Estudos de Técnicas Avançadas de Integração Numérica- Integração da Série de Taylor e Decomposição Adomiana

Vitor Dal Bó Abella

Orientador: Prof. Dr. Pedro Fernandes Bolognese

Introdução

Em função do uso cada vez maior de modelos computacionais nas Ciências Exatas e consequentemente na Engenharia, é necessário o desenvolvimento de métodos numéricos para a simulação de modelos de processo que permitam aplicações tais como estimação de parâmetros, otimização e controle, os quais apresentam grande demanda computacional. Deste modo, o objetivo desta pesquisa consiste no estudo de métodos numéricos mais eficientes para a resolução de problemas de valor inicial (PVI) constituídos por sistemas de equações diferenciais ordinárias (ODEs). Em particular, foi testada a aplicação de dois métodos para modelos típicos da Engenharia Química: método da integração da série de Taylor (TSM) e método da decomposição Adomiana (ADM).

Estratégia

Foram desenvolvidas em ambiente de programação Python, método de Taylor, e Matlab, método de Adomian, rotinas de integração para o problema de valor inicial de EDOs autônomas, eq. (1). Em seguida houve a realização de testes dos métodos.

$$(1)\quad \begin{cases} \dfrac{dy_i}{dt} = f_i\left(y_1(t), y_2(t), \dots, y_p(t)\right) = f_i(y\,(t)) \\ \boldsymbol{F} = [f_1(y(t)); f_2(y(t)); \dots; f_p(y(t))] \\ y(t_0) = y_0 \end{cases}$$

Método de Taylor

O método da série de Taylor foi truncado no termo de segunda ordem *(m=2)*, conforme a eq. (2), para um passo *h*. Foi inserido um controle de passo, eq. (3), para uma dada tolerância *tol*. A derivada de F em eq. (2) corresponde à matriz Jacobiana e foi obtida através de três formas diferentes, dando origem a três rotinas utilizando:

- Pacote ALGOPY de diferenciação automática;
- Diferenças finitas de segunda ordem;
- Jacobiana Exata.

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

$$(3)\quad h = \left(\frac{tol}{||y^{[m]}||}\right)^{\frac{1}{m}}$$

Método de Adomian

O método de Adomian propõe uma solução para eq. (1) como mostrado em eq. (4) onde o termo h_i e g_i representam, respectivamente, a soma dos termos lineares e não lineares de f_i. $Então, g_i$ é decomposto em uma série chamada de polinômios de Adomian, eq. (5), cuja expressão clássica está em eq. (6), onde n é sua ordem.

$$(4)\quad y_i(t) = y_i(0) + \int \left(h_i(y(t)) + g_i(y(t))\right) \mathrm{dt}$$

$$(5)\quad g_i(y(t)) = \sum_{n=0}^{\infty} A_{(i,n)}(y(t))$$

$$(6)\quad A_{(i,n)}\left(y_{(i,0)}(t) + y_{(i,1)}(t), \dots, y_{(i,n)}(t)\right) = \frac{1}{n!}\left(\frac{d^n}{d\lambda^n}\left[g_i\left(\sum_{j=0}^{\infty} \lambda^i y_{(i,j)}(\mathrm{t})\right)\right]\right)_{\lambda=0}$$

Testes

O método de Taylor foi estudado com um sistema de 41 EDOs de uma coluna de destilação com Jacobiana exata conhecida, verificando erro quadrático médio (EQM) para o último vetor de composições calculado com relação à solução do solver ODEint do Python, e tempo de processamento.

Já o método de Adomian foi testado em várias ordens para um sistema de EDOs de uma coluna de destilação de 5 estágios e um reator CSTR. A solução foi comparada com o ode45, solver em Matlab.

Departamento de Engenharia Química (DEQUI) - UFRGS, Rua Eng. Luiz Englert, s/nº - Prédio 12204, 90040-040 Porto Alegre, Brasil, vitor.abella@gmail.com

Resultados

Pode-se acompanhar a evolução do tempo de processamento e EQM para diferentes tolerâncias na fig. (1). O Algopy não respondeu bem para a resolução de um sistema de 41 EDOs da coluna de destilação, desta forma não aparece na fig. (1).

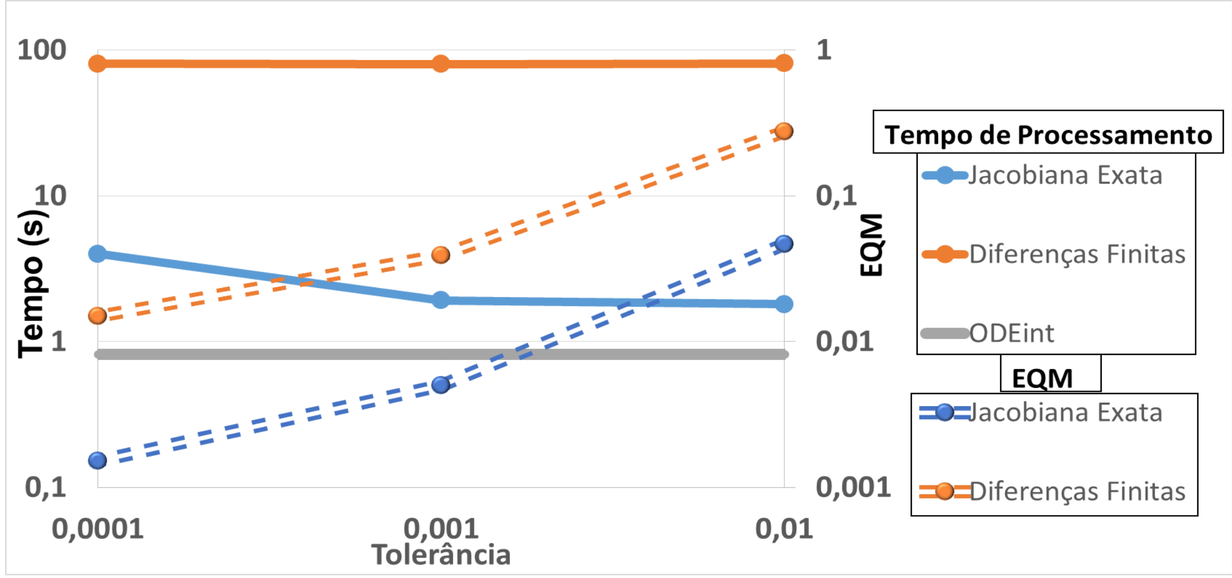

Figura 1. Relação entre rotinas, EQM , tempo de processamento e tolerância para o método de Taylor.

Diferenças Finitas de 2° ordem obtiveram tempo de processamento relativamente grande. Outro fato a destacar é a solução a partir da Jacobiana exata obteve maior tempo de processamento que o ODEint.

Para o método de Adomian, a coluna de destilação se mostrou demasiado complexa, o qual o Matlab não suportou realizar sucessivas integrais simbólicas. Para o reator CSTR obteve-se a fig. (2).

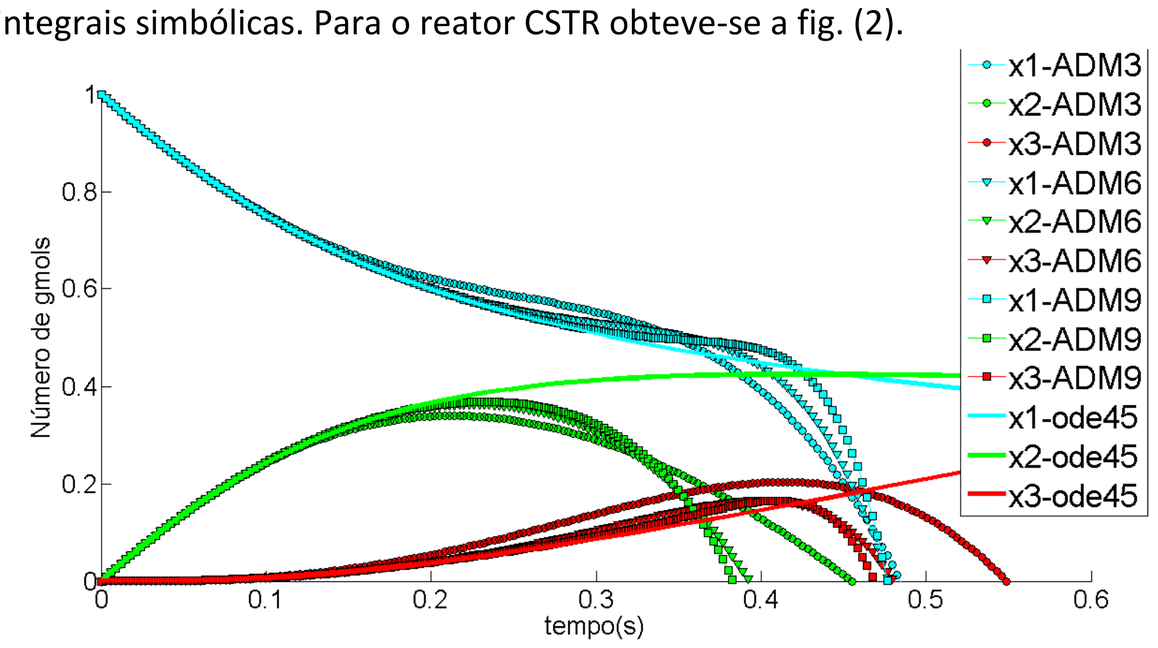

Figura 2. Solução para Sist. de ODEs através de diferentes ordens do polinômio de Adomian.

Onde o x(nº) representa a variável do sistema e o número que sucede ADM representa a ordem do polinômio de Adomian.

O tempo médio de execução da rotina em Matlab para obtenção da solução nas ordens 3, 6 e 9 foi respectivamente 0.01, 0.22 e 5.09s.

Como se percebe a solução obtida para um polinômio de Adomian de ordem 6 e 9, apesar de estarem próximas, apresentam significativa diferença em tempo de execução. O polinômio de ordem 3, apesar de mais se distanciar da solução proposta pelo solver ode45, foi o mais rápido.

Conclusão

Em uma busca constante de soluções mais rápidas e mais eficientes, a técnica de derivação automática acoplada ao TSM pode ser testada futuramente com outro pacote ou em outra linguagem computacional para averiguar seu real desempenho em sistemas de ODEs complexos.

Além disso, uma rotina de estimação de erro para o método de Adomian pode torná-lo em uma ferramenta poderosa para resolução de ODEs, pois seu tempo de execução foi bastante atrativo.

Referências

[1] Barrio Roberto, Applied Mathematics and Computation, 2005, Vol.163(2), p. 525-545.
[2] Jarod M. Younker, Numerical Integration of the Chemical Rate Equations via a Discretized Adomian Decomposition, ACS Publications, 2011, Vol.50(6), p. 3100-3109
[3] Hooman Fatoorehchi, On Calculation of Adomian Polynomials by MATLAB, Journal of Applied Computer Science & Mathematics, 2011, Vol.5(11), p.85.