web*.



Fonte: Do autor (2024).

5. CONCLUSÃO

Através do exposto, é possível concluir que os códigos apresentaram resultados compatíveis com os exemplos da bibliografia de H. Scott Fogler (2022a), Elementos de Engenharia das Reações Químicas, e ainda possibilitou a variação dos dados da DTR e parâmetros cinéticos, permitindo a análise do impacto de cada variável no processo.

De forma geral, percebe-se que o algoritmo permite o cálculo de conversão do reator com base em diferentes modelos, como de segregação, mistura máxima, tanques em série (TES), com desvio e espaço morto e tanques interconectados. Além disso, os momentos da DTR, tempo de residência médio (t_m), variância (σ^2) e inclinação (s^3) são calculados, possibilitando a comparação do reator real com um reator ideal. Por fim, o programa desenvolvido retorna gráficos das funções de concentração de traçador, $C(t)$, Distribuição de Tempos de Residência, $E(t)$, e distribuição cumulativa, $F(t)$, possibilitando que o usuário visualize a diferença dos tipos de perturbação e identifique as não idealidades do processo de forma simples e didática.

Portanto, entende-se que o trabalho gerou resultados importantes e relevantes no contexto de ensino de Engenharia Química, bem como para profissionais da área que atuam no projeto de reatores, uma vez que não há *site* similar disponível gratuitamente em português. A ferramenta desenvolvida é robusta, sendo capaz de prever a conversão de reatores não ideais a partir de cinco modelos, além de permitir a variação dos dados de entrada e o estudo do impacto de cada variável na conversão. Além disso, destaca-se a alta acessibilidade proporcionada pelo recurso *web* totalmente gratuito, sendo um *site* que pode ser acessado em qualquer sistema operacional com acesso à internet, seja em computadores antigos ou em celulares, eliminando a necessidade de *download* e alta capacidade de processamento para executar o programa.

Apesar da robustez do projeto, algumas melhorias podem ser feitas em trabalhos futuros. Uma delas é devido ao programa estar hospedado em um servidor gratuito, o que acarreta algumas limitações de processamento. Dentre as soluções possíveis, a serem avaliadas, pode-se mencionar a avaliação da troca do servidor e a otimização dos códigos, possibilitando uma navegação mais fluida. Acrescenta-se também a necessidade do programa

ser testado para diferentes condições de entrada, de forma ampla, a fim de verificar possíveis limitações e erros.

Outras sugestões de melhorias incluem a implementação dos modelos para diferentes ordens de reação, e de mais modelos, como o de dispersão, além da construção de códigos específicos para reatores tanque e tubulares, o que permitirá abordagens mais específicas para dados tipos de problemas. Ademais, pode-se repensar a forma de inserção dos dados de concentração de traçador e tempo no formulário, como exemplo, permitindo a importação desses dados de uma planilha.

REFERÊNCIAS

- ABEDSOLTAN, H. COVID -19 and the chemical industry: impacts, challenges, and opportunities. **Journal of chemical technology and biotechnology**, v. 98(12), p. 2789-2797, 2023.
- ASE DEVELOPERS. **Atomic simulation environment documentation**. 2024. Disponível em: <<https://wiki.fysik.dtu.dk/ase/>>. Acesso em: 27 abr. 2024.
- ABIQUIM. **O desempenho da indústria química brasileira**. São Paulo, 2023. Disponível em: <https://www.enaiq.org.br/wp-content/uploads/2023/12/desempenho_industria_quimica_brasileira.pdf>. Acesso em: 03 mai. 2024.
- BARBOSA, R. S; EVANGELISTA, N. S.; OLIVEIRA, N. M. B. **Projetando reatores**. 2024. Disponível em: <<https://projetandoreatores.fly.dev/exemplos/dtr1/>>. Acesso em: 03 mai. 2024.
- CACCAVALE, F. et al. SPyCE: a structured and tailored series of Python courses for (bio)chemical engineers. **Education for chemical engineers**, v. 45, p. 90–103, 2023.
- CANTERA DEVELOPERS. **Cantera homepage**. 2024. Disponível em: <<https://cantera.org/>>. Acesso em: 27 abr. 2024.
- COATINGS WORLD. **Chemical industry agrees on global standard for calculating product carbon footprint**. 2022. Disponível em: <https://www.coatingsworld.com/issues/2022-10-01/view_breaking-news/chemical-industry-agrees-on-global-standard-for-calculating-product-carbon-footprint/?widget=listSection>. Acesso em: 28 abr. 2024.
- DAHLGREN, B. ChemPy: a package useful for chemistry written in Python. **The journal of open source software**, v. 3(24), p. 565, 2018.
- DJANGO PROJECT. **Django documentation**. 2021. Disponível em: <<https://docs.djangoproject.com/en/4.2/>>. Acesso em: 11 mar. 2024.
- FLORIAN, B. et al. **The state of the chemical industry - it is getting more complex**. 2020. Disponível em: <https://www.mckinsey.com/industries/chemicals/our-insights/the-state-of-the-chemical-industry-it-is-getting-more-complex#>. Acesso em: 19 abr. 2024.
- FOGLER, H. S. **Elementos de engenharia das reações químicas**. 6. ed. Rio de Janeiro: LTC, 2022a.
- FOGLER, H. S. **Elementos de engenharia das reações químicas**. University of Michigan, 2022b. Disponível em: <<https://public.websites.umich.edu/~elements/5e/index.html>>. Acesso em: 27 abr. 2024.

- GAO, H. et al. A mini review of the recent progress in coarse-grained simulation of polymer systems. **Chinese journal of structural chemistry**, v. 43(5), 2024.
- GAO, Y.; MUZZIO, F. J.; IERAPETRITOU, M. G. A review of the Residence Time Distribution (RTD) applications in solid unit operations. **Powder technology**, v. 228, p. 416-423, 2012.
- LEITE, J. P. et al. Avaliação preliminar do impacto de softwares de simulação no ensino das engenharias química e de petróleo. **Engevista**, v. 16(1), p. 28-40, 2014. Disponível em: <<https://periodicos.uff.br/engevista/article/view/8993>>. Acesso em: 12 nov. 2023.
- LEVENSPIEL, Octave. **Engenharia das reações químicas**. São Paulo, SP: E. Blücher, c2000.
- MARKETS AND MARKETS. **Global chemical industry outlook 2024**: summary document. 2024. Disponível em: <<https://www.marketsandmarkets.com/blog/CM/Global-Chemical-Industry-Outlook-2024-Summary-Document>>. Acesso em: 28 abr. 2024.
- NIST. **FiPy**: a finite volume PDE solver using python. 2024. Disponível em: <<https://www.ctcms.nist.gov/fipy/>>. Acesso em: 27abr. 2024.
- PYFR TEAM. **PyFR homepage**. 2024. Disponível em: < <https://pyfr.org/>>. Acesso em: 27 abr. 2024.
- PYTHON SOFTWARE FOUNDATION. **Python documentation**. 2023. Disponível em: <<https://docs.python.org/3/>>. Acesso em: 11 mar. 2024.
- SILVA, R. A.; VASCONCELOS, F. C. G. C. Softwares de simulação no ensino de química: uma perspectiva através do m-learning. **Revista brasileira de ensino de ciência e tecnologia**, v. 14(1), 2021.
- RAMÍREZ, J. et al. A virtual laboratory to support chemical reaction engineering courses using real-life problems and industrial software. **Education for chemical engineers**, v. 33, p. 36-44, 2020.
- SUTHAR, K. J.; JOSHIPURA, M. H. The integrative approach of learning chemical engineering thermodynamics by using simulation-based exercises. **Education for chemical engineers**, v. 45, p. 122-129, 2023.
- YANG, J. et al. Sustainability performance of global chemical industry based on green total factor productivity. **The science of the total environment**, v. 830, p. 154787, 2022.

APÊNDICE A – Enunciados e resultados dos problemas

A.1 DTR1 - Modelos de segregação e mistura máxima

Trata-se da DTR em um reator com perturbação do tipo pulso, sendo usados os modelos de segregação e de mistura máxima para estimar a conversão, com base no Exemplo 17.4 da 6ª edição do livro Elementos de Engenharia das Reações Químicas de H. Scott Fogler (2022a).

Considera-se a dimerização de segunda ordem em fase líquida, conforme Equação A.1, cuja lei de velocidade é dada pela Equação A.2, para a qual $k = 0,01 \text{ dm}^3/(\text{mol} \cdot \text{min})$. A reação ocorre a 320 K, com alimentação é A puro, sendo $C_{A0} = 8 \text{ mol/dm}^3$. O reator é não ideal, de 1.000 dm^3 e possui vazão de alimentação de $25 \text{ dm}^3/\text{min}$.



$$r_A = -kC_A^2 \quad (\text{A.2})$$

Deseja-se conhecer os limites para a conversão, considerando diferentes graus possíveis de micromistura, a partir de ensaio de DTR realizado nesse reator. Quais são esses limites?

Sabe-se que foi realizado um teste lançando 100 g de traçador ($N_0 = 100 \text{ g}$) no reator, com a vazão (v_0) de $25 \text{ dm}^3/\text{min}$ e, em seguida, foram medidas as concentrações de saída, $C(t)$, em função do tempo t . A curva C é ajustada por um polinômio de grau n , como mostra a Equação A.3, grau esse que pode ser definido pelo usuário.

$$C(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + \dots + a_nt^n \quad (\text{A.3})$$

Os resultados para este exemplo, tanto os apresentados por Fogler (2022a) quanto os obtidos pelos códigos desenvolvidos neste trabalho, estão apresentados na Figura A.1.

Figura A.1 - Resultados referência (a) de Fogler (2022a) e os obtidos (b) pela ferramenta *web* para o Exemplo 17.4 relativo à aplicação dos modelos sem parâmetros ajustáveis.

(a)	(b)
$X_{PFR} = 0,76$	$t_m = 38.5 \text{ min}$
$X_{Seg} = 0,61$	$\sigma^2 = 1.51e+03 \text{ min}^2$
$X_{CSTR} = 0,58$	$S^3 = 7.17e+01 \text{ min}^3$
$X_{mm} = 0,56$	$XCSTR = 0.5758$
	$XPFR = 0.7619$
	$Xseg = 0.6124$
	$XMM = 0.5623$

Fonte: Do autor (2024).

A.2 DTR2 - Modelo de espaço morto e desvio

Trata-se do uso de traçador para determinar os parâmetros do modelo considerando um CSTR com espaço morto e desvio de fluxo (Figura 9), o qual será usado para representar um reator tanque como o ilustrado na Figura 8. Esse caso foi baseado no Exemplo 18.5 da 6ª edição do livro Elementos de Engenharia das Reações Químicas de H. Scott Fogler (2022a).

Os dados de concentração com tempo do traçador foram obtidos a partir de uma perturbação em degrau, sendo a concentração de entrada do traçador $C_{T0} = 2.000 \text{ mg/dm}^3$. Os parâmetros α e β do modelo podem ser obtidos pela regressão linear como apresentada na Equação 57 e a conversão prevista pelo modelo é encontrada a partir da Equação 33, considerando o balanço molar no reator ideal, com volume V_s , como mostra a Equação 34.

Para este exemplo, pede-se que sejam determinados os parâmetros do modelo α e β , como definidos pelas Equações 29 e 30, respectivamente. Já os dados cinéticos para cálculo da conversão são: $C_{A0} = 2 \text{ kmol/m}^3$, $\tau = 10 \text{ min}$ e $k = 0,28 \text{ m}^3/(\text{kmol}\cdot\text{min})$, indicando uma reação de segunda ordem. Os resultados obtidos para este exemplo estão apresentados na Figura A.2.

Figura A.2 - Resultados referência (a) de Fogler (2022a) e os obtidos (b) pela ferramenta *web* para o Exemplo 18.5 relativo à modelagem de tanque com espaço morto e desvio.

(a)	(b)
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $\alpha = 0,7e$ $\beta = 0,2$ </div>	$t_m = 3.76 \text{ min}$ $\sigma^2 = 3.97e-01 \text{ min}^2$ $S^3 = 7.16e+02 \text{ min}^3$
$\tau_s = \frac{V_s}{v_s} = 8,7 \text{ min}$	$X_{ideal} = 0.6574$ $X_{modelo} = 0.5218$
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $X_{modelo} = 0,51$ $X_{ideal} = 0,66$ </div>	$\text{alfa} = 0.68$ $\text{beta} = 0.19$ $\text{tauS} = 8.43 \text{ min}$

Fonte: Do autor (2024).

A.3 DTR3 - Modelo de tanques em série

Para avaliar a modelagem de tanques em série, foi realizado um experimento com uso de traçador para determinar o número de reatores tanques em série (n) necessários, com base no Exemplo 18.2 da 6ª edição do livro Elementos de Engenharia das Reações Químicas de H. Scott Fogler (2022a), como descrito a seguir.

O ensaio experimental foi realizado em um reator tubular com 10 cm de diâmetro e 6,36 m de comprimento para obter dados de concentração do traçador ao longo do tempo, a partir de uma perturbação do tipo pulso. Neste reator ocorre uma reação com $k = 0,25 \text{ min}^{-1}$.

Qual será o número de reatores em série (n) necessário para simular o comportamento desse reator tubular e a conversão prevista pelo modelo TES? Os resultados apresentados na referência (FOGLER, 2022a) e os obtidos neste trabalho, para este exemplo, são os apresentados na Figura A.3.

Figura A.3 - Resultados referência (a) de Fogler (2022a) e os obtidos (b) pela ferramenta *web* para o Exemplo 18.2 relativo à modelagem de reator tubular como n CSTR em série.

(a)	(b)
$t_m = 5,15 \text{ minutos}$ $\sigma^2 = 6,2 \text{ minutos}^2$	
$n = 4,35$	$t_m = 5.01 \text{ min}$ $\sigma^2 = 5.05e+00 \text{ min}^2$ $S^3 = 3.97e-01 \text{ min}^3$
PFR: $X = 72,5\%$ Tanques em série: $X = 67,7\%$ Único CSTR: $X = 56,3\%$	$XTES = 0.6724$ $n = 4.96$

Fonte: Do autor (2024).

A.4 DTR4 - Modelo de tanques ideais interconectados

Para avaliação deste modelo com dois parâmetros ajustáveis, um reator tanque (Figura 10) é modelado como dois CSTR interconectados (Figura 11), com base no Exemplo 18.6 da 6ª edição do livro Elementos de Engenharia das Reações Químicas de H. Scott Fogler (2022a).

Neste caso, um teste em degrau com traçador foi realizado no reator de 1000 dm³, com $C_{T0} = 2.000 \text{ mg/dm}^3$ e $v_0 = 25 \text{ dm}^3/\text{min}$.

A concentração de traçador ao longo do tempo segue o comportamento da Equação 43, obtida a partir do balanço do traçador nos reatores 1 e 2 da Figura 11, com m_1 e m_2 definidos na Equação 44, sendo a conversão prevista pelo modelo para uma reação de primeira ordem definida na Equação 40.

Para os dados apresentados, pede-se que sejam determinados os parâmetros de permuta α e β , a conversão para uma reação com $k = 0,03 \text{ min}^{-1}$, além das conversões correspondentes em um CSTR e em um PFR. Os resultados apresentados por Fogler (2022a), bem como os obtidos na ferramenta *web*, para este exemplo estão apresentados na Figura A.4.

Figura A.4 - Resultados referência (a) de Fogler (2022a) e os obtidos (b) pela ferramenta *web* para o Exemplo 18.6 relativo à modelagem de reator tanque por CSTR interconectados.

(a)

$$(X_{\text{modelo}} = 0,51) < (X_{\text{CSTR}} = 0,55) < (X_{\text{PFR}} = 0,7)$$

(b)

$$\begin{aligned} t_m &= 36.56 \text{ min} \\ \sigma^2 &= 1.74e+03 \text{ min}^2 \\ S^3 &= 2.26e+00 \text{ min}^3 \\ X &= 0.5202 \\ X(\text{CSTR}) &= 0.5455 \\ X(\text{PFR}) &= 0.6988 \\ m_1 &= -0.0018 \\ m_2 &= -0.95 \\ \text{Intersecao} &= 0.0158 \end{aligned}$$

Fonte: Do autor (2024).