Computers and Structures 193 (2017) 226–238

Contents lists available at ScienceDirect

Computers and Structures

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

# On the advantages of using the first-order generalised-alpha scheme for structural dynamic problems

C. Kadapa *, W.G. Dettmer, D. Perić

*Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Fabian Way, Swansea SA1 8EN, Wales, UK*

ARTICLE INFO

*Article history:*
Received 26 April 2017
Accepted 22 August 2017
Available online 5 September 2017

*Keywords:*
Structural dynamics
Time integration
Generalised-alpha scheme
Numerical dissipation
Overshoot

ABSTRACT

The advantages of using the generalised-alpha scheme for first-order systems for computing the numerical solutions of second-order equations encountered in structural dynamics are presented. The governing equations are rewritten so that the second-order equations can be solved directly without having to convert them into state-space. The stability, accuracy, dissipation and dispersion characteristics of the scheme are discussed. It is proved through spectral analysis that the proposed scheme has improved dissipation properties when compared with the standard generalised-alpha scheme for second-order equations. It is also proved that the proposed scheme does not suffer from overshoot. Towards demonstrating the application to practical problems, proposed scheme is applied to the benchmark example of three degrees of freedom stiff-flexible spring-mass system, two-dimensional Howe truss model, and elastic pendulum problem discretised with non-linear truss finite elements.



## 1. Introduction

Obtaining stable and accurate solutions of second-order dynamical systems encountered in science and engineering has been one of the important areas of research on numerical schemes for initial value problems (IVPs). In literature, there are several time integration schemes for solving structural dynamic problems. Several classifications exist for such schemes: *implicit* or *explicit* and *single-step* or *multi-step* being the most prominent ones. The detailed discussion of such schemes is beyond the scope of this paper and any standard book on numerical schemes for initial value problems, e.g. [1–7], may be consulted for this purpose.

Implicit schemes generally possess better stability characteristics than explicit schemes. An implicit scheme allows use of large steps for obtaining numerical solutions, hence, such schemes require less time and effort. However, it is now an established fact that use of large time steps, in implicit schemes, results in undesirable numerical dissipation in the low-frequency range. On the other hand, for structural dynamics problems discretised with finite elements, it is advantageous to be able to control the amount of numerical damping so that adverse effects of spurious higher-frequency modes on the numerical solution can be avoided. Therefore, a time integration scheme with controllable numerical damping for high-frequency modes and at the same time with less numerical dissipation in the low-frequency range is desirable. Following Hilber and Hughes [7,8], a competitive numerical scheme for structural dynamic problems should possess the following important characteristics:

1. Unconditional stability when applied to a linear problem.
2. No more than one set of implicit equations should have to be solved at each time step.
3. Second-order accuracy.
4. Controllable algorithmic dissipation in the higher modes.
5. Self-starting.
6. The scheme should not suffer from *overshoot* behaviour.

A considerable amount of research has gone into developing implicit schemes which possess the above-listed attributes. Newmark-$\beta$ scheme [9], Wilson-$\theta$ scheme [10], HHT-$\alpha$ scheme [11], Collocation scheme [8], WBZ-$\alpha$ scheme [12], HP-$\theta_1$ scheme [13], CH-$\alpha$ scheme [14] and G-$\alpha$ scheme [15] are few such schemes which satisfy some or all of the above listed criteria. Though all these schemes are unconditionally stable, implicit, single-step and second-order in nature, their differences are in the amount of numerical dissipation and whether or not they suffer from overshoot. HHT-$\alpha$, CH-$\alpha$ and WBZ-$\alpha$ schemes have been proven to suffer from overshooting, see [15] and references therein. Erlicher et al. [16] have proven the overshoot behaviour of CH-$\alpha$ scheme

* Corresponding author.
*E-mail address:* c.kadapa@swansea.ac.uk (C. Kadapa).

http://dx.doi.org/10.1016/j.compstruc.2017.08.013
0045-7949/

in the context of non-linear dynamic problems. KaiPing [15] has improved upon CH-α scheme and devised a new family of generalised-α schemes without overshoot. Though NOHHT-α and NOWBZ-α schemes, proposed by [15], are without overshoot and have better dissipation properties when compared with their counterparts with overshoot, the amount of numerical dissipation of NOCH-α remained exactly same as that of the CH-α scheme. On the similar lines, Kuhl and Crisfield [17] developed energy conserving generalised energy-momentum methods based on CH-α scheme.

It is important to note that all the schemes listed above are single-step schemes for second-order IVPs. To the knowledge of the authors, there are only a few direct multi-step schemes for second-order IVPs. Two-step composite scheme by Bathe and Baig [18] and the three-step scheme by Wen et al. [19] are two such multi-step schemes for structural dynamic problems. We refer to a recent article by Zhang et al. [20] for a comprehensive numerical analysis of such composite schemes. Though these multi-step schemes do not contain any adjustable parameter, their main disadvantage is that for the same time step size, they are computationally expensive when compared with single-step schemes. For example, for a linear problem, and for a given time step, the computational cost of Bathe's two-step scheme is twice that of a single-step scheme; and for the three-step scheme by Wen et al. [19] the computational cost is three times that of a single-step scheme. In addition, the task of book-keeping and storing variables for intermediate steps in multi-step schemes adds to unnecessary computational overheads. Furthermore, the cost and complexity of the algorithm of multi-step schemes increase many folds for non-linear problems.

In this paper, we propose to use the generalised-α scheme for the first-order dynamic systems, proposed by Jansen et al. [21] and referred as JWH-α from this point onwards, for obtaining the numerical solutions of structural dynamic problems. Recently, this scheme has been applied to nearly incompressible elasticity by Rossi et al. [22] and viscoelasticity by Zeng et al. [23]. This work is motivated by the need for a consistent time integration scheme for fluid-structure interaction (FSI) problems. CH-α and JWH-α schemes have been extensively used as time integration schemes for fluid and solid sub-problems, respectively, in numerical schemes for coupled fluid-structure interaction, see [24–30]. Investigation of time integration schemes for fluids [24] showed the excellent performance of the JWH-α scheme. Dettmer and Perić [31] used CH-α and JWH-α schemes, respectively, for fluid and solid sub-solvers to obtain second order accurate unconditionally stable weakly coupled solution scheme for FSI with small to moderate added mass effects. All of these motivate the development of the unified framework in which a single time integration scheme is used for both fluid and solid sub-problems.

The outline of the paper is as follows. The governing equations and proposed scheme are presented in Section 2. Stability and accuracy analysis are carried out in Section 3. Dissipation and dispersion characteristics of the scheme are studied in Section 4. In Section 5, it is proved that the proposed scheme does not suffer from overshoot behaviour. Finally, the algorithm is applied to three multi-degree of freedom (MDOF) examples in Section 6 and the performance of the proposed scheme is compared against CH-α and Bathe's schemes. Conclusions are drawn in Section 7.

## 2. Governing equations and the proposed time integration scheme

The governing equation for the general linear structural dynamic problem can be written in matrix-vector form as,

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F} \tag{1}$$

$$\mathbf{d}(t=0) = \mathbf{d}_0 \tag{2}$$

$$\dot{\mathbf{d}}(t=0) = \dot{\mathbf{d}}_0 \tag{3}$$

where $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are the mass, damping and stiffness matrices, respectively; $\mathbf{d}$ is the vector of displacements (including rotational degree of freedom) and $\dot{\mathbf{d}} = \mathrm{d}\mathbf{d}/\mathrm{d}t, \ddot{\mathbf{d}} = \mathrm{d}^2\mathbf{d}/\mathrm{d}t^2$ are the velocity and acceleration vectors; $\mathbf{F}$ is the vector of external nodal forces; $\mathbf{d}_0$ and $\dot{\mathbf{d}}_0$ are the initial displacement and velocity, respectively. In order for the formulation to be consistent and balance total energy, the initial acceleration should be computed as,

$$\ddot{\mathbf{d}}_0 = \mathbf{M}^{-1}\left[\mathbf{F}(t=0) - \mathbf{C}\dot{\mathbf{d}}_0 - \mathbf{K}\mathbf{d}_0\right] \tag{4}$$

In order to apply the JWH-α scheme, the second-order equation Eq. (1) is first converted into a system of first-order equations. By introducing an auxiliary variable $\mathbf{v} = \dot{\mathbf{d}}$, the equivalent first-order system can be written in the matrix form as,

$$\begin{bmatrix}\mathbf{M} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}\end{bmatrix}\begin{Bmatrix}\dot{\mathbf{d}}\\ \dot{\mathbf{v}}\end{Bmatrix} + \begin{bmatrix}\mathbf{0} & -\mathbf{M}\\ \mathbf{K} & \mathbf{C}\end{bmatrix}\begin{Bmatrix}\mathbf{d}\\ \mathbf{v}\end{Bmatrix} = \begin{Bmatrix}\mathbf{0}\\ \mathbf{F}\end{Bmatrix} \tag{5}$$

By applying the JWH-α scheme to Eq. (5), the following first-order system is obtained.

$$\begin{bmatrix}\mathbf{M} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}\end{bmatrix}\begin{Bmatrix}\dot{\mathbf{d}}_{n+\alpha_m}\\ \dot{\mathbf{v}}_{n+\alpha_m}\end{Bmatrix} + \begin{bmatrix}\mathbf{0} & -\mathbf{M}\\ \mathbf{K} & \mathbf{C}\end{bmatrix}\begin{Bmatrix}\mathbf{d}_{n+\alpha_f}\\ \mathbf{v}_{n+\alpha_f}\end{Bmatrix} = \begin{Bmatrix}\mathbf{0}\\ \mathbf{F}_{n+\alpha_f}\end{Bmatrix} \tag{6}$$

with,

$$\dot{\mathbf{d}}_{n+\alpha_m} = \alpha_m\dot{\mathbf{d}}_{n+1} + (1-\alpha_m)\dot{\mathbf{d}}_n \tag{7}$$

$$\dot{\mathbf{v}}_{n+\alpha_m} = \alpha_m\dot{\mathbf{v}}_{n+1} + (1-\alpha_m)\dot{\mathbf{v}}_n \tag{8}$$

$$\mathbf{d}_{n+\alpha_f} = \alpha_f\mathbf{d}_{n+1} + (1-\alpha_f)\mathbf{d}_n \tag{9}$$

$$\mathbf{v}_{n+\alpha_f} = \alpha_f\mathbf{v}_{n+1} + (1-\alpha_f)\mathbf{v}_n \tag{10}$$

$$\mathbf{F}_{n+\alpha_f} = \alpha_f\mathbf{F}_{n+1} + (1-\alpha_f)\mathbf{F}_n \tag{11}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\left[\gamma\dot{\mathbf{d}}_{n+1} + (1-\gamma)\dot{\mathbf{d}}_n\right] \tag{12}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\left[\gamma\dot{\mathbf{v}}_{n+1} + (1-\gamma)\dot{\mathbf{v}}_n\right] \tag{13}$$

For convenience, Eqs. (12) and (13) are rewritten as,

$$\dot{\mathbf{d}}_{n+1} = \frac{1}{\gamma\Delta t}[\mathbf{d}_{n+1} - \mathbf{d}_n] + \frac{\gamma-1}{\gamma}\dot{\mathbf{d}}_n \tag{14}$$

$$\dot{\mathbf{v}}_{n+1} = \frac{1}{\gamma\Delta t}[\mathbf{v}_{n+1} - \mathbf{v}_n] + \frac{\gamma-1}{\gamma}\dot{\mathbf{v}}_n \tag{15}$$

Now, using the Eqs. (7)–(13), the first-order matrix system Eq. (6) can be solved for $\{\mathbf{d}_{n+1}\ \mathbf{v}_{n+1}\}^T$. However, this is not a wise choice as this would require solving a matrix system which is twice as large as the original one. Even though the resulting overhead might be insignificant for small problems, the cost would increase substantially for large problems, especially when the matrix system needs to be solved at every iteration of every time step for a non-linear problem. Therefore, in the present work, we rewrite the Eq. (6) so that we only need to solve a matrix system that is exactly the same size as that of the original system.

A close observation of Eq. (6) reveals that, its first equation can be simplified to an equation involving only vectors. Accordingly, we get,

$$\dot{\mathbf{d}}_{n+\alpha_m} = \mathbf{v}_{n+\alpha_f} \tag{16}$$