

SANDIA REPORT

SAND2013-10281

Unlimited Release

Printed December 2013

Edge Remap for Solids

James R. Kamm, Edward Love, Allen C. Robinson, Joseph S. Young, Denis Ridzal

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

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

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

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

Printed in the United States of America. This report has been reproduced directly from the best available copy.

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

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

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

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



Edge Remap for Solids

James R. Kamm
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545
jrkamm@lanl.gov

Edward Love, Allen C. Robinson
Computational Shock and Multi-Physics Department
Sandia National Laboratories, P.O. Box 5800
Albuquerque, NM 87185-1323
elove@sandia.gov

Joseph S. Young
Numerical Analysis and Applications Department
Sandia National Laboratories, P.O. Box 5800
Albuquerque, NM 87185-1320

Denis Ridzal
Optimization and Uncertainty Quantification Department
Sandia National Laboratories, P.O. Box 5800
Albuquerque, NM 87185-1318
dridzal@sandia.gov

Abstract

We review the edge element formulation for describing the kinematics of hyperelastic solids. This approach is used to frame the problem of remapping the inverse deformation gradient for Arbitrary Lagrangian-Eulerian (ALE) simulations of solid dynamics. For hyperelastic materials, the stress state is completely determined by the deformation gradient, so remapping this quantity effectively updates the stress state of the material. A method, inspired by the constrained transport remap in electromagnetics, is reviewed, according to which the zero-curl constraint on the inverse deformation gradient is implicitly satisfied. Open issues related to the accuracy of this approach are identified. An optimization-based approach is implemented to enforce positivity of the determinant of the deformation gradient. The efficacy of this approach is illustrated with numerical examples.

Acknowledgment

This work was supported by the Computer Science Advanced Research program at Sandia National Laboratories.

This page intentionally left blank.

Contents

1	Introduction	11
2	Element Interpolation	13
3	Kinematics	15
	The deformation gradient	15
	Constraints	17
4	Curl-Preserving Remap	19
	Reconstruction	19
	Nodal values	20
	High-order edge representation	20
	Update	21
	Curl-preserving remap implementation	25
5	Numerical Examples of Curl-Preserving Remap	27
	Convergence rates for problems with exact solutions	27
	Exponential vortical flow	28
	ABC vortical flow	32
6	Enforcing Extrema-Preservation on the Remap	37
	Constrained optimization overview	37
	Constrained optimization implementation	39
7	Numerical Examples of Extrema-Preserving Remap	43

Exponential vortical flow	43
ABC vortical flow	48
8 Summary	53
References	54
Appendix	
A Curl of a Tensor	59
B Angular Flow Exact Solutions	61
C The Matrix Adjugate	63

List of Figures

3.1	Notional depiction of the deformation of material	16
4.1	Depiction of the 2D case for the parent cell that is the pre-image of the k th mesh element	20
4.2	Depiction of the integration path for the remap scheme in 2D	22
4.3	Depiction of the orientation convention used for the remap scheme	23
4.4	Determination of the upwind element for a node in the Lagrangian-updated mesh .	24
5.1	Magnitude of the stretch tensor for the exponential vortical flow problem, Lagrangian (no remap) and Eulerian (Lagrangian plus zero-curl remap every timestep) calculations	29
5.2	Magnitude of the stretch tensor for the ABC vortical flow problem, Lagrangian (no remap) and Eulerian (Lagrangian plus zero-curl remap every timestep) calculations	33
7.1	Magnitude of the stretch tensor for the exponential vortical flow problem, QR+clipped remap every timestep and optimized zero-curl remap every timestep calculations . .	45
7.2	Magnitude of the stretch tensor for the ABC vortical flow problem, QR+clipped remap every timestep and optimized zero-curl remap every timestep calculations . .	49

List of Tables

5.1	Stretch matrix convergence rates for the exponential vortical flow problem, Lagrangian (no remap) and Eulerian (Lagrangian plus zero-curl remap every timestep) calculations	30
5.2	Rotation matrix convergence rates for the exponential vortical flow problem, Lagrangian (no remap) and Eulerian (Lagrangian plus zero-curl remap every timestep) calculations	31
5.3	Stretch matrix convergence rates for the ABC vortical flow problem, Lagrangian (no remap) and Eulerian (Lagrangian plus zero-curl remap every timestep) calculations	34
5.4	Rotation matrix convergence rates for the ABC vortical flow problem, Lagrangian (no remap) and Eulerian (Lagrangian plus zero-curl remap every timestep) calculations	35
7.1	Stretch matrix convergence rates for the exponential vortical flow problem, QR+clipped remap every timestep and optimized zero-curl remap every timestep calculations . .	46
7.2	Rotation matrix convergence rates for the exponential vortical flow problem, QR+clipped remap every timestep and optimized zero-curl remap every timestep calculations . .	47
7.3	Stretch matrix convergence rates for the ABC vortical flow problem, QR+clipped remap every timestep and optimized zero-curl remap every timestep calculations . .	50
7.4	Rotation matrix convergence rates for the ABC vortical flow problem, QR+clipped remap every timestep and optimized zero-curl remap every timestep calculations . .	51

Chapter 1

Introduction

The focus of this work is remapping for computational solid mechanics simulations in the Arbitrary Lagrangian-Eulerian (ALE) framework. In the ALE approach, the Lagrangian step updates the equations of motion in the frame of reference of the moving material. In the Eulerian or remap step, material properties are mapped to coordinates on another mesh. We restrict our analysis to ALE methods for which the remap mesh corresponds to the original mesh configuration. Robust and accurate formulation of the remap step remains a formidable problem. One reason is that any computational scheme for the kinematics of solid deformation will break down for sufficiently great material deformation. Another reason is the challenge of satisfying the fundamental constraints of solid dynamics in the computational remap procedures. The methods described in this report intrinsically comply with certain constraints implied by the mathematical modeling of solid dynamics.

The mathematics that underpin the techniques used here involves the coupling of algebraic topology with exterior calculus, applied in a discrete setting. This combination of topics has been discussed by several researchers. Tonti, in [43] and subsequent works (see also Mattiussi [37]), was among the first to develop a framework for categorizing the governing equations for numerous physical phenomena in this manner, thereby laying the foundation for this approach. This hybrid subject, aspects of which are discussed in the volume edited by Arnold et al. [4], the recent review of Christiansen et al. [23], and, more abstractly, e.g., by Dezin [25] and Hirani [31], is succinctly summarized by Bochev [6].

The original and most mature development of these ideas is in the field of electromagnetics. Bossavit [11, 12, 13, 14, 15, 16, 17, 18, 19, 20] laid out the combination of differential geometry, exterior calculus, and algebraic topology for computational electromagnetics (see also Gross and Kotiuga [29]). The seminal work of Evans and Hawley [28] introduced the constrained transport algorithm, a mimetic method for advection of magnetic flux density, $\mathbf{B}$, on structured meshes. This method preserves a discrete version of the $\nabla \cdot \mathbf{B} = 0$ property. As originally presented, the constrained transport algorithm is applicable to a staggered field representation with the magnetic flux represented on cell faces and the electric fields on cell edges. In the finite element context, the appropriate generalization of this representation is given by edge and face finite element bases. Bochev and Robinson [10] and Day et al. [24] developed these concepts further. Bochev et al. [7] applied these constrained transport ideas to the remapping of magnetic flux in the ALE framework for unstructured meshes.

Research has also been conducted on developing a comparable formalism for the dynamics of elastic solids. A careful derivation of the conservation laws for these dynamics is given by Wagner [45]; see also [44]. Arnold et al. [5, 3] describe a merging of exterior calculus with the finite element method for elasticity, leading to the development of the so-called elastic complex. A related and more general framework is developed by Eastwood [26, 27]. Yavari [46] and Angoshtari & Yavari [2] propose a different approach to the discretization of elasticity, pursuing a fundamentally geometric perspective. Bossavit [21], building upon his monumental contributions to computational electromagnetics, considers magnetoelasticity in the edge element framework, identifying many interesting analogies between the two subjects.

Robinson et al. [41], motivated by the work of Bochev et al. [10, 7] on computational MHD, extended that approach to the remap problem in solid dynamics, applying constrained transport to the Lagrangian positions. One fundamental constraint for elastodynamics is that the deformation gradient representation remains in a curl-free space. One way to achieve this is the reference map technique of Kamrin et al. [32], who pursue an Eulerian-frame finite-difference implementation. Instead, Robinson et al. considered the Lagrangian coordinates to be the “potentials” of the inverse deformation gradient fields. To achieve this, a space of curl-free edge elements is constructed, and the deformation gradient is represented in terms of these edge elements.

The structure of this report is as follows. The basic edge element formulation is reviewed in Chapter 2. In Chapter 3, the representation of the inverse deformation gradient in terms of edge basis functions is derived. The application of this representation for remapping the inverse deformation gradient is described in Chapter 4, which also contains a summary of the algorithmic implementation of this method. Numerical examples are evaluated in Chapter 5. Chapter 6 presents one possible approach by which to ensure no spurious maxima are introduced to the curl-free remap result; numerical results of this optimized approach are given in Chapter 7. We summarize this work in Chapter 8.

Chapter 2

Element Interpolation

Consider a finite element mesh consisting of a set of convex, nondegenerate hexahedra. Following the notation of Bochev and Robinson [10] and Bochev et al. [10, 7], let $\mathcal{N}$, $\overrightarrow{\mathcal{E}}$, $\overrightarrow{\mathcal{F}}$, and $\mathcal{K}$ denote the sets of all nodes, oriented edges, oriented faces, and hexahedra that constitute the mesh. The union of all hexahedra comprises the entire grid, denoted $\Omega \subset \mathbb{R}^3$.

Consider first the set of edges $\overrightarrow{\mathcal{E}}$ in the mesh. Associated with edge $\overrightarrow{\mathcal{E}}_i \in \overrightarrow{\mathcal{E}}$ is a vector-valued interpolation function $\overrightarrow{\mathbf{W}}_i : \Omega \rightarrow \mathbb{R}^3$ with compact support. Assume that the edge interpolation functions satisfy the orthogonality property

$$\int_{\overrightarrow{\mathcal{E}}_i} \overrightarrow{\mathbf{W}}_j \cdot \mathbf{t} \, ds = \delta_{ij} , \quad (2.1)$$

where $\mathbf{t} \in \mathbb{R}^3$ is the (unit) tangent vector to the edge and s parameterizes the arc length along the edge. Bochev et al. [7] give definitions of such edge interpolation functions, used in constructing solutions of Maxwell's equations. The primary use of these basis functions is to approximate the fields within a computational element. As pointed out by Bochev and Robinson [10] and Rieben et al. [40], across shared faces of adjacent elements quantities expressed as sums of these basis functions have (i) continuous tangential components and (ii) discontinuous normal components.

Now consider the set $\mathcal{N}$ of all nodes in the mesh. Associated with the A th node is a scalar-valued interpolation function $N_A : \Omega \rightarrow \mathbb{R}$ with compact support. Additionally, let $f : \Omega \rightarrow \mathbb{R}$ be a scalar function defined by nodal values f_A , $A = 1, \dots, \dim \mathcal{N}$. One may extend the definition of f to all $\mathbf{x} \in \Omega$ by interpolation of these node values as:

$$f(\mathbf{x}) = \sum_{A=1}^{\dim \mathcal{N}} f_A N_A(\mathbf{x}) . \quad (2.2)$$

This scalar-valued function has a vector-valued gradient, computed as

$$\nabla f = \sum_{A=1}^{\dim \mathcal{N}} f_A \nabla N_A . \quad (2.3)$$

As shown by Bochev et al. [7], there exists a unique set of scalar parameters a_i such that

$$\nabla f = \sum_{i=1}^{\dim \overrightarrow{\mathcal{E}}} a_i \overrightarrow{\mathbf{W}}_i . \quad (2.4)$$

To evaluate the a_i , begin by denoting Δf_i as the increment of the function f along an edge $\vec{\mathcal{E}}_i$. For example, with edge $\vec{\mathcal{E}}_i$ defined as the directed line segment connecting two nodes $\mathcal{N}_A$ and $\mathcal{N}_B$, Δf_i is the (tangent-directed) difference of the values of f at the two nodes delimiting this edge. Each edge has an orientation that determines the sign of this difference. Line integration along this edge yields

$$\begin{aligned}
\Delta f_i &= \int_{\vec{\mathcal{E}}_i} \nabla f \cdot \mathbf{t} \, ds \\
&= \int_{\vec{\mathcal{E}}_i} \left[\sum_{j=1}^{\dim \vec{\mathcal{E}}} a_j \vec{\mathbf{W}}_j \right] \cdot \mathbf{t} \, ds \\
&= \sum_{j=1}^{\dim \vec{\mathcal{E}}} a_j \left[\int_{\vec{\mathcal{E}}_i} \vec{\mathbf{W}}_j \cdot \mathbf{t} \, ds \right] \\
&= \sum_{j=1}^{\dim \vec{\mathcal{E}}} a_j \delta_{ij} \\
&= a_i .
\end{aligned} \tag{2.5}$$

Thus the gradient of the scalar function $f(\mathbf{x})$ can be written as

$$\nabla f(\mathbf{x}) = \sum_{e=1}^{\dim \vec{\mathcal{E}}} \Delta f_e \vec{\mathbf{W}}_e(\mathbf{x}) .$$

(2.6)

Chapter 3

Kinematics

This section contains a review of basic kinematics associated with the edge-based remap approach. Denote by $\mathbf{X}$ the material coordinates relative to a fixed Cartesian coordinate system with basis $\{\mathbf{E}_A\}$, such that $\mathbf{X} = X_A \mathbf{E}_A$. Similarly, denote by $\mathbf{x}$ spatial coordinates relative to a fixed Cartesian coordinate system with basis $\{\mathbf{e}_i\}$, such that $\mathbf{x} = x_i \mathbf{e}_i$ ¹. Given a vector or tensor $\mathbf{T}$, let $[\mathbf{T}]$ be the matrix representation of $\mathbf{T}$ in the appropriate fixed Cartesian basis.

The deformation gradient

The standard kinematics between the material coordinates and the physical coordinates of the deformed body are depicted notionally in Fig. 3.1. The deformation gradient is defined as

$$\mathbf{F} := \frac{\partial x_i}{\partial X_A} \mathbf{e}_i \otimes \mathbf{E}_A, \quad (3.1)$$

where $\mathbf{e}_i \otimes \mathbf{E}_A$ is the tensor product of the basis vectors $\mathbf{e}_i$ and $\mathbf{E}_A$. Consistent with this nomenclature, define the (covariant) spatial basis vectors

$$\mathbf{g}_A := \frac{\partial \mathbf{x}}{\partial X_A} = \frac{\partial x_i}{\partial X_A} \mathbf{e}_i. \quad (3.2)$$

Thus, the deformation gradient can be written in matrix form, with respect to the $\mathbf{e}_i \otimes \mathbf{E}_A$ basis, as

$$[\mathbf{F}] = \left[\begin{array}{c|c|c} [\mathbf{g}_1] & [\mathbf{g}_2] & [\mathbf{g}_3] \end{array} \right] \quad (3.3)$$

where $[\mathbf{g}_A]$ is the A th column of $[\mathbf{F}]$.

The inverse deformation gradient (IDG) is defined similarly as

$$\mathbf{F}^{-1} := \frac{\partial X_A}{\partial x_i} \mathbf{E}_A \otimes \mathbf{e}_i. \quad (3.4)$$

¹We often—but not always—use the convention that repeated indices imply that the quantities are to be summed, such as in this expression. Throughout this report, whether or not this implicit summation applies should be obvious by the context.

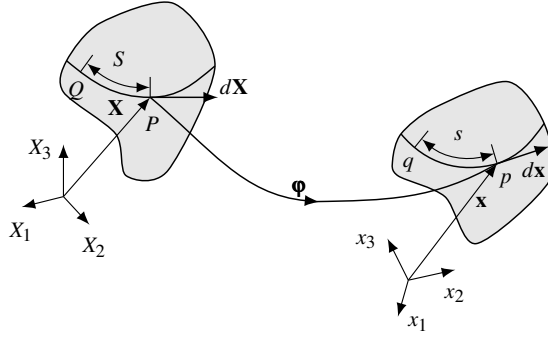


Figure 3.1. Notional depiction of the deformation of material. The motion $\boldsymbol{\varphi}$ maps the material coordinates $\mathbf{X}$ to the spatial coordinates $\mathbf{x}$, both Cartesian systems. Under the action of the motion $\boldsymbol{\varphi}$, the arc S along a material line between the points Q and P is mapped to the arc s between the corresponding image points q and p in physical coordinates. Correspondingly, $d\mathbf{X}$, the local tangent vector at point P , is mapped to local tangent vector $d\mathbf{x}$. These local tangent vectors are related to each other through the deformation gradient: $d\mathbf{x} = \mathbf{F}d\mathbf{X}$. As this relation holds locally for all material lines through P and their images through p , the deformation gradient $\mathbf{F}$ represents the mapping between the two tangent spaces at $\mathbf{X}$ and $\mathbf{x}$.

Consistent with this representation, define the (contravariant) spatial basis vectors

$$\mathbf{g}^A = \frac{\partial X_A}{\partial \mathbf{x}} = \frac{\partial X_A}{\partial x_i} \mathbf{e}_i = \nabla X_A . \quad (3.5)$$

By definition, $\mathbf{g}^A \cdot \mathbf{g}_B = \delta^A_B$. Consistent with (3.3), $\mathbf{F}^{-1}$ can be written in matrix form, with respect to the $\mathbf{E}_A \otimes \mathbf{e}_i$ basis, as

$$[\mathbf{F}^{-1}] = \begin{bmatrix} [\mathbf{g}^1]^\top \\ \vdots \\ [\mathbf{g}^2]^\top \\ \vdots \\ [\mathbf{g}^3]^\top \end{bmatrix} , \quad (3.6)$$

i.e., $[\mathbf{g}^A]^\top$ is the A th row of $[\mathbf{F}^{-1}]$. By this construction

$$\begin{aligned} \mathbf{F} \mathbf{E}_A &= \mathbf{g}_A \iff \mathbf{F} = \mathbf{g}_A \otimes \mathbf{E}_A \\ &\iff \mathbf{F}^{-1} = \mathbf{E}_A \otimes \mathbf{g}^A \\ &\iff \mathbf{F}^{-\top} = \mathbf{g}^A \otimes \mathbf{E}_A \\ &\iff \mathbf{g}^A = \mathbf{F}^{-\top} \mathbf{E}_A . \end{aligned} \quad (3.7)$$

Consider now the edges of the mesh and denote by $\Delta \mathbf{X}_e$ the increment in $\mathbf{X}$ along the e th edge.

Then, from (3.5) and (2.6),

$$\begin{aligned}
\mathbf{g}^A &= \nabla X_A \\
&= \nabla(\mathbf{X} \cdot \mathbf{E}_A) \\
&= \sum_{e=1}^{\dim \vec{\mathcal{E}}} (\Delta \mathbf{X}_e \cdot \mathbf{E}_A) \vec{\mathbf{W}}_e \\
&= \left[\sum_{e=1}^{\dim \vec{\mathcal{E}}} \vec{\mathbf{W}}_e \otimes \Delta \mathbf{X}_e \right] \mathbf{E}_A,
\end{aligned} \tag{3.8}$$

where we have invoked the equality $(\mathbf{a} \otimes \mathbf{b}) \mathbf{c} = \mathbf{a}(\mathbf{b} \cdot \mathbf{c})$. Comparing this result with (3.7) clearly implies that

$$\mathbf{F}^{-\top} = \sum_{e=1}^{\dim \vec{\mathcal{E}}} \vec{\mathbf{W}}_e \otimes \Delta \mathbf{X}_e. \tag{3.9}$$

Taking the transpose, one obtains the following expression for the inverse deformation gradient:

$$\boxed{\mathbf{F}^{-1} = \sum_{e=1}^{\dim \vec{\mathcal{E}}} \Delta \mathbf{X}_e \otimes \vec{\mathbf{W}}_e.} \tag{3.10}$$

This expression can be numerically evaluated at any point $\mathbf{x}$ inside a mesh element, using the values of the edge basis functions $\vec{\mathbf{W}}_e$ and the increments in material coordinates along the edges $\Delta \mathbf{X}_e$.

Constraints

There are several constraints that restrict which material deformations are admissible. The associated numerical solutions, likewise, must satisfy discrete analogues of these constraints. The first constraint follows from the equality of second derivatives of the material coordinates relative to the current configuration:

$$\frac{\partial^2 X_A}{\partial x_i \partial x_j} = \frac{\partial^2 X_A}{\partial x_j \partial x_i}. \tag{3.11}$$

Using the definition of the inverse deformation gradient in (3.4), this equality implies

$$F_{Aj,i}^{-1} = F_{Ai,j}^{-1}, \tag{3.12}$$

where the comma (,) denotes differentiation. The curl of a tensor $\mathbf{T} := T_{Ai} \mathbf{E}_A \otimes \mathbf{e}_i$ is defined [22] to satisfy the relation

$$(\text{curl } \mathbf{T}) \mathbf{v} := \text{curl} \left(\mathbf{T}^\top \mathbf{v} \right) \tag{3.13}$$

for all vectors $\mathbf{v}$ that are independent of $\mathbf{x}$, where the curl on the RHS of the definition above is the usual vector-curl operator. As shown in Appendix A, this definition implies

$$\text{curl } \mathbf{T} := -\epsilon_{ijk} T_{Li,j} \mathbf{e}_k \otimes \mathbf{E}_L, \tag{3.14}$$

where ε_{ijk} is the usual permutation symbol. Using this expression, the equality of (3.12) implies that

$$\boxed{\operatorname{curl} \mathbf{F}^{-1} = \mathbf{0}} \quad (3.15)$$

for twice-differentiable material coordinates.

The second constraint follows from the local continuity of the motion. Let the mapping between spatial coordinates $\mathbf{x}$ and the material coordinates $\mathbf{X}$ be denoted $\boldsymbol{\varphi}$, so that $\mathbf{x} = \boldsymbol{\varphi}(\mathbf{X}, t)$. Consistent with the above assumptions, assume that $\boldsymbol{\varphi}$ is sufficiently differentiable with respect to $\mathbf{X}$. The second constraint is summarized by Lubliner ([34], p. 466, emphasis in the original, which uses the symbol $\boldsymbol{\chi}$ for $\boldsymbol{\varphi}$):

If, in a neighborhood of the material point $\mathbf{X}$, the function $\boldsymbol{\varphi}(\mathbf{X}, t)$ is invertible — in other words, if the material points in the neighborhood are in one-to-one correspondence with their displaced positions — then, by the **implicit function theorem** of advanced calculus, the matrix of components of $\mathbf{F}(\mathbf{X}, t)$ (the *Jacobian* matrix) must be nonsingular, that is, $J(\mathbf{X}, t) \neq 0$, where $J(\mathbf{X}, t) \stackrel{\text{def}}{=} \det \mathbf{F}(\mathbf{X}, t)$ is the *Jacobian determinant*. If we consider only displaced configurations that can evolve continuously from one another, then since $J = 1$ when the displaced and reference configurations coincide, we obtain the stronger condition $J(\mathbf{X}, t) > 0$.

We restate this constraint as:

$$\boxed{\det \mathbf{F} > 0.} \quad (3.16)$$

More specifically, $0 < \det \mathbf{F} < \infty$, i.e., under a polar decomposition $\mathbf{F} = \mathbf{V}\mathbf{R}$, the principal stretches (eigenvalues of $\mathbf{V}$) are all positive and finite, and the rotation tensor $\mathbf{R}$ is proper orthogonal (i.e., $\mathbf{R} \in SO(3)$).

Chapter 4

Curl-Preserving Remap

As discussed in Chapter 1, remapping plays a central role in the ALE framework. There are three primary considerations that affect the fidelity of the ALE remap. The first two are related to the constraints described above: (1) the zero-curl constraint of (3.15) and (2) the positive-determinant constraint of (3.16). The third is associated with the inevitable loss of accuracy due to finite-precision arithmetic for very small or very large strains; this is manifest as very small or very large eigenvalues in the stretch tensor $\mathbf{V}$. While the second and third remain active research topics, in this section we provide details of a method that satisfies the first of these constraints.

The edge element formulation ensures the curl-free property of the remapped inverse deformation gradient. The degrees of freedom on each edge are the coordinate differences $\Delta \mathbf{X}_e$, which are initialized with the point-wise, signed difference of the corresponding initial Lagrangian coordinates. The Lagrangian update step guarantees that these values are unchanged. For the remap, we define a high-order representation of this field by extending the edge element description to include gradients of the edge-based coordinate differences, in effect providing a higher-order representation of the coordinate difference field. With this enhanced representation, an upwind nodal flux contribution is calculated at each node. The gradient of this flux is used to update the edge-based coordinate differences. This construction ensures that the curl of the inverse deformation gradient remains identically zero after the remap. While this approach implies consistency of the solution with respect to the curl-free condition, it does not guarantee the numerical accuracy of the computed result. Various aspects of the actual implementation affect the overall accuracy of the algorithm.

Reconstruction

In this section, we obtain an edge-element based, high-order representation for the inverse deformation gradient. This is achieved in two steps: the inverse deformation gradient is first mapped to the nodes of an element, and then a high-order representation for the material coordinate differences on edges is developed.

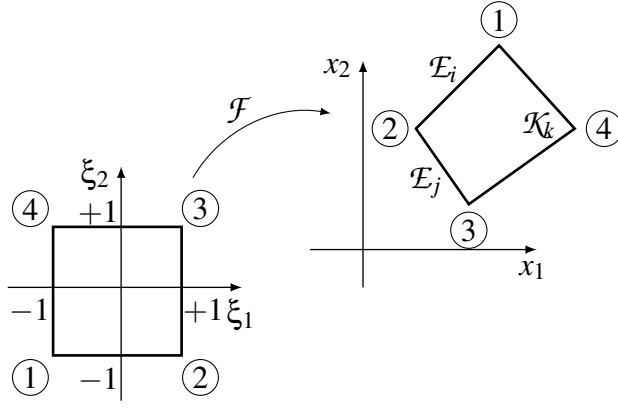


Figure 4.1. Depiction of the 2D case for the parent cell (left) that is the pre-image of the mesh element K_k (right) under the mapping $\mathcal{F}$ between local natural and physical coordinates. The corresponding nodes are identically numbered. In the case depicted, for the edge E_i the parallel natural coordinate $\xi_{\parallel}$ is given by ξ_1 , while for edge E_j the parallel natural coordinate $\xi_{\parallel}$ is given by ξ_2 .

Nodal values

Equation (3.10) provides a compact expression for $\mathbf{F}^{-1}$. Assume that this representation corresponds to the Lagrangian-updated values, so that this expression furnishes value of $\mathbf{F}^{-1}$ at any point within a cell. At a given node, however, the edge element representations from the adjacent cells need not be continuous. To extend the definition of $\mathbf{F}^{-1}$ relative to a given cell, one must first use this information to assign values at a given node. This process is well known in the finite element community as patch recovery, according to which nodal values are inferred from the surrounding patch of elements. Therefore, for the case at hand, a patch recovery operator was used [49, 48, 1].

High-order edge representation

This section contains a procedure to obtain a high-order representation of $\Delta \mathbf{X}$ at the nodes. Let $\xi_{\parallel}$ denote the natural coordinate (i.e., of the parent cell) that runs tangent to edge E_e . Without loss of generality, the natural (Cartesian) coordinates of the parent element of a given cell are specified to run from -1 to $+1$; see Fig. 4.1. To obtain high-order accurate estimates of $\mathbf{F}^{-1}$ at the nodes, the definition of the edge element coefficient is extended to include linear variation with $\xi_{\parallel}$:

$$\Delta \mathbf{X}_e(\xi_{\parallel}) := \overline{\Delta \mathbf{X}}_e + S_e \xi_{\parallel} . \quad (4.1)$$

where $\overline{\Delta \mathbf{X}}_e$ is the value of the material coordinate difference along edge e ; this value is unchanged during the Lagrangian update step. Evaluating this relation at each node on this edge implies:

$$S_e(-1) = -(\Delta \mathbf{X}_e(-1) - \overline{\Delta \mathbf{X}}_e) \quad \text{and} \quad S_e(+1) = \Delta \mathbf{X}_e(+1) - \overline{\Delta \mathbf{X}}_e . \quad (4.2)$$

Values for the edge-based coordinate differences at each node on this edge, i.e., $\Delta\mathbf{X}_e(-1)$ and $\Delta\mathbf{X}_e(+1)$ in (4.2), can be obtained as follows. Application of the chain rule shows that the edge-based coordinate difference can be expressed as

$$\Delta\mathbf{X}_e = \int_{-1}^1 \frac{\partial\mathbf{X}}{\partial\xi_{\parallel}} d\xi_{\parallel} = \int_{-1}^1 \frac{\partial\mathbf{X}}{\partial\mathbf{x}} \frac{\partial\mathbf{x}}{\partial\xi_{\parallel}} d\xi_{\parallel} = \int_{-1}^1 \mathbf{F}^{-1} \frac{\partial\mathbf{x}}{\partial\xi_{\parallel}} d\xi_{\parallel} . \quad (4.3)$$

Approximate the integral in this expression using the value of its argument at each of the nodes:

$$\Delta\mathbf{X}_e(\pm 1) \approx \mathbf{F}^{-1}(\pm 1) \frac{\partial\mathbf{x}}{\partial\xi_{\parallel}} \int_{-1}^1 d\xi_{\parallel} = 2\mathbf{F}^{-1}(\pm 1) \frac{\partial\mathbf{x}}{\partial\xi_{\parallel}} , \quad (4.4)$$

where the argument ± 1 refers to the value of $\xi_{\parallel}$ at which $\mathbf{F}^{-1}$ is evaluated. The term $\partial\mathbf{x}/\partial\xi_{\parallel}$ is just the tangent vector along this edge of the mapping $\mathcal{F}$ from natural to physical coordinates; denote this vector as $\mathbf{V}_{\xi_{\parallel}}$. Simplifying (4.4) leads to the following values for the coordinate differences at the two nodes on edge e :

$$\Delta\mathbf{X}_e(-1) = 2\mathbf{F}^{-1}(-1) \mathbf{V}_{\xi_{\parallel}} \quad \text{and} \quad \Delta\mathbf{X}_e(+1) = 2\mathbf{F}^{-1}(+1) \mathbf{V}_{\xi_{\parallel}} . \quad (4.5)$$

Substituting these values into (4.2) gives unique values for the two possible slopes in the expression (4.1). To choose between these values, a slope limiter is used. Perhaps the simplest way to implement such a limiter is to limit each component separately. With the minmod limiter [33], this can be written compactly in vector form as:

$$\mathbf{S}_e = \text{minmod}(\mathbf{S}_e(-1), \mathbf{S}_e(+1)) , \quad (4.6)$$

where the minmod function operates on each component of its vector arguments, i.e.,

$$\text{minmod}(\mathbf{V}, \mathbf{W}) := \sum_{A=1}^3 \text{minmod}(\mathbf{V} \cdot \mathbf{E}_A, \mathbf{W} \cdot \mathbf{E}_A) \mathbf{E}_A , \quad (4.7)$$

with

$$\text{minmod}(a, b) := \begin{cases} \min(a, b) & \text{if } a, b > 0, \\ \max(a, b) & \text{if } a, b < 0, \\ 0 & \text{otherwise.} \end{cases} \quad (4.8)$$

The effect of the slope limiting on the robustness of this approach remains to be investigated. For example, other, less diffusive limiters (as described, e.g., in [33]) could be used in place of the minmod limiter. A novel development would be the use of intrinsically vector-based limiters; see the recent work of Luttwak and Falcovitz [35] or Maire et al. [36] for possible approaches.

Update

In this section, an expression for the value of $\Delta\mathbf{X}$ on the edge of remapped elements is obtained. The generalized Stokes' theorem for tensors [22] is written:

$$\oint_{\partial S} \mathbf{T} \mathbf{t} ds = \int_S (\text{curl } \mathbf{T})^T \mathbf{n} dA . \quad (4.9)$$

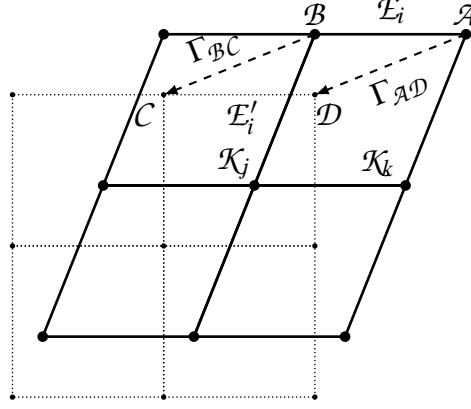


Figure 4.2. Depiction of the integration path for the remap scheme in 2D: the solid lines represent a subset of the Lagrangian-updated (deformed) mesh and the dotted lines represent part of the target (remap) mesh. The inverse deformation gradient is to be remapped from the deformed mesh to the target mesh. The union of the segments $\mathcal{E}_i$, Γ_{BC} , $\mathcal{E}'_i$, and Γ_{AD} comprises a closed curve, associated with nodes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, around which the integral in (4.10) is to be evaluated.

Applying this relation to a sufficiently smooth inverse deformation gradient, which by (3.15) has zero curl, this equality implies

$$\oint_{\partial S} \mathbf{F}^{-1} \mathbf{t} ds = \mathbf{0}. \quad (4.10)$$

That is, the integral of the $\mathbf{F}^{-1}$ about any closed path vanishes identically. The application of this relation is the key concept behind the constrained transport algorithm for remap. The path ∂S should be specified so that (i) all quantities are well-defined along each element of the path, (ii) there is some simplification of the integral along certain elements of the path, so that (iii) an updated value of the integral is obtained on an element of the path of interest. This closed path ∂S is chosen to contain the edge $\mathcal{E}_i$ in the updated configuration (i.e., after the Lagrangian update), as well as the image of that edge under the remap, which we denote $\mathcal{E}'_i$. The nodes of these edges are naturally associated through the position-offset vectors that connect them. In the terminology of computational advection, the post-remap nodes are the upwind nodes of the edge under the action of the remap.

Figure 4.2 gives a notional 2D depiction, with the updated mesh as the solid lines and the target (remap) mesh as dotted lines. In this figure, the closed curve of interest, ∂S , is comprised of the edges of the quadrilateral delimited by nodes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$. Write this relation as

$$\partial S := \overrightarrow{\mathcal{A}\mathcal{B}} \cup \overrightarrow{\mathcal{B}\mathcal{C}} \cup \overrightarrow{\mathcal{C}\mathcal{D}} \cup \overrightarrow{\mathcal{D}\mathcal{A}}, \quad (4.11)$$

where the $\overrightarrow{PQ}$ denotes segment directed from node P to node Q . Relative to the previous notation and modulo a sign, $\overrightarrow{\mathcal{A}\mathcal{B}}$ is the edge $\mathcal{E}_i$, $\overrightarrow{\mathcal{C}\mathcal{D}}$ is the edge $\mathcal{E}'_i$, and $\overrightarrow{\mathcal{B}\mathcal{C}}$ and $\overrightarrow{\mathcal{D}\mathcal{A}}$ are the “connecting”

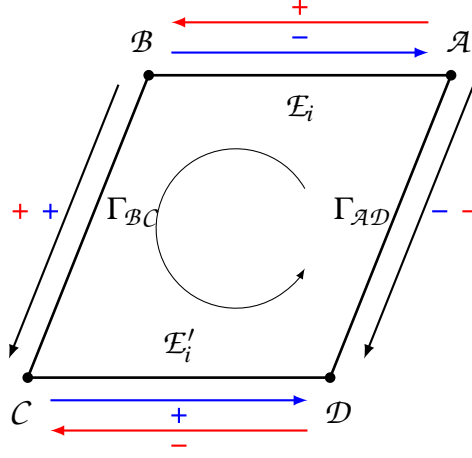


Figure 4.3. Depiction of the orientation convention used for the closed loop of the remap scheme. The union of the edges $\mathcal{E}_i$, $\Gamma_{\mathcal{BC}}$, $\mathcal{E}'_i$, and $\Gamma_{\mathcal{AD}}$ is a closed curve, associated with nodes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$. The integral in (4.13) is to be evaluated along this closed path, in the sense of the circular arrow. The edges $\mathcal{E}_i$ and $\mathcal{E}'_i$ are consistently oriented, e.g., if $\mathcal{E}_i$ is directed from $\mathcal{A}$ to $\mathcal{B}$, then $\mathcal{E}'_i$ is directed from $\mathcal{D}$ to $\mathcal{C}$. The two choices of orientation for these edges are indicated with the red and blue arrows. The orientation on the fictitious edges $\Gamma_{\mathcal{BC}}$ and $\Gamma_{\mathcal{AD}}$ is assigned to be in the direction from the node of the updated edge (i.e., $\mathcal{A}$ or $\mathcal{B}$) to the corresponding upwind node (i.e., $\mathcal{D}$ or $\mathcal{C}$). The “+” and “−” signs, correlated in color with the indicated orientations, correspond to the sign of the addends in (4.13).

segments $\Gamma_{\mathcal{BC}}$ and $\Gamma_{\mathcal{AD}}$, respectively. The orientation of the segments plays a crucial role in the correct evaluation in this analysis.

From (4.10) and (4.11), it is seen that

$$\mathbf{0} = \oint_{\partial S} \mathbf{F}^{-1} \mathbf{t} ds = \int_{\overrightarrow{\mathcal{AB}}} \mathbf{F}^{-1} \mathbf{t} ds + \int_{\overrightarrow{\mathcal{BC}}} \mathbf{F}^{-1} \mathbf{t} ds + \int_{\overrightarrow{\mathcal{CD}}} \mathbf{F}^{-1} \mathbf{t} ds + \int_{\overrightarrow{\mathcal{DA}}} \mathbf{F}^{-1} \mathbf{t} ds. \quad (4.12)$$

In this work, the orientation of the edges is chosen as indicated in Fig. 4.3. Depending on the choice of orientation of the edges $\mathcal{E}_i$ and $\mathcal{E}'_i$, as described in the caption of that figure, the expression in (4.12) evaluates to:

$$\mathbf{0} = \oint_{\partial S} \mathbf{F}^{-1} \mathbf{t} ds = \mp \Delta \mathbf{X}_{\mathcal{E}_i} + \Delta \mathbf{X}_{\Gamma_{\mathcal{BC}}} \pm \Delta \mathbf{X}_{\mathcal{E}'_i} - \Delta \mathbf{X}_{\Gamma_{\mathcal{AD}}}, \quad (4.13)$$

$$\text{where} \quad \Delta \mathbf{X}_C := \int_C \mathbf{F}^{-1} \mathbf{t} ds \quad (4.14)$$

along segment C .

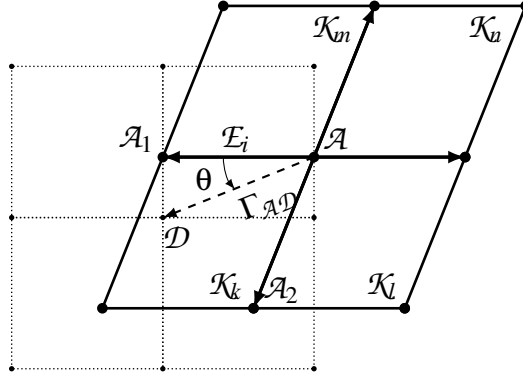


Figure 4.4. Determination of the upwind element for node $\mathcal{A}$ (c.f. Fig. 4.2): the inner product is positive for the directed vector along edge $\Gamma_{\mathcal{A}\mathcal{D}}$ with the vectors along the two directed edges connected to $\mathcal{A}$ in element $\mathcal{K}_k$ (e.g., $|\theta| < \pi/2$). At least one of the corresponding inner products is negative for the other elements associated with node $\mathcal{A}$ (i.e., $\mathcal{K}_l$, $\mathcal{K}_m$, $\mathcal{K}_u$). Therefore, element $\mathcal{K}_k$ is the upwind element for node $\mathcal{A}$.

For a given edge $\mathcal{E}_i$, one determines the corresponding upwind edge $\mathcal{E}'_i$ as follows. The 2D version of this process is illustrated in Fig. 4.4. Consider node $\mathcal{A}$ associated with edge $\mathcal{E}_i$. Satisfaction of the CFL condition guarantees that the point $\mathbf{x}^{\mathcal{D}}$ is located inside one of the elements associated with the node $\mathcal{A}$. Consider an element $\mathcal{K}_k$ connected to node $\mathcal{A}$; the goal is to determine if $\mathbf{x}^{\mathcal{D}} \in \mathcal{K}_k$. Define the tangent vectors associated with the edges of $\mathcal{K}_k$ connected to node $\mathcal{A}$ as

$$\mathbf{V}_\ell := \mathbf{x}^{\mathcal{A}_\ell} - \mathbf{x}^{\mathcal{A}} \quad \text{for } \ell = 1, 2 \text{ in 2D or } \ell = 1, 2, 3 \text{ in 3D}, \quad (4.15)$$

where the node $\mathcal{A}_\ell \neq \mathcal{A}$ shares an edge on element $\mathcal{K}_k$ with node $\mathcal{A}$. If, for all edges connected to node $\mathcal{A}$, it is true that

$$\left(\mathbf{x}^{\mathcal{D}} - \mathbf{x}^{\mathcal{A}} \right) \cdot \mathbf{V}_\ell \geq 0, \quad (4.16)$$

then $\mathbf{x}^{\mathcal{D}} \in \mathcal{K}_k$, i.e., element $\mathcal{K}_k$ is the upwind element of node $\mathcal{A}$.

In the configuration of Fig. 4.2, the value of $\Delta \mathbf{X}$ is *known* along edge $\mathcal{E}_i$ and is *desired* along edge $\mathcal{E}'_i$ (which may span more than one element). One must approximate the values along paths $\Gamma_{\mathcal{B}\mathcal{C}}$ and $\Gamma_{\mathcal{A}\mathcal{D}}$. By construction, each of the segments $\Gamma_{\mathcal{B}\mathcal{C}}$ and $\Gamma_{\mathcal{A}\mathcal{D}}$ is completely contained in a single element of the deformed mesh. For example, in Fig. 4.2, $\Gamma_{\mathcal{A}\mathcal{D}} \subset \mathcal{K}_k$ and $\Gamma_{\mathcal{B}\mathcal{C}} \subset \mathcal{K}_j$. On each of these elements, there is a codified representation of $\mathbf{F}^{-1}$, given by (3.10).

Consider the integral along segment $\Gamma_{\mathcal{A}\mathcal{D}}$. Along this path, the integrand can be rewritten in terms of the natural coordinates of the associated parent element:

$$\int_{\Gamma_{\mathcal{A}\mathcal{D}}} \mathbf{F}^{-1} \mathbf{t} ds = \int_{\Gamma_{\mathcal{A}\mathcal{D}}} \mathbf{F}^{-1} \frac{\partial \mathbf{x}}{\partial \xi_\alpha} d\xi_\alpha, \quad (4.17)$$

where the summation convention on α applies. Perhaps the simplest approximation of the integrals is given by a single point integration rule evaluated at the node-centered mid-point of the integration domain $\Gamma_{\mathcal{AD}}$. There are two obvious ways to evaluate this mid-point. The first is to identify $\boldsymbol{\xi}^{\mathcal{A}} = \mathcal{F}^{-1}(\mathbf{x}^{\mathcal{A}})$ and $\boldsymbol{\xi}^{\mathcal{D}} = \mathcal{F}^{-1}(\mathbf{x}^{\mathcal{D}})$ as the natural coordinates of nodes $\mathcal{A}$ and $\mathcal{D}$, respectively, in element $\mathcal{K}_k$, and then to associate the mid-point with the mean value of these natural coordinate vectors:

$$\hat{\boldsymbol{\xi}} := \frac{1}{2} (\boldsymbol{\xi}^{\mathcal{A}} + \boldsymbol{\xi}^{\mathcal{D}}) . \quad (4.18)$$

The second is to associate the mid-point with natural coordinate of the physical-space midpoint:

$$\hat{\boldsymbol{\xi}} := \mathcal{F}^{-1} \left(\frac{1}{2} (\mathbf{x}_{\mathcal{A}} + \mathbf{x}_{\mathcal{D}}) \right) . \quad (4.19)$$

However one evaluates the mid-point, the associated approximation of the integral in (4.17) is given by

$$\int_{\Gamma_{\mathcal{AD}}} \mathbf{F}^{-1} \mathbf{t} ds \approx \mathbf{F}^{-1}(\hat{\boldsymbol{\xi}}) \frac{\partial \mathbf{x}}{\partial \xi_{\alpha}}(\hat{\boldsymbol{\xi}}) \int_{\Gamma_{\mathcal{AD}}} d\xi_{\alpha} = \mathbf{F}^{-1}(\hat{\boldsymbol{\xi}}) \frac{\partial \mathbf{x}}{\partial \xi_{\alpha}}(\hat{\boldsymbol{\xi}}) \delta \xi_{\alpha} , \quad (4.20)$$

where $\delta \boldsymbol{\xi} := \boldsymbol{\xi}^{\mathcal{D}} - \boldsymbol{\xi}^{\mathcal{A}}$. An alternative approximation of the integral in (4.17), with the same order of accuracy, is

$$\int_{\Gamma_{\mathcal{AD}}} \mathbf{F}^{-1} \mathbf{t} ds \approx \mathbf{F}^{-1}(\hat{\boldsymbol{\xi}}) \int_{\Gamma_{\mathcal{AD}}} \mathbf{t} ds = \mathbf{F}^{-1}(\hat{\boldsymbol{\xi}}) (\mathbf{x}^{\mathcal{D}} - \mathbf{x}^{\mathcal{A}}) . \quad (4.21)$$

From (3.10) and (4.1), there is a representation of $\mathbf{F}^{-1}$ on this element. One evaluates that expression at $\hat{\boldsymbol{\xi}}$ as

$$\mathbf{F}^{-1}(\hat{\boldsymbol{\xi}}) = \sum_{e=1}^{\dim \vec{\mathcal{E}}} \Delta \mathbf{X}_e(\hat{\boldsymbol{\xi}}) \otimes \vec{\mathbf{W}}_e(\hat{\boldsymbol{\xi}}) \quad (4.22)$$

and substitutes this result into either (4.20) or (4.21) to obtain a numerical approximation of the integral of the inverse deformation gradient along segment $\Gamma_{\mathcal{AD}}$. In (4.22) the sum is over all edges; recall, however, that most edge basis functions vanish identically outside of the element in question.

Curl-preserving remap implementation

This section outlines the sequence of steps by which one can implement the constrained-transport edge-based remap described above. The proposed approach is mindful of improved computational efficiency by reducing the effort associated with searching for nodes in the mesh. Each of the following steps corresponds to a unique loop over the nodes or edges.

- i) **Project $\mathbf{F}^{-1}$:** The first step is to project the tensor field $\mathbf{F}^{-1}$, given within elements, to the nodes. For a patch recovery, this requires looping over all the nodes $\mathcal{N}$ in the mesh (depending on the details of the projection, this procedure may involve a loop over elements). For each node, a projected (i.e., interpolated) value of $\mathbf{F}^{-1}$ is computed using the values of $\mathbf{F}^{-1}$ in

the set of elements connected to (surrounding) the node. There many different ways to do this [49, 48, 1]; details are omitted.

- ii) **Reconstruct slopes:** This step involves computing the slopes $\mathbf{S}_e$ on all the edges $\mathcal{E}$ of the mesh. The details of this procedure are in Chapter 4. This step requires looping over all edges $\mathcal{E}$ of the mesh.
- iii) **Determine upwind nodes:** The upwind position of each node in the mesh must now be located. This process requires looping over the nodes $\mathcal{N}$ of the mesh. For a given node $\mathcal{A} \in \mathcal{N}$, the upwind position of the node is calculated using the algorithm outlined in subsection 4. Denote the upwind location of $\mathcal{A}$ by the (fictitious¹) node $\mathcal{D}$; this defines a (fictitious) edge $\Gamma_{\mathcal{AD}}$. On the edge $\Gamma_{\mathcal{AD}}$, compute

$$\Delta \mathbf{X}_{\mathcal{AD}} := \int_{\Gamma_{\mathcal{AD}}} \mathbf{F}^{-1} \mathbf{t} ds, \quad (4.23)$$

and store this value, associating it with node $\mathcal{A}$. The computation of this integral is described in subsection 4.

- iv) **Remap $\Delta \mathbf{X}_e$:** The last step is to compute updated (remapped) values of $\Delta \mathbf{X}_e$ on each edge. This is a final loop over all edges $\mathcal{E}$ of the mesh. For each edge $\mathcal{E}_i$ equation (4.13) is solved for the value of $\Delta \mathbf{X}$ on $\mathcal{E}'_i$,

$$\Delta \mathbf{X}_{\mathcal{E}'_i} := \int_{\mathcal{E}'_i} \mathbf{F}^{-1} \mathbf{t} ds. \quad (4.24)$$

The data needed to solve this equation have been computed and stored in the immediately preceding step (iii).

¹This node and edge are identified as “fictitious” insofar as there are no new data structures associated with them: information computed in this step is associated with node $\mathcal{A}$.

Chapter 5

Numerical Examples of Curl-Preserving Remap

In this section, test problems with exact solutions are evaluated with the curl-preserving remap method. We consider only 2-D deformations; in particular, we restrict our attention to deformations associated with velocity fields that depend solely on angle about a central, fixed location. Those exact solutions are used to compute convergence rates for the implemented algorithm.

Convergence rates for problems with exact solutions

If we assume that the exact deformation is known using the methods of Appendix B, then the associated convergence rate for the computed solutions can be calculated. In particular, the convergence rates of the eigenvalues of the stretch tensor $\mathbf{V}$ and rotation tensor $\mathbf{R}$ in the decomposition of the deformation gradient as $\mathbf{F} = \mathbf{V}\mathbf{R}$ can be evaluated. Convergence rates are computed using the three standard norms typically used to quantify the error, L_1 , L_2 , and L_∞ , where for the quantity $u_{ij} := u(x_i, y_j)$ given on the set $i = 1, \dots, n_i$, $j = 1, \dots, n_j$,

$$\|u\|_{L_1} = \sum_{i,j} |u_{ij}| \mathcal{V}_{ij}, \quad \|u\|_{L_2} = \left(\sum_{i,j} u_{ij}^2 \mathcal{V}_{ij} \right)^{1/2}, \quad \|u\|_{L_\infty} = \max_{i,j} |u_{ij}|, \quad (5.1)$$

where $\mathcal{V}_{ij} = \text{meas}(\mathcal{K}_{ij})$. These norms are evaluated for the difference between the exact and computed values of a given matrix, i.e., $u = M_{ab}^{\text{exact}} - M_{ab}^{\text{comp}}$, where, say, M_{ab} is the (a, b) component of matrix $\mathbf{M} = \mathbf{V}, \mathbf{R}$. With these values, the convergence rate σ for the results on the the coarse (c) and fine (f) grids, with the respective number of elements initially in each linear direction denoted as n_c and n_f , is evaluated as

$$\sigma = \frac{\log(\|u_c\|/\|u_f\|)}{\log(n_f/n_c)}. \quad (5.2)$$

Exponential vortical flow

This problem contains a smoothly varying angular flow field with an exponential radial decay. The equation for the magnitude of the azimuthal flow velocity is

$$v_\theta = \frac{\Gamma}{2\pi} r \exp\left(-\frac{br^2}{2}\right). \quad (5.3)$$

The deformation gradient associated with this angular velocity field can be calculated explicitly. As a specific example, let $\Gamma = 2\pi$ and $b = 1$, choose the two-dimensional computational domain to be the square $[-5, 5] \times [-5, 5]$, and let the time interval for the simulation be $[0, 5]$. The domain is discretized with a mesh of $n_{\text{el}} \times n_{\text{el}}$ initially square finite elements, where $n_{\text{el}} = \{16, 32, 64, 128, 256\}$. The problem was run using both pure Lagrangian (i.e., no remap) and zero-curl Eulerian (i.e., Lagrange plus zero-curl remap every timestep) methodologies. Figure 5.1 contains plots of $\|\mathbf{V}\|$, on the five different mesh resolutions, at the final time $t = 5$.

Table 5.1 contains the convergence rates for the stretch tensor for both the Lagrangian and zero-curl Eulerian methodologies. Table 5.2 documents the corresponding values for the rotation tensor. Since the velocity field given in (5.3) is smooth, the computed convergence rate should be approximately equal to the designed rate of the algorithm. Here, the underlying hydrocode, ALEGRA, has second-order algorithms for the fundamental conservation laws, so we expect second-order convergence under refinement of the spatial mesh.¹ The deformation gradient for this problem is also smooth, so the convergence rate in any error norm is representative of solution behavior. The results of Tables 5.1 and 5.2 show that the remapped values (i.e., the results of the zero-curl Eulerian calculations) of both the stretch and rotation matrices converge at almost the same rate as their counterparts in the Lagrangian (i.e., unremapped) case.

¹The timestep is related to the spatial cell size through the CFL condition, so spatial refinement implies temporal refinement, as well.

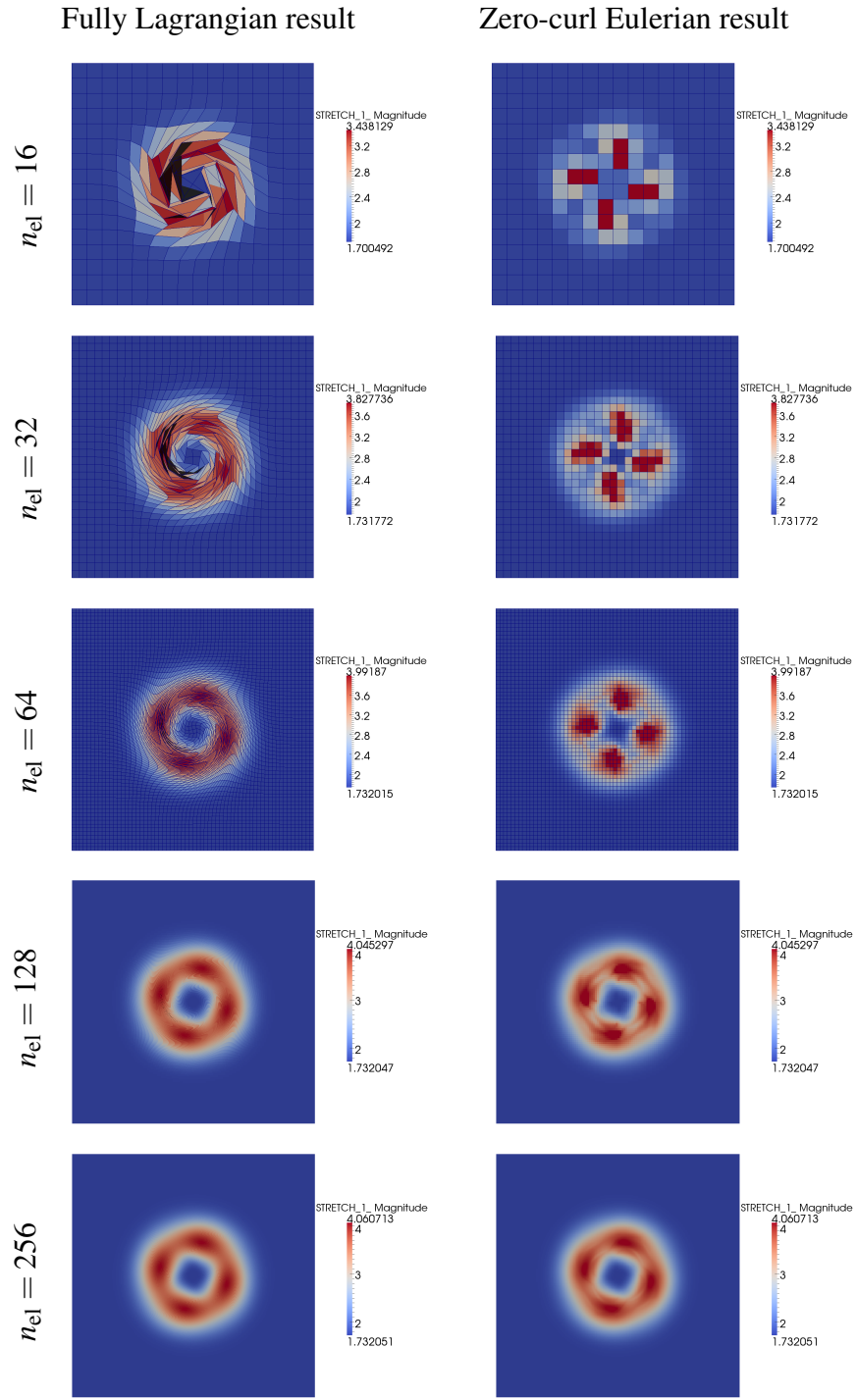


Figure 5.1. Color-coded plots of the magnitude of the stretch tensor, $\|\mathbf{V}\|$, at simulation time $t = 5$ for the fully Lagrangian (no remap) calculation (left column), and the Eulerian (Lagrangian plus zero-curl remap every timestep) calculation (right column) of the exponential vortical flow problem. The rows correspond to the mesh resolution, from the coarsest (16×16) on top to the finest (256×256) on the bottom. In the top three rows, the Lagrangian mesh has become highly deformed, while the Eulerian mesh remains unchanged; the mesh is not shown in the bottom two rows.

Table 5.1. Convergence rates for components of the stretch matrix $\mathbf{V}$ (in $\mathbf{F} = \mathbf{VR}$) for the exponential vortical flow problem, calculated with the norms indicated in the column headings, at $t = 5$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-5, 5] \times [-5, 5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

Lagrangian (no remap)				Eulerian (zero-curl remap)				
n_{el}		L_1	L_2	L_∞	L_1	L_2	L_∞	
V_{xx}	16	$\rangle$						
	32		1.80858	1.99496	1.81303	1.36353	1.80395	1.38775
	64		1.94630	1.93120	1.95788	1.41513	1.68420	1.50672
	128		1.98764	1.98053	1.94142	1.78994	1.81272	2.04941
	256		1.99688	1.99496	1.98349	1.96797	1.95608	1.91443
n_{el}		L_1	L_2	L_∞	L_1	L_2	L_∞	
V_{xy}	16	$\rangle$						
	32		1.84398	1.76823	1.48767	1.30707	1.42913	1.55579
	64		1.94807	1.93106	1.83133	1.29262	1.14898	0.828250
	128		1.98085	1.98257	1.94669	1.74493	1.62300	1.17611
	256		1.99663	1.99562	1.99574	2.01182	2.00453	1.83988
n_{el}		L_1	L_2	L_∞	L_1	L_2	L_∞	
V_{yy}	16	$\rangle$						
	32		1.80858	1.79393	1.81303	1.36353	1.80395	1.38775
	64		1.94630	1.93120	1.95788	1.41513	1.68420	1.50672
	128		1.98764	1.98053	1.94142	1.78994	1.81272	2.04941
	256		1.99688	1.99496	1.98349	1.96797	1.95608	1.91443

Table 5.2. Convergence rates for components of the rotation matrix $\mathbf{R}$ (in $\mathbf{F} = \mathbf{VR}$) for the exponential vortex problem, calculated with the norms indicated in the column headings, at $t = 5$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-5, 5] \times [-5, 5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

		Lagrangian (no remap)			Eulerian (zero-curl remap)		
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{xx}	16	>					
	32						
	64						
	128						
	256						
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{xy}	16	>					
	32						
	64						
	128						
	256						
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{yx}	16	>					
	32						
	64						
	128						
	256						
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{yy}	16	>					
	32						
	64						
	128						
	256						
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞

ABC vortical flow

This problem contains a cylindrical block that rotates at constant angular velocity, surrounded by an irrotational circular flow region with constant circulation, which, in turn, is surrounded by a flow with negative vorticity until the flow decreases to zero velocity at finite radius. The flow field has discontinuous velocity gradients. The equations for the magnitude of the azimuthal flow velocity are:

$$v_\theta = \begin{cases} \omega_0 r, & 0 < r < a, \\ \frac{\omega_0 a^2}{r}, & a < r < b, \\ \frac{\omega_0 a^2}{r} \left(\frac{c^2 - r^2}{c^2 - b^2} \right), & b < r < c, \\ 0, & c < r. \end{cases} \quad (5.4)$$

This problem provides a region of pure local rotation in the immediate vicinity of the origin, followed by, in order of increasing radial distance from the origin, a region of pure local stretch, a mixed region with both stretch and rotation, and, finally, a quiescent region. The deformation gradient associated with this angular velocity field can be calculated explicitly.

As a specific numerical example, let $\omega_0 = 2\pi$, $a = 0.1$, $b = 0.3$ and $c = 0.4$. The two-dimensional computational domain is the square $[-0.5, 0.5] \times [-0.5, 0.5]$, and the time interval for the simulation is $[0, 0.5]$. The domain is discretized with a mesh of $n_{el} \times n_{el}$ initially square finite elements, where $n_{el} = \{16, 32, 64, 128, 256\}$. The problem was run using both pure Lagrangian (i.e., no remap) and Eulerian (i.e., Lagrange plus zero-curl remap every timestep) methodologies. Figure 5.2 shows plots of $\|\mathbf{V}\|$, on the five different mesh resolutions, at the final time of 0.5.

Table 5.3 contains the convergence rates for the stretch tensor for both the Lagrangian and Eulerian methodologies. Table 5.4 documents the corresponding values for the rotation tensor. Unlike the previous example, for this problem the velocity field given in (5.4) is continuous in space at the boundaries $r = a$ and $r = b$, but the spatial gradient of this field is discontinuous at these locations. Due to this discontinuity, one should expect no better than first-order convergence under refinement of the spatial mesh. As the deformation gradient for this problem is discontinuous, the L_1 -norm convergence rates are perhaps the best indicator of solution behavior; that the L_∞ -norm convergence rates are indeterminate is not unexpected. The remapped values (i.e., the results of the Eulerian calculations) of both the stretch and rotation matrices converge at a slightly lower but almost the same rate as their counterparts in the Lagrangian (i.e., unremapped) case, despite the intrinsically discontinuous nature of the solution.

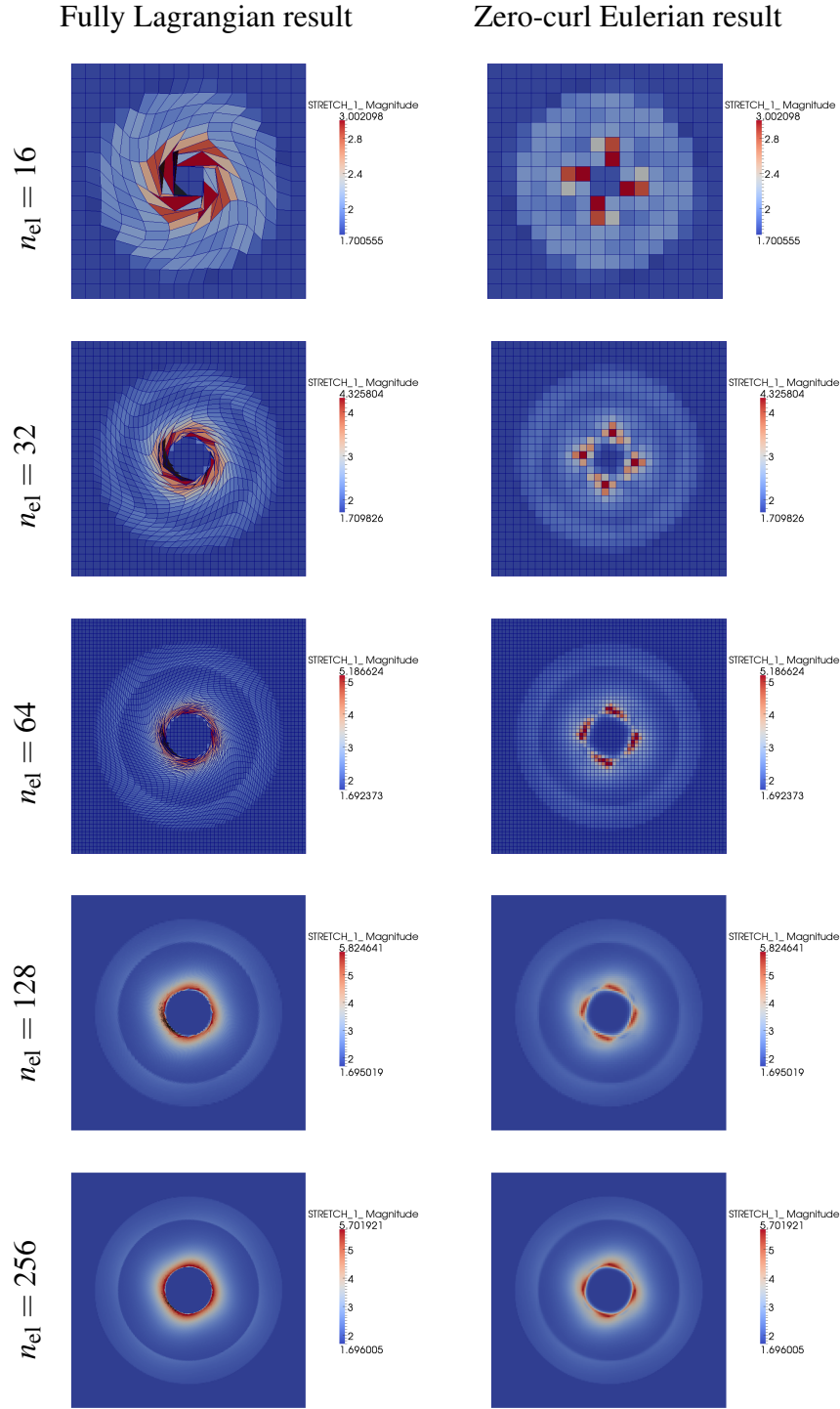


Figure 5.2. Color-coded plots of the magnitude of the stretch tensor, $\|\mathbf{V}\|$, at simulation time $t = 0.5$ for the fully Lagrangian (no remap) calculation (left column), and the Eulerian (Lagrangian plus zero-curl remap every timestep) calculation (right column) of the ABC vortical flow problem. The rows correspond to the mesh resolution, from the coarsest (16×16) on top to the finest (256×256) on the bottom. In the top three rows, the Lagrangian mesh has become highly deformed, while the Eulerian mesh remains unchanged; the mesh is not shown in the bottom two rows.

Table 5.3. Convergence rates for components of the stretch matrix $\mathbf{V}$ (in $\mathbf{F} = \mathbf{VR}$) for the ABC vortical flow problem, calculated with the norms indicated in the column headings, at $t = 0.5$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-0.5, 0.5] \times [-0.5, 0.5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

		Lagrangian (no remap)			Eulerian (zero-curl remap)		
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
V_{xx}	16	$\rangle$	0.973322	0.380938	-1.01659	0.438729	-0.212137
	32						
	64						
	128						
	256						
V_{xy}	16	$\rangle$	1.16954	1.07195	0.494820	0.686648	0.311300
	32						
	64						
	128						
	256						
V_{yy}	16	$\rangle$	0.973322	0.380938	-1.01659	0.437219	-0.208572
	32						
	64						
	128						
	256						

Table 5.4. Convergence rates for components of the rotation matrix $\mathbf{R}$ (in $\mathbf{F} = \mathbf{VR}$) for the ABC vortical flow problem, calculated with the norms indicated in the column headings, at $t = 0.5$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-0.5, 0.5] \times [-0.5, 0.5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

Lagrangian (no remap)				Eulerian (zero-curl remap)				
n_{el}		L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{xx}	16	$\rangle$	0.762453	0.405455	−0.132523	1.88348	2.11960	1.36358
	32							
	64							
	128							
	256							
n_{el}		L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{xy}	16	$\rangle$	0.914064	0.360033	−0.0082190	1.37845	1.11204	0.328109
	32							
	64							
	128							
	256							
n_{el}		L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{yx}	16	$\rangle$	0.914064	0.360033	−0.0082190	1.37845	1.11204	0.328109
	32							
	64							
	128							
	256							
n_{el}		L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{yy}	16	$\rangle$	0.762453	0.405455	−0.132523	1.88348	2.11960	1.36358
	32							
	64							
	128							
	256							

This page intentionally left blank.

Chapter 6

Enforcing Extrema-Preservation on the Remap

The results of the previous chapter demonstrate that the zero-curl constraint-preserving remap method is convergent on the given test problems at the chosen times. Running these test problems to later times revealed an unanticipated aspect of the computed results: comparison of the new remap results to the exact solution and to a pure Lagrangian simulation showed that the new algorithm eventually provided inaccurate results, with the computed stretches becoming unbounded. We hypothesize that the absence of any explicit limiting in the calculation of the remapped solution allows spurious extrema, as evinced in the determinant of the deformation gradient.

New and expected extremal stretches¹ in the Lagrangian step are a manifestation of the material deformation and lead to variations in the determinant of the IDG. For accurate ALE simulations, it is imperative that the remap step not artificially amplify the legitimate deformation, so that the remap is an extremum-preserving operation. To achieve this goal, we have developed an optimization-based approach that enforces the zero-curl constraint *and* bounds the value of the determinant of the Jacobian in each remapped element. This chapter contains a description of that augmented method.

Constrained optimization overview

The remap method previously described provides a solution that obeys the zero-curl condition, but does not enforce any constraints on the determinant of the IDG. Using the remapped solution as the initial guess, a black-box optimization approach is used that retains the zero-curl condition and imposes the restriction that the determinant of the IDG not decrease. These two requirements are formulated as constraints in a numerical optimization scheme.

The general formulation of optimization techniques used is described in detail in the work of Ridzal, Young, Aguiló, and Heinkenschloss [39, 47]. Stated abstractly, the overarching goal of this optimization problem is to calculate a minimum of an objective function f of the state u that is

¹Recall, the material stretches are the eigenvalues of the symmetric, positive definite tensor in the polar decomposition of the deformation gradient.

subject to both linear equality constraints g and nonlinear inequality constraints h :

$$\min_u f(u) \quad \text{subject to} \quad g(u) = 0 \quad \text{and} \quad h(u) > 0. \quad (6.1)$$

In the current application:

- the state is the array consisting of the increments of directed material coordinates on all edges of the mesh;
- the linear equality constraint represents the vanishing curl condition calculated on the faces of each element in the mesh; and
- the nonlinear inequality constraint limits the value of the determinant in each element of the mesh to be greater than the minimum determinant value, over the *entire* mesh, associated with the Lagrangian step but before the remap.

Notionally, the j th element of the inequality constraint array can be written:

$$h_j(u) := \det_j(u) - \varepsilon > 0 \quad \text{with} \quad \varepsilon := \min_{k \in \mathcal{K}} \{\det_k(u^L)\}, \quad (6.2)$$

where $\det_j(u)$ is the determinant of the IDG on element j , the minimum is taken over all elements in the mesh, and u^L represents u after the Lagrangian step.

The objective function to be minimized is the square of the L_2 norm of the difference between the state u and the zero-curl remap state $\hat{u}$ (i.e., the solution obtained with the zero-curl remap of the previous chapters):

$$f(u) := \frac{1}{2} \sum_i (u_i - \hat{u}_i)^2. \quad (6.3)$$

The methods used require that the inequality constraints be affine; however, the determinant constraint does not satisfy this requirement. Therefore, the optimization problem in (6.1) is rewritten in the following form, which introduces an artificial (“slack”) variable s :

$$\min_u f(u) \quad \text{subject to} \quad g(u) = 0, \quad h(u) - s = 0 \quad \text{and} \quad s - \varepsilon > 0. \quad (6.4)$$

Thus, the equality constraint is expanded to include the definition of the slack variable, and the inequality constraint becomes the trivially affine relation, $s > 0$. We rewrite this system as:

$$\min_z \mathcal{F}(z) \quad \text{subject to} \quad \mathcal{G}(z) = 0 \quad \text{and} \quad \mathcal{H}(z) > 0, \quad \text{where} \quad (6.5a)$$

$$z := [u, s]^T, \quad \mathcal{F}(z) := f(u), \quad \mathcal{G}(z) := [g(u), h(u) - s]^T, \quad \mathcal{H}(z) := s - \varepsilon. \quad (6.5b)$$

Constrained optimization implementation

We now consider the dimensionality of the terms in these equations. First, define the parameters:

$$n_{\text{el}} = \text{the number of elements in the mesh}, \quad (6.6a)$$

$$n_{\text{ed}} = \text{the number of edges in the mesh}, \quad (6.6b)$$

$$n_{\text{fa}} = \text{the number of faces in the mesh}, \quad (6.6c)$$

where, in two dimensions, $n_{\text{fa}} = n_{\text{ed}}$. The fundamental unknown, u , represents the components of the increment in displacement along each edge; in two dimensions, there are two such components per edge, while in three dimensions, there are three. Thus:

$$\dim\{u\} = D n_{\text{ed}}, \quad (6.7)$$

where D is the dimensionality of the physical geometry. The dimensionality of the slack variable, s , equals the dimensionality of the original inequality constraints, h ; there is one such constraint per element of the mesh, so that

$$\dim\{s\} = \dim\{h\} = n_{\text{el}}. \quad (6.8)$$

Thus, the dimensionality of the argument, z , of the extended objective function, $\mathcal{F}$, is given by:

$$\dim\{z\} = D n_{\text{ed}} + n_{\text{el}}. \quad (6.9)$$

The dimensionality of g equals the number of the linear constraints, which consist of the number of elements in the curl constraint on each face times the number of faces of all mesh elements:

$$\dim\{g\} = D n_{\text{fa}}. \quad (6.10)$$

The dimensionality of the terms in the effective system of equations in (6.5a,6.5b) follows immediately:

$$\dim\{\mathcal{F}\} = \dim\{f\} = 1, \quad (6.11a)$$

$$\dim\{\mathcal{G}\} = \dim\{g\} + \dim\{h\} = D n_{\text{fa}} + n_{\text{el}}, \quad (6.11b)$$

$$\dim\{\mathcal{H}\} = n_{\text{el}}. \quad (6.11c)$$

In terms of the dimensionality of their domains and codomains, these functions map between the following spaces:

$$\mathcal{F} : \mathbb{R}^{D n_{\text{ed}} + n_{\text{el}}} \rightarrow \mathbb{R}, \quad (6.12a)$$

$$\mathcal{G} : \mathbb{R}^{D n_{\text{ed}} + n_{\text{el}}} \rightarrow \mathbb{R}^{D n_{\text{fa}} + n_{\text{el}}}, \quad (6.12b)$$

$$\mathcal{H} : \mathbb{R}^{D n_{\text{ed}} + n_{\text{el}}} \rightarrow \mathbb{R}^{n_{\text{el}}}. \quad (6.12c)$$

In the numerical optimization routines, expressions for these terms and their derivatives are required. The first derivatives of the functions in the extended system (and their dimensionality) are evaluated as:

$$\mathcal{F}' = \underbrace{1}_{Dn_{\text{ed}}} \left\{ \underbrace{\left[\frac{\partial f}{\partial u} \right]}_{n_{\text{el}}} \right\} \quad (6.13a)$$

$$\mathcal{G}' = \underbrace{Dn_{\text{ed}}}_{n_{\text{el}}} \left\{ \underbrace{\left[\begin{array}{c|c} \frac{\partial g}{\partial u} & \mathbf{0} \\ \hline \frac{\partial h}{\partial u} & -\mathbf{I} \end{array} \right]}_{n_{\text{el}}} \right\} \quad (6.13b)$$

$$\mathcal{H}' = \underbrace{n_{\text{el}}}_{Dn_{\text{ed}}} \left\{ \underbrace{\left[\begin{array}{c|c} \mathbf{0} & \mathbf{I} \end{array} \right]}_{n_{\text{el}}} \right\} \quad (6.13c)$$

The first derivatives in these terms can be directly evaluated. For $\mathcal{F}'$, from the definition of f in (6.3) it is seen that

$$\frac{\partial f}{\partial u_i} = u_i - \hat{u}_i. \quad (6.14)$$

For $\mathcal{G}'$, we focus on $\partial g/\partial u$ and $\partial h/\partial u$. For the former, we recall that $g(u) = 0$ represents the curl constraint, which is linear in the increments in material coordinates, $\Delta \mathbf{X}$, as in (4.13). Thus, the majority of elements $\partial g_i/\partial u_j$ are zero, with the few non-zero elements (depending on the ordering of the array u) having the value ± 1 , according to the orientation chosen (as represented in (4.13)). Recalling that $h(u)$ represents the determinant of the IDG on the elements, for $\partial h/\partial u$ we require an expression for the derivative of a determinant with respect to its entries. To obtain such a representation, we use Jacobi's formula: for an invertible matrix $\mathbf{A}(\zeta)$ that is differentiable in the argument ζ , the derivative of the determinant of the matrix with respect to ζ is given by

$$\frac{d(\det \mathbf{A})}{d\zeta} = \text{tr} \left(\mathbf{A}^{\text{A}} \frac{d\mathbf{A}}{d\zeta} \right) = \left(\mathbf{A}^{\text{A}} \right)^{\text{T}} : \frac{d\mathbf{A}}{d\zeta}, \quad (6.15)$$

where $\mathbf{A} : \mathbf{B} := A_{ij}B_{ij}$ for 2×2 matrices, and $\mathbf{A}^{\text{A}}$ is the adjugate of $\mathbf{A}$ (see Appendix C for a brief review of the matrix adjugate). For the case at hand, the matrix is the IDG and the independent variable is the increment in material coordinates along the e th edge:

$$\frac{d(\det \mathbf{F}^{-1})}{d(\Delta \mathbf{X}_e)} = \left(\mathbf{F}^{-\text{T}} \right)^{\text{A}} : \frac{d\mathbf{F}^{-1}}{d(\Delta \mathbf{X}_e)}. \quad (6.16)$$

We evaluate the second term on the RHS using (3.10):

$$\mathbf{F}^{-1} = \sum_{e=1}^{\dim \vec{E}} \Delta \mathbf{X}_e \otimes \vec{\mathbf{W}}_e \Rightarrow \frac{d\mathbf{F}^{-1}}{d(\Delta \mathbf{X}_e)} = \vec{\mathbf{W}}_e. \quad (6.17)$$

Thus, the expression in (6.16) becomes:

$$\frac{d(\det \mathbf{F}^{-1})}{d(\Delta \mathbf{X}_e)} = \left(\mathbf{F}^{-\top} \right)^A \vec{\mathbf{W}}_e. \quad (6.18)$$

The expressions for the adjugate of $\mathbf{F}^{-1}$ are evaluated directly; with these values, explicit expressions for the matrix-vector product of (6.18) were obtained using Mathematica.

The second derivatives of these terms are not explicitly employed in the optimization routines; however, the contraction of the Hessian with other factors is needed. For $\mathcal{F}$, one requires the contraction of the Hessian with an element from the domain, i.e., $\mathcal{F}''z$. From (6.13a), there is only one non-zero block in $\mathcal{F}''$, and this term involves

$$\frac{\partial^2 f}{\partial u^2} \quad \text{where} \quad \left(\frac{\partial^2 f}{\partial u^2} \right)_{ij} = \frac{\partial^2 f}{\partial u_i \partial u_j}, \quad i, j = 1, \dots, Dn_{\text{ed}}. \quad (6.19)$$

From the definition of f in (6.3), it is seen that

$$\frac{\partial^2 f}{\partial u_i \partial u_j} = \delta_{ij}, \quad (6.20)$$

so the elements of $\mathcal{F}''z$ can be evaluated directly. For $\mathcal{G}$ and $\mathcal{H}$, the required term is slightly more complicated. Specifically, one requires an expression for the operator

$$(\Phi''x)^\dagger y. \quad (6.21)$$

where Φ represents $\mathcal{G}$ or $\mathcal{H}$, with $\Phi: \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the appropriate (finite-dimensional) spaces. Here, the superscript $\dagger$ denotes the adjoint of the operator: if $\mathcal{L}: \mathcal{X} \rightarrow \mathcal{Y}$, then $\mathcal{L}^\dagger: \mathcal{Y} \rightarrow \mathcal{X}$ such that

$$\langle \mathcal{L}x, y \rangle = \langle x, \mathcal{L}^\dagger y \rangle, \quad (6.22)$$

where $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $\langle \cdot, \cdot \rangle$ is the inner product on the appropriate space. For the operators under consideration, the k th term in (6.21) can be expressed, in indicial notation, as

$$\left[(\Phi''x)^\dagger y \right]_k = \frac{\partial^2 \Phi_i}{\partial x_j \partial x_k} x_j y_i, \quad k = 1, \dots, \dim \mathcal{X}, \quad (6.23)$$

with sums over repeated indices. For $\Phi = \mathcal{H}$, the expression for $\mathcal{H}'$ in (6.13c) implies that all higher-than-first derivatives of $\mathcal{H}$ vanish, so the expression in (6.21) is simply the null operator. For $\mathcal{G}$, the equality constraint $g(u) = 0$ is linear in its sole argument u , so the second derivative of g with respect to u vanishes; obviously, the second derivative of g with respect to s vanishes, as well. The second element of $\mathcal{G}$, i.e., $h(u) - s$, only has non-zero second derivative terms arising from $h''(u)$: thus, one requires an expression for the second derivative of the determinant of the

IDG with respect to the increments in material coordinates along the edges (i.e., the elements of u). There are general forms for the second derivative of a determinant of a matrix with respect to its arguments [38]; however, these representations (i) involve the inverse of the matrix (which we prefer not to evaluate explicitly) and (ii) may become unreliable as the matrix becomes singular (which is precisely the neighborhood of interest). Therefore, we resort yet again to expressions based on the matrix adjugate. See Appendix C for additional details.

Chapter 7

Numerical Examples of Extrema-Preserving Remap

In this section, the test problems considered in Chapter 5 are computed with the extremum-preserving remap of the previous section. In this section, we run the problems out to later times, in order to evaluate the effect of the optimization procedure on the solution.

Neither the pure Lagrangian calculations nor the zero-curl remapped calculations could run to these later times: the former led to inverted elements, and the later failed due to negative determinants. Therefore, in this section we compare the optimized zero-curl remap results with those obtained by remapping the VR-decomposition of the IDG directly. In that approach, the individual components of the stretch tensor $\mathbf{V}$ and rotation tensor $\mathbf{R}$ are remapped directly using a cell-centered finite-volume approach. The rotation tensor is represented as a unit quaternion; the quaternion is re-normalized after the remap is complete. For the stretch tensor, clipping is used; excessively large or small remapped stretches are clipped so that $10^{-6} < \lambda_{\min}, \lambda_{\max} < 10^6$. See Robinson et al. [41] for a discussion of this method.

The optimized zero-curl remap approach proved to be computationally intensive, with calculations taking two to three orders of magnitude longer than the standard approach. The initial computations with this method took even longer than the results cited, which were completed after preliminary efforts to tune certain optimization parameters.

Previous experience with optimizations method suggests that careful restatement of the overall optimization problem could lead to notable speed-up. It is unlikely, however, that the present approach could achieve run-times that would be competitive with existing methods.

Exponential vortical flow

This problem is the same as that described in Chapter 5: smoothly varying angular flow with an exponential radial decay. The final time has been increased to $t = 20$, significantly greater than the time ($t = 5$) evaluated previously.

For this analysis, the domain is discretized with a mesh of $n_{\text{el}} \times n_{\text{el}}$ initially square finite elements, where $n_{\text{el}} = \{16, 32, 64\}$. Figure 7.1 contains plots of the absolute magnitude of the the

stretch tensor, $\|\mathbf{V}\|$, on the three different mesh resolutions, at the final time $t = 20$. There is significant difference between the maximum stretch magnitude computed with the two methods. Because of this difference, these plots are displayed on different scales for the different methods; if the plot had been over the same range, then either one or the other result would exhibit little discernible structure. The gross features of the results of the two methods differ, as well. In particular, the apparent radius of the appreciably stretched material appears greater for the optimized remap case, which suggests that this method may have greater effective radial diffusion.

Table 7.1 contains the convergence rates for the stretch tensor for both the QR+clipped remap and optimized remap methodologies. Table 7.2 provides the corresponding values for the rotation tensor. Since the all fields associated with this problem are smooth, the computed convergence rate should be approximately equal to the designed rate of the underlying algorithm, which is two. For the matrix elements of both the stretch and rotation matrices, the QR+clipping method is much closer to the design convergence rate for the finest mesh. The optimized remap results are not converging near the theoretical rate, indicating problems with this approach. While it would be possible to ascertain if the QR+clipping method more closely approaches the theoretical characteristics of the algorithm on finer meshes, such an investigation for the optimized, zero-curl approach would be prohibitively expensive at this time.

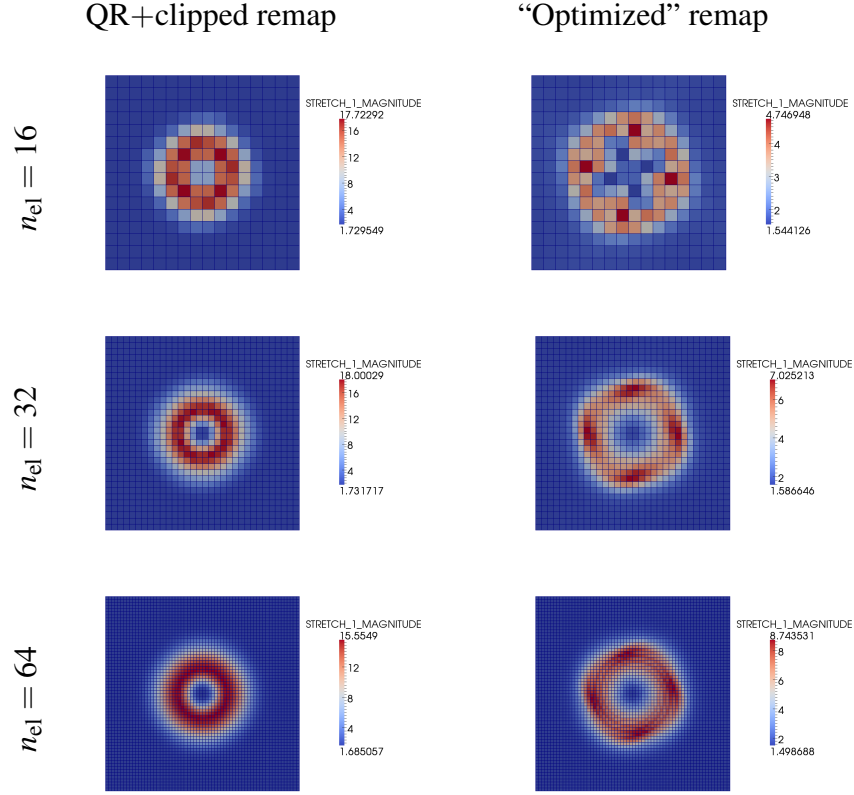


Figure 7.1. Color-coded plots of the magnitude of the stretch tensor, $\|\mathbf{V}\|$, at simulation time $t = 20$ for the QR+clipped remap every timestep calculation (left column), and the optimized zero-curl remap every timestep calculation (right column) of the exponential vortical flow problem. The rows correspond to the mesh resolution, from the coarsest (16×16) on top to the finest (64×64) on the bottom.

Table 7.1. Convergence rates for components of the stretch matrix $\mathbf{V}$ (in $\mathbf{F} = \mathbf{VR}$) for the exponential vortical flow problem, calculated with the norms indicated in the column headings, at $t = 20$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-5, 5] \times [-5, 5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

QR+clipped remap				“Optimized” remap		
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
V_{xx}	16	$\rangle$				
	32					
	64					
	1.08913	0.796569	0.458533	0.314500	0.308998	0.216502
	1.85106	1.62649	1.313899	0.318817	0.378219	0.256597
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
V_{xy}	16	$\rangle$				
	32					
	64					
	1.45983	1.02780	0.340759	0.439205	0.378651	0.316431
	1.492420	1.35599	1.30612	0.552993	0.427437	0.251303
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
V_{yy}	16	$\rangle$				
	32					
	64					
	1.09782	0.797821	0.429801	0.314500	0.308998	0.216502
	1.84142	1.61439	1.31608	0.318817	0.378219	0.256597

Table 7.2. Convergence rates for components of the rotation matrix $\mathbf{R}$ (in $\mathbf{F} = \mathbf{VR}$) for the exponential vortical flow problem, calculated with the norms indicated in the column headings, at $t = 20$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-5, 5] \times [-5, 5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

QR+clipped remap				“Optimized” remap		
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{xx}	16	$\rangle$	$\rangle$	$\rangle$	$\rangle$	$\rangle$
	32					
	64					
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{xy}	16	$\rangle$	$\rangle$	$\rangle$	$\rangle$	$\rangle$
	32					
	64					
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{yx}	16	$\rangle$	$\rangle$	$\rangle$	$\rangle$	$\rangle$
	32					
	64					
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
R_{yy}	16	$\rangle$	$\rangle$	$\rangle$	$\rangle$	$\rangle$
	32					
	64					

ABC vortical flow

This problem is the same as that described in Chapter 5 and consists of four separate cylindrical regions, with continuous velocity and discontinuous velocity gradients at the interfaces between the regions. The final time has been increased to $t = 2$, significantly greater than the time ($t = 0.5$) evaluated previously.

Figure 7.2 contains plots of the absolute magnitude of the the stretch tensor, $\|\mathbf{V}\|$, on the three different mesh resolutions, at the final time $t = 2$. As for the previous test problem, there is significant difference between the maximum stretch magnitude for the different methods. Again, because of these differences, these plots for the different methods are displayed on different scales. The detailed structure between the two methods again differs. The radius of material with maximum stretch is greater for the optimized remap case once again; moreover, the area of increased stretch (which is of lower absolute magnitude) appears more broadly distributed. These results suggest again that the optimized remap method has greater effective radial diffusion.

Table 7.3 contains the convergence rates for the stretch tensor elements for both the QR+clipped remap and “optimized” zero-curl remap methodologies. Table 7.4 documents the corresponding values for the rotation tensor. Given the discontinuities in this problem, one should anticipate first-order convergence in the L_1 norm for a properly implemented (and convergent) algorithm. The QR+clipping method is reasonably close to the design convergence rate for the finest mesh for both the stretch and rotation matrices. Consistent with results on the previous problem, the optimized remap results are not converging at near the theoretical rate.

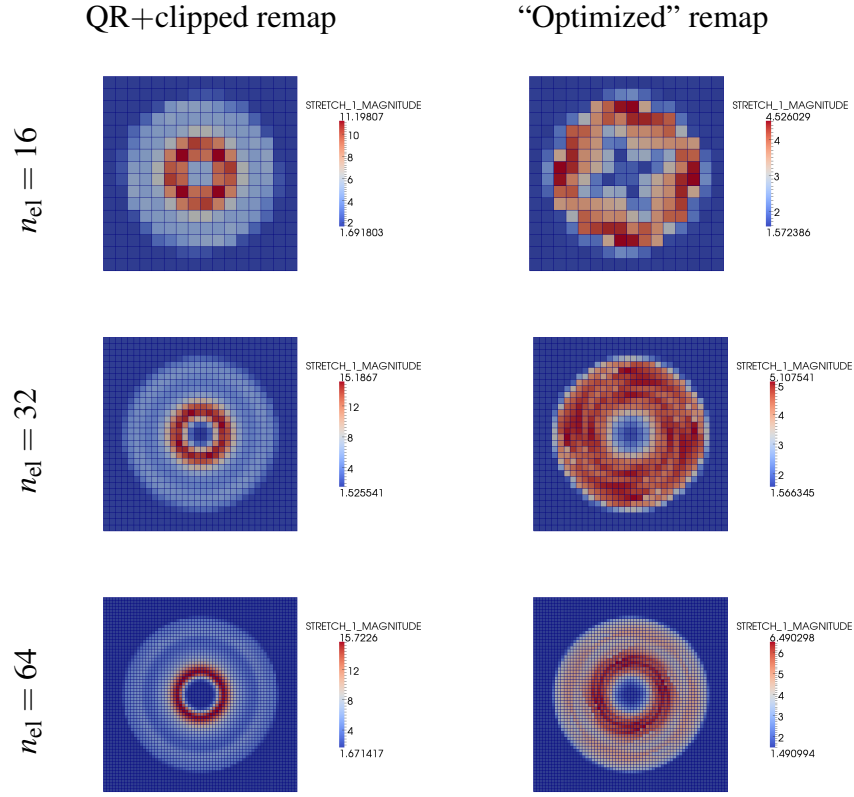


Figure 7.2. Color-coded plots of the magnitude of the stretch tensor, $\|\mathbf{V}\|$, at simulation time $t = 2$ for the QR+clipped remap every timestep calculation (left column), and the optimized zero-curl remap every timestep calculation (right column) of the ABC vortical flow problem. The rows correspond to the mesh resolution, from the coarsest (16×16) on top to the finest (64×64) on the bottom.

Table 7.3. Convergence rates for components of the stretch matrix $\mathbf{V}$ (in $\mathbf{F} = \mathbf{VR}$) for the ABC vortex problem, calculated with the norms indicated in the column headings, at $t = 2$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-0.5, 0.5] \times [-0.5, 0.5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

QR+clipped remap				“Optimized” remap		
n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞
V_{xx}	16	$\rangle$		0.0790837	-0.343008	-1.14195
	32					
	64					
V_{xy}	16	$\rangle$		0.288347	0.0279148	-0.512174
	32					
	64					
V_{yy}	16	$\rangle$		0.0790837	-0.343008	-1.14195
	32					
	64					

Table 7.4. Convergence rates for components of the rotation matrix $\mathbf{R}$ (in $\mathbf{F} = \mathbf{VR}$) for the ABC vortical flow problem, calculated with the norms indicated in the column headings, at $t = 2$. The value n_{el} is the number of elements in each linear direction of the mesh on $[-0.5, 0.5] \times [-0.5, 0.5]$. Convergence rates are evaluated from calculations at two mesh resolutions together with the exact solution.

QR+clipped remap				“Optimized” remap				
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{xx}	16	$\rangle$						
	32		0.130699	-0.331789	-0.359335	0.509182	0.857022	0.575082
	64		1.36229	0.945824	-0.0196008	-0.0452588	-0.406977	-0.640735
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{xy}	16	$\rangle$						
	32		0.840856	0.454478	-0.166108	0.404932	0.0787619	-0.762331
	64		0.681295	0.172749	-0.0314592	0.319195	0.264502	0.351374
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{yx}	16	$\rangle$						
	32		0.840856	0.454478	-0.166108	0.404932	0.0787619	-0.762331
	64		0.681295	0.172749	-0.0314592	0.319195	0.264502	0.351374
	n_{el}	L_1	L_2	L_∞	L_1	L_2	L_∞	
R_{yy}	16	$\rangle$						
	32		1.36229	-0.331789	-0.359335	0.509182	0.857022	0.575082
	64		0.130699	0.945824	-0.0196008	-0.0452588	-0.406977	-0.640735

This page intentionally left blank.

Chapter 8

Summary

This work presents a constraint-preserving approach to the problem of remapping solutions for computational solid mechanics simulations in the Arbitrary Lagrangian-Eulerian (ALE) framework using the edge element formulation. We restrict consideration to ALE methods on hexahedral meshes for which the remap mesh corresponds to the original mesh configuration. Two conditions—(1) zero-curl of the inverse deformation gradient and (2) positivity of the determinant of the deformation gradient—constrain allowable deformations. We describe a constraint-preserving remap method that is inspired by the constrained transport approach for numerical MHD. This approach intrinsically satisfies the zero-curl constraint on the inverse deformation gradient. Computed results on two kinematics-only test problems indicate that this basic method is convergent for small-but-finite deformations. Running these problems to later simulation times, however, reveals a shortcoming of this approach, namely, that non-physical extrema in the material stretches can be introduced.

In an attempt to ameliorate this issue, we formulated an associated constrained optimization problem that maintains the zero-curl property of the solution while providing solutions that do not introduce new extrema in the product of the stretches. Computed results, using a black-box optimization routine to simultaneously enforce the zero-curl and positive-determinant constraints, are both of different magnitude and more diffusive than the results calculated with a non-constraint preserving, element-by-element remap with clipped extremal stretches. In addition to this discrepancy, an additional drawback of the current optimization approach is that it is computationally intensive, being two to three orders of magnitude slower than the standard, element-by-element remap approach. Previous experience with optimization methods (see, e.g., [8, 9]), shows that careful restatement of the optimization problem can lead to appreciable speed-up. It is unclear, however, whether the present approach could achieve run-times competitive with existing methods. It is possible that further refinements of the underlying curl-preserving remap algorithm might increase accuracy slightly, e.g., by using more sophisticated quadrature schemes or higher-order elements.

Although the approaches discussed in this report appear, in theory, to be superior to the ad hoc methods in common use, it remains to be seen if this theoretical promise can be realized in practice.

This page intentionally left blank.

References

- [1] M. Ainsworth and J.T. Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142:1–88, 1997.
- [2] A. Angoshtari and A. Yavari. A geometric structure-preserving discretization scheme for incompressible linearized elasticity. *Computer Methods in Applied Mechanics and Engineering*, 259:130–153, 2013.
- [3] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus: From Hodge theory to numerical stability. *Bulletin of the American Mathematical Society*, 47:281–354, 2010.
- [4] D.N. Arnold, P.B. Bochev, R.B. Lehoucq, R.A. Nicolaides, and M. Shashkov. *Compatible Spatial Discretizations*. The IMA Volumes in Mathematics and its Applications, Volume 142. Springer, New York, 2006.
- [5] D.N. Arnold, R.S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, 2006.
- [6] P.B. Bochev. Variational and Geometric Aspects of Compatible Discretizations, Part II. Technical Report SAND2009-6223P, Sandia National Laboratories, 2009.
- [7] P.B. Bochev, J.J. Hu, A.C. Robinson, and R.S. Tuminaro. Towards robust 3D Z-pinch simulations: discretization and fast solvers for magnetic diffusion in heterogeneous conductors. *Electronic Transactions on Numerical Analysis*, 15:186–210, 2003.
- [8] P.B. Bochev, D. Ridzal, G. Scovazzi, and M. Shashkov. Formulation, analysis and numerical study of an optimization-based conservative interpolation (remap) of scalar fields for arbitrary Lagrangian-Eulerian methods. *J. Comput. Phys.*, 230:5199–5225, 2011.
- [9] P.B. Bochev, D. Ridzal, and M. Shashkov. Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. *J. Comput. Phys.*, 246:37–57, 2013.
- [10] P.B. Bochev and A.C. Robinson. Matching algorithms with physics: Exact sequences of finite element spaces. In D. Estep and S. Tavener, editors, *Collected Lectures on the Preservation of Stability under Discretization*, pages 145–165. Society for Industrial and Applied Mathematics, 2002.
- [11] Alain Bossavit. On the geometry of electromagnetism (1): Affine space. *J. Japan Soc. Appl. Electromag. Mech.*, 6:17–28, 1998.
- [12] Alain Bossavit. On the geometry of electromagnetism (2): Geometrical objects. *J. Japan Soc. Appl. Electromag. Mech.*, 6:114–123, 1998.

- [13] Alain Bossavit. On the geometry of electromagnetism (3): Integration, Stokes, Faraday's law. *J. Japan Soc. Appl. Electromag. Mech.*, 6:233–240, 1998.
- [14] Alain Bossavit. On the geometry of electromagnetism (4): “Maxwell's house”. *J. Japan Soc. Appl. Electromag. Mech.*, 6:318–326, 1998.
- [15] Alain Bossavit. Computational electromagnetism and geometry: Building a finite-dimensional “Maxwell's house” (1): Network equations. *J. Japan Soc. Appl. Electromag. Mech.*, 7:150–159, 1999.
- [16] Alain Bossavit. Computational electromagnetism and geometry: Building a finite-dimensional “Maxwell's house” (2): Network constitutive laws. *J. Japan Soc. Appl. Electromag. Mech.*, 7:294–301, 1999.
- [17] Alain Bossavit. Computational electromagnetism and geometry: Building a finite-dimensional “Maxwell's house” (3): Convergence. *J. Japan Soc. Appl. Electromag. Mech.*, 7:401–408, 1999.
- [18] Alain Bossavit. Computational electromagnetism and geometry: Building a finite-dimensional “Maxwell's house” (4): From degrees of freedom to fields. *J. Japan Soc. Appl. Electromag. Mech.*, 8:102–109, 2000.
- [19] Alain Bossavit. Computational electromagnetism and geometry: Building a finite-dimensional “Maxwell's house” (5): The “Galerkin hodge”. *J. Japan Soc. Appl. Electromag. Mech.*, 7:203–209, 2000.
- [20] Alain Bossavit. Computational electromagnetism and geometry: Building a finite-dimensional “Maxwell's house” (6): Some questions and answers. *J. Japan Soc. Appl. Electromag. Mech.*, 8:372–377, 2000.
- [21] Alain Bossavit. Discrete magneto-elasticity: A geometrical approach. *IEEE Transactions on Magnetics*, 46:3485–3491, 2010.
- [22] D.S. Chandrasekharaiah. Stokes type theorem for tensors. *Indian J. Pure Appl. Math*, 21:379–383, 1990.
- [23] S. H. Christiansen, H. Z. Munthe-Kaas, and B. Owren. Topics in structure-preserving discretization. *Acta Numerica*, 20:1–119, 2011.
- [24] D. Day, P. Bochev, K. Pendley, C. Weiss, and A. Robinson. Geophysical Subsurface Imaging and Interface Identification. Technical Report SAND2005-6272, Sandia National Laboratories, 2005.
- [25] A. A. Dezin. *Multidimensional Analysis and Discrete Models*. CRC Press, Boca Raton, 1995.
- [26] M. Eastwood. A complex from linear elasticity. In J. Slovák and M. Čadek, editors, *Proceedings of the 19th Winter School “Geometry and Physics”*. *Circolo Matematico di Palermo*, pages 23–29, 2000.

- [27] M. Eastwood. Ricci curvature and the mechanics of solids. *Australian Mathematical Society Gazette*, 37:238–241, 2010. See: arXiv:1005.2010v1.
- [28] C. R. Evans and J. F. Hawley. Simulation of magnetohydrodynamic flows: a constrained transport method. *The Astrophysical Journal*, 332:659–677, 1988.
- [29] Paul W. Gross and P. Robert Kotiuga. *Electromagnetic Theory and Computation: A Topological Approach*. Cambridge University Press, Cambridge, 2004. Mathematical Sciences Research Institute Publications Vol. 48.
- [30] R. D. Hill and E. E. Underwood. On the matrix adjoint (adjugate). *SIAM J. Alg. Disc. Meth.*, 6:731–737, 1985.
- [31] A. Hirani. *Discrete Exterior Calculus*. PhD thesis, California Institute of Technology, 2003.
- [32] K. Kamrin, C. Rycroft, and J.-C. Nave. Reference map technique for finite-strain elasticity and fluid-solid interaction. *Journal of the Mechanics and Physics of Solids*, 60:1952–1969, 2012.
- [33] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.
- [34] J. Lubliner. *Plasticity Theory*. Dover Publications, Inc., Mineola, NY, 2006. <http://www.ce.berkeley.edu/~coby/plas/pdf/book.pdf>.
- [35] G. Luttwak and J. Falcovitz. Slope limiting for vectors: A novel vector limiting algorithm. *International Journal for Numerical Methods in Fluids*, 65:1365–1375, 2011.
- [36] P.-H. Maire, R. Loubère, and P. Váchal. Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme. *Communications in Computational Physics*, 10:940–978, 2011.
- [37] C. Mattiussi. A reference discretization strategy for the numerical solution of physical field problems. *Advances in Imaging and Electron Physics*, 121:143–279, 2002.
- [38] K. B. Petersen and M. S. Pedersen. The matrix cookbook. Technical report, Technical University of Denmark, 2012. Version 20121115.
- [39] Denis Ridzal, Miguel Aguiló, and Matthias Heinkenschloss. Numerical study of a matrix-free trust-region sqp method for equality constrained optimization. Technical Report SAND2011-9346, Sandia National Laboratories, 2011.
- [40] R. N. Rieben, D. A. White, B. K. Wallin, and J. M. Solvberg. An arbitrary Lagrangian-Eulerian discretization of MHD on 3D unstructured grids. *Journal of Computational Physics*, 226:534–570, 2007.
- [41] A. C. Robinson, David I. Ketcheson, Thomas L. Ames, and Grant V. Farnsworth. A comparison of Lagrangian/Eulerian approaches for tracking the kinematics of high deformation solid motion. Technical Report SAND2009-5154, Sandia National Laboratories, 2009.

- [42] G. W. Stewart. On the adjugate matrix. *Linear Algebra and its Applications*, 283:151–164, 1998.
- [43] E. Tonti. *On the Formal Structure of Physical Theories*. Consiglio Nazionale delle Ricerche, Milan, 1975.
- [44] D. H. Wagner. Conservation laws, coordinate transformations, and differential forms. In J. Glimm, M.J. Graham, J.W. Grove, and B.J. Plohr, editors, *Proceedings of the Fifth International Conference on Hyperbolic Problems: Theory, Numerics, and Applications*, pages 471–477, Singapore, 1996. World Scientific Publishers.
- [45] D. H. Wagner. Symmetric hyperbolic equations of motion for a hyperelastic material. *Journal of Hyperbolic Differential Equations*, 6:615–630, 2009.
- [46] A. Yavari. On geometric discretization of elasticity. *Journal of Mathematical Physics*, 49:022901–1–022901–35, 2009.
- [47] Joseph Young, Denis Ridzal, and Miguel Aguiló. Efficient handling of inequality constraints in a matrix-free trust-region sqp algorithm. Technical Report SAND2012-2495C, Sandia National Laboratories, 2012.
- [48] O.C. Zienkiewicz, B. Boroomand, and J.Z. Zhu. Recovery procedures in error estimation and adaptivity. Part I: Adaptivity in linear problems. *Computer Methods in Applied Mechanics and Engineering*, 176:111–125, 1999.
- [49] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part I: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33:1331–1364, 1992.

Appendix A

Curl of a Tensor

Let $\mathbf{T}$ be a tensor and $\mathbf{v}$ be a vector. From the definition in (3.13),

$$(\text{curl } \mathbf{T}) \mathbf{v} := \text{curl} \left(\mathbf{T}^\top \mathbf{v} \right) \quad (\text{A.1})$$

for all constant vectors $\mathbf{v}$. The i th component of the LHS of (A.1) is evaluated as

$$((\text{curl } \mathbf{T}) \mathbf{v})_i = (\text{curl } \mathbf{T})_{il} v_l. \quad (\text{A.2})$$

Also, by definition,

$$(\mathbf{T} \mathbf{v})_i = T_{il} v_l, \quad (\text{A.3})$$

so that

$$\left(\mathbf{T}^\top \mathbf{v} \right)_i = T_{li} v_l, \quad (\text{A.4})$$

which is the i th component of the RHS of (A.1).

The curl operator is defined on the vector $\mathbf{w}$ such that the i th component of $\text{curl } \mathbf{w}$ is given by

$$(\text{curl } \mathbf{w})_i := \varepsilon_{ijk} w_{k,j}. \quad (\text{A.5})$$

Therefore, the i th component of the curl of the vector $\mathbf{T}^\top \mathbf{v}$ (see (A.4)) can be written as:

$$\begin{aligned} \left(\text{curl} \left(\mathbf{T}^\top \mathbf{v} \right) \right)_i &= \varepsilon_{ijk} \left(\mathbf{T}^\top \mathbf{v} \right)_{k,j} \\ &= \varepsilon_{ijk} (T_{lk} v_l)_{,j} \\ &= \varepsilon_{ijk} T_{lk,j} v_l \end{aligned} \quad (\text{A.6})$$

where we assume that $\mathbf{v}$ is independent of $\mathbf{x}$. From the definition in (A.1), this expression equals the i th component of $(\text{curl } \mathbf{T}) \mathbf{v}$, which is given in (A.2). Since the equality in (A.1) is true for all $\mathbf{a}$, it follows that

$$(\text{curl } \mathbf{T})_{il} = \varepsilon_{ijk} T_{lk,j}. \quad (\text{A.7})$$

Renaming the dummy indices in this expression,

$$(\text{curl } \mathbf{T})_{kl} = \varepsilon_{kji} T_{li,j}. \quad (\text{A.8})$$

Using properties of the permutation symbol,

$$(\text{curl } \mathbf{T})_{kl} = -\varepsilon_{ijk} T_{li,j}, \quad (\text{A.9})$$

which is the result of (3.14).

This page intentionally left blank.

Appendix B

Angular Flow Exact Solutions

In this appendix, expressions are derived for the tangent vectors and deformation gradient for rotational flow. Let (r, θ) denote the plane polar coordinates in the spatial (deformed) frame, and let (R, Θ) represent the plane polar coordinates in the material (undeformed) frame.

We consider deformations related to velocity fields in which the flow depends only on angle. Thus, the mapping between the material and spatial frames is given by:

$$\begin{aligned} r &= R, \\ \theta &= f(R) + \Theta. \end{aligned} \quad (\text{B.1})$$

The material position $\mathbf{X}$ is given in terms of the material-frame polar coordinates (R, Θ) as:

$$\mathbf{X} := (R \cos \Theta) \mathbf{E}_1 + (R \sin \Theta) \mathbf{E}_2, \quad (\text{B.2})$$

where $\mathbf{E}_1$ and $\mathbf{E}_2$ are the Cartesian unit vectors in the material frame. The material covariant basis vectors are defined as the local derivatives of the material positions with respect to the polar coordinates:

$$\begin{aligned} \mathbf{G}_1 &:= \frac{\partial \mathbf{X}}{\partial R} = (\cos \Theta) \mathbf{E}_1 + (\sin \Theta) \mathbf{E}_2, \\ \mathbf{G}_2 &:= \frac{\partial \mathbf{X}}{\partial \Theta} = (-R \sin \Theta) \mathbf{E}_1 + (R \cos \Theta) \mathbf{E}_2. \end{aligned} \quad (\text{B.3})$$

Requiring that the contravariant basis vectors be dual to the covariant basis vectors, i.e., $\mathbf{G}_i \cdot \mathbf{G}^j = \delta_i^j$, it is seen by inspection that the contravariant basis vectors are

$$\begin{aligned} \mathbf{G}^1 &= (\cos \Theta) \mathbf{E}_1 + (\sin \Theta) \mathbf{E}_2, \\ \mathbf{G}^2 &= \left(-\frac{1}{R} \sin \Theta \right) \mathbf{E}_1 + \left(\frac{1}{R} \cos \Theta \right) \mathbf{E}_2. \end{aligned} \quad (\text{B.4})$$

Similarly, in the spatial frame, the position vector is, in terms of plane polar coordinates,

$$\mathbf{x} = (r \cos \theta) \mathbf{e}_1 + (r \sin \theta) \mathbf{e}_2, \quad (\text{B.5})$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are the Cartesian unit vectors in the spatial frame. The relation between the plane Cartesian and plane polar unit vectors in the spatial frame is given by:

$$\begin{aligned} \mathbf{e}_r &:= (\cos \theta) \mathbf{e}_1 + (\sin \theta) \mathbf{e}_2, \\ \mathbf{e}_\theta &:= (-\sin \theta) \mathbf{e}_1 + (\cos \theta) \mathbf{e}_2. \end{aligned} \quad (\text{B.6})$$

The spatial covariant basis vectors are defined as:

$$\begin{aligned}\mathbf{g}_1 &:= \frac{\partial \mathbf{x}}{\partial R} = (\cos \theta - r \sin \theta f'(R)) \mathbf{e}_1 + (\sin \theta + r \cos \theta f'(R)) \mathbf{e}_2, \\ \mathbf{g}_2 &:= \frac{\partial \mathbf{x}}{\partial \Theta} = (-r \sin \theta) \mathbf{e}_1 + (r \cos \theta) \mathbf{e}_2.\end{aligned}\tag{B.7}$$

In terms of the polar unit vectors of (B.6), the spatial covariant basis vectors can be expressed as:

$$\begin{aligned}\mathbf{g}_1 &= \mathbf{e}_r + R f'(R) \mathbf{e}_\theta, \\ \mathbf{g}_2 &= R \mathbf{e}_\theta.\end{aligned}\tag{B.8}$$

To obtain convenient expressions for material contravariant basis vectors, introduce the following notation. Let the angular difference, $\Delta\theta$, be the difference between the spatial polar coordinate angle θ and the material polar coordinate angle, Θ . From (B.1), this term can be written as:

$$\Delta\theta := \theta - \Theta = f(R).\tag{B.9}$$

From trigonometric formulas for the difference of two angles,

$$\begin{aligned}\cos \Theta &= \cos(\theta - \Delta\theta) = (\cos \theta) (\cos \Delta\theta) + (\sin \theta) (\sin \Delta\theta), \\ \sin \Theta &= \sin(\theta - \Delta\theta) = (\sin \theta) (\cos \Delta\theta) - (\cos \theta) (\sin \Delta\theta).\end{aligned}\tag{B.10}$$

Without loss of generality, assume $\mathbf{e}_1 = \mathbf{E}_1$ and $\mathbf{e}_2 = \mathbf{E}_2$. Then, from (B.4), (B.6), and (B.10) it can be shown that

$$\begin{aligned}\mathbf{G}^1 &= (\cos \Delta\theta) \mathbf{e}_r - (\sin \Delta\theta) \mathbf{e}_\theta, \\ \mathbf{G}^2 &= \frac{1}{R} (\sin \Delta\theta) \mathbf{e}_r + \frac{1}{R} (\cos \Delta\theta) \mathbf{e}_\theta.\end{aligned}\tag{B.11}$$

Using these results, we seek to represent the deformation gradient, which is the unique linear mapping $\mathbf{F}$ such that

$$\mathbf{F} \mathbf{G}_A = \mathbf{g}_A, \quad A = 1, 2.\tag{B.12}$$

The solution for $\mathbf{F}$ in this equation is given by

$$\mathbf{F} = \sum_{A=1}^2 \mathbf{g}_A \otimes \mathbf{G}^A.\tag{B.13}$$

Therefore, from (B.8) and (B.11) one obtains closed-form expressions for the deformation gradient. In practice, these expressions are evaluated numerically.

Appendix C

The Matrix Adjugate

As indicated in Chapter 6, implementation of the optimization routines requires expressions for the first and second derivatives of the determinant of a matrix. The standard expressions for these quantities (see, e.g., [38]) contain the inverse of the matrix. Such representations are undesirable for the present purposes from the perspective of both computational efficiency and numerical accuracy as the matrix becomes singular, which is the neighborhood in which the optimization algorithm operates.

For an invertible matrix $\mathbf{A}(\zeta)$ that is differentiable in the argument ζ , Jacobi's formula provides an expression for the derivative of the determinant of the matrix with respect to ζ :

$$\frac{d}{d\zeta}(\det \mathbf{A}) = \text{tr} \left(\mathbf{A}^A \frac{d\mathbf{A}}{d\zeta} \right) = \left(\mathbf{A}^A \right)^T : \frac{d\mathbf{A}}{d\zeta}, \quad (\text{C.1})$$

where $\mathbf{A}^A$ is the adjugate of $\mathbf{A}$. The adjugate [30, 42] of the $n \times n$ matrix $\mathbf{A}$ is the $n \times n$ matrix $\mathbf{A}^A$ defined as the transpose of the matrix of the cofactors of the elements of $\mathbf{A}$, so that the (i, j) -element of $\mathbf{A}^A$ is given by:

$$(\mathbf{A}^A)_{ij} := (-1)^{i+j} \det \mathbf{A}(i|j), \quad (\text{C.2})$$

where $\mathbf{A}(i|j)$ represents the $(n-1) \times (n-1)$ sub-matrix of $\mathbf{A}$ obtained by deleting the i th row and j th column. Any square matrix satisfies the relation

$$\mathbf{A} \mathbf{A}^A = \mathbf{A}^A \mathbf{A} = (\det \mathbf{A}) \mathbf{I}, \quad (\text{C.3})$$

where $\mathbf{I}$ is the unit matrix. From (C.3), when $\mathbf{A}$ is invertible, the adjugate can be expressed as:

$$\mathbf{A}^A = (\det \mathbf{A}) \mathbf{A}^{-1}. \quad (\text{C.4})$$

As stated by Stewart [42], "... although $\mathbf{A}^{-1}$ and $\mathbf{A}^A$ differ only by a scalar factor, the matrix $\mathbf{A}^{-1}$ has singularities while $\mathbf{A}^A$ is analytic—in fact, it is a multinomial in the elements of $\mathbf{A}$." Thus, the adjugate of a matrix remains smooth even as the matrix itself becomes singular, which is important for the present application.

We seek to evaluate (C.1) in the case where matrix is the IDG and the independent variable is the increment in material coordinates along the e th edge:

$$\frac{d(\det \mathbf{F}^{-T})}{d(\Delta \mathbf{X}_e)} = (\mathbf{F}^{-1})^A : \frac{d\mathbf{F}^{-1}}{d(\Delta \mathbf{X}_e)}. \quad (\text{C.5})$$

We evaluate the second term on the RHS using (3.10):

$$\mathbf{F}^{-1} = \sum_{e=1}^{\dim \vec{\mathcal{E}}} \Delta \mathbf{X}_e \otimes \vec{\mathbf{W}}_e \Rightarrow \frac{d\mathbf{F}^{-1}}{d(\Delta \mathbf{X}_e)} = \vec{\mathbf{W}}_e. \quad (\text{C.6})$$

Thus, the expression in (C.5) becomes:

$$\frac{d(\det \mathbf{F}^{-1})}{d(\Delta \mathbf{X}_e)} = \left(\mathbf{F}^{-\top} \right)^A \vec{\mathbf{W}}_e. \quad (\text{C.7})$$

Using the definitions above, expressions for the adjugate of $\mathbf{F}^{-1}$ are evaluated directly. Using Mathematica, explicit expressions for the matrix-vector product of (C.7) were obtained, as well as expressions for the second derivative of this quantity.

Evaluating the derivative(s) of (C.7) requires evaluation of the derivatives of $(\mathbf{F}^{-1})^A$. To be more specific, let $\mathbf{G}$ be a matrix with components

$$[\mathbf{G}] = \left[\begin{array}{c|c|c} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{array} \right]. \quad (\text{C.8})$$

The adjugate of $\mathbf{G}$ is

$$[\mathbf{G}^A] = \left[\begin{array}{c|c|c} -G_{23}G_{32} + G_{22}G_{33} & G_{13}G_{32} - G_{12}G_{33} & -G_{13}G_{22} + G_{12}G_{23} \\ G_{23}G_{31} - G_{21}G_{33} & -G_{13}G_{31} + G_{11}G_{33} & G_{13}G_{21} - G_{11}G_{23} \\ -G_{22}G_{31} + G_{21}G_{32} & G_{12}G_{31} - G_{11}G_{32} & -G_{12}G_{21} + G_{11}G_{22} \end{array} \right]. \quad (\text{C.9})$$

This example illustrates that the components of the adjugate are polynomial functions of the components of the matrix. Thus, computing derivatives is tedious but straightforward. For example,

$$\frac{d[\mathbf{G}^A]}{dG_{11}} = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ 0 & -G_{33} & -G_{23} \\ 0 & -G_{32} & G_{22} \end{array} \right]. \quad (\text{C.10})$$

Expressions for the two-dimensional case are developed analogously. Now let $\mathbf{G} := \mathbf{F}^{-1}$. Then the (pseudo-time/directional) derivative of (C.7) can now be computed as

$$\begin{aligned} \frac{d}{dt} \left(\mathbf{G}^A \vec{\mathbf{W}}_e \right) &= \left(\frac{d\mathbf{G}^A}{dG_{ij}} \dot{G}_{ij} \right) \vec{\mathbf{W}}_e \\ &= \frac{d\mathbf{G}^A}{dG_{ij}} (\mathbf{e}_i \cdot \dot{\mathbf{G}} \mathbf{e}_j) \vec{\mathbf{W}}_e \\ &= \frac{d\mathbf{G}^A}{dG_{ij}} \left[\mathbf{e}_i \cdot (\Delta \dot{\mathbf{X}}_f \otimes \vec{\mathbf{W}}_f) \mathbf{e}_j \right] \vec{\mathbf{W}}_e \\ &= \frac{d\mathbf{G}^A}{dG_{ij}} \left[(\mathbf{e}_i \cdot \dot{\mathbf{X}}_f) (\vec{\mathbf{W}}_f \cdot \mathbf{e}_j) \right] \vec{\mathbf{W}}_e \\ &= \left[\frac{d\mathbf{G}^A}{dG_{ij}} (\vec{\mathbf{W}}_f \cdot \mathbf{e}_j) (\vec{\mathbf{W}}_e \otimes \mathbf{e}_i) \right] \Delta \dot{\mathbf{X}}_f. \end{aligned} \quad (\text{C.11})$$

DISTRIBUTION:

1 MS 0380	D. Day, 01542 (electronic copy)
1 MS 0380	M. Heinstein, 01542 (electronic copy)
1 MS 0380	K. Pierson, 01542 (electronic copy)
1 MS 0828	B. Carnes, 01544 (electronic copy)
1 MS 0828	K. Copps, 01544 (electronic copy)
1 MS 0836	S. Domino, 01541 (electronic copy)
1 MS 0845	M. Aguiló, 01542 (electronic copy)
1 MS 1318	J. Stewart, 01441 (electronic copy)
1 MS 1318	T. Trucano, 01440 (electronic copy)
1 MS 1320	P. Bochev, 01442 (electronic copy)
1 MS 1320	D. Ridzal, 01441 (electronic copy)
1 MS 1323	E. Love, 01443 (electronic copy)
1 MS 1323	A. Robinson, 01443 (electronic copy)
1 MS 1323	O.E. Strack, 01443 (electronic copy)
1 MS 0899	Technical Library, 9536 (electronic copy)

This page intentionally left blank.

