Computers and Structures 195 (2018) 74–84

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

A composite collocation method with low-period elongation for structural dynamics problems

Ce Huang, Minghui Fu *

School of Engineering, Sun Yat-Sen University, No. 135 West Xingang Road, Guangzhou, China

ARTICLE INFO

Article history:
Received 15 March 2017
Accepted 27 September 2017

Keywords:
Structural dynamics
Time integration algorithms
Collocation methods
High-order accuracy
Sparse linear algebra equations system
Controllable numerical dissipation

ABSTRACT

This paper presents a novel time integration algorithm for solving linear structural dynamic problems in the framework of the high-order collocation method. When two Gauss points in the integration interval are selected as collocation points, both an A-stable algorithm with third order accuracy and a non-dissipative algorithm with fourth order accuracy can be derived from a second order collocation polynomial. The only difference is that the former obtains a numerical solution at the middle point of the time interval, while the latter has a solution at the end of the interval. A new composite method is established through applying these two algorithms alternately, which combines the advantages of the numerical dissipation property of the third order algorithm and the high-order accuracy of the fourth order algorithm. The usage frequency of the two algorithms during the whole step-by-step integration procedure is an important parameter affecting the numerical dissipation, which is investigated in this study. As the algebraic equations systems solved by the two algorithms are exactly same, no extra computation effort is introduced.

1. Introduction

Step-by-step direct time integration algorithms have been widely used in structural dynamics analysis [1–3]. The time integration algorithms can be classified into two categories, explicit algorithms [4–6] and implicit algorithms [7–9]. Explicit algorithms calculate system response in the next time-step from displacements, velocities, and accelerations in the current and previous time-steps. Explicit methods are usually conditional stable. The time-step has to be restricted below a critical value to maintain numerical stability. Implicit algorithms obtain the response in the next time-step by solving algebra equations systems, which involve the responses in the current, previous and next time-steps. Implicit algorithms need more computational cost in each time-step, however, basically maintain unconditional stability, which indicates that implicit algorithms can use a lager time-step compared with conditionally explicit stable schemes. The Newmark method [10], $Wilson-\theta$ method [11], $CH-\alpha$ method [7], $HHT-\alpha$ method [12], and $WBZ-\alpha$ method [12] are considered as implicit methods. However, most widely used explicit and implicit methods only have 2nd-order accuracy, which may result in unacceptable accumulation of error in long-term simulations. Therefore, it is necessary to develop high-order algorithms with unconditional stability.

Many high-order algorithms have been proposed using different methodologies [13,14]. Based on the Galerkin methodology, Fung generalized the classical diagonal and first sun-diagonal Padé approximants to the exponential function using the complex time step method [15,16], the least-squares method [17], the weighted residual method [13,14], the collocation method [18] and the differential quadrature method [19,20]. Mancuso and Ubertini [21,22] obtained high-order unconditional stable algorithms based on the collocation and Nørsett methodologies. Algorithms based on p-order collocation polynomials require solving p sets of linear algebraic equations in each time-step. Although these approaches lead to desired accuracy, they basically need solving a large and complex system of algebraic equations. Therefore, special care must be taken in solving the system of algebraic equations so that these methods can be efficiently implemented.

In the numerical analysis of standard finite-element semi-discrete systems, only a few low-frequency modes approximate the original continuous system. The quality of numerical algorithms depends crucially on their dissipative property. Fung [18] shows that A-stable algorithms with 3rd-order accuracy and a non-dissipative algorithm with 4th-order accuracy can be developed respectively using two collocation points. However, the non-dissipative algorithm with 4th-order accuracy does not have

* Corresponding author.
E-mail addresses: stsfmh@mail.sysu.edu.cn, huangcealg@163.com (M. Fu).

https://doi.org/10.1016/j.compstruc.2017.09.012
0045-7949/

algorithmic damping to dissipate fictitious high-frequency responses. In this paper, an A-stable algorithm with 3rd-order accuracy and a non-dissipative algorithm with 4th-order accuracy are alternately employed to establish a composite method, which has high accuracy and numerical dissipation property. In addition, as the algebraic equations system obtained by both algorithms of order 3 and order 4 are exactly same; the proposed composite algorithm doesn't introduce extra computational work. Compared with the 3rd-order algorithm, the proposed method improves the computational accuracy. Furthermore, the coefficient matrix of the algebraic equations system obtained by the proposed composite method shows a lot of specific properties after elimination and simplification operations. However, conventional direct solution methods cannot take advantage of these benefits to save computational cost. Particular solution methods for the algebraic equations system are presented considering two types of coefficient matrices.

2. Collocation methods

The motion equation of a discretized linear structure with n-degree-of-freedom can be described by,

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{F}(t) \tag{1}$$

with the initial condition, $\mathbf{x}(0) = \mathbf{x}_0, \dot{\mathbf{x}}(0) = \dot{\mathbf{x}}_0$, where $\mathbf{M}, \mathbf{C}, \mathbf{K}$ are the mass, damping, and stiffness matrices, respectively; $\mathbf{F}(t)$ is the vector of externally applied loads; and $\mathbf{x}$, $\dot{\mathbf{x}}$, $\ddot{\mathbf{x}}$ are the displacement, velocity and acceleration vectors of the system.

Assuming $\mathbf{q} = \mathbf{x}$, $\mathbf{p} = \mathbf{M}\dot{\mathbf{q}}$, $\mathbf{z} = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix}$, Eq. (1) can be transformed into a first order differential equation, namely

$$\begin{cases} \dot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{p} \\ \dot{\mathbf{p}} = \mathbf{F}(t) - \mathbf{C}\mathbf{M}^{-1}\mathbf{p} - \mathbf{K}\mathbf{q} \end{cases} \quad or \quad \dot{\mathbf{z}} = f(t, \mathbf{z}) \tag{2}$$

where

$$f(t, \mathbf{z}) = \begin{bmatrix} \mathbf{0} & \mathbf{M}^{-1} \\ -\mathbf{K} & -\mathbf{C}\mathbf{M}^{-1} \end{bmatrix} \mathbf{z} + \begin{bmatrix} \mathbf{0} \\ \mathbf{F}(t) \end{bmatrix}$$

By applying time discretization with a time-step τ, the solution of Eq. (2) in the time interval $[t_k, t_{k+1}]$ can be approximated by the following second order polynomial

$$\begin{aligned} \mathbf{z}(t) = {} & \frac{(t - t_k - \eta_1\tau)(t - t_k - \eta_2\tau)}{\eta_1\eta_2\tau^2}\mathbf{z}(t_k) \\ & + \frac{(t - t_k)(t - t_k - \eta_2\tau)}{\eta_1(\eta_1 - \eta_2)\tau^2}\mathbf{z}(t_k + \eta_1\tau) \\ & + \frac{(t - t_0)(t - t_k - \eta_1\tau)}{\eta_2(\eta_2 - \eta_1)\tau^2}\mathbf{z}(t_k + \eta_2\tau) \end{aligned} \tag{3}$$

where $t_k = k\tau$ and $t_{k+1} = (k+1)\tau$ are two consecutive time instants, and $t_k + \eta_1\tau, t_k + \eta_2\tau$ are two chosen collocation points in the interval $[t_k, t_{k+1}]$. The derivative of Eq. (3) with respect to time can be written as

$$\begin{aligned} \dot{\mathbf{z}}(t) = {} & \frac{2t - 2t_k - \tau(\eta_1 + \eta_2)}{\tau^2\eta_1\eta_2}\mathbf{z}(t_k) + \frac{2t - 2t_k - \tau\eta_2}{\tau^2\eta_1(\eta_1 - \eta_2)}\mathbf{z}(t_k + \eta_1\tau) \\ & + \frac{-2t + 2t_k + \tau\eta_1}{\tau^2(\eta_1 - \eta_2)\eta_2}\mathbf{z}(t_k + \eta_2\tau) \end{aligned} \tag{4}$$

The four unknowns $\mathbf{q}(t_k + \eta_1\tau)$, $\mathbf{q}(t_k + \eta_2\tau)$, $\mathbf{p}(t_k + \eta_1\tau)$, $\mathbf{p}(t_k + \eta_2\tau)$ can be solved by substituting Eqs. (3) and (4) into Eq. (2) and letting the residual be zero at the two collocation points. During this process, an algebraic equation system with $4n$ unknowns needs to be solved.

According to Ref. [18], the numerical solution approximated by Eq. (3) obtains at least second-order accuracy. However, if the collocation parameters η_1, η_2 are given by the solutions of the following Eq. (5), the accuracy of the resultant algorithms can be improved to be third order or fourth order.

$$3(\mu + 1)\eta^2 - 2(\mu + 2)\eta + 1 = 0 \tag{5}$$

The solutions are

$$\eta_{1,2} = \frac{\mu + 2 \pm \sqrt{1 + \mu + \mu^2}}{3(1 + \mu)} \tag{6}$$

If μ equal to any value within $[-1, 1]$, the resultant algorithm is unconditional stable and has third order accuracy. When $\mu = 1$, $\eta_{1,2} = 1/2 \pm \sqrt{3}/6$, the two collocation points are the gauss points in the interval $[t_k, t_{k+1}]$. Then the collocation method becomes the same as the Gauss-Legendre Symplectic Runge-Kutta (GLSRK) method [26], which is non-dissipative and has fourth order accuracy.

3. A composite collocation method

The collocation method with fourth order accuracy (*CM*4) gives very accurate numerical results, which is good for long-term prediction of system response and preservation of system invariant (such as energy and momentum). Moreover, larger time-step can be used to produce desired accuracy compared with those methods with lower-order accuracy. However, in structural dynamic analysis, high-frequency responses produced by the spatial discretization process usually do not represent the physical oscillation behavior of the original system and will reduce the computational accuracy. The *CM*4 method does not have numerical dissipation to damp out the fictitious high-frequency mode responses. In order to circumvent this drawback, a composite collocation method (*CCM*) is proposed in this paper, which has high-order accuracy and controllable algorithmic dissipation.

It can be seen from Eqs.(5) and (6)that, when $\mu = -0.5, \eta_{1,2} = 1 \pm \sqrt{3}/3$, the resultant third order collocation method (*CM*3) is A-stable and has numerical dissipation, and the collocation points used in the interval $[t_k, t_{k+1}]$ are $t_k + \tau \pm \sqrt{3}\tau/3$. However, if the step-size τ is reduced to 0.5τ, the two collocation points would be $t_k + \tau/2 \pm \sqrt{3}\tau/6$, which are exactly same as that using *CM*4 with step-size τ. Fig. 1 depicts the collocation points used in the two algorithms. Apparently, the unknowns using *CM*4 with step-size τ and *CM*3 with step-size 0.5τ are both $\mathbf{z}(t_k + \tau/2 \pm \sqrt{3}\tau/6)$. The algebraic equations system required to be solved using both algorithms is exactly same. The only difference is that *CM*4 obtains the numerical result of $\mathbf{z}(t_k + \tau)$, while *CM*3 obtains that of $\mathbf{z}(t_k + 0.5\tau)$. In other words, *CM*3 with step-size 0.5τ and *CM*4 with step-size τ can be regarded as two schemes of the two-point-collocation methods family. Therefore, a high-order composite method with numerical dissipation can be developed by using *CM*4 and *CM*3 alternately. Assuming CCM^{c_1/c_2} denotes that c_1 times of *CM*4 with step-size τ and c_2 times of *CM*3 with step-size 0.5τ are used alternately, $c_1\tau + 0.5c_2\tau$ would be the new step-size. By adjusting the parameter c_1/c_2, the amount of numerical dissipation can be modified. Moreover, on the premise of same computational cost, the speeds advancing over integration time for different methods would be different. Table 1 shows that, with same computational steps, the advancing speeds over time for $CCM^{2/1}$, $CCM^{1/2}$ are obviously faster than that for *CM*3 with step-size 0.5τ.

4. Solution of the algebraic equations system

For high-order time integration algorithms, computational efficiency is crucial to make them truly applicable. Compared with