

Testes Computacionais com o Método da Integração de Série de Taylor para a Simulação Dinâmica de Processos da Engenharia Química

Lucas Foppa^a

Orientador: Prof. Dr. Pedro Bolognese Fernandes

Introdução

Uma das opções para a redução do tempo computacional necessário para a simulação dinâmica de modelos de processo envolve o uso de métodos numéricos mais eficientes para a simulação do sistema. Essas técnicas aprimoradas representam vias alternativas para a resolução de problemas de valor inicial.

Um exemplo deste tipo de técnica é a **integração da série de Taylor**, que consiste em obter a solução por integração de sua expansão de Taylor, truncada após o número de termos desejado. Apesar de não ser novo, este método despertou atenção renovada na literatura^{[1][2]} principalmente em função do surgimento de algoritmos para diferenciação automática de sistemas de equações, que facilitam a obtenção da série. Scott e Martini^[3], por exemplo, obtiveram algoritmos aproximadamente 15 vezes mais rápidos para o cálculo de trajetórias de naves espaciais lançando mão de uma rotina de integração via série de Taylor com ajuste de passo.

Objetivo

Investigar o método de Taylor para resolução numérica de sistemas de equações diferenciais, analisando seu desempenho e praticidade de utilização para modelos de interesse da engenharia de processos.

Estratégia

Primeiramente, foram desenvolvidas, em ambiente de programação Python, rotinas de integração para o problema de valor inicial (1), de EDOs autônomas, utilizando o método da série de Taylor truncado no termo de segunda ordem, conforme a equação (2), para um passo h . A derivada de F em (2) corresponde à matriz Jacobiana e foi obtida através de duas formas diferentes, dando origem a duas rotinas:

- Utilizando o pacote ALGOPY de diferenciação automática
- Utilizando diferenças finitas de primeira ordem

$$(1) \quad \begin{cases} \frac{dy}{dt} = \mathbf{F}(y_1, y_2, \dots, y_n) = \begin{pmatrix} f_1(y_1, y_2, \dots, y_n) \\ f_2(y_1, y_2, \dots, y_n) \\ \dots \\ f_n(y_1, y_2, \dots, y_n) \end{pmatrix} \\ \mathbf{y}(t_0) = \mathbf{y}_0 \end{cases}$$

$$(2) \quad \mathbf{y}(t_n) \approx \mathbf{y}(t_{n-1}) + h * \mathbf{F}(\mathbf{y}(t_{n-1})) + h^2 * \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}(t_{n-1})) * \mathbf{F}(\mathbf{y}(t_{n-1}))$$

Em seguida, as performances das rotinas obtidas foram comparadas àquelas do solver Odeint (baseado nos métodos BDF e Adams) e do método clássico de Runge-Kutta, para a resolução de problemas “benchmarks” da engenharia química tais como o reator CST R (Continuous Stirred Tank Reactor) de Van De Vusse, cujos resultados são mostrados a seguir. O desempenho computacional foi avaliado através de dois fatores:

- erros globais relativos associados
- tempo necessário para o cálculo

Resultados

As rotinas de integração desenvolvidas apresentam erros absolutos menores (em relação à solução de Odeint) do que aqueles associados ao método Runge-Kutta. Além disso, os erros e a forma da solução independem da estratégia de obtenção da matriz Jacobiana (via ALGOPY ou diferenças finitas).

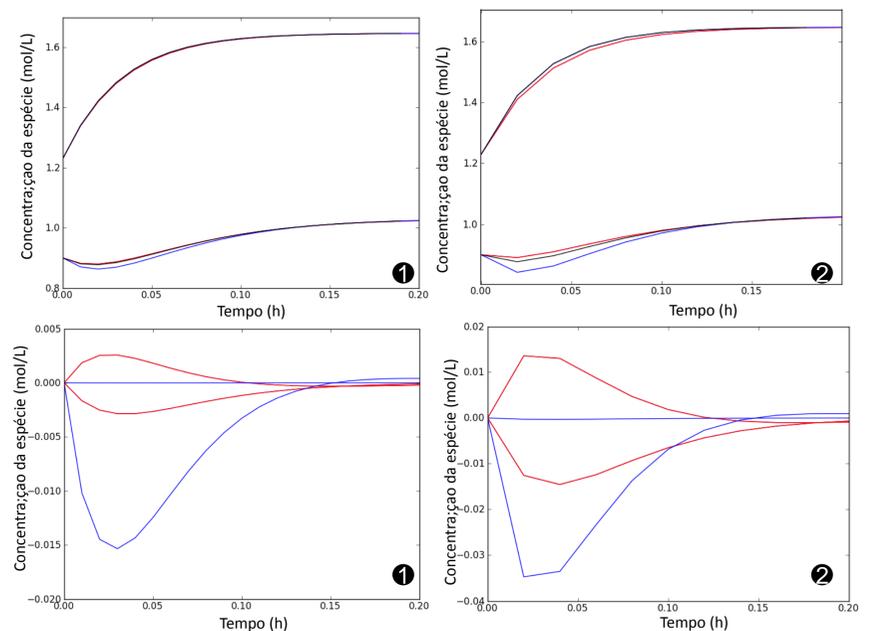


Figura 1. Soluções do PVI do reator de Van De Vusse obtidas com diversas rotinas de integração para passos de 0.01 e 0.02 (acima) e erros relativos à solução do solver Odeint (abaixo). preto: Odeint, azul: Runge-Kutta, vermelho: Taylor com diferenciação automática, lilás: Taylor com diferenças finitas.

Para pequenos valores de h (situação 1), a resolução do problema de valor inicial usando as rotinas desenvolvidas é muito mais lenta do que com o solver Odeint. No entanto, quando o passo é aumentado (situação 2), temos tempos semelhantes – e até menores – de cálculo, com erros da ordem de 1%.

A rotina que utiliza o método da série de Taylor com diferenças finitas para cálculo da matriz jacobiana apresentou os melhores resultados. É mais rápida do que a sua análoga que utiliza diferenciação automática (ALGOPY) e do que o método Runge-Kutta para os exemplos estudados.

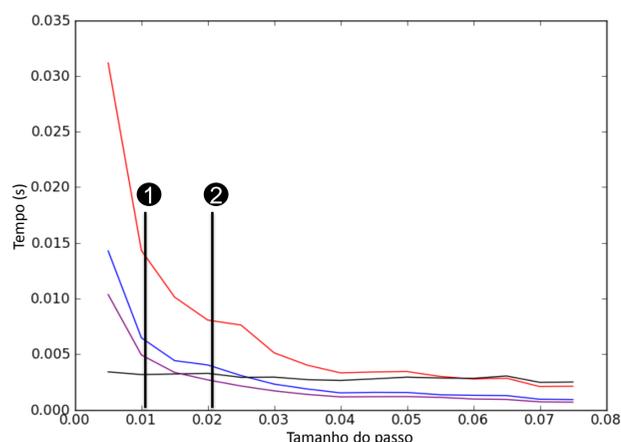


Figura 2. Tempo absoluto de cálculo da solução (s) versus tamanho de passo para as diversas rotinas. preto: Odeint, azul: Runge-Kutta, vermelho: Taylor com diferenciação automática, lilás: Taylor com diferenças finitas.

Conclusão

Os testes computacionais apresentados são aplicações da técnica da série de Taylor em problemas específicos. Contudo, seus resultados sugerem tendências que auxiliarão no aprimoramento deste método, como por exemplo a inclusão de controle do passo, aumento do número de termos na série de Taylor e testes com outros módulos de diferenciação automática.

Referências

- [1] Barrio, R, Applied Mathematics and Computation. 2005, 163, p. 525-545.
- [2] E. Miletics, G. Molnárka. Journal of Computational Methods in Science and Engineering. 2004, 4, 1-2, p. 105-114.
- [3] Scott, James R.; Martini, Michael C. 2011, NASA Tech Briefs, January 2011; 37.