Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 486171, 11 pages
http://dx.doi.org/10.1155/2014/486171

Research Article

Implicit Damping Iterative Algorithm to Solve Elastoplastic Static and Dynamic Equations

Huaifa Ma,[1,2] Jikai Zhou,[3] and Guoping Liang[4]

[1] *State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research, Beijing 100038, China*
[2] *Earthquake Engineering Research Center, China Institute of Water Resources and Hydropower Research, Beijing 100048, China*
[3] *College of Civil and Transportation Engineering, Hohai University, Jiangsu, Nanjing 210098, China*
[4] *Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China*

Correspondence should be addressed to Huaifa Ma; mahf@iwhr.com

Received 26 May 2014; Revised 6 August 2014; Accepted 6 August 2014; Published 31 August 2014

Academic Editor: Giuseppe Marino

This paper presents an implicit damping iterative algorithm to simultaneously solve equilibrium equations, yield function, and plastic flow equations, without requiring an explicit expression of elastoplastic stiffness matrices and local iteration for "return mapping" stresses to the yield surface. In addition, a damping factor is introduced to improve the stiffness matrix conformation in the nonlinear iterative process. The incremental iterative scheme and whole amount iterative scheme are derived to solve the dynamical and static and dynamical elastoplastic problems. To validate the proposed algorithms, computation procedures are designed and the numerical tests are implemented. The computational results verify the correctness and reliability of the proposed implicit iteration algorithms.

1. Introduction

The solution of the nonlinear problem using FEM eventually boils down to solving discretized nonlinear equations. FEM uses a series of modified linear approximate solutions to approach the solution of a nonlinear problem based on an iterative process. There are many methods available for solving nonlinear equations [1, 2], which may be divided into two categories: the direct iteration method and the Newton-Raphson method, or simply Newton method. The convergence rate of the direct iteration method is highly dependent on the choice of initial values. For problems with many degrees of freedom, instability may occur. In addition, for problems related to deformation history, the application of this method is quite limited. For these reasons, the direct method is rarely used. The Newton-Raphson method is probably the most popular method for solving nonlinear equations, and extensive researches and investigations have been performed. There are many derivatives of these methods, including the modified Newton-Raphson method, Quasi-Newton method, and incremental method (which can be regarded as the incremental form of the Newton-Raphson method). Researches on the Newton-Raphson method are focused on two aspects, namely, the computational efficiency and the stability of the solution. The Aitken acceleration method [3] and linear-search method [4, 5] are used in conjunction with Newton's method to reduce the number of computational iterations. Wempner [6] and Riks [7] proposed the arc-length method, and Forde and Stiemer [8], Müller [9], and others improved it. These improvements made it possible to analyze and solve the ultimate load of structure as well as the material weakening problems.

The elastoplastic problem is a typical nonlinear problem. In the loading and unloading processes, elastoplastic materials show different deformation characteristics: plastic hardening or weakening when loading, but elastic deformation when unloading. Two basic issues must be addressed for solving elastoplastic problems using FEM, namely, the linear scheme of nonlinear equations and its solution algorithm and the material constitutive relationship and its integration. The linear scheme of nonlinear equations and its solution process often depend on the material properties, load magnitude, loading history, and loading method. Therefore, the method of incremental stress-strain constitutive equations combined with iterative schemes is widely used, but the iterative operations will inevitably cause the effective stress to deviate from the yield surface, that is, the "drift" phenomenon.

At present, the Newton-Raphson method is still an effective way to solve plasticity problems. In particular for serious nonlinear processes involving material softening, it is more effective to combine the Newton-Raphson method with the arc-length method or displacement controlled method. In order to describe the complex deformation processes of elastoplastic problems, a variety of constitutive equation integration methods [10] have been proposed, and the most widely used one is the return-mapping algorithm [11–16]. The return-mapping algorithm is actually an elastic forecast and plastic amendment process which requires the local iteration to correct the plastic parameters in the iterative process for solving nonlinear equations and return the forecast stress to the yield surface. This paper proposes a new nonlinear iteration method, that is, the implicit damping iterative method. Different from existing methods, this new method is designed to simultaneously solve equilibrium equations, yield functions, and plastic flow equations of elastoplastic static and dynamical problems. In the iterative process of certain nonlinear problems, the global stiffness matrix of a finite element formulation tends to be ill-conditioned. In order to avoid the singularity of the coefficient matrix, the authors introduce a damping factor in the numerical iterative process, which is stable, fast, and easy to use and involves no local iteration for "return mapping" stress to the yield surfaces.

2. Basic Concept of the Implicit Damping Iterative Algorithm

The constitutive relation of elastoplastic material is usually given in the incremental form by the yield function and flow rule. Constitutive relations are generally expressed as an explicit form, namely, $\Delta\boldsymbol{\sigma} = \mathbf{D}^{\mathbf{ep}}\Delta\boldsymbol{\varepsilon}$, where $\Delta\boldsymbol{\sigma}$ and $\Delta\boldsymbol{\varepsilon}$ are the stress increment and the strain increment, respectively, and $\mathbf{D}^{\mathbf{ep}}$ is the elastoplastic matrix. The elastoplastic matrix will be adjusted according to the stress-strain state. Here the displacement (strain) will be obtained by simultaneously solving the equilibrium equations, yield function, and plastic flow equations. The explicit form of elastoplastic matrix is not required in solving the plastic strain based on the loading-unloading process and the current deformation.

Based on the above concepts, the implicit damping iterative method is proposed to solve the general material nonlinear problem. Supposing that $\boldsymbol{\varepsilon}_n$, $\boldsymbol{\varepsilon}_n^{\mathbf{p}}$, and $\boldsymbol{\sigma}_n$ are, respectively, the strain, plastic strain, and stress at the nth load step, and $\Delta\boldsymbol{\varepsilon}_n$, $\Delta\boldsymbol{\varepsilon}_n^{\mathbf{p}}$, and $\Delta\boldsymbol{\sigma}_n$ are, respectively, the strain increment, plastic strain increment, and stress increment at the $n+1$th load step, then the total strain, plastic strain, and stress are, respectively, expressed as:

$$\begin{aligned} \boldsymbol{\varepsilon}_{n+1} &= \boldsymbol{\varepsilon}_n + \Delta\boldsymbol{\varepsilon}_n; \\ \boldsymbol{\varepsilon}_{n+1}^{\mathbf{p}} &= \boldsymbol{\varepsilon}_n^{\mathbf{p}} + \Delta\boldsymbol{\varepsilon}_n^{\mathbf{p}}; \\ \boldsymbol{\sigma}_{n+1} &= \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_n. \end{aligned} \tag{1}$$

In the $n+1$th load step, the equations are

$$\begin{gathered} (\boldsymbol{\sigma}_{n+1}, \delta\boldsymbol{\varepsilon}) = (\mathbf{F}_{n+1}, \delta\mathbf{u}) \quad \text{Equalibrium equations,} \\ f(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\kappa}_{n+1}) = 0 \quad \text{Yield function,} \\ \Delta\boldsymbol{\varepsilon}_{n+1}^{\mathbf{p}} = \Delta\lambda_n \frac{\partial f}{\partial\boldsymbol{\sigma}} \quad \text{Flow rules.} \end{gathered} \tag{2a}$$

In the above formulas, the plastic potential function is the same as the yield function. Assuming that the yield function is a function of the stress and internal variable κ, then $\mathbf{F}$ is the external distributed load, and f_{n+1}, λ_{n+1}, and κ_{n+1}, respectively, represent the yield function value, plastic factor, and internal plastic variable value at $n+1$th load step. The notation $(*, *)$ in the first equation of (2a) and the formulas in the following section indicate the inner product of two functions, and the vectors and matrix are expressed in bold characters. In the iterative process, the value of yield function ($f_n \neq 0$) in the former load step is allowed to not be on the yield surface, that is, $f_n \neq 0$, but it can be automatically corrected by the second equation, so that the value of the yield function at $n+1$th load step meets the yield condition $f_{n+1} = 0$ and returns to the yield surface.

Equation (2a) is rewritten as follows:

$$\begin{gathered} (\boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_n, \delta\varepsilon) = (\mathbf{F}_{n+1}, \delta\mathbf{u}), \\ f(\boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_n, \kappa_n + \Delta\kappa_n) = 0, \\ \Delta\boldsymbol{\varepsilon}_{n+1}^{\mathbf{p}} = \Delta\lambda_n \frac{\partial f}{\partial\boldsymbol{\sigma}}. \end{gathered} \tag{2b}$$

Let $\Delta\kappa_n = \Delta\lambda_n H(\boldsymbol{\sigma}, \kappa)$, where $H(\boldsymbol{\sigma}, \kappa)$ is defined as the hardening function; thus the stress increment is expressed as follows:

$$\Delta\boldsymbol{\sigma}_n = \mathbf{D}^{\mathbf{e}}(\Delta\boldsymbol{\varepsilon}_n - \Delta\boldsymbol{\varepsilon}_n^{\mathbf{p}}) = \mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n - \Delta\lambda_n \mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}}. \tag{3}$$

Here $\mathbf{D}^{\mathbf{e}}$ is the elastic matrix. By using these notations, we then extend $f(\boldsymbol{\sigma}_{n+1}, \kappa_{n+1})$ at $(\boldsymbol{\sigma}_n, \kappa_n)$ to obtain

$$\begin{aligned} & f(\boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_n, \kappa_n + \Delta\kappa_n) \\ &\quad = f_n + \left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T \Delta\boldsymbol{\sigma}_n + \frac{\partial f}{\partial\kappa}\Delta\kappa_n \\ &\quad = \left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T \left(\mathbf{D}^{\mathbf{e}}\Delta\varepsilon_n - \Delta\lambda_n \mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}}\right) \\ &\qquad + \Delta\lambda_n \frac{\partial f}{\partial\kappa} H(\boldsymbol{\sigma}, \kappa) \\ &\quad = f_n + \left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T \mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n \\ &\qquad - \Delta\lambda_n \left\{\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T \mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}} - H(\boldsymbol{\sigma}, \kappa)\frac{\partial f}{\partial\kappa}\right\} \\ &\quad = 0. \end{aligned} \tag{4}$$

Here and in the following sections the superscript T in $(*)^T$ indicates the transposition of a matrix:

$$\text{Set}\quad A_0 = \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T \mathbf{D}^{\mathbf{e}} \frac{\partial f}{\partial \boldsymbol{\sigma}} - H(\boldsymbol{\sigma},\kappa)\frac{\partial f}{\partial \kappa}. \tag{5}$$

The plastic factor can be calculated by the following formula:

$$\Delta\lambda^n = \frac{1}{A_0}\left[\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T \mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n + f_n\right]. \tag{6}$$

Substituting (6) into (3), we obtain

$$\begin{aligned}\Delta\boldsymbol{\sigma}_n &= \mathbf{D}^{\mathbf{e}}\left(\Delta\boldsymbol{\varepsilon}_n - \Delta\boldsymbol{\varepsilon}_n^{\mathbf{p}}\right)\\ &= \mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n - \Delta\lambda_n\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}\\ &= \mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n - \frac{1}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}\left(\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n + f_n\right)\\ &= \left[\mathbf{D}^{\mathbf{e}} - \frac{1}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\right]\Delta\boldsymbol{\varepsilon}_n - \frac{f_n}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}.\end{aligned} \tag{7}$$

That is,

$$\Delta\boldsymbol{\sigma}_n = \mathbf{D}^{\mathbf{ep}}\Delta\boldsymbol{\varepsilon}_n - \frac{f_n}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}, \tag{8}$$

where

$$\mathbf{D}^{\mathbf{ep}} = \left[\mathbf{D}^{\mathbf{e}} - \frac{1}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\right]. \tag{9}$$

As usual, $\mathbf{D}^{\mathbf{ep}}$ is referred to as the elastoplastic matrix. If $f_n = 0$, the stress increments are given by $\Delta\boldsymbol{\sigma}_n = \mathbf{D}^{\mathbf{ep}}\Delta\boldsymbol{\varepsilon}_n$.

The first term in (2b) can be expressed as follows:

$$\left(\Delta\boldsymbol{\sigma}_n, \delta\boldsymbol{\varepsilon}\right) = \left(\mathbf{F}_{n+1}, \delta\mathbf{u}\right) - \left(\boldsymbol{\sigma}_n, \delta\boldsymbol{\varepsilon}\right). \tag{10}$$

Substituting (7) into the above equation, we obtain

$$\begin{aligned}\left(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n, \delta\boldsymbol{\varepsilon}\right) - \left(\Delta\lambda_n\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}, \delta\boldsymbol{\varepsilon}\right)&\\ = \left(\mathbf{F}_{n+1}, \delta\mathbf{u}\right) - \left(\boldsymbol{\sigma}_n, \delta\boldsymbol{\varepsilon}\right)&.\end{aligned} \tag{11}$$

Eliminating $\Delta\lambda_n$ by (6), the basic equation to solve the static elastoplastic problem can be expressed in the integral weak forms as follows:

$$\begin{aligned}&\left(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n, \delta\boldsymbol{\varepsilon}\right) - \left(\frac{1}{A_0}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n, \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\ &\quad= \left(\frac{1}{A_0}f_n, \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right) + \left(\mathbf{F}_{n+1}, \delta\mathbf{u}\right) - \left(\boldsymbol{\sigma}_n, \delta\boldsymbol{\varepsilon}\right).\end{aligned} \tag{12a}$$

Equation (12a) can be rewritten explicitly as follows:

$$\begin{aligned}\left(\mathbf{D}^{\mathbf{ep}}\Delta\boldsymbol{\varepsilon}_n, \delta\boldsymbol{\varepsilon}\right) &= \left(\frac{f_n}{A_0}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}, \delta\boldsymbol{\varepsilon}\right)\\ &\quad+ \left(\mathbf{F}_{n+1}, \delta\mathbf{u}\right) - \left(\boldsymbol{\sigma}_n, \delta\boldsymbol{\varepsilon}\right).\end{aligned} \tag{12b}$$

In accordance with Newton iteration method, at the $k+1$th iterative step the nonlinear equation $f(x^{(k+1)}) = 0$, after linearization at $x^{(k)}$, $f(x^{(k)}) + df(x^{(k)})/dx\Delta x^{(k)} = 0$.

Here $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$, when $\Delta x^{(k)} \to 0$, $x^{(k+1)}$ is the solution of the problem. Substituting $df(x^{(k)})/dx$ by coefficients A, the equation $f(x^{(k)}) + df(x^{(k)})/dx\Delta x^{(k)} = 0$ will be changed into $f(x^{(k)}) + A\Delta x^{(k)} = 0$, as long as $\Delta x^{(k)} \to 0$, and $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$; $x^{(k+1)}$ also tends to the solution of the problem. To avoid the singularity of the coefficient matrix of the equation, the damping factor μ is introduced, so that $A = A_0 + \mu$; that is,

$$A = \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}} - \frac{\partial f}{\partial \kappa}H + \mu. \tag{13}$$

The value of μ can be modified according to actual situations. We attempt to keep the absolute value of $|A|$ far from zero but not so high that it will affect the iterative convergence rate.

(1) If

$$A_0 = \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}} - \frac{\partial f}{\partial \kappa}H > 0, \tag{14}$$

that is,

$$\begin{gathered}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}} > \frac{\partial f}{\partial \kappa}H,\\ \text{set}\quad \mu = C_1\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}};\end{gathered} \tag{15}$$

(2) if

$$A_0 = \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}} - \frac{\partial f}{\partial \kappa}H < 0; \tag{16}$$

that is,

$$\begin{gathered}\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}} < \frac{\partial f}{\partial \kappa}H,\\ \text{set}\quad \mu = -C_2\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}.\end{gathered} \tag{17}$$

Here C_1 and C_2 are all real numbers greater than zero. When A_0 is equal to or approach zero, then μ can be given a greater number. In this paper, C_1 and C_2 are taken as 2, and the damping factor μ is determined by the following formula:

$$\mu = \begin{cases} 2\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}} & \frac{\partial f}{\partial \kappa}H < \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}\\ -2\left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}} & \frac{\partial f}{\partial \kappa}H > \left(\frac{\partial f}{\partial \boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial \boldsymbol{\sigma}}. \end{cases} \tag{18}$$

The constants in (18) are dependent on actual cases. The forms of the damping factors may be different from the yield function and flow rules. It is not difficult to extend this implicit damping iterative method to solve general material nonlinear problems with stiff damage weakening (degradation), but the representation of the yield function and flow rule may be different.

3. Iterative Schemes

3.1. Construction of Iterative Formulas. The unknowns in (12a) are displacement increments. In the elastic loading or plastic unloading, the plastic factor $\Delta\lambda_n$ is less than zero; then (12a) is transformed into the general linear elastic equation: $(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n, \delta\boldsymbol{\varepsilon}) = (\mathbf{F}_{n+1}, \delta\mathbf{u}) - (\boldsymbol{\sigma}_n, \delta\boldsymbol{\varepsilon})$; when the stress-strain falls into the plastic zone, the plastic factor $\Delta\lambda_n$ is greater than zero. Therefore, one may solve elastic equations if in the iterative process the plasticity actor value is negative or the plastic equation is positive.

The following section introduces two iterative methods: one is an incremental iterative method, and the other is the full amount displacements of iteration.

There are two algorithms for displacement increment iteration: (1) the iteration algorithm based on independent unknowns of strain increments and plastic factor and (2) the numerical solution based on the only unknowns of strain increment.

3.1.1. Increment Iterative Scheme (I). At each load step, the external load remains constant, by (12a) after inducing the damping factor at the kth iteration:

$$\begin{aligned}&\left(\mathbf{D}^{\mathbf{e}}\Delta\Delta\boldsymbol{\varepsilon}^{(k)},\delta\boldsymbol{\varepsilon}\right)-\left(\frac{1}{A}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\Delta\Delta\boldsymbol{\varepsilon}^{(k)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\&\quad=\left(\frac{1}{A}f_{n+1}^{(k-1)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\&\qquad+\left(\mathbf{F}_{n+1},\delta\mathbf{u}\right)-\left(\boldsymbol{\sigma}_{n+1}^{(k-1)},\delta\boldsymbol{\varepsilon}\right).\end{aligned}\tag{19}$$

We then obtain the displacement subincrements $\Delta\Delta\mathbf{u}^{(k)}$ by the iterative equation (19) at the kth iteration. In fact, (19) can be converted to the following equivalent form:

$$\begin{aligned}&\left(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k)},\delta\boldsymbol{\varepsilon}\right)-\left(\frac{1}{A}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\&\quad=\left(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k-1)},\delta\boldsymbol{\varepsilon}\right)\\&\qquad-\left(\frac{1}{A}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k-1)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\&\qquad+\left(\frac{1}{A}f_{n+1}^{(k-1)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\&\qquad+\left(\mathbf{F}_{n+1},\delta\mathbf{u}\right)-\left(\boldsymbol{\sigma}_{n+1}^{(k-1)},\delta\boldsymbol{\varepsilon}\right).\end{aligned}\tag{20}$$

From iterative equation (20), we can directly obtain the total displacement increments $\Delta\mathbf{u}_n^{(k)}$ and $\Delta\boldsymbol{\varepsilon}_n^{(k)}$ at the current load step. We then compute stresses $\boldsymbol{\sigma}_{n+1}^{(k)}$ by means of the following formula:

$$\begin{aligned}\boldsymbol{\sigma}_{n+1}^{(k)}&=\boldsymbol{\sigma}_n+\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k)}-\frac{1}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k)}\\&\quad-\frac{1}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}}f_n\left(\boldsymbol{\sigma}_n,\kappa_n\right).\end{aligned}\tag{21}$$

Note that when calculating the stresses by (21), A_0 is used, rather than A, to compute the real stresses. We then compute $\partial f/\partial\boldsymbol{\sigma}$ by difference, and let $\partial f/\partial\sigma_{ij} \approx [f(\sigma_{ij}+\delta)-f(\sigma_{ij})]/\delta$. In this paper δ takes the 10^{-3} of the material ultimate stress. When the iteration converges, $\Delta\boldsymbol{\varepsilon}_n^{(k)} \rightarrow \Delta\boldsymbol{\varepsilon}_n$, $\boldsymbol{\varepsilon}_{n+1}^{(k)} = \boldsymbol{\varepsilon}_{n+1}^{(k-1)} + \Delta\boldsymbol{\varepsilon}_n^{(k)} \rightarrow \boldsymbol{\varepsilon}_{n+1}$, and $\boldsymbol{\sigma}_{n+1}^{(k)} \rightarrow \boldsymbol{\sigma}_{n+1}$, the external loads are in equilibrium with internal forces, and the equilibrium equations tend to the following form:

$$\left(\mathbf{F}_{n+1},\delta\mathbf{u}\right)-\left(\boldsymbol{\sigma}_{n+1},\delta\boldsymbol{\varepsilon}\right)=0.\tag{22}$$

The procedure of increment iterative scheme (I) is listed in Box 1.

3.1.2. Increment Iterative Scheme (II). Equation (6) is written in the form of weak form and associated with (11):

$$\begin{aligned}&\left(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n,\delta\boldsymbol{\varepsilon}\right)-\left(\Delta\lambda_n\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}},\delta\boldsymbol{\varepsilon}\right)=\left(\mathbf{F}_{n+1},\delta\mathbf{u}\right)-\left(\boldsymbol{\sigma}_n,\delta\boldsymbol{\varepsilon}\right),\\&\left(A_0\Delta\lambda_n,\delta\lambda\right)-\left(\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n,\delta\lambda\right)=\left(f_n,\delta\lambda\right).\end{aligned}\tag{23}$$

The strain increments $\Delta\mathbf{u}$ and the plastic factor $\Delta\lambda_n$ are taken as an independent unknown, and these are simultaneously solved. The iterative scheme's integral weak form is as follows:

$$\begin{aligned}&\left(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k)},\delta\boldsymbol{\varepsilon}\right)-\left(\Delta\lambda_n^{(k)}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}},\delta\boldsymbol{\varepsilon}\right)\\&\qquad+\left(A\Delta\lambda_n^{(k)},\delta\lambda\right)-\left(\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k)},\delta\lambda\right)\\&\quad=\left(\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k-1)},\delta\boldsymbol{\varepsilon}\right)-\left(\Delta\lambda_n^{(k-1)}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}},\delta\boldsymbol{\varepsilon}\right)\\&\qquad+\left(A\Delta\lambda_n^{(k-1)},\delta\lambda\right)-\left(\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k-1)},\delta\lambda\right)\\&\qquad+\left(f_{n+1}^{(k-1)},\delta\lambda\right)+\left(\mathbf{F}_{n+1},\delta\mathbf{u}\right)-\left(\boldsymbol{\sigma}_{n+1}^{(k-1)},\delta\boldsymbol{\varepsilon}\right).\end{aligned}\tag{24}$$

In the iterative process of (24), if the plastic factor $\Delta\lambda_n^{(k)}$ is less than zero, let it be equal to zero, and then calculate the stress increments according to the elastic matrix. In this paper, if the calculated A is very small (e.g., 10^{-10}), a large number is set (e.g., 10^{20}). The main difference between the iteration scheme (II) and the program flow of the previous iteration (I) lies in the processing of the plastic factor.

(i) The given initial values include $\mathbf{u}_0$, $\Delta\mathbf{u}_0$, $\boldsymbol{\sigma}_0$, κ_0, material and iterative controlling parameters;

(ii) According to the acting loads $\mathbf{F}_{n+1}$, execute iteration within $n+1$th loading step, to find displacement increments $\Delta\mathbf{u}_n^{(k)}$ by (20) until achieving convergence:

(1) For the kth iteration:

We assume $\mathbf{u}_{n+1}^{(0)} = \mathbf{u}_n$; $\mathbf{u}_{n+1}^{(0)} = \boldsymbol{\sigma}_n$; and $\Delta\mathbf{u}_{n+1}^{(0)} = \mathbf{0}$, then execute the iterative loops by (20):

① Compute the strain increments $\Delta\boldsymbol{\varepsilon}_n^{(k)}$, stresses $\boldsymbol{\sigma}_{n+1}^{(k)}$, yield function value $f_{n+1}^{(k)}$, $\partial f/\partial\boldsymbol{\sigma}$ and the other coefficients.

② Compute the plastic parameter $d\lambda = (\partial f/\partial\boldsymbol{\sigma})^T\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}_n^{(k)} + f_n(\boldsymbol{\sigma}_n, \kappa_n)$, and determine the stress state:

If $d\lambda > 0$, the stress is located in the plastic area; if $d\lambda \le 0$, then the stress is in the elastic area.

③ Update stiffness matrix by the following regulations:

(a) In the elastic area, eliminate the terms containing A in (20).

(b) In the plastic area, compute A by (13) and (18).

④ Solve $\Delta\mathbf{u}_n^{(k)}$ by (20), calculate $err = \left\|\Delta\mathbf{u}_n^{(k)} - \Delta\mathbf{u}_n^{(k-1)}\right\|$.

⑤ Determine whether or not err is less than $\alpha\left\|\Delta\mathbf{u}_n^{(k)}\right\|$, ($\alpha = 10^{-9}$ in this paper).

If not, repeat Steps ①–④, and conduct the $k+1$th iteration, until achieving convergence. Then return to Step (2).

(2) Re-compute the plastic parameter $d\lambda$ by the displacements at the final iteration, determine the stress state, compute the total stresses $\boldsymbol{\sigma}_{n+1}$ by (21), as well as inner variable κ_{n+1}, plastic strains, and so on.

(iii) Repeat Step (ii), continue the computation of the next load step, determine whether or not failure takes place, and obtain the ultimate load.

Box 1: Increment iterative scheme (I).

For the kth iteration:

We assume that $\mathbf{u}_{n+1}^{(0)} = \mathbf{u}_n$, $\boldsymbol{\sigma}_{n+1}^{(0)} = \boldsymbol{\sigma}_n$ and $\Delta\mathbf{u}_{n+1}^{(0)} = \mathbf{0}$, then execute the iterative loops by (24):

① Compute the total strain increments $\Delta\boldsymbol{\varepsilon}_n^{(k)}$, stresses $\boldsymbol{\sigma}_{n+1}^{(k)}$, yield function value $f_{n+1}^{(k)}$, $\partial f/\partial\boldsymbol{\sigma}$ and the other coefficients.

② Compute A by (13) and (18). If A is very small (e.g. $<10^{-10}$), let A equal a high number (e.g. 10^{20}), then obtain $\Delta\mathbf{u}_n^{(k)}$ and $\Delta\lambda_n^{(k)}$ by (24).

③ Determine the stress state: If $\Delta\lambda_n^{(k)} < 0$, let $\Delta\lambda_n^{(k)} = 0$.

④ Calculate the errors of $\Delta\mathbf{u}_n^{(k)}$ and $\Delta\lambda_n^{(k)}$, and see if they satisfy the tolerance error requirement.

⑤ If not, repeat Steps ①–④, then conduct the $k+1$th iteration, until achieving convergence.

Box 2: Increment iterative scheme (II).

Corresponding to Box 1, in Box 2 we only list a portion of the nonlinear iterative process.

Equation (20) is rewritten as the total strain (displacement) iterative formula for a structural system to withstand the current loads:

$$\begin{aligned}\left(\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}_{n+1}^{(k)},\delta\boldsymbol{\varepsilon}\right)&-\left(\frac{1}{A}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}_{n+1}^{(k)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\ &=\left(\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}_{n+1}^{(k-1)},\delta\boldsymbol{\varepsilon}\right)\\ &\quad-\left(\frac{1}{A}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}_{n+1}^{(k-1)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\ &\quad+\left(\frac{1}{A}f_{n+1}^{(k-1)},\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^{T}\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\ &\quad+\left(\mathbf{F}_{n+1},\delta\mathbf{u}\right)-\left(\boldsymbol{\sigma}_{n+1}^{(k-1)},\delta\boldsymbol{\varepsilon}\right).\end{aligned}\tag{25}$$

Clearly, (25) is essentially another form of the iteration equation (20). $\boldsymbol{\varepsilon}_{n+1}^{(k-1)}$ and $\boldsymbol{\varepsilon}_{n+1}^{(k)}$, respectively, represent the total strain components of the $k-1$th and kth with the current load $\mathbf{F}_{n+1}$. The total displacements $\mathbf{u}_{n+1}^{(k)}$ can be solved at the kth iterative step by (25). We then compute the displacement increments $\Delta\mathbf{u}_n^{(k)} = \mathbf{u}_{n+1}^{(k)} - \mathbf{u}_n$ and strain increments $\Delta\boldsymbol{\varepsilon}_n^{(k)} = \boldsymbol{\varepsilon}_{n+1}^{(k)} - \boldsymbol{\varepsilon}_n$ at the current load step and then substitute $\Delta\boldsymbol{\varepsilon}_n^{(k)}$ into (21), after which the total stress components of the current iteration step can be obtained. When the iteration converges, that is, $\boldsymbol{\varepsilon}_{n+1}^{(k)} \to \boldsymbol{\varepsilon}_{n+1}^{(k-1)}$, $f_{n+1}^{(k)} \to 0$, and $\boldsymbol{\sigma}_{n+1}^{(k)} \to \boldsymbol{\sigma}_{n+1}$, then the same equilibrium as that in (22) is reached. The nonlinear iteration procedure of whole amount iterative scheme is almost the same as that of the increment iterative scheme (I).

In (20) the displacement increments are taken as independent unknowns, and in (24) the plastic factors also act as independent unknowns. However, both of these incremental iterative formulas are completely equivalent, except for some slight differences in their expressions. Equation (25), the total displacement iteration scheme of displacements, is also equivalent to the incremental iteration schemes, by which the total strains (displacements) can be directly obtained under the current loads.

3.2. FEPG Script Codes.

In order to verify the above iteration algorithm, the FEM FORTRAN program for the above iterative algorithm is developed based on the Finite Element Program Generator (FEPG) [17, 18]. The essence of FEPG lies in adopting both component programming techniques and human intelligence technology to automatically generate finite element source code from the given partial differential equation (PDE) codes and algorithm expressions (nonlinear finite element (NFE)). FEPG is based on the unified mathematical theory for FEM and its intrinsic rule, and the program generation is similar to the deduction process of mathematical formulas. The finite element computation program is composed of five components, namely, START, BFT, SOLV, E, and U, among which E and U are automatically generated according to the expressions given by users, while the other three components, which do not change with the expressions, are given by the system.

According to the language rules of FEPG, the integral weak equations (20), (24), and (25) are written by the PDE; the iterative algorithm and loops are written by NFE codes; and the coupling relation of the displacement with stress fields is described by the Generate Command stream file and NFE codes (GCN) and either Generate IO (GIO) or Multidisciplinary Input (MDI) codes. Running the GIO or MDI command can generate the source codes for the calculation procedures [17]. In the following section, these source codes are executed to assess the iterative procedures previously established.

3.3. Validation of Iteration Method. We take a circular tunnel excavation problem as a testing example. Tunnel excavation will cause rock stress redistribution. The tunnel surrounding may enter the plastic state. Similar to the thick-walled cylinder, the out boundary will be subject to uniform external pressure, as shown in Figure 1. Adopting the Mohr-Coulomb rule, here we let the cohesive strength be equal to 1.0 kPa and let the circular orifice radius be 1.0 m, and then the expansion processes of the plastic zone are simulated. Taking the friction coefficients f as 0.2 and 0.4, the radius of the elastoplastic interface is computed. The curves of plastic radii via pressure are given in Figure 2. As a whole, the plastic zone radii increase with confining pressure and agree well with the theoretical solutions. When friction coefficient is 0.4, especially, the computational results coincide almost exactly with the theoretical solution.

It is worth noting that far away from the outer boundary of the rectangle area, the plastic area is in good agreement with the theoretical solutions, but when the plastic area reaches the outer edge, it no longer meets the infinite boundary conditions of the theoretical solution, which may result in a larger calculated error.

4. Solving the Nonlinear Dynamic Equations

4.1. Basic Equations of Dynamic Problems. For dynamic problems, the various physical quantities of the object are a function of time. At time t the strain, plastic strain, and stress are, respectively, expressed as $\boldsymbol{\varepsilon}_t$, $\boldsymbol{\varepsilon}_t^{\mathbf{p}}$, $\boldsymbol{\sigma}_t$, the strain increment $\Delta\boldsymbol{\varepsilon}_t$, the plastic strain increment $\Delta\boldsymbol{\varepsilon}_t^{\mathbf{p}}$, and stress increment $\Delta\boldsymbol{\sigma}_t$. At time $t+\Delta t$, a total strain, plastic strain, and stress are expressed as

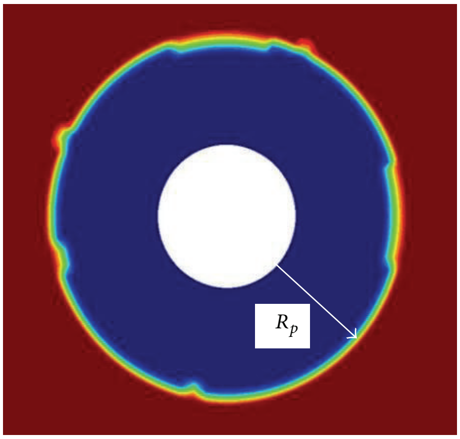


FIGURE 1: Plastic area.

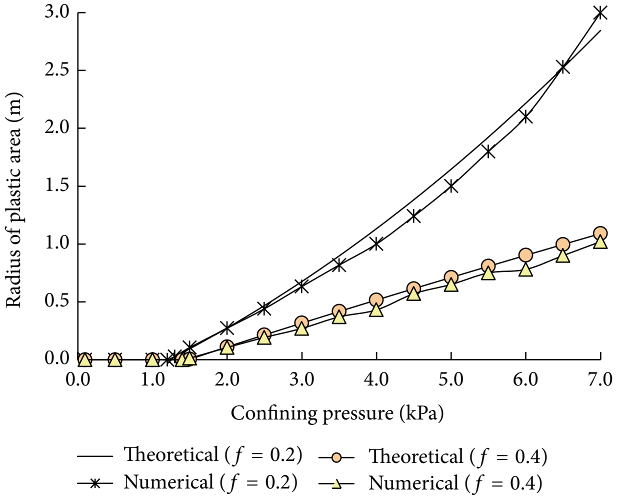


FIGURE 2: Plastic radii via pressure.

$$\begin{aligned}\boldsymbol{\varepsilon}_{t+\Delta t} &= \boldsymbol{\varepsilon}_t + \Delta\boldsymbol{\varepsilon}_t;\\ \boldsymbol{\varepsilon}_{t+\Delta t}^{\mathbf{p}} &= \boldsymbol{\varepsilon}_t^{\mathbf{p}} + \Delta\boldsymbol{\varepsilon}_t^{\mathbf{p}};\\ \boldsymbol{\sigma}_{t+\Delta t} &= \boldsymbol{\sigma}_t + \Delta\boldsymbol{\sigma}_t.\end{aligned} \tag{26}$$

By analogy with (2a) and (2b), the dynamic equations at time $t+\Delta t$ are

$$\begin{aligned}\left(\Delta\boldsymbol{\sigma}_t, \delta\boldsymbol{\varepsilon}\right) + \left(\rho\ddot{\mathbf{u}}_{t+\Delta t}, \delta\mathbf{u}\right) + \left(\mu\dot{\mathbf{u}}_{t+\Delta t}, \delta\mathbf{u}\right)\\ = \left(\mathbf{F}_{t+\Delta t}, \delta\mathbf{u}\right) - \left(\boldsymbol{\sigma}^{(t)}, \delta\boldsymbol{\varepsilon}\right),\\ f_{t+\Delta t} = 0,\\ \Delta\boldsymbol{\varepsilon}^{\mathbf{p}} = \Delta\lambda_t \frac{\partial f}{\partial \boldsymbol{\sigma}}.\end{aligned} \tag{27}$$

Here ρ is the mass density; μ is the damping coefficient, whose meaning is different from that of (13); $\ddot{\mathbf{u}}$ and $\dot{\mathbf{u}}$, respectively, represent the components of the acceleration and velocities.

```
(i) Given time interval $\Delta t$, the integration constants; material and iterative controlling parameters; initial values,
    including $\mathbf{u}_0, \dot{\mathbf{u}}_0, \ddot{\mathbf{u}}_0, \boldsymbol{\sigma}_0, \kappa_0$, and their increments.
(ii) According to acting loads $\mathbf{F}_{t+\Delta t}$ or time-displacement history, execute iteration within and $t + \Delta t$, to find total
    displacements $\mathbf{u}^{(k)}_{t+\Delta t}$ by (31) until achieving convergence:
(1) For the $k$th nonlinear iteration (not update $\dot{\mathbf{u}}_{t+\Delta t}$ and $\ddot{\mathbf{u}}_{t+\Delta t}$ for time being in nonlinear iterative loops):
    Assume $\mathbf{u}^{(0)}_{t+\Delta t} = \mathbf{u}_t, \boldsymbol{\sigma}^{(0)}_{t+\Delta t} = \boldsymbol{\sigma}_t$ and $\Delta\mathbf{u}^{(0)}_{t+\Delta t} = \mathbf{0}$, then execute the iterative loops and determine $\mathbf{u}^{(k)}_{t+\Delta t}$ by (31):
① Compute the displacement increments $\Delta\mathbf{u}^{(k)}_t = \mathbf{u}^{(k)}_{t+\Delta t} - \mathbf{u}_t$, strain increments $\Delta\boldsymbol{\varepsilon}^{(k)}_t = \boldsymbol{\varepsilon}^{(k)}_t - \boldsymbol{\varepsilon}_t, \boldsymbol{\sigma}^{(k)}_{t+\Delta t}, f^{(k)}_{t+\Delta t}, \partial f/\partial\boldsymbol{\sigma}$
    and the other coefficients.
② Compute plastic parameter $d\lambda = (\partial f/\partial\boldsymbol{\sigma})^T\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}^{(k)}_t + f_t(\boldsymbol{\sigma}_t, \kappa_t)$, and determine the stress state: If $d\lambda > 0$, the stress is
    located in the plastic area; if $d\lambda \le 0$, the stress is in the elastic area.
③ Update the stiffness matrix by the following regulations:
      (a) In the elastic area, eliminate the terms containing $A$ in (28).
      (b) In the plastic area, compute $A$ by (13) and (18).
④ Find $\mathbf{u}^{(k)}_{t+\Delta t}$ by (31), calculate $\Delta\mathbf{u}^{(k)}_t = \mathbf{u}^{(k)}_{t+\Delta t} - \mathbf{u}_t$, $err = \left\|\Delta\mathbf{u}^{(k)}_t - \Delta\mathbf{u}^{(k-1)}_t\right\|$.
⑤ Determine whether or not $err$ is less than $\alpha\left\|\Delta\mathbf{u}^{(k)}_t\right\|$; if not, repeat Steps ①–④, conduct the $k + 1$th iteration until
    achieving convergence, then return to Step (2).
(2) Update $\mathbf{u}_{t+\Delta t} = \mathbf{u}^{(k)}_{t+\Delta t}, \ddot{\mathbf{u}}_{t+\Delta t} = a_0\Delta\mathbf{u}^{(k)}_t - a_2\dot{\mathbf{u}}_t - a_3\ddot{\mathbf{u}}_t$, and $\dot{\mathbf{u}}_{t+\Delta t} = \dot{\mathbf{u}}_t + a_6\ddot{\mathbf{u}}_t + a_7\ddot{\mathbf{u}}_{t+\Delta t}$.
(3) Re-compute the plastic parameter $d\lambda$ by the displacements at the final iteration, determine the stress state, and compute
    the total stresses and inner variable $\kappa_{t+\Delta t}$, plastic strains and so on, for the next time step.
(iii) Repeat Step (ii), continue the computation of the next load step, determine whether or not failure take places, and obtain
    the ultimate load.
```

Box 3: Whole amount iterative scheme of elastoplastic dynamical equations.

4.2. Iterative Methods of Dynamic Problems. The implicit damping iterative method proposed above can be easily expanded and applied to elastoplastic dynamic problems. By analogy with the total amount iterative algorithm (see (25)) of static problems, the total amount iterative formula of dynamic problems can be written as follows:

$$\begin{aligned}
&\left(\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}^{(k)}_{t+\Delta t}, \delta\boldsymbol{\varepsilon}\right) - \left(\frac{1}{A}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}^{(k)}_{t+\Delta t}, \left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\
&\qquad + \left(\rho\ddot{\mathbf{u}}^{(k)}_{t+\Delta t}, \delta\mathbf{u}\right) + \left(\mu\dot{\mathbf{u}}^{(k)}_{t+\Delta t}, \delta\mathbf{u}\right)\\
&\quad = \left(\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}^{(k-1)}_{t+\Delta t}, \delta\boldsymbol{\varepsilon}\right)\\
&\qquad - \left(\frac{1}{A}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\boldsymbol{\varepsilon}^{(k-1)}_{t+\Delta t}, \left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\
&\qquad + \left(\frac{1}{A}f^{(k-1)}_{t+\Delta t}, \left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\delta\boldsymbol{\varepsilon}\right)\\
&\qquad + \left(\mathbf{F}_{t+\Delta t}, \delta\mathbf{u}\right) - \left(\boldsymbol{\sigma}^{(k-1)}_{t+\Delta t}, \delta\boldsymbol{\varepsilon}\right).
\end{aligned} \tag{28}$$

The strain increments $\Delta\boldsymbol{\varepsilon}^{t(k)} = \boldsymbol{\varepsilon}^{t+\Delta t(k)} - \boldsymbol{\varepsilon}^t$ at time $t+\Delta t$ up to the kth iteration. The stress is calculated by the following:

$$\begin{aligned}
\boldsymbol{\sigma}^{(k)}_{t+\Delta t} &= \boldsymbol{\sigma}_t + \mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}^{(k)}_t - \frac{1}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}}\left(\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)^T\mathbf{D}^{\mathbf{e}}\Delta\boldsymbol{\varepsilon}^{(k)}_t\\
&\quad - \frac{1}{A_0}\mathbf{D}^{\mathbf{e}}\frac{\partial f}{\partial\boldsymbol{\sigma}}f_t\left(\boldsymbol{\sigma}_t, \kappa_t\right).
\end{aligned} \tag{29}$$

The iterative process of (25) has been rewritten as a dynamic time-dependent loading process. When $\boldsymbol{\varepsilon}^{(k)}_{t+\Delta t} \to \boldsymbol{\varepsilon}^{(k-1)}_{t+\Delta t} = \boldsymbol{\varepsilon}_{t+\Delta t}$, $f^{(k)}_{t+\Delta t} \to 0$, (28) converges to the following kinetic equation:

$$\left(\rho\ddot{\mathbf{u}}_{t+\Delta t}, \delta\mathbf{u}\right) + \left(\mu\dot{\mathbf{u}}_{t+\Delta t}, \delta\mathbf{u}\right) = \left(\mathbf{F}_{t+\Delta t}, \delta\mathbf{u}\right) - \left(\boldsymbol{\sigma}_{t+\Delta t}, \delta\boldsymbol{\varepsilon}\right). \tag{30}$$

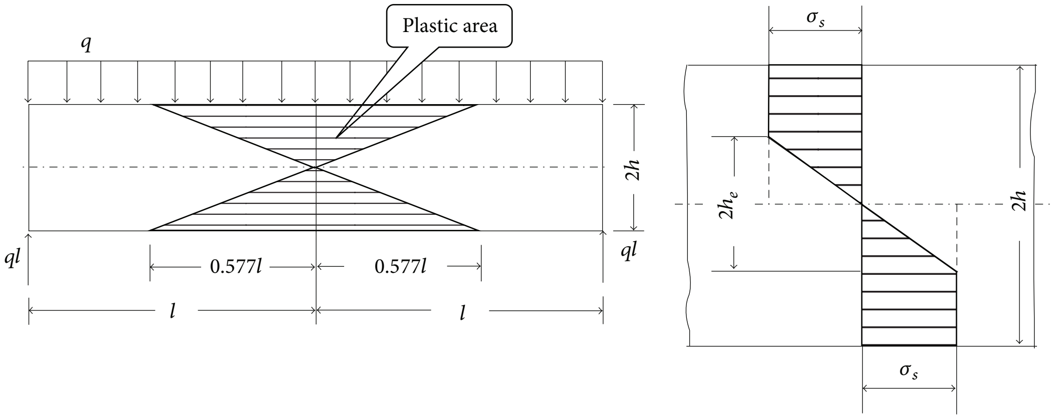
The iterative equation (28) is discretized in the time domain using the basic assumptions of the Newmark integration method. The dynamic finite element equations are obtained as follows:

$$\begin{aligned}
&\left(\mathbf{K}^{(k-1)}_t + a_0\mathbf{M} + a_1\mathbf{C}\right)\mathbf{u}^{(k)}_{t+\Delta t}\\
&\quad = \mathbf{K}^{(k-1)}_t\mathbf{u}^{(k-1)}_{t+\Delta t} + \Delta\mathbf{F}^{(k-1)}_{t+\Delta t}\\
&\qquad + \mathbf{M}\left[a_0\mathbf{u}^t_t + a_2\dot{\mathbf{u}}^t_t + a_3\ddot{\mathbf{u}}^t_t\right]\\
&\qquad + \mathbf{C}\left[a_1\mathbf{u}^t_t + a_4\dot{\mathbf{u}}^t_t + a_5\ddot{\mathbf{u}}^t_t\right].
\end{aligned} \tag{31}$$

Here $\mathbf{K}^{(k)}_t$, $\mathbf{M}$, and $\mathbf{C}$, respectively, represent the stiffness matrix, mass matrix, and damping matrix, and $\Delta\mathbf{F}^{(k)}_{t+\Delta t}$ is the load increment generated by the last three terms on the right side of (28). Note that the nodal forces produced by $\mathbf{K}_t\mathbf{u}_{t+\Delta t}$ are not necessarily equal to the nodal force caused by $\boldsymbol{\sigma}_{t+\Delta t}$, due to possible plastic deformation. If not, $\mathbf{K}_t\mathbf{u}_{t+\Delta t} + \Delta\mathbf{F}_{t+\Delta t} = \mathbf{F}_{t+\Delta t}$ and (28) tend to be the full amount of the Newmark integral equation of a linear elastic problem.

Carrying numerical iterations until $\mathbf{u}_{t+\Delta t} = \mathbf{u}^{(k)}_{t+\Delta t} \approx \mathbf{u}^{(k-1)}_{t+\Delta t}$ and $\Delta\mathbf{u}_t = \mathbf{u}_{t+\Delta t} - \mathbf{u}_t$, by which one can obtain the acceleration $\ddot{\mathbf{u}}_{t+\Delta t} = a_0\Delta\mathbf{u}_t - a_2\dot{\mathbf{u}}_t - a_3\ddot{\mathbf{u}}_t$ at the time of $t + \Delta t$ and velocities $\dot{\mathbf{u}}_{t+\Delta t} = \dot{\mathbf{u}}_t + a_6\ddot{\mathbf{u}}_t + a_7\ddot{\mathbf{u}}_{t+\Delta t}$, where $a_0 \sim a_7$ indicate the Newmark integration constants, given as follows: $a_0 = 1/\gamma\Delta t^2$, $a_1 = \delta/\gamma\Delta t$, $a_2 = 1/\gamma\Delta t$, $a_3 = 1/2\gamma - 1$, $a_4 = \delta/\gamma - 1$, $a_5 = (\Delta t/2)(\delta/\gamma - 2)$, $a_6 = \Delta t(1 - \gamma)$, $a_7 = \gamma\Delta t$, and $\delta \ge 0.5, \gamma = 0.25(0.5 + \delta)^2$. Listed in Box 3 is the total amount iterative scheme of the elastoplastic dynamical equations.

Shown in (28) is the implicit damping iteration scheme to solve the total displacements of elastoplastic dynamic problems. This creates the boundary conditions given in the form of whole amount displacements or loads, thus



(a) Theoretical plastic area ($M_p = 1.5M_e$)

(b) Stress in the cross-section

Figure 3: Theoretical plastic area of flexural beam under static load.

providing a great convenience to conduct the nonlinear response analyses.

4.3. Numerical Examples. The example is the elastoplastic bending of a simply supported beam subjected to a uniform load of intensity q with the rectangular cross-section ($b \times 2h$), as shown in Figure 3. Using the perfectly elastoplastic model and Mises' yield criterion, when $h_e = h$, the stresses on the upper and lower edges of the middle part of the beam just reach the plastic limit. If the cross-section is in the fully elastic state, then the elastic limit moment at the middle beam is $M_e = (2/3)bh^2\sigma_s$; when $h_e = 0$, the cross-section of the beam fully accesses the full plastic state, formatting a plastic hinge, and then the plastic limit moments are $M_p = bh^2\sigma_s$ and $M_p = 1.5M_e$.

The above theoretical solutions are obtained without gravity weight. For the convenience of comparison with the theoretical solutions, this study does not consider the weight of the beam and instead considers the mass of the beam material in the dynamic calculation and takes the damping proportional to mass; that is, $\mathbf{C} = \alpha\mathbf{M}$, with the dynamic loading rate of 100.0 kN/s.

We conduct static numerical simulations by exerting static distributed load with the increments of $0.01q_s$, where q_s (100 kN/m^2) denotes the plastic limit load ($M_p = 1.5M_e$). In order to verify the accuracy and efficiency of the proposed method, the numerical tests were executed, respectively, by using the Implicit Damping Iterative Algorithm (IDIA) and the Return-Mapping Algorithm provided by the ABAQUS software. The curves of deflection via the loading history are obtained as shown in Figure 4. It is shown that the two curves are almost identical. Then, by exerting static distributed load with the increments of the increments of $0.10q_s$, $0.05q_s$, and $0.01q_s$, respectively, we obtain the three curves using IDIA, as shown in Figure 5, which almost completely overlap with each other. The plastic zones as shown in Figure 6, which are calculated by using both IDIA and ABAQUS, under the plastic limit moments, are all very close to the theoretical solution. Furthermore, we noted that the plastic zone calculated under the plastic limit load is not affected by loading increments. Those numerical tests demonstrate that the proposed implicit iteration algorithm has both strong stability and reliability.

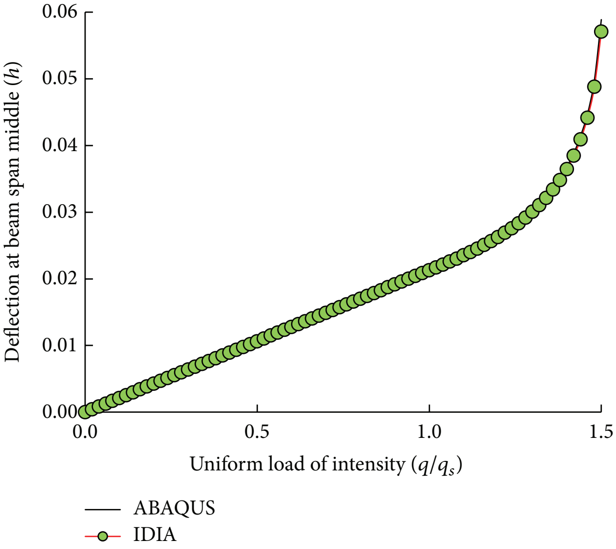


Figure 4: Computational results by IDIA compared with ABAQUS.

The dynamical tests were conducted with 0.001 s of time step and 100.0 kN/s of loading rate. The curves of deflection under dynamic load ($\alpha = 30.0$), obtained, respectively, by IDIA and ABAQUS, as shown in Figure 7, are completely overlapped. In addition, we have similar results when the damping coefficient α is, respectively, taken as 0 and 10. The distributions of the plastic zones in Figure 8 show that the dynamic flexural strength is higher than the static flexural strength, and as damping coefficient increases, the corresponding plastic zone becomes smaller and smaller. Figures 8(a)(A) and 8(b)(A) show the case with no damping ($\alpha = 0$) and the dynamic effect induced only by the inertia force; the plastic zone is a bit smaller than that of the static load. In general, the above computational results obtained

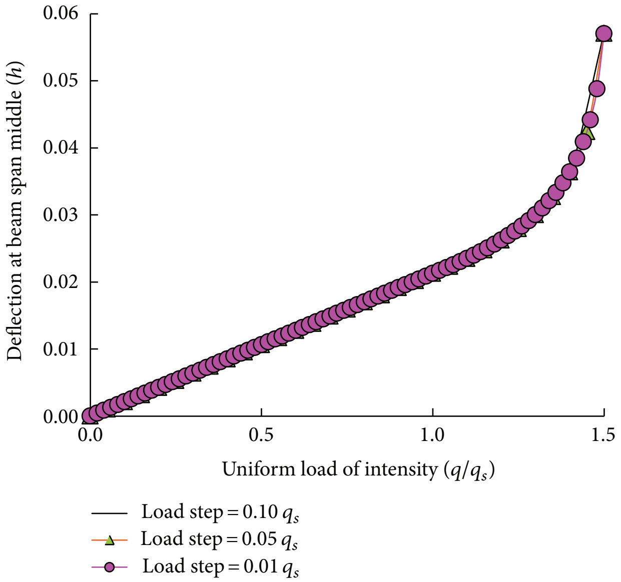
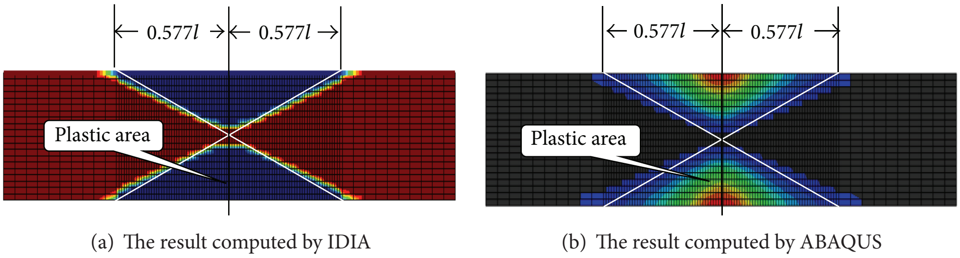


FIGURE 5: Stability analysis of IDIA.



(a) The result computed by IDIA

(b) The result computed by ABAQUS

FIGURE 6: Plastic area under static load (M_p = 1.5M_e).

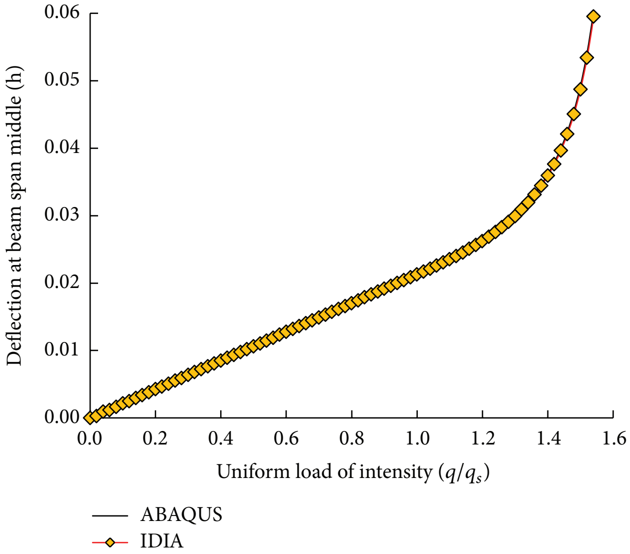
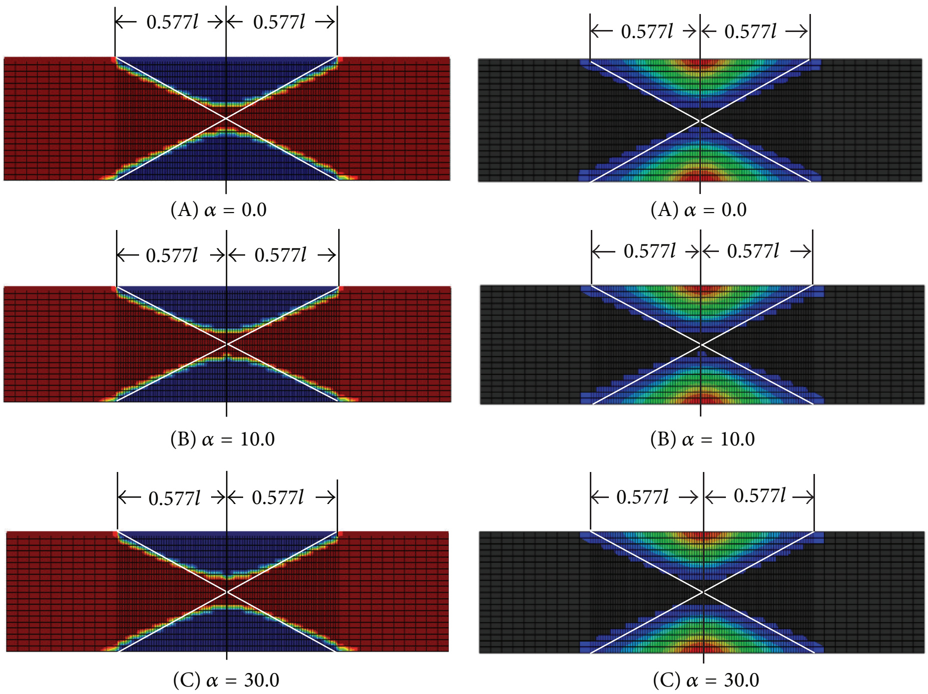


FIGURE 7: Dynamical results computed, respectively, by IDIA and ABAQUS (α = 30.0).



(A) α = 0.0

(B) α = 10.0

(C) α = 30.0

(a) The result computed by IDIA

(A) α = 0.0

(B) α = 10.0

(C) α = 30.0

(b) The result computed by ABAQUS

FIGURE 8: Plastic area under dynamical load ($M_p = 1.5M_e$), respectively, by IDIA and ABAQUS.

from IDIA and ABAQUS completely agree well and in line with the general rule of the dynamic effect.

These numerical experiments are carried out on a PC configured as Intel Core i5-2400 CPU@3.10 GHz, MEM 2.99 GB. In the examples given, it iterates only 3 or 4 times to converge to solution: the first example with 16,320 DOF, 80-step calculation, taking less than 20 minutes; the second example, with 4662 DOF, running 15 loading steps to form a plastic hinge, taking less than one minute. It is shown from these calculations that the iterative algorithm proposed has good computational efficiency.

In addition, a series of numerical tests performed by using the above examples showed the magnitude of the constants C_1 and C_2 in (18) affects little the iteration convergence rate and solution accuracy.

5. Conclusion

This paper first describes the basic concept of the implicit damping iterative method and then gives the displacement incremental iterative scheme as well as the whole iterative scheme to solve the static and dynamical elastoplastic problems. At the same time, the authors present the corresponding computing procedures and script files of the iterative schemes in accordance with the FEPG language rules and generate the FORTRAN programs. The circular tunnel excavation and elastoplastic bending of simply supported beam problems are numerically calculated, and the results verify the correctness and reliability of the proposed implicit iteration algorithms.

The elastoplastic damping implicit iterative algorithm will simultaneously solve equilibrium equations, yield functions, and plastic flow equations. The method does not require an explicit expression of the elastoplastic stiffness matrix and local iteration for "return mapping" stress to the yield surface. In addition, the damping factor introduced in the paper improves the stiffness matrix conformation. Although the numerical algorithm proposed is based on the elastoplastic problem, it can easily be expanded and applied to solve the general material nonlinearity problem. In particular, the whole amount implicit damping iterative scheme allows the provision of displacement or stress boundary conditions in the form of the whole amount. The method will bring great convenience to solving nonlinear dynamic response of high concrete dams with complex seismic ground motion inputs and other similar problems.

The drawback of the implicit damping iterative method proposed in this paper is that it requires recalculation of the stiffness matrix in each iteration step. However, as computer processing power increases, especially with the development of high performance computers, computing speed and scale will no longer be the bottleneck of scientific computing. Therefore, the shortcomings of the proposed algorithm will not be the outstanding problems of the application. In addition, the authors only carried out the numerical experiments of the incremental loading process, but if under displacement load, the implicit damping iterative algorithm can also be employed to simulate material plastic softening processes without much additional effort.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This project is supported by the National Natural Science Foundation of China (no. 51079164), China Water Resources Public Welfare Project (nos. 201201053 and 201301057), and Research Special of the China Institute of Water Resources and Hydropower Research (no. KJ1242).

References

[1] O. C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, New York, NY, USA, 3rd edition, 1977.

[2] H. Matthies and G. Strang, "The solution of nonlinear finite element equations," *International Journal for Numerical Methods in Engineering*, vol. 14, no. 11, pp. 1613–1626, 1979.

[3] B. M. Irons and R. C. Tuck, "A version of the Aitken accelerator for computer iteration," *International Journal for Numerical Methods in Engineering*, vol. 1, no. 3, pp. 275–277, 1969.

[4] M. A. Crisfield, "Arc-length Method Including Line Searches and accelerations," *International Journal for Numerical Methods in Engineering*, vol. 19, no. 9, pp. 1269–1289, 1983.

[5] T. Seifert and I. Schmidt, "Line-search methods in general return mapping algorithms with application to porous plasticity," *International Journal for Numerical Methods in Engineering*, vol. 73, no. 10, pp. 1468–1495, 2008.

[6] G. A. Wempner, "Discrete approximations related to nonlinear theories of solids," *International Journal of Solids and Structures*, vol. 7, no. 11, pp. 1581–1599, 1971.

[7] E. Riks, "The application of Newton's method to the problem of elastic stability," *Journal of Applied Mechanics*, vol. 39, no. 4, pp. 1060–1065, 1972.

[8] B. W. R. Forde and S. F. Stiemer, "Improved arc length orthogonality methods for nonlinear finite element analysis," *Computers and Structures*, vol. 27, no. 5, pp. 625–630, 1987.

[9] M. Müller, "Passing of instability points by applying a stabilized Newton-Raphson scheme to a finite element formulation: comparison to arc-length method," *Computational Mechanics*, vol. 40, no. 4, pp. 683–705, 2007.

[10] J. C. Simo and T. J. W. Hughes, *Computational Inelasticity*, Springer, New York, NY, USA, 1998.

[11] J. C. Simo and R. L. Taylor, "A return mapping algorithm for plane stress elastoplasticity," *International Journal for Numerical Methods in Engineering*, vol. 22, no. 3, pp. 649–670, 1986.

[12] B. Moran, M. Ortiz, and C. F. Shih, "Formulation of implicit finite element methods for multiplicative finite deformation plasticity," *International Journal for Numerical Methods in Engineering*, vol. 29, no. 3, pp. 483–514, 1990.

[13] D. Peirce, C. F. Shih, and A. Needleman, "A tangent modulus method for rate dependent solids," *Computers and Structures*, vol. 18, no. 5, pp. 875–887, 1984.

[14] B. Moran, "A finite element formulation for transient analysis of viscoplastic solids with application to stress wave propagation problems," *Computers and Structures*, vol. 27, no. 2, pp. 241–247, 1987.

[15] Z. L. Zhang, "Explicit consistent tangent moduli with a return mapping algorithm for pressure-dependent elastoplasticity models," *Computer Methods in Applied Mechanics and Engineering*, vol. 121, no. 1–4, pp. 29–44, 1995.

[16] M. A. Keavey, "A simplified canonical form algorithm with application to porous metal plasticity," *International Journal for Numerical Methods in Engineering*, vol. 65, no. 5, pp. 679–700, 2006.

[17] G. P. Liang and Y. F. Zhou, *The Finite Element Language*, Science Press, Beijing, China, 2013, (Chinese).

[18] G. P. Liang, "Finite element program generator and finite element language," *Advances in Mechanics*, vol. 20, no. 2, pp. 199–204, 1990 (Chinese).