SAND2007-8146C

Numerical Eigenvalue Analysis with Applications to Constitutive Modeling

William Scherzinger

Sandia National Laboratories

Albuquerque, NM 87185-0372

Sandia National Laboratories

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

National Nuclear Security Administration

Overview

- Kinematics in Principal Coordinates
  - Strains
  - Strain Rates
  - Numerical Issues
- Eigenvalue/Eigenvector Solution
- Applications
- Conclusions

Kinematics

- "Strain" – two states
  - Deformation gradient

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X}$$

- "Strain Rate" – one state
  - Rate of deformation

$$d\mathbf{v} = \mathbf{L} \cdot d\mathbf{x}$$

$$\mathbf{L} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1}$$

Kinematics

- Strains derived from deformation gradient
  - Polar Decomposition

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{R}$$

  - Green-Lagrange

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}\right) = \frac{1}{2}\left(\mathbf{U}^2 - \mathbf{I}\right)$$

  - Seth-Hill

$$\mathbf{E}^{(n)} = \frac{1}{n}\left(\mathbf{U}^n - \mathbf{I}\right)$$

Kinematics

- Strains in principal coordinates
  - Right Stretch Tensor

$$\mathbf{U} = \sum_{i=1}^{3} \lambda_i \overline{\mathbf{e}}_i \overline{\mathbf{e}}_i \qquad \left(\mathbf{U} = \sum_{i=1}^{3} \lambda_i \overline{\mathbf{e}}_i \otimes \overline{\mathbf{e}}_i \right)$$

  - Seth-Hill

$$\mathbf{E}^{(n)} = \frac{1}{n} \sum_{i=1}^{3} \left(\lambda_i^n - 1 \right) \overline{\mathbf{e}}_i \overline{\mathbf{e}}_i$$

Sandia National Laboratories

NNSA National Nuclear Security Administration

Kinematics

- Strains in finite element codes
  - Finite deformation finite element codes are solved incrementally

$$t_0, t_1, t_2, \ldots, t_n, t_{n+1}$$

$$t_{n+1} = t_n + \Delta t$$

  - Approach lends itself well to strains – two states

$$d\mathbf{x}_{n+1} = \mathbf{F} \cdot d\mathbf{X} \quad ; \quad t_0, t_{n+1}$$

$$d\mathbf{x}_{n+1} = \hat{\mathbf{F}} \cdot d\mathbf{x}_n \quad ; \quad t_n, t_{n+1}$$

Kinematics

- Strain rates
  - Strain rates are relatively simple

$$\dot{\mathbf{E}} = \frac{1}{2}\left(\dot{\mathbf{F}}^T \cdot \mathbf{F} + \mathbf{F}^T \cdot \dot{\mathbf{F}}\right)$$

  - But we tend to use the rate of deformation for constitutive models (work conjugate issues in a hypoelastic formulations)

$$\mathbf{D} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^T\right)$$

Kinematics

- Rate of Deformation
  - Occurs at a point in time – problems for FE codes
  - In a finite element solution, when do we calculate it?

$$\mathbf{D}_n = \frac{1}{2}\left(\mathbf{L}_n + \mathbf{L}_n^T\right) \quad ; \quad \mathbf{L}_n = \frac{\partial \mathbf{v}}{\partial \mathbf{x}_n}$$

$$\mathbf{D}_{n+1} = \frac{1}{2}\left(\mathbf{L}_{n+1} + \mathbf{L}_{n+1}^T\right) \quad ; \quad \mathbf{L}_{n+1} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}_{n+1}}$$

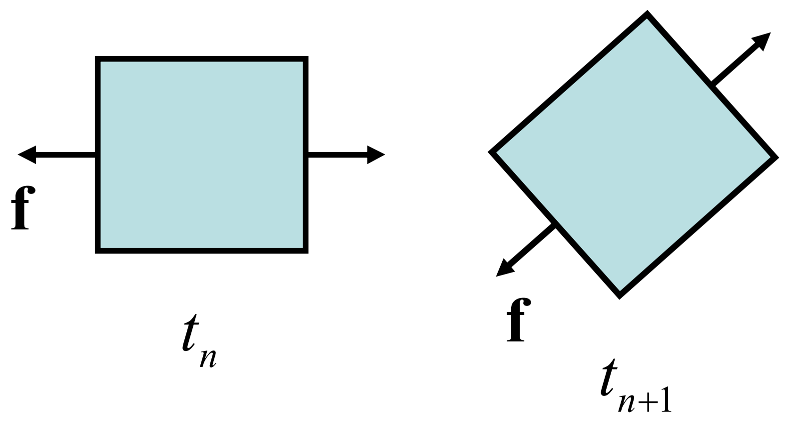
$$\mathbf{D}_{n+\alpha} \frac{1}{2}\left(\mathbf{L}_{n+\alpha} + \mathbf{L}_{n+\alpha}^T\right) \quad ; \quad \mathbf{L}_{n+\alpha} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}_{n+\alpha}}$$

Kinematics

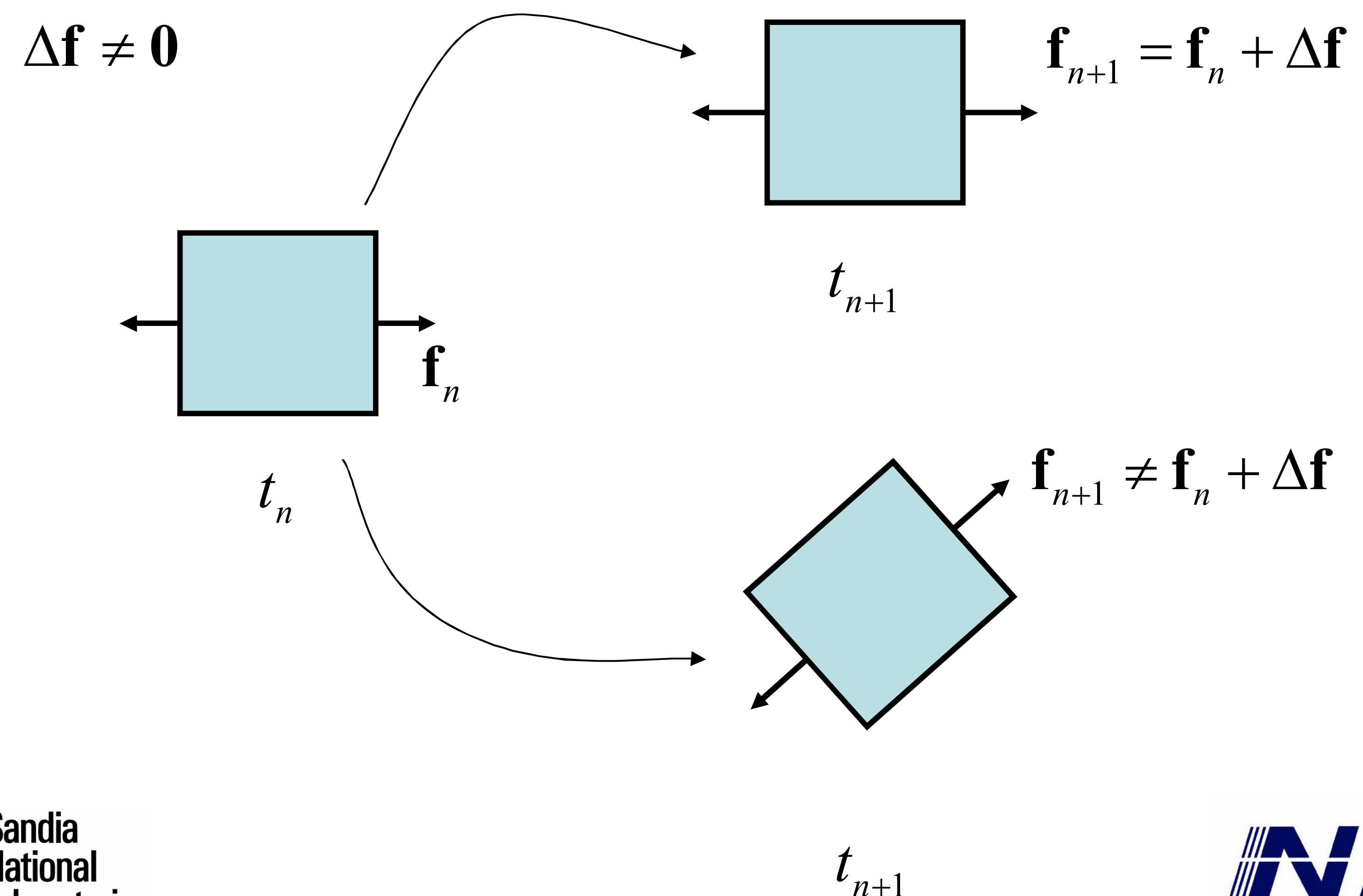
- Midstep Rate of Deformation – Incremental Objectivity
  - Hughes and Winget
  - Calculate rate of deformation with α=1/2

$$\mathbf{x}_{n+1/2} = \mathbf{x}_n + \frac{1}{2}\Delta\mathbf{x}$$

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}_{n+1/2}}$$



Kinematics

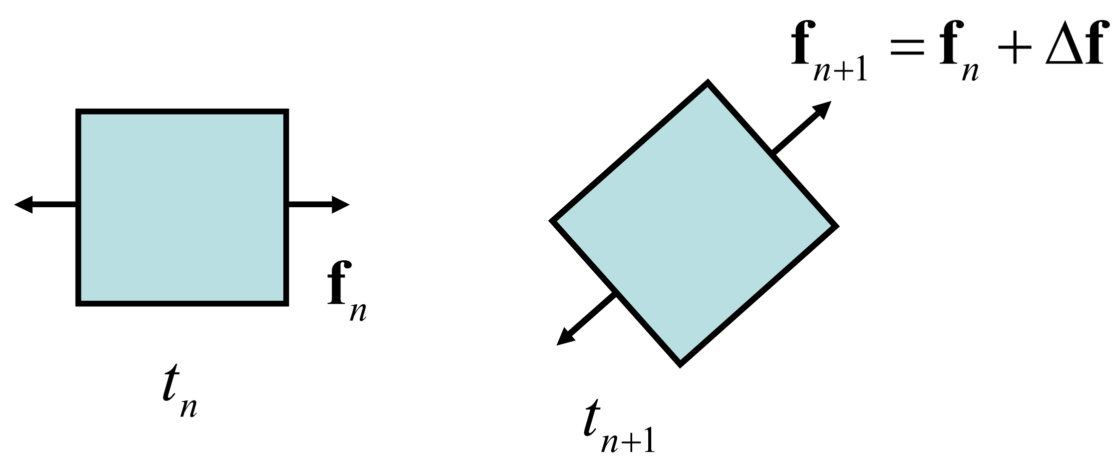

$\Delta \mathbf{f} \neq 0$
$\mathbf{f}_n$
t_n
$\mathbf{f}_{n+1} = \mathbf{f}_n + \Delta \mathbf{f}$
t_{n+1}
$\mathbf{f}_{n+1} \neq \mathbf{f}_n + \Delta \mathbf{f}$
t_{n+1}

Kinematics

- Strong Incremental Objectivity
  - Rashid
  - Calculate rate of deformation based on logarithmic strain increment

$$\hat{\mathbf{F}} = \hat{\mathbf{R}} \cdot \hat{\mathbf{U}}$$

$$\mathbf{D} = \frac{1}{\Delta t} \ln \hat{\mathbf{U}}$$



Kinematics

- Numerical Implementation
  - Tried a lot of things – all approximate
  - Spectral Decomposition – eigenvalues and eigenvectors

$$\hat{\mathbf{U}} = \sum_{i=1}^{3} \lambda_i \overline{\mathbf{e}}_i \overline{\mathbf{e}}_i$$

$$\mathbf{D} = \frac{1}{\Delta t} \ln \hat{\mathbf{U}} = \frac{1}{\Delta t} \sum_{i=1}^{3} \ln \lambda_i \overline{\mathbf{e}}_i \overline{\mathbf{e}}_i$$

Kinematics

- Numerical Implementation
  - Calculate current gradient operator

B_{iI}^{n+1} discrete gradient operator (also used for divergence)

  - Calculate inverse incremental deformation gradient

$$\hat{F}_{ij}^{-1} = \frac{\partial x_n}{\partial x_{n+1}} = \delta_{ij} - \hat{u}_{iI} B_{jI}^{n+1}$$

$$\hat{u}_{iI} = \Delta t v_{iI}$$

Kinematics

- Numerical Implementation
  - Form inverse incremental right Cauchy-Green tensor

$$\hat{C}_{ij}^{-1} = \hat{F}_{ik}^{-1}\hat{F}_{jk}^{-1}$$

  - **Find eigenvalues and eigenvectors**

$$\hat{\mathbf{C}}^{-1} = \sum_{i=1}^{3} \xi_i \overline{\mathbf{e}}_i \overline{\mathbf{e}}_i$$

$$\hat{\mathbf{C}}^{-1} = \hat{\mathbf{U}}^{-2} \quad ; \quad \hat{\mathbf{U}} = \sum_{i=1}^{3} \lambda_i \overline{\mathbf{e}}_i \overline{\mathbf{e}}_i$$

$$\xi_i = \lambda_i^{-2}$$

Kinematics

- Numerical Implementation
  - Calculate rate of deformation

$$\mathbf{D} = -\frac{1}{2\Delta t}\sum_{i=1}^{3} \ln \xi_i \overline{\mathbf{e}}_i \overline{\mathbf{e}}_i$$

- How do we perform the spectral decomposition?

Eigenvalue/Eigenvector Solver

- Basic approach is in Malvern
- Special case needs special attention
  - 2 nearly identical eigenvalues

- Subtle problem
  - It is NOT an problem with large deformations
  - It is often a problem with small deformations

$$\varepsilon_i \approx \varepsilon_j \quad i \neq j$$

$$\lambda_i \approx \lambda_j$$

- If not handled properly it leads to convergence problems
  - Bulletproof algorithm

Sandia National Laboratories

National Nuclear Security Administration

Eigenvalue/Eigenvector Solver

- Symmetric 3x3 Matrix

$$[\mathbf{A}] = \begin{bmatrix} A_{11} & A_{12} & A_{31} \\ A_{12} & A_{22} & A_{23} \\ A_{31} & A_{23} & A_{33} \end{bmatrix}$$

  - Can be used to represent right stretch, left stretch, right Cauchy-Green, left Cauchy-Green, strain, etc.
  - Can also be used to represent Cauchy stress, Kirchoff stress, Second Piola-Kirchoff stress.

Eigenvalue/Eigenvector Solver

- Eigenvalue problem

$$[\mathbf{A}]\{\mathbf{v}\} = \lambda\{\mathbf{v}\}$$

- Modified eigenvalue problem

$$[\mathbf{A}']\{\mathbf{v}\} = \eta\{\mathbf{v}\}$$

$$[\mathbf{A}'] = [\mathbf{A}] - \frac{1}{3}\left(\mathrm{tr}[\mathbf{A}]\right)[\mathbf{I}]$$

$$\eta = \lambda - \frac{1}{3}\left(\mathrm{tr}[\mathbf{A}]\right)$$

Eigenvectors are the same

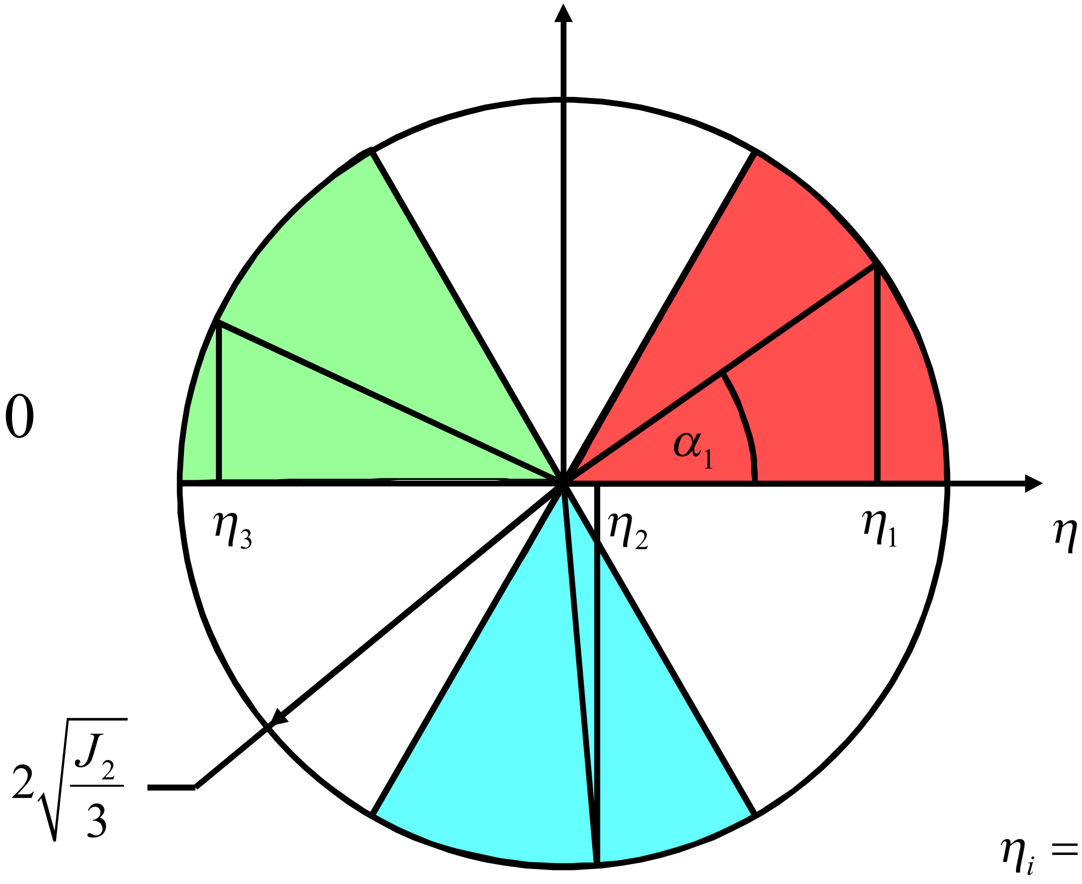
Eigenvalues are shifted

Eigenvalue/Eigenvector Solver

Shifted Eigenvalues

$$\eta_1 \geq \eta_2 \geq \eta_3$$

$$\eta_1 + \eta_2 + \eta_3 = 0$$



$$\eta_i = 2\sqrt{\frac{J_2}{3}} \cos \alpha_i$$

Eigenvalue/Eigenvector Solver

- Two nearly identical eigenvalues

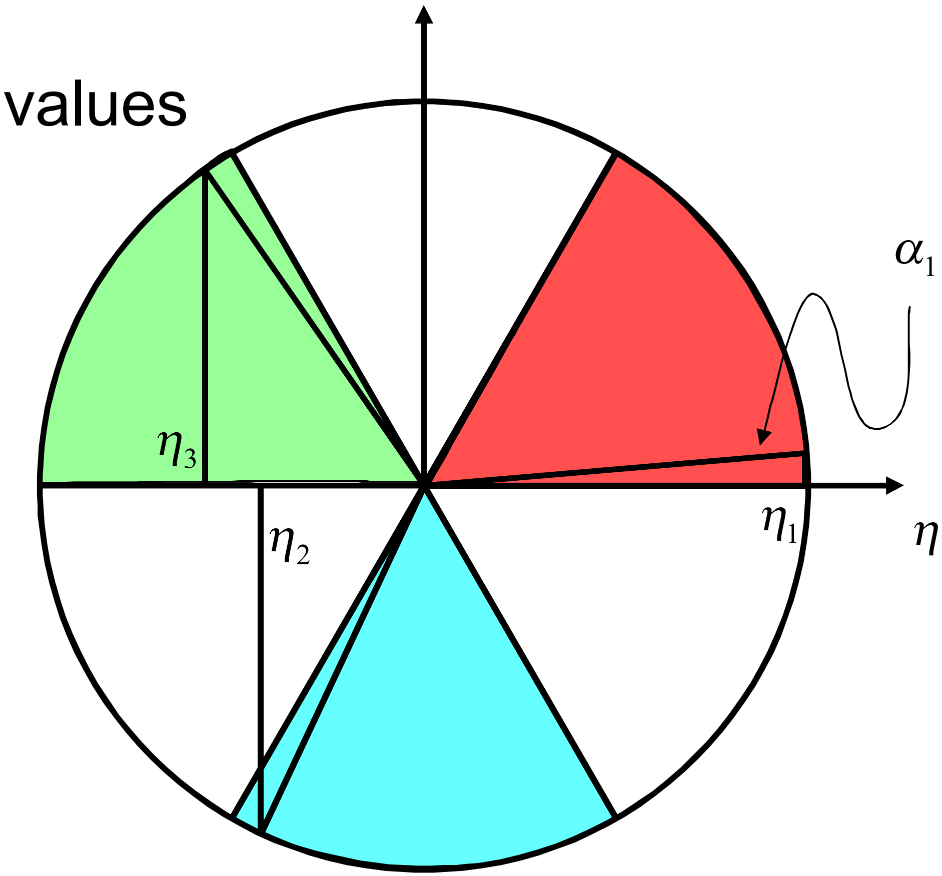
$$\alpha_1 \rightarrow 0$$

$$\eta_2 \rightarrow \eta_3$$

- Alternatively

$$\alpha_1 \rightarrow \frac{\pi}{3}$$

$$\eta_2 \rightarrow \eta_1$$



Eigenvalue/Eigenvector Solver

- Eigenvalues

$$\lambda_1 \quad ; \quad \lambda_2 \quad ; \quad \lambda_3 = \lambda_2 + \Delta\lambda$$

- Deviatoric eigenvalue problem

$$\hat{\eta}_1 = \bar{\eta}\left(1-\varepsilon\right) \quad ; \quad \hat{\eta}_2 = -\frac{1}{2}\bar{\eta}\left(1+2\varepsilon\right) \quad ; \quad \hat{\eta}_2 = -\frac{1}{2}\bar{\eta}\left(1-4\varepsilon\right)$$

$$\bar{\eta} = \frac{2}{3}\left(\lambda_1 - \lambda_2\right) \quad ; \quad \varepsilon = \frac{\Delta\lambda}{3\bar{\eta}}$$

Eigenvalue/Eigenvector Solver

- Asymptotic analysis

$$\cos 3\alpha = \operatorname{sgn}\left(\bar{\eta}\right)\left[1 - \frac{27}{2}\varepsilon^2 - 27\varepsilon^3 + O\left(\varepsilon^4\right)\right]$$

- Two cases:

$$\alpha = 0, \frac{2\pi}{3}, \frac{4\pi}{3} \quad \text{if} \quad \bar{\eta} > 0 \quad \Rightarrow \quad \eta_2 \rightarrow \eta_3$$

$$\alpha = \frac{\pi}{3}, \pi, \frac{5\pi}{3} \quad \text{if} \quad \bar{\eta} < 0 \quad \Rightarrow \quad \eta_2 \rightarrow \eta_1$$

Eigenvalue/Eigenvector Solver

- Case 1: $\bar{\eta} > 0 \quad \left(\lambda_1 > \lambda_2 \;\; ; \;\; \alpha_1 \to 0\right)$

$$\eta_1 = \bar{\eta}\left(1-\varepsilon\right) \quad ; \quad \eta_2 = \eta_3 = -\frac{1}{2}\bar{\eta}\left(1-\varepsilon\right)$$

Good **Bad**

Method can't distinguish between nearly identical eigenvalues

Method can find most distinct eigenvalue

Eigenvalue/Eigenvector Solver

- Case 2: $\bar{\eta} < 0 \quad \left(\lambda_1 < \lambda_2 \;\; ; \;\; \alpha_1 \to \pi/3\right)$

$$\eta_1 = \eta_2 = -\frac{1}{2}\bar{\eta}\left(1-\varepsilon\right) \quad ; \quad \eta_3 = \bar{\eta}\left(1-\varepsilon\right)$$

Bad **Good**

Find most distinct eigenvalue first!

Eigenvalue/Eigenvector Solver

- How do we find most distinct eigenvalue?
  - Solve for α

$$\alpha < \frac{\pi}{6} \quad \Rightarrow \quad \eta_1$$

$$\alpha > \frac{\pi}{6} \quad \Rightarrow \quad \eta_3$$

This algorithm can only find the most distinct eigenvalue accurately

Eigenvalue/Eigenvector Solver

- After most distinct eigenvalue is found...
  - Find eigenvector corresponding to this eigenvalue
  - Reduce matrix to a 2x2

$$\left[\mathbf{A}'\right] = \begin{bmatrix} \eta_1 & 0 & 0 \\ 0 & \bar{A}'_{22} & \bar{A}'_{23} \\ 0 & \bar{A}'_{23} & \bar{A}'_{33} \end{bmatrix}$$

  - Could use quadratic equation, but you will loose accuracy

Eigenvalue/Eigenvector Solver

- Use Wilkinson shift

$$\eta_2 = \bar{A}'_{33} + \frac{\bar{A}'_{22} - \bar{A}'_{33}}{2} - \frac{1}{2}\operatorname{sgn}\left(\bar{A}'_{22} - \bar{A}'_{33}\right)\sqrt{\left(\bar{A}'_{22} - \bar{A}'_{33}\right)^2 - 4\bar{A}'_{23}\bar{A}'_{23}}$$

$$\eta_3 = \bar{A}'_{22} + \bar{A}'_{33} - \eta_2$$

- Asymptotic analysis gives

$$\eta_2 = -\frac{1}{2}\bar{\eta}\left(1-\varepsilon\right) + \frac{3}{2}\bar{\eta}\operatorname{sgn}\left(\varepsilon\right)\varepsilon$$

Eigenvalue/Eigenvector Solver

- Final results

$$\eta_1 = \bar{\eta}\left(1-\varepsilon\right)$$

$$\eta_2 = \frac{\bar{\eta}}{2}\left[\left(1+3\,\mathrm{sgn}\left(\varepsilon\right)\right)\varepsilon - 1\right]$$

$$\eta_3 = -\left(\eta_1 + \eta_2\right)$$

$$\varepsilon = \frac{\Delta\lambda}{2\left(\lambda_1 - \lambda_2\right)}$$

Eigenvalue/Eigenvector Solver

- Final results

$$\varepsilon > 0 \quad \Rightarrow \quad \begin{aligned} \eta_1 &= \bar{\eta}\left(1-\varepsilon\right) \\ \eta_2 &= -\frac{1}{2}\bar{\eta}\left(1-4\varepsilon\right) \\ \eta_3 &= -\frac{1}{2}\bar{\eta}\left(1+2\varepsilon\right) \end{aligned}$$

$$\varepsilon < 0 \quad \Rightarrow \quad \begin{aligned} \eta_1 &= \bar{\eta}\left(1-\varepsilon\right) \\ \eta_2 &= -\frac{1}{2}\bar{\eta}\left(1+2\varepsilon\right) \\ \eta_3 &= -\frac{1}{2}\bar{\eta}\left(1-4\varepsilon\right) \end{aligned}$$

Numerical Examples

- Randomly generated eigenvalues

$$\lambda_1 = 5\left(2m_1 - 1\right)$$

random numbers: $m_1, m_2, m_3 \in \left[0,1\right]$

$$\lambda_2 = 5\left(2m_2 - 1\right)$$

$$\lambda_3 = \lambda_2 + \varepsilon\left(2m_3 - 1\right)$$

ε ← perturbation amplitude

$$\varepsilon \in \left[10^{-15}, 1\right]$$

Numerical Examples

- Randomly generated rotation matrix

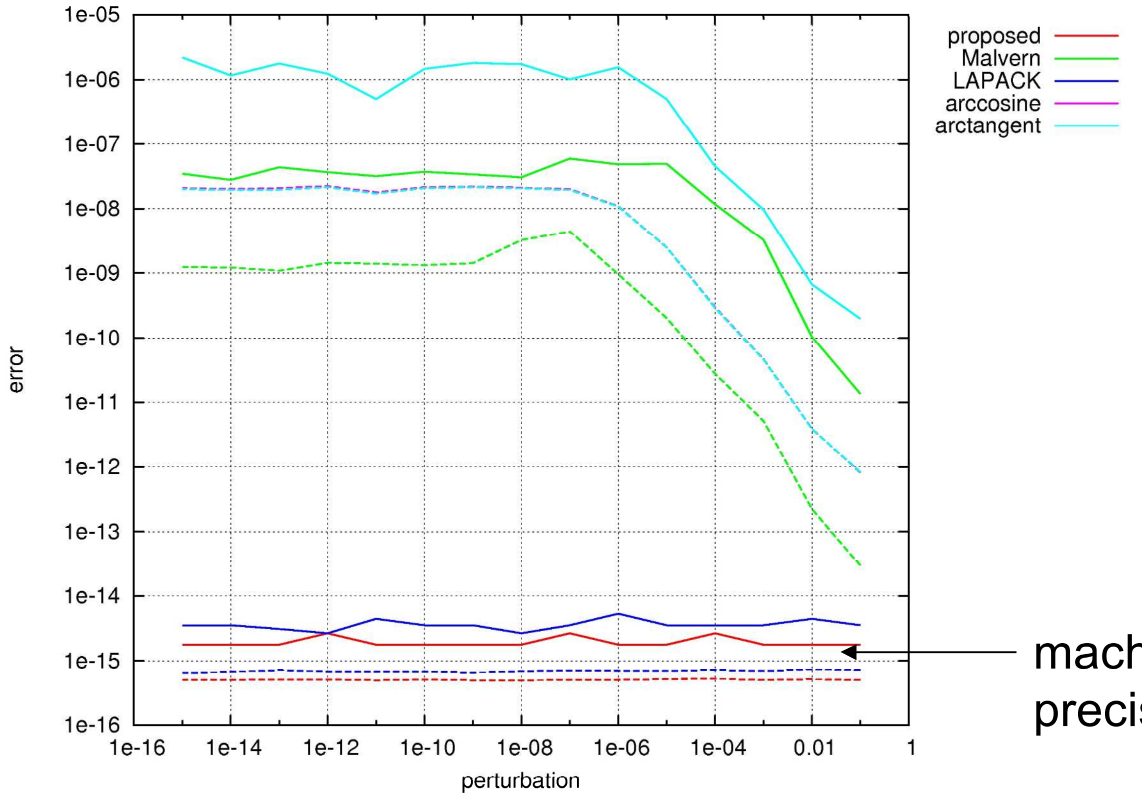
$$\mathbf{A} = \mathbf{Q}^T(\mathbf{n},\theta) \cdot \mathbf{\Lambda} \cdot \mathbf{Q}(\mathbf{n},\theta)$$

axis of rotation and angle of rotation

- Pass $\mathbf{A}$ to eigenvalue/eigenvector routine
  - Check eigenvalues
  - Check eigenvectors (reassemble matrix and check values)
- Do this billions of times
  - Keep track of maximum error

Numerical Examples

Maximum error in eigenvalues – 2 nearly identical eigenvalues



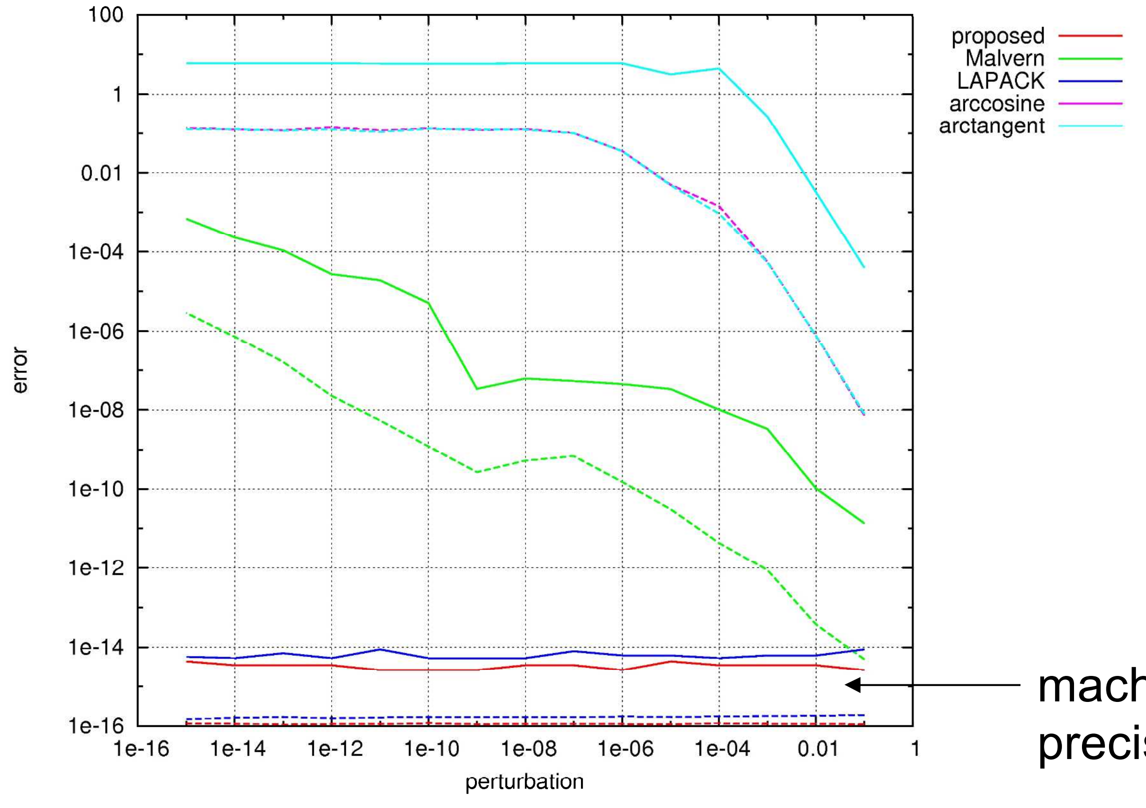
machine precision

Sandia National Laboratories

NNSA National Nuclear Security Administration

Numerical Examples

Maximum error in matrix entries – 2 nearly identical eigenvalues



machine precision

Sandia National Laboratories

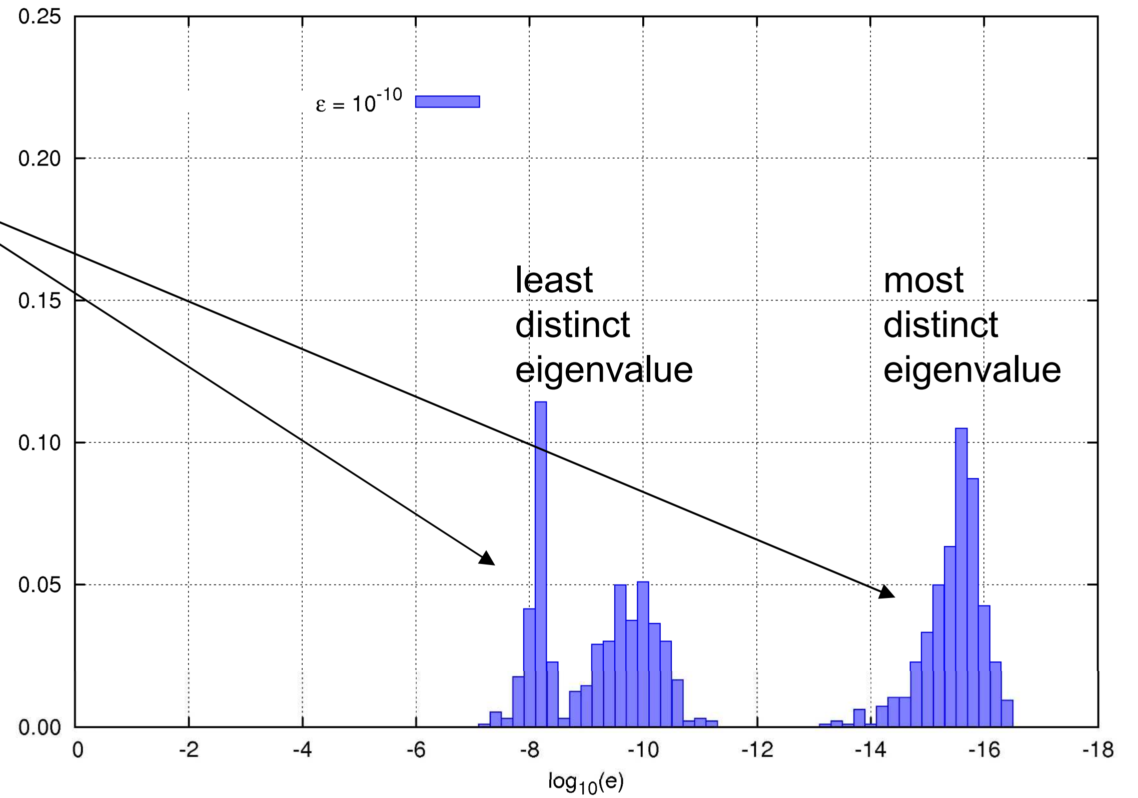
National Nuclear Security Administration

Numerical Examples

Distribution of error in numerical algorithm based on Malvern

$\varepsilon = 10^{-10}$

two distinct regions

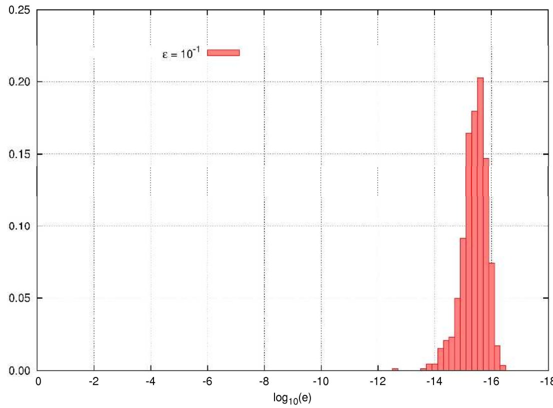


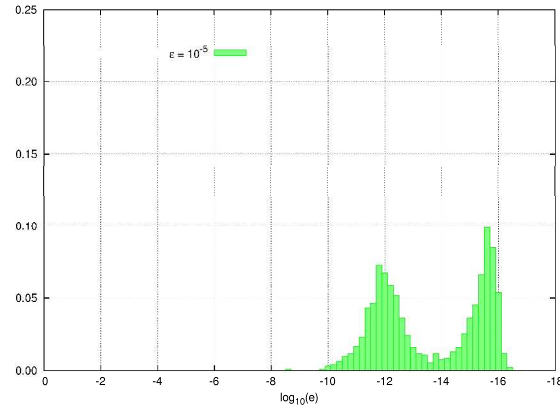
The key to the algorithm is to solve for the most distinct eigenvalue first!

Numerical Examples

Distribution of error in numerical algorithm based on Malvern

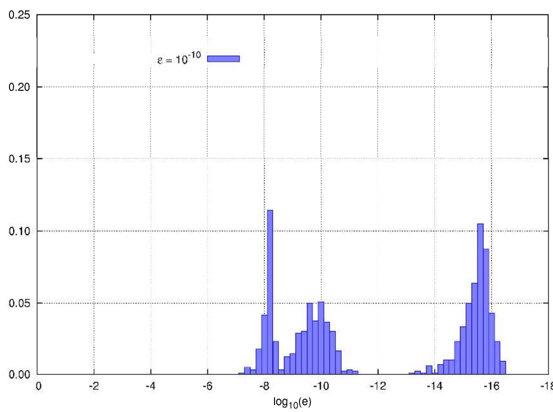
$\varepsilon = 10^{-1}$

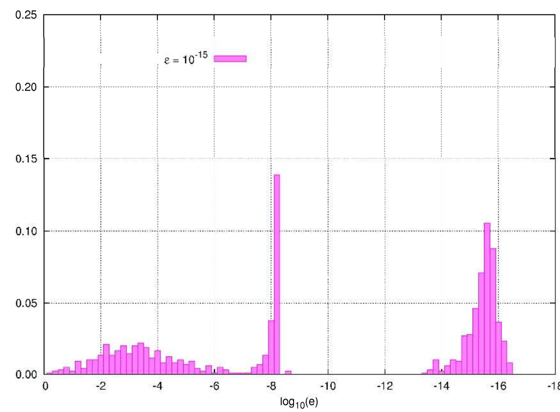




$\varepsilon = 10^{-5}$

$\varepsilon = 10^{-10}$





$\varepsilon = 10^{-15}$

Other Applications

- Polar Decomposition
- Alternative strain measures
- Rate of alternative strain measures (Ogden)

$$\boldsymbol{\varepsilon} = \ln \mathbf{U} \quad ; \quad \dot{\boldsymbol{\varepsilon}} = \frac{1}{2}\frac{\partial \ln \mathbf{C}}{\partial \mathbf{C}} : \dot{\mathbf{C}}$$

- Rate of rotation
- Consistent tangent moduli for any Green-McInnis stress rate?

Conclusions

- An algorithm was presented that accurately solves for eigenvalues and eigenvectors of symmetric 3x3 matrices
- The key to the algorithm is to find the most distinct eigenvalue first
- The algorithm is applicable in general purpose numerical codes
- Applications are abundant in computational mechanics