



PHƯƠNG PHÁP TÍCH PHÂN TỪNG BƯỚC CẢI TIẾN TRONG PHÂN TÍCH PHẢN ỨNG ĐỘNG CỦA HỆ KẾT CẤU NHIỀU BẬC TỰ DO

Nguyễn Xuân Thành^{1*}

Tóm tắt: Từ ý tưởng trong công bố của Razavi và cộng sự (vcs) [1], bài báo này đề xuất hai phương pháp tích phân từng bước cải tiến trong phân tích phản ứng động của các kết cấu nhiều bậc tự do. Tương tự một số phương pháp khác, gia tốc của hệ kết cấu được giả thiết biến thiên với quy luật bậc hai (hoặc bậc ba). Do đó, sự biến thiên của chuyển vị của hệ có dạng đa thức bậc bốn (hoặc bậc năm). Khi đó, sẽ có năm (hoặc sáu) hệ số của đa thức cần phải xác định giá trị trong mỗi bước thời gian. Các phương trình để tìm các giá trị này bao gồm hai điều kiện ban đầu nhận được từ bước phân tích trước đó, phương trình cân bằng của hệ ở thời điểm đầu và/hoặc cuối của bước thời gian hiện thời, và hai phương trình cuối là điều kiện để cho tích phân của bình phương sai số của phương trình chuyển động trong bước thời gian đang xét đạt cực tiểu. Cách thiết lập này dẫn đến dạng đối xứng của ma trận hệ số - là dạng được ưa thích hơn trong các tính toán xử lý số - trong phương trình để tìm a_n và b_n của đa thức xấp xỉ chuyển vị. Các kết quả bằng số nhận được trong bài báo này chỉ ra rằng, với cùng các điều kiện giải bài toán như nhau, phương pháp được đề xuất ở đây cho lời giải bằng số chính xác hơn một số phương pháp thông dụng khác hiện nay.

Từ khóa: động lực học, phương pháp số, tích phân trực tiếp, tích phân từng bước, độ chính xác.

Improved time step integration methods in dynamic analysis of MDOF structures

Abstract: From the idea proposed by Razavi et al [1], this article proposes two improved schemes of time integration for dynamic analysis of multi-degree-of-freedom (MDOF) structures. As in some other methods, the author assumed a quadratic (or cubic) variation of the accelerations of MDOF structure. Therefore, in term of displacements, there are five (or six) unknown coefficients in the polynomial functions to be found. The equations to find these coefficients are the two initial conditions from previous step, the one(s) from the system equilibrium at the end(s) of the current time step, and the last two being the conditions for optimum value of integral of square of residue over the step length. The the formulation of the proposed scheme leads to the symmetric form of the coefficient matrix in the equation for finding a_n and b_n which is preferable in numerical computations. In addition, numerical results obtained in this article show that, with the same problem settings, the proposed scheme attains higher accuracy compared with ones from other commonly used methods.

Keywords: dynamics, numerical method, direct integration, time-step integration, accuracy.

Nhận ngày 9/8/2017; sửa xong 20/9/2017; chấp nhận đăng 26/9/2017

Received: August 9th, 2017; revised: September 20th, 2017; accepted: September 26th, 2017



1. Giới thiệu

Lời giải tìm phản ứng của hệ kết cấu, cả tuyến tính lẫn phi tuyến, chịu tải trọng biến thiên theo thời gian thường rất hãn hữu nhận được ở dạng giải tích. Trong các tình huống này thì cần phải sử dụng các phương pháp số mà trong một số chúng là phương pháp tích phân từng bước theo thời gian. Các phương pháp này được phân nhóm thành các phương pháp tường minh và các phương pháp không tường minh [2,3]. Một phương pháp được gọi là tường minh nếu phương trình chuyển động được thỏa mãn tại thời điểm bắt đầu của bước thời gian đang xét và phương pháp tính toán các giá trị chuyển vị và vận tốc tại thời điểm cuối của bước thời gian đang xét. Ngược lại, nếu các phương trình chuyển động được thỏa mãn tại thời điểm cuối của bước thời gian đang xét thì phương pháp được gọi là phương pháp không tường minh. Các

¹ TS, Khoa Xây dựng DD & CN, Trường Đại học Xây dựng.

* Tác giả chính. E-mail: thanhhx@nuce.edu.vn.

phương pháp tường minh yêu cầu tính toán ít hơn trong mỗi bước thời gian nhưng số bước thời gian nhiều, trong khi đó các phương pháp không tường minh lại yêu cầu tính toán nhiều hơn trong mỗi bước thời gian, với bước lớn hơn nên số lượng các bước sẽ ít hơn [4].

Hiệu quả của một phương pháp tích phân từng bước theo thời gian được đánh giá qua các đặc trưng ổn định, độ chính xác và tính hội tụ của phương pháp. Trong các phương pháp ổn định có điều kiện, độ lớn của mỗi bước chia theo thời gian không được phép vượt quá bước thời gian tới hạn vì nếu vượt quá giá trị này thì lời giải sẽ nhanh chóng trở nên không bị chặn và sự mất ổn định số của nghiệm của bài toán sẽ xảy ra. Sai số là vấn đề không thể tránh khỏi đối với bất kỳ một phương pháp số nào để giải phương trình chuyển động của hệ kết cấu. Để đánh giá sai số của phương pháp tích phân từng bước theo thời gian, người ta thường khảo sát hai chỉ số là sai số về chu kỳ riêng (dispersion), và suy giảm biên độ (dissipation). Tính ổn định và độ chính xác số được chỉ ra là phụ thuộc vào tỷ số giữa độ lớn của bước thời gian và chu kỳ dao động riêng cơ bản của hệ. Theo định lý Lax, thì tính tương thích và ổn định của phương pháp sẽ đảm bảo lời giải hội tụ đến giá trị chính xác [3].

Đã có nhiều phương pháp tích phân từng bước theo thời gian được đề xuất, trong đó có những phương pháp quen thuộc như các phương pháp sai phân, các phương pháp trong họ phương pháp β -Newmark, phương pháp θ -Wilson. Một số phương pháp khác ít được biết đến hơn nhưng cũng có một số ưu điểm nhất định nào đó, chẳng hạn như phương pháp được [1, 5, 6] đề xuất. Trong [1], Razavi vcs đã đề xuất một phương pháp với nhiều tính năng ưu việt. Phương pháp này có bước thời gian tới hạn lên đến $1,24T$ với T là chu kỳ dao động riêng không cản của hệ kết cấu một bậc tự do. Ngoài ra, phương pháp này có độ chính xác cao hơn một số phương pháp thông dụng khác. Bậc hội tụ của phương pháp này là bốn, trong khi đó bậc hội tụ của phương pháp β -Newmark chỉ là hai. Tuy nhiên, khi áp dụng phương pháp cho hệ nhiều bậc tự do thì sai số tăng lên, làm giảm độ chính xác của phương pháp. Năm 2013, Nguyen [7] đề xuất một cải tiến nâng cao độ chính xác của phương pháp Razavi. Các đặc trưng tính hiệu quả khác của phương pháp số cũng được cải thiện đáng kể, được thể hiện trong bài báo Nguyen vcs [8] vào năm 2014. Vào năm 2015, phương pháp này cũng đã được áp dụng vào giải quyết các bài toán phi tuyến vật liệu [9]. Việc áp dụng phương pháp này vào bài toán hệ kết cấu nhiều bậc tự do chưa được xét tới. Trong nghiên cứu này, tác giả sẽ đề xuất hai phương pháp cải tiến để có thể áp dụng vào giải bài toán hệ nhiều bậc tự do với độ chính xác được nâng cao. Các đặc trưng số khác như độ ổn định và độ hội tụ sẽ được trình bày trong các nghiên cứu tiếp theo.

Bài báo bắt đầu với việc giới thiệu tóm tắt các phương pháp do Razavi vcs đề xuất năm 2007 [1] và Nguyen đề xuất năm 2013 [7]. Sau đó, bài báo triển khai công thức cụ thể của các phương pháp cải tiến. Trong các triển khai này, tác giả tiếp cận giải bài toán theo cả 2 hướng: giải trực tiếp phản ứng của hệ, và giải thông qua tổ hợp các phản ứng theo các mode. Các biểu thức nhận được từ các triển khai này được cụ thể hóa trong chương trình mã nguồn mở được viết theo ngôn ngữ Julia là DirectStepIntegration.jl cho phép người sử dụng tiếp cận và phát triển cải tiến chương trình một cách dễ dàng. Các ví dụ bằng số có so sánh với kết quả nhận được từ các phương pháp thường dùng khác sẽ minh họa cho hiệu quả của các phương pháp đề xuất.



2. Tóm tắt phương pháp được đề xuất trong [1] và [7]

Trong [1], chuyển vị động $u(\tau)$ của hệ một bậc tự do ở bước thời gian thứ n được giả thiết là hàm đa thức bậc bốn:

$$u(\tau) = a_n \tau^4 + b_n \tau^3 + c_n \tau^2 + d_n \tau + e_n \quad (1)$$

trong đó: a_n, b_n, \dots, e_n là các hệ số cần phải được xác định. Biến τ chạy trong khoảng $[0, h]$, trong đó h là độ lớn của bước thời gian đã được chọn trước.

Giá trị ban đầu u_n và \dot{u}_n ở bước thời gian thứ n đang xét, bằng với các phản ứng của hệ ở thời điểm cuối của bước thời gian trước đó. Với điều kiện này, khi cho $\tau = 0$, ta có được hai hệ số e_n và d_n như sau:

$$e_n = u_n; d_n = \dot{u}_n \quad (2)$$

Từ các biểu thức (1) và (2), kết hợp với phương trình chuyển động ở thời điểm ban đầu của bước thời gian thứ n , xác định được hệ số c_n như sau:

$$c_n = (P_n - C\dot{u}_n - Ku_n) / (2M) \quad (3)$$

trong đó: M, C, K tương ứng là khối lượng, hệ số cản nhớt và độ cứng của hệ; P_n là giá trị của lực tác dụng ở thời điểm t_n . Chỉ còn cần xác định tiếp a_n và b_n . Razavi đề xuất dùng hai phương trình sau đây. Phương trình thứ nhất là phương trình chuyển động của hệ ở thời điểm cuối của bước thời gian thứ n đang xét:



$$M(12a_n h^2 + 6b_n h + 2c_n) + C(4a_n h^3 + 3b_n h^2 + 2c_n h + d_n) + K(a_n h^4 + b_n h^3 + c_n h^2 + d_n h + e_n) = P_{n+1} \quad (4)$$

trong đó: các hệ số c_n, d_n và e_n đã biết ở các biểu thức (2) và (3).

Phương trình thứ hai là phương trình làm cho hàm tích phân của sai số trong khoảng thời gian đang xét, được định nghĩa như trong phương trình (5) dưới đây bằng 0.

$$I \equiv \int_0^h (M\ddot{u}(\tau) + C\dot{u}(\tau) + Ku(\tau) - P(\tau)) d\tau = 0 \quad (5)$$

Rõ ràng rằng, phương trình này dẫn đến kết quả chỉ tốt khi số dư (biểu thức dưới dấu tích phân) có một dấu (dương hoặc âm), điều mà chúng ta không chắc chắn được. Để khắc phục điều này, năm 2013, Nguyen [7] đã đề xuất dùng các phương trình cực tiểu hóa hàm bình phương sai số trong bước thời gian đang xét như sau:

$$I \equiv \int_0^h (M\ddot{u}(\tau) + C\dot{u}(\tau) + Ku(\tau) - P(\tau))^2 d\tau \rightarrow \min \quad (6)$$

trong đó: hàm đa thức xấp xỉ chuyển vị có bậc là năm:

$$u = a_n \tau^5 + b_n \tau^4 + c_n \tau^3 + d_n \tau^2 + e_n \tau + f_n \quad (7)$$

Điều kiện để hàm số I trong phương trình (6) đạt cực tiểu sẽ cho ta hệ phương trình để tìm a_n và b_n :

$$\frac{\partial I}{\partial a_n} = 0; \quad \frac{\partial I}{\partial b_n} = 0 \quad (8)$$

Có thể xem chi tiết các biểu thức nhận được của a_n và b_n trong [7]. Các kết quả giải bài toán một bậc tự do chịu các nguyên nhân khác nhau cho thấy phương pháp đề xuất có độ chính xác cao [7].



3. Đề xuất cải tiến các phương pháp và ứng dụng giải hệ nhiều bậc tự do

Các phương pháp số có thể được áp dụng trực tiếp vào giải hệ nhiều bậc tự do với các ẩn số là chuyển vị thực của hệ hoặc có thể được áp dụng vào giải hệ nhiều bậc tự do với các ẩn số là các chuyển vị theo mode của hệ [10]. Trong nghiên cứu này, các phương pháp cải tiến được đề xuất và áp dụng vào giải bài toán trong cả hai trường hợp với ẩn số là chuyển vị thực của hệ và ẩn số là chuyển vị theo mode của hệ. Các công thức của phương pháp tương ứng cho cả hai trường hợp này đều sẽ được trình bày dưới đây.

3.1 Cải tiến phương pháp được đề xuất trong Razavi vcs [1]

Trong cải tiến này, với hệ nhiều bậc tự do, hàm đa thức cho véc-tơ chuyển vị \mathbf{u} là hàm bậc bốn như trong phương trình (9).

$$\mathbf{u}(\tau) = \mathbf{a}_n \tau^4 + \mathbf{b}_n \tau^3 + \mathbf{c}_n \tau^2 + \mathbf{d}_n \tau + \mathbf{e}_n \quad (9)$$

trong đó: với trường hợp hệ nhiều bậc tự do, các hệ số cần được xác định $\mathbf{a}_n, \mathbf{b}_n, \dots, \mathbf{e}_n$ đều là các véc-tơ.

Giá trị ban đầu của véc-tơ chuyển vị \mathbf{u} và véc-tơ vận tốc $\dot{\mathbf{u}}$ của hệ ở bước thời gian thứ n đang xét bằng với các phản ứng của hệ ở thời điểm cuối của bước thời gian trước đó. Từ đó, ta có:

$$\mathbf{e}_n = \mathbf{u}_n; \quad \mathbf{d}_n = \dot{\mathbf{u}}_n \quad (10)$$

Phương trình chuyển động ở thời điểm ban đầu của bước thời gian thứ n như sau:

$$\mathbf{M}\ddot{\mathbf{u}}(0) + \mathbf{C}\dot{\mathbf{u}}(0) + \mathbf{K}\mathbf{u}(0) = 2\mathbf{M}\mathbf{c}_n + \mathbf{C}\mathbf{d}_n + \mathbf{K}\mathbf{e}_n = \mathbf{P}_n \quad (11)$$

trong đó: $\mathbf{M}, \mathbf{C}, \mathbf{K}$ tương ứng là các ma trận khối lượng, ma trận cản nhớt, và ma trận độ cứng của hệ; \mathbf{P}_n là véc-tơ lực tác dụng tại thời điểm t_n . Từ các biểu thức (9) và (10), ta xác định được hệ số \mathbf{c}_n như sau:

$$\mathbf{c}_n = \mathbf{M}^{-1}(\mathbf{P}_n - \mathbf{C}\dot{\mathbf{u}}_n - \mathbf{K}\mathbf{u}_n) / 2 \quad (12)$$

Sau khi tìm được các hệ số $\mathbf{e}_n, \mathbf{d}_n$ và \mathbf{c}_n từ phương trình (10) và (12), các điều kiện để cho phiếm hàm I đạt cực tiểu được sử dụng, với I được xác định như trong phương trình (13). Đây là một cải tiến so với đề xuất ban đầu của Razavi vcs [1] như được thể hiện trong phương trình (5) cho hệ có một bậc tự do.

$$I \equiv \int_0^h \mathbf{R}^T \mathbf{R} d\tau \rightarrow \min \quad (13)$$

trong đó: \mathbf{R} là véc-tơ các số dư có số thành phần đúng bằng số bậc tự do của hệ:

$$\mathbf{R} \equiv \mathbf{M}\ddot{\mathbf{u}}(\tau) + \mathbf{C}\dot{\mathbf{u}}(\tau) + \mathbf{K}\mathbf{u}(\tau) - \mathbf{P}(\tau) \quad (14)$$

với $\mathbf{P}(\tau)$ là véc-tơ các hàm tải trọng trong khoảng thời gian đang xét. Nếu hàm này khả tích, thì có thể nhận được biểu thức giải tích của I dễ dàng và chính xác. Trong nhiều trường hợp, hàm $\mathbf{P}(\tau)$ có thể phức tạp hoặc được cho dưới dạng bảng các giá trị rời rạc ứng với các thời điểm nhất định, nên phải tính tích phân (13) gần đúng bằng một phương pháp xấp xỉ nào đó. Khi đó, cũng có thể xấp xỉ $\mathbf{P}(\tau)$ bằng hàm đa thức hoặc một hàm khả tích khác để đưa vào tính toán phiếm hàm I . Khi xấp xỉ $\mathbf{P}(\tau)$ bằng hàm đa thức, độ chính xác của kết quả nhận được sẽ được cải thiện khi sử dụng các đa thức bậc cao. Để đơn giản, trong nghiên cứu này, sự biến thiên của $\mathbf{P}(\tau)$ trong khoảng thời gian đang xét được giả thiết có dạng tuyến tính:

$$\mathbf{P}(\tau) = \mathbf{P}_n + \frac{(\mathbf{P}_{n+1} - \mathbf{P}_n)\tau}{h} \quad (15)$$

Điều kiện để cho phiếm hàm I đạt cực trị, được thể hiện trong phương trình (16), sẽ cho phép xác định được các hệ số \mathbf{a}_n và \mathbf{b}_n :

$$\frac{\partial I}{\partial \mathbf{a}_n} = 0; \quad \frac{\partial I}{\partial \mathbf{b}_n} = 0 \quad (16)$$

Cụ thể, từ phương trình (16), ta có:

$$\begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{a}_n \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} \mathbf{r}_n \\ \mathbf{s}_n \end{bmatrix} \quad (17)$$

trong đó:

$$\mathbf{Q}_{11} = \frac{\partial^2 I}{\partial \mathbf{a}_n \partial \mathbf{a}_n^T}; \quad \mathbf{Q}_{12} = \frac{\partial^2 I}{\partial \mathbf{a}_n \partial \mathbf{a}_n^T}; \quad \mathbf{Q}_{21} = \frac{\partial^2 I}{\partial \mathbf{a}_n \partial \mathbf{b}_n^T} = \mathbf{Q}_{12}^T; \quad \text{và} \quad \mathbf{Q}_{22} = \frac{\partial^2 I}{\partial \mathbf{b}_n \partial \mathbf{b}_n^T} \quad (18)$$

và $\mathbf{r}_n, \mathbf{s}_n$ tương ứng là phần còn lại trong các biểu thức (16a) và (16b). Về mặt hình thức, có thể viết:

$$\mathbf{r}_n = \frac{\partial I}{\partial \mathbf{a}_n} - \mathbf{q}_{11}\mathbf{a}_n - \mathbf{q}_{12}\mathbf{b}_n; \quad \text{và} \quad \mathbf{s}_n = \frac{\partial I}{\partial \mathbf{b}_n} - \mathbf{q}_{21}\mathbf{a}_n - \mathbf{q}_{22}\mathbf{b}_n \quad (19)$$

Các biểu thức triển khai của $\mathbf{Q}_{11}, \mathbf{Q}_{12}, \mathbf{Q}_{21}, \mathbf{Q}_{22}, \mathbf{r}_n$ và \mathbf{s}_n cho bước thời gian thứ n được cho cụ thể trong gói chương trình tính toán DirectStepIntegration.jl được nói đến trong Mục 4.

Sau khi đã xác định được các đại lượng trên, ta có:

$$\begin{bmatrix} \mathbf{a}_n \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_n \\ \mathbf{s}_n \end{bmatrix} \quad (20)$$

Từ đó, phản ứng của hệ ở thời điểm cuối của bước thời gian đang xét là:

$$\mathbf{u}_{n+1} = \mathbf{a}_n h^4 + \mathbf{b}_n h^3 + \mathbf{c}_n h^2 + \mathbf{d}_n h + \mathbf{e}_n; \quad \text{và} \quad \dot{\mathbf{u}}_{n+1} = 4\mathbf{a}_n h^3 + 3\mathbf{b}_n h^2 + 2\mathbf{c}_n h + \mathbf{d}_n \quad (21)$$

Các véc-tơ này sẽ được sử dụng làm các điều kiện ban đầu cho tính toán phản ứng của hệ trong bước thời gian tiếp theo. Quá trình này sẽ lặp lại cho đến khi tìm được phản ứng của hệ ở bước thời gian cuối cùng trong khoảng thời gian cho trước. Trong trường hợp lấy ẩn số là chuyển vị theo mode, thì mỗi phản ứng theo mode sẽ là nghiệm của một bài toán một bậc tự do ứng theo từng mode. Lúc này, tất cả các đại lượng véc-tơ xuất hiện trong các công thức của phương pháp cải tiến trên đây lúc này đều suy biến thành các đại lượng vô hướng tương ứng. Sau khi xác định được phản ứng theo mode của hệ, các phản ứng theo mode này được tổ hợp tuyến tính với nhau để được phản ứng thực của hệ [10].

3.2 Cải tiến phương pháp được đề xuất trong Nguyen [7]

Trong [7], đa thức xấp xỉ của \mathbf{u} trong khoảng thời gian thứ n được giả thiết có bậc 5, lớn hơn bậc của đa thức xấp xỉ trong Razavi vcs [1] một bậc. Một cải tiến khác được đề xuất trong [7] là việc sử dụng các phương trình cực tiểu hóa hàm bình phương sai số trong bước thời gian đang xét, như trong các phương trình (6) và (8). Tuy nhiên, các công thức nhận được mới chỉ áp dụng cho trường hợp hệ một bậc tự do. Với hệ nhiều bậc tự do, sai số của phương trình chuyển động không còn là một hàm số (hay một đại lượng vô hướng) thay đổi theo thời gian nữa mà là một véc-tơ \mathbf{R} có số thành phần bằng số bậc tự do của hệ. Với sự thay đổi này, cần phải điều chỉnh lại các công thức đã nêu trong Nguyen [7]. Việc điều chỉnh các công thức này bắt đầu với việc giả thiết đa thức xấp xỉ véc-tơ chuyển vị \mathbf{u} vẫn là bậc năm theo thời gian:

$$\mathbf{u}(\tau) = \mathbf{a}_n \tau^5 + \mathbf{b}_n \tau^4 + \mathbf{c}_n \tau^3 + \mathbf{d}_n \tau^2 + \mathbf{e}_n \tau + \mathbf{f}_n \quad (22)$$

trong đó: các hệ số $\mathbf{a}_n, \mathbf{b}_n, \dots, \mathbf{f}_n$ đều là các véc-tơ cần tìm.

Từ các điều kiện ban đầu của chuyển vị của hệ ở bước thời gian thứ n đang xét, ta có:

$$\mathbf{f}_n = \mathbf{u}_n; \quad \mathbf{e}_n = \dot{\mathbf{u}}_n \quad (23)$$



Từ phương trình chuyển động ở thời điểm ban đầu của bước thời gian thứ n :

$$\mathbf{M}\ddot{\mathbf{u}}(0) + \mathbf{C}\dot{\mathbf{u}}(0) + \mathbf{K}\mathbf{u}(0) = 2\mathbf{M}\mathbf{d}_n + \mathbf{C}\mathbf{e}_n + \mathbf{K}\mathbf{f}_n = \mathbf{P}_n \quad (24)$$

ta xác định được hệ số \mathbf{d}_n như sau:

$$\mathbf{d}_n = \mathbf{M}^{-1}(\mathbf{P}_n - \mathbf{C}\dot{\mathbf{u}}_n - \mathbf{K}\mathbf{u}_n) / 2 \quad (25)$$

Lúc này, còn ba hệ số cần phải xác định là \mathbf{a}_n , \mathbf{b}_n và \mathbf{c}_n . Sử dụng phương trình chuyển động ở thời điểm cuối của bước thời gian thứ n :

$$\mathbf{M}(20\mathbf{a}_n h^3 + 12\mathbf{b}_n h^2 + 6\mathbf{c}_n h + \mathbf{d}_n) + \mathbf{C}(5\mathbf{a}_n h^4 + 4\mathbf{b}_n h^3 + 3\mathbf{c}_n h^2 + 2\mathbf{d}_n h + \mathbf{e}_n) + \mathbf{K}(\mathbf{a}_n h^5 + \mathbf{b}_n h^4 + \mathbf{c}_n h^3 + \mathbf{d}_n h^2 + \mathbf{e}_n h + \mathbf{f}_n) = \mathbf{P}_{n+1} \quad (26)$$

ta xác định được hệ số \mathbf{c}_n là hàm của hai hệ số \mathbf{a}_n và \mathbf{b}_n như sau:

$$\mathbf{c}_n = -(\mathbf{6M}h + 3\mathbf{C}h^2 + \mathbf{K}h^3)^{-1} \left[(\mathbf{20M}h^3 + 5\mathbf{C}h^4 + \mathbf{K}h^5)\mathbf{a}_n + (\mathbf{12M}h^2 + 4\mathbf{C}h^3 + \mathbf{K}h^4)\mathbf{b}_n + (\mathbf{M} + \mathbf{C}h + \mathbf{K}h^2)\mathbf{d}_n + (\mathbf{C} + \mathbf{K}h)\mathbf{e}_n + \mathbf{K}\mathbf{f}_n - \mathbf{P}_{n+1} \right] \quad (27)$$

Tiếp theo, sau khi thay \mathbf{f}_n , \mathbf{e}_n , \mathbf{d}_n và \mathbf{c}_n từ các phương trình (23), (25) và (27) vào các biểu thức của $\mathbf{u}(\tau)$, $\dot{\mathbf{u}}(\tau)$ và $\ddot{\mathbf{u}}(\tau)$ và thế vào công thức (14) xác định sai số \mathbf{R} trong bước thời gian đang xét, ta thấy \mathbf{R} lúc này chỉ còn là hàm của hai hệ số chưa biết là \mathbf{a}_n và \mathbf{b}_n .

$$\mathbf{R} \equiv \mathbf{M}\ddot{\mathbf{u}}(\tau) + \mathbf{C}\dot{\mathbf{u}}(\tau) + \mathbf{K}\mathbf{u}(\tau) - \mathbf{P}(\tau) \quad (28)$$

Các hệ số chưa biết này được xác định từ điều kiện để cho I như trong biểu thức (13) đạt cực tiểu.

$$\frac{\partial I}{\partial \mathbf{a}_n} = 0; \quad \frac{\partial I}{\partial \mathbf{b}_n} = 0 \quad (29)$$

Các biến đổi biểu thức trên, tuy cồng kềnh, nhưng không phức tạp và có thể được thực hiện thủ công hoặc nhờ sự trợ giúp của các chương trình toán học có khả năng tính toán trên ký hiệu, chẳng hạn như wxMaxima, Mathematica, ... Do khuôn khổ có hạn, bài báo tác giả không trình bày các công thức cụ thể xác định \mathbf{a}_n và \mathbf{b}_n trong công thức (20) ở mục trước cũng như ở phương pháp cải tiến này. Người đọc có thể tham khảo trực tiếp các công thức này trong phần mã nguồn của gói chương trình DirectStepIntegration.jl được nhắc đến trong Mục 4.

Sau khi xác định tất cả các hệ số, phản ứng của hệ ở thời điểm cuối của bước thời gian đang xét là:

$$\mathbf{u}_{n+1} = \mathbf{a}_n h^5 + \mathbf{b}_n h^4 + \mathbf{c}_n h^3 + \mathbf{d}_n h^2 + \mathbf{e}_n h + \mathbf{f}_n; \quad (30)$$

$$\dot{\mathbf{u}}_{n+1} = 5\mathbf{a}_n h^4 + 4\mathbf{b}_n h^3 + 3\mathbf{c}_n h^2 + 2\mathbf{d}_n h + \mathbf{e}_n$$

Các véc-tơ này sẽ được sử dụng làm các điều kiện ban đầu cho tính toán phản ứng của hệ trong bước thời gian tiếp theo. Quá trình này sẽ lặp lại cho đến khi tìm được phản ứng của hệ ở bước thời gian cuối cùng. Trong trường hợp lấy ẩn số là chuyển vị theo mode, cũng giống như trong Mục 3.1, thì các công thức trên đây được vận dụng cho các hệ một bậc tự do với ẩn số là các phản ứng theo từng mode. Các phản ứng theo mode này sau đó được tổ hợp tuyến tính với nhau để được phản ứng thực của hệ.



4. Gói chương trình DirectStepIntegration.jl

Các chương trình tự động hóa tính toán được viết trên ngôn ngữ Julia [11]. Ngôn ngữ Julia là một ngôn ngữ mới ra đời nhưng có nhiều ưu điểm vượt trội so với các ngôn ngữ tính toán khoa học khác [12]. Các chương trình tkris4 và tkris5 tương ứng với hai phương pháp cải tiến trên, và được đóng gói với tên chung là DirectStepIntegration.jl. Người dùng đã cài đặt ngôn ngữ Julia có thể tải xuống gói này để có thể thực hiện lại các ví dụ của bài báo hoặc tự do sử dụng trong các bài toán riêng của mình (Với những người dùng MATLAB, do gói chương trình DirectStepIntegration.jl có mã nguồn mở và có cú pháp lệnh gần giống như các lệnh của MATLAB nên mã nguồn của gói chương trình này có thể chuyển đổi dễ dàng sang các lệnh tương ứng của MATLAB.) Các bước thực hiện tính toán rất đơn giản với các lệnh được thực hiện từ dấu nhắc của Julia như sau (nội dung phía sau dấu # là ghi chú, nếu thấy không cần thiết, có thể bỏ qua):

Các bước khởi tạo

Pkg.clone("https://github.com/tkris1004/DirectStepIntegration.git");

push!(LOAD_PATH,pwd());

using DirectStepIntegration;

Tải gói chương trình về

Đặt đường dẫn

Khai báo sử dụng gói

Sử dụng *tkris4* và *tkris5* với các tham số mặc định

$u, u_d, u_{2d} = \text{tkris4}();$ # Véc-tơ chuyển vị: u , véc-tơ vận tốc: u_d , véc-tơ gia tốc: u_{2d}

hoặc: $u, u_d, u_{2d} = \text{tkris5}();$ # Véc-tơ chuyển vị: u , véc-tơ vận tốc: u_d , véc-tơ gia tốc: u_{2d}

Sử dụng *tkris4* và *tkris5* với các tham số được xác định từ người dùng

$u, u_d, u_{2d} = \text{tkris4}(dth, tz, ic, p);$ # Véc-tơ chuyển vị: u , véc-tơ vận tốc: u_d , véc-tơ gia tốc: u_{2d}

hoặc: $u, u_d, u_{2d} = \text{tkris5}(dth, tz, ic, p);$ # Véc-tơ chuyển vị: u , véc-tơ vận tốc: u_d , véc-tơ gia tốc: u_{2d}

Các tham số cần được khai báo trước khi sử dụng các câu lệnh *tkris4* và *tkris5* gồm có:

- *dth*: khai báo về đặc trưng của hệ dưới dạng:

$dth = (M, C, K);$ # Với M, C, K là ma trận khối lượng, ma trận cản nhớt, và ma trận độ cứng của hệ.

- *tz*: khai báo về các thời điểm cách đều nhau cần xác định phản ứng của hệ dưới dạng:

$tz = (h, t_end);$ # Với h là độ lớn bước chia theo thời gian và t_end là thời điểm cuối;

t_end phải lớn hơn h nhưng không nhất thiết phải là bội số của h .

- *ic*: khai báo về điều kiện ban đầu ở thời điểm t_0 của hệ, như sau:

$ic = (cvbd, vtbd);$

trong đó: *cvbd* là véc-tơ chuyển vị ban đầu và *vtbd* là véc-tơ vận tốc ban đầu của hệ. Số thành phần trong mỗi véc-tơ này đúng bằng số bậc tự do của hệ.

- *p*: khai báo về tải trọng, chương trình có thể tự động nhận biết khi *p* được khai báo ở một trong hai dạng sau:

$p = (td_i, tt_i);$

hoặc: $p = (\text{delta}_t, tt_i);$

trong đó: $td_i = [t_1; t_2; \dots; t_N]$ là véc-tơ các thời điểm mà tại đó có khai báo các giá trị tải trọng $tt_i = [p_1; p_2; \dots; p_N]$ tương ứng, với p_i là véc-tơ hàng giá trị của các tải trọng tác dụng vào các bậc tự do của hệ. Các thời điểm này không cần phải cách đều nhau. Thời điểm cuối cùng t_N có thể lớn hơn, nhỏ hơn hoặc bằng t_end . Khi các thời điểm cách đều nhau một khoảng là delta_t thì khai báo *p* dùng theo cách thứ 2.

Giá trị mặc định của các tham số (theo ví dụ 13.8 trong [13]) như sau: $M = [2 \ 0; 0 \ 1];$ $C = [0 \ 0; 0 \ 0];$ $tz = (0.1, 10.0);$ $K = [96 \ -32; -32 \ 32];$ $dth = (M, C, K);$ $cvbd = [0 \ 0; 0 \ 0];$ $vtbd = [0 \ 0; 0 \ 0];$ $ic = (cvbd, vtbd);$ $p = (0.1, \text{ones}(15, 1)) * [0 \ 100];$



5. Các ví dụ minh họa

Mục này chọn trình bày ba ví dụ: (1) hệ một bậc tự do chịu tác dụng của tải trọng dạng xung nửa hình sin; (2) hệ hai bậc tự do chịu tác dụng của tải trọng hằng số đặt đột ngột; và (3) hệ hai bậc tự do chịu tác dụng của tải trọng động đất. Trong hai ví dụ đầu, tác giả so sánh kết quả của các phương pháp đề xuất và một số phương pháp thông dụng khác như phương pháp gia tốc tuyến tính, phương pháp Fox-Goodwin (trong nhóm phương pháp β -Newmark), phương pháp θ -Wilson, ... với nghiệm giải tích. Trong hai ví dụ cuối, kết quả nhận được từ cách giải trực tiếp ra phản ứng thực và cách giải gián tiếp thông qua tổ hợp phản ứng theo mode được trình bày. Ảnh hưởng của độ lớn bước chia theo thời gian cũng được khảo sát và được chỉ ra trong ví dụ cuối. Trong các ví dụ này, để đảm bảo việc so sánh được nhất quán, độ lớn bước thời gian được lấy theo các ví dụ đã có - thường là nhỏ hơn 1/10 chu kỳ dao động riêng nhỏ nhất - và đều thỏa mãn nhỏ hơn bước thời gian tới hạn để đảm bảo độ ổn định số của các kết quả.

5.1 Ví dụ 1 (Ví dụ 5.4 trong [10])

Xét hệ một bậc tự do có các đặc trưng $M = 0,2533 \text{kip.s}^2/\text{in.}$, $C = 0,1592 \text{kip.s}/\text{in.}$, và $K = 10 \text{kips}/\text{in.}$ Khi bắt đầu chịu tác dụng của tải trọng, hệ ở trạng thái nghỉ. Phản ứng của hệ được xác định ở các thời điểm cách nhau 0,1s (ứng với $h = T_n/10$). Tải trọng xung có dạng nửa bước sóng hình sin trong 0,6s đầu tiên với $p(t) = 10 \sin(\pi t/0,6) \text{kips}$. Từ Bảng 1 ta nhận xét rằng các kết quả nhận được ở các thời điểm từ phương pháp cải tiến thường bám sát nghiệm giải tích hơn. Bước chia theo thời gian mịn hơn cũng đã được khảo sát cho ra các kết quả còn khả quan hơn nữa khi so sánh với các phương pháp khác.



5.2 Ví dụ 2 (Ví dụ 13.8 trong [13])

Xét hệ hai bậc tự do không cản, chịu tác dụng của tải trọng, với các đặc trưng như sau:

$$\mathbf{M} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \mathbf{K} = \begin{bmatrix} 96 & -32 \\ -32 & 32 \end{bmatrix}; \text{ và } \mathbf{p}(t) = \begin{bmatrix} 0 \\ 100 \end{bmatrix} \quad (31)$$

Khi bắt đầu chịu tác dụng của tải trọng, hệ ở trạng thái nghỉ. Phản ứng của hệ được xác định ở các thời điểm cách nhau 0,1s. Bảng 2 so sánh các kết quả chuyển vị u_{1n} và u_{2n} tương ứng ở bậc tự do 1 và 2. Giá trị trong cột (a) nhận được khi giải trực tiếp ra phản ứng thực của hệ, giá trị trong cột (b) nhận được khi giải thông các phản ứng theo mode $q_1(t)$ và $q_2(t)$ rồi tổ hợp lại theo phương trình (32) dưới đây.

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = [\phi_1 \quad \phi_2] \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} \quad (32)$$

với $\phi_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ và $\phi_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ là các mode dao động riêng của hệ, được xác định từ bài toán giá trị

riêng. Ta cũng thấy rằng các phương pháp đề xuất bám sát nghiệm giải tích, chính xác hơn đôi chút so với phương pháp của Razavi vcs [1], và có độ chính xác vượt trội so với các phương pháp khác. Các giá trị trong hai cột (a) và (b) rất sát nhau (hầu như trùng nhau), chứng tỏ rằng, về mặt tính toán số, cách giải trực tiếp tìm phản ứng thực của hệ có thể thay thế hoàn toàn tương đương cho phương pháp giải gián tiếp thông qua tổ hợp các phản ứng theo mode. Điều này có một thuận lợi là ta có thể tránh được các bài toán giá trị riêng.

Bảng 1. So sánh các phương pháp - hệ một bậc tự do

	t_n (s)	Giải tích	Gia tốc tuyến tính	Fox-Goodwin	θ -Wilson ($\theta = 1,4$)	Razavi	tkris4	tkris5
u_n	0,1	0,0328	0,0300	0,0155	0,0280	0,0318	0,0318	0,0318
	0,2	0,2332	0,2193	0,2056	0,2053	0,2275	0,2275	0,2274
	0,3	0,6487	0,6166	0,6223	0,5791	0,6335	0,6336	0,6336
	0,4	1,1605	1,1130	1,1462	1,0544	1,1337	1,1338	1,1339
	0,5	1,5242	1,4782	1,5281	1,4242	1,4893	1,4893	1,4895
	0,6	1,4814	1,4625	1,5019	1,4568	1,4478	1,4476	1,4480
	0,7	0,9245	0,9514	0,9357	1,0329	0,9038	0,9034	0,9036
	0,8	0,0593	0,1273	0,0558	0,2958	0,0584	0,0580	0,0579
	0,9	-0,7751	-0,6954	-0,7929	-0,4913	-0,7571	-0,7573	-0,7577
	1,0	-1,2718	-1,2208	-1,2973	-1,0669	-1,2428	-1,2425	-1,2432
\dot{u}_n	0,1	0,9567	0,8995	0,9273	0,8414	0,9351	0,9358	0,9354
	0,2	3,1383	2,9819	3,0621	2,7942	3,0674	3,0682	3,0680
	0,3	4,9674	4,7716	4,8623	4,5146	4,8553	4,8552	4,8558
	0,4	4,8408	4,7419	4,7526	4,6201	4,7319	4,7304	4,7317
	0,5	1,9783	2,1082	1,9592	2,3568	1,9347	1,9320	1,9333
	0,6	-3,0849	-2,6911	-3,0021	-1,9762	-3,0138	-3,0164	-3,0161
	0,7	-7,4096	-7,1468	-7,4842	-6,1924	-7,4611	-7,4612	-7,4631
	0,8	-8,7279	-8,7758	-8,9256	-8,0835	-8,8759	-8,8729	-8,8762
	0,9	-6,7347	-7,1539	-6,9710	-7,1976	-6,9192	-6,9141	-6,9171
	1,0	-2,3696	-3,0508	-2,5461	-4,0084	-2,5205	-2,5155	-2,5165

Bảng 2. So sánh các phương pháp - hệ nhiều bậc tự do

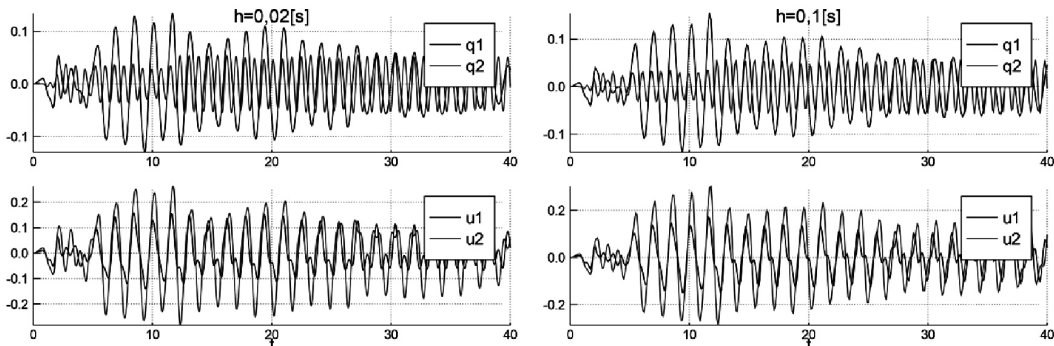
	t_n (s)	Giải tích	Gia tốc tuyến tính	Fox-Goodwin	θ -Wilson ($\theta=1,4$)	Razavi	tkris4		tkris5	
							(a)	(b)	(a)	(b)
u_{1n}	0,1	0,006	0,012	0,006	0,015	0,007	0,007	0,007	0,006	0,006
	0,2	0,096	0,109	0,095	0,124	0,096	0,096	0,096	0,096	0,096
	0,3	0,424	0,430	0,423	0,446	0,424	0,424	0,424	0,424	0,424
	0,4	1,103	1,081	1,104	1,057	1,103	1,104	1,104	1,103	1,103
	0,5	2,089	2,027	2,091	1,922	2,089	2,089	2,089	2,089	2,089
	0,6	3,144	3,060	3,147	2,876	3,144	3,144	3,144	3,144	3,144
	0,7	3,929	3,867	3,931	3,670	3,929	3,929	3,929	3,929	3,929
	0,8	4,160	4,165	4,159	4,060	4,160	4,159	4,159	4,160	4,160
	0,9	3,748	3,837	3,745	3,915	3,747	3,747	3,747	3,748	3,748
	1,0	2,848	2,993	2,845	3,265	2,848	2,849	2,849	2,848	2,848
u_{2n}	0,1	0,487	0,475	0,487	0,468	0,487	0,487	0,487	0,487	0,487
	0,2	1,800	1,763	1,801	1,715	1,800	1,800	1,800	1,800	1,800
	0,3	3,562	3,510	3,563	3,409	3,561	3,561	3,561	3,562	3,562
	0,4	5,329	5,286	5,329	5,166	5,329	5,329	5,329	5,329	5,329
	0,5	6,762	6,750	6,761	6,665	6,762	6,762	6,762	6,762	6,762
	0,6	7,714	7,732	7,713	7,717	7,715	7,714	7,714	7,714	7,714
	0,7	8,209	8,233	8,208	8,274	8,210	8,210	8,210	8,209	8,209
	0,8	8,330	8,331	8,330	8,377	8,330	8,330	8,330	8,330	8,330
	0,9	8,107	8,081	8,109	8,090	8,106	8,107	8,107	8,107	8,107
	1,0	7,487	7,465	7,486	7,448	7,486	7,486	7,486	7,487	7,487

5.3. Ví dụ 3

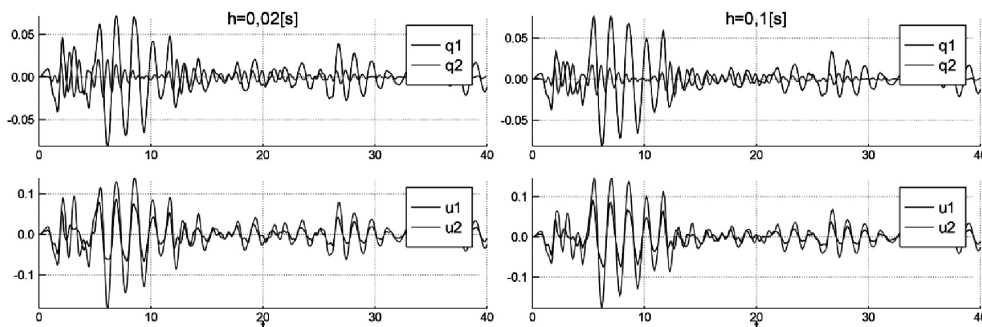
Xem xét hệ hai bậc tự do có **M** và **K** như trong Ví dụ 2. Xét hệ trong trường hợp không cản và có cản.

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; \text{ hoặc } C = \begin{bmatrix} 4/3 & -4/15 \\ -4/15 & 8/15 \end{bmatrix} \quad (33)$$

Hệ chịu tác dụng của chuyển động nền đất được cho theo bản ghi gia tốc nền của trận động đất El Centro 1940 (phụ lục 6 trong [10]). Khi bắt đầu chịu tác dụng của tải trọng động đất, hệ ở trạng thái nghỉ. Độ lớn bước thời gian lần lượt bằng 0,02s và 0,1s. Các kết quả được chỉ ra trên Hình 1 và Hình 2. Phương pháp đề xuất thứ hai được sử dụng do nó có độ chính xác cao hơn. Việc giải trực tiếp tìm phản ứng thực ($u_1(t)$ và $u_2(t)$) của hệ cho kết quả hoàn toàn trùng khớp với việc giải bài toán thông qua tổ hợp các phản ứng theo mode của hệ ($q_1(t)$ và $q_2(t)$) như trong phương trình (32). Hơn nữa, khi so sánh các đồ thị ở cột bên trái với các đồ thị ở cột bên phải, ta thấy có thể sử dụng bước chia h tương đối lớn (trong ví dụ cụ thể này $h = 0,1s$, lớn hơn 1/10 chu kỳ dao động riêng nhỏ nhất là $T_2 = 2\pi/8$) trong quá trình giải bài toán.



Hình 1. Phản ứng của hệ hai bậc tự do với độ lớn bước chia theo thời gian khác nhau - hệ không cản



Hình 2. Phản ứng của hệ hai bậc tự do với độ lớn bước chia theo thời gian khác nhau - hệ có cản

6. Kết luận

Bài báo đã trình bày hai đề xuất cải tiến phương pháp tích phân từng bước theo thời gian, sử dụng tiêu chuẩn cực tiểu của phiếm hàm tích phân của bình phương sai số trong bước thời gian đang xét (Mục 3.1), và kết hợp với việc nâng bậc của đa thức xấp xỉ chuyển vị (Mục 3.2). Các công thức đều cho hệ tuyến tính tham số hằng nhiều bậc tự do. Độ chính xác của các phương pháp cải tiến này đều cao hơn một số phương pháp tích phân từng bước thông dụng khác. Bài báo cũng đã giới thiệu gói chương trình mã nguồn mở DirectStepIntegration.jl để người sử dụng có thể tùy biến áp dụng vào các bài toán của mình.

Lời cảm ơn: Tác giả bài báo bày tỏ sự cảm ơn đến hỗ trợ của đề tài nghiên cứu khoa học trọng điểm cấp trường, mang mã số 145-2017/KHXD-TĐ.

Tài liệu tham khảo

1. Razavi S., Abolmaali A., Ghassemieh M. (2007), "A Weighted Residual Parabolic Acceleration Time Integration Method for Problems in Structural Dynamics," *Computational Methods in Applied Mathematics*, 7(3):227-238.
2. Dokainish M.A., Subbaraj K. (1989), "A survey of direct time integration methods in computational structural dynamics - I. Explicit methods", *Computers & Structures*, 32(6):1371-1386.
3. Hughes T.J.R., Belytschko T. (1983), "A precis of developments in computational methods for transient analysis", *Journal of Applied Mechanics*, 50(4b):1033-1041.
4. Subbaraj K., Dokainish M.A. (1989), "A survey of direct time-integration methods in computational structural dynamics - II. Implicit methods", *Computers & Structures*, 32(6):1387-1401.
5. Lim H., Taylor R.L. (2001), "An explicit-implicit method for flexible-rigid multibody systems", *Finite elements in analysis and design*, 37(11):881-900.
6. Soroushian A. (2008), "A technique for time integration analysis with steps larger than the excitation steps", *Communications in Numerical Methods in Engineering*, 24(12):2087-2011.
7. Nguyen X.T. (2013), "A New Time Integration Scheme for Dynamic Analysis of Structures", *Tuyển tập Hội nghị Khoa học toàn quốc Cơ học Vật rắn biến dạng lần thứ XI*, Ho-Chi-Minh City, Vietnam, 2:1064-1072.
8. Nguyen X.T., Pham A.H., Razavi H. (2014), "Numerical Aspects of a Time Integration Scheme for Dynamic Analysis of Structures", *Proceedings of the 3rd International Conference on Engineering Mechanics and Automation (ICEMA 3)*, Hanoi, 1:474-480.
9. Nguyen X.T., Pham A.H. (2015), "Application of a time stepping scheme in analysis of nonlinear dynamics problems", *Tuyển tập Hội nghị Khoa học toàn quốc Cơ học Vật rắn biến dạng lần thứ XII*, Danang, Vietnam, 2:1270-1277.
10. Chopra A.K. (2012), *Dynamics of structures (Theory and applications to earthquake engineering)*, 4th Ed., Prentice Hall.
11. "Wikipedia," Wikimedia Foundation, Inc., (2017). [https://en.wikipedia.org/wiki/Julia_\(programming_language\)](https://en.wikipedia.org/wiki/Julia_(programming_language)) truy cập ngày 01/8/2017.
12. "Julia," JuliaLang (2017). <https://julia.org/> truy cập ngày 01/8/2017.
13. Humar J.L. (1990), *Dynamics of Structures*, Englewood Cliffs, N.J., Prentice Hall.