Computational Mechanics 25 (2000) 430–437 © Springer-Verlag 2000

An improved predictor/multi-corrector algorithm for a time-discontinuous Galerkin finite element method in structural dynamics

C.-C. Chien, T.-Y. Wu

Abstract This work presents an improved predictor/multi-corrector algorithm for linear structural dynamics problems, based on the time-discontinuous Galerkin finite element method. The improved algorithm employs the Gauss–Seidel method to calculate iteratively the solutions that exist in the phase of the predictor/multi-corrector of the numerical implementation. Stability analyses of iterative algorithms reveal that such an improved scheme retains the unconditionally stable behavior with greater efficiency than another iterative algorithm. Also, numerical examples are presented, demonstrating that the proposed method is more stable and accurate than several commonly used algorithms in structural dynamic applications.

1 Introduction

The equations of motion for linear structural dynamic problems after spatial discretization using the finite element method can be written as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{F}(t), \quad t \in I = (0, T) \qquad (1)$$

where **M**, **C**, and **K** are the mass, the damping matrix and the stiffness matrix, respectively. In addition, $\mathbf{F}(t)$ is the applied load vector, and $\mathbf{u}(t)$ is the unknown displacement vector which, in general, is function of time, t. The dots indicate differentiation with respect to t. The initial conditions are given as follows:

$$\mathbf{u}(0) = \mathbf{u}_0, \quad \dot{\mathbf{u}}(0) = \mathbf{v}_0 \qquad (2)$$

Various direct integration methods or step-by-step time integration methods have been widely used to obtain numerical solutions for structural dynamic problems. Among these, second-order accurate methods, such as the Newmark method (1959), the Wilson-θ method (1973), the Houbolt method (1950), the Park method (1975), and the HHT-α method (1977), are most frequently used. These methods are based on semi-discretizations: finite elements are used in space to reduce a system of second-order ordinary differential equations in time. These, in turn, are discretized by traditional finite difference methods for ordinary differential equations. As widely contented, the finite element method is superior to the finite difference method. From a computational perspective, the semi-discrete method does not seem to be a pure finite element method. If, indeed, finite elements have advantages in space, they should also exhibit advantages in time (Hughes and Hulbert, 1988). Thus, many researchers have attempted to employ finite elements in the time domain (Argyris and Scharpf, 1969; Fried, 1969; Oden, 1969; Zienkiewicz and Taylor, 1991). Kinematic and mixed formulation for dynamics were derived from a general form and multibody simulations were carried out using finite elements in the time domain (Mello et al., 1990; Borri et al., 1990). Further, a general framework for interpreting time finite element formulations was proposed by Borri and Bottasso (1993). The goal of their study was to provide a unified view, where different methods emanate from the same general statement of the problem of motion expressed by Hamilton's law of varying action. Within this framework, the bi-discontinuous form and the discontinuous Galerkin form for integration algorithms were derived respectively, according to different boundary terms.

Based on a time-discontinuous Galerkin (TDG) method, the unknown fields are permitted to be discontinuous at the discrete time levels. The TDG method has been successfully applied to first-order hyperbolic problems (fluid mechanics) and parabolic problems (transient heat conduction), (Johnson, Nävert, and Pitkaranta, 1984; Johnson, 1987; Thomée, 1984). Recently, Hughes and Hulbert first applied this novel approach to the area of structural dynamics (Hughes and Hulbert, 1988; Hulbert, 1989, 1992; Hulbert and Hughes, 1990). According to their results, the TDG method possesses considerable potential absent from the traditional semi-discrete methods. In addition to lead to A-stable, higher-order accurate solution algorithms to solve ordinary differential equations, this method also achieves the asymptotic annihilation of the spurious high frequency response (if proper time finite elements are chosen). Li and Wiberg (1996, 1998) recently applied the TDG method to 2-D structural dynamic problems. Those investigations dealt with the specific TDG method that uses piecewise linear interpolations for both displacements and velocities, i.e., the P1–P1 two-field formulation element (Hulbert, 1992). They designed the predictor/multi-corrector solution algorithm in the numerical implementation of the TDG method. The algorithm involves a factorization for each fixed time step size and a few

Received 18 June 1999

C.-C. Chien (✉), T.-Y. Wu
Department of Civil Engineering,
Chung-Yuan University
Chung-Li, 32023, Taiwan

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 88-2211-E-033-003.

iterations at each step for solving the resulted system of coupled equations. Since only external dynamic loads with simple time variations are applied to the problems, the numerical implementation of this approach may seem uncomplicated. However, while the prescribed loading conditions vary arbitrarily with time (for instance, the earthquake-induced motion), a large number of iterations arise from their proposed algorithm. While the TDG method has the advantages of higher accuracy and a longer time step, its drawback is its higher computational cost.

In this paper, we present an improved predictor/multi-corrector solution algorithm for linear structural dynamics problems, based on the TDG finite element method. The proposed algorithm uses the Gauss–Seidel iteration method which can decrease the number of the iterations and can provide an accuracy comparable to the Li and Wiberg method. Two numerical examples are presented. The stability, accuracy and efficiency of the present analysis are established, by comparing with numerical results obtained by several commonly used algorithms.

2
Time-discontinuous Galerkin finite element method

Let $0 = t_1 < t_2 \cdots < t_n < t_{n+1} < \cdots < t_{N+1} = T$ be a partition of the time domain $I = (0, T)$ with corresponding time steps) $\Delta t_n = t_{n+1} - t_n$ and $I_n = (t_n, t_{n+1})$. Let the time finite element space be described by:

$$\mathbf{V}^h = \left\{ \mathbf{w}^h = \bigcup_{n=1}^{N} (P^1(I_n))^{n_{eq}} \right\} \tag{3}$$

where $\mathbf{P}^1$denotes the first-order polynomial, and each member of $\mathbf{V}^h$ is a vector consisting of $\mathbf{n}_{\text{eq}}$ linear functions on each time step I_n. All trial displacements and velocities and their corresponding weighting functions are chosen from the space $\mathbf{V}^h$. Notably, the functions in $\mathbf{V}^h$ may be discontinuous at the discrete time levels t_n. To account for this, we introduce the notation:

$$\mathbf{w}_n^+ = \lim_{\varepsilon \to 0^+} \mathbf{w}(t_n + \varepsilon) \tag{4a}$$

$$\mathbf{w}_n^- = \lim_{\varepsilon \to 0^-} \mathbf{w}(t_n + \boldsymbol{\varepsilon}) \tag{4b}$$

$$[\mathbf{w}_n] = \mathbf{w}_n^+ - \mathbf{w}_n^- \tag{4c}$$

where $[\mathbf{w}_n]$ represents the jump of $\mathbf{w}_n$ at t_n, and we denote the inner product on I_n by:

$$(\mathbf{w}, \mathbf{u})_{I_n} = \int_{I_n} \mathbf{w} \cdot \mathbf{u} \, \mathrm{d}t \tag{5}$$

The time-discontinuous Galerkin finite element method can now be formulated as follows:
Find $\mathbf{U}^h = \{\mathbf{u}^h, \mathbf{v}^h\} \in \mathbf{V}^h \times \mathbf{V}^h$ such that for all $\mathbf{W}^h = \{\mathbf{w}_1^h, \mathbf{w}_2^h\} \in \mathbf{V}^h \times \mathbf{V}^h$,

$$\mathbf{B}_{DG}(\mathbf{W}^h, \mathbf{U}^h)_n = L_{DG}(\mathbf{W}^h)_n \quad n = 1, \ 2, \ \ldots, \ N \tag{6}$$

where

$$\begin{aligned} \mathbf{B}_{DG}(\mathbf{W}^h, \mathbf{U}^h)_n = {} & (\mathbf{w}_2^h, \mathbf{L}_1\mathbf{U}^h)_{I_n} + (\mathbf{w}_1^h, \mathbf{K}\mathbf{L}_2\mathbf{U}^h)_{I_n} \\ & + \mathbf{w}_2^h(t_n^+) \cdot \mathbf{M}\mathbf{v}^h(t_n^+) + \mathbf{w}_1^h(t_n^+) \cdot \mathbf{K}\mathbf{u}^h(t_n^+) \\ & n = 1, \ 2, \ \ldots, \ N \end{aligned} \tag{7}$$

$$\begin{aligned} \mathbf{L}_{DG}(\mathbf{W}^h)_n = {} & (\mathbf{w}_2^h, \mathbf{F})_{I_n} + \mathbf{w}_2^h(t_n^+) \cdot \mathbf{M}\mathbf{v}^h(t_n^-) \\ & + \mathbf{w}_1^h(t_n^+) \cdot \mathbf{K}\mathbf{u}^h(t_n^-) \quad n = 2, \ \ldots, \ N \end{aligned} \tag{8}$$

$$\begin{aligned} \mathbf{L}_{DG}(\mathbf{W}^h)_1 = {} & (\mathbf{w}_2^h, \mathbf{F})_{I_1} + \mathbf{w}_2^h(0^+) \cdot \mathbf{M}\mathbf{v}_0 \\ & + \mathbf{w}_1^h(0^+) \cdot \mathbf{K}\mathbf{u}_0 \quad n = 1 \end{aligned} \tag{9}$$

and

$$\mathbf{L}_1\mathbf{U}^h = \mathbf{M}\dot{\mathbf{v}}^h + \mathbf{C}\mathbf{v}^h + \mathbf{K}\mathbf{u}^h \tag{10}$$

$$\mathbf{L}_2\mathbf{U}^h = \dot{\mathbf{u}}^h - \mathbf{v}^h \tag{11}$$

It has been demonstrated that this time-discontinuous Galerkin method is unconditionally stable and of third-order accuracy if the P1-P1 two-field element is used (Hulbert, 1989, 1992).

3
Solution algorithm

In this section, we present an improved iterative solution algorithm to obtain the solutions for the resulting system of equations from Eq. (6). Considering a typical time step $I_n = (t_n, t_{n+1})$, let $\mathbf{u}_1$ and $\mathbf{v}_1$ denote the nodal displacements and velocities at t_n^+, respectively, and $\mathbf{u}_2$ and $\mathbf{v}_2$ the nodal displacements and velocities at t_{n+1}^-, respectively. Also, let $\mathbf{u}_1^-$ and $\mathbf{v}_1^-$ represent the nodal displacements and velocities at t_n^-, respectively, which are determined from either the previous step's calculations or, if $n = 1$, the initial data. Thus, the displacements and velocities at an arbitrary time $t \in (t_n, t_{n+1})$ can be expressed as follows:

$$\mathbf{u}^h(t) = \boldsymbol{\phi}_1(t)\mathbf{u}_1 + \boldsymbol{\phi}_2(t)\mathbf{u}_2 \tag{12a}$$

$$\mathbf{v}^h(t) = \boldsymbol{\phi}_1(t)\mathbf{v}_1 + \boldsymbol{\phi}_2(t)\mathbf{v}_2 \tag{12b}$$

where $\boldsymbol{\phi}_1(t) = \frac{t_{n+1} - t}{\Delta t_n}$ and $\boldsymbol{\phi}_2(t) = \frac{t - t_n}{\Delta t_n}$, i.e., the P1-P1 element is defined for the displacement and velocity fields (Hulbert, 1989, 1992). By substituting Eq. (12) and their corresponding weighting functions into Eq. (6), and performing the integration explicitly, one can obtain the following matrix equation as follows:

$$\begin{bmatrix} \frac{1}{2}\mathbf{K} & \frac{1}{2}\mathbf{K} & -\frac{1}{3}\Delta t_n\mathbf{K} & -\frac{1}{3}\Delta t_n\mathbf{K} \\ -\frac{1}{2}\mathbf{K} & \frac{1}{2}\mathbf{K} & -\frac{1}{6}\Delta t_n\mathbf{K} & -\frac{1}{3}\Delta t_n\mathbf{K} \\ \frac{1}{3}\Delta t_n\mathbf{K} & \frac{1}{6}\Delta t_n\mathbf{K} & \frac{1}{2}\mathbf{M} + \frac{1}{3}\Delta t_n\mathbf{C} & \frac{1}{2}\mathbf{M} + \frac{1}{6}\Delta t_n\mathbf{C} \\ \frac{1}{6}\Delta t_n\mathbf{K} & \frac{1}{3}\Delta t_n\mathbf{K} & -\frac{1}{2}\mathbf{M} + \frac{1}{6}\Delta t_n\mathbf{C} & \frac{1}{2}\mathbf{M} + \frac{1}{3}\Delta t_n\mathbf{C} \end{bmatrix} \times \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{K}\mathbf{u}_1^- \\ 0 \\ \mathbf{F}_1 + \mathbf{M}\mathbf{v}_1^- \\ \mathbf{F}_2 \end{Bmatrix} \tag{13}$$

where

$$\mathbf{F}_1 = \int_{I_n} \boldsymbol{\phi}_1(t)\mathbf{F} \, \mathrm{d}t; \quad \mathbf{F}_2 = \int_{I_n} \boldsymbol{\phi}_2(t)\mathbf{F} \, \mathrm{d}t \tag{14}$$

Solving Eq. (13) is a non-trivial task to solve Eq. (13), because this equation is coupled and four times larger than the original Eq. (1). An iterative predictor/multi-corrector algorithm is designed to decrease the computational cost.. We first recast the system as follows:

$$\begin{bmatrix} \mathbf{K} & 0 & -\frac{1}{6}\Delta t_n\mathbf{K} & \frac{1}{6}\Delta t_n\mathbf{K} \\ 0 & \mathbf{K} & -\frac{1}{2}\Delta t_n\mathbf{K} & -\frac{1}{2}\Delta t_n\mathbf{K} \\ 0 & 0 & \mathbf{M}^* & \frac{2}{3}\mathbf{M}+\frac{1}{6}\Delta t_n\mathbf{C} \\ 0 & 0 & \frac{1}{2}\Delta t_n\mathbf{C}+\frac{1}{3}\Delta t_n^2\mathbf{K} & \mathbf{M}^* \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{K}\mathbf{u}_1^- \\ \mathbf{K}\mathbf{u}_2^- \\ \mathbf{F}_1^* \\ \mathbf{F}_2^* \end{Bmatrix} \tag{14}$$

where

$$\mathbf{M}^* = \mathbf{M} + \tfrac{1}{2}\Delta t_n\mathbf{C} + \tfrac{1}{6}\Delta t_n^2\mathbf{K} \tag{15a}$$

$$\mathbf{F}_1^* = \tfrac{5}{3}\left(\mathbf{F}_1 + \mathbf{M}\mathbf{v}_1^-\right) - \tfrac{1}{3}\mathbf{F}_2 - \tfrac{2}{3}\Delta t_n\mathbf{K}\mathbf{u}_1^- \tag{15b}$$

$$\mathbf{F}_2^* = \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{M}\mathbf{v}_1^- - \Delta t_n\mathbf{K}\mathbf{u}_1^- \tag{15c}$$

are the effective mass matrix and force vectors. Clearly, the third and fourth rows in Eq. (14) have been partially decoupled from the first and second ones, and they can be solved separately as follows:

$$\begin{bmatrix} \mathbf{M}^* & \frac{2}{3}\mathbf{M}+\frac{1}{6}\Delta t_n\mathbf{C} \\ \frac{1}{2}\Delta t_n\mathbf{C}+\frac{1}{3}\Delta t_n^2\mathbf{K} & \mathbf{M}^* \end{bmatrix} \begin{Bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_1^* \\ \mathbf{F}_2^* \end{Bmatrix} \tag{16}$$

The above equation can be expressed more concisely in matrix form as follows:

$$[\mathbf{A}]\{\mathbf{V}\} = \{\mathbf{F}^*\} \tag{17}$$

If initial predictor values of $\mathbf{v}_1$ and $\mathbf{v}_2$ are given, for instance, chosen being $\mathbf{v}_1^-$, Eq. (16) is solved iteratively for the corrected values of $\mathbf{v}_1$ and $\mathbf{v}_2$, until the required accuracy is obtained. In the numerical examples, an accuracy requirement of $\varepsilon = 10^{-6}$ is specified, and the iteration process is performed until $\varepsilon = \|(\mathbf{v}_1^{k+1} - \mathbf{v}_1^k), (\mathbf{v}_2^{k+1} - \mathbf{v}_2^k)\|$ is met. Then, the first row in Eq. (14) is used for determining $\mathbf{u}_1$ and the second one for determining $\mathbf{u}_2$. Table 1 summarizes the above solution algorithm, demonstrating that the implementation of the TDG method is not complex. Notably, an improved algorithm using Gauss–Seidel iteration scheme in step **C** of the time integration can effectively reduce the number of iterations in the computation. This procedure is superior to that described in the paper by Li and Wiberg (1996) (using Gauss–Jacobi iteration method).

Table 1. Implementation of the TDG method

A. Data input and initialization
$\mathbf{u}_1^- = \mathbf{u}_0;\ \mathbf{v}_1^- = \mathbf{v}_0;\ t = 0$

B. Form effective mass matrix and perform factorization
$\mathbf{M}^* = \mathbf{M} + \frac{1}{2}\Delta t_n\mathbf{C} + \frac{1}{6}\Delta t_n^2\mathbf{K}$

C. Time integration
(a). Form generalized force vectors
$\mathbf{F}_1^* = \frac{5}{3}(\mathbf{F}_1 + \mathbf{M}\mathbf{v}_1^-) - \frac{1}{3}\mathbf{F}_2 - \frac{2}{3}\Delta t_n\mathbf{K}\mathbf{u}_1^-$
$\mathbf{F}_2^* = \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{M}\mathbf{v}_1^- - \Delta t_n\mathbf{K}\mathbf{u}_1^-$
(b). Predictor phase
$\mathbf{v}_1 = \mathbf{v}_1^-, \quad k = 0$
(c). Multi-corrector phase (using Gauss-Seidel block iteration)
$\mathbf{M}^*\mathbf{v}_2^{k+1} = \mathbf{F}_2^* - (\frac{1}{2}\Delta t_n\mathbf{C} + \frac{1}{3}\Delta t_n^2\mathbf{K})\mathbf{v}_1^k$
$\mathbf{M}^*\mathbf{v}_1^{k+1} = \mathbf{F}_1^* - (\frac{2}{3}\mathbf{M} + \frac{1}{6}\Delta t_n\mathbf{C})\mathbf{v}_2^{k+1}$
If $\bar{\varepsilon} > \varepsilon$ go to (c)
Else $k = k + 1$
(d). Output solution
$\mathbf{v}_1^- \leftarrow \mathbf{v}_2$
$\mathbf{u}_1^- \leftarrow \mathbf{u}_1^- + \frac{\Delta t_n}{2}(\mathbf{v}_1 + \mathbf{v}_2)$
$t \leftarrow t + \Delta t_{n+1}$
If $t < T$ go to (a), else terminate

4 Stability analysis of iteration

Here, we attempt a proof of the stability analysis for the above-mentioned two iterative solution techniques. Consider a one degree of freedom system given by the motion equation

$$\ddot{\mathbf{u}} + 2\xi\omega\dot{\mathbf{u}} + \omega^2\mathbf{u} = 0 \tag{18}$$

If we split the matrix [**A**] in Eq. (17) into the matrices [**M**] and [**N**], i.e., $[\mathbf{A}] = [\mathbf{M}] - [\mathbf{N}]$, then Eq. (17) becomes

$$([\mathbf{M}] - [\mathbf{N}])\{\mathbf{V}\} = \{\mathbf{F}^*\} \tag{19a}$$

or

$$[\mathbf{M}]\{\mathbf{V}\} = [\mathbf{N}]\{\mathbf{V}\} + \{\mathbf{F}^*\} \tag{19b}$$

Multiplying the whole Eq. (19b) by $[\mathbf{M}]^{-1}$ results in the following matrix equation:

$$\{\mathbf{V}\} = [\mathbf{M}]^{-1}[\mathbf{N}]\{\mathbf{V}\} + [\mathbf{M}]^{-1}\{\mathbf{F}^*\} \tag{20}$$

or in compact form:

$$\{\mathbf{V}\} = [\mathbf{Z}]\{\mathbf{V}\} + \{\mathbf{Y}\} \tag{21}$$

Herein, an arbitrary initial value $\mathbf{V}^0$ may be assumed and iterative calculations performed until the acceptable error ε is achieved. In other words, Eq. (21) can be rewritten as follows:

$$\{\mathbf{V}^{k+1}\} = [\mathbf{Z}]\{\mathbf{V}^k\} + \{\mathbf{Y}\}, \quad k = 1,\ 2,\ 3,\ \ldots \tag{22}$$

The spectral radius ρ of the matrix [**Z**] should be smaller than 1 to allow the stability and accuracy characteristics for iterative methods to be obtained. On the other hand, if the spilt [**A**] can be chosen to make the spectral radius ρ as small as possible, a very efficient iterative method may be obtained (Asaithambi, 1995). The stability analyses including the Gauss–Jacobi and Gauss–Seidel iterative methods are discussed below:

(1) The Gauss–Jacobi iterative method

In this method, the above-mentioned matrix [**M**] is constructed by the principal diagonal elements of matrix [**A**]. Another matrix [**N**]is determined by the off-diagonal ele-

ments of [**A**]. Matrix [**Z**] involving Δt, ξ and ω can be written in explicit form as follows:

$$[\mathbf{Z}] = \begin{bmatrix} 0 & \frac{\frac{2}{3}+\frac{1}{3}\Delta t\omega}{1+\Delta t\omega\xi+\frac{1}{6}(\Delta t\omega)^2} \\ \frac{\Delta t\omega\xi+\frac{1}{3}(\Delta t\omega)^2}{1+\Delta t\omega\xi+\frac{1}{6}(\Delta t\omega)^2} & 0 \end{bmatrix} \tag{23}$$

Then, the spectral radii ρ of the matrix [**Z**] can be computed as follows:

$$\rho = \frac{\sqrt{[\Delta t\omega\xi + \frac{1}{3}(\Delta t\omega)^2][\frac{2}{3} + \frac{1}{3}\Delta t\omega]}}{1 + \Delta t\omega\xi + \frac{1}{6}(\Delta t\omega)^2} < 1 \tag{24}$$

The relationship of spectral radii ρ versus $\Delta t\omega$ is plotted in Fig. 1a. Li and Wiberg used this algorithm (called the Li algorithm) to solve Eq. (16).

(2) The Gauss–Seidel iterative method

In this method, the principal diagonal elements and, strictly, the lower triangular part of matrix [**A**] are incorporated into the [**M**]. Matrix [**N**] only contains the remaining upper triangular part of matrix [**A**]. Matrix [**Z**] involving Δt, ξ and ω can be written in the explicit form as follows:

$$[\mathbf{Z}] = \begin{bmatrix} 0 & \frac{\frac{2}{3}+\frac{1}{3}\Delta t\omega}{1+\Delta t\omega\xi+\frac{1}{6}(\Delta t\omega)^2} \\ 0 & \frac{\left[\Delta t\omega\xi+\frac{1}{3}(\Delta t\omega)^2\right]\left(\frac{2}{3}+\frac{1}{3}\Delta t\omega\right)}{[1+\Delta t\omega\xi+\frac{1}{6}(\Delta t\omega)^2]^2} \end{bmatrix} \tag{25}$$

Then, the spectral radii ρ of the matrix [**Z**] can be computed as follows:

$$\rho = \frac{[\Delta t\omega\xi + \frac{1}{3}(\Delta t\omega)^2][\frac{2}{3} + \frac{1}{3}\Delta t\omega]}{[1 + \Delta t\omega\xi + \frac{1}{6}(\Delta t\omega)^2]^2} < 1 \tag{26}$$

The relationship of spectral radii ρ versus $\Delta t\omega$ is depicted in Fig. 1b. The present analysis employs this algorithm (called the improved algorithm) to solve Eq. (16).

Figures 1a, 1b show the stability characteristics of the above two methods, respectively. It is noted that both iterative methods are unconditional stable. However, the Gauss–Seidel method refers to the present analysis, and they exhibit a better behavior of convergence compared with the Gauss–Jacobi method. The spectral radii ρ of the matrix [**Z**] for the Gauss-Seidel method (Fig. 1b) are much smaller than those for the Gauss–Jacobi method (Fig. 1a), and hence the Gauss–Seidel iteration method improves the computational efficiency in iteration stage. It is also observed that increasing the damping ratio will produces the decrease of the computational efficiency particularly in the

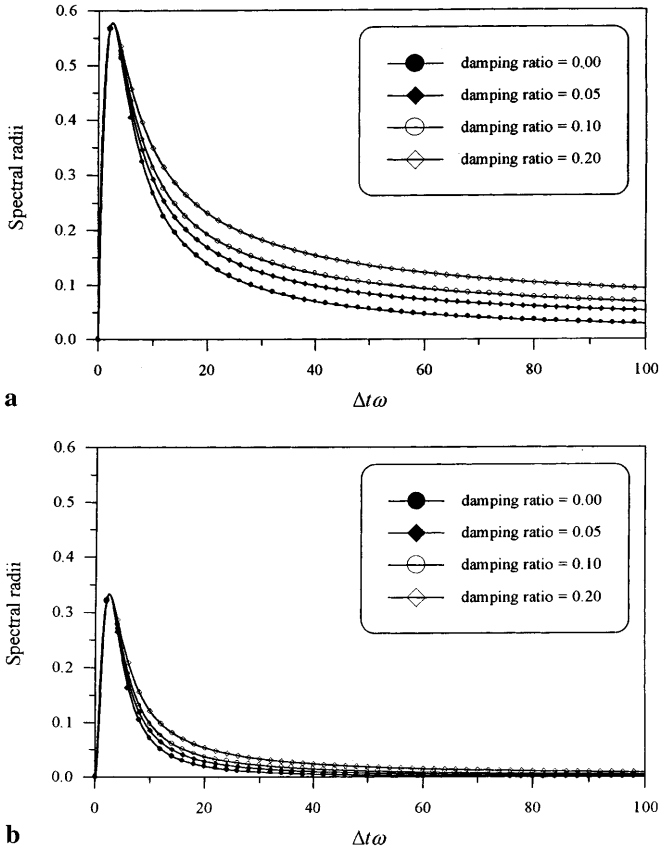

Fig. 1a, b. Spectral radius $\rho(Z)$ for **a** Gauss–Jacobi iteration method **b** Gauss–Seidel iteration method

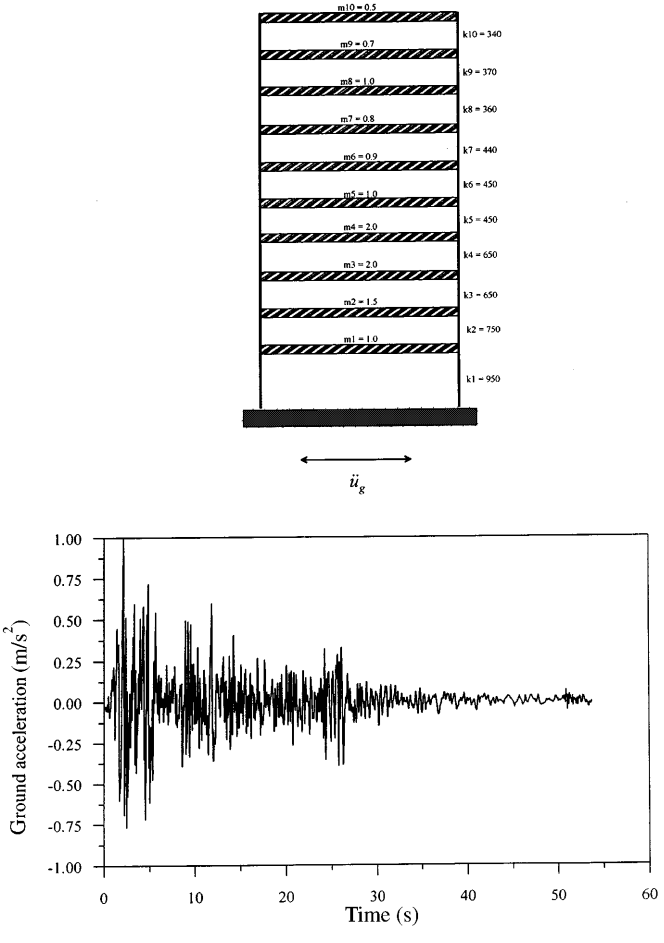

Fig. 2. A ten-story shear building subjected to El Centro ground acceleration

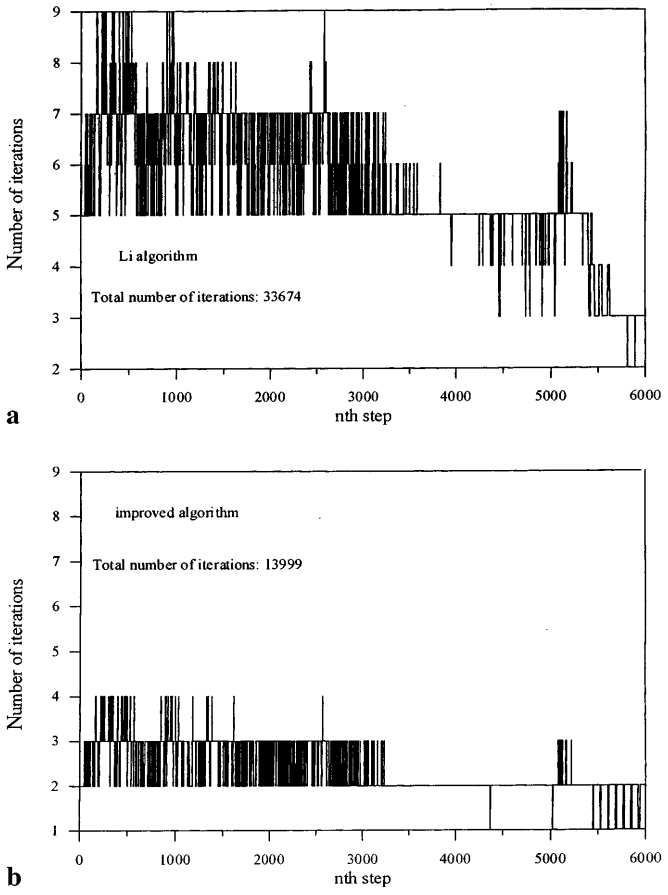

Fig. 3a, b. Numbers of iteration needed for every time step **a** the Li algorithm **b** the improved algorithm

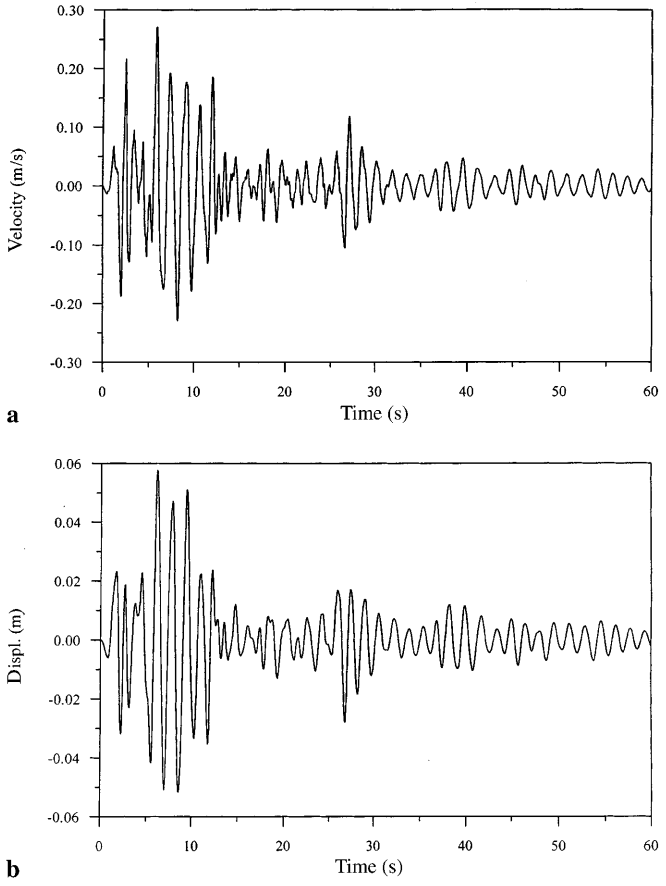

Fig. 4a, b. Response histories of the ten-story shear building **a** velocity at the top **b** displacement at the top

case of the higher frequency or the use of longer time steps. For the Gauss–Seidel method, one can only predict the velocity $\mathbf{v}_1$ that is determined by the velocity $\mathbf{v}_2$ (i.e., $\mathbf{v}_1^-$) of the previous time step. Thus, the increasing of the iteration numbers can be avoided and the accuracy still can be achieved even longer time step is used. In fact, one can further increase the speed of the computation using some accelerated factors (for instance, SOR method) for the Gauss–Seidel method that is not included in the present analysis.

5 Numerical examples

To confirm the validity and accuracy of the improved predictor/multi-corrector method discussed so far, two problems are presented based on the TDG formulation (P1-P1 elements) using the Gauss–Seidel iterative techniques. The problems presented in this study are dynamic structural problems, but the problem and the formulation are equally valid for any other dynamic problem governed by second-order ordinary differential equations. The accuracy of the present method is assessed by comparing the computed values with the equivalent analytical values (whenever possible) and other commonly used algorithms. It is also shown that the improved method for the present analysis has potential advantage over the method using the Gauss–Jacobi iteration (Li and Wiberg, 1996).

Example 1. The first problem in this example concerns the ten-story shear building depicted in Fig. 2. The base of the building is subjected to El Centro ground motion shown in Fig. 2 that is much more irregular than a base-acceleration impulse in the literature (Li and Wiberg, 1996). Due to the restriction on the sampling time step 0.02 s for the earthquake acceleration, the time step $\Delta t = 0.01$ s is used to iteratively calculate the responses. Comparing with the Li algorithm and improved algorithm, the total number of iterations needed for the computed responses are 33674 and 13999, respectively, shown in Fig. 3a, b. It is seen clearly that the improved algorithm is more efficient than the Li algorithm. However, based on these two different solution algorithms, the results with same accuracy for the velocity and displacement responses at the tenth floor are plotted in Fig. 4a, b, respectively.

Example 2. Consider a one-dimensional straight bar subjected to an end load P of a step time variation, as depicted in Fig. 5. The numerical values used in this problem are $P = 10000$ N and three different time steps Δt, with the material properties Young's modulus $E = 10^7$ kPa and mass density $\rho = 1000$ kg/m^3. The spatial domain is modeled by 250 linear finite elements. Our purpose is to observe the stress behaviors at the fixed end of the bar within the time 0.2 s. In the present analysis, the

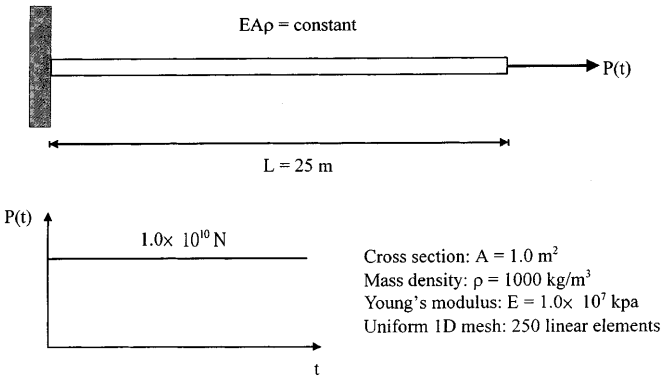

Fig. 5. A uniform bar subjected to a step time of impulse

responses are evaluated using three different time steps $\Delta t = 0.0004$, 0.0002 and 0.0001 s and hence the corresponding numbers of time steps 500, 1000 and 2000 are required, respectively. The computed time histories of the displacement and velocity can be compared to other step-by-step integration methods, showing good agreement. Due to the amount of data, those results are not presented here. However, the time histories for the stress response at the fixed end of the bar are plotted from Figs. 6–9. Here, the proposed TDG method is presented, and compared with available analytical solutions and three commonly used step-by-step integration methods, i.e., the HHT-α

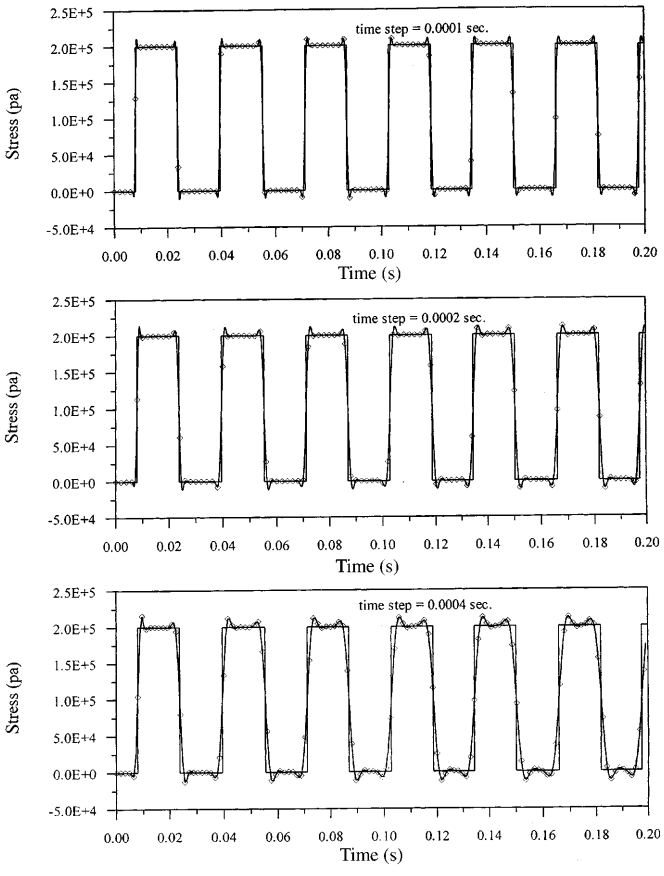

Fig. 6. Time history of stress at the fixed end of a uniform bar obtained by the TDG method

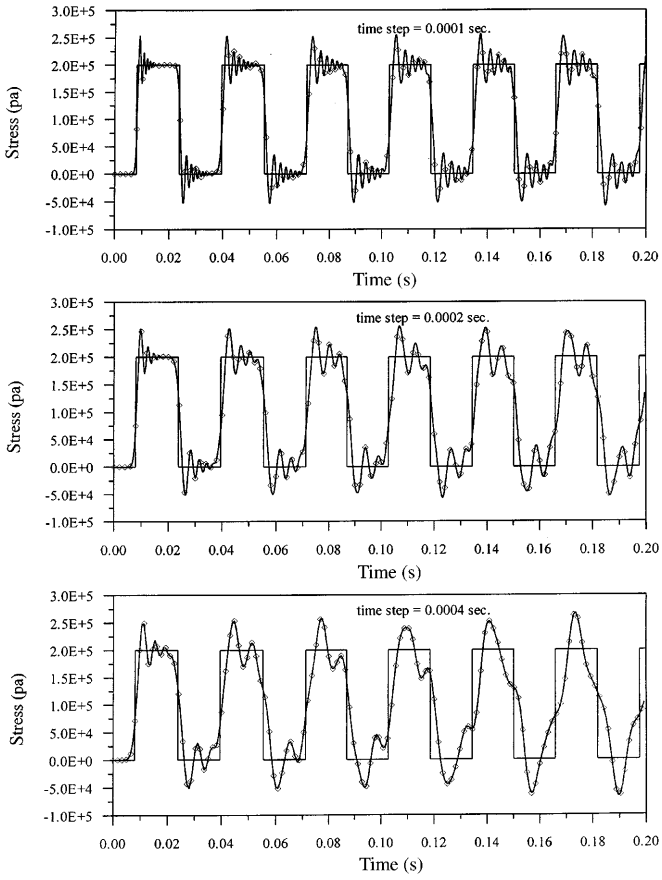

Fig. 7. Time history of stress at the fixed end of a uniform bar obtained by the HHT-α method

($\alpha = -0.1$, $\beta = 0.3025$, $\gamma = 0.5$), the Houbolt method and the Park method. It is quite obvious that the TDG results obtained using P1-P1 element are excellent among the methods. The comparison of required CPU time for computation is plotted in Fig. 10. Although the computation effort needed within each time step of the TDG method is higher than the commonly used methods, many advantages of the TDG method such as longer time steps, third-order accuracy and the possibility to filter out the spurious partitions of higher modes can be demonstrated. But for the time-step-time integration methods, one changes the time steps or even adjusts the time integration parameters, no accurate results can be obtained. In Fig. 6 it indicates that a very accurate solution can be obtained by the TDG method either for the Li or improved algorithm. However, when using the time step $\Delta t = 0.0001$ s, the number of the predictor/multi-corrector iterations for the Li algorithm with a requirement of $\varepsilon = 10^{-6}$ is 5 or 6 and the number of iterations using the improved algorithm is only 2 or 3. Further, when the time step Δt is changed to 0.0004 s, the number of iterations increases to $7 \sim 9$ for the Li algorithm and only 5 or 6 times is needed for the improved algorithm. Notably, a better initial estimate of the velocity for the improved method can reduce the number of iterations, especially as longer time step is adopted. Compared with the Li method by the Gauss–Jacobi algorithm, an improved procedure of the proposed method has better stability properties in Fig. 1 and, at the

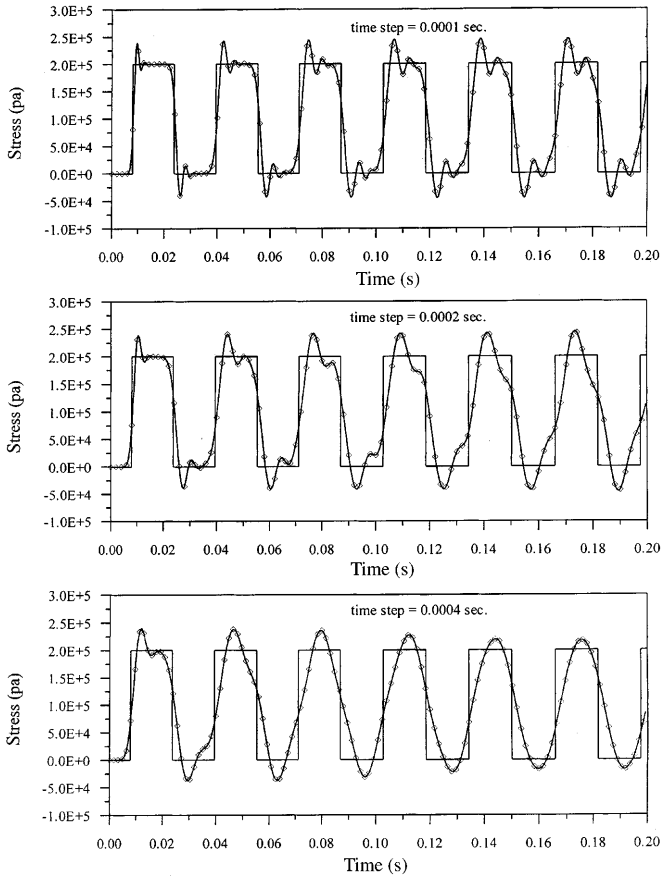

Fig. 8. Time history of stress at the fixed end of a uniform bar obtained by the Houbolt method

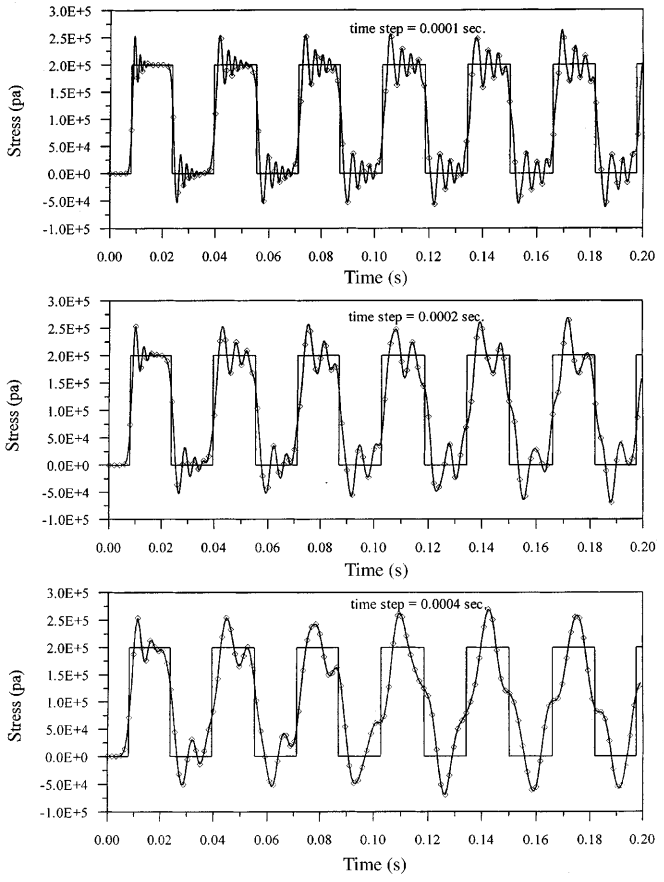

Fig. 9. Time history of stress at the fixed end of a uniform bar obtained by the Park method

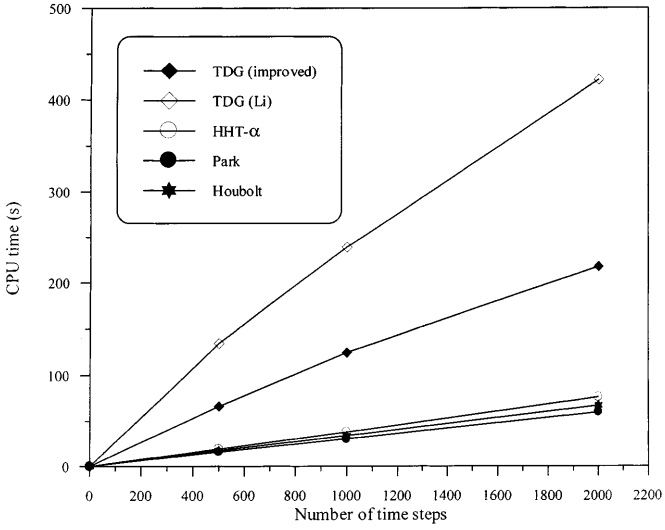

Fig. 10. Comparison of CPU time (133 MHz CPU) needed for five different methods for Example 2

same time, can achieve more efficient numerical computation, as seen in Fig. 10.

6 Conclusions

An improved predictor/multi-corrector algorithm using the Gauss–Seidel block iteration technique is presented. Based on the time discontinuous Galerkin finite element method, this method is used for the analysis of structural dynamic problems. The proposed method is shown analytically to be unconditionally stable regardless of the time step, and its third-order accurate when P1-P1 element is used. By comparing the results with those obtained by the traditional methods based on the finite difference formulas, the accuracy, stability and efficiency is established. Although the computational effort of the TDG method within each time step is considerably higher, this is compensated by the use of a larger time step size and its accuracy and, thus, making the total cost comparable. The present method can be extended to the implementation of other higher-order TDG finite elements or three-dimensional structural dynamic problems.

References

Argyris J, Scharpf DW (1969) Finite elements in time and space. Nucl. Eng. Design. 10: 456–464

Asaithambi NS (1995) Numerical Analysis, Theory and Practice, Saunders College Publishing

Borri M, Bottasso C (1993) A general framework for interpreting time finite element formulations. Comp. Mech. 13: 133–142

Borri M, Mello F, Atluri SN (1990) Time finite element methods for large rotational dynamics of multibody systems. Comput. Struct. 37: 231–240

Borri M, Mello F, Atluri SN (1990) Variational approaches for dynamics and time-finite-elements: numerical studies. Comp. Mech. 7: 49–76

Fried I (1969) Finite element analysis of time-dependent phenomena. AIAA J. 7: 1170–1173
Hilber HM, Hughes TJR, Taylor RL (1977) Improved numerical dissipation for time integration algorithms in structural dynamics. Earthquake Eng. Struct. Dyn. 5: 283–292
Houbolt JC (1950) A recurrence matrix solution for the dynamic response of elastic aircraft. J. Aeronaut. Sci. 17: 540–550
Hughes TJR, Hulbert G (1988) Space-time finite element methods for elastodynamics: formulation and error estimates. Comput. Meth. Appl. Mech. Eng. 66: 393–363
Hulbert G (1989) Space-time finite element methods for second order Hyperbolic equations, Ph.D. thesis, Dept. of Mechanical Engineering, Stanford University, Stanford
Hulbert G (1992) Time finite element methods for structural dynamics. Int. J. Num. Meth. Eng. 33: 307–331
Johnson C, Nävert U, Pitkaranta J (1984) Finite element methods for linear hyperbolic problem. Comput. Meth. Appl. Mech. Eng. 45: 285–312
Johnson C (1987) Numerical Solutions of Partial Differential Equations by the Finite Element Method. Cambridge Univ. Press, Cambridge
Johnson C (1993) Discontinuous Galerkin finite element methods for second order hyperbolic problem. Comput. Meth. Appl. Mech. Eng. 107: 117–129
Li XD, Wiberg N-E (1996) Structural dynamic analysis by a time-discontinuous Galerkin finite element method. Int. J. Num. Meth. Eng. 39: 2131–2152
Li XD, Wiberg N-E (1998) Implementation and adaptivity of a space-time finite element method. Comput. Meth. Appl. Mech. Eng. 156: 211–229
Newmark NM (1959) A method of computation for structural dynamics. J. Eng. Mech. Div. ASCE. 8: 67–94
Oden JT (1969) A general theory of finite elements II, Applications. Int. J. Num. Meth. Eng. 1: 247–259
Park KC (1975) Evaluating time integration methods for nonlinear dynamic analysis. In: Belytschko T, et al., (eds) Finite Element Analysis of Transient Nonlinear Behavior, AMD 14. New York: ASME, pp. 35–58
Thomée V (1984) Galerkin Finite Element Methods for Parabolic Problems. Springer-Verlag, New York
Wilson EL, Farhoomand I, Bathe KJ (1973) Nonlinear dynamic analysis of complex structures. Earthquake Eng. Struct. Dyn. 1: 242–252
Zienkiewicz OC, Taylor RL (1991) The Finite Element Method, Solid and Fluid Mechanics, Dynamics and Nonlinearity, McGraw-Hill