



DESENVOLVIMENTO DE UMA FERRAMENTA ORIENTADA A EQUAÇÕES DE CÓDIGO ABERTO PARA MODELAGEM E SIMULAÇÃO DE PROCESSOS

Hanniel Ferreira S. de Freitas^{1,2} – hannel.freitas@ifrn.edu.br

José Eduardo Olivo² – jeolivo@uem.br

Cid Marcos Gonçalves Andrade² – cmgandrade@uem.br

1 – Instituto Federal do Rio Grande do Norte – IFRN Currais Novos

2 – Universidade Estadual de Maringá - UEM

Resumo: Face as demandas atuais para os processos industriais - que tipicamente envolvem aspectos sociais, ambientais e tecnológicos, aliados aos típicos objetivos econômicos - o *design* racional dos processos tem figurado como um importante ponto de consideração, bem como a associação de sistemas de controle, realização de estudos de otimização e análises exploratória de diferentes cenários produtivos. Nesse intento, as ferramentas para modelagem e simulação de flowsheets de processos mostram-se cruciais para a concepção e manutenção adequada de uma planta industrial. O presente trabalho tem como objetiva apresentar o desenvolvimento de uma ferramenta de código aberto em linguagem Python para a modelagem e simulação de flowsheets de processos, usando a abordagem orientada a equações, referida como numericamente superior para os problemas de grande escala, tipicamente encontrados na modelagem de unidades industriais. Uma breve análise acerca das equações algébrico-diferenciais é apresentada, bem como as especificações norteadoras utilizadas no desenvolvimento da ferramenta, tais como a utilização de ferramentas estado-da-arte da linguagem Python, mecanismo para coerência dimensional na definição das equações e a possibilidade de o usuário escrevê-las utilizando uma sintaxe bastante natural. Os resultados obtidos por meio da ferramenta desenvolvida para alguns estudos de caso são apresentados e discutidos, endossando as capacidades desta na resolução de problemas de modelagem e simulação de processos, bem

como a relevância da ferramenta com seu código aberto e disponível para ser expandido pelo usuário como um recurso para o estudo dos tópicos de interesse da simulação de processos a partir de uma abordagem orientada a equações.

Palavras-chave: Open-source; Simulador orientado a equações; Flowsheet; DAE; Python

1. Introdução

Nos dias atuais, as demandas industriais não se limitam àquelas de natureza econômica (produtividade, custo de produção versus lucro, etc) e contemplam também aspectos ambientais, sociais e tecnológicos. Essas demandas muitas vezes são definidas na etapa de *design* do processo, entretanto ocorre destas surgirem em momento posterior a esta etapa inicial, sendo garantidas por meio de sistemas de controle e/ou modificações de especificação. Assim, existe uma grande urgência para a realização de testes de performance sob variadas condições, o que vai de encontro a frequente impossibilidade de se interromper o processo produtivo para a realização de testes. Face a essa demanda, a modelagem e simulação de processos figura como uma importante ferramenta, permitindo que sejam realizados estudos de caso e análises exploratórias acerca da alteração estrutural destes (LUYBEN, 1989).

No tocante à modelagem e simulação de flowsheets de processos e das operações

unitárias envolvidas, duas abordagens diferentes podem ser definidas neste intento: A abordagem sequencial modular (SM), em que as equações definidas para cada equipamento são resolvidas, seguindo-se para o próximo; abordagem orientada a equações ou simultânea (EO, do inglês *Equation Oriented*), nas quais as equações que definem cada uma das operações que compõem o processo são reunidas e resolvidas simultaneamente. Conforme será descrito na seção de fundamentação teórica, esta abordagem apresenta vantagens à luz de estudos de caso de otimização, operação de recírculos (mássicos, energéticos), estimação de parâmetros, entre outros (DOWNLING & BIEGLER, 2015; FU et al., 2015). Contudo, a maioria dos simuladores comerciais adotam a abordagem sequencial-modular, ou uma estratégia híbrida entre as duas mencionadas acima.

O presente trabalho intenta apresentar uma ferramenta de código aberto (*open-source*) para a modelagem e simulação de processos utilizando a abordagem orientada a equações, utilizando uma linguagem de programação de alto nível (Python) e concebida com vistas a introduzir os usuários nos conceitos estruturantes da modelagem e simulação orientada a equações, e servir de base para aplicações neste intento. Convém notar que existem diversas ferramentas de modelagem e simulação EO *open-source* (dentre as quais podemos destacar EMSO, ASCEND, DAETOOLS, entre outras), entretanto muitas vezes estes softwares não são concebidos com o objetivo de fornecer uma plataforma amigável ao usuário interessado em expandir tais ferramentas ou basear-se nelas para o desenvolvimento de sua própria. Outro ponto norteador no desenvolvimento da ferramenta apresentada neste trabalho é que esta é desenvolvida em Python, uma vez que essa linguagem computacional apresenta um notável estado da arte voltado à computação científica, permitindo que diversas ferramentas sejam integradas.

O presente trabalho foi doravante

dividido de acordo com a seguinte estrutura: Na seção 2, a fundamentação teórica do mesmo é apresentada; a seguir, na seção 3, os conceitos importantes no desenvolvimento da ferramenta são descritos, bem como os casos de teste utilizados para a verificação da performance da mesma; na seção 4, os resultados preliminares obtidos com a ferramenta a partir de casos de estudo serão discutidos; na seção 5 as conclusões e perspectivas futuras são apresentadas, e em seguida, as referências utilizadas ao longo do texto.

2. Fundamentação teórica

A operação de um processo industrial, a exemplo de uma planta química, pode ser representando por um conjunto de equações algébricas não lineares, quando o processo encontra-se em estado estacionário (SHACHAM et al. 1982). Expandindo essa definição para os processos em regime transiente, e adotando um paradigma de modelagem em parâmetros concentrados, somam-se a este as equações diferenciais, formando um sistema chamado de algébrico-diferencial (DAE, do inglês *differential-algebraic equation*). Em termos matemáticos, estes sistemas podem ser descritos conforme o sistema canônico a seguir (WETTER et al., 2016).

$$F(t, x(t), \dot{x}(t), u(t), y(t), \theta) = 0 \quad (1)$$

$$Y(t, x(t), u(t), y(t), \theta) = 0 \quad (2)$$

$$F_0(x(t_0), \dot{x}(t_0), u(t_0), y(t_0), \theta) = 0 \quad (3)$$

Nas Equações (1-3), os termos $F(\dots)$ descrevem as equações representando taxas de mudança ao longo do tempo, $Y(\dots)$ as restrições algébricas, $F_0(\dots)$ definem implicitamente as condições iniciais, $\theta \in R^p$ o vetor dos parâmetros utilizados na definição do modelo, o vetor de estados $x: \mathbb{R} \rightarrow \mathbb{R}^{n_x}$, uma função de controle $u: \mathbb{R} \rightarrow \mathbb{R}^{n_u}$, e

$y: \mathbb{R} \rightarrow \mathbb{R}^{n_y}$ o vetor de variáveis algébricas.

De acordo com Fu et al. (2015), a abordagem EO apresenta-se como a mais adequada para a simulação de flowsheets de grande escala, bem como aqueles que apresentem correntes de reciclo e especificações implícitas. Na referida metodologia, todas as equações e variáveis são resolvidas simultaneamente, (KAMATH et al., 2010 apud FU et al., 2015). De acordo com os autores, a referida abordagem mostra-se especialmente mais vantajosa caso estejam disponíveis diretamente as matrizes Jacobiana e Hessiana. Outra vantagem da já mencionada abordagem é apresentada no trabalho de Åkesson et al. (2010), no qual os autores relatam que a mesma converte as equações dos modelos presentes no *flowsheet* para uma forma adequada para a solução de problemas de otimização não-linear de grande escala, os quais também são referidos na literatura especializada como *problemas de alta dimensionalidade*.

Quanto à resolução de sistemas DAE, um procedimento pode ser visto de maneira simplificada no trabalho de Soares (2003). No princípio, ocorre a etapa de *inicialização*, na qual são determinados os valores de todas as variáveis e suas derivadas no ponto inicial, por meio da resolução do sistema algébrico linear ou não-linear (mais frequente). Essa etapa tem por objetivo a obtenção de valores consistentes para proceder a resolução das equações diferenciais; em seguida, ocorre a etapa de *integração*, na qual as equações diferenciais são integradas ao longo do curso do problema. Na hipótese de ocorrerem descontinuidades e os valores das derivadas das variáveis tornem-se inconsistentes, é necessária a nova determinação dos valores das variáveis e suas derivadas, etapa essa referida como *reinicialização*. Convém notar que no caso de problemas estacionários, faz-

se necessária apenas a etapa de resolução do sistema algébrico.

3. Desenvolvimento da ferramenta

3.1 A concepção da ferramenta

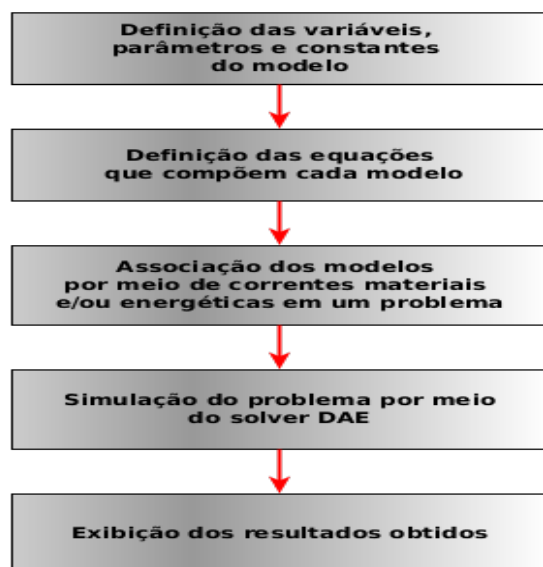
Conforme já foi descrito, no presente trabalho foi desenvolvida uma ferramenta na linguagem Python para a modelagem e simulação de processos utilizando a abordagem simultânea ou orientada a equações (EO). A ferramenta foi chamada de *SLOTH* (<https://github.com/hfsf/sloth>), e ainda se encontra sob constante desenvolvimento. A seguir, serão listadas as especificações utilizadas durante o desenvolvimento da mesma. Estas são, em síntese, as mesmas encontradas em outras ferramentas dedicadas à modelagem e simulação de processos por meio da abordagem simultânea, conforme descrito no trabalho antológico de Shacham et al. (1982).

- I. Desenvolvimento de acordo com as recomendações de consistência para estruturação eficiente do código (VAN ROSSUM et al., 2018).
- II. Mecanismo para consistência dimensional em todas as operações realizadas.
- III. Mecanismo para entrada de equações utilizando uma sintaxe muito próxima à da escrita matemática. A título de exemplo, a equação $F_{in} - F_{out} + KC$ pode ser escrita conforme a seguir: `F_in() - F_out() + K()*C()`
- IV. Mecanismo para análise dos graus de liberdade associados a cada modelo.
- V. Possibilidade de o usuário associar vários modelos (tipicamente, um para cada operação unitária), podendo utilizar-se de *templates* previamente estabelecidos, e conectá-los, formando um problema de estudo.
- VI. Utilização de ferramentas estado-da-arte da linguagem Python para as diversas etapas envolvidas nas simulações (computação simbólica, *solver* para os sistema de equações, operações matriciais, etc)

3.2. Etapas do processo de simulação na ferramenta

A seguir, será apresentado esquematicamente o procedimento para a realização de uma simulação utilizando a ferramenta *SLOTH*, conforme pode ser visto na Figura 1. Convém mencionar que uma descrição criteriosa da implementação do software está além do escopo deste trabalho. O código-fonte da mesma pode ser acessado por meio de seu repositório (<https://github.com/hfsf/sloth>), onde um usuário pode também contribuir com o seu processo de desenvolvimento, uma vez que a ferramenta é *open-source*.

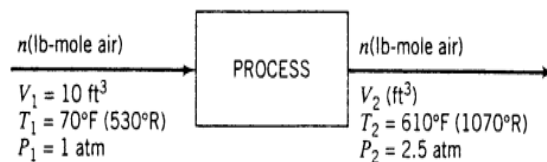
Figura 1-Representação esquemática do procedimento utilizado na modelagem simulação de flowsheets por meio da ferramenta desenvolvida



3.3. Casos de estudo

A título de ilustrar o funcionamento da ferramenta, ainda que em uma aplicação que encontra-se aquém do seu escopo, será analisado um caso de estudo correspondente a um processo genérico de aquecimento-compressão de uma corrente gasosa apresentado em Felder & Rousseau (1986), ilustrado abaixo, na Figura 2.

Figura 2-Caso de estudo de um processo de aquecimento e compressão de uma corrente gasosa. Adaptado de Felder & Rousseau (1986).



No referido processo, 10 pés cúbicos de ar (0,2832 m³) a 70 ° F (294, 26 K ou 530 °R) e a 1 atm são aquecidos a 610 ° F (594,26 K ou 1070 °R) e comprimidos a 2,5 atm. Deseja-se determinar o volume ocupado pelo gás ao final do processo de aquecimento-compressão, assumindo um comportamento de gás ideal.

4. Resultados obtidos

A partir do caso de estudo, utilizando-se da ferramenta *SLOTH*, foi criado um código para a resolução do mesmo. Um excerto do código pode ser visto abaixo, na Figura 3.

Figura 3-Excerto do código utilizado para a resolução do caso de estudo

```

import solvers
from core.equation_operators import *
from core.template_units import *

class modelTest1(model.Model):

    def __init__(self, name, description):

        super().__init__(name, description)

        self.P1 = self.createParameter("P1", atm, "Pressure-1")
        self.V1 = self.createParameter("V1", m**3, "Volume-1")
        self.T1 = self.createParameter("T1", K, "Temperature-1")
        self.P1.setValue(1)
        self.V1.setValue(10.)
        self.T1.setValue(530.)

        self.P2 = self.createParameter("P2", atm, "Pressure-2")
        self.V2 = self.createVariable("V2", m**3, "Volume-2")
        self.T2 = self.createParameter("T2", K, "Temperature-2")
        self.P2.setValue(2.5)
        self.T2.setValue(1070.)

    def DeclareEquations(self):

        expr = self.V2() - self.V1()*(self.P1()/self.P2())*(self.T2()/self.T1())

        self.eq = self.createEquation("eq1", "Volume-2 calculation", expr)

prob = problem.Problem("test_problem", "A problem for testing purposes")
  
```

Conforme já foi mencionado, a ferramenta também conta com mecanismos para a análise do conjunto das equações que compõem o processo, permitindo uma visualização amigável das variáveis, parâmetros e constantes definidas. Na Figura 4, um excerto da saída da ferramenta para o caso de estudo em tela é exibida, na qual também é possível ver o resultado obtido pela ferramenta quanto ao valor final ocupado pelo gás após o processo de aquecimento-compressão, correspondendo a um valor de 8,0754 ft³ (0,2287 m³).

Figura 4-Saída retornada pela ferramenta para o caso de estudo, apresentando um análise das equações que compõem o mesmo, bem como a saída obtida pela ferramenta para a resolução destas.



Equation name	Description	Representation	Variables	Parameters	Equation type
eq1	Volume-2 calculati	$V_2 = 8.07547169811321$	V_2	$T_2(^{\circ}), T_1(^{\circ}), V_1(^{\circ}), P_1(^{\circ}), P_2$	linear
eq2	Express	Z_1	Z_1	$T_1(^{\circ}), P_1(^{\circ})$	

is linear? True
is nonlinear? False
is differential? False

unit.py

>>(V2: 8.07547169811321)
>>: [8.07547169811321]

5. Conclusões e perspectivas futuras

No presente trabalho foi desenvolvido um protótipo inicial de uma ferramenta para a resolução de problemas de modelagem e simulação de flowsheets de processos, a partir de uma abordagem orientada a equações, desenvolvida na linguagem Python, e de código aberto. Um caso de estudo apresentado na literatura de um simples processo industrial foi utilizado para avaliar a performance da ferramenta, que se mostrou capaz de resolver o problema solicitado, ainda . Conforme foi mencionado anteriormente, a abordagem orientada a equações mostra-se como uma importante alternativa para a simulação de processos industriais, especialmente aqueles que possuem um caráter mais complexo. Assim, a ferramenta

aqui apresentada tem o seu valor endossado, uma vez que seu objetivo é figurar como uma alternativa para os estudos em torno desse tema, haja vista que ela possui código aberto e vem sendo desenvolvida com vistas a ser simples, acessível e extensível. Ainda que tenha sido utilizado um exemplo bastante simples, a ferramenta é capaz de lidar com problemas consideravelmente maiores. Como perspectivas futuras de trabalho, entende-se que o desenvolvimento da ferramenta deve continuar, contemplando a resolução de problemas algébrico diferenciais de grande escala, bem como a implementação de uma biblioteca das principais operações unitárias encontradas nas plantas químicas, de modo a permitir que o usuário utilize-se desta para as suas próprias simulações. Por fim, a inclusão de mecanismos dedicados aos estudos de otimização, análise de sensibilidade e controle também mostram-se como importantes rotas de trabalho.

REFERÊNCIAS

- LUYBEN, William L. **Process modeling, simulation and control for chemical engineers**. McGraw-Hill Higher Education, 1989.
- DOWLING, Alexander W.; BIEGLER, Lorenz T. **A framework for efficient large scale equation-oriented flowsheet optimization**. Computers & Chemical Engineering, v. 72, p. 3-20, 2015
- FU, Qiwen; ZHU, Lingyu; CHEN, Xi. **Complete equation-oriented approach for process analysis and optimization of a cryogenic air separation unit**. Industrial & Engineering Chemistry Research, v. 54, n. 48, p. 12096-12107, 2015.
- SHACHAM, M. et al. **Equation oriented approach to process flowsheeting**. Computers & Chemical Engineering, v. 6, n. 2, p. 79-95, 1982.
- KAMATH, Ravindra S.; GROSSMANN, Ignacio E.; BIEGLER, Lorenz T. **Aggregate models based on improved group methods for simulation and optimization of distillation systems**. Computers & chemical engineering, v. 34, n. 8, p. 1312-1319, 2010.
- WETTER, Michael; BONVINI, Marco; NOUIDUI, Thierry S. **Equation-based languages—A new paradigm for building energy modeling, simulation and optimization**. Energy and Buildings, v. 117, p. 290-300, 2016.

ÅKESSON, Johan et al. **Modeling and optimization with Optimica and JModelica. org—Languages and tools for solving large-scale dynamic optimization problems**. Computers & Chemical Engineering, v. 34, n. 11, p. 1737-1749, 2010.

VAN ROSSUM, Guido; WARSAW, Barry; COGHLAN, Nick. **PEP 8: style guide for Python code**. Python. org, 2013 . Disponível em: <<https://www.python.org/dev/peps/pep-0008/>>. Acesso em 5 nov. 2018.

SOARES, Rafael de Pelegrini. **Desenvolvimento de um simulador genérico de processos dinâmicos**. 2003. Dissertação (Dissertação em Engenharia Química) – UFRGS. Porto Alegre. 2003.

FELDER, Richard M.; ROUSSEAU, Ronald W. **Elementary principles of chemical processes**. NY etc.: Wiley, 1986.