

# An Automatic Newton–Raphson Scheme

Daichao Sheng,\* Scott W. Sloan, and Andrew J. Abbo

---

Received September 2001

Senior Lecturer (ARC ARF/QEII Fellow), Department of  
Civil, Surveying & Environmental Engineering, The  
University of Newcastle, NSW 2308, Australia  
Tel: +61 2 49215746, Fax: +61 2 49216991  
E-mail: Daichao.Sheng@newcastle.edu.au

Professor of Civil Engineering, Department of Civil,  
Surveying & Environmental Engineering, The University  
of Newcastle, NSW 2308, Australia  
Tel: +61 2 49215746, Fax: +61 2 49216059  
E-mail: scott.sloan@newcastle.edu.au

Software Engineer/Research Fellow, Department of Civil,  
Surveying & Environmental Engineering, The University  
of Newcastle, NSW 2308, Australia  
Tel: +61 2 49215746, Fax: +61 2 49216058

---

**ABSTRACT.** *This article presents an automatic Newton–Raphson method for solving nonlinear finite element equations. It automatically subincrements a series of given coarse load increments so that the local load path error in the displacements is held below a prescribed threshold. The local error is measured by taking the difference between two iterative solutions obtained from the backward Euler method and the SS21 method. By computing both the displacement rates and the displacements this error estimate is obtained cheaply.*

*The performance of the new automatic scheme is compared with the standard Newton–Raphson scheme, the modified Newton–Raphson scheme with line search, and two other automatic schemes that are based on explicit Euler methods. Through analyses of a wide variety of problems, it is shown that the automatic Newton–Raphson scheme is superior to the standard Newton–Raphson methods in terms of efficiency, robustness, and accuracy.*

---

*Key Words and Phrases.* Newton–Raphson, error control, subincrementation, nonlinear, finite element method.

---

\* All correspondence to first author. E-mail: Daichao.Sheng@newcastle.edu.au

## I. INTRODUCTION

In nonlinear displacement finite element methods the global stiffness equations are solved using a variety of explicit/incremental or implicit/iterative methods. Incremental schemes treat the global stiffness equations as a set of ordinary differential equations and solve them using a series of piecewise linear steps. Iterative methods treat the governing relations as a set of nonlinear equations and iterate within each load increment until the unbalanced forces are smaller than a preset tolerance. Detailed reviews of these methods can be found, for example, in Crisfield [1] and Potts and Zdravkovic [2]. Although advanced methods with enhanced accuracy, robustness, and efficiency have been developed during the last 30 years or so, the standard Newton Raphson method and its variants are still widely used in commercial finite element codes.

The Newton–Raphson method solves the governing equations by applying the unbalanced forces, computing the corresponding displacements, and then iterating until the drift from equilibrium is small. The chief advantages of this scheme are its accuracy and rapid (but conditional) convergence. Its major disadvantage is that it may not always converge, particularly when the material or problem is strongly nonlinear. To overcome this, various techniques have been developed to stabilize and accelerate the convergence of Newton–Raphson schemes. These include the line search techniques of Matthies and Strang [3] and Crisfield [1], as well as the arc length control procedures developed by Wempner [4] and Riks [5,6]. Line search methods attempt to stabilize Newton–Raphson iterations by shrinking or expanding the current displacement increment to minimize the resulting unbalanced forces. In cases where the current search direction is poor, or where the unbalanced forces are nonsmooth functions of the displacements, line searches may be of limited use. The philosophy behind arc length methods is to force the Newton–Raphson iterations to remain within the vicinity of the last converged equilibrium point. This means that the applied load must be reduced as the iterations proceed, but greatly reduces the risk of divergence for strongly nonlinear problems. A detailed discussion of various arc length methods and their practical implementation can be found in Crisfield [1]. In a fundamental development, Simo and Taylor [7] derived the consistent tangent technique for use with the Newton–Raphson scheme. By incorporating high order terms that are usually ignored in the standard form of the elastoplastic stiffness relations, this procedure gives a quadratic rate of convergence. Although powerful, the method is difficult to implement for complex yield criteria because it needs second derivatives of the yield function to be evaluated.

This article presents a Newton–Raphson method which automatically subincrements a given set of coarse load increments. The performance of the new method is compared with the performance of the standard Newton–Raphson method, the Newton–Raphson method with line search, and two explicit automatic methods developed by Abbo and Sloan [8,9]. The behavior of all the methods is studied by considering a number of boundary value problems involving different soil constitutive models.

## II. NEWTON–RAPHSON SCHEMES

### A. Standard Newton–Raphson Scheme

In the displacement finite element method, the global elastoplastic stiffness equations to be solved for each load increment can be written in rate form as

$$\dot{\mathbf{U}} = [\mathbf{K}_{ep}(\mathbf{U})]^{-1} \dot{\mathbf{F}}^{ext} = [\mathbf{K}_{ep}(\mathbf{U})]^{-1} \frac{\Delta \mathbf{F}^{ext}}{\Delta t} \quad (1)$$

where  $\dot{\mathbf{U}}$  is a vector of unknown displacement rates,  $\mathbf{K}_{ep}$  is the tangent stiffness matrix,  $\Delta \mathbf{F}^{ext}$  is a vector of external force increments which are applied over an arbitrary time interval  $\Delta t$ , and the superior dot denotes a derivative with respect to time. For rate independent problems it is convenient to introduce a pseudo time,  $T$ , defined by

$$T = \frac{t - t_0}{\Delta t} \quad (2)$$

where  $t_0$  and  $t_0 + \Delta t$  are, respectively, the times at the start and end of the load increment and  $0 \leq T \leq 1$ . Using the chain rule for  $\dot{\mathbf{U}}$  in (1) and substituting for  $T$  from (2) yields

$$\frac{d\mathbf{U}}{dT} = [\mathbf{K}_{ep}(\mathbf{U})]^{-1} \Delta \mathbf{F}^{ext} \quad (3)$$

Equation (3) has the form of a classical initial value problem since  $\Delta \mathbf{F}^{ext}$  is known, the right-hand side is a function of  $\mathbf{U}$ , and the initial conditions are the known displacements, denoted as  $\mathbf{U}_0$ , at the start of the load increment where  $T = 0$ .

Since the stiffness matrix  $\mathbf{K}_{ep}$  depends on the displacements  $\mathbf{U}$ , or more correctly on the stresses corresponding to these displacements, it does not remain constant during the application of a load increment. For this reason, one of the chief concerns in solving (3) is how to compute a representative value of  $\mathbf{K}_{ep}$  over the load step. A further concern is that, regardless of the numerical approximation used to solve equation (3), the solution will always give rise to unbalanced forces. If left unchecked, these forces accumulate in subsequent increments and may eventually cause the solution to become grossly inaccurate. Different load stepping strategies for solving (3) are distinguished from one another mainly in the way that these two concerns are dealt with.

The standard Newton-Raphson scheme solves the governing relations by applying the external forces, computing the corresponding displacement increments, and iterating until the drift from equilibrium is small. The overall process is summarized by the following sequence of steps:

$$\begin{aligned} \mathbf{R}^{i-1} &= \mathbf{F}^{ext} - \int_V \mathbf{B}^T \boldsymbol{\sigma}^{i-1} dV \\ \delta \mathbf{U} &= [\mathbf{K}_{ep}(\mathbf{U}^{i-1})]^{-1} \mathbf{R}^{i-1} \\ \mathbf{U}^i &= \mathbf{U}^{i-1} + \delta \mathbf{U} \end{aligned}$$

where  $\mathbf{R}^{i-1}$ ,  $\boldsymbol{\sigma}^{i-1}$ , and  $\mathbf{U}^{i-1}$  are, respectively, the unbalanced forces, stresses, and displacements at the start of the  $i$ th iteration,  $\mathbf{F}^{\text{ext}}$  are the external forces at the end of the current load increment, and  $\delta\mathbf{U}^i$  are the iterative displacements for the  $i$ th iteration. When updating the stresses in the above equations, it is advisable to integrate the constitutive relationships using incremental, rather than iterative, strains according to

$$\Delta\mathbf{e} = \mathbf{B}\{\mathbf{u}^i - \mathbf{u}_0\}$$

where  $\mathbf{u}_0$  and  $\mathbf{u}^i$  are, respectively, the element displacements at the start of the current load increment and the end of the  $i$ th iteration. As noted by Crisfield [1], this strategy avoids the complicated strain paths and spurious unloading that occurs when iterative strains are used, and has been found to improve the reliability of most iterative solution schemes.

Following conventional practice, the Newton-Raphson iterations are terminated once the unbalanced forces are small in comparison to the externally applied loads according to

$$\|\mathbf{R}\|_2 \leq ITOL \|\mathbf{F}^{\text{ext}}\|_2 \quad (4)$$

where  $ITOL$  is typically in the range  $10^{-3}$  to  $10^{-6}$ .

## B. Newton-Raphson Scheme with Line Search

The Newton-Raphson scheme (or any of its variants) can be implemented with a line search to improve its stability. In such a scheme, the displacements are updated according to

$$\mathbf{U}^i = \mathbf{U}^{i-1} + \beta_1 \delta\mathbf{U}$$

where the parameter  $\beta_1$  is used to control the iterative step length. In the absence of better information, this parameter is initially set to 1 and may, if necessary, be modified by minimizing the total potential energy. In each iteration, once the iterative displacements  $\mathbf{U}^i$  and the new unbalanced forces  $\mathbf{R}^i$  are known, two inner products,  $s_0$  and  $s_i$ , representing the potential energy increments at the start and end of the iteration, respectively, are computed as follows

$$s_0 = \delta\mathbf{U}^T \mathbf{R}^{i-1}$$

$$s_i = \delta\mathbf{U}^T \mathbf{R}^i$$

If  $s_i$  is small in comparison with  $s_0$ , i.e.,

$$|r_1| = \left| \frac{s_i}{s_0} \right| < LTOL \quad (5)$$

where  $LTOL$  is a small line search tolerance, then  $\beta_1$  is accepted and the current iteration is regarded as successful. Otherwise, we compute a new  $\beta_{1n}$  according to

$$\beta_{1n} = \frac{\beta_0 r_0 - \beta_1 r_1}{r_0 - r_1}$$

where  $\beta_0$  is initially set to 0,  $r_0$  is initially set to 1, and  $\beta_1$  is initially set to 1. If  $\beta_{1n}$  is found to be between  $\beta_0$  and  $\beta_1$  (interpolation), then  $\beta_1$  is reset as

$$\beta_1 = \beta_{1n}$$

If  $\beta_{1n}$  is found to be outside the interval  $\beta_0$  to  $\beta_1$  (extrapolation), then we reset

$$\beta_0 = \max\{\beta_0, \beta_1\}$$

$$r_0 = \min\{r_0, r_1\}$$

$$\beta_1 = \beta_{1n}$$

With the new  $\beta_1$ , the displacements  $\mathbf{U}^i$ , the unbalanced forces  $\mathbf{R}^i$ , the inner products and the absolute ratio  $|r_1|$  are recomputed. This process is repeated until the condition (5) is satisfied.

The line search tolerance  $LTOL$  used in equation (5) is typically set to 0.8 [1]. In practice,  $\beta_1$  should be limited within a certain range so that the current displacement increments  $\delta\mathbf{U}$  are not expanded too dramatically. In this study, we set  $0 \leq \beta_1 \leq 10$ .

### III. AUTOMATIC NEWTON-RAPHSON SCHEME

As mentioned in the introduction, one major disadvantage of the Newton-Raphson method is that its convergence is not guaranteed, particularly for highly nonlinear problems. Since the convergence properties of this algorithm are largely governed by the load increment size, it is often necessary to resort to a trial and error procedure unless some form of automatic load step adjustment is available. Though convergence is usually guaranteed by choosing a large number of load steps, this option is both laborious and inefficient for large-scale problems (especially when the nonlinearity in the load path is highly localized).

Because plasticity theory assumes infinitesimal increments, the use of discrete load steps in finite element analysis inevitably gives rise to load path error. This error is not necessarily removed by equilibrium iterations, increases with increasing load increment size, and may be significant for problems which give rise to strongly nonlinear strain paths. One approach is to control the load path error in a direct and rational way, and automatically choose small load increments where they are needed. Abbo and Sloan [8,9] used this strategy to develop automatic

load stepping schemes based on a pair of explicit Euler methods and these proved to be accurate, robust, and fast for a wide range of constitutive models [8,10]. We now use the same principle to develop an automatic implicit Newton–Raphson scheme.

### A. First-Order and Second-Order Accurate Solutions

The global stiffness equations for each load increment can be rewritten in term of nodal velocities as

$$\mathbf{K}_{\text{ep}} \mathbf{V}_n = \dot{\mathbf{F}}_n^{\text{ext}} = \frac{\Delta \mathbf{F}_n^{\text{ext}}}{h_n} \quad (6)$$

where  $h_n$  is the time step when the external load increment  $\Delta \mathbf{F}_n^{\text{ext}}$  is applied, and  $\mathbf{V}_n$  is an average estimate of the velocities over  $h_n$ . This equation can be solved for  $\mathbf{V}_n$  using the Newton–Raphson iteration procedure and will be discussed later in this section. Once  $\mathbf{V}_n$  is known, the displacements can be updated according to

$$\bar{\mathbf{U}}_n = \mathbf{U}_{n-1} + h_n \mathbf{V}_n \quad (7)$$

The solution procedure above is equivalent to the backward Euler method and the displacements updated using (7) are of first-order accuracy.

Alternatively, the global stiffness equations can be written in terms of the accelerations as

$$h_n \mathbf{K}_{\text{ep}} \mathbf{A}_n = \dot{\mathbf{F}}_n^{\text{ext}} - \mathbf{K}_{\text{ep}} \mathbf{V}_{n-1} \quad (8)$$

where  $\mathbf{V}_{n-1}$  is the velocity vector for the previous time step and

$$\mathbf{A}_n = \frac{\mathbf{V}_n - \mathbf{V}_{n-1}}{h_n} \quad (9)$$

is an average estimate of the accelerations over the current time step. With known  $\mathbf{V}_{n-1}$ , equation (8) can be solved for  $\mathbf{A}_n$ . Once  $\mathbf{A}_n$  is known, the displacements and velocities can be updated according to

$$\begin{aligned} \mathbf{U}_n &= \mathbf{U}_{n-1} + h_n \mathbf{V}_{n-1} + \frac{1}{2} h_n^2 \mathbf{A}_n \\ \mathbf{V}_n &= \mathbf{V}_{n-1} + h_n \mathbf{A}_n \end{aligned} \quad (10)$$

This type of displacement update, which requires information from two consecutive time steps, is equivalent to the SS21 algorithm for solving systems of first order differential equations (see, for example, Zienkiewicz et al, 1984; Thomas and Gladwell, 1988), and is of second order accuracy.

Since equations (8) are identical to (6), we do not need to solve them to obtain the second-order accurate updates. Instead, we can solve the system (6) for the velocities  $\mathbf{V}_n$  and find the accelerations  $\mathbf{A}_n$  from (9). Once  $\mathbf{V}_n$  and  $\mathbf{A}_n$  are known, the second-order accurate updates can be found from (10). Alternatively, it is possible to propagate the solution using the first-order accurate solution (7) and just use (10) to estimate the local error in the displacements. Because of its simplicity and accuracy, this approach is used here.

### B. Error Control and Subincrementation

Since the local truncation errors in the updates (7) and (10) are, respectively,  $O(h^2)$  and  $O(h^3)$ , the lower order estimate may be subtracted from the higher order estimate to give the local truncation error measure

$$\mathbf{E}_n = \mathbf{U}_n - \bar{\mathbf{U}}_n = h_n \mathbf{V}_{n-1} + \frac{1}{2} h_n^2 \mathbf{A} - h_n \mathbf{V}_n = \frac{1}{2} h_n (\mathbf{V}_{n-1} - \mathbf{V}_n)$$

For the purposes of error control,  $\mathbf{E}_n$  may be replaced by the more useful dimensionless relative error measure

$$R_n = \frac{\|\mathbf{E}_n\|}{\|\mathbf{U}_n\|}$$

This relative error  $R_n$  is then used to control the size of the load subincrement. The current subincrement is accepted if  $R_n$  is less than a user specified tolerance,  $DTOL$ , and rejected otherwise. In either case, the size of the next load subincrement is found from

$$h_{n+1} = q h_n$$

where  $q$  is a factor chosen to limit the predicted local error. Since the local error for the next load subincrement,  $R_{n+1}$ , is related to  $R_n$  by

$$R_{n+1} = q^2 R_n$$

the required factor  $q$  is found by insisting that  $R_{n+1} \leq DTOL$  to give

$$q \leq \sqrt{DTOL/R_n}$$

To prevent the step control mechanism from choosing subincrements which just fail to meet the local error tolerance, a safety factor should be applied to  $h_{n+1}$  according to

$$q = 0.8\sqrt{DTOL/R_n}$$

Moreover, to minimize the number of rejected substeps,  $q$  is constrained to lie between 0.1 and 2.0. These limits do not greatly influence the overall performance of the scheme, but are added for robustness.

### C. Newton–Raphson Iteration for Velocity

Rewriting the system of equations (6), to define a residual  $\mathbf{R}$ , gives

$$\mathbf{R}(\mathbf{V}) = \dot{\mathbf{F}}_n^{\text{ext}} - \mathbf{K}_{\text{ep}} \mathbf{V}_n = 0$$

The solution to these equations may be found using the Newton–Raphson algorithm. Letting the superscript  $i$  denote iteration number, this scheme takes the form

$$\begin{aligned} \delta \mathbf{V}^i &= \left\{ \frac{\partial \mathbf{R}}{\partial \mathbf{V}} \right\}^{-1} \mathbf{R}(\mathbf{V}^{i-1}) = \left\{ \mathbf{K}_{\text{ep}}(\bar{\mathbf{U}}_n^{i-1}) \right\}^{-1} \left( \dot{\mathbf{F}}_n^{\text{ext}} - \mathbf{K}_{\text{ep}}(\bar{\mathbf{U}}_n^{i-1}) \mathbf{V}_n^{i-1} \right) \\ \mathbf{V}_n^i &= \mathbf{V}_n^{i-1} + \delta \mathbf{V}^{i-1} \\ \bar{\mathbf{U}}_n^i &= \mathbf{U}_{n-1} + h_n \mathbf{V}_n^i \end{aligned} \quad (11)$$

The iteration process can be started with

$$\mathbf{V}_n^0 = \mathbf{V}_{n-1} \quad (12)$$

$$\bar{\mathbf{U}}_n^0 = \mathbf{U}_{n-1} + h \mathbf{V}_n^0 \quad (13)$$

to accelerate the convergence.

Noting that

$$\mathbf{K}_{\text{ep}} \mathbf{V}_n^{i-1} = \int_V \mathbf{B}^T \dot{\boldsymbol{\sigma}}_n^{i-1} dV = \dot{\mathbf{F}}_n^{\text{int}}$$

and using the first order approximations

$$\dot{\mathbf{F}}_n^{\text{ext}} = (\mathbf{F}_n^{\text{ext}} - \mathbf{F}_{n-1}^{\text{ext}})/h_n + \mathbf{0}(h_n)$$

$$\dot{\mathbf{F}}_n^{\text{int}} = (\mathbf{F}_n^{\text{int}} - \mathbf{F}_{n-1}^{\text{int}})/h_n + \mathbf{0}(h_n)$$

equation (11) may be rewritten in the alternative form

$$\begin{aligned} \delta \mathbf{V}^i &= \left\{ \mathbf{K}_{\text{ep}}(\bar{\mathbf{U}}_n^{i-1}) \right\}^{-1} \left( \frac{\mathbf{F}_n^{\text{ext}} - \mathbf{F}_n^{\text{int}}}{h_n} - \frac{\mathbf{F}_{n-1}^{\text{ext}} - \mathbf{F}_{n-1}^{\text{int}}}{h_n} \right) \\ &= \left\{ \mathbf{K}_{\text{ep}}(\bar{\mathbf{U}}_n^{i-1}) \right\}^{-1} \left( \frac{\mathbf{F}_n^{\text{ext}} - \mathbf{F}_n^{\text{int}} - \mathbf{F}_{n-1}^{\text{unb}}}{h_n} \right) \end{aligned} \quad (14)$$

where  $\mathbf{F}_n^{\text{ext}}$  is the applied external force vector at the end of the load subincrement,  $\mathbf{F}_n^{\text{int}}$  is the internal force vector estimated based on displacements  $\bar{\mathbf{U}}_n^{i-1}$ , and  $\mathbf{F}_{n-1}^{\text{unb}}$  the unbalanced forces at the start of the load subincrement. Provided the iteration convergence tolerance is small, the unbalanced forces  $\mathbf{F}_{n-1}^{\text{unb}}$  will also be small. Adding these forces to the forces on the right-hand side of (14), and assuming they are dissipated at a constant rate over the interval  $h_n$ , leads to the following expression for the iterative velocities

$$\delta \mathbf{V}^i = \left\{ \mathbf{K}_{\text{ep}}(\bar{\mathbf{U}}_n^{i-1}) \right\}^{-1} \left( \frac{\mathbf{F}_n^{\text{ext}} - \mathbf{F}_n^{\text{int}}}{h_n} \right) \quad (15)$$

The iterations are terminated once the total unbalanced force rate is small in comparison to the applied force rate, i.e.,

$$\|\mathbf{R}\| \leq ITOL \|\dot{\mathbf{F}}_n^{\text{ext}}\| \quad (16)$$

If this condition is not satisfied within a prescribed maximum number of iterations, the size of the subincrement  $h_n$  is reduced using the simple rule

$$h_n \leftarrow 0.25h_n \quad (17)$$

#### D. Initial Velocity $\mathbf{V}_0$

For the very first subincrement, a starting velocity vector  $\mathbf{V}_0$  is needed to compute the local error. This velocity at  $t = 0$  can be estimated from

$$\mathbf{V}_0 + \{\mathbf{K}_{\text{ep}}(\mathbf{U}_0)\}^{-1} \frac{\dot{\mathbf{F}}_0^{\text{ext}}}{ht}$$

where  $\mathbf{U}_0$  are the displacements at  $t = 0$  and  $\dot{\mathbf{F}}_0^{\text{ext}}$  is the rate of the external load at  $t = 0$ . Note that this initial velocity should be computed whenever the type (or rate) of external loading changes (e.g., from body force loading to pressure loading).

Inspection of equation (15) reveals that the velocity computed from the Newton–Raphson iteration  $\mathbf{V}_1$  will approach

$$\mathbf{V}_0 = \{\mathbf{K}_{\text{ep}}^{-1}(\mathbf{U}_0)\} \frac{-\mathbf{F}_0^{\text{unb}}}{h_1}$$

as the step size  $h_1$  approaches zero. Therefore,  $\mathbf{V}_1$  may drift away from  $\mathbf{V}_0$  for very small  $h_1$ . To avoid this problem an initial set of displacements, corresponding to the unbalanced force at the start of the subincrement, is also computed as

$$\Delta \mathbf{U}_0^{\text{unb}} = \{\mathbf{K}_{\text{ep}}(\mathbf{U}_0)\}^{-1} \mathbf{F}_0^{\text{unb}}$$

The displacement error for the very first subincrement is then found using

$$\mathbf{E}_0 = \frac{1}{2} h_1 \left( \mathbf{V}_0 - \mathbf{V}_1 - \frac{\Delta \mathbf{U}_0^{\text{unb}}}{h_1} \right) \quad (18)$$

This error estimate ensures that the absolute error approaches zero as the step size approaches zero, and is employed for the very first subincrement when the total displacement at the start of the subincrement is zero.

## E. Implementation

The automatic Newton–Raphson algorithm described above requires the user to specify a series of coarse load increments. These are then automatically subincremented, if necessary, so that the relative load path error in the computed deflections is close to a user–specified tolerance  $DTOL$ . This tolerance governs the number of load subincrements that are generated, with smaller values resulting in larger numbers of subincrements. Typical values for  $DTOL$  lie in the range  $10^{-2}$  to  $10^{-4}$ . To avoid the build up of excessive unbalanced forces, which may cause the automatic error control to try to select an infinitesimal step size, the iteration tolerance should be set so that  $ITOL \leq DTOL$ .

The complete load integration algorithm can be implemented as follows:

1. Enter with the time at the start of the coarse increment  $t_0$ , the current stress state at each integration point  $(\boldsymbol{\sigma}_{t_0}, \kappa_{t_0})$ , the current displacements  $\mathbf{U}_{t_0}$  and their corresponding velocities  $\mathbf{V}_{t_0}$ , the coarse time increment  $\Delta t$ , the last trial time increment  $h_{\text{trial}}$ , the external forces over

the current coarse increment  $\mathbf{F}^{\text{ext}}$ , the iteration tolerance  $ITOL$ , and the displacement error tolerance  $DTOL$ .

2. Set  $t = t_0$  and  $h = \min\{h_{\text{trial}}, \Delta t\}$ .
3. While  $t < t_0 + \Delta t$  complete steps 4 to 10.
4. Compute  $\mathbf{V}_{t+h}$  for  $\bar{\mathbf{U}}_{t+h}$  using the Newton–Raphson algorithm. If the solution fails to converge, set  $h = 0.25h$  and try again.
5. Estimate the local error for current load subincrement using

$$E_{t+h} = \frac{h}{2} \|\mathbf{V}_{t+h} - \mathbf{V}_t\|$$

6. Compute relative error for current load subincrement using

$$R_{t+h} = \max\{EPS, E_{t+h}/\|\bar{\mathbf{U}}_{t+h}\|\}$$

where  $EPS$  is a machine constant representing the smallest relative error that can be computed.

7. If  $R_{t+h} \leq DTOL$  go to step 8. Otherwise current load subincrement has failed, so estimate a smaller pseudo time step using

$$q = \max\{0.8\sqrt{DTOL/R_{t+h}}, 0.1\}$$

$$h \leftarrow qh$$

and return to step 4 to repeat the subincrement.

8. Current load subincrement is successful. Update displacements and stress states using

$$\mathbf{U}_{t+h} = \bar{\mathbf{U}}_{t+h}$$

$$\boldsymbol{\sigma}_{t+h} = \bar{\boldsymbol{\sigma}}_{t+h}$$

$$\boldsymbol{\kappa}_{t+h} = \bar{\boldsymbol{\kappa}}_{t+h}$$

9. If the previous subincrement was unsuccessful, go to step 10. Otherwise, estimate a new subincrement size by computing

$$q = \min\{0.8\sqrt{DTOL/R_{t+h}}, 2.0\}$$

and then setting

$$h \leftarrow qh$$

10. Store the trial subincrement size  $h_{\text{trial}} = h$  and, before returning to step 3, update time and check that integration does not proceed beyond  $t_0 + \Delta t$  by setting

$$h \leftarrow \min\{h, t_0 + \Delta t - t\}$$

$$t \leftarrow t + h$$

11. Exit with displacements  $\mathbf{U}_{t_0+\Delta t}$  and the stress states  $(\boldsymbol{\sigma}_{t_0+\Delta t}, \boldsymbol{\kappa}_{t_0+\Delta t})$  at the end of the coarse load increment.

In step 1, the last trial subincrement size for the very first load step can be set to  $\Delta t$ . The Newton–Raphson procedure for solving the displacement velocities in step 4 may be summarized as follows:

1. Enter with the current stress states  $(\boldsymbol{\sigma}_t, \boldsymbol{\kappa}_t)$ , the current step size  $h$ , the external forces at the start and end of the current step  $\mathbf{F}_t^{\text{ext}}$  and  $\mathbf{F}_{t+h}^{\text{ext}}$ , the internal forces at the start of the current step  $\mathbf{F}_t^{\text{int}}$  and  $\mathbf{F}_{t+h}^{\text{ext}}$ , and the iteration error tolerance  $ITOL$ .
2. Set

$$\begin{aligned}\mathbf{V}_{t+h}^0 &= \mathbf{V}_t \\ \bar{\mathbf{U}}_{t+h}^0 &= \mathbf{U}_t + h\mathbf{V}_{t+h}^0\end{aligned}$$

3. Compute

$$\begin{aligned}\Delta\mathbf{U}^0 &= \mathbf{V}_{t+h}^0 h \\ \Delta\mathbf{e} &= \mathbf{B}\Delta\mathbf{u}^0 \\ \bar{\mathbf{s}}_{t+h} &= \mathbf{s}_t + \Delta\mathbf{s} \\ \mathbf{F}_{t+h}^{\text{int}} &= \mathbf{F}_{t+h}^{\text{int}}(\bar{\mathbf{s}}_{t+h})\end{aligned}$$

4. Repeat steps 5 to 8 for  $i = 1$  to  $MAXITS$ .
5. Compute residual vector

$$\mathbf{R}^i = \frac{\mathbf{F}_{t+h}^{\text{ext}} - \mathbf{F}_{t+h}^{\text{int}}}{h}$$

6. If

$$h\|\mathbf{R}^i\|_2 \leq ITOL\|\mathbf{F}^{\text{ext}}\|_2$$

and  $i > 1$ , go to step 10.

7. Otherwise, solve iterative velocity increments

$$\delta\mathbf{V} = \{\mathbf{K}_{\text{ep}}\}^{-1} \mathbf{R}^i$$

8. Update the velocities, the displacements, the stress states, and the internal forces

$$\begin{aligned}\mathbf{V}_{t+h}^i &= \mathbf{V}_{t+h}^{i-1} + \delta\mathbf{V} \\ \Delta\mathbf{U}^i &= \mathbf{V}_{t+h}^i h \\ \Delta\mathbf{e} &= \mathbf{B}\Delta\mathbf{u}^i\end{aligned}$$

$$\begin{aligned}\bar{\boldsymbol{\sigma}}_{t+h} &= \boldsymbol{\sigma}_t + \Delta\boldsymbol{\sigma} \\ \bar{\kappa}_{t+h} &= \kappa_t + \Delta\kappa \\ \mathbf{F}_{t+h}^{\text{int}} &= \int \mathbf{B}^T \bar{\boldsymbol{\sigma}}_{t+h} dV \\ \bar{\mathbf{U}}_{t+h} &= \mathbf{U}_t + \Delta\mathbf{U}^i\end{aligned}$$

9. Maximum number of iterations exceeded. Exit with “no convergence” warning.  
10. Exit with  $\mathbf{V}_{t+h}$ ,  $\bar{\mathbf{U}}_{t+h}$ ,  $\bar{\boldsymbol{\sigma}}_{t+h}$ , and  $\bar{\kappa}_{t+h}$ .

The maximum number of iterations in step 9 above is set to 10 in this article. This number does not affect the accuracy and robustness of the algorithm, but does somewhat influence its efficiency. In general, this number should not be too large to avoid unnecessary iterations. In addition, if the norm of  $\mathbf{R}$  is not reduced by a factor of 0.5 in two consecutive iterations, the Newton–Raphson iteration is considered to be a failure and is exited (step 9). This is again to avoid unnecessary iterations.

#### IV. APPLICATIONS

In this section, the three Newton–Raphson schemes described in the previous sections are used to analyze a number of practical problems. The line search technique is used with the modified Newton–Raphson scheme instead of the standard Newton–Raphson scheme, as suggested by Crisfield [1]. The performance of the three schemes is compared with the two automatic schemes developed by Abbo and Sloan [8,9]. These two automatic schemes, which use the explicit forward Euler and the modified Euler pair to estimate the local displacement error, were recently found to outperform most standard incremental and iterative schemes for critical state models [10].

In the following analyses, both the classic Mohr–Coulomb model and advanced critical state models are used to represent soil behavior. These constitutive laws are all integrated using an explicit substepping scheme with error control [11]. This scheme is a refined version of the algorithm originally published by Sloan [12], and uses the Euler-modified Euler pair with adaptive substepping to control the integration error in the computed stresses. In the iterative load stepping schemes, the constitutive relationships are always integrated using incremental, rather than iterative, strains. As noted by Crisfield [1], this implies that the iterative stress states are found by integrating the constitutive law from the last converged equilibrium state and avoids the complicated strain paths and spurious unloading that occurs when iterative strains are used. For the results presented in this article, the constitutive laws are integrated very accurately by using a stress error tolerance,  $STOL$ , of  $10^{-4}$  in conjunction with a yield surface drift tolerance,  $YTOL$ , of  $10^{-9}$  (see Sloan et al. [11]). For the iterative schemes, the iteration tolerance  $ITOL$  in equations (4) and (16) is set to  $10^{-3}$ . If convergence is not obtained after 1000 iterations in one single increment, the analysis is halted and regarded as a failure. For the automatic schemes, the displacement error tolerance,  $DTOL$ , is set to  $10^{-3}$ . To investigate the effect of this tolerance, an additional run with  $DTOL = 10^{-1}$  is made for each set of analyses using the automatic Newton–Raphson scheme. Typical values for  $DTOL$  lie in the range  $10^{-2}$  to  $10^{-4}$ , with smaller values resulting in more accurate solutions and more load substeps.

Due to the rarity of analytical solutions for problems using critical state models, the accuracy of the various load stepping schemes is checked by comparing their results against an accurate reference solution. Where possible, the latter is obtained from a Newton-Raphson scheme with 1000 load increments, a stress tolerance of  $10^{-6}$ , and an iteration tolerance of  $10^{-6}$ . For cases where the Newton-Raphson method fails, the reference solution is found using the incremental automatic scheme of Abbo and Sloan [8] with a stringent error tolerance of  $DTOL = 10^{-6}$ .

In the tables that follow, all the CPU times presented are for a Pentium III 700 MHz processor with 256 Mb of memory and the Compaq Visual FORTRAN compiler.

### A. Expansion of Thick Cylinder

The expansion of a thick cylinder is a simple test problem. The finite element mesh, shown in Figure 1, models the geometry of the axisymmetric cylinder with 6-noded triangles. The external load is applied in the form of prescribed displacements at the cavity wall and these are increased until the cavity pressure  $\sigma'_c$  reaches its limiting value. To compute a uniform cavity pressure which is equivalent to the imposed cavity displacements, the radial reactions at the inner nodes are summed and divided by the cavity surface area. The cavity displacements are first applied in 10 equal increments and the problem is analyzed using the Newton-Raphson method (NR), the modified Newton-Raphson method with line search (MNRLS), the automatic Newton-Raphson method (AUTONR), the automatic incremental method (AUTOINC) of Abbo and Sloan [8], and the automatic iterative method (AUTOITR) of Abbo and Sloan [9]. Each of these analyses is then repeated using 50 equal sized load increments. Note that, when applying the automatic schemes, the imposed displacement increments actually correspond to a series of trial load increments which are, if required, subdivided into a series of smaller subincrements. Depending on the local nonlinearity of the load deformation response, the size of these subincrements may vary substantially over a given trial load step.

In the following, the cylinder is assumed to be composed of a medium sand, a normally consolidated clay, and an overconsolidated clay. The behavior of the sand is represented by a nonassociated Mohr-Coulomb model, while the two clays are modeled by the Modified Cam Clay model. The results obtained for each of these cases are discussed separately.

#### 1. Expansion of Thick Cylinder in Medium Sand (MC-MS)

In the first set of cylinder analyses, a nonassociated Mohr-Coulomb model is used to simulate the expansion of the cylinder in a medium density sand. The material parameters used in the study are

$$E = 500 \text{ kPa}, c = 1 \text{ kPa}, \phi = 30^\circ, \psi = 10^\circ, \mu = 0.3$$

where  $E$  is Young's modulus,  $c$  is the cohesion,  $\mu$  is Poisson's ratio,  $\phi$  is the friction angle, and  $\psi$  is the dilation angle. The initial stress state is assumed to be uniformly isotropic and the radial, axial, and hoop components are all equal to 5 KPa. A total displacement of  $0.1a$ , which is sufficient to initiate collapse, is imposed at the inner surface of the cylinder.

The results for this problem are shown in Table 1. The standard Newton-Raphson scheme (NR) and the modified Newton-Raphson scheme with line search (MNRLS) use similar numbers

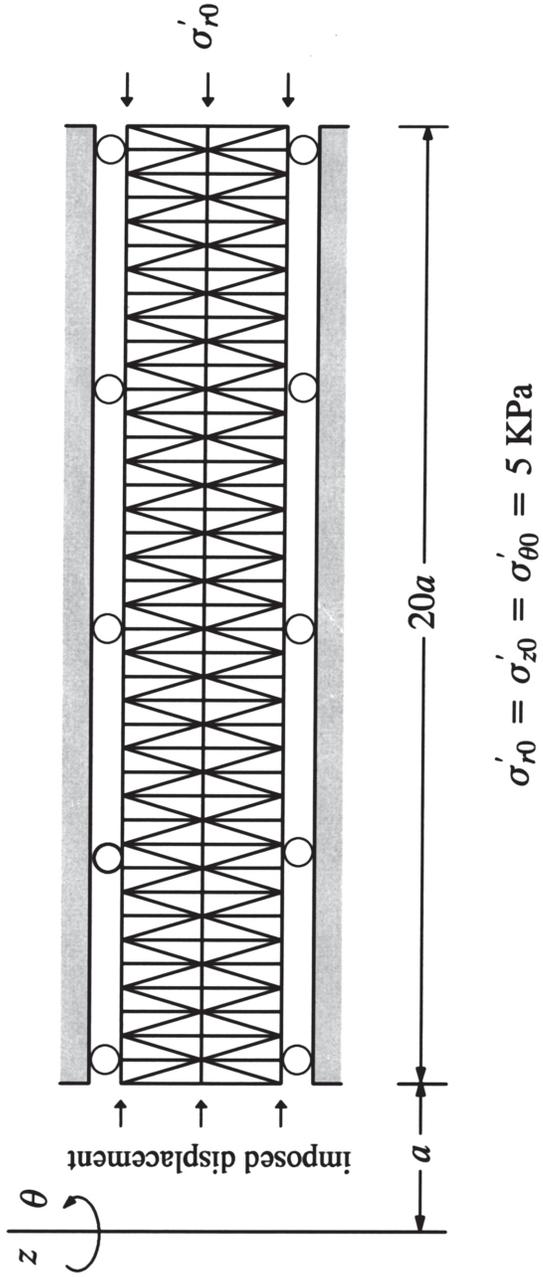


FIGURE 1. Thick cylinder mesh (160 elements, 405 nodes, 648 degrees of freedom).

**TABLE 1**  
**Results for Thick Cylinder of Medium Sand (MC-MS)**  
**( $E = 5000$  KPa,  $c = 1$  KPa,  $\phi = 30^\circ$ ,  $\psi = 10^\circ$ ,  $\mu = 0.3$ )**

Scheme	Total Iterations		CPU Time (sec)		Final Cavity Pressure (kPa)	
	10 inc's	50 inc's	10 inc's	50 inc's	10 inc's	50 inc's
NR	47	99	2	3	49.520	49.520
MNRLS	46	99	2	3	49.520	49.520
AUTONR <sup>1</sup>	36(13 <sup>3</sup> )	51(50)	2	3	49.520	49.520
AUTONR <sup>2</sup>	57(53)	77(75)	3	4	49.520	49.520
AUTOINC	0(40)	0(85)	2	5	49.520	49.520
AUTOITR	93(46)	163(88)	4	8	49.520	49.520
Reference Solution <sup>4</sup>	1356		52		49.520	

<sup>1</sup>  $DTOL = 10^{-1}$ .

<sup>2</sup>  $DTOL = 10^{-3}$ .

<sup>3</sup> Total number of subincrements.

<sup>4</sup> Computed using NR iteration with 1000 load increments and  $ITOL = 10^{-6}$ ,  $STOL = 10^{-6}$ .

of iterations, with about 5 iterations per load increment for the 10 increment analysis and only 2 iterations per increment for the 50-increment analysis. The automatic Newton–Raphson scheme with  $DTOL = 10^{-1}$  (AUTONR<sup>1</sup>) requires the fewest iterations for the analyses with 10 and 50 increments. This scheme does not generate any subincrements for the 50-increment analysis and requires fewer iterations than the standard Newton–Raphson scheme. The reason for the latter is that the starting procedure used in the AUTONR scheme, defined by equations (12) and (13), serves to accelerate the convergence of the Newton-Raphson solver. The automatic Newton–Raphson scheme with  $DTOL = 10^{-3}$  (AUTONR<sup>2</sup>) uses about 50% more iterations than the AUTONR<sup>1</sup> alternative, but gives identical cavity collapse pressures. The automatic iterative scheme AUTOITR of Abbo and Sloan [9] requires more iterations than the NR scheme, and more load subincrements than AUTONR<sup>2</sup> (which was run with the same value of  $DTOL$ ) for the 50-increment analysis. The automatic (noniterative) incremental scheme AUTOINC of Abbo and Sloan [8] generates comparatively high load subincrements.

In terms of CPU time used, the NR, MNRLS, and AUTONR<sup>1</sup> schemes use the same amount of CPU time for both the 10- and 50-increment analyses. The AUTONR<sup>2</sup> scheme is about 1.5 to 1.3 times slower than the NR scheme. Abbo and Sloan's automatic scheme AUTOITR is the slowest for this problem.

The final cavity pressures shown in Table 1 are identical for all the schemes. Interestingly, the accuracy of the automatic Newton–Raphson scheme is not sensitive to the displacement error tolerance  $DTOL$ . The cavity expansion curves plotted in Figure 2 and Figure 3 also indicate that all the schemes give similar results for this simple problem.

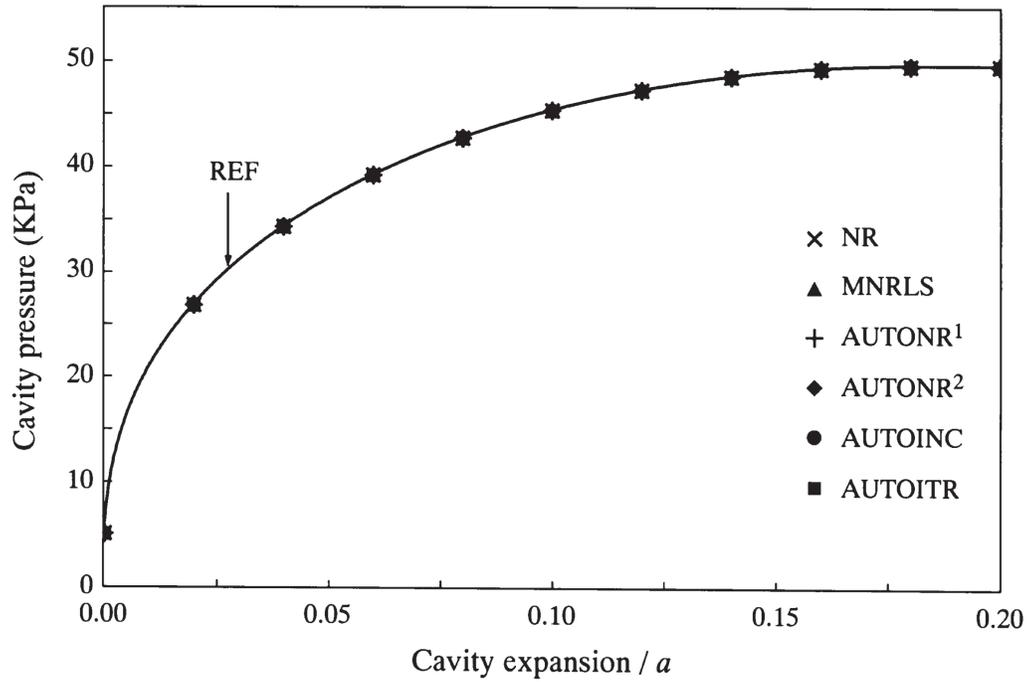


FIGURE 2. Cavity expansion curves for cylinder of medium sand (MC-MS, 10 load increments).

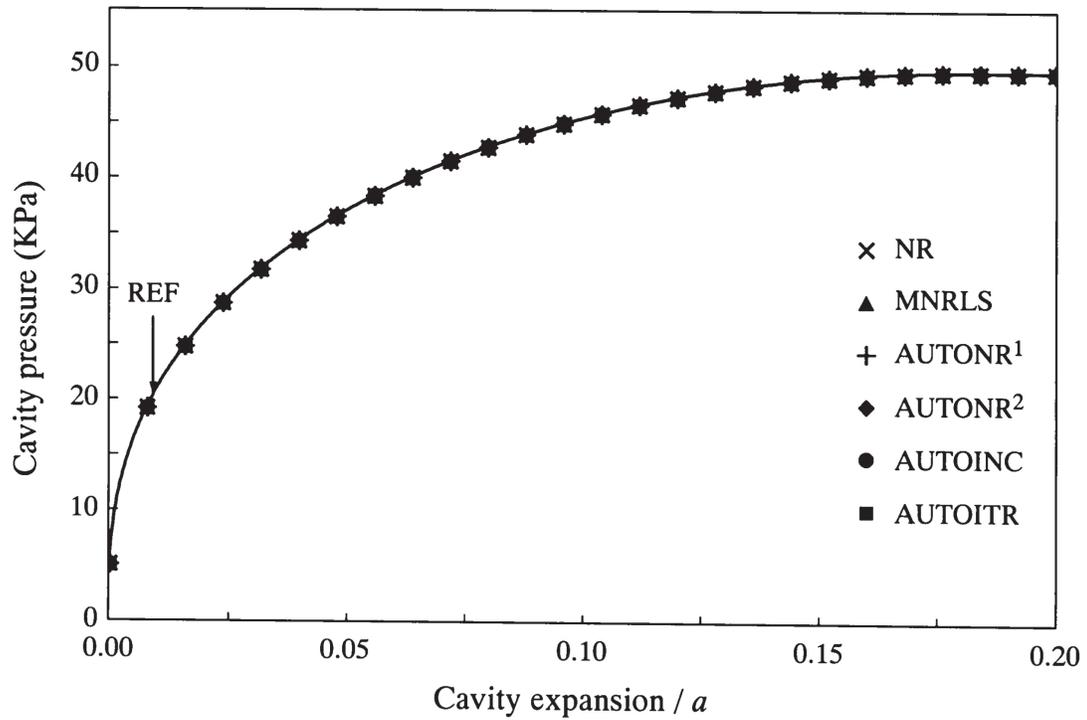


FIGURE 3. Cavity expansion curves for cylinder of medium sand (MC-MS, 50 load increments).

## 2. Expansion of Thick Cylinder in Normally Consolidated Clay (MCC-NC)

In this example, the material behavior is simulated by the classical Modified Cam Clay model (MCC) of Roscoe and Burland [13]. An associated flow rule is used and a rounded Mohr-Coulomb failure surface is employed in the deviatoric plane (Sheng et al., [14]). The material parameters used are

$$M = 0.772, \lambda = 0.1, \kappa = 0.01, \mu = 0.3, N = 3.0$$

where  $M$  is the slope of the critical state line with a value corresponding to an internal friction angle of  $20^\circ$ ,  $\lambda$  is the slope of the isotropic normal compression line,  $\kappa$  is the slope of the unloading/reloading line,  $\mu$  is Poisson's ratio, and  $N$  is the value of the specific volume on the isotropic normal compression line at unit effective mean stress. The three principal initial stresses in the cylinder are again set to 5 KPa to give an isotropic stress state. A total radial expansion of  $6a_0$  is imposed at the inner radius.

The results for this problem, shown in Table 2, indicate that each of the load stepping schemes furnishes a solution for both the 10- and 50-increment analyses. Of the iterative procedures, the automatic Newton-Raphson scheme with  $DTOL = 10^{-1}$  uses the least iterations for the two sets of analyses, while the standard Newton-Raphson scheme uses the most iterations. The modified Newton-Raphson scheme with line search uses about one fourth to one third of the iterations of the standard Newton-Raphson scheme, with the biggest savings for the analysis with the largest load steps. The automatic Newton-Raphson scheme requires about 13% more iterations when  $DTOL$  is decreased from  $10^{-1}$  to  $10^{-3}$ . Compared with the automatic schemes AUTOINC and AUTOITR, the automatic Newton-Raphson scheme with  $DTOL = 10^{-3}$  generates

**TABLE 2**  
**Results for Thick Cylinder of Normally Consolidated Clay (MCC-NC)**  
**( $M = 0.772, \lambda = 0.1, \kappa = 0.01, \mu = 0.3, N = 3.0$ )**

Scheme	Total Iterations		CPU Time (sec)		Final Cavity Pressure (kPa)	
	10 inc's	50 inc's	10 inc's	50 inc's	10 inc's	50 inc's
NR	3897	2157	5914	1009	24,700	24.763
MNRLS	941	753	1737	418	24.700	24.763
AUTONR <sup>1</sup>	572(458 <sup>3</sup> )	561(460)	197	171	24.766	24.767
AUTONR <sup>2</sup>	647(551)	638(542)	190	163	24.766	24.766
AUTOINC	0(57)	0(105)	34	29	25.336	24.808
AUTOITR	1358(58)	2042(108)	694	615	24.751	24.763
Reference Solution <sup>4</sup>	7845		483		24.766	

<sup>1</sup>  $DTOL = 10^{-1}$ .

<sup>2</sup>  $DTOL = 10^{-3}$ .

<sup>3</sup> Total number of subincrements.

<sup>4</sup> Computed using NR iteration with 1000 load increments and  $ITOL = 10^{-6}$ ,  $STOL = 10^{-6}$ .

5 to 10 times more subincrements, which suggests that its error control is more stringent. Comparing the results for the analyses using 10 and 50 coarse increments, respectively, we note that the total iteration and subincrement counts for the automatic Newton–Raphson scheme are not significantly influenced by the number of coarse load increments. This feature is not observed in the other schemes, and suggests the automatic Newton–Raphson algorithm has superior error control.

In terms of CPU time, the Sloan and Abbo scheme AUTOINC is the fastest while the NR scheme is the slowest. The MNRLS scheme gives a saving of 60 to 70% in CPU time over the NR scheme. Relative to the NR scheme, the automatic Newton–Raphson scheme with  $DTOL = 10^{-1}$  furnishes CPU time savings in the range 84 to 97%. Interestingly, tightening  $DTOL$  from  $10^{-1}$  to  $10^{-3}$  with the automatic Newton–Raphson procedure reduces the CPU time by about 5%. This reflects the fact that the Newton–Raphson iterations may converge more quickly with a good initial guess. Relative to the automatic schemes of Abbo and Sloan, the automatic Newton–Raphson scheme with  $DTOL = 10^{-3}$  is faster than the AUTOITR scheme but slower than the AUTOINC scheme.

In terms of accuracy, all the schemes give accurate predictions of the final cavity pressure. The maximum deviation from the reference solution is less than 3%, and occurs for the AUTOINC scheme with the 10-increment analysis. The complete load–displacement curves in Figure 4 and Figure 5 confirm that all the schemes give solutions of acceptable accuracy for this problem.

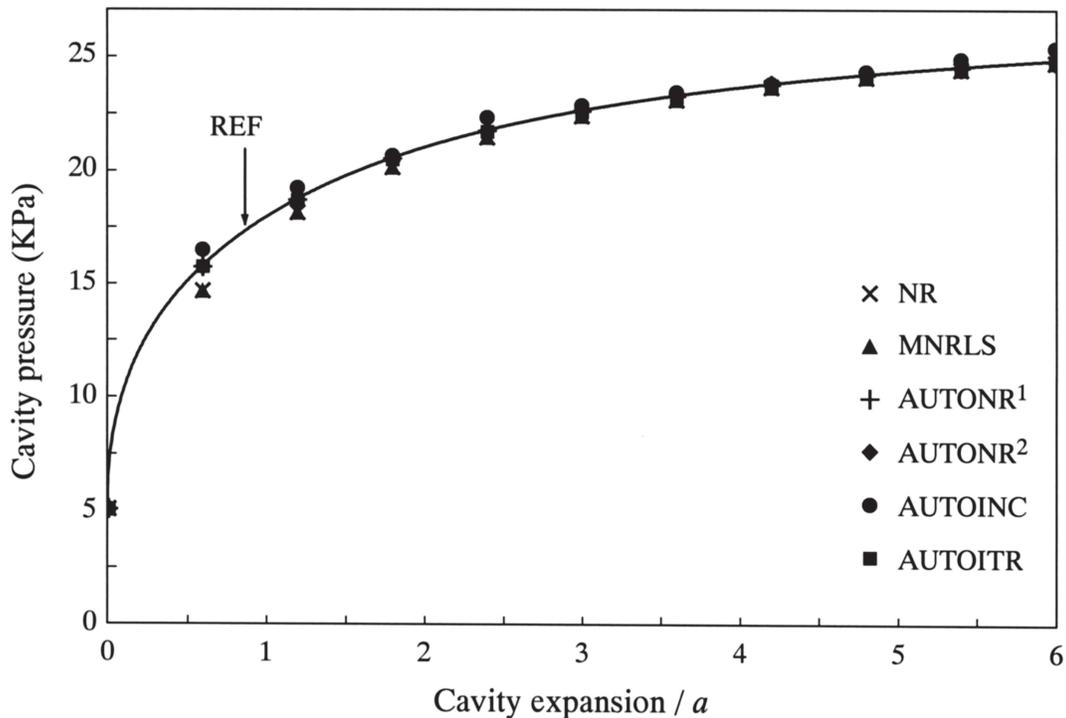


FIGURE 4. Cavity expansion curves for cylinder of normally consolidated clay (MCC–NC, 10 load increments).

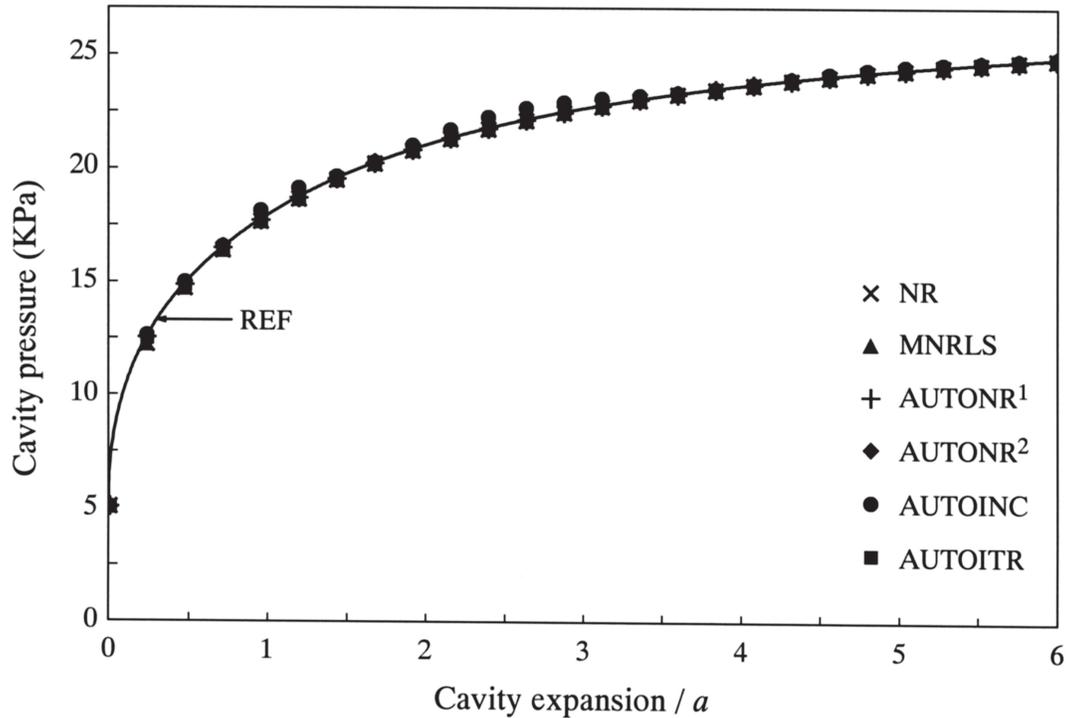


FIGURE 5. Cavity expansion curves for cylinder of normally consolidated clay (MCC-NC, 50 load increments).

### 3. Expansion of Thick Cylinder in Overconsolidated Clay (MCC-OC)

In the last cavity expansion example the Modified Cam Clay model, with a rounded Mohr-Coulomb yield surface in the deviatoric plane, is used to simulate the expansion of a cylinder of overconsolidated clay. The material parameters and initial stress state used are the same as for the previous case, with the only difference being that the overconsolidation ratio is set to 10 to determine the initial yield surface location. The final radial expansion is reduced from  $6a$  to  $1a$ .

The numerical results for the various MCC-OC analyses are presented in Table 3. Overall, the iteration counts, the CPU times, and the final cavity pressures display similar trends to those observed for the case of the normally consolidated clay (MCC-NC). The MNRLS scheme uses about one third of the iterations of the standard NR scheme for the 10-increment analysis, but uses only 5% fewer iterations for the 50-increment analysis. The automatic Newton-Raphson scheme with  $DTOL = 10^{-1}$  again uses the least iterations among all the iterative schemes. Tightening the displacement error tolerance to  $DTOL = 10^{-3}$  with this method results in a negligible increase in the iteration count. Compared to the Abbo and Sloan schemes, the automatic Newton-Raphson scheme again generates more load subincrements.

In terms of CPU time, the incremental automatic scheme AUTOINC is again the fastest, while the NR is again the slowest. The MNRLS scheme gives a saving of 60% in CPU time over the NR scheme for the 10-increment analysis, and 15% for the 50-increment analysis. The automatic Newton-Raphson scheme with  $DTOL = 10^{-1}$  scheme gives a saving of 85% in CPU time over the NR scheme for the 10-increment analysis, and about 29% for the 50-increment

**TABLE 3**  
**Results for Thick Cylinder of Overconsolidated Clay (MCC–OC)**  
**( $M = 0.772$ ,  $\lambda = 0.1$ ,  $\kappa = 0.01$ ,  $\mu = 0.3$ ,  $N = 3.0$ ,  $OCR = 10$ )**

Scheme	Total Iterations		CPU Time (sec)		Final Cavity Pressure (kPa)	
	10 inc's	50 inc's	10 inc's	50 inc's	10 inc's	50 inc's
NR	717	372	346	52	50.364	50.371
MNRLS	240	354	132	45	50.364	50.371
AUTONR <sup>1</sup>	226(184 <sup>3</sup> )	239(132)	51	39	50.373	50.373
AUTONR <sup>2</sup>	236(192)	368(54)	47	71	50.372	50.373
AUTOINC	0(47)	0(87)	8	9	50.816	50.430
AUTOITR	628(48)	417(91)	117	44	50.359	50.371
Reference Solution <sup>4</sup>	3071		112		50.370	

<sup>1</sup>  $DTOL = 10^{-1}$ .

<sup>2</sup>  $DTOL = 10^{-3}$ .

<sup>3</sup> Total number of subincrements.

<sup>4</sup> Computed using NR iteration with 1000 load increments and  $ITOL = 10^{-6}$ ,  $STOL = 10^{-6}$ .

analysis. The AUTONR scheme with  $DTOL = 10^{-3}$  is roughly 6 to 4 times slower than the AUTOINC scheme, but is about 2.5 times faster than the AUTOITR scheme for the 10-increment analysis.

All the schemes in Table 3 predict a final cavity pressure which is close to the reference solution obtained from the standard Newton–Raphson scheme with  $ITOL = 10^{-6}$  and  $STOL = 10^{-6}$ . The least accurate prediction differs by less than 1% from the reference solution, and is given by the automatic scheme AUTOINC with 10 coarse increments. The complete load–displacement curves in Figure 6 and Figure 7 indicate that all the schemes give accurate load predictions at all displacement levels.

## B. Rigid Footing

We now consider the problem of a smooth rigid strip footing, of width  $B$ , resting on an elastoplastic soil layer. The finite element mesh (comprising 6-noded triangles) and boundary conditions for the various analyses are shown in Figure 8. To simulate the behavior of a rigid foundation, the footing is subjected to a set of uniform vertical displacements and an equivalent pressure is computed by summing the appropriate vertical nodal reactions. Because of the singularity at the footing edge and the strong rotation of the principal stresses, this problem is a challenging test for any solution scheme, especially when large load increments are used.

As in the previous cavity expansion study, each of the various load stepping schemes is run using 10 and 50 equal-sized displacement increments. The three different soil types considered

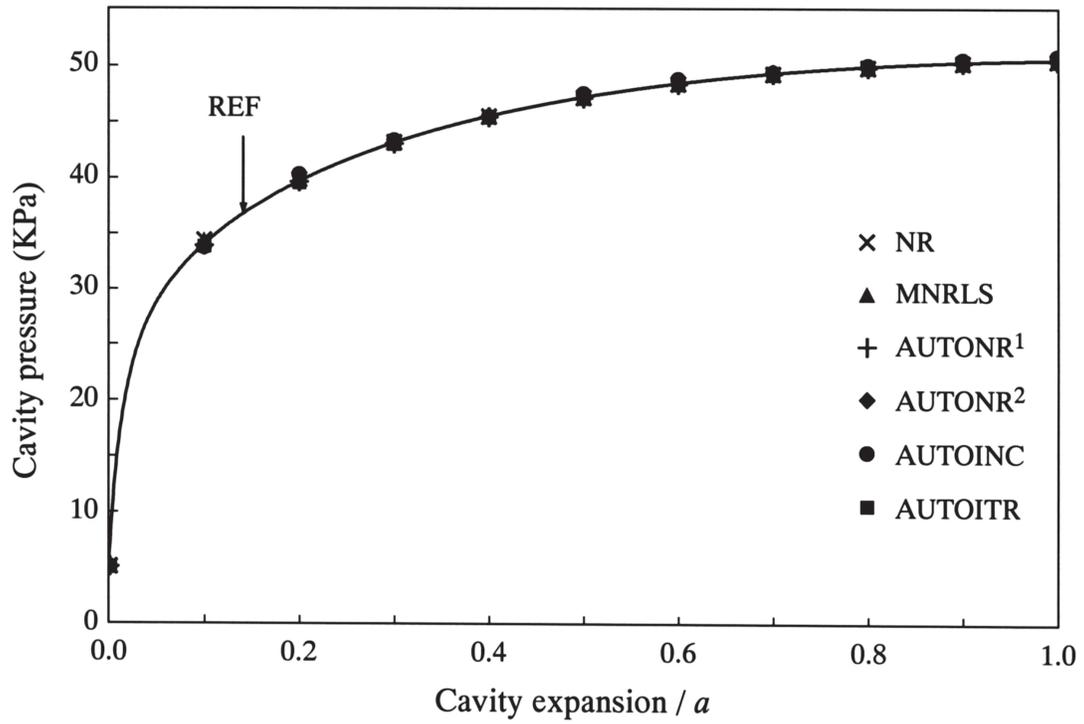


FIGURE 6. Cavity expansion curves for cylinder of overconsolidated clay (MCC-OC, 10 load increments).

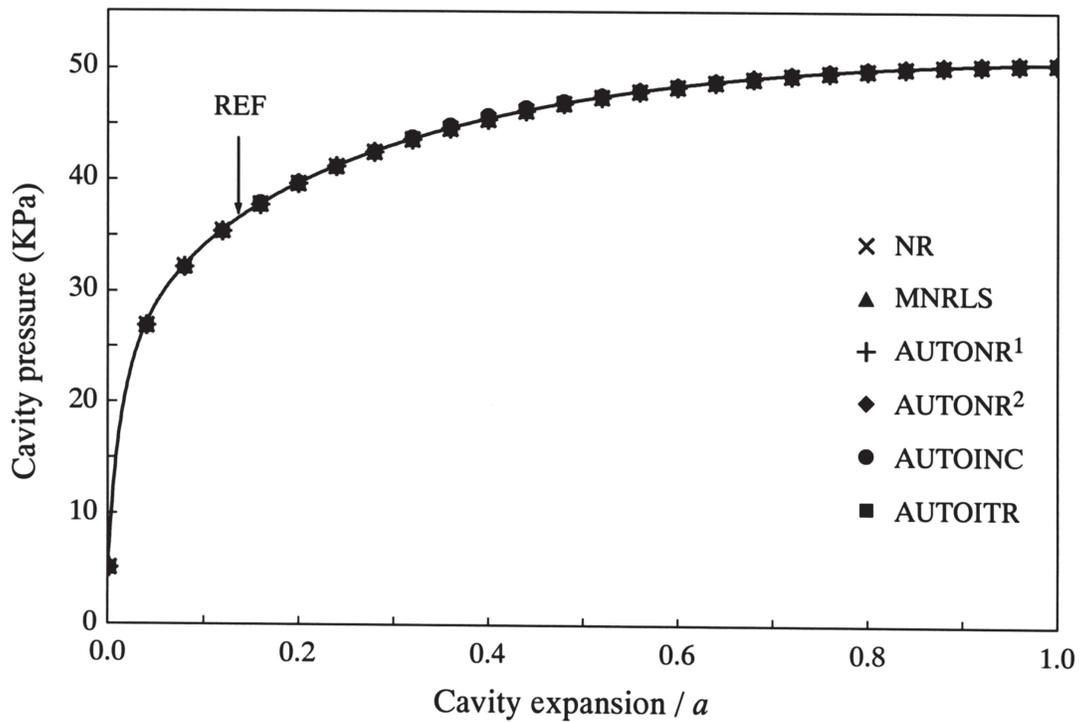


FIGURE 7. Cavity expansion curves for cylinder of overconsolidated clay (MCC-OC, 50 load increments).

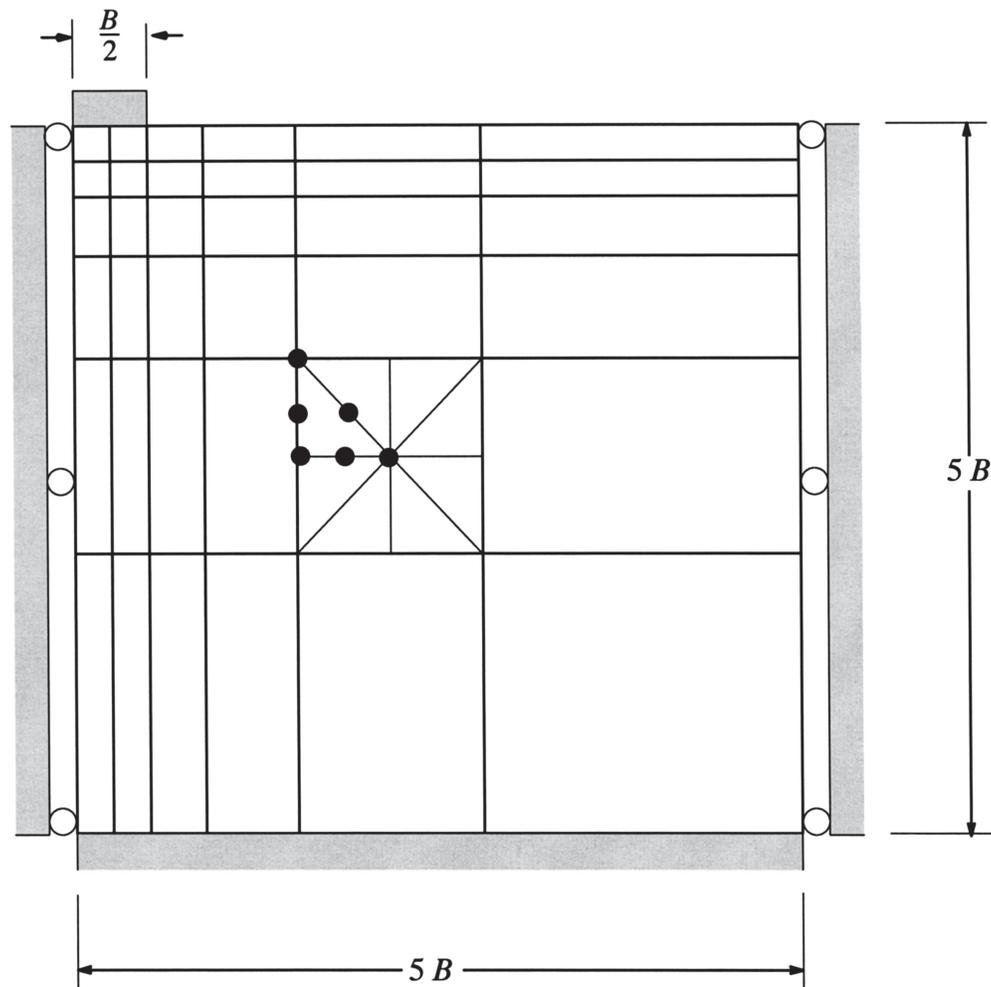


FIGURE 8. Rigid footing mesh (625 nodes, 288 elements, 1143 degrees of freedom).

are a medium density sand, a normally consolidated clay, and an overconsolidated clay. The sand is modeled by a nonassociated Mohr–Coulomb model, while the two clays are modeled using the Generalized Cam Clay constitutive law (GCC-NC and GCC-OC) of Sheng et al. [14] The total displacement imposed on the footing causes large-scale plastic yielding and is sufficient to initiate a state of collapse.

For all footing analyses, the initial stresses in the layer are generated using the body loads which correspond to the submerged soil density. During this stage, we assume that the material is elastic and apply the body loads in small increments. The Poisson ratio used in this body loading is calculated from the given lateral stress coefficient  $K_0$ . Once the initial stresses are established, the initial yield surface locations for the critical state models are determined from a given overconsolidation ratio. Note that the total iterations and CPU times given in the following tables are due only to the application of the footing load, and not the establishment of the initial stresses.

### 1. Footing on Medium Sand (MC-MS)

In the first set of footing analyses, the nonassociated Mohr–Coulomb model is used to represent the sand behavior and the material parameters are the same as those used for the thick cylinder problem. The buoyant unit weight, which is needed to establish the initial stresses, is set to 6 KN/m<sup>3</sup>. Once the initial stress field is established, a total displacement of 0.075*B*, which is sufficient to initiate collapse, is imposed on the footing.

The results, shown in Table 4, demonstrate that both the NR scheme and the MNRLS scheme with line search fail to furnish a solution for the 10-increment analyses. Among the successful iterative procedures, the AUTONR method with  $DTOL = 10^{-1}$  uses the least iterations, while the automatic scheme AUTOITR uses the most iterations. For the 50-increment analysis, the MNRLS scheme uses over four times as many iterations as the NR scheme. The automatic Newton–Raphson scheme with  $DTOL = 10^{-1}$  uses about 10% fewer iterations than the NR scheme, and about 50% fewer iterations than the AUTOITR scheme of Abbo and Sloan. Unlike previous examples, the two alternative forms of the automatic Newton–Raphson scheme uses less load subincrements than the two automatic schemes of Abbo and Sloan. As before, the total iterations and total subincrements used in the automatic Newton–Raphson scheme do not vary significantly as the number of coarse increments is increased. This suggests that the error control mechanism is working as expected.

For the 10-increment analysis, the AUTOINC scheme of Abbo and Sloan is the fastest, while their AUTOITR scheme is the slowest. For the 50-increment analysis, the Newton–Raphson scheme is the fastest, whereas the modified Newton–Raphson scheme with line search is the slowest. In terms of CPU time, the automatic Newton–Raphson scheme is competitive with all other iterative methods for both values of  $DTOL$ .

**TABLE 4**  
**Results for Rigid Footing on Medium Sand (MC-MS)**  
*(E = 5000 KPa, c = 1 KPa,  $\phi = 30^\circ$ ,  $\psi = 10^\circ$ ,  $\mu = 0.3$ )*

Scheme	Total Iterations		CPU Time (sec)		Final Footing Pressure (kPa)	
	10 inc's	50 inc's	10 inc's	50 inc's	10 inc's	50 inc's
NR	fail	167	fail	67	fail	96.346
MNRLS	fail	722	fail	186	fail	96.340
AUTONR <sup>2</sup>	143(71 <sup>3</sup> )	149(67)	106	72	96.352	96.348
AUTONR <sup>2</sup>	152(100)	158(105)	91	78	96.349	96.346
AUTOINC	0(115)	0(156)	48	72	96.335	96.333
AUTOITR	322(134)	309(181)	117	119	96.358	96.352
Reference Solution <sup>4</sup>	2579		826		96.311	

<sup>1</sup>  $DTOL = 10^{-1}$ .

<sup>2</sup>  $DTOL = 10^{-3}$ .

<sup>3</sup> Total number of subincrements.

<sup>4</sup> Computed using NR iteration with 1000 load increments and  $ITOL=10^{-6}$ ,  $STOL=10^{-6}$ .

In terms of accuracy, each successful scheme predicts a final footing pressure which is very close to the reference solution obtained from the NR scheme. The worst estimate, obtained using the AUTOITR procedure for the 10-increment run, is only 0.05% higher than the reference solution. The AUTOINC scheme provides the most accurate estimation of the final footing pressure, with a maximum overestimation of only 0.02%. The complete load–displacement curves, shown in Figure 9 for the 10-increment run and in Figure 10 for the 50-increment run, indicate that all schemes give good predictions of the footing pressures at all displacements.

## 2. Footing on Normally Consolidated Clay (GCC–NC)

In this footing example, the behavior of a normally consolidated clay is simulated using a Generalized Cam clay (GCC) model with a rounded Mohr-Coulomb yield surface and an associated flow rule (Sheng et al. [14]). The parameters adopted are as follows

$$M = 0.984, \lambda = 0.2, \kappa = 0.05, \mu = 0.3, N = 2.8, \beta' = 0.5, \gamma' = 6, K_0 = 0.72$$

where  $M$ ,  $\lambda$ ,  $\kappa$ ,  $\mu$ , and  $N$  are the same as in the Modified Cam Clay model,  $\beta'$  is used to adjust the shape of the yield surface in the meridional plane,  $\gamma'$  is the submerged unit weight, and  $K_0$  is the ratio of the horizontal and vertical *in situ* stresses. The last two parameters are used to generate

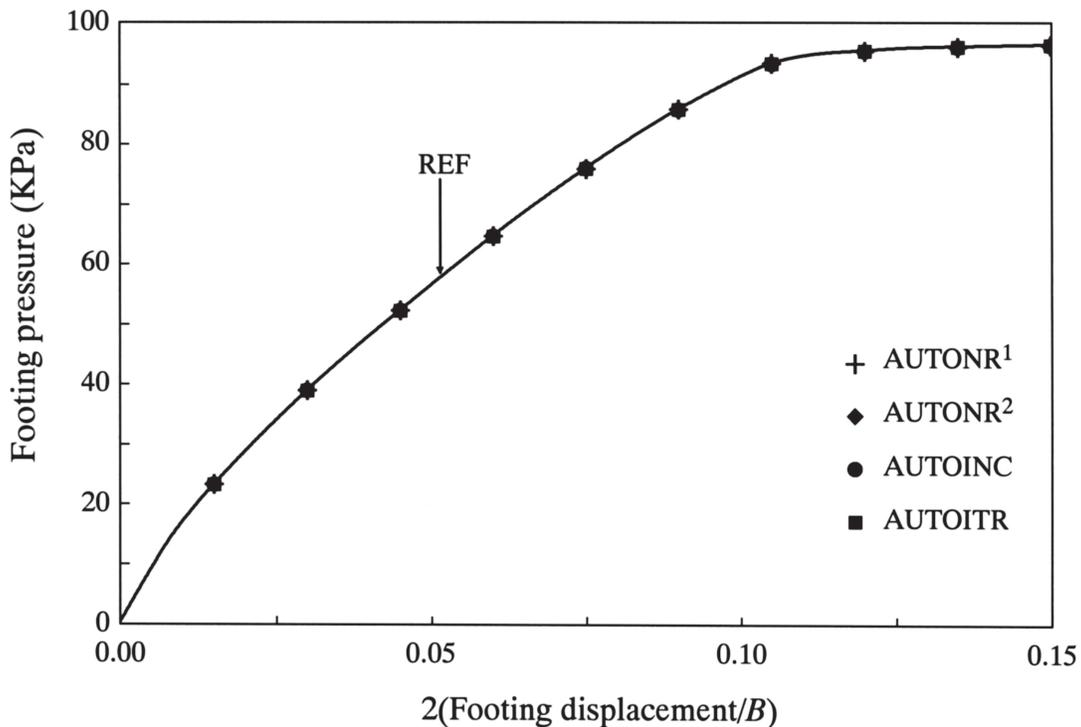


FIGURE 9. Load-displacement response of rigid footing on medium sand (MC–MS, 10 load increments).

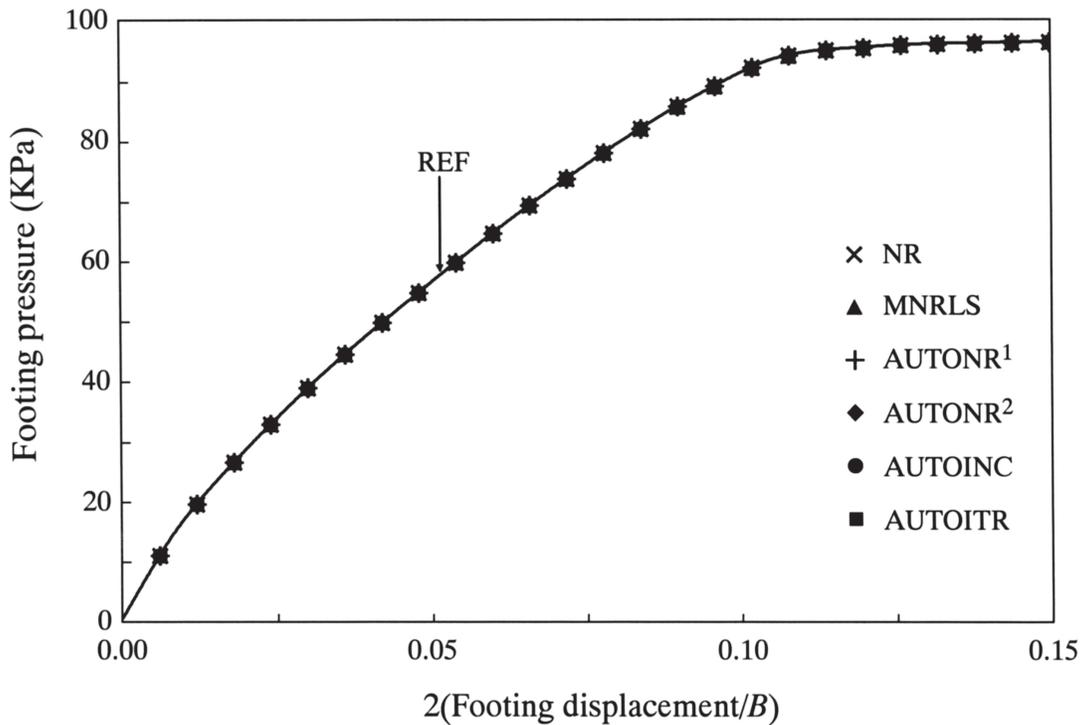


FIGURE 10. Load-displacement response of rigid footing on medium sand (MC–MS, 50 load increments).

the *in situ* stresses before the footing is displaced. Note that the Poisson ratio of  $\mu = 0.3$  is only used during footing loading, not the gravity loading. In order to initiate a state of near collapse in the soil, the footing is loaded to a total displacement of  $3B/4$ , which is 10 times larger than that for the Mohr–Coulomb sand in the previous example.

The numerical results for the various methods, shown in Table 5, show that all the schemes furnish a solution for this problem. In terms of iteration counts, the NR scheme uses the most iterations for the 10-increment analyses and the automatic scheme AUTOITR uses the most iterations for the 50-increment analyses. The automatic Newton–Raphson scheme with  $DTOL = 10^{-1}$  uses the least iterations for both sets of analyses.

The MNRLS procedure uses about 50% fewer iterations than the NR procedure for the 10-increment analysis, but uses slightly more iterations for the 50-increment analysis. Tightening the displacement error tolerance from  $10^{-1}$  to  $10^{-3}$  with the automatic Newton–Raphson scheme results in a 40 to 50% increase in iterations.

In terms of CPU time used, the fastest scheme for both sets of analyses is again the incremental automatic scheme (AUTOINC) of Abbo and Sloan. The slowest scheme for the 10-increment analysis is the Newton–Raphson scheme (NR), while the slowest method for the 50-increment analysis is the automatic Newton–Raphson scheme with  $DTOL = 10^{-3}$ . Compared with the NR scheme, the MNRLS procedure gives a saving of 27% in CPU time for the 10-increment analysis and a saving of 20% for the 50-increment analysis. A similar comparison for the automatic Newton–Raphson scheme with  $DTOL = 10^{-1}$  gives a saving of 75% for the 10-increment analysis, but a 7% increase for the 50-increment analysis.

**TABLE 5**  
**Results for Rigid Footing on Normally Consolidated Clay (GCC–NC)**  
**( $M = 0.984$ ,  $\lambda = 0.20$ ,  $\kappa = 0.05$ ,  $\mu = 0.3$ ,  $N = 2.8$ ,  $\beta' = 0.5$ ,  $\gamma' = 6$ ,  $K_0 = 0.72$ )**

Scheme	Total Iterations		CPU Time (sec)		Final Footing Pressure (kPa)	
	10 inc's	50 inc's	10 inc's	50 inc's	10 inc's	50 inc's
NR	365	171	605	93	36.837	36.867
MNRLS	197	174	403	74	36.841	36.861
AUTONR <sup>1</sup>	122(96 <sup>3</sup> )	140(77)	155	100	36.858	36.861
AUTONR <sup>2</sup>	188(163)	205(149)	136	105	36.862	36.854
AUTOINC	0(84)	0(128)	44	58	36.940	36.913
AUTOITR	318(77)	233(114)	159	103	36.893	36.867
Reference Solution <sup>4</sup>	2168		498		36.842	

<sup>1</sup>  $DTOL = 10^{-1}$ .

<sup>2</sup>  $DTOL = 10^{-3}$ .

<sup>3</sup> Total number of subincrements.

<sup>4</sup> Computed using NR iteration with 1000 load increments and  $ITOL = 10^{-6}$ ,  $STOL = 10^{-6}$ .

The complete load–displacement curves, shown in Figure 11 and Figure 12, indicate that all the schemes again give accurate predictions of the footing pressures over the full range of loading.

### 3. Footing on Overconsolidated Clay (GCC–OC)

Again we use the Generalized Cam clay model to simulate the behavior of overconsolidated clay. The material parameters are given as follows:

$$M = 0.984, \lambda = 0.1, \kappa = 0.01, \mu = 0.3, N = 2.8, \beta' = 0.5, \gamma' = 6, K_0 = 1.0$$

The overconsolidation ratio is set according to

$$OCR = (50 + p'_{in}) / p'_{in}$$

where  $p'_{in}$  is the *in situ* mean effective stress. This equation implies that, at the ground surface where  $p'_{in} = 0$ , the soil is preconsolidated to 50 kPa (about a depth of 3 m of the same soil). In order to approach a state of collapse, a total displacement of  $B/20$  is imposed on the footing, which is one fifteenth of that applied to the normally consolidated clay in the previous example.

The numerical results, shown in Table 6, indicate that the NR scheme and the MNRLS scheme both fail to give a solution for this problem, even for the runs with 50 load steps. The three

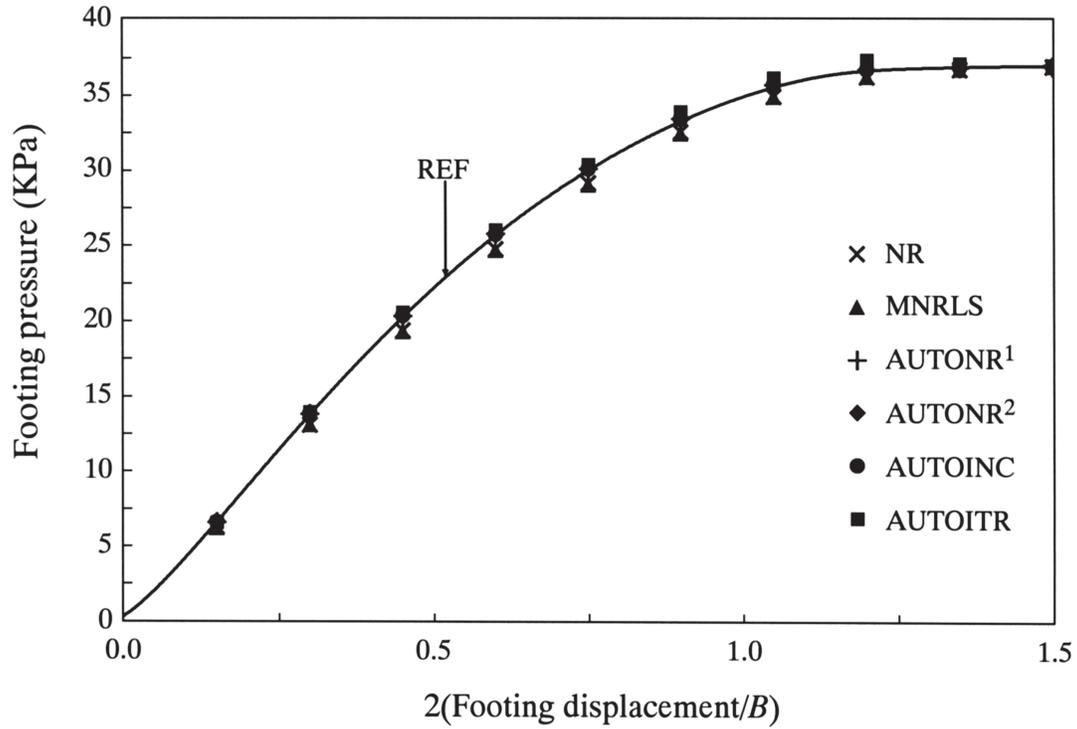


FIGURE 11. Load-displacement response of rigid footing on normally consolidated clay (GCC-NC, 10 load increments).

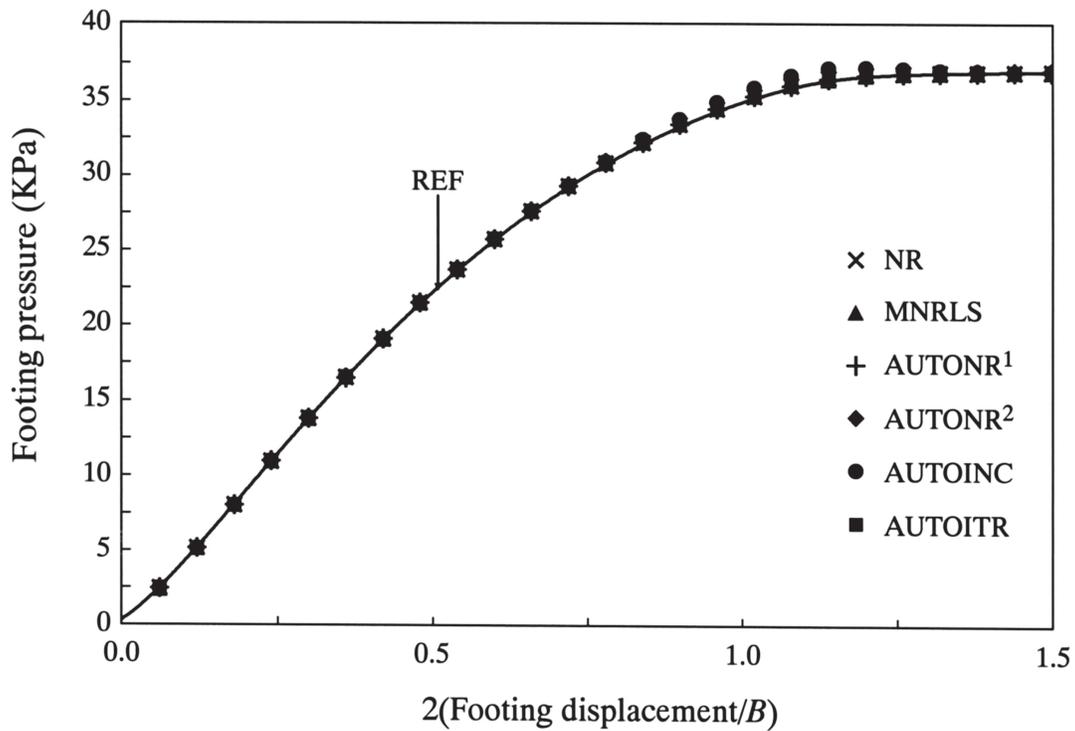


FIGURE 12. Load-displacement response of rigid footing on normally consolidated clay (GCC-NC, 50 load increments).

**TABLE 6**  
**Results for Rigid Footing on Normally Consolidated Clay (GCC-OC)**  
**( $M = 0.984$ ,  $\lambda = 0.1$ ,  $\kappa = 0.01$ ,  $\mu = 0.3$ ,  $N = 2.8$ ,  $\beta' = 0.5$ ,  $\gamma' = 6$ ,  $K_0 = 1.0$ )**

Scheme	Total Iterations		CPU Time (sec)		Final Footing Pressure (kPa)	
	10 inc's	50 inc's	10 inc's	50 inc's	10 inc's	50 inc's
NR	fail	fail	fail	fail	fail	fail
MNRLS	fail	fail	fail	fail	fail	fail
AUTONR <sup>1</sup>	156(96 <sup>3</sup> )	155(92)	93	74	66.864	66.840
AUTONR <sup>2</sup>	208(167)	213(170)	87	80	66.752	66.748
AUTOINC	0(130)	0(168)	37	50	66.735	66.906
AUTOITR	345(145)	340(192)	89	88	66.776	66.845
Reference Solution <sup>4</sup>	2521		498		66.900	

<sup>1</sup>  $DTOL = 10^{-1}$ .

<sup>2</sup>  $DTOL = 10^{-3}$ .

<sup>3</sup> Total number of subincrements.

<sup>4</sup> Computed using NR with 1000 load increments and  $ITOL = 10^{-6}$ ,  $STOL = 10^{-6}$ .

automatic schemes, on the other hand, all furnish accurate solutions. In terms of iteration counts, the automatic Newton–Raphson scheme performs better than the iterative automatic scheme of Abbo and Sloan [9]. This advantage is also reflected in the CPU times. As in most cases discussed so far, the total iteration and subincrement counts for the automatic Newton–Raphson scheme are largely insensitive to the prescribed number of coarse load increments.

In terms of accuracy, the four automatic schemes all give accurate predictions of the final footing pressure (Table 6) as well as the complete load–displacement curve (Figure 13 and Figure 14).

## V. CONCLUSIONS

This article presents an automatic Newton–Raphson method for solving nonlinear finite element equations. It automatically subincrements a series of given coarse load increments so that the local load path error in the displacements is held below a prescribed threshold. The local error is measured by taking the difference between the iterative solutions obtained from the backward Euler method and the SS21 method. By storing the nodal velocities, in addition to the nodal displacements, the error control mechanism can be implemented with negligible overhead. Some key conclusions that may be drawn from the study are:

1. The new automatic Newton–Raphson scheme is more robust and more efficient than the standard Newton–Raphson scheme or the modified Newton–Raphson scheme with line

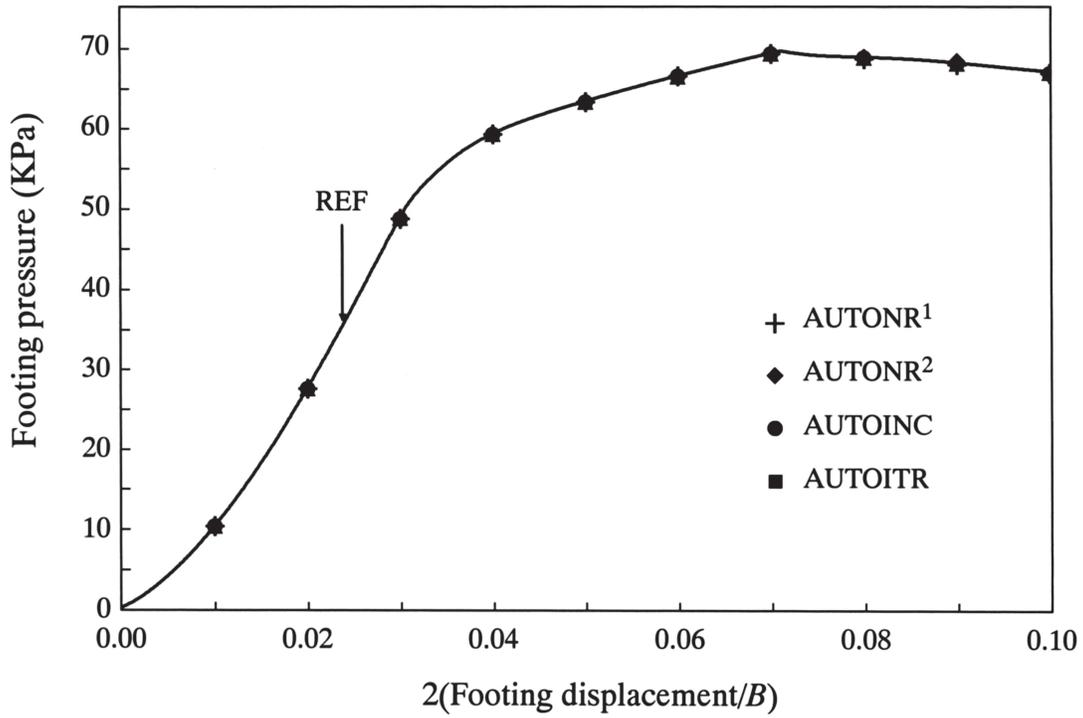


FIGURE 13. Load-displacement response of rigid footing on overconsolidated clay (GCC-OC, 10 load increments).

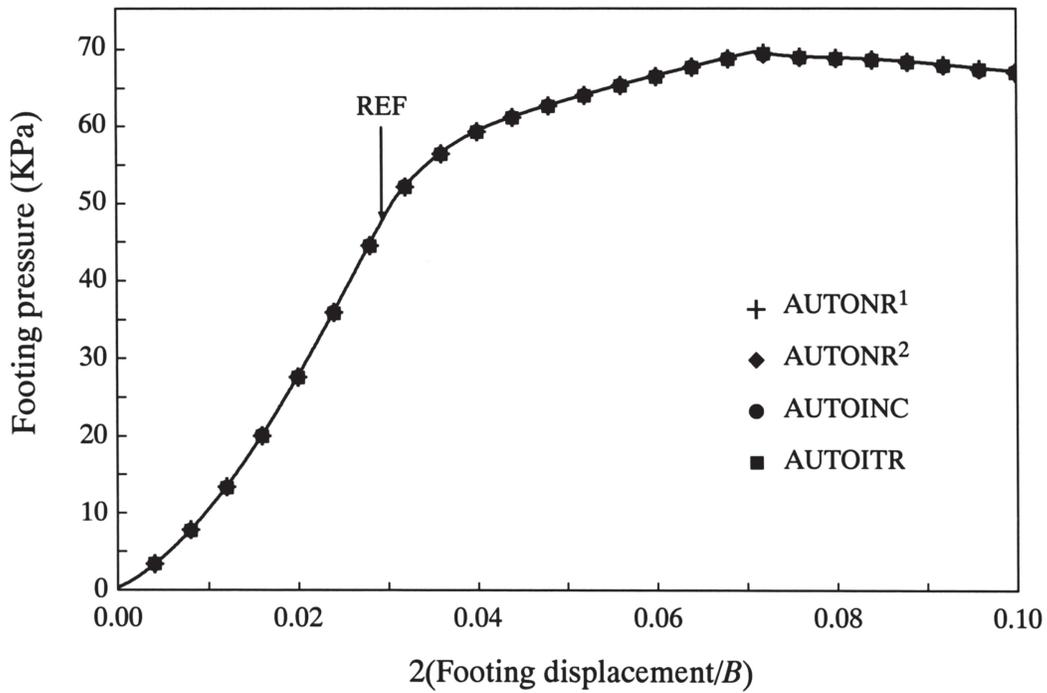


FIGURE 14. Load-displacement response of rigid footing on overconsolidated clay (GCC-OC, 50 load increments).

search. It furnishes a solution for every problem, whereas the latter two schemes fail in a number of cases. For problems involving critical state models, it can be significantly faster than the Newton–Raphson scheme and the modified Newton–Raphson scheme with line search.

2. The automatic Newton–Raphson scheme is generally faster than the iterative automatic scheme of Abbo and Sloan [9], but slower than the incremental automatic scheme of Abbo and Sloan [8]. An advantage of the proposed automatic Newton–Raphson scheme is that its accuracy is largely insensitive to the displacement error tolerance  $DTOL$ , at least for problems with moderate nonlinearity.
3. The total iteration and subincrement counts for the automatic Newton–Raphson scheme are not significantly influenced by the prescribed number of coarse load increments. This suggests its error control mechanism is robust, since it behaves as expected.
4. For the geomechanics examples considered here, the modified Newton–Raphson scheme incorporated with a line search is generally more efficient than the standard Newton–Raphson scheme. The line search, however, does not seem to improve the robustness of the Newton–Raphson scheme.

## REFERENCES

- [1] **M.A. Crisfield**, *Non-linear Finite Element Analysis of Solid and Structures*, Vol. 1, John Wiley & Sons, Chichester, 1991.
- [2] **D.M. Potts and L. Zdravkovic**, *Finite Element Analysis in Geotechnical Engineering*, Theory, Thomas Telford, London, 1999.
- [3] **H., Matthies, and G. Strang**, The solution of non-linear finite element equations, *International Journal for Numerical Methods in Engineering*, **14**, 1613–1626, 1979.
- [4] **G.A. Wempner**, Discrete approximations related to nonlinear theories of solids, *International Journal of Solids and Structures*, **7**, 1581–1599, 1971.
- [5] **E. Riks**, The application of Newton’s method to the problem of elastic stability, *Journal of Applied Mechanics*, **39**, 1060–1066, 1972.
- [6] **E. Riks**, An incremental approach to the solution of snapping and buckling problems, *International Journal of Solids and Structures*, **15**, 529–551, 1979.
- [7] **J.C. Simo and R.L. Taylor**, Consistent tangent operators for rate-independent elasto-plasticity, *Computer Methods in Applied Mechanics and Engineering*, **48**, 101–118, 1985.
- [8] **A.J. Abbo and S.W. Sloan**, An automatic load stepping algorithm with error control, *International Journal for Numerical Methods in Engineering*, **39**, 1737–1759, 1996.
- [9] **A.J. Abbo and S.W. Sloan**, Load path control of iterative solution schemes, in *Computational Plasticity, Proceedings of 5th International Conference* (D.R.J. Owen, E. Onate, and E. Hinton, Eds.), International Center for Numerical Methods in Engineering, Barcelona, Part 1, 325–333, 1997.
- [10] **D. Sheng and S.W. Sloan**, Load stepping schemes for critical state models, *International Journal for Numerical Methods in Engineering*, **50**, 67–93, 2001.
- [11] **S.W. Sloan, A.J. Abbo, and D. Sheng**, Refined explicit integration of elastoplastic models with automatic error control, *Engineering Computations*, **18**, 121–154, 2001.
- [12] **S.W. Sloan**, Substepping schemes for the numerical integration of elastoplastic stress-strain relations, *International Journal for Numerical Methods in Engineering*, **24**, 893–911, 1987.
- [13] **K.H. Roscoe and J.B. Burland**, On generalisation stress-strain behaviour of “wet” clay, *Engineering Plasticity*, Cambridge University Press, 535–560, 1968.

- [14] **D. Sheng, S.W. Sloan, and H.S. Yu**, Aspects of finite element implementation of critical state models, *Computational Mechanics*, **26**, 185–196, 2000.
- [15] **R.W. Thomas and I. Gladwell**, Variable–order variable step algorithms for second–order systems. Part 1: The methods, *International Journal for Numerical Methods in Engineering*, **26**, 39–53, 1988.
- [16] **O.C. Zienkiewicz, W.L. Wood, and R.L. Taylor**, A unified set of single step algorithms. Part 1: General formulation and applications, *International Journal for Numerical Methods in Engineering*, **20**, 1529–1552, 1984.