SAND REPORT

SAND2003-4510
Unlimited Release
Printed December 2003

Improved Kinematic Options in ALEGRA

Grant V. Farnsworth and Allen C. Robinson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-4510
Unlimited Release
Printed December 2003

Improved Kinematic Options in ALEGRA

Grant V. Farnsworth and Allen C. Robinson
Computational Physics Research & Development
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0819

Abstract

Algorithms for higher order accuracy modeling of kinematic behavior within the ALEGRA framework are presented. These techniques improve the behavior of the code when kinematic errors are found, ensure orthonormality of the rotation tensor at each time step, and increase the accuracy of the Lagrangian stretch and rotation tensor update algorithm. The implementation of these improvements in ALEGRA is described. A short discussion of issues related to improving the accuracy of the stress update procedures is also included.

Acknowledgment

This work was supported by the Mathematics, Information, and Computer Science program, as well as ASCI and Computer Science Research Foundation funds at Sandia National Laboratories.

Outside of the ALEGRA group Rebecca Brannon and Clark Dohrmann were kind enough to give useful feedback on this manuscript. We also thank Rebecca Brannon for her code examples as well as to Clark Dohrmann and William Scherzinger for the eigenvalue/eigenvector decomposition routines. Discussions and review by members of the ALEGRA team and the ALEGRA algorithms group have been very useful.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Contents

Figures

Nomenclature

Term	Definition
ALE	Arbitrary Lagrangian Eulerian
ALEGRA	Arbitrary Lagrangian Eulerian General Research Application

Improved Kinematic Options in ALEGRA

1 Introduction

Solid dynamics modeling is integral to the simulations capability of the Arbitrary Lagrangian Eulerian (ALE) code ALEGRA [11]. Research into alternative methods for computing and correcting solid kinematic information has uncovered theoretical and practical weaknesses in ALEGRA's current implementation. In the course of this research, alternative solutions have been implemented as options, which may replace the default behavior once they have been sufficiently tested. This document is intended as an interim or "work in progress" report published in this format for the purpose of providing a permanent record of these changes to ALEGRA and a basis for discussions with ALEGRA users on their usage and potential impact. Feedback on the use of these options is encouraged.

1.1 ALEGRA's Default Algorithm

The kinematics of modeling continuum solids is described via a motion, $\mathbf{x}(\mathbf{a},t)$, where $\mathbf{x}$ is the current position as a function of the initial Lagrangian coordinates, $\mathbf{a}$, and time, t. Of particular interest is the deformation gradient tensor, $\partial\mathbf{x}/\partial\mathbf{a}$ which is needed in order to properly compute material stress states through the constitutive assumptions.

The approach to kinematics in ALEGRA is to decompose the deformation gradient at the initial time,

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{a}} = \mathbf{V}\mathbf{R} \tag{1}$$

into a symmetric positive definite left stretch tensor, $\mathbf{V}$, and an orthonormal rotation tensor, $\mathbf{R}$. The $\mathbf{V}$ and $\mathbf{R}$ matrices are then updated separately in a Lagrangian sense using a rate equation based on the mid-point velocity gradient tensor $\mathbf{L}$ [2, 6, 3]. That is,

$$\mathbf{L}_{n+1/2} = \nabla\mathbf{v} \text{ where } (\nabla\mathbf{v})_{ij} = \frac{\partial v_i}{\partial x_j} \tag{2}$$

and v_i and x_j represent Cartesian components which are both centered at the mid-time level,

$n+1/2$. Then

$$\mathbf{D}_{n+1/2} = \frac{1}{2}(\mathbf{L}_{n+1/2} + \mathbf{L}_{n+1/2}^T) \quad (3)$$

$$\mathbf{W}_{n+1/2} = \frac{1}{2}(\mathbf{L}_{n+1/2} - \mathbf{L}_{n+1/2}^T). \quad (4)$$

An angular velocity (or spin) tensor Ω is calculated from $\mathbf{D}_{n+1/2}$ and $\mathbf{V}_n$ using

$$\omega_{n+1/2} = \mathbf{w}_{n+1/2} + [tr(\mathbf{V}_n)\mathbf{I} - \mathbf{V}_n]^{-1}\mathbf{z}_{n+1/2}, \quad (5)$$

where $\mathbf{z}_{n+1/2}$, $\mathbf{w}_{n+1/2}$ and $\omega_{n+1/2}$ are the axial vectors corresponding to the skew-symmetric tensors $\mathbf{Z}_{n+1/2} = \mathbf{D}_{n+1/2}\mathbf{V}_n - \mathbf{V}_n\mathbf{D}_{n+1/2}$, $\mathbf{W}_{n+1/2}$ and $\Omega_{n+1/2}$, respectively. An axial vector, $\mathbf{w}$, corresponds to the skew-symmetric tensor, $\mathbf{W}$, if $\mathbf{Wv} = \mathbf{w} \times \mathbf{v}$ for every vector $\mathbf{v}$.

The standard update algorithm for $\mathbf{V}_n$ and $\mathbf{R}_n$ [2, 3], is

$$\mathbf{V}_{n+1} = \mathbf{V}_n + \Delta t(\mathbf{L}_{n+1/2}\mathbf{V}_n - \mathbf{V}_n\Omega_{n+1/2}) \quad (6)$$

$$\mathbf{R}_{n+1} = [\mathbf{I} - \frac{1}{2}\Delta t\Omega_{n+1/2}]^{-1}[\mathbf{I} + \frac{1}{2}\Delta t\Omega_{n+1/2}]\mathbf{R}_n. \quad (7)$$

This is a first-order algorithm in time since the derivative is computed using $\mathbf{V}_n$ instead of $\mathbf{V}_{n+1/2}$ in Equation 5 as will be discussed in Section 2.

In the remap step, the stretch and rotation are advected componentwise using a volume-based element remap algorithm [8]. This algorithm preserves the symmetry of $\mathbf{V}$, but it may not preserve the orthonormality of $\mathbf{R}$. By way of correction, the rows of $\mathbf{R}$ are independently normalized to unit magnitude in the three dimensional case. In two dimensions both rows are divided by the norm of the first. The methodology in three-dimensions does not project the tensor into the proper orthonormal space. Over time, the "rotation" tensor deviates from the orthonormality property required by the kinematic theory.

For large deformations, successive remaps may also cause the stretch tensor to lose the property of positive definiteness. This represents a fundamental error relative to the modeling assumption that $\det(\mathbf{F}) > 0$. As a precaution against this error, $\mathbf{V}$ is checked for positive definiteness using a determinant test after each Lagrangian step but before the remap. If a bad stretch tensor is found, the default behavior of the code is to terminate the simulation but the user may request that this behavior be ignored by resetting the offending tensor to the identity and continue. The loss of information about the stretch induced by a reset of $\mathbf{V}$ causes an inaccuracy that propagates rapidly throughout the mesh. For some problems, this may significantly affect the validity of the results, as will be seen in Section 3.

The principal reason for calculating **R** is for use in satisfying the principle of material frame indifference in material models and it is convenient to cast these models in the unrotated frame. For many models the stress state, **T**, is stored in the unrotated or reference stress configuration, σ. They are related by

$$\sigma_n = \mathbf{R}_n^T \mathbf{T}_n \mathbf{R}_n. \tag{8}$$

For example, a hypo-elastic material model may take the form

$$\begin{aligned} \mathbf{d}_{n+1/2} &= \mathbf{R}_{n+1}^T \mathbf{D}_{n+1/2} \mathbf{R}_{n+1} && (9) \\ \sigma_{n+1} &= \sigma_n + \Delta t \mathbf{f}(\mathbf{d}_{n+1/2}, \sigma_n). && (10) \end{aligned}$$

where we see that the rate-of-strain tensor, $\mathbf{D}_{n+1/2}$, is rotated into the material frame as $\mathbf{d}_{n+1/2}$ for use by the constitutive equations. Whenever **T** is needed, it is extracted using

$$\mathbf{T}_{n+1} = \mathbf{R}_{n+1} \sigma_{n+1} \mathbf{R}_{n+1}^T. \tag{11}$$

This stress update algorithm leads to first-order stress updates in time due to the use of $\mathbf{R}_{n+1}$ in Equation 9.

1.2 Summary of Changes

The code changes addressed in this paper implement options for the following improvements to the above method:

- A second-order Lagrangian update to **V** and **R**.
- A more rational way to handle non-positive definite stretch tensors **V**.
- A method for projecting **R** into the proper orthonormal space.

It is intended that these new options become the default options in ALEGRA in the future after sufficient experience is gained. The updates described here can be considered as minor improvements to the current algorithm.

This is not to say that alternatives to the stress integration algorithm viewpoint should not be considered. For example, in a Lagrangian code it is easy to compute the inverse deformation gradient with respect to a given set of spatial coordinates at any time level [1, 9]. Only a fast polar decomposition algorithm is required to compute **R**. Detailed comparisons of these approaches are certainly in order, but this is outside the scope of this

document. We emphasize, however, that remapping errors would continue to exist and would have to be dealt with the ALE or Eulerian case. We expect that constrained transport algorithms for curl-free fields (similar to constrained transport algorithms for divergence-free fields) will be required to properly advect either the coordinates themselves or the inverse deformation gradient. A straightforward component-by-component remapping of the inverse deformation gradient may be possible but the attendant cost of the eventual loss of the gradient character of **F** is unknown.

Important issues related to second-order stress updates are discussed in Section 5, although no particular algorithm has yet been implemented.

2 An Improved Lagrangian VR Update

The update of **V** and **R** done in the Lagrangian step is first-order accurate in time because the velocity gradient **L** is known only at the mid-point time, but the **V** and **R** tensors are known at the time step. After obtaining Ω_n, we can compute an estimate of **V** at the mid-point time using

$$\mathbf{V}_{n+1/2} = \mathbf{V}_n + \frac{1}{2}\Delta t(\mathbf{L}_{n+1/2}\mathbf{V}_n - \mathbf{V}_n\Omega_n) \tag{12}$$

and then use this estimate of $\mathbf{V}_{n+1/2}$ to compute $\Omega_{n+1/2}$ as in Equation 5.

$$\omega_{n+1/2} = \mathbf{w}_{n+1/2} + [tr(\mathbf{V}_{n+1/2})\mathbf{I} - \mathbf{V}_{n+1/2}]^{-1}\mathbf{z}_{n+1/2}, \tag{13}$$

with $\mathbf{Z}_{n+1/2} = \mathbf{D}_{n+1/2}\mathbf{V}_{n+1/2} - \mathbf{V}_{n+1/2}\mathbf{D}_{n+1/2}$. The final update then becomes

$$\mathbf{V}_{n+1} = \mathbf{V}_n + \Delta t(\mathbf{L}_{n+1/2}\mathbf{V}_{n+1/2} - \mathbf{V}_{n+1/2}\Omega_{n+1/2}), \tag{14}$$

$$\mathbf{R}_{n+1} = [\mathbf{I} - \frac{1}{2}\Delta t\Omega_{n+1/2}]^{-1}[\mathbf{I} + \frac{1}{2}\Delta t\Omega_{n+1/2}]\mathbf{R}_n. \tag{15}$$

Equations 12 through 14 amount to a mid-point Runge Kutta integration and thus lead to a second-order accurate integration algorithm. Both the first-order accuracy of the original algorithm and the second-order accuracy of the algorithm described in this section have been verified numerically. The computational cost involves little more than one extra tensor inversion per cycle. Equation 15 represents an application of the Cayley transformation to obtain an orthogonal rotation update [1].

This improvement can be enabled through the line

```
vr_update_method,{first order|second order}
```

in the solid dynamics section of the input file. First-order updating is selected by default.

3 Stretch Tensor Reset After Remap

A negative eigenvalue in the stretch tensor indicates a breakdown in the kinematic description as a result of finite grid resolution and remap errors. For sufficiently large deformations, these errors are inevitable, but the errors are often spatially localized and it may be desirable to minimize the impact of these errors on the rest of the grid while pushing the computation through to completion.

The improved kinematic error fix uses a fast spectral decomposition solver to examine the eigenvalues of the stretch tensor at each time step and force them to remain positive. The eigenvalue solver used was provided by Scherzinger [9].

Let

$$\mathbf{VQ} = \mathbf{Q}\Lambda \tag{16}$$

represent an eigenvalue/eigenvector diagonalization of $\mathbf{V}$ and let λ_s be a preselected floor value (10^{-6} by default). If $\lambda_1 \leq \lambda_2 \leq \lambda_3$ are the ordered eigenvalues of $\mathbf{V}$, then we modify the elements of Λ according to

$$\hat{\lambda_1} = \max(\lambda_1, \lambda_s) \tag{17}$$

$$\hat{\lambda_2} = \lambda_2 \tag{18}$$

$$\hat{\lambda_3} = \lambda_3 \tag{19}$$

and compute the "corrected" or "fixed" $\mathbf{V}$ using

$$\mathbf{V} = \mathbf{Q}\hat{\Lambda}\mathbf{Q}^T. \tag{20}$$

Although this modification of $\mathbf{V}$ does not correct the underlying problem associated with finite grid resolution and remapping, it lessens its impact on the rest of the calculation.

Preliminary tests indicate that the use of this option on problems that experience kinematic errors may significantly improve the quality of the solution. Figure 1 compares the accuracy of the eigenvalue floor option and the reset to identity option in a problem of incompressible rotational motion in the Lagrangian/Eulerian framework. The estimated stretches using the eigenvalue limiting methods match the exact solution much more closely than the solution which resets the stretch to the identity. Note that these errors have nothing to do with element inversions in the Lagrangian step. The mesh in this case is a Cartesian grid which deforms only slightly each time step. The errors observed are a result of many Lagrangian/remap cycles and will occur at some point in time for any fixed grid resolution. The true stretch tensor becomes so highly close to singular that truncation errors in the remap will push the stretch tensor outside of the theoretical bounds.

If no steps are take by the user, ALEGRA will abort upon encountering a non-positive definite stretch tensor. For this case a simple check for positive definiteness of the stretch tensor rather than an eigenvalue decomposition is used. No eigenvector decomposition is needed. However, if the user desires to proceed anyway the "ignore kinematic errors" keyword can be enabled.

This option is enabled through the keyword

```
ignore kinematic errors[,reset to identity| limit smallest eigenvalue]
   [,1.0e-6] [,silent]
```

in the solid dynamics section of the input file. By default, the "reset to identity" option is selected when the "ignore kinematic errors" keyword is found in the input. No eigenvector decomposition is required is this case. The eigenvector decomposition is required for the "limit smallest eigenvalue" case. The floor value applies only to the "limit smallest eigenvalue" option. To modify the floor value, replace 1.0e-6 with the desired positive value. The "silent" keyword turns off all warnings with respect to application of either correction method to the stretch tensor. Currently, this stretch correction algorithm occurs in the Lagrangian step in the routine *Material::Update_Rotation_and_Stretch*. A more logical place for this correction is in the *Material::Update_After_Remap* function since this the correction is meant to apply to errors generated during remap. This code reorganization may occur in the future.

4 Renormalization of R after Remap

Since the remap algorithm does not maintain the orthonormality of the rotation tensor, we should project **R** into the space of orthonormal tensors. The solution to this minimization problem under the Frobenius norm is the result of doing a polar decomposition as in (1) and setting **R** to the resulting rotation tensor, essentially throwing away the stretch part, which we consider remap error. Fast iterative methods for polar decomposition of non-singular matrices have been studied in detail [4, 5, 7]. We have implemented methods suggested by Brannon [1].

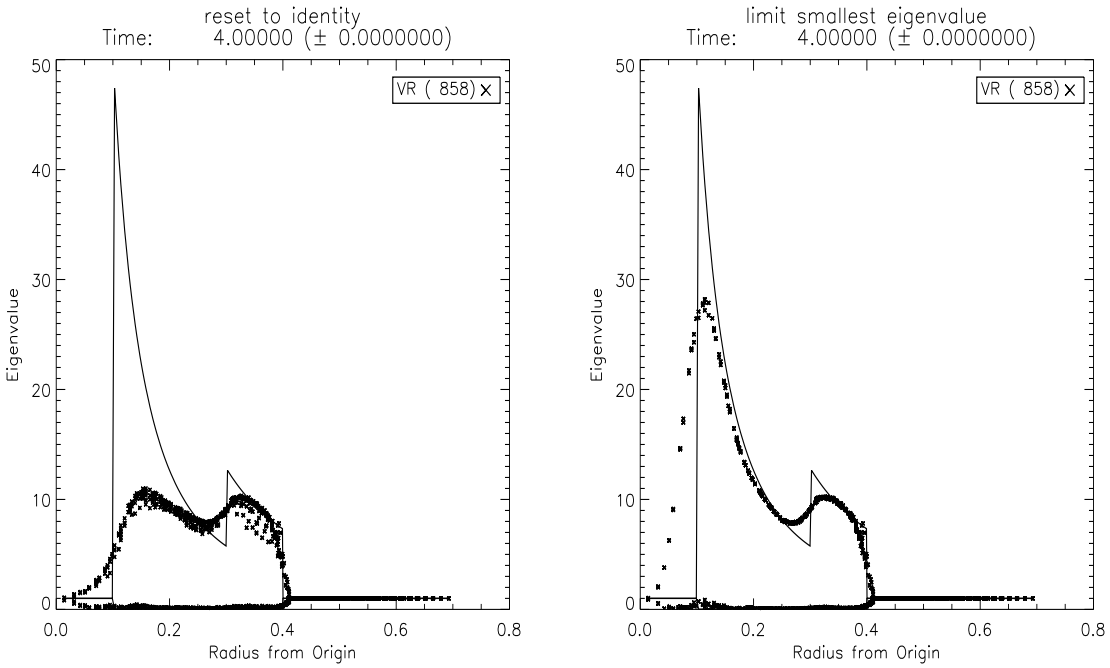

Figure 1. Stretch eigenvalues for a rotation problem with a discontinuous vorticity field. Plotted is the eigenvalue vs. radius from origin with the exact solution plotted in the solid line. Reset to identity (left) and limit smallest eigenvalue (right).

In two dimensions, the orthogonalization can be performed explicitly by

$$\hat{R}_{11} = (R_{11}+R_{22})/a \tag{21}$$
$$\hat{R}_{21} = (R_{21}-R_{12})/a \tag{22}$$
$$\hat{R}_{12} = (R_{12}-R_{21})/a \tag{23}$$
$$\hat{R}_{22} = (R_{11}+R_{22})/a \tag{24}$$

where

$$a = \sqrt{(R_{11}+R_{22})^2+(R_{21}-R_{12})^2} \tag{25}$$

This algorithm actually requires fewer operations than the current row normalization method. The denominator will not be zero for matrices which have a non-zero determinant. In particular we expect the remapped rotation matrices to be close to proper orthogonal rotation matrices and this algorithm will perform well.

In three dimensions, the rotational part of **R** may be extracted by a fixed point iteration, which converges if the maximum stretch of **R** is less than $\sqrt{3}$. Since the orthonormal part

of a tensor is invariant to scalar multiples, we can first rescale $\mathbf{R}$ according to

$$\mathbf{R}^0 = \sqrt{\frac{3}{tr(\mathbf{R}^T\mathbf{R})}}\mathbf{R}. \tag{26}$$

If $\lambda_1^2 \leq \lambda_2^2 \leq \lambda_3^2$ are the non-zero ordered eigenvalues of $(\mathbf{R}^0)^T\mathbf{R}^0$, then by construction $\lambda_1^2+\lambda_2^2+\lambda_3^2 = 3$ and thus $\lambda_3^2 < 3$. The iteration

$$\mathbf{R}^{m+1} = \frac{1}{2}\mathbf{R}^m[3\mathbf{I} - (\mathbf{R}^m)^T\mathbf{R}^m]. \tag{27}$$

is thus guaranteed to converge to the nearest orthonormal tensor. In fact, due to the fact that the remap algorithm generally produces updated rotation tensors that are "nearly" orthonormal, this iteration will converges to machine precision in just a few iterations. The iteration stops when $||\mathbf{R}^T\mathbf{R} - \mathbf{I}||^2 < \varepsilon^2$ where ε is an small number close to machine precision. The routine will abort after 15 iterations.

The normalization method can be specified by using the keyword

```
rotation normalization method,{rows|projection}
```

in the solid dynamics section of the input file. By default, the row normalization is performed. The rotation projection algorithm occurs in the *Material::Update_After_Remap* function.

5 The ALEGRA Material Model Interface and Second Order Stress Updates

In order to reap the benefits of a second-order Lagrangian **VR** update, the stress update should achieve the same level of accuracy. The Lagrangian step ALEGRA call sequence and material model interface is set up such that the rotation matrix is updated to $\mathbf{R}_{n+1}$ before the call to the material model. The time centered deformation rate provided by ALEGRA is rotated to the material frame at the top of the material model interface coding.

In order to have a well-defined and separable interface between the host code, ALEGRA, and a general material model, which can be ported between different host code frameworks, material modelers must assume they will have available a time centered deformation rate tensor in the unrotated or material configuration. That is, they expect to

have

$$\mathbf{d} = \mathbf{R}_{n+1/2}^T \mathbf{D}_{n+1/2} \mathbf{R}_{n+1/2} \tag{28}$$

available along with a time step Δt and state variables at time t_n in order to be able to generate a second-order accurate time integration algorithm. For example, one possible second-order algorithm valid for some material models would be

$$\begin{aligned} \mathbf{d}_{n+1/2} &= \mathbf{R}_{n+1/2}^T \mathbf{D}_{n+1/2} \mathbf{R}_{n+1/2} & (29) \\ \sigma_{n+1/2} &= \sigma_n + \frac{\Delta t}{2} \mathbf{f}(\mathbf{d}_{n+1/2}, \sigma_n) & (30) \\ \sigma_{n+1} &= \sigma_n + \Delta t \mathbf{f}(\mathbf{d}_{n+1/2}, \sigma_{n+1/2}) & (31) \end{aligned}$$

which is second-order accurate in time. Another example is an implicit trapezoidal rule for the stress update as suggested by Trangenstein in conjunction with a Lagrange multiplier for the yield condition [12]. We expect many possible algorithmic and modeling variations [10].

Unfortunately what the current ALEGRA framework provides to material modelers is the time step Δt and

$$\mathbf{d}_{n+1/2} = \mathbf{R}_{n+1}^T \mathbf{D}_{n+1/2} \mathbf{R}_{n+1}. \tag{32}$$

Thus $\mathbf{d}_{n+1/2}$ has a first-order time truncation error which will make it impossible to develop a standalone stress integration algorithm which is second-order in time. Although we have not yet verified this statement experimentally in ALEGRA, the issue clearly requires further investigation.

Improvements to this situation could be accomplished with some amount of code restructuring and support work within ALEGRA and the material model interface. New approaches should be considered. We point out that the rotation update equation

$$\mathbf{R}_{n+1} = \mathbf{Q}\mathbf{R}_n = [\mathbf{I} - \frac{\Delta t}{2}\Omega_{n+1/2}]^{-1}[\mathbf{I} + \frac{\Delta t}{2}\Omega_{n+1/2}]\mathbf{R}_n, \tag{33}$$

which is termed the Cayley transformation, is only one of several forms for updating rotation tensors in terms of $\Omega_{n+1/2}$. What is needed by the Lagrangian algorithm and material models is both $\mathbf{R}_{n+1/2}$ (e.g. for stress integration algorithms) and $\mathbf{R}_{n+1}$ for rotation to the current frame as in Equation 11. Due to the Lie algebra generated by skew symmetric tensors, very tight and efficient closed form equations can be obtained for these quantities in different approximations. Various forms of the Euler-Rodrigues formula or exponential map are possible [1] and [10] (Chapter 8.3.2) which has the advantages of being an exact solution for constant spins. $\mathbf{Q}^{1/2}\mathbf{R}_n$ has also been proposed for $\mathbf{R}_{n+1/2}$, e.g. [12] (p. 79). In all cases, simple, efficient expressions are possible in terms of $\Omega_{n+1/2}$.

The most obvious and simple change to the ALEGRA interface would be to compute and store both $\mathbf{R}_{n+1/2}$ in addition to $\mathbf{R}_{n+1}$ just before the call to the material models interface. Alternatively, the three unique components of $\Omega_{n+1/2}$ could be stored and $\mathbf{R}_{n+1/2}$ could be computed from $\mathbf{R}_{n+1}$ as needed. Another approach would be to compute $\mathbf{R}_{n+1/2}$ for use in the polar stress integration algorithms and store $\Omega_{n+1/2}$ for use in updating to $\mathbf{R}_{n+1}$ after the material model interface. A tightly specified and possibly expanded concept of the material model interface is necessary. We believe that the issue of upgrading the ALEGRA interface to clearly satisfy the input requirement that material models be able to generate consistent second-order integration algorithms must be addressed. Those models which may not already be designed for second-order accuracy could also be upgraded.

The issue of the development and testing for second-order accuracy is one aspect of improving the quality of the ALEGRA material model interface support. Brannon has suggested that a general unit testing interface be made available to check for time accuracy and frame indifference under refinement with specified displacements. We concur that such support would be extremely useful for verification purposes as material models can be extremely complicated and as such demand robust testing within the ALEGRA code framework itself. Unit testing outside of the framework can only be part of the answer to verification requirements.

6 Conclusions

Each of the changes mentioned in this paper can potentially have a positive effect on simulation quality. The effect of the stretch tensor reset fix is particularly noticeable, if the problem generates bad stretch tensors. Our evidence suggests that the limit smallest eigenvalue scheme should become the default method of patching the stretch tensor. The projection renormalization of **R** is clearly an appropriate improvement over the row normalization technique.

The **VR** update improvement entails only a minor cost and at least gives an improved accuracy for **R** for use by stress update algorithms and thus may be a useful permanent algorithmic change. However, the gains from improving the Lagrangian **VR** update may not be particularly significant due to the centering of the unrotated deformation rate as currently calculated in material models using stress integration procedures. This issue deserves further study. A robust unit testing interface for verification purposes is suggested as a means of moving in the direction of a more robust, verifiable and supportable ALEGRA material model interface.

References

[1] R. M. Brannon. Rotation and reflection: A comprehensive review of the theory of real orthogonal tensors in the context of engineering mechanics. Technical Report unpublished draft document, Sandia National Laboratories, Albuquerque, New Mexico, 2003.

[2] J. K. Dienes. On the analysis of rotation and stress rate in deforming bodies. *Acta Mechanica*, 32:217–232, 1979.

[3] D. P. Flanagan and L. M. Taylor. An accurate numerical algorithm for stress integration with finite rotations. *Computer Methods in Applied Mechanical Engineering*, 62:305–320, 1987.

[4] Nicholas J. Higham. Computing the polar decomposition - with applications. *SIAM Journal Scientific and Statistical Computing*, 7(4):1160–1174, October 1986.

[5] Nicholas J. Higham and Robert S. Schreiber. Fast polar decomposition of an arbitrary matrix. *SIAM Journal Scientific and Statistical Computing*, 11(4):648–655, July 1990.

[6] Thomas J. R. Hughes and James Winget. Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis. *Computer Methods in Applied Mechanical Engineering*, 15(12):1862–1867, 1980.

[7] Charles Kenney and Alan J. Laub. On scaling newton's method for polar decompositions and the matrix sign function. *SIAM Journal Matrix Analysis*, 13(3):688–706, July 1992.

[8] James S. Peery and Daniel E. Carroll. Multi-material ALE methods in unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 187:591–619, 2000.

[9] William M. Scherzinger and Clark R. Dohrmann. Strong incremental objectivity and improved kinematics in ASCI solid mechanics codes. Technical Report unpublished draft document, Sandia National Laboratories, Albuquerque, New Mexico, March 2002.

[10] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Springer-Verlag New York, Inc., 1998.

[11] R. M. Summers, J. S. Peery, and M. W. Wong. Recent progress in ALEGRA development and application to ballistic impacts. *Int. J. Impact Engng.*, 20:779–788, 1997.

[12] John A. Trangenstein and Phillip Colella. A higher-order Godunov method for modeling finite deformation in elastic-plastic solids. *Communications on Pure and Applied Mathematics*, 44:41–100, 1991.

DISTRIBUTION:

1 Filbey, Gordon L., Jr.
U.S. Army Research Lab
AMSRD-WT-TA
Aberdeen Proving Ground, MD
21005-5066

1 Doney, Bobby
U.S. Army Research Lab
AMSRD-WT-TA
Aberdeen Proving Ground, MD
21005-5066

1 Fermen-Coker, M.
U.S. Army Research Lab
AMSRD-WT-TC
Aberdeen Proving Ground, MD
21005-5066

1 Kimsey, Kent D.
U.S. Army Research Lab
AMSRD-WT-TC
Aberdeen Proving Ground, MD
21005-5066

1 Bruchey, William
U.S. Army Research Lab
AMSRD-WT-TD
Aberdeen Proving Ground, MD
21005-5066

1 Kingman, Pat
U.S. Army Research Lab
AMSRD-WT-TD
Aberdeen Proving Ground, MD
21005-5066

1 Schraml, Stephan J.
U.S. Army Research Lab
AMSRD-WM-TD
Aberdeen Proving Ground, MD
21005-5066

1 MS 0321
W. J. Camp, 09200

1 MS 0670
D. E. Carroll, 06524

1 MS 0751
R. M. Brannon, 06117

1 MS 0318
P. Yarrington, 09230

1 MS 0819
R. M. Summers, 09231

1 MS 0819
K. H. Brown, 09231

1 MS 0819
S. P. Burns, 09231

1 MS 0819
M. A. Christon, 09231

1 MS 0819
R. R. Drake, 09231

5 MS 0819
G. V. Farnsworth, 09231

1 MS 0819
D. I. Ketcheson, 09231

1 MS 0819
T. G. Trucano, 09211

1 MS 0819
S. Carroll, 09231

1 MS 0819
D. M. Hensinger, 09231

1 MS 0819
S. V. Petney, 09231

1 MS 0819
J. Robbins, 09231

5 MS 0819
A. C. Robinson, 09231

1 MS 0819
T. E. Voth, 09231

1 MS 0819
M. K. Wong, 09231

1 MS 0828
W. L. Oberkampf, 09133

1 MS 0820
P. Chavez, 09232

1 MS 0820
P. A. Taylor, 09232

1 MS 0835
J. M. McGlaun, 09140

1 MS 0835
K. F. Alvin, 09142

1 MS 0835
J. D. Hales, 09142

1 MS 0893
W. M. Scherzinger, 09123

1 MS 0847
M. L. Blanford, 09142

1 MS 0847
C. R. Dohrmann, 09124

1 MS 0847
A. S. Gullerud, 09142

1 MS 0847
M. W. Heinstein, 09142

1 MS 0847
S. W. Key, 09142

1 MS 0847
J. R. Koteras, 09142

1 MS 0847
J. A. Mitchell, 09142

1 MS 0847
K. H. Pierson, 09142

1 MS 1110
D. Womble, 9214

1 MS 1186
T. A. Mehlhorn, 01674

1 MS 1186
C. J. Garasi, 01674

1 MS 1186
T. A. Haill, 01674

1 MS 1186
R. W. Lemke, 01674

2 MS 0899
Technical Library, 9616

1 MS 9018
Central Technical Files, 8945-1