

# Application of Improved Truncation Error Estimation Techniques to Adjoint Based Error Estimation and Grid Adaptation

**Joseph M. Derlaga**

*Aerospace and Ocean Engineering Dept.  
Virginia Tech*

**AMS Seminar, July 9, 2015**

**PhD Defense, July 1, 2015**

# Introduction

**Solutions obtained via Computational Fluid Dynamics (CFD) are increasingly trusted as the ‘true’ solution to a fluid dynamics problem, despite the solution being subject to a variety of modeling and numerical errors.**

- **Of these numerical errors, discretization error (DE) is often the largest, and most difficult to properly estimate.**
- **DE is caused by the need to discretize and truncate the governing equations of interest, and this process results in the so called truncation error (TE).**
- **Different methods of estimating DE are available, but most are dependent on proper TE estimates.**
- **What is the most appropriate TE estimation procedure and how do the DE estimation procedures compare?**

# Outline

- **Background/Review**
  - TE and DE
  - Local DE estimation procedures
  - Adjoint based functional DE estimation
  - TE estimation procedures
- **Quasi-1D Euler results**
- **SENSEI CFD solver**
- **2D Euler results**
- **Future work and contributions**

# Review of TE and DE

## Truncation Error: $\tau_h(u)$

is the difference between the discrete *equations* and the governing *equations* and can be expressed using the Generalized Truncation Error Expression (GTEE), see Roy (2009)

$$L_h(u) = L(u) + \tau_h(u)$$

where  $L_h(u_h) = 0$

- $L_h(\cdot)$  is the discretized equation
- $u_h$  is the numerical solution to the discretized equation

and  $L(\tilde{u}) = 0$

- $L(\cdot)$  is the governing equation
- $\tilde{u}$  is the exact solution to the governing equation

# Review of TE and DE

## Discretization Error: $\varepsilon_h$

is the difference between the exact *solution* to the discrete equations,  $u_h$ , and the exact *solution* to the governing equations,  $\tilde{u}$

$$\varepsilon_h = u_h - \tilde{u}$$

If the governing equations are linear ( or linearized ), and with some algebraic manipulation, we can show that the GTEE becomes:

$$L(\varepsilon_h) = -\tau_h(I_h^q u_h)$$

# Review of TE and DE

## Discretization Error: $\varepsilon_h$

is the difference between the exact *solution* to the discrete equations,  $u_h$ , and the exact *solution* to the governing equations,  $\tilde{u}$

$$\varepsilon_h = u_h - \tilde{u}$$

If the governing equations are linear ( or linearized ), and with some algebraic manipulation, we can show that the GTEE becomes:

$$L(\varepsilon_h) = -\tau_h(I_h^q u_h)$$

Or discretely:

$$L_h(\varepsilon_h) = \tau_h(\tilde{u})$$

Therefore, truncation error acts as the local source of discretization error.

# DE Estimation

## Defect Correction (DC)

- The truncation error, or its estimate, can be used as a source term to drive the numerical solution towards the exact solution of the exact governing equations

$$L_h(u_{h,H0}) = \tau_h(I_h^q u_h)$$

## Defect Correction (DC)

- The truncation error, or its estimate, can be used as a source term to drive the numerical solution towards the exact solution of the exact governing equations

$$L_h(u_{h,H0}) = \tau_h(I_h^q u_h)$$

- **Pros:**
  - Extremely easy to implement in any flow solver that can accept a source terms
- **Cons:**
  - Need to solve a nonlinear system, but good initial conditions are generally available
- See the works of Fox (1947), Pereyra (1967/1968), and Phillips (2014)

# DE Estimation

## Error Transport Equations (ETEs)

- Through a Taylor series expansion of the GTEE, we can state:

$$\frac{\partial L_h(u_h)}{\partial u_h} \bar{\varepsilon}_h = -\tau_h(I_h^q u_h) + O(\bar{\varepsilon}_h^2)$$

- This linear system allows us to directly solve for a DE estimate

## Error Transport Equations (ETEs)

- Through a Taylor series expansion of the GTEE, we can state:

$$\frac{\partial L_h(u_h)}{\partial u_h} \bar{\varepsilon}_h = -\tau_h(I_h^q u_h) + O(\bar{\varepsilon}_h^2)$$

- This linear system allows us to directly solve for a DE estimate
- Pros:
  - Linearized problem which is easier to solve compared to DC
- Cons:
  - Need to fully linearize the discrete operator
- See the works of Zhang et al. (2000), Qin et al. (2002/2006), Cavallo et al. (2007/2008), and Phillips (2014)

# Adjoint Methods

## Adjoint methods estimate DE in functional outputs

- Unlike DC and ETEs, which locally estimates the DE, the adjoint method only estimates the error in a single functional output per adjoint/dual solve

# Adjoint Methods

## Adjoint methods estimate DE in functional outputs

- Unlike DC and ETEs, which locally estimates the DE, the adjoint method only estimates the error in a single functional output per adjoint/dual solve
- Pros:
  - Linearized problem which is easier to solve compared to the primal problem
  - Can be used to drive adaptation and design methods
- Cons:
  - Need to fully linearize the discrete (or continuous) operator
  - Only obtain a single DE estimate per dual solve

# Adjoint Methods

## Review of adjoint based error estimation

**Adjoint methods seek to improve a functional estimate by determining its sensitivity to TE perturbations**

# Adjoint Methods

## Review of adjoint based error estimation

Adjoint methods seek to improve a functional estimate by determining its sensitivity to TE perturbations

- Assume we have a Taylor series expansion of the exact governing equation about a general function,  $u$

$$L(\tilde{u}) = L(u) + \left. \frac{\partial L}{\partial U} \right|_{U=u} (\tilde{u} - u) + \left. \frac{\partial^2 L}{\partial U^2} \right|_{U=u} \frac{(\tilde{u} - u)^2}{2} + \dots = 0$$

- And a Taylor series expansion of the functional of interest

$$J(\tilde{u}) = J(u) + \left. \frac{\partial J}{\partial U} \right|_{U=u} (\tilde{u} - u) + \left. \frac{\partial^2 J}{\partial U^2} \right|_{U=u} \frac{(\tilde{u} - u)^2}{2} + \dots$$

## Review of adjoint based error estimation

We can then combine them through Lagrange multipliers (neglecting H.O.T.'s) :

$$J(\tilde{u}) \approx J(u) + \left. \frac{\partial J}{\partial U} \right|_{U=u} (\tilde{u} - u) + \left. \frac{\partial^2 J}{\partial U^2} \right|_{U=u} \frac{(\tilde{u} - u)^2}{2} + \lambda \left[ L(u) + \left. \frac{\partial L}{\partial U} \right|_{U=u} (\tilde{u} - u) + \left. \frac{\partial^2 L}{\partial U^2} \right|_{U=u} \frac{(\tilde{u} - u)^2}{2} \right]$$

# Adjoint Methods

## Review of adjoint based error estimation

We can then combine them through Lagrange multipliers (neglecting H.O.T.'s) :

$$J(\tilde{u}) \approx J(u) + \frac{\partial J}{\partial U} \Big|_{U=u} (\tilde{u} - u) + \frac{\partial^2 J}{\partial U^2} \Big|_{U=u} \frac{(\tilde{u} - u)^2}{2} + \lambda \left[ L(u) + \frac{\partial L}{\partial U} \Big|_{U=u} (\tilde{u} - u) + \frac{\partial^2 L}{\partial U^2} \Big|_{U=u} \frac{(\tilde{u} - u)^2}{2} \right]$$

If we say that  $u$  is a discrete solution prolonged to a  $q^{\text{th}}$  order polynomial,  $I_h^q u_h$ , we have:

$$J(\tilde{u}) \approx J(I_h^q u_h) - \frac{\partial J}{\partial U} \Big|_{I_h^q u_h} \epsilon_h + \frac{\partial^2 J}{\partial U^2} \Big|_{I_h^q u_h} \frac{\epsilon_h^2}{2} + \lambda \left[ L(I_h^q u_h) - \frac{\partial L}{\partial U} \Big|_{I_h^q u_h} \epsilon_h + \frac{\partial^2 L}{\partial U^2} \Big|_{I_h^q u_h} \frac{\epsilon_h^2}{2} \right]$$

# Adjoint Methods

## Review of adjoint based error estimation

We can then combine them through Lagrange multipliers (neglecting H.O.T.'s) :

$$J(\tilde{u}) \approx J(u) + \frac{\partial J}{\partial U} \Big|_{U=u} (\tilde{u} - u) + \frac{\partial^2 J}{\partial U^2} \Big|_{U=u} \frac{(\tilde{u} - u)^2}{2} + \lambda \left[ L(u) + \frac{\partial L}{\partial U} \Big|_{U=u} (\tilde{u} - u) + \frac{\partial^2 L}{\partial U^2} \Big|_{U=u} \frac{(\tilde{u} - u)^2}{2} \right]$$

If we say that  $u$  is a discrete solution prolonged to a  $q^{\text{th}}$  order polynomial,  $I_h^q u_h$ , we have:

$$J(\tilde{u}) \approx J(I_h^q u_h) - \frac{\partial J}{\partial U} \Big|_{I_h^q u_h} \epsilon_h + \frac{\partial^2 J}{\partial U^2} \Big|_{I_h^q u_h} \frac{\epsilon_h^2}{2} + \lambda \left[ L(I_h^q u_h) - \frac{\partial L}{\partial U} \Big|_{I_h^q u_h} \epsilon_h + \frac{\partial^2 L}{\partial U^2} \Big|_{I_h^q u_h} \frac{\epsilon_h^2}{2} \right]$$

By use of the GTEE,  $L(I_h^q u_h) = -\tau_h(I_h^q u_h)$ , we can rewrite the above as:

$$J(\tilde{u}) \approx J(I_h^q u_h) - \lambda \tau_h(I_h^q u_h) - \epsilon_h \left[ \frac{\partial J}{\partial U} \Big|_{I_h^q u_h} + \lambda \frac{\partial L}{\partial U} \Big|_{I_h^q u_h} \right] + \frac{\epsilon_h^2}{2} \left[ \frac{\partial^2 J}{\partial U^2} \Big|_{I_h^q u_h} + \lambda \frac{\partial^2 L}{\partial U^2} \Big|_{I_h^q u_h} \right]$$

# Adjoint Implementation

## **This work uses the discrete adjoint approach**

- **Complex, steady-state, viscous solutions to the Navier-Stokes equations generally require implicit methods to avoid CFL restrictions**
  - **If the Jacobians already exist, you might as well use them to form the discrete adjoint**
- **If a fully implicit formulation is in place, then the boundary condition perturbations will already be included**
- **Ease of implementing new functionals**
  - **Discrete differentiation of functional can be easier than continuous differentiation**

- **Multiple grid approaches**
  - **Discrete residual operates on a finer/coarser grid which has a reconstruction/restriction of the discrete solution**
    - **Venditti & Darmofal (2000/2002/2003) approach used by adjoint community**
    - **Phillips (2013/2014) extended this using a formal order of accuracy argument for both coarse- and fine-grid variations**

- **Multiple grid approaches**
  - Discrete residual operates on a finer/coarser grid which has a reconstruction/restriction of the discrete solution
    - Venditti & Darmofal (2000/2002/2003) approach used by adjoint community
    - Phillips (2013/2014) extended this using a formal order of accuracy argument for both coarse- and fine-grid variations
- **Single grid approaches**
  - Discrete residual approach where the solution is approximated through best fit approximations (Park 2007, Park 2011)
  - Continuous residual approach where the solution is reconstructed and the strong (Giles & Pierce 1999/2000) or weak governing equations are operated upon it (Derlaga 2013, Phillips 2013/2014)

- **Fine-grid method:**

$$\tau_h(I_h u_h) \approx -I_{h/r}^h L_{h/r} (I_q^{h/r} I_h^q u_h) \left( \frac{r^p}{r^p - 1} \right)$$

- **Coarse-grid method:**

$$\tau_h(I_h u_h) \approx I_q^h I_{rh}^q L_{rh} (I_h^{rh} u_h) \left( \frac{1}{r^p - 1} \right)$$

- **Continuous Residual method:**

$$\tau_h(I_h^q u_h) \approx -I^h L(I_h^q u_h)$$

- **Fine-grid method:**

$$\tau_h(I_h u_h) \approx -I_{h/r}^h L_{h/r} (I_q^{h/r} I_h^q u_h) \left( \frac{r^p}{r^p - 1} \right)$$

- **Coarse-grid method:**

$$\tau_h(I_h u_h) \approx I_q^h I_{rh}^q L_{rh} (I_h^{rh} u_h) \left( \frac{1}{r^p - 1} \right)$$

- **Continuous Residual method:**

$$\tau_h(I_h^q u_h) \approx -I^h L(I_h^q u_h)$$

- **Want to satisfy conservation of the mean:**

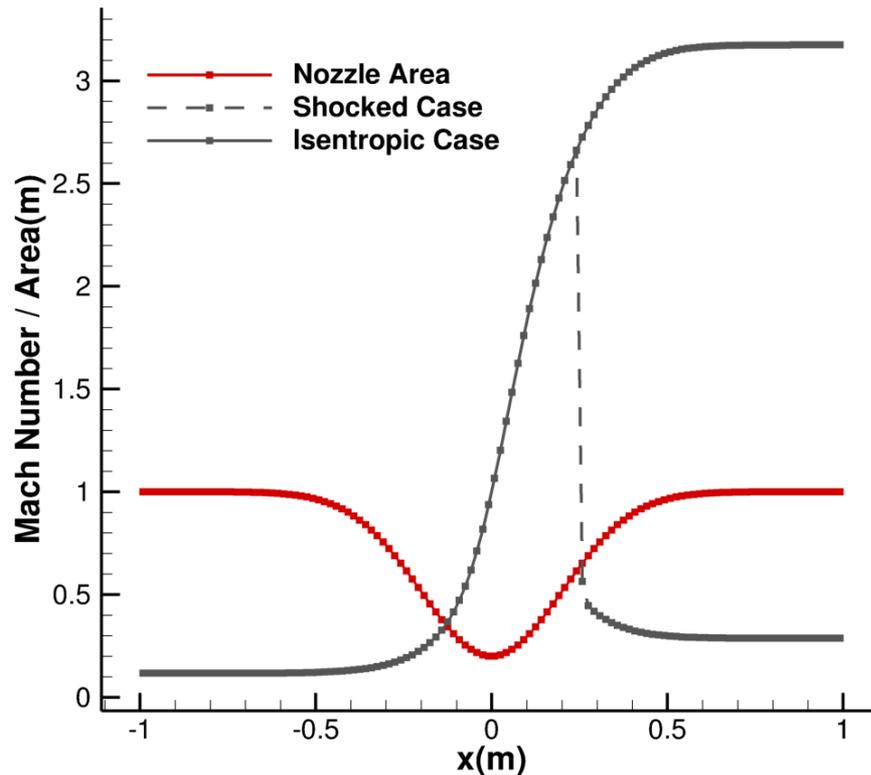
$$u_h = I_q^h I_h^q u_h$$

# Outline

- Background/Review
  - TE and DE
  - Local DE estimation procedures
  - Adjoint based functional DE estimation
  - TE estimation procedures
- **Quasi-1D Euler results**
- **SENSEI CFD solver**
- **2D Euler results**
- **Future work and contributions**

# Quasi-1D Euler Equations

- Special case of 2D Euler flow for finite volume methods
- Examine both isentropic and non-isentropic cases
- Roe's FDS, van Albada limiter, MUSCL with  $\kappa = 1/3$
- Functional is an integral of pressure along the domain

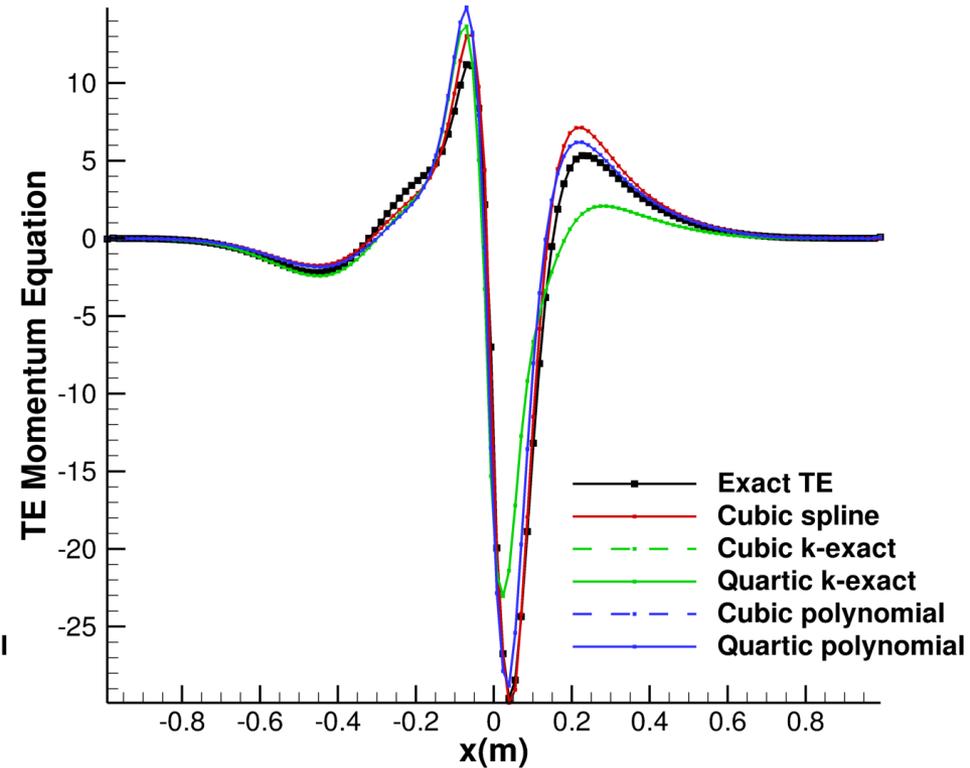
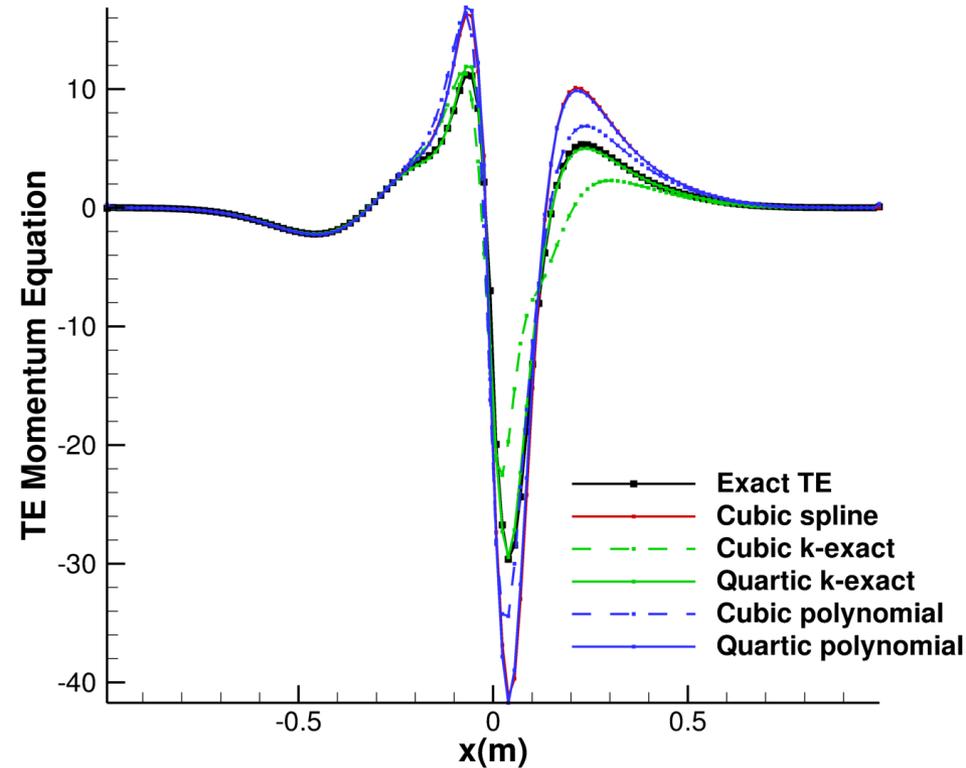


# Quasi-1D Euler Equations

## TE Estimation Comparison

Primitive Variable Reconstruction / Weak Form TE Estimation

Primitive Variable Reconstruction / Embedded Grid TE Estimation

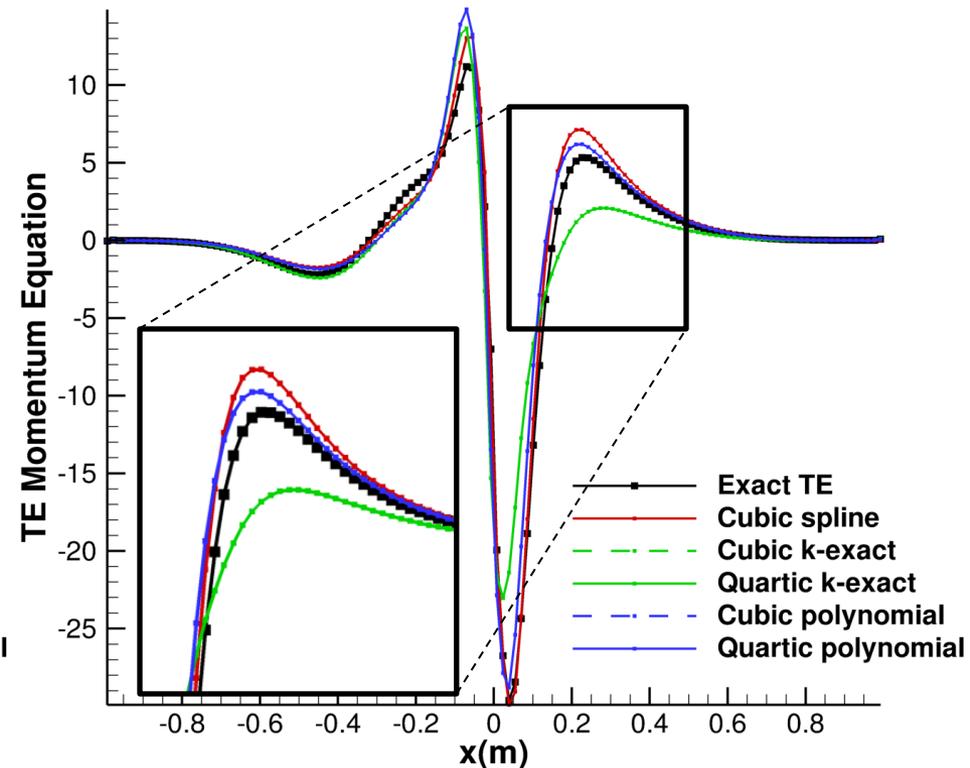
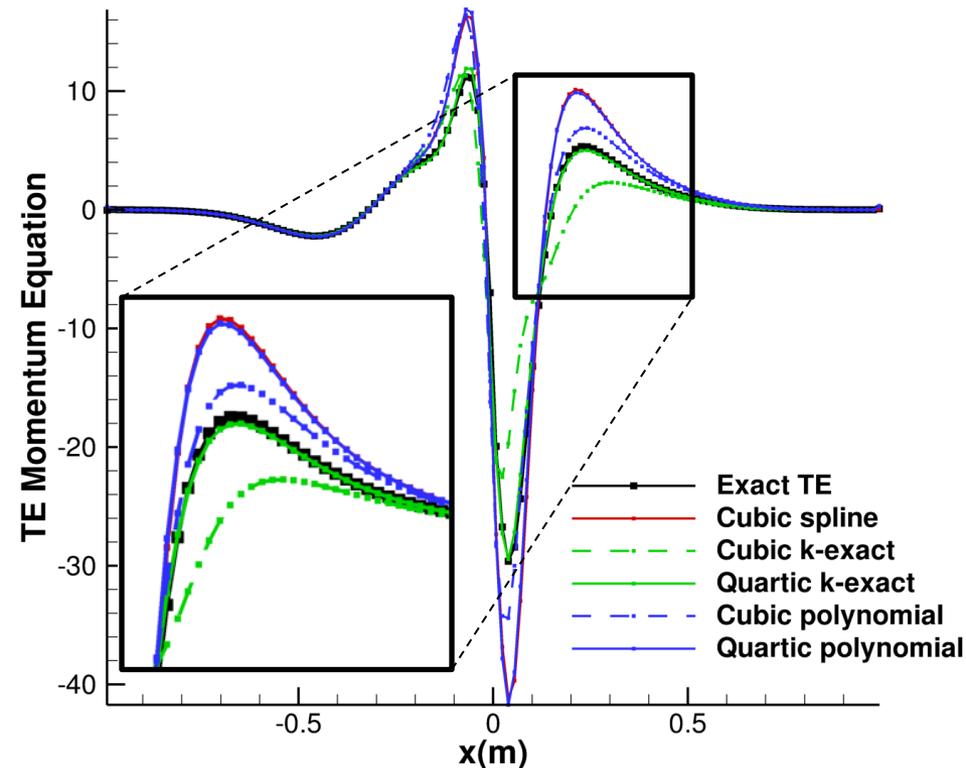


# Quasi-1D Euler Equations

## TE Estimation Comparison

Primitive Variable Reconstruction / Weak Form TE Estimation

Primitive Variable Reconstruction / Embedded Grid TE Estimation

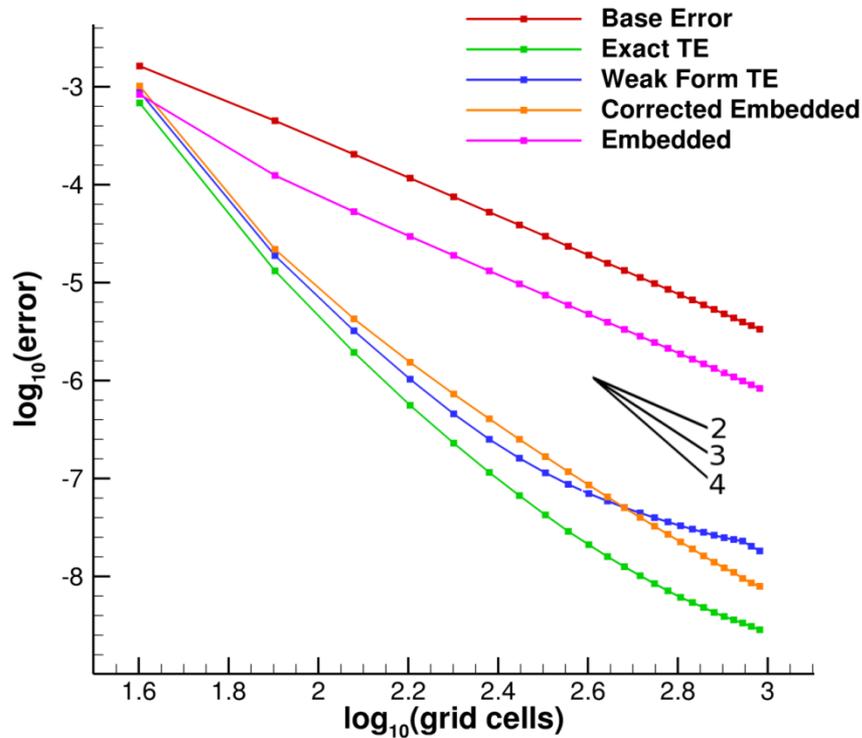


# Quasi-1D Euler Equations

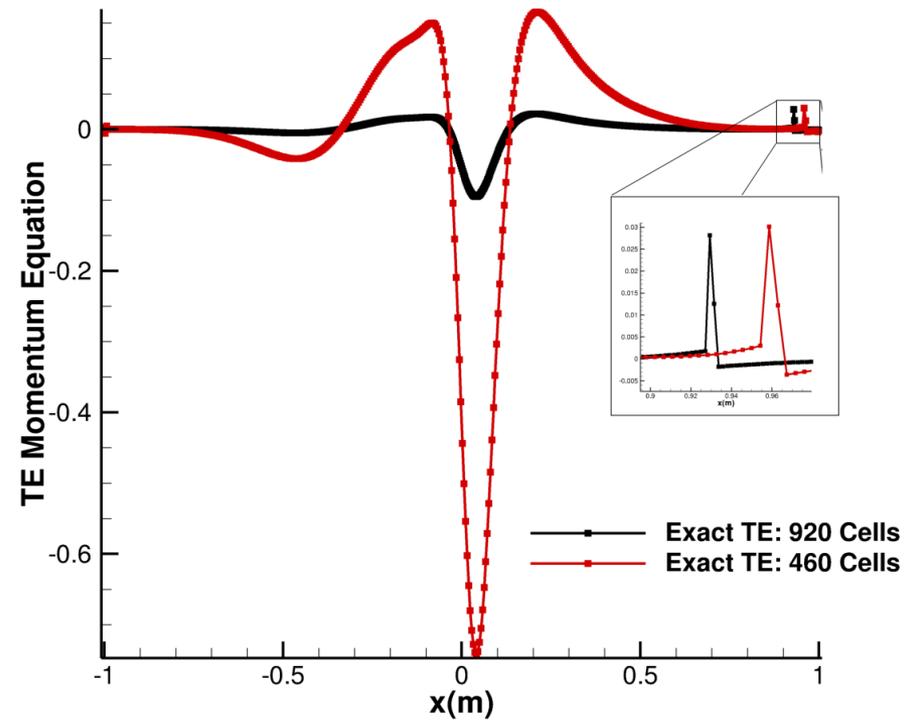


## Isentropic Case

### Base and Remaining DE



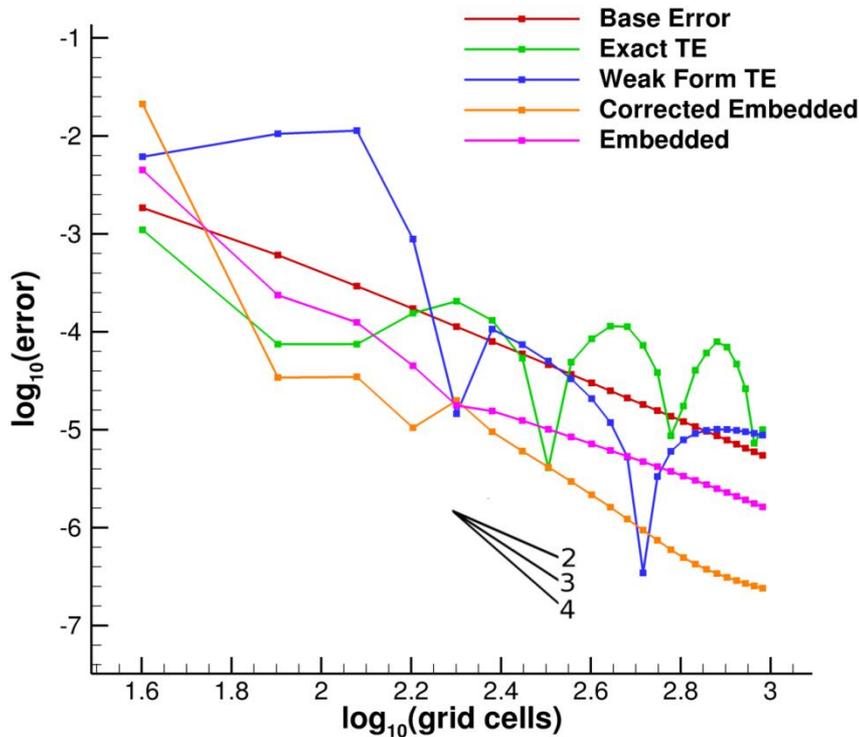
### Truncation Error



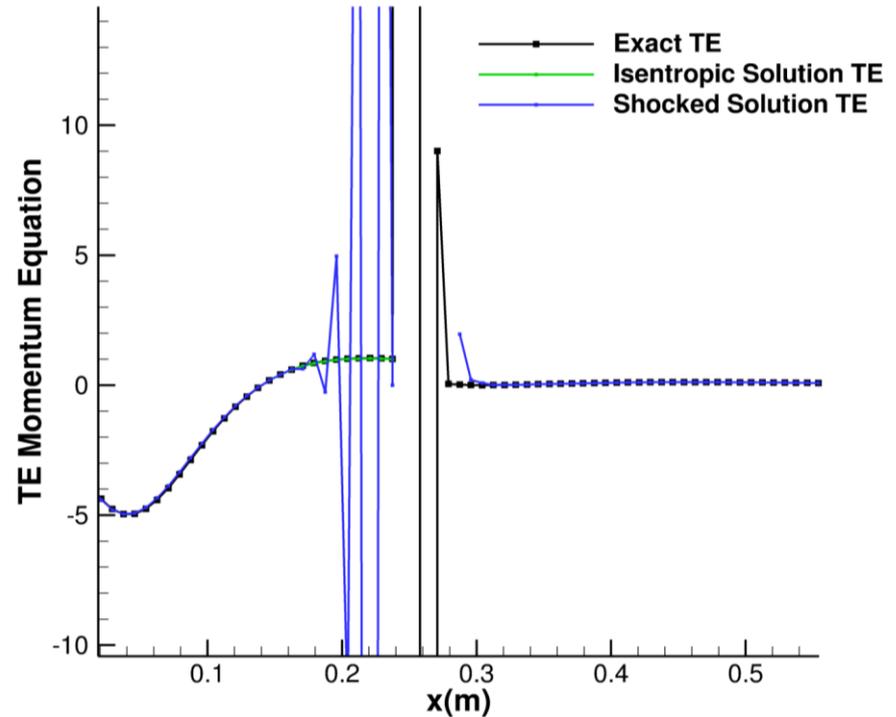
# Quasi-1D Euler Equations

## Non-isentropic Case

### Base and Remaining DE

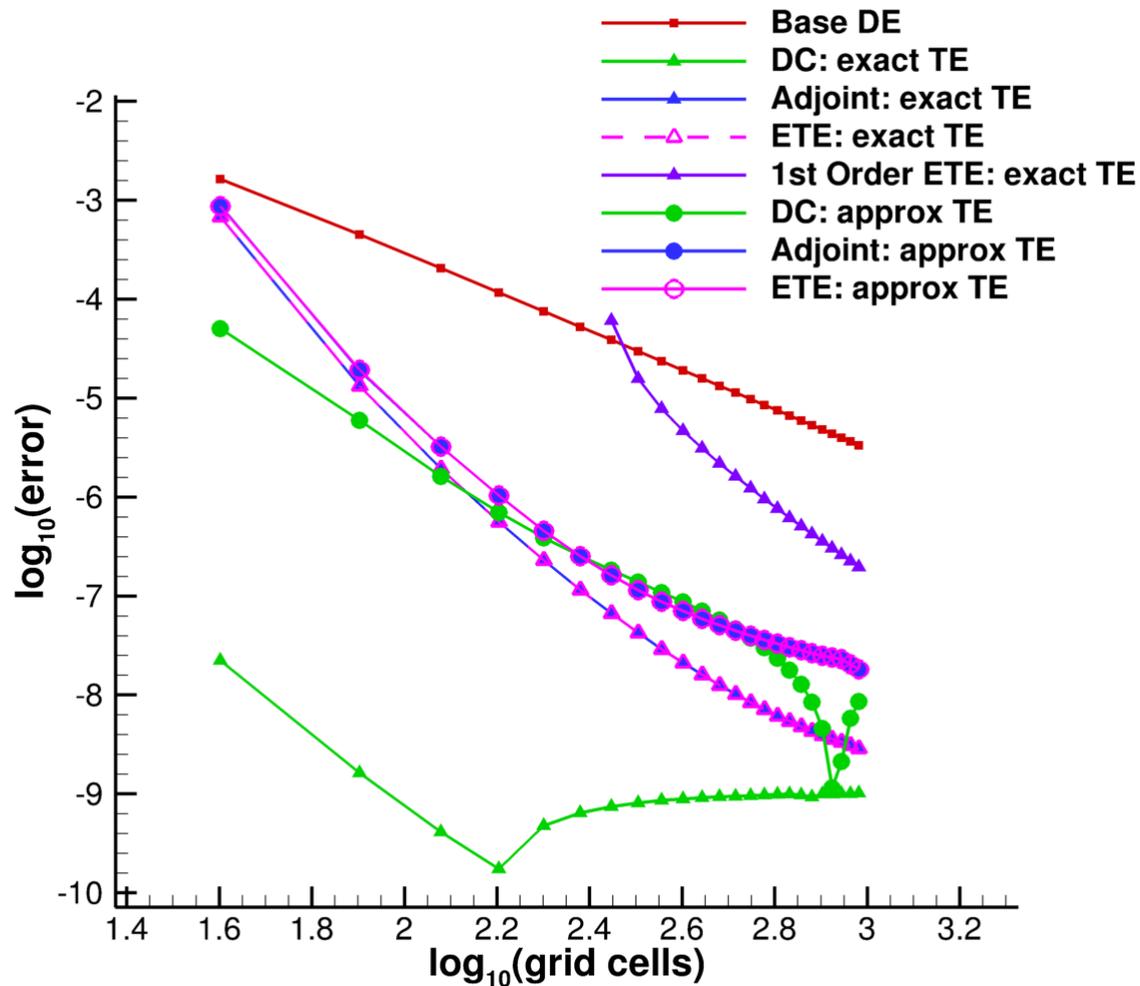


### Truncation Error



# Quasi-1D Euler Equations

## Functional DE Estimation



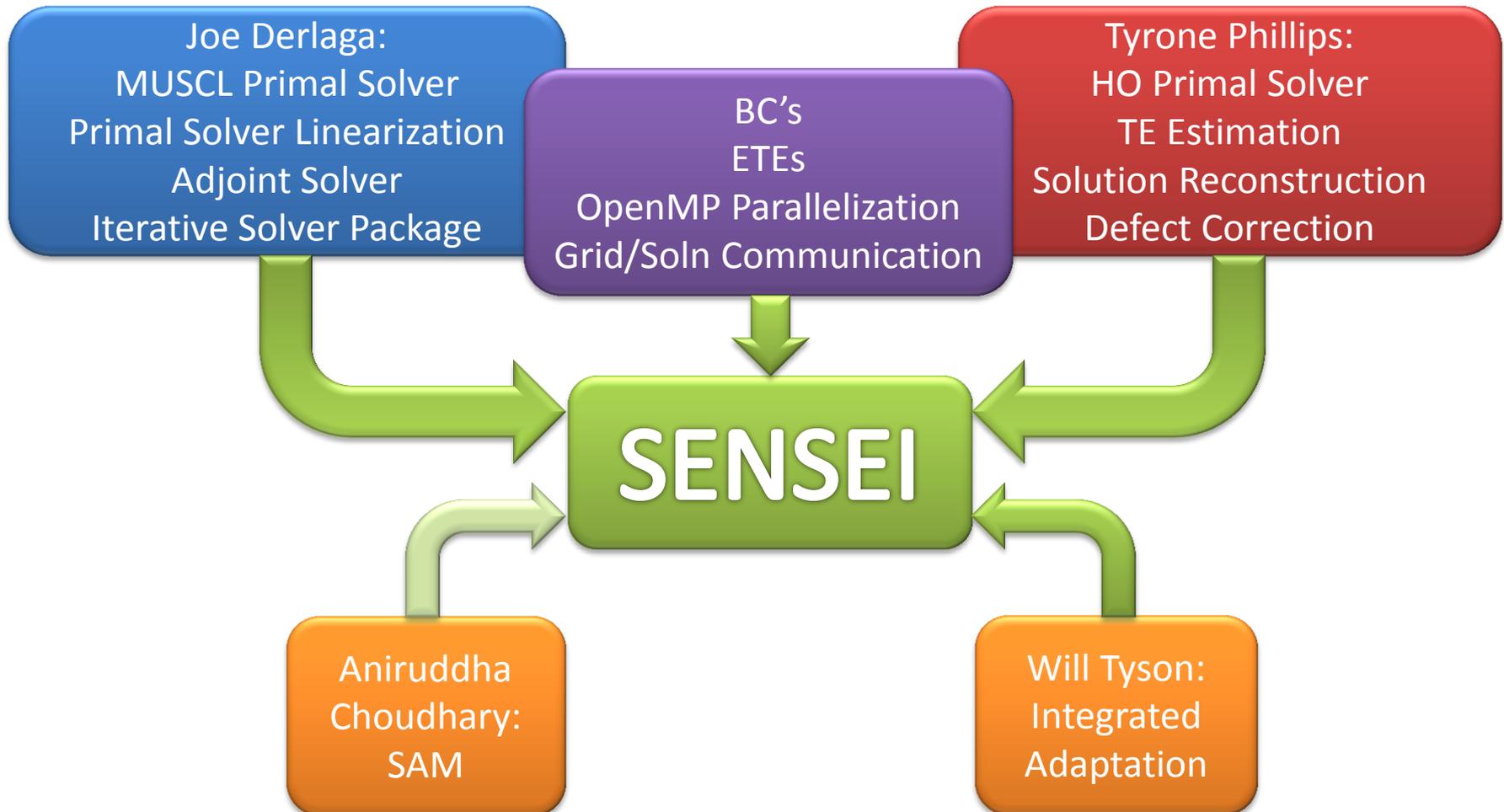
# Quasi-1D Euler Review

- **Developed a new method of estimating TE using the weak formulation of the governing equations and reconstructions of the discrete solution in a mapped physical space**
  - **Demonstrated that reconstructions of the primitive variables were the most appropriate**
- **Compared the impact of TE estimation procedures on the prediction of functional error**
  - **Flows containing shocks are difficult for the weak formulation to properly predict, however, a discrete residual method proved to be acceptable**
- **Demonstrated for one test case that DC, ETEs, and the adjoint method perform in a similar manner for the same TE estimate**
  - **Showed that a full linearization is necessary for the ETEs**

## Structured, Euler/Navier-Stokes Explicit/Implicit Solver

- Cell-centered, finite volume formulation using MUSCL extrapolation for 2<sup>nd</sup> order accuracy of inviscid fluxes and a Green's theorem approximation of derivatives for the viscous fluxes
- 2D/axisymmetric and 3D, multi-block, structured grids with point matched interfaces and ghost cells for interblock communication
- Time marching via explicit M-step Runge-Kutta or implicit Euler method
- OpenMP parallelized over blocks and written using modern Fortran 03/08 with software engineering best practices in mind

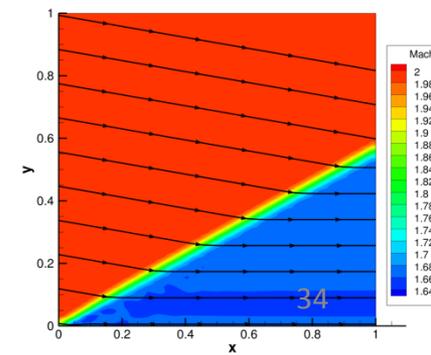
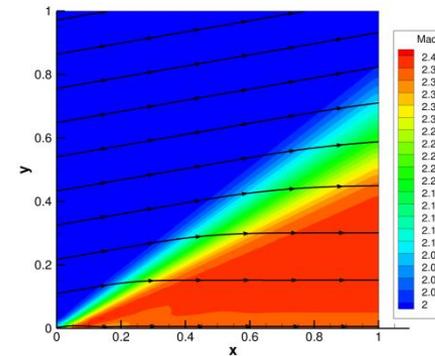
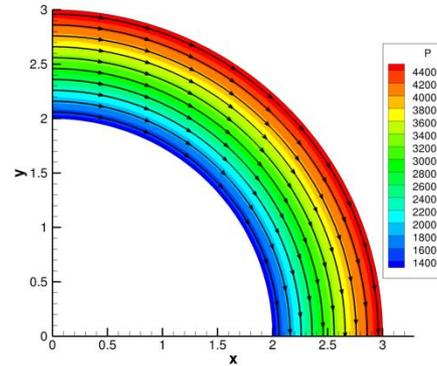
## Combines the work of multiple researchers



- **Discussed the importance of proper software engineering practices in and how it applies to modern engineering code development**
  - Often ignored by engineers, following best practices from the software engineering world results in code which is easier to develop, modify, and maintain
- **Developed a new MMS technique that is vastly simpler than the traditional formulation**
  - Tool to encourage the adoption of code verification techniques by reducing the implementation barrier
- **Discussed how the use of modern Fortran 03/08 greatly improves code quality and readability**
  - Familiar Fortran, but new abstractions make for easier to understand code and allow for future expansion

# SENSEI Test Cases

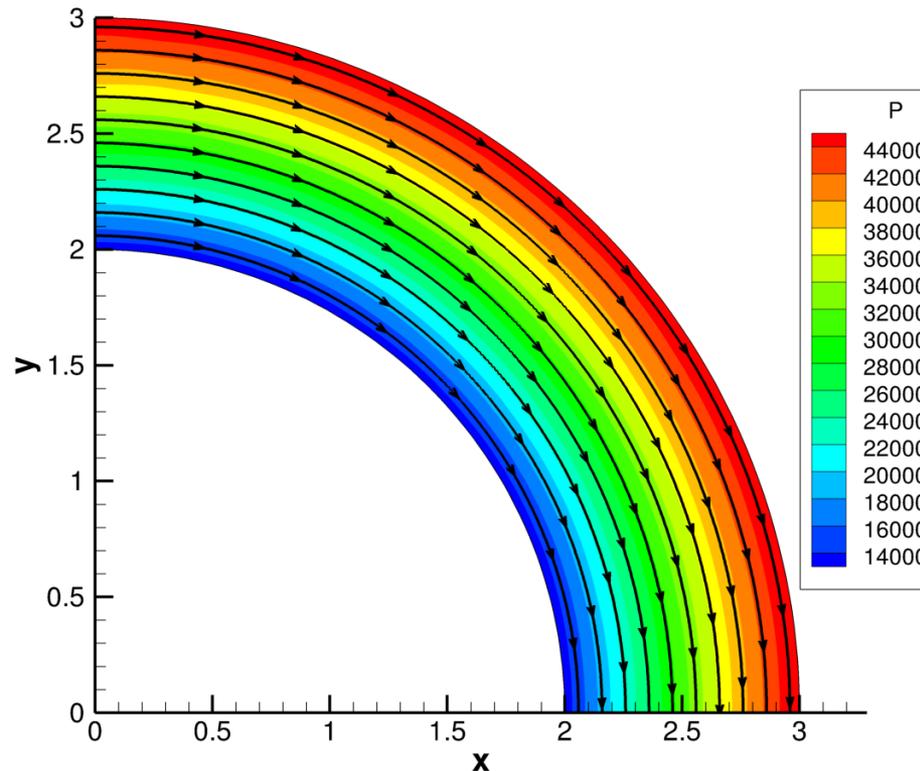
- Known exact solutions
- Varying order of accuracy
- van Leer FVS, van Albada limiter, MUSCL with  $\kappa = 1/3$
- DE in force functionals
- Qualitative DE comparisons using 'kexact' and 'fgc' methods
- Adaptation driven by TE and adjoint weighted TE weight functions using SAM



# Supersonic Vortex Flow

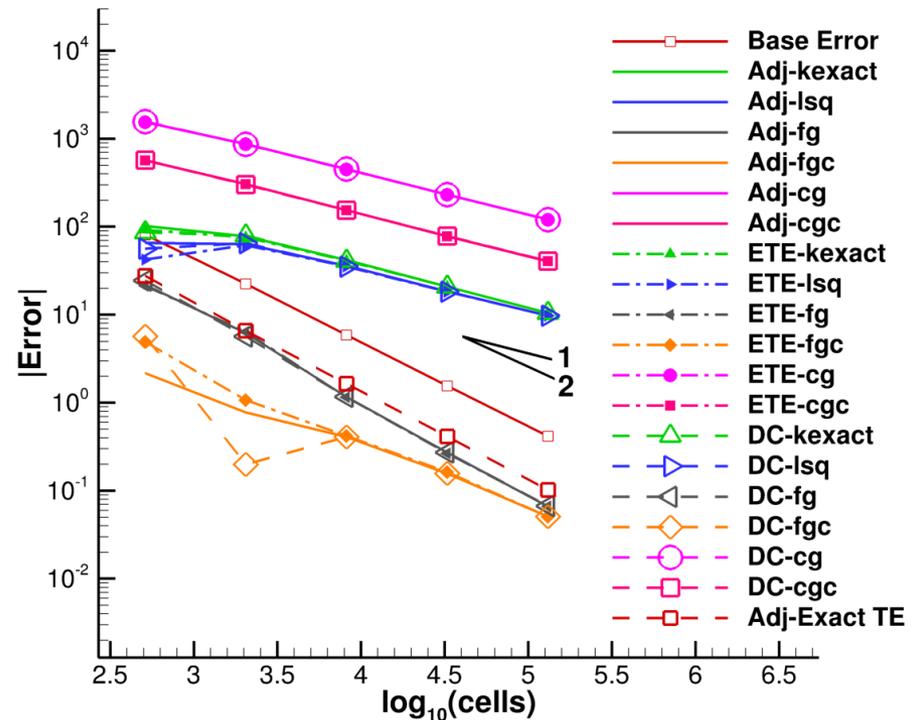
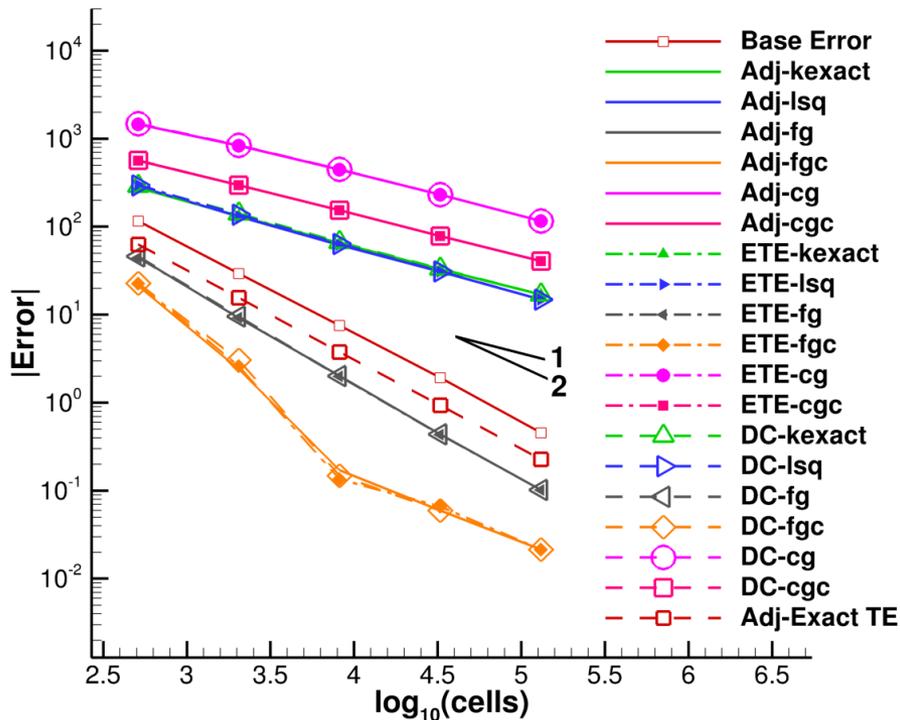
## Smooth turn of supersonic flow with OOA = 2

### Pressure Distribution



# Supersonic Vortex Flow

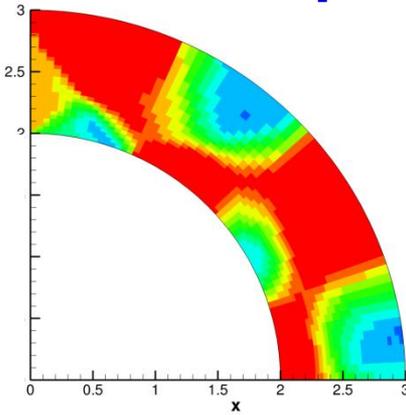
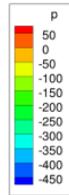
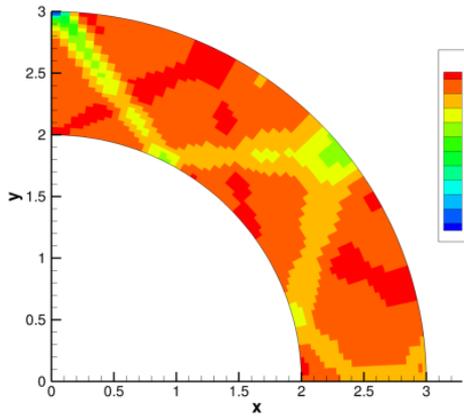
## Base and Remaining Functional DE Lift Drag



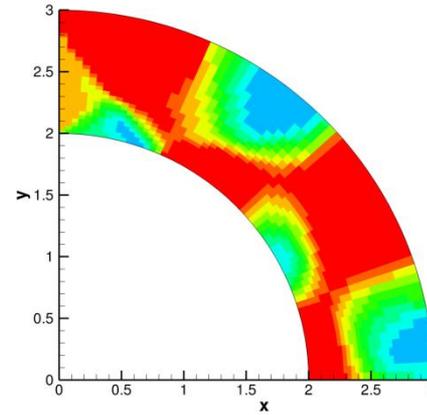
# Supersonic Vortex Flow

## Qualitative Comparison of DE in Pressure

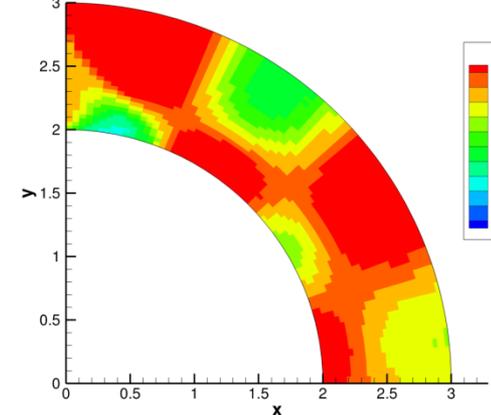
'kexact'



DC

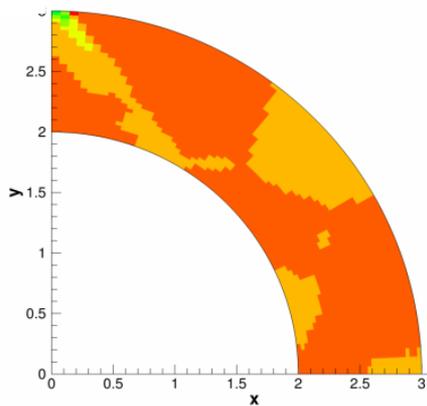


ETEs

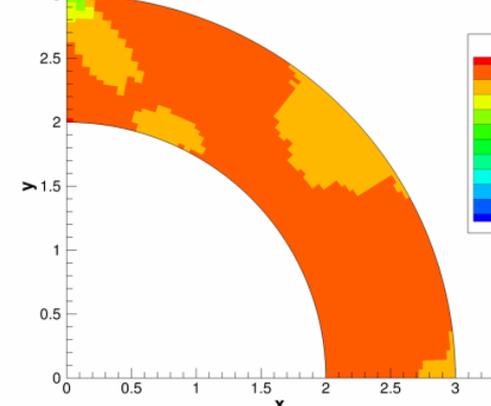
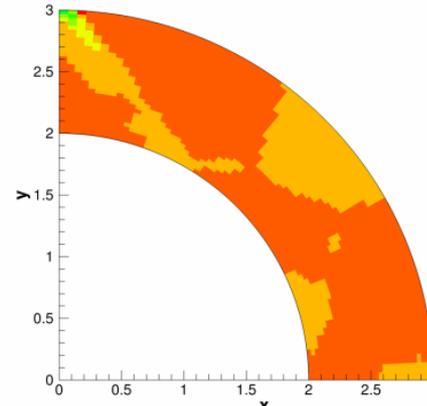


Approx. ETEs

Exact DE



'fgc'

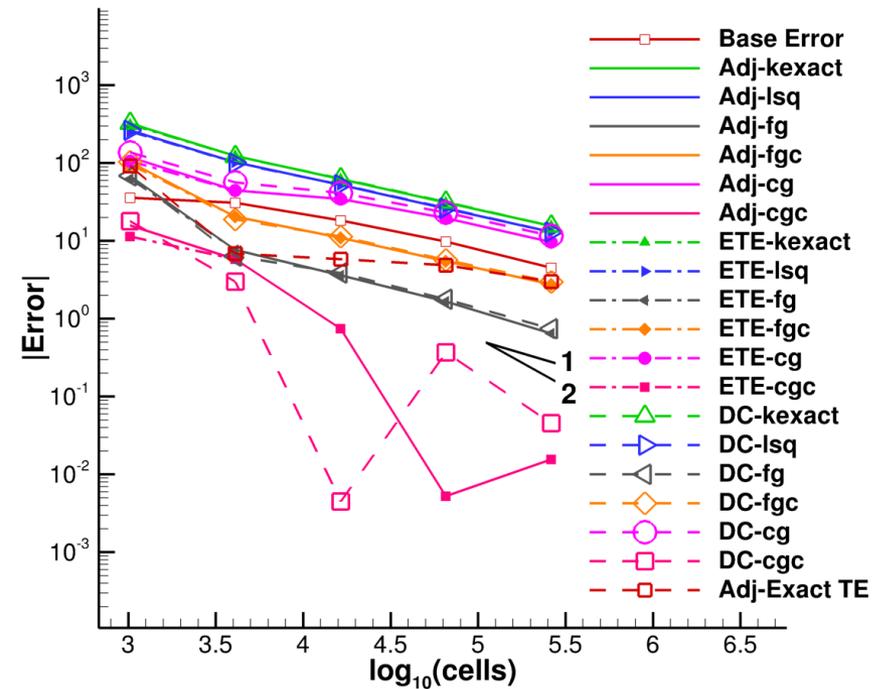
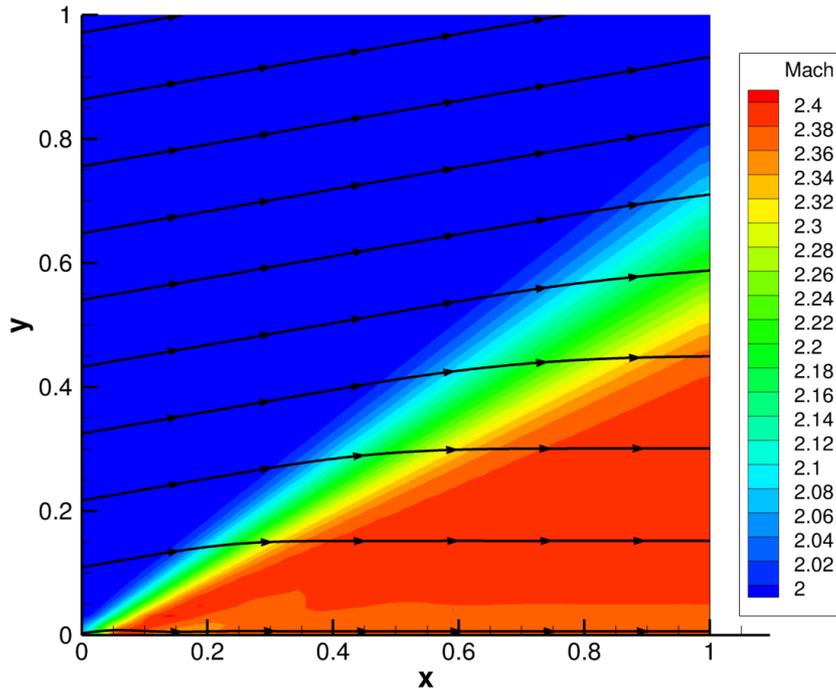


# Expansion Fan

Supersonic expansion around corner, OOA = 1

Base and Remaining DE  
in Force Normal to Plate

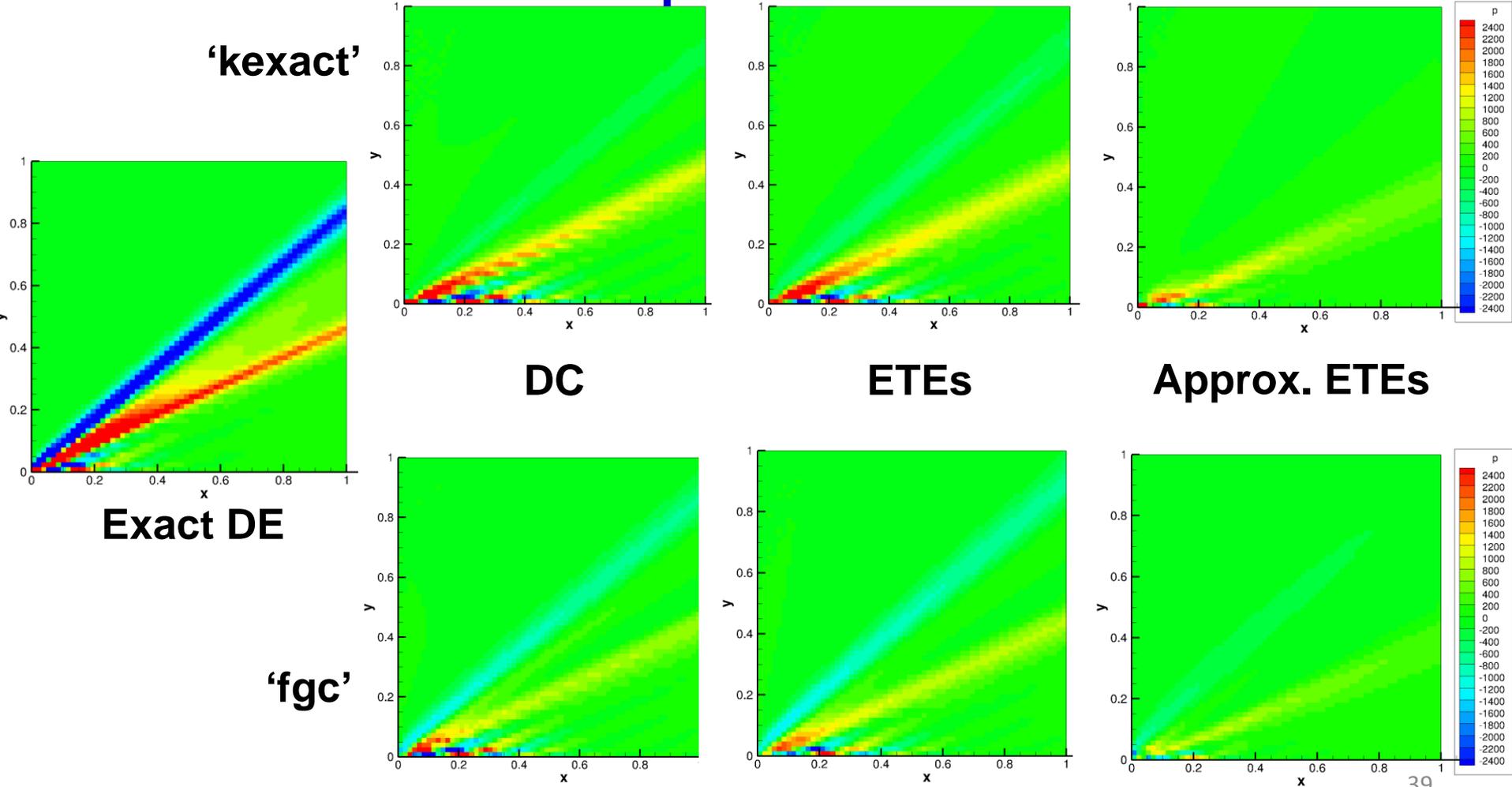
Mach Distribution



# Expansion Fan

## Qualitative Comparison of DE in Pressure

'kexact'



Exact DE

'fgc'

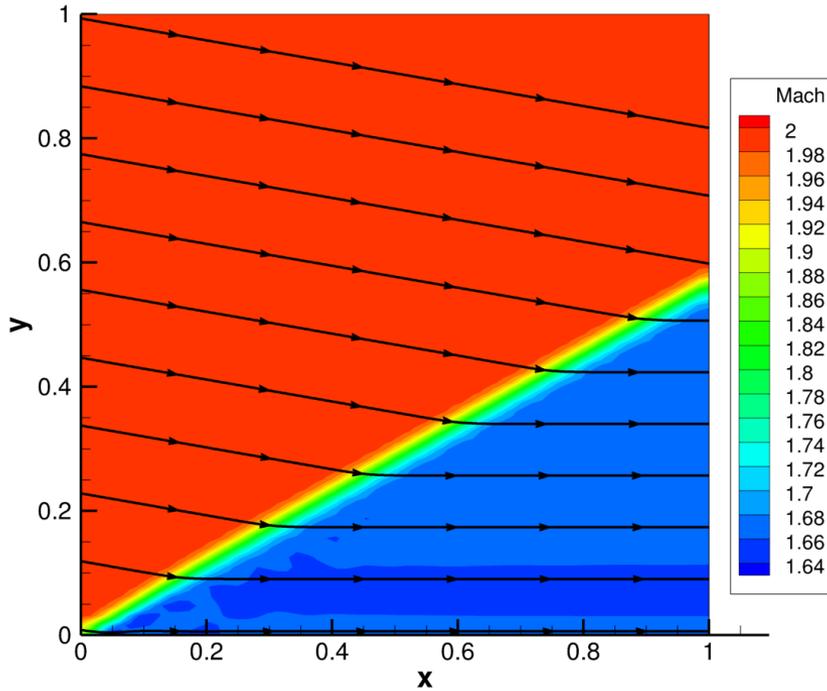
DC

ETEs

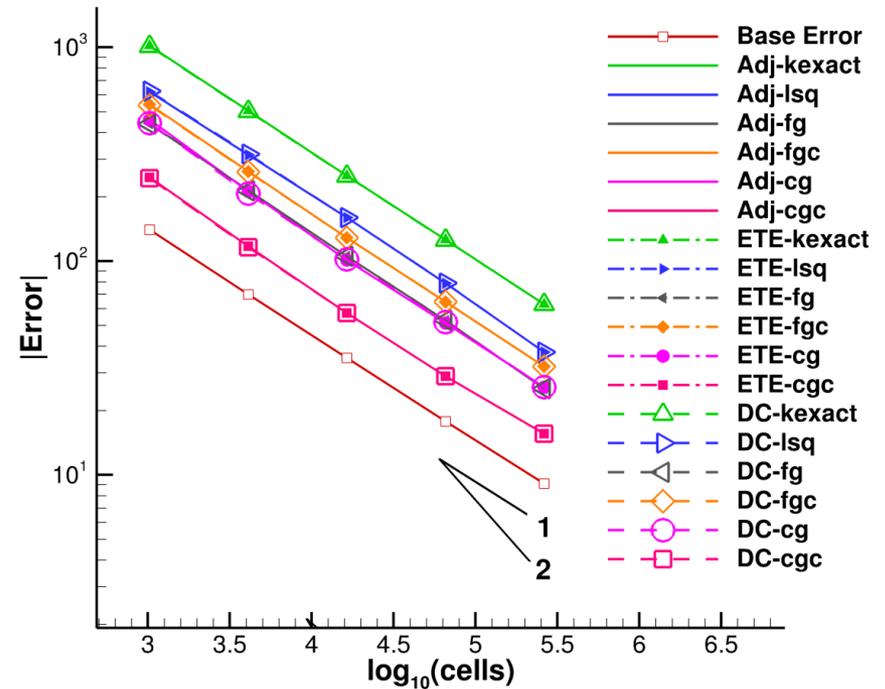
Approx. ETEs

## Supersonic compression, OOA = 1

### Mach Distribution

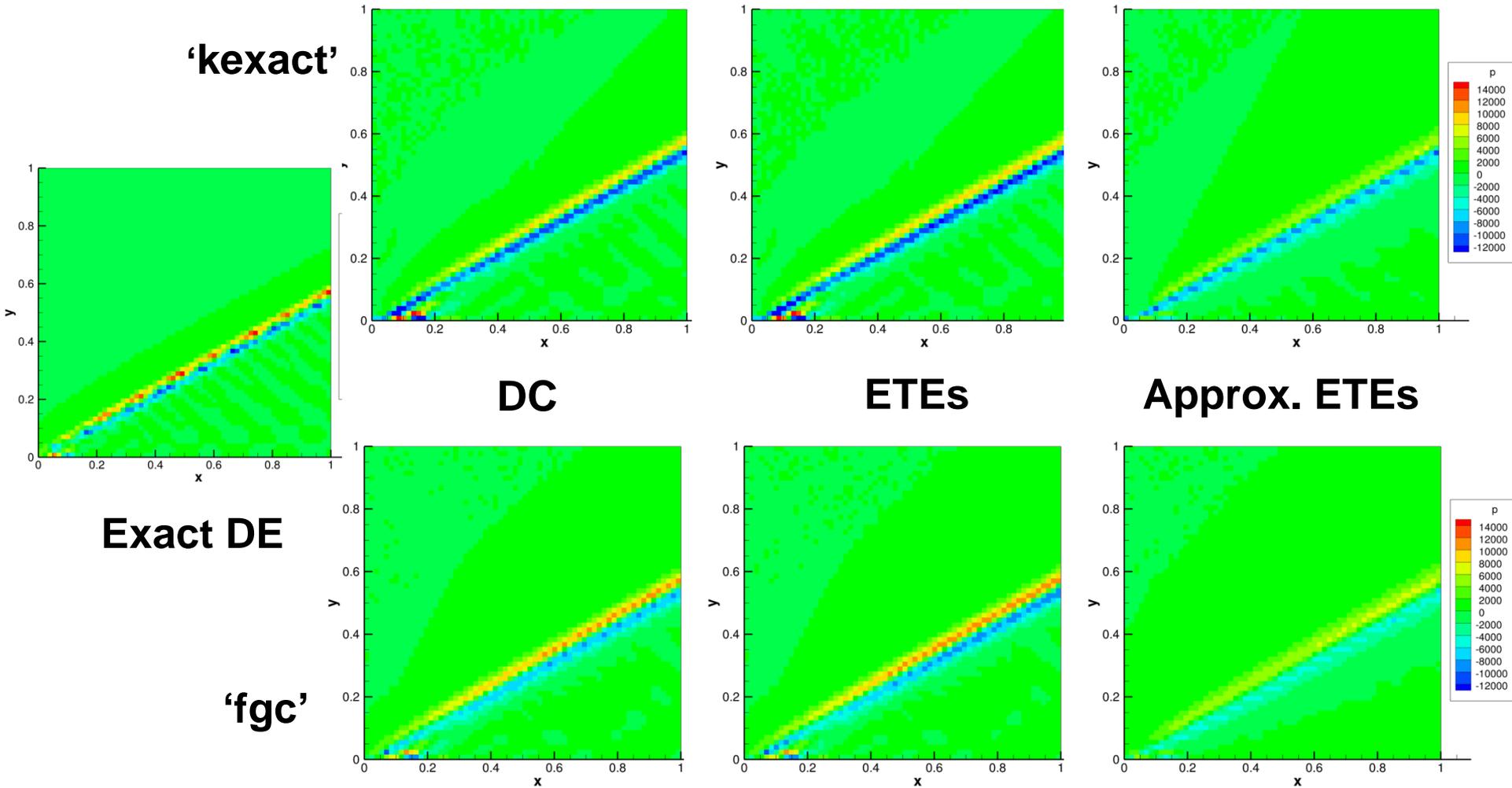


### Base and Remaining DE in Force Normal to Plate



# Shocked Flow

## Qualitative Comparison of DE in Pressure



## 2D Euler Review

- **Demonstrated that DC, ETEs, and the adjoint method produce comparable results for the same TE estimate**
  - Reinforced that a full linearization is necessary for the ETEs if they are to be competitive with the adjoint method
  - Gave guidelines for which error estimation procedure is the most appropriate to implement
- **Indicated that non-smoothness of mesh metrics on adapted meshes may be the largest contributor to poor TE estimates and therefore poor DE estimates**

# Contributions

- **Compared DC, ETEs, and adjoint methods for the first time in the literature and showed that each method performs in a comparable manner for the same TE estimate**
  - **If only DE estimates are needed, then DC makes the most sense to implement, but if computational speed is important, the ETEs make the most sense, but carry the caveat of increased development costs**
    - **The DC method costs about 40% of the computational time of the primal solve whereas the ETEs may only need 10%-25% of the primal solve computational time**
  - **But, if adaptation is important, then the adjoint method may need to be implemented, at a comparable development and solution cost to the ETEs**

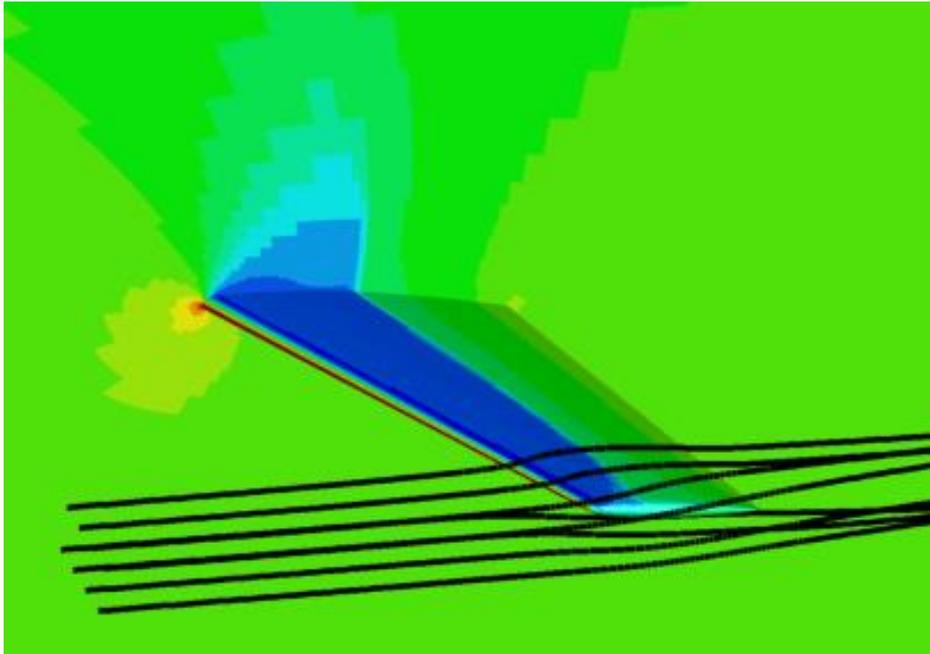
# Contributions

- **Demonstrated that reconstructions of the primitive variables were appropriate for TE estimation**
- **Developed a new CFD solver incorporating software engineering best practices**
  - **Integrated a new MMS procedure, iterative solver package, and enabled error estimation comparisons**

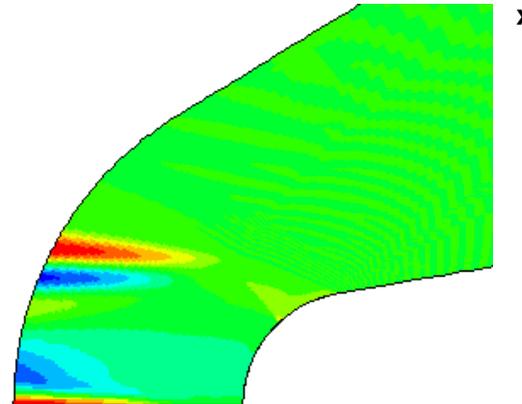
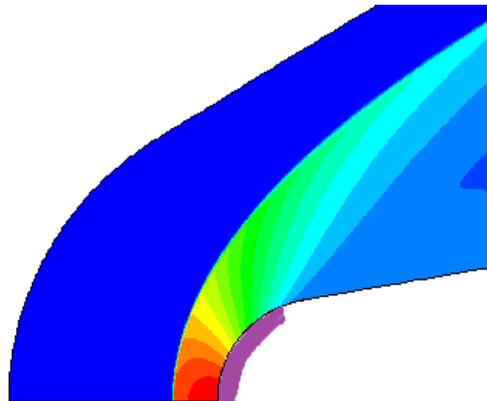
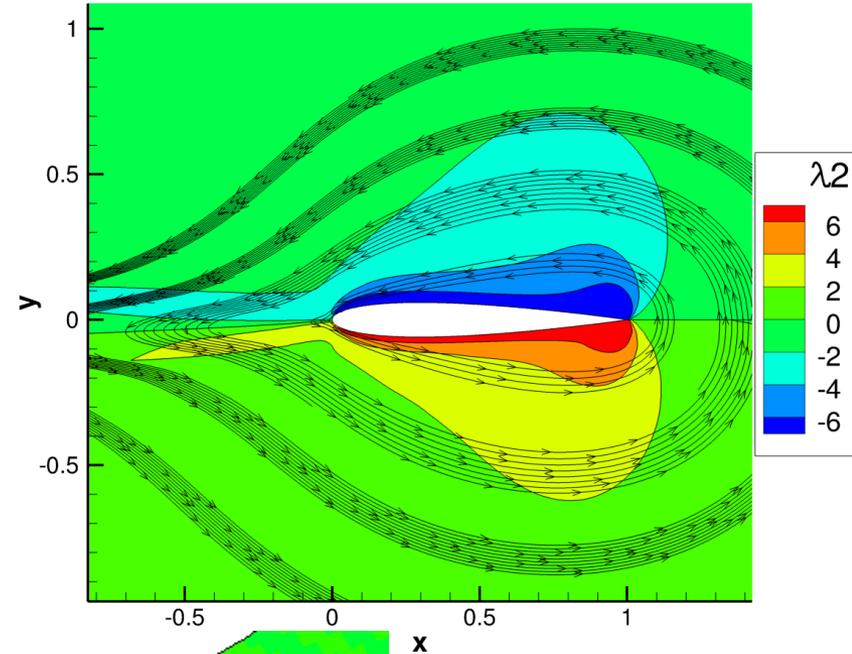
# Future Work

- **Re-examine the behavior of TE near domain and interblock boundaries**
  - Hybrid schemes may be the most appropriate near domain boundaries
- **Consider solution reconstructions in physical space in order to remove grid metrics entirely**
  - May be more applicable to unstructured meshes
- **Examine the use of improved adaptation procedures**
  - Could potentially fix grid metric issues that hamper the TE estimation on adapted meshes
- **Examine the use of the entropy adjoint in grid adaptation**
- **Extend the ETEs to multi-block domains**
- **Complete 3D, multi-block Navier-Stokes development**

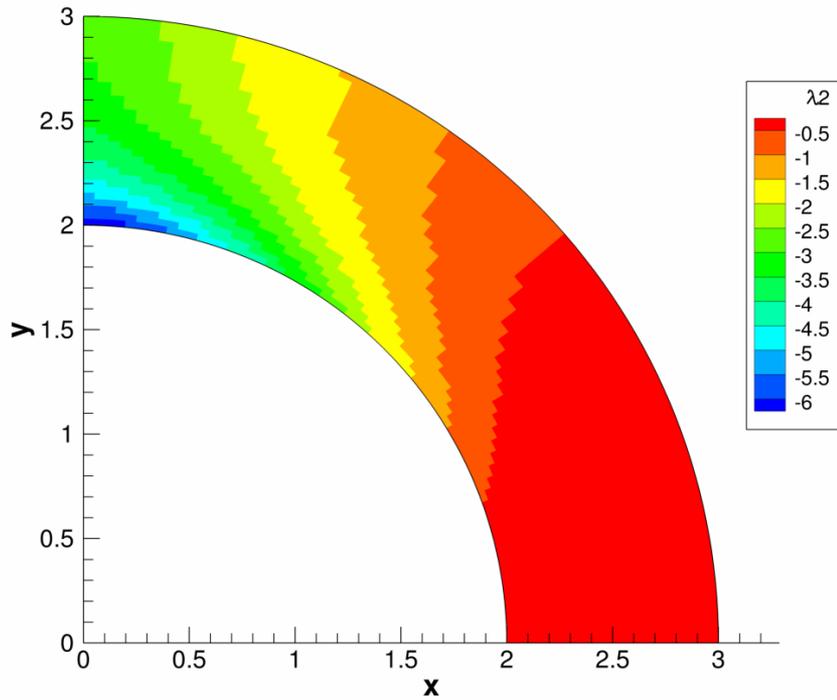
# Questions?



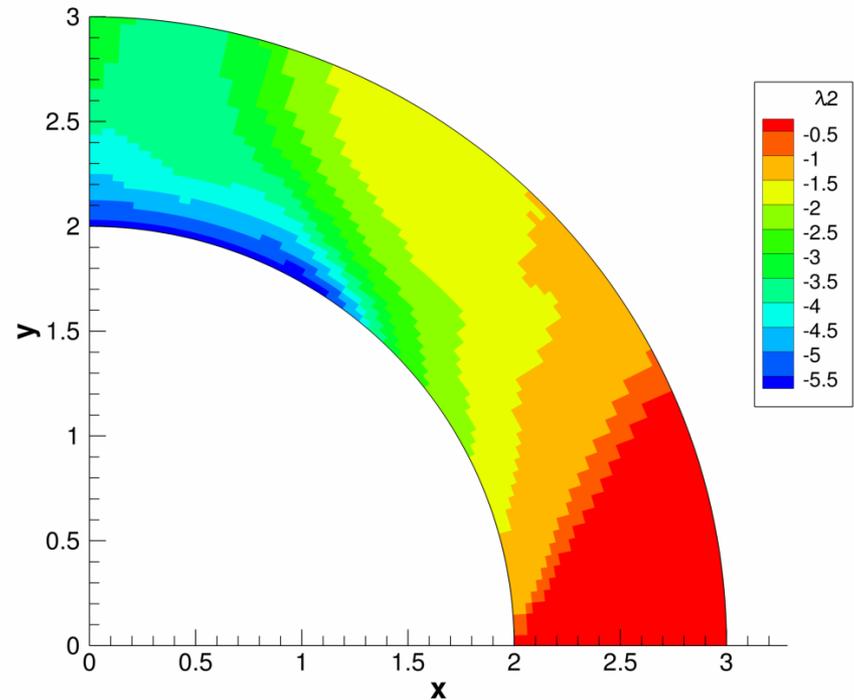
NACA 0012, Mach = 0.7, 1 Degree AoA



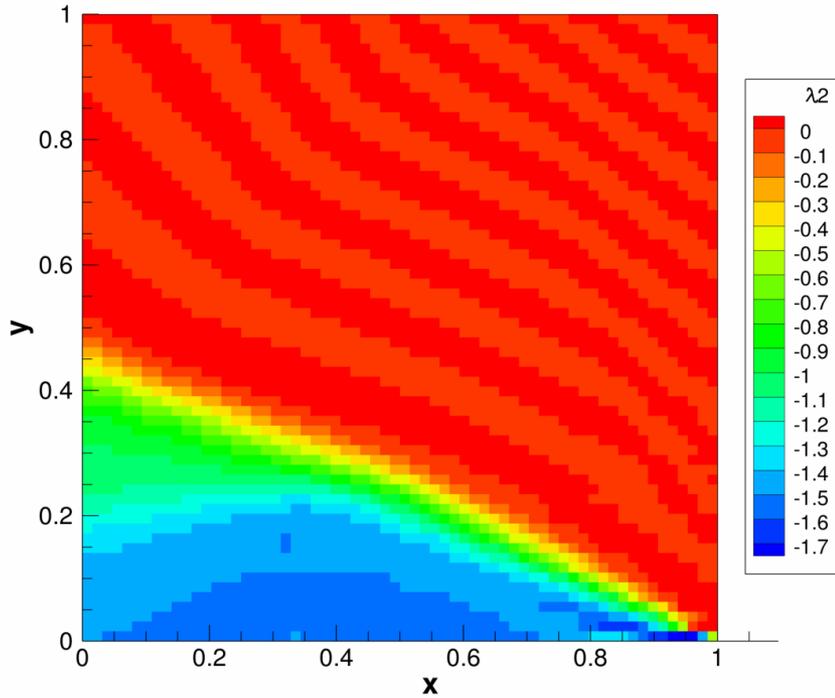
## SVF Lift Adjoint



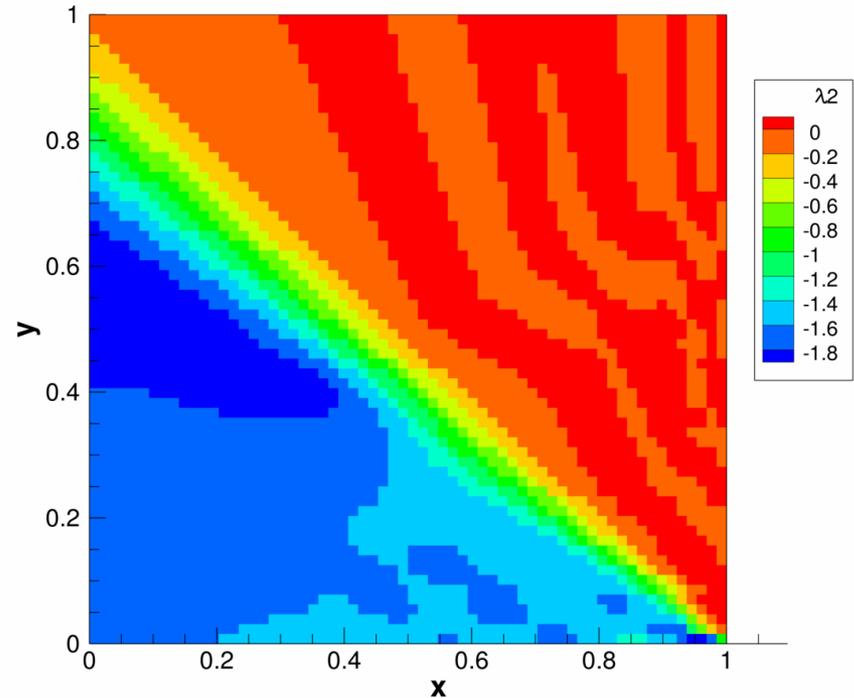
## SVF Drag Adjoint



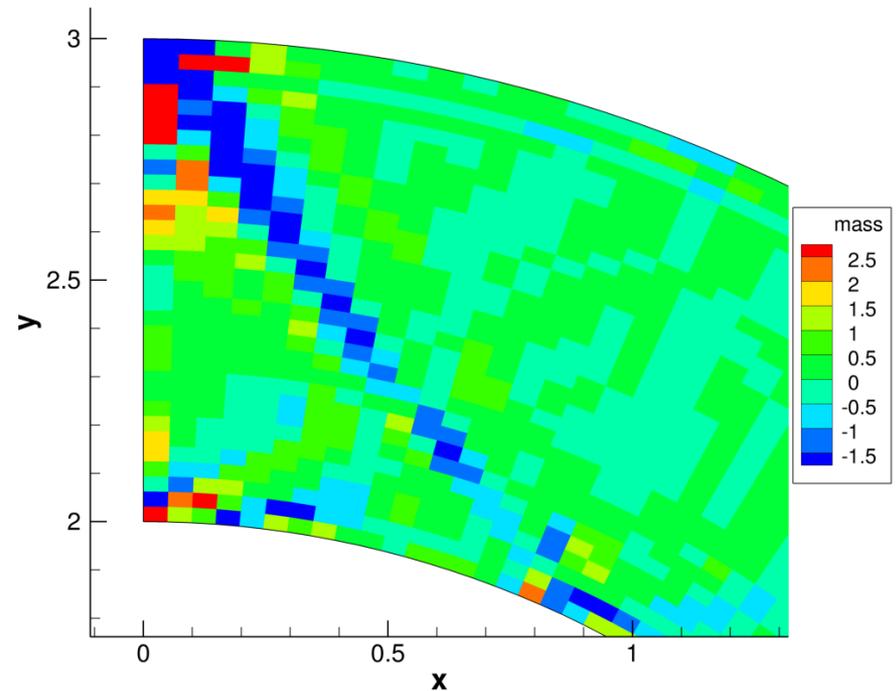
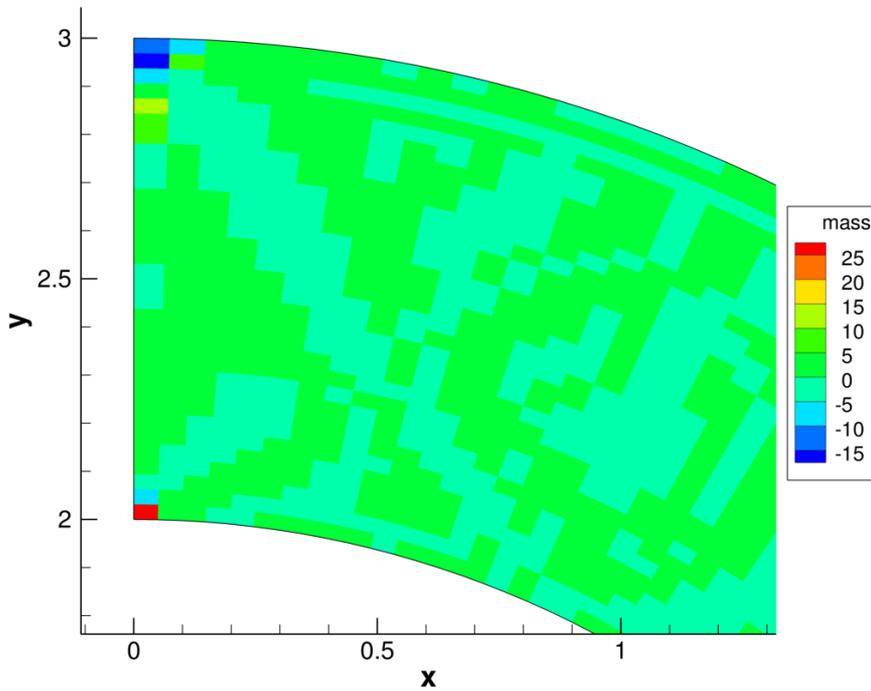
## Expansion Fan Adjoint



## Shock Adjoint

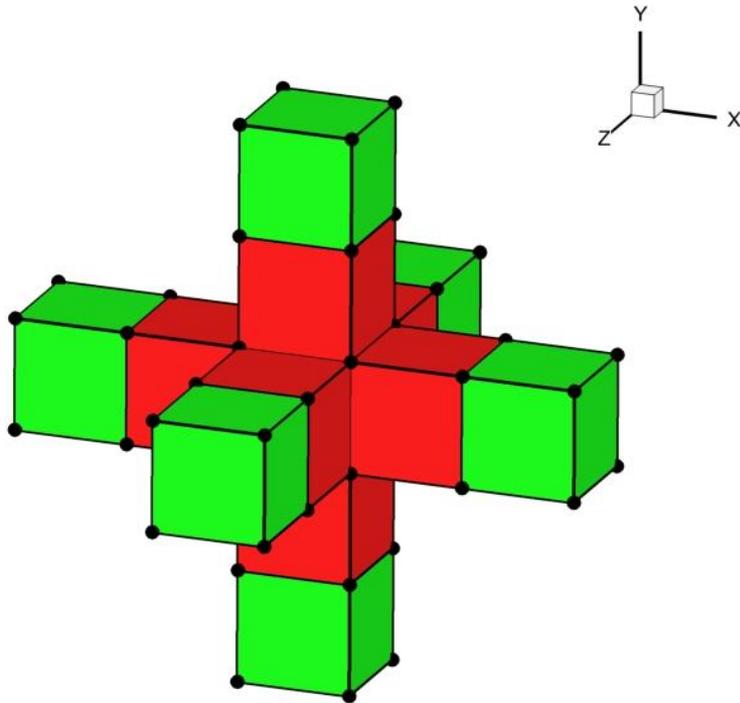


## SVF 'kexact' Estimated TE in Continuity Equation

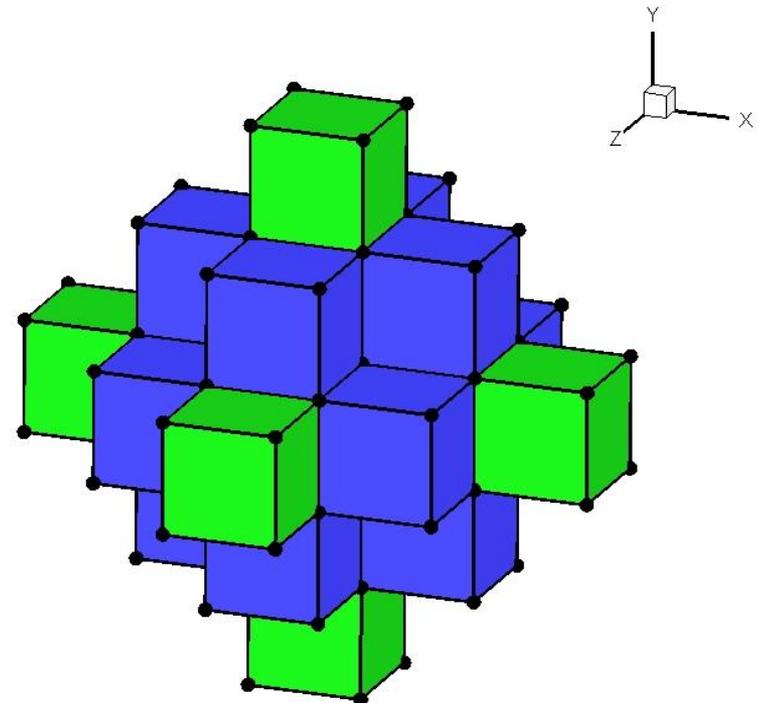


## SENSEI Stencils

### Euler Equations

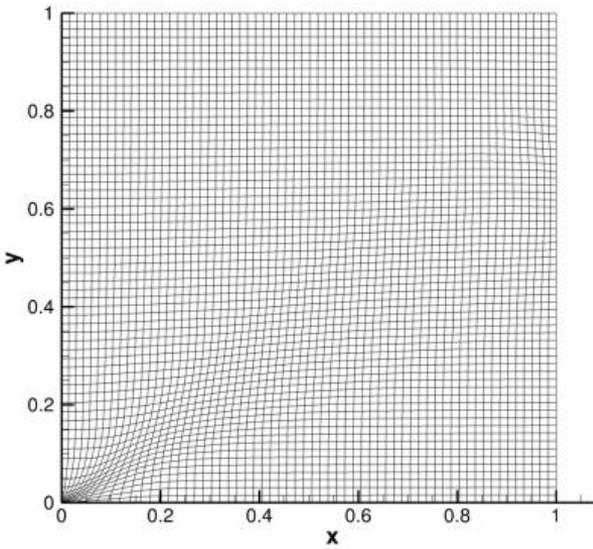


### Laminar NS

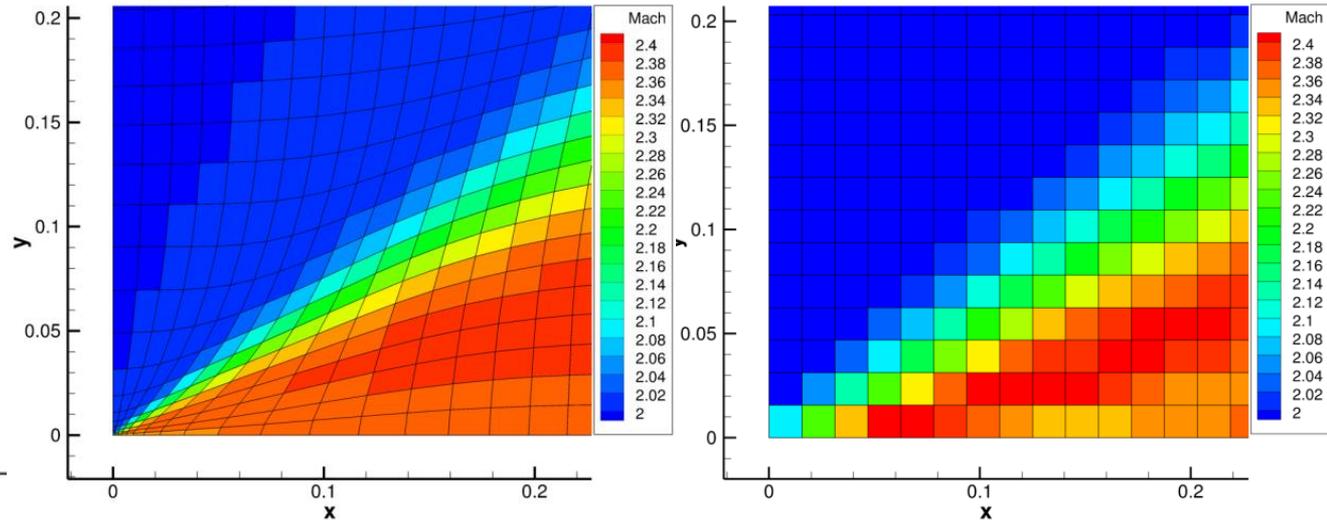


# Expansion Fan

## TE Adapted Mesh

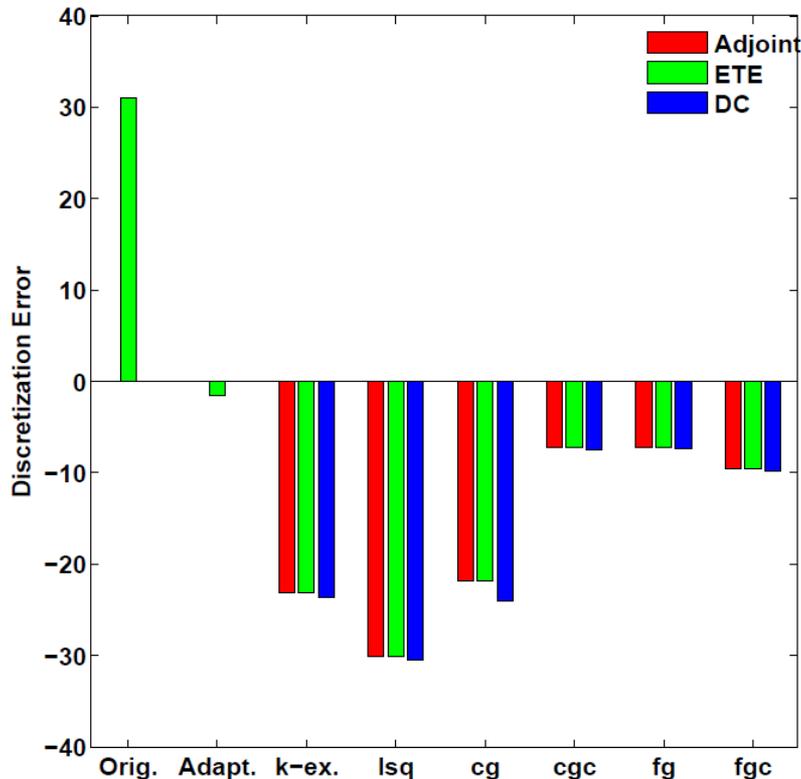


## Comparison of Mach Distributions

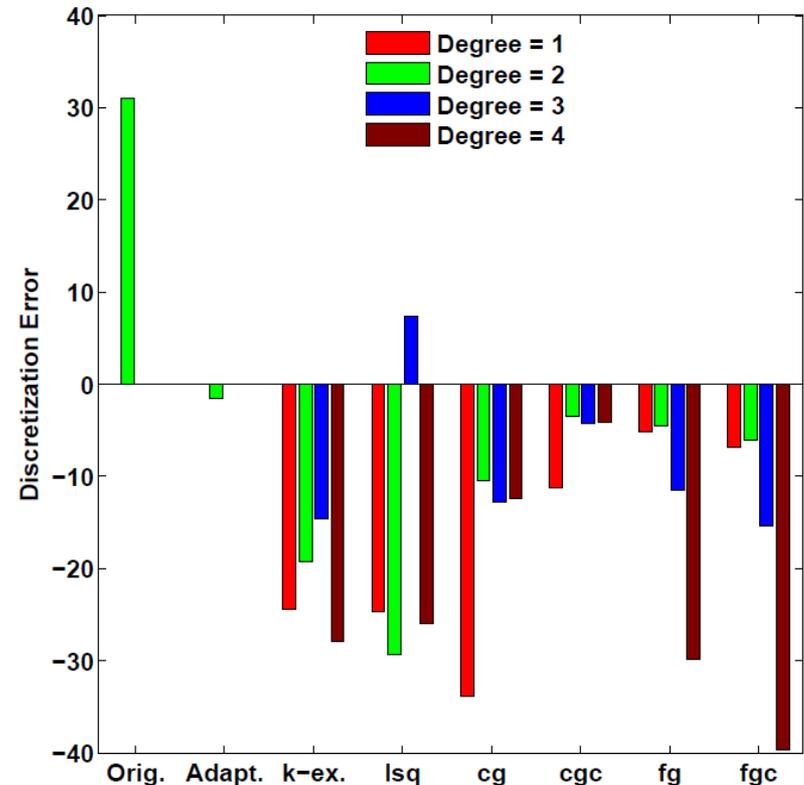


# Expansion Fan

## Functional DE Estimate on TE Adapted Mesh

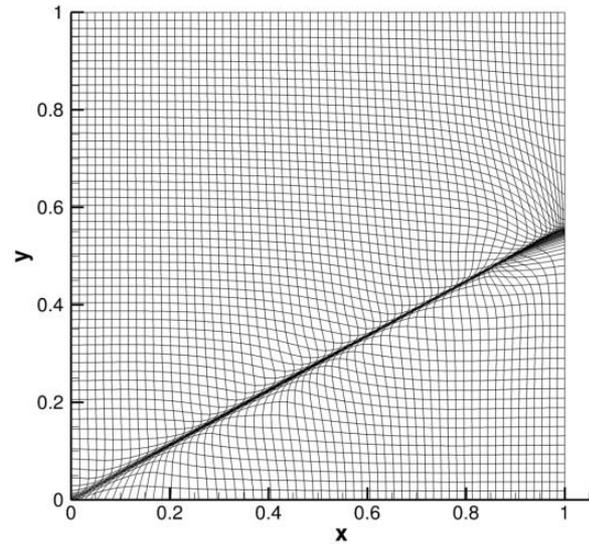
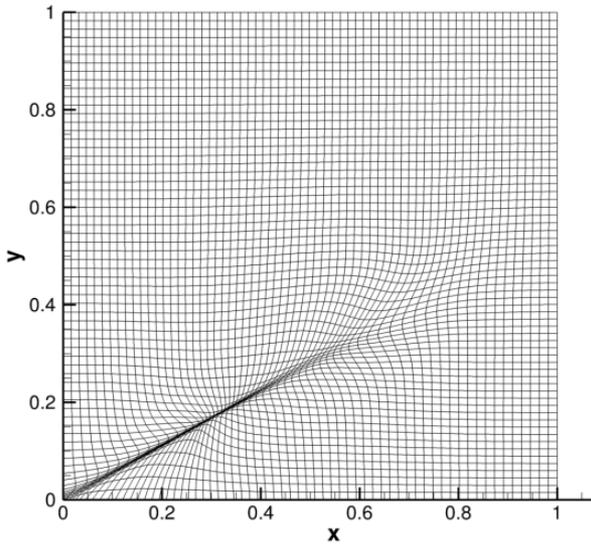


## Comparison of Effect of Polynomial Degree on Adjoint DE Estimate

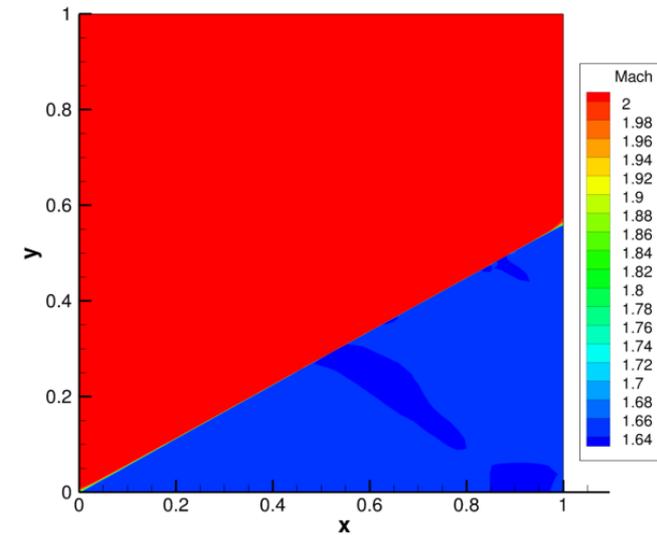


# Shocked Flow

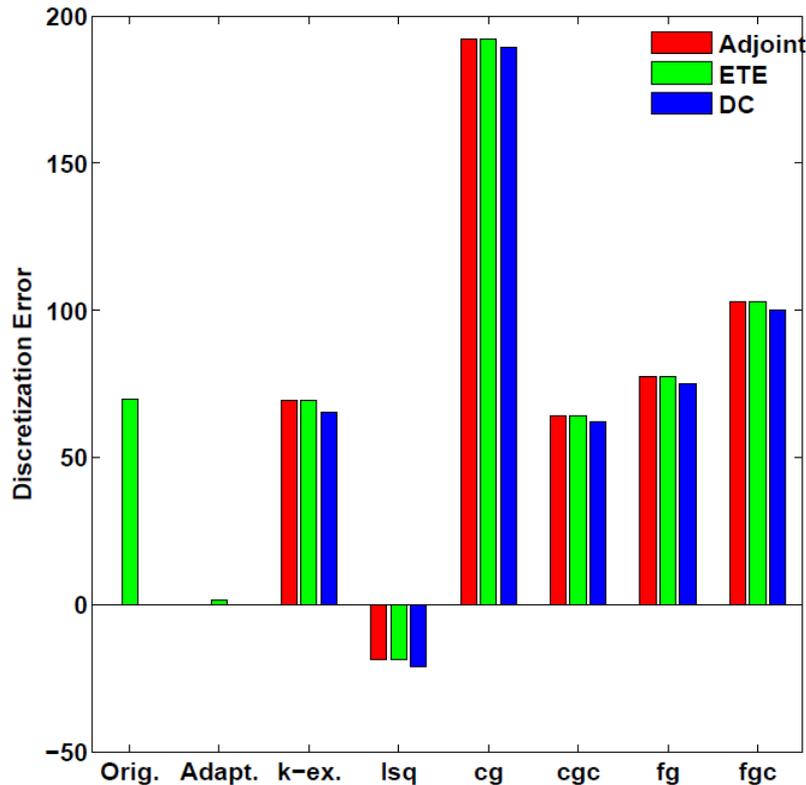
## Adjoint and TE Adapted Meshes



## Mach Contour



## Functional DE Estimate on TE Adapted Mesh



## Comparison of Effect of Polynomial Degree on Adjoint DE Estimate

