An Improved Time Integration Method Based on Galerkin Weak Form with Controllable Numerical Dissipation

Weixuan Wang, Qinyan Xing * and Qinghao Yang

Department of Civil Engineering, Tsinghua University, Beijing 100084, China; Wang-Weixuan@outlook.com (W.W); renjay@163.com (Q.Y.)
* Correspondence: xingqy@tsinghua.edu.cn

Abstract: Based on the newly proposed generalized Galerkin weak form (GGW) method, a two-step time integration method with controllable numerical dissipation is presented. In the first sub-step, the GGW method is used, and in the second sub-step, a new parameter is introduced by using the idea of a trapezoidal integral. According to the numerical analysis, it can be concluded that this method is unconditionally stable and its numerical damping is controllable with the change in introduced parameters. Compared with the GGW method, this two-step scheme avoids the fast numerical dissipation in a low-frequency range. To highlight the performance of the proposed method, some numerical problems are presented and illustrated which show that this method possesses superior accuracy, stability and efficiency compared with conventional trapezoidal rule, the Wilson method, and the Bathe method. High accuracy in a low-frequency range and controllable numerical dissipation in a high-frequency range are both the merits of the method.

Keywords: nonlinear dynamics; time integration; Galerkin method; numerical dissipation control

1. Introduction

To solve general structural dynamic problems, the direct integration method has been proven to be effective and has been widely used in engineering. The commonly used integration methods can be categorized into the explicit one and the implicit one. The explicit method is conditionally stable, and a smaller time step should be used in the calculation. Thus, for structural vibration analysis, the implicit method is usually considered to be more effective [1]. As for implicit method, it can be designed to be unconditionally stable and to possess controllable numerical dissipation. Although the calculation needs matrix factorization and is cost-consuming [2], the implicit method can allow a larger time step due to its superior numerical stability, which brings much more efficiency in solving dynamic problems [3]. At present, there are some well-known implicit algorithms, such as the Newmark method [4], the Wilson-θ method [5], the generalized-α method [6], the HHT method [7], and the Bathe method [8], etc.

However, some algorithms may lead to instability in the nonlinear range [9] due to non-conservation of energy and momentum [10]. Taking trapezoidal rule for example, in the linear range, it is unconditionally stable, and has second-order accuracy without amplitude decay [11,12]. However, this method can easily become unstable when extended to the nonlinear problems [12,13], and the response will diverge [3,10,14] when using an inappropriate length of time steps. To solve this problem, numerical damping is introduced into the Newmark method by adjusting the parameters, but it results in first-order accuracy [15].

For most of the structural dynamic problems, the main contribution of the response comes from its low-order frequency modes [16]. Considering the impact of discretization, the high-order frequency modes of the structure will change, which affects the computational accuracy. Therefore, the numerical damping which can control numerical dissipation is necessary for both low-order and high-order frequency ranges. In addition, it can also
improve the convergence of the iterative solution and increase the numerical stability. Nevertheless, unreasonable numerical damping will result in a great loss of accuracy, which leads to a large attenuation of structural response. It is usually caused by excessive numerical dissipation in a low-frequency range. To evaluate the numerical dissipation of the algorithm, the spectral radius $\rho$ is used. The change in dissipation is reflected in the spectral radius $\rho$; that is, the reduction in spectral radius illustrates the dissipation of the algorithm. When the time step is large, the spectral radius will quickly attenuate for some algorithms, such as part of the Newmark method, Houbolt method, Wilson-$\theta$ method, and part of the generalized $\alpha$ method, etc. [17]

In summary, it is of necessity to find a time integration algorithm that has both reasonable and controllable numerical damping. This algorithm can generate good accuracy in a low-frequency range with controlled high dissipation in a high-frequency range [7,18,19], and is expected to perform well in nonlinear field.

Many researchers have proposed useful methods to control the accuracy of algorithms in different frequency ranges. Some composite algorithms with several steps perform quite well [20–22]. A typical one is the Bathe method proposed by Bathe K.J. et al., which divides a time step into two sub-steps to combine two well-known time integration schemes [3,8,12,23]. The trapezoidal integration method and the three-point Euler backward method are respectively used in the first and second sub-step. The Bathe method has reasonable numerical damping and stability but cannot control the dissipation [24]. Based on this, the $\rho\theta$-Bathe method and $\beta_1/\beta_2$-Bathe method were proposed [23,25]. According to the evaluation, these methods possess effective numerical dissipation, and the $\beta_1/\beta_2$-Bathe method is the special case of the $\rho\theta$-Bathe method [26]. Compared with the generalized-$\alpha$ method, this series of Bathe methods show much better performance [24]. Some multi-step algorithms are further proposed based on the methods mentioned above [27–30]. They are usually composed of a non-dissipative algorithm, e.g., trapezoidal rule, and a strongly dissipative one [29].

Focusing on the nonlinear problems, a time integration method based on Galerkin weak form, i.e., the Galerkin weak form (GGW) method is proposed [31], considering weighted residuals of equations [32]. The Newton–Raphson iteration method is used first so that the iterative linear procedure can be constructed. To solve the resulting linear equations, a time-domain finite element method (FEM) based on Galerkin weak form is established [33–35], in which the Gaussian numerical integration is used to solve the integral term. For different degrees $m$ ($m=1,2,3$) of the time elements, the implicit time integration schemes are derived. At the same time, the numerical damping is controlled by modifying the numerical integration form. According to the analysis, the GGW method is an unconditionally stable, self-starting and one-step method. Compared with the Newmark method, Wilson-$\theta$ method, and Bathe method, the GGW method has better stability and efficiency. However, for some representative nonlinear problems, it is found that in the GGW method, the numerical dissipation may decay too fast and the speed of the dissipation cannot be controlled in some cases. Hence, the GGW method is expected to be improved to possess a more effective numerical dissipation performance.

Based on the deep study of the GGW method and inspired by the idea of the Bathe method, an improved two-step time integration method is proposed hereinafter. The time step of this method is divided into two sub-steps. In the first sub-step, the GGW method is used, and in the second sub-step, another parameter is introduced by the trapezoidal integral, i.e., a non-dissipative algorithm is combined with the GGW method so that the improved algorithm can solve the defects of the GGW method but retaining its excellent characteristics. After that, the corresponding iterative procedure is established and presented, and then the accuracy and numerical stability of the algorithm are discussed. Typical numerical examples including an undamped single-degree-of-freedom (single-DOF) system, the nonlinear pendulum, and a large-span plane skeletal system are solved, through which it can be seen that the numerical dissipation is controllable in the proposed two-step time integration method and sensitivity of parameters is reduced. It dramatically
improves the performance of the GGW method. Compared with other algorithms, such as the trapezoid rule, the Wilson-β method, and the Bathe method, the results sufficiently show the high accuracy and computational efficiency of the proposed method.

2. Basic Principle

2.1. Governing Equations and Iterative Schemes

The nonlinear dynamic problems can be usually attributed to the following nonlinear second-order ordinary differential equations (ODEs), which are named dynamic equations hereinafter:

\[
\begin{aligned}
\{ \mathbf{M} \ddot{\mathbf{u}} + \mathbf{C}(\dot{\mathbf{u}}) + \mathbf{K}(\mathbf{u}) - \mathbf{f}_\text{ext}(t) = 0, & \quad 0 < t \leq t_f \\
\mathbf{u}(0) = \mathbf{u}_0, & \quad \mathbf{u}(0) = \dot{\mathbf{u}}_0
\end{aligned}
\]

(1)

where \( \mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}} \) are the nodal displacement vector, velocity vector, and acceleration vector respectively, \( \mathbf{M} \) is the mass matrix, \( \mathbf{f}_d \) is the damping force vector, \( \mathbf{s} \) is the resilience vector, \( \mathbf{f}_\text{ext} \) is the vector of externally applied forces, and \( \mathbf{u}_0, \dot{\mathbf{u}}_0 \) are the known displacement and velocity vectors at the initial time, respectively. The time domain is \( (0, t_f) \).

Regarding the time interval \([t_n, t_{n+1}]\), when \( \mathbf{f}_d \) and \( \mathbf{s} \) are the nonlinear functions related to \( \mathbf{u} \) and \( \dot{\mathbf{u}} \), the Newton–Raphson method can be used to transform the nonlinear second-order ODEs (Equation (1)) into linear equations as follows:

\[
\begin{aligned}
\{ \mathbf{M}(k+1) \ddot{\mathbf{u}}^{(k+1)} + \mathbf{C}^{(k)} \dot{\mathbf{u}}^{(k+1)} + \mathbf{K}^{(k)} \mathbf{u}^{(k+1)} = \mathbf{f}_\text{ext} - \mathbf{f}_\text{int}^{(k)} = \mathbf{f} \\
\mathbf{u}^{(k+1)}(t_n) = \mathbf{u}_n, & \quad \mathbf{u}^{(k+1)}(t_{n+1}) = \dot{\mathbf{u}}_n
\end{aligned}
\]

(2)

where

\[
\mathbf{C}^{(k)} = \left. \frac{\partial \mathbf{f}_d}{\partial \mathbf{u}} \right|_{\mathbf{u}^{(k)}} \quad \mathbf{K}^{(k)} = \left. \frac{\partial \mathbf{s}}{\partial \mathbf{u}} \right|_{\mathbf{u}^{(k)}}
\]

(3)

\[
\mathbf{f}_\text{int}^{(k)} = \left( \mathbf{f}_d^{(k)} \right|_{\mathbf{u}^{(k)}} - \mathbf{C}^{(k)} \dot{\mathbf{u}}^{(k)}(k) + \left( \mathbf{s}^{(k)} \right|_{\mathbf{u}^{(k)}} - \mathbf{K}^{(k)} \mathbf{u}^{(k)}(k)
\]

(4)

In Equations (2)–(4) \( \mathbf{u}^{(k)}, \dot{\mathbf{u}}^{(k)}, \text{ and } \ddot{\mathbf{u}}^{(k)} \) are solutions of the independent variable time \( t \) obtained in the \( k \)-th step of iteration, where the tangent damping \( \mathbf{C}^{(k)} \), the tangent stiffness \( \mathbf{K}^{(k)} \), and the internal force \( \mathbf{f}_\text{int}^{(k)} \) are all functions of time \( t \). Therefore, through Newton’s method, the process of solving nonlinear Equation (1) is transformed into an iterative solution of linear Equation (2). For convenience, the superscript \( k \) standing for the \( k \)-th iteration step is omitted in Equation (2), and Equation (2) is rewritten as follows:

\[
\begin{aligned}
\{ \mathbf{M} \ddot{\mathbf{u}} + \mathbf{C} \dot{\mathbf{u}} + \mathbf{K} \mathbf{u} = \mathbf{f} \\
\mathbf{u}(t_n) = \mathbf{u}_n, & \quad \dot{\mathbf{u}}(t_n) = \dot{\mathbf{u}}_n
\end{aligned}
\]

(5)

2.2. Galerkin Finite Element Method

Many methods can be used to solve Equation (5), and Ref. [31] uses the time FEM based on the Galerkin weak form to solve Equation (5). Setting \( \mathbf{v} \in H^1_2 \) to be the test function and \( \mathbf{u}^h \in H^1_2 \) to be the trial function, the weighted residual weak form of Equation (5) in \([t_n, t_{n+1}]\) is established as follows:

\[
\int_{t_n}^{t_{n+1}} \mathbf{v}^T (\mathbf{M} \ddot{\mathbf{u}}^h + \mathbf{C} \dot{\mathbf{u}}^h + \mathbf{K} \mathbf{u}^h) \, dt = \int_{t_n}^{t_{n+1}} \mathbf{v}^T \mathbf{f} \, dt
\]

(6)

where \( H^1_2 \) and \( H^1_2 \) are the trial function space and the test function space, respectively. Regarding the time interval \([t_n, t_{n+1}]\) as a single time element of length \( \Delta t = t_{n+1} - t_n \), the test function and trial function can be expressed with Equation (7), in which the
time interval \([t_n, t_{n+1}]\) is divided into \(m\) \((m = 1, 2, 3, \ldots)\) intervals so that the \(m\)-degree interpolation polynomial is used.

\[
u^h = N\mathbf{d} = \sum_{i=1}^{m+1} N_i(t)d_i, \quad \mathbf{v}^h = N\mathbf{v} = \sum_{i=1}^{m+1} N_i(t)v_i
\]  

(7)

where \(N = [N_1(t)I, N_2(t)I, \ldots, N_{m+1}(t)I]\) is the shape function matrix and \(m\) is the degree of time elements. \(N_i(t)\) \((i = 1, 2, \ldots, m+1)\) is the shape function of Lagrange polynomial form and \(I\) is the unit matrix. \(\mathbf{d} = [d_1^T, d_2^T, \ldots, d_{m+1}^T]^T\) is the matrix of nodal displacement vectors in time interval \([t_n, t_{n+1}]\) and \(\mathbf{v} = [v_1^T, v_2^T, \ldots, v_{m+1}^T]^T\) is the matrix of nodal test displacement vectors.

Substituting Equation (7) into Equation (6) and implementing partial integration, the following equation can be derived by coordinate mapping, in which time \(t \in [t_n, t_{n+1}]\) is converted to \(\zeta \in [-1, 1]\).

\[
\left(-\frac{4}{\Delta t^2}m\mathbf{K} + \frac{2}{\Delta t}K + k\mathbf{K}\right)\mathbf{d} + \frac{2M}{\Delta t} (N(1)\mathbf{u}_{n+1} - N(-1)\mathbf{u}_n) = \mathbf{P}
\]  

(8)

where

\[
m\mathbf{K} = \int_{-1}^{1} \mathbf{N}^T \mathbf{M} \mathbf{N} d\zeta, \quad k\mathbf{K} = \int_{-1}^{1} \mathbf{N}^T \mathbf{C}(\zeta) \mathbf{N} d\zeta
\]  

(9)

\[
P = \int_{-1}^{1} \mathbf{N}^T \mathbf{f} d\zeta
\]  

(10)

From Equation (8), for any positive integer \(m\), we can derive and obtain a time integration scheme. In fact, considering both the calculation efficiency and accuracy, letting \(m = 1\) or \(2\) or \(3\) is sufficient [31]. At the same time, it can be proved that when reduced Gaussian integration is used, the derived scheme is unconditionally stable for linear problems [33]. When \(m = 1\) and using the Gaussian integration, the time integration scheme from Equation (8) can be presented as

\[
\begin{bmatrix}
K_{11}^{[k]} & 0 \\
K_{21}^{[k]} & \frac{2M}{\Delta t}
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}_{n+1}^{(k+1)} \\
\mathbf{u}_{n+1}^{(k+1)}
\end{bmatrix} =
\begin{bmatrix}
P_1^{[k]} - K_{11}^{[k]} \mathbf{u}_n + 2M\mathbf{u}_n/\Delta t \\
P_2^{[k]} - K_{21}^{[k]} \mathbf{u}_n
\end{bmatrix}
\]  

(11)

where

\[
K_{ij} = -\frac{4}{\Delta t^2} \sum_{l=1}^{n_i} \omega_l N_j(\zeta_l) M N_j(\zeta_l) + \frac{4}{\Delta t} \sum_{l=1}^{n_i} \omega_l N_j(\zeta_l) C(\zeta_l) N_j(\zeta_l)
\]  

(12)

\[
P_j = \sum_{l=1}^{n_i} \omega_l N_j(\zeta_l) f(\zeta_l, \mathbf{u})
\]  

(13)

where \(\omega_l, \zeta_l\) are the Gaussian integral coefficients and locations, respectively, and \(n_i\) is the number of Gaussian points. Finally, when the reduced Gaussian integration with one Gaussian integration point is used, it can be simplified to the following equations:

\[
K_{11} = -\frac{2}{\Delta t^2} \mathbf{M} + \frac{1}{\Delta t} \mathbf{C}(\mathbf{u}) + \frac{1}{4} \mathbf{K}(\mathbf{u}) \quad K_{12} = \frac{2}{\Delta t^2} \mathbf{M} - \frac{1}{\Delta t} \mathbf{C}(\mathbf{u}) + \frac{1}{4} \mathbf{K}(\mathbf{u})
\]  

(14)

\[
K_{21} = \frac{2}{\Delta t^2} \mathbf{M} + \frac{1}{\Delta t} \mathbf{C}(\mathbf{u}) + \frac{1}{4} \mathbf{K}(\mathbf{u}) \quad K_{22} = -\frac{2}{\Delta t^2} \mathbf{M} - \frac{1}{\Delta t} \mathbf{C}(\mathbf{u}) + \frac{1}{4} \mathbf{K}(\mathbf{u})
\]  

(15)

In summary, the nonlinear dynamic equation, Equation (1), is converted into the linear dynamic equation, Equation (5), with Newton-Raphson iteration, and then the Galerkin FEM is used to solve this ODE. This is the basic idea and the key strategy of the GGM method. However, the numerical damping is not introduced until Equations (14) and (15).
2.3. Improvement of Algorithm

In order to introduce numerical dissipation into the GGW method, Ref. [31] proposes a feasible time integration scheme. A parameter \( \alpha \) is introduced to control the numerical damping by changing the form of the test function as Equation (16). At the same time, to calculate the integral terms in Equations (12) and (13), a specific numerical integration rule as Equation (17) is constructed using Equation (16).

\[
\tilde{N}_1 = -\frac{(\zeta - 1 - \alpha)}{2}, \quad \tilde{N}_2 = \frac{(\zeta + 1 - \alpha)}{2}
\]

(16)

\[
\int_{-1}^{1} f(\zeta) d\zeta = 2f(\frac{\alpha}{2})
\]

(17)

In this way, the time integration scheme with numerical dissipation for linear time elements is established based on the Galerkin weak form. However, the parameter \( \alpha \) in Equations (16) and (17) turns out to be too sensitive under certain circumstances, which will result in too fast decay (this is demonstrated in Section 4.3).

To overcome this shortcoming of the GGW method, here we present an improved method in which the time step is divided into two subsections and another parameter \( \beta (0 < \beta < 1) \) is introduced in addition to \( \alpha \).

In the first step, we take the step length to be \( \beta \Delta t \), and by using the GGW method, we obtain the solutions \( u_{n+\beta} \), \( u_{n+\beta} \) and \( u_{n+1} \) at \( t_n + \beta \Delta t \). In the second step, we use \( u_{n+\beta} \), \( \tilde{u}_{n+\beta} \) and \( \tilde{u}_{n+1} \) to compute the solutions \( u_{n+1} \), \( \tilde{u}_{n+1} \) at \( t_{n+1} \) \( (t_{n+1} = t_n + \Delta t) \). The detailed computational strategy of the second sub-step is as follows:

Considering the definition of integral as Equation (18) shown as follows:

\[
\begin{cases}
  u_{n+1} - u_n & = \int_{t_n}^{t_{n+1}} \dot{u}(t) dt = \int_{t_n}^{t_{n+1}} \dot{u}(t) dt + \int_{t_n}^{t_{n+1}} \dot{u}(t) dt \\
  \tilde{u}_{n+1} - \tilde{u}_n & = \int_{t_n}^{t_{n+1}} \tilde{u}(t) dt = \int_{t_n}^{t_{n+1}} \tilde{u}(t) dt + \int_{t_n}^{t_{n+1}} \tilde{u}(t) dt
\end{cases}
\]

(18)

and using the algorithm of trapezoidal integral, Equation (18) can be rewritten into Equation (19).

\[
\begin{cases}
  u_{n+1} - u_n & = \Delta t \left[ (\tilde{u}_n + \tilde{u}_{n+1}) \beta + (\tilde{u}_{n+1} + \tilde{u}_{n+1}) (1 - \beta) \right] \\
  \tilde{u}_{n+1} - \tilde{u}_n & = \Delta t \left[ (\tilde{u}_n + \tilde{u}_{n+1}) \beta + (\tilde{u}_{n+1} + \tilde{u}_{n+1}) (1 - \beta) \right]
\end{cases}
\]

(19)

where \( \tilde{u}_{n+\beta} \) and \( \tilde{u}_{n+1} \) are solutions solved from the first sub-step. Furthermore, Equation (19) can be simplified into the following:

\[
\begin{align*}
  u_{n+1} & = u_n + (c_1 \tilde{u}_n + c_2 \tilde{u}_{n+1} + c_3 \tilde{u}_{n+1}) \Delta t \\
  \tilde{u}_{n+1} & = \tilde{u}_n + (c_1 \tilde{u}_n + c_2 \tilde{u}_{n+1} + c_3 \tilde{u}_{n+1}) \Delta t
\end{align*}
\]

(20)

where \( c_1, c_2 \) and \( c_3 \) are parameters, and \( c_1 = \frac{\beta}{2}, c_2 = \frac{1}{2}, c_3 = \frac{1 - \beta}{2} \). Generally, when \( \beta = 0.5 \), \( c_1 = c_3 = 1/4, c_2 = 1/2 \). In addition. \( c_1 + c_2 + c_3 = 1 \).

It is obvious that, when \( \beta = 1 \), the second sub-step vanishes and the proposed method degenerates to the GGW method; when \( \beta = 0 \), the first sub-step vanishes and the proposed method degenerates to the Newmark method.
The iterative algorithm of the proposed method is finally summarized into the following steps (Algorithm 1).

**Algorithm 1.** The proposed two sub-step method based on Galerkin Finite Element Method (FEM)

1 **Initial condition**

1.1 Determine the initial conditions: \(u_0, \dot{u}_0, \ddot{u}_0\);

1.2 Determine the time step \(\Delta t\), total time \(t_f\) and error tolerance \(Tol_{iter}\).

1.3 Determine parameters in the algorithm: \(\beta, \alpha, \) and calculate \(c_1, c_2, c_3\).

2 **Aim** Solve the equation at \(t_n + \Delta t\) when knowing the result at \(t_n\).

2 // — **First sub-step:** solve the dynamic response \(u_{n+\beta}, \dot{u}_{n+\beta}, \ddot{u}_{n+\beta}\) at \((t_n + \beta \Delta t)\)

Use the step-by-step time integration algorithm based on the GGW method:

\[
\begin{bmatrix}
K_{11} & K_{12} \\
K_{21} & K_{22}
\end{bmatrix}
\begin{bmatrix}
u_n \\
v_{n+\beta}
\end{bmatrix}
+ \frac{2M}{\Delta t^2}
\begin{bmatrix}
-u_n \\
v_{n+\beta}
\end{bmatrix}
= \begin{bmatrix}
P_1 \\
P_2
\end{bmatrix}
\]

2.1 Initial iterative step by setting: \(u^{(1)}_{n+\beta} = u_n, \quad \dot{u}^{(1)}_{n+\beta} = \dot{u}_n, \quad \ddot{u}^{(1)}_{n+\beta} = \ddot{u}_n\).

Let \(k\) be the number of iteration steps, starting from 1 to meet the requirements.

2.2 Since \(u^{(k)}_{n+\beta}, \dot{u}^{(k)}_{n+\beta}\) and \(\ddot{u}^{(k)}_{n+\beta}\) are known, the \(k+1\)th step of iteration is performed:

2.3 Determine the parameters:

\[
\begin{align*}
\alpha &= \frac{1}{2}(1 - \beta)u_n + \frac{1}{2}(1 + \beta)u^{(k)}_{n+\beta} + \dot{u}^{(k)}_{n+\beta} = -\frac{1}{2}u_n + \frac{1}{2}u^{(k)}_{n+\beta} \\
\beta &= \frac{1}{2}d, \quad \dot{\beta} = \frac{1}{2}d, \quad \ddot{\beta} = \frac{1}{2}d
\end{align*}
\]

2.4 Solve the \(u^{(k+1)}_{n+\beta}, \dot{u}^{(k+1)}_{n+\beta}, \ddot{u}^{(k+1)}_{n+\beta}\) using the following equation:

\[
\begin{align*}
-\frac{2}{\Delta t^2}M - \frac{1}{2}d(\ddot{c}^{(k)}_{n+1}) + \frac{1}{2}(1 - \frac{\beta^2}{4})\ddot{K}^{(k)}_{n+2} u^{(k+1)}_{n+1} &= (1 + \frac{\beta}{2})f^{(k)}_{n+1} \\
+ \frac{2}{\Delta t^2}M + \frac{1}{2}d(1 + \frac{\beta}{2})\ddot{C}^{(k)}_{n+1} - \frac{1}{2}(1 - \frac{\beta^2}{4})\dddot{K}^{(k)}_{n+1} u_n &= 2M M^{(k)}_n \\
\left(\frac{2}{\Delta t^2}M + \frac{1}{2}d(1 - \frac{\beta}{2})\ddot{C}^{(k)}_{n+1} + \frac{1}{2}(1 - \frac{\beta^2}{4})\dddot{K}^{(k)}_{n+1} + \frac{2M}{\Delta t}u^{(k)}_{n+1} = \right)
\end{align*}
\]

2.5 If \(|u^{(k+1)}_{n+\beta} - u^{(k)}_{n+\beta}| > Tol_{iter}\) then \(k = k+1\) and return to 2.2, otherwise go to the next step.

3 // — **Second sub-step:** solve the dynamic response \(u_{n+1}, \dot{u}_{n+1}, \ddot{u}_{n+1}\) with \(u_n, \dot{u}_n, \ddot{u}_n\) at \((t_n + \beta \Delta t)\) already obtained

\[
\begin{bmatrix}
u_{n+1} \\
v_{n+1}
\end{bmatrix}
= \begin{bmatrix}
(1 + c_1 \Delta t + c_2 \Delta t + c_3 \Delta t^2) u_n + c_1 \dot{u}_n + c_2 \ddot{u}_n + c_3 \dddot{u}_n + c_4 \ddot{u}_n + c_5 \dddot{u}_n \Delta t \\
(1 + c_1 \Delta t + c_2 \Delta t + c_3 \Delta t^2) \dot{u}_n + c_1 \ddot{u}_n + c_2 \dddot{u}_n + c_3 \ddot{u}_n + c_4 \dddot{u}_n + c_5 \dddot{u}_n \Delta t
\end{bmatrix}
\]

3.1 Initial iterative step: \(u^{(1)}_{n+1} = u_n, \quad \dot{u}^{(1)}_{n+1} = \dot{u}_n, \quad \ddot{u}^{(1)}_{n+1} = \ddot{u}_n;\)

\(k\) is the number of iterations, starting from 1 to meet the requirements.

3.2 When \(u^{(k)}_{n+1}, \dot{u}^{(k)}_{n+1}\) and \(\ddot{u}^{(k)}_{n+1}\) are known, the \((k+1)\)th step of iteration is performed:

3.3 Determine the parameters:

\[
\begin{align*}
\dddot{C}^{(k)}_{n+1} &= C^{(k)}_{n+1}, \quad \dddot{K}^{(k)}_{n+1} &= K^{(k)}_{n+1}, \quad \dddot{K}^{(k)}_{n+1} = \frac{M}{\Delta t} + c^{(k)}_{n+1}, \\
\dddot{f} &= f^{ext} - \dddot{f}^{(k)}_{n+1} - f^{(k)}_{n+1} \\
&= -\frac{M}{\Delta t}c_{n+1} - c_{n+1} \Delta t - c_{n+1} \Delta t^2 - c_{n+1} \Delta t^2
\end{align*}
\]

Solve the response by the following equations:

\[
K\dddot{u} = \dddot{f};
\]

3.4 Calculate \(u^{(k+1)}_{n+1}, \dot{u}^{(k+1)}_{n+1}\) and \(\ddot{u}^{(k+1)}_{n+1}:

\[
\begin{bmatrix}
u^{(k+1)}_{n+1} \\
v^{(k+1)}_{n+1} \\
v^{(k+1)}_{n+1}
\end{bmatrix}
= \begin{bmatrix}
n^{(k+1)}_{n+1} + \Delta u \\
n^{(k+1)}_{n+1} + \Delta u \\
n^{(k+1)}_{n+1} + \Delta u
\end{bmatrix}
\]

3.5 If \(|u^{(k+1)}_{n+1} - u^{(k)}_{n+1}| > Tol_{iter}\) then \(k = k+1\) and return to step 3.2, otherwise go to the next step.

3.6 Get the answers \(u_{n+1}, \dot{u}_{n+1}, \ddot{u}_{n+1}\).
3. Analysis of Numerical Stability and Accuracy

The following single-DOF system is considered to analyze the stability and accuracy of the proposed method:

\[ \ddot{u} + \omega^2 u = 0 \]  

(21)

where \( \omega \) is the circular frequency of free vibration. The relationship between the response of the \((n + 1)\)-th time step and the \(n\)-th time step can be obtained by referring to Equations (11) and (20), that is

\[
\begin{align*}
\frac{u_{n+1} - c_3 u_{n+1} \Delta t}{\Delta t} &= u_n + (c_1 u_n + c_2 u_{n+1}) \Delta t \\
\frac{u_{n+1} - c_3 u_{n+1} \Delta t}{\Delta t} &= u_n + (c_1 u_n + c_2 u_{n+1}) \Delta t \\
\frac{u_{n+1} + \omega^2 u_{n+1}}{2} &= 0
\end{align*}
\]  

(22)

which leads to

\[
\begin{bmatrix}
1 & -c_3 \Delta t \\
\frac{1}{c_3 \omega^2 \Delta t} & 1
\end{bmatrix}
\begin{bmatrix}
u_{n+1} \\
u_{n+1}
\end{bmatrix} = \begin{bmatrix}
1 & c_1 \Delta t \\
-c_1 \omega^2 \Delta t & 1
\end{bmatrix}
\begin{bmatrix}u_n \\
u_n
\end{bmatrix} + \begin{bmatrix}0 & c_2 \Delta t \\
-c_2 \omega^2 \Delta t & 0
\end{bmatrix}
\begin{bmatrix}u_{n+1} \\
u_{n+1}
\end{bmatrix}
\]  

(23)

As we know, the transfer matrix \( A \) in the following equation is generally used to analyze the stability of the method:

\[
\begin{bmatrix}u_{n+1} \\
u_{n+1}
\end{bmatrix} = A
\begin{bmatrix}u_n \\
u_n
\end{bmatrix}
\]  

(24)

Using \( \rho(A) \) to represent the spectral radius of \( A \), it is well-known that when \( \rho(A) \leq 1 \), the algorithm is stable. Referring to Equation (23), for the proposed two-step algorithm, the transfer matrix \( A \) can be written into the following formula:

\[
A = \begin{bmatrix}
1 & -c_3 \Delta t \\
\frac{1}{c_3 \omega^2 \Delta t} & 1
\end{bmatrix} \left( \begin{bmatrix}
1 & c_1 \Delta t \\
-c_1 \omega^2 \Delta t & 1
\end{bmatrix} + \begin{bmatrix}0 & c_2 \Delta t \\
-c_2 \omega^2 \Delta t & 0
\end{bmatrix} \right) \begin{bmatrix}
u_{n+1} \\
u_{n+1}
\end{bmatrix}
\]  

(25)

where \( B \) is the transfer matrix computed in the first sub-step, i.e., \( B \) satisfies

\[
\begin{bmatrix}u_{n+1} \\
u_{n+1}
\end{bmatrix} = B
\begin{bmatrix}u_n \\
u_n
\end{bmatrix}
\]  

(26)

As the transfer matrix of the GGW method, \( B \) was derived from Ref. [31]:

\[
B = \frac{1}{a^2 \Lambda^2 + 4a \Lambda^2 + 4 \Lambda^2 + 16}
\begin{bmatrix}
a^2 \Lambda^2 - 4 \Lambda^2 + 16 & 16 \beta \Delta t \\
-16 \Lambda^2 / \beta \Delta t & a^2 \Lambda^2 - 4 \Lambda^2 + 16
\end{bmatrix}
\]  

(27)

where \( \Lambda = \omega \beta \Delta t \), \( a \) is the parameter in the GGW method to control the numerical damping. It has already been concluded that the spectral radius \( \rho(B) \leq 1 \) [33,34] when using reduced Gaussian integration in Equation (8) and the GGW method is unconditionally stable [31]. By using Equations (25) and (27), the spectral radius of the present two-step algorithm is derived:

\[
\rho(A) = \frac{(4c_1 + 4c_2 - a \beta - 2\beta) \Omega i + (2c_4 \beta - 2c_2 \beta + ac_1 \beta + ac_2 \beta) \Omega^2 + 4}{(2 + a) \beta + 4c_3 \Omega i + (2c_3 \beta + ac_3 \beta) \Omega^2 - 4}
\]  

(28)

where \( \Omega = \omega \Delta t \). With \( T \) denoting the period of free vibration, when \( \Delta t / T \) tends to infinity, the spectral radius \( \rho(A) \) turns out to be

\[
\rho_\infty \equiv \rho(A)_{\Delta t \to \infty} = \frac{2c_1 - 2c_2 + ac_1 + ac_2}{2c_3 + ac_3}
\]  

(29)
Through Equation (29), the stability of the algorithm with different parameters can be predicted. Generally, we take $\beta = 1/2$ and then $c_1 = c_3 = 1/4, c_2 = 1/2$, which results in the following expression:

$$\rho_\infty \equiv \rho(A) \nrightarrow \infty = \frac{3\alpha - 2}{2 + \alpha}$$  

(30)

When $\rho_\infty$ both and $\beta$ are given, the corresponding parameter $\alpha$ can be computed from Equation (29). Generally, when $\Delta t/T$ tends to infinity, the suggested $\rho_\infty$ satisfies $\rho_\infty \in [0.5, 0.8]$. Therefore, according to the Equation (30), when $\beta = 1/2$, $\alpha \in [2/19, 2/7]$. In this case, the change in $\rho(A)$ relating to $\Delta t/T$ is plotted in Figure 1 with $\rho_\infty = 0, 0.4, 0.8$ and $\beta = 0.5, 0.65, 0.8$, respectively, which aims to evaluate the impact of $\alpha$ and $\beta$ on $\rho(A)$. Furthermore, to evaluate the errors of this method, the period elongation (PE) and amplitude decay (AD) are plotted in Figures 2 and 3 as well.

![Figure 1](image1.png)

**Figure 1.** The spectral radius of transfer matrix of the proposed method with different numerical damping.

![Figure 2](image2.png)

**Figure 2.** Percentage of period elongation of the proposed method with different numerical damping.
is a constant, the decay rate of this algorithm turns out to be larger when $\beta$ is smaller. As shown in Figures 2 and 3, the PE and AD are also related to $\rho_\infty$ and $\beta$. When $\rho_\infty$ maintains a constant value, the trends of PE and $\beta$ are positively correlated, while those of AD and $\beta$ are opposite. Moreover, when $\rho_\infty$ becomes larger, both AD and PE become smaller.

The comparison of the GGW method and the proposed method is shown in Figure 4. When $\Delta t/T$ is small (e.g., $\Delta t/T \approx 10^{-2}$ in Figure 4), the spectral radius of the GGW method starts to reduce, while the spectral radius of the proposed method starts to reduce when $\Delta t/T$ is larger (e.g., $\Delta t/T \approx 10^{-1}$ in Figure 4). This illustrates that the proposed method improved the disadvantage of too fast numerical dissipation in the GGW method.

![Figure 4](image_url)

**Figure 4.** The spectral radius of transfer matrix of the proposed method ($\beta = 1/2$) and the GGW method when $\rho_\infty = 0$.

4. Numerical Examples

In this section, three numerical examples are considered to demonstrate the accuracy and efficiency of the proposed two-step method. Section 4.1 gives an undamped standard single-DOF system, Section 4.2 gives an oscillating nonlinear single pendulum, and Section 4.3 gives a long span planar skeletal structure. These examples are solved with the Wilson method ($\theta = 1.4$), Bathe method, GGW method, and the trapezoidal rule as well in
order to give a brief analysis of the performance of the proposed method. All examples are programed in MATLAB 2020b under the same computing environment.

4.1. A Single-DOF System

In order to evaluate the accuracy of the proposed method, an undamped standard single-DOF system is firstly considered [19]. Its governing equation is:

\[ \ddot{u}(t) + \omega^2 u(t) = 0; u(0) = 1, \dot{u}(0) = 0 \]  

(31)

Letting \( \omega = \pi \), the exact solution can be written as:

\[ u(t) = \cos(\omega t) \]  

(32)

The free-vibration period \( T \) of this example is 2 s, the time step \( \Delta t = T/10 = 0.2 \) s, and the parameters are chosen to be \( \alpha = 0, \beta = 0.5, c_1 = c_3 = 1/4, c_2 = 1/2 \). The vibrational responses within 40–50 s of the proposed two-step method, the exact solution, and other time integration methods are shown in Figures 5–7. The computational time of these methods are listed in Table 1.

![Displacement responses](image)

**Figure 5.** Displacement responses with the Wilson method, Bathe method, GGW method, trapezoidal rule, and the proposed method from 40 to 50 s.

![Velocity responses](image)

**Figure 6.** Velocity responses with the Wilson method, Bathe method, GGW method, trapezoidal rule, and the proposed method from 40 to 50 s.
1.9999992384, the pendulum will oscillate, and
\[ \theta_2 \] such that the movement mode of the
0
\[-\bigg( \frac{g \theta}{L} \bigg) \sin \left( \frac{t}{2 \omega_0} \right) \],
0
\[ \omega_0 \]
\[ \theta \]
0
\[ \kappa \]

where
\[ \omega = gL \]
\[ \kappa \]
\[ \vartheta \]

It can be observed that the Wilson method has a larger amplitude decay when the
time step is a little longer, e.g., \( \Delta t = 0.2 \) s in this example, whereas the trapezoid rule,
Bathe method, GGW method, and the proposed method can still maintain good accuracy.
However, the Bathe method will bring a certain attenuation. Among all these methods, the
proposed algorithm shows a lower period elongation. Furthermore, as shown in Table 1,
the computational times of these methods are close to each other, thus the numerical
efficiency of the proposed method can be proved.

4.2. Nonlinear Single Pendulum

In order to verify the effectiveness of the proposed algorithm in nonlinear cases, an oscil-
lating nonlinear single pendulum [36] described in Figure 8 is considered in this subsection.

![Figure 7. Acceleration responses with the Wilson method, Bathe method, GGW method, trapezoidal rule, and the proposed method from 40 to 50 s.](image)

![Figure 8. The oscillating nonlinear single pendulum.](image)
The governing equation is

\[ \ddot{\theta} + \omega^2 \sin(\theta) = 0; \quad \theta(0) = \theta_0, \dot{\theta}(0) = \dot{\theta}_0 \]  

(33)

where \( \omega = \sqrt{g/L} \), \( \theta(t) \) is the angle between the rod and the vertical plane at time \( t \), \( L \) is the length of the rod regardless of mass, and \( g \) is the acceleration of gravity, assuming a dimensionless case of \( L = g = 1 \) so that \( \omega = 1 \).

Large motion causes the nonlinearity, the degree of which can be adjusted by changing the initial conditions. In this case, the energy \( E_{\text{total}} \) is conserved, which can be written as [37,38]

\[ \frac{1}{2} \left( \frac{d\theta}{dt} \right)^2 - \omega^2 \cos(\theta) = E_{\text{total}} \]  

(34)

Considering the initial conditions \( \theta_0 \) and \( \dot{\theta}_0 \), the equation can be expressed as:

\[ t = \pm \int_{\theta_0}^{\theta} \frac{1}{\sqrt{\theta_0^2 + 2\omega^2(\cos(\theta) - \cos(\theta_0))}} d\theta \]  

(35)

Letting \( \theta_0 = 0 \), the relationship of Equation (35) can be rewritten as:

\[ t = \pm \frac{1}{\dot{\theta}_0} \int_{0}^{\theta} \frac{1}{\sqrt{1 - \kappa^2 \sin^2 \left( \frac{\theta}{2} \right)}} d\theta = \frac{2}{|\dot{\theta}_0|} \text{Ei} \left( \sin \left( \frac{\theta}{2} \right), \kappa \right) \]  

(36)

where \( \kappa = 2\omega/\dot{\theta}_0 \) and \( \text{Ei}(z,k) \) is the elliptic integral of the first kind. With the change in \( \kappa \), the pendulum has two movement modes: if \( \kappa < 1 \), the pendulum will rotate around the center and the angle of rotation will keep increasing; if \( \kappa \geq 1 \), the pendulum will oscillate around the center of rotation and the angle will be limited into a certain range and form a periodic change. The maximum angle becomes:

\[ \theta_{\text{max}} = 2 \sin^{-1} \left( \frac{\theta_0}{2\omega} \right) \]  

(37)

Here we consider a highly nonlinear case; that is, when the speed of rotation is zero, the maximum oscillating angle reaches 179.9°. According to the proposed equation, the initial condition is set to be \( \theta_0 = 1.999999238456499 \) such that the movement mode of the pendulum is close to the critical state. This case can be used to verify the validity of the algorithm. For the proposed method, the parameters are \( c_1 = c_3 = 1/4, \alpha = 0, \beta = 0.5 \) and for the GGW method, the parameter is \( \alpha = 0 \).

The corresponding results are shown in Figure 9 with \( \Delta t/T = 10^{-4} \). It can be seen that the trapezoid rule, Wilson method, and Bathe method all give a result that the pendulum tends to rotate around the center, which indeed does not reflect the actual movement. Instead, the proposed method and GGW method generate numerical solutions which are very close to the exact solution. It can be also found that when the time step is \( \Delta t = T/50000 \), as shown in Figure 10, the solution of the proposed method possesses high accuracy compared with the exact solution.
obtain stable results. The corresponding results are shown in Figure 10, no matter how small the Δt is, the trapezoidal rule cannot obtain stable results.

![Graph](image)

**Figure 9.** Numerical solutions with the Wilson method, Bathe method, GGW method, trapezoidal rule and the proposed method when Δt/T = 1/10,000.

![Graph](image)

**Figure 10.** Results of the proposed method with different time steps Δt = T/10,000, T/20,000, T/50,000.

The computational time of these methods are listed in Table 2. We can find that the computational time of the proposed method, the GGW method, and the Bathe method are close to each other, which is a little longer than that of Wilson method, since these three methods are two-step ones. It takes the shortest time for computation with trapezoidal rule, but as shown in Figure 11, no matter how small the Δt is, the trapezoidal rule cannot obtain stable results.

**Table 2.** Computational time of different methods for Problem 4.2.

| Method          | 10^-4       |
|-----------------|------------|
| Proposed method | 0.947 s    |
| GGW method      | 1.100 s    |
| Bathe method    | 0.954 s    |
| Wilson method   | 0.744 s    |
| Trapezoidal rule| 0.052 s    |
4.3. Long Span Planar Skeletal Structure

A 25 m (length) × 1 m (height) large span truss model is considered to demonstrate the numerical stability and dissipation of the proposed algorithm. The model shown in Figure 12 is composed of 101 member bars, and each bar is a single element with hinged constraints at both ends. As for the mechanical properties of the model, the elastic modulus is \( E = 2.0 \times 10^5 \text{ MPa} \), the density is 7800 kg/m\(^3\), and the cross-section area of each bar is 100 mm\(^2\). An initial velocity \( v = 100 \text{ m/s} \) is set on the points A and B located near the middle of the span, as shown in Figure 12.

![Long span planar skeletal structure.](image_url)

Setting \( \Delta t = 0.01 \text{ s} \), the total solution time \( t_f = 10 \text{ s} \), parameters of the proposed method = 0.1, \( \beta = 0.5 \), \( c_1 = c_3 = 1/4 \), and the parameter of the GGW method \( \alpha = 0 \), the responses of point A obtained from different methods are shown in Figures 13–16. The response calculated by the trapezoidal rule starts to diverge after 3 s, and at around 8 s, it fails to give a reasonable result. However, other methods can all obtain convergent solutions with such a length of time step.

![Displacement responses of Point A with the Wilson method, Bathe method, GGW method, trapezoidal rule, and the proposed method: (a) responses from 0 to 10 s; (b) partial enlarged drawing of responses from 9 to 10 s.](image_url)
Figure 14. Velocity responses of Point A with the Wilson method, Bathe method, GGW method, trapezoidal rule, and the proposed method: (a) responses from 0 to 10 s; (b) partial enlarged drawing of responses from 9 to 10 s.

Figure 15. Acceleration responses of Point A with the Wilson method, Bathe method, GGW method, trapezoidal rule and the proposed method: (a) responses from 0 to 10 s; (b) partial enlarged drawing of responses from 9 to 10 s.

Figure 16. Partial enlarged drawing of velocity responses from 3 to 8 s of Point A with the Wilson method, Bathe method, GGW method, and the proposed method.

As shown in Figure 14, the velocity response calculated by the GGW method is larger than that calculated by other methods, i.e., the GGW method does not filter the effects of high-frequency oscillations. In order to demonstrate this effect more clearly, the velocity response from 3 to 8 s in Figure 14 is replotted in Figure 16. The presented method has much improved the performance of the GGW method on high-frequency oscillation problems. Figure 16 also shows that after filtering out high-frequency effects, the velocity responses
calculated by the proposed method, Bathe method and Wilson method oscillate between −10 and 10 m/s, while the response of the GGW method oscillates between −50 and 50 m/s.

To study the influence of parameter $\alpha$ on the results of the proposed method, setting the time step $\Delta t = 0.001$ s, the response of this example is computed with different $\alpha$ ($\alpha = 0.005, 0.1, 0.25, 0.5, 0.667$) respectively, and the results are plotted in Figures 17–19. According to Figures 17–19, as $\alpha$ becomes larger, the high-frequency oscillation gradually weakens while the low-frequency response is well preserved. It demonstrates the superiority of the proposed method with controllable numerical damping.

![Figure 17](image1.png)

Figure 17. Displacement responses of Point A for various parameter $\alpha$: (a) displacement responses from 0 to 1.5 s; (b) partial enlarged drawing of responses.

![Figure 18](image2.png)

Figure 18. Velocity responses of Point A for various parameter $\alpha$: (a) velocity responses from 0 to 1.5 s; (b) partial enlarged drawing of responses.

![Figure 19](image3.png)

Figure 19. Acceleration responses of Point A for various parameter $\alpha$: (a) acceleration responses from 0 to 1.5 s; (b) partial enlarged drawing of responses.
To verify the decaying speed of the numerical solution with the proposed method, the time steps are set to be $\Delta t = 0.001, 0.005, 0.01$, and $0.02 \text{ s}$, respectively, and the corresponding results are shown in Figures 20 and 21, where $\alpha = 2/7$ so that the corresponding final spectral radius is $0.5$, and low frequencies are retained while high frequencies are filtered out.

![Figure 20](image1.png)

**Figure 20.** Displacement responses of Point A with different time steps.

![Figure 21](image2.png)

**Figure 21.** Velocity responses of Point A with different time steps.

It can be found that when the time step increases, the high-frequency part of the numerical solution gradually disappears while the low-frequency part is retained. This conclusion shows that the numerical result obtained from the proposed method gradually attenuates, but the accurate and effective low-frequency solution can still be obtained. This result can help to improve the computational efficiency.

When filtering out high-frequency numerical solutions, it is also necessary to consider the influence of numerical damping on low-frequency solutions. Due to the rapid numerical damping and amplitude decay, the low-frequency solution of some algorithms will gradually lose its value during calculation. When using the Wilson method, with the passage of time, the amplitude of numerical solution gradually decays. As for the GGW method, the numerical damping can be controlled by changing the parameter $\alpha$, but its solution attenuates too fast, which will also cause numerical dissipation in the low-frequency mode when filtering out the high-order modes.

As a significant improvement to the GGW method, the proposed method can effectively avoid such a problem. To verify the superior accuracy of the proposed method, the responses are computed with the Wilson method, GGW method, and proposed method.
when Δt = 0.01 s. The corresponding results are shown in Figure 22, where α = 0.05 for the GGW method and α = 2/7 for the proposed method.

![Figure 22](image_url)

**Figure 22.** Displacement responses of Point A within 8–15 s using the Wilson method, Bathe method, and the proposed method when Δt = 0.01 s.

It can be found from Figure 22 that the accuracy of the proposed method is better than both the GGW method and the Wilson method with the same time step. This is because of the unnecessary numerical dissipation of these two methods. Using a smaller time step can improve the accuracy of these two methods, but this will increase the cost of computation.

To compare the efficiency of the proposed method and the Bathe method, the displacement responses are computed with different time steps: Δt = 0.03, 0.02, and 0.01 s. Letting α = 1/7, the results are shown in Figure 23 and the computational time versus different Δt is listed in Table 3.

![Figure 23](image_url)

**Figure 23.** Displacement responses of Point A from 14 to 15 s with the Bathe method and the proposed method using different time steps.

**Table 3.** Computational time of Bathe method and proposed method.

| Method       | Δt 0.03 s | Δt 0.02 s | Δt 0.01 s |
|--------------|-----------|-----------|-----------|
| Bathe method | 3.895 s   | 5.293 s   | 10.186 s  |
| Proposed method | 5.039 s   | 7.409 s   | 14.133 s  |
As shown in Table 3, the computational time of the Bathe method is shorter than the proposed method when the same time step is used. However, we can find from Figure 23 that the accuracy of the proposed method is better than the Bathe method with the same time step. That is to say, under the conditions of the same accuracy, the proposed method calculates faster with larger time step than Bathe method with smaller time step. For example, in Figure 23, the Bathe method uses 10.186s with $\Delta t = 0.01$ s to obtain results of the same accuracy as the proposed method, which takes 5.039s with $\Delta t = 0.03$ s. From this point of view, the proposed method is sufficiently efficient in both the computational time and accuracy.

5. Conclusions

This paper proposes a new implicit time integration method based on the GGW method. To control the numerical damping effectively, the time step is divided into two sub-steps. In the first step, the GGW method is used, and in the second step, a new parameter is introduced according to the idea of trapezoidal integration. The numerical analysis of numerical stability and accuracy shows that spectral radius, period extension rate, and amplitude attenuation rate of this method can all be controlled by changing the parameters $\alpha$ and $\beta$. Generally, we take $\beta = 1/2$ and $\alpha \in [2/19, 2/7]$, thus $\rho_\infty \in [0.5, 0.8]$. Several numerical examples are given to demonstrate the effectiveness of the proposed method.

The first representative example is an undamped single-DOF problem. The results show that the proposed method has better accuracy, smaller period extension, and smaller amplitude attenuation compared with other well-known time integration methods. The second example is a large displacement nonlinear problem. When using the same time step, the Bathe method, trapezoidal rule, and Wilson method cannot compute reasonable results, while the results of the proposed method are very similar to the exact solutions, which proves that the method is suitable for nonlinear problems. The third example is a large-span planar structure problem. The results show that the proposed method can not only possess high dissipation in a high-frequency range, but also preserve good accuracy in a low-frequency range. Hence, a larger time step can be chosen to solve such problems.

In summary, the proposed method has the following merits:

1. Reasonable and controllable numerical damping. It possesses high dissipation in a high-frequency range and preserves good accuracy in a low-frequency range as well;
2. Applicability of both linear and nonlinear problems. This method can maintain high accuracy in large-displacement nonlinear problems;
3. Due to the advantages mentioned above, a stable and reasonable numerical result can still be obtained when a larger time step is selected, which results in the superior efficiency.

The proposed method is expected to have a bright future in nonlinear vibration problems.

**Author Contributions:** Conceptualization, Q.X.; Methodology, Q.X., W.W. and Q.Y.; software, W.W. and Q.Y.; validation, Q.X. and W.W.; formal analysis, W.W.; investigation, Q.X. and W.W.; resources, Q.X.; data curation, Q.X. and W.W.; writing—original draft preparation, W.W.; writing—review and editing, Q.X. and W.W.; visualization, Q.X.; supervision, Q.X.; project administration, Q.X.; funding acquisition, Q.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Natural Science Foundation of China (No. 51508305). The authors are solely responsible for the content.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.
References

1. Bathe, K.-J. Finite Element Procedures; Prentice Hall: Denver, CO, USA, 2006.
2. Kim, W. A simple explicit single step time integration algorithm for structural dynamics. *Int. J. Numer. Methods Eng.* 2019, 119, 383–403. [CrossRef]
3. Bathe, K.J. Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration scheme. *Comput. Struct.* 2007, 85, 437–445. [CrossRef]
4. Newmark, N.M. A method of computation for structural dynamics. *J. Eng. Mech. Div.* 1959, 85, 67–94. [CrossRef]
5. Wilson, E.; Farhoomand, I.; Bathe, K. Nonlinear dynamic analysis of complex structures. *Earthq. Eng. Struct. Dyn.* 1972, 1, 241–252. [CrossRef]
6. Chung, J.; Hulbert, G. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-$\alpha$ method. *J. Appl. Mech.* 1993, 60, 371–375. [CrossRef]
7. Hilber, H.M.; Hughes, T.J.; Taylor, R.L. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthq. Eng. Struct. Dyn.* 1977, 5, 283–292. [CrossRef]
8. Bathe, K.J.; Baig, M.M.I. On a composite implicit time integration procedure for nonlinear dynamics. *Comput. Struct.* 2005, 83, 2513–2524. [CrossRef]
9. Armero, F.; Romero, I. On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. Part I: Low-order methods for two model problems and nonlinear elastodynamics. *Comput. Methods Appl. Mech. Eng.* 2001, 190, 2603–2649. [CrossRef]
10. Kuhl, D.; Crisfield, M. Energy-conserving and decaying algorithms in non-linear structural dynamics. *Int. J. Numer. Methods Eng.* 1999, 45, 569–599. [CrossRef]
11. Hughes, T.J. Analysis of transient algorithms with particular reference to stability behavior. *Comput. Methods Transient Anal. Amst. N.-Holl.* 1983, 67–155.
12. Bathe, K.; Noh, G. Insight into an implicit time integration scheme for structural dynamics. *Comput. Struct.* 2012, 98–99, 1–6. [CrossRef]
13. Kim, W.; Choi, S.Y. An improved implicit time integration algorithm: The generalized composite time integration algorithm. *Comput. Struct.* 2018, 196, 341–354. [CrossRef]
14. Hughes, T.J. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis; Dover Publications: New York, NY, USA, 2012.
15. Zhang, J.; Liu, Y.H.; Liu, D.H. Accuracy of a composite implicit time integration scheme for structural dynamics. *Int. J. Numer. Methods Eng.* 2017, 109, 368–406. [CrossRef]
16. Subbaraj, K.; Dokainish, M. A survey of direct time-integration methods in computational structural dynamics—II. Implicit methods. *Comput. Struct.* 1989, 32, 1387–1401. [CrossRef]
17. Zhang, X. *Computational Dynamics*; Tsinghua University Press: Beijing, China, 2015. (In Chinese)
18. Marburg, S. Numerical damping in the acoustic boundary element method. *Acta Acust. United Acust.* 2016, 102, 415–418. [CrossRef]
19. Xing, Y.F.; Ji, Y.; Zhang, H.M. On the the construction of a type of composite time integration methods. *Comput. Struct.* 2019, 221, 157–178. [CrossRef]
20. Dong, S. BDF-like methods for nonlinear dynamic analysis. *J. Comput. Phys.* 2010, 229, 3019–3045. [CrossRef]
21. Rezaiee-Pajand, M.; Sarafrazi, S.R. A mixed and multi-step higher-order implicit time integration family. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* 2010, 224, 2097–2108. [CrossRef]
22. Akbar Gholampour, A.; Ghassemieh, M.; Karimi-Rad, M. A new unconditionally stable time integration method for analysis of nonlinear structural dynamics. *J. Appl. Mech.* 2013, 80, 021024. [CrossRef]
23. Noh, G.; Bathe, K.-J. The Bathe time integration method with controllable spectral radius: The $\rho_{\infty}$-Bathe method. *Comput. Struct.* 2019, 212, 299–310. [CrossRef]
24. Kim, W.; Reddy, J. An improved time integration algorithm: A collocation time finite element approach. *Int. J. Struct. Stab. Dyn.* 2017, 17, 1750024. [CrossRef]
25. Malakiyeh, M.M.; Shojaee, S.; Bathe, K.-J. The Bathe time integration method revisited for prescribing desired numerical dissipation. *Comput. Struct.* 2019, 212, 289–298. [CrossRef]
26. Noh, G. For direct time integrations: A comparison of the Newmark and $\rho_{\infty}$-Bathe schemes. *Comput. Struct.* 2019, 225, 106079. [CrossRef]
27. Ghassemieh, M.; Karimi-Rad, M. A Parabolic Acceleration Time Integration for Dynamic Problems. *J. Civ. Surv. Eng.* 2011, 45, 35–43.
28. Chandra, Y.; Zhou, Y.; Stanciulescu, I.; Eason, T.; Spottswood, S. A robust composite time integration scheme for snap-through problems. *Comput. Mech.* 2015, 55, 1041–1056. [CrossRef]
29. Rezaiee-Pajand, M.; Karimi-Rad, M. More accurate and stable time integration scheme. *Eng. Comput.* 2015, 31, 791–812. [CrossRef]
30. Wen, W.; Wei, K.; Lei, H.; Duan, S.; Fang, D. A novel sub-step composite implicit time integration scheme for structural dynamics. *Comput. Struct.* 2017, 182, 176–186. [CrossRef]
31. Xing, Q.; Yang, Q.; Wang, W. A Time Integration Method Based on Galerkin Weak Form for Nonlinear Structural Dynamics. *Appl. Sci.* 2019, 9, 3076. [CrossRef]
32. Zienkiewicz, O.C. A new look at the Newmark, Houbolt and other time stepping formulas. A weighted residual approach. *Earthq. Eng. Struct. Dyn.* 1977, 5, 413–418. [CrossRef]
33. Borri, M.; Ghiringhelli, G.; Lanz, M.; Mantegazza, P.; Merlini, T. Dynamic response of mechanical systems by a weak Hamiltonian formulation. *Comput. Struct.* 1985, 20, 495–508. [CrossRef]

34. Xing, X.-H.; Zhang, X.; Lu, M.-W. A time integration method based on the weak form Galerkin method. *Eng. Mech.* 2006, 23, 8–12. (In Chinese)

35. Xing, Q.; Yang, X. An EEP adaptive strategy of the Galerkin FEM for dynamic equations of discrete systems. *Appl. Math. Mech* 2017, 38, 133–143. (In Chinese)

36. Kim, W.; Lee, J.H. An improved explicit time integration method for linear and nonlinear structural dynamics. *Comput. Struct.* 2018, 206, 42–53. [CrossRef]

37. Fung, T. Solving initial value problems by differential quadrature method—Part 1: First-order equations. *Int. J. Numer. Methods Eng.* 2001, 50, 1411–1427. [CrossRef]

38. Fung, T. Solving initial value problems by differential quadrature method—Part 2: Second-and higher-order equations. *Int. J. Numer. Methods Eng.* 2001, 50, 1429–1454. [CrossRef]