XXVIII Congresso {virtual} de Iniciação Científica da Unicamp 2020

Equações diferenciais ordinárias *stiff*: um estudo computacional*

Gabriel Grillo† Profa. Dra. Sandra Augusta Santos‡

Resumo

As equações diferenciais ordinárias (EDO's) tiveram um grande sucesso em descrever o desenvolvimento de fenômenos dos mais variados assuntos no decorrer da história da ciência e tecnologia. Nesse sentido, o estudo de métodos numéricos foi fundamental para a solução dessas equações quando conhecemos as condições inciais do sistema, o que chamamos de problemas de valor inicial (PVI's). Entretanto, alguns desses PVI's, aos quais se deu o nome de *stiff*, mostraram-se excessivamente caros para serem integrados, mesmo com bons métodos. Alguns aspectos de *stiffness* são tratados neste trabalho, além de dois métodos desenvolvidos para tentar contornar essa dificuldade. Por fim, um problema aparentemente simples, porém desafiador, ilustra um PVI *stiff*, em que usamos uma rotina moderna (`ode15s` do `Matlab`[1]) para efetuar a integração.

Palavras-chave— Equações diferenciais ordinárias; problemas *stiff*; experimentos computacionais.

1 Introdução

Este trabalho aborda métodos para solução numérica de Problemas de Valor Inicial (PVI's) da forma

$$\frac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(t, \boldsymbol{y}), \quad t \in [t_0, t_f], \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \tag{1}$$

em que $\boldsymbol{f} : \mathbb{R}^{m+1} \to \mathbb{R}^m$ e $\boldsymbol{y}_0 \in \mathbb{R}^m$.

Conforme [1, Teorema 1.1], se em (1) assumirmos para $\boldsymbol{f}(t, \boldsymbol{y})$ continuidade na região $\mathcal{D} = \{t \in [t_0, t_f], \|\boldsymbol{y}\| < \infty\}$ e continuidade Lipschitz em $\boldsymbol{y}$ na mesma região, então para qualquer $\boldsymbol{y}_0 \in \mathbb{R}^m$ o PVI (1) possui solução única, $\boldsymbol{y}(t)$, no intervalo $[t_0, t_f]$ e essa solução é diferenciável. Assim, trataremos de problemas que seguem essas hipóteses.

Inspirados por [1] e [2] podemos descrever de forma geral o passo de um método numérico para se obter aproximação $\boldsymbol{y}_n$ da solução exata $\boldsymbol{y}(t_n)$ de (1) , como

$$\sum_{j=0}^{k} \alpha_j \boldsymbol{y}_{n-j} = h_n \boldsymbol{\psi}_{\boldsymbol{f}}(h_n, t_n, \ldots, t_{n-k}, \boldsymbol{y}_n, \ldots, \boldsymbol{y}_{n-k}), \tag{2}$$

em que $k \geq 1$ determina a quantidade de passos anteriores que se utiliza para calcular $\boldsymbol{y}_n$, os valores α_j, $j = 1, \ldots, k$, são constantes características do método e a função $\boldsymbol{\psi}_{\boldsymbol{f}}$ é definida a partir de avaliações de $\boldsymbol{f}(t, \boldsymbol{y})$, sendo também característica do método.

Se $k = 1$, dizemos que se trata de um método de passo simples e usa apenas $\boldsymbol{y}_{n-1}$ para calcular $\boldsymbol{y}_n$. Por outro lado, se $k > 1$, dizemos que é um método de passo múltiplo e usa $\boldsymbol{y}_{n-1}, \ldots, \boldsymbol{y}_{n-k}$ para calcular $\boldsymbol{y}_n$. Caso $\boldsymbol{\psi}_{\boldsymbol{f}}$ seja linear nas avaliações de $\boldsymbol{f}$, dizemos que o método é linear. Ainda, se $\boldsymbol{\psi}_{\boldsymbol{f}}$ depender de $\boldsymbol{y}_n$, dizemos que o método é implícito, e caso contrário o chamamos de explícito.

Por exemplo, o método de Euler adiantado, $\boldsymbol{y}_n = \boldsymbol{y}_{n-1} + h_n \boldsymbol{f}(t_{n-1}, \boldsymbol{y}_{n-1})$, é um método de passo simples, linear e explícito.

*Projeto financiado pela FAPESP (Número do Processo: 2019/24339-3).

†g216439@dac.unicamp.br

‡sasantos@unicamp.br

[1]Versão R2020a

Um método do tipo (2) é dito *convergente* de ordem p se

$$\boldsymbol{y}_n - \boldsymbol{y}(t_n) = \mathcal{O}(h^p), \quad \forall n = 1, \ldots, N,$$

em que $h = \max_{1 \leq n \leq N}\{h_n\}$ e $t_N = t_f$.

Definimos o *erro de truncamento local* como

$$\boldsymbol{d}_n = \frac{1}{h_n}\left(\sum_{j=0}^{k} \alpha_j \boldsymbol{y}(t_{n-j})\right) - \boldsymbol{\psi}_{\boldsymbol{f}}(h_n, t_n, \ldots, t_{n-k}, \boldsymbol{y}(t_n), \ldots, \boldsymbol{y}(t_{n-k})).$$

Esse valor nos dá o resíduo da equação que define o método quando trocamos $\boldsymbol{y}_{n-j}$ por $\boldsymbol{y}(t_{n-j})$ (compare a expressão para $\boldsymbol{d}_n$ e (2)).

Usando o erro de truncamento local, podemos definir um método como *consistente* de ordem p se $\boldsymbol{d}_n = \mathcal{O}(h_n^p)$.

Por fim, definimos um método como *0-estável* se todas as raízes ξ_i do polinômio $\rho(\xi) = \sum_{j=0}^{k} \alpha_j \xi^{k-j}$, dito polinômio característico de (2), forem tais que $|\xi_i| \leq 1$, e caso $|\xi_i| = 1$, então ξ_i é raiz simples, $i = 1, \ldots, k$.

Esta última definição fala sobre um aspecto bastante amplo dos métodos numéricos para PVI's, e existem diversos tipos de estabilidades que são definidos em alguns contextos. Entretanto, todos se interessam em entender como perturbações controladas, por exemplos do valor inicial $\boldsymbol{y}_0$ ou de parâmetros da função $\boldsymbol{f}(t, \boldsymbol{y})$, afetam o sistema como um todo, i.e. se as soluções obtidas serão suficientemente próximas. Um resultado importante ([1, Teorema 3.1]) diz que se um método for *0-estável* e *consistente* de ordem p, então o método é *convergente* de ordem p.

Feitas essas definições, podemos seguir para o tópico que esse trabalho se concentra, que é entender uma classe de PVI's em particular denominados *stiff*, ou aqueles em que ocorre o fenômeno de *stiffness*. A definição de um sistema de equações diferenciais *stiff* não é precisa e consolidada na literatura. Por exemplo, em [1] e [2] os autores apresentam definições semelhantes apresentando um PVI como sendo *stiff* se, ao aplicarmos um método do tipo (2), for necessário reduzir excessivamente o tamanho de passo h_n para que se mantenha a estabilidade em comparação ao que seria necessário para se manter a acurácia desejada, i.e. o erro local de aproximação para a solução sob controle.

Por outro lado, em [4] o autor define um PVI *stiff* quando a solução exata do problema, que vimos que é única $\boldsymbol{y}(t)$, tem uma variação pequena, mas as outras soluções da EDO associada próximas a $\boldsymbol{y}(t)$ apresentam grande variação. Como analogia, o autor ilustra esse processo como a solução exata estando em um vale que possui ao seu redor montanhas íngremes onde as outras soluções da EDO estariam. Em [3] os autores fazem essa interpretação graficamente exibindo as inclinações de algumas dessas soluções ao redor da solução exata. Na Figura 1 fazemos exatamente o mesmo para um exemplo.

Ainda, em [2] os autores destacam também que o fenômeno de *stiffness* em um PVI pode estar associado a sistemas em que uma das componentes varia muito mais rápido que outra, o que é comum, por exemplo, em sistemas que modelam reações químicas (ver [1, Exercício 5.9]).

Em geral, por conta de limitações em característica de estabilidade, os métodos explícitos para solução de PVI não são apropriados para problemas *stiff*, pois levariam a passos tão pequenos que a integração seria impraticável, ou por uma limitação de tempo, ou por uma limitação de erros de arredondamento acumulados. Assim, como os métodos implícitos apresentam caraterísticas de estabilidades mais robustas, eles são escolhidos para integrar problemas *stiff*. Nestes métodos, é preciso resolver um sistema não linear de equações em cada passo, e em geral se usa uma variação do método de Newton. Para a resolução deste sistema é importante ter informação sobre a matriz jacobiana $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}$, pois caso contrário esta deverá ser estimada numericamente, o que pode incorrer em maior erro e custo para o método.

Veja então que os métodos implícitos, ao realizarem passos mais caros numericamente, podem atingir maior estabilidade, e portanto realizar passos mais longos, de forma que darão menos passos na integração. Conforme comentado em [4], a preocupação para se desenhar métodos específicos para problemas *stiff* é por conta da eficiência, pois a princípio qualquer método será capaz de integrar um problema deste tipo, basta tomar passos suficientemente curtos, possuir capacidade computacional para armazenar a solução e evitando que o erro de arredondamento domine e ter tempo disponível para aguardar a integração completa.

Um dos métodos mais importantes para a integração de problemas *stiff*, introduzido em [3], é o *Backward differentiation formula - BDF*. Um método *BDF* de k passos se baseia em, para calcular $\boldsymbol{y}_n$, determinar um polinômio interpolador $\boldsymbol{p}(t)$ nos pontos $(t_{n-k}, \boldsymbol{y}_{n-k}), \ldots, (t_{n-1}, \boldsymbol{y}_{n-1})$ e fazer $\boldsymbol{p}'(t_n) = \boldsymbol{f}(t_n, \boldsymbol{y}_n)$. Caso os valores $t_{n-k}, \ldots, t_n$ estejam igualmente espaçados e esse espaçamento seja h, um passo do método *BDF* de k passos é dado por

$$\sum_{i=1}^{k} \frac{1}{i} \nabla^i \boldsymbol{y}_n = h\boldsymbol{f}(t_n, \boldsymbol{y}_n), \tag{3}$$

em que o símbolo $\nabla^i \boldsymbol{y}_n$ representa a diferença atrasada dada recursivamente por $\nabla^0 \boldsymbol{y}_l = \boldsymbol{y}_l$ e $\nabla^i \boldsymbol{y}_l = \nabla^{i-1}\boldsymbol{y}_l - \nabla^{i-1}\boldsymbol{y}_{l-1}$.

Veja que os métodos *BDF*'s são de passo múltiplo, linear e implícitos. Além disso, a única avaliação de $\boldsymbol{f}(t, \boldsymbol{y})$ está dada implicitamente em $(t_n, \boldsymbol{y}_n)$, o que dá ao método características de estabilidade importantes para tratar de problemas *stiff*. Entretanto, é possível mostrar que os métodos *BDF* de $k > 6$ passos não são 0-estáveis, de forma que esse métodos não têm utilidade prática. Um método *BDF* de k passos tem ordem de consistência $p = k$, e os métodos de $1 \leq k \leq 6$ passos são 0-estáveis, assim esses métodos têm ordem de convergência k.

Uma variação importante dos métodos *BDF* são os métodos *Numerical Differentiation Formulas - NDF*, expressos por

$$\sum_{i=1}^{k} \frac{1}{i} \nabla^i \boldsymbol{y}_n = h\boldsymbol{f}(t_n, \boldsymbol{y}_n) + \kappa\gamma_k(\boldsymbol{y}_n - \boldsymbol{y}_n^{(0)}), \tag{4}$$

em que $\gamma_k = \sum_{j=1}^{k} \frac{1}{j}$, κ é um parâmetro e $\boldsymbol{y}_n^{(0)}$ é o valor utilizado para iniciar o método de Newton modificado usado para resolver o sistema de equações não lineares que definem o passo. Este valor é escolhido como $\boldsymbol{y}_n^{(0)} = \sum_{j=0}^{k} \nabla^j \boldsymbol{y}_{n-1}$. Esses métodos são de ordem $p = k$.

Conforme apresentado em [5], os *NDF*'s são implementados no `Matlab` na rotina `ode15s` com escolhas para o parâmetro κ para cada ordem do método com o intuito de aumentar a acurácia em relação aos métodos *BDF* com uma pequena perda em estabilidade. Entretanto, é possível acionar uma opção em que a rotina `ode15s` usa os métodos *BDF*'s. Em qualquer um dos casos, trata-se de uma rotina de passo e ordem variáveis. Os números `1` e `5` no nome indicam que são usadas ordens de 1 a 5 e a letra `s` indica que a rotina é prescrita para problemas *stiff*. A sexta ordem tanto do *BDF* quanto do *NDF* não são implementadas por terem características de estabilidade limitadas.

2 Um problema teste

Em [3], artigo que introduz os métodos *BDF*, os autores tratam de PVI's cuja EDO associada é da forma

$$\frac{d\boldsymbol{y}}{dt}(t) = \frac{\boldsymbol{y}(t) - \boldsymbol{g}(t)}{a(t, \boldsymbol{y})}, \tag{5}$$

em que $a : \mathbb{R}^{m+1} \to \mathbb{R}$ e $\boldsymbol{g} : \mathbb{R}^m \to \mathbb{R}^m$, sendo $\boldsymbol{g}(t)$ uma função limitada. Nesse artigo os autores definem um problema *stiff* quando

$$\left| \frac{a(t, \boldsymbol{y})}{h} \right| \ll 1, \tag{6}$$

em que h é o passo na malha em t.

Trataremos de um problema teste que está nos moldes de (5), com $m = 1$, $a(t, \boldsymbol{y}) = 1/\alpha$, $\alpha > 0$ constante, e $g(t, y) = t^2$. Assim

$$y' = f(t, y) = \alpha(y - t^2), \tag{7}$$

cuja solução analítica é

$$y(t) = t^2 + \frac{2t}{\alpha} + \frac{2}{\alpha^2} + C \exp \alpha t, \qquad C \in \mathbb{R}. \tag{8}$$

Nosso PVI será tratado no intervalo $0 \leq t \leq 5$, com valor inicial $y(0) = 2/\alpha^2$, de forma que a sua solução é a expressão (8) com $C = 0$. Com isso, temos como solução um polinômio, mas o espaço de soluções da EDO (7) apresenta funções com parcela exponencial, o que nos dá a característica *stiff* desse problema. Podemos acompanhar na Figura 1 o campo de velocidades associado à EDO (7), em que podemos ver que ao redor da solução polinomial da EDO existe a tendência de grande crescimento ou grande decrescimento, que não é característica de polinômios, mas sim de exponenciais. Além disso, como α aparece multiplicando t no argumento da exponencial em (7), vemos que quanto maior for α, mais *stiff* será o problema. Outra forma de ver isso é analisando (6) e lembrando que para nosso problema temos $a(t, y) = 1/\alpha$.

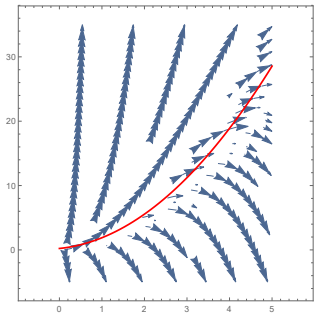


Figura 1: Campo de velocidades associado à EDO (7) em azul e a solução analítica (8) dessa EDO em vermelho, considerando $\alpha = 3$ e $C = 0$

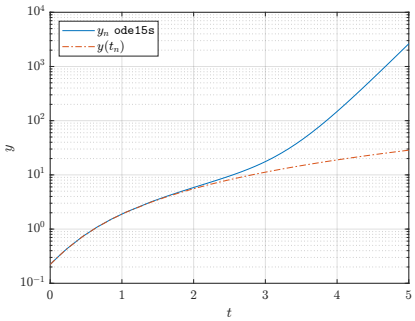


Figura 2: Solução de (9) com a rotina `ode15s` em *default*, juntamente com a solução exata dada por (8)

Para testar o método *BDF* em [3] os autores resolvem um PVI com EDO como em (7), usando $\alpha = 5$. Aqui definiu-se $\alpha = 3$ como na Figura 1, que nos experimentos preliminares se mostrou um valor capaz de exibir as dificuldades do problema de forma satisfatória, sem que a sua solução fosse demasiadamente custosa. O intervalo de integração foi escolhido com a mesma finalidade. Assim, o PVI a ser resolvido é

$$y' = f(t, y) = 3(y - t^2), \qquad 0 \leq t \leq 5, \qquad y(0) = \frac{2}{9}. \tag{9}$$

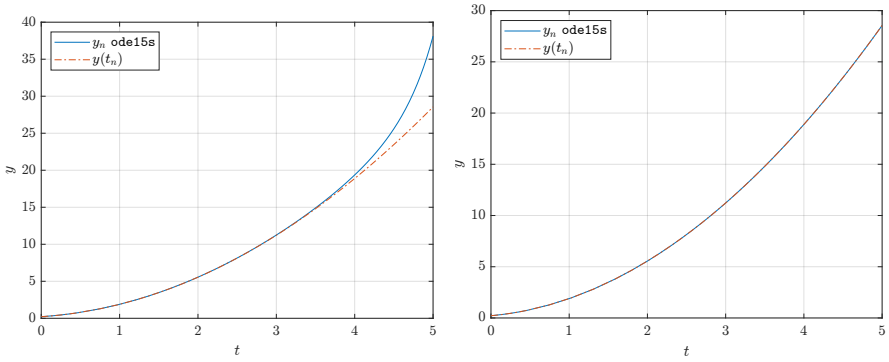
Com a finalidade de integrar (9) usaremos a rotina `ode15s` , já que segundo a documentação do `Matlab` esta é a primeira opção que o usuário deveria testar quando ele acredita que seu PVI tem características *stiff*. Além disso, como já dito, nela está implementado o método *NDF* e opcionalmente o método *BDF*, e este método foi apresentado em [3] resolvendo um problema similar a (9). Na Figura 2 podemos ver a solução obtida por `ode15s` em *default* em escala semilogarítmica em y. Observe que depois de $t = 3$ a solução passa a descrever uma reta na escala semilogarítmica em y, o que indica que a solução numérica obtida se comporta como uma exponencial. Com isso, podemos ver que o resultado obtido com a rotina `ode15s` 'escapa' para uma solução exponencial da EDO (7), seguindo uma linha de velocidade da Figura 1.

Sendo essa solução insatisfatória, podemos definir opções para que a rotina `ode15s` possa atingir melhores resultados. Uma tentativa válida seria ligar a opção para que o *solver* use o método *BDF* no lugar do *NDF*, já que aquele tem melhores características de estabilidade do que este. Entretanto, ao fazermos essa tentativa os resultados foram semelhantes. Assim, outra opção seria reduzir os valores de tolerâncias relativa e absoluta. Por padrão, a tolerância relativa é 1e−3 e a absoluta é 1e−6, então podemos fazer testes com ambas valendo `tol` $\in \{1e{-}6, 1e{-}8, 1e{-}10, 1e{-}12\}$. Para esses testes, dados como número de passos aceitos, passos descartados e avaliações de $\boldsymbol{f}(t, \boldsymbol{y})$ dizem respeito ao custo computacional da integração, e $e_{max} = \max_{1 \leq n \leq N} |y_n - y(t_n)|$, em que $t_N = 5$, diz respeito à qualidade da integração. Esses resultados podem ser acompanhados na Tabela 1.

`tol`	passos aceitos	passos descartados	avaliações de $f(t, y)$	e_{max}
1e−6	68	10	159	9.56e+0
1e−8	67	8	153	8.66e−2
1e−10	77	10	177	9.68e−4
1e−12	75	7	162	8.65e−6

Tabela 1: Dados da resolução numérica do PVI (9) usando `ode15s` com *NDF* e diferentes valores para as tolerâncias

Observando os dados da Tabela 1 vemos que valores menores de `tol` nos dão menores valores de e_{max}, como esperado. Entretanto, a maior exigência de tolerância não se reflete em um custo computacional substancialmente maior, de forma que os dados de passos aceitos, passos descartados e avaliações de $f(t, y)$ não variam muito. Assim, usando a rotina `ode15s` para esse problema é barato se obter integrações melhores tomando valores mais exigentes para `tol`, que farão com o que os mecanismos internos de controle de passo da rotina possam perceber com maior sensibilidade os momentos em que o passo pode aumentar e os momentos em que precisa diminuir. Graficamente, podemos acompanhar essa melhora observando a Figura 3 e também comparando-a com a Figura 2.



(a) `tol` = 1e−6 (b) `tol` = 1e−12

Figura 3: Solução de (9) com a rotina `ode15s` com *NDF* e tolerâncias definidas como `tol` e a solução exata dada por (8)

3 Conclusão

Descrevemos dois métodos para resolução de PVI's *stiff*, mas existem outros e também outras rotinas implementadas em `Matlab` (ver [5]). Além disso, apresentamos aqui uma instância de um problema *stiff*, mas existem outros problemas que são *stiff* por conta de outras propriedades. Em [2] são apresentados diversos problemas que são usados como teste, e em [5] esses problemas são integrados utilizando *stiff solvers* implementados em `Matlab`.

Destacamos que um problema aparentemente simples como (9) é capaz de causar complicações mesmo em rotinas modernas, o que nos surpreendeu, pois ao ler [3] imaginamos que poderíamos usar $\alpha = 50$ ou $\alpha = 100$ em (7), já que os autores resolvem esse problema com $\alpha = 5$ e nós teríamos à disposição uma rotina consolidada (`ode15s`) e uma capacidade computacional maior que a dos autores. Além disso, esse problema nos mostrou que ainda existe espaço para melhorias em *stiff solvers*, e que seu uso deve ser feito de forma crítica.

Referências

[1] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, 1998.

[2] R. Ashino, M. Nagase, and R. Vaillancourt, *Behind and Beyond the MATLAB ODE Suite*, Computers & Mathematics with Applications, 40 (2000), pp. 491–512.

[3] C. F. Curtiss and J. O. Hirschfelder, *Integration of Stiff Equations*, Proceedings of the National Academy of Sciences of the United States of America, 38 (1952), p. 235.

[4] C. B. Moler, *Numerical Computing with MATLAB*, SIAM, Philadelphia, 2004.

[5] L. F. Shampine and M. W. Reichelt, *The MATLAB ODE suite*, SIAM Journal on Scientific Computing, 18 (1997), pp. 1–22.