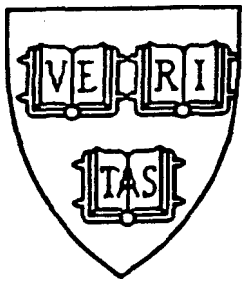


MECH - 178

A USER-MATERIAL SUBROUTINE INCORPORATING SINGLE CRYSTAL PLASTICITY IN THE ABAQUS FINITE ELEMENT PROGRAM

Yonggang Huang

Division of Applied Sciences
Harvard University
Cambridge, Massachusetts
02138

June 1991

A USER-MATERIAL SUBROUTINE INCORPORATING SINGLE CRYSTAL PLASTICITY IN THE ABAQUS FINITE ELEMENT PROGRAM

Yonggang Huang
Division of Applied Sciences
Harvard University
Cambridge, MA 02138

June 1991

Abstract

A user-material subroutine has been written to incorporate single crystal plasticity in the finite element program ABAQUS. The finite-element formulation of elastic-plastic and viscoplastic single crystal deformation is reviewed in this paper, including versions for small deformation theory and for the rigorous theory of finite-strain and finite-rotation. Inelastic deformation of a single crystal arises from crystalline slip, which is assumed here to obey the Schmid law. Various self and latent hardening relations between resolved shear stress and shear strain in slip systems are presented and incorporated as options in the subroutine.

1. Introduction

The finite element program ABAQUS has been widely used in the deformation and stress analysis of solids. Besides a broad range of constitutive models, ABAQUS also provides an interface whereby the user may write his or her own constitutive model in a subroutine denoted UMAT in a very general way (ABAQUS User's Manual, 1989). The stresses, strains and solution dependent state variables are solved incrementally by ABAQUS. When the subroutine UMAT is called, it is provided with the state at the start of the increment (stress, solution dependent state variables) and with the strain increments and the time increment. The subroutine UMAT performs two functions: it updates the stresses and the solution dependent state variables to their values at the end of the increment, and it provides the material Jacobian matrix, $\partial\Delta\boldsymbol{\sigma}/\partial\Delta\boldsymbol{\varepsilon}$, for the constitutive model as required for an iterative Newton-Rhapson solution.

The main objective of this paper is to provide the finite element code ABAQUS a user-material subroutine for the constitutive relation of single crystals in a continuum framework. The kinematical structure here falls within the framework laid out by Rice (1971) and Hill and Rice (1972), rigorously accommodating finite deformation effects. The plastic deformation is assumed due solely to the crystallographic dislocation slip; deformation by diffusion, twinning and grain boundary sliding is not considered here. The Schmid stress, or resolved shear stress on a slip system, is assumed here to be the driving force for slip. The completion of this subroutine makes it possible to use ABAQUS for stress and fracture analyses of single crystals and bicrystals. The finite element analysis of single crystals was first studied by Peirce, Asaro and Needleman (1982, 1983).

2. Review of Elastic-Plastic Constitutive Formulation for Single Crystals

2.1 Kinematics

The kinematical theory for the mechanics of crystals outlined here follows the pioneering work of Taylor (1938) and its precise mathematical theory by Hill (1966), Rice (1971), and Hill and Rice (1972). The following is a simple summary of the theory, followed Asaro and Rice (1977) and Asaro (1983).

A crystalline material is embedded on its lattice which undergoes elastic deformation and rotation. The inelastic deformation of a single crystal is assumed here to arise solely from crystalline slip. The material flows through the crystal lattice via dislocation motion. The total deformation gradient $\mathbf{F}$ is given by

$$\mathbf{F}=\mathbf{F}^{*}\cdot\mathbf{F}^{p} \tag{2.1.1}$$

where $\mathbf{F}^P$ denotes plastic shear of the material to an intermediate reference configuration in which lattice orientation and spacing are the same as in the original reference configuration, and where $\mathbf{F}^*$ denotes stretching and rotation of the lattice. Elastic properties are assumed to be unaffected by slip, in the sense that stress is determined solely by $\mathbf{F}^*$. The rate of change of $\mathbf{F}^P$ is related to the slipping rate $\dot{\gamma}^{(\alpha)}$ of the α slip system by

$$\dot{\mathbf{F}}^P \cdot \mathbf{F}^{P-1} = \sum_{\alpha} \dot{\gamma}^{(\alpha)} \mathbf{s}^{(\alpha)} \mathbf{m}^{(\alpha)} \tag{2.1.1a}$$

where the sum ranges over all activated slip systems, unit vectors $\mathbf{s}^{(\alpha)}$ and $\mathbf{m}^{(\alpha)}$ are the slip direction and normal to slip plane in the reference configuration, respectively.

It is convenient to define the vector $\mathbf{s}^{*(\alpha)}$, lying along the slip direction of system α in the deformed configuration, by

$$\mathbf{s}^{*(\alpha)} = \mathbf{F}^* \cdot \mathbf{s}^{(\alpha)} \tag{2.1.2a}$$

A normal to the slip plane which is the reciprocal base vector to all such vectors in the slip plane is

$$\mathbf{m}^{*(\alpha)} = \mathbf{m}^{(\alpha)} \cdot \mathbf{F}^{*-1} \tag{2.1.2b}$$

The velocity gradient in the current state is

$$\mathbf{L} \equiv \dot{\mathbf{F}} \cdot \mathbf{F}^{-1} = \mathbf{D} + \mathbf{\Omega} \tag{2.1.3}$$

where the symmetric rate of stretching $\mathbf{D}$ and the antisymmetric spin tensor $\mathbf{\Omega}$ may be decomposed into lattice parts (superscript *) and plastic parts (superscript P) as follows:

$$\mathbf{D} = \mathbf{D}^* + \mathbf{D}^P, \quad \mathbf{\Omega} = \mathbf{\Omega}^* + \mathbf{\Omega}^P \tag{2.1.4}$$

satisfying

$$\mathbf{D}^* + \mathbf{\Omega}^* = \dot{\mathbf{F}}^* \cdot \mathbf{F}^{*-1}, \quad \mathbf{D}^P + \mathbf{\Omega}^P = \sum_{\alpha} \dot{\gamma}^{(\alpha)} \mathbf{s}^{*(\alpha)} \mathbf{m}^{*(\alpha)} \tag{2.1.5}$$

2.2 Constitutive laws

Following Hill and Rice (1972), the existence of an elastic potential, $\Phi = \Phi(\mathbf{F}^*)$, assures that the relation between the symmetric rate of stretching of the lattice, $\mathbf{D}^*$, and the Jaumann rate of Cauchy stress $\boldsymbol{\sigma}$, $\overset{\nabla^*}{\boldsymbol{\sigma}}$, is given by

$$\overset{\nabla^*}{\boldsymbol{\sigma}} + \boldsymbol{\sigma}\left(\mathbf{I}:\mathbf{D}^*\right) = \mathbf{L}:\mathbf{D}^* \tag{2.2.1}$$

where $\mathbf{I}$ is the second order identical tensor, $\mathbf{L}$ is the tensor of elastic moduli having the full set of symmetries $L_{ijkl}=L_{jikl}=L_{ijlk}=L_{klij}$, the Jaumann rate $\overset{\nabla^*}{\boldsymbol{\sigma}}$ is the corotational stress rate on axes that rotate with the crystal lattice, which is related to the corotational stress rate on axes rotating with the material, $\overset{\nabla}{\boldsymbol{\sigma}}$, by

$$\overset{\nabla^*}{\boldsymbol{\sigma}} = \overset{\nabla}{\boldsymbol{\sigma}} + \left(\mathbf{\Omega} - \mathbf{\Omega}^*\right) \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \left(\mathbf{\Omega} - \mathbf{\Omega}^*\right) \tag{2.2.2}$$

where $\overset{\nabla}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} - \mathbf{\Omega} \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \mathbf{\Omega}$.

The crystalline slip is assumed here to obey Schmid's law, i.e. the slipping rate $\dot{\gamma}^{(\alpha)}$ in any particular slip system α is assumed to depend on the current σ solely through the so-called Schmid stress, $\tau^{(\alpha)}$. The Schmid stress is just the resolved shear stress when elastic lattice distortions are negligible. There are many possible generalizations in the presence of finite elastic distortions, some discussed by Asaro and Rice (1977). Here we use the version based on the Rice's (1971) thermodynamic stress conjugate to slip, which Rice has shown to precisely preserve the normality structure of the small deformation theory (Mandel, 1965; Hill, 1967; Rice, 1970) in terms of work conjugate stress and strain measures for finite deformation. Thus we use the definition

$$\tau^{(\alpha)} = \mathbf{m}^{*(\alpha)} \cdot \frac{\rho_0}{\rho} \boldsymbol{\sigma} \cdot \mathbf{s}^{*(\alpha)} \tag{2.2.3}$$

where ρ_0 and ρ are the mass density in the reference and current states; Hill and Rice (1972) note that this $\tau^{(\alpha)}$ is $\tau^{m}{}_{s}$, the mixed shear component of Kirchhoff stress $\boldsymbol{\tau}$ on coordinates which convect with the lattice. The rate of change of this Schmid stress is given by

$$\dot{\tau}^{(\alpha)} = \mathbf{m}^{*(\alpha)} \cdot \left[\overset{\nabla^*}{\boldsymbol{\sigma}} + \boldsymbol{\sigma}\left(\mathbf{I}:\mathbf{D}^*\right) - \mathbf{D}^* \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \mathbf{D}^* \right] \cdot \mathbf{s}^{*(\alpha)} \tag{2.2.4}$$

2.3 Hardening of rate-dependent crystalline materials

It has been noted by Peirce, Asaro and Needleman (1983) that rate-independent plasticity may be treated as the limit of rate-dependent viscoplasticity. The constitutive formulation in the present report is given within this viscoplastic framework. The hardening of single crystals has been discussed by many authors (e.g. see the review article by Asaro, 1983a; and Wu, Bassani and Laird, 1991). Based on the Schmid law, the slipping rate $\dot{\gamma}^{(\alpha)}$ of the αth slip system in a rate-dependent crystalline solid is determined by the corresponding resolved shear stress $\tau^{(\alpha)}$ as

$$\dot{\gamma}^{(\alpha)} = \dot{a}^{(\alpha)} f^{(\alpha)}\left(\tau^{(\alpha)} / g^{(\alpha)}\right) \tag{2.3.1}$$

where the constant $\dot{a}^{(\alpha)}$ is the reference strain rate on slip system α, $g^{(\alpha)}$ is a variable which describes the current strength of that system, and the nondimensional function $f^{(\alpha)}$ is a general function describing the dependence of strain rate on the stress. Hutchinson (1976) used a simple power law for polycrystalline creep:

$$f^{(\alpha)}(x) = x|x|^{n-1} \tag{2.3.1a}$$

where n is the rate sensitivity exponent. In the limit as $n \to \infty$ this power law approaches that of a rate-independent material.

The strain hardening is characterized by the evolution of the strengths $g^{(\alpha)}$ through the incremental relation:

$$\dot{g}^{(\alpha)} = \sum_{\beta} h_{\alpha\beta} \dot{\gamma}^{(\beta)} \tag{2.3.2}$$

where $h_{\alpha\beta}$ are the slip hardening moduli, the sum ranges over all activated slip systems. Here $h_{\alpha\alpha}$ (no sum) and $h_{\alpha\beta}$ ($\alpha\neq\beta$) are called self and latent hardening moduli, respectively.

Peirce, Asaro and Needleman (1982), and Asaro (1983a, b) have used a simple form for the self hardening moduli:

$$h_{\alpha\alpha} = h(\gamma) = h_0 \operatorname{sech}^2\left|\frac{h_0\gamma}{\tau_s - \tau_0}\right| \quad \text{(no sum on } \alpha\text{)} \tag{2.3.3a}$$

where h_0 is the initial hardening modulus, τ_0 is the yield stress which equals the initial value of current strength $g^{(\alpha)}(0)$, τ_s is the stage I stress (or the break-through stress where large plastic flow initiates), and γ is the Taylor cumulative shear strain on all slip systems, i.e.

$$\gamma = \sum_{\alpha} \int_0^t \left|\dot{\gamma}^{(\alpha)}\right| dt \tag{2.3.3b}$$

The latent hardening moduli are given by

$$h_{\alpha\beta} = q h(\gamma) \quad (\alpha \neq \beta) \tag{2.3.3c}$$

where q is a constant. These expressions of hardening moduli neglect the Bauschinger effect in a crystalline solid.

Bassani and Wu (1991) have used a different expression for the hardening moduli to describe the three stage hardening of crystalline materials. Their expression depends on the shear strains $\gamma^{(\alpha)}$ of all slip systems:

$$h_{\alpha\alpha} = \left\{(h_0 - h_s)\operatorname{sech}^2\left[\frac{(h_0 - h_s)\gamma^{(\alpha)}}{\tau_s - \tau_0}\right] + h_s\right\} G\left(\gamma^{(\beta)}; \beta \neq \alpha\right) \quad \text{(no sum on } \alpha\text{)} \tag{2.3.4a}$$

$$h_{\beta\alpha} = q h_{\alpha\alpha} \quad (\beta \neq \alpha) \tag{2.3.4b}$$

where the newly introduced h_s is the hardening modulus during easy glide within the stage I hardening, the function G is associated with interactive (cross) hardening and given by

$$G\left(\gamma^{(\beta)}; \beta \neq \alpha\right) = 1 + \sum_{\beta\neq\alpha} f_{\alpha\beta} \tanh\left(\gamma^{(\beta)} / \gamma_0\right) \tag{2.3.4c}$$

where γ_0 is the amount of slip after which the interaction between slip systems reaches the peak strength, and each component $f_{\alpha\beta}$ represents the magnitude of the strength of a particular slip interaction. For example, coplanar interactions tend to be weaker than non-coplanar ones. For FCC single crystals there are five distinct slip interactions, i.e. there are at most five independent components of $f_{\alpha\beta}$.

In these formulations there is no explicit yielding; if the resolved shear stress on a system is non-zero, then plastic shearing occurs. However, for large values of the rate sensitivity exponent n ($n \geq 50$) the plastic shearing rate on slip systems with a resolved shear stress less than τ_0 is exceedingly small compared to the reference rate $\dot{a}$. Since within the present formulation all systems are potentially active, it is neither necessary nor convenient to consider $(\mathbf{s}^{*(\alpha)}, \mathbf{m}^{*(\alpha)})$ and

$(-\mathbf{s}^{*(\alpha)}, \mathbf{m}^{*(\alpha)})$ as separate slip systems on each of which only positive slip is allowed. Thus, we permit $\dot{\gamma}^{(\alpha)}$ to be negative if the corresponding $\tau^{(\alpha)}$ is negative, as in eq. (2.3.1a).

There are other types of models on the slip hardening (e.g., Zarka, 1975) which also fall into the above general framework (2.3.1) and (2.3.2) though more parameters, which may be considered as the internal variables in classical plasticity theory, are introduced.

3. Forward Gradient Time Integration Scheme and the Incremental Formulation

Two time integration schemes are used in the present paper. The first one assumes a linear relation among the increments of stresses, strains and state variables such as shear strains, resolved shear stresses, current strengths in slip systems, as described in Sections 3.1-3.3. The stresses and state variables are evaluated at the start of the time increment. The second scheme solves the nonlinear incremental equations by a Newton-Rhapson iterative method, as discussed in Section 3.4. An implicit time integration scheme is used in which the stresses and state variables are evaluated at the end of the time increment.

3.1 Forward gradient time integration scheme

The tangent modulus method for rate dependent solid developed by Peirce, Shih, and Needleman (1984) is used in the subroutine. We define the increment of shear strain $\gamma^{(\alpha)}$ in slip system α within the time increment Δt by

$$\Delta\gamma^{(\alpha)} = \gamma^{(\alpha)}(t+\Delta t) - \gamma^{(\alpha)}(t) \tag{3.1.1}$$

and employ a linear interpolation within Δt:

$$\Delta\gamma^{(\alpha)} = \Delta t\left[(1-\theta)\dot{\gamma}_t^{(\alpha)} + \theta\dot{\gamma}_{t+\Delta t}^{(\alpha)}\right] \tag{3.1.2}$$

where the subscript is the time at which the slipping rate $\dot{\gamma}^{(\alpha)}$ is the evaluated. The parameter θ ranges from 0 to 1, with θ=0 corresponding to the simple Euler time integration scheme. A choice of θ between 0.5 and 1 is recommended (Peirce *et al*, 1984).

The slipping rate $\dot{\gamma}^{(\alpha)}$ in general is a function of the resolved shear stress $\tau^{(\alpha)}$ and the current strength $g^{(\alpha)}$ (see eq. (2.3.1)). The Taylor expansion of slipping rate gives

$$\dot{\gamma}_{t+\Delta t}^{(\alpha)} = \dot{\gamma}_t^{(\alpha)} + \frac{\partial\dot{\gamma}^{(\alpha)}}{\partial\tau^{(\alpha)}}\Delta\tau^{(\alpha)} + \frac{\partial\dot{\gamma}^{(\alpha)}}{\partial g^{(\alpha)}}\Delta g^{(\alpha)} \tag{3.1.3}$$

where $\Delta\tau^{(\alpha)}$ and $\Delta g^{(\alpha)}$ are the increments of resolved shear stress and current strength in slip system α within the time increment Δt, respectively. Eqs. (3.1.1)-(3.1.3) have been rearranged to give the following incremental relation:

$$\Delta\gamma^{(\alpha)} = \Delta t\left[\dot{\gamma}_t^{(\alpha)} + \theta\frac{\partial\dot{\gamma}^{(\alpha)}}{\partial\tau^{(\alpha)}}\Delta\tau^{(\alpha)} + \theta\frac{\partial\dot{\gamma}^{(\alpha)}}{\partial g^{(\alpha)}}\Delta g^{(\alpha)}\right] \tag{3.1.4}$$

3.2 Incremental Formulation

The relations are derived in this section for the increments of shear strain $\Delta\gamma^{(\alpha)}$, resolved shear stress $\Delta\tau^{(\alpha)}$, current strength $\Delta g^{(\alpha)}$ in all slip systems, in terms of the strain increments $\Delta\varepsilon_{ij}$ and time increment Δt. The corotational stress increments $\Delta\sigma_{ij} = \overset{\nabla}{\sigma}_{ij}\Delta t$ are also expressed in terms of strain increments $\Delta\varepsilon_{ij}$. This definition of stress increment is consistent with the finite element code ABAQUS (ABAQUS theory manual, 1989; also Hughes and Winget, 1980) for finite deformation analysis.

It is convenient to introduce for each slip system the "Schmid factor" $\mu_{ij}^{(\alpha)}$ and tensor $\omega_{ij}^{(\alpha)}$ defined as

$$\mu_{ij}^{(\alpha)} = \frac{1}{2}\left[s_i^{*(\alpha)}m_j^{*(\alpha)} + s_j^{*(\alpha)}m_i^{*(\alpha)}\right] \tag{3.2.1a}$$

$$\omega_{ij}^{(\alpha)} = \frac{1}{2}\left[s_i^{*(\alpha)}m_j^{*(\alpha)} - s_j^{*(\alpha)}m_i^{*(\alpha)}\right] \tag{3.2.1b}$$

The tensor $\omega_{ij}^{(\alpha)}$ is related to the spin tensors Ω and Ω^* by

$$\Omega_{ij} - \Omega_{ij}^* = \sum_{\alpha}\omega_{ij}^{(\alpha)}\dot{\gamma}^{(\alpha)} \tag{3.2.1c}$$

From the general hardening equation of crystalline slip (2.3.2), the increments of current hardening function $\Delta g^{(\alpha)}$ are given by

$$\Delta g^{(\alpha)} = \sum_{\beta} h_{\alpha\beta}\Delta\gamma^{(\beta)} \tag{3.2.2}$$

The increments of resolved shear stress $\Delta\tau^{(\alpha)}$ are related to the strain increments $\Delta\varepsilon_{ij}$ through eq. (2.2.4), the elastic constitutive law (2.2.1), and the decomposition of strain increments to lattice parts and plastic parts (2.1.4), (2.1.5),

$$\Delta\tau^{(\alpha)} = \left[L_{ijkl}\mu_{kl}^{(\alpha)} + \omega_{ik}^{(\alpha)}\sigma_{jk} + \omega_{jk}^{(\alpha)}\sigma_{ik}\right]\cdot\left[\Delta\varepsilon_{ij} - \sum_{\beta}\mu_{ij}^{(\beta)}\Delta\gamma^{(\beta)}\right] \tag{3.2.3}$$

where L_{ijkl} are the elastic moduli. The corotational stress increments $\Delta\sigma_{ij}$ are given by eq. (2.2.2) as

$$\Delta\sigma_{ij} = L_{ijkl}\Delta\varepsilon_{kl} - \sigma_{ij}\Delta\varepsilon_{kk} - \sum_{\alpha}\left[L_{ijkl}\mu_{kl}^{(\alpha)} + \omega_{ik}^{(\alpha)}\sigma_{jk} + \omega_{jk}^{(\alpha)}\sigma_{ik}\right]\Delta\gamma^{(\alpha)} \tag{3.2.4}$$

For given strain increments $\Delta\varepsilon_{ij}$, the increments of shear strain $\Delta\gamma^{(\alpha)}$ in the slip systems are uniquely determined by the following linear algebraic equation, which is obtained by substituting the above incremental relations (3.2.2) and (3.2.3) into (3.1.4),

$$\sum_{\beta}\left\{\delta_{\alpha\beta}+\theta\Delta t\frac{\partial\dot{\gamma}^{(\alpha)}}{\partial\tau^{(\alpha)}}\left[L_{ijkl}\mu_{kl}^{(\alpha)}+\omega_{ik}^{(\alpha)}\sigma_{jk}+\omega_{jk}^{(\alpha)}\sigma_{ik}\right]\mu_{ij}^{(\beta)}-\theta\Delta t\frac{\partial\dot{\gamma}^{(\alpha)}}{\partial g^{(\alpha)}}h_{\alpha\beta}\text{sign}\left(\dot{\gamma}_t^{(\beta)}\right)\right\}\Delta\gamma^{(\beta)}$$

$$=\dot{\gamma}_t^{(\alpha)}\Delta t+\theta\Delta t\frac{\partial\dot{\gamma}^{(\alpha)}}{\partial\tau^{(\alpha)}}\left[L_{ijkl}\mu_{kl}^{(\alpha)}+\omega_{ik}^{(\alpha)}\sigma_{jk}+\omega_{jk}^{(\alpha)}\sigma_{ik}\right]\Delta\varepsilon_{ij} \tag{3.2.5}$$

where $\delta_{\alpha\beta}$ is the Kronecker delta. Once the $\Delta\gamma^{(\alpha)}$ are known in terms of the strain increments $\Delta\varepsilon_{ij}$, all other increments can be found through eqs. (3.2.2)-(3.2.4).

3.3 Lattice Rotation

The crystal lattice undergoes distortion and rotation as the crystal deforms; however, the effect of lattice rotation does not explicitly appear in the constitutive equations in Section 2 when all rate quantities are formed on this rotating lattice frame (2.1.2a, b) (also, see Asaro and Rice, 1977; Asaro, 1983b). The lattice deformation and rotation are fully characterized by the reciprocal vectors coinciding with slip directions, $\mathbf{s}^{*(\alpha)}$, and normals to slip planes, $\mathbf{m}^{*(\alpha)}$, in the deformed configuration. By differentiating eqs. (2.1.2a, b), one finds

$$\dot{\mathbf{s}}^{*(\alpha)}=\left(\mathbf{D}^{*}+\boldsymbol{\Omega}^{*}\right)\cdot\mathbf{s}^{*(\alpha)} \tag{3.3.1a}$$

$$\dot{\mathbf{m}}^{*(\alpha)}=-\mathbf{m}^{*(\alpha)}\cdot\left(\mathbf{D}^{*}+\boldsymbol{\Omega}^{*}\right) \tag{3.3.1b}$$

The corresponding increments in terms of the strain increments $\Delta\varepsilon_{ij}$ and increments of shear strain $\Delta\gamma^{(\alpha)}$ in slip systems are given by

$$\Delta s_i^{*(\alpha)}=\left\{\Delta\varepsilon_{ij}+\Omega_{ij}\Delta t-\sum_{\beta}\left[\mu_{ij}^{(\beta)}+\omega_{ij}^{(\beta)}\right]\Delta\gamma^{(\beta)}\right\}s_j^{*(\alpha)} \tag{3.3.2a}$$

$$\Delta m_i^{*(\alpha)}=-m_j^{*(\alpha)}\left\{\Delta\varepsilon_{ji}+\Omega_{ji}\Delta t-\sum_{\beta}\left[\mu_{ji}^{(\beta)}+\omega_{ji}^{(\beta)}\right]\Delta\gamma^{(\beta)}\right\} \tag{3.3.2b}$$

The $\mathbf{s}^{*(\alpha)}$ and $\mathbf{m}^{*(\alpha)}$ are updated at each time step so as to obtain the "Schmid factor" $\mu_{ij}^{(\alpha)}$ and tensor $\omega_{ij}^{(\alpha)}$ defined in eqs. (3.2.1a, b) at the current state.

3.4 Nonlinear incremental formulations

The incremental equation (3.1.2) for the shear strain $\gamma^{(\alpha)}$ of slip system α still holds, but the Taylor expansion of the slipping rate (3.1.3) is not used in this section. All incremental equations except (3.2.5) in Sections 3.2 and 3.3 also hold and become nonlinear since the stresses and state variables are evaluated at the end of the time increment. The linear eq. (3.2.5) for the increments of shear strains $\Delta\gamma^{(\beta)}$ in slip systems is replaced by the following nonlinear equation,

which is obtained by substituting the general expression of the slipping rates (2.3.1) into the incremental equation (3.1.2),

$$\Delta\gamma^{(\alpha)} - (1-\theta)\Delta t\dot{\gamma}_t^{(\alpha)} - \theta\Delta t\dot{a}^{(\alpha)} f^{(\alpha)}\left(\frac{\tau_t^{(\alpha)} + \Delta\tau^{(\alpha)}}{g_t^{(\alpha)} + \Delta g^{(\alpha)}}\right) = 0 \qquad (3.4.1)$$

where the increments of resolved shear stress $\Delta\tau^{(\alpha)}$ and current strengths $\Delta g^{(\alpha)}$ are nonlinear functions of $\Delta\gamma^{(\alpha)}$ determined by eqs. (3.2.2) and (3.2.3). The above nonlinear equation of $\Delta\gamma^{(\alpha)}$ is solved by a Newton-Rhapson iterative method, while the linear solution by eq. (3.2.5) is taken as an initial guess. All other increments are determined through the same iterative procedure.

4. Subroutine for ABAQUS

4.1 User-material subroutine UMAT

A FORTRAN subroutine called UMAT has been written particularly for the finite element code ABAQUS as a "user-material" subroutine for the above constitutive model of single crystals with the forward gradient time integration scheme in Section 3. The subroutine includes options of small deformation theory and the theory of finite-strain and finite-rotation. The format of an input file for subroutine UMAT is discussed in detail in Appendix A, while an example of an input file including the source code of subroutine UMAT is given in Appendix C for a single crystal bar subject to uniaxial tension.

In subroutine UMAT for single crystals, the current strengths $g^{(\alpha)}$, shear strains $\gamma^{(\alpha)}$, resolved shear stresses $\tau^{(\alpha)}$, normals to slip planes $\mathbf{m}^{*(\alpha)}$, slip directions $\mathbf{s}^{*(\alpha)}$, and total cumulative shear strain γ on all slip systems (defined by eq. (2.3.3b)) are considered as solution dependent state variables. The format of the output for these solution dependent state variables is given in Appendix B. The stresses, strains and state variables are solved incrementally by ABAQUS. When the subroutine is called, it is provided with the state at the start of the increment (stresses, solution dependent state variables) and with the (estimated) strain increments and the time increment. The subroutine UMAT performs two functions: it updates the stresses and the solution dependent state variables to their values at the end of the increment, and it provides the material Jacobian matrix, $\partial\Delta\boldsymbol{\sigma}/\partial\Delta\boldsymbol{\varepsilon}$, for the constitutive model. This matrix depends on the forward gradient time integration scheme in Section 3 since this single crystal model is in the rate form and is integrated numerically in the subroutine.

The subroutine UMAT provides an option of using the linearized solution procedure in Sections 3.1-3.3 and evaluating the stress and solution dependent state variables at the start of the time increment (time t), or using the Newton-Rhapson iterative method to solve nonlinear incremental equations in Section 3.4 and evaluating the stress and solution dependent state

variables at the end of the time increment ($t+\Delta t$). A larger time increment is allowed for the nonlinear solution procedure since the incremental relations are more stable. In the Newton-Rhapson iterative method the Jacobian matrix $\partial\Delta\sigma / \partial\Delta\varepsilon$ has been simplified by neglecting the derivative of increments of normals to slip planes and slip directions with respect to the strain increments, $\frac{\partial\Delta\mathbf{m}^*}{\partial\Delta\varepsilon}$ and $\frac{\partial\Delta\mathbf{s}^*}{\partial\Delta\varepsilon}$. This simplification produces no error when the effect of lattice rotation is not considered. The error is on the order of the elastic strain increments, $O(\mathbf{D}^*\Delta t)$, compared with 1, if the effect of lattice rotation is included.

The increments of rotation $\Omega_{ij}\Delta t$ which are needed in eqs. (3.3.2a, b) are not provided from the interface between the user-material subroutine and the present version of main ABAQUS program. One may derive $\mathbf{\Omega}\Delta t$ from the provided rotation increment matrix $\Delta\mathbf{R}$ by

$$\Delta\mathbf{R} = \left(\mathbf{I} - \frac{1}{2}\mathbf{\Omega}\Delta t\right)^{-1} \cdot \left(\mathbf{I} + \frac{1}{2}\mathbf{\Omega}\Delta t\right) \qquad (4.1.1)$$

(ABAQUS theory manual, 1989).

The present version of subroutine UMAT is written for cubic crystals, although it may be generalized for non-cubic crystals, as will be discussed in the next section. The subroutine can accept, as input, up to three sets of slip systems for each cubic crystal. There is observation of the activation of slip systems {110}<111>, {121}<111> and {123}<111> in BCC metal crystals, and {111}<110> in FCC metal crystals (see, e.g. Hull and Bacon, 1984).

There are seven user-supplied function subprograms, F, DFDX, HSELF, HLATNT, GSLP0, DHSELF and DHLATN in the main subroutine UMAT. These characterize the crystalline slip and hardening of slip systems. The function subprogram F provides the slipping rate $\dot{\gamma}^{(\alpha)}$ by eq. (2.3.1) at the start of the increment, while function subprogram DFDX gives its derivative, $\frac{d\dot{\gamma}^{(\alpha)}}{d\left(\tau^{(\alpha)} / g^{(\alpha)}\right)}$. The power law form (2.3.1a) has been used for the general function f in eq. (2.3.1). The function subprograms HSELF and HLATNT provide the self and latent hardening moduli defined in the incremental formulation (2.3.2). The default is either Peirce *et al* (1982) and Asaro's (1983a, b) law (2.3.3), or Bassani and Wu's (1991) formulation (2.3.4), depending on the format of input data detailed in Appendix A. The function subprogram GSLP0 provides the initial value of the current strength $g^{(\alpha)}(0)$, and its default is the yield stress τ_0 in eq. (2.3.3a) or (2.3.4a). The function subprograms DHSELF and DHLATN, which are necessary only when the

Newton-Rhapson iterative method is used, give the derivative of self and latent hardening moduli, $\frac{dh_{\alpha\eta}}{d\gamma^{(\beta)}}$.

Each user-supplied function subprogram assumes physical variables are the same within a given set of crystallographically identical slip systems at the reference state (before loading is applied), although it could be different from those in other sets. In other words, all parameters (e.g. yield stress, initial hardening modulus) are the same within same set of slip systems.

It is easy to modify the power law (2.3.1a) for the slipping rate to other forms, if the expression of slipping rate still falls into the general framework of (2.3.1). Only function subprograms F and DFDX need to be changed. Similarly, only function subprograms HSELF, HLATNT and GSLP0 (also DHSELF and DHLATN if the Newton-Rhapson iterative method is used) have to be modified if a different formulation for the self and latent hardening moduli is used, as long as the general incremental relation (2.3.2) still holds. Further modifications are discussed in the next section.

The user must provide the main subroutine UMAT with the following seven groups of data in the ABAQUS input file for the problem addressed:

(1) Elastic moduli of cubic crystals;

(2) Number of sets of potentially activated slip systems;
A typical slip plane, e.g. (110) for a BCC crystal, for each set of slip systems; and
A typical slip direction, e.g. [111] for a BCC crystal, for the same set of slip systems;

(3) Initial orientation of cubic crystals in global system at the reference state;

(4) Slipping rate dependence on resolved shear stress and current strength (see eq. (2.3.1)), e.g. reference strain rate $\dot{a}^{(\alpha)}$, exponent n in power law (2.3.1a);

(5) Self and latent hardening moduli;

(6) Forward gradient time integration parameter θ; and
Parameter NLGEOM which determines whether the small deformation theory or the theory of finite-strain and finite-rotation is used in the analysis;

(7) Parameters for the iteration method (if method of Section 3.4 is used);

The structure of an input file and the more detailed format of input data are discussed in Section 4.2 and in Appendix A.

There are eight subroutines, ROTATION, SLIPSYS, STRAINRATE, LATENTHARDEN, GSLPINIT, ITERATION, LUDCMP, and LUBKSB, in the main subroutine UMAT. The relation of the first five with the main subroutine UMAT and function subprograms are shown in Fig.1. The subroutine ROTATION determines the initial orientation of a cubic crystal in the global system, while SLIPSYS generates all slip systems (slip directions and normals

to slip planes) in the same set for a cubic crystal in the reference state. The subroutine STRAINRATE, which calls function subprograms F and DFDX, calculates the slip rates in all slip systems at the start of the increment. The function subprogram F is also called by the main subroutine UMAT if the iteration method is used. The subroutine LATENTHARDEN, which calls function subprograms HSELF and HLATNT, generates the hardening matrix, i.e. the self hardening moduli on the diagonal and the latent hardening moduli on the off-diagonal. The subroutine GSLPINIT, which calls the function subprogram GSLP0, calculates the initial value of the current strength in all slip systems at the reference state. The subroutine ITERATION, which calls the function subprograms DHSELF and DHLATN, provides the arrays for the Newton-Rhapson iterative method. The last two subroutines have been used together to solve linear equations; LUDCMP does the LU decomposition, LUBKSB completes the backward substitution.

Users who are interested in an electronic copy of an .INP file containing the subroutine UMAT may write to Professor James R. Rice, Division of Applied Sciences, Harvard University, Cambridge, MA 02138, USA (e-mail: RICE@GEMS.HARVARD.EDU), giving a telnet number for their cpu, an account name and other access information so that a copy can be placed, by ftp, in their directory. The source code is stored in the VAX3100 computer GEMS of the solid mechanics group at Harvard University, and located in the account GEMS$DKB500:[RICE] as file UMATCRYSPL.INP.

4.2 Modification in ABAQUS input files for use of UMAT subroutine

The user-material subroutine UMAT must be part of the ABAQUS input file (.INP) as a material definition. Once a mesh has been defined by statements in the .INP file, the following procedures should be followed to incorporate elastic-plastic single crystal response:

(1) There must be a *USER MATERIAL card following the *MATERIAL card in the input file to define a single crystal solid. There are two parameters, CONSTANTS and UNSYMM, in the *USER MATERIAL card. The first one is required, which is the maximum number of material parameters in the model. In the present version of UMAT, this number is set to be 160 to include all seven groups of data in Section 4.1. More details of the input format are given in the Appendix A. The second parameter, UNSYMM, is used in the general case when the Jacobian matrix $\partial\Delta\sigma / \partial\Delta\varepsilon$ is not symmetric. This parameter UNSYMM may be omitted when the deformation is small and there is no slip hardening.

(2) Following the card *DEPVAR the user must provide the number of solution dependent state variables. This number equals nine times the total number of independent slip systems NSLPTL plus five, i.e. 9*NSLPTL+5. As discussed in Section 2.3, the slip system $(-\mathbf{s}^{*(\alpha)},\mathbf{m}^{*(\alpha)})$ is not considered as independent of $(\mathbf{s}^{*(\alpha)},\mathbf{m}^{*(\alpha)})$ for a cubic crystal. There are nine solution dependent state variables in each slip system, namely the current strength $g^{(\alpha)}$,

shear strain $\gamma^{(\alpha)}$, resolved shear stress $\tau^{(\alpha)}$, normal to slip plane $\mathbf{m}^{*(\alpha)}$ and slip direction $\mathbf{s}^{*(\alpha)}$. The total cumulative shear strain γ on all slip systems is also considered as a solution dependent state variable. For a FCC metal crystal the number of solution dependent state variables should be 113 (=9*12+5).

(3) There must be a *USER SUBROUTINE card followed by the source code of subroutine UMAT.

(4) To include finite strain and finite rotation effect of single crystals, the user must give parameter NLGEOM mentioned above a non-zero value. Meanwhile, the user must indicate geometric nonlinearity in the *STEP card in the.INP file.

The 160 material parameters for subroutine UMAT are put in twenty data cards with eight parameters per card, having the following distributions for these seven group of input data: three cards for elastic moduli of crystals, four for potentially activated slip systems, two for initial orientation of crystals, three for slipping rate dependence, six for self and latent hardening moduli, one for the time integration scheme and analysis of finite deformation, and one for the iteration method. Further details are shown in Appendix A.

In the analysis of bicrystals or composites of more than two crystals, the user must repeat the above procedures for each material. More specifically, the user must give the corresponding twenty data cards following *USER MATERIAL card, and the number of solution dependent state variables following *DEPVAR card, for each single crystal. The source code of subroutine UMAT following *USER SUBROUTINE card needs not be repeated.

5. Modification and Improvement

The modification of subroutine UMAT is fairly straightforward, as long as the distribution of input data cards mentioned above is not changed. There follows some guidelines for further modification and improvement of the subroutine UMAT:

5.1 Non-cubic crystals

Only subroutines ROTATION and SLIPSYS must be changed accordingly to include the effect of the aspect ratio of a non-cubic crystal and relative orientation of base vectors for a non-orthotropic crystal. For example, the user must realize that for an orthotropic single crystal the [110] direction may not be normal to the [-110] direction, nor normal to the (-111) plane.

5.2 Other models of slipping rates and hardening moduli

The subroutine UMAT may be modified easily to accommodate other models of single crystals. As discussed earlier, a different expression for slipping rates in slip systems other than

the power law form of eq. (2.3.1a) can be accommodated easily by changing function subprograms F and DFDX, as long as the general expression still holds. However, the subroutine STRAINRATE has to be modified if the expression of slipping rates in slip systems are more general, such as

$$\dot{\gamma}^{(\alpha)} = \dot{a}^{(\alpha)} f^{(\alpha)}\left(\tau^{(\alpha)}, g^{(\alpha)}, T\right) \tag{5.2.1}$$

where T are other internal variables, such as temperature.

Only function subprograms HSELF, HLATNT and GSLP0 (also DHSELF and DHLATN if the Newton-Rhapson iterative method is used) have to be changed accordingly for different expressions of self and latent hardening moduli other than eqs. (2.3.3) by Peirce *et al* (1982) and Asaro (1983a, b) or (2.3.4) by Bassani and Wu (1991), assuming the general incremental relation of eq. (2.3.2) holds. Otherwise the subroutine LATENTHARDEN needs further modification.

The number of state variables following *DEPVAR card must be increased accordingly if more solution dependent state variables are necessary in other models of rate sensitivity and slip hardening.

5.3 Non-Schmid effect

Asaro and Rice (1977) have discussed the modelling of single crystals which do not obey the Schmid law. The subroutine UMAT has to be modified accordingly by changing the subroutine STRAINRATE and function subprograms F and DFDX. More state variables have to be introduced in the main subroutine UMAT to accommodate this non-Schmid effect. These new state variables must be passed to subroutine STRAINRATE.

5.4 Numerical time integration scheme and the iteration method

Numerical time integration schemes other than the forward gradient time integration scheme in Section 3.1, or the Newton-Rhapson iterative method in Section 3.4, require the modification of main subroutine UMAT.

5.5 Format of the input file for subroutine UMAT

It is recommended that users not change the structure of the input file for subroutine UMAT, i.e. the structure of these twenty data cards. Otherwise users have to modify the array PROPS which stores all the input data in subroutine UMAT.

Acknowledgement

The author gratefully appreciates the suggestions and comments from Professor J.R. Rice. This work was supported under grant ONR N00014-90-J-1379 from the Office of Naval Research, Mechanics Program, to Prof. J.R. Rice at Harvard University. The ABAQUS program was kindly made available by Hibbitt, Karlsson & Sorensen, Inc. under an academic license.

References

ABAQUS Manual, (1989) version 4.8, Hibbitt, Karlsson & Sorensen, Inc., Providence, Rhode Island.

Asaro, R. J., (1983a) "Micromechanics of Crystals and Polycrystals", Adv. Appl. Mech., v. 23, p. 1.

Asaro, R. J., (1983b) "Crystal Plasticity", J. Appl. Mech., v. 50, p. 921.

Asaro, R. J., and Rice, J. R., (1977) "Strain Localization in Ductile Single Crystals", J. Mech. Phys. Solids, v. 25, p. 309.

Bassani, J. L., and Wu, T., (1991) "Latent Hardening in Single Crystals, Part II, Analytical Characterization and Predictions", Proc. Roy. Soc. Series, A, in press.

Hill, R., (1966) "Generalized Constitutive Relations for Incremental Deformation of Metal Crystals by Multislip", J. Mech. Phys. Solids, v. 14, p. 95.

Hill, R., (1967) "The Essential Structure of Constitutive Laws for Metal Composites and Polycrystals", J. Mech. Phys. Solids, v. 15, p. 79.

Hill, R., and Rice, J. R., (1972) "Constitutive Analysis of Elastic-Plastic Crystals at Arbitrary Strain", J. Mech. Phys. Solids, v. 20, p. 401.

Hughes, T.J.R., and Winget, J., (1980) "Finite Rotation Effects in Numerical Integration of Rate Constitutive Equations Arising in Large-Deformation Analysis", Int. J. Num. Methods Engng., v. 15, p. 1862.

Hull, D., and Bacon, D. J., (1984) *Introduction to Dislocations*, Pergamon Press.

Hutchinson, J. W., (1976) "Bounds and Self-Consistent Estimates for Creep of Polycrystalline Materials", Proc. Roy. Soc. Series, A, v. 348, p. 101.

Mandel, J., (1965) "Generalisation de la Theorie de Plasticite de W.T. Koiter", Int. J. Solids Struct., v. 1, p. 273.

Peirce, D., Asaro, R. J., and Needleman, A., (1982) "An Analysis of Nonuniform and Localized Deformation in Ductile Single Crystals", Acta Metall. v. 30, p.1087.

Peirce, D., Asaro, R. J., and Needleman, A., (1983) "Material Rate Dependence and Localized Deformation in Crystalline Solids", Acta Metall. v. 31, p.1951.

Peirce, D., Shih, C. F. and Needleman, A., (1984) "A Tangent Modulus Method for Rate Dependent Solids", Computers & Structures, v. 18, p. 875.

Rice, J.R., (1970) "On the Structure of Stress-Strain Relations for Time-Dependent Plastic Deformation in Metals", J. Appl. Mech., v. 37, p. 728.

Rice, J. R., (1971), "Inelastic Constitutive Relations for Solids: An Internal-Variable Theory and its Application to Metal Plasticity", J. Mech. Phys. Solids, v. 19, p. 433.

Taylor, G. I., (1938) "Plastic Strain in Metals", J. Inst. Metals, v. 62, p. 307.

Wu, T., Bassani, J. L., and Laird, C. (1991) "Latent Hardening in Single Crystals, Part I, Theory and Experiments", Proc. Roy. Soc. Series, A, in press.

Zarka, J., (1975) *Constitutive Equations in Plasticity*, ed. Argon, A. S., MIT Press, p. 359.

Appendix A. Input File for UMAT Subroutine for Single Crystals

All material parameters for subroutine UMAT follow the card *USER MATERIAL in the .INP file. There are twenty data cards, with up to eight parameters per card. ABAQUS requires all data in these cards be real variables. For example, unity should be written as 1. or 1.0. Here is the format of these input data:

A1. Elastic moduli of crystals (three data cards)

Only the first data card is needed for cubic crystals. The second and third cards are reserved for future generalization of the user-material subroutine for orthotropic and general anisotropic crystals.

The elastic moduli of crystals are given in the local cubic system, i.e. the base vectors are [100], [010] and [001] directions. For a cubic crystal there are three independent moduli c_{11} ($=L_{1111}$), c_{12} ($=L_{1122}$), c_{44} ($=L_{1212}$). The input cards must give the moduli in the order:

c_{11}, c_{12}, c_{44}	(card #1)
0.	(card #2)
0.	(card #3)

If a cubic crystal has elastic isotropy, there are only two independent elastic constants. The input cards may give elastic modulus E and Poisson's ratio ν in the order:

E, ν	(card #1)
0.	(card #2)
0.	(card #3)

(The present version of subroutine UMAT can also generate the elastic moduli matrix for orthotropic crystals and general anisotropic crystals. For an orthotropic crystal, the input cards may give elastic moduli in the order:

L_{1111}, L_{1122}, L_{2222}, L_{1133}, L_{2233}, L_{3333}, L_{1212}, L_{1313} (card #1)
L_{2323} (card #2)
0. (card #3)

For a general anisotropic crystal, the input cards may give elastic moduli in the order:

L_{1111}, L_{1122}, L_{2222}, L_{1133}, L_{2233}, L_{3333}, L_{1112}, L_{2212} (card #1)
L_{3312}, L_{1212}, L_{1113}, L_{2213}, L_{3313}, L_{1213}, L_{1313}, L_{1123} (card #2)
L_{2223}, L_{3323}, L_{1223}, L_{1323}, L_{2323} (card #3))

A2 Slip systems (four data cards)

Subroutine UMAT can generate up to three sets of slip systems for a cubic crystal. The first card in this group (card #4) gives the number of sets of potentially active slip systems, NSET, which must be less or equal to three:

NSET (card #4)

The following card (card #5) gives for the first set of slip systems a normal to slip plane (m_1, m_2, m_3) and a slip direction (s_1, s_2, s_3), e.g. (111) and [110] for FCC metal crystals, in the following order:

m_1, m_2, m_3, s_1, s_2, s_3 (card #5)

where all the data must be real variables. If there is more than one set of slip systems, the following two cards (cards #6 and #7) give normals to slip planes and slip directions in the same order as the first set of slip system. Otherwise, real number zero (0.) should be put at the beginning of the corresponding card.

A3 Initial orientation of crystals (two data cards)

The orientation of a cubic crystal is uniquely determined by giving components of two non-parallel vectors in the local cubic system and the global system. The first card in this group (card #8) gives the first vector in the order:

p_1, p_2, p_3, P_1, P_2, P_3 (card #8)

where $[p_1\ p_2\ p_3]$ and $[P_1\ P_2\ P_3]$ are the components of first vector in the local cubic system and global system, respectively. The magnitudes of **p** and **P** need not be identical. The second card in this group (card #9) gives the second vector in the same order.

A4 Slipping rate dependence (three data cards)

Three cards in this group provide the modelling parameters for slipping rate dependence on resolved shear stress and current strength (see eq. (2.3.1) in the paper). The power law (2.3.1a) is used in subroutine UMAT, while the rate sensitivity exponent n and reference strain rate $\dot{a}$ are assumed the same as others within the same set of crystallographically identical slip systems, although it could be different from those in other sets.

The first card in this group (card #10) gives the rate sensitivity exponent n and reference strain rate $\dot{a}$ for the first set of slip systems in the order:

n, $\dot{a}$ (card #10)

If there is more than one set of slip systems, the following two cards (cards #11 and #12) follow the same order. Otherwise, real number zero (0.) should be put at the beginning of the corresponding card.

A5 Self and latent hardening moduli (six data cards)

Six cards in this group provide data for models of self and latent hardening moduli (see eq. (2.3.2)). Peirce *et al* (1982) and Asaro's (1983a, b) hardening law (2.3.3), or Bassani and Wu's (1991) formulation (2.3.4) are used in subroutine UMAT. All parameters in these slip hardening models are assumed the same as others within the same set of crystallographically identical slip systems, although it could be different from those in other sets.

The input format for these two formulations (2.3.3) and (2.3.4) is different. The format of input data controls which formulation is used in subroutine UMAT. The first two cards in this group (cards #13 and #14) provide input data for the first set of slip systems. The remaining four cards (cards #15-#18) are reserved for the other two slip systems (if there are).

For Peirce *et al* (1982) and Asaro's (1983a, b) hardening law (2.3.3), the first card in this group (cards #13) gives the initial hardening modulus h_0, stage I stress τ_s and initial yield stress τ_0 in the order:

h_0, τ_s, τ_0 (card #13)

The second card in this group (card #14) provides latent hardening parameters q, as defined in eq. (2.3.3c):

q, q_1 (card #14)

where q is the ratio of latent over self hardening moduli within the same set of slip systems, and q_1 is the ratio for slip systems in other sets (if there are).

For Bassani and Wu's (1991) law (2.3.4), the first card (card #13) provides self hardening parameters in the order:

h_0, τ_s, τ_0, h_s, γ_0, γ_1, f_0, f_1 (card #13)

where h_0 is the initial hardening modulus, τ_s is the stage I stress, τ_0 is the initial yield stress, h_s is the hardening modulus during easy glide within stage I hardening, γ_0 and f_0 are parameters of slip interaction (defined in eq. (2.3.4c)) within same set of slip systems, and γ_1 and f_1 are the corresponding values between different sets of slip systems.

The format of the second card in this group (card #14) for Bassani and Wu's model (1991) has exactly the same order for latent hardening parameters q and q_1 as that for Peirce *et al* (1982) and Asaro's (1983a, b) above.

A6 Other parameters (one data card)

This card (card #19) provides additional parameters (no more than eight) beyond the five groups above. This card gives the forward gradient time integration parameter θ in Section 3.1 and parameter NLGEOM. Any non-zero (but real) values of NLGEOM will initiate the finite strain and finite rotation computation within subroutine UMAT. Here is format of the card:

θ, NLGEOM (card #19)

A7 Parameters for the iteration method (one data card)

This card (card #20) provides the parameters for the iteration, parameter ITRATN, maximum number of iteration ITRMAX and the absolute error of shear strains in slip systems γ_{err}. Any non-zero (but real) values of ITRATN will initiate the iteration process. If the iterative method does not lead to a convergent solution within ITRMAX step of iteration, the non-iterative solution

(3.2.5) is used instead. The parameter γ_{err} is the tolerance of absolute error of the shear stains. Here is the format of the card:

ITPATN, ITRMAX, γ_{err} (card #20)

Appendix B Output for the Solution Dependent State Variables

In subroutine UMAT the shear strains $\gamma^{(\alpha)}$, as well as other solution dependent state variables are stored in the array STATEV in the order:

STATEV(1)	- STATEV(NSLPTL):	current strengths $g^{(\alpha)}$
STATEV(NSLPTL+1)	- STATEV(2*NSLPTL):	shear strains $\gamma^{(\alpha)}$
STATEV(2*NSLPTL+1)	- STATEV(3*NSLPTL):	resolved shear stresses $\tau^{(\alpha)}$
STATEV(3*NSLPTL+1)	- STATEV(6*NSLPTL):	normals to slip planes $\mathbf{m}^{*(\alpha)}$
STATEV(6*NSLPTL+1)	- STATEV(9*NSLPTL):	slip directions $\mathbf{s}^{*(\alpha)}$
STATEV(9*NSLPTL+1)	:	total cumulative shear strain γ

where NSLPTL is the total number of slip systems in all sets. For example, NSLPTL is 12 for {111}<110> slip systems in FCC metal crystals, and NSLPTL is 48 for {110}<111>, {121}<111> and {123}<111> slip systems in BCC metal crystals.

The format of the output for solution dependent state variables in ABAQUS is

SDV for all solution dependent state variables

or

SDVn for the solution dependent state variable n

For example, the format of the output for shear strains $\gamma^{(\alpha)}$ and the total cumulative shear strain γ on all slip systems in an FCC metal crystal with {111}<110> slip system is

SDV13,SDV14,SDV15,SDV16,SDV17,SDV18,SDV19,SDV20
SDV21,SDV22,SDV23,SDV24,SDV109

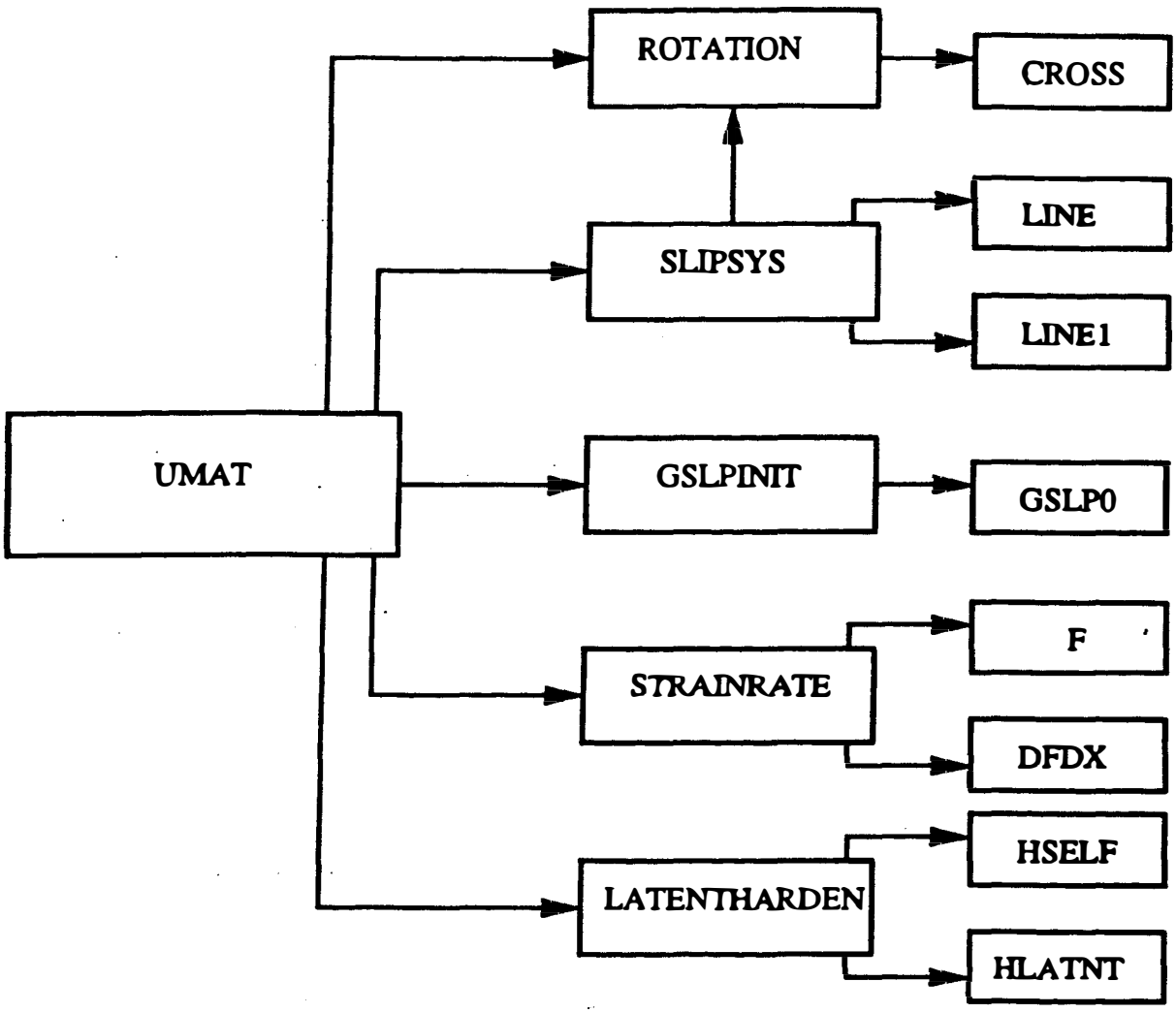
(ABAQUS allows no more than eight data per card.) The order of corresponding slip systems (generated by subroutine SLIPSYS) may be found in the .DAT file by searching for the symbol "#".

Appendix C An Example of an Input File

An example of an input file including the source code of subroutine UMAT is given in this appendix for a copper single crystal bar subject to uniaxial tension. The bar is 100mm long, with a square cross section 10mm * 10mm. The two edges of the square cross section coincide with the crystal [010] and [-101] directions, respectively.

Copper has a FCC structure, with elastic moduli c_{11}=168400MPa, c_{12}=121400MPa, and c_{44}=75400MPa. There is one set of slip systems, {111}<110>. The rate sensitivity exponent n and reference strain rate $\dot{a}$ are takén to be 10 and 0.001 sec^{-1}, respectively. Peirce *et al* (1982) and Asaro's (1983a, b) hardening law (2.3.3) is used, with their recommended values of initial hardening modulus h_0=541.5MPa, stage I stress τ_s=109.5MPa and initial yield stress τ_0=60.8MPa, which are obtained by fitting the experimental data for a copper single crystal. The ratio of latent over self hardening moduli, q, is taken to be unity (Taylor's hardening). The forward gradient time integration parameter θ is 0.5.

The copper single crystal bar is subject to uniaxial tensile stress 200MPa along the axial direction. The effect of finite strain and finite rotation is included by setting parameter NLGEOM=1.0. The Newton-Rhapson iterative method is used (ITRATN=1.0), with maximum number of iteration ITRMAX=10.0 and the absolute error of shear strains $\gamma_{err}=10^{-5}$.



UMAT --- main subroutine

ROTATION --- orientation of local cubic system in global system
CORSS --- cross product of two vectors

SLIPSYS --- generating all slip systems
LINE --- [mmm] type of slip systems
LINE1 --- [0mn] type of slip systems

GSLPINIT --- initial values of current strain hardening functions in all slip systems
GSLP0 --- USER-supplied functional subroutine for the initial value in each system

STRAINRATE --- shear strain-rates in all slip systems
F --- USER-supplied functional subroutines for the shear strain-rate in each system
DFDX --- USER-supplied functional subroutine for the derivative of function F

LATENTHARDEN -- hardening matrix, i.e. self- and latent-hardening in all slip systems
HSELF -- USER-supplied functional subroutine for the self-hardening modulus
HLATNT -- USER-supplied functional subroutine for the latent-hardening modulus

Figure 1

```
** One-element  Test:
**   (via *UMAT procedure)
**
** Model is intended to represent a single crystal metallic bar
**   subjected to uniaxial tension
**
**  This program is based on the "finite strain" version of the
**    constitutive law of a single crystal metal following the Schmid
**    rule, with various hardening options.  It involves a single
**    element.
**
**
*HEADING
Single-Crystal One Element Model; Finite Strain and Finite Rotation
**
**  lengths in mm, stress and moduli in MPa
**
*NODE, NSET=NODEALL
   1,    0.,     0.,     0.
   2,    0.,     10.,    0.
   3,    0.,     10.,    10.
   4,    0.,     0.,     10.
   5,    100.,   0.,     0.
   6,    100.,   10.,    0.
   7,    100.,   10.,    10.
   8,    100.,   0.,     10.
   9,    0.,     5.,     0.
  10,    0.,     10.,    5.
  11,    0.,     5.,     10.
  12,    0.,     0.,     5.
  13,    100.,   5.,     0.
  14,    100.,   10.,    5.
  15,    100.,   5.,     10.
  16,    100.,   0.,     5.
  17,    50.,    0.,     0.
  18,    50.,    10.,    0.
  19,    50.,    10.,    10.
  20,    50.,    0.,     10.
**
*ELEMENT, TYPE=C3D20R
  1,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
 16, 17, 18, 19, 20
*ELSET,ELSET=ONE
1
**
*BOUNDARY
1,PINNED
2,1
2,3
3,1
4,1
9,1
10,1
11,1
12,1
**
**
*SOLID SECTION, ELSET=ONE, MATERIAL=CRYSTAL
*MATERIAL,NAME=CRYSTAL
**
*USER MATERIAL,CONSTANTS=160,UNSYMM
**
** All the constants below must be real numbers!
**
   168400., 121400., 75400. ,
**    c11  ,   c12  ,   c44  , (elastic constants of copper crystal)
**    MPa  ,   MPa  ,   MPa  ,
**
      0.   ,
** constants only used for an elastic orthotropic or anisotropic material
**    MPa  ,
**
      0.   ,
** constants only used for an elastic anisotropic material
**    MPa  ,
**
** The elastic constants above are relative to crystal axes, where
**   1 -- [100],  2 -- [010],  3 -- [001] .  These elastic constants
**   are arranged in the following order:
**   eight constants each line (data card)
**
** (1) isotropic:
**      E    ,  Nu       (Young's modulus and Poisson's ratio)
**      0.
**      0.
**
** (2) cubic:
**      c11  ,  c12  ,  c44
**      0.
**      0.
**
** (3) orthotropic:
**      D1111,  D1122,  D2222,  D1133,  D2233,  D3333,  D1212,  D1313,
**      D2233
**      0.
**
** (4) anisotropic:
**      D1111,  D1122,  D2222,  D1133,  D2233,  D3333,  D1112,  D2212,
**      D3312,  D1212,  D1113,  D2213,  D3313,  D1213,  D1313,  D1123,
**      D2223,  D3323,  D1223,  D1323,  D2323
**
**
      1.   ,
** number of sets of slip systems
**    --   ,
**
      1.   ,   1.   ,   1.   ,   1.   ,   1.   ,   0.   ,
**    normal to slip plane   ,       slip direction      , of the 1st set
**    --   ,   --   ,   --   ,   --   ,   --   ,   --   ,
**
      0.   ,
**    normal to slip plane   ,       slip direction      , of the 2nd set
**    --   ,   --   ,   --   ,   --   ,   --   ,   --   ,
**
      0.   ,
**    normal to slip plane   ,       slip direction      , of the 3rd set
**    --   ,
**
**
     -1.   ,   0.   ,   1.   ,   0.   ,   0.   ,   1.   ,
** direction in local system ,       global system       , of the 1st vector
**    ---  ,   --   ,   --   ,   --   ,   --   ,   --   ,
** (the first vector to determine crystal orientation in global system)
**
      0.   ,   1.   ,   0.   ,   0.   ,   1.   ,   0.   ,
** direction in local system ,       global system       , of the 2nd vector
**    --   ,   --   ,   --   ,   --   ,   --   ,   --   ,
** (the second vector to determine crystal orientation in global system)
```

```
**
** constraint:  The angle between two non-parallel vectors in the local
**              and global systems should be the same.  The relative
**              difference must be less than 0.1%.
**
**
      10.  ,  .001  ,
**     n    ,  adot  , of 1st set of slip systems
**    ---   ,  1/sec ,
** (power hardening exponent and hardening coefficient)
**  gammadot = adot * ( tau / g ) ** n
**
** Users who want to use their own constitutive relation may change the
**   function subprograms F and DFDX called by the subroutine
**   STRAINRATE and provide the necessary data (no more than 8) in the
**   above line (data card).
**
**
      0.   ,   0.
**    n    ,  adot  , of 2nd set of slip systems
**    ---  ,  1/sec ,
**
      0.   ,   0.   ,
**    n    ,  adot  , of 3rd set of slip systems
**    --   ,  1/sec ,
**
**
     541.5 ,  109.5 ,  60.8  ,
**    h0   ,  taus  ,  tau0  , of 1st set of slip systems
**    MPa  ,   MPa  ,   MPa  ,
** (initial hardening modulus, saturation stress and initial critical
**  resolved shear stress)
**  H = H0 * ( sech | H0 * gamma / (taus - tau0 ) | ) ** 2
**
** Users who want to use their own self-hardening law may change the
**   function subprogram HSELF called by the subroutine LATENTHARDEN
**   and provide the necessary data (no more than 8) in the above line
**   (data card).
**
**
      1.   ,   1.   ,
**    q    ,   ql   , latent hardening of 1st set of slip systems
**    --   ,   --   ,
** (ratios of latent to self-hardening in the same and different sets
**  of slip systems)
**
** Users who want to use their own latent-hardening may change the
**   function subprogram HLATNT called by the subroutine LATENTHARDEN
**   and provide the additional data (beyond the self-hardening data,
**   no more than 8) in the above line (data card).
**
**
      0.   ,
**    h0   ,  taus  ,  tau0  , of 2nd set of slip systems
**    MPa  ,   MPa  ,   Mpa  ,
**
      0.   ,
**    q    ,   ql   , of 2nd set of slip systems
**    --   ,   --   ,
**
      0.   ,
**    h0   ,  taus  ,  tau0  , of 3rd set of slip systems
**    MPa  ,   MPa  ,   MPa  ,
**
      0.   ,
**    q    ,   ql   , of 3rd set of slip systems
**    --   ,   --   ,
**
**
      .5   ,   1.   ,
**   THETA ,  NLGEOM ,
**    --   ,   --   ,
**
** THETA:  implicit integration parameter, between 0 and 1
**
** NLGEOM:  parameter determining whether finite deformation of single
**   crystal is considered
**
**   NLGEOM=0. --- small deformation
**   otherwise --- finite rotation and finite strain,  Users must
**                 declare "NLGEOM" in the input file, at the *STEP
**                 card
**
**
      1.   ,   10.  , 1.E-5  ,
**  ITRATN , ITRMAX , GAMERR ,
**    --   ,   --   ,   --   ,
** ITRATN:  parameter determining whether iteration method is used to
**   solve increments of stresses and state variables in terms of
**   strain increments
**
**   ITRATN=0. --- no iteration
**   otherwise --- iteration
**
** ITRMAX:  maximum number of iterations
**
** GAMERR:  absolute error of shear strains in slip systems
**
**
*DEPVAR
113
** number of state dependent variables, must be larger than (or equal
**   to) nine times total number of slip systems in all sets, plus
**   five, plus the additional number of state variables users
**   introduced for their own single crystal model
**
** For example, {110}<111> has twelve slip systems.  There are
**   12*9+5=113 state dependent variables.
**
**
**
*USER SUBROUTINE
**
      SUBROUTINE UMAT (STRESS, STATEV, DDSDDE, SSE, SPD, SCD, RPL,
     *                 DDSDDT, DRPLDE, DRPLDT, STRAN, DSTRAN, TIME,
     *                 DTIME, TEMP, DTEMP, PREDEF, DPRED, CMNAME, NDI,
     *                 NSHR, NTENS, NSTATV, PROPS, NPROPS, COORDS,
     *                 DROT)

C-----  Use single precision on Cray by
C      (1) deleting the statement "IMPLICIT*8 (A-H,O-Z)";
C      (2) changing "REAL*8 FUNCTION" to "FUNCTION";
C      (3) changing double precision functions DSIGN to SIGN.

C-----  Subroutines:
C
C       ROTATION     -- forming rotation matrix, i.e. the direction
C                       cosines of cubic crystal [100], [010] and [001]
```

```
C                     directions in global system at the initial
C                     state
C
C       SLIPSYS       -- calculating number of slip systems, unit
C                        vectors in slip directions and unit normals to
C                        slip planes in a cubic crystal at the initial
C                        state
C
C       GSLPINIT      -- calculating initial value of current strengths
C                        at initial state
C
C       STRAINRATE    -- based on current values of resolved shear
C                        stresses and current strength, calculating
C                        shear strain-rates in slip systems
C
C       LATENTHARDEN -- forming self- and latent-hardening matrix
C
C       ITERATION     -- generating arrays for the Newton-Rhapson
C                        iteration
C
C       LUDCMP        -- LU decomposition
C
C       LUBKSB        -- linear equation solver based on LU
C                        decomposition method (must call LUDCMP first)


C-----  Function subprogram:

C       F -- shear strain-rates in slip systems


C-----  Variables:
C
C       STRESS -- stresses (INPUT & OUTPUT)
C                 Cauchy stresses for finite deformation
C       STATEV -- solution dependent state variables (INPUT & OUTPUT)
C       DDSDDE -- Jacobian matrix (OUTPUT)

C-----  Variables passed in for information:
C
C       STRAN  -- strains
C                 logarithmic strain for finite deformation
C                 (actually, integral of the symmetric part of velocity
C                  gradient with respect to time)
C       DSTRAN -- increments of strains
C       CMNAME -- name given in the *MATERIAL option
C       NDI    -- number of direct stress components
C       NSHR   -- number of engineering shear stress components
C       NTENS  -- NDI+NSHR
C       NSTATV -- number of solution dependent state variables (as
C                 defined in the *DEPVAR option)
C       PROPS  -- material constants entered in the *USER MATERIAL
C                 option
C       NPROPS -- number of material constants
C

C-----  This subroutine provides the plastic constitutive relation of
C     single crystals for finite element code ABAQUS.  The plastic slip
C     of single crystal obeys the Schmid law.  The program gives the
C     choice of small deformation theory and theory of finite rotation
C     and finite strain.
C       The strain increment is composed of elastic part and plastic
C     part.  The elastic strain increment corresponds to lattice
C     stretching, the plastic part is the sum over all slip systems of
C     plastic slip.  The shear strain increment for each slip system is
C     assumed a function of the ratio of corresponding resolved shear
C     stress over current strength, and of the time step.  The resolved
C     shear stress is the double product of stress tensor with the slip
C     deformation tensor (Schmid factor), and the increment of current
C     strength is related to shear strain increments over all slip
C     systems through self- and latent-hardening functions.

C-----  The implicit integration method proposed by Peirce, Shih and
C     Needleman (1984) is used here.  The subroutine provides an option
C     of iteration to solve stresses and solution dependent state
C     variables within each increment.

C-----  The present program is for a single CUBIC crystal.  However,
C     this code can be generalized for other crystals (e.g. HCP,
C     Tetragonal, Orthotropic, etc.).  Only subroutines ROTATION and
C     SLIPSYS need to be modified to include the effect of crystal
C     aspect ratio.
C

C-----  Important notice:
C
C     (1) The number of state variables NSTATV must be larger than (or
C         equal to) NINE (9) times the total number of slip systems in
C         all sets, NSLPTL, plus FIVE (5)
C              NSTATV >= 9 * NSLPTL + 5
C         Denote s as a slip direction and m as normal to a slip plane.
C         Here (s,-m), (-s,m) and (-s,-m) are NOT considered
C         independent of (s,m).  The number of slip systems in each set
C         could be either 6, 12, 24 or 48 for a cubic crystal, e.g. 12
C         for {110}<111>.
C
C         Users who need more parameters to characterize the
C         constitutive law of single crystal, e.g. the framework
C         proposed by Zarke, should make NSTATV larger than (or equal
C         to) the number of those parameters NPARMT plus nine times
C         the total number of slip systems, NSLPTL, plus five
C              NSTATV >= NPARMT + 9 * NSLPTL + 5
C
C     (2) The tangent stiffness matrix in general is not symmetric if
C         latent hardening is considered.  Users must declare "UNSYMM"
C         in the input file, at the *USER MATERIAL card.
C


C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (ND=150)
C-----  The parameter ND determines the dimensions of the arrays in
C     this subroutine.  The current choice 150 is a upper bound for a
C     cubic crystal with up to three sets of slip systems activated.
C     Users may reduce the parameter ND to any number as long as larger
C     than or equal to the total number of slip systems in all sets.
C     For example, if {110}<111> is the only set of slip system
C     potentially activated, ND could be taken as twelve (12).

      CHARACTER*8 CMNAME
      EXTERNAL F
      DIMENSION STRESS(NTENS), STATEV(NSTATV), DDSDDE(NTENS,NTENS),
     2          DDSDDT(NTENS), DRPLDE(NTENS), STRAN(NTENS),
     3          DSTRAN(NTENS), PREDEF(1), DPRED(1), PROPS(NPROPS),
     4          COORDS(3), DROT(3,3)
```

```
      DIMENSION ISPDIR(3), ISPNOR(3), NSLIP(3),
     2          SLPDIR(3,ND), SLPNOR(3,ND), SLPDEF(6,ND),
     3          SLPSPN(3,ND), DSPDIR(3,ND), DSPNOR(3,ND),
     4          DLOCAL(6,6), D(6,6), ROTD(6,6), ROTATE(3,3),
     5          FSLIP(ND), DFDXSP(ND), DDEMSD(6,ND),
     6          H(ND,ND), DDGDDE(ND,6),
     7          DSTRES(6), DELATS(6), DSPIN(3), DVGRAD(3,3),
     8          DGAMMA(ND), DTAUSP(ND), DGSLIP(ND),
     9          WORKST(ND,ND), INDX(ND), TERM(3,3), TRMO(3,3), ITRM(3)

      DIMENSION FSLIP1(ND), STRES1(6), GAMMA1(ND), TAUSP1(ND),
     2          GSLP1(ND), SPNOR1(3,ND), SPDIR1(3,ND), DDSDE1(6,6),
     3          DSOLD(6), DGAMOD(ND), DTAUOD(ND), DGSPOD(ND),
     4          DSPNRO(3,ND), DSPDRO(3,ND),
     5          DHDGDG(ND,ND)


C-----  NSLIP  -- number of slip systems in each set
C-----  SLPDIR -- slip directions (unit vectors in the initial state)
C-----  SLPNOR -- normals to slip planes (unit normals in the initial
C                 state)
C-----  SLPDEF -- slip deformation tensors (Schmid factors)
C                 SLPDEF(1,i) -- SLPDIR(1,i)*SLPNOR(1,i)
C                 SLPDEF(2,i) -- SLPDIR(2,i)*SLPNOR(2,i)
C                 SLPDEF(3,i) -- SLPDIR(3,i)*SLPNOR(3,i)
C                 SLPDEF(4,i) -- SLPDIR(1,i)*SLPNOR(2,i)+
C                                SLPDIR(2,i)*SLPNOR(1,i)
C                 SLPDEF(5,i) -- SLPDIR(1,i)*SLPNOR(3,i)+
C                                SLPDIR(3,i)*SLPNOR(1,i)
C                 SLPDEF(6,i) -- SLPDIR(2,i)*SLPNOR(3,i)+
C                                SLPDIR(3,i)*SLPNOR(2,i)
C                 where index i corresponds to the ith slip system
C-----  SLPSPN -- slip spin tensors (only needed for finite rotation)
C                 SLPSPN(1,i) -- [SLPDIR(1,i)*SLPNOR(2,i)-
C                                 SLPDIR(2,i)*SLPNOR(1,i)]/2
C                 SLPSPN(2,i) -- [SLPDIR(3,i)*SLPNOR(1,i)-
C                                 SLPDIR(1,i)*SLPNOR(3,i)]/2
C                 SLPSPN(3,i) -- [SLPDIR(2,i)*SLPNOR(3,i)-
C                                 SLPDIR(3,i)*SLPNOR(2,i)]/2
C                 where index i corresponds to the ith slip system
C-----  DSPDIR -- increments of slip directions
C-----  DSPNOR -- increments of normals to slip planes
C
C-----  DLOCAL -- elastic matrix in local cubic crystal system
C-----  D      -- elastic matrix in global system
C-----  ROTD   -- rotation matrix transforming DLOCAL to D
C
C-----  ROTATE -- rotation matrix, direction cosines of [100], [010]
C                 and [001] of cubic crystal in global system
C
C-----  FSLIP  -- shear strain-rates in slip systems
C-----  DFDXSP -- derivatives of FSLIP w.r.t x=TAUSLP/GSLIP, where
C                 TAUSLP is the resolved shear stress and GSLIP is the
C                 current strength
C
C-----  DDEMSD -- double dot product of the elastic moduli tensor with
C                 the slip deformation tensor plus, only for finite
C                 rotation, the dot product of slip spin tensor with
C                 the stress
C
C-----  H      -- self- and latent-hardening matrix
C                 H(i,i) -- self hardening modulus of the ith slip
C                           system (no sum over i)
C                 H(i,j) -- latent hardening molulus of the ith slip
C                           system due to a slip in the jth slip system
C                           (i not equal j)
C
C-----  DDGDDE -- derivatice of the shear strain increments in slip
C                 systems w.r.t. the increment of strains
C
C-----  DSTRES -- Jaumann increments of stresses, i.e. corotational
C                 stress-increments formed on axes spinning with the
C                 material
C-----  DELATS -- strain-increments associated with lattice stretching
C                 DELATS(1) - DELATS(3) -- normal strain increments
C                 DELATS(4) - DELATS(6) -- engineering shear strain
C                                          increments
C-----  DSPIN  -- spin-increments associated with the material element
C                 DSPIN(1) -- component 12 of the spin tensor
C                 DSPIN(2) -- component 31 of the spin tensor
C                 DSPIN(3) -- component 23 of the spin tensor
C
C-----  DVGRAD -- increments of deformation gradient in the current
C                 state, i.e. velocity gradient times the increment of
C                 time
C
C-----  DGAMMA -- increment of shear strains in slip systems
C-----  DTAUSP -- increment of resolved shear stresses in slip systems
C-----  DGSLIP -- increment of current strengths in slip systems
C
C
C-----  Arrays for iteration:
C
C           FSLIP1, STRES1, GAMMA1, TAUSP1, GSLP1 , SPNOR1, SPDIR1,
C           DDSDE1, DSOLD , DGAMOD, DTAUOD, DGSPOD, DSPNRO, DSPDRO,
C           DHDGDG
C
C
C-----  Solution dependent state variable STATEV:
C           Denote the number of total slip systems by NSLPTL, which
C           will be calculated in this code.
C
C       Array STATEV:
C       1          - NSLPTL     : current strength in slip systems
C       NSLPTL+1   - 2*NSLPTL   : shear strain in slip systems
C       2*NSLPTL+1 - 3*NSLPTL   : resolved shear stress in slip systems
C
C       3*NSLPTL+1 - 6*NSLPTL   : current components of normals to slip
C                                 slip planes
C       6*NSLPTL+1 - 9*NSLPTL   : current components of slip directions
C
C       9*NSLPTL+1              : total cumulative shear strain on all
C                                 slip systems (sum of the absolute
C                                 values of shear strains in all slip
C                                 systems)
C
C       9*NSLPTL+2 - NSTATV-4   : additional parameters users may need
C                                 to characterize the constitutive law
C                                 of a single crystal (if there are
C                                 any).
C
C       NSTATV-3                : number of slip systems in the 1st set
C       NSTATV-2                : number of slip systems in the 2nd set
C       NSTATV-1                : number of slip systems in the 3rd set
C       NSTATV                  : total number of slip systems in all
C                                 sets
C
C
```

```
C-----  Material constants PROPS:
C
C       PROPS(1) - PROPS(21) -- elastic constants for a general elastic
C                               anisotropic material
C
C            isotropic   : PROPS(1)=0  for  1>2
C                          PROPS(1) -- Young's modulus
C                          PROPS(2) -- Poisson's ratio
C
C            cubic       : PROPS(1)=0  for 1>3
C                          PROPS(1) -- c11
C                          PROPS(2) -- c12
C                          PROPS(3) -- c44
C
C            orthotropic : PORPS(1)=0  for  1>9
C                          PROPS(1) - PROPS(9) are D1111, D1122, D2222,
C                          D1133, D2233, D3333, D1212, D1313, D2323,
C                          respectively, which has the same definition
C                          as ABAQUS for orthotropic materials
C                          (see *ELASTIC card)
C
C            anisotropic : PROPS(1) - PROPS(21) are D1111, D1122,
C                          D2222, D1133, D2233, D3333, D1112, D2212,
C                          D3312, D1212, D1113, D2213, D3313, D1213,
C                          D1313, D1123, D2223, D3323, D1223, D1323,
C                          D2323, respectively, which has the same
C                          definition as ABAQUS for anisotropic
C                          materials (see *ELASTIC card)
C
C
C       PROPS(25) - PROPS(56) -- parameters characterizing all slip
C                                systems to be activated in a cubic
C                                crystal
C
C            PROPS(25) -- number of sets of slip systems (maximum 3),
C                         e.g. (110)[1-11] and (101)[11-1] are in the
C                         same set of slip systems, (110)[1-11] and
C                         (121)[1-11] belong to different sets of slip
C                         systems
C                         (It must be a real number, e.g. 3., not 3 !)
C
C            PROPS(33) - PROPS(35) -- normal to a typical slip plane in
C                                     the first set of slip systems,
C                                     e.g. (1 1 0)
C                                     (They must be real numbers, e.g.
C                                      1. 1. 0., not 1 1 0 !)
C            PROPS(36) - PROPS(38) -- a typical slip direction in the
C                                     first set of slip systems, e.g.
C                                     [1 1 1]
C                                     (They must be real numbers, e.g.
C                                      1. 1. 1., not 1 1 1 !)
C
C            PROPS(41) - PROPS(43) -- normal to a typical slip plane in
C                                     the second set of slip systems
C                                     (real numbers)
C            PROPS(44) - PROPS(46) -- a typical slip direction in the
C                                     second set of slip systems
C                                     (real numbers)
C
C            PROPS(49) - PROPS(51) -- normal to a typical slip plane in
C                                     the third set of slip systems
C                                     (real numbers)
C            PROPS(52) - PROPS(54) -- a typical slip direction in the
C                                     third set of slip systems
C                                     (real numbers)
C
C
C       PROPS(57) - PROPS(72) -- parameters characterizing the initial
C                                orientation of a single crystal in
C                                global system
C            The directions in global system and directions in local
C            cubic crystal system of two nonparallel vectors are needed
C            to determine the crystal orientation.
C
C            PROPS(57) - PROPS(59) -- [p1 p2 p3), direction of first
C                                     vector in local cubic crystal
C                                     system, e.g. [1 1 0]
C                                     (They must be real numbers, e.g.
C                                      1. 1. 0., not 1 1 0 !)
C            PROPS(60) - PROPS(62) -- [P1 P2 P3), direction of first
C                                     vector in global system, e.g.
C                                     [2. 1. 0.]
C                                     (It does not have to be a unit
C                                      vector)
C
C            PROPS(65) - PROPS(67) -- direction of second vector in
C                                     local cubic crystal system (real
C                                     numbers)
C            PROPS(68) - PROPS(70) -- direction of second vector in
C                                     global system
C
C
C       PROPS(73) - PROPS(96) -- parameters characterizing the visco-
C                                plastic constitutive law (shear
C                                strain-rate vs. resolved shear
C                                stress), e.g. a power-law relation
C
C            PROPS(73) - PROPS(80) -- parameters for the first set of
C                                     slip systems
C            PROPS(81) - PROPS(88) -- parameters for the second set of
C                                     slip systems
C            PROPS(89) - PROPS(96) -- parameters for the third set of
C                                     slip systems
C
C
C       PROPS(97) - PROPS(144)-- parameters characterizing the self-
C                                and latent-hardening laws of slip
C                                systems
C
C            PROPS(97) - PROPS(104)-- self-hardening parameters for the
C                                     first set of slip systems
C            PROPS(105)- PROPS(112)-- latent-hardening parameters for
C                                     the first set of slip systems and
C                                     interaction with other sets of
C                                     slip systems
C
C            PROPS(113)- PROPS(120)-- self-hardening parameters for the
C                                     second set of slip systems
C            PROPS(121)- PROPS(128)-- latent-hardening parameters for
C                                     the second set of slip systems
C                                     and interaction with other sets
C                                     of slip systems
C
C            PROPS(129)- PROPS(136)-- self-hardening parameters for the
C                                     third set of slip systems
C            PROPS(137)- PROPS(144)-- latent-hardening parameters for
C                                     the third set of slip systems and
C                                     interaction with other sets of
```

```
C                               slip systems
C
C
C       PROPS(145)- PROPS(152)-- parameters characterizing forward time
C                               integration scheme and finite
C                               deformation
C
C            PROPS(145) -- parameter theta controlling the implicit
C                          integration, which is between 0 and 1
C                          0.  : explicit integration
C                          0.5 : recommended value
C                          1.  : fully implicit integration
C
C            PROPS(146) -- parameter NLGEOM controlling whether the
C                          effect of finite rotation and finite strain
C                          of crystal is considered,
C                          0.          : small deformation theory
C                          otherwise : theory of finite rotation and
C                                      finite strain
C
C
C       PROPS(153)- PROPS(160)-- parameters characterizing iteration
C                               method
C
C            PROPS(153) -- parameter ITRATN controlling whether the
C                          iteration method is used,
C                          0.          : no iteration
C                          otherwise : iteration
C
C            PROPS(154) -- maximum number of iteration ITRMAX
C
C            PROPS(155) -- absolute error of shear strains in slip
C                          systems GAMERR
C


C-----  Elastic matrix in local cubic crystal system: DLOCAL
      DO J=1,6
         DO I=1,6
            DLOCAL(I,J)=0.
         END DO
      END DO

      CHECK=0.
      DO J=10,21
         CHECK=CHECK+ABS(PROPS(J))
      END DO

      IF (CHECK.EQ.0.) THEN
         DO J=4,9
            CHECK=CHECK+ABS(PROPS(J))
         END DO

         IF (CHECK.EQ.0.) THEN

            IF (PROPS(3).EQ.0.) THEN

C-----  Isotropic material
               GSHEAR=PROPS(1)/2./(1.+PROPS(2))
               E11=2.*GSHEAR*(1.-PROPS(2))/(1.-2.*PROPS(2))
               E12=2.*GSHEAR*PROPS(2)/(1.-2.*PROPS(2))

               DO J=1,3
                  DLOCAL(J,J)=E11
                  DO I=1,3
                     IF (I.NE.J) DLOCAL(I,J)=E12
                  END DO

                  DLOCAL(J+3,J+3)=GSHEAR
               END DO

            ELSE

C-----  Cubic material
               DO J=1,3
                  DLOCAL(J,J)=PROPS(1)

                  DO I=1,3
                     IF (I.NE.J) DLOCAL(I,J)=PROPS(2)
                  END DO

                  DLOCAL(J+3,J+3)=PROPS(3)
               END DO

            END IF

         ELSE

C-----  Orthotropic metarial
            DLOCAL(1,1)=PROPS(1)
            DLOCAL(1,2)=PROPS(2)
            DLOCAL(2,1)=PROPS(2)
            DLOCAL(2,2)=PROPS(3)

            DLOCAL(1,3)=PROPS(4)
            DLOCAL(3,1)=PROPS(4)
            DLOCAL(2,3)=PROPS(5)
            DLOCAL(3,2)=PROPS(5)
            DLOCAL(3,3)=PROPS(6)

            DLOCAL(4,4)=PROPS(7)
            DLOCAL(5,5)=PROPS(8)
            DLOCAL(6,6)=PROPS(9)

         END IF

      ELSE

C-----  General anisotropic material
         ID=0
         DO J=1,6
            DO I=1,J
               ID=ID+1
               DLOCAL(I,J)=PROPS(ID)
               DLOCAL(J,I)=DLOCAL(I,J)
            END DO
         END DO
      END IF

C-----  Rotation matrix: ROTATE, i.e. direction cosines of [100], [010]
C     and [001) of a cubic crystal in global system
C
      CALL ROTATION (PROPS(57), ROTATE)

C-----  Rotation matrix: ROTD to transform local elastic matrix DLOCAL
C     to global elastic matrix D
C
```

```
      DO J=1,3
         J1=1+J/3
         J2=2+J/2

         DO I=1,3
            I1=1+I/3
            I2=2+I/2

            ROTD(I,J)=ROTATE(I,J)**2
            ROTD(I,J+3)=2.*ROTATE(I,J1)*ROTATE(I,J2)
            ROTD(I+3,J)=ROTATE(I1,J)*ROTATE(I2,J)
            ROTD(I+3,J+3)=ROTATE(I1,J1)*ROTATE(I2,J2)+
     2                    ROTATE(I1,J2)*ROTATE(I2,J1)

         END DO
      END DO

C-----  Elastic matrix in global system: D
C     (D) = (ROTD) * (DLOCAL) * (ROTD)transpose
C
      DO J=1,6
         DO I=1,6
            D(I,J)=0.
         END DO
      END DO

      DO J=1,6
         DO I=1,J

            DO K=1,6
               DO L=1,6
                  D(I,J)=D(I,J)+DLOCAL(K,L)*ROTD(I,K)*ROTD(J,L)
               END DO
            END DO

            D(J,I)=D(I,J)

         END DO
      END DO

C-----  Total number of sets of slip systems: NSET
      NSET=NINT(PROPS(25))
      IF (NSET.LT.1) THEN
         WRITE (6,*) '***ERROR - zero sets of slip systems'
         STOP
      ELSE IF (NSET.GT.3) THEN
         WRITE (6,*)
     2     '***ERROR - more than three sets of slip systems'
         STOP
      END IF

C-----  Implicit integration parameter: THETA
      THETA=PROPS(145)

C-----  Finite deformation ?
C-----  NLGEOM = 0,   small deformation theory
C       otherwise, theory of finite rotation and finite strain, Users
C     must declare "NLGEOM" in the input file, at the *STEP card
C
      IF (PROPS(146).EQ.0.) THEN
         NLGEOM=0
      ELSE
         NLGEOM=1
      END IF

C-----  Iteration?
C-----  ITRATN = 0, no iteration
C       otherwise, iteration (solving increments of stresses and
C     solution dependent state variables)
C
      IF (PROPS(153).EQ.0.) THEN
         ITRATN=0
      ELSE
         ITRATN=1
      END IF

      ITRMAX=NINT(PROPS(154))
      GAMERR=PROPS(155)

      NITRTN=-1

      DO I=1,NTENS
         DSOLD(I)=0.
      END DO

      DO J=1,ND
         DGAMOD(J)=0.
         DTAUOD(J)=0.
         DGSPOD(J)=0.
         DO I=1,3
            DSPNRO(I,J)=0.
            DSPDRO(I,J)=0.
         END DO
      END DO

C-----  Increment of spin associated with the material element: DSPIN
C     (only needed for finite rotation)
C
      IF (NLGEOM.NE.0) THEN
         DO J=1,3
            DO I=1,3
               TERM(I,J)=DROT(J,I)
               TRM0(I,J)=DROT(J,I)
            END DO

            TERM(J,J)=TERM(J,J)+1.D0
            TRM0(J,J)=TRM0(J,J)-1.D0
         END DO

         CALL LUDCMP (TERM, 3, 3, ITRM, DDCMP)

         DO J=1,3
            CALL LUBKSB (TERM, 3, 3, ITRM, TRM0(1,J))
         END DO

         DSPIN(1)=TRM0(2,1)-TRM0(1,2)
         DSPIN(2)=TRM0(1,3)-TRM0(3,1)
         DSPIN(3)=TRM0(3,2)-TRM0(2,3)

      END IF

C-----  Increment of dilatational strain: DEV
      DEV=0.D0
      DO I=1,NDI
         DEV=DEV+DSTRAN(I)
      END DO

C-----  Iteration starts (only when iteration method is used)
```

```
1000  CONTINUE

C-----  Parameter NITRTN: number of iterations
C       NITRTN = 0 --- no-iteration solution
C
      NITRTN=NITRTN+1

C-----  Check whether the current stress state is the initial state
      IF (STATEV(1).EQ.0.) THEN

C-----  Initial state
C
C-----  Generating the following parameters and variables at initial
C     state:
C          Total number of slip systems in all the sets NSLPTL
C          Number of slip systems in each set NSLIP
C          Unit vectors in initial slip directions SLPDIR
C          Unit normals to initial slip planes SLPNOR
C
         NSLPTL=0
         DO I=1,NSET
            ISPNOR(1)=NINT(PROPS(25+8*I))
            ISPNOR(2)=NINT(PROPS(26+8*I))
            ISPNOR(3)=NINT(PROPS(27+8*I))

            ISPDIR(1)=NINT(PROPS(28+8*I))
            ISPDIR(2)=NINT(PROPS(29+8*I))
            ISPDIR(3)=NINT(PROPS(30+8*I))

            CALL SLIPSYS (ISPDIR, ISPNOR, NSLIP(I), SLPDIR(1,NSLPTL+1),
     2                   SLPNOR(1,NSLPTL+1), ROTATE)

            NSLPTL=NSLPTL+NSLIP(I)
         END DO

         IF (ND.LT.NSLPTL) THEN
            WRITE (6,*)
     2 '***ERROR - parameter ND chosen by the present user is less than
     3            the total number of slip systems NSLPTL'
            STOP
         END IF

C-----  Slip deformation tensor: SLPDEF (Schmid factors)
         DO J=1,NSLPTL
            SLPDEF(1,J)=SLPDIR(1,J)*SLPNOR(1,J)
            SLPDEF(2,J)=SLPDIR(2,J)*SLPNOR(2,J)
            SLPDEF(3,J)=SLPDIR(3,J)*SLPNOR(3,J)
            SLPDEF(4,J)=SLPDIR(1,J)*SLPNOR(2,J)+SLPDIR(2,J)*SLPNOR(1,J)
            SLPDEF(5,J)=SLPDIR(1,J)*SLPNOR(3,J)+SLPDIR(3,J)*SLPNOR(1,J)
            SLPDEF(6,J)=SLPDIR(2,J)*SLPNOR(3,J)+SLPDIR(3,J)*SLPNOR(2,J)
         END DO

C-----  Initial value of state variables: unit normal to a slip plane
C     and unit vector in a slip direction
C
         STATEV(NSTATV)=FLOAT(NSLPTL)
         DO I=1,NSET
            STATEV(NSTATV-4+I)=FLOAT(NSLIP(I))
         END DO

         IDNOR=3*NSLPTL
         IDDIR=6*NSLPTL
         DO J=1,NSLPTL
            DO I=1,3
               IDNOR=IDNOR+1
               STATEV(IDNOR)=SLPNOR(I,J)

               IDDIR=IDDIR+1
               STATEV(IDDIR)=SLPDIR(I,J)
            END DO
         END DO

C-----  Initial value of the current strength for all slip systems
C
         CALL GSLPINIT (STATEV(1), NSLIP, NSLPTL, NSET, PROPS(97))

C-----  Initial value of shear strain in slip systems
         DO I=1,NSLPTL
            STATEV(NSLPTL+I)=0.
         END DO

         STATEV(9*NSLPTL+1)=0.

C-----  Initial value of the resolved shear stress in slip systems
         DO I=1,NSLPTL
            TERM1=0.

            DO J=1,NTENS
               IF (J.LE.NDI) THEN
                  TERM1=TERM1+SLPDEF(J,I)*STRESS(J)
               ELSE
                  TERM1=TERM1+SLPDEF(J-NDI+3,I)*STRESS(J)
               END IF
            END DO

            STATEV(2*NSLPTL+I)=TERM1
         END DO

      ELSE

C-----  Current stress state
C
C-----  Copying from the array of state variables STATVE the following
C     parameters and variables at current stress state:
C          Total number of slip systems in all the sets NSLPTL
C          Number of slip systems in each set NSLIP
C          Current slip directions SLPDIR
C          Normals to current slip planes SLPNOR
C
         NSLPTL=NINT(STATEV(NSTATV))
         DO I=1,NSET
            NSLIP(I)=NINT(STATEV(NSTATV-4+I))
         END DO

         IDNOR=3*NSLPTL
         IDDIR=6*NSLPTL
         DO J=1,NSLPTL
            DO I=1,3
               IDNOR=IDNOR+1
               SLPNOR(I,J)=STATEV(IDNOR)

               IDDIR=IDDIR+1
               SLPDIR(I,J)=STATEV(IDDIR)
            END DO
         END DO

C-----  Slip deformation tensor: SLPDEF (Schmid factors)
         DO J=1,NSLPTL
```

```
            SLPDEF(1,J)=SLPDIR(1,J)*SLPNOR(1,J)
            SLPDEF(2,J)=SLPDIR(2,J)*SLPNOR(2,J)
            SLPDEF(3,J)=SLPDIR(3,J)*SLPNOR(3,J)
            SLPDEF(4,J)=SLPDIR(1,J)*SLPNOR(2,J)+SLPDIR(2,J)*SLPNOR(1,J)
            SLPDEF(5,J)=SLPDIR(1,J)*SLPNOR(3,J)+SLPDIR(3,J)*SLPNOR(1,J)
            SLPDEF(6,J)=SLPDIR(2,J)*SLPNOR(3,J)+SLPDIR(3,J)*SLPNOR(2,J)
         END DO

      END IF

C-----  Slip spin tensor: SLPSPN (only needed for finite rotation)
      IF (NLGEOM.NE.0) THEN
         DO J=1,NSLPTL
            SLPSPN(1,J)=0.5*(SLPDIR(1,J)*SLPNOR(2,J)-
     2                      SLPDIR(2,J)*SLPNOR(1,J))
            SLPSPN(2,J)=0.5*(SLPDIR(3,J)*SLPNOR(1,J)-
     2                      SLPDIR(1,J)*SLPNOR(3,J))
            SLPSPN(3,J)=0.5*(SLPDIR(2,J)*SLPNOR(3,J)-
     2                      SLPDIR(3,J)*SLPNOR(2,J))
         END DO
      END IF

C-----  Double dot product of elastic moduli tensor with the slip
C     deformation tensor (Schmid factors) plus, only for finite
C     rotation, the dot product of slip spin tensor with the stress:
C     DDEMSD
C
      DO J=1,NSLPTL
         DO I=1,6
            DDEMSD(I,J)=0.
            DO K=1,6
               DDEMSD(I,J)=DDEMSD(I,J)+D(K,I)*SLPDEF(K,J)
            END DO
         END DO
      END DO

      IF (NLGEOM.NE.0) THEN
         DO J=1,NSLPTL

            DDEMSD(4,J)=DDEMSD(4,J)-SLPSPN(1,J)*STRESS(1)
            DDEMSD(5,J)=DDEMSD(5,J)+SLPSPN(2,J)*STRESS(1)

            IF (NDI.GT.1) THEN
               DDEMSD(4,J)=DDEMSD(4,J)+SLPSPN(1,J)*STRESS(2)
               DDEMSD(6,J)=DDEMSD(6,J)-SLPSPN(3,J)*STRESS(2)
            END IF

            IF (NDI.GT.2) THEN
               DDEMSD(5,J)=DDEMSD(5,J)-SLPSPN(2,J)*STRESS(3)
               DDEMSD(6,J)=DDEMSD(6,J)+SLPSPN(3,J)*STRESS(3)
            END IF

            IF (NSHR.GE.1) THEN
               DDEMSD(1,J)=DDEMSD(1,J)+SLPSPN(1,J)*STRESS(NDI+1)
               DDEMSD(2,J)=DDEMSD(2,J)-SLPSPN(1,J)*STRESS(NDI+1)
               DDEMSD(5,J)=DDEMSD(5,J)-SLPSPN(3,J)*STRESS(NDI+1)
               DDEMSD(6,J)=DDEMSD(6,J)+SLPSPN(2,J)*STRESS(NDI+1)
            END IF

            IF (NSHR.GE.2) THEN
               DDEMSD(1,J)=DDEMSD(1,J)-SLPSPN(2,J)*STRESS(NDI+2)
               DDEMSD(3,J)=DDEMSD(3,J)+SLPSPN(2,J)*STRESS(NDI+2)
               DDEMSD(4,J)=DDEMSD(4,J)+SLPSPN(3,J)*STRESS(NDI+2)
               DDEMSD(6,J)=DDEMSD(6,J)-SLPSPN(1,J)*STRESS(NDI+2)
            END IF

            IF (NSHR.EQ.3) THEN
               DDEMSD(2,J)=DDEMSD(2,J)+SLPSPN(3,J)*STRESS(NDI+3)
               DDEMSD(3,J)=DDEMSD(3,J)-SLPSPN(3,J)*STRESS(NDI+3)
               DDEMSD(4,J)=DDEMSD(4,J)-SLPSPN(2,J)*STRESS(NDI+3)
               DDEMSD(5,J)=DDEMSD(5,J)+SLPSPN(1,J)*STRESS(NDI+3)
            END IF

         END DO
      END IF

C-----  Shear strain-rate in a slip system at the start of increment:
C     FSLIP, and its derivative: DFDXSP
C
      ID=1
      DO I=1,NSET
         IF (I.GT.1) ID=ID+NSLIP(I-1)
         CALL STRAINRATE (STATEV(NSLPTL+ID), STATEV(2*NSLPTL+ID),
     2                    STATEV(ID), NSLIP(I), FSLIP(ID), DFDXSP(ID),
     3                    PROPS(65+8*I))
      END DO

C-----  Self- and latent-hardening laws
      CALL LATENTHARDEN (STATEV(NSLPTL+1), STATEV(2*NSLPTL+1),
     2                   STATEV(1), STATEV(9*NSLPTL+1), NSLIP, NSLPTL,
     3                   NSET, H(1,1), PROPS(97), ND)

C-----  LU decomposition to solve the increment of shear strain in a
C     slip system
C
      TERM1=THETA*DTIME
      DO I=1,NSLPTL
         TAUSLP=STATEV(2*NSLPTL+I)
         GSLIP=STATEV(I)
         X=TAUSLP/GSLIP
         TERM2=TERM1*DFDXSP(I)/GSLIP
         TERM3=TERM1*X*DFDXSP(I)/GSLIP

         DO J=1,NSLPTL
            TERM4=0.
            DO K=1,6
               TERM4=TERM4+DDEMSD(K,I)*SLPDEF(K,J)
            END DO

            WORKST(I,J)=-TERM2*TERM4+H(I,J)*TERM3*DSIGN(1.D0,FSLIP(J))

            IF (NITRTN.GT.0) WORKST(I,J)=WORKST(I,J)+TERM3*DHDGDG(I,J)

         END DO

         WORKST(I,I)=WORKST(I,I)+1.
      END DO

      CALL LUDCMP (WORKST, NSLPTL, ND, INDX, DDCMP)

C-----  Increment of shear strain in a slip system: DGAMMA
      TERM1=THETA*DTIME
      DO I=1,NSLPTL

         IF (NITRTN.EQ.0) THEN
            TAUSLP=STATEV(2*NSLPTL+I)
            GSLIP=STATEV(I)
            X=TAUSLP/GSLIP
```

```
            TERM2=TERM1*DFDXSP(I)/GSLIP

            DGAMMA(I)=0.
            DO J=1,NDI
               DGAMMA(I)=DGAMMA(I)+DDEMSD(J,I)*DSTRAN(J)
            END DO

            IF (NSHR.GT.0) THEN
               DO J=1,NSHR
                  DGAMMA(I)=DGAMMA(I)+DDEMSD(J+3,I)*DSTRAN(J+NDI)
               END DO
            END IF

            DGAMMA(I)=DGAMMA(I)*TERM2+FSLIP(I)*DTIME

         ELSE
            DGAMMA(I)=TERM1*(FSLIP(I)-FSLIP1(I))+FSLIP1(I)*DTIME
     2              -DGAMOD(I)

         END IF

      END DO

      CALL LUBKSB (WORKST, NSLPTL, ND, INDX, DGAMMA)

      DO I=1,NSLPTL
         DGAMMA(I)=DGAMMA(I)+DGAMOD(I)
      END DO

C----- Update the shear strain in a slip system: STATEV(NSLPTL+1) -
C     STATEV(2*NSLPTL)
C
      DO I=1,NSLPTL
         STATEV(NSLPTL+I)=STATEV(NSLPTL+I)+DGAMMA(I)-DGAMOD(I)
      END DO

C----- Increment of current strength in a slip system: DGSLIP
      DO I=1,NSLPTL
         DGSLIP(I)=0.
         DO J=1,NSLPTL
            DGSLIP(I)=DGSLIP(I)+H(I,J)*ABS(DGAMMA(J))
         END DO
      END DO

C----- Update the current strength in a slip system: STATEV(I) -
C     STATEV(NSLPTL)
C
      DO I=1,NSLPTL
         STATEV(I)=STATEV(I)+DGSLIP(I)-DGSPOD(I)
      END DO

C----- Increment of strain associated with lattice stretching: DELATS
      DO J=1,6
         DELATS(J)=0.
      END DO

      DO J=1,3
         IF (J.LE.NDI) DELATS(J)=DSTRAN(J)
         DO I=1,NSLPTL
            DELATS(J)=DELATS(J)-SLPDEF(J,I)*DGAMMA(I)
         END DO
      END DO

      DO J=1,3
         IF (J.LE.NSHR) DELATS(J+3)=DSTRAN(J+NDI)
         DO I=1,NSLPTL
            DELATS(J+3)=DELATS(J+3)-SLPDEF(J+3,I)*DGAMMA(I)
         END DO
      END DO

C----- Increment of deformation gradient associated with lattice
C     stretching in the current state, i.e. the velocity gradient
C     (associated with lattice stretching) times the increment of time:
C     DVGRAD (only needed for finite rotation)
C
      IF (NLGEOM.NE.0) THEN
         DO J=1,3
            DO I=1,3
               IF (I.EQ.J) THEN
                  DVGRAD(I,J)=DELATS(I)
               ELSE
                  DVGRAD(I,J)=DELATS(I+J+1)
               END IF
            END DO
         END DO

         DO J=1,3
            DO I=1,J
               IF (J.GT.I) THEN
                  IJ2=I+J-2
                  IF (MOD(IJ2,2).EQ.1) THEN
                     TERM1=1.
                  ELSE
                     TERM1=-1.
                  END IF

                  DVGRAD(I,J)=DVGRAD(I,J)+TERM1*DSPIN(IJ2)
                  DVGRAD(J,I)=DVGRAD(J,I)-TERM1*DSPIN(IJ2)

                  DO K=1,NSLPTL
                     DVGRAD(I,J)=DVGRAD(I,J)-TERM1*DGAMMA(K)*
     2                                       SLPSPN(IJ2,K)
                     DVGRAD(J,I)=DVGRAD(J,I)+TERM1*DGAMMA(K)*
     2                                       SLPSPN(IJ2,K)
                  END DO
               END IF

            END DO
         END DO

      END IF

C----- Increment of resolved shear stress in a slip system: DTAUSP
      DO I=1,NSLPTL
         DTAUSP(I)=0.
         DO J=1,6
            DTAUSP(I)=DTAUSP(I)+DDEMSD(J,I)*DELATS(J)
         END DO
      END DO

C----- Update the resolved shear stress in a slip system:
C     STATEV(2*NSLPTL+1) - STATEV(3*NSLPTL)
C
      DO I=1,NSLPTL
         STATEV(2*NSLPTL+I)=STATEV(2*NSLPTL+I)+DTAUSP(I)-DTAUOD(I)
      END DO

C----- Increment of stress: DSTRES
```

```
      IF (NLGEOM.EQ.0) THEN
         DO I=1,NTENS
            DSTRES(I)=0.
         END DO
      ELSE
         DO I=1,NTENS
            DSTRES(I)=-STRESS(I)*DEV
         END DO
      END IF

      DO I=1,NDI
         DO J=1,NDI
            DSTRES(I)=DSTRES(I)+D(I,J)*DSTRAN(J)
         END DO

         IF (NSHR.GT.0) THEN
            DO J=1,NSHR
               DSTRES(I)=DSTRES(I)+D(I,J+3)*DSTRAN(J+NDI)
            END DO
         END IF

         DO J=1,NSLPTL
            DSTRES(I)=DSTRES(I)-DDEMSD(I,J)*DGAMMA(J)
         END DO
      END DO

      IF (NSHR.GT.0) THEN
         DO I=1,NSHR

            DO J=1,NDI
               DSTRES(I+NDI)=DSTRES(I+NDI)+D(I+3,J)*DSTRAN(J)
            END DO

            DO J=1,NSHR
               DSTRES(I+NDI)=DSTRES(I+NDI)+D(I+3,J+3)*DSTRAN(J+NDI)
            END DO

            DO J=1,NSLPTL
               DSTRES(I+NDI)=DSTRES(I+NDI)-DDEMSD(I+3,J)*DGAMMA(J)
            END DO

         END DO
      END IF

C-----  Update the stress: STRESS
      DO I=1,NTENS
         STRESS(I)=STRESS(I)+DSTRES(I)-DSOLD(I)
      END DO

C-----  Increment of normal to a slip plane and a slip direction (only
C     needed for finite rotation)
C
      IF (NLGEOM.NE.0) THEN
         DO J=1,NSLPTL
            DO I=1,3
               DSPNOR(I,J)=0.
               DSPDIR(I,J)=0.

               DO K=1,3
                  DSPNOR(I,J)=DSPNOR(I,J)-SLPNOR(K,J)*DVGRAD(K,I)
                  DSPDIR(I,J)=DSPDIR(I,J)+SLPDIR(K,J)*DVGRAD(I,K)
               END DO

            END DO
         END DO

C-----  Update the normal to a slip plane and a slip direction (only
C     needed for finite rotation)
C
         IDNOR=3*NSLPTL
         IDDIR=6*NSLPTL
         DO J=1,NSLPTL
            DO I=1,3
               IDNOR=IDNOR+1
               STATEV(IDNOR)=STATEV(IDNOR)+DSPNOR(I,J)-DSPNRO(I,J)

               IDDIR=IDDIR+1
               STATEV(IDDIR)=STATEV(IDDIR)+DSPDIR(I,J)-DSPDRO(I,J)
            END DO
         END DO

      END IF

C-----  Derivative of shear strain increment in a slip system w.r.t.
C     strain increment: DDGDDE
C
      TERM1=THETA*DTIME
      DO I=1,NTENS
         DO J=1,NSLPTL
            TAUSLP=STATEV(2*NSLPTL+J)
            GSLIP=STATEV(J)
            X=TAUSLP/GSLIP
            TERM2=TERM1*DFDXSP(J)/GSLIP
            IF (I.LE.NDI) THEN
               DDGDDE(J,I)=TERM2*DDEMSD(I,J)
            ELSE
               DDGDDE(J,I)=TERM2*DDEMSD(I-NDI+3,J)
            END IF
         END DO

         CALL LUBKSB (WORKST, NSLPTL, ND, INDX, DDGDDE(1,I))

      END DO

C-----  Derivative of stress increment w.r.t. strain increment, i.e.
C     Jacobian matrix
C
C-----  Jacobian matrix: elastic part
      DO J=1,NTENS
         DO I=1,NTENS
            DDSDDE(I,J)=0.
         END DO
      END DO

      DO J=1,NDI
         DO I=1,NDI
            DDSDDE(I,J)=D(I,J)
            IF (NLGEOM.NE.0) DDSDDE(I,J)=DDSDDE(I,J)-STRESS(I)
         END DO
      END DO

      IF (NSHR.GT.0) THEN
         DO J=1,NSHR
            DO I=1,NSHR
               DDSDDE(I+NDI,J+NDI)=D(I+3,J+3)
            END DO

            DO I=1,NDI
```

```
            DDSDDE(I,J+NDI)=D(I,J+3)
            DDSDDE(J+NDI,I)=D(J+3,I)
            IF (NLGEOM.NE.0)
     2         DDSDDE(J+NDI,I)=DDSDDE(J+NDI,I)-STRESS(J+NDI)
          END DO
        END DO
      END IF

C-----  Jacobian matrix: plastic part (slip)
      DO J=1,NDI
         DO I=1,NDI
            DO K=1,NSLPTL
               DDSDDE(I,J)=DDSDDE(I,J)-DDEMSD(I,K)*DDGDDE(K,J)
            END DO
         END DO
      END DO

      IF (NSHR.GT.0) THEN
         DO J=1,NSHR

            DO I=1,NSHR
               DO K=1,NSLPTL
                  DDSDDE(I+NDI,J+NDI)=DDSDDE(I+NDI,J+NDI)-
     2                                DDEMSD(I+3,K)*DDGDDE(K,J+NDI)
               END DO
            END DO

            DO I=1,NDI
               DO K=1,NSLPTL
                  DDSDDE(I,J+NDI)=DDSDDE(I,J+NDI)-
     2                            DDEMSD(I,K)*DDGDDE(K,J+NDI)
                  DDSDDE(J+NDI,I)=DDSDDE(J+NDI,I)-
     2                            DDEMSD(J+3,K)*DDGDDE(K,I)
               END DO
            END DO

         END DO
      END IF

      IF (ITRATN.NE.0) THEN
         DO J=1,NTENS
            DO I=1,NTENS
               DDSDDE(I,J)=DDSDDE(I,J)/(1.+DEV)
            END DO
         END DO
      END IF

C-----  Iteration ?
      IF (ITRATN.NE.0) THEN

C-----  Save solutions (without iteration):
C           Shear strain-rate in a slip system FSLIP1
C           Current strength in a slip system GSLP1
C           Shear strain in a slip system GAMMA1
C           Resolved shear stress in a slip system TAUSP1
C           Normal to a slip plane SPNOR1
C           Slip direction SPDIR1
C           Stress STRES1
C           Jacobian matrix DDSDE1
C
         IF (NITRTN.EQ.0) THEN

            IDNOR=3*NSLPTL
            IDDIR=6*NSLPTL

            DO J=1,NSLPTL
               FSLIP1(J)=FSLIP(J)
               GSLP1(J)=STATEV(J)
               GAMMA1(J)=STATEV(NSLPTL+J)
               TAUSP1(J)=STATEV(2*NSLPTL+J)
               DO I=1,3
                  IDNOR=IDNOR+1
                  SPNOR1(I,J)=STATEV(IDNOR)

                  IDDIR=IDDIR+1
                  SPDIR1(I,J)=STATEV(IDDIR)
               END DO
            END DO

            DO J=1,NTENS
               STRES1(J)=STRESS(J)
               DO I=1,NTENS
                  DDSDE1(I,J)=DDSDDE(I,J)
               END DO
            END DO

         END IF

C-----  Increments of stress DSOLD, and solution dependent state
C     variables DGAMOD, DTAUOD, DGSPOD, DSPNRO, DSPDRO (for the next
C     iteration)
C
         DO I=1,NTENS
            DSOLD(I)=DSTRES(I)
         END DO

         DO J=1,NSLPTL
            DGAMOD(J)=DGAMMA(J)
            DTAUOD(J)=DTAUSP(J)
            DGSPOD(J)=DGSLIP(J)
            DO I=1,3
               DSPNRO(I,J)=DSPNOR(I,J)
               DSPDRO(I,J)=DSPDIR(I,J)
            END DO
         END DO

C-----  Check if the iteration solution converges
         IDBACK=0
         ID=0
         DO I=1,NSET
            DO J=1,NSLIP(I)
               ID=ID+1
               X=STATEV(2*NSLPTL+ID)/STATEV(ID)
               RESIDU=THETA*DTIME*F(X,PROPS(65+8*I))+DTIME*(1.0-THETA)*
     2                FSLIP1(ID)-DGAMMA(ID)
               IF (ABS(RESIDU).GT.GAMERR) IDBACK=1
            END DO
         END DO

         IF (IDBACK.NE.0.AND.NITRTN.LT.ITRMAX) THEN
C-----  Iteration: arrays for iteration
            CALL ITERATION (STATEV(NSLPTL+1), STATEV(2*NSLPTL+1),
     2                      STATEV(1), STATEV(9*NSLPTL+1), NSLPTL,
     3                      NSET, NSLIP, ND, PROPS(97), DGAMOD, DHDGDG)

            GO TO 1000

         ELSE IF (NITRTN.GE.ITRMAX) THEN
C-----  Solution not converge within maximum number of iteration (the
```

```
C     solution without iteration will be used)
C
            DO J=1,NTENS
               STRESS(J)=STRES1(J)
               DO I=1,NTENS
                  DDSDDE(I,J)=DDSDE1(I,J)
               END DO
            END DO

            IDNOR=3*NSLPTL
            IDDIR=6*NSLPTL
            DO J=1,NSLPTL
               STATEV(J)=GSLP1(J)
               STATEV(NSLPTL+J)=GAMMA1(J)
               STATEV(2*NSLPTL+J)=TAUSP1(J)

               DO I=1,3
                  IDNOR=IDNOR+1
                  STATEV(IDNOR)=SPNOR1(I,J)

                  IDDIR=IDDIR+1
                  STATEV(IDDIR)=SPDIR1(I,J)
               END DO
            END DO

         END IF

      END IF

C-----  Total cumulative shear strains on all slip systems (sum of the
C     absolute values of shear strains in all slip systems)
C
      DO I=1,NSLPTL
         STATEV(9*NSLPTL+1)=STATEV(9*NSLPTL+1)+ABS(DGAMMA(I))
      END DO

      RETURN
      END


C-----------------------------------------------------------------------


      SUBROUTINE ROTATION (PROP, ROTATE)

C-----  This subroutine calculates the rotation matrix, i.e. the
C     direction cosines of cubic crystal [100], [010] and [001]
C     directions in global system

C-----  The rotation matrix is stored in the array ROTATE.

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PROP(16), ROTATE(3,3), TERM1(3,3), TERM2(3,3), INDX(3)

C-----  Subroutines:
C
C       CROSS  -- cross product of two vectors
C
C       LUDCMP -- LU decomposition
C
C       LUBKSB -- linear equation solver based on LU decomposition
C                 method (must call LUDCMP first)

C-----  PROP -- constants characterizing the crystal orientation
C               (INPUT)
C
C            PROP(1) - PROP(3) -- direction of the first vector in
C                                 local cubic crystal system
C            PROP(4) - PROP(6) -- direction of the first vector in
C                                 global system
C
C            PROP(9) - PROP(11)-- direction of the second vector in
C                                 local cubic crystal system
C            PROP(12)- PROP(14)-- direction of the second vector in
C                                 global system
C
C-----  ROTATE -- rotation matrix (OUTPUT):
C
C            ROTATE(1,1) -- direction cosines of direction [1 0 0] in
C                           local cubic crystal system
C            ROTATE(1,2) -- direction cosines of direction [0 1 0] in
C                           local cubic crystal system
C            ROTATE(1,3) -- direction cosines of direction [0 0 1] in
C                           local cubic crystal system

C-----  local matrix: TERM1
      CALL CROSS (PROP(1), PROP(9), TERM1, ANGLE1)

C-----  LU decomposition of TERM1
      CALL LUDCMP (TERM1, 3, 3, INDX, DCMP)

C-----  inverse matrix of TERM1: TERM2
      DO J=1,3
         DO I=1,3
            IF (I.EQ.J) THEN
               TERM2(I,J)=1.
            ELSE
               TERM2(I,J)=0.
            END IF
         END DO
      END DO

      DO J=1,3
         CALL LUBKSB (TERM1, 3, 3, INDX, TERM2(1,J))
      END DO

C-----  global matrix: TERM1
      CALL CROSS (PROP(4), PROP(12), TERM1, ANGLE2)

C-----  Check: the angle between first and second vector in local and
C     global systems must be the same.  The relative difference must be
C     less than 0.1%.
C
      IF (ABS(ANGLE1/ANGLE2-1.).GT.0.001) THEN
         WRITE (6,*)
     2      '***ERROR - angles between two vectors are not the same'
         STOP
      END IF

C-----  rotation matrix: ROTATE
      DO J=1,3
         DO I=1,3
            ROTATE(I,J)=0.
            DO K=1,3
               ROTATE(I,J)=ROTATE(I,J)+TERM1(I,K)*TERM2(K,J)
```

```
         END DO
      END DO
   END DO

   RETURN
   END


C-----------------------------------


      SUBROUTINE CROSS (A, B, C, ANGLE)

C-----  (1) normalize vectors A and B to unit vectors
C       (2) store A, B and A*B (cross product) in C

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(3), B(3), C(3,3)

      SUM1=SQRT(A(1)**2+A(2)**2+A(3)**2)
      SUM2=SQRT(B(1)**2+B(2)**2+B(3)**2)

      IF (SUM1.EQ.0.) THEN
         WRITE (6,*) '***ERROR - first vector is zero'
         STOP
      ELSE
         DO I=1,3
            C(I,1)=A(I)/SUM1
         END DO
      END IF

      IF (SUM2.EQ.0.) THEN
         WRITE (6,*) '***ERROR - second vector is zero'
         STOP
      ELSE
         DO I=1,3
            C(I,2)=B(I)/SUM2
         END DO
      END IF

      ANGLE=0.
      DO I=1,3
         ANGLE=ANGLE+C(I,1)*C(I,2)
      END DO
      ANGLE=ACOS(ANGLE)

      C(1,3)=C(2,1)*C(3,2)-C(3,1)*C(2,2)
      C(2,3)=C(3,1)*C(1,2)-C(1,1)*C(3,2)
      C(3,3)=C(1,1)*C(2,2)-C(2,1)*C(1,2)
      SUM3=SQRT(C(1,3)**2+C(2,3)**2+C(3,3)**2)
      IF (SUM3.LT.1.E-8) THEN
         WRITE (6,*)
     2      '***ERROR - first and second vectors are parallel'
         STOP
      END IF

      RETURN
      END


C-----------------------------------------------------------------------
```

```
      SUBROUTINE SLIPSYS (ISPDIR, ISPNOR, NSLIP, SLPDIR, SLPNOR,
     2                    ROTATE)

C-----  This subroutine generates all slip systems in the same set for
C       a CUBIC crystal.  For other crystals (e.g., HCP, Tetragonal,
C       Orthotropic, ...), it has to be modified to include the effect of
C       crystal aspect ratio.

C-----  Denote s as a slip direction and m as normal to a slip plane.
C       In a cubic crystal, (s,-m), (-s,m) and (-s,-m) are NOT considered
C       independent of (s,m).

C-----  Subroutines:  LINE1 and LINE

C-----  Variables:
C
C       ISPDIR -- a typical slip direction in this set of slip systems
C                 (integer)  (INPUT)
C       ISPNOR -- a typical normal to slip plane in this set of slip
C                 systems (integer)  (INPUT)
C       NSLIP  -- number of independent slip systems in this set
C                 (OUTPUT)
C       SLPDIR -- unit vectors of all slip directions  (OUTPUT)
C       SLPNOR -- unit normals to all slip planes  (OUTPUT)
C       ROTATE -- rotation matrix (INPUT)
C            ROTATE(1,1) -- direction cosines of [100] in global system
C            ROTATE(1,2) -- direction cosines of [010] in global system
C            ROTATE(1,3) -- direction cosines of [001] in global system
C
C       NSPDIR -- number of all possible slip directions in this set
C       NSPNOR -- number of all possible slip planes in this set
C       IWKDIR -- all possible slip directions (integer)
C       IWKNOR -- all possible slip planes (integer)


C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ISPDIR(3), ISPNOR(3), SLPDIR(3,50), SLPNOR(3,50),
     *          ROTATE(3,3), IWKDIR(3,24), IWKNOR(3,24), TERM(3)

      NSLIP=0
      NSPDIR=0
      NSPNOR=0

C-----  Generating all possible slip directions in this set
C
C       Denote the slip direction by [lmn].  I1 is the minimum of the
C       absolute value of l, m and n, I3 is the maximum and I2 is the
C       mode, e.g. (1 -3 2), I1=1, I2=2 and I3=3.  I1<=I2<=I3.

      I1=MIN(IABS(ISPDIR(1)),IABS(ISPDIR(2)),IABS(ISPDIR(3)))
      I3=MAX(IABS(ISPDIR(1)),IABS(ISPDIR(2)),IABS(ISPDIR(3)))
      I2=IABS(ISPDIR(1))+IABS(ISPDIR(2))+IABS(ISPDIR(3))-I1-I3

      RMODIR=SQRT(FLOAT(I1*I1+I2*I2+I3*I3))

C     I1=I2=I3=0
      IF (I3.EQ.0) THEN
         WRITE (6,*) '***ERROR - slip direction is [000]'
         STOP

C     I1=I2=0, I3>0   ---   [001] type
```

```
      ELSE IF (I2.EQ.0) THEN
         NSPDIR=3
         DO J=1,3
            DO I=1,3
               IWKDIR(I,J)=0
               IF (I.EQ.J) IWKDIR(I,J)=I3
            END DO
         END DO

C     I1=0, I3>=I2>0
      ELSE IF (I1.EQ.0) THEN

C        I1=0, I3=I2>0   ---   [011] type
         IF (I2.EQ.I3) THEN
            NSPDIR=6
            DO J=1,6
               DO I=1,3
                  IWKDIR(I,J)=I2
                  IF (I.EQ.J.OR.J-I.EQ.3) IWKDIR(1,J)=0
                  IWKDIR(1,6)=-I2
                  IWKDIR(2,4)=-I2
                  IWKDIR(3,5)=-I2
               END DO
            END DO

C        I1=0, I3>I2>0   ---   [012] type
         ELSE
            NSPDIR=12
            CALL LINE1 (I2, I3, IWKDIR(1,1), 1)
            CALL LINE1 (I3, I2, IWKDIR(1,3), 1)
            CALL LINE1 (I2, I3, IWKDIR(1,5), 2)
            CALL LINE1 (I3, I2, IWKDIR(1,7), 2)
            CALL LINE1 (I2, I3, IWKDIR(1,9), 3)
            CALL LINE1 (I3, I2, IWKDIR(1,11), 3)

         END IF

C     I1=I2=I3>0   ---   [111] type
      ELSE IF (I1.EQ.I3) THEN
         NSPDIR=4
         CALL LINE (I1, I1, I1, IWKDIR)

C     I3>I2=I1>0   ---   [112] type
      ELSE IF (I1.EQ.I2) THEN
         NSPDIR=12
         CALL LINE (I1, I1, I3, IWKDIR(1,1))
         CALL LINE (I1, I3, I1, IWKDIR(1,5))
         CALL LINE (I3, I1, I1, IWKDIR(1,9))

C     I3=I2>I1>0   ---   [122] type
      ELSE IF (I2.EQ.I3) THEN
         NSPDIR=12
         CALL LINE (I1, I2, I2, IWKDIR(1,1))
         CALL LINE (I2, I1, I2, IWKDIR(1,5))
         CALL LINE (I2, I2, I1, IWKDIR(1,9))

C     I3>I2>I1>0   ---   [123] type
      ELSE
         NSPDIR=24
         CALL LINE (I1, I2, I3, IWKDIR(1,1))
         CALL LINE (I3, I1, I2, IWKDIR(1,5))
         CALL LINE (I2, I3, I1, IWKDIR(1,9))
         CALL LINE (I1, I3, I2, IWKDIR(1,13))
         CALL LINE (I2, I1, I3, IWKDIR(1,17))
         CALL LINE (I3, I2, I1, IWKDIR(1,21))

      END IF

C-----  Generating all possible slip planes in this set
C
C       Denote the normal to slip plane by (pqr).  J1 is the minimum of
C     the absolute value of p, q and r, J3 is the maximum and J2 is the
C     mode, e.g. (1 -2 1), J1=1, J2=1 and J3=2.  J1<=J2<=J3.

      J1=MIN(IABS(ISPNOR(1)),IABS(ISPNOR(2)),IABS(ISPNOR(3)))
      J3=MAX(IABS(ISPNOR(1)),IABS(ISPNOR(2)),IABS(ISPNOR(3)))
      J2=IABS(ISPNOR(1))+IABS(ISPNOR(2))+IABS(ISPNOR(3))-J1-J3

      RMONOR=SQRT(FLOAT(J1*J1+J2*J2+J3*J3))

      IF (J3.EQ.0) THEN
         WRITE (6,*) '***ERROR - slip plane is [000]'
         STOP

C     (001) type
      ELSE IF (J2.EQ.0) THEN
         NSPNOR=3
         DO J=1,3
            DO I=1,3
               IWKNOR(I,J)=0
               IF (I.EQ.J) IWKNOR(I,J)=J3
            END DO
         END DO

      ELSE IF (J1.EQ.0) THEN

C     (011) type
         IF (J2.EQ.J3) THEN
            NSPNOR=6
            DO J=1,6
               DO I=1,3
                  IWKNOR(I,J)=J2
                  IF (I.EQ.J.OR.J-I.EQ.3) IWKNOR(I,J)=0
                  IWKNOR(1,6)=-J2
                  IWKNOR(2,4)=-J2
                  IWKNOR(3,5)=-J2
               END DO
            END DO

C     (012) type
         ELSE
            NSPNOR=12
            CALL LINE1 (J2, J3, IWKNOR(1,1), 1)
            CALL LINE1 (J3, J2, IWKNOR(1,3), 1)
            CALL LINE1 (J2, J3, IWKNOR(1,5), 2)
            CALL LINE1 (J3, J2, IWKNOR(1,7), 2)
            CALL LINE1 (J2, J3, IWKNOR(1,9), 3)
            CALL LINE1 (J3, J2, IWKNOR(1,11), 3)

         END IF

C     (111) type
      ELSE IF (J1.EQ.J3) THEN
         NSPNOR=4
         CALL LINE (J1, J1, J1, IWKNOR)

C     (112) type
      ELSE IF (J1.EQ.J2) THEN
```

```
        NSPNOR=12
        CALL LINE (J1, J1, J3, IWKNOR(1,1))
        CALL LINE (J1, J3, J1, IWKNOR(1,5))
        CALL LINE (J3, J1, J1, IWKNOR(1,9))

      (122) type
      ELSE IF (J2.EQ.J3) THEN
        NSPNOR=12
        CALL LINE (J1, J2, J2, IWKNOR(1,1))
        CALL LINE (J2, J1, J2, IWKNOR(1,5))
        CALL LINE (J2, J2, J1, IWKNOR(1,9))

:     (123) type
      ELSE
        NSPNOR=24
        CALL LINE (J1, J2, J3, IWKNOR(1,1))
        CALL LINE (J3, J1, J2, IWKNOR(1,5))
        CALL LINE (J2, J3, J1, IWKNOR(1,9))
        CALL LINE (J1, J3, J2, IWKNOR(1,13))
        CALL LINE (J2, J1, J3, IWKNOR(1,17))
        CALL LINE (J3, J2, J1, IWKNOR(1,21))

      END IF

:-----  Generating all slip systems in this set
:
:-----  Unit vectors in slip directions: SLPDIR, and unit normals to
:     slip planes: SLPNOR in local cubic crystal system
:
      WRITE (6,*) '
      WRITE (6,*) ' #          Slip plane          Slip direction'

      DO J=1,NSPNOR
        DO I=1,NSPDIR

          IDOT=0
          DO K=1,3
            IDOT=IDOT+IWKDIR(K,I)*IWKNOR(K,J)
          END DO

          IF (IDOT.EQ.0) THEN
            NSLIP=NSLIP+1
            DO K=1,3
              SLPDIR(K,NSLIP)=IWKDIR(K,I)/RMODIR
              SLPNOR(K,NSLIP)=IWKNOR(K,J)/RMONOR
            END DO

            WRITE (6,10) NSLIP,
     2                   (IWKNOR(K,J),K=1,3), (IWKDIR(K,I),K=1,3)

          END IF

        END DO
      END DO
10    FORMAT(1X,I2,9X,'(',3(1X,I2),1X,')',10X,'[',3(1X,I2),1X,']')

      WRITE (6,*) 'Number of slip systems in this set = ',NSLIP
      WRITE (6,*) '          '

      IF (NSLIP.EQ.0) THEN
        WRITE (6,*)
     *    'There is no slip direction normal to the slip planes!'
        STOP
      ELSE

C-----  Unit vectors in slip directions: SLPDIR, and unit normals to
C     slip planes: SLPNOR in global system
C
        DO J=1,NSLIP
          DO I=1,3
            TERM(I)=0.
            DO K=1,3
              TERM(I)=TERM(I)+ROTATE(I,K)*SLPDIR(K,J)
            END DO
          END DO
          DO I=1,3
            SLPDIR(I,J)=TERM(I)
          END DO

          DO I=1,3
            TERM(I)=0.
            DO K=1,3
              TERM(I)=TERM(I)+ROTATE(I,K)*SLPNOR(K,J)
            END DO
          END DO
          DO I=1,3
            SLPNOR(I,J)=TERM(I)
          END DO
        END DO

      END IF

      RETURN
      END


C---------------------------------


          SUBROUTINE LINE (I1, I2, I3, IARRAY)

C-----  Generating all possible slip directions <lmn> (or slip planes
C     {lmn}) for a cubic crystal, where l,m,n are not zeros.

C-----  Use single precision on cray
C
          IMPLICIT REAL*8 (A-H,O-Z)
          DIMENSION IARRAY(3,4)

          DO J=1,4
            IARRAY(1,J)=I1
            IARRAY(2,J)=I2
            IARRAY(3,J)=I3
          END DO

          DO I=1,3
            DO J=1,4
              IF (J.EQ.I+1) IARRAY(I,J)=-IARRAY(I,J)
            END DO
          END DO

          RETURN
          END


C---------------------------------
```

```
      SUBROUTINE LINE1 (J1, J2, IARRAY, ID)

C-----  Generating all possible slip directions <0mn> (or slip planes
C     {0mn}) for a cubic crystal, where m,n are not zeros and m does
C     not equal n.

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION IARRAY(3,2)

      IARRAY(ID,1)=0
      IARRAY(ID,2)=0

      ID1=ID+1
      IF (ID1.GT.3) ID1=ID1-3
      IARRAY(ID1,1)=J1
      IARRAY(ID1,2)=J1

      ID2=ID+2
      IF (ID2.GT.3) ID2=ID2-3
      IARRAY(ID2,1)=J2
      IARRAY(ID2,2)=-J2

      RETURN
      END


C-----------------------------------------------------------------------


      SUBROUTINE GSLPINIT (GSLIP0, NSLIP, NSLPTL, NSET, PROP)

C-----  This subroutine calculates the initial value of current
C     strength for each slip system in a rate-dependent single crystal.
C     Two sets of initial values, proposed by Asaro, Pierce et al, and
C     by Bassani, respectively, are used here.  Both sets assume that
C     the initial values for all slip systems are the same (initially
C     isotropic).

C-----  These initial values are assumed the same for all slip systems
C     in each set, though they could be different from set to set, e.g.
C     <110>{111} and <110>{100}.

C-----  Users who want to use their own initial values may change the
C     function subprogram GSLP0.  The parameters characterizing these
C     initial values are passed into GSLP0 through array PROP.

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL GSLP0
      DIMENSION GSLIP0(NSLPTL), NSLIP(NSET), PROP(16,NSET)

C-----  Function subprograms:
C
C       GSLP0 -- User-supplied function subprogram given the initial
C                value of current strength at initial state

C-----  Variables:
C
C     GSLIP0 -- initial value of current strength (OUTPUT)
C
C     NSLIP  -- number of slip systems in each set (INPUT)
C     NSLPTL -- total number of slip systems in all the sets (INPUT)
C     NSET   -- number of sets of slip systems (INPUT)
C
C     PROP   -- material constants characterizing the initial value of
C               current strength (INPUT)
C
C               For Asaro, Pierce et al's law
C               PROP(1,I) -- initial hardening modulus H0 in the Ith
C                            set of slip systems
C               PROP(2,I) -- saturation stress TAUs in the Ith set of
C                            slip systems
C               PROP(3,I) -- initial critical resolved shear stress
C                            TAU0 in the Ith set of slip systems
C
C               For Bassani's law
C               PROP(1,I) -- initial hardening modulus H0 in the Ith
C                            set of slip systems
C               PROP(2,I) -- stage I stress TAUI in the Ith set of
C                            slip systems (or the breakthrough stress
C                            where large plastic flow initiates)
C               PROP(3,I) -- initial critical resolved shear stress
C                            TAU0 in the Ith set of slip systems
C

      ID=0
      DO I=1,NSET
         ISET=I
         DO J=1,NSLIP(I)
            ID=ID+1
            GSLIP0(ID)=GSLP0(NSLPTL,NSET,NSLIP,PROP(1,I),ID,ISET)
         END DO
      END DO

      RETURN
      END


C---------------------------------


C-----  Use single precision on cray
C
      REAL*8 FUNCTION GSLP0(NSLPTL,NSET,NSLIP,PROP,ISLIP,ISET)

C-----     User-supplied function subprogram given the initial value of
C       current strength at initial state

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION NSLIP(NSET), PROP(16)

      GSLP0=PROP(3)

      RETURN
      END


C-----------------------------------------------------------------------


      SUBROUTINE STRAINRATE (GAMMA, TAUSLP, GSLIP, NSLIP, FSLIP,
     2                       DFDXSP, PROP)
```

```
C-----  This subroutine calculates the shear strain-rate in each slip
C      system for a rate-dependent single crystal.  The POWER LAW
C      relation between shear strain-rate and resolved shear stress
C      proposed by Hutchinson, Pan and Rice, is used here.

C-----  The power law exponents are assumed the same for all slip
C      systems in each set, though they could be different from set to
C      set, e.g. <110>{111} and <110>{100}.  The strain-rate coefficient
C      in front of the power law form are also assumed the same for all
C      slip systems in each set.

C-----  Users who want to use their own constitutive relation may
C      change the function subprograms F and its derivative DFDX,
C      where F is the strain hardening law, dGAMMA/dt = F(X),
C      X=TAUSLP/GSLIP.  The parameters characterizing F are passed into
C      F and DFDX through array PROP.

C-----  Function subprograms:
C
C       F     -- User-supplied function subprogram which gives shear
C                strain-rate for each slip system based on current
C                values of resolved shear stress and current strength
C
C       DFDX  -- User-supplied function subprogram dF/dX, where x is the
C                ratio of resolved shear stress over current strength

C-----  Variables:
C
C      GAMMA  -- shear strain in each slip system at the start of time
C                step  (INPUT)
C      TAUSLP -- resolved shear stress in each slip system (INPUT)
C      GSLIP  -- current strength (INPUT)
C      NSLIP  -- number of slip systems in this set (INPUT)
C
C      FSLIP  -- current value of F for each slip system (OUTPUT)
C      DFDXSP -- current value of DFDX for each slip system (OUTPUT)
C
C      PROP   -- material constants characterizing the strain hardening
C                law (INPUT)
C
C                For the current power law strain hardening law
C                PROP(1) -- power law hardening exponent
C                PROP(1) = infinity corresponds to a rate-independent
C                material
C                PROP(2) -- coefficient in front of power law hardening


C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL F, DFDX
      DIMENSION GAMMA(NSLIP), TAUSLP(NSLIP), GSLIP(NSLIP),
     2          FSLIP(NSLIP), DFDXSP(NSLIP), PROP(8)

      DO I=1,NSLIP
         X=TAUSLP(I)/GSLIP(I)
         FSLIP(I)=F(X,PROP)
         DFDXSP(I)=DFDX(X,PROP)
      END DO

      RETURN
      END

C-----------------------------------


C-----  Use single precision on cray
C
          REAL*8 FUNCTION F(X,PROP)

C-----     User-supplied function subprogram which gives shear
C        strain-rate for each slip system based on current values of
C        resolved shear stress and current strength
C
C-----  Use single precision on cray
C
          IMPLICIT REAL*8 (A-H,O-Z)
          DIMENSION PROP(8)

          F=PROP(2)*(ABS(X))**PROP(1)*DSIGN(1.D0,X)

          RETURN
          END


C-----------------------------------


C-----  Use single precision on cray
C
          REAL*8 FUNCTION DFDX(X,PROP)

C-----     User-supplied function subprogram dF/dX, where x is the
C        ratio of resolved shear stress over current strength

C-----  Use single precision on cray
C
          IMPLICIT REAL*8 (A-H,O-Z)
          DIMENSION PROP(8)

          DFDX=PROP(1)*PROP(2)*(ABS(X))**(PROP(1)-1.)

          RETURN
          END


C-----------------------------------------------------------------------


     SUBROUTINE LATENTHARDEN (GAMMA, TAUSLP, GSLIP, GAMTOL, NSLIP,
    2                         NSLPTL, NSET, H, PROP, ND)

C-----  This subroutine calculates the current self- and latent-
C      hardening moduli for all slip systems in a rate-dependent single
C      crystal.  Two kinds of hardening law are used here.  The first
C      law, proposed by Asaro, and Pierce et al, assumes a HYPER SECANT
C      relation between self- and latent-hardening moduli and overall
C      shear strain.  The Bauschinger effect has been neglected.  The
C      second is Bassani's hardening law, which gives an explicit
C      expression of slip interactions between slip systems.  The
C      classical three stage hardening for FCC single crystal could be
C      simulated.

C-----  The hardening coefficients are assumed the same for all slip
C      systems in each set, though they could be different from set to
C      set, e.g. <110>{111} and <110>{100}.
```

```
C-----  Users who want to use their own self- and latent-hardening law
C      may change the function subprograms HSELF (self hardening) and
C      HLATNT (latent hardening).  The parameters characterizing these
C      hardening laws are passed into HSELF and HLATNT through array
C      PROP.


C-----  Function subprograms:
C
C       HSELF  -- User-supplied self-hardening function in a slip
C                 system
C
C       HLATNT -- User-supplied latent-hardening function

C-----  Variables:
C
C      GAMMA  -- shear strain in all slip systems at the start of time
C                step  (INPUT)
C      TAUSLP -- resolved shear stress in all slip systems (INPUT)
C      GSLIP  -- current strength (INPUT)
C      GAMTOL -- total cumulative shear strains over all slip systems
C                (INPUT)
C      NSLIP  -- number of slip systems in each set (INPUT)
C      NSLPTL -- total number of slip systems in all the sets (INPUT)
C      NSET   -- number of sets of slip systems (INPUT)
C
C      H      -- current value of self- and latent-hardening moduli
C                (OUTPUT)
C                H(I,I) -- self-hardening modulus of the ith slip system
C                          (no sum over I)
C                H(I,J) -- latent-hardening molulus of the ith slip
C                          system due to a slip in the jth slip system
C                          (I not equal J)
C
C      PROP   -- material constants characterizing the self- and latent-
C                hardening law (INPUT)
C
C                For the HYPER SECANT hardening law
C                PROP(1,I) -- initial hardening modulus H0 in the ith
C                             set of slip systems
C                PROP(2,I) -- saturation stress TAUs in the ith set of
C                             slip systems
C                PROP(3,I) -- initial critical resolved shear stress
C                             TAU0 in the ith set of slip systems
C                PROP(9,I) -- ratio of latent to self-hardening Q in the
C                             ith set of slip systems
C                PROP(10,I)-- ratio of latent-hardening from other sets
C                             of slip systems to self-hardening in the
C                             ith set of slip systems Q1
C
C                For Hassani's hardening law
C                PROP(1,I) -- initial hardening modulus H0 in the ith
C                             set of slip systems
C                PROP(2,I) -- stage I stress TAUI in the ith set of
C                             slip systems (or the breakthrough stress
C                             where large plastic flow initiates)
C                PROP(3,I) -- initial critical resolved shear stress
C                             TAU0 in the ith set of slip systems
C                PROP(4,I) -- hardening modulus during easy glide Hs in
C                             the ith set of slip systems
C                PROP(5,I) -- amount of slip Gamma0 after which a given
C                             interaction between slip systems in the
C                             ith set reaches peak strength
C                PROP(6,I) -- amount of slip Gamma0 after which a given
C                             interaction between slip systems in the
C                             ith set and jth set (I not equal J)
C                             reaches peak strength
C                PROP(7,I) -- representing the magnitude of the strength
C                             of interaction in the ith set of slip
C                             system
C                PROP(8,I) -- representing the magnitude of the strength
C                             of interaction between the ith set and jth
C                             set of system
C                PROP(9,I) -- ratio of latent to self-hardening Q in the
C                             ith set of slip systems
C                PROP(10,I)-- ratio of latent-hardening from other sets
C                             of slip systems to self-hardening in the
C                             ith set of slip systems Q1
C
C     ND     -- leading dimension of arrays defined in subroutine UMAT
C               (INPUT)


C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL HSELF, HLATNT
      DIMENSION GAMMA(NSLPTL), TAUSLP(NSLPTL), GSLIP(NSLPTL),
     2          NSLIP(NSET), PROP(16,NSET), H(ND,NSLPTL)

      CHECK=0.
      DO I=1,NSET
         DO J=4,8
            CHECK=CHECK+ABS(PROP(J,I))
         END DO
      END DO

C-----  CHECK=0   --  HYPER SECANT hardening law
C       otherwise --  Bassani's hardening law

      ISELF=0
      DO I=1,NSET
         ISET=I
         DO J=1,NSLIP(I)
            ISELF=ISELF+1

            DO LATENT=1,NSLPTL
               IF (LATENT.EQ.ISELF) THEN
                  H(LATENT,ISELF)=HSELF(GAMMA,GAMTOL,NSLPTL,NSET,NSLIP,
     2                                  PROP(1,I),CHECK,ISELF,ISET)
               ELSE
                  H(LATENT,ISELF)=HLATNT(GAMMA,GAMTOL,NSLPTL,NSET,
     2                                   NSLIP,PROP(1,I),CHECK,ISELF,
     3                                   ISET,LATENT)
               END IF
            END DO

         END DO
      END DO

      RETURN
      END

C---------------------------------
```

```
C-----  Use single precision on cray
C
      REAL*8 FUNCTION HSELF(GAMMA,GAMTOL,NSLPTL,NSET,NSLIP,PROP,
     2                      CHECK,ISELF,ISET)

C-----     User-supplied self-hardening function in a slip system

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION GAMMA(NSLPTL), NSLIP(NSET), PROP(16)

      IF (CHECK.EQ.0.) THEN

C-----  HYPER SECANT hardening law by Asaro, Pierce et al
        TERM1=PROP(1)*GAMTOL/(PROP(2)-PROP(3))
        TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
        HSELF=PROP(1)*TERM2**2

      ELSE

C-----  Bassani's hardening law
        TERM1=(PROP(1)-PROP(4))*GAMMA(ISELF)/(PROP(2)-PROP(3))
        TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
        F=(PROP(1)-PROP(4))*TERM2**2+PROP(4)

        ID=0
        G=1.
        DO I=1,NSET
          IF (I.EQ.ISET) THEN
            GAMMA0=PROP(5)
            FAB=PROP(7)
          ELSE
            GAMMA0=PROP(6)
            FAB=PROP(8)
          END IF

          DO J=1,NSLIP(I)
            ID=ID+1
            IF (ID.NE.ISELF) G=G+FAB*TANH(GAMMA(ID)/GAMMA0)
          END DO
        END DO

        HSELF=F*G

      END IF

      RETURN
      END


C-----------------------------------


C-----  Use single precision on cray
C
      REAL*8 FUNCTION HLATNT(GAMMA,GAMTOL,NSLPTL,NSET,NSLIP,PROP,
     2                       CHECK,ISELF,ISET,LATENT)

C-----     User-supplied latent-hardening function

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION GAMMA(NSLPTL), NSLIP(NSET), PROP(16)

      ILOWER=0
      IUPPER=NSLIP(1)
      IF (ISET.GT.1) THEN
        DO K=2,ISET
          ILOWER=ILOWER+NSLIP(K-1)
          IUPPER=IUPPER+NSLIP(K)
        END DO
      END IF

      IF (LATENT.GT.ILOWER.AND.LATENT.LE.IUPPER) THEN
        Q=PROP(9)
      ELSE
        Q=PROP(10)
      END IF

      IF (CHECK.EQ.0.) THEN

C-----  HYPER SECANT hardening law by Asaro, Pierce et al
        TERM1=PROP(1)*GAMTOL/(PROP(2)-PROP(3))
        TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
        HLATNT=PROP(1)*TERM2**2*Q

      ELSE

C-----  Bassani's hardening law
        TERM1=(PROP(1)-PROP(4))*GAMMA(ISELF)/(PROP(2)-PROP(3))
        TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
        F=(PROP(1)-PROP(4))*TERM2**2+PROP(4)

        ID=0
        G=1.
        DO I=1,NSET
          IF (I.EQ.ISET) THEN
            GAMMA0=PROP(5)
            FAB=PROP(7)
          ELSE
            GAMMA0=PROP(6)
            FAB=PROP(8)
          END IF

          DO J=1,NSLIP(I)
            ID=ID+1
            IF (ID.NE.ISELF) G=G+FAB*TANH(GAMMA(ID)/GAMMA0)
          END DO
        END DO

        HLATNT=F*G*Q

      END IF

      RETURN
      END


C-----------------------------------------------------------------------


      SUBROUTINE ITERATION (GAMMA, TAUSLP, GSLIP, GAMTOL, NSLPTL, NSET,
     2                      NSLIP, ND, PROP, DGAMOD, DHDGDG)

C-----  This subroutine generates arrays for the Newton-Rhapson
C      iteration method.
```

```
C-----  Users who want to use their own self- and latent-hardening law
C     may change the function subprograms DHSELF (self hardening) and
C     DHLATN (latent hardening).  The parameters characterizing these
C     hardening laws are passed into DHSELF and DHLATN through array
C     PROP.


C-----  Function subprograms:
C
C       DHSELF -- User-supplied function of the derivative of self-
C                 hardening modul1
C
C       DHLATN -- User-supplied function of the derivative of latent-
C                 hardening modul1

C-----  Variables:
C
C     GAMMA  -- shear strain in all slip systems at the start of time
C               step  (INPUT)
C     TAUSLP -- resolved shear stress in all slip systems (INPUT)
C     GSLIP  -- current strength (INPUT)
C     GAMTOL -- total cumulative shear strains over all slip systems
C               (INPUT)
C     NSLPTL -- total number of slip systems in all the sets (INPUT)
C     NSET   -- number of sets of slip systems (INPUT)
C     NSLIP  -- number of slip systems in each set (INPUT)
C     ND     -- leading dimension of arrays defined in subroutine UMAT
C               (INPUT)
C
C     PROP   -- material constants characterizing the self- and latent-
C               hardening law (INPUT)
C
C                For the HYPER SECANT hardening law
C                PROP(1,i) -- initial hardening modulus H0 in the ith
C                             set of slip systems
C                PROP(2,i) -- saturation stress TAUs in the ith set of
C                             slip systems
C                PROP(3,i) -- initial critical resolved shear stress
C                             TAU0 in the ith set of slip systems
C                PROP(9,i) -- ratio of latent to self-hardening Q in the
C                             ith set of slip systems
C                PROP(10,i)-- ratio of latent-hardening from other sets
C                             of slip systems to self-hardening in the
C                             ith set of slip systems Q1
C
C                For Bassani's hardening law
C                PROP(1,i) -- initial hardening modulus H0 in the ith
C                             set of slip systems
C                PROP(2,i) -- stage I stress TAUI in the ith set of
C                             slip systems (or the breakthrough stress
C                             where large plastic flow initiates)
C                PROP(3,i) -- initial critical resolved shear stress
C                             TAU0 in the ith set of slip systems
C                PROP(4,i) -- hardening modulus during easy glide Hs in
C                             the ith set of slip systems
C                PROP(5,i) -- amount of slip Gamma0 after which a given
C                             interaction between slip systems in the
C                             ith set reaches peak strength
C                PROP(6,i) -- amount of slip Gamma0 after which a given
C                             interaction between slip systems in the
C                             ith set and jth set (i not equal j)
C                             reaches peak strength
C                PROP(7,i) -- representing the magnitude of the strength
C                             of interaction in the ith set of slip
C                             system
C                PROP(8,i) -- representing the magnitude of the strength
C                             of interaction between the ith set and jth
C                             set of system
C                PROP(9,i) -- ratio of latent to self-hardening Q in the
C                             ith set of slip systems
C                PROP(10,i)-- ratio of latent-hardening from other sets
C                             of slip systems to self-hardening in the
C                             ith set of slip systems Q1
C
C-----  Arrays for iteration:
C
C       DGAMOD (INPUT)
C
C       DHDGDG (OUTPUT)
C

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL DHSELF, DHLATN
      DIMENSION GAMMA(NSLPTL), TAUSLP(NSLPTL), GSLIP(NSLPTL),
     2          NSLIP(NSET), PROP(16,NSET),
     3          DGAMOD(NSLPTL), DHDGDG(ND,NSLPTL)

      CHECK=0.
      DO I=1,NSET
         DO J=4,8
            CHECK=CHECK+ABS(PROP(J,I))
         END DO
      END DO

C-----  CHECK=0   --  HYPER SECANT hardening law
C       otherwise --  Bassani's hardening law

      ISELF=0
      DO I=1,NSET
         ISET=I
         DO J=1,NSLIP(I)
            ISELF=ISELF+1

            DO KDERIV=1,NSLPTL
               DHDGDG(ISELF,KDERIV)=0.

               DO LATENT=1,NSLPTL
                  IF (LATENT.EQ.ISELF) THEN
                     DHDG=DHSELF(GAMMA,GAMTOL,NSLPTL,NSET,NSLIP,
     2                           PROP(1,I),CHECK,ISELF,ISET,KDERIV)
                  ELSE
                     DHDG=DHLATN(GAMMA,GAMTOL,NSLPTL,NSET,NSLIP,
     2                           PROP(1,I),CHECK,ISELF,ISET,LATENT,
     3                           KDERIV)
                  END IF

                  DHDGDG(ISELF,KDERIV)=DHDGDG(ISELF,KDERIV)+
     2                                 DHDG*ABS(DGAMOD(LATENT))
               END DO

            END DO
         END DO
      END DO

      RETURN
```

```
      END


C-----------------------------------


C-----  Use single precision on cray
C
          REAL*8 FUNCTION DHSELF(GAMMA,GAMTOL,NSLPTL,NSET,NSLIP,PROP,
     2                           CHECK,ISELF,ISET,KDERIV)

C-----  User-supplied function of the derivative of self-hardening
C     moduli

C-----  Use single precision on cray
C
          IMPLICIT REAL*8 (A-H,O-Z)
          DIMENSION GAMMA(NSLPTL), NSLIP(NSET), PROP(16)

          IF (CHECK.EQ.0.) THEN

C-----  HYPER SECANT hardening law by Asaro, Pierce et al
            TERM1=PROP(1)*GAMTOL/(PROP(2)-PROP(3))
            TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
            TERM3=PROP(1)/(PROP(2)-PROP(3))*DSIGN(1.D0,GAMMA(KDERIV))
            DHSELF=-2.*PROP(1)*TERM2**2*TANH(TERM1)*TERM3

          ELSE

C-----  Bassani's hardening law
            TERM1=(PROP(1)-PROP(4))*GAMMA(ISELF)/(PROP(2)-PROP(3))
            TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
            TERM3=(PROP(1)-PROP(4))/(PROP(2)-PROP(3))

            IF (KDERIV.EQ.ISELF) THEN
               F=-2.*(PROP(1)-PROP(4))*TERM2**2*TANH(TERM1)*TERM3
               ID=0
               G=1.
               DO I=1,NSET
                  IF (I.EQ.ISET) THEN
                     GAMMA0=PROP(5)
                     FAB=PROP(7)
                  ELSE
                     GAMMA0=PROP(6)
                     FAB=PROP(8)
                  END IF

                  DO J=1,NSLIP(I)
                     ID=ID+1
                     IF (ID.NE.ISELF) G=G+FAB*TANH(GAMMA(ID)/GAMMA0)
                  END DO
               END DO

            ELSE
               F=(PROP(1)-PROP(4))*TERM2**2+PROP(4)
               ILOWER=0
               IUPPER=NSLIP(1)
               IF (ISET.GT.1) THEN
                  DO K=2,ISET
                     ILOWER=ILOWER+NSLIP(K-1)
                     IUPPER=IUPPER+NSLIP(K)
                  END DO
               END IF

               IF (KDERIV.GT.ILOWER.AND.KDERIV.LE.IUPPER) THEN
                  GAMMA0=PROP(5)
                  FAB=PROP(7)
               ELSE
                  GAMMA0=PROP(6)
                  FAB=PROP(8)
               END IF

               TERM4=GAMMA(KDERIV)/GAMMA0
               TERM5=2.*EXP(-TERM4)/(1.+EXP(-2.*TERM4))
               G=FAB/GAMMA0*TERM5**2

            END IF

            DHSELF=F*G

          END IF

          RETURN
          END


C-----------------------------------


C-----  Use single precision on cray
C
          REAL*8 FUNCTION DHLATN(GAMMA,GAMTOL,NSLPTL,NSET,NSLIP,PROP,
     2                           CHECK,ISELF,ISET,LATENT,KDERIV)

C-----  User-supplied function of the derivative of latent-hardening
C     moduli

C-----  Use single precision on cray
C
          IMPLICIT REAL*8 (A-H,O-Z)
          DIMENSION GAMMA(NSLPTL), NSLIP(NSET), PROP(16)

          ILOWER=0
          IUPPER=NSLIP(1)
          IF (ISET.GT.1) THEN
             DO K=2,ISET
                ILOWER=ILOWER+NSLIP(K-1)
                IUPPER=IUPPER+NSLIP(K)
             END DO
          END IF

          IF (LATENT.GT.ILOWER.AND.LATENT.LE.IUPPER) THEN
             Q=PROP(9)
          ELSE
             Q=PROP(10)
          END IF

          IF (CHECK.EQ.0.) THEN

C-----  HYPER SECANT hardening law by Asaro, Pierce et al
            TERM1=PROP(1)*GAMTOL/(PROP(2)-PROP(3))
            TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
            TERM3=PROP(1)/(PROP(2)-PROP(3))*DSIGN(1.D0,GAMMA(KDERIV))
            DHLATN=-2.*PROP(1)*TERM2**2*TANH(TERM1)*TERM3*Q

          ELSE

C-----  Bassani's hardening law
```

```
      TERM1=(PROP(1)-PROP(4))*GAMMA(ISELF)/(PROP(2)-PROP(3))
      TERM2=2.*EXP(-TERM1)/(1.+EXP(-2.*TERM1))
      TERM3=(PROP(1)-PROP(4))/(PROP(2)-PROP(3))

      IF (KDERIV.EQ.ISELF) THEN
         F=-2.*(PROP(1)-PROP(4))*TERM2**2*TANH(TERM1)*TERM3
         ID=0
         G=1.
         DO I=1,NSET
            IF (I.EQ.ISET) THEN
               GAMMA0=PROP(5)
               FAB=PROP(7)
            ELSE
               GAMMA0=PROP(6)
               FAB=PROP(8)
            END IF

            DO J=1,NSLIP(I)
               ID=ID+1
               IF (ID.NE.ISELF) G=G+FAB*TANH(GAMMA(ID)/GAMMA0)
            END DO
         END DO

      ELSE
         F=(PROP(1)-PROP(4))*TERM2**2+PROP(4)
         ILOWER=0
         IUPPER=NSLIP(1)
         IF (ISET.GT.1) THEN
            DO K=2,ISET
               ILOWER=ILOWER+NSLIP(K-1)
               IUPPER=IUPPER+NSLIP(K)
            END DO
         END IF

         IF (KDERIV.GT.ILOWER.AND.KDERIV.LE.IUPPER) THEN
            GAMMA0=PROP(5)
            FAB=PROP(7)
         ELSE
            GAMMA0=PROP(6)
            FAB=PROP(8)
         END IF

         TERM4=GAMMA(KDERIV)/GAMMA0
         TERM5=2.*EXP(-TERM4)/(1.+EXP(-2.*TERM4))
         G=FAB/GAMMA0*TERM5**2

      END IF

      DHLATN=F*G*Q

   END IF

   RETURN
   END

C-----------------------------------------------------------------------

      SUBROUTINE LUDCMP (A, N, NP, INDX, D)

C-----  LU decomposition

C-----  Use single precision on cray

C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (NMAX=200, TINY=1.0E-20)
      DIMENSION A(NP,NP), INDX(N), VV(NMAX)

      D=1.
      DO I=1,N
         AAMAX=0.

         DO J=1,N
            IF (ABS(A(I,J)).GT.AAMAX) AAMAX=ABS(A(I,J))
         END DO

         IF (AAMAX.EQ.0.) PAUSE 'Singular matrix.'
         VV(I)=1./AAMAX
      END DO

      DO J=1,N
         DO I=1,J-1
            SUM=A(I,J)

            DO K=1,I-1
               SUM=SUM-A(I,K)*A(K,J)
            END DO

            A(I,J)=SUM
         END DO
         AAMAX=0.

         DO I=J,N
            SUM=A(I,J)

            DO K=1,J-1
               SUM=SUM-A(I,K)*A(K,J)
            END DO

            A(I,J)=SUM
            DUM=VV(I)*ABS(SUM)
            IF (DUM.GE.AAMAX) THEN
               IMAX=I
               AAMAX=DUM
            END IF
         END DO

         IF (J.NE.IMAX) THEN
            DO K=1,N
               DUM=A(IMAX,K)
               A(IMAX,K)=A(J,K)
               A(J,K)=DUM
            END DO

            D=-D
            VV(IMAX)=VV(J)
         END IF

         INDX(J)=IMAX
         IF (A(J,J).EQ.0.) A(J,J)=TINY
         IF (J.NE.N) THEN
            DUM=1./A(J,J)
            DO I=J+1,N
               A(I,J)=A(I,J)*DUM
            END DO
         END IF
```

```
      END DO

      RETURN
      END


C-----------------------------------------------------------------------


      SUBROUTINE LUBKSB (A, N, NP, INDX, B)

C-----  Linear equation solver based on LU decomposition

C-----  Use single precision on cray
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NP,NP), INDX(N), B(N)

      II=0
      DO I=1,N
         LL=INDX(I)
         SUM=B(LL)
         B(LL)=B(I)

         IF (II.NE.0) THEN
            DO J=II,I-1
               SUM=SUM-A(I,J)*B(J)
            END DO
         ELSE IF (SUM.NE.0.) THEN
            II=I
         END IF

         B(I)=SUM
      END DO

      DO I=N,1,-1
         SUM=B(I)

         IF (I.LT.N) THEN
            DO J=I+1,N
               SUM=SUM-A(I,J)*B(J)
            END DO
         END IF

         B(I)=SUM/A(I,I)
      END DO

      RETURN
      END
**
**
**
********** Load step follows **********
**
*RESTART,WRITE,FREQUENCY=50
**
*STEP,INC=500,CYCLE=25,NLGEOM,ROTTOL=0.02
*STATIC,PTOL=0.2
0.01,1.0,0.00001,0.2
*DLOAD
ONE,P2,-2.0E2
**
*NODE PRINT,FREQUENCY=500
U,RF
*EL PRINT,FREQUENCY=500
S
*EL PRINT,FREQUENCY=500
E
*EL PRINT,FREQUENCY=500
SDV13,SDV14,SDV15,SDV16,SDV17,SDV18,SDV19,SDV20
SDV21,SDV22,SDV23,SDV24,SDV109
**
*NODE FILE,FREQUENCY=1
U
*EL FILE,FREQUENCY=1
S,E
*EL FILE,FREQUENCY=1
SDV
**
*END STEP
```