Sensitivity Analysis for Chaotic, Turbulent Flows

Patrick J. Blonigan

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

8/28/2012

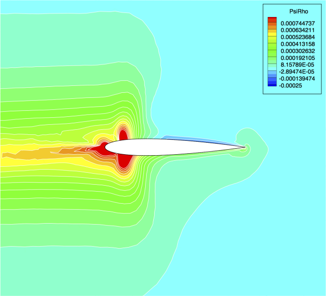
Outline

1. Motivation

2. Chaotic Sensitivity Analysis Issues
   - Overview
   - Kuramoto-Shivashinsky Equation
   - NACA 0012

3. Least Squares Sensitivity Method
   - Overview
   - Multigrid Elimination

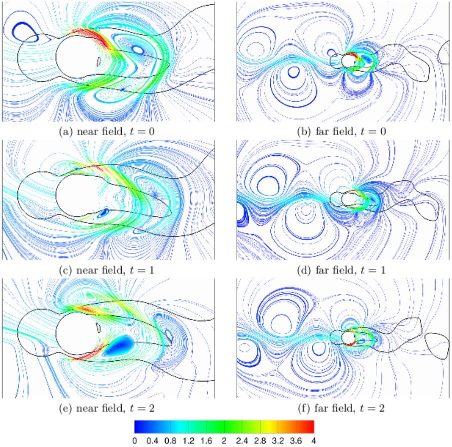
Sensitivity Anaylsis



- Sensitivity Analysis Methods compute derivatives of outputs with respect to inputs.
- With the adjoint, we go backwards in time to find the sensitivity of outputs to inputs.
- The computational cost of the Adjoint method DOES NOT scale with the number of gradients computed.

Adjoint Flow-Field

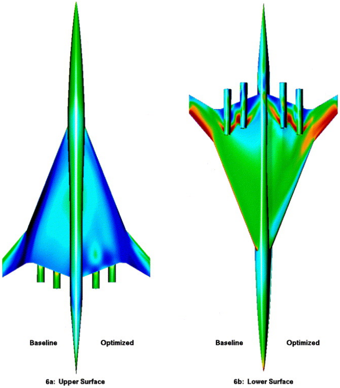
Sensitivities propagate upstream

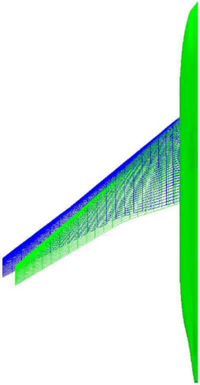


From Wang and Gao, 2012

Sensitivity Analysis Applications

Aerodynamic Shape Optimization

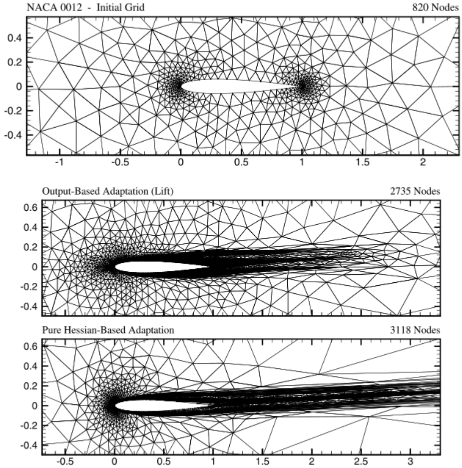




From Jameson 2004

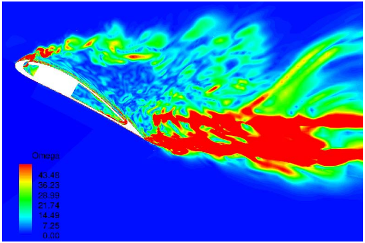
Sensitivity Analysis Applications

Error Estimation and Mesh Adaptation



From Venditti and Darmofal, 2003

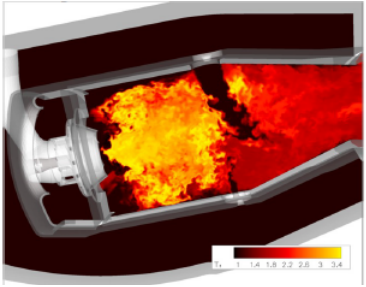
Sensitivity Analysis Applications
Other Applications



From University of Miami CCS

- Flow Control
- Uncertainty Quantification
- and many more...

High Fidelity Model Issue

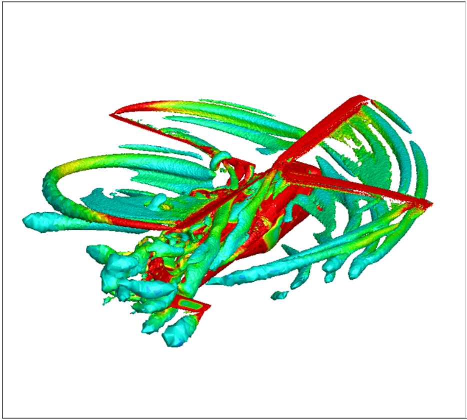


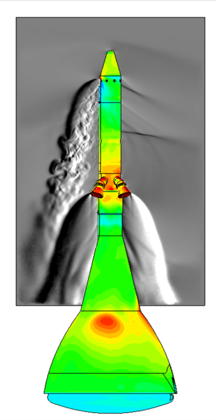
From DOE

- As computers become more powerful, high fidelity turbulence models such as LES will become increasingly popular.
- High fidelity models capture the chaotic nature of turbulent flows.
- However, traditional sensitivity analysis methods break down when applied to chaotic fluid flows.

Chaotic, Turbulent Flow-fields

Unsteady Wakes

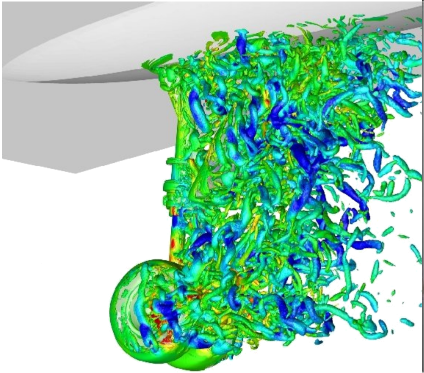


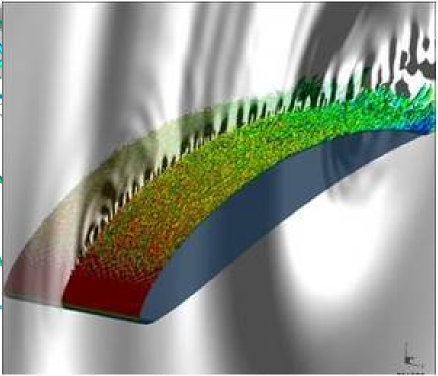


From E. Nielsen

Chaotic, Turbulent Flow-fields

Aeroaccoutics

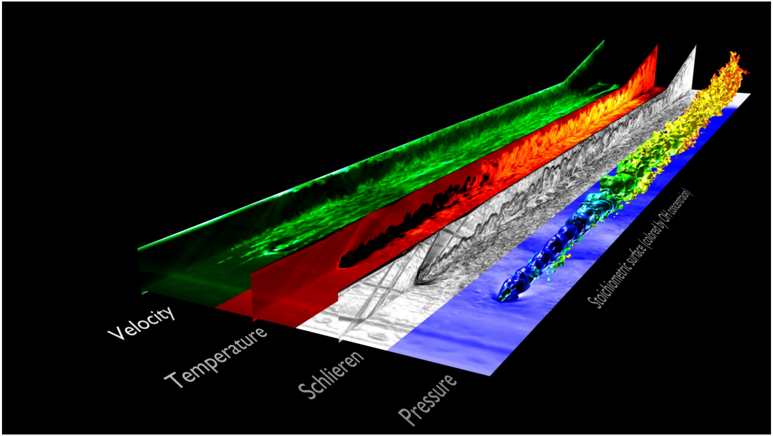




LEFT: From E. Nielsen RIGHT: From TU Berlin

Chaotic, Turbulent Flow-fields

Mixing



From J. Larsson, Stanford University

Traditional Forward Sensitivity Analysis

- Interested in the long time averaged quantity J, governed by a system of equations f with some design parameter(s) s:

$$\bar{J} = \int_0^T J(u,s)dt, \quad \frac{\partial u}{\partial t} = f(u,s)$$

- Solve the tangent equation for $v = \frac{\partial u}{\partial s}$:

$$\frac{\partial v}{\partial t} = \frac{\partial f}{\partial u} v + \frac{\partial f}{\partial s}$$

- Compute the sensitivity of $\bar{J}$ to s:

$$\frac{d\bar{J}^T}{ds} = \int_0^T \frac{\partial J}{\partial u} v + \frac{\partial J}{\partial s} dt$$

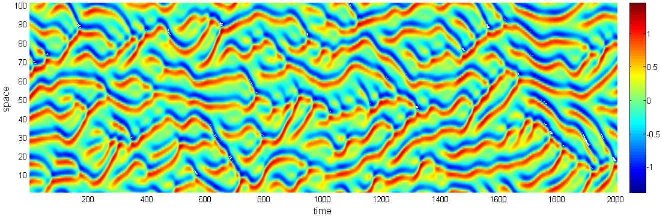
Main Issue

- For chaotic systems, this does not work, because:

$$\frac{d\bar{J}^{\infty}}{ds} \neq \lim_{T\to\infty} \frac{d\bar{J}^{T}}{ds}$$

- This is because the tangent solution v diverges for chaotic systems. Counter-intuitively, increasing T can exacerbate this divergence.
- Adjoint sensitivity analysis breaks down for a similar reason.
- This property of chaotic systems has been shown by Lea et al. for the Lorenz Attractor.
- This problem exists for chaotic PDEs as well.

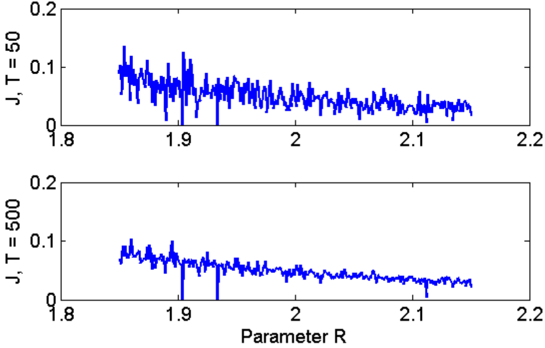
Chaotic KS Equation Solution



$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{1}{R}\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$$

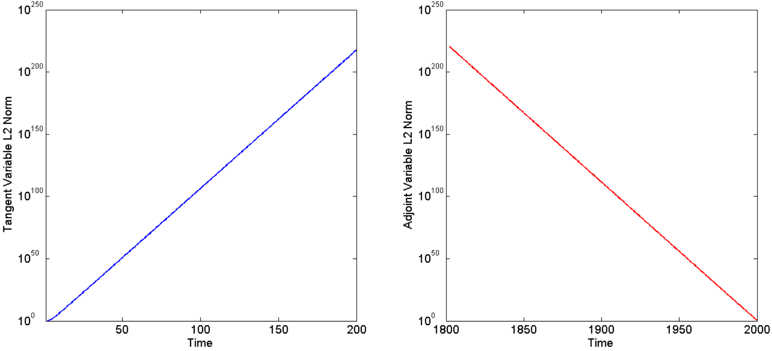
- $R = 2.0$ for Chaos in space and time.

Objective Function



$$\bar{J}^T = \frac{1}{T}\int_0^T \frac{\partial u}{\partial t} dt$$

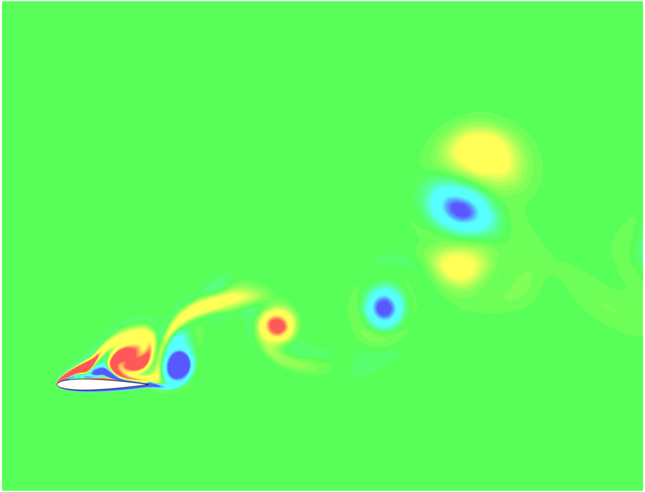
Tangent and Adjoint Solutions



- Both Tangent and Adjoint solutions diverge exponentially for the KS equation.

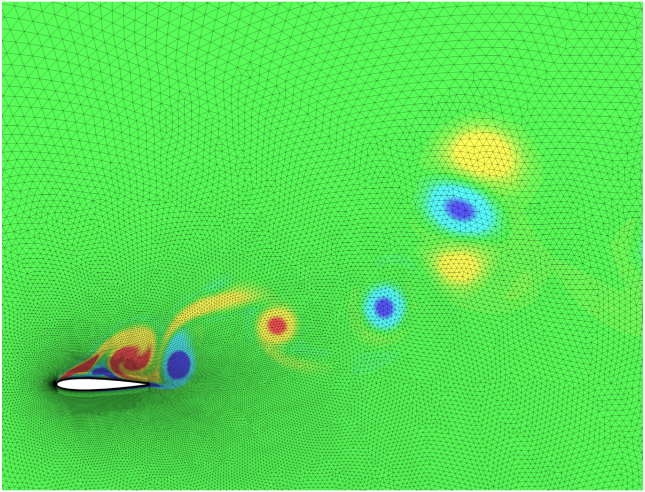
NACA 0012 Airfoil Vorticity Contours

Mach 0.1, Angle of Attack 20°, $Re = 10000$



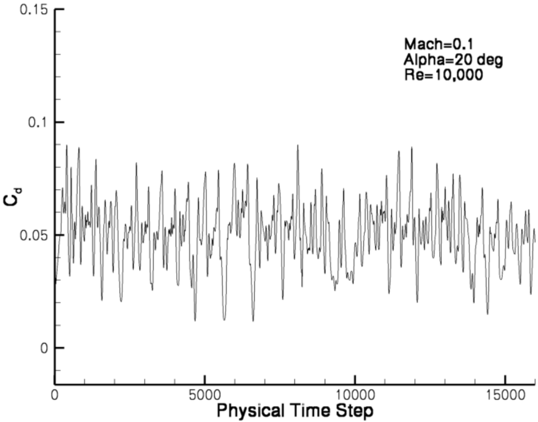
NACA 0012 Airfoil Vorticity Contours

Mach 0.1, Angle of Attack 20°, $Re = 10000$

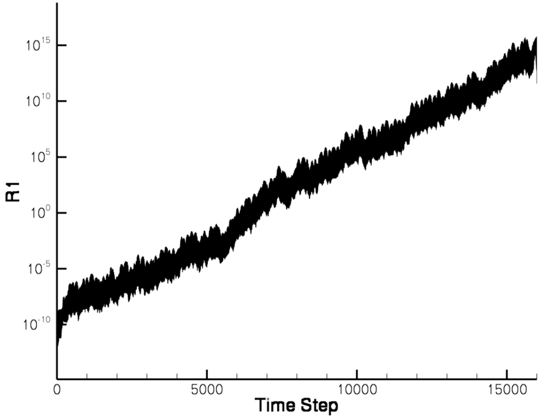


Drag Coefficient Time History

Aperiodicity indicates that the flow is chaotic



Adjoint Residual L2 Norm



Least Squares Sensitivity Method: The Basics

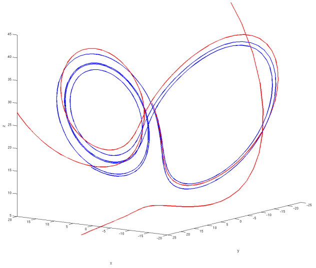
- A chaotic system has at least three different modes.
  - An unstable mode, associated with a positive Lyapunov Exponent.
  - A stable mode, associated with a negative Lyapunov Exponent.
  - A neutrally stable mode, associated with a zero Lyapunov Exponent.
- The unstable mode is responsible for the divergence of the tangent and adjoint equations.

Least Squares Sensitivity Method: The Basics (cont'd)

- Solve for stable modes forwards in time and solve unstable modes backward in time to prevent divergence of tangent and adjoint solutions.
- This solution is called the "Shadow Trajectory" (Wang, 2012) and is the least divergent tangent solution.
- Find the shadow trajectory by solving the following linearly contrained, least squares problem:

$$\min_{\eta, v(t), 0<t<T} \|v\|_2, \quad s.t. \quad \frac{\partial v}{\partial t} = \frac{\partial f}{\partial u} v + \frac{\partial f}{\partial d} + \eta f\,, \quad 0 < t < T$$

Shadow Trajectory



- Divergent and Shadow trajectories for the Lorenz Attractor.

LSS as a "Black Box"

$$\bar{J} = \int_0^T J(u,s)dt, \quad \frac{\partial u}{\partial t} = f(u,s)$$

Inputs:

- Forward Solution: u_i
- Design Variable(s): s
- Operator values f_i
- Operator design parameter sensitivity $\frac{\partial f}{\partial s}_i$
- Objective Function Sensitivity: $\frac{\partial J}{\partial u}_i$
- Objective Function Sensitivity: $\frac{\partial J}{\partial s}_i$
- Jacobian matricies: $\frac{\partial f_i}{\partial u_i}$

Ouputs:

- Sensitivities: $\frac{d\bar{J}}{ds}$

KKT System

$$
\left(\begin{array}{ccccc|c|cccc}
I & & & & & & F_0^T & & & \\
& I & & & & & G_1^T & F_1^T & & \\
& & \ddots & & & & & G_2^T & \ddots & \\
& & & I & & & & & \ddots & F_{m-1}^T \\
& & & & I & & & & & G_m^T \\ \hline
& & & & & & f_1^T & f_2^T & \dots & f_m^T \\ \hline
F_0 & G_1 & & & & f_1 & & & & \\
& F_1 & G_2 & & & f_2 & & & & \\
& & \ddots & \ddots & & \vdots & & & & \\
& & & F_{m-1} & G_m & f_m & & & &
\end{array}\right)
\left(\begin{array}{c}
v_0 \\ v_1 \\ \vdots \\ \vdots \\ v_m \\ \hline \eta \\ \hline \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m
\end{array}\right)
= -
\left(\begin{array}{c}
0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \hline 0 \\ \hline b_1 \\ b_2 \\ \vdots \\ b_m
\end{array}\right)
$$

$$
F_i = \frac{I}{\Delta t}, \quad G_i = -\frac{I}{\Delta t} + \frac{\partial f}{\partial u}(u_n, s), \quad b_i = \frac{\partial f}{\partial d}(u_i, s), \quad f_i = f(u_i), \quad i = 0, \dots, m
$$

- The KKT matrix is a large, symmetric block matrix, where each block is n by n for an n state system. Total size is $2mn + n + 1$ by $2mn + n + 1$ for m time steps. For a discretization with a five element stencil, there are approximately $23mn$ non-zero elements in the matrix.
- For the Airfoil simulation shown earlier the KKT matrix would be 3.2×10^9 by 3.2×10^9 with 3.7×10^{10} non-zero elements.

Multigrid Elimination

- New method to reduce memory usage when solving the LSS KKT system.
- Gaussian Elimination conducted like 1D Multigrid.
- Eliminate every 2nd equation, reduce the system from $2mn + n + 1$ to $n + 1$ equations.
- No need to save coefficients on every grid.
- Potentially Parallelizable.
- Method can be used to solve any unsteady system and its adjoint simultaneously.

Schur Complement

- The KKT matrix is symmetric indefinite, so it becomes singular on coarser grids due to poor scaling.
- Instead, conduct ME on the KKT system's Schur Complement, which is SPD (ignoring the constraint equation).
- Original System:

$$\begin{bmatrix} I & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} v \\ \lambda \end{bmatrix} = - \begin{bmatrix} 0 \\ b \end{bmatrix}$$

- Schur Complement:

$$BB^T \lambda = b$$

Elimination Scheme

Fine Grid Equations

$$L_{i-1}\lambda_{i-2} + D_{i-1}\lambda_{i-1} + U_{i-1}\lambda_i + f_{i-1}\eta = b_{i-1} \tag{1}$$

$$L_i\lambda_{i-1} + D_i\lambda_i + U_i\lambda_{i+1} + f_i\eta = b_i \tag{2}$$

$$L_{i+1}\lambda_i + D_{i+1}\lambda_{i+1} + U_{i+1}\lambda_{i+2} + f_{i+1}\eta = b_{i+1} \tag{3}$$

Elimination Scheme

Coarse Grid Equation

$$L_I \lambda_{i-2} + D_I \lambda_i + U_I \lambda_{i+2} + f_I \eta = b_I$$

Where:

$$\begin{aligned}
L_I &= -L_i D_{i-1}^{-1} L_{i-1} \\
D_I &= -L_i D_{i-1}^{-1} U_{i-1} + D_i - U_i D_{i+1}^{-1} L_{i+1} \\
U_I &= -U_i D_{i+1}^{-1} U_{i+1} \\
f_I &= -L_i D_{i-1}^{-1} f_{i-1} + f_i - U_i D_{i+1}^{-1} f_{i+1} \\
b_I &= -L_i D_{i-1}^{-1} b_{i-1} + b_i - U_i D_{i+1}^{-1} b_{i+1}
\end{aligned}$$

- A similar method is used to restrict the constraint equation and the equation for the long-time averaged gradient of interest.

LSS and ME applied to the Lorenz Equations

- Lorenz Equations:

$$\frac{dx}{dt} = s(y - x), \quad \frac{dy}{dt} = x(r - z) - y, \quad \frac{dz}{dt} = xy - bz$$

- Long time averaged z gradients computed by LSS/ME:

$$\frac{d\bar{z}}{ds} = 0.1545, \quad \frac{d\bar{z}}{dr} = 0.9709, \quad \frac{d\bar{z}}{db} = -1.8014$$

- Gradients computed by finite difference/linear regression:

$$\frac{d\bar{z}}{ds} = 0.16 \pm 0.02, \quad \frac{d\bar{z}}{dr} = 1.01 \pm 0.04, \quad \frac{d\bar{z}}{db} = -1.68 \pm 0.15$$

Avoiding inverting Matrices

- ME can be implemented so that no Jacobian matrices need to be inverted
- Consider the following system:

$$\begin{pmatrix} D_1 & U_1 & 0 \\ L_2 & D_2 & U_2 \\ 0 & L_3 & D_3 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

- The system is restricted using ME:

$$A\lambda_2 = b$$

with:

$$\begin{aligned} A =& \quad -L_2 D_1^{-1} U_1 + D_2 - U_2 D_3^{-1} L_3 \\ b =& \quad -L_2 D_1^{-1} b_1 + b_2 - U_2 D_3^{-1} b_3 \end{aligned}$$

Avoiding inverting Matrices (cont'd)

- This system can be solved iteratively, using some preconditioner P:

$$P\Delta x = b - Ax_k, \quad x_{k+1} = x_k + \Delta x$$

  Where x_k is the value of λ_2 after k iterations.

- Decompose Ax_k into three parts:

$$Ax_k = -L_2 D_1^{-1} U_1 x_k + D_2 x_k - U_2 D_3^{-1} L_3 x_k = \alpha + \beta + \gamma$$

Avoiding inverting Matrices (cont'd)

- Consider α:

$$-L_2 D_1^{-1} U_1 x_k = \alpha$$

- Compute $y_k = U_1 x_k$:

$$-L_2 D_1^{-1} y_k = \alpha$$

- Next, define $z_k = D_1^{-1} y_k$. Itertively solve:

$$D_1 z_k = y_k$$

- Use the result to compute α:

$$\alpha = -L_2 z_k$$

- This idea can be applied to a much larger system and allows ME to be conducted without inverting any Jacobian matricies.

Conclusion

- Traditional sensitivity analysis methods are unable to compute sensitivities of long-time averaged quantities in CFD simulations.
- The LSS method could compute these quantities in an efficient manner if applied with Multigrid elimination.

- Future Work
  - Further develop and implement ME without inverting Jacobians, ideally in C, C++ or Fortran.
  - Apply LSS/ME to the KS equation.
  - Validate LSS on aerodynamic test cases.

Acknowledgements

- I would like the acknowledge the following people for their guidance and support this summer:
  - Boris Diskin
  - Eric Nielsen
  - NASA LaRC Computational AeroSciences Branch
  - LARSS Program
  - The Center for Turbulence Research
  - Johan Larsson
  - Qiqi Wang