Pergamon

0045-7949(94)00339-4

Computers & Structures Vol. 54, No. 3, pp. 427–441, 1995

0045-7949/95 \$9.50 + 0.00

A SMOOTH HYPERBOLIC APPROXIMATION TO THE MOHR–COULOMB YIELD CRITERION

A. J. Abbo and **S. W. Sloan**

Department of Civil Engineering and Surveying, University of Newcastle, NSW 2308, Australia

(*Received* 22 *August* 1993)

Abstract—The Mohr–Coulomb yield criterion is used widely in elastoplastic geotechnical analysis. There are computational difficulties with this model, however, due to the gradient discontinuities which occur at both the edges and the tip of the hexagonal yield surface pyramid. It is well known that these singularities often cause stress integration schemes to perform inefficiently or fail. This paper describes a simple hyperbolic yield surface that eliminates the singular tip from the Mohr–Coulomb surface. The hyperbolic surface can be generalized to a family of Mohr–Coulomb yield criteria which are also rounded in the octahedral plane, thus eliminating the singularities that occur at the edge intersections as well. This type of yield surface is both continuous and differentiable at all stress states, and can be made to approximate the Mohr–Coulomb yield function as closely as required by adjusting two parameters. The yield surface and its gradients are presented in a form which is suitable for finite element programming with either explicit or implicit stress integration schemes. Two efficient FORTRAN 77 subroutines are given to illustrate how the new yield surface can be implemented in practice.

INTRODUCTION

The Mohr–Coulomb yield criterion, while superseded by more complex soil models for advanced applications, is still employed extensively in geotechnical analysis. Important advantages of the Mohr–Coulomb model include its simplicity and the fact that it permits finite element solutions to be compared with a wide variety of classical plasticity solutions. The latter feature is especially useful in validating finite element codes.

In terms of the principal stresses ($\sigma_1 \geqslant \sigma_2 \geqslant \sigma_3$), with tensile stresses taken as positive, the Mohr–Coulomb yield criterion can be expressed as

$$F = (\sigma_1 - \sigma_3) + (\sigma_1 + \sigma_3)\sin\phi - 2c\cos\phi = 0, \tag{1}$$

where c and ϕ represent the cohesion and friction angle of the soil.

To avoid calculating the principal stresses explicitly, which may become complicated for axisymmetric and three-dimensional deformation, isotropic yield functions are often expressed in terms of stress invariants. One particularly convenient set of stress invariants, proposed by Nayak and Zienkiewicz [1], is

$$\sigma_m = \tfrac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$$

$$\bar\sigma = \sqrt{\tfrac{1}{2}(s_x^2 + s_y^2 + s_z^2) + \tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2}$$

$$\theta = \tfrac{1}{3}\sin^{-1}\left(-\frac{3\sqrt{3}}{2}\frac{J_3}{\bar\sigma^3}\right), \quad -30^\circ \leqslant \theta \leqslant 30^\circ,$$

where

$$J_3 = s_x s_y s_z + 2\tau_{xy}\tau_{yz}\tau_{zx} - s_x\tau_{yz}^2 - s_y\tau_{xz}^2 - s_z\tau_{xy}^2$$

and

$$s_x = \sigma_x - \sigma_m, \quad s_y = \sigma_y - \sigma_m, \quad s_z = \sigma_z - \sigma_m.$$

In terms of these invariants, the principal stresses are

$$\sigma_1 = \frac{2}{\sqrt{3}}\bar\sigma\sin(\theta + 120^\circ) + \sigma_m \tag{2}$$

$$\sigma_2 = \frac{2}{\sqrt{3}}\bar\sigma\sin(\theta) + \sigma_m$$

$$\sigma_3 = \frac{2}{\sqrt{3}}\bar\sigma\sin(\theta - 120^\circ) + \sigma_m. \tag{3}$$

Substituting eqns (2) and (3) in (1), the Mohr–Coulomb yield function can be expressed as

$$F = \sigma_m\sin\phi + \bar\sigma K(\theta) - c\cos\phi = 0 \tag{4}$$

with

$$K(\theta) = \cos\theta - \frac{1}{\sqrt{3}}\sin\phi\sin\theta. \tag{5}$$

The relationship between $\bar\sigma$ and σ_m, for a constant θ, defines a meridional section of the yield surface. For the Mohr–Coulomb criterion, this relationship

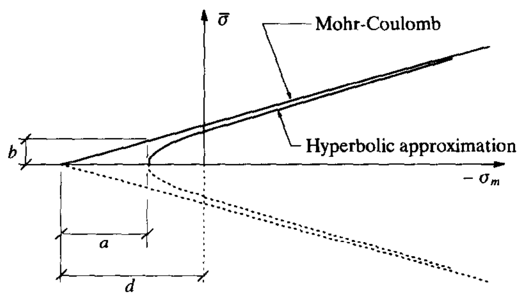

Fig. 1. Hyperbolic approximation to Mohr–Coulomb yield function.

can be represented as a straight line in $(\sigma_m, \bar{\sigma})$ space as shown in Fig. 1. The point where the line cuts the σ_m-axis corresponds to the tip of the hexagonal Mohr–Coulomb pyramid, and it is here that the gradient of the yield surface is undefined.

Removal of the apex singularity can be accomplished by modelling the Mohr–Coulomb yield surface with a continuous and differentiable surface. A comprehensive discussion of various smooth approximations to the Mohr–Coulomb criterion has been given by Zienkiewicz and Pande [2]. These authors noted that a hyperbolic approximation, as shown in Fig. 1, can be made to model the original surface as closely as desired by adjusting a single parameter. Another attractive feature of the hyperbolic surface is that it asymptotes rapidly to the Mohr–Coulomb yield surface as the compressive hydrostatic stress increases. Because the hyperbolic surface is an internal approximation, the corresponding soil strength is always less than the strength that would be found from a Mohr–Coulomb model with the same cohesion and friction angle. It should be noted that the removal of the apex singularity can result in a significant computational saving for problems which involve tensile hydrostatic stress states. These types of problems often arise in the analysis of soils with significant friction angles and low cohesions [2], and may cause stress integration schemes to become unstable.

In the octahedral plane, defined by $\sigma_m = \text{constant}$, the shape of the yield function is defined by the relationship between $\bar{\sigma}$ and θ. When viewed in this plane, the Mohr–Coulomb surface has sharp vertices (and hence gradient discontinuities) at $\theta = \pm 30^\circ$ where each of the sides meet. It is necessary to address these singularities because stress states lying at, or near, the vertices are often encountered in finite element analysis. Various techniques for eliminating these corners have been discussed by Zienkiewicz and Pande [2], Owen and Hinton [3] and Sloan and Booker [4]. Of these techniques, the Sloan and Booker rounding has the advantage that it models the Mohr–Coulomb yield surface very closely, since it uses a trigonometric approximation only in the region of the vertices. Relative to the strength predicted by the Mohr–Coulomb criterion, this approximation also ensures that the strength is modelled conservatively. Except for tensile hydrostatic stress states, the resulting yield surface is continuous and differentiable, and can be made to model the Mohr–Coulomb yield surface closely by adjusting a single parameter.

This paper describes a rounded hyperbolic yield surface that eliminates all singularities from the Mohr–Coulomb yield criterion. The new surface uses a hyperbolic approximation in the meridional plane, in conjunction with the trigonometric rounding techniques of Sloan and Booker [4] in the octahedral plane. It is both continuous and differentiable at all stress states, and can be fitted to the Mohr–Coulomb yield surface by adjusting two-parameters.

Two FORTRAN 77 subroutines are presented to illustrate how the rounded hyperbolic surface may be implemented efficiently in a finite element code. The subroutines, YIELD and GRAD, return the value of the yield function and the gradient vector, respectively, and are applicable to two-dimensional plasticity with associated or non-associated flow. They are also able to calculate values for the usual Mohr–Coulomb and Tresca yield criteria (both of which are rounded in the octahedral plane).

HYPERBOLIC YIELD SURFACE

The equation of the straight line defining the Mohr–Coulomb yield function in the meridional plane can be determined directly from eqn (4) as

$$\bar{\sigma} = \frac{1}{K(\theta)}(c \cos \phi - \sigma_m \sin \phi).$$

The slope of this line is $-\sin \phi / K(\theta)$ and it intercepts the σ_m-axis at $\sigma_m = c \cot \phi$. Following Zienkiewicz and Pande [2], a close approximation to the straight line which defines the Mohr–Coulomb yield surface can be obtained using an asymptotic hyperbola. The general equation of such a hyperbola, in $(\sigma_m, \bar{\sigma})$ space, is

$$\frac{(\sigma_m - d)^2}{a^2} - \frac{\bar{\sigma}^2}{b^2} = 1, \tag{6}$$

where a, b and d are the distances defined in Fig. 1. The upper asymptote to this hyperbola has slope $-b/a$ and crosses the σ_m-axis at $\sigma_m = d$. Equating the slope and intercept of the Mohr–Coulomb surface to the slope and intercept of the hyperbolic surface asymptote yields the two relations

$$\frac{b}{a} = \frac{\sin \phi}{K(\theta)}, \quad d = c \cot \phi.$$

On substitution into eqn (6), the required yield surface can be derived as

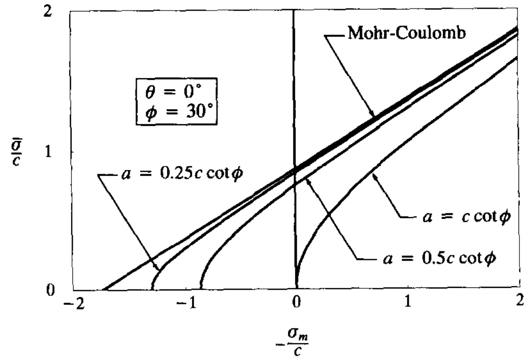

Fig. 2. Hyperbolic approximation to Mohr–Coulomb meridional section.

$$F = \sigma_m + \sqrt{\bar{\sigma}^2 K^2(\theta) + a^2 \sin^2\phi} - c\cos\phi = 0, \tag{7}$$

where the negative branch of the hyperbola has been chosen. This function can be made to model the Mohr–Coulomb yield function as closely as desired by adjusting the parameter a. Moreover, the Mohr–Coulomb yield function is recovered if a is set to zero. Various meridional sections of the hyperbolic yield surface are plotted in Fig. 2. For $a \leqslant 0.25c\cot\phi$, the hyperbolic surface closely represents the Mohr–Coulomb surface. In practice, setting $a = 0.05c\cot\phi$ has been found to give results which are almost identical to those from the Mohr–Coulomb model.

ROUNDED HYPERBOLIC YIELD SURFACE

With a suitable choice for $K(\theta)$, the hyperbolic surface of eqn (7) can be generalized to form a family of useful yield functions which do not possess singularities in the octahedral plane. For the purposes of this paper, $K(\theta)$ will be selected so that the octahedral cross-section is similar to the Mohr–Coulomb cross-section, except that it is rounded.

A suitable choice for $K(\theta)$, which rounds the vertices of the Mohr–Coulomb surface in the octahedral plane using a simple trigonometric approximation, has been described by Sloan and Booker [4]. Away from the singular vertices, which occur at $\theta = \pm 30°$ in the octahedral plane, Sloan and Booker's rounded yield surface is identical to the Mohr–Coulomb yield surface so that $K(\theta)$ is defined by eqn (5). In the vicinity of the singularities, where $|\theta| > \theta_T$ and θ_T is a specified transition angle, an alternative form of $K(\theta)$ is defined. Sloan and Booker's rounded Mohr–Coulomb yield surface thus retains the form of eqn (4), but redefines $K(\theta)$ as

$$K(\theta) = \begin{cases} (A - B\sin 3\theta), & |\theta| > \theta_T \\ \left(\cos\theta - \frac{1}{\sqrt{3}}\sin\phi\sin\theta\right), & |\theta| \leqslant \theta_T \end{cases} \tag{8}$$

where

$$A = \tfrac{1}{3}\cos\theta_T\left(3 + \tan\theta_T\tan 3\theta_T + \frac{1}{\sqrt{3}}\operatorname{sign}(\theta)(\tan 3\theta_T - 3\tan\theta_T)\sin\phi\right) \tag{9}$$

$$B = \frac{1}{3\cos 3\theta_T}\left(\operatorname{sign}(\theta)\sin\theta_T + \frac{1}{\sqrt{3}}\sin\phi\cos\theta_T\right)$$

$$\operatorname{sign}(\theta) = \begin{cases} +1 & \text{for } \theta \geqslant 0° \\ -1 & \text{for } \theta < 0°. \end{cases} \tag{10}$$

The value of the transition angle lies within the range $0 \leqslant \theta_T \leqslant 30°$, with larger values giving better fits to the Mohr–Coulomb cross-section in the octahedral plane. In practice, θ_T should not be too near 30° to avoid ill-conditioning of the approximation, and a typical value is 25°. Once the transition angle is specified, the coefficients A and B may be computed efficiently by evaluating all of the constant terms in eqns (9) and (10), respectively.

A hyperbolic yield function, which is rounded in both the meridional plane and the octahedral plane, can be defined by using eqn (7) with $K(\theta)$ given by (8). The resulting yield stress is continuous and differentiable for all stress states, and the Mohr–Coulomb yield surface can be modelled as closely as desired by adjusting the two parameters a and θ_T. Indeed, the Mohr–Coulomb function can be recovered by substituting $a = 0$ and $\theta_T = 30°$. A comparison between the π-plane sections of the rounded hyperbolic surface and the Mohr–Coulomb surface is illustrated in Fig. 3. For a meridional rounding parameter $a = 0.05c\cot\phi$, and an octahedral rounding parameter of $\theta_T = 25°$, the $\bar{\sigma}/c$ values predicted by the rounded hyperbolic surface differ from those of the rounded Mohr–Coulomb surface by a maximum of 0.13%. As the compressive mean normal stress increases, this difference is reduced even further by the asymptotic nature of the hyperbolic surface.

It should be noted that four different yield surfaces can be defined using the two forms of yield functions in conjunction with the two forms of $K(\theta)$. Equations (4) and (5) define the traditional Mohr–Coulomb function with discontinuities at both the tip and the

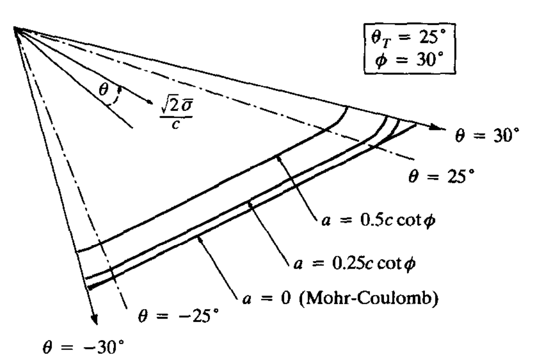

Fig. 3. Rounded hyperbolic yield surface in the π-plane.

edges of the pyramidal yield surface. Sloan and Booker's rounded Mohr–Coulomb surface, with only the tip discontinuity, is given by eqns (4) and (8). The tip singularity may be removed from both of these surfaces by replacing the yield function (4) with the hyperbolic yield function (7). Thus a hyperbolic yield function with edge discontinuities is defined using equations (7) and (5) while, as mentioned previously, a rounded hyperbolic surface with no discontinuities at all is described by eqns (7) and (8).

YIELD SURFACE GRADIENTS

The gradients of the yield surface and plastic potential play an essential role in elastoplastic finite element analysis. These quantities are used to calculate the elastoplastic stress–strain matrices and, in explicit stress integration schemes, to correct for drift from the yield surface. As the gradients are usually calculated many times in a single analysis, they need to be evaluated efficiently. Nayak and Zienkiewicz [1] proposed a convenient method for computing the gradient $\mathbf{a}$ of an isotropic function which is of the form

$$\mathbf{a} = \frac{\partial F}{\partial \boldsymbol{\sigma}} = C_1 \frac{\partial \sigma_m}{\partial \boldsymbol{\sigma}} + C_2 \frac{\partial \bar{\sigma}}{\partial \boldsymbol{\sigma}} + C_3 \frac{\partial J_3}{\partial \boldsymbol{\sigma}}, \tag{11}$$

where

$$C_1 = \frac{\partial F}{\partial \sigma_m}$$

$$C_2 = \frac{\partial F}{\partial \bar{\sigma}} - \frac{\tan 3\theta}{\bar{\sigma}} \frac{\partial F}{\partial \theta}$$

$$C_3 = -\frac{\sqrt{3}}{2 \cos 3\theta \bar{\sigma}^3} \frac{\partial F}{\partial \theta}$$

$$\frac{\partial \sigma_m}{\partial \boldsymbol{\sigma}} = \frac{1}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

$$\frac{\partial \bar{\sigma}}{\partial \boldsymbol{\sigma}} = \frac{1}{2\bar{\sigma}} \begin{Bmatrix} s_x \\ s_y \\ s_z \\ 2\tau_{xy} \\ 2\tau_{yz} \\ 2\tau_{xz} \end{Bmatrix}$$

$$\frac{\partial J_3}{\partial \boldsymbol{\sigma}} = \begin{Bmatrix} s_y s_z - \tau_{yz}^2 \\ s_x s_z - \tau_{xz}^2 \\ s_x s_y - \tau_{xy}^2 \\ 2(\tau_{yz}\tau_{xz} - s_z \tau_{xy}) \\ 2(\tau_{xz}\tau_{xy} - s_x \tau_{yz}) \\ 2(\tau_{xy}\tau_{yz} - s_y \tau_{xz}) \end{Bmatrix} + \frac{\bar{\sigma}^2}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{12}$$

and $\boldsymbol{\sigma}^T = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{xz}\}$ is the vector of stress components. This arrangement permits different yield criteria to be implemented by simply calculating the appropriate coefficients C_1, C_2 and C_3, since all of the other derivatives are independent of F. For the Mohr–Coulomb yield criterion, these constants are found by differentiating (4) to give

$$C_1^{mc} = \sin \phi, \quad C_2^{mc} = K - \tan 3\theta \frac{\mathrm{d}K}{\mathrm{d}\theta},$$

$$C_3^{mc} = -\frac{\sqrt{3}}{2 \cos 3\theta \bar{\sigma}^2} \frac{\mathrm{d}K}{\mathrm{d}\theta}, \tag{13}$$

where $K = K(\theta)$. Similarly, the coefficients for the hyperbolic yield surface are calculated by differentiating eqn (7). It is interesting to note that these can be expressed very simply in terms of the above Mohr–Coulomb coefficients as

$$C_1^h = C_1^{mc}, \quad C_2^h = \alpha C_2^{mc}, \quad C_3^h = \alpha C_3^{mc}, \tag{14}$$

where

$$\alpha = \frac{\bar{\sigma} K}{\sqrt{\bar{\sigma}^2 K^2 + a^2 \sin^2 \phi}}.$$

Gradients for the Mohr–Coulomb and hyperbolic surfaces, with unrounded octahedral cross-sections, may be evaluated using equations (13) and (14), respectively, together with eqn (5) to define $K(\theta)$. The same procedure is followed for the rounded forms of the Mohr–Coulomb and hyperbolic surfaces, except that all of the gradient coefficients are now computed using the rounded $K(\theta)$ function which is given by eqn (8). In these cases, the derivative of $K(\theta)$ is then

$$\frac{\mathrm{d}K}{\mathrm{d}\theta} = \begin{cases} -3B \cos 3\theta, & |\theta| > \theta_T \\ -\sin \theta - \frac{1}{\sqrt{3}} \sin \phi \cos \theta, & |\theta| \leqslant \theta_T. \end{cases} \tag{15}$$

GRADIENT DERIVATIVES

In many implicit stress integration methods, such as the backward Euler return algorithm discussed by Crisfield [5], it is necessary to compute the derivatives of the gradient vector with respect to the stresses.

Since implicit integration schemes are widely used in finite element codes, expressions for the gradient derivatives of the rounded hyperbolic surface are now derived. For the sake of simplicity, a two-dimensional stress vector is assumed with $\boldsymbol{\sigma}^T = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}\}$.

Differentiating eqn (11) gives

$$\frac{\partial \mathbf{a}}{\partial \boldsymbol{\sigma}} = \frac{\partial C_2}{\partial \boldsymbol{\sigma}}\frac{\partial \bar{\sigma}}{\partial \boldsymbol{\sigma}} + C_2\frac{\partial^2 \bar{\sigma}}{\partial \boldsymbol{\sigma}^2} + \frac{\partial C_3}{\partial \boldsymbol{\sigma}}\frac{\partial J_3}{\partial \boldsymbol{\sigma}} + C_3\frac{\partial^2 J_3}{\partial \boldsymbol{\sigma}^2}, \quad (16)$$

where $\partial\bar{\sigma}/\partial\boldsymbol{\sigma}$ and $\partial J_3/\partial\boldsymbol{\sigma}$ are defined by (12) and

Similarly, for the hyperbolic yield surface

$$\frac{\partial C_2^h}{\partial \boldsymbol{\sigma}} = \alpha\frac{\partial C_2^{mc}}{\partial \boldsymbol{\sigma}} + C_2^{mc}\frac{\partial \alpha}{\partial \boldsymbol{\sigma}} \quad (19)$$

$$\frac{\partial C_3^h}{\partial \boldsymbol{\sigma}} = \alpha\frac{\partial C_3^{mc}}{\partial \boldsymbol{\sigma}} + C_3^{mc}\frac{\partial \alpha}{\partial \boldsymbol{\sigma}} \quad (20)$$

with

$$\frac{\partial \alpha}{\partial \boldsymbol{\sigma}} = \frac{1-\alpha^2}{\sqrt{\bar{\sigma}^2K^2 + a^2\sin^2\theta}}\left(\frac{\partial\bar{\sigma}}{\partial\boldsymbol{\sigma}}K + \bar{\sigma}\frac{\partial K}{\partial\theta}\frac{\partial\theta}{\partial\boldsymbol{\sigma}}\right). \quad (21)$$

$$\frac{\partial^2\bar{\sigma}}{\partial\boldsymbol{\sigma}^2} = \frac{1}{\bar{\sigma}}\begin{bmatrix} \frac{1}{3} - \frac{s_x s_x}{4\bar{\sigma}^2} & & & \\ -\frac{1}{6} - \frac{s_x s_y}{4\bar{\sigma}^2} & \frac{1}{3} - \frac{s_y s_y}{4\bar{\sigma}^2} & \text{symmetric} & \\ -\frac{1}{6} - \frac{s_x s_z}{4\bar{\sigma}^2} & -\frac{1}{6} - \frac{s_y s_z}{4\bar{\sigma}^2} & \frac{1}{3} - \frac{s_z s_z}{2\bar{\sigma}^2} & \\ -\frac{\tau_{xy} s_x}{2\bar{\sigma}^2} & -\frac{\tau_{xy} s_y}{2\bar{\sigma}^2} & -\frac{\tau_{xy} s_z}{2\bar{\sigma}^2} & 1 - \frac{\tau_{xy}\tau_{xy}}{\bar{\sigma}^2} \end{bmatrix}$$

$$\frac{\partial^2 J_3}{\partial\boldsymbol{\sigma}^2} = \frac{1}{3}\begin{bmatrix} s_x - s_y - s_z & & & \\ 2s_z & s_y - s_x - s_z & \text{symmetric} & \\ 2s_y & 2s_x & s_z - s_x - s_y & \\ 2\tau_{xy} & 2\tau_{xy} & -4\tau_{xy} & -6s_z \end{bmatrix}.$$

To complete the formation of the gradient derivatives, the derivatives of C_1 and C_2 with respect to the stresses need to be evaluated for each type of yield function. For the Mohr–Coulomb criterion these derivatives are

$$\frac{\partial C_2^{mc}}{\partial\boldsymbol{\sigma}} = \frac{\partial\theta}{\partial\boldsymbol{\sigma}}\left(\frac{\partial K}{\partial\theta} + \frac{\partial^2 K}{\partial\theta^2}\tan 3\theta - 3\frac{\partial K}{\partial\theta}\sec^2 3\theta\right) \quad (17)$$

$$\frac{\partial C_3^{mc}}{\partial\boldsymbol{\sigma}} = \frac{\sqrt{3}}{2\bar{\sigma}^2\cos 3\theta}\left[\frac{\partial\theta}{\partial\boldsymbol{\sigma}}\left(\frac{\partial^2 K}{\partial\theta^2} - 3\frac{\partial K}{\partial\theta}\tan 3\theta\right) + \frac{2}{\bar{\sigma}}\frac{\partial K}{\partial\theta}\frac{\partial\bar{\sigma}}{\partial\boldsymbol{\sigma}}\right], \quad (18)$$

where

$$\frac{\partial\theta}{\partial\boldsymbol{\sigma}} = \frac{-\sqrt{3}}{2\bar{\sigma}^3\cos 3\theta}\left(\frac{\partial J_3}{\partial\boldsymbol{\sigma}} - \frac{3J_3}{\bar{\sigma}}\frac{\partial\bar{\sigma}}{\partial\boldsymbol{\sigma}}\right).$$

Gradient derivatives for rounded and unrounded octahedral cross-sections of the Mohr–Coulomb criterion are, respectively, obtained from (16) by substituting either eqn (8) or eqn (5) in (13), (17) and (18).

Thus a hyperbolic surface with a rounded octahedral cross-section is obtained from (16) by using (8) to define $K(\theta)$ in eqns (13) and (17)–(21).

IMPLEMENTATION

Two FORTRAN 77 subroutines, presented in Appendices A and B, illustrate how the hyperbolic rounded surface may be implemented efficiently in a finite element code. The subroutines, YIELD and GRAD, return the value of the yield function and the gradient vector, respectively, for a specified stress state. They are applicable to two-dimensional plasticity, with either associated or non-associated flow, and assume that the stress vector is $\boldsymbol{\sigma}^T = \{\sigma_x, \sigma_y, \tau_{xy}, \sigma_z\}$. For the case of a non-associated flow rule, the gradients are found by assuming that the plastic potential is of the same form as the yield function, with the only difference being that the dilatancy angle replaces the friction angle. As well as incorporating the rounded hyperbolic surface, the subroutines also model the usual Mohr–Coulomb and Tresca yield criteria (both of which are rounded in the octahedral plane). Since this code is executed a large number of times during the course of a typical finite element computation, considerable attention has been paid to implementing the models with a minimum amount of arithmetic.

CONCLUSION

A smooth hyperbolic approximation to the Mohr–Coulomb yield function is derived. The rounded hyperbolic surface is continuous and differentiable for all stress states, and can be fitted to the Mohr–Coulomb yield surface by adjusting two parameters. To complete the description, two FORTRAN 77 subroutines are listed to illustrate how the new surface can be implemented in an elastoplastic finite element code.

Acknowledgement—Part of the research reported in this paper was funded by the Australian Research Council. The authors are thankful for this support.

REFERENCES

1. G. C. Nayak and O. C. Zienkiewicz, Convenient form of stress invariants for plasticity. *J. Struct. Div. ASCE* **98,** 949–954 (1972).
2. O. C. Zienkiewicz and G. N. Pande, Some useful forms of isotropic yield surfaces for soil and rock mechanics. In *Finite Elements in Geomechanics*, pp. 179–190 (Edited by G. Gudehus). Wiley, Chichester (1977).
3. D. R. J. Owen and E. Hinton, *Finite Elements in Plasticity: Theory and Practice.* Pineridge Press, Swansea (1980).
4. S. W. Sloan and J. R. Booker, Removal of singularities in Tresca and Mohr–Coulomb yield functions. *Commun. Appl. Numer. Meth.* **2,** 173–179 (1986).
5. M. A. Crisfield, *Non-linear Finite Element Analysis of Solids and Structures*, Vol. 1. Wiley, Chichester, (1991).

APPENDIX A

```
      SUBROUTINE YIELD(YLD,SIGXX,SIGYY,SIGXY,SIGZZ,MPROP,NMP,IOW)
************************************************************************
*     PURPOSE:
*     --------
*     This subroutine returns the value of the yield function at the
*     given stress state for plane strain and axisymmetric plasticity.
*     Smooth approximations to the Mohr-Coulomb and Tresca criteria are
*     used.
*
*     INPUT:
*     ------
*     YLD        -  Undefined
*     SIGXX      -  XX-component of normal stress
*     SIGYY      -  YY-component of normal stress
*     SIGXY      -  XY-component of shear  stress
*     SIGZZ      -  ZZ-component of normal stress
*     MPROP      -  Vector of dimension (NMP)
*                -  Contains material parameters
*                -  MPROP(10) = (a*SIN(friction angle))**2
*                           a = hyperbolic rounding parameter
*                -  MPROP(11) = SIN(friction angle)
*                -  MPROP(12) = COS(friction angle)
*                -  MPROP(17) = Cohesion
*                -  MPROP(20) = Value defining type of yield function
*                           1 = Mohr-Coulomb rounded in octahedral plane
*                           2 = Hyperbolic Mohr-Coulomb rounded in
*                               octahedral plane
*                           3 = Tresca rounded in octahedral plane
*     NMP        -  Parameter specifying number of material parameters
*     IOW        -  Unit number of output file
*
*     OUTPUT:
*     -------
*     YLD        -  Value of yield function
*     SIGXX      -  Unchanged
*     SIGYY      -  Unchanged
*     SIGXY      -  Unchanged
*     SIGZZ      -  Unchanged
*     MPROP      -  Unchanged
*     NMP        -  Unchanged
*     IOW        -  Unchanged
*
*     SUBROUTINES CALLED:   NONE
*     -------------------
*
*     PROGRAMMER:   Andrew Abbo
*     -----------
*
*     LAST MODIFIED:  May 1993      Andrew Abbo
*     --------------
*
************************************************************************
      INTEGER YFTYPE,NMP,IOW
*
      DOUBLE PRECISION STA,CTA,A,B,K,SGN,YLD
      DOUBLE PRECISION SIGXX,SIGYY,SIGZZ,SIGXY
      DOUBLE PRECISION DSIGX,DSIGY,DSIGZ
      DOUBLE PRECISION CPHI,SPHI,COH,ASPHI2
      DOUBLE PRECISION THETA,J2,J3,S3TA,SIGM,SBAR
*
      DOUBLE PRECISION MPROP(NMP)
*
*     Set constants
*
      INTEGER MOHR,HYPER,TRESCA
      PARAMETER( MOHR = 1 )
      PARAMETER( HYPER = 2 )
      PARAMETER( TRESCA = 3 )
*
```

```
      DOUBLE PRECISION C00001,C004P5,C000P5,C00000
      PARAMETER( C00001 = 1.0D0 )
      PARAMETER( C004P5 = 4.5D0 )
      PARAMETER( C000P5 = 0.5D0 )
      PARAMETER( C00000 = 0.0D0 )
*
      DOUBLE PRECISION C000R3,CP3333,C00IR3
      PARAMETER( C000R3 = 1.732050807568877D0 )
      PARAMETER( CP3333 = 0.333333333333333D0 )
      PARAMETER( C00IR3 = 0.577350269189626D0 )
*
*     Constants for rounded K function
*
      DOUBLE PRECISION A1,A2,B1,B2,ATTRAN
*
*     Rounding constants for theta > 25 degrees
*
      PARAMETER( A1 = 1.432052062044227D0 )
      PARAMETER( A2 = 0.406941858374615D0 )
      PARAMETER( B1 = 0.544290524902313D0 )
      PARAMETER( B2 = 0.673903324498392D0 )
      PARAMETER( ATTRAN = 0.436332312998582D0 )
*
*     Calculate value of invariants
*
      SIGM = CP3333*(SIGXX+SIGYY+SIGZZ)
      DSIGX = SIGXX-SIGM
      DSIGY = SIGYY-SIGM
      DSIGZ = SIGZZ-SIGM
      J2 = C000P5*(DSIGX*DSIGX+DSIGY*DSIGY+DSIGZ*DSIGZ) + SIGXY*SIGXY
      J3 = DSIGZ*(DSIGX*DSIGY-SIGXY*SIGXY)
      SBAR = SQRT(J2)
      IF (J2.GT.C00000) THEN
*
*       Calculate third stress invariant
*
        S3TA = -C004P5*J3/(C000R3*SBAR*J2)
        IF (S3TA.LT.-C00001) THEN
          S3TA = -C00001
        ELSEIF (S3TA.GT.C00001) THEN
          S3TA = C00001
        ENDIF
        THETA = CP3333*ASIN(S3TA)
*
      ELSE
*
*       Special case of zero deviatoric stress
*
         S3TA  = C00000
         THETA = C00000
*
      ENDIF
*
*     Extract form of yield function from MPROP vector
*
      YFTYPE = INT(MPROP(20))
*
      IF ((YFTYPE.EQ.MOHR).OR.(YFTYPE.EQ.HYPER)) THEN
*-----------------------------------------------------------------------
*       Mohr-Coulomb or hyperbolic Mohr-Coulomb yield function
*       Set value of material parameters used in yield function
*
        COH    =  MPROP(17)
        SPHI   =  MPROP(11)
        CPHI   =  MPROP(12)
        ASPHI2 =  MPROP(10)
*
*       Calculate K function
*
        IF (ABS(THETA).LT.ATTRAN) THEN
*
```

```
*         Calculate K function for unrounded region of octahedral plane
*
          STA = SIN(THETA)
          CTA = COS(THETA)
          K = CTA-STA*SPHI*C00IR3
*
        ELSE
*
*         Calculate K function for rounded region of octahedral plane
*
          SGN = SIGN(C00001,THETA)
          A = A1 + A2*SGN*SPHI
          B = B1*SGN + B2*SPHI
          K = A-B*S3TA
*
        ENDIF
*
*       Calculate value of yield function
*
        IF (YFTYPE.EQ.HYPER) THEN
*
*         Hyperbolic Mohr-Coulomb surface
*
          YLD = SIGM*SPHI+SQRT(SBAR*SBAR*K*K+ASPHI2)-COH*CPHI
*
        ELSE
*
*         Mohr-Coulomb surface
*
          YLD = SIGM*SPHI+SBAR*K-COH*CPHI
*
        ENDIF
*
      ELSEIF (YFTYPE.EQ.TRESCA) THEN
*-----------------------------------------------------------------------
*       Tresca yield function
*       Set value of material parameters used in yield function
*
        COH = MPROP(17)
*
        IF (ABS(THETA).LT.ATTRAN) THEN
*
*         Calculate K function for unrounded region of octahedral plane
*
          K = COS(THETA)
*
        ELSE
*
*         Calculate K function for rounded region of octahedral plane
*
          SGN = SIGN(C00001,THETA)
          A = A1
          B = B1*SGN
          K = A-B*S3TA
*
        ENDIF
*
*       Calculate value of yield function
*
        YLD = SBAR*K-COH
*
      ELSE
*-----------------------------------------------------------------------
*       Invalid yield function type
*
        WRITE(IOW,'('' *** ERROR IN SUBROUTINE YIELD ***'')')
        WRITE(IOW,'('' INVALID YIELD FUNCTION - YFTYPE = '',I4)')YFTYPE
        STOP
*
      ENDIF
      END
```

APPENDIX B

```
      SUBROUTINE GRAD(GY1,GY2,GY3,GY4,GP1,GP2,GP3,GP4,SIGXX,SIGYY,SIGXY,
     +                SIGZZ,MPROP,NMP,IOW)
************************************************************************
*     PURPOSE:
*     --------
*
*     This subroutine returns the value of the gradient to the yield
*     surface and plastic potential at a given stress state for plane
*     strain and axisymmetric plasticity. Smooth approximations to the
*     Mohr-Coulomb and Tresca criteria are used. The routine is designed
*     for both associated and non-associated flow rules.
*
*     INPUT:
*     ------
*     GY1..GY4   -  Undefined on entry
*     GP1..GP4   -  Undefined on entry
*     SIGXX      -  XX-component of normal stress
*     SIGYY      -  YY-component of normal stress
*     SIGXY      -  XY-component of shear  stress
*     SIGZZ      -  ZZ-component of normal stress
*     MPROP      -  Vector of dimension (NMP)
*                -  Contains material parameters
*                -  MPROP(8)   = Value specifying type of flow rule
*                            0 = associated flow
*                            1 = non-associated flow
*                -  MPROP(9)   = (a*SIN(dilation angle))**2
*                            a = hyperbolic rounding parameter
*                -  MPROP(10)  = (a*SIN(friction angle))**2
*                            a = hyperbolic rounding parameter
*                -  MPROP(11)  = SIN(friction angle)
*                -  MPROP(13)  = SIN(dilation angle)
*                -  MPROP(20)  = Value defining type of yield function
*                            1 = Mohr-Coulomb rounded in octahedral plane
*                            2 = Hyperbolic Mohr-Coulomb rounded in
*                                octahedral plane
*                            3 = Tresca rounded in octahedral plane
*     NMP        -  Parameter specifying number of material parameters
*     IOW        -  Unit number of output file
*
*     OUTPUT:
*     -------
*     GY1        -  Derivative of yield function wrt SIGXX
*     GY2        -  Derivative of yield function wrt SIGYY
*     GY3        -  Derivative of yield function wrt SIGXY
*     GY4        -  Derivative of yield function wrt SIGZZ
*     GP1        -  Derivative of plastic potential wrt SIGXX
*     GP2        -  Derivative of plastic potential wrt SIGYY
*     GP3        -  Derivative of plastic potential wrt SIGXY
*     GP4        -  Derivative of plastic potential wrt SIGZZ
*     SIGXX      -  Unchanged
*     SIGYY      -  Unchanged
*     SIGXY      -  Unchanged
*     SIGZZ      -  Unchanged
*     MPROP      -  Unchanged
*     NMP        -  Unchanged
*     FLAG       -  Unchanged
*     IOW        -  Unchanged
*
*     SUBROUTINES CALLED:   NONE
*     -------------------
*
*     PROGRAMMER:   Andrew Abbo
*     -----------
*
*     LAST MODIFIED:  May 1993      Andrew Abbo
*     --------------
*
************************************************************************
```

```
      INTEGER YFTYPE,NMP,IOW
      INTEGER FLOW
*
      DOUBLE PRECISION SIGXX,SIGYY,SIGZZ,SIGXY
      DOUBLE PRECISION SPHI,SPSI,ASPHI2,ASPSI2
      DOUBLE PRECISION DSIGX,DSIGY,DSIGZ
      DOUBLE PRECISION THETA,SIGM,J2,J3,J23,SBAR,ALPHA
      DOUBLE PRECISION STA,CTA,C3TA,S3TA,T3TA
      DOUBLE PRECISION A,B,SGN,K,DK
      DOUBLE PRECISION C1,C2,C3
      DOUBLE PRECISION GY1,GY2,GY3,GY4
      DOUBLE PRECISION GP1,GP2,GP3,GP4
*
      DOUBLE PRECISION MPROP(NMP)
*
*     Set constants
*
      INTEGER MOHR,HYPER,TRESCA,ASSOC
      PARAMETER( MOHR = 1 )
      PARAMETER( HYPER = 2 )
      PARAMETER( TRESCA = 3 )
      PARAMETER( ASSOC = 0 )
*
      DOUBLE PRECISION TINY
      PARAMETER( TINY = 1.0D-15 )
*
      DOUBLE PRECISION C004P5,C000P5,CP3333
      PARAMETER( C004P5 = 4.5D0 )
      PARAMETER( C000P5 = 0.5D0 )
      PARAMETER( CP3333 = 0.3333333333333333D0 )
*
      DOUBLE PRECISION C00000,C00001,C00002,C00003,C00004
      PARAMETER( C00000 = 0.0D0 )
      PARAMETER( C00001 = 1.0D0 )
      PARAMETER( C00002 = 2.0D0 )
      PARAMETER( C00003 = 3.0D0 )
      PARAMETER( C00004 = 4.0D0 )
*
      DOUBLE PRECISION  C000R3,C00IR3,CP8660
      PARAMETER( C000R3 = 1.732050807568877D0 )
      PARAMETER( C00IR3 = 0.5773502691896258D0 )
      PARAMETER( CP8660 = 0.866025403784439D0 )
*
*     Constants for rounded K function
*
      DOUBLE PRECISION A1,A2,B1,B2,ATTRAN
*
*     Rounding constants for theta > 25 degrees
*
      PARAMETER( A1 = 1.432052062044227D0 )
      PARAMETER( A2 = 0.406941858374615D0 )
      PARAMETER( B1 = 0.544290524902313D0 )
      PARAMETER( B2 = 0.673903324498392D0 )
      PARAMETER( ATTRAN=0.436332312998582D0 )
*
*     Calculate value of invariants for the current stress state.
*
      SIGM = CP3333*(SIGXX+SIGYY+SIGZZ)
      DSIGX = SIGXX-SIGM
      DSIGY = SIGYY-SIGM
      DSIGZ = SIGZZ-SIGM
      J2 = C000P5*(DSIGX*DSIGX+DSIGY*DSIGY+DSIGZ*DSIGZ) + SIGXY*SIGXY
      J3 = DSIGZ*(DSIGX*DSIGY-SIGXY*SIGXY)
*
      SBAR=SQRT(J2)
*
*     Store type of flow rule
*     If MPROP(8)=1 then have non-associated flow rule
*     If have associated flow rule, then the gradients to the plastic
*     potential and the yield function will be set equal
```

```
*
      FLOW=INT(MPROP(8))
*
*     Extract form of yield function from MPROP vector
*
      YFTYPE=INT(MPROP(20))
*
      IF ((YFTYPE.EQ.MOHR).OR.(YFTYPE.EQ.HYPER)) THEN
*-----------------------------------------------------------------------
*       Mohr-Coulomb or hyperbolic Mohr-Coulomb yield function
*
        IF (J2.GT.C00000) THEN
*
*         Calculate third stress invariant
*
          S3TA = -C004P5*J3/(C000R3*SBAR*J2)
          IF (S3TA.LT.-C00001) THEN
            S3TA = -C00001
          ELSEIF (S3TA.GT.C00001) THEN
            S3TA = C00001
          ENDIF
          THETA  = CP3333*ASIN(S3TA)
*
        ELSE
*
*         Special case of zero deviatoric stress
*
          J2     = TINY
          SBAR   = C00000
          THETA  = C00000
          S3TA   = C00000
*
        ENDIF
*
*       Set value of material parameters used in gradient calculations
*
        SPHI   = MPROP(11)
        ASPHI2 = MPROP(10)
*
        CTA  = COS(THETA)
        C3TA = CTA*(C00004*CTA*CTA-C00003)
        T3TA = S3TA/C3TA
*
*       Calculate K function and its derivative wrt theta DK
*
        IF (ABS(THETA).LT.ATTRAN) THEN
*
*         Unrounded surface
*
          STA = S3TA/(C00004*CTA*CTA-C00001)
          K   = CTA-STA*SPHI*C00IR3
          DK  = STA+CTA*SPHI*C00IR3
*
        ELSE
*
*         Rounded surface
*
          SGN = SIGN(C00001,THETA)
          A   = A1 + A2*SGN*SPHI
          B   = B1*SGN + B2*SPHI
*
          K   = A-B*S3TA
          DK  = C00003*B*C3TA
*
        ENDIF
*
*       Calculate gradient coefficients for Mohr-Coulomb surface
*
```

```
        C1 = SPHI
        C2 = K+T3TA*DK
        C3 = CP8660*DK/(J2*C3TA)
*
*       Adjust coefficients for hyperbolic Mohr-Coulomb surface
*
        IF (YFTYPE.EQ.HYPER) THEN
          ALPHA = SBAR*K
          ALPHA = ALPHA/SQRT(ALPHA*ALPHA + ASPHI2)
          C2    = C2*ALPHA
          C3    = C3*ALPHA
        ENDIF
*
      ELSEIF ( YFTYPE.EQ.TRESCA) THEN
*-----------------------------------------------------------------------
*       Tresca Yield Function
*
        IF (J2.GT.C00000) THEN
*
*         Calculate third stress invariant
*
          S3TA = -C004P5*J3/(C000R3*SBAR*J2)
          IF (S3TA.LT.-C00001) THEN
            S3TA = -C00001
          ELSEIF (S3TA.GT.C00001) THEN
            S3TA = C00001
          ENDIF
          THETA = CP3333*ASIN(S3TA)
*
        ELSE
*
*         Cannot have yielding at zero deviatoric stress for Tresca
*
          WRITE(IOW,'('' *** ERROR IN SUBROUTINE GRAD ***'')')
          WRITE(IOW,'('' J2=0 FOR TRESCA YIELD FUNCTION'')')
          STOP
*
        ENDIF
*
*       Set value of parameters used in gradient calculations
*
        CTA   = COS(THETA)
        C3TA  = CTA*(C00004*CTA*CTA-C00003)
        T3TA  = S3TA/C3TA
*
*       Calculate K function and its derivative wrt theta DK
*
        IF (ABS(THETA).LT.ATTRAN) THEN
*
*         Unrounded surface
*
          STA = S3TA/(C00004*CTA*CTA-C00001)
          K   = CTA
          DK  = STA
*
        ELSE
*
*         Rounded surface
*
          SGN = SIGN(C00001,THETA)
          A   = A1
          B   = B1*SGN
*
          K   = A-B*S3TA
          DK  = C00003*B*C3TA
*
        ENDIF
*
*       Calculate gradient coefficients
*
```

```
          C1 = C00000
          C2 = K+T3TA*DK
          C3 = CP8660*DK/(J2*C3TA)
*
        ELSE
*-----------------------------------------------------------------------
*         Invalid yield function type
*
          WRITE(IOW,'('' *** ERROR IN SUBROUTINE GRAD ***'')')
          WRITE(IOW,'('' INVALID YIELD FUNCTION - YFTYPE = '',I4)')YFTYPE
          STOP
*
        ENDIF
*
*       Compose gradient to yield function
*
        J23  = J2*CP3333
        C2   = C2*C000P5/SBAR
        C1   = C1*CP3333
        GY1 = C1 +    C2*DSIGX + C3*(DSIGY*DSIGZ + J23)
        GY2 = C1 +    C2*DSIGY + C3*(DSIGX*DSIGZ + J23)
        GY3 = C00002*SIGXY*(C2 - C3* DSIGZ)
        GY4 = C1 +    C2*DSIGZ + C3*(DSIGX*DSIGY-SIGXY*SIGXY+J23)
*
*       Calculate gradient to potential for associated case
*
        IF (FLOW.EQ.ASSOC) THEN
          GP1 = GY1
          GP2 = GY2
          GP3 = GY3
          GP4 = GY4
          RETURN
        ENDIF
*
*       If non-associated flow calculate gradient to plastic potential
*       Assume that the plastic potential has the same form as the yield
*       function except that the dilation angle is substituted for the
*       friction angle
*
        IF ((YFTYPE.EQ.MOHR).OR.(YFTYPE.EQ.HYPER)) THEN
*-----------------------------------------------------------------------
*         Mohr-Coulomb or hyperbolic Mohr-Coulomb plastic potential
*         Extract material parameters
*
          SPSI   = MPROP(13)
          ASPSI2 = MPROP(9)
*
*         Calculate K function and its derivative wrt theta
*
          IF (ABS(THETA).LT.ATTRAN) THEN
*
*           Unrounded surface
*
            K  = CTA-STA*SPSI*C00IR3
            DK = STA+CTA*SPSI*C00IR3
*
          ELSE
*
*           Rounded surface
*
            A  = A1 + A2*SGN*SPSI
            B  = B1*SGN+B2*SPSI
*
            K  = A-B*S3TA
            DK = C00003*B*C3TA
*
          ENDIF
*
*         Calculate gradient coefficients for Mohr-Coulomb surface
*
```

```
        C1 = SPSI
        C2 = K+T3TA*DK
        C3 = CP8660*DK/(J2*C3TA)
*
*       Adjust coefficients for hyperbolic Mohr-Coulomb surface
*
        IF (YFTYPE.EQ.HYPER) THEN
          ALPHA = SBAR*K
          ALPHA = ALPHA/SQRT(ALPHA*ALPHA + ASPSI2)
          C2    = C2*ALPHA
          C3    = C3*ALPHA
        ENDIF
*
      ELSEIF ( YFTYPE.EQ.TRESCA) THEN
*-----------------------------------------------------------------------------
*       Tresca Yield Function
*
        GP1 = GY1
        GP2 = GY2
        GP3 = GY3
        GP4 = GY4
        RETURN
*
      ENDIF
*
*     Compose gradient to plastic potential
*
      C1  = C1*CP3333
      C2  = C2*C000P5/SBAR
      GP1 = C1 +    C2*DSIGX + C3*(DSIGY*DSIGZ + J23 )
      GP2 = C1 +    C2*DSIGY + C3*(DSIGX*DSIGZ + J23 )
      GP3 = C00002*SIGXY*(C2 - C3* DSIGZ)
      GP4 = C1 +    C2*DSIGZ + C3*(DSIGX*DSIGY-SIGXY*SIGXY+J23)
*
      END
```