Received: 3 June 2021 | Revised: 19 August 2021 | Accepted: 10 September 2021

DOI: 10.1002/msd2.12007

RESEARCH ARTICLE

IJMSD International Journal of Mechanical System Dynamics Open Access WILEY

A generalized approach for implicit time integration of piecewise linear/nonlinear systems

Huimin Zhang[1,2] | Runsen Zhang[1,2] | Andrea Zanoni[2] | Pierangelo Masarati[2]

[1]School of Aeronautic Science and Engineering, Beihang University, Beijing, China

[2]Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Milano, Italy

Correspondence

Ms. Huimin Zhang, School of Aeronautic Science and Engineering, Beihang University, Beijing 100083, China.
Email: huimin.zhang@polimi.it

Funding information

China Scholarship Council, Grant/Award Numbers: 201906020240, 201906020241

Abstract

A generalized solution scheme using implicit time integrators for piecewise linear and nonlinear systems is developed. The piecewise linear characteristic has been well-discussed in previous studies, in which the original problem has been transformed into linear complementarity problems (LCPs) and then solved via the Lemke algorithm for each time step. The proposed scheme, instead, uses the projection function to describe the discontinuity in the dynamics equations, and solves for each step the nonlinear equations obtained from the implicit integrator by the semismooth Newton iteration. Compared with the LCP-based scheme, the new scheme offers a more general choice by allowing other nonlinearities in the governing equations. To assess its performances, several illustrative examples are solved. The numerical solutions demonstrate that the new scheme can not only predict satisfactory results for piecewise nonlinear systems, but also exhibits substantial efficiency advantages over the LCP-based scheme when applied to piecewise linear systems.

KEYWORDS

implicit time integrator, piecewise linear and nonlinear systems, projection function

1 | INTRODUCTION

Piecewise linear and nonlinear systems, such as gap-activated springs in vibrating machines,[1,2] structures with damage or clearance,[3] gear backlashes,[4] and drag torques,[5] are commonly used in civil engineering, aerospace, mechanical engineering, and infrastructures. Because of the piecewise linear and nonlinear characteristics, these systems exhibit very complex and diverse dynamic behaviors. However, this kind of nonlinearity also renders these problems more difficult to solve analytically. For single-degree-of-freedom (SDOF) systems, there are some traditional methods that can be used, such as the averaging method,[6] the perturbation method,[7] and the harmonic balance method,[8] but when the number of degrees-of-freedom increases, these methods become tedious and impractical. Therefore, numerical integration is a more general choice for such problems to seek approximate solutions.

As the most commonly used numerical tool for solving semidiscrete dynamic equations, time integration methods, which are generally divided into explicit and implicit categories, are simply reviewed here. Explicit methods, including the central difference method, the Tchamwa–Wielgosz method,[9] the explicit generalized-α method,[10] the Noh–Bathe method,[11] and many others, are easy to implement, but they are all conditionally stable, which means that their available time step sizes are limited by the stability domain of the method for linear problems, and are hard to be determined for nonlinear problems. As a result, explicit methods are more commonly used when the allowable step size is at the same level as the step size that is required to describe the problem within the desired accuracy, such as in wave propagation and impact problems. Implicit methods, including the Newmark method,[12] the HHT-α method developed by Hilber, Hughes and Taylor,[13] the generalized-α method,[14] the Bathe method,[15] the linear two-step method[16,17] and many others,[18,19] can be designed to realize unconditional stability for linear problems, so their time step sizes only depend on the desired solution accuracy, at the cost of greater computational effort. When applied to linear systems, implicit methods require the factorization of the effective stiffness matrix; for nonlinear systems, an iterative solution approach is inevitable. A more comprehensive review of time integration methods can be found in Tamma et al.[20]

Considering piecewise linear and nonlinear problems, a simple way for numerical solutions is to employ an explicit integrator directly, like in [4,21]. These methods update the current step by known state variables, and then make the updated variables satisfy the piecewise characteristics. However, as mentioned above, one drawback of explicit methods is their conditional stability, and since the dynamic equations cannot be satisfied implicitly at each time step, their solutions for nonlinear problems may be not so reliable.

On the other hand, there are also some attempts to provide available schemes using implicit integrators. Their main difficulty is to solve the implicit dynamics equations with piecewise characteristics at each time step. Yu[22] and Fadaee et al.[23] proposed that piecewise linear equations can be transformed into equivalent linear complementarity problems (LCPs),[24] and then solved by the Lemke algorithm. They use the Bozzak–Newmark method[25] as the implicit integrator in the schemes. A similar idea was employed by He et al.,[26] based on the precise integration method (PIM)[27] and the Lemke algorithm. These methods, referred to as the LCP-based schemes, are applicable only when the piecewise linear characteristic is the only nonlinearity in the dynamics equations. They have been successfully applied to simulate the dynamic response of multidegree-of-freedom (MDOF) linear systems in [3,28,29].

As mentioned earlier, existing LCP-based schemes can handle piecewise linear problems; however, if other nonlinearities are involved, only explicit integrators can be used. To overcome this limitation, this study aims to provide a more elegant and versatile solution scheme for piecewise linear and nonlinear systems using implicit time integration schemes. The projection function,[30,31] which has been used to deal with the discontinuities caused by impacts and frictions, is introduced to model the piecewise linear and nonlinear characteristics. Without loss of generality, the generalized-α method[14] is employed to integrate the dynamics equations, and the resulting nonlinear equations are solved at each time step by the semismooth Newton iteration.[32] Compared with the LCP-based schemes, the proposed approach can be directly used to solve general piecewise nonlinear systems. At the same time, the proposed approach shows significant efficiency advantages for systems with a large number of piecewise features, due to the fast convergence rate of the Newton iteration. The proposed approach is applied to several numerical examples, to assess its numerical performances.

This paper is organized as follows. In Section 2, the dynamic equations for piecewise linear and nonlinear systems are presented using the projection function. The computational procedures are provided in Section 3, based on the generalized-α method and the semismooth Newton iteration. Numerical experiments are implemented in Section 4. Conclusions are finally drawn in Section 5.

2 | FORMULATION

For illustrative purposes, the simple piecewise linear SDOF system shown in Figure 1 is discussed first. Its elastic restoring force, $N(x)$, can be formulated as follows:

$$N(x) = \begin{cases} k_1 x, & x \le \Delta \\ k_1 x + k_2(x - \Delta), & x > \Delta \end{cases} \tag{1}$$

where x is the displacement of the cart, Δ is the initial clearance, and k_1 and k_2 are the stiffness parameters of the system. Using this simple model, the basic ideas of the LCP-based scheme are reviewed here. From Equation (1), the complementarity relationship can be established by introducing two auxiliary variables, g and y, as

$$g = y - x + \Delta, \tag{2a}$$

$$y \ge 0, \tag{2b}$$

$$g \ge 0, \tag{2c}$$

$$y \cdot g = 0. \tag{2d}$$

Using the variable y, Equation (1) can be transformed into

$$N(x) = k_1 x + k_2 y. \tag{3}$$

After applying the desired time integration scheme, the numerical displacement x_k at time t_k satisfies

$$k^* x_k + k_2 y_k = r_k^* \tag{4}$$

where k^* is the effective stiffness value, which is a constant for linear systems; r_k^* is the effective load, which needs to be updated for each time step. By combining Equations (2) and (4), one obtains the LCP

$$y_k = (k^* + k_2)^{-1} k^* g_k + (k^* + k_2)^{-1}(r_k^* - k^*\Delta), \tag{5a}$$

$$y_k \ge 0, \tag{5b}$$

$$g_k \ge 0, \tag{5c}$$

$$y_k \cdot g_k = 0. \tag{5d}$$

At each step, y_k and g_k are first obtained by solving the LCP via the Lemke algorithm. Then x_k is computed from Equation (2a) as $x_k = y_k - g_k + \Delta$, and other state variables are updated according to the recursive scheme. The above procedure can be easily extended

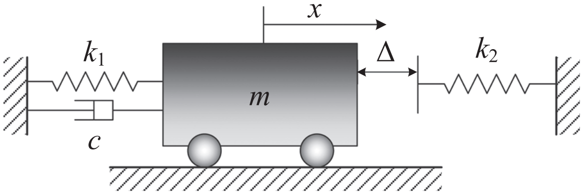

FIGURE 1 Piecewise linear SDOF system

to MDOF systems. However, due to the lack of an efficient solver for nonlinear complementarity problems, the LCP-based schemes are designed only for piecewise linear systems. To our knowledge, so far there is no available solution scheme using implicit integrators for piecewise nonlinear systems in the literature.

In the proposed scheme, the projection function is introduced to formulate more general forms of piecewise linear and nonlinear characteristics. From convex analysis, the projection of a point w in a convex set C is the closest point in C to w, defined as follows:

$$\mathrm{proj}_C(w) = \arg\min_{z \in C} |w - z|. \tag{6}$$

It follows that if , and otherwise, $\mathrm{proj}_C(w)$ is the nearest boundary point of C to w. For example,

$$\mathrm{proj}_{[a,+\infty)}(w) = \begin{cases} a, & w \le a \\ w, & w > a \end{cases}, \tag{7a}$$

$$\mathrm{proj}_{[a,b)}(w) = \begin{cases} a, & w \le a \\ w, & a < w \le b, \\ b, & w > b \end{cases} \tag{7b}$$

$$a, b, w \in \mathbb{R}. \tag{7c}$$

By using the projection function, Equation (1) can be transformed into

$$N(x) = k_1 x + k_2 \mathrm{proj}_{\mathbb{R}_0^+}(x - \Delta). \tag{8}$$

It indicates that the projection function in Equation (8) can replace the role of the auxiliary variable y of the LCP-based scheme in Equation (3). For general multistage piecewise linear systems, as follows:

$$N(x) = \begin{cases} k_1 x + z_1, & x \le x_1 \\ k_2 x + z_2, & x_1 < x \le x_2 \\ \cdots & \\ k_n x + z_n, & x > x_{n-1} \end{cases} \tag{9}$$

where the constants $z_1, z_2, \ldots, z_n$ are defined as required to make $N(x)$ continuous at $x_1, x_2, \ldots, x_{n-1}$, the equivalent form using the projection functions can be expressed as follows:

$$N(x) = k_1 x + z_1 + \sum_{i=2}^{n} (k_i - k_{i-1}) \mathrm{proj}_{\mathbb{R}_0^+}(x - x_{i-1}) \tag{10}$$

where the convex sets are set as $\mathbb{R}_0^+$ uniformly. Some special cases, such as

$$N(x) = \begin{cases} k(x + \Delta_1), & x \le -\Delta_1 \\ 0, & -\Delta_1 < x \le \Delta_2 \\ k(x - \Delta_2), & x > \Delta_2 \end{cases} \tag{11}$$

where Δ_1 and Δ_2 are the initial clearances, can be transformed into the simpler form

$$N(x) = k\left(x - \mathrm{proj}_{[-\Delta_1, \Delta_2]}(x)\right). \tag{12}$$

When extended to piecewise nonlinear systems, such as

$$N(x) = \begin{cases} k_1 x^{m_1} + z_1, & x \le x_1 \\ k_2 x^{m_2} + z_2, & x_1 < x \le x_2 \\ \cdots & \\ k_n x^{m_n} + z_n, & x > x_{n-1} \end{cases} \tag{13}$$

the equivalent form can be written as follows:

$$\begin{aligned} N(x) = k_1 x^{m_1} + z_1 + \sum_{i=2}^{n} \Bigg[& k_i \left(\mathrm{proj}^{m_i}_{[x_{i-1},+\infty)}(x) - x_{i-1}^{m_i} \right) \\ & - k_{i-1} \left(\mathrm{proj}^{m_{i-1}}_{[x_{i-1},+\infty)}(x) - x_{i-1}^{m_{i-1}} \right) \Bigg]. \end{aligned} \tag{14}$$

The contribution of each term in Equation (14) can be decoupled as follows:

$$\begin{aligned} & k_1 x^{m_1} + z_1 - k_1 \left(\mathrm{proj}^{m_1}_{[x_1,+\infty)}(x) - x_1^{m_1} \right) \\ & = \begin{cases} k_1 x^{m_1} + z_1, & x \le x_1 \\ k_1 x_1^{m_1} + z_1, & x > x_1 \end{cases}, \end{aligned} \tag{15a}$$

$$\begin{aligned} & k_i \left(\mathrm{proj}^{m_i}_{[x_{i-1},+\infty)}(x) - x_{i-1}^{m_i} \right) - k_i \left(\mathrm{proj}^{m_i}_{[x_i,+\infty)}(x) - x_i^{m_i} \right) \\ & = \begin{cases} 0, & x \le x_{i-1} \\ k_i \left(x^{m_i} - x_{i-1}^{m_i} \right), & x_{i-1} < x \le x_i \\ k_i \left(x_i^{m_i} - x_{i-1}^{m_i} \right), & x > x_i \end{cases}, i = 2, 3, \cdots, n-1, \end{aligned} \tag{15b}$$

$$k_n \left(\mathrm{proj}^{m_n}_{[x_{n-1},+\infty)}(x) - x_{n-1}^{m_n} \right) = \begin{cases} 0, & x \le x_{n-1} \\ k_n \left(x^{m_n} - x_{n-1}^{m_n} \right), & x > x_{n-1} \end{cases}. \tag{15c}$$

Combined with the continuity conditions, that is $k_{i-1} x_{i-1}^{m_{i-1}} + z_{i-1} = k_i x_{i-1}^{m_i} + z_i$, $i = 2, 3, \ldots, n$, it can be easily checked that Equation (14) is the same as the original Equation (13). In a similar way, for general polynomial functions, as

$$N(x) = \begin{cases} \sum_{j=1}^{p_1} k_{1,j} x^{m_{1,j}} + z_1, & x \le x_1 \\ \sum_{j=1}^{p_2} k_{2,j} x^{m_{2,j}} + z_2, & x_1 < x \le x_2 \\ \cdots & \\ \sum_{j=1}^{p_n} k_{n,j} x^{m_{n,j}} + z_n, & x > x_{n-1} \end{cases} \tag{16}$$

where $p_1, p_2, \ldots, p_n$ are positive integers, the equivalent form using the projection function is

$$\begin{aligned} N(x) = & \sum_{j=1}^{p_1} k_{1,j} x^{m_{1,j}} + z_1 \\ & + \sum_{i=2}^{n} \Bigg[\sum_{j=1}^{p_i} k_{i,j} \left(\mathrm{proj}^{m_{i,j}}_{[x_{i-1},+\infty)}(x) - x_{i-1}^{m_{i,j}} \right) \\ & - \sum_{j=1}^{p_{i-1}} k_{i-1,j} \left(\mathrm{proj}^{m_{i-1,j}}_{[x_{i-1},+\infty)}(x) - x_{i-1}^{m_{i-1,j}} \right) \Bigg]. \end{aligned} \tag{17}$$

Some special cases, like

$$N(x)=\begin{cases}\sum_{j=1}^{p}k_j(x+\varDelta_1)^{m_j}, & x\le-\varDelta_1\\ 0, & -\varDelta_1<x\le\varDelta_2\\ \sum_{j=1}^{p}k_j(x-\varDelta_2)^{m_j}, & x>\varDelta_2\end{cases}\tag{18}$$

can also be formulated as the simpler form

$$N(x)=\sum_{j=1}^{p}k_j\Big(x-\mathrm{proj}_{[-\varDelta_1,\varDelta_2]}(x)\Big)^{m_j}.\tag{19}$$

Since formulating nonlinearities as polynomial functions can cover most situations produced in structural dynamics, other possible forms, which can also be expressed using the projection function by selecting proper projected functions and convex sets, are not discussed one by one. The projection function offers a more elegant, versatile and flexible manner to express both piecewise linear and nonlinear characteristics. The solution of nonlinear equations involving projection functions can be included in the solution program for general nonlinear equations without additional computational procedures, as discussed in more detail in Section 3.

To present a general formulation of piecewise linear and nonlinear MDOF systems, an auxiliary variable $\boldsymbol{y}\in\mathbb{R}^{q\times1}$ is introduced as follows:

$$\boldsymbol{y}=\boldsymbol{p}(\boldsymbol{x})=\begin{bmatrix}\mathrm{proj}_{C_1}(w_1(\boldsymbol{x}))\\ \mathrm{proj}_{C_2}(w_2(\boldsymbol{x}))\\ \cdots\\ \mathrm{proj}_{C_q}(w_q(\boldsymbol{x}))\end{bmatrix}\tag{20}$$

where $\boldsymbol{x}\in\mathbb{R}^{s\times1}$ denotes the displacement variable, $\boldsymbol{w}(\boldsymbol{x})=[w_1(\boldsymbol{x}),w_2(\boldsymbol{x}),\ldots,w_q(\boldsymbol{x})]^{\mathrm{T}}$ are projected functions with respect to $\boldsymbol{x}$. Then, the general form of the dynamic equations can be expressed as follows:

$$\boldsymbol{M}\ddot{\boldsymbol{x}}+\boldsymbol{C}\dot{\boldsymbol{x}}+\boldsymbol{N}(\boldsymbol{x},\boldsymbol{y},t)=\boldsymbol{f}(t)\tag{21}$$

where $\boldsymbol{M}\in\mathbb{R}^{s\times s}$ and $\boldsymbol{C}\in\mathbb{R}^{s\times s}$ are the mass and damping matrices, assumed constant for simplicity, the over dot indicates differentiation with respect to time t, $\boldsymbol{N}(\boldsymbol{x},\boldsymbol{y},t)\in\mathbb{R}^{s\times1}$ is the elastic force vector, and $\boldsymbol{f}(t)\in\mathbb{R}^{s\times1}$ is the external load vector. Equations (20)–(21) constitute the dynamics equations to be solved. The introduction of the variable $\boldsymbol{y}$ is not strictly necessary, but it makes the expressions more readable. In particular, the dynamics equations of piecewise linear systems can be written as follows:

$$\boldsymbol{M}\ddot{\boldsymbol{x}}+\boldsymbol{C}\dot{\boldsymbol{x}}+\boldsymbol{K}_x\boldsymbol{x}+\boldsymbol{K}_y\boldsymbol{y}=\boldsymbol{f}(t)\tag{22}$$

and

$$\boldsymbol{y}=\boldsymbol{p}(\boldsymbol{x})=\begin{bmatrix}\mathrm{proj}_{C_1}(\boldsymbol{W}_1\boldsymbol{x})\\ \mathrm{proj}_{C_2}(\boldsymbol{W}_2\boldsymbol{x})\\ \cdots\\ \mathrm{proj}_{C_q}(\boldsymbol{W}_q\boldsymbol{x})\end{bmatrix},\ \boldsymbol{W}=\begin{bmatrix}\boldsymbol{W}_1\\ \boldsymbol{W}_2\\ \cdots\\ \boldsymbol{W}_q\end{bmatrix}\tag{23}$$

where $\boldsymbol{K}_x\in\mathbb{R}^{s\times s}$, $\boldsymbol{K}_y\in\mathbb{R}^{s\times q}$, $\boldsymbol{W}_i\in\mathbb{R}^{1\times s}$, $\boldsymbol{W}\in\mathbb{R}^{q\times s}$ are constant matrices. For piecewise linear systems, only Equation (23) needs to be solved iteratively, so their computational procedure is discussed separately.

3 | COMPUTATIONAL PROCEDURE

Without loss of generality, the generalized-α method is employed to illustrate the computational procedures. The standard generalized-α method in Chung and Hulbert[14] only shows first-order accuracy for acceleration, so an improved scheme in Arnold and Brüls,[33] which holds second-order accuracy for displacement, velocity, and acceleration, is adopted. It introduces an acceleration-like variable $\boldsymbol{a}$ as follows:

$$(1-\alpha)\boldsymbol{a}_{k+1}+\alpha\boldsymbol{a}_k=(1-\delta)\ddot{\boldsymbol{x}}_{k+1}+\delta\ddot{\boldsymbol{x}}_k,\ \boldsymbol{a}_0=\ddot{\boldsymbol{x}}_0\tag{24}$$

where the subscript k denotes the state variables of step k, α, and δ are parameters of the algorithm. Using the acceleration-like variable $\boldsymbol{a}$, the displacement and velocity of step $k+1$ are updated according to

$$\begin{aligned}\boldsymbol{x}_{k+1}&=\boldsymbol{x}_k+h\dot{\boldsymbol{x}}_k+\frac{h^2}{2}((1-2\beta)\boldsymbol{a}_k+2\beta\boldsymbol{a}_{k+1})\\ \dot{\boldsymbol{x}}_{k+1}&=\dot{\boldsymbol{x}}_k+h((1-\gamma)\boldsymbol{a}_k+\gamma\boldsymbol{a}_{k+1})\end{aligned}\tag{25}$$

where $h=t_{k+1}-t_k$ is the time step size, and β and γ are additional parameters of the algorithm. The state variables of step $k+1$ satisfy the discrete dynamic equations, as

$$\boldsymbol{M}\ddot{\boldsymbol{x}}_{k+1}+\boldsymbol{C}\dot{\boldsymbol{x}}_{k+1}+\boldsymbol{N}(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1},t_{k+1})=\boldsymbol{f}(t_{k+1})\tag{26}$$

with

$$\boldsymbol{y}_{k+1}=\boldsymbol{p}(\boldsymbol{x}_{k+1})=\begin{bmatrix}\mathrm{proj}_{C_1}(w_1(\boldsymbol{x}_{k+1}))\\ \mathrm{proj}_{C_2}(w_2(\boldsymbol{x}_{k+1}))\\ \cdots\\ \mathrm{proj}_{C_q}(w_q(\boldsymbol{x}_{k+1}))\end{bmatrix}.\tag{27}$$

For piecewise linear systems, the dynamic equations to be solved are

$$\boldsymbol{M}\ddot{\boldsymbol{x}}_{k+1}+\boldsymbol{C}\dot{\boldsymbol{x}}_{k+1}+\boldsymbol{K}_x\boldsymbol{x}_{k+1}+\boldsymbol{K}_y\boldsymbol{y}_{k+1}=\boldsymbol{f}(t_{k+1})\tag{28}$$

with

$$\boldsymbol{y}_{k+1}=\boldsymbol{p}(\boldsymbol{x}_{k+1})=\begin{bmatrix}\mathrm{proj}_{C_1}(\boldsymbol{W}_1\boldsymbol{x}_{k+1})\\ \mathrm{proj}_{C_2}(\boldsymbol{W}_2\boldsymbol{x}_{k+1})\\ \cdots\\ \mathrm{proj}_{C_q}(\boldsymbol{W}_q\boldsymbol{x}_{k+1})\end{bmatrix}.\tag{29}$$

The generalized-α method embraces several well-known single-step schemes. For instance, if $\alpha=0$, it reduces to the

HHT-α method;[13] if $\delta = 0$, it reduces to the Bossak–Newmark method;[25] if $\alpha = \delta = 0$, it reduces to the Newmark method;[12] if $\alpha = \delta = 0, \gamma = 1/2, \beta = 1/4$, it reduces to the trapezoidal rule (TR). From linear spectral analysis, a set of optimal parameters, controlled by the spectral radius ρ_∞ at the high-frequency limit, for the generalized-α method was given as follows:

$$\begin{aligned} &\alpha = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \delta = \frac{\rho_\infty}{\rho_\infty + 1}, \\ &\beta = \frac{1}{(\rho_\infty + 1)^2}, \gamma = \frac{3 - \rho_\infty}{2(\rho_\infty + 1)}, \rho_\infty \in [0, 1]. \end{aligned} \tag{30}$$

A smaller ρ_∞ denotes worse accuracy and stronger numerical damping, which can be used to filter out inaccurate high-frequency contributions. Using Equation (30), the generalized-α method becomes a single-parameter scheme with respect to ρ_∞, and to avoid excessive loss of accuracy, $\rho_\infty \geq 0.6$ is recommended for general purpose integration.

3.1 | Computational procedure for piecewise linear systems

Considering the dynamic equations of piecewise linear systems in Equations (28)–(29), only Equation (29) needs to be solved iteratively, so the computational procedure is relatively simple. Substituting Equations (24)–(25) into Equation (28) yields

$$\boldsymbol{x}_{k+1} = \boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y \boldsymbol{y}_{k+1}) \tag{31}$$

where

$$\boldsymbol{S} = \frac{1-\alpha}{(1-\delta)\beta h^2}\boldsymbol{M} + \frac{\gamma}{\beta h}\boldsymbol{C} + \boldsymbol{K}_x, \tag{32a}$$

$$\begin{aligned} \boldsymbol{r}_{k+1} = \boldsymbol{f}(t_{k+1}) + \boldsymbol{M}\Bigg(&\frac{1-\alpha}{(1-\delta)\beta h^2}\boldsymbol{x}_k + \frac{1-\alpha}{(1-\delta)\beta h}\dot{\boldsymbol{x}}_k \\ &+ \frac{\delta}{1-\delta}\ddot{\boldsymbol{x}}_k + \frac{1-\alpha-2\beta}{2(1-\delta)\beta}\boldsymbol{a}_k\Bigg) + \boldsymbol{C}\Bigg(\frac{\gamma}{\beta h}\boldsymbol{x}_k + \frac{\gamma-\beta}{\beta}\dot{\boldsymbol{x}}_k \\ &+ \frac{\gamma-2\beta}{2\beta}h\boldsymbol{a}_k\Bigg). \end{aligned} \tag{32b}$$

Since the effective stiffness matrix $\boldsymbol{S}$ is constant, $\boldsymbol{S}^{-1}$ can be prepared before the step-by-step solutions. Using Equation (31), Equation (29) can be rewritten as

$$\begin{aligned} \boldsymbol{y}_{k+1} &= \boldsymbol{p}(\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y \boldsymbol{y}_{k+1})) \\ &= \begin{bmatrix} \mathrm{proj}_{C_1}(\boldsymbol{W}_1\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y \boldsymbol{y}_{k+1})) \\ \mathrm{proj}_{C_2}(\boldsymbol{W}_2\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y \boldsymbol{y}_{k+1})) \\ \cdots \\ \mathrm{proj}_{C_q}(\boldsymbol{W}_q\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y \boldsymbol{y}_{k+1})) \end{bmatrix}. \end{aligned} \tag{33}$$

The projection function $\mathrm{proj}_C(x)$ is nondifferentiable at the boundary of the convex set C. According to the semismooth Newton method,[32] the generalized derivative of $\mathrm{proj}_C(x)$ can be defined as follows:

$$\lambda_C(x) = \begin{cases} 0, & \text{if } x \notin C \\ \dfrac{1}{2}, & \text{if } x \in \partial C. \\ 1, & \text{else} \end{cases} \tag{34}$$

Consequently, at each step, the nonlinear equation to be solved is

$$\boldsymbol{g}(\boldsymbol{y}_{k+1}) = \boldsymbol{y}_{k+1} - \boldsymbol{p}(\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y \boldsymbol{y}_{k+1})). \tag{35}$$

Its Jacobian matrix with respect to $\boldsymbol{y}_{k+1}$ is

$$\boldsymbol{G}(\boldsymbol{y}_{k+1}) = \boldsymbol{I} + \boldsymbol{\Lambda}(\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y \boldsymbol{y}_{k+1}))\boldsymbol{W}\boldsymbol{S}^{-1}\boldsymbol{K}_y \tag{36}$$

where $\boldsymbol{I} \in \mathbb{R}^{q \times q}$ is the unit matrix, and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_{C_i}(\boldsymbol{W}_i\boldsymbol{x})), i = 1, 2, \ldots, q$. The iterative scheme can be written as follows:

$$\boldsymbol{y}_{k+1}^{l+1} = \boldsymbol{y}_{k+1}^{l} - \boldsymbol{G}\left(\boldsymbol{y}_{k+1}^{l}\right)^{-1}\boldsymbol{g}\left(\boldsymbol{y}_{k+1}^{l}\right) \tag{37}$$

where the superscript l denotes the iteration number. In Equation (36), only $\boldsymbol{\Lambda}$ needs to be updated at each iteration. To avoid repeated calculations, matrix $\boldsymbol{W}\boldsymbol{S}^{-1}\boldsymbol{K}_y$ can be precomputed. When the convergence condition is achieved, that is $|\boldsymbol{g}| < \epsilon$, where ϵ is the allowable tolerance error, $\boldsymbol{y}_{k+1}$ is obtained and $\boldsymbol{x}_{k+1}$ can be updated by Equation (31). Then other state variables are computed by

$$\begin{aligned} \ddot{\boldsymbol{x}}_{k+1} = &\frac{1-\alpha}{(1-\delta)\beta h^2}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) - \frac{1-\alpha}{(1-\delta)\beta h}\dot{\boldsymbol{x}}_k \\ &- \frac{\delta}{1-\delta}\ddot{\boldsymbol{x}}_k - \frac{1-\alpha-2\beta}{2(1-\delta)\beta}\boldsymbol{a}_k, \end{aligned} \tag{38a}$$

$$\dot{\boldsymbol{x}}_{k+1} = \frac{\gamma}{\beta h}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) - \frac{\gamma-\beta}{\beta}\dot{\boldsymbol{x}}_k - \frac{\gamma-2\beta}{2\beta}h\boldsymbol{a}_k, \tag{38b}$$

$$\boldsymbol{a}_{k+1} = \frac{1-\delta}{1-\alpha}\ddot{\boldsymbol{x}}_{k+1} + \frac{\delta}{1-\alpha}\ddot{\boldsymbol{x}}_k - \frac{\alpha}{1-\alpha}\boldsymbol{a}_k. \tag{38c}$$

For clarity, the step-by-step solution using the generalized-α method for piecewise linear systems is summarized in Table 1. Other implicit integrators can be used in a similar way. Compared with the LCP-based scheme, the proposed scheme employs the semismooth Newton iteration instead of the Lemke algorithm to solve $\boldsymbol{y}_{k+1}$. Since the semismooth Newton iteration has locally super-linear convergence rate, the state of the last step is used as the prediction. Consequently, if the current time step is in a smooth interval, the problem degenerates to a linear problem, requiring only one iteration. Only when the switching point is passed, the proposed scheme may need to use more iterations to reach a convergent solution. The

TABLE 1 Step-by-step solution using the generalized-α method for piecewise linear systems

```
A. Initial calculations
1. From the matrices $\boldsymbol{M}, \boldsymbol{C}, \boldsymbol{K}_x, \boldsymbol{K}_y, \boldsymbol{W}$, and the functions $\boldsymbol{f}(t), \boldsymbol{p}(\mathbf{x}), \Lambda(\mathbf{x})$;
2. Initialize $\mathbf{x}_0, \dot{\mathbf{x}}_0, \ddot{\mathbf{x}}_0, \boldsymbol{a}_0$, and $\boldsymbol{y}_0$;
3. Select the time step size $h$, the algorithmic parameters $\rho_\infty$,
   the tolerance error $\epsilon$, and the maximum number of iterations $N$;
4. Calculate integration constants:
   $\alpha = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \delta = \frac{\rho_\infty}{\rho_\infty + 1}, \beta = \frac{1}{(\rho_\infty + 1)^2}, \gamma = \frac{3 - \rho_\infty}{2(\rho_\infty + 1)}$,
   $a_0 = \frac{1-\alpha}{(1-\delta)\beta h^2}, a_1 = \frac{1-\alpha}{(1-\delta)\beta h}, a_2 = \frac{\delta}{1-\delta}, a_3 = \frac{1-\alpha-2\beta}{2(1-\delta)\beta}, b_0 = \frac{\gamma}{\beta h}, b_1 = \frac{\gamma-\beta}{\beta}$,
   $b_2 = \frac{\gamma-2\beta}{2\beta}h, c_0 = \frac{1-\delta}{1-\alpha}, c_1 = \frac{\delta}{1-\alpha}, c_2 = \frac{\alpha}{1-\alpha}$,
   $\boldsymbol{S}^{-1} = (a_0\boldsymbol{M} + b_0\boldsymbol{C} + \boldsymbol{K}_x)^{-1}, \boldsymbol{A} = \boldsymbol{W}\boldsymbol{S}^{-1}\boldsymbol{K}_y$.
B. For each time step
1. Calculate the effective load $\boldsymbol{r}_{k+1}$:
   $\boldsymbol{r}_{k+1} = \boldsymbol{f}(t_{k+1}) + \boldsymbol{M}(a_0\mathbf{x}_k + a_1\dot{\mathbf{x}}_k + a_2\ddot{\mathbf{x}}_k + a_3\boldsymbol{a}_k) + \boldsymbol{C}(b_0\mathbf{x}_k + b_1\dot{\mathbf{x}}_k + b_2\boldsymbol{a}_k)$;
2. Solve for $\boldsymbol{y}_{k+1}$:
   a. Predict $\boldsymbol{y}_{k+1}$
      $l = 0, \boldsymbol{y}_{k+1} = \boldsymbol{y}_k$;
   b. Compute $\boldsymbol{g}(\boldsymbol{y}_{k+1})$ and $\boldsymbol{G}(\boldsymbol{y}_{k+1})$
      $\boldsymbol{g}(\boldsymbol{y}_{k+1}) = \boldsymbol{y}_{k+1} - \boldsymbol{p}(\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y\boldsymbol{y}_{k+1}))$,
      $\boldsymbol{G}(\boldsymbol{y}_{k+1}) = \boldsymbol{I} + \Lambda(\boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y\boldsymbol{y}_{k+1}))\boldsymbol{A}$;
   c. Update $\boldsymbol{y}_{k+1}$
      $l = l + 1, \boldsymbol{y}_{k+1} = \boldsymbol{y}_{k+1} - \boldsymbol{G}(\boldsymbol{y}_{k+1})^{-1}\boldsymbol{g}(\boldsymbol{y}_{k+1})$;
   d. If $l < N$ and $|\boldsymbol{g}| > \epsilon$, go to b; If $l < N$ and $|\boldsymbol{g}| \le \epsilon$, go to 3; If $l \equiv N$,
      abort.
3. Solve for $\mathbf{x}_{k+1}, \ddot{\mathbf{x}}_{k+1}, \dot{\mathbf{x}}_{k+1}$, and $\boldsymbol{a}_{k+1}$:
   $\mathbf{x}_{k+1} = \boldsymbol{S}^{-1}(\boldsymbol{r}_{k+1} - \boldsymbol{K}_y\boldsymbol{y}_{k+1}), \ddot{\mathbf{x}}_{k+1} = a_0(\mathbf{x}_{k+1} - \mathbf{x}_k) - a_1\dot{\mathbf{x}}_k - a_2\ddot{\mathbf{x}}_k - a_3\boldsymbol{a}_k$,
   $\dot{\mathbf{x}}_{k+1} = b_0(\mathbf{x}_{k+1} - \mathbf{x}_k) - b_1\dot{\mathbf{x}}_k - b_2\boldsymbol{a}_k, \boldsymbol{a}_{k+1} = c_0\ddot{\mathbf{x}}_{k+1} + c_1\ddot{\mathbf{x}}_k - c_2\boldsymbol{a}_k$.
```

computational efficiency of the proposed scheme and the LCP-based scheme is compared based on the numerical example in Section 4.

3.2 | Computational procedure for piecewise nonlinear systems

Considering the dynamic equations of piecewise nonlinear systems of Equations (26)–(27), the nonlinear equation that need to be solved at each time step is

$$\boldsymbol{g}(\mathbf{x}_{k+1}) = \boldsymbol{M}\ddot{\mathbf{x}}_{k+1} + \boldsymbol{C}\dot{\mathbf{x}}_{k+1} + \boldsymbol{N}(\mathbf{x}_{k+1}, \boldsymbol{y}_{k+1}, t_{k+1}) - \boldsymbol{f}(t_{k+1}) \tag{39}$$

where $\boldsymbol{y}_{k+1} = \boldsymbol{p}(\mathbf{x}_{k+1}), \dot{\mathbf{x}}_{k+1}$ and $\ddot{\mathbf{x}}_{k+1}$ can be expressed as functions of $\mathbf{x}_{k+1}$ using Equation (38). The Jacobian matrix of $\boldsymbol{g}(\mathbf{x}_{k+1})$ with respect to $\mathbf{x}_{k+1}$ has the form

$$\begin{aligned}\boldsymbol{G}(\mathbf{x}_{k+1}) = \frac{1-\alpha}{(1-\delta)\beta h^2}\boldsymbol{M} + \frac{\gamma}{\beta h}\boldsymbol{C} + \boldsymbol{K}_x(\mathbf{x}_{k+1}, \boldsymbol{y}_{k+1}, t_{k+1}) \\ + \boldsymbol{K}_y(\mathbf{x}_{k+1}, \boldsymbol{y}_{k+1}, t_{k+1})\Lambda(\mathbf{x}_{k+1})\boldsymbol{W}(\mathbf{x}_{k+1})\end{aligned} \tag{40}$$

where

$$\boldsymbol{K}_x(\mathbf{x}, \boldsymbol{y}, t) = \frac{\partial \boldsymbol{N}(\mathbf{x}, \boldsymbol{y}, t)}{\partial \mathbf{x}^{\mathrm{T}}}, \tag{41a}$$

$$\boldsymbol{K}_y(\mathbf{x}, \boldsymbol{y}, t) = \frac{\partial \boldsymbol{N}(\mathbf{x}, \boldsymbol{y}, t)}{\partial \boldsymbol{y}^{\mathrm{T}}}, \tag{41b}$$

$$\boldsymbol{W}(\mathbf{x}) = \frac{\mathrm{d}\boldsymbol{w}(\mathbf{x})}{\mathrm{d}\mathbf{x}^{\mathrm{T}}}, \tag{41c}$$

$$\Lambda(\mathbf{x}) = \mathrm{diag}(\lambda_{C_i}(w_i(\mathbf{x}))), i = 1, 2, \cdots, q. \tag{41d}$$

The iterative scheme is

$$\mathbf{x}_{k+1}^{l+1} = \mathbf{x}_{k+1}^{l} - \boldsymbol{G}\left(\mathbf{x}_{k+1}^{l}\right)^{-1}\boldsymbol{g}\left(\mathbf{x}_{k+1}^{l}\right). \tag{42}$$

In this case, matrices $\boldsymbol{K}_x, \boldsymbol{K}_y, \boldsymbol{W}$, and Λ need to be updated at each iteration. Until the convergence condition is satisfied, $\mathbf{x}_{k+1}$ is obtained, and other state variables can be computed using Equation (38). The step-by-step solution is presented in Table 2.

LCP-based schemes have not yet been applied to piecewise nonlinear problems. Therefore, the previously described solution scheme, based on the implicit integration, is proposed for the first time in this study for such problems. Its numerical performance is assessed using the numerical examples presented in Section 4.

4 | NUMERICAL EXAMPLES

Some illustrative examples, including two piecewise linear and three piecewise nonlinear cases, are simulated to show the performance of the proposed scheme. The piecewise linear cases are employed to show the accuracy order and computational efficiency, compared to that of LCP-based methods, whereas the piecewise nonlinear cases are used to demonstrate the effectiveness for such type of problems. Note that the accuracy of numerical results depends on the used time integrator, so for piecewise linear systems, if the proposed scheme and the LCP-based scheme use the same integrator and the same step size, they will present the same solutions. As a result, only the computational efficiency of these two schemes is compared considering the MDOF case. In all the examples, the reference solutions are obtained using a fourth-order explicit Runge–Kutta method with a very small time step.

TABLE 2 Step-by-step solution using the generalized-α method for piecewise nonlinear systems

A. Initial calculations

1. From the matrices $\boldsymbol{M}$, $\boldsymbol{C}$, and the functions $\boldsymbol{N}(\boldsymbol{x},\boldsymbol{y},t)$, $\boldsymbol{K}_x(\boldsymbol{x},\boldsymbol{y},t)$, $\boldsymbol{K}_y(\boldsymbol{x},\boldsymbol{y},t)$, $\boldsymbol{f}(t)$, $\boldsymbol{w}(\boldsymbol{x})$, $\boldsymbol{W}(\boldsymbol{x})$, $\boldsymbol{p}(\boldsymbol{x})$, $\boldsymbol{\Lambda}(\boldsymbol{x})$;

2. Initialize $\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0, \ddot{\boldsymbol{x}}_0, \boldsymbol{a}_0$, and $\boldsymbol{y}_0$;

3. Select the time step size h, the algorithmic parameters ρ_∞, the tolerance error ϵ, and the maximum number of iterations N;

4. Calculate integration constants:

$$\alpha=\frac{2\rho_\infty-1}{\rho_\infty+1},\ \delta=\frac{\rho_\infty}{\rho_\infty+1},\ \beta=\frac{1}{(\rho_\infty+1)^2},\ \gamma=\frac{3-\rho_\infty}{2(\rho_\infty+1)},$$

$$a_0=\frac{1-\alpha}{(1-\delta)\beta h^2},\ a_1=\frac{1-\alpha}{(1-\delta)\beta h},\ a_2=\frac{\delta}{1-\delta},\ a_3=\frac{1-\alpha-2\beta}{2(1-\delta)\beta},\ b_0=\frac{\gamma}{\beta h},\ b_1=\frac{\gamma-\beta}{\beta},$$

$$b_2=\frac{\gamma-2\beta}{2\beta}h,\ c_0=\frac{1-\delta}{1-\alpha},\ c_1=\frac{\delta}{1-\alpha},\ c_2=\frac{\alpha}{1-\alpha}.$$

B. For each time step

1. Predict $\boldsymbol{x}_{k+1}$

$$l=0,\ \boldsymbol{x}_{k+1}=\boldsymbol{x}_k+h\dot{\boldsymbol{x}}_k;$$

2. Update $\boldsymbol{y}_{k+1}, \ddot{\boldsymbol{x}}_{k+1}, \dot{\boldsymbol{x}}_{k+1}$ and $\boldsymbol{a}_{k+1}$:

$$\boldsymbol{y}_{k+1}=\boldsymbol{p}(\boldsymbol{x}_{k+1}),\ \ddot{\boldsymbol{x}}_{k+1}=a_0(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k)-a_1\dot{\boldsymbol{x}}_k-a_2\ddot{\boldsymbol{x}}_k-a_3\boldsymbol{a}_k,$$

$$\dot{\boldsymbol{x}}_{k+1}=b_0(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k)-b_1\dot{\boldsymbol{x}}_k-b_2\boldsymbol{a}_k,\ \boldsymbol{a}_{k+1}=c_0\ddot{\boldsymbol{x}}_{k+1}+c_1\ddot{\boldsymbol{x}}_k-c_2\boldsymbol{a}_k;$$

3. Compute $\boldsymbol{g}(\boldsymbol{x}_{k+1})$ and $\boldsymbol{G}(\boldsymbol{x}_{k+1})$:

$$\boldsymbol{g}(\boldsymbol{x}_{k+1})=\boldsymbol{M}\ddot{\boldsymbol{x}}_{k+1}+\boldsymbol{C}\dot{\boldsymbol{x}}_{k+1}+\boldsymbol{N}(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1},t_{k+1})-\boldsymbol{f}(t_{k+1}),$$

$$\boldsymbol{G}(\boldsymbol{x}_{k+1})=a_0\boldsymbol{M}+b_0\boldsymbol{C}+\boldsymbol{K}_x(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1},t_{k+1})$$
$$+\boldsymbol{K}_y(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1},t_{k+1})\boldsymbol{\Lambda}(\boldsymbol{x}_{k+1})\boldsymbol{W}(\boldsymbol{x}_{k+1});$$

4. Update $\boldsymbol{x}_{k+1}$:

$$l=l+1,\ \boldsymbol{x}_{k+1}=\boldsymbol{x}_{k+1}-\boldsymbol{G}(\boldsymbol{x}_{k+1})^{-1}\boldsymbol{g}(\boldsymbol{x}_{k+1});$$

5. If $l<N$ and $|\boldsymbol{g}|>\epsilon$, go to 2; If $l<N$ and $|\boldsymbol{g}|\le\epsilon$, go to next step; If $l\equiv N$, abort.

4.1 | Piecewise linear examples

4.1.1 | SDOF case

The generalized-α method is known as second-order accurate for smooth systems, but previous studies[34,35] show that discontinuities in velocity or acceleration caused by impacts or friction can reduce its accuracy to first-order. For this reason, it is necessary to discover the accuracy order of this method for systems with piecewise linear characteristics. Therefore, the SDOF system employed in Worden et al.[36] to model a cracked beam is solved to assess the convergence rate. The equation of motion can be written as follows:

$$m\ddot{x}+c\dot{x}+\begin{cases}kx, & x<0\\ \alpha kx, & x\ge 0\end{cases}=F\sin(\omega t). \tag{43}$$

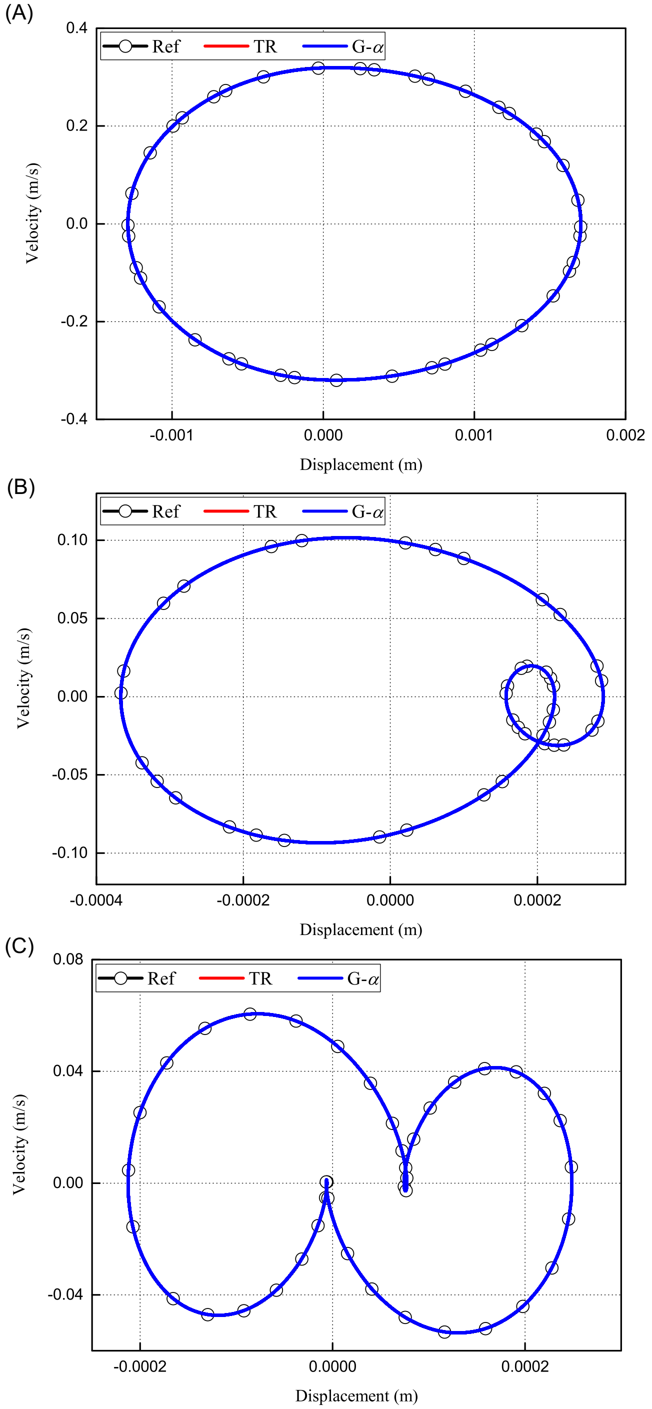

FIGURE 2 Phase diagrams of steady-state responses for the piecewise linear SDOF case: (A) $\omega = 214$ rad/s, (B) $\omega = 384$ rad/s, (C) $\omega = 561$ rad/s

With the projection function, Equation (43) can be rewritten as follows:

$$m\ddot{x}+c\dot{x}+kx+(\alpha-1)k\,\mathrm{proj}_{\mathbb{R}_0^+}(x)=F\sin(\omega t). \tag{44}$$

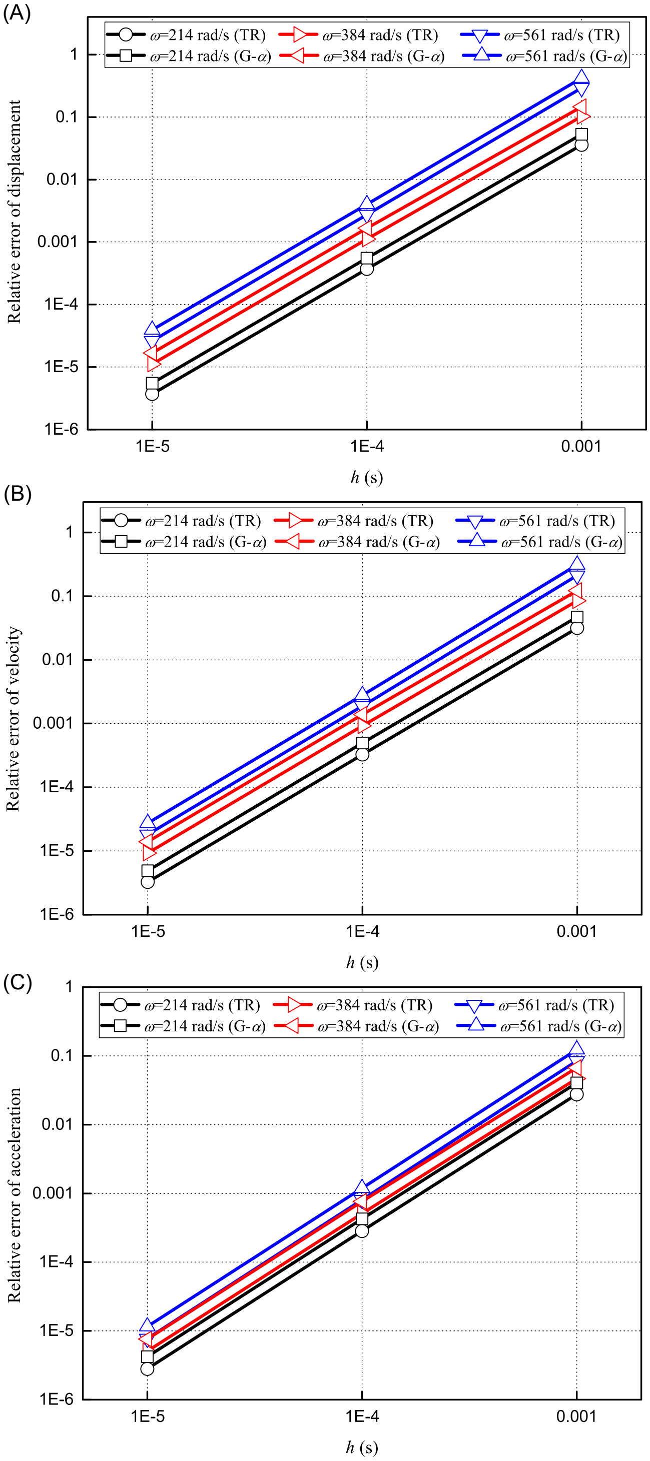

FIGURE 3 Relative errors versus the step size h: (A) displacement, (B) velocity, and (C) acceleration

The system parameters are set as $m = 0.6$ kg, $c = 2.6$ N · s/m, $k = 27346$ N/m, $\alpha = 0.6$, $F = 10$ N. The initial conditions are $x_0 = 10^{-5}$ m, $\dot{x}_0 = -\alpha k x_0 / c$. As discussed in Worden et al.,[36] three frequencies, $\omega = 214$, 384, and 561 rad/s, which will result in harmonic motion, and subharmonic motions of Periods 2 and 3, respectively, are selected. Using $h = T/2^{10}$, where $T = 2\pi/\omega$, the phase diagrams of steady-states responses within $[900T, 1000T]$ by the TR and the generalized-α method with $\rho_\infty = 0.5$ (G-α) are shown in Figure 2. Both methods accurately predict the solution: the numerical solutions almost overlap the reference one.

To show the convergence rate, the relative errors of displacement, velocity, and acceleration versus the step size h are plotted in Figure 3. Here the relative errors within [0, 10 s] are computed according to

$$\text{Relative error} = \frac{\sum_{k=1}^{n} \left| x_k - x(t_k) \right|}{\sum_{k=1}^{n} \left| x(t_k) \right|} \tag{45}$$

where n is the total number of steps, x_k and $x(t_k)$, respectively, denote the numerical and reference solutions. It can be observed that the integrators still hold second-order accuracy for displacement, velocity, and acceleration, so the piecewise characteristic does not cause the reduction of accuracy order. This is because all state variables in the recursive schemes are continuous, and no sudden change brings additional numerical errors. The conclusion can be naturally extended to general MDOF systems, which are equivalent to the superposition of a set of SDOF systems.

Besides, the results also show that TR ($\rho_\infty = 1$) is more accurate than the dissipative schemes, so it is employed in the following examples. Despite this, the dissipative schemes are very useful in large finite element systems and stiff problems, to filter out the high-frequency dynamics and make the solutions smoother and more stable.

4.1.2 | MDOF case

To compare the computational efficiency of the proposed scheme with that of LCP-based ones, a set of MDOF cases, as

$$\mathbf{M} = \begin{bmatrix} m & & & & \\ & m & & & \\ & & m & & \\ & & & \ddots & \\ & & & & m \end{bmatrix}, \tag{46a}$$

$$\mathbf{K}_x = \begin{bmatrix} 2k_1 & -k_1 & & & \\ -k_1 & 2k_1 & -k_1 & & \\ & -k_1 & \ddots & \ddots & \\ & & \ddots & 2k_1 & -k_1 \\ & & & -k_1 & k_1 \end{bmatrix}, \tag{46b}$$

$$\mathbf{K}_y = \begin{bmatrix} k_2 & & & & \\ & k_2 & & & \\ & & k_2 & & \\ & & & \ddots & \\ & & & & k_2 \end{bmatrix}, \tag{46c}$$

$$\mathbf{C} = 0.1\mathbf{M} + 0.01\mathbf{K}_x, \tag{46d}$$

$$\mathbf{f}(t) = \begin{bmatrix} A_0 + A\sin(2\pi f t) \\ A_0 + A\sin(2\pi f t) \\ A_0 + A\sin(2\pi f t) \\ \vdots \\ A_0 + A\sin(2\pi f t) \end{bmatrix}, \tag{46e}$$

TABLE 3 Required CPU time (s) for the piecewise linear MDOF cases of the proposed scheme and the LCP-based scheme

Number of piecewise characteristics	TR/Newton iteration	TR/Lemke's algorithm
10	0.1834	0.1024
50	1.0867	1.0722
100	4.0558	6.5924
1000	654.6712	22489.3308

$$\boldsymbol{p}(\boldsymbol{x}) = \begin{bmatrix} \mathrm{proj}_{\mathbb{R}_0^+}(x_1) \\ \mathrm{proj}_{\mathbb{R}_0^+}(x_2) \\ \mathrm{proj}_{\mathbb{R}_0^+}(x_3) \\ \vdots \\ \mathrm{proj}_{\mathbb{R}_0^+}(x_s) \end{bmatrix}, \tag{46f}$$

with different number of DOFs are solved. The number of piecewise characteristics is the same as the number of DOFs. Both schemes use TR as the time integrator. Other system parameters are assumed as $m = 1$ kg, $k_1 = 1000$ N/m, $k_2 = 2000$ N/m, $f = 100$ Hz, $A_0 = 0.1$ N, $A = 10$ N. The initial displacements satisfy static equilibrium.

Using $h = 0.001$ s, the transient response within [0, 1 s] is tracked, and the required CPU times for these two schemes are listed in Table 3. It follows that as the number of piecewise characteristic increases, the new method gradually exhibits a significant efficiency advantage. During the numerical experiments, the Newton iteration always reaches convergence in no more than five iterations, but the number of pivoting required by the Lemke algorithm increases rapidly as the number of piecewise characteristics increases. As shown in Murthy,[37] Lemke's algorithm needs exponential complexity in the worst case, which means that it requires up to 2^q pivoting steps to reach the final result. Therefore, the proposed scheme appears to be substantially more efficient when applied to systems with a large number of piecewise features.

Figure 4 shows the solutions of x_1, x_{500}, and x_{1000} when the number of piecewise characteristics is 1000. Since the errors of the numerical results mainly come from the accumulation of the local truncation error of the time integration scheme, which is TR used here, the proposed scheme and the LCP-based scheme show overlapping results. Therefore, using different solvers for solving the piecewise linear equations per step has little effect on the accuracy.

4.2 | Piecewise nonlinear examples

4.2.1 | Rolling mill

As shown in Figure 5, the rolling mill system,[2] a typical piecewise nonlinear problem, is considered. The governing equation can be written as follows:

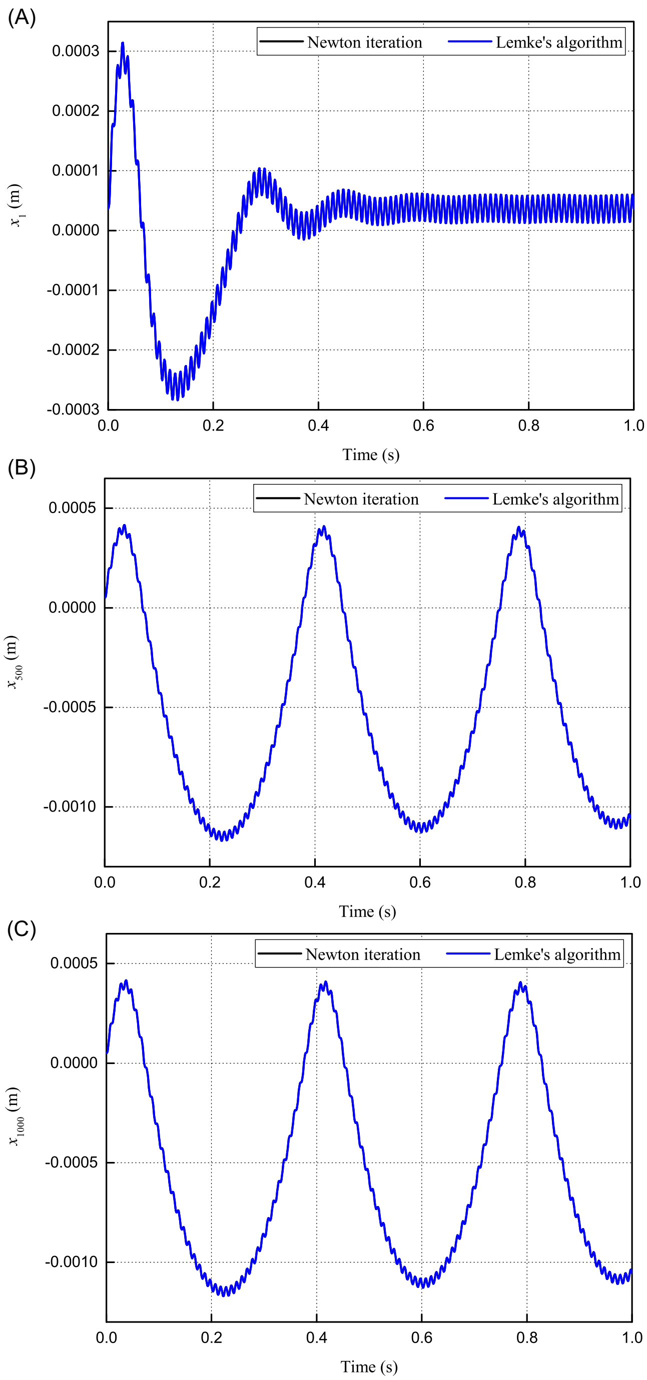

FIGURE 4 Numerical solutions for the piecewise linear MDOF case (A) x_1, (B) x_{500}, and (C) x_{1000}

$$m\ddot{x} + c\dot{x} + k_1 x + \begin{cases} k_2 x + k_3 x^3, & x \le 0 \\ k_4 x + k_5 x^3, & x > 0 \end{cases} = F\cos(\omega t). \tag{47}$$

The equivalent form of Equation (47) can be expressed as follows:

$$\begin{aligned} & m\ddot{x} + c\dot{x} + k_1 x + k_2 x + k_3 x^3 + (k_4 - k_2)\mathrm{proj}_{\mathbb{R}_0^+}(x) \\ & + (k_5 - k_3)\mathrm{proj}^3_{\mathbb{R}_0^+}(x) = F\cos(\omega t). \end{aligned} \tag{48}$$

The system parameters are set as $m = 2.16 \times 10^4$ kg, $c = 1.08 \times 10^4$ N · s/m,$k_1 = 1.17 \times 10^7$ N/m, $k_2 = 1.59 \times 10^8$ N/m, $k_3 = 1.59 \times 10^7$ N/

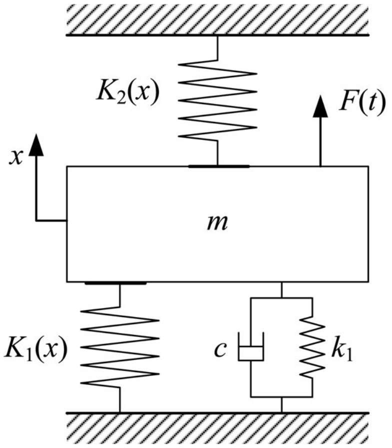

FIGURE 5 Dynamics model of a rolling mill

m,$k_4 = 5 \times 10^6$ N/m, $k_5 = -5 \times 10^5$ N/m and ω =2.0086 rad/s. As discussed in Si et al.,[2] three external force amplitudes, $F = 3.5, 4.6$, and 5 m, which will result in Period 2 motion, Period 3 motion, and chaotic motion, respectively, are employed. Using the proposed scheme with TR and $h = T/2^{10}$, the phase diagrams of steady-state response within $[900T, 1000T]$ are plotted in Figure 6. The numerical results are coincident with the reference ones, which demonstrates that the proposed scheme performs very well for the piecewise nonlinear example.

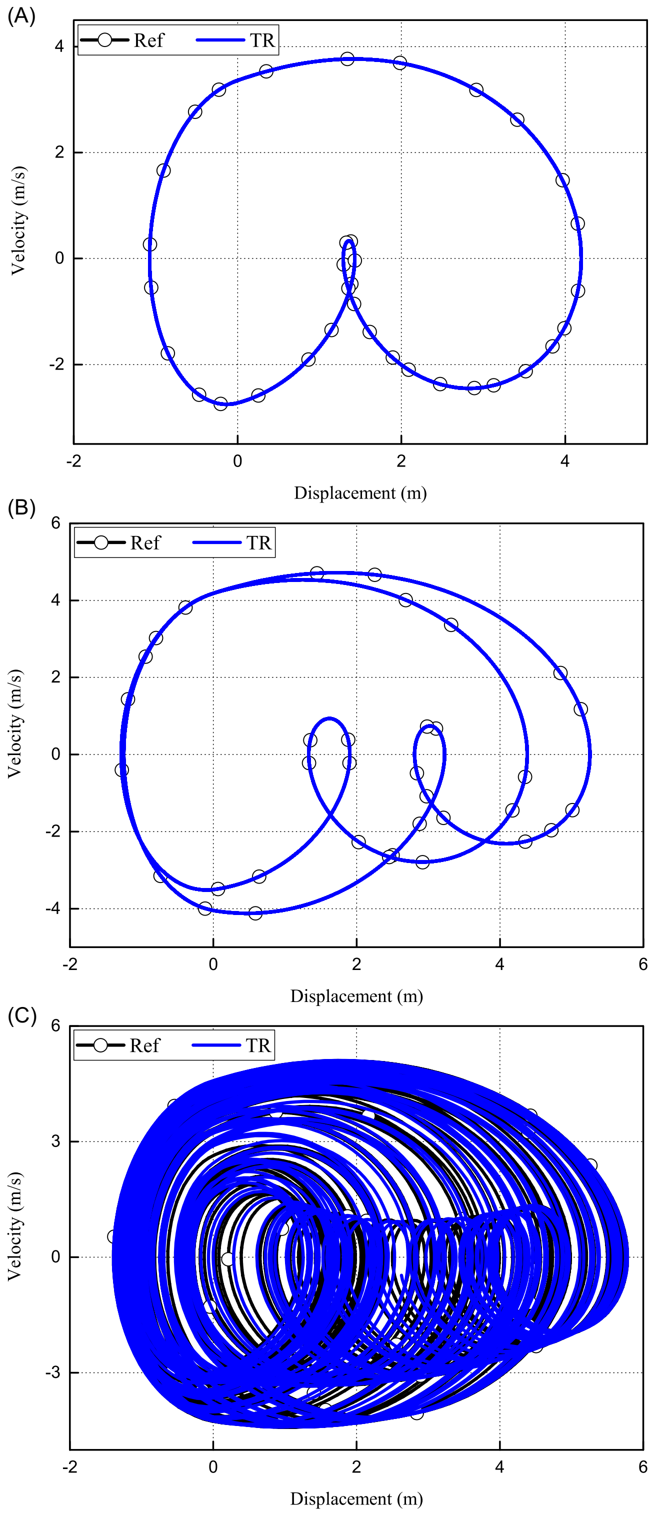

FIGURE 6 Phase diagrams of steady-state responses for the rolling mill: (A) $F = 3.5$ m, (B) $F = 4.6$ m, and (C) $F = 5$ m

4.2.2 | Piecewise-nonlinear time-varying oscillator

In this case, the piecewise-nonlinear time-varying oscillator investigated in Ma and Kahraman[38] is considered. The motion equation in dimensionless form is

$$\ddot{x} + 2\xi\dot{x} + r(t)N(x) = f(t) \tag{49}$$

where $\xi = 0.05$, $f(t) = 0.5$, and

$$r(t) = 1 + 0.3\sin\omega t + 0.15\sin 2\omega t + 0.1\sin 3\omega t, \tag{50a}$$

$$N(x) = \begin{cases} (x-1) + 0.1(x-1)^2 + 0.2(x-1)^3, & x > 1 \\ 0, & -1 \le x \le 1. \\ (x+1) - 0.1(x+1)^2 + 0.2(x+1)^3, & x < -1 \end{cases} \tag{50b}$$

Using the projection function, $k(x)$ can be converted to

$$\begin{aligned} N(x) = {} & (x - \mathrm{proj}_{[-1,1]}(x)) \\ & + 0.1\mathrm{proj}_{[-1,1]}(x)(x - \mathrm{proj}_{[-1,1]}(x))^2 \\ & + 0.2\mathrm{proj}^2_{[-1,1]}(x)(x - \mathrm{proj}_{[-1,1]}(x))^3. \end{aligned} \tag{51}$$

According to the conclusions in Ma and Kahraman,[38] different stable solutions may appear for a certain ω. When the stable solution satisfies $|x| > 1$, it is called no-impact (NI) motion; when $x_{\max} > 1$ and $-1 < x_{\min} < 1$ ($x_{\min} < -1$ and $-1 < x_{\max} < 1$), it is called single-side impact (SSI) motion; when $x_{\max} > 1$ and $x_{\min} < -1$, it is called double-side impact (DSI) motion.

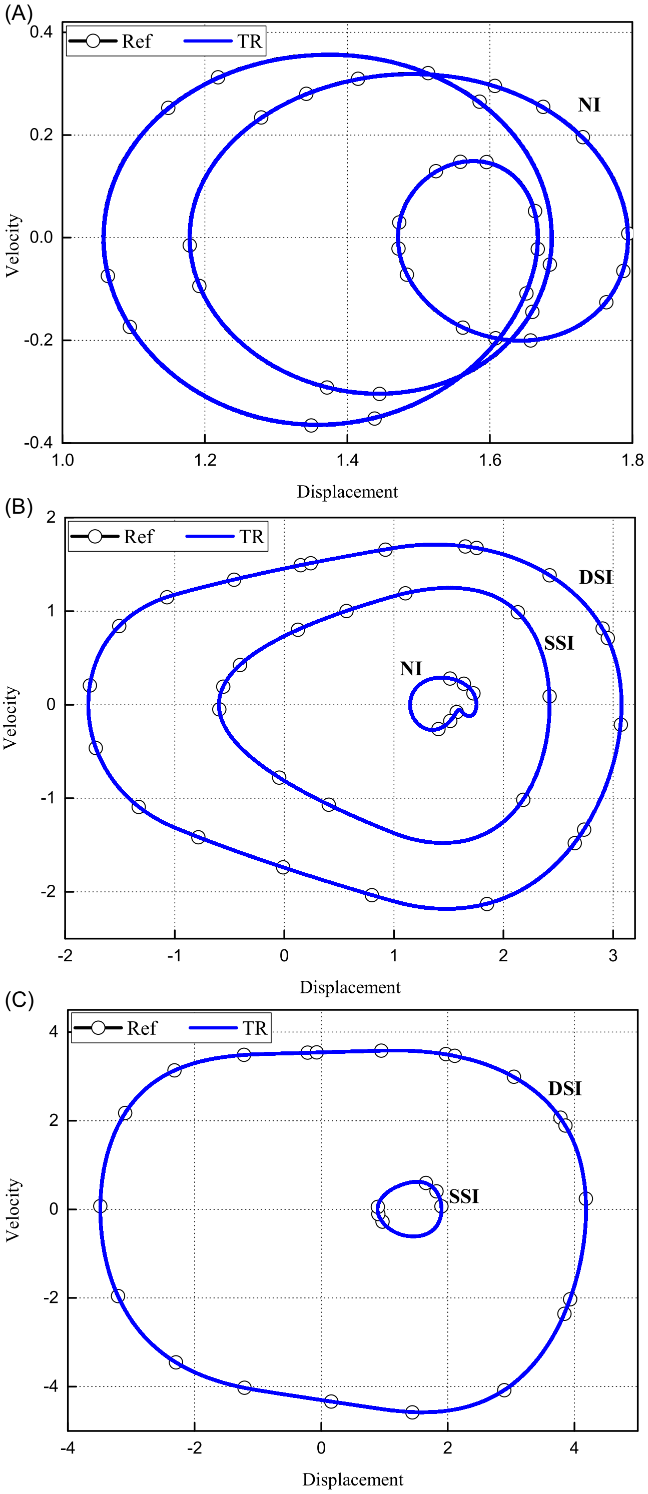

FIGURE 7 Phase diagrams of steady-state responses for the piecewise-nonlinear time-varying oscillator: (A) $\omega = 0.4$, (B) $\omega = 0.8$, and (C) $\omega = 1.2$

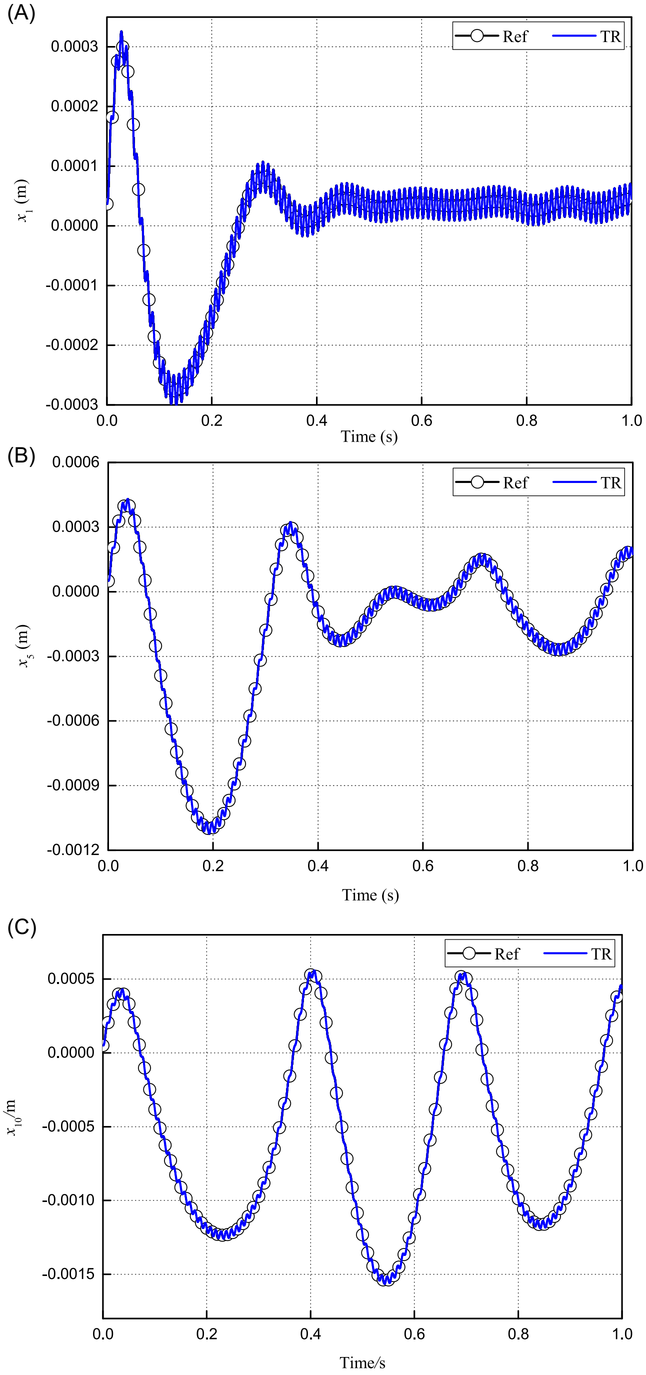

FIGURE 8 Numerical solutions for the piecewise nonlinear MDOF case (A) x_1, (B) x_5, and (C) x_{10}

Three values of ω, 0.4, 0.8, and 1.2, are employed. The corresponding phase diagrams of steady-state response in $[900T, 1000T]$ are computed using the proposed scheme with TR and $h = T/2^{10}$, as shown in Figure 7. By trying different initial conditions, all possible stable solutions for a certain ω are captured. The results are consistent with the analytical solutions[38] obtained using the harmonic balance method. For example, when $\omega = 0.8$, three stable motions, NI, SSI, and DSI, are all possible. Therefore, the implicit integration schemes proposed here can be used to study more complex piecewise nonlinear systems.

4.2.3 | MDOF case

The MDOF model in Section 4.1 is reused here, but the elastic force is replaced by

$$\begin{aligned}\boldsymbol{N} = &\begin{bmatrix} 2k_1 & -k_1 & & & \\ -k_1 & 2k_1 & -k_1 & & \\ & -k_1 & \ddots & \ddots & \\ & & \ddots & 2k_1 & -k_1 \\ & & & -k_1 & k_1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_s \end{bmatrix} \\ &+\begin{bmatrix} k_2 & & & & \\ & k_2 & & & \\ & & k_2 & & \\ & & & \ddots & \\ & & & & k_2 \end{bmatrix}\begin{bmatrix} \mathrm{proj}_{\mathbb{R}_0^+}(x_1) + \mathrm{proj}^3_{\mathbb{R}_0^+}(x_1) \\ \mathrm{proj}_{\mathbb{R}_0^+}(x_2) + \mathrm{proj}^3_{\mathbb{R}_0^+}(x_2) \\ \mathrm{proj}_{\mathbb{R}_0^+}(x_3) + \mathrm{proj}^3_{\mathbb{R}_0^+}(x_3) \\ \vdots \\ \mathrm{proj}_{\mathbb{R}_0^+}(x_s) + \mathrm{proj}^3_{\mathbb{R}_0^+}(x_s) \end{bmatrix}.\end{aligned} \tag{52}$$

Other sets and parameters remain unchanged. The initial displacements satisfy static equilibrium.

Using 10 DOFs and $h = 0.0001\,\mathrm{s}$, Figure 8 shows the solutions of x_1, x_5, and x_{10}. As can be seen, the numerical results agree very well with the reference ones. It demonstrates that the proposed scheme works very well also for piecewise nonlinear MDOF system. For more complex problems, the convergence of the Newton iteration greatly depends on the predicted value at each step. A small time step, an accurate predicted value, and appropriate algorithmic dissipation are all helpful to improve the convergence.

5 | CONCLUSIONS

This study provides a solution scheme using implicit integrators for solving the dynamic response of general piecewise linear and nonlinear systems. The proposed scheme employs the projection function to express the piecewise characteristics, instead of converting them into equivalent LCPs. Compared with the existing LCP-based scheme, the new scheme possesses two benefits. First, it can be applied to general piecewise nonlinear systems, because the discrete nonlinear equations containing the projection functions can be solved uniformly using the semismooth Newton method. Additionally, according to the numerical experiment, the proposed scheme is more efficient than the LCP-based scheme for the system with many piecewise features. The computational procedures are presented using the generalized-α method and the semismooth Newton iteration. Other implicit integrators can also be utilized in a similar manner. Numerical experiments demonstrate that the Newton iteration is more efficient than the Lemke algorithm, especially when the system has a lot of piecewise linear features. When applied to piecewise nonlinear problems, the numerical solutions of the new scheme present a high consistency with the reference ones.

ACKNOWLEDGMENT

The first and second authors acknowledge the financial support by the China Scholarship Council (No. 201906020240, No. 201906020241).

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Huimin Zhang http://orcid.org/0000-0003-3129-0170
Pierangelo Masarati http://orcid.org/0000-0002-9347-7654

REFERENCES

1. Ma YL, YuShu D, Wang DL. Nonlinear vibrational behavior of multi-body dynamical systems with bi-directional piecewise linear spring constraints. *J Vibration Control*. 2016;22(7):1808-1819.
2. Si C, Tian R, Feng J, Yang X. Bifurcation and chaos for piecewise nonlinear roll system of rolling mill. *Adv Mech Eng*. 2017;9(12):1687814017742313.
3. Joglekar DM, Mitra M. Nonlinear analysis of flexural wave propagation through 1D waveguides with a breathing crack. *J Sound Vibration*. 2015;344:242-257.
4. Boere SW, Shukla A, Fey RHB, Nijmeijer H. Severity of tip-out induced impacts in drive line systems with backlash. *J Computat Nonlin Dynamics*. 2010;5(2).
5. Yoon J-y, Lee H. Dynamic vibratory motion analysis of a multi-degree-of-freedom torsional system with strongly stiff non-linearities. *Proc Instit Mech Eng, Part C: J Mech Eng Sci*. 2015;229(8):1399-1414.
6. Nayfeh AH. *Introduction to Perturbation Techniques*. John Wiley & Sons; 2011.
7. Narimani A, Golnaraghi ME, Jazar GN. Frequency response of a piecewise linear vibration isolator. *J Vibration Control*. 2004;10(12):1775-1794.
8. Comparin RJ, Singh R. Frequency response characteristics of a multi-degree-of-freedom system with clearances. *J Sound Vibration*. 1990;42(1):101-124.
9. Tchamwa B, Conway T, Wielgosz C. An accurate explicit direct time integration method for computational structural dynamics. *Comput Struct*. 1999;398:77-84.
10. Chung J, Lee JM. A new family of explicit time integration methods for linear and non-linear structural dynamics. *Int J Num Methods Eng*. 1994;37(23):3961-3976.
11. Noh G, Bathe K-J. An explicit time integration scheme for the analysis of wave propagations. *Comput Struct*. 2013;129:178-193.
12. Newmark NM. A method of computation for structural dynamics. *J Eng Mech Div*. 1959;85(3):67-94.
13. Hilber HM, Hughes TJ, Taylor RL. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Eng Struct Dynam*. 1977;5(3):283-292.

14. Chung J, Hulbert G. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *J Appl Mech*. 1993;60:371-375.
15. Noh G, Bathe K-J. The Bathe time integration method with controllable spectral radius: the ρ_∞-Bathe method. *Comput Struct*. 2019; 212:299-310.
16. Masarati P, Lanz M, Mantegazza P. Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dinamics applications In: *XVI Congresso Nazionale AIDAA*; 2001:1-10.
17. Masarati P, Morandini M, Mantegazza P. An efficient formulation for general-purpose multibody/multiphysics analysis. *J Computat Nonlin Dyn*. 2014;9(4).
18. Zhang H, Zhang R, Masarati P. Improved second-order unconditionally stable schemes of linear multi-step and equivalent single-step integration methods. *Computat Mech*. 2020:1-25.
19. Zhang H, Zhang R, Xing Y, Masarati P. On the optimization of n-substep composite time integration methods. *Nonlin Dynam*. 2020; 102(3):1939-1962.
20. Tamma KK, Har J, Zhou X, Shimada M, Hoitink A. An overview and recent advances in vector and scalar formalisms: space/time discretizations in computational dynamics–a unified approach. *Archi Computat Methods Eng*. 2011;18(2):119-283.
21. Bi Q, Chen X, Kurths J, Zhang Z. Nonlinear behaviors as well as the mechanism in a piecewise-linear dynamical system with two time scales. *Nonlin Dynam*. 2016;85(4):2233-2245.
22. Yu S. An efficient computational method for vibration analysis of unsymmetric piecewise-linear dynamical systems with multiple degrees of freedom. *Nonlin Dynam*. 2013;71(3):493-504.
23. Fadaee M, Yu S. Dynamic behaviour of MDOF oscillators subjected to multiple visco-elastic contact constraints. *Int J Mechl Sci*. 2017; 131:218-226.
24. Cottle RW, Pang J-S, Stone RE. *The Linear Complementarity Problem*. SIAM; 2009.
25. Wood WL, Bossak M, Zienkiewicz OC. An alpha modification of Newmark's method. *Int J Num Method Eng*. 1980;15(10): 1562-1566.
26. He D, Gao Q, Zhong W. An efficient method for simulating the dynamic behavior of periodic structures with piecewise linearity. *Nonlin Dynam*. 2018;94(3):2059-2075.
27. Zhong WX, Williams FW. A precise time step integration method. *Proc Institut Mech Eng, Part C: J Mech Eng Sci*. 1994;208(6):427-430.
28. Fadaee M, Yu S. Vibrational behavior of MDOF oscillators subjected to multiple contact constraints. *J Mech Sci Technol*. 2017;31(4):1551-1560.
29. Zhang X, Kong X, Wen B, Zhao C. Numerical and experimental study on synchronization of two exciters in a nonlinear vibrating system with multiple resonant types. *Nonlin Dynam*. 2015;82(1-2):987-999.
30. Rockafellar RT. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math Operat Res*. 1976;1(2):97-116.
31. Alart P, Curnier A. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput Method Appl Mech Eng*. 1991;92(3):353-375.
32. Qi Liqun, Sun Jie. A nonsmooth version of Newton's method. *Math Program*. 1993;58(1-3):353-367.
33. Arnold M, Brüls O. Convergence of the generalized-α scheme for constrained mechanical systems. *Multibody Syst Dynam*. 2007;18(2): 185-202.
34. Chen Q-z, Acary V, Virlez G Brüls O. A nonsmooth generalized-α scheme for flexible multibody systems with unilateral constraints. *Int J Num Methods Eng*. 2013;96(8):487-511.
35. Zhang R, Yu Y, Wang Q, Wang Q. An improved implicit method for mechanical systems with set-valued friction. *Multibody Syst Dynam*. 2020;48(2):211-238.
36. Worden K, Farrar CR, Haywood J, Todd M. A review of nonlinear dynamics applications to structural health monitoring. *Struct Control Health Monitor*. 2008;15(4):540-567.
37. Murty KG. Computational complexity of complementary pivot methods. In *Complementarity and fixed point problems*. Springer; 1978:61-73.
38. Ma Q, Kahraman A. Period-one motions of a mechanical oscillator with periodically time-varying, piecewise-nonlinear stiffness. *J Sound Vibration*. 2005;284(3-5):893-914.

How to cite this article: Zhang H, Zhang R, Zanoni A, Masarati P. A generalized approach for implicit time integration of piecewise linear/nonlinear systems. *Int J Mech Syst Dyn*. 2021; 1:108-120. doi:10.1002/msd2.12007