*Evolution of Plastic Anisotropy for
High-Strain-Rate Computations*

Sheila K. Schiferl

Paul J. Maudlin

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Los Alamos
NATIONAL LABORATORY

Los Alamos, New Mexico 87545

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

EVOLUTION OF PLASTIC ANISOTROPY FOR HIGH-STRAIN-RATE COMPUTATIONS

by

Sheila K. Schiferl and Paul J. Maudlin

ABSTRACT

A model for anisotropic material strength, and for changes in the anisotropy due to plastic strain, is described. This model has been developed for use in high-rate, explicit, Lagrangian multidimensional continuum-mechanics codes. The model handles anisotropies in single-phase materials, in particular the anisotropies due to crystallographic texture—preferred orientations of the single-crystal grains. Textural anisotropies, and the changes in these anisotropies, depend overwhelmingly on the crystal structure of the material and on the deformation history. The changes, particularly for complex deformations, are not amenable to simple analytical forms. To handle this problem, the material model described here includes a texture code, or micromechanical calculation, coupled to a continuum code. The texture code updates grain orientations as a function of tensor plastic strain, and calculates the yield strength in different directions. A yield function is fitted to these yield "points." For each computational cell in the continuum simulation, the texture code tracks a particular set of grain orientations. The orientations will change due to the tensor strain history, and the yield function will change accordingly. Hence, the continuum code supplies a tensor strain to the texture code, and the texture code supplies an updated yield function to the continuum code. Since significant texture changes require relatively large strains—typically, a few percent or more—the texture code is not called very often, and the increase in computer time is not excessive. The model was implemented, using a finite-element continuum code and a texture code specialized for hexagonal-close-packed crystal structures. The results for several uniaxial stress problems and an explosive-forming problem are shown.

I. INTRODUCTION

Many substances have directional properties. Typical anisotropic solids include multiphase materials such as composites, as well as a surprising variety of single-phase materials, including metals. Some of these materials have significant directionality in plastic properties, which can influence response to large deformations. Such directional plastic response can produce unique features, like earing of deep-drawn cups,^{1,2} that are not seen in isotropic materials. In this work, we will concentrate on single-phase plastic anisotropy and introduce a method for modeling anisotropic change during high-rate deformations.

The major underlying cause of single-phase anisotropy is crystallographic texture, the preferred orientation of single-crystal grains in a polycrystalline solid. Single crystals are typically anisotropic; if the grains are not randomly oriented, the material will exhibit anisotropic properties. The kinds of texture and any resulting plastic directionality depend on both history (particularly the tensor strain history) and the particular crystal structure of the grains. For hexagonal-close-packed (hcp) metals such as titanium and zirconium, where there is a unique crystal direction, plastic anisotropies due to texture can be very large.³⁻⁵ For cross-rolling textures, the strong direction is normal to the rolled sheet. Yield anisotropies can be greater than 2:1, and plastic strain ratios (R-values) typically range from 3 to 7. For extrusion, the anisotropies are again large, but the extrusion axis is now the weak direction. For cubic metals, where there is no unique crystal direction, yield anisotropies tend to be small, but strain anisotropies can be significant ($R \sim 0.6$ for rolled copper; $R > 2.5$ for some steels).⁵

A strong initial texture is usually the result of a large deformation of a particular type (extrusion, rolling, torsion, etc.). Subsequent deformation of a different type can change this texture to a different pattern. The texture pattern, at any point during the deformation, will depend on the crystal structure, the initial texture, and the subsequent history (primarily the tensor strain and secondarily the physical conditions—temperature and strain-rate—that can influence the crystal response to deformation).

We have previously described constitutive models and solution algorithms for handling plastic anisotropy in high-rate continuum mechanics codes.⁶ For a code utilizing only isotropic response, addition of an anisotropic

model typically requires modification of a number of routines: stress rate (polar decomposition, rather than Jaumann rate), solutions to elastoplastic equations (associative flow, rather than radial return), frame of reference for stress updating (a frame that rotates with a computational element, rather than the laboratory frame), as well as an anisotropic form for the yield function. In this work, we extend the implementation of Ref. 6 to include a model for evolution of anisotropy in textured materials.

The most important consideration for this modeling is the extreme history-dependence of texture change—a simple analytical form probably does not exist for changes of anisotropy in an arbitrary strain path.⁷ Hence, to implement an anisotropic model that includes evolution of anisotropy during a dynamic deformation, it is desirable to be simulating certain features of the microstructural response and updating the material model as the deformation proceeds. One way of handling such a two-level simulation is by linking an element in a Lagrangian-formulation continuum code to a set of oriented grains in a mechanistic texture calculation. The continuum code calculates a bulk (macroscopic) deformation. The texture code calculates changes in preferred orientation (texture evolution) for a set of single-crystal grains, with the entire set subject to the bulk deformation from the continuum code. The texture code is also used to calculate a yield stress for the set of grains and return the yield stress value to the continuum code.

This approach—input a strain, update the texture, output a stress for each code cycle—has proved workable for implicit codes and metal-forming applications.⁸ In the sections below, we introduce a variation of this approach for explicit codes and high-rate applications. In this variation, we integrate the deformation tensor over a number of time steps (100 to 1000 steps) before calling a texture calculation to update the grain orientations. We also update the yield function, by fitting a set of texture-code stress calculations, as part of our elastoplastic constitutive model.

The details of the construction of this coupled continuum code/texture code for high-rate applications are given in the sections below. Sections II.A–C contain a description of texture modeling. In Sec. II.D, texture modeling is adapted to yield calculations. Section III contains a description of our particular implementation, including modifications to standard texture codes and continuum codes, and details of our continuum code/texture code interface. The use of these coupled codes for high-rate, large-deformation

problems is described in Sec. IV. Two kinds of test problems are addressed: uniaxial stress and explosive forming.

II. TEXTURE METHODS

In this section, we will discuss the mathematics of texture calculations. The methods for these calculations have been described in detail elsewhere;⁹⁻¹¹ we give an outline here.

A. General Principles

Texture calculations are based on simulation of the dominant physical mechanisms of plastic deformation that underlie deformation-texture evolution. These mechanisms are reasonably well understood for a variety of conditions, including shocks and high temperatures. We consider plastic deformation of a polycrystalline material to be a result of shear in each crystal, according to two basic mechanisms: crystallographic slip and mechanical twinning.¹² For high strain-rate calculations, we ignore grain-boundary sliding.

In both slip and twinning, the "stack" of crystal planes shears in a particular direction. The shear only occurs on certain crystal planes and along certain crystal directions. Slip involves the whole crystal, and movement can be either forward and backward along certain crystal directions. Both slip and twinning rotate the crystal lattice.

In mechanical twinning, a band of the crystal undergoes both shear and a slight shuffle in atom positions. Hence, twinning does not generally involve the whole crystal. The important physical quantities are the volume fraction of the twin band(s) and the twinning shear. The latter is fixed, both in magnitude and direction, and depends on the crystal structure. The twinned material is effectively rotated relative to the untwinned matrix. This rotation can be very large; in hcp crystals, the *c* axis (the "hard" direction) rotates by $\sim 85^\circ$ for a common type of tensile twin. This is not a physical rotation, but the shear plus the shuffle in atomic positions is crystallographically indistinguishable from a rotation and can be treated mathematically as such.¹³

We consider the activation of any slip or twinning system (where a system consists of a plane plus a direction) to require a certain critical-

resolved-shear stress (CRSS). The particular systems that can exist depend on the crystal structure. The relative difficulty of operation of different kinds of systems, and hence the relative CRSSs, depends on the particular material and the conditions, particularly temperature and strain-rate. Systems that are related by crystal symmetry have essentially the same properties. Mathematically, symmetry-related systems are assigned the same CRSSs.

For texture calculations, we assume that the symmetric, deviatoric portion of an incremental deformation tensor can be accommodated by crystallographic slip and/or twinning, plus grain rotation. The object of the calculations is to find which crystallographic systems are active and how the incremental shear is divided among these systems. The amounts of shear and the system geometries can then be used to calculate the rotation. Strain compatibility (homogeneous strain over the set of grains) is generally assumed. The key to texture changes (and resulting changes in anisotropy) is the grain rotation, or lattice rotation, of each grain relative to other grains. The ordinary rigid-body rotation of the whole set of grains can be treated separately from the individual lattice rotations. Rigid-body rotation is driven by the anti-symmetric part of the velocity-gradient tensor. In general, there will be different lattice rotations for grains of different orientations.

A deformation step corresponds to a single texture-code cycle. The step is large enough so that the deformation is essentially all plastic; we assume that elastic strains can be neglected compared to plastic ones. However, texture code formulas use expressions for "infinitesimal" strains, so the deformation steps need to be kept small, albeit plastic. For typical texture calculations, a reasonable step size is on the order of a few percent, and an input deformation tensor can be subdivided, if necessary. The result of lattice rotations for a set of grains, integrated over many deformation steps, is texture evolution. Evolution calculations, including special considerations when twinning is present, are described below in Secs. II.B-C.

The response of a single crystal to plastic strain, in addition to a lattice rotation, is a stress. Due to the discontinuous nature of slip, this stress is not necessarily in the same direction as the original strain and will depend on the grain orientation relative to the applied strain. The calculation of the yielding behavior of a set of grains is described in Sec. II.D.

In all of these sections we give the full 3D geometric form for all texture equations. For many situations, it is not necessary to constrain the

dimensions of the texture equations, even when coupled continuum simulations possess a higher symmetry. Physically and computationally, when a deformation possesses a certain symmetry, the overall texture evolves with the same symmetry, as long as the initial texture possesses the same or higher symmetry.¹² We will be discussing axisymmetric continuum simulations in some detail with initially axisymmetric textures (no texture-equation constraint). The problem of a simulation with an initially lower-symmetry texture will not be handled here.

Finally, in texture calculations, the orientations of a set of grains are specified with regard to a particular reference frame. The frame used in all of these simulations is the "material" frame, a reference frame that rotates and translates with an element of material, relative to the laboratory frame.⁶ From the point of view of an observer in this material frame, the (continuum) material element experiences only deformation without rigid body motion (pure stretch). Constitutive modeling in this frame is convenient in that it can be performed without regard to rigid-body rotation.

B. Texture Evolution Calculations

A texture calculation models the effects of deformation on the orientations of a set of single-crystal grains. The orientation of each grain is initially specified; and texture evolution tracks the lattice rotation and subsequent orientation of each grain, as the grains respond to one or more deformation steps. It is mathematically convenient to specify a grain orientation by three Euler angles. If a grain is initially oriented with its (x,y,z) axes coincident with those of a reference frame (the "material" frame), any orientation can be specified by three successive rotations around the grain axes:

ϕ_1 = rotation around the z axis,

Φ = rotation around the new x axis, and

ϕ_2 = rotation around the new z axis.

A texture calculation, for a given deformation step, will first find the operating slip and/or twin systems in each grain and the amount of shear on each that satisfies the deviatoric strain components. This solution will then be used to determine the amount of lattice rotation. To find the set of systems, two assumptions are generally made:¹⁰

- 1) The plastic strain of each grain is equal to the macroscopic plastic strain of the set of grains (or, in our case, equal to the strain of the continuum code element corresponding to the set of grains). This assures strain compatibility.
- 2) The solution is a linear combination of shear on 5* independent systems, to accommodate 5 independent components of deviatoric strain.

Mathematically, assumption 2 allows many parts of the problem to be handled with 5 "vector" components for stress and strain, etc., rather than 9 tensor components. (Transformations between reference frames, of course, need to be handled with the proper rotation rule.) A convenient "5-space" form for deviatoric stress is:¹¹

$$S_1 = \frac{1}{\sqrt{2}} (S_{22} - S_{11}), \quad (2.1a)$$

$$S_2 = \frac{\sqrt{3}}{\sqrt{2}} (S_{33}), \quad (2.1b)$$

$$S_3 = \frac{1}{\sqrt{2}} (S_{23} + S_{32}), \quad (2.1c)$$

$$S_4 = \frac{1}{\sqrt{2}} (S_{13} + S_{31}), \quad (2.1d)$$

and

$$S_5 = \frac{1}{\sqrt{2}} (S_{12} + S_{21}). \quad (2.1e)$$

Exactly the same equations can be used to convert incremental deviatoric strain to a 5-space form. These transformations result in the equality:

$$\delta W = \sum_{i=1}^3 \sum_{j=1}^3 \delta \epsilon_{ij} S_{ij} = \sum_{k=1}^5 \delta \epsilon_k S_k, \quad (2.2)$$

*The deviatoric strain tensor is symmetric, with 6 non-zero components. The 3 diagonal components, however, sum to zero, so there are 2 independent diagonal components, plus 3 independent off-diagonal components, for a total of 5 independent components.

where δW represents the incremental plastic work. Equation (2.2) is particularly useful, since texture simulations typically require time-consuming calculations of plastic work.

The steps required to calculate texture changes for a single deformation step are then, for each grain:

- 1) Transform the incremental deviatoric strain tensor, $\Delta\epsilon$, from the material frame, m , to the crystal's reference frame, c , where T is the coordinate-transformation tensor for the particular grain:

$$\Delta\epsilon^c = T\Delta\epsilon^m T^t. \quad (2.3)$$

For this form, T_{ij} is the direction cosine between the j -direction in the material frame and the i -direction in the crystal frame. In terms of the Euler angles defined above,

$$T = \begin{bmatrix} \cos\phi_1\cos\phi_2 - \sin\phi_1\sin\phi_2\cos\Phi & \sin\phi_1\cos\phi_2 + \cos\phi_1\sin\phi_2\cos\Phi & \sin\phi_2\sin\Phi \\ -\cos\phi_1\sin\phi_2 - \sin\phi_1\cos\phi_2\cos\Phi & -\sin\phi_1\sin\phi_2 + \cos\phi_1\cos\phi_2\cos\Phi & \cos\phi_2\sin\Phi \\ \sin\phi_1\sin\Phi & -\cos\phi_1\sin\Phi & \cos\Phi \end{bmatrix}. \quad (2.4)$$

- 2) Transform the resulting strain tensor to the 5-component "vector" form.
- 3) Solve the 5×5 set of linear equations ($k = 1,5$) for the 5 incremental shear strains ($\delta\gamma^s$, $s = 1,5$):

$$\Delta\epsilon_k = m_k^s \delta\gamma^s. \quad (2.5)$$

The m_k^s are the symmetric Schmidt (geometrical) factors for the slip or twin system s , in the 5-component form. In tensor form, the symmetric Schmidt components are:

$$m_{ij}^s = \frac{1}{2}(b_i^s a_j^s + b_j^s a_i^s), \quad (2.6)$$

where i, j are directions in the crystal; $\hat{a}, \hat{b}$ are unit vectors, $\hat{a}$ is normal to the slip (or twin) plane, and $\hat{b}$ is in the slip (or twin) direction.

- 4) Find the "best" set of 5 systems. Mathematically, the best set should satisfy a minimum work criterion—the incremental work from shear on the 5 independent systems in the set should be a minimum, where

$$\delta W = \sum_{s=1}^5 \delta \gamma^s \tau^s \quad (2.7)$$

and τ_s is the CRSS of system s . In texture computations, the shear on a slip system is always normal to the stress surface. Hence, this criterion is formally equivalent to yield surface normality.¹⁴ Methods for determining the best set of systems are described below.

- 5) Once the best set of 5 systems is found, the incremental lattice rotation can be calculated. The form for the lattice rotation depends on whether deformation is by slip or twinning. For a slip system, the shear strain and the rotation involve the whole grain. The incremental lattice rotation $\delta \Omega_k$ due to all active slip systems becomes

$$\delta \Omega_k(\text{slip}) = \sum_{s=1}^n \delta \gamma^s M_k^s, \quad (2.8)$$

where n ($n \leq 5$) is the number of active slip systems. The M_k^s are antisymmetric Schmidt components, and both $\delta \Omega_k$ and M_k^s are in the 5-space ($k = 1, 5$) vector form.

For a twin system, the true variable is not incremental shear but the volume fraction of the twin band(s). However, it is possible to define an incremental shear $\delta \gamma^s$ for a twin system:¹⁰

$$\delta \gamma^s = v^s \gamma_0^s, \quad (2.9)$$

where v^s is the (variable) volume fraction and γ_0^s is the fixed twin shear. The incremental shears for slip and twin systems can then be

treated as mathematically similar quantities. For the twinned volume fraction of the grain, there are two successive lattice rotations: the rotation due to slip, plus an effective twinning rotation of 180° around the normal to the twinning plane. In tensor form, the lattice rotation due to twin system s is

$$\Omega_{ij,\text{twin}}^s = 2a_i^s a_j^s - \delta_{ij}, \quad (2.10)$$

where a is as defined for Eq. (2.6) and δ_{ij} is the Dirac delta function.

- 6) For slip, the new transformation tensor T' for the grain (from material frame to crystal frame) is

$$T'_{\text{slip}} = \{I + \delta\Omega_{\text{slip}}\} T, \quad (2.11)$$

where I is the identity matrix, T is the old transformation tensor, and the incremental lattice rotation $\delta\Omega$ is in tensor form. The rotation is defined relative to the material frame (see discussion in Sec. II.A above). The material frame rotates with the finite-element cell, so there is no rigid-body rotation included in Eq. (2.11).

For each active twin system, the twin band orientation is a result of both slip and twin rotations [Eqs. (2.8), (2.10)]. The new transformation tensor for a twin is

$$T'_{\text{twin}} = \Omega_{\text{twin}}^s \{I + \delta\Omega_{\text{slip}}\} T. \quad (2.12)$$

In effect, the activation of a twin system produces two smaller grains, with volume fraction v^s for the twin and $1-v^s$ for the untwinned part. There will be two different orientations for these two parts. Strategies for handling the extra orientations that result from twinning will be discussed in Sec. II.C.

It is desirable to make the new transformation tensors exactly orthonormal. Gram-Schmidt orthonormalization, or a similar process, should be applied to the quantity in the brackets above, before calculating T' . The three Euler angles for the new orientation can then be found.

In practice, there are two methods for solving the set of linear equations, Eq. (2.5), with the constraint, Eq. (2.7).¹⁰ The first is the Taylor method. The linear equations are solved directly, via a kind of steepest descent to the minimum work, using a SIMPLEX algorithm.¹⁵ This method has two advantages: a very simple setup (compare with the description of the single-crystal yield surface, below) and the property that different CRSSs can be varied during a deformation as temperature and/or strain-rate changes. The method is fast enough for stand-alone texture calculations or for an implicit code simulation with about 250 grains corresponding to one continuum element: ~1000–2000 grains/s, for each deformation step, on a CRAY-XMP. (Timing considerations for texture evolution in an explicit code will be described in more detail in Sec. III.)

The second method, the algorithm that we prefer in these calculations, is the Taylor-Bishop-Hill method. This procedure allows less flexibility with regard to the CRSSs and requires a more complex setup but is 10- to 100-times faster on a vector machine and considerably faster on a parallel machine as well. The Taylor-Bishop-Hill algorithm represents the “dual” to the “primal” SIMPLEX described above.¹⁶

To set up a Taylor-Bishop-Hill calculation, we consider that each slip system has a CRSS. The intersection of CRSSs from 5 independent systems is called a stress vertex. The set of systems at each vertex are the independent systems in the linear equation for the shears. However, rather than handle many sets of linear equations while running a texture simulation, we can find the independent systems, plus the stress at each vertex, beforehand. This setup procedure is called, “calculating a single-crystal yield surface (SCYS).” Tabulations of vertex stresses and lists of the systems at each vertex have been published for cubic materials and for special cases in hcp materials. An algorithm to generate the SCYS for the general case of an hcp or lower symmetry material is also available.¹¹

The Taylor-Bishop-Hill method requires calculation of the work at each SCYS vertex, rather than the solution of linear equations. The constraint in Eq. (2.7) transforms to finding the best vertex, the vertex where the (external) work is a maximum. In the 5-space representation:

$$\delta W_{\text{ext}} = \sum_{k=1}^5 \delta \epsilon_k S_k, \quad (2.13)$$

where S_k are the stress components of each vertex, and $\delta\epsilon_k$ are the incremental deviatoric strain components, all in the crystal frame. Once we have found the vertex giving the maximum work, we know the 5 systems involved; and we need to solve a single linear equation for the shears. Mathematically, this has been proved to be exactly equivalent to the SIMPLEX solution.¹⁶ The Taylor-Bishop-Hill method has the obvious advantage of requiring only the determination of the maximum of a 5-component dot product, calculated for each vertex, plus the solution of a system of 5 linear equations. The number of vertices is generally not too unwieldy; it ranges from a few dozen, for typical cubic materials, to a few hundred for hcp crystals, considering systems that are commonly seen in metals.

C. Special Cases

1. Nonuniqueness of Rotations. The preceding equations (2.1–2.13) describe a rate-independent, rigid-plastic response to (relatively) large-strain deformation. If enough independent slip systems exist, a stress solution and a rotation can always be found. However, there may be more than one possible rotation for a particular stress vertex. This nonuniqueness occurs when there is more than one set of 5 independent systems associated with a particular vertex. Crystal symmetry tends to produce this situation for a significant minority of vertices; the number of 5-system sets at a vertex can vary between 1 and 12, depending on the kind of crystal (and sometimes on the choice of operating systems and relative CRSSs).

Techniques available to resolve the nonuniqueness tend to require either the postulating of rate-dependent equations or the choice of a particular set of systems. Typically, rate-dependent calculations require a nonlinear, iterative solution, in 5 dimensions, for a rate-dependent form of the stress.¹⁷ In essence, the mathematical form "smoothes out" the vertex corners. This method has been used for texture evolution in implicit codes, for cubic materials. The equations are somewhat stiff,¹⁸ requiring many iterations. We have tried this method for a hexagonal material, as part of the implementation of texture evolution described in this work. The equations are indeed stiff and subject to convergence problems.

A method for system choice is based on the observation that different sets of slip systems that ideally give the same values for the stress are never quite identical (and hence, the vertex stresses are not quite identical) in a real

material. Experimentally, the particular set of systems operating in a particular small deformation cannot be predicted exactly. A reasonable approach, under these conditions, is to consider all possible sets of systems at a vertex, and to make a Monte-Carlo choice of one set.¹⁹ Computationally, this requires only a call to a random number generator to choose the set, plus some extra space to store the information about different sets of systems at a vertex. We use this method in our present calculations.

2. Twin Orientations. The preceding method for calculating texture changes is practical for materials where slip is the dominant deformation mechanism (e.g., copper, aluminum). The method is not practical for materials where twinning is important. The major difficulty in modeling twinning is the multiplicity of extra orientations produced when twin bands are formed. If a new grain orientation were added every time a twin was part of the 5-system shear solution, the number of orientations would quickly become unwieldy for most deformations in polycrystalline hcp materials, such as titanium, or for dynamic deformations of bcc materials such as iron.

To handle the extra orientations, we use an approach first suggested by Tome *et al.*²⁰ Instead of tracking individual grain orientations, we use an "Euler space," or 3D grid of Euler angles, rather than individual orientations. The range of Euler space needed for the grid can be reduced by considering crystal symmetry.⁹ (For example, the necessary range of ϕ_2 for an hcp material is only 0–60°.)

To obtain a Cartesian metric for Euler space, we use the coordinates $(\phi_1, \cos\Phi, \phi_2)$, rather than the standard Euler angles (ϕ_1, Φ, ϕ_2) .²¹ The individual grain orientations are replaced by weights in the grid boxes; the center of each grid box is treated as the average orientation. Twinning can then be efficiently handled by transferring weight fractions (usually to non-contiguous locations) within the grid. Slip is handled by moving the non-twinned contents of the grid boxes according to the associated small incremental rotations. Hence, slip resembles advection in an Eulerian framework.

With this "Eulerian" framework for the micromechanical part of these calculations, it is possible to calculate texture evolution involving both slip and twinning by the methods in Sec. II.B. The individual grid boxes can be treated as separate grains in Eqs. (2.1–2.13); the center of each box provides the Euler angles and transformation matrix required for the calculations. The

only difference between the treatment of individual grains and the treatment of the orientation grid is in the final step. In the case of individual grains, we calculate the new Euler angles from each new transformation matrix. In the case of the grid, we move all of the box contents according to the new transformation matrices and calculate the new set of orientation weights.

D. Yield-Surface Determination

Texture evolution tracks the orientations (or orientation weights; see Sec. II.C-2) of a set of single-crystal grains, as the grains rotate due to plastic deformation. Changes in the orientation distribution generally cause changes in the directionality of yielding. Experimentally, bulk anisotropic plastic response can be measured by probing a polycrystalline piece of material in different directions with small (but plastic) strains; the components of the yield stresses can then be determined. In practice, this procedure is complicated by material inhomogeneity and by changes in material properties due to the probe deformations themselves.

Mathematically, we can simulate this process, taking advantage of some of the mechanisms already built into a texture code. If we input a deviatoric strain plus a set of orientations, the texture code can calculate the stress for each single-crystal lattice in each orientation in the set. In terms of the method in Sec. II.B, we only need to find the stress vertex solution for the probe strain. There is no rotation calculation and no uniqueness problem. Furthermore, there is no computational problem with finite (vs. infinitesimal) strain probes—the choice of stress vertex depends on the direction of strain tensor components, not the absolute size. We can obtain a deviatoric stress for the polycrystalline sample as a whole by averaging over the set of orientations. However, the appropriate kind of yield determination is different for explicit and implicit codes.

In implicit codes, a deformation step (or a subdivision of a deformation step) is a relatively large strain. The step itself is the probe, and the single-crystal stresses can be averaged to obtain a rigid-plastic yield point. The computationally-intensive maximum-work calculation for the crystal stress must be done at each deformation step. This is a reasonable model for a typical forming simulation, where the deformation rate is low, the strain steps are large, and the number of strain steps is relatively small.

For dynamic deformation problems using explicit codes, the requirements for a constitutive model are different. First, we need an elastic-plastic, not a rigid-plastic yield determination. This suggests the use of a yield surface, or yield function, and the generation of a number of points to fit the function, not just a single point. However, performing maximum-work calculations for a number of strain probes and for every orientation in a set for each probe at each computational step would require prohibitive amounts of computer time, even with vector and/or massively parallel machines. Fortunately, such time-consuming calculations do not make physical sense. Changes in grain orientations tend to require relatively large strains (from ~0.5% to a few percent), and significant changes in anisotropic yield tend to require even larger strains (several percent, in general: hundreds or even thousands of times the strain of a computational step in a dynamic deformation). Hence, a yield function, once fitted, can be used as part of the material model for a great many computational steps before it needs to be recalculated. We describe the general process of computing points on such a yield surface below. The choice of functions to fit these points, and the determination of yield parameters will be discussed in Sec. III.A.

Mathematically, we first need a set of strain probes; the calculated response to each probe will be a stress tensor. We then can fit an analytic or piecewise function to the set of stresses. The number of stress components needed for the fit is limited by the dimension of the intended simulation: two diagonal components and only one nonzero off-diagonal component are needed for a 2D simulation; three off-diagonal components are needed for a 3D simulation. The range of the input strains is similar.

A convenient way to distribute the strain probes (and hence, to distribute the resulting stress points) is to use the form of a stress or strain "space" introduced in Sec. II.B. Each independent component of the stress or strain tensor corresponds to a direction. As an example, we consider axisymmetric cases, which can be addressed with 2D codes. The corresponding stress space is 3D: two independent diagonal components and one off-diagonal component. We can construct an orthogonal set $\{S_x, S_y, S_z\}$, with S_x and S_y in the π -plane, the plane in which the diagonal deviatoric stresses are at 120° to each other. One possible linear combination of tensor components is:

$$\begin{pmatrix} S_x \\ S_y \\ S_z \end{pmatrix} = \begin{pmatrix} \frac{-\sqrt{3}}{2} & -\sqrt{3} & 0 \\ \frac{3}{2} & 0 & 0 \\ 0 & 0 & \sqrt{3} \end{pmatrix} \begin{pmatrix} S_{11} \\ S_{22} \\ S_{12} \end{pmatrix}. \quad (2.14)$$

This representation has a number of advantages. It allows for easy fitting of a yield function and easy visualization of stress paths on the yield surface. For the corresponding 3D strain-"space," the probes can be distributed on the surface of a sphere. Once the distribution of strain probes is determined, the probes can be transformed into tensors. The set of strain tensors can then be used to calculate a yield surface, as follows:

For each probe, take the set of grain orientations that represent the textured material. For each orientation:

- 1) Transform the probe strain tensor from the material frame to the crystal reference frame [Eq. (2.3)].
- 2) Transform the resulting strain tensor to the 5-component form [Eqs. (2.1)].
- 3) Find the stress vertex that gives the maximum work [Eq. (2.13)].
- 4) Transform the 5-component form of the stress to the tensor form [Eqs. (2.1)].
- 5) Transform the stress tensor, S , from the crystal frame, c , to the material frame, m , where T is the coordinate-transformation tensor for the particular grain [compare with Eq. (2.3)]:

$$S^m = T^t S^c T. \quad (2.15)$$

- 6) Average the stress tensor over all the crystal orientations. (Alternately the corresponding "vectors" can be averaged; the results for tensor and "vector" averaging are equivalent.)

This process gives one stress tensor (or stress point, if the "vector" form is used) for each strain probe. The collection of points can be viewed as defining a yield surface, albeit a very general one, not restricted to a particular analytical form. The generality of this step is useful when dealing with textured yield surfaces, since there are particular difficulties in finding any single analytical form that suits a range of different materials as well as different amounts of texturing (including isotropy).

- 7) Choose a form for a yield function, and find the yield-surface parameters by fitting the function to the stress tensors (or points).

The characteristic features of the yield surface for a particular material depend heavily on the crystal symmetry, as well as on the grain orientation distribution. The yield surface for a strong texture, such as in heavily-rolled or extruded material, can differ significantly from a simple shape. For example, in rolled copper (fcc crystal structure), the π -plane yield surface has definite facets;²² for α -titanium (hcp crystal structure), the corresponding yield surface resembles a flattened ellipse.²³ The designing of suitable yield functions (including both global and piecewise forms) for texture-dependent anisotropy is discussed in detail in Ref. 6. In the next section, we will discuss the adaptation and implementation of a variant of Hill's¹ quadratic function, a form that has been widely used in anisotropic modeling.

III. IMPLEMENTATION OF ANISOTROPIC EVOLUTION

So far, we have discussed how a texture code can be used to track changes in the grain orientation distribution or to generate a yield surface. The former calculation requires input in the form of a deformation tensor (or set of tensors), plus a set of initial grain orientations (or weights in a grid of orientations). The output is a new set of orientations (or orientation weights). The latter requires input in the form of a set of deformation tensor probes, plus the input orientation distribution. The output is a set of stress tensors (or "points," in the "stress space" conventions of Sec. II.D). These two processes are often sufficient for a stand-alone texture code. However, in this

work we want to use a texture code as part of a constitutive model for the dynamic deformation of a material. Moreover, this model is to be updated as deformation proceeds.

For the updating of a path-dependent process, such as texture change during a dynamic deformation, we have coupled the texture code to an explicit continuum code. The coupling occurs at two places. First, the deformation for each material element needs to be communicated from the continuum code to the texture code. This information is used to update the preferred orientations in the different parts of the material. Second, information about anisotropic yielding, as a function of preferred orientation, needs to be communicated from the texture code to the continuum code. For computational efficiency, it is desirable to treat anisotropic yielding in terms of a specified function and to communicate the fitted parameters from the texture code to the continuum code.

The gradual nature of anisotropic change allows great reductions in the computational burden of a texture-code material model. Deformations can be collected and integrated over time, and updates of the preferred orientation and yield function need not be done very often. We describe the timing of these updates and other details of our texture code/continuum code interface in Sec. III.C below.

We handle the connections between continuum code and texture code on the level of a computational cell, where the cell consists of one or more element in a finite-element continuum code. A particular cell is represented in the texture code by an orientation distribution—a sample of many single-crystal grains (or weighted orientations in a grid), with each grain, or weighted orientation, subject to the deformation of the computational cell. The yielding behavior of this polycrystalline sample is represented in the continuum code by an analytic function, with fitted parameters updated during the simulation by the texture code. For such a two-level simulation, it is necessary to adapt routines in the texture code (primarily for speed) and in the continuum code (primarily for anisotropic yield). Details of these modifications are described in Secs. III.A and III.B below.

A. Texture-Code Modifications

Texture evolution and yield-surface determination need to be adapted and/or fine-tuned if they are to be run as part of a continuum code material

model. We discuss several considerations for texture-code modifications, including texture-system size, computational strategies for reducing computer time, implementation of a particular yield function, and tying the material model to existing constitutive relations in the continuum code.

Any texture calculation operates on a set of N single-crystal grains (or weighted orientations that function like grains, except for the final orientation update). Texture calculations are computationally very intensive; and the number of grains (or orientations), N , is central to virtually all of these calculations. This number needs to be small enough to be tractable and large enough to give approximate yield isotropy for a random distribution, as well as adequate resolution of yield anisotropy for other likely distributions. The minimum acceptable N will depend on crystal structure, the relative sizes of different CRSSs, and the range of textures expected. For our simulations of (hexagonal-close-packed) titanium, both strong textures and approximate yield isotropy were expected to occur. We used $N = 512$ for these calculations; this number of orientations gave yield isotropy to $\pm 1\%$ for a random distribution.

The number of grains is virtually always the largest operating array size in these calculations. For nonparallel machines, and vector machines in particular, it is desirable to have an innermost loop over N grains for texture-code operations. Such a loop over grains is involved in the transformation of Euler angles to a rotation matrix (and its reverse), in the transformation of the strain tensor (or strain probe) from material to crystal frames, in the maximum-work calculation, in the lattice-spin calculations, etc.

The evolution calculations involve linear-equation solving. We have used a standard set of algorithms—lower-upper decomposition with Cholesky factorization and partial pivoting, etc. The coefficients (Schmidt components) are mathematically well-behaved and do not appear to require scaling. However, the linear equation solutions can take up a large fraction of the computer time for texture calculations. Accordingly, it is worth optimizing the linear equation routine. For vector machines in particular, rewriting a linear-equation solver to place a loop over grains as the innermost loop in every operation can dramatically reduce the overall computer time for texture evolution. We note that, for parallel machines, the above strategies for reducing CPU time need to be reversed, with the loop over grains replaced by a distribution among parallel processors.

The yield-surface determination described in Sec. II.D requires no linear-equation solving. Instead, the texture-code calculations simply find the deviatoric stress (averaged over N grains) for each of M strain probes. This set of stresses is amenable to fitting with a variety of functions, either global or piecewise. An appropriate functional form depends primarily on the crystal structure and secondarily on the kinds of texture.

Probably the simplest analytical function for anisotropy is the quadratic form proposed by Hill.¹ In terms of deviatoric stresses S_{ij} we have:

$$f = \frac{1}{2} \left\{ (G+H) S_{11}^2 + (F+H) S_{22}^2 + (F+G) S_{33}^2 - 2H S_{11}S_{22} - 2G S_{11}S_{33} - 2F S_{22}S_{33} \right. \\ \left. + 2N S_{12}^2 + 2M S_{13}^2 + 2L S_{23}^2 \right\} - \sigma^2 = 0. \quad (3.1)$$

This function is conveniently analytical and can be used for 3D problems. For a 2D problem, recalling that the trace of the deviatoric stress tensor is zero, Hill's yield function reduces to

$$f = \frac{1}{2} \left\{ U S_{11}^2 + V S_{22}^2 + W S_{12}^2 + 2X S_{11}S_{22} \right\} - \sigma^2 = 0. \quad (3.2)$$

A more general quadratic surface, allowing additional cross-terms, gives

$$f = \frac{1}{2} \left\{ U S_{11}^2 + V S_{22}^2 + W S_{12}^2 + 2X S_{11}S_{22} + 2Y S_{11}S_{12} + 2Z S_{22}S_{12} \right\} - \sigma^2 = 0, \quad (3.3)$$

where the new parameters Y and Z are rotational coefficients describing rotations in the S_{11} - S_{12} and the S_{22} - S_{12} planes, respectively. This form of the function is particularly useful for a material frame that rotates with the finite element. For a 2D simulation, using the three stress "directions" S_x, S_y, S_z in Eq. (2.14), this function becomes

$$f = A S_x^2 + B S_y^2 + C S_z^2 + 2D S_x S_y + 2E S_x S_z + 2F S_y S_z - \sigma^2 = 0. \quad (3.4)$$

For the special case where $A = B = C = 1$ and $D = E = F = 0$, Eq. (3.4) simplifies to the equation of a sphere:

$$f = S_x^2 + S_y^2 + S_z^2 - \sigma^2 = 0. \quad (3.5)$$

This is exactly equivalent to the von Mises yield surface.

There are some problems, however, in simply fitting texture-code yield points to the quadratic form in Eq. (3.4). Our simulations involve strongly textured hexagonal materials, where the yield points do resemble an ellipsoid, but a flattened one, not a quadratic shape. As a result, standard fitting routines may produce questionable results. At best, quadratic fitting routines tend to underestimate the eccentricity.

We use a faster and more stable, if less elegant approach. Consider a 2D simulation, where the "stress space" has three dimensions. The calculation of parameters for the yield function takes the form of a search in this stress space, first for the maximum "vector," then for the "vector" with the minimum component orthogonal to the first (i.e., major and minor axes), then for the maximum component in the direction orthogonal to the first two axes. The six parameters in Eq. (3.4) can be determined from these calculations. This approach can be extended to the 5D stress space for 3D simulations, as long as no cross-products are used.

Finally, for a general texture code, the absolute scale of the stresses (and hence the scale of any yield surface) is determined by the values of the CRSSs. To connect the texture simulation with experimental bulk properties, it is most efficient to use the texture code only for the shape—the anisotropy—of the yield surface. We note that the yield points for a randomly oriented set of grains should give a sphere, approximately: Eq. (3.5), with σ as the radius. Any subsequent yield function can be normalized to a value of 1 by multiplying the components of the stress points by the random-texture value of $1/\sigma$ before fitting. The magnitude of the yield stress, plus hardening and thermal softening, can then be handled on the macroscopic continuum-code level.

B. Continuum Code Modifications

A continuum code using only isotropic material models will generally need modifications in several different routines to handle plastic anisotropy. We will describe these changes briefly in the section below; for a detailed description, see Ref. 6. With the code interface described in Sec. III.C,

anisotropic evolution can also be added to the material model. We have used EPIC,²⁴ a finite-element, explicit Lagrangian code, as our continuum base. For a high-rate time-dependent code such as EPIC, it is not necessary to construct a stiffness matrix. It is necessary for the code to have a more generalized elastic-plastic capability. The emphasis of this modeling is on plasticity as being the dominant feature of high deformation applications. For these applications, the isotropic form of Hooke's law is used, assuming an isotropic elastic loading out to an anisotropic yield surface.

There are three kinds of continuum code modifications that need to be implemented for calculations using anisotropic plasticity (some of these forms are already used for other kinds of material models). First, the calculation of element rotations is sensitive to the kind of stress rate used. For anisotropic calculations, the Green-Naghdi stress rate (or polar decomposition)²⁵ is essential; the Jaumann rate can give spurious results for this kind of material.²⁶ Second, updating of the stress state is best done with an associative flow algorithm (see Ref. 6 for an anisotropic version of Nemat-Nasser's²⁷ efficient method). The fast radial-return method is not appropriate here, since return would not be normal to the anisotropic yield surface, in general and would produce a nonassociative flow rule.

Finally, all stress updates in the continuum code are done in the material frame⁶—a frame rotating with the finite element. This simplifies equations in both texture and continuum codes, since both evolution and the yield function are calculated in the material frame. We update the stress in the material frame and transform the resulting tensor back to the laboratory frame afterwards. Use of the material frame requires the storage of extra internal variables for the continuum code: the stretch tensor, plus the integrated rotation angle of the finite element for a 2D simulation, and three integrated angles for a 3D simulation.

C. Continuum Code/Texture Code Interface

We have described algorithms to track changes in grain orientations due to plastic deformation and algorithms to generate and fit the corresponding yield surfaces. We have also described algorithms to handle anisotropy in a continuum code. What is needed now is to connect the microscale evolution of anisotropy with the continuum simulation. This connection takes the form of a code interface.

A continuum code/texture code interface needs to communicate two types of information: the deviatoric strain history of a segment of material (from the continuum code) and the parameters of the yield function (from the texture code). To construct this interface, we need to address two main issues: the size of the computational cell that corresponds to a texture sample of N grains and how often and in what form the codes communicate.

The computational cell need not necessarily be a single finite element. The main requirements for computational-cell size are that deformation and rotation of elements within a piece of material be reasonably homogeneous. The desired degree of homogeneity will depend on the nature of the problem and mesh geometry and the kind of detail required. For a number of 2D axially symmetric dynamic problems, including Taylor cylinder impacts and explosive forming, we have found that a quadrilateral cell, enclosing four crossed-triangle finite elements, is reasonable.

The nature of texture changes will influence how often the two codes need to communicate. What makes these calculations possible in an explicit code is exploitation of the relative "slowness" of texture change: for processes dominated by slip, significant orientation changes typically require ~10–20% strain or more. In addition, the yield function is, in general, not too sensitive to small preferred-orientation changes. Hence, we can integrate the deviatoric strain tensor for a computational cell until the deformation is large enough to recalculate the yield function. It is necessary, however, to provide for subdivision of the strain steps when the evolution routine is called, since the texture evolution equations are for infinitesimal, not finite strains. Texture evolution is typically calculated as a succession of small (2.5–5% strain) finite steps.

A flow chart for the interface is shown in Fig. 1. Information from the continuum code is communicated in vectors of 64 finite elements. For our purposes, these elements are treated as vectors of 16 computational cells (quads). Four control constants are set beforehand: ECALL, the minimum incremental equivalent strain in any one cell required to send the vector to the texture routine; EFIT, the minimum incremental equivalent strain required to recalculate the yield function for any cell; ESTEP, the maximum equivalent strain per evolution step (for subdivision of strain steps); and EMIN, the minimum equivalent strain to calculate texture evolution in a cell.

Fig. 1. Flow chart for the continuum code/texture code interface. Double lines enclose the texture-code operations.

Shared variables between the texture code and a 2D continuum code include, for each computational cell, three plastic incremental deviatoric strain tensor elements (Δe_{ij}^p or $\dot{e}_{ij}^p \Delta t$) in the material frame, an incremental equivalent plastic strain (Δe_{eq}^p), and the current yield surface parameters. The evolution routine is not called unless Δe_{eq}^p for at least one computational cell in the vector is greater than ECALL. (It is necessary to zero out the incremental strains in "fractured" cells before testing for ECALL). If the evolution routine is called, the quantity EINC (equivalent strain since the last yield-surface calculation) will be incremented for each cell. However, only

cells with an Δe_{eq}^p greater than EMIN will have evolution calculated. If evolution is calculated, the strain increments will be divided into strain steps, with the equivalent strain no greater than ESTEP. For an evolved cell, the new Euler angles will be calculated and stored, and the incremental strain tensors and Δe_{eq}^p will be zeroed out. For any cell with EINC greater than EFIT, new yield function parameters will be calculated; and EINC will be zeroed out.

These procedures keep track of the changes in the orientation distribution as the calculation proceeds, and the yield function parameters are updated when warranted. Also, small strain increments are not lost, even if evolution is not calculated, since the tensors and equivalent strains are incremented until used.

D. Initialization

The previous discussion was concerned only with ongoing texture evolution and yield-surface updating. These routines also require initialization. We can divide the initialization procedure into four parts: interface control, slip system information, evolution setup, and yield-surface setup. The code interface was described in the previous section. To initialize the interface, we need to set the computational cell arrays of incremental strains to zero and assign the strain measures ECALL, EFIT, ESTEP, and EMIN.

Slip system information includes the Schmidt factors and the CRSSs, as well as the SCYS information (see Sec. II.B). Schmidt factors are linear combinations of products of slip and slip-plane-normal components. For the most important slip systems of a particular crystal structure, the necessary factors can be computed once and stored. The CRSS for each kind of slip system can be read in at program initialization. The SCYS is read in at initialization; it consists of the individual stress components of each yield vertex, a list of the sets of five independent systems at that vertex, and the list of systems in each set.

To start an evolution calculation for any computational cell, we need slip and twin system information and an initial set of grain orientations (or a grid of orientations, plus orientation weights). We represent each grain orientation by the 3 Euler angles (ϕ_1, Φ, ϕ_2); the orientation of a grid block is the 3D Cartesian center of the block, in terms of two Euler angles and an Euler

angle function $(\phi_1, \cos\Phi, \phi_2)$. It is possible to set up different initial textures in different computational cells, but the simplest and most generally physical initial condition is to read in the set of Euler angles or the orientation weights for one texture sample and then to copy the texture information to all computational cells.

To start the continuum code, we need an initial set of yield-surface parameters. To calculate the initial yield surface, we need several kinds of information, all to be set up as part of the initialization: the set of grain orientations, slip system information, strain probes, and a normalization constant [see the discussion following Eq. (3.5)]. Given the slip system information and the strain probes, we can calculate a yield surface from a random distribution of grain orientations. The resulting isotropic yield function in "stress space" should be approximately spherical; the radius is the desired normalization constant for stresses. For all subsequent yield-surface calculations, all stresses will be divided by this constant. The initial yield parameters, for the initial orientations, can then be calculated; and the initialization of the material model is complete.

IV. TEST PROBLEMS

In this section, we examine the consequences of texture evolution for different loading paths, with different initial textures of one particular material. These test problems include large strain, high strain-rate processes: uniaxial stress, both compression and tension, plus explosive forming.

The material to be modeled has the general characteristics of hcp titanium and zirconium.³⁻⁵ For this class of materials, a single crystal has a unique strong direction, the c-axis, or the 6-fold symmetry axis of the crystal. Typical textures of polycrystalline material after initial processing (e.g., rolling, and extrusion) also have a unique strong direction and very strong plastic anisotropy—as large as 2:1 for the ratio of yield strengths in different directions. If the initial processing is axisymmetric, the texture and the yield anisotropy of the processed materials are also axisymmetric.¹²

In general, texture evolution is a 3D problem, and the texture calculations described in earlier sections of this work are fully 3D. However, to simplify our coupled calculations, the deformations in the test problems are symmetric about a central axis. This axisymmetry is compatible with

texture and texture evolution if the initial texture is at least axisymmetric. Experimentally, textures evolve under deformation to match the symmetry of the principal strains. If the initial texture has symmetry equal to or higher than axisymmetric and the principal strains are axisymmetric, the evolving texture will be axisymmetric.¹² To maintain compatibility of deformation symmetry and texture symmetry, the two different initial textures used in these simulations are both symmetric about a central axis. This axisymmetry enables us to use a 2D continuum code rather than a 3D code in the coupled simulations. The details of our material model, including both texture-code and continuum quantities, are described below.

A. Material Model

For texture-code calculations we use the single-crystal properties typical of hcp titanium at moderately high temperatures (~ 0 to several hundred degrees C) and very high strain-rates ($>10^4 \text{ s}^{-1}$). The slip and twin systems characteristic of α -titanium at various temperatures, plus CRSSs for these systems, have been summarized by Conrad.²⁸

The changes in dominant systems for shock and high strain-rate conditions for a number of metals, including hcp structures, are discussed by Meyers and Murr²⁹ and Murr.³⁰ The general effect of high strain-rates is to shift the dominant systems from higher-temperature to lower-temperature modes. Twinning, in general, is promoted at high strain-rate and shock conditions, or low temperatures. Hence, we model the system activity at moderate temperatures and favor twinning (lower CRSS) over the high-temperature (or low-purity) slip systems (higher CRSS).

The active slip and twin systems and their CRSSs are given below in Table I. Prismatic slip, in the $(10\bar{1}0)$ plane and the $(11\bar{2}0)$ direction, is generally agreed to be the easiest slip system for hcp materials for a wide variety of conditions. The CRSSs are normalized, as ratios to the value for prismatic slip. Basal slip, in the (0001) plane and $(11\bar{2}0)$ direction, is more difficult, and relatively uncommon in hcp materials like titanium and zirconium;^{3,28} it is ignored here. Neither prismatic nor basal slip accommodate deformation in the c-axis direction. The easiest mode for deformation in tension along the c-axis direction is tensile twinning. A comparable compressive twinning mode is generally not seen at high strain-rates;³¹ compression with a component

TABLE I. Active deformation modes and CRSSs.

Designation	Plane	Direction	CRSS ^a
Prismatic slip	(10 $\bar{1}$ 0)	$\langle 11\bar{2}0 \rangle$	1
Pyramidal $\langle a+c \rangle$ slip	(10 $\bar{1}$ 1)	$\langle 11\bar{2}3 \rangle$	7
Tensile twin	(10 $\bar{1}$ 2)	$\langle \bar{1}011 \rangle$	5

^aNormalized to the CRSS for prismatic slip.

along the c-axis is difficult, but can be accommodated by the pyramidal $\langle a+c \rangle$ slip system.³²

Experimental values of CRSSs are generally available only as approximate ranges for particular conditions; and ratios of CRSSs for different kinds of systems are only approximate, particularly for comparison of twin to slip systems. For these simulations, we have chosen ratios of CRSSs both to reflect the difficulty of c-axis deformation and to fit typical experimental values for R-values and yield anisotropy to the texture of a cross-rolled sample.

We use two initial anisotropic textures, one modeled after the preferred orientation of heavily cross-rolled titanium,⁴ and one typical of extruded titanium.³³ Basal (0002) pole figures for these textures are given in Figs. 2 and 3. The contours represent a density plot for the c-axes, or hard directions, of individual grains. For cross-rolled titanium, the c-axes cluster around a direction pointing out of the plane of the sheet; for extruded titanium, the c-axes point radially.

The parameters of the yield functions for these two textures were obtained with the texture code (see Secs. II.D and III.A) and the CRSSs from Table I, using Eq. (3.4). For the cross-rolling texture, we have $A = 3.74$, $B = 1.09$, $C = 0.52$. The strong to weak yield-stress ratio is 1.85 in transformed stress space (S_x, S_y, S_z), which corresponds to ~ 1.68 in a laboratory frame (i.e., S_{rr}, S_{zz}, S_{rz}), where z is the axial direction and r is the radial direction. The hard direction is *parallel* to the z -axis. For the extrusion texture, we have $A = 0.83$, $B = 2.84$, $C = 0.96$. The strong to weak yield-stress ratio is approximately the same as for the cross-rolling texture, but the hard direction is now *perpendicular* to the z -axis. For both textures, the parameters D-F for

Fig. 2. The (0002) pole figure for the cross-rolling initial texture. Intensities in multiples of a random distribution (m.r.d.); contours are 3.5, 7.0,...m.r.d. Stereographic projection; projection plane is parallel to the plane of the metal sheet.

Fig. 3. The (0002) pole figure for the extrusion initial texture. Intensities in multiples of a random distribution (m.r.d.); contours are 1,2,...m.r.d. Stereographic projection; projection plane is normal to the axis of the extrusion.

the cross-terms in Eq. (3.4) were negligible. The mapping of these initial textures to computational cells for the continuum simulations is described in the next section.

For these simulations, hardening and thermal softening are handled by a macroscopic flow stress model. The texture modeling provides the shape of the yield surface [anisotropy parameters A , B , etc., in Eq. (3.4)], and the flow stress model [σ in Eq. (3.4)] provides the scale. The evolution of σ is provided by the Johnson-Cook model³⁴ for titanium with an initial yield strength of 200 MPa. For all of these simulations, a Mie-Grüneisen equation of state was used.

B. Uniaxial Stress

We first examine the consequences of texture evolution for the simple case of uniaxial stress, with no work-hardening or thermal softening. Four simulations were performed: (1) Compression, with the extrusion initial texture; (2) Compression, with the cross-rolling initial texture; (3) Tension, with the extrusion initial texture; and (4) Tension, with the cross-rolling initial texture.

Each simulation used a single computational cell consisting of four crossed-triangle finite elements, with the cell texture modeled by a sample of 512 oriented grains. The symmetry axis for each initial texture corresponded to the loading axis for the continuum simulation. A call to the texture evolution routine was triggered, and the yield function was recalculated for every incremental equivalent strain of 0.01. The total incremental equivalent strain for each of these simulations was 1.5.

The deviatoric stress component histories for these four cases are shown in Fig. 4a–b. The corresponding histories for the yield-function parameters A , B , and C are shown in Fig. 5a–b. (The rotational coefficients D – F are not shown; these were of magnitude 0.1 or less, and were not significant.) It can be seen that significant changes in yield anisotropy only occur for two of these cases: compression, with an initial extrusion texture, and tension, with an initial compression texture. In both cases, the initial anisotropy essentially disappears, then the opposite anisotropy develops.

These results agree with the nature of texture evolution seen in experiments on hcp materials. The key factor in texture change and the corresponding anisotropy change is twinning, which can substantially change

Figs. 4a-b. Material frame stress components versus equivalent strain for the uniaxial stress simulations.

Figs. 4c-d. Material frame stress components versus equivalent strain for the uniaxial stress simulations.

Figs. 5a-b. Yield function parameters A, B, C versus equivalent strain for the uniaxial stress simulations.

Figs. 5c-d. Yield function parameters A, B, C versus equivalent strain for the uniaxial stress simulations.

a preferred orientation in less than 20% equivalent strain. If there is tension along the c-axes, which promotes "tensile" twinning, the material will twin and the volume fraction twinned will effectively rotate by $\sim 85^\circ$ —the c-axes of parent and twinned material will be approximately perpendicular. For an extrusion texture, the majority of crystal c-axes point radially; twinning occurs in compression (tension along the c-axes). For a cross-rolling texture, the majority of crystal c-axes point along the axis; twinning occurs in tension.

For an extrusion texture in tension and a cross-rolling texture in compression there is compression along the c-axes, and the material can shear along the c-axes with difficulty but cannot twin. These initial textures are stable; previous processing resulted in the strong initial textures, and subsequent deformation represents continued processing along the same loading path. The subsequent deformations will not change the preferred orientation in any significant way.

C. Explosive Forming

The preceding uniaxial tests help establish that the coupled simulations give physically reasonable results even for the extreme case of fast texture evolution due to twinning. We now examine the consequences of texture evolution for a case of explosive forming.

A schematic of the initial geometry is illustrated in Fig. 6. The setup consists of a titanium sheet 2.5 mm thick backed by a layer of high explosive 1.4 mm thick. The metal sheet initially rests on a ceramic die having a concave parabolic shape described by the equation: $z = -3.5 \text{ cm} + 0.389 \text{ cm}^{-1}r^2$; this problem is symmetric around the z-axis. The high explosive is initiated with a plane wave that then presses the sheet into the die. In these simulations, the die is treated as a rigid body.

Three simulations were performed; one for each strong initial texture and one for a random initial texture with a correspondingly isotropic yield surface. For all of these simulations, the full Johnson-Cook model, incorporating hardening and thermal softening, was employed.

Each simulation used 160 computational cells—4 cells across the thickness of the metal sheet, and 40 cells radially, with each cell consisting of four crossed-triangle finite elements. Each initial texture was mapped onto

Fig. 6. Initial axisymmetric geometry of the explosive-forming problem.

the computational cells so that the symmetry axis for the texture was parallel to the through-thickness direction of the metal sheet. Hence, for the cross-rolling initial texture, the hard direction for the sheet was normal to the sheet; for the extrusion initial texture, the soft direction was normal to the sheet.

A call to the texture evolution routine was triggered, and the yield function recalculated for an incremental equivalent strain of 0.05. The maximum equivalent strain step for the texture evolution routine was 0.02. That is, if the incremental equivalent strain for a computational cell was greater than 0.02, the texture evolution calculation would be divided into two or more steps. Texture evolution increased the computer time for an explosive-forming simulation by about a factor of 5, compared to an anisotropic simulation without evolution.

These simulations were also performed for a finer resolution of the texture changes. The call to the texture routine and the yield-function recalculation were set at an incremental equivalent strain of 0.025; the

Figs. 7a-b. Profiles of the final simulated geometry of the explosive-forming problem, for different initial textures, for a region within a radius of 2 mm around the apex of the die. Simulations with texture evolution (a) (changing yield anisotropy) versus simulation without texture evolution (b) (constant yield anisotropy) are shown.

maximum equivalent-strain step for the texture routine was 0.01. The results for final shape of the metal sheet were not significantly different from the coarser equivalent-strain parameters of 0.05 and 0.02, respectively.

The final shapes of the metal sheet for the three different initial textures (cross-rolling, isotropic, extrusion) are shown in Fig. 7a. For comparison, the final shapes for these initial textures, but without any texture evolution, are shown in Fig. 7b. The constant-anisotropy calculations were described in more detail in a previous article.⁶

It can be seen that the final shape of the metal sheet is significantly influenced by yield anisotropy—both by the initial anisotropy and by subsequent evolution. In the absence of texture evolution, the extrusion texture produces severe localization at the apex of the sheet. Metal sheets with isotropic and cross-rolling textures do not show significant localization, but the rolling texture produces significantly less thinning than the isotropic texture.

When texture evolution is added, a metal sheet with an initial extrusion texture is considerably more stable; no significant localization is seen during explosive forming. However, a sheet with an initial cross-rolling texture is more stable yet, with less thinning at the apex than sheets with either isotropic or extrusion textures. Plastic anisotropy can have a powerful influence on localization, but changes in yield anisotropy with deformation should be taken into account when evaluating the stability of a particular material.

V. SUMMARY AND CONCLUSIONS

We have developed a method for modeling changes in plastic anisotropy during high-rate deformation, where the anisotropy is due to crystallographic texture—preferred orientation of the single-crystal grains in a material. The method consists of a micromechanical subroutine (a texture code) coupled to an explicit Lagrangian code.

The coupling is through the deviatoric strain tensor (from the continuum code to the texture code) and the yield function (from the texture code to the continuum code). Each computational cell in the Lagrangian code supplies a strain tensor to the texture code. The texture code calculates deformations—shear and rotation—of a set of single-crystal grains by the

mechanisms of slip and twinning within each grain. For a particular strain tensor, grains of different orientations rotate by different amounts. The texture code rotates the grains, and tracks the grain orientations.

When the grain orientations for a particular computational cell are updated, the texture code can also update the anisotropic yield function for the cell. To obtain a yield function, the set of oriented grains is "probed" with different deviatoric strain tensors. The result is a set of deviatoric stress tensors, which are fitted to a quadratic anisotropic form. The parameters of the updated yield function then can be used by the continuum code as part of its constitutive model.

Texture changes require relatively large strains, at least a few percent for any significant preferred-orientation changes. Accordingly, it is not necessary for an explicit continuum code to call the texture code at each step. Instead, the strain tensors are accumulated over a number of time steps until a predetermined maximum equivalent strain triggers a call to update the texture. We find that the coupled texture/continuum codes increase the computer time by about a factor of five, compared to simulations without texture evolution.

In separate studies, we have investigated the changes in the yield function for uniaxial stress—compression and tension—applied to textured titanium. The results are in accord with experiments; the texture, and hence the anisotropy, will only change significantly if the loading path promotes twinning. The final state of the material is generally not isotropic; if there is a significant amount of twinning, a new anisotropy will evolve that is the reverse of the original state.

We have also investigated the effects of changes in the yield function for explosive forming of titanium sheet. The final shapes are significantly affected by yield anisotropy; cross-rolled material is more stable against localization than isotropic material, while extruded material is less stable. Allowing the anisotropy to evolve with deformation reduces localization effects, compared with calculations using constant anisotropy.

ACKNOWLEDGMENTS

This work was supported by the Department of Defense (DoD) and the Department of Energy (DOE), through the Joint DoD/DOE Munitions Technology Development Program.

REFERENCES

1. R. Hill, "A Theory of the Yielding and Plastic Flow of Anisotropic Metals," *Proc. Roy. Soc. London A* **193**, 281-297 (1948).
2. L. Bourne and R. Hill, "On the Correlation of the Directional Properties of Rolled Sheet in Tension and Cupping Tests," *Phil. Mag.* **41**, 671-681 (1950).
3. R. G. Ballinger and R. M. Pelloux, "The Effect of Anisotropy on the Mechanical Behavior of Zircaloy-2," *J. Nucl. Mater.* **97**, 231-253 (1981).
4. F. Larson and A. Zarkades, "Properties of Textured Titanium Alloys," Metals and Ceramics Information Center Report MCIC74-20 (June 1974).
5. W. F. Hosford and R. M. Caddell, *Metal Forming: Mechanics and Metallurgy* (Prentice-Hall, Englewood Cliffs, New Jersey, 1983).
6. P. J. Maudlin and S. K. Schiferl, "Computational Anisotropic Plasticity for High-Rate Forming Applications," Los Alamos National Laboratory document LA-UR-93-1563, submitted to *Computer Methods in Applied Mechanics and Engineering*.
7. R. Hill, S. S. Hecker, and M. G. Stout, "An Investigation of Plastic Flow and Differential Work Hardening in Orthotropic Brass Tubes Under Fluid Pressure and Axial Load," *Inter. J. Solids and Struct.* **21**, 2999-3021 (1994).
8. K. K. Mathur and P. R. Dawson, "On Modeling the Development of Crystallographic Texture in Bulk Forming Processes," *Int. J. Plasticity* **5**, 67-94 (1989).
9. J. Gil Sevillano, P. van Houtte, and E. Aernoudt, "Large Strain Work Hardening and Textures," *Prog. Mater. Sci.* **25**, 69-412 (1981).

10. P. van Houtte and F. Wagner, "Development of Textures by Slip and Twinning," in *Preferred Orientation in Deformed Metals and Rocks: An Introduction to Modern Texture Analysis*, H.-R. Wenk, Ed. (Academic Press, Orlando, Florida, 1985), pp. 233-258.
11. C. Tome and U. F. Kocks, "The Yield Surface of H.C.P. Crystals," *Acta Metall.* **33**, 603-621 (1985).
12. C. Barrett and T. B. Massalski, *Structure of Metals*, 3rd Edition (Pergamon, New York, 1980).
13. B. D. Cullity, *Elements of X-Ray Diffraction* (Addison-Wesley, Reading, Massachusetts, 1967).
14. R. Hill, *The Mathematical Theory of Plasticity* (Clarendon, Oxford, 1950).
15. K. G. Murty, *Linear and Combinatorial Programming* (John Wiley & Sons, Inc., New York, 1967), Chapter 2.
16. G. Y. Chin and W. L. Mammel, "Generalization and Equivalence of the Minimum Work (Taylor) and Maximum Work (Bishop-Hill) Principles for Crystal Plasticity," *Trans. Met. Soc. AIME* **245**, 1211-1214 (1969).
17. G. R. Canova, C. Fressengeas, A. Molinari, and U. F. Kocks, "Effect of Rate Sensitivity on Slip System Activity and Lattice Rotation," *Acta Metall.* **36**, 1961-1970 (1988).
18. K. Mathur, Thinking Machines Corporation, personal communication, October 1991.
19. G. P. van Houtte, "Some Recent Developments in the Theories for Deformation Texture Prediction," in *Proceedings of the Seventh International Conference on Textures of Materials*, C. M. Brakman, P. Jongenburger, and E. J. Mittermeijer, Eds. (Netherlands Society for Materials Science, Holland, 1984).
20. C. Tome, G. R. Canova, S. Ahzi, and A. Molinari, "Simulation of Texture Development in Hexagonal Materials Using a Self-Consistent Approach," in *Eighth International Conference on Textures of Materials*, J. S. Kallend and G. Gottstein, Eds. (Santa Fe, New Mexico, 1987).

21. H. J. Bunge, "Representation of Preferred Orientations," Chap. 4 in *Preferred Orientation in Deformed Metals and Rocks: An Introduction to Modern Texture Analysis*, H.-R. Wenk, Ed. (Academic Press, Orlando, Florida, 1985), pp. 73–108.
22. G. R. Canova, U. F. Kocks, C. N. Tome, and J. J. Jonas, "The Yield Surface of Textured Polycrystals," *J. Mech. Phys. Solids* **33**, 371–397 (1985).
23. S. K. Schiferl and P. J. Maudlin, "Texture-Induced Anisotropy and High-Strain-Rate Deformation in Metals," in *Shock Waves and High-Strain-Rate Phenomena in Materials*, M. A. Meyers, L. E. Murr, and K. P. Staudhammer, Eds. (Dekker, New York, 1991).
24. G. R. Johnson and R. A. Stryk, "User Instructions for the EPIC-2 Code," Air Force Armament Laboratory, Eglin Air Force Base Report AFATL-TR-86-51, (August 1986).
25. A. E. Green and P. M. Naghdi, "A General Theory of Elastic-Plastic Continuum," *Arch. Rat. Mech. Anal.* **18**, 251–275 (1965).
26. J. K. Dienes, "On Rotation and Stress Rate in Deforming Bodies," *Acta Mechanica* **32**, 217–232 (1979). For implementation, see: D. P. Flanagan and L. M. Taylor, *Comp. Meth. Appl. Mech. Eng.* **62**, 305–320 (1987).
27. S. Nemat-Nasser, "Rate-Independent Finite-Deformation Elastoplasticity: A New Explicit Constitutive Algorithm," *Mechanics of Materials* **11**, 235–249 (1991).
28. H. Conrad, "Effect of Interstitial Solutes on the Strength and Ductility of Titanium," *Prog. Mater. Sci.* **26**, 123–403 (1981).
29. M. A. Meyers and L. E. Murr, "Defect Generation in Shock-Wave Deformation," in *Shock Waves and High-Strain-Rate Phenomena in Metals*, M. A. Meyers and L. E. Murr, Eds. (Plenum, New York, 1980), pp. 487–530.
30. L. E. Murr, "Residual Microstructure—Mechanical Property Relationships in Shock-Loaded Metals and Alloys," in *Shock Waves and High-Strain-Rate Phenomena in Metals*, M. A. Meyers and L. E. Murr, Eds. (Plenum, New York, 1980), pp. 607–673.
31. G. T. Gray III, Los Alamos National Laboratory, personal communication, December 1993.

32. N. E. Paton and W. A. Backofen, "Plastic Deformation of Titanium at Elevated Temperatures," *Met. Trans.* **1**, 2839-2847 (1970).
33. C. Yin, M. Doner, and H. Conrad, "Effects of Interstitial Content on the Texture of Swaged and of Recrystallized Titanium Wire, *J. Less-Common Met.* **33**, 229-238 (1973).
34. G. R. Johnson and W. H. Cook, "A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates, and High Temperatures," *Proceedings of the Seventh International Symposium on Ballistics* (The Hague, Holland, April 1983).