

XÂY DỰNG TRÌNH THỰC THI CHO PHƯƠNG PHÁP SAI PHÂN LÙI DẠNG KHỐI LIÊN TỤC

Dinh Văn Tiệp*, Phạm Thị Thu Hằng

Trường Đại học Kỹ thuật Công nghiệp - ĐH Thái Nguyên

TÓM TẮT

Phương pháp đa bước sai phân lùi dạng khối là một cải tiến nổi trội để tìm xấp xỉ nghiệm của phương trình vi với điều kiện ban đầu. Phương pháp này với miền ổn định tuyệt đối lớn nên đặc biệt thích hợp cho lớp các bài toán stiff. Bài báo này trình bày cách xây dựng trình thực thi cho phương pháp này bằng cách sử dụng phương pháp lặp Newton cho khối. Một chương trình bằng Ngôn ngữ Matlab được đưa ra để mô tả thuật toán và trình thực thi với số bước k cụ thể. Các ví dụ trình bày đưa ra sự so sánh về tính hiệu quả và chính xác của phương pháp.

Từ khóa: *phương pháp tuyến tính đa bước; phương pháp sai phân lùi; phương pháp sai phân lùi dạng khối liên tục; bài toán stiff; miền ổn định tuyệt đối.*

Ngày nhận bài: 19/5/2020; Ngày hoàn thiện: 27/5/2020; Ngày đăng: 29/5/2020

CONSTRUCTING THE IMPLEMENTATION TO THE CONTINUOUS BLOCK BDF METHODS

Dinh Van Tiep*, Pham Thi Thu Hang

TNU - University of Technology

ABSTRACT

The k-step methods of continuous block backward difference formula (or BDF) have great benefit to approximate the differential equation with the initial condition. These methods possessing very large absolute stability regions are especially useful for the stiff equations. This article presents a method of implementation to these k-step scheme by using the Newton's iteration for the root finding problem of the non-linear multivariable case. Also, a program written in Matlab with a particular k is presented. The numerical experiments introduced to illustrate the efficiency and exactness of these scheme comparing with some conventional BDF methods.

Keywords: *linear multistep method; backward difference formula; continuous block backward difference formula; stiffness; absolute stability region.*

Received: 19/5/2020; Revised: 27/5/2020; Published: 29/5/2020

* Corresponding author. Email: tiepdinhvan@gmail.com

1. Mở đầu

Xem xét bài toán giá trị ban đầu

$$y' = f(t, y), a \leq t \leq b, y(a) = \alpha, \quad (1)$$

với giả thiết điều kiện Lipschitz theo y của hàm f . Các phương pháp xấp xỉ nghiệm của phương trình bao gồm chủ đạo các phương pháp đơn bước và đa bước. Họ các phương pháp sai phân lùi (ký hiệu BDF) với số bước k nhỏ, $k < 6$, thể hiện ưu điểm lớn với miền ổn định tuyệt đối gần chiếm trọn mặt phẳng phức. Thực tế các miền đó của các phương pháp này chứa toàn bộ phần âm của trục thực trên mặt phẳng phức. Điều đó cho thấy khả năng áp dụng của các phương pháp này cho các bài toán stiff, một trong những vấn đề gây khó khăn lớn đối với đa số các phương pháp xấp xỉ khác.

Tuy nhiên, một trong những trở ngại cho tính áp dụng của các phương pháp họ sai phân lùi, cũng giống như các phương pháp đa bước bậc cao khác, chẳng hạn các phương họ Adams, đó là chúng đều ở dạng ẩn. Cách tiếp cận để xử lý vấn đề này gồm các hướng sử dụng dạng dự báo - hiệu chỉnh và hướng sử dụng các phép lặp cho bản thân mỗi phương pháp. Với hướng tiếp cận dự báo - hiệu chỉnh, miền ổn định tuyệt đối khá bé. Không may, điều này đặc biệt đúng với các cặp dự báo - hiệu chỉnh với kiểu hiệu chỉnh họ sai phân lùi. Khi đó, các ưu điểm nổi trội của họ sai phân lùi bị triệt tiêu. Đối với hướng sử dụng kiểu lặp có hạn chế ở việc nghiệm hội tụ đến nghiệm của phương trình sai phân, thay vì đến nghiệm của phương trình vi phân.

Hướng tiếp cận sai phân lùi dạng khối liên tục được đưa ra bởi N.M. Yao và cộng sự với bậc thấp $k \leq 6$ (xem [1], [2]) thể hiện sự nổi trội về tính áp dụng đối với lớp các bài toán stiff. Đặc biệt hơn, khả năng tự khởi động của phương pháp tạo nên sự thống nhất giữa khớp nối mà nhiều phương pháp đa bước gặp phải.

Trình thực thi của phương pháp khối liên tục có hai hướng tiếp cận chủ đạo, một là sau bước khởi động, không dùng quy trình lặp để

giảm khối lượng tính toán, hai là các bước lặp khối sẽ tiến hành cho đến cuối cùng. Với hướng tiếp cận thứ hai, độ chính xác tốt hơn, song yêu cầu số bước chia phải là một bội nguyên lần số bước của phương pháp.

2. Công thức sai phân lùi dạng khối liên tục

Trong [1], [2], N. M. Yao và cộng sự đã trình bày quy trình xây dựng phương pháp tổng quát, và đưa ra các công thức khối cụ thể cho các trường hợp $n = 2, 3, 4, 6$. Đồng thời, trong các bài báo này, các tác giả cũng đề cập đến vấn đề về tính ổn định, bậc và các hệ số bậc của từng công thức khối. Đối với các công thức $n = 2, 3$, các công thức khối dạng này là A-ổn định (tức A-stable). Với $n = 4, 6$, đặc tính đó có yếu đi chút ít song chúng vẫn là L_0 -ổn định (tức L_0 -stable) theo định nghĩa của Cash [3]. Các tính chất miền tuyệt đối này được bảo tồn từ các phương pháp BDF thông thường. Như vậy, các ưu điểm vượt trội của phương pháp BDF không bị mất đi với $k \leq 6$.

Ở đây, khác so với trình thực thi giới thiệu trong [1], ta sẽ xây dựng quy trình lặp tổng quát cho khối được tạo ra theo phương pháp này với chứng minh sự hội tụ của nghiệm xấp xỉ khi $h \rightarrow 0^+$.

Xét công thức dạng khối liên tục xây dựng kiểu sai phân lùi [1]:

$$A^{(1)}Y_{n+1} = A^{(0)}Y_n + hB^{(1)}F_{n+1}, \quad (2)$$

$$\text{với } Y_{n+1} = (y_{n+1}, y_{n+2}, \dots, y_{n+k})^T,$$

$$Y_n = (y_{n-k+1}, y_{n+k}, \dots, y_n)^T,$$

$$F_{n+1} = (f_{n+1}, f_{n+2}, \dots, f_{n+k})^T,$$

$$y_j \approx y(t_j), f_j \approx f(t_j, y(t_j)), \forall j = 1, 2, \dots,$$

$A^{(1)}, A^{(0)}, B^{(1)}$ là các ma trận cấp $k \times k$ thu được từ xây dựng phương pháp. Ở đây, giả sử phương pháp BDF cho bởi công thức:

$$hf_{n+i} = \sum_{j=0}^{k-1} a_{ij}y_{n+j} + hb_i f_{n+k},$$

$$\forall i = 1, \dots, k-1. \quad (3)$$

$$y_{n+k} = \sum_{j=0}^{k-1} a_j y_{n+j} + hb_k f_{n+k},$$

$A^{(1)}$ có cột k là $(0, \dots, 1)^T$, và hàng k là $-(a_1, \dots, a_{n-k+1}, -1)$, cột thứ j của ma trận là $-(a_{1j}, a_{2j}, \dots, a_j)^T$; ma trận $A^{(0)}$ có tất cả các cột bằng vectơ không, ngoại trừ cột cuối cùng là vectơ $(a_{10}, \dots, a_{k-1,0}, a_0)^T$; ma trận $B^{(1)}$ có dạng tam giác trên với các phần tử khác không nằm trên đường chéo chính và cột thứ k , trong đó đường chéo chính là vectơ $(-1, \dots, b_k)$, và cột k là vectơ $(b_1, \dots, b_k)^T$.

Các ma trận $A^{(1)}$ cho $k = 2, 3, 4, 5, 6$ đều là các ma trận khả nghịch. Do đó, phương trình (2) có thể đưa được về dạng:

$$Y_{n+1} = CY_n + hBF_{n+1}, \quad (4)$$

với C là ma trận cấp $k \times k$ có các phần tử khác 0 ngoại trừ có thể các phần tử cột k , ký hiệu là:

$$A = (c_1, c_2, \dots, c_k)^T. \quad (4')$$

Trong (3), $F_{n+1} = F(t_{n+1}, Y_{n+1})$

$$= (f(t_{n+1}, y_{n+1}), \dots, f(t_{n+k}, y_{n+k})),$$

phụ thuộc vào Y_{n+1} . Chính vì thế, phương trình (4) có thể xấp xỉ nghiệm bằng phương pháp lặp Newton hoặc Newton suy rộng với giả thiết hàm f đủ khả vi. Quá trình lặp Newton được áp dụng cho phương trình $G(X) = 0$ với mỗi $q \geq 0$ là:

$$G(X) = y_{qk}A + hBF(t_{qk+1}, X) - X, \quad (5)$$

$X = (x_1, \dots, x_k)^T \in \mathbb{R}^k$, để tìm nghiệm xấp xỉ của phương trình này. Thực tế, khi $h \rightarrow 0$, nghiệm của (5) hội tụ về nghiệm đúng:

$$Y(t_{qk+1}) = (y(t_{qk+1}), \dots, y(t_{qk+k}))^T,$$

khi giả thiết $y_{qk} = y(t_{qk})$. Điều này được chứng minh trong Định lý 1 dưới đây. Bậc và hệ số bậc của phương pháp khối là bậc nhỏ nhất và hệ số bậc nhỏ nhất để xấp xỉ các $y(t_{qk+i}), \forall i = 1, \dots, k$, bằng các phương trình sai phân thứ i tương ứng trong (3). Công thức lặp nghiệm cho (5) (tại bước thứ q) là:

$$X^{(m+1)} = X^{(m)} - [J_G(X^{(m)})]^{-1}G(X^{(m)}), \quad (6)$$

với

$$X^{(0)} = \begin{cases} (y_{(q-1)k+1}, \dots, y_{qk}) & \text{nếu } q \geq 1 \\ (0, \dots, y_0), y_0 = \alpha, & \text{nếu } q = 0. \end{cases}$$

Jacobian của G là $J_G(X) =$

$$hBdiag(f_y(t_{qk+1}, x_1), \dots, f_y(t_{qk+k}, x_k)) - I_k,$$

trong đó

$$diag(f_y(t_{qk+1}, x_1), \dots, f_y(t_{qk+k}, x_k))$$

là ma trận đường chéo với các phần tử này xếp dọc đường chéo chính và I_k là ma trận đơn vị cấp k .

Giả sử số đoạn chia của $[a, b]$ là $N = kp$, tức N là bội nguyên lần của k . Từ điều kiện ban đầu $y_0 = \alpha$, với $q = 0$, áp dụng (5) ta được $Y_1 = (y_1, y_2, \dots, y_k)^T$.

Tiếp tục với $q = 1$, áp dụng (5) thu được:

$$Y_{k+1} = (y_{k+1}, y_{k+2}, \dots, y_{2k})^T.$$

Cứ như thế, với $q = (p - 1)$, thu được:

$$Y_{k(p-1)+1} = (y_{k(p-1)+1}, y_{k(p-1)+2}, \dots, y_N)^T.$$

Như vậy, sau quá trình thu được dãy vectơ

$$Y_1, Y_{k+1}, \dots, Y_{k(p-1)+1}$$

xấp xỉ nghiệm của (1). Mệnh đề dưới đây chỉ ra một đặc trưng của trình thực thi ta đang xây dựng.

Mệnh đề 1. Ma trận A trong (4') có tất cả các thành phần là 1, tức $A = (1, 1, \dots, 1)^T$.

Chứng minh. Từ (3), ta được:

$$A^{(1)} = \begin{bmatrix} -a_{11} & -a_{12} & \dots & -a_{1,k-1} & 0 \\ -a_{21} & -a_{22} & \dots & -a_{2,k-1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -a_{k-1,1} & -a_{k-1,2} & \dots & -a_{k-1,k-1} & 0 \\ -a_1 & -a_2 & \dots & -a_{k-1} & 1 \end{bmatrix},$$

$$A^{(0)} = \begin{bmatrix} 0 & \dots & 0 & a_{10} \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & a_{k-1,0} \\ 0 & \dots & 0 & a_0 \end{bmatrix}, a_0 = 1,$$

$$B^{(1)} = \begin{bmatrix} -1 & 0 & \cdots & 0 & b_1 \\ 0 & -1 & \ddots & \vdots & b_2 \\ \vdots & 0 & \ddots & 0 & \vdots \\ 0 & \vdots & \ddots & -1 & b_{k-1} \\ 0 & 0 & \cdots & 0 & b_k \end{bmatrix}.$$

Viết ma trận $A^{(1)}, [A^{(1)}]^{-1}$ dưới dạng khối:

$$A^{(1)} = \begin{bmatrix} D & \mathbf{0} \\ U & 1 \end{bmatrix}, [A^{(1)}]^{-1} = \begin{bmatrix} D^{-1} & \mathbf{0} \\ V & 1 \end{bmatrix},$$

trong đó, D^{-1} là nghịch đảo của D ,

$$D = (-a_{ij})_{(k-1) \times (k-1)}, D^{-1} \in \mathbb{M}(k-1),$$

$$\mathbf{0} = (0, \dots, 0)^T \in \mathbb{M}(k-1, 1),$$

$$U = (-a_1, \dots, -a_{k-1}), V \in \mathbb{M}(1, k-1).$$

Giả sử $V = (x_1, \dots, x_{k-1})^T$ và D^{-1} có các cột lần lượt là D^1, D^2, \dots, D^{k-1} , hay viết là:

$$D^{-1} = [D^1, D^2, \dots, D^{k-1}].$$

Ký hiệu E_l là vectơ cột đơn vị thứ l của $\mathbb{M}(k-1, 1) = \mathbb{R}^{k-1}, \forall l = 1, \dots, k-1$,

$$E_l = (0, \dots, 0, 1, 0, \dots, 0)$$

với số 1 ở vị trí thứ l . Gọi I_{k-1} là ma trận đơn vị cấp $k-1$. Ta có:

$$D^{-1}D = I_{k-1} = [E_1, \dots, E_{k-1}],$$

$$[V \ 1] \begin{bmatrix} D \\ U \end{bmatrix} = (0, \dots, 0) \in \mathbb{M}(1, k-1) = \mathbb{R}^{k-1},$$

nên

$$\sum_{j=1}^{k-1} (-a_{jl})D^j = E_l, \forall l = 1, \dots, k-1, \quad (7)$$

$$\sum_{j=0}^{k-1} a_{jl}x_j + a_l = 0, \forall l = 1, \dots, k-1. \quad (8)$$

Vì k phương trình sai phân ở (3) đều có bậc $r > 0$ nên hệ số bậc 0 của các phương trình này là 0, tức ta có:

$$\sum_{l=0}^{k-1} a_{jl} = 0, \forall j = 1, \dots, k-1, \quad (7')$$

$$\sum_{j=0}^{k-1} a_j = 1. \quad (8')$$

Kết hợp (7) và (7') được:

$$\begin{aligned} \sum_{j=1}^{k-1} a_{j0}D^j &= \sum_{j=1}^{k-1} \left(-\sum_{l=1}^{k-1} a_{jl} \right) D^j \\ &= \sum_{l=1}^{k-1} \left(\sum_{j=1}^{k-1} (-a_{jl}) D^j \right) = \sum_{l=1}^{k-1} \left(\sum_{j=1}^{k-1} a_{jl} D^j \right) \\ &= \sum_{l=1}^{k-1} E_l = (1, \dots, 1)^T \in \mathbb{M}(k-1, 1) = \mathbb{R}^{k-1}. \end{aligned}$$

Đẳng thức này là $k-1$ thành phần đầu tiên của vectơ $A = (1, \dots, 1)^T \in \mathbb{R}^k$.

Kết hợp (8) và (8') được:

$$\begin{aligned} \sum_{j=1}^{k-1} a_{j0}x_j &= \sum_{j=1}^{k-1} \left(-\sum_{l=1}^{k-1} a_{jl} \right) x_j = \\ \sum_{l=1}^{k-1} \left(-\sum_{j=1}^{k-1} a_{jl} x_j \right) &= -\sum_{l=1}^{k-1} a_l = a_0 - 1 = 0. \end{aligned}$$

Đẳng thức này là thành phần thứ k của vectơ $A = (1, \dots, 1)^T \in \mathbb{R}^k$. \square

Định lý sau khẳng định sự hội tụ (khi $h \rightarrow 0^+$) của nghiệm xấp xỉ của phương trình (5) về nghiệm của phương trình vi phân (1). Điều này cho thấy, dù phương trình phi tuyến (5), tại mỗi giá trị $q \geq 0$, có thể có nhiều nghiệm, tuy nhiên với kích thước bước h đủ nhỏ, việc lựa chọn y_{qk} và vectơ

$$\begin{aligned} X^{(0)} &= Y_{(q-1)k+1} \\ &= (y_{(q-1)k+1}, y_{(q-1)k+2}, \dots, y_{qk}) \end{aligned}$$

như là xấp xỉ ban đầu của mỗi quá trình lặp Newton (6) là đúng đắn. Tức là nghiệm của phương trình phi tuyến (5) thu được từ sự hội tụ của dãy (6), với mỗi $q \geq 0$, có thể gần tùy ý đến nghiệm đúng của (1), chỉ cần điều chỉnh giá trị h đủ nhỏ.

Định lý 1. Có định mỗi $q \geq 0$. Giả sử nghiệm thu được ở bước thứ $q-1$ là nghiệm đúng, tức $Y_{qk+1} = Y(t_{qk+1})$ nếu $q \geq 1$ và $y_{qk} = y(t_{qk})$ nếu $q = 0$. Gọi $X = Y_*$ là nghiệm đúng của phương trình (5), $G(X) = 0$, thu được từ sự hội tụ của dãy (6), $\lim_{m \rightarrow \infty} X^{(m)} = Y_*$,

tức Y_* phụ thuộc Y_{qk} . Gọi $Y^* = Y(t_{qk+1}) = (y(t_{qk+1}), \dots, y(t_{qk+k}))^T$ là nghiệm đúng của phương trình vi phân (1). Khi đó,

$$\lim_{h \rightarrow 0^+} \|Y_* - Y^*\| = 0.$$

Chứng minh. Giả sử $\rho_k(h)h^s$ là véc tơ có các thành phần là sai số làm tròn (truncation error) của các phương trình sai phân lần lượt trong (3) với s là số lớn nhất sao cho h^s là nhân tử chung của k thành phần này. Vì các hệ số bậc 0 và 1 của những phương trình này bằng 0, nên $s \geq 2$. Ta có:

$$Y^* = y(t_{qk})A + hBF(t_{qk+1}, Y^*) + \rho_k(h)h^s,$$

$$Y_* = y_{qk}A + hBF(t_{qk+1}, Y_*).$$

Do đó, $\|Y^* - Y_*\| \leq h\|B\|\|F(t_{qk+1}, Y^*) - F(t_{qk+1}, Y_*)\| + \rho_k(h)h^s \leq hL_F\|B\|\|Y^* - Y_*\| + \rho_k(h)h^s$. (Ở đây, sử dụng điều kiện Lipschitz của f suy ra sự tồn tại của hằng số Lipschitz L_F theo Y cho hàm véc tơ $F(t, Y)$) Từ đó, với $h > 0$ đủ nhỏ, sao cho

$$\frac{1}{2} < (1 - hL_F\|B\|),$$

$$\|Y^* - Y_*\| \leq \frac{\rho_k(h)h^s}{1 - hL_F\|B\|} < 2\rho_k(h)h^s.$$

Do đó, ta có điều phải chứng minh.

Trên đây là hướng tiếp cận thứ hai của trình thực thi như đã đề cập ở đoạn cuối phần mở đầu bài báo. Hướng tiếp cận thứ nhất chỉ xét đến quá trình lặp Newton một lần để xấp xỉ Y_1 phần còn lại sẽ sử dụng đồng thời phương trình thứ k của (3) như là phép hiệu chỉnh, trong đó về phải có chứa f_{k+1} mà giá trị này sẽ được tính toán bằng cách sử dụng phương trình liền trước nhất thứ $k - 1$, và phương trình này được coi như phép dự báo. Như thế, ở hướng tiếp cận thứ 2 này, ta có một cặp dự báo - hiệu chỉnh. Tuy nhiên, cặp này có thể làm hẹp đi đáng kể độ lớn của miền ổn định tuyệt đối mà các phương pháp BDF bậc thấp ($k \leq 6$) vốn có.

Đối với $k = 3$, ta có $X = (x_1, x_2, x_3)^T$,

$$A^{(1)} = \begin{bmatrix} 4/11 & -8/11 & 0 \\ 14/11 & -23/22 & 0 \\ 9/11 & -18/11 & 1 \end{bmatrix},$$

$$A^{(0)} = \begin{bmatrix} 0 & 0 & -4/11 \\ 0 & 0 & 5/22 \\ 0 & 0 & 2/11 \end{bmatrix},$$

$$B^{(1)} = \begin{bmatrix} -1 & 0 & -1/11 \\ 0 & -1 & 2/11 \\ 0 & 0 & 6/11 \end{bmatrix}, A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 23 & -4 & 5 \\ 12 & -3 & 12 \\ 7 & 2 & 1 \\ 3 & -3 & 3 \\ 9 & 0 & 3 \\ 4 & 0 & 4 \end{bmatrix},$$

$$G(X) = y_{qk}A + hBF(t_{qk+1}, X) - X$$

$$= y_k \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + hB \begin{bmatrix} f(t_{n+1}, x_1) \\ f(t_{n+2}, x_2) \\ f(t_{n+3}, x_3) \end{bmatrix},$$

$J_G(X)$

$$= hB \begin{bmatrix} f_y(t_{n+1}, x_1) & 0 & 0 \\ 0 & f_y(t_{n+2}, x_2) & 0 \\ 0 & 0 & f_y(t_{n+3}, x_3) \end{bmatrix} - I_3.$$

Với $k = 5$, các ma trận khối trong (4) và (4') được cho như sau, các khối này do tác giả xây dựng,

$$A = (1,1,1,1,1)^T,$$

$$B = \begin{bmatrix} 1901 & -1387 & 109 & -637 & 251 \\ 720 & -360 & 30 & -360 & 720 \\ 269 & 133 & 49 & 73 & 29 \\ 90 & -45 & 15 & -45 & 90 \\ 237 & 99 & 39 & -69 & 27 \\ 80 & -40 & 10 & -40 & 80 \\ 134 & 116 & 68 & -56 & 14 \\ 45 & -45 & 15 & -45 & 45 \\ 425 & 175 & 25 & -25 & 95 \\ 144 & -72 & 6 & -72 & 144 \end{bmatrix}.$$

3. Trình thực thi và thuật toán

Trình thực thi của phương pháp khối liên tục dạng BDF bậc k được trình bày dạng giải mã sau đây. Trong trình thực thi này, hằng số $tol > 0$ được đưa vào làm tiêu chuẩn dừng cho phép lặp Newton, đảm bảo sai số phép lặp mỗi bước không vượt quá số này. Ngoài ra, hằng số M đưa ra giới hạn cho số lần lặp

được phép mà nếu vượt quá số này, quá trình lặp thất bại. Số đoạn chia N được yêu cầu là bội nguyên lần của k .

INPUT hàm $f(t, y)$, $[a, b]$, giá trị ban đầu $y(a) = \alpha$, số đoạn chia $N = kp$, độ tin cậy $tol > 0$, số lần lặp tối đa M mỗi quá trình lặp.

OUTPUT dãy xấp xỉ w_i của nghiệm $y(t_i), \forall i = 0, 1, \dots, N$,

hoặc thông báo quá trình thất bại.

Trình thực thi

Bước 1. Khởi tạo:

$t_0 := a;$

$w_0 := \alpha;$

$h := \frac{b-a}{N};$

$Y := (0, \dots, w_0)^T;$ % Viết $Y = (y_1, \dots, y_k)$.

ma trận $A, B;$

$i := 0;$

hàm $G(X) = w_0 A + hBF(t_{i+1}, X) - X;$

$FLAG := 0;$ % Điều kiện thoát vòng While

Bước 2. Quy trình lặp:

While ($i < N$) and ($FLAG = 0$) do

$k := 1;$

While $k \leq M$ do % Vòng While trong

tính $J_G(Y), G(Y);$

giải hệ phương trình $J_G(Y)X = -G(Y);$

cập nhật $Y := Y + X;$

If $\|X\| < tol$ then

cập nhật $w_0 := y_k;$

chấp nhận xấp xỉ $Y_{i+1} := Y;$

exit;%Thoát vòng While

trong

Else If ($k = M$) then

OUT="fail";

$FLAG := 1;$

% Thông báo số vòng lặp vượt quá

$M.$

% Thoát vòng lặp While ngoài.

End If.

$k := k + 1;$

End do. % Kết thúc vòng While trong.

$i := i + 3;$

End do. % Kết thúc vòng While ngoài.

Bước 3. Xuất kết quả $Y_1, Y_{k+1}, \dots, Y_{k(p-1)+1}$, hoặc thông báo quá trình thất bại do vượt quá số vòng lặp M quy định.

Chương trình sau đây lập trình cho phương pháp với $k = 3$.

Chương trình Matlab

```
function
outp=BDFblock(f,a,b,alpha,N,tol,M)
% NOTE: N must be a multiple
of k.
%----Exact solution to
compare----
syms p(r);
format long;
eqn=diff(p,r)==f(r,p);
icd=p(a)==alpha;
as=dsolve(eqn,icd);
xi=double(subs(as,r,a:(b-a)/N:b));
ti=a:(b-a)/N:b;
%-----STEP 1-----
t0=a;
y0=alpha;
h=(b-a)/N;
A=[1;1;1];
B=[23/12,-4/3,5/12;7/3,-2/3,1/3;9/4,0,3/4];
syms s0 s u z s1 s2 s3;
K=diag([subs(diff(f(s0,z),z),[s0,z],[s+h,s1]),subs(diff(f(s0,z),z),[s0,z],[s+2*h,s2]),subs(diff(f(s0,z),z),[s0,z],[s+3*h,s3])]);
Jg=h*B*K-eye(3);% Jacobian of
G
```

```

G=u*A+h*B*[f(s+h,s1);f(s+2*h,s
2);f(s+3*h,s3)]-
[s1;s2;s3];%function G
t=[t0];
w=[y0];
Y=[0;0;y0];
%-----STEP 2-----
---
i=0;
FLAG=0;
%Use to STOP outer while-loop
while (i<N) & (FLAG==0)% Outer
while
    k=1;
while k<=M % Inner while-loop
J=double(subs(Jg,[s,s1,s2,s3],
[t0,Y(1),Y(2),Y(3)]));
G1=double(subs(G,[u,s,s1,s2,s3],
[y0,t0,Y(1),Y(2),Y(3)]));
X=linsolve(J,-G1);
Y=Y+X;
if (norm(X)<tol)
    w=[w,Y(1),Y(2),Y(3)];
    y0=Y(3);
    =[t,t0+h,t0+2*h,t0+3*h];
    t0=t0+3*h;
    break
elseif k==M
    fprintf('Fail!');
    FLAG=1;
end
k=k+1;
end
i=i+3;
end
%-----Step 3-----
---
outp=[t',w',xi',abs(w-xi)'];
end
%-----END-----
---
```

4. So sánh thực nghiệm

Phương pháp thực nghiệm được đưa ra ở đây để so sánh tính chính xác của phương pháp BDF dạng khối liên tục (với $k = 3$) và phương pháp BDF thông thường sử dụng trực tiếp vòng lặp Newton bậc 4 và bậc 5. Các kết quả được thể hiện trong bảng 1 dưới đây.

Ví dụ. Xét các phương trình sau (xem [4])

$$y' = y - t^2 + 1, y(0) = \frac{1}{2}, 0 \leq t \leq 2, (9)$$

$$\begin{cases} y' = 5e^{5t}(y - t)^2 + 1 \\ y(0) = -1, t \in [0,1] \end{cases} (10)$$

$$\begin{cases} y' = -20y + 20 \cos t - \sin t \\ y(0) = 0, t \in [0,2] \end{cases} (11)$$

$$\begin{cases} y' = -20(y - t^2) + 2t \\ y(0) = 1/3, t \in [0,1] \end{cases} (12)$$

Các kết quả thực nghiệm được cho trong bảng 1 để đánh giá sai số cho thấy ưu điểm của các phương pháp BDF dạng khối so với các phương pháp BDF thông thường ở việc tạo ra các giá trị xấp xỉ chính xác cao hơn với kích thước bước h vừa phải. Độ chính xác của BDF dạng khối với $k = 3$, ký hiệu BDFblock3, giảm dần khi h tăng lên so với các phương pháp BDF4 và BDF5 (có bậc 4, và 5 tương ứng). Điều này cũng dễ hiểu từ việc các phương pháp này có bậc cao hơn so với phương pháp được mang ra đánh giá. Tuy nhiên, nếu so sánh với phương pháp cùng bậc, thì các phương pháp BDF dạng khối liên tục rõ ràng nổi trội hơn. Điều này càng được thể hiện rõ hơn trong các bài toán stiff với độ “nhất thời” (tức “transient”) càng lớn.

5. Kết luận

Việc xây dựng trình thực thi cho các phương pháp BDF dạng khối theo phương pháp lặp Newton giúp khai thác tốt độ chính xác của các phương pháp này, đặc biệt trong các bài toán stiff.

Cách tiếp cận thứ hai của trình thực thi dựa vào việc giải phương trình phi tuyến tính (5) liên tiếp qua mỗi đoạn k - bước được giải quyết tốt nhờ thuật toán Newton.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1]. O. A. Akinfenwa, S. N. Jator, and N. M. Yao, "Continuous block backward differentiation formula for solving stiff ordinary differential equations," *Computers & Mathematics with Applications*, vol. 65, no. 7, pp. 996-1005, 2013.
- [2]. O. A. Akinfenwa, S. N. Jator, and N. M. Yao, "On The Stability of Continuous Block Backward Differentiation Formula For Solving Stiff Ordinary Differential Equations," *J. of Mod. Meth. in Numer. Math.*, vol. 3, no. 2, pp. 50-58, 2012.
- [3]. J. R. Cash, "Review paper. Efficient numerical methods for the solution of stiff initial-value problems and differential algebraic equations," *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, vol. 459, no. 2032, pp. 797-815, 2003.
- [4]. R. L. Burden, and J. D. Faires, *Numerical Analysis*, 9th Edition, Brooks/Cole, 2010.

Bảng 1. Các giá trị sai số tuyệt đối ở bước cuối cùng tạo ra bởi các phương pháp BDF4, BDF5 thông thường và phương pháp BDF3 dạng khối liên tục (ký hiệu BDFblock3).

Equations	BDF4	BDF5	BDFblock3
(9), M=10, N=6,12, 30, tol=0.001,	5.3×10^{-3} , 6.5×10^{-4} , 2.4×10^{-5}	1.4×10^{-3} , 1.1×10^{-4} , 1.8×10^{-6}	6.13×10^{-2} , 5.64×10^{-3} , 3.05×10^{-4}
(10), M=10, N=6, 12, 30, tol=0.001	3.95×10^{-3} , 8.3×10^{-5} , 3.2×10^{-6}	1.8×10^{-3} , 4.4×10^{-5} , 3.5×10^{-6}	3.1×10^{-4} , 2.5×10^{-5} , 6.5×10^{-6}
(11), M=10, N=6,12, 30, 300 tol=0.001	7.8×10^3 , 7.3×10^{-2} , 9.95×10^{-8} , 1.84×10^{-11}	1.84×10^6 , 1.28, 1.1×10^{-5} , 4.6×10^{-14}	5.5×10^{-4} , 5.7×10^{-6} , 2.4×10^{-7} , 5.6×10^{-10}
(12), M=10, N= 6, 12, 30, tol=0.001	0.42, 8.74×10^{-5} , 1.02×10^{-9}	1.13, 2.2×10^{-4} , 2.53×10^{-7}	1.48×10^{-4} , 3.79×10^{-8} , 2.62×10^{-10}