

Non-steady homogeneous deformations: Computational techniques using Lie theory, and application to ellipsoidal markers in naturally deformed rocks

Joshua R. Davis^{1*}, Sarah J. Titus², Eric Horsman³

¹*Department of Mathematics, Carleton College, 1 N. College St., Northfield, MN 55057; 507-645-4402; jdavis@carleton.edu; *corresponding author*

²*Department of Geology, Carleton College; stitus@carleton.edu*

³*Department of Geological Sciences, East Carolina University; horsmane@ecu.edu*

Abstract

The dynamic theory of deformable ellipsoidal inclusions in slow viscous flows was worked out by J.D. Eshelby in the 1950s, and further developed and applied by various authors. We describe three approaches to computing Eshelby's ellipsoid dynamics and other homogeneous deformations. The most sophisticated of our methods uses differential-geometric techniques on Lie groups. This Lie group method is faster and more precise than earlier methods, and perfectly preserves certain geometric properties of the ellipsoids, including volume. We apply our method to the analysis of naturally deformed clasts from the Gem Lake shear zone in the Sierra Nevada mountains of California, USA. This application demonstrates how, given three-dimensional strain data, we can solve simultaneously for best-fit bulk kinematics of the shear zone, as well as relative viscosities of clasts and matrix rocks.

Keywords: velocity gradient, deformable ellipsoid, viscous flow, Lie group, inverse model

1. Introduction

Geologists often use elliptical and ellipsoidal markers to characterize strain and infer rheology. Depending on the structural or tectonic context, the ratio r between a clast's viscosity and the host rock's viscosity may vary widely. Passive markers, where $r = 1$, are perhaps the most thoroughly understood case (e.g., Ramsay, 1967; Dunnet, 1969; Elliott, 1970; Matthews et al., 1974; Lisle, 1977). Examples include reduction spots (e.g., Tullis and Wood, 1975) and ooids (e.g., Cloos, 1947, 1971). Rigid clasts, where $r = \infty$, are another important special case. Jeffery (1922) developed a dynamic theory of rigid ellipsoid rotation, which has been applied extensively (Gay, 1968a,b; Ghosh and Ramberg, 1976; Passchier, 1987; De Paor, 1988; Jezek et al., 1996; Simpson and De Paor, 1997; Jessup et al., 2007). Voids, where $r = 0$, represent the other end-member case. Voids have been used in volcanology to study the kinematics of flowing lavas (e.g., Rust et al., 2003) and to estimate magma viscosity (e.g., Manga et al., 1998).

However, ellipsoidal markers in rocks are not always well-modeled by these three idealized special cases. Competent clasts may exhibit viscosity ratios $1 < r < \infty$, and incompetent clasts may exhibit $0 < r < 1$ (Fig. 1). A dynamic theory of deformable ellipsoids in slow viscous flows was developed by Eshelby (1957, 1959) and Bilby et al. (1975). This theory handles all viscosity ratios, including the special cases of passive, rigid, and void ellip-

soids. The theory has been extended with new computational approaches (Freeman, 1987; Schmid and Podladchikov, 2003; Mulchrone and Walsh, 2006; Jiang, 2007a), and modified to handle interacting clasts (Mandal et al., 2003), different clast or matrix properties (e.g., Fletcher, 2004, 2009; Dabrowski and Schmid, 2011; Mancktelow, 2011), and other clast shapes (Treagus and Treagus, 2001; Treagus, 2002).

In this paper, we present new approaches to computing Eshelby's deforming ellipsoids. Our main method relies on a mathematical structure called a *Lie group* (see, e.g., Belinfante and Kolman, 1972; Gilmore, 1974; Curtis, 1984; Hall, 2003; Pollatsek, 2009), explained in more detail in Section 3.3. Like earlier methods, our method produces a numerical approximation to the deformation of the ellipsoids, but with three advantages. First, the method automatically preserves desirable characteristics of the ellipsoids, such as their volumes and basic ellipsoidal shape. Second, the method applies not just to Eshelby's ellipsoids, but to all non-steady homogeneous deformations. Third, the method is faster and more precise than earlier methods. Such speed is of practical value to geologists, because the analysis of naturally deformed rocks may require the simulation of many ellipsoid deformations. As an example application, we inverse-model deformed clast data from the Gem Lake shear zone in the Sierra Nevada mountains of California, USA (Section 5). This application demonstrates how, given three-dimensional strain data, we can

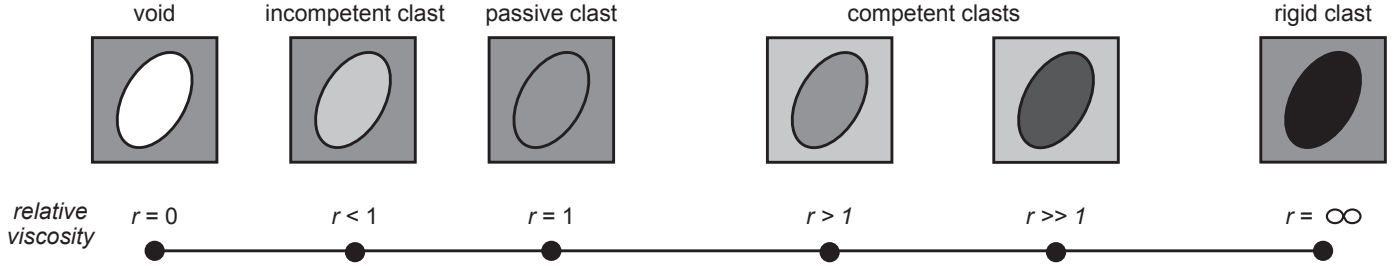


Figure 1: The spectrum of possible ellipsoid problems, organized by viscosity ratio r .

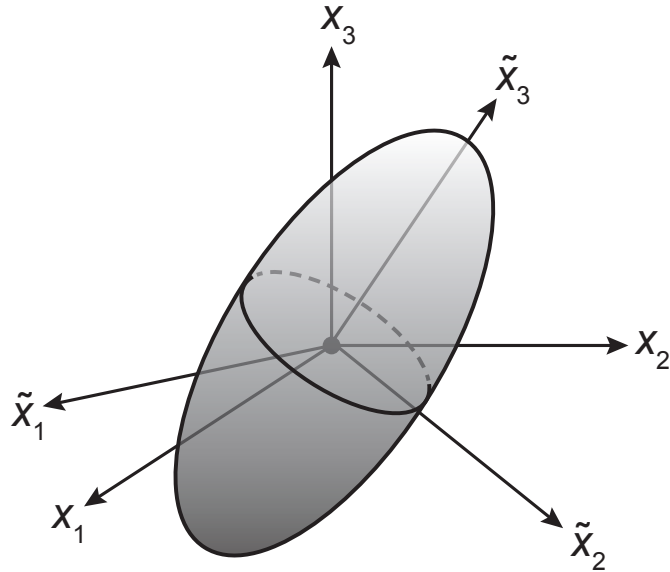


Figure 2: Two coordinate systems are employed. The \mathbf{x} coordinates are fixed. The $\tilde{\mathbf{x}}$ coordinates are aligned with the axes of the ellipsoid, and rotate over time.

solve simultaneously for best-fit bulk kinematics of the shear zone as well as viscosity ratios of several clast types. We compare our results with those from a previously published study that relied on traditional strain modeling.

2. Mathematical framework

Consider a rock that contains a single ellipsoidally shaped inclusion of a different viscosity. We subject the rock to a volume-preserving homogeneous deformation. As the matrix rock deforms, the inclusion deforms differently due to the viscosity contrast. We assume that both the matrix and the inclusion materials remain homogeneous, isotropic, and of constant viscosity at all times.

In coordinates $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$ centered on the inclusion (Fig. 2), the velocity $\dot{\mathbf{x}}$ at each point in the host rock is linearly related to the position vector \mathbf{x} at that point by a velocity gradient tensor \mathbf{L} :

$$\dot{\mathbf{x}} = \mathbf{L}\mathbf{x}. \quad (1)$$

We are denoting by \mathbf{L} both the tensor and its matrix representation in the \mathbf{x} coordinates. For simplicity of presentation, we assume that the deformation is steady, so that

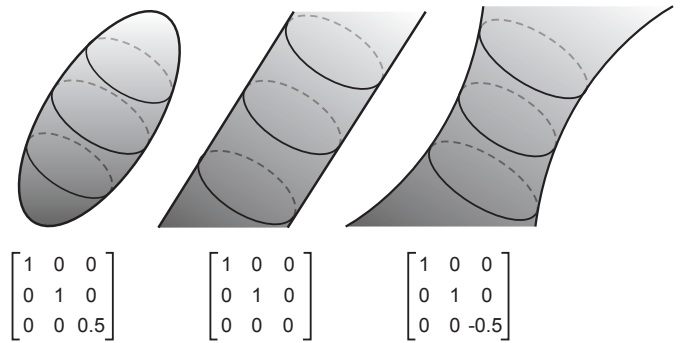


Figure 3: The ellipsoid tensor \mathbf{E} should have only positive eigenvalues. Numerical errors can cause the eigenvalues to drift, sometimes even producing an unphysical shape such as a hyperboloid.

\mathbf{L} is constant. The techniques of this paper could be extended to time-dependent \mathbf{L} easily. Because the flow preserves volume, $\text{tr } \mathbf{L} = 0$. There are no other assumptions or restrictions on \mathbf{L} .

At any given time, the boundary ellipsoid of the inclusion can be described by a tensor \mathbf{E} (e.g., Flinn, 1979), in that the ellipsoid is the set of points \mathbf{x} such that

$$\mathbf{x}^T \mathbf{E} \mathbf{x} = 1. \quad (2)$$

Mathematically, \mathbf{E} is a symmetric, positive-definite $(0, 2)$ -tensor. Symmetry and positive-definiteness mean that Eq. (2) defines an ellipsoid, rather than a hyperboloid or other unphysical shape, and that \mathbf{E} diagonalizes as $\mathbf{E} = \mathbf{Q}^T \tilde{\mathbf{E}} \mathbf{Q}$, where \mathbf{Q} is a rotation matrix,

$$\tilde{\mathbf{E}} = \begin{bmatrix} a_1^{-2} & 0 & 0 \\ 0 & a_2^{-2} & 0 \\ 0 & 0 & a_3^{-2} \end{bmatrix}, \quad (3)$$

and the $a_i > 0$ (Fig. 3). The rows of \mathbf{Q} are unit vectors (in \mathbf{x} coordinates) indicating the directions of the ellipsoid semi-axes in a right-handed manner, and the a_i are the semi-axis lengths. The matrix $\tilde{\mathbf{E}}$ is the tensor \mathbf{E} rendered in a coordinate system $\tilde{\mathbf{x}} = [\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3]^T$ aligned with the ellipsoid's axes (Fig. 2). When two of the semi-axis lengths a_i and a_j are equal, there is an apparent ambiguity in choosing the $\tilde{\mathbf{x}}$ coordinates, or equivalently \mathbf{Q} . However, there is a unique correct way to resolve this ambiguity (see Jiang (2007a) and Appendix B). The volume of the

ellipsoid is

$$\frac{4}{3}\pi a_1 a_2 a_3 = \frac{4\pi}{3\sqrt{\det \mathbf{E}}}.$$

Under the assumption of slow viscous flow, Eshelby (1957) and Bilby et al. (1975) showed that, if the inclusion is an ellipsoid, then its deformation is also homogeneous. That is, for all \mathbf{x} inside the inclusion and along its boundary,

$$\dot{\mathbf{x}} = \mathbf{K}\mathbf{x}, \quad (4)$$

for some velocity gradient tensor \mathbf{K} (viewed as a matrix, in the \mathbf{x} coordinates). The dynamics of the ellipsoid are most easily described in the $\tilde{\mathbf{x}}$ coordinates (Appendix A). As the ellipsoid deforms, $\det \mathbf{E}$ and the ellipsoid volume remain constant, but \mathbf{K} continually changes. Thus Eshelby's theory is an example of a non-steady homogeneous deformation. In contrast to the steady homogeneous case (Provost et al., 2004; Davis and Titus, 2011), the differential equation that governs the non-steady case (Eq. (4)) admits no closed-form solution. One must resort to an iterative algorithm that produces a numerical approximation to the deformation. An inherent trade-off exists between the precision of the simulation and the computational time required. Imprecision causes the ellipsoid to drift away from its true size, shape, and orientation. In some cases, imprecision may lead to a catastrophic failure, such as an ellipsoid that degenerates to a cylinder or hyperboloid (Fig. 3). The issue is that, while all ellipsoids are represented by tensors, not all tensors represent ellipsoids. Arithmetic operations on ellipsoid tensors can produce tensors that are not physically meaningful. We describe this phenomenon in detail in the next section.

In summary, \mathbf{K} describes the flow inside the ellipsoid, and \mathbf{L} describes the flow at points distant from the ellipsoid. Eshelby (1959) describes the flow at intermediate points, but we do not require that part of the theory.

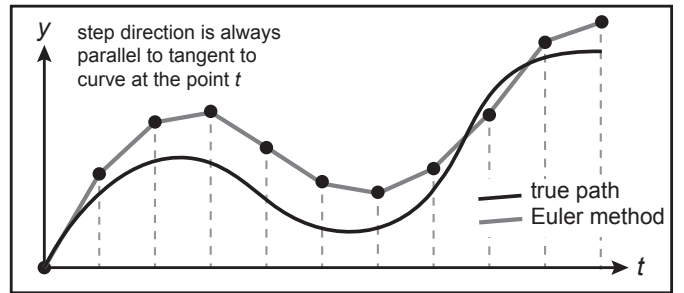
3. Simulation of deformable ellipsoids

3.1. Simulating the size and orientation separately ((\mathbf{Q}, a_i) method)

It is common in the geological literature (e.g., Freeman, 1987; Jiang, 2007a, 2012) to simulate the semi-axis lengths a_1, a_2, a_3 and orientation \mathbf{Q} of the deforming ellipsoid separately. That is, each of these four quantities is governed by its own differential equation, although the four equations are strongly coupled. For example, let $\tilde{\mathbf{C}}$ be the stretching tensor of the inclusion's deformation, expressed in $\tilde{\mathbf{x}}$ coordinates, and \mathbf{W} and $\tilde{\mathbf{W}}$ the vorticity tensor of the host rock deformation, expressed in \mathbf{x} and $\tilde{\mathbf{x}}$ coordinates respectively. Appendix A gives explicit expressions for these quantities in terms of \mathbf{Q} and the a_i . By Eqs. (A.16) and (A.17),

$$\begin{aligned} \dot{a}_i &= f_i(\mathbf{Q}, a_1, a_2, a_3), \\ \dot{\mathbf{Q}} &= g(\mathbf{Q}, a_1, a_2, a_3), \end{aligned}$$

Euler method



Runge-Kutta method

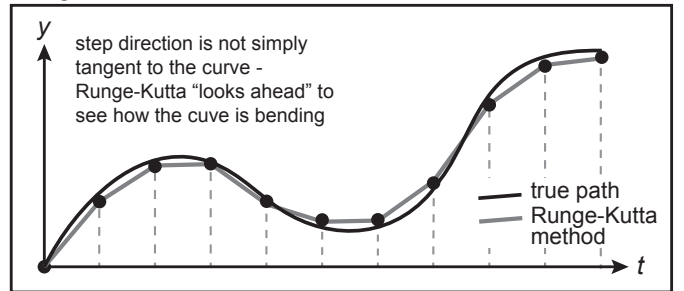


Figure 4: Cartoon of the Euler and Runge-Kutta methods. Both methods approximate the solution to a differential equation using a sequence of discrete, steady steps. The latter method typically produces less error.

where $f_i(\mathbf{Q}, a_1, a_2, a_3) = a_i \tilde{C}_{ii}$ and $g(\mathbf{Q}, a_1, a_2, a_3) = \tilde{\mathbf{W}}\mathbf{Q} - \mathbf{Q}\tilde{\mathbf{W}}$. We can solve these differential equations using the classic Euler method, Runge-Kutta method, etc. (e.g., Gerald and Wheatley, 1984; Burden and Faires, 2001; Chapra and Canale, 2002), to obtain a numerical approximation to the evolution of \mathbf{Q} and the a_i over time. We briefly review these methods now.

The key idea of the Euler method is to divide the deformation time into subintervals, and to approximate $\dot{\mathbf{Q}}$ and the \dot{a}_i as constant over each subinterval. Geometrically, the true solution of the differential equation is a curved path, which the Euler method approximates by a sequence of straight line segments (Fig. 4). To make this idea precise, suppose that time t runs from 0 to 1, and divide the time interval $[0, 1]$ into n steps, each of length $h = 1/n$. For $s = 0, 1, \dots, n$, we compute approximations $\mathbf{Q}_s \approx \mathbf{Q}(sh)$ and $a_{i,s} \approx a_i(sh)$ to \mathbf{Q} and a_i at time $t = s/n = sh$. That is, $\mathbf{Q}_0 = \mathbf{Q}(0)$ is the initial orientation, $\mathbf{Q}_n \approx \mathbf{Q}(1)$ is the final orientation, and similarly for the a_i . Starting from an initial ellipsoid $\mathbf{E}_0 = \mathbf{E}(0)$, compute \mathbf{Q}_0 and the $a_{i,0}$. Then apply the iteration formulas

$$\begin{aligned} a_{i,s+1} &= a_{i,s} + hf_i(\mathbf{Q}_s, a_{1,s}, a_{2,s}, a_{3,s}) \\ \mathbf{Q}_{s+1} &= \mathbf{Q}_s + hg(\mathbf{Q}_s, a_{1,s}, a_{2,s}, a_{3,s}) \end{aligned}$$

to obtain $\mathbf{Q}_1, a_{1,1}, a_{2,1}, a_{3,1}$, and then $\mathbf{Q}_2, a_{1,2}, a_{2,2}, a_{3,2}$, and so on, up to the final values $\mathbf{Q}_n, a_{1,n}, a_{2,n}, a_{3,n}$.

For the sake of brevity, let $y = (\mathbf{Q}, a_1, a_2, a_3)$ and $f = (g, f_1, f_2, f_3)$. In this simpler notation, the differential equations are $\dot{y} = f(y)$, and the Euler method iteration formulas are $y_{s+1} = y_s + hf(y_s)$.

The Euler method generalizes to the higher-order Runge-Kutta methods (Appendix D). Like the Euler method, a Runge-Kutta method approximates the solution of the differential equation as a sequence of line segments (Fig. 4). However, a Runge-Kutta method evaluates the derivative \dot{y} (that is, the function f) several times per step, to better aim this line segment, so that it ends closer to the true solution. Concretely, the classic fourth-order Runge-Kutta method proceeds as follows, given $y_s = (\mathbf{Q}_s, a_{1,s}, a_{2,s}, a_{3,s})$:

1. Let $k_1 = f(y_s)$.
2. Let $k_2 = f(y_s + \frac{h}{2}k_1)$.
3. Let $k_3 = f(y_s + \frac{h}{2}k_2)$.
4. Let $k_4 = f(y_s + hk_3)$.
5. Then $y_{s+1} = y_s + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.

Any numerical approximation method for solving differential equations is subject to error. However, some errors are worse than others, because they drastically affect the geometry of the problem. For example, let

$$\mathbf{L} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -4 \end{bmatrix},$$

let \mathbf{E}_0 be a sphere of radius one, and let $r = 1$. After the first step of the Euler method with $n = 3$ steps (so $h = 0.33$), the semiaxis lengths are 2.33, 1, and -0.33 . The ellipsoid has degenerated to a hyperboloid, as in Fig. 3. Admittedly, this example is extreme, in that \mathbf{L} represents a high rate of strain and the step count n is low. A smaller \mathbf{L} or a smaller step size might at least preserve the clast as an ellipsoid. On the other hand, the chosen viscosity ratio r is not extreme, and we have not introduced the complication of non-coaxial deformation. It is not clear how small of a step size is needed to keep the clast an ellipsoid, and the necessary step size may vary over the course of the simulation. Even if the clast remains an ellipsoid, its shape and volume may deviate wildly from their true values (Section 3.4). The orientation \mathbf{Q} also accumulates error, such that \mathbf{Q} deviates from being a rotation and $\mathbf{Q}^\top \dot{\mathbf{E}} \mathbf{Q}$ is no longer the diagonalization of \mathbf{E} . More detailed performance analysis is given in Section 3.4.

The spheroid case, where two of the ellipsoid semi-axis lengths a_i and a_j are equal, is noteworthy. The typical expression (Eq. (A.20)) for \tilde{W}_{ij} becomes undefined. Jiang (2007a, 2012) handles this case by declaring that $(\dot{\mathbf{Q}}\mathbf{Q}^\top)_{ij} = 0$, or equivalently that $\tilde{W}_{ij} = (\mathbf{Q}\mathbf{W}\mathbf{Q}^\top)_{ij}$. In Appendix C we handle this case differently, by computing a limit. (See Mulchrone (2007) for a similar argument in two dimensions.) Of course, a random triaxial ellipsoid will deform into a spheroid only rarely. On the other hand, in an analysis of naturally deformed rock, one may wish to model the deformation of ellipsoids that are assumed to begin as spheres, where $a_1 = a_2 = a_3$. Such a simulation could be dramatically affected by the handling of the spheroid case.

3.2. Simulating the ellipsoid tensor (\mathbf{E} method)

In this section, we again use classic numerical methods to simulate the deforming ellipsoid. However, this approach simulates the ellipsoid tensor \mathbf{E} rather than treating the orientation and size separately. The governing differential equation is Eq. (A.15):

$$\dot{\mathbf{E}} = f(\mathbf{E}),$$

where $f(\mathbf{E}) = -\mathbf{K}^\top \mathbf{E} - \mathbf{E} \mathbf{K}$ is computable using Appendix A. By replacing y with \mathbf{E} in the preceding section, we obtain Euler and Runge-Kutta methods for evolving \mathbf{E} .

In the previous section, we note that simulating \mathbf{Q} and the a_i can cause significant errors in geometry. Analogous issues crop up in simulating \mathbf{E} . Namely, we wish \mathbf{E} to remain symmetric, positive-definite, and determinant-one (assuming that $\det \mathbf{E}_0 = 1$) at all times. By transposing Eq. (A.15), it is easy to see that $\dot{\mathbf{E}}$ is symmetric. It follows that the Euler and Runge-Kutta methods just described do preserve the symmetry of \mathbf{E} . On the other hand, they do not preserve positive-definiteness or determinant. Repeating the example of the previous section results in a hyperboloid after one step of this Euler method. So this approach to simulating the deforming ellipsoid is also unreliable for low step counts.

This \mathbf{E} method enjoys two small advantages over the (\mathbf{Q}, a_i) method. It does not require special handling of the spheroid case (Appendix C). Computer experiments also suggest that this approach slightly outperforms the (\mathbf{Q}, a_i) method (Section 3.4).

3.3. Simulating the finite deformation (\mathbf{F}^{-1} method)

The differential equation, Eq. (4), that governs the deformation of the ellipsoidal clast results in a homogeneous finite deformation: At any given time t , the cumulative effect up to that time can be represented as a finite deformation tensor (position gradient tensor) $\mathbf{F}(t)$ such that $\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(0)$ for all points \mathbf{x} . More briefly, we write $\mathbf{x} = \mathbf{F}\mathbf{x}(0)$. Combining this equation with Eq. (2), we have

$$\mathbf{x}(0)^\top \mathbf{E}(0) \mathbf{x}(0) = 1 = \mathbf{x}(0)^\top \mathbf{F}^\top \mathbf{E} \mathbf{F} \mathbf{x}(0).$$

Because this equation holds for all points \mathbf{x} on the ellipsoidal boundary, it follows that

$$\mathbf{E} = (\mathbf{F}^{-1})^\top \mathbf{E}(0) \mathbf{F}^{-1}. \quad (5)$$

This equation suggests that we can simulate \mathbf{E} by simulating \mathbf{F} or \mathbf{F}^{-1} and computing \mathbf{E} from \mathbf{F}^{-1} . It is well known that

$$\dot{\mathbf{F}} = \mathbf{K} \mathbf{F} \quad (6)$$

(see Bown (1989, p. 261), Pollard and Fletcher (2005, p. 178), etc.). We could apply the Euler method or Runge-Kutta method to this differential equation, to evolve \mathbf{F} and thus \mathbf{E} . However, we do not pursue that approach. Instead, we recognize that \mathbf{F} and \mathbf{F}^{-1} are curves in the

Lie group of volume-preserving homogeneous finite deformations, and we apply Lie group numerical methods. For clarity of exposition, we develop this strategy in several stages.

Suppose for the moment that \mathbf{K} is not time-dependent. That is, the ellipsoid's deformation is steady. Then $\mathbf{F}(t) = \exp t\mathbf{K}$, where \exp is the matrix exponential function (Provost et al., 2004; Davis and Titus, 2011). Equivalently, $\mathbf{F}^{-1} = \exp -t\mathbf{K}$. Thus

$$\mathbf{E} = (\exp -t\mathbf{K})^\top \mathbf{E}(0) (\exp -t\mathbf{K}). \quad (7)$$

This equation is a new method for simulating the deforming ellipsoid. Unlike the approaches of Sections 3.1 and 3.2, this method does preserve the key geometric properties of \mathbf{E} at all times t . First, transposing Eq. (7) shows that \mathbf{E} is symmetric. Second, the determinant is

$$\begin{aligned} \det \mathbf{E} &= \det(\exp -t\mathbf{K})^\top \det \mathbf{E}(0) \det(\exp -t\mathbf{K}) \\ &= (\det \exp -t\mathbf{K})^2 \det \mathbf{E}(0). \end{aligned}$$

Because $\det \exp \mathbf{M} = \exp \operatorname{tr} \mathbf{M}$ for all matrices \mathbf{M} ,

$$\det \mathbf{E} = (e^{\operatorname{tr} -t\mathbf{K}})^2 \det \mathbf{E}(0) = (e^{-t \operatorname{tr} \mathbf{K}})^2 \det \mathbf{E}(0).$$

However, \mathbf{K} has trace zero, because the deformation is volume-preserving (Appendix A). Thus $\det \mathbf{E} = \det \mathbf{E}(0)$. The constancy of the determinant implies that the ellipsoid's volume is preserved, and that \mathbf{E} never has zero as an eigenvalue. Finally, the eigenvalues of \mathbf{E} are always real, because \mathbf{E} is always symmetric. They depend continuously on \mathbf{E} , which depends continuously on t . They are positive at $t = 0$ and they are never zero. Thus they must be positive at all times t .

The foregoing description assumes that the ellipsoid deformation is steady, which is not realistic. However, as we discuss in Section 3.1, the key idea of the Euler method is to solve a non-steady differential equation by taking such steady steps. So let us again use time steps of size $h = 1/n$, and compute approximations $\mathbf{E}_s \approx \mathbf{E}(sh)$ to the ellipsoid at time $t = sh$. Let \mathbf{K}_s denote the velocity gradient tensor \mathbf{K} computed from \mathbf{E}_s . Running Eq. (7) for one time step yields

$$\mathbf{E}_{s+1} = (\exp -h\mathbf{K}_s)^\top \mathbf{E}_s (\exp -h\mathbf{K}_s).$$

This formula is an Euler method for \mathbf{E} , distinct from that of Section 3.2. Because each step of this new method preserves the geometric properties of \mathbf{E} , so does the entire method. That is, the final ellipsoid tensor $\mathbf{E}_n \approx \mathbf{E}(1)$ is guaranteed to be symmetric, positive-definite, and of the same volume as the initial ellipsoid $\mathbf{E}_0 = \mathbf{E}(0)$.

The corresponding Runge-Kutta method for \mathbf{E} is not obvious. To derive it, we first use Eq. (5) to rewrite the Euler method as

$$\begin{aligned} & (\mathbf{F}_{s+1}^{-1})^\top \mathbf{E}(0) \mathbf{F}_{s+1}^{-1} \\ &= (\exp -h\mathbf{K}_s)^\top (\mathbf{F}_s^{-1})^\top \mathbf{E}(0) \mathbf{F}_s^{-1} (\exp -h\mathbf{K}_s), \end{aligned}$$

where \mathbf{F}_s is an approximation to $\mathbf{F}(sh)$. This equation is equivalent to

$$\mathbf{F}_{s+1}^{-1} = \mathbf{F}_s^{-1} \exp -h\mathbf{K}_s, \quad (8)$$

which is an Euler method for the inverse cumulative finite deformation \mathbf{F}^{-1} . Notice that, if \mathbf{F}_s has determinant one, then so does \mathbf{F}_{s+1} , because \mathbf{K}_s has trace zero. Each step of this Euler method preserves the most basic geometric property of \mathbf{F} : its volume preservation.

The fact that these methods respect the geometry of the deformation is no accident. It is a symptom of an underlying mathematical structure called *Lie theory* (see, e.g., Belinfante and Kolman, 1972; Gilmore, 1974; Curtis, 1984; Hall, 2003; Pollatsek, 2009). In any rock deformation problem, the set of finite deformations forms a *Lie group*, the set of instantaneous deformations forms the associated *Lie algebra*, and an *exponential* function maps the latter into the former. In the context of homogeneous deformations, the Lie group is the set of position gradient tensors (finite deformation tensors), the Lie algebra is the set of velocity gradient tensors, and the exponential is the matrix exponential.

If we restrict our attention to volume-preserving homogeneous deformations, then the Lie group is the *special linear group*, denoted $\operatorname{SL}(3, \mathbb{R})$, which is the set of all (real, 3×3) matrices of determinant one. The Lie algebra, denoted $\mathfrak{sl}(3, \mathbb{R})$, is the set of all matrices of trace zero. The Lie group is a curved, eight-dimensional hypersurface in the nine-dimensional Euclidean space of all 3×3 matrices (Fig. 5). The classical Euler and Runge-Kutta methods on matrices travel along straight lines in the Euclidean space, and hence cannot stay in the curved Lie group. In contrast, the Euler method of Eq. (8) travels along certain *geodesic curves* in the Lie group. It is a natural generalization of the Euclidean Euler method to a curved setting. Although it suffers from some error, like any approximation, its error is entirely within the set of physically relevant deformation tensors. Consequently, when we simulate E in terms of F^{-1} using Eq. (8), our results are guaranteed to satisfy our basic physical expectations.

We do not attempt to summarize Lie theory here. For our purposes, it is enough simply to recognize the evolution of \mathbf{F} or \mathbf{F}^{-1} as a differential equation on a Lie group, so that we can apply Lie group numerical methods. We follow the fourth-order Runge-Kutta method described by Munthe-Kaas (1998) and summarized in Appendix D. The method uses the *bracket* operation, which for any two matrices \mathbf{M} and \mathbf{N} is defined as $[\mathbf{M}, \mathbf{N}] = \mathbf{M}\mathbf{N} - \mathbf{N}\mathbf{M}$. Denote by $f(\mathbf{M})$ the function that takes a matrix \mathbf{M} as input, computes $\mathbf{E} = \mathbf{M}^\top \mathbf{E}(0) \mathbf{M}$, and then computes $-\mathbf{K}$ from \mathbf{E} using Appendix A. Here is the Runge-Kutta method over a single time step of size $h = 1/n$.

1. Let $k_1 = f(\mathbf{F}_s^{-1})$.
2. Let $k_2 = f(\mathbf{F}_s^{-1} \exp \frac{h}{2} k_1)$.
3. Let $k_3 = f\left(\mathbf{F}_s^{-1} \exp\left(\frac{h}{2} k_2 + \frac{h^2}{24} [k_1, k_2]\right)\right)$.

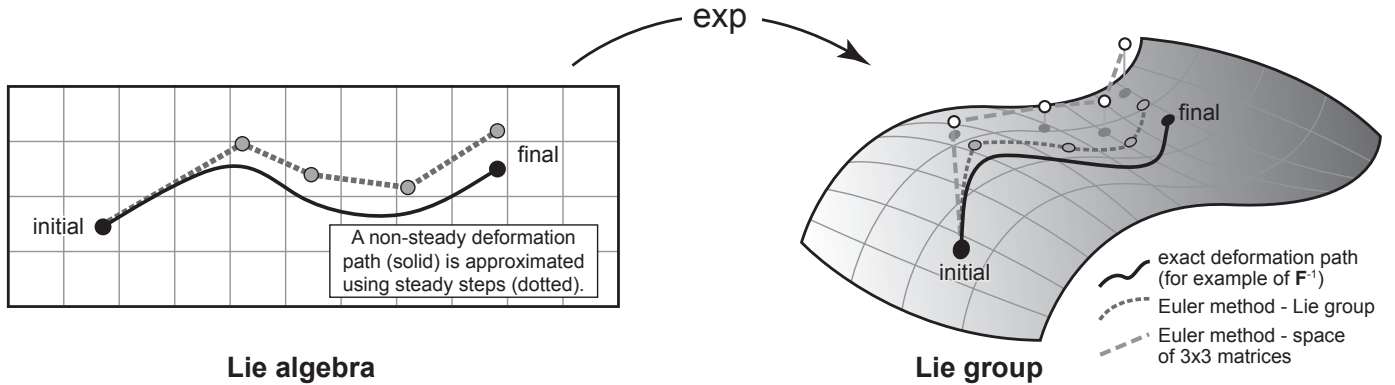


Figure 5: Cartoon of computing a deformation path in a Lie group. A solution using classic numerical methods on matrices departs from the group, while a solution using Lie group methods remains in the group at all times. The Lie group methods can be viewed as classic methods operating on the Lie algebra (Munthe-Kaas, 1999).

4. Let $k_4 = f\left(\mathbf{F}_s^{-1} \exp\left(hk_3 + \frac{h^2}{6}[k_1, k_3]\right)\right)$.
5. Let $v = \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.
6. Let $w = \frac{h}{24}(3k_1 + 2k_2 + 2k_3 - k_4)$.
7. Then

$$\mathbf{F}_{s+1}^{-1} = \mathbf{F}_s^{-1} \exp(v + [w, v]). \quad (9)$$

To gain some intuition for this algorithm, pretend for the moment that matrix multiplication is commutative. Then all of the bracket terms are zero, and w is unnecessary. If we strike these parts of the algorithm, then what remains closely resembles the Euclidean Runge-Kutta method presented in Section 3.1. However, whereas each part of that method resembles the Euclidean Euler method, each part of the Lie group Runge-Kutta method resembles the Lie group Euler method (Eq. (8)). So, intuitively, the Lie group Runge-Kutta method is analogous to the Euclidean Runge-Kutta method, but with corrections for the curvature and the non-commutativity of the Lie group.

Here is our final method for simulating the deforming ellipsoid, stated explicitly. We begin with $\mathbf{E}_0 = \mathbf{E}(0)$ and $\mathbf{F}_0^{-1} = \mathbf{I}$, the identity tensor. For $s = 1, \dots, n$, we compute \mathbf{F}_s^{-1} using the Lie group Runge-Kutta method. Then we compute $(\mathbf{F}_n^{-1})^\top \mathbf{E}(0) \mathbf{F}_n^{-1} = \mathbf{E}_n \approx \mathbf{E}(1)$.

3.4. Performance comparisons

We have discussed three basic strategies for numerically solving the deforming ellipsoid problem: a (\mathbf{Q}, a_i) method (Section 3.1), an \mathbf{E} method (Section 3.2), and a Lie group \mathbf{F}^{-1} method (Section 3.3). For each strategy we have described an Euler method and a fourth-order Runge-Kutta method, so we have six methods in all. In this section we report the results of computer tests on randomly generated instances of the problem. We compare the six methods in terms of speed, overall precision, and precision in preserving volume.

For each test, we randomly generate an instance $(\mathbf{E}, \mathbf{L}, r)$ of the problem, by choosing an orientation \mathbf{Q} , choosing $0.25 \leq a_1, a_2 \leq 4$, setting $a_3 = (a_1 a_2)^{-1}$, choosing $-2 \leq L_{ij} \leq 2$, and choosing $0.1 \leq r \leq 100$ such that

	(\mathbf{Q}, a_i)	\mathbf{E}	\mathbf{F}^{-1}
Euler	0.0106	0.0106	0.0107
Runge-Kutta	0.0452	0.0472	0.0455

Table 1: Average computer processor time per step, in seconds, for the six deformable ellipsoid methods.

$\log_{10} r$ is uniform. We compute three approximately equal answers for the final ellipsoid $\mathbf{E}(1)$, by running the three Runge-Kutta methods with increasing step count n until their answers disagree by no more than 10^{-5} . For speed and simplicity we measure the distance between any two ellipsoid tensors \mathbf{E} and \mathbf{E}' using the Frobenius norm of their difference:

$$|\mathbf{E} - \mathbf{E}'| = \sqrt{\text{tr}(\mathbf{E} - \mathbf{E}')^\top (\mathbf{E} - \mathbf{E}')}$$

(Appendix E describes another way to measure distance, that handles the curved geometry of the problem more precisely.) Then we run each of the six methods repeatedly, with increasing step count n . For each n , we record the computer processor time required to compute $\mathbf{E}(1)$, the distance from that $\mathbf{E}(1)$ to the nearest precomputed correct answer, and the determinant of that $\mathbf{E}(1)$.

We have run 375 such tests. For each method, Table 1 shows the average computer processor time required per step. We have attempted to control for various sources of noise and bias in these timing data. The six methods are implemented in the same language and library (Mathematica), at similar levels of abstraction and optimization. As the table shows, the Euler methods all require approximately equal time per step, and so do the Runge-Kutta methods. The Runge-Kutta methods require about four or five times as much time per step as the Euler methods. We expect these patterns to hold for other implementations, at least approximately, with the specific numbers varying by implementation. Henceforth, we state all deformable ellipsoid test results in terms of step count rather than time.

Fig. 6 summarizes the performance of all six methods,

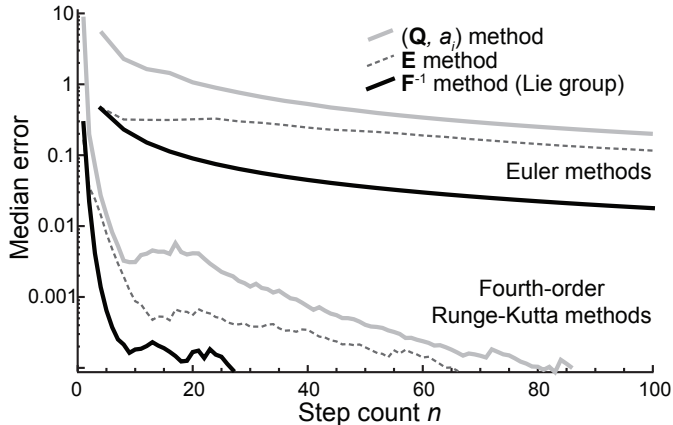


Figure 6: Median error of all six methods for the deformable ellipsoid.

by showing, for each step count n , the median error in our 375 tests. This graph can be read in two ways: the precision attainable in a given step count, or the step count required to achieve a given precision. The Runge-Kutta methods perform distinctly better than the Euler methods, even if we account for the Euler methods' requiring one fourth the processor time per step. This situation is typical of Euler and Runge-Kutta methods (e.g., Gerald and Wheatley, 1984, p. 308), and justifies the increased programming difficulty of the latter. Henceforth we focus on the Runge-Kutta methods.

It is also clear from Fig. 6 that the Lie group \mathbf{F}^{-1} approach significantly outperforms the (\mathbf{Q}, a_i) and the \mathbf{E} approaches. Fig. 7 compares the performance of these three Runge-Kutta methods in greater detail. For each n , the two panels show the errors of the (\mathbf{Q}, a_i) and \mathbf{E} methods divided by the error of the \mathbf{F}^{-1} method. For $n = 20$, for example, the \mathbf{E} method produces at worst about 100 times the error of the \mathbf{F}^{-1} method, while the (\mathbf{Q}, a_i) method produces at worst about 10,000 times the error of the \mathbf{F}^{-1} method. Whether to achieve a given precision in the shortest possible time or to achieve the highest possible precision in a given time, the \mathbf{F}^{-1} method is preferable to the other methods.

Fig. 8 shows the worst-case volume-preservation behavior of the three Runge-Kutta methods, by showing their maximum and minimum computed values for the determinant of \mathbf{E} , in our 375 tests. The determinant should always be one. The (\mathbf{Q}, a_i) and \mathbf{E} Runge-Kutta methods may deviate far from one at low step counts n , but do not deviate significantly from one after about $n = 20$. The \mathbf{F}^{-1} Runge-Kutta method preserves the determinant exactly, even at low step counts.

4. Simulation of rigid ellipsoids

As the viscosity contrast r goes to infinity, the deformable ellipsoid model reproduces the rigid ellipsoid model of Jeffery (1922) (Freeman, 1987; Jiang, 2007b). We compute this limiting case in Appendix A. In short,

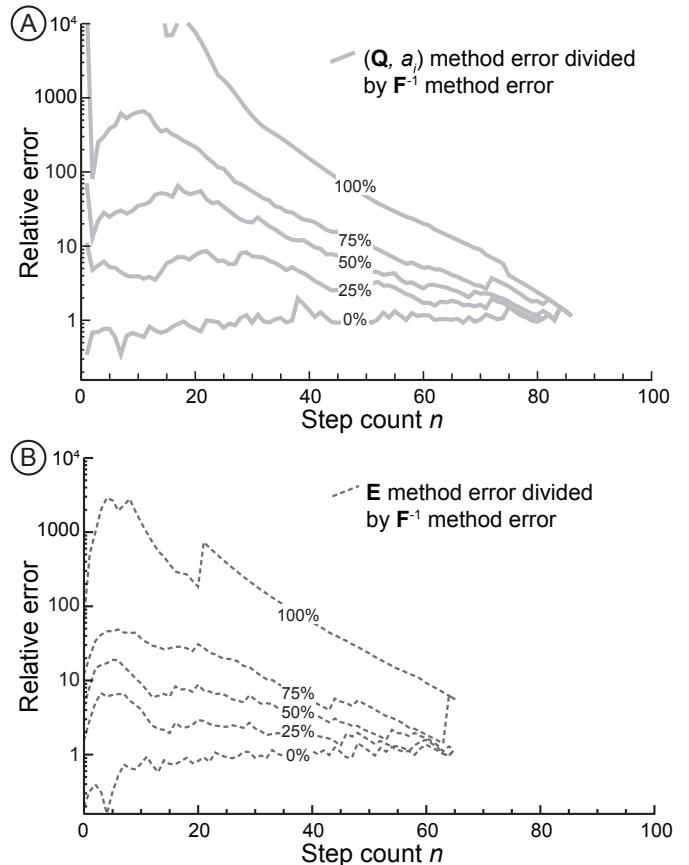


Figure 7: The error of (A) the (\mathbf{Q}, a_i) Runge-Kutta method and (B) the \mathbf{E} Runge-Kutta method, relative to the \mathbf{F}^{-1} error. The five curves indicate percentiles in the 375 tests.

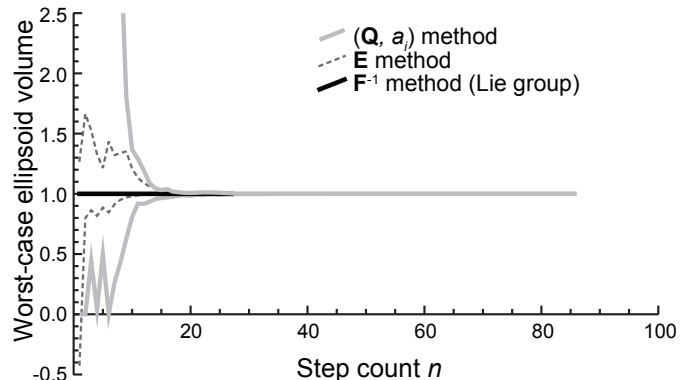


Figure 8: Worst-case determinants of the deformable ellipsoid tensor \mathbf{E} .

the stretching component of deformation vanishes, and the vorticity component simplifies. In this section, we describe and compare two approaches for simulating the rotation of these rigid ellipsoids.

The first approach is simply that of Section 3.1, specifically Eq. (A.17). Recall that this approach evolves \mathbf{Q} as a 3×3 matrix, using Euclidean numerical methods. As one would expect, Eq. (A.16), which governs the evolution of the a_i , reduces to $\dot{a}_i = 0$. Although the special case

	Euclidean	Lie
Euler	0.000134	0.000171
Runge-Kutta	0.000553	0.000759

Table 2: Computer processor time per step, in seconds, for the four rigid ellipsoid methods.

$a_i = a_j$ is problematic in this approach (Eq. (A.21)), the problem vanishes as $r \rightarrow \infty$ (Eq. (A.24)).

The second approach treats \mathbf{Q} as an element of a Lie group, in a manner analogous to Section 3.3. The Lie group in question is no longer the volume-preserving homogeneous deformations $SL(3, \mathbb{R})$. It is now the smaller group $SO(3)$, consisting of finite rotations — matrices \mathbf{R} such that $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ and $\det \mathbf{R} = 1$. The Lie algebra is no longer $\mathfrak{sl}(3, \mathbb{R})$ but rather the smaller Lie algebra $\mathfrak{so}(3)$ consisting of instantaneous rotations — matrices $\mathbf{\Omega}$ such that $\mathbf{\Omega}^T = -\mathbf{\Omega}$. The matrix exponential sends $\mathfrak{so}(3)$ to $SO(3)$.

Here is the Lie group approach in detail. The diagonalization $\mathbf{E} = \mathbf{Q}^T \tilde{\mathbf{E}} \mathbf{Q}$, combined with Eq. (5), yields

$$\mathbf{Q}_s^T \tilde{\mathbf{E}} \mathbf{Q}_s = (\mathbf{F}_s^{-1})^T \mathbf{Q}_0^T \tilde{\mathbf{E}} \mathbf{Q}_0 \mathbf{F}_s^{-1}.$$

It follows that $\mathbf{Q}_s = \mathbf{Q}_0 \mathbf{F}_s^{-1}$. Then, using the Lie group Euler method for \mathbf{F}^{-1} (Eq. (8)), we obtain

$$\mathbf{Q}_{s+1} = \mathbf{Q}_0 \mathbf{F}_{s+1}^{-1} = \mathbf{Q}_0 \mathbf{F}_s^{-1} \exp -h\mathbf{K}_s = \mathbf{Q}_s \exp -h\mathbf{K}_s.$$

This is the Euler method for \mathbf{Q} . It is of the same form as Eq. (8), with \mathbf{Q} filling the role of \mathbf{F}^{-1} . Replacing \mathbf{F}^{-1} with \mathbf{Q} in the Runge-Kutta method Eq. (9) yields the corresponding Runge-Kutta method for \mathbf{Q} . Either of these methods, combined with $\mathbf{E} = \mathbf{Q}^T \tilde{\mathbf{E}} \mathbf{Q}$, yields a Lie group method for the rigid ellipsoid tensor \mathbf{E} .

Thus, we have two approaches to simulating the rotation of rigid ellipsoids, each in an Euler version and a Runge-Kutta version, for four methods in all. In a testing procedure similar to that of Section 3.4, we have tested the four methods on 2,500 randomly generated instances of the rigid ellipsoid problem. Table 2 shows the computer processor time required for one step of each of the four methods. Although these timing data are implementation-dependent, some patterns are noteworthy. As in the deformable case (Table 1), the Runge-Kutta methods require four or five times as much processor time per step as the corresponding Euler methods. Unlike the deformable methods, the rigid Lie group methods require about one-third more time per step than their Euclidean counterparts. The extra time arises primarily from computing brackets and exponentials.

Fig. 9(A) summarizes the performance of the four methods based on the step count n . As expected, the Runge-Kutta methods outperform the Euler methods dramatically. The Lie group Euler method significantly outperforms the Euclidean Euler method. On the other hand, the Euclidean Runge-Kutta method matches or beats the Lie group Runge-Kutta method, especially once processor

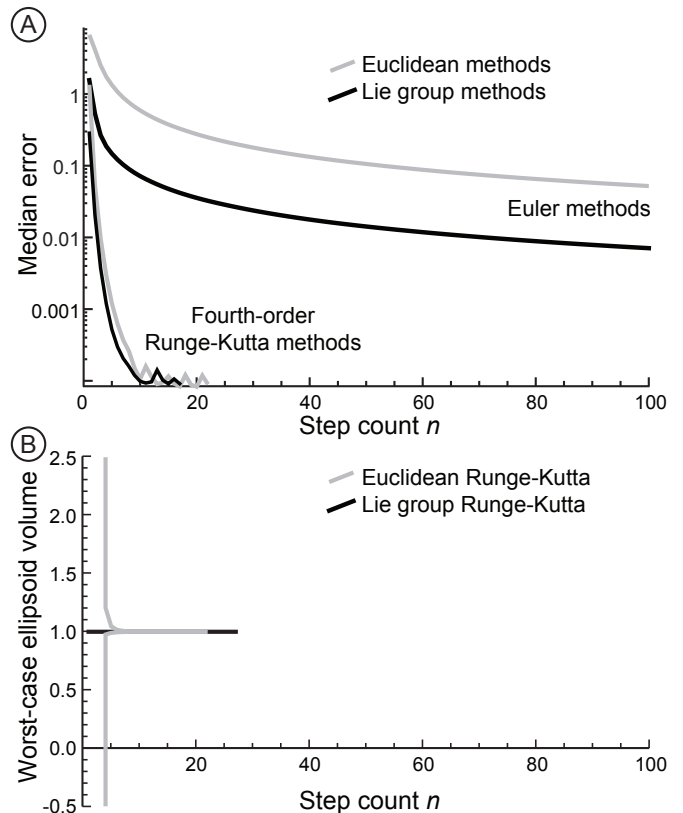


Figure 9: (A) Median error of the four rigid methods. (B) Worst-case error in the ellipsoid volume, for the two Runge-Kutta rigid methods.

time is taken into account. Fig. 9(B) shows the worst-case volume-preservation behavior of the two Runge-Kutta methods, by showing their maximum and minimum computed values for $\det \mathbf{E}$, which should always be one, in our 2,500 tests. As in the deformable case (Fig. 8), the Lie group method never deviates from determinant one. The Euclidean method does not deviate significantly from determinant one if at least $n = 5$ steps are used.

In summary, one can apply Lie group methods to the simulation of rigid ellipsoids as well as deformable ellipsoids. However, Lie group methods do not appear to enjoy any practical advantage over simpler, Euclidean methods in the rigid case.

5. Application to the Gem Lake shear zone

As an application of our deformable ellipsoid techniques, we reanalyze three-dimensional strain and fabric data of Horsman et al. (2008) from the Gem Lake shear zone in the Sierra Nevada mountains of California. Our goal is to quantify the outcrop-scale deformation in the shear zone based on the behavior of deformed ellipsoidal clasts. We first describe the geologic setting, data, and model. Then, we describe numerical experiments showing that our inverse modeling approach is feasible. Finally, we apply the approach to the Gem Lake shear zone data.

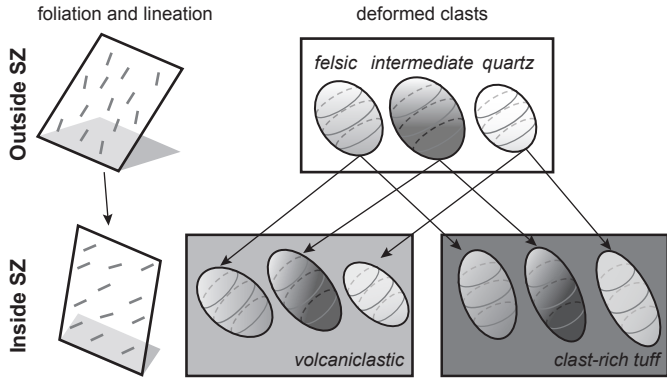


Figure 10: Cartoon of the Gem Lake shear zone modeling process. Each of seven subproblems is represented by an arrow. We seek the deformation that best fits all seven subproblems simultaneously.

5.1. Geologic setting and data

The Gem Lake shear zone is part of the dextral transpressional Late Cretaceous Sierra Crest shear zone system in the Sierra Nevada of California (Tikoff and Greene, 1997). In the vicinity of Gem Lake, the shear zone has a strike and dip of about 150° and 78° . The shear zone is about 1 km wide, and it may have accommodated 20 km or more of dextral offset (Greene and Schweickert, 1995).

Horsman et al. (2008) reported measurements of lineation and foliation from volcaniclastic rocks and clast-rich tuffs inside the shear zone, and from rocks outside the shear zone. The lineation outside the shear zone pitches steeply down dip, while inside the zone it pitches moderately to the northwest. The rocks contain clasts (idealized as ellipsoids) of felsic, intermediate, quartz, and pumice composition. Horsman et al. (2008) reported measurements, including both ellipsoid orientation and magnitude, of each of these clast types, inside and outside the shear zone.

5.2. Model

We assume that the large-scale shear zone deformation was steady, homogeneous, and volume-preserving. Choose time units so that this deformation runs from $t = 0$ to $t = 1$, and let \mathbf{L} be its as-yet-unknown velocity gradient tensor. The rocks outside the shear zone may have been deformed prior to the shear zone deformation. We assume that the prior deformation was homogeneous and volume-preserving. We seek the \mathbf{L} and rock viscosity ratios that best solve seven subproblems related to lineation and foliation, and the three clast types within each of the two host rock types (Fig. 10).

The first subproblem asks \mathbf{L} to match the lineations and foliations. We assume that those observed outside the shear zone are aligned with the finite strain axes of the prior deformation. Using Appendix E, we average the two lineation-foliation pairs inside the shear zone into a single lineation-foliation pair, which we assume to be aligned with the finite strain axes of the sequential superposition

of the prior and shear zone deformations. Let \mathbf{E}_0 be the ellipsoid tensor for the finite strain ellipsoid outside the shear zone, and \mathbf{E}_1 the finite strain ellipsoid tensor inside the shear zone. Then $\mathbf{E}_1 = (\exp -\mathbf{L})^\top \mathbf{E}_0 (\exp -\mathbf{L})$. (This equation is the passive, steady case of Eq. (5).) Unfortunately, we do not know the magnitudes of these ellipsoids, but only their orientations. Working in fixed geographic coordinates, let \mathbf{l}_0 and \mathbf{f}_0 be unit vectors aligned with the lineation and foliation pole outside the shear zone, and \mathbf{R}_0 the matrix whose rows are \mathbf{f}_0 , $\mathbf{l}_0 \times \mathbf{f}_0$, and \mathbf{l}_0 . Let $b_1 \leq b_2 \leq b_3 = (b_1 b_2)^{-1}$ be the unknown finite strain axes of the prior deformation, and

$$\Delta_0 = \begin{bmatrix} b_1^{-2} & 0 & 0 \\ 0 & b_2^{-2} & 0 \\ 0 & 0 & b_1^2 b_2^2 \end{bmatrix},$$

so that $\mathbf{E}_0 = \mathbf{R}_0^\top \Delta_0 \mathbf{R}_0$. Similarly, form a matrix \mathbf{R}_1 from the averaged lineation and foliation inside the shear zone, so that $\mathbf{E}_1 = \mathbf{R}_1^\top \Delta_1 \mathbf{R}_1$ for an unknown Δ_1 . Then

$$\Delta_1 = \mathbf{R}_1 (\exp -\mathbf{L})^\top \mathbf{R}_0^\top \Delta_0 \mathbf{R}_0 (\exp -\mathbf{L}) \mathbf{R}_1^\top.$$

That is, the deformation \mathbf{L} matches the lineation and foliation data $\mathbf{R}_0, \mathbf{R}_1$ if and only if $b_1 \leq b_2 \leq (b_1 b_2)^{-1}$ exist to make the symmetric matrix Δ_1 diagonal with decreasing diagonal entries. This condition amounts to three scalar equations and four inequalities on \mathbf{L} and the b_i .

The remaining six subproblems ask \mathbf{L} to match the deformed clast data. Using Appendix E, the ellipsoids for each combination of clast type and matrix rock type can be averaged into a single ellipsoid. We assume that the clasts outside and inside the shear zone represent a single clast population before and after the shear zone deformation (Fig. 10). For example, the shear zone deformation has deformed the felsic ellipsoid outside the shear zone to the felsic-in-volcanoclastic ellipsoid, using some unknown felsic/volcanoclastic viscosity ratio. Also, the deformation changed the felsic-outside ellipsoid to the felsic-in-tuff ellipsoid using some unknown felsic/tuff viscosity ratio, and similarly for the intermediate and quartz ellipsoids. We ignore the pumice clast data, because Horsman et al. (2008) suspect volume loss in those clasts, and our deforming ellipsoid model assumes volume preservation. For any given \mathbf{L} and set of viscosity ratios, we can simulate these six deforming ellipsoids, using the Lie group \mathbf{F}^{-1} method of Section 3.3. (We use $n = 10$ steps, for a step size of $h = 0.1$.) The ellipsoids computed at $t = 1$ should match the observed clast ellipsoids. Each of these six subproblems amounts to one equation of ellipsoid tensors, and hence six scalar equations on \mathbf{L} and the unknown viscosity ratios.

In summary, the parameters of our model are the velocity gradient tensor \mathbf{L} , five unknown viscosities (felsic, intermediate, quartz, volcaniclastic, and tuff), and two finite strain axis magnitudes b_1 and b_2 . Because \mathbf{L} is thus far required only to preserve volume, it may have as many as eight degrees of freedom, for a total of up to 15 degrees of freedom in the model. The model must satisfy 39

highly nonlinear equations, as well as four inequalities for the finite strain axis magnitudes. We can expect no exact solution. However, using numerical optimization software such as Mathematica, we can find a least-squares best-fit solution, as in Davis and Titus (2011, Section 5.2). Appendix E explains one last detail: In each of the seven subproblems that lead to the 39 equations, we compute the error in the *logarithms* of the matrices, rather than the matrices themselves. For example, we ask $\log \mathbf{\Delta}_1$, rather than $\mathbf{\Delta}_1$, to be diagonal.

Although this approach can be applied to any volume-preserving \mathbf{L} , transpression deformations are most relevant to the geologic setting. The dip of the shear zone, and the fact that the lineations are neither vertical nor horizontal, suggests that monoclinic transpression is not an appropriate model. Like Horsman et al. (2008), we model the shear zone as an inclined transpression (Lin et al., 1998; Jones et al., 2004; Davis and Titus, 2011):

$$\mathbf{L} = \mathbf{S} \begin{bmatrix} 0 & -\frac{\cot \alpha}{\cos \delta} \log(1-s) & 0 \\ 0 & \log(1-s) & 0 \\ 0 & -\tan \delta \log(1-s) & -\log(1-s) \end{bmatrix} \mathbf{S}^\top,$$

where

$$\mathbf{S} = \begin{bmatrix} \cos \sigma & -\sin \sigma & 0 \\ \sin \sigma & \cos \sigma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \delta & -\sin \delta \\ 0 & \sin \delta & \cos \delta \end{bmatrix},$$

the strike of the shear plane is $-\pi/2 - \sigma$, the dip of the shear plane is $\pi/2 - \delta$ (in a right-handed sense), α is the angle of oblique convergence, and $0 < s < 1$ is the fractional shortening across the shear zone (Fig. 11). Hence there are four parameters in \mathbf{L} and 11 parameters overall.

We impose various additional constraints on the parameters, to reflect the geometry of the Gem Lake shear zone. The strike and dip of the shear plane are constrained between 145° and 155° and between 73° and 83° , respectively. The angle of oblique convergence is between 1° and 89° , representing all possible dextral deformations, while the shortening is between 0.1 and 0.9. The five viscosities are assumed to be between 0.01 and 100. (The units on these viscosities are not important, because only their ratios enter into the problem.) Thus, we consider viscosity ratios as small as 10^{-4} and as large as 10^4 .

5.3. Numerical experiments

Before we apply this model to the Gem Lake shear zone data, we describe two numerical experiments. Both experiments use randomly generated, synthetic data sets, of the same form as the Gem Lake shear zone data set. The experiments demonstrate that our inverse modeling approach is capable of recovering the deformation from such data, and give some idea of how variations in the data affect the uncertainty of the best-fit model.

The first experiment consists of 107 tests. In each test, we randomly generate a fictitious initial state and shear

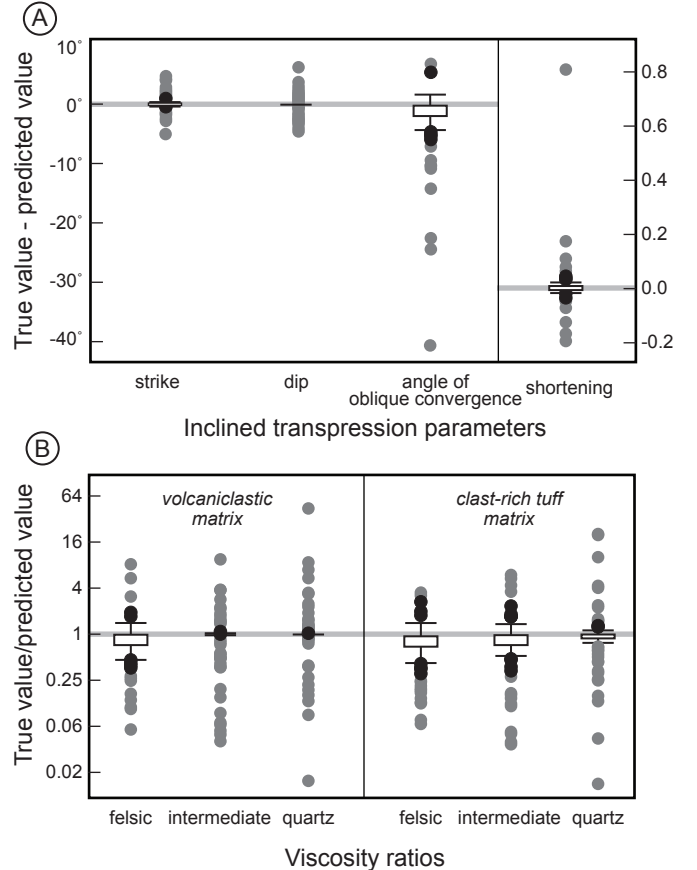
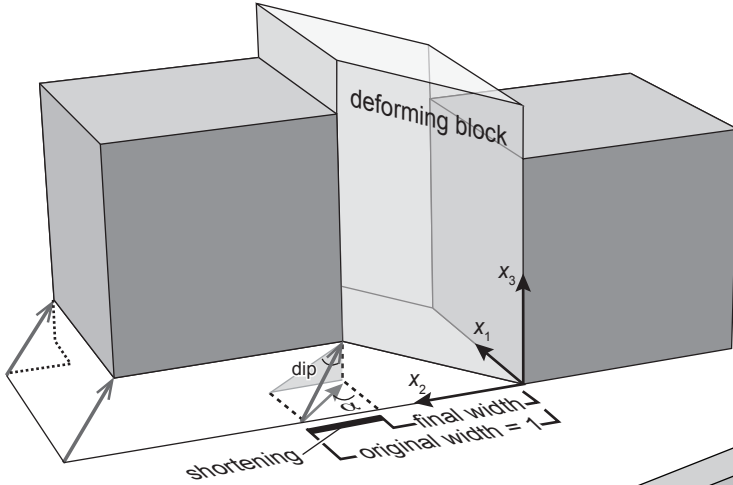


Figure 12: Box-and-whisker plots for the first numerical experiment. (A) The difference between the true and predicted values for strike, dip, angle of oblique convergence, and shortening. (B) The ratio between the true and predicted viscosity ratios.

zone deformation. (The transpression parameters are chosen uniformly from their constraint intervals. The ellipsoid orientations are chosen uniformly, and their semi-axis lengths are chosen between 0.33 and 3.0 such that their logarithms are uniform. The clast viscosities are chosen between 1 and 100, and the matrix viscosities between 0.01 and 1, such that their logarithms are uniform.) We forward-model to obtain the resulting clast ellipsoids and finite strain ellipsoid inside the shear zone. Then we inverse-model the deformation and viscosities from the data inside and outside the shear zone, exactly as described in the preceding section. Finally, we compare the best-fit parameter values from the inverse model to their true values chosen at the start of the test.

Fig. 12 shows box-and-whisker plots for the results of the first experiment. Panel (A) shows the differences between the true and the predicted values for the inclined transpression parameters. The predicted values closely match their true values, but there is more inaccuracy in the angle of convergence and shortening, than in the strike and dip of the shear plane. Panel (B) shows the ratios between the true and predicted viscosity ratios. The model almost always recovers the viscosity ratios to within a factor of four.

(A) example of inclined transpression in local coordinates



KEY

- total displacement of rigid block
- components of displacement (1 pure shear and 2 simple shears)
- lines parallel to components of local coordinate system
- α angle of oblique convergence
- x_1, x_2, x_3 local coordinate system

(B) same deformation in geographic coordinates

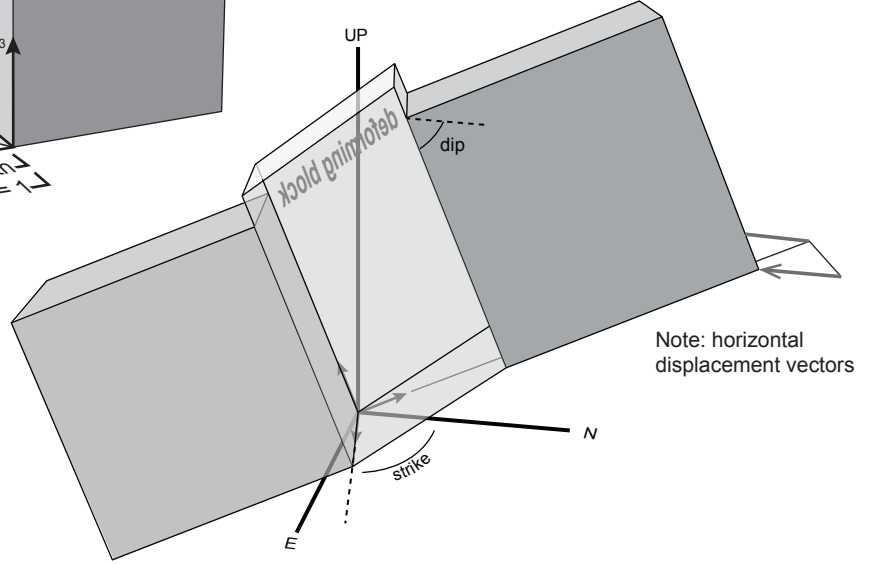


Figure 11: Diagram illustrating the kinematic model of inclined transpression used for the Gem Lake shear zone. There are four degrees of freedom related to inclined transpression that we incorporate into our modeling: two come from the shear zone coordinate system (one for strike, one for dip), one comes from the amount of convergence, and one comes from the angle of convergence. Modified from Jones et al. (2004).

One particular test accounts for the most extreme inaccuracies in angle of oblique convergence, shortening, and several of the viscosity ratios. We conclude that our inverse modeling approach and numerical optimization software are capable of recovering deformation from data like those of Horsman et al. (2008).

The second experiment consists of 1,000 tests. In each test, we randomly generate an initial state and deformation, and compute the final state from them, as in the first experiment. The error squared of the deformation, in matching the data, is of course zero. We perturb the orientations of the 11 ellipsoids, by rotating each one through 5° about a random axis in space, and observe how the error squared of the deformation deviates from zero. We also perturb the magnitudes of the clast ellipsoids, by raising each semi-axis length to a power of $(1.03)^5$. This perturbation is designed to exaggerate the shapes of the clasts by roughly 10%, while preserving their volumes. As we perturb the magnitudes, we again observe how the error squared of the deformation changes. Finally, we perturb the orientations and magnitudes simultaneously, and observe how the error squared changes. We regard these 5° and/or 10% perturbations to be “small” variations in the data. Probably, many field measurements of ellipsoids con-

tain this much error. We then repeat the entire process with “large” variations in the data: 10° perturbations in orientations and roughly 20% perturbations in clast magnitudes.

In 95% of the small-variation results, the error squared of the deformation does not exceed 3.37. In 95% of the large-variation results, the error squared does not exceed 15.9.

5.4. Results

Now that we have demonstrated the feasibility of our inverse-modeling approach, we apply this approach to the Gem Lake shear zone data. Computing the best-fit inclined transpression requires about 180,000 simulations of deformable ellipsoids, for about 17 hours of processor time on our computer. The error squared of the best fit is 3.05. This number has no physical meaning, but recall from Section 5.3 that small variations in the data tend to produce error squared up to 3.37, and large variations produce error squared up to 15.9. The following figures include “pseudo-confidence regions” based on those two values. We emphasize that these confidence regions are not statistically rigorous. Further, they are constructed by letting a single parameter vary, while other parameters are held fixed

	Constraint	Best Fit	Pseudo-CI
Strike	145°-155°	155°	153°-163°
Dip	73°-83°	83°	75°-89°
Ang. Obl. Conv.	1°-89°	7°	4°-10°
Shortening	0.1-0.9	0.27	0.18-0.34
Felsic	0.01-100	58	30-142
Intermediate	0.01-100	43	25-90
Quartz	0.01-100	62	33-167
Volcaniclastic	0.01-100	8	4-14
Clast-Rich Tuff	0.01-100	14	8-23

Table 3: Model parameters: the imposed constraint, the best-fit value, and the small-variation pseudo-confidence interval defined by an error squared of 3.37.

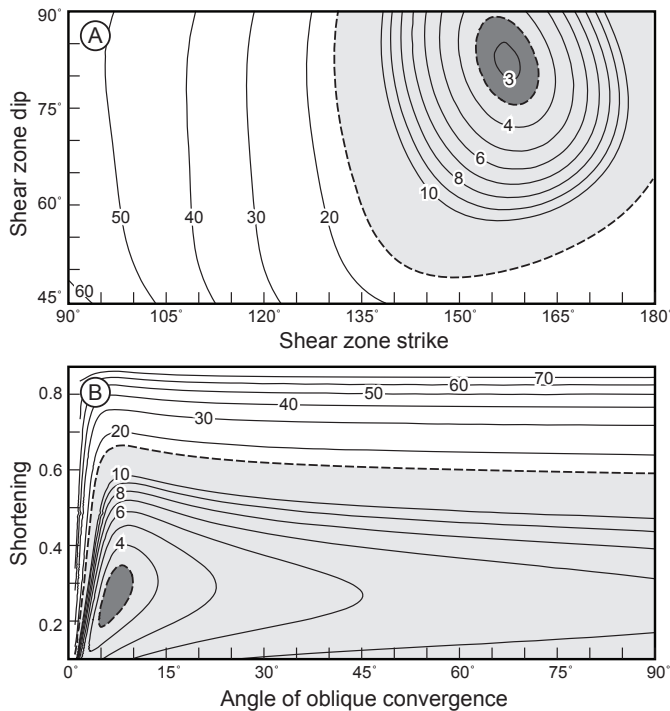


Figure 13: The effect of the inclined transpression parameters on the model fitness. The contours show the error squared of the fit. The pseudo-confidence regions defined by the values 3.37 and 15.9 are shaded. (A) Strike and dip of the shear plane. (B) Angle of oblique convergence and shortening.

to their best-fit values. So they describe the uncertainty in the best-fit parameters incompletely at best. The constraints and best-fit values for the model parameters are summarized in Table 3, along with small-variation pseudo-confidence intervals derived from the figures.

Fig. 13 is one view of how the fitness of the model varies around the best fit. Panel (A) shows contours of error squared, as the strike and dip vary, with all other parameters held fixed to their best-fit values. The lowest error squared occurs near a strike of 158° and a dip of 82°. This point lies outside our strike and dip constraints, which explains why the best-fit optimization saturates those constraints. Similarly, panel (B) shows con-

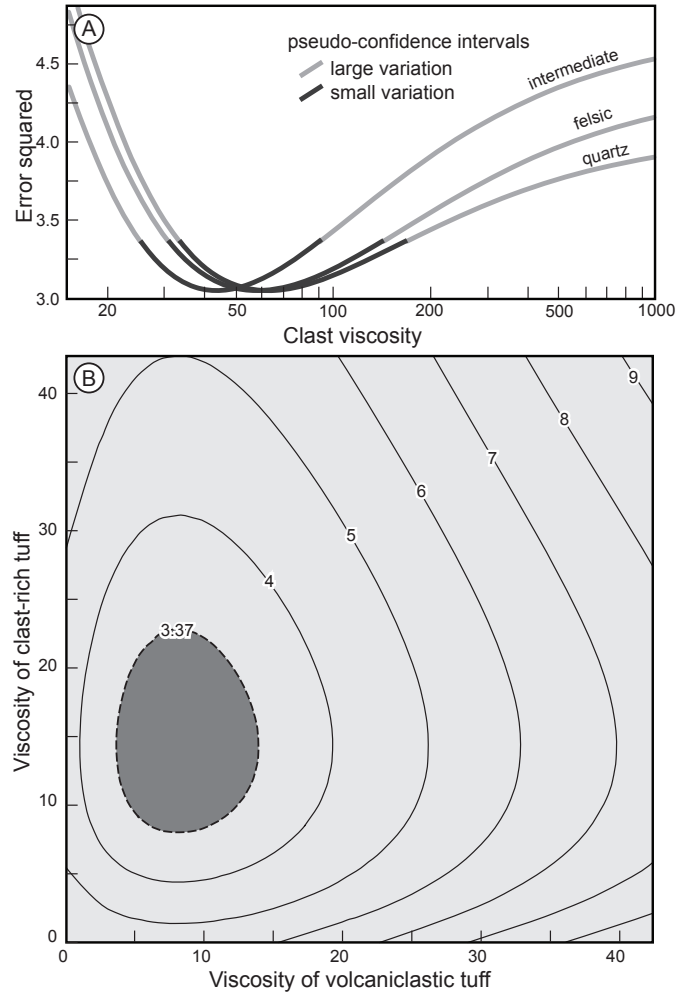


Figure 14: The effect of the rock viscosity parameters on the fitness of the model. The pseudo-confidence regions defined by the values 3.37 and 15.9 are shaded. (A) The error squared plotted against the viscosities of the three clast types. (B) The viscosities of the two matrix rocks, with contours showing the error squared.

tours of error squared as the angle of oblique convergence and the shortening vary. Based on the small-variation pseudo-confidence regions, this model predicts a shear zone strike of 153°-163° and dip of 75°-89°. The model predicts an angle of oblique convergence of 4°-10° and shortening of 0.18-0.34.

The best-fit viscosities in Table 3 imply clast-matrix viscosity ratios between 3 and 8. The small-variation pseudo-confidence intervals consistently predict viscosity ratios greater than 1. Fig. 14 explores the fitness of the model as we vary the viscosities. In panel (A), each clast viscosity is varied, while the other four viscosities (and other model parameters) are held fixed to their best-fit values. The pseudo-confidence regions span nearly one order of magnitude. In panel (B), each matrix rock viscosity is varied. Although the pseudo-confidence region for the matrix viscosities is narrower than for the clast viscosities, the analysis as a whole yields rather broad constraints on the viscosity ratios.

Clast	Matrix	Observed	Predicted
Felsic	Volc.	0.60, 1.05, 1.59	0.55, 0.91, 2.00
Felsic	Tuff	0.49, 1.36, 1.48	0.50, 0.92, 2.20
Interm.	Volc.	0.49, 0.96, 2.13	0.46, 1.18, 1.84
Interm.	Tuff	0.36, 1.25, 2.20	0.40, 1.28, 1.98
Quartz	Volc.	0.58, 1.12, 1.54	0.45, 1.08, 2.05
Quartz	Tuff	0.34, 1.05, 2.79	0.40, 1.12, 2.22

Table 4: Semi-axis lengths for the deformable clasts: data observed in the shear zone, and values predicted by the best-fit model.

Table 4 shows how the best-fit model matches the clast magnitudes. In some cases, such as the intermediate-composition clasts, the predicted semi-axis lengths are close to the observed ones. In many cases, however, the predicted lengths seem to be significant over- or underestimates of the observed values. Fig. 15 shows how the model matches the orientational aspects of the data. In panel (A), the lineation and foliation are matched to within a few degrees. In panel (B), all of the predicted short axes are quite close to their observations. In three of the cases, the other two axes match their observations to within a few degrees. In two other cases, the predictions are less close to, but still consistent with, the observations. The worst match seems to occur in the felsic-in-tuff case, where the long and intermediate axes are switched. However, the observed orientations of these axes may not be well-defined, because their lengths are so similar (Table 4).

6. Discussion

6.1. Computational methods

In Section 3, we discuss three methods for simulating the deformation of ellipsoidal clasts. We focus on the (\mathbf{Q}, a_i) method because this kind of approach — evolving the orientation and semi-axes of the ellipsoid separately — dominates the geology literature on this problem (e.g., Freeman, 1987; Jiang, 2007a, 2012). We focus on the \mathbf{E} method because it is elegant. The ellipsoid tensor captures orientation and magnitude in a single quantity, governed by a fairly simple differential equation. We focus on the Lie group \mathbf{F}^{-1} method because it illustrates the role of Lie theory in describing rock deformation, and because it is the fastest method that we have tried.

This list of three methods is not meant to be exhaustive. For example, we mention in Section 3.3 that one could evolve \mathbf{F} or \mathbf{F}^{-1} using Euclidean methods, instead of Lie group methods. For another example, it is easy to adapt the (\mathbf{Q}, a_i) method to simulate the logarithms of the semi-axes. If we let $b_i = \log a_i$, then Eq. (A.16) becomes $\dot{b}_i = \tilde{C}_{ii}$, and a (\mathbf{Q}, b_i) method follows immediately. Also, in this paper we have discussed only Euler and fourth-order Runge-Kutta methods, for the sake of clarity. More sophisticated methods exist in Euclidean spaces, including higher-order and multi-step methods. Munthe-Kaas (1999) and Faltinsen et al. (2001) develop some of these methods on Lie groups.

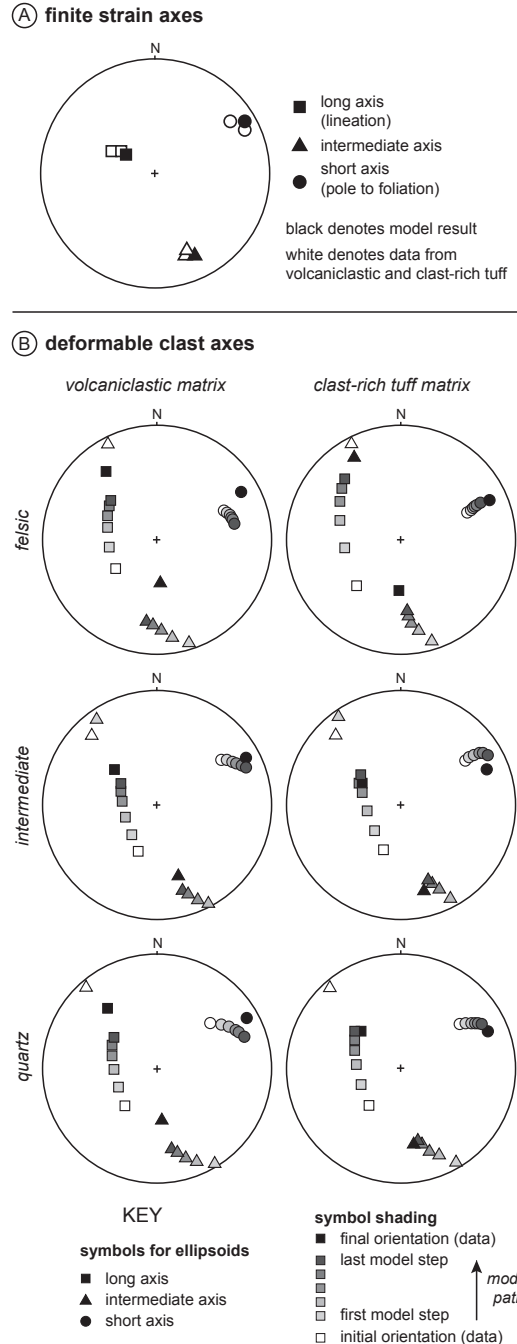


Figure 15: Equal-area, lower-hemisphere plots showing how the best-fit model matches the orientational data. (A) The subproblem of matching lineation and foliation. (B) The six deformable clast subproblems.

The main practical reason for contemplating various methods is that some are faster than others. For example, to achieve an error of 0.001 in the ellipsoid tensor, the Lie group \mathbf{F}^{-1} method requires only $n = 4$ steps, while the Euclidean (\mathbf{Q}, a_i) method requires $n = 36$ steps (Fig. 6). In our implementations, the processor time required by the \mathbf{F}^{-1} method is proportionally smaller, but implementations in a different language or library could produce

different processor times.

Speed is of practical value to geologists because the analysis of naturally deformed rocks may require many ellipsoid simulations. Computing the best-fit inclined transpression for the Gem Lake shear zone requires around 180,000 simulations, or about one day of computation on our computer. The analysis might require a week of computation if it were based on the previously available methods. Fast methods allow us to explore a wide variety of models.

The \mathbf{E} and \mathbf{F}^{-1} methods also offer secondary advantages over the (\mathbf{Q}, a_i) method. Neither method requires the computation of \mathbf{Q} , which is difficult in the spheroid case (Appendix C). The Lie group \mathbf{F}^{-1} method always preserves the volume of the ellipsoid (Fig. 8) and avoids catastrophic failures of geometry (Fig. 3).

6.2. Application of the methods

In the application section of this paper, we use our Lie group ellipsoid method to inverse-model the bulk deformation and viscosity ratios in the Gem Lake shear zone. Horsman et al. (2008) used different methods to estimate similar parameters, including dextral offset. In this section, we compare our results to theirs and others from the literature.

For characterizing the bulk kinematics, Horsman et al. (2008) fixed the strike (147°) and dip (78° SW) of the shear zone based on field fabrics, so these were not variables in their modeling. This shear plane is not dissimilar from our optimal value of $158^\circ/82^\circ$ SW, and it lies within our large-variation pseudo-confidence interval (Fig. 13). Horsman et al. (2008) found an angle of oblique convergence of 5° – 25° using a model of inclined transpression designed to match strain values within the shear zone. That estimate is consistent with our best-fit value of 7° and range of 4° – 10° . Also, their shortening values of 0.2–0.6 are consistent with our best-fit value of 0.27 and range of 0.18–0.34. In short, our estimates of deformation coincide with the lower end of the estimates of Horsman et al. (2008).

Horsman et al. (2008) used cleavage orientations in the two host rocks, coupled with a model of cleavage refraction in layered media (Treagus, 1983), to estimate relative rock viscosities. Those authors concluded that the volcanoclastic rock is more viscous than the clast-rich tuff, with a viscosity ratio between 2 and 5. Our analysis predicts the opposite relationship: a best-fit ratio of 0.57, with a pseudo-confidence interval of 0.17 to 1.8 (Table 3). The discrepancy between these results is not especially worrisome. It is clear from Fig. 14 that, although our analysis estimates the viscosities of all five rock types, these estimates are not tightly constrained. For example, the quartz/volcanoclastic ratio may lie anywhere between 2 and 42.

Using the parameters from our best-fit inclined transpression, we compute a total dextral offset of 3.1 km across the Gem Lake shear zone. As the dip, angle

of oblique convergence, and shortening range over their pseudo-confidence regions, the dextral offset varies between 1.2 km and 7.6 km. (This range is more conservative than that predicted by the pseudo-confidence region in Fig. 13, because the most extreme combinations of values lie outside that region.) The tectonic-scale kinematic analysis of Horsman et al. (2008) also resulted in an estimated offset of only a few kilometers across the shear zone. These estimates broadly agree with the 1.2–8 km estimate for the Rosy Finch shear zone (Tikoff and Teyssier, 1992; Greene and Schweickert, 1995), which is the southwest continuation of the Gem Lake shear zone.

The Gem Lake shear zone offset estimates are considerably smaller than the 20 km or more hypothesized by Greene and Schweickert (1995). That estimate was based on matching marble slivers between two volcanic pendants. Those authors note that the displacement could be considerably smaller, if the two marbles were derived from different sources. If the offset were truly as large as 20 km, then clast interaction might provide an explanation for our smaller estimate. In a deforming rock with multiple clasts, interactions among clasts can appreciably affect deformation, unless the clasts are separated by about four radii or more (Mandal et al., 2003). The clasts in the Gem Lake shear zone rocks are not tightly packed, but nor are they consistently separated by four radii (Horsman et al., 2008, Fig. 4). Thus, our model, in treating the clasts as isolated, is overly simplified and not entirely realistic. Deformation partitioning at multiple scales provides another possible explanation for the discrepancy. For instance, our model does not consider the possibility that deformation is partitioned differently among the two matrix rocks in the shear zone. Further, it is possible that deformation was accommodated outside of the narrow 1 km-wide region studied near Gem Lake. Along most of its mapped length, the shear zone includes multiple anastomosing strands (Greene and Schweickert, 1995, Fig. 2). In the Gem Lake region, only one strand has been recognized. Perhaps more detailed mapping would demonstrate the existence of another strand.

6.3. Implications

Lie groups enjoy a rich theory going back well over a century (Hawkins, 2000). We believe that that this theory can benefit geologists both conceptually, by supplying new vocabulary for describing deformation, and practically, by enabling new approaches to computation of deformation. A Lie group may seem abstract or unnecessary at first. However, it is actually more physical than the more familiar space of all 3×3 matrices, because it captures exactly the volume-preserving deformations (or rigid rotations, in Section 4) that are meaningful in the problem. This paper improves upon our previous Lie theory application (Davis and Titus, 2011) in several ways.

First, whereas the previous paper treated all *steady* homogeneous deformations, this paper applies to all homogeneous deformations, which are described by Eqs. (4) and

(6). The deforming ellipsoid of Eshelby (1957) is actually just one example. For another example, consider any homogeneous transpressional system. If the strike of the shear zone, direction of motion, or speed of motion changes during deformation, then the velocity gradient tensor \mathbf{L} is non-steady. There is no closed-form solution for the finite deformation \mathbf{F} , but our methods for \mathbf{F} and \mathbf{F}^{-1} (Section 3.3 and Appendix D) yield a numerical approximation with high speed and precision. For a final example, consider a deforming host rock that contains a roughly ellipsoidal xenolith, which itself contains a smaller ellipsoidal clast. This situation can be modeled as a nested Eshelby problem. Appendix A lets us compute the velocity gradient tensor in the xenolith from that in the host rock, and the velocity gradient tensor in the clast from that in the xenolith. The \mathbf{E} and \mathbf{F}^{-1} methods can then simulate the deformation of all three rocks simultaneously. Similar problems have been studied by Mancktelow (2013) and Jiang and Bentley (2012).

Second, our previous paper showed how to integrate a variety of data types into the computation of a best-fit steady model (Davis and Titus, 2011, Fig. 7). This paper adds a new data type — deformed ellipsoidal clasts — and gives an example where the orientation, but not the magnitude, of the finite strain ellipsoid is known.

Third, our new techniques aid the computation of best-fit non-steady models. For example, suppose that we have data about the finite strain ellipsoid $(\mathbf{F}\mathbf{F}^\top)^{-1}$ of a homogeneous finite deformation \mathbf{F} . If we assume a constant velocity gradient tensor \mathbf{L} , then the data constrain \mathbf{L} via the equation $(\mathbf{F}\mathbf{F}^\top)^{-1} = (\exp \mathbf{L} \exp \mathbf{L}^\top)^{-1}$. If \mathbf{L} is not constant, then that equation does not hold, but our numerical methods for \mathbf{F}^{-1} or \mathbf{F} still let us compute $(\mathbf{F}\mathbf{F}^\top)^{-1}$ from \mathbf{L} quickly and precisely. This computation is the fundamental step in fitting a non-steady \mathbf{L} to finite strain ellipsoid data.

Finally, whereas Davis and Titus (2011) presented its best-fit models as unique and certain, this paper attempts to describe the uncertainty in the best-fit parameters for the Gem Lake shear zone, through the numerical experiments of Section 5.3. Although the pseudo-confidence regions that we construct are not statistically rigorous, they provide useful constraints on the bulk deformation (Fig. 13). The rigor of the analysis could be improved through Markov chain Monte Carlo methods (e.g., Gregory, 2005), for example. Like the numerical optimization algorithm employed in this paper, a Markov chain Monte Carlo approach would require many deforming ellipsoid simulations. Fast methods for computing these simulations are always valuable.

7. Acknowledgments

Basil Tikoff gave constructive criticism on an early draft of this paper. Comments by Kieran Mulchrone, an anonymous referee, and Bill Dunne improved the paper after submission. We also thank Mark Brandon, Neil

Ramsamooj, Chelsea Scott, Peter Spaeth, and Bill Titus for helpful discussions. This research was funded by NSF-EAR 1151851 (Davis and Titus). Support for writing was provided by a GAIN workshop funded by NSF-HRD grants 0620101 (Mary Anne Holmes) and 0620087 (Suzanne O’Connell). Computing resources were supported in part by a grant from the Howard Hughes Medical Institute to Carleton College.

8. References

- Belinfante, J.G., Kolman, B., 1972. A Survey of Lie Groups and Lie Algebras with Applications and Computational Methods. Society for Industrial and Applied Mathematics, Philadelphia.
- Bilby, B., Eshelby, J., Kundu, A., 1975. The change of shape of a viscous ellipsoidal region embedded in a slowly deforming matrix having a different viscosity. *Tectonophysics* 28, 265–274.
- Bilby, B., Kolbuszewski, M., 1957. The finite deformation of an inhomogeneity in two-dimensional slow viscous incompressible flow. *Proceedings of the Royal Society of London A* 355, 335–353.
- Bown, R.M., 1989. *Introduction to Continuum Mechanics for Engineers*. Plenum Press, New York.
- Brandon, M.T., 1995. Analysis of geologic strain data in strain-magnitude space. *Journal of Structural Geology* 17, 1375–1385.
- Burden, R.L., Faires, J.D., 2001. *Numerical Analysis*. Brooks/Cole. 7th edition.
- Chapra, S.C., Canale, R.P., 2002. *Numerical Methods for Engineers*. McGraw-Hill. 4th edition.
- Cloos, E., 1947. Oolite deformation in the South Mountain fold, Maryland. *Geological Society of America Bulletin* 58, 843–917.
- Cloos, E., 1971. *Microtectonics along the Western Edge of the Blue Ridge, Maryland and Virginia*. Johns Hopkins University, Studies in Geology, Baltimore.
- Curtis, M.L., 1984. *Matrix Groups*. Springer-Verlag, New York. 2nd edition.
- Dabrowski, M., Schmid, D.W., 2011. A rigid circular inclusion in an anisotropic host subject to simple shear. *Journal of Structural Geology* 33, 1169–1177.
- Davis, J.R., Titus, S.J., 2011. Homogeneous steady deformation: A review of computational techniques. *Journal of Structural Geology* 33, 1046–1062.
- De Paor, D.G., 1988. R-f/phi strain analysis using and orientation net. *Journal of Structural Geology* 10, 323–333.
- Dunnet, D., 1969. A technique of finite strain analysis using elliptical particles. *Tectonophysics* 7, 117–136.
- Elliott, D., 1970. Determination of finite strain and initial shape from deformed elliptical objects. *Geological Society of America Bulletin* 81, 2221–2236.
- Eshelby, J., 1957. The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proceedings of the Royal Society of London A* 241, 376–396.
- Eshelby, J., 1959. The elastic field outside an ellipsoidal inclusion. *Proceedings of the Royal Society of London A* 252, 561–569.
- Faltinsen, S., Marthinsen, A., Munthe-Kaas, H.Z., 2001. Multistep methods integrating ordinary differential equations on manifolds. *Applied Numerical Mathematics* 39, 349–365.
- Fletcher, R.C., 2004. Anisotropic viscosity of a dispersion of aligned elliptical cylindrical clasts in viscous matrix. *Journal of Structural Geology* 26, 1977–1987.
- Fletcher, R.C., 2009. Deformable, rigid, and inviscid elliptical inclusions in a homogeneous incompressible anisotropic viscous fluid. *Journal of Structural Geology* 31, 382–387.
- Flinn, D., 1979. The deformation matrix and the deformation ellipsoid. *Journal of Structural Geology* 1, 299–307.
- Freeman, B., 1987. The behavior of deformable ellipsoidal particles in three-dimensional slow flows: implications for geological strain analysis. *Tectonophysics* 132, 297–309.

- Gay, N.C., 1968a. The motion of rigid particles embedded in a viscous fluid during pure shear deformation of the fluid. *Tectonophysics* 5, 81–88.
- Gay, N.C., 1968b. Pure shear and simple shear deformation of inhomogeneous viscous fluids. 1. theory. *Tectonophysics* 5, 211–234.
- Gerald, C.F., Wheatley, P.O., 1984. *Applied Numerical Analysis*. Addison-Wesley, 3rd edition.
- Ghosh, S.K., Ramberg, H., 1976. Reorientation of inclusions by combination of pure and simple shear. *Tectonophysics* 34, 1–70.
- Gilmore, R., 1974. *Lie Groups, Lie Algebras, and Some of Their Applications*. John Wiley & Sons, New York.
- Greene, D.C., Schweickert, R.A., 1995. The gem lake shear zone: Cretaceous dextral transpression in the northern Ritter Range pendant, eastern Sierra Nevada, California. *Tectonics* 14, 945–961.
- Gregory, P., 2005. *Bayesian Logical Data Analysis for the Physical Sciences*. Cambridge University Press, Cambridge, UK.
- Hall, B.C., 2003. *Lie groups, Lie algebras, and representations*. Springer-Verlag, New York.
- Hawkins, T., 2000. *Emergence of the Theory of Lie Groups*. Springer-Verlag, New York.
- Horsman, E., Tikoff, B., Czeck, D., 2008. Rheological implications of heterogeneous deformation at multiple scales in the late Cretaceous Sierra Nevada, California. *Geological Society of America Bulletin* 120, 238–255.
- Jeffery, G., 1922. The motion of ellipsoidal particles immersed in a viscous fluid. *Proceedings of the Royal Society of London A* 102, 161–179.
- Jessup, M.J., Law, R.D., Frassi, C., 2007. The Rigid Grain Net (RGN): an alternative method for estimating mean kinematic vorticity number (W_m). *Journal of Structural Geology* 29, 411–421.
- Jezek, J., Schulmann, K., Segeth, K., 1996. Fabric evolution of rigid inclusions during mixed coaxial and simple shear flows. *Tectonophysics* 257, 203–221.
- Jiang, D., 2007a. Numerical modeling of the motion of deformable ellipsoidal objects in slow viscous flows. *Journal of Structural Geology* 29, 435–452.
- Jiang, D., 2007b. Numerical modeling of the motion of rigid ellipsoidal objects in slow viscous flows: A new approach. *Journal of Structural Geology* 29, 189–200.
- Jiang, D., 2012. A general approach for modeling the motion of rigid and deformable ellipsoids in ductile flows. *Computers & Geosciences* 38, 52–61.
- Jiang, D., Bentley, C., 2012. A micromechanical approach for simulating multi-scale fabrics in large-scale high-strain zones: theory and application. *Journal of Geophysical Research (Solid Earth)* 117.
- Jones, R.R., Holdsworth, R.E., Clegg, P., McCaffrey, K., Tavarnelli, E., 2004. Inclined transpression. *Journal of Structural Geology* 26, 1531–1548.
- Lay, D.C., 1994. *Linear Algebra and Its Applications*. Addison Wesley, Reading, Massachusetts.
- Lin, S., Jiang, D., Williams, P.F., 1998. Transpression (or transtension) zones of triclinic symmetry; natural example and theoretical modelling, in: Holdsworth, R.E., Strachan, R.A., Dewey, J.F. (Eds.), *Continental Transpressional and Transtensional Tectonics*. Geological Society, London. volume 135, pp. 41–57.
- Lisle, R.J., 1977. Estimation of the tectonic strain ratio from the mean shape of deformed elliptical markers. *Geologie en Mijnbouw* 56, 140–144.
- Mancktelow, N.S., 2011. Deformation of an elliptical inclusion in two-dimensional incompressible power-law viscous flow. *Journal of Structural Geology* 33, doi:10.1016/j.jsg.2011.06.005, 1378–1393.
- Mancktelow, N.S., 2013. Behavior of an isolated rimmed elliptical inclusion in 2D slow incompressible flow. *Journal of Structural Geology* 46, 235–254.
- Mandal, N., Samanta, S.K., Bhattacharyya, G., Chakraborty, C., 2003. Deformation of ductile inclusions in a multiple inclusion system in pure shear. *Journal of Structural Geology* 25, 1359–1370.
- Manga, M., Castro, J., Cashman, K., Loewenberg, M., 1998. Rheology of bubble-bearing magmas. *Journal of Volcanology and Geothermal Research* 87, 15–28.
- Matthews, P.E., Bond, R.A.B., Van den Berg, J.J., 1974. An algebraic method of strain analysis using elliptical markers. *Tectonophysics* 24, 31–67.
- Mulchrone, K., 2007. Avoiding singularities in the numerical solution of the motion of a deformable ellipse immersed in a viscous fluid. *Mathematical Geology* 39, 647–655.
- Mulchrone, K.F., Walsh, K., 2006. The motion of a non-rigid ellipse in a general 2D deformation. *Journal of Structural Geology* 28, 392–407.
- Munthe-Kaas, H., 1998. Runge-Kutta methods on Lie groups. *BIT* 38, 92–111.
- Munthe-Kaas, H., 1999. High order Runge-Kutta methods on manifolds. *Applied Numerical Mathematics* 29, 115–127.
- Passchier, C.W., 1987. Stable positions of rigid objects in non-coaxial flow; a study in vorticity analysis. *Journal of Structural Geology* 9, 679–690.
- Pollard, D.D., Fletcher, R.C., 2005. *Fundamentals of Structural Geology*. Cambridge Univ. Press, Cambridge.
- Pollatsek, H., 2009. *Lie Groups: A Problem-Oriented Introduction via Matrix Groups*. Mathematical Association of America.
- Provost, A., Buisson, C., Merle, O., 2004. From progressive to finite deformation and back. *Journal of Geophysical Research* 109, doi:10.1029/2001JB001734.
- Ramsay, J.G., 1967. *Folding and fracturing of rocks*. McGraw-Hill, New York.
- Rust, A.C., Manga, M., Cashman, K.V., 2003. Determining flow type, shear rate and shear stress in magmas from bubble shapes and orientations. *Journal of Volcanology and Geothermal Research* 122, 111–132.
- Schmid, D.W., Podladchikov, Y.Y., 2003. Analytical solutions for deformable elliptical inclusions in general shear. *Geophysical Journal International* 155, 1269–288.
- Simpson, C., De Paor, D.G., 1997. Practical analysis of general shear zones using porphyroblast hyperbolic distribution method: an example from the Scandinavian Caledonides, in: Sengupta, S. (Ed.), *Evolution of Geological Structures in Micro- to Macroscales*. Chapman and Hall, London, pp. 169–184.
- Tikoff, B., Greene, D., 1997. Stretching lineations in transpressional shear zones: An example from the Sierra Nevada batholith, California. *Journal of Structural Geology* 19, 2939.
- Tikoff, B., Teyssier, C., 1992. Crustal-scale, en echelon “P-shear” tensional bridges: A possible solution to the batholithic room problems. *Geology* 20, 927–930.
- Treagus, S.H., 1983. A theory of finite strain variation through contrasting layers, and its bearing on cleavage refraction. *Journal of Structural Geology* 5, 351–368.
- Treagus, S.H., 2002. Modelling the bulk viscosity of two-phase mixtures in terms of clast shape. *Journal of Structural Geology* 24, 57–76.
- Treagus, S.H., Treagus, J.E., 2001. Effects of object ellipticity on strain, and implications for clast–matrix rocks. *Journal of Structural Geology* 23, 601–608.
- Tullis, T.E., Wood, D.S., 1975. Correlation of finite strain from both reduction bodies and preferred orientations of mica in slate from Wales. *Geological Society of America Bulletin* 86, 632–638.
- Wheeler, J., 1986. Average properties of ellipsoidal fabrics: implications for two- and three-dimensional methods of strain analysis. *Tectonophysics* 126, 259–270.

Appendix A. Dynamics

The dynamics of the deformation of the ellipsoidal inclusion were worked out by Eshelby (1957, 1959) and Bilby et al. (1975) and further developed by Bilby and Kolbuszewski (1957); Freeman (1987); Jiang (2007a), and others. Here we recapitulate the key results without rederiving them.

We employ two coordinate systems centered on the center of the ellipsoid. The \mathbf{x} coordinate axes are fixed (for example, geographic), while the $\tilde{\mathbf{x}}$ coordinate axes rotate with the axes of the ellipsoid (Fig. 2). The two coordinate systems are related by

$$\tilde{\mathbf{x}} = \mathbf{Q}\mathbf{x}, \quad (\text{A.1})$$

where \mathbf{Q} is a rotation matrix. The rows of \mathbf{Q} are unit vectors (in \mathbf{x} coordinates) indicating the directions of the ellipsoid semi-axes in a right-handed manner. Differentiating the orthogonality condition $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$, we obtain

$$\dot{\mathbf{Q}}\mathbf{Q}^\top + \mathbf{Q}\dot{\mathbf{Q}}^\top = \mathbf{0}. \quad (\text{A.2})$$

The ellipsoid tensor \mathbf{E} is the symmetric, positive-definite (0,2)-tensor such that the ellipsoid is the set of points \mathbf{x} satisfying $\mathbf{x}^\top \mathbf{E} \mathbf{x} = 1$. This tensor \mathbf{E} diagonalizes as $\mathbf{E} = \mathbf{Q}^\top \tilde{\mathbf{E}} \mathbf{Q}$, where

$$\tilde{\mathbf{E}} = \begin{bmatrix} a_1^{-2} & 0 & 0 \\ 0 & a_2^{-2} & 0 \\ 0 & 0 & a_3^{-2} \end{bmatrix},$$

and the $a_i > 0$ are the semi-axis lengths of the ellipsoid (Fig. 3). This $\tilde{\mathbf{E}}$ is \mathbf{E} expressed in $\tilde{\mathbf{x}}$ coordinates. When expressing tensors in $\tilde{\mathbf{x}}$ coordinates, we consistently use the tilde $\tilde{\cdot}$.

The deformation of the host rock is defined by $\dot{\mathbf{x}} = \mathbf{L}\mathbf{x}$ (Eq. (1)), where \mathbf{L} is a constant tensor of trace zero. After substituting Eq. (A.1) into Eq. (1) we have $\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{L}}\tilde{\mathbf{x}}$, where

$$\tilde{\mathbf{L}} = \mathbf{Q}\mathbf{L}\mathbf{Q}^\top - \mathbf{Q}\dot{\mathbf{Q}}^\top. \quad (\text{A.3})$$

Let r be the ratio of the viscosity of the inclusion to the viscosity of the host rock. We assume that r is constant. The deformation of the ellipsoidal inclusion is homogeneous (Eshelby, 1957; Bilby et al., 1975): For all \mathbf{x} inside the inclusion and along its boundary, $\dot{\mathbf{x}} = \mathbf{K}\mathbf{x}$, for some time-dependent velocity gradient tensor \mathbf{K} . Describing \mathbf{K} and related quantities is the goal of this appendix. In preparation, recall that any matrix \mathbf{M} can be decomposed as a sum

$$\mathbf{M} = \frac{1}{2}(\mathbf{M} + \mathbf{M}^\top) + \frac{1}{2}(\mathbf{M} - \mathbf{M}^\top)$$

of symmetric and antisymmetric matrices. Let $\mathbf{L} = \mathbf{D} + \mathbf{W}$, $\tilde{\mathbf{L}} = \tilde{\mathbf{D}} + \tilde{\mathbf{W}}$, and

$$\tilde{\mathbf{K}} = \tilde{\mathbf{C}} + \tilde{\mathbf{V}} \quad (\text{A.4})$$

be these decompositions of \mathbf{L} , $\tilde{\mathbf{L}}$, and $\tilde{\mathbf{K}}$. Then from Eqs. (A.3) and (A.2) it follows that

$$\tilde{\mathbf{D}} = \mathbf{Q}\mathbf{D}\mathbf{Q}^\top, \quad (\text{A.5})$$

$$\tilde{\mathbf{W}} = \mathbf{Q}\mathbf{W}\mathbf{Q}^\top - \mathbf{Q}\dot{\mathbf{Q}}^\top. \quad (\text{A.6})$$

The tensor $\tilde{\mathbf{C}}$ is given by

$$\tilde{C}_{ij} = \tilde{D}_{ij} + (1-r) \sum_{k=1}^3 \sum_{l=1}^3 \tilde{S}_{ijkl} \tilde{C}_{kl}, \quad (\text{A.7})$$

where

$$\begin{aligned} \tilde{S}_{iijj} &= \frac{3}{4\pi} a_j^2 J_{ij}, \\ \tilde{S}_{ijij} &= \tilde{S}_{ijji} = \frac{3}{8\pi} (a_i^2 + a_j^2) J_{ij}, \end{aligned}$$

all other \tilde{S}_{ijkl} are zero,

$$\begin{aligned} J_{ii} &= 2\pi a_1 a_2 a_3 \int_0^\infty \frac{du}{(a_i^2 + u)^2 \lambda}, \\ J_{ij} &= \frac{2}{3} \pi a_1 a_2 a_3 \int_0^\infty \frac{du}{(a_i^2 + u)(a_j^2 + u)\lambda} \end{aligned}$$

for $i \neq j$, and

$$\lambda = \sqrt{(a_1^2 + u)(a_2^2 + u)(a_3^2 + u)}.$$

For $i \neq j$, Eq. (A.7) simplifies to

$$\tilde{C}_{ij} = \frac{\tilde{D}_{ij}}{1 + \frac{3(r-1)}{4\pi} (a_i^2 + a_j^2) J_{ij}}. \quad (\text{A.8})$$

For $i = j$, Eq. (A.7) simplifies to

$$\tilde{C}_{ii} = \tilde{D}_{ii} - \frac{3(r-1)}{4\pi} \sum_{k=1}^3 a_k^2 J_{ik} \tilde{C}_{kk}. \quad (\text{A.9})$$

Let \mathbf{J} be a matrix whose (i, j) th entry is J_{ij} as defined above. Then Eq. (A.9) can be rewritten as

$$\mathbf{A} \begin{bmatrix} \tilde{C}_{11} \\ \tilde{C}_{22} \\ \tilde{C}_{33} \end{bmatrix} = \begin{bmatrix} \tilde{D}_{11} \\ \tilde{D}_{22} \\ \tilde{D}_{33} \end{bmatrix}, \quad (\text{A.10})$$

where $\mathbf{A} = \mathbf{I} + \frac{3(r-1)}{4\pi} \mathbf{J} \tilde{\mathbf{E}}^{-1}$.

This dynamical model assumes that the deformation is incompressible (Bilby et al., 1975). Incompressibility corresponds to the velocity gradient tensor's having trace zero. From Eq. (A.9) and the identity

$$J_{i1} + J_{i2} + J_{i3} = \frac{4\pi}{3a_i^2} \quad (\text{A.11})$$

(Eshelby, 1957, (3.11)), it is easily proved that $r \operatorname{tr} \tilde{\mathbf{C}} = \operatorname{tr} \tilde{\mathbf{D}}$. Meanwhile, $\operatorname{tr} \tilde{\mathbf{C}} = \operatorname{tr} \tilde{\mathbf{K}} = \operatorname{tr} \mathbf{K}$ and $\operatorname{tr} \tilde{\mathbf{D}} = \operatorname{tr} \mathbf{D} = \operatorname{tr} \mathbf{L}$. Thus $r \operatorname{tr} \mathbf{K} = \operatorname{tr} \mathbf{L}$, so the inclusion's deformation is incompressible if the ambient deformation is. The expressions for \tilde{C}_{ii} in Freeman (1987) and Jiang (2007a) are equivalent to the volume-preserving case of Eq. (A.10).

The tensor $\tilde{\mathbf{V}}$ is given by

$$\tilde{V}_{ij} = \tilde{W}_{ij} + (1-r) \sum_{k=1}^3 \sum_{l=1}^3 \tilde{\Pi}_{ijkl} \tilde{C}_{kl},$$

where

$$\begin{aligned} \tilde{\Pi}_{ijij} &= \tilde{\Pi}_{ijji} = \frac{1}{8\pi} (J_j - J_i), \\ J_i &= 2\pi a_1 a_2 a_3 \int_0^\infty \frac{du}{(a_i^2 + u)\lambda}, \end{aligned}$$

and all other $\tilde{\Pi}_{ijkl} = 0$. This equation simplifies to

$$\tilde{V}_{ij} = \tilde{W}_{ij} + \frac{r-1}{4\pi}(J_i - J_j)\tilde{C}_{ij}. \quad (\text{A.12})$$

The approaches of Sections 3.2 and 3.3 require us to compute \mathbf{K} from \mathbf{E} , \mathbf{L} , and r at each time step. In analogy with Eq. (A.3), $\tilde{\mathbf{K}} = \mathbf{Q}\mathbf{K}\mathbf{Q}^\top - \mathbf{Q}\dot{\mathbf{Q}}^\top$. From this equation and Eq. (A.6) it follows that

$$\mathbf{K} = \mathbf{Q}^\top (\tilde{\mathbf{K}} - \tilde{\mathbf{W}}) \mathbf{Q} + \mathbf{W}, \quad (\text{A.13})$$

where

$$(\tilde{\mathbf{K}} - \tilde{\mathbf{W}})_{ij} = \left(1 + \frac{r-1}{4\pi}(J_i - J_j)\right)\tilde{C}_{ij} \quad (\text{A.14})$$

by Eqs. (A.4) and (A.12). The approach of Section 3.2 further requires us to compute $\dot{\mathbf{E}}$ at each time step. Differentiating Eq. (2) and using Eq. (4), we have

$$\begin{aligned} 0 &= \dot{\mathbf{x}}^\top \mathbf{E} \mathbf{x} + \mathbf{x}^\top \dot{\mathbf{E}} \mathbf{x} + \mathbf{x}^\top \mathbf{E} \dot{\mathbf{x}} \\ &= \mathbf{x}^\top (\mathbf{K}^\top \mathbf{E} + \dot{\mathbf{E}} + \mathbf{E} \mathbf{K}) \mathbf{x}. \end{aligned}$$

Because that equation holds for all \mathbf{x} on the boundary of the ellipsoid, the tensor in the middle must be zero, and so

$$\dot{\mathbf{E}} = -\mathbf{K}^\top \mathbf{E} - \mathbf{E} \mathbf{K}. \quad (\text{A.15})$$

The approach of Section 3.1 requires us to compute \dot{a}_i and $\dot{\mathbf{Q}}$ at each time step. In $\tilde{\mathbf{x}}$ coordinates, Eq. (A.15) is $\dot{\tilde{\mathbf{E}}} = -\tilde{\mathbf{K}}^\top \tilde{\mathbf{E}} - \tilde{\mathbf{E}} \tilde{\mathbf{K}}$. Using Eqs. (A.4) and (3), we can rewrite this matrix equation as nine scalar equations

$$\dot{\tilde{E}}_{ij} = a_j^{-2} \tilde{V}_{ij} - a_j^{-2} \tilde{C}_{ij} - a_i^{-2} \tilde{C}_{ij} - a_i^{-2} \tilde{V}_{ij}.$$

In the $i = j$ case, $\frac{d}{dt}(a_i^{-2}) = -2a_i^{-2} \dot{a}_i$, which implies that

$$\dot{a}_i = a_i \dot{\tilde{C}}_{ii}. \quad (\text{A.16})$$

This differential equation describes the evolution of the ellipsoid axis magnitudes over time. A differential equation governing the orientation \mathbf{Q} is found by rearranging Eq. (A.6):

$$\dot{\mathbf{Q}} = \tilde{\mathbf{W}}\mathbf{Q} - \mathbf{Q}\mathbf{W}. \quad (\text{A.17})$$

It remains to compute $\tilde{\mathbf{W}}$. Because $\tilde{\mathbf{W}}$ is antisymmetric, $\tilde{W}_{ii} = 0$ for all i . To compute \tilde{W}_{ij} for $i \neq j$, notice that $\tilde{\mathbf{E}}$ is always diagonal. Therefore $\dot{\tilde{E}}_{ij} = 0$ and

$$(a_i^2 + a_j^2)\tilde{C}_{ij} = (a_i^2 - a_j^2)\tilde{V}_{ij}. \quad (\text{A.18})$$

In the typical situation, when $a_i \neq a_j$, we have

$$\tilde{V}_{ij} = \frac{a_i^2 + a_j^2}{a_i^2 - a_j^2} \tilde{C}_{ij}. \quad (\text{A.19})$$

Combining this equation with Eqs. (A.8) and (A.12), we have

$$\tilde{W}_{ij} = \left(\frac{a_i^2 + a_j^2}{a_i^2 - a_j^2} + \frac{1-r}{4\pi}(J_i - J_j) \right) \tilde{C}_{ij} \quad (\text{A.20})$$

$$= \frac{4\pi \frac{a_i^2 + a_j^2}{a_i^2 - a_j^2} + (r-1)(J_i - J_j)}{4\pi + 3(r-1)(a_i^2 + a_j^2)J_{ij}} \tilde{D}_{ij}. \quad (\text{A.21})$$

We discuss the special case of $a_i = a_j$ in Appendix C.

In Eqs. (A.8), (A.9), and (A.12), setting $r = 1$ produces $\tilde{\mathbf{C}} = \tilde{\mathbf{D}}$ and $\tilde{\mathbf{V}} = \tilde{\mathbf{W}}$. This result makes sense. There is no viscosity contrast, so the inclusion deforms passively, as part of the surrounding rock (Freeman, 1987; Jiang, 2007a).

Consider the rigid limit $r \rightarrow \infty$. We can rewrite Eq. (A.10) as

$$\begin{bmatrix} \tilde{C}_{11} \\ \tilde{C}_{22} \\ \tilde{C}_{33} \end{bmatrix} = \frac{\text{adj } \mathbf{A}}{\det \mathbf{A}} \begin{bmatrix} \tilde{D}_{11} \\ \tilde{D}_{22} \\ \tilde{D}_{33} \end{bmatrix},$$

where $\text{adj } \mathbf{A}$ is the adjugate of \mathbf{A} (e.g., Lay, 1994, p. 179). It is easily seen that $\det \mathbf{A}$ is cubic in r , while the entries of $\text{adj } \mathbf{A}$ are only quadratic in r . Hence $\text{adj } \mathbf{A} / \det \mathbf{A} \rightarrow \mathbf{0}$ and the three $\tilde{C}_{ii} \rightarrow 0$ as $r \rightarrow \infty$. In Eq. (A.8) we have $\tilde{C}_{ij} \rightarrow 0$ for $i \neq j$. Thus $\tilde{\mathbf{C}} \rightarrow \mathbf{0}$, which makes sense: The rigid inclusion does not deform. Now if all of the ellipsoid semi-axis lengths are distinct, then Eq. (A.19) implies that $\tilde{\mathbf{V}} \rightarrow \mathbf{0}$ as well. However, the inclusion may still rotate. In analogy with Eqs. (A.5) and (A.6), we have

$$\tilde{\mathbf{C}} = \mathbf{Q}\mathbf{C}\mathbf{Q}^\top, \quad \tilde{\mathbf{V}} = \mathbf{Q}\mathbf{V}\mathbf{Q}^\top - \mathbf{Q}\dot{\mathbf{Q}}^\top. \quad (\text{A.22})$$

From these equations, Eqs. (A.6) and (A.4), and $\tilde{\mathbf{C}} = \tilde{\mathbf{V}} = \mathbf{0}$ it follows that

$$\mathbf{K} = \mathbf{V} = \mathbf{W} - \mathbf{Q}^\top \tilde{\mathbf{W}}\mathbf{Q} \quad (\text{A.23})$$

in the limit $r \rightarrow \infty$. By Eq. (A.21),

$$\tilde{W}_{ij} \rightarrow \frac{J_j - J_i}{3(a_i^2 + a_j^2)J_{ij}} \tilde{D}_{ij} = \frac{a_i^2 - a_j^2}{a_i^2 + a_j^2} \tilde{D}_{ij}, \quad (\text{A.24})$$

because $J_{ij} = (J_j - J_i)/(3a_i^2 - 3a_j^2)$ (Eshelby, 1957, (3.13)). This equation is identical to Jiang (2012, Eq. (15)), which matches Jeffery (1922). The equation suggests that $\tilde{W}_{ij} = 0$ when $a_i = a_j$, but see also Appendix C.

Appendix B. Spheroid case: Choosing \mathbf{Q}

When two of the semi-axis lengths a_i and a_j are equal, the ellipsoid is a spheroid. There is an ambiguity in how we choose the orientations of the two equal semi-axes. Namely, they can be freely rotated about the third semi-axis, and still remain semi-axes of the spheroid. We now explain how the foregoing dynamical calculations require a particular choice of semi-axes.

When $i \neq j$ but $a_i = a_j$, Eq. (A.18) implies that $\tilde{C}_{ij} = 0$. Then, by Eqs. (A.8) and (A.5),

$$\tilde{D}_{ij} = (\mathbf{Q}\mathbf{D}\mathbf{Q}^\top)_{ij} = 0. \quad (\text{B.1})$$

Geometrically, Eq. (B.1) means that, as the deforming ellipsoid degenerates to a spheroid, the ellipsoid axes come into partial alignment with the instantaneous stretching axes of the ambient flow. In order to mitigate errors in our numerical simulation — especially when starting from an initial spheroid — we check this constraint on each time

step. That is, if the ellipsoid is a spheroid, then we adjust \mathbf{Q} to satisfy Eq. (B.1), as follows. Given \mathbf{E} , diagonalize it as $\mathbf{E} = \mathbf{Q}^\top \tilde{\mathbf{E}} \mathbf{Q}$, where $\tilde{\mathbf{E}}$ is as in Eq. (3). If $a_1 = a_2$, then compute $\tilde{\mathbf{D}} = \mathbf{Q} \mathbf{D} \mathbf{Q}^\top$,

$$\theta = \frac{1}{2} \arctan \frac{2\tilde{D}_{12}}{\tilde{D}_{22} - \tilde{D}_{11}},$$

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

(If $\tilde{D}_{22} = \tilde{D}_{11}$, then set $\theta = \pi/4$.) It is easily checked that $(\mathbf{R} \mathbf{Q} \mathbf{D} \mathbf{Q}^\top \mathbf{R}^\top)_{12} = 0$. So replace the previous choice of \mathbf{Q} with $\mathbf{R} \mathbf{Q}$. Similarly, if $a_1 = a_3$, then compute $\tilde{\mathbf{D}} = \mathbf{Q} \mathbf{D} \mathbf{Q}^\top$,

$$\theta = \frac{1}{2} \arctan \frac{2\tilde{D}_{13}}{\tilde{D}_{11} - \tilde{D}_{33}},$$

$$\mathbf{R} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix},$$

and replace \mathbf{Q} with $\mathbf{R} \mathbf{Q}$. If $a_2 = a_3$, then compute $\tilde{\mathbf{D}} = \mathbf{Q} \mathbf{D} \mathbf{Q}^\top$,

$$\theta = \frac{1}{2} \arctan \frac{2\tilde{D}_{23}}{\tilde{D}_{33} - \tilde{D}_{22}},$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix},$$

and replace \mathbf{Q} with $\mathbf{R} \mathbf{Q}$. If all three semi-axes are equal, then all three of these corrections should be performed. The effect is to completely align the ellipsoid axes with the instantaneous stretching axes of the ambient deformation. These corrections appear different from, but are in fact identical to, those of Jiang (2007a).

Appendix C. Spheroid case: Computing $\tilde{\mathbf{W}}$

Other than ensuring that \mathbf{Q} is compatible with \mathbf{D} according to Appendix B, the approaches of Sections 3.2 and 3.3 do not need to treat spheroids as a special case. The approach of Section 3.1, on the other hand, breaks down at Eq. (A.21). The indeterminate form of $\tilde{\mathbf{W}}$ in that equation suggests that we should compute a limit. Several complications arise.

First, the orientation matrix \mathbf{Q} does not depend continuously on \mathbf{E} . For example, let θ be any angle and

$$\mathbf{Q}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}.$$

Let $a_1 \neq a_2$ be any two semi-axis lengths,

$$\tilde{\mathbf{E}}(\tau) = \begin{bmatrix} a_1^{-2} & 0 & 0 \\ 0 & (a_2 + \tau)^{-2} & 0 \\ 0 & 0 & (a_2 - \tau)^{-2} \end{bmatrix},$$

and $\mathbf{E}(\theta, \tau) = \mathbf{Q}(\theta)^\top \tilde{\mathbf{E}}(\tau) \mathbf{Q}(\theta)$. For small $\tau \neq 0$, $\mathbf{E}(\theta, \tau)$ is a triaxial ellipsoid, that has $\mathbf{Q}(\theta)$ as its unique (up to permutations of the axes) orientation matrix \mathbf{Q} . That is, if we hold θ fixed, then $\lim_{\tau \rightarrow 0} \mathbf{Q} = \mathbf{Q}(\theta)$. However, all of these ellipsoids $\mathbf{E}(\theta, \tau)$ have the same limit as $\tau \rightarrow 0$, namely the spheroid $\mathbf{E}(\theta, 0) = \tilde{\mathbf{E}}(0)$. Thus the limit of \mathbf{Q} , as \mathbf{E} approaches this spheroid, depends on the direction θ of approach, and \mathbf{Q} is discontinuous in \mathbf{E} .

Nevertheless, we assume without proof that, in any particular deforming ellipsoid problem, \mathbf{Q} is a sufficiently smooth function of time t . This assumption is physically reasonable, and if it does not hold, then simulating \mathbf{Q} using numerical approximations is apparently hopeless. Then, by Eq. (A.6), $\tilde{\mathbf{W}}$ is also a smooth function of t .

Fix indices $i \neq j$, and let k be the other index, so that $i \neq k \neq j$. Suppose that a deforming ellipsoid $\mathbf{E} = \mathbf{E}(t)$ degenerates to a spheroid, so that $a_i = a_j \neq a_k$ at $t = 0$. Then, by Eqs. (A.21) and (A.5),

$$\lim_{t \rightarrow 0} \tilde{W}_{ij} = b \lim_{t \rightarrow 0} \frac{(\mathbf{Q} \mathbf{D} \mathbf{Q}^\top)_{ij}}{a_i - a_j},$$

where

$$b = \frac{a_i(0)}{1 + \frac{3(r-1)}{2\pi} a_i^2(0) J_{ij}(0)}$$

(assuming that the limit on the right exists). The limit on the right is of the form “0/0”, and thus equals

$$\frac{\lim_{t \rightarrow 0} (\dot{\mathbf{Q}} \mathbf{D} \mathbf{Q}^\top + \mathbf{Q} \mathbf{D} \dot{\mathbf{Q}}^\top)_{ij}}{\lim_{t \rightarrow 0} \dot{a}_i - \dot{a}_j}$$

by l'Hôpital's rule (assuming that these new limits exist, and that the bottom limit is nonzero).

To compute the top limit, we use Eqs. (A.5), (A.2), and (A.17). We obtain

$$c - (\tilde{D}_{ii} - \tilde{D}_{jj}) \lim_{t \rightarrow 0} \tilde{W}_{ij},$$

where

$$c = (\tilde{\mathbf{D}} \mathbf{Q} \mathbf{W} \mathbf{Q}^\top - \mathbf{Q} \mathbf{W} \mathbf{Q}^\top \tilde{\mathbf{D}})_{ij} + \tilde{W}_{ik} \tilde{D}_{kj} - \tilde{D}_{ik} \tilde{W}_{kj}$$

evaluated at $t = 0$. The bottom limit is

$$\lim_{t \rightarrow 0} (a_i \tilde{C}_{ii} - a_j \tilde{C}_{jj}) = a_i(0) \lim_{t \rightarrow 0} (\tilde{C}_{ii} - \tilde{C}_{jj})$$

by Eq. (A.16). Assume without loss of generality that $a_1 a_2 a_3 = 1$. Also, to keep the notation manageable, suppose for the moment that $i = 2$ and $j = 3$. Then, at $t = 0$, we have $a_1 = a_2^{-2} = a_3^{-2}$, $J_{22} = J_{33} = 3J_{23}$, and

$$\mathbf{J} = \begin{bmatrix} \frac{4\pi a_2^6 - 8\pi}{3a_2^2} + 8J_{23} & \frac{4\pi}{3a_2^2} - 4J_{23} & \frac{4\pi}{3a_2^2} - 4J_{23} \\ \frac{4\pi}{3a_2^2} - 4J_{23} & 3J_{23} & J_{23} \\ \frac{4\pi}{3a_2^2} - 4J_{23} & J_{23} & 3J_{23} \end{bmatrix},$$

by Eq. (A.11). The matrix \mathbf{A} of Eq. (A.10) is now simple enough to be inverted symbolically. Using that equation and the incompressibility condition $\text{tr } \tilde{\mathbf{C}} = \text{tr } \tilde{\mathbf{D}} = 0$, we

find that $\tilde{C}_{22} - \tilde{C}_{33} = b(\tilde{D}_{22} - \tilde{D}_{33})/a_2$ at $t = 0$. Similarly, in any $a_i = a_j$ case, we have

$$\tilde{C}_{ii} - \tilde{C}_{jj} = b(\tilde{D}_{ii} - \tilde{D}_{jj})/a_i$$

at $t = 0$, and the bottom limit is $b(\tilde{D}_{ii}(0) - \tilde{D}_{jj}(0))$. Typically, $\tilde{D}_{ii}(0) \neq \tilde{D}_{jj}(0)$, so the bottom limit is nonzero. Then we have

$$\lim_{t \rightarrow 0} \tilde{W}_{ij} = b \frac{c - (\tilde{D}_{ii}(0) - \tilde{D}_{jj}(0)) \lim_{t \rightarrow 0} \tilde{W}_{ij}}{b(\tilde{D}_{ii}(0) - \tilde{D}_{jj}(0))},$$

which is easily solved for the limit of \tilde{W}_{ij} . In summary, when $a_i = a_j \neq a_k$ and $\tilde{D}_{ii} \neq \tilde{D}_{jj}$,

$$\tilde{W}_{ij} = \frac{(\tilde{\mathbf{D}}\mathbf{Q}\mathbf{W}\mathbf{Q}^\top - \mathbf{Q}\mathbf{W}\mathbf{Q}^\top\tilde{\mathbf{D}})_{ij} + \tilde{W}_{ik}\tilde{D}_{kj} - \tilde{D}_{ik}\tilde{W}_{kj}}{2(\tilde{D}_{ii} - \tilde{D}_{jj})}$$

at $t = 0$.

In this appendix, we do not attempt to resolve the special subcases where $\tilde{D}_{ii} = \tilde{D}_{jj}$ or $a_i = a_j = a_k$. The calculations seem arduous, and they are not needed by our preferred methods (Sections 3.2 and 3.3). In retrospect, those methods appear to be simpler than the method of Section 3.1.

Jiang (2007a, 2012) handles the $a_i = a_j$ case by declaring, without strong justification, that $\tilde{W}_{ij} = \tilde{V}_{ij} = (\mathbf{Q}\mathbf{W}\mathbf{Q}^\top)_{ij}$ (in our notation, and using Eq. (A.12)). To compare that answer to ours, we have performed a series of numerical experiments. In each experiment, we randomly choose a spheroid \mathbf{E} , a velocity gradient tensor \mathbf{L} , and a viscosity ratio r . We deform \mathbf{E} by \mathbf{L} for a short time period, to obtain a triaxial ellipsoid that is close to the spheroid. Assuming that $\tilde{\mathbf{W}}$ varies continuously with time, the triaxial $\tilde{\mathbf{W}}$ should be close to the spheroid $\tilde{\mathbf{W}}$. The experiments consistently show the triaxial $\tilde{\mathbf{W}}$ to be significantly closer to our spheroid $\tilde{\mathbf{W}}$ than to the spheroid $\tilde{\mathbf{W}}$ of Jiang (2007a, 2012), suggesting that ours is the correct approach to computing $\tilde{\mathbf{W}}$ in the spheroid case.

Eq. (A.24) suggests that $\tilde{W}_{ij} = 0$ for rigid spheroids with $a_i = a_j$. It is noteworthy that neither our answer for the deformable spheroid \tilde{W}_{ij} , nor that of Jiang (2007a, 2012), limits to zero as $r \rightarrow \infty$. This example illustrates some of the difficulties of working with multiple limits ($a_i - a_j \rightarrow 0$ and $r \rightarrow \infty$) simultaneously.

Appendix D. Runge-Kutta methods

First we review the classic fourth-order Runge-Kutta method. We wish to solve an ordinary differential equation

$$\dot{y} = f(t, y),$$

where $y = y(t)$ is a path in some Euclidean space \mathbb{R}^m . Like the Euler method, the Runge-Kutta method divides the time interval $[0, 1]$ into n equal subintervals, each of length h , and computes approximations y_0, y_1, \dots, y_n , with $y_0 = y(0)$ and $y_n \approx y(1)$. It computes y_{s+1} from y_s by this algorithm:

1. Let $k_1 = f(hs, y_s)$.
2. Let $k_2 = f(h(s + \frac{1}{2}), y_s + \frac{h}{2}k_1)$.
3. Let $k_3 = f(h(s + \frac{1}{2}), y_s + \frac{h}{2}k_2)$.
4. Let $k_4 = f(h(s + 1), y_s + hk_3)$.
5. Then $y_{s+1} = y_s + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.

A typical Lie group is non-commutative. As a consequence, there arise two distinct but similar senses of differential geometry on the Lie group, called the *left-* and *right-invariant affine connections*. For each of these geometries, there is a Runge-Kutta method. As we present them here, the left and right Runge-Kutta methods appear dissimilar from each other, because we have tried to alter them only minimally from the existing literature. In both methods, G is a matrix Lie group, \mathfrak{g} is the corresponding Lie algebra, the bracket on \mathfrak{g} is defined by $[\mathbf{M}, \mathbf{N}] = \mathbf{M}\mathbf{N} - \mathbf{N}\mathbf{M}$, and the exponential map $\exp : \mathfrak{g} \rightarrow G$ is the matrix exponential.

Suppose that we wish to solve

$$\dot{y} = f(t, y) \cdot y,$$

where $y = y(t)$ is a path in G with given initial point $y(0)$, and $f : \mathbb{R} \times G \rightarrow \mathfrak{g}$ is any function. For example, Eq. (6) is of this form, as long as \mathbf{K} depends on t and \mathbf{F} only. Then the right-invariant fourth-order Runge-Kutta method of Munthe-Kaas (1999) applies. It evolves y_s to y_{s+1} over a time step of size h as follows:

1. Let $k_1 = f(hs, y_s)$.
2. Let $k_2 = \text{dexpinv}(\frac{h}{2}k_1, f(h(s + \frac{1}{2}), \exp(\frac{h}{2}k_1)y_s))$.
3. Let $k_3 = \text{dexpinv}(\frac{h}{2}k_2, f(h(s + \frac{1}{2}), \exp(\frac{h}{2}k_2)y_s))$.
4. Let $k_4 = \text{dexpinv}(hk_3, f(h(s + 1), \exp(hk_3)y_s))$.
5. Then $y_{s+1} = \exp(\frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4))y_s$.

Here, $\text{dexpinv} : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$ is defined (to fourth order) by

$$\text{dexpinv}(u, v) = v - \frac{1}{2}[u, v] + \frac{1}{24}[u, [u, v]].$$

Similarly, suppose that we wish to solve

$$\dot{y} = y \cdot f(t, y). \quad (\text{D.1})$$

Then the left-invariant fourth-order Runge-Kutta method of Munthe-Kaas (1998) applies. This method evolves y_s to y_{s+1} over a time step of size h as follows.

1. Let $k_1 = f(hs, y_s)$.
2. Let $k_2 = f(h(s + \frac{1}{2}), y_s \exp(\frac{h}{2}k_1))$.
3. Let $k_3 = f(h(s + \frac{1}{2}), y_s \exp(\frac{h}{2}k_2 + \frac{h^2}{24}[k_1, k_2]))$.
4. Let $k_4 = f(h(s + 1), y_s \exp(hk_3 + \frac{h^2}{6}[k_1, k_3]))$.
5. Let $v = \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.
6. Let $w = \frac{h}{24}(3k_1 + 2k_2 + 2k_3 - k_4)$.
7. Then $y_{s+1} = y_s \exp(v + [w, v])$.

The equation $\dot{\mathbf{F}} = \mathbf{K}\mathbf{F}$ can be rewritten in the form of Eq. (D.1), as follows. By the product rule for differentiation of matrices,

$$0 = \frac{d}{dt}(\mathbf{F}\mathbf{F}^{-1}) = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1} + \mathbf{F} \cdot \frac{d}{dt}(\mathbf{F}^{-1}).$$

Combining this equation with $\dot{\mathbf{F}} = \mathbf{K}\mathbf{F}$ yields

$$\frac{d}{dt}(\mathbf{F}^{-1}) = \mathbf{F}^{-1} \cdot -\mathbf{K},$$

which is of the desired form as long as \mathbf{K} depends on t and \mathbf{F}^{-1} only.

Appendix E. Miscellaneous computations

In this appendix, we describe two averaging techniques with a Lie-theoretic flavor: averaging ellipsoids, and averaging lineation-foliation pairs. These concepts are used in Section 5.2. We also describe how logarithms enter into the 39 equations of that subsection.

For any set of m ellipsoid tensors $\mathbf{E}_1, \dots, \mathbf{E}_m$, one can define the arithmetic and geometric means by

$$\begin{aligned} \text{arith}(\mathbf{E}_1, \dots, \mathbf{E}_m) &= \frac{1}{m} \sum_{i=1}^m \mathbf{E}_i, \\ \text{geom}(\mathbf{E}_1, \dots, \mathbf{E}_m) &= \exp \frac{1}{m} \sum_{i=1}^m \log \mathbf{E}_i \end{aligned}$$

(Wheeler, 1986; Brandon, 1995), where \exp and \log denote the matrix exponential and logarithm functions (Provost et al., 2004; Davis and Titus, 2011). These concepts of mean are called ‘‘arithmetic’’ and ‘‘geometric’’ because, when they are applied to symmetric, positive-definite 1×1 tensors, they reproduce the ordinary arithmetic and geometric means of numbers.

It is easy to show that, because the \mathbf{E}_i are symmetric and positive-definite, so are the arithmetic and geometric means. That is, both means constitute well-defined notions of average ellipsoid. However, the geometric mean enjoys several convenient properties that the arithmetic mean lacks. For example, if all \mathbf{E}_i are normalized to have determinant one, then the geometric mean has determinant one as well. For another example, when the \mathbf{E}_i have the same orientation but differ in magnitude, the geometric mean of the tensors amounts to a geometric mean of semi-axis lengths. The arithmetic mean of the tensors amounts to a more complicated operation on these lengths.

Similarly, we can average lineation-foliation pairs, as follows. First, let \mathbf{l} and \mathbf{f} be unit vectors aligned with the lineation and foliation pole. Fix a coordinate system. Let \mathbf{R} be the matrix whose rows are \mathbf{l} , $\mathbf{f} \times \mathbf{l}$, and \mathbf{f} . Then \mathbf{R} is a rotation matrix, and $\{\mathbf{l}, \mathbf{f} \times \mathbf{l}, \mathbf{f}\}$ is a right-handed orthonormal coordinate frame, obtained by applying the rotation $\mathbf{R}^{-1} = \mathbf{R}^\top$ to the basis vectors of the geographic coordinate frame. In this manner, a lineation-foliation pair can be regarded as an oriented coordinate frame, or as a rotation, which is an element of the Lie group $\text{SO}(3)$ (Section 3.3).

The question, then, is how to compute averages in $\text{SO}(3)$. Let $\mathbf{R}_1, \dots, \mathbf{R}_m$ be m rotations. Unless it happens to be a rotation by $\pi = 180^\circ$, each \mathbf{R}_i has a principal matrix logarithm $\log \mathbf{R}_i$, all of whose eigenvalues have

imaginary part in the interval $(-\pi, \pi)$ (Davis and Titus, 2011, Appendix C). Each $\log \mathbf{R}_i$ is an element of $\mathfrak{so}(3)$, meaning that it is antisymmetric. Thus $\frac{1}{m} \sum_i \log \mathbf{R}_i$ is antisymmetric, and hence

$$\bar{\mathbf{R}} = \exp \frac{1}{m} \sum_{i=1}^m \log \mathbf{R}_i$$

is a rotation, which we take to be the average of the m given rotations.

This procedure sometimes produces unexpected results, due to the many-to-one nature of the exponential map from $\mathfrak{so}(3)$ to $\text{SO}(3)$. In other words, the logarithm involves a branch cut, which we have chosen to place along the set of rotations by π . There is an analogous problem in two dimensions, in which a heading of 359° and a heading of 001° average to 180° rather than 000° , due to a branch cut at 000° . We do not attempt to describe a general, fully reliable rotation-averaging method here. The procedure just described suffices for our application in Section 5.

Finally, in Section 5.2, in each deforming clast subproblem, we must compute the difference between an observed clast ellipsoid and a predicted clast ellipsoid. Because the logarithms inhabit a vector space and the ellipsoid tensors do not, it makes more mathematical sense to use a Euclidean distance on the logarithms than on the ellipsoid tensors themselves. Thus the distance between ellipsoids \mathbf{E} and \mathbf{E}' is computed as

$$\text{dist}(\mathbf{E}, \mathbf{E}') = \sqrt{\sum_{i=1}^3 \sum_{j=1}^3 (\log \mathbf{E} - \log \mathbf{E}')_{ij}^2}.$$

In the lineation/foliation subproblem, we similarly ask the logarithm of Δ_1 to be diagonal, rather than Δ_1 itself. This tactic makes the weight of the lineation/foliation subproblem more commensurate with the weights of the clast subproblems. If this tactic is not employed, then the total error of the model is highly sensitive to noise in the lineation/foliation data, when finite strain ellipsoid magnitudes are large.