INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 39, 1737–1759 (1996)

AN AUTOMATIC LOAD STEPPING ALGORITHM WITH ERROR CONTROL

A. J. ABBO AND S. W. SLOAN*

Department of Civil Engineering and Surveying, University of Newcastle, Callaghan N.S.W. 2308, Australia

SUMMARY

This paper presents an algorithm for controlling the error in non-linear finite element analysis which is caused by the use of finite load steps. In contrast to most recent schemes, the proposed technique is non-iterative and treats the governing load–deflection relations as a system of ordinary differential equations. This permits the governing equations to be integrated adaptively where the step size is controlled by monitoring the local truncation error. The latter is measured by computing the difference between two estimates of the displacement increments for each load step, with the initial estimate being found from the first-order Euler scheme and the improved estimate being found from the second-order modified Euler scheme. If the local truncation error exceeds a specified tolerance, then the load step is abandoned and the integration is repeated with a smaller load step whose size is found by local extrapolation. Local extrapolation is also used to predict the size of the next load step following a successful update. In order to control not only the local load path error, but also the global load path error, the proposed scheme incorporates a correction for the unbalanced forces. Overall, the cost of the automatic error control is modest since it requires only one additional equation solution for each successful load step. Because the solution scheme is non-iterative and founded on successful techniques for integrating systems of ordinary differential equations, it is particularly robust. To illustrate the ability of the scheme to constrain the load path error to lie near a desired tolerance, detailed results are presented for a variety of elastoplastic boundary value problems.

KEY WORDS: automatic; load stepping; load incrementation; algorithm; error control; elastoplastic

INTRODUCTION

When the displacement finite element method is used to analyse the behaviour of non-linear materials, the total load is broken up into a number of increments and each of these is applied in sequence. The size of these increments, which is not necessarily uniform, typically has a major influence on the accuracy of the solution that is obtained for a given mesh. Plasticity theory, for example, assumes infinitesimal steps and analysis with finite load increments inevitably causes some load path error in the resulting displacements and stresses. Ignoring rounding effects, this error typically decreases as the increment size is decreased.

Techniques for solving the global equations associated with non-linear finite element analysis can be broadly classified as either iterative or incremental. Iterative schemes treat the governing relations as a system of non-linear equations and attempt to solve them by applying the unbalanced forces, computing the corresponding displacement increments, and iterating until the drift from equilibrium is small. Newton–Raphson, modified Newton–Raphson, and initial stress

* Author to whom correspondence should be addressed

CCC 0029–5981/96/101737–23

Received 22 August 1994
Revised 23 May 1995

methods are all iterative techniques. One major disadvantage of the Newton–Raphson family of algorithms is that the iterations may not converge, particularly when the behaviour is strongly non-linear. This may force various stabilizing measures to be used, such as line searches or arc length control, and the algorithms can rapidly become very complex in an effort to maintain robustness. A second, equally serious, disadvantage is that even though equilibrium may be satisfied, there is no estimate of the load path error. Although an iterative scheme may converge to an equilibrium state, there is no guarantee that this is sufficiently close to the true equilibrium state (unless a large number of load increments is used). The need for small load increments is especially important for plasticity problems, where significant load path errors can be caused by the use of load steps which are too coarse. Unfortunately, the size of the load increments required for an accurate solution varies throughout the loading range and, not surprisingly, is highly problem dependent. In general, accurate solutions with large load increments can only be obtained for cases where the strain path is only mildly non-linear, since it is necessary to assume that the strain rate is constant throughout the iteration process.

Incremental schemes treat the governing relations as a system of ordinary differential equations and, in their usual form, involve no iteration since the solution is generated using a series of piece-wise linear steps which approximate the non-linear load–deformation response of the system. Provided the global stiffness matrix remains well conditioned, these techniques have proved to be very robust and are especially useful for highly non-linear problems involving complex constitutive behaviour. With a sufficient number of load increments, they seldom fail to furnish a solution of acceptable accuracy. One commonly perceived problem with incremental schemes is that they tend to 'drift' from equilibrium as the solution proceeds. This effect can be minimized by calculating the residual (or unbalanced) forces at the end of each load increment and adding these to the applied loads for the next increment. This simple correction requires a relatively small amount of computation time, yet ensures that equilibrium is approximately satisfied at all times.

The method described in this paper is essentially an incremental scheme with automatic step size control. The integration process selects each step so that the local truncation error in the computed deflections is below a prescribed value, and also includes an unbalanced force correction to prevent accumulation of global error. A key feature of the scheme is that by automatically controlling the load path, the load path error in the resulting final displacements can be constrained to lie near a user-specified tolerance. The scheme is particularly robust and permits a broad class of load–deformation paths to be integrated with only a small amount of drift from equilibrium. Since the method does not exploit any special features of the governing equations, it can be used to deal with a wide range of constitutive models. Moreover, complicated loading paths, such as those associated with unloading and excavation sequences, do not present any special problems. The motivation for the scheme comes from the successful application by Sloan[1] of a similar idea to the automatic integration of complex constitutive laws.

Somewhat surprisingly, the design of robust and efficient load incrementation strategies for non-linear finite element analysis has not received wide attention in the literature. Because of the inherent complexity of the problem, and the tendency to focus on iterative solution schemes, most published techniques are heuristic in nature and often require some intervention by the user. Some of the more successful algorithms that have been developed use a variety of parameters including the curvature of the non-linear path,[2] the 'current stiffness parameter',[3,4] and the number of iterations required to restore equilibrium.[5] More recently Schellekens *et al.*[6] proposed a method based on strain energy. In contrast to the primary aim of the proposed algorithm, which attempts to control the load path error in the solution directly, the main purpose of these schemes is to ensure convergence of various iterative procedures.

SOLUTION PROCEDURES

The system of differential equations to be solved for each load increment can be expressed in rate form as

$$\dot{\mathbf{u}} = [\mathbf{K}(\mathbf{u})]^{-1}\dot{\mathbf{f}} = [\mathbf{K}(\mathbf{u})]^{-1}\,\Delta\mathbf{f}/\Delta t \tag{1}$$

where $\dot{\mathbf{u}}$ is a vector of unknown displacement rates, $\mathbf{K}$ is the tangent stiffness matrix, $\dot{\mathbf{f}}$ is a vector of externally applied force rates (which are assumed constant over some time interval Δt), and the superior dot denotes a derivative with respect to time. For rate independent problems it is convenient to introduce a 'dimensionless time', T, such that

$$T = (t - t_0)/\Delta t \tag{2}$$

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

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}T} = [\mathbf{K}(\mathbf{u})]^{-1}\,\Delta\mathbf{f} \tag{3}$$

Equation (3) has the form of a classical initial value problem since $\Delta\mathbf{f}$ 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 $t = t_0$ and $T = 0$. The traditional and crudest method for solving such a system of differential equations is the first-order forward Euler scheme. This explicit method calculates the displacements at the end of the load increment, where $t = t_0 + \Delta t$ and $T = 1$, using the relationship

$$\mathbf{u} = \mathbf{u}_0 + [\mathbf{K}(\mathbf{u}_0)]^{-1}\,\Delta\mathbf{f}$$

Although the forward Euler scheme provides a simple means for solving the governing load–deflection equations, it is only accurate for small load steps. The numerical performance of the simple Euler scheme can be improved greatly by computing the unbalanced forces at the end of each increment and adding these to the applied loads for the next load increment according to

$$\mathbf{u} = \mathbf{u}_0 + [\mathbf{K}(\mathbf{u}_0)]^{-1}\{\Delta\mathbf{f} + \mathbf{f}^{\mathrm{unb}}(\mathbf{u}_0)\} \tag{4}$$

This minimizes the tendency of the solution to drift from equilibrium as the integration proceeds. The accuracy of the Euler solution can, of course, be further improved by dividing the applied load increment into N subincrements of equal size. Equation (4) is then replaced by the recurrence relation

$$\mathbf{u}_i = \mathbf{u}_{i-1} + [\mathbf{K}(\mathbf{u}_{i-1})]^{-1}\{\Delta\mathbf{f}_i + \mathbf{f}^{\mathrm{unb}}(\mathbf{u}_{i-1})\}$$

where $\Delta\mathbf{f}_i = \Delta\mathbf{f}\,\Delta T_i$ is the subincremental force vector, $\Delta T_i = 1/N$ is the dimensionless time subincrement, and $i = 1, 2, \ldots, N$. The initial conditions at the start of the load increment are $\mathbf{u} = \mathbf{u}_0$ whilst at the end of the load increment the displacements are given by $\mathbf{u} = \mathbf{u}_N$. For this type of scheme to be efficient, it is necessary to be able to estimate the number of load subincrements that are required to produce a solution of specified accuracy. One such estimate, first proposed by Wissmann and Hauck[7] for the integration of the stress–strain laws, can be obtained by using one full-size load increment and then two half-size subincrements. The difference between these two solutions for the displacements provides a measure of the local truncation error which, in turn, can be used to predict the number of load subincrements required to achieve a specified accuracy. Any of the well known schemes for integrating systems of ordinary differential equations can be used with this type of step size control, including the Euler method.

Modified Euler scheme with adaptive load subincrementation

Adaptive integration schemes, which automatically adjust the step size to keep the local truncation error near a specified tolerance, are standard methods in numerical analysis for the solution of initial value problems. Embedded integration formulas, such as those described by Fehlberg[8], provide a particularly efficient means of step size control since they require few function evaluations to estimate the local truncation error. The key idea of these techniques is to use two integration schemes, whose order of accuracy differs by one, to predict the solution at the end of each step. The difference between the highest-order solution and the lowest-order solution provides an estimate of the local truncation error for the current step size. If this error is less than a specified level, the solution is accepted and the next step size is predicted by using local extrapolation of the dominant error term. Otherwise, the solution is rejected and the stage is repeated with a smaller step whose size is again computed from local extrapolation. In this way, the step size may increase or decrease, in accordance with the local non-linearity, as the integration proceeds. A variety of adaptive stress subincrementation schemes for integrating complex constitutive laws have been proposed by Sloan[1] and Sloan and Booker.[9] These methods have proved to be most effective in practice, and can be readily modified to integrate global finite element equations.

Neglecting, for the moment, the effects of unbalanced forces, each stage of the proposed scheme computes two estimates of the displacements. These are based on the Euler and modified Euler formulas and may be written, respectively, as

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta\mathbf{u}_1 \tag{5}$$

$$\hat{\mathbf{u}}_i = \mathbf{u}_{i-1} + \tfrac{1}{2}(\Delta\mathbf{u}_1 + \Delta\mathbf{u}_2) \tag{6}$$

where

$$\Delta\mathbf{u}_1 = [\mathbf{K}(\mathbf{u}_{i-1})]^{-1}\,\Delta\mathbf{f}_i \tag{7}$$

$$\Delta\mathbf{u}_2 = [\mathbf{K}(\mathbf{u}_{i-1} + \Delta\mathbf{u}_1)]^{-1}\,\Delta\mathbf{f}_i \tag{8}$$

and $\Delta\mathbf{f}_i = \Delta\mathbf{f}\,\Delta T_i$ is the subincremental force vector with $0 < \Delta T_i \leqslant 1$. Since the local truncation errors in $\mathbf{u}_i$ and $\hat{\mathbf{u}}_i$ are, respectively, $O(\Delta T_i^2)$ and $O(\Delta T_i^3)$, the error in $\mathbf{u}_i$ can be estimated by subtracting the lower-order solution from the higher-order solution to give

$$\|\mathbf{E}_i\| \approx \tfrac{1}{2}\,\|(\Delta\mathbf{u}_2 - \Delta\mathbf{u}_1)\| \tag{9}$$

where any convenient norm may be used. This quantity, which predicts the absolute truncation error, can be divided by $\|\hat{\mathbf{u}}_i\|$ to furnish a more useful dimensionless relative error measure

$$R_i = \|\mathbf{E}_i\|/\|\hat{\mathbf{u}}_i\| \tag{10}$$

The current load subincrement is accepted if R_i is less than some specified tolerance, DTOL, and rejected otherwise. In either case, the size of the next dimensionless time step ΔT_{i+1} is found from

$$\Delta T_{i+1} = q\,\Delta T_i \tag{11}$$

where q is a factor which is chosen to limit the predicted truncation error. As the local truncation error in $\mathbf{u}_i$ is $O(\Delta T_i^2)$, it follows from (11) that the truncation error for the next load subincrement, R_{i+1}, is related to the truncation error for the current load subincrement, R_i, according to

$$R_{i+1} \approx q^2 R_i$$

The required factor q is found by insisting that $R_{i+1} \leqslant \text{DTOL}$, so that

$$q \leqslant \sqrt{\frac{\text{DTOL}}{R_i}}$$

Because local extrapolation may become inaccurate for strongly non-linear behaviour, it is wise to choose q conservatively to minimize the number of rejected load subincrements. Numerical experiments on a wide variety of plasticity problems suggest that a suitable strategy for computing q is to set

$$q = 0{\cdot}7 \sqrt{\frac{\text{DTOL}}{R_i}} \tag{12}$$

and also constrain it to lie within the limits

$$0{\cdot}1 \leqslant q \leqslant 1{\cdot}1 \tag{13}$$

The coefficient of 0·7 acts merely as a safety factor, since it usually prevents the step control mechanism from choosing load subincrements which just fail to meet the local error tolerance. Restricting the growth of consecutive load subincrements to 10 per cent also has this effect. Numerical experiments indicate that increasing the subincrement reduction coefficient to 0·9, as well as increasing the maximum growth factor for consecutive subincrements to 400 per cent, has little influence on the performance of the algorithm. Relaxing these constraints leads to larger subincrement sizes and hence fewer load subincrements overall, but this saving is counteracted by the increased number of failed load subincrements. A final control, of lesser importance than the above refinements, is to prohibit the step size from growing immediately after a failed load subincrement. This ensures that there are at least two load subincrements of the same size following a failure, and is useful for cases where the load path has sharp changes in curvature.

The integration scheme is started by applying (5) and (6) with the known incremental force vector $\Delta\mathbf{f}$, the initial displacements $\mathbf{u}_0$, and an initial guess for ΔT_1. For the first load increment ΔT_1 is typically set to unity, but in subsequent load increments ΔT_1 may be initialized to the value of ΔT that gave the last successful subincrement. If the relative error in the resulting displacements, as defined by equation (10), is less than or equal to the specified tolerance DTOL, then the current load subincrement is accepted and the displacements are updated using (5). The step size for the next load subincrement is then found using (12) and (13). This may increase, decrease, or stay the same, depending on the error that is calculated from equation (10). If the specified error tolerance is not met, so that $R_i > \text{DTOL}$, then the solution is rejected and a smaller step size is computed using equations (12) and (13). This stage is then repeated and, if necessary, the step size is reduced further until a successful load subincrement size is obtained. The end of the integration procedure is reached when the entire increment of load is applied so that

$$\sum_{i=1}^{N} \Delta T_i = 1$$

A naïve analysis of the proposed algorithm would suggest that each successful load subincrement requires two formations/factorizations of the stiffness matrix and two equation solutions. With a minor change to the computation sequence, however, only one stiffness matrix formation/factorization and one equation solution are needed, thus improving the efficiency of the scheme substantially. The key point to note is that the displacements $\Delta\mathbf{u}_2$, which are calculated for error control, can be multiplied by the subincrement size factor q to provide the first-order

displacements for the next load subincrement. Thus, after a successful load subincrement, the first-order displacements for the next load subincrement are given by

$$\Delta \mathbf{u}_1 = q\Delta \mathbf{u}_2 \tag{14}$$

This feature is an important advantage of using the Euler-modified Euler pair for error control. Compared with a simple forward Euler scheme using the same load path, the adaptive method described above requires only one additional formation/factorization of the stiffness matrix and one extra equation solution for each coarse load increment $\Delta \mathbf{f}$. This extra work is minor for cases where the number of subincrements in each coarse load increment is significant.

Correcting for drift from equilibrium

As mentioned previously, solutions from incremental schemes tend to drift from equilibrium as the integration proceeds and a load imbalance develops between the externally applied forces and the forces supported by the internal stresses. One option for minimizing this effect is to augment the externally applied force vector for the current subincrement with the unbalanced forces at the end of the previous subincrement, equations (7) and (8) are then replaced by

$$\Delta \mathbf{u}_1 = [\mathbf{K}(\mathbf{u}_{i-1})]^{-1}\{\Delta \mathbf{f}_i + \mathbf{f}^{\text{unb}}(\mathbf{u}_{i-1})\} \tag{15}$$

$$\Delta \mathbf{u}_2 = [\mathbf{K}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_1)]^{-1}\{\Delta \mathbf{f}_i + \mathbf{f}^{\text{unb}}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_1)\} \tag{16}$$

while the rest of the algorithm remains unchanged. Although simple and seemingly efficient, equations (15) and (16) do not lead to a scheme with good step size control. This is because the contribution of the unbalanced forces to the deflections is independent of ΔT_i. Even if ΔT_i is shrunk to zero, so that $\Delta \mathbf{f}_i = \Delta \mathbf{f} \Delta T_i = 0$, the error computed from (10) may still exceed the specified tolerance due to the effect of the unbalanced forces. For strongly non-linear behaviour, this may result in the algorithm adopting tiny load subincrements and not being able to advance the solution.

One strategy for avoiding this problem is to control the subincrement size by using only the local truncation error due to the applied loads, as before, and add the effect of the unbalanced forces separately. The subincremental displacements are then divided into two parts, one part being due to the applied load and the other part being due to the unbalanced force. Expanding equations (15) and (16), the Euler and modified Euler updates may be rewritten as

$$\begin{aligned} \mathbf{u}_i &= \mathbf{u}_{i-1} + \Delta \mathbf{u}_1 + \Delta \mathbf{u}_1^{\text{unb}} \\ \hat{\mathbf{u}}_i &= \mathbf{u}_{i-1} + \tfrac{1}{2}(\Delta \mathbf{u}_1 + \Delta \mathbf{u}_2) + \tfrac{1}{2}(\Delta \mathbf{u}_1^{\text{unb}} + \Delta \mathbf{u}_2^{\text{unb}}) \end{aligned} \tag{17}$$

where

$$\begin{aligned} \Delta \mathbf{u}_1 &= [\mathbf{K}(\mathbf{u}_{i-1})]^{-1}\{\Delta \mathbf{f}_i\} \\ \Delta \mathbf{u}_1^{\text{unb}} &= [\mathbf{K}(\mathbf{u}_{i-1})]^{-1}\{\mathbf{f}^{\text{unb}}(\mathbf{u}_{i-1})\} \\ \Delta \mathbf{u}_2 &= [\mathbf{K}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_1 + \Delta \mathbf{u}_1^{\text{unb}})]^{-1}\{\Delta \mathbf{f}_i\} \\ \Delta \mathbf{u}_2^{\text{unb}} &= [\mathbf{K}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_1 + \Delta \mathbf{u}_1^{\text{unb}})]^{-1}\{\mathbf{f}^{\text{unb}}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_1 + \Delta \mathbf{u}_1^{\text{unb}})\} \end{aligned}$$

For reasons just discussed, the unbalanced force displacement contributions must be neglected when computing the local truncation error. Thus equation (9) is still used to estimate $\mathbf{E}_i$ but the displacements are now updated using (17) after a successful load subincrement. The other change is that (10) is no longer used to estimate the relative truncation error. Instead, this quantity is

computed from

$$R_i = \|\mathbf{E}_i\|/\|\mathbf{u}_i\|$$

Since the above modifications make it unnecessary to compute the quantities $\hat{\mathbf{u}}_i$ or $\Delta\mathbf{u}_2^{unb}$, the revised scheme nominally requires two formations/factorizations of the stiffness matrix and three equation solutions for each successful load subincrement. As described in the previous section, however, one of these formations/factorizations and equation solutions can be removed easily by exploiting the common evaluation points of the Euler and modified Euler formulas. The savings occur immediately after a successful load subincrement, where the first-order displacements for the next subincrement can again be found from (14). When compared with a simple forward Euler scheme using the same load subincrement sizes, the adaptive method with an unbalanced force correction thus requires one additional equation solution per subincrement. One additional formation/factorization of the stiffness matrix and one extra equation solution is also required for each coarse load increment.

Prescribed force loadings

Due to the stiffness matrix becoming singular, difficulties may arise when collapse studies are performed with prescribed force loading. No such problem, of course, occurs for prescribed displacement analyses. Since the current scheme makes no attempt to integrate post-peak behaviour with prescribed force loadings, a reliable method for terminating the solution process gracefully is required. One simple approach for detecting imminent collapse under these conditions is to monitor the relative stiffness of the system. The current stiffness parameter of Bergan *et al.*[3] is often used for this purpose, but it is best suited to proportional loading and needs to be modified for the use of prescribed displacements.

An alternative measure of the relative stiffness can be found by first defining a scalar which is a least squares fit to the incremental stiffness equations. Such a scalar, denoted by K_i, minimizes the quantity

$$(K_i\Delta\mathbf{u}_i - \Delta\mathbf{f}_i)^T(K_i\Delta\mathbf{u}_i - \Delta\mathbf{f}_i)$$

where $\Delta\mathbf{f}_i$ and $\Delta\mathbf{u}_i$ are, respectively, the incremental forces and displacements for the ith step. Expanding this dot product and minimizing with respect to K_i gives

$$K_i = \Delta\mathbf{f}_i^T\Delta\mathbf{u}_i/\Delta\mathbf{u}_i^T\Delta\mathbf{u}_i$$

At the end of the ith step, the relative stiffness can now be estimated as

$$K = \frac{K_i}{K_0} = \frac{\Delta\mathbf{f}_i^T\Delta\mathbf{u}_i}{\Delta\mathbf{u}_i^T\Delta\mathbf{u}_i} \times \frac{\Delta\mathbf{u}_0^T\Delta\mathbf{u}_0}{\Delta\mathbf{f}_0^T\Delta\mathbf{u}_0} \tag{18}$$

where $\Delta\mathbf{f}_0$ and $\Delta\mathbf{u}_0$ are the incremental forces and displacements for the first step. The incremental stiffness parameter defined by equation (18) is applicable at both the incremental and subincremental level, and can be used to monitor the solution procedure. Once K falls below a small threshold value under force prescribed loading, then the stiffness matrix is nearly singular and the analysis should be terminated.

IMPLEMENTATION OF INTEGRATION ALGORITHM WITH ERROR CONTROL

The implementation of the automatic integration scheme is now described. The algorithm requires the user to specify a series of coarse load increments that define $\Delta\mathbf{f}$, and then attempts to

subincrement these so that the relative load path error in the computed deflections is near a user-specified tolerance DTOL.

The load integration algorithm may be implemented as follows:

1. Enter with current stresses $\boldsymbol{\sigma}$, current displacements $\mathbf{u}$, external force increment $\Delta\mathbf{f}$, unbalanced force vector for current displacements $\mathbf{f}^{unb}$, previous subincrement size ΔT_{last}, and displacement error tolerance DTOL.
2. Set $T = 0$ and $\Delta T = \min\{\Delta T_{last}, 1\}$
3. Compute $\Delta\mathbf{u}_1$ according to
$$\Delta\mathbf{u}_1 = [\mathbf{K}(\mathbf{u})]^{-1}\,\Delta T\{\Delta\mathbf{f}\}$$
4. While $T < 1$ do steps 5–13.
5. Compute $\Delta\mathbf{u}_1^{unb}$ according to
$$\Delta\mathbf{u}_1^{unb} = [\mathbf{K}(\mathbf{u})]^{-1}\{\mathbf{f}^{unb}\}$$
6. Calculate first-order displacement update as
$$\mathbf{u}_1 = \mathbf{u} + \Delta\mathbf{u}_1 + \Delta\mathbf{u}_1^{unb}$$
and then integrate constitutive law to find corresponding stress state $\boldsymbol{\sigma}_1$.
7. Compute $\Delta\mathbf{u}_2$ according to
$$\Delta\mathbf{u}_2 = \Delta T[\mathbf{K}(\mathbf{u}_1)]^{-1}\{\Delta\mathbf{f}\}$$
8. Estimate local truncation error for current load subincrement using
$$\mathbf{E} = \tfrac{1}{2}(\Delta\mathbf{u}_2 - \Delta\mathbf{u}_1)$$
9. Compute relative error for current load subincrement using
$$R = \max\left\{EPS, \frac{\|\mathbf{E}\|_{max}}{\|\mathbf{u}_1\|_{max}}\right\}$$
where EPS is a machine constant.
10. If $R \leqslant$ DTOL go to step 11. Else current load subincrement has failed, so estimate a smaller dimensionless time step using
$$q = \max\left\{0{\cdot}7\sqrt{\frac{\text{DTOL}}{R}}, 0{\cdot}1\right\}$$
and set
$$\Delta T \leftarrow q\,\Delta T$$
Scale $\Delta\mathbf{u}_1$ according to
$$\Delta\mathbf{u}_1 \leftarrow q\,\Delta\mathbf{u}_1$$
and return to step 6 to repeat the subincrement.
11. Current load subincrement is successful. Update dimensionless time, displacements, and stresses according to
$$T \leftarrow T + \Delta T$$
$$\mathbf{u} = \mathbf{u}_1$$
$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_1$$
Then find unbalanced forces $\mathbf{f}^{unb}$ corresponding to displacements $\mathbf{u}_1$.

12. Compute least squares estimate of incremental stiffness from

$$K_i = \Delta \mathbf{f}^{\mathrm{T}} \Delta \mathbf{u}_1 / \Delta \mathbf{u}_1^{\mathrm{T}} \Delta \mathbf{u}_1$$

If this is the first subincrement of the first coarse load increment, set $K_0 = K_i$. Calculate incremental stiffness parameter from

$$K = K_i / K_0$$

If $|K| \leqslant$ KTOL and loading is by prescribed forces, exit to step 14 and abandon subsequent coarse load increments.

13. Save size of last subincrement

$$\Delta T_{\text{last}} = \Delta T$$

Compute first estimate of step size growth factor using

$$q = \min \left\{ 0{\cdot}7 \sqrt{\frac{\text{DTOL}}{R}}, 1{\cdot}1, \frac{1 - T}{\Delta T} \right\}$$

If previous load subincrement was unsuccessful, do not allow subincrement size to grow by enforcing

$$q \leftarrow \min \{q, 1\}$$

Compute step size and first-order displacement prediction for next load subincrement

$$\Delta T \leftarrow q \Delta T$$

$$\Delta \mathbf{u}_1 = q \Delta \mathbf{u}_2$$

14. Exit with displacements $\mathbf{u}$ and stresses $\boldsymbol{\sigma}$ at end of coarse load increment.

The variable ΔT_{last} stores the size of the second last subincrement that was used successfully in the previous coarse load step. This is used as a trial value at the start of each new coarse load step in order to minimize the number of rejected subincrements. It is necessary to store the second last subincrement size, rather than the last subincrement size, since the latter may be severely truncated to avoid over-shooting the end of the integration interval.

In step 9, EPS represents the smallest relative error that can be computed on the host machine. For double precision arithmetic on a 32-bit architecture, EPS is typically set to around 10^{-16}. The value of KTOL, which is used in step 12 to detect singularity in the stiffness matrix for force prescribed loading, may be set anywhere in the range 10^{-3}–10^{-6}. Numerical experiments suggest that a suitable value for detecting imminent collapse in plasticity problems is 10^{-4}.

Finally, in step 13, care must be taken that the integration does not proceed beyond $T = 1$. This is implemented by insisting that the step growth factor, q, is less than or equal to $(1 - T)/\Delta T$, where ΔT and T are, respectively, the subincrement size and dimensionless time for the last successful substep.

APPLICATIONS

In this section, the automatic load stepping algorithm is used to analyse the behaviour of four elastoplastic boundary value problems. Detailed results are presented for the expansion of a thick cylinder, the collapse of a rigid strip footing, the collapse of a flexible strip footing and the stability of a trapdoor. Each of these boundary value problems is modelled using fully integrated cubic

strain triangles, which are known to give reliable and accurate results for plasticity problems.[10] With the exception of the trapdoor problem, which assumes a purely cohesive Tresca model, all of the results are for cohesive-frictional constitutive behaviour and employ a rounded Mohr–Coulomb yield surface. The rounding procedure, described in Sloan and Booker[11] and Abbo and Sloan,[12] adopts a hyperbolic approximation in the meridional plane and a trigonometric approximation in the octahedral plane and requires only two parameters to fit the exact Mohr–Coulomb criterion as closely as desired. This approach removes the need to develop special constitutive integration procedures in the vicinity of the tip and corners, and is simple to implement. In the case of the trapdoor problem, the corners of the Tresca criterion are also rounded using an identical process.

At the stress point level, the elastoplastic constitutive laws are integrated using an explicit scheme with error control as described in Abbo and Sloan.[13] This scheme is a refined version of the algorithm originally published by Sloan,[1] and uses the Euler-modified Euler pair with adaptive substepping to achieve a user specified accuracy in the stresses. For the results presented in this paper, the constitutive laws are integrated very accurately by using a relative error tolerance of 10^{-6} for the stresses in conjunction with an absolute tolerance of 10^{-9} for drift from the yield surface.

As mentioned previously, the proposed integration algorithm requires the loading to be defined as a series of coarse load increments. For all of the problems, except the flexible footing, the coarse load steps are applied as prescribed displacement increments of equal size, such that the total imposed displacement induces a state of collapse in the soil mass. In the case of the flexible footing, collapse is induced by applying the coarse load steps as prescribed force increments of equal size.

To assess the performance of the automatic integration algorithm, the global load path error in the final displacements $\mathbf{u}$ is estimated using

$$u_{\text{error}} = \|\mathbf{u}_{\text{ref}} - \mathbf{u}\|_{\max} / \|\mathbf{u}_{\text{ref}}\|_{\max} \tag{19}$$

where $\mathbf{u}_{\text{ref}}$ are a set of reference displacements. The reference displacements, which have very small load path errors, are found using a first-order Euler algorithm with an equilibrium correction and a very large number of equal-size load increments. For the automatic scheme, the error calculated from (19) may be compared with the displacement error tolerance DTOL to assess its performance. As a further measure of the accuracy of the various analyses, the unbalanced forces are computed to check the equilibrium between the forces supported by the internal stresses and the forces applied externally. Any drift from equilibrium is found using

$$f_{\text{error}} = \|\mathbf{f}^{\text{unb}}\|_{\max} / \|\mathbf{f}\|_{\max} \tag{20}$$

where all values are computed at the end of each analysis. Equation (20) provides a useful check on the accuracy of the reference solutions, which give $f_{\text{error}} \leqslant 10^{-9}$ for all cases.

In the results that follow, various timing statistics are given to indicate the efficiency of the proposed load incrementation scheme. All of these are for a HP710 workstation with the HP FORTRAN 77 compiler and level 3 optimization.

Thick cylinder

Since it is essentially one dimensional in nature, the expansion of a thick cylinder of cohesive-frictional material provides a simple test case for the automatic integration algorithm. The mesh and material properties used to model this problem are shown in Figure 1. The cylinder is loaded to failure by uniform prescribed displacements at its inner radius, where an equivalent uniform

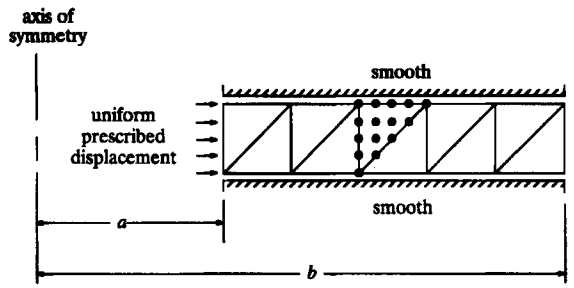

Figure 1. Expansion of thick cylinder of cohesive-frictional material

Table I. Results for thick cylinder using Euler scheme

No. load increments	CPU time (s)	Collapse load (p/c)	Displacement error (u_{error})	Equilibrium error (f_{error})
10	2	1·0277	$1{\cdot}9\times10^{-3}$	$0{\cdot}36\times10^{-2}$
100	9	1·0175	$2{\cdot}0\times10^{-5}$	$0{\cdot}14\times10^{-4}$
1000	84	1·0174	$5{\cdot}3\times10^{-9}$	$0{\cdot}68\times10^{-12}$
100 000	8276	1·0174	—	$0{\cdot}28\times10^{-12}$

pressure, p, is computed by summing the appropriate nodal reactions. The analytic solution to this problem, which is useful for checking displacement finite element codes, can be found in Yu[14] and predicts that collapse occurs when $p/c = 1{\cdot}1074$. In order to obtain a set of reference displacements which contain very small load path errors, a forward Euler analysis with an equilibrium correction and 100 000 load increments is used. These displacements, and the corresponding stresses, are used in equations (19) and (20) to estimate the displacement error, u_{error}, and the equilibrium error, f_{error}, at the end of each load path.

To gauge the performance of a conventional integration scheme, the thick cylinder is analysed using the forward Euler method with equilibrium correction and various numbers of equal size load increments. The CPU times, collapse loads, displacement errors and equilibriums errors for runs with 10, 100, 1000 and 100 000 load steps are presented in Table I. As expected, the displacement load path errors for the Euler scheme decrease as the number of load increments is increased. Due to the mildly non-linear behaviour of this problem, only 10 increments are required to achieve a load path error of roughly 10^{-3} or better in the final displacements. For a load path error of around 10^{-5}, about one hundred increments are necessary. It is interesting to note that, with the simple equilibrium correction, the global error of the Euler scheme is at least a quadratic function of the increment size. Since the global error of the traditional Euler scheme is a linear function of the step size, the merit of incorporating this correction is clear. In a similar manner to the displacement error, the equilibrium error also decreases as the number of increments is increased. With 100 000 load increments, the Euler analysis predicts the exact collapse load of $p/c = 1{\cdot}0174$ and gives an equilibrium error of less than 10^{-12}.

To assess the performance of the automatic integration algorithm, it is used to analyse the thick cylinder with various values of the displacement error tolerance, DTOL, ranging from 10^{-1} to

Table II. Results for thick cylinder using automatic scheme

Each tolerance DTOL	No. coarse load steps	No. load subincrements		CPU time (s)	Collapse load (p/c)	Displacement error (u_{error})	Equilibrium error (f_{error})
		Successful	Failed				
10^{-1}	10	10	0	2·3	1·0277	$0{\cdot}2\times10^{-2}$	$0{\cdot}36\times10^{-1}$
	5	5	0	1·7	1·0596	$0{\cdot}7\times10^{-2}$	$0{\cdot}10\times10^{0}$
10^{-2}	10	10	0	2·3	1·0277	$0{\cdot}2\times10^{-2}$	$0{\cdot}36\times10^{-1}$
	5	5	0	1·7	1·0596	$0{\cdot}7\times10^{-2}$	$0{\cdot}10\times10^{0}$
10^{-3}	10	19	1	7	1·0195	$0{\cdot}4\times10^{-3}$	$0{\cdot}16\times10^{-1}$
	5	18	2	6	1·0185	$0{\cdot}2\times10^{-3}$	$0{\cdot}17\times10^{-1}$
10^{-4}	10	55	1	28	1·0176	$0{\cdot}5\times10^{-4}$	$0{\cdot}40\times10^{-2}$
	5	56	2	29	1·0175	$0{\cdot}2\times10^{-4}$	$0{\cdot}14\times10^{-2}$

10^{-4}. Each run with a fixed tolerance is performed twice, once with five coarse load steps and once with ten coarse load steps, to test the sensitivity of the results to this parameter. The selected load subincrements, CPU times, collapse loads, displacement errors and equilibrium errors are recorded in Table II for each analysis. When five or ten coarse load steps are used, the scheme chooses not to subincrement for values of DTOL equal to 10^{-1} and 10^{-2}. The computed load path errors in the displacements confirm that load subincrementation is indeed unnecessary for these tolerances, since the maximum error is only $0{\cdot}7\times10^{-2}$ for the five increment analysis. As the error tolerance is tightened to 10^{-3} or 10^{-4}, load subincrementation is required and the performance of the automatic scheme can be assessed by comparing the computed displacement errors with DTOL. Inspection of the results shown in Table II indicate that the resulting displacement errors are all within an order of magnitude of the specified tolerance. Analysis with ten coarse load steps, for example, gives a displacement error of $0{\cdot}4\times10^{-3}$ when an error tolerance of 10^{-3} is specified, while for a stricter tolerance of 10^{-4}, a displacement error of $0{\cdot}5\times10^{-4}$ results. As expected, the equilibrium errors also decrease as the error tolerance is tightened but, in general, these are two orders of magnitude larger than the displacement errors. It is pleasing to note that, for all of the runs with the same error tolerance, the results computed using five coarse load increments are very similar to those computed using ten coarse load increments. Indeed, the errors for these two sets of runs are of the same order of magnitude in each case, since the algorithm always chooses a similar pattern of load subincrements. For typical error tolerances in the range 10^{-2}–10^{-3}, the CPU time requirement of the automatic scheme is very modest, and would certainly be competitive with that for a fast iterative solution scheme.

The displacement load path errors cited in Tables I and II are at the end of the loading range, where collapse has taken place, and give no indication of how the error varies during the analysis. This question is addressed in Figure 2, which shows the displacement error variation versus load level for a run with 10 coarse load steps and DTOL = 10^{-3}. In this analysis, the displacement error is calculated at the end of each coarse load increment. For most of the loading range, it can be seen that the automatic integration algorithm maintains an error of approximately $0{\cdot}5\times10^{-3}$, which is equal to half the specified displacement tolerance. Some oscillation in this type of plot is to be expected due to the effect of truncated subincrements, which occur at the end of each coarse load step, and the complexity of the differential equations being integrated. To give an indication of the performance of a more conventional scheme, Figure 2 also shows the displacement errors for the Euler algorithm with ten and one hundred equal-size load increments. These results

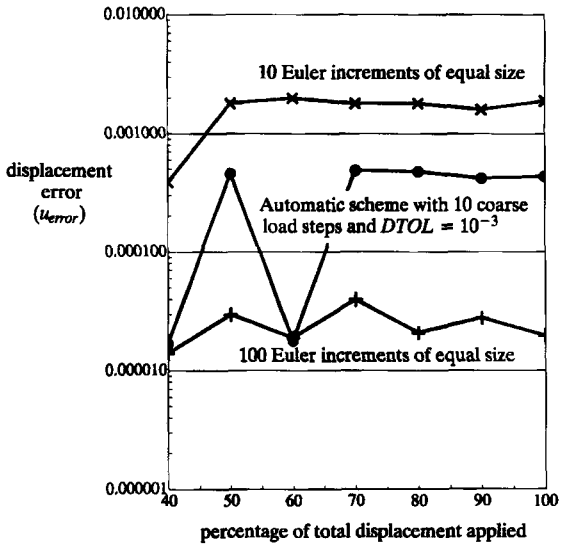

Figure 2. Variation of displacement load path error with load level for thick cylinder

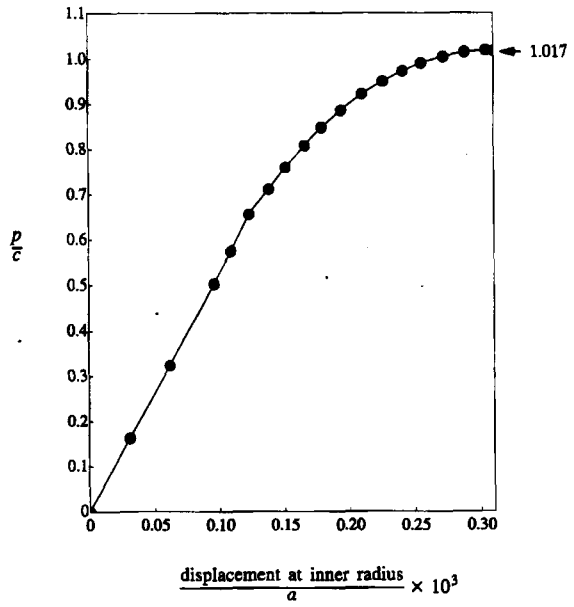

Figure 3. Automatic load subincrement selection for thick cylinder with single coarse load step and DTOL = 10^{-3}

suggest that the use of fixed-size increments is a good strategy when using the Euler method to analyse the thick cylinder, since this keeps the load path error relatively constant.

The performance of the automatic algorithm is investigated further by applying all of the load to the cylinder in a single coarse load step. The resulting load–displacement curve, obtained with a load path error tolerance of 10^{-3}, is shown in Figure 3. Although an extreme test, the automatic

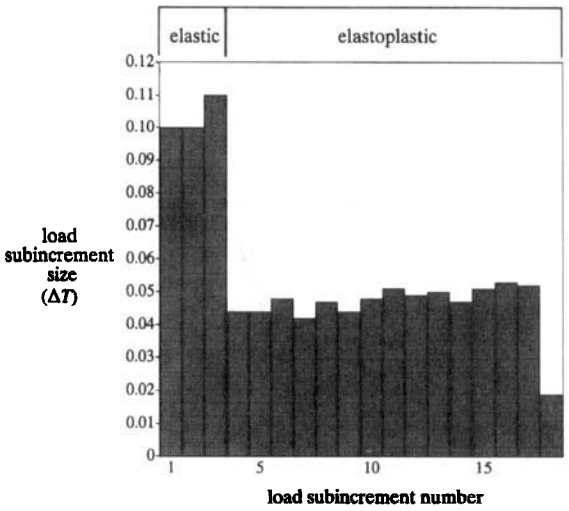

Figure 4. Automatic load subincrement selection for thick cylinder with single coarse load step and DTOL = 10^{-3}

scheme successfully reduces the initial increment size and then adjusts this throughout the integration process to suit the non-linearity of the behaviour. A more detailed plot of the load subincrement sizes, shown in Figure 4, reveals that the scheme chooses three large subincrements in the elastic range and then fifteen, roughly uniform, subincrements in the elastoplastic range. The fact that scheme selects load steps of roughly equal size in the non-linear range confirms the previous observation that this is the optimum strategy for the Euler method when analysing the thick cylinder problem.

Rigid strip footing

This section considers the problem of a smooth rapid strip footing resting on a weightless cohesive-frictional soil. Due to the singularity at the edge of the footing and the strong rotation of the principal stresses, this case is a severe test for non-linear solution schemes. The finite element mesh and soil parameters used in the analysis are shown in Figure 5. Vertical load is applied to the footing by a set of uniform prescribed displacements and an equivalent pressure is again computed by summing the appropriate nodal reactions. A total of 48 cubic strain triangles is used in the grid, and these are concentrated under the edge of the footing in an effort to model the singularity. The exact collapse load, derived by Prandtl, is given as $p/c = 30{\cdot}1396$. As with the thick cylinder problem, the reference displacements are calculated using the Euler scheme with an equilibrium correction and 100 000 load increments of equal size.

The results shown in Table III summarize the performance of the Euler method when it is used with an equilibrium correction and various numbers of equal-size increments. Approximately one hundred steps are required to achieve a load path error of 10^{-2} or better in the final displacements. For a load path error of 10^{-5}, more than one thousand increments are necessary. Due to the influence of the simple equilibrium correction, the global load path error of the Euler scheme appears to be a quadratic function of the increment size. The equilibrium error for this problem shows a strong correlation with the displacement error measure, and is negligible for an analysis with one thousand load increments or more. With only ten load increments the error in the

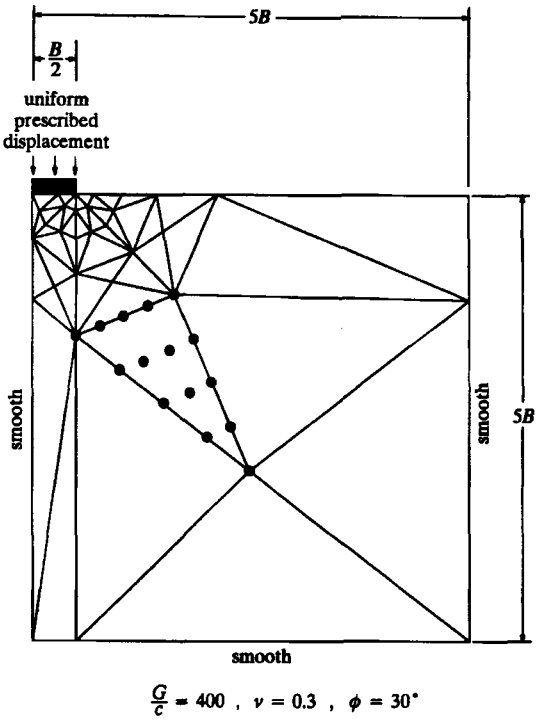

Figure 5. Smooth rigid strip footing on cohesive-frictional soil

Table III. Results for smooth rigid strip footing using Euler scheme

No. load increments	CPU time (s)	Collapse load (p/c)	Displacement error (u_{error})	Equilibrium error (f_{error})
10	49	32·4508	$5{\cdot}5 \times 10^{-1}$	$0{\cdot}31 \times 10^{0}$
100	96	30·7530	$6{\cdot}4 \times 10^{-3}$	$0{\cdot}80 \times 10^{-3}$
1000	472	30·7522	$3{\cdot}2 \times 10^{-5}$	$0{\cdot}57 \times 10^{-5}$
100000	45835	30·7522	—	$0{\cdot}42 \times 10^{-9}$

computed collapse load is roughly 8 per cent. This is reduced to approximately 2 per cent when the analysis is performed with one thousand load increments.

The results for the footing analysis with the automatic integration algorithm are presented in Table IV. Runs are performed with DTOL ranging from 10^{-1} to 10^{-4} and the loading is applied in five and ten coarse steps for each tolerance. In all cases, the displacement load path error is well controlled by the automatic scheme and is of the same order of magnitude as the specified error tolerance. For example, with DTOL set to 10^{-1}, the runs with five and ten coarse steps give, respectively, observed load path errors of $2{\cdot}7 \times 10^{-1}$ and $5{\cdot}5 \times 10^{-1}$. With a tighter tolerance of 10^{-4}, the corresponding errors are $1{\cdot}6 \times 10^{-4}$ and $1{\cdot}5 \times 10^{-4}$. As in the thick cylinder example, the load path subincrementation performed by the automatic algorithm is largely independent of the number of coarse load steps used in each analysis. For example, with an error tolerance of 10^{-4}, the algorithm creates 744 subincrements from five coarse load steps and 731 subincrements

Table IV. Results for smooth rigid strip footing using automatic scheme

Error tolerance DTOL	No. coarse load steps	No. load subincrements: Successful	No. load subincrements: Failed	CPU time (s)	Collapse load (p/c)	Displacement error (u_{error})	Equilibrium error (f_{error})
10^{-1}	10	10	0	49	32·4509	$5{\cdot}5\times10^{-1}$	$0{\cdot}31\times10^{0}$
	5	16	1	59	31·2404	$2{\cdot}7\times10^{-1}$	$0{\cdot}26\times10^{0}$
10^{-2}	10	49	2	85	30·7724	$3{\cdot}2\times10^{-2}$	$0{\cdot}41\times10^{-2}$
	5	44	3	78	30·7724	$6{\cdot}2\times10^{-2}$	$0{\cdot}61\times10^{-2}$
10^{-3}	10	172	7	126	30·7525	$2{\cdot}0\times10^{-3}$	$0{\cdot}43\times10^{-3}$
	5	167	8	127	30·7525	$2{\cdot}1\times10^{-3}$	$0{\cdot}41\times10^{-3}$
10^{-4}	10	731	48	359	30·7522	$1{\cdot}5\times10^{-4}$	$0{\cdot}11\times10^{-3}$
	5	744	50	369	30·7522	$1{\cdot}6\times10^{-4}$	$0{\cdot}13\times10^{-3}$

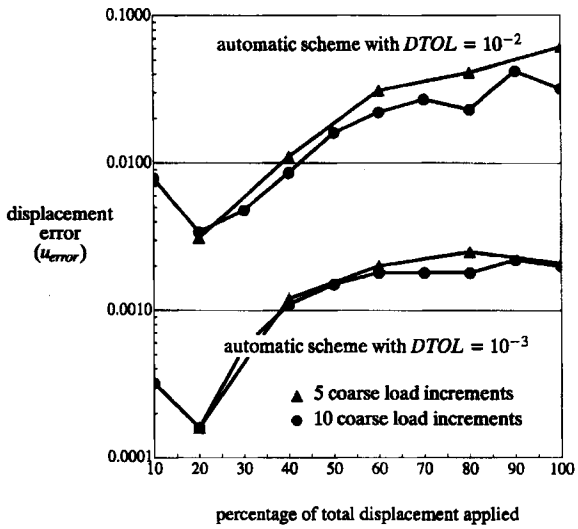

Figure 6. Variation of displacement load path error with load level for rigid strip footing

from ten coarse load steps. When ten coarse steps are used with DTOL = 10^{-1} and DTOL = 10^{-4}, the error in the respective collapse load ranges from 8 to 2 per cent. The timing data in Table IV indicates that, with roughly 750 load subincrements of variable size, the automatic scheme takes approximately 0·50 s per subincrement. This compares favourably with the performance of the Euler scheme which, from the statistics shown in Table III, takes approximately 0·47 s per load step for a thousand steps of fixed size.

The variation of the load path error with load level for the strip footing analysis is shown in Figure 6. Results are presented for analyses using five and ten coarse load steps with displacement tolerances of 10^{-2} and 10^{-3}. In each case the displacement errors are seen to be within an order of magnitude of the specified error tolerances. This plot also indicates that, for each tolerance, the load path errors are similar regardless of the number of coarse load increments used in the analysis.

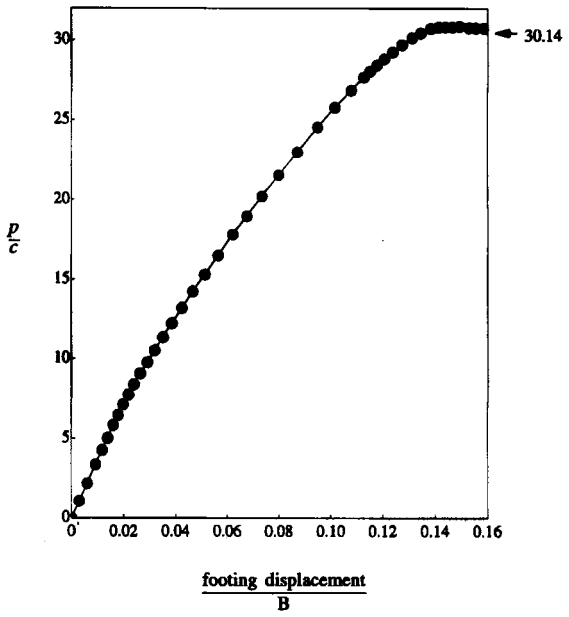

Figure 7. Automatic load subincrement selection for rigid footing with single coarse load step and DTOL = 10^{-2}

Figure 7 shows the load–displacement curve obtained from the automatic scheme with a single coarse load step and an error tolerance of 10^{-2}. This case highlights the adaptive nature of the integration algorithm, which automatically chooses small load subincrements in regions of highly non-linear behaviour. The slight kink in the load–deformation curve, just after yielding commences, is detected by the scheme and small load subincrements are used. The substeps then grow over the following portion of the curve which is nearly linear. As a state of collapse is approached, the subincrement size is once again reduced to enable accurate integration of the governing load–displacement equations. A more detailed picture of the variation of the load subincrement sizes is presented in Figure 8.

Flexible strip footing

To demonstrate the ability of the automatic algorithm to analyse prescribed force loading up to the point of collapse, the behaviour of a smooth flexible strip footing resulting on a cohesive-frictional soil is studied. The mesh, soil properties and theoretical collapse load for this example are identical to those for the rigid footing case, the only difference is that the footing is now loaded by nodal forces rather than by nodal displacements. This problem is difficult because the stiffness matrix becomes progressively ill-conditioned as collapse is approached. In order to terminate the automatic scheme gracefully, the incremental stiffness parameter of equation (18) is used to detect singularity with a threshold value of 10^{-4}.

The only results presented for this case are the load–displacement curve of Figure 9 and the plot of successful load subincrement sizes shown in Figure 10. As before, these are obtained from analysis with a single coarse load increment and a load path error tolerance of 10^{-2}. Inspection of Figure 9 indicates that the automatic scheme identifies the point of incipient collapse both correctly and sharply. The load subincrement sizes, shown in Figure 10, clearly reflect the non-linearity of the load-displacement curve.

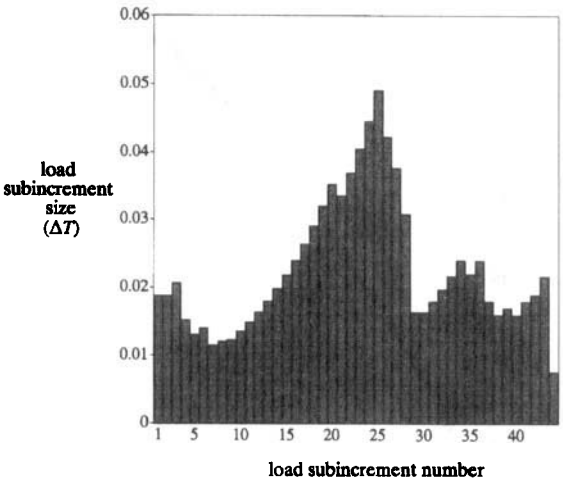

Figure 8. Automatic load subincrement selection for rigid footing with single coarse load step and DTOL = 10^{-2}

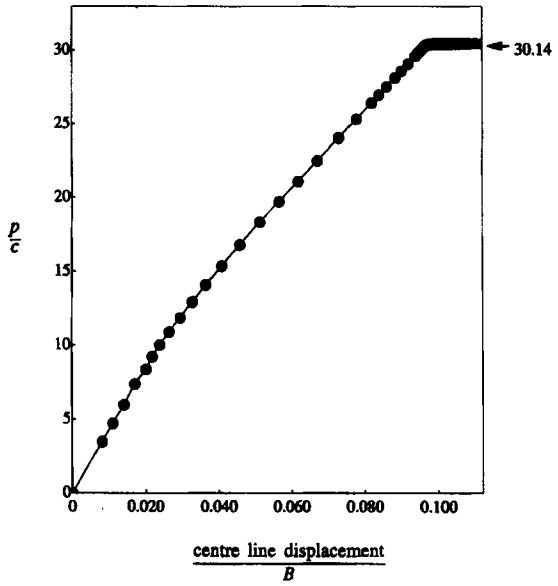

Figure 9. Automatic load subincrement selection for flexible footing with single coarse load step and DTOL = 10^{-2}

Rough trapdoor

The undrained stability of a trapdoor provides another good test of the automatic integration algorithm since the collapse mechanism is dominated by shear failure. The mesh and soil properties for the problem are shown in Figure 11. To avoid the development of a displacement discontinuity at the trapdoor edge, the element side immediately adjacent to the trapdoor is subject to a linear variation of imposed displacement. This displacement matches the trapdoor displacement at one end of the side and decreases to zero at the other to satisfy the boundary

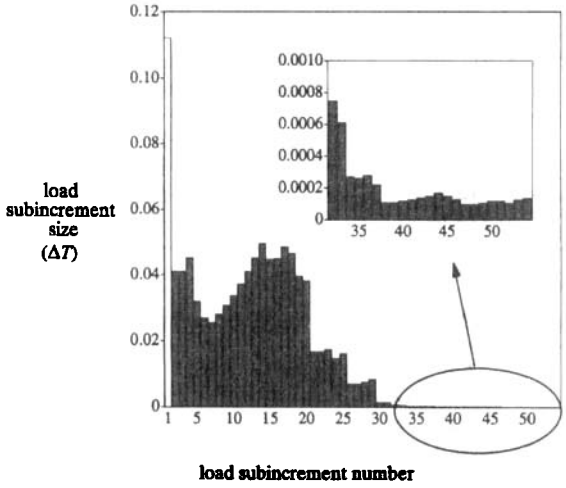

Figure 10. Automatic load subincrement selection for flexible footing with single coarse load step and DTOL = 10^{-2}

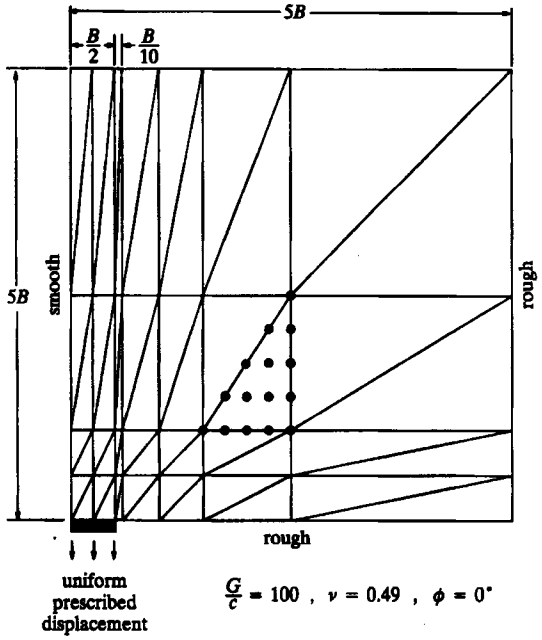

Figure 11. Rough rigid trapdoor in purely cohesive soil

condition. Somewhat surprisingly, the exact collapse load for a trapdoor in purely cohesive soil is still unknown, although rigorous upper and lower bounds have been derived by Sloan *et al.*[15] For the trapdoor analysed in this paper, these upper and lower bounds are respectively $p/c = 6{\cdot}34$ and $p/c = 5{\cdot}77$, where p is an equivalent pressure. To obtain a set of reference displacements for this problem, the Euler method with an equilibrium correction is used with 50 000 load increments.

Results for the corrected Euler method, with various numbers of fixed size increments, are shown in Table V. With this scheme, only 10 increments are required to achieve a load path error

Table V. Results for rough trapdoor using Euler scheme

No. load increments	CPU time (s)	Collapse load (p/c)	Displacement error (u_{error})	Equilibrium error (f_{error})
10	38	5·9455	$2{\cdot}0 \times 10^{-2}$	$0{\cdot}25 \times 10^{-1}$
100	160	5·9318	$1{\cdot}3 \times 10^{-4}$	$0{\cdot}52 \times 10^{-3}$
1000	1406	5·9316	$9{\cdot}6 \times 10^{-7}$	$0{\cdot}90 \times 10^{-6}$
50 000	72 826	5·9316	—	$0{\cdot}35 \times 10^{-9}$

Table VI. Results for rough trapdoor using automatic scheme

Error tolerance DTOL	No. coarse load steps	No. load subincrements		CPU time (s)	Collapse load (p/c)	Displacement error (u_{error})	Equilibrium error (f_{error})
		Successful	Failed				
10^{-1}	10	10	0	48	5·9456	$0{\cdot}2 \times 10^{-1}$	$0{\cdot}25 \times 10^{-1}$
	5	5	0	37	5·9348	$0{\cdot}6 \times 10^{-1}$	$0{\cdot}10 \times 10^{0}$
	1	1	0	5	7·6139	$1{\cdot}0 \times 10^{-1}$	$0{\cdot}33 \times 10^{0}$
10^{-2}	10	57	4	119	5·9344	$0{\cdot}06 \times 10^{-2}$	$0{\cdot}44 \times 10^{-2}$
	5	53	4	106	5·9360	$0{\cdot}1 \times 10^{-2}$	$0{\cdot}82 \times 10^{-2}$
	1	51	6	101	5·9373	$0{\cdot}1 \times 10^{-2}$	$0{\cdot}57 \times 10^{-2}$
10^{-3}	10	133	4	224	5·9330	$0{\cdot}3 \times 10^{-3}$	$0{\cdot}37 \times 10^{-2}$
	5	128	5	208	5·9349	$0{\cdot}6 \times 10^{-3}$	$0{\cdot}63 \times 10^{-2}$
	1	130	6	208	5·9342	$0{\cdot}7 \times 10^{-3}$	$0{\cdot}42 \times 10^{-2}$
10^{-4}	10	637	29	950	5·9319	$0{\cdot}3 \times 10^{-4}$	$0{\cdot}52 \times 10^{-3}$
	5	650	34	955	5·9319	$0{\cdot}3 \times 10^{-4}$	$0{\cdot}11 \times 10^{-2}$
	1	633	31	925	5·9319	$0{\cdot}3 \times 10^{-4}$	$0{\cdot}30 \times 10^{-2}$

of 10^{-2} or better in the final displacements. For a load path error of 10^{-4}, around one hundred steps are necessary. Using 50 000 load increments, the Euler analysis predicts a collapse load of $p/c = 5{\cdot}9316$ which falls between the bounds of Sloan *et al.*[15] Since this analysis gives an equilibrium error of less than 10^{-9}, there would appear to be very little load path error in the reference displacements. The influence of the equilibrium correction on the performance of the Euler scheme is once again clearly apparent, as the global load path error decreases quadratically with decreasing increment size.

The results for analysis of the trapdoor using the automatic integration algorithm are shown in Table VI. Data for runs with error tolerances ranging from 10^{-1} to 10^{-4} are presented, with each tolerance being analysed with one, five and ten coarse load steps. In all of these analyses, the observed load path errors are considerably less than the specified error tolerance, which indicates that the automatic scheme is rather conservative in its step size control for this problem. With ten coarse load steps, for example, a displacement error of $0{\cdot}2 \times 10^{-1}$ is obtained using an error tolerance of 10^{-1}. When the latter is tightened to a value of 10^{-4}, the observed displacement error is $0{\cdot}6 \times 10^{-4}$. As in the previous examples, the subincremental strategy chosen by the algorithm is largely independent of the initial number of coarse load steps. For coarse load steps of one, five and ten, the algorithm uses a minimum of 633 and a maximum of 650 load subincrements to integrate to an error tolerance of 10^{-4}. The collapse loads for the trapdoor range from $p/c = 7{\cdot}6139$ to $p/c = 5{\cdot}9319$, where the former value is computed using a single coarse load step

with a tolerance of 10^{-1} and the latter value is found from all of the analyses with a tolerance of 10^{-4}. The CPU times for the automatic scheme are again competitive for typical tolerances of 10^{-2} to 10^{-3}.

Figure 12 shows the load–displacement curve for a run performed with only a single coarse load step and an error tolerance of 10^{-2}. A more detailed profile of the load subincrement sizes for this case, shown in Figure 13, clearly indicates that a small step size is required at the beginning of the analysis while a larger step size is permissible as the trapdoor collapses. This variation of load increments is not intuitive and is unlikely to be chosen by even the most

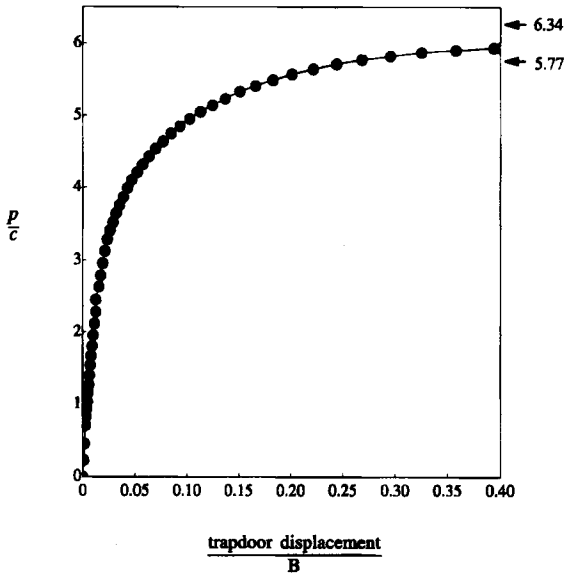

Figure 12. Automatic load subincrement selection for rigid trapdoor with single coarse load step and DTOL = 10^{-2}

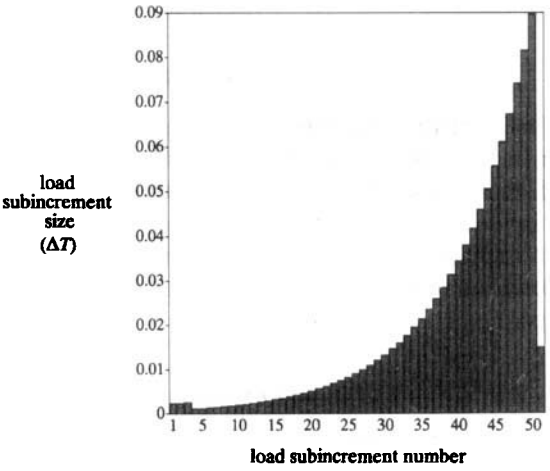

Figure 13. Automatic load subincrement selection for rigid trapdoor with single coarse load step and DTOL = 10^{-2}

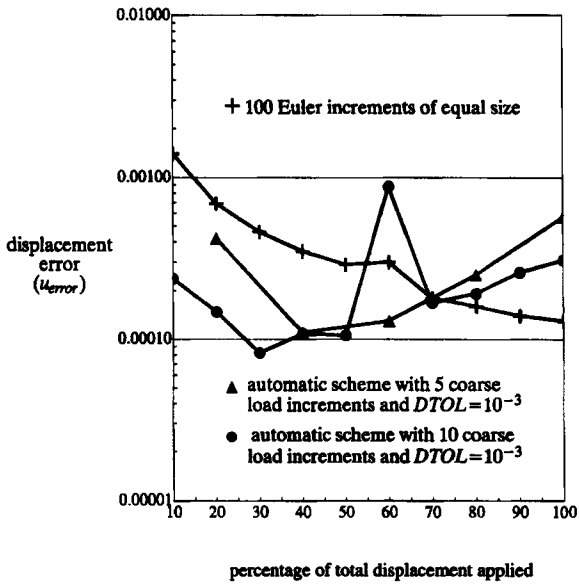

Figure 14. Variation of displacement load path error with load level for rigid trapdoor

experienced analyst. Figure 13 also suggests that the step size control during the analysis is restricted by the rule which limits the growth of consecutive load subincrement sizes to 10 per cent.

Figure 14 shows the variation of the load path error with load level for various analyses of the trapdoor. The results for the automatic scheme, using five and ten coarse load increments with an error tolerance of 10^{-3}, suggest that the load path error is always kept below the desired threshold. The plot for the Euler scheme, obtained using an equilibrium correction and one hundred fixed size steps, indicates that the use of equal size increments leads to decreasing load path error with increasing load level. This data supports the strategy of the automatic scheme, which tries to keep the load path error constant by increasing the subincrement size as the load is increased.

CONCLUSIONS

The automatic integration algorithm presented in this paper is an attractive method for controlling the load path error in non-linear finite element analysis. By subincrementing a number of user-defined coarse load steps, the scheme is able to integrate the governing stiffness equations so that the load path error in the final displacements lies near a specified tolerance. The algorithm is fast, robust and simple to implement. It successfully controls the load path error independently of the number of coarse load increments supplied by the user, and may even be employed with a single coarse load step.

REFERENCES

1. S. W. Sloan, 'Substepping schemes for the numerical integration of elastoplastic stress-strain relations', *Int. j. numer. methods eng.*, **24**, 893–911 (1987).
2. C. Den Heijer and W. C. Rheinboldt, 'On step length algorithms for a class of continuation methods', *SIAM J. Numer. Anal.*, **18**, 925–948 (1981).

3. P. G. Bergan, G. Horrigmoe, B. Krakeland and T. H. Soreide, 'Solution techniques for nonlinear finite element problems', *Int. j. numer. methods eng.*, **12**, 1677–1696 (1978).
4. P. G. Bergan and T. Soreide, 'Solution of large displacement and instability problems using the current stiffness parameter' in P. G. Bergan *et al.* (eds.), *Finite Elements in Nonlinear Mechanics*, Tapir Press, Trondheim, 1978, pp. 647–669.
5. M. A. Crisfield, 'A fast incremental/iterative solution procedure that handles 'snap-through', *Comp. Struct.*, **13**, 55–62 (1981).
6. J. C. J. Schellekens, P. H. Feenstra, and de Borst, 'A self-adaptive load estimator based on strain energy', *Proc. 3rd Int. Conf. Computational Plasticity*, Vol. 1, Barcelona, 1992, pp. 187–198.
7. J. W. Wissmann and C. Hauck, 'Efficient elastic-plastic finite element analysis with higher order stress point algorithms', *Comp. Struct.*, **17**, 89–95 (1983).
8. E. Fehlberg, 'Klassische Runge–Kutta Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungs-probleme', *Computing*, **6**, 61–71 (1970).
9. S. W. Sloan and J. R. Booker, 'Integration of Tresca and Mohr-Coulomb constitutive relations in plane strain elastoplasticity', *Int. j. numer. methods eng.*, **33**, 163–196 (1992).
10. S. W. Sloan and M. F. Randolph, 'Numerical prediction of collapse loads using finite element methods', *Int. j. numer. anal. methods geomech.*, **6**, 47–76 (1982).
11. S. W. Sloan and J. R. Booker, 'Removal of singularities in Tresca and Mohr–Coulomb yield functions', *Commun. appl. numer. methods* **2**, 173–179 (1986).
12. A. J. Abbo and S. W. Sloan, 'A smooth hyperbolic approximation to the Mohr–Coulomb yield criterion', *Comp. Struct.*, **54**, 427–441 (1993).
13. A. J. Abbo and S. W. Sloan, 'A comparison of integration schemes for elastoplastic constitutive laws', *Research Report* 091.02.1994, Department of Civil Engineering and Surveying, University of Newcastle, Australia, 1994.
14. H. S. Yu, 'Expansion of a thick cylinder of soil', *Comp. Geotechnics*, **14**, 21–41, (1992).
15. S. W. Sloan, A. Assadi and N. Purushothaman, 'Undrained stability of a trapdoor', *Geotechnique*, **40**, 45–62 (1990).