Beyond Guesswork: Code Verification for Acoustic Duct Mode Prediction

Jeffrey Severino

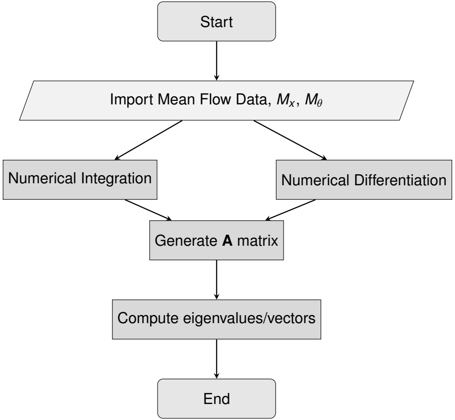
Spring 2024 NASA Acoustics Technical Working Group
Dr. Ray Hixon
Supported by AATT

March 20, 2024

Section 1

Basis of Technical Work

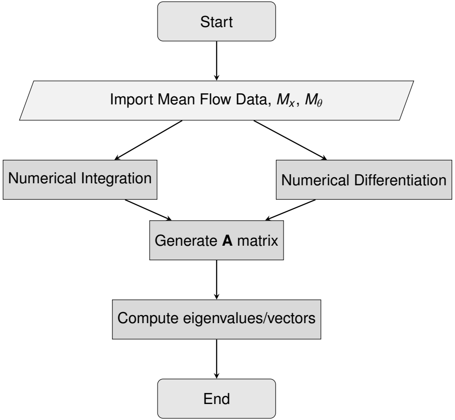
Code Implementation



The *Swirl* Code

- Code was first published by Kousen [1995]

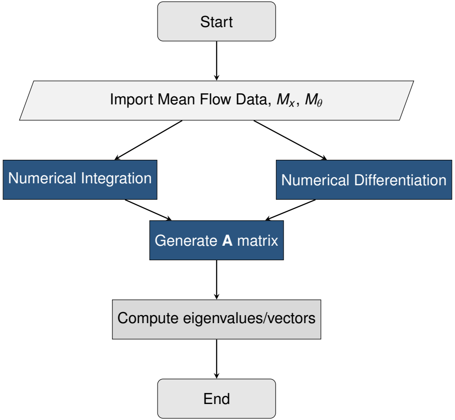
Code Implementation



The *Swirl* Code

- Code was first published by Kousen [1995]
- Contains three main numerical components:

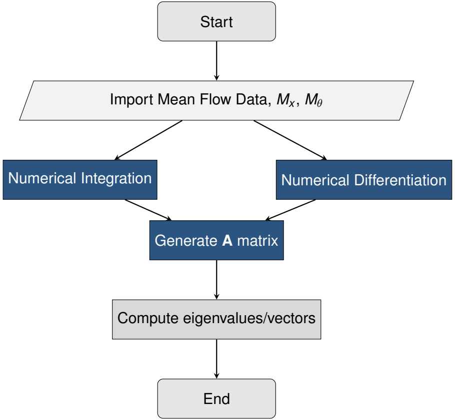
Code Implementation



The *Swirl* Code

- Code was first published by Kousen [1995]
- Contains three main numerical components:
  1. Numerical Integration

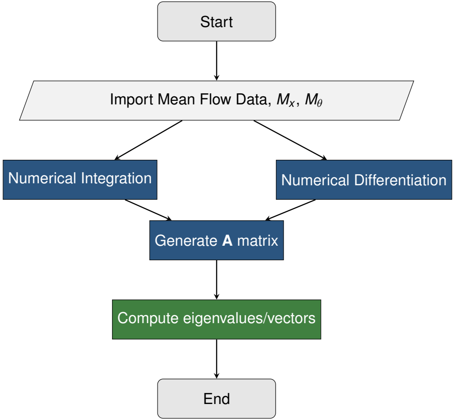
Code Implementation



The *Swirl* Code

- Code was first published by Kousen [1995]
- Contains three main numerical components:
  1. Numerical Integration
  2. Finite Difference Approximation for Derivative

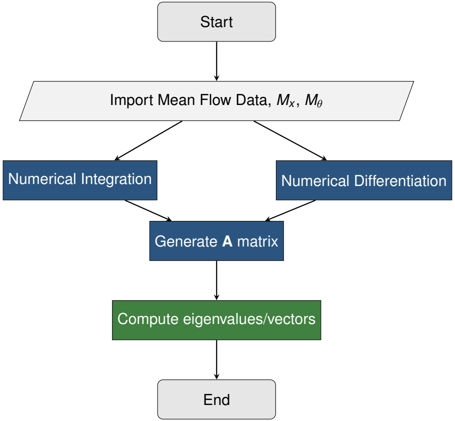
Code Implementation



The *Swirl* Code

- Code was first published by Kousen [1995]
- Contains three main numerical components:
  1. Numerical Integration
  2. Finite Difference Approximation for Derivative
  3. Eigenvalue/vector calculation

Code Implementation



The *Swirl* Code

- Code was first published by Kousen [1995]
- Contains three main numerical components:
  1. Numerical Integration
  2. Finite Difference Approximation for Derivative
  3. Eigenvalue/vector calculation

Goal: Code Verification by the Method of Manufactured and Exact Solutions (MMS/MES)

Verification Vs Validation

Whats the difference?

Often times, verification and validation are used interchangeably. Blottner has made an accepted distinction:

Verification Vs Validation

Whats the difference?

Often times, verification and validation are used interchangeably. Blottner has made an accepted distinction:

Verification

[Are we] solving the governing equations right?

Verification Vs Validation

Whats the difference?

Often times, verification and validation are used interchangeably. Blottner has made an accepted distinction:

Verification

[Are we] solving the governing equations right?

Validation

[Are we] solving the right governing equations?

Verification Vs Validation

Whats the difference?

Often times, verification and validation are used interchangeably. Blottner has made an accepted distinction:

Verification

[Are we] solving the governing equations right?

Code Verification

Concerned with error evaluation and correct algorithm implementation

Validation

[Are we] solving the right governing equations?

Verification Vs Validation

Whats the difference?

Often times, verification and validation are used interchangeably. Blottner has made an accepted distinction:

Verification

[Are we] solving the governing equations right?

Validation

[Are we] solving the right governing equations?

Code Verification

Concerned with error evaluation and correct algorithm implementation

Solution Verification

Concerned with error estimation that always occur in given numerical PDE solution

Verification Vs Validation

Whats the difference?

Often times, verification and validation are used interchangeably. Blottner has made an accepted distinction:

Verification

[Are we] solving the governing equations right?

Validation

[Are we] solving the right governing equations?

Code Verification

Concerned with error evaluation and correct algorithm implementation

Solution Verification

Concerned with error estimation that always occur in given numerical PDE solution

This work is focused code verification through the order-of-accuracy (OOA) test

Section 2

Verification of Numerical Integration

Numerical Integration for the Speed of Sound

Defining Error and Order of Accuracy

- Error for the speed of sound is defined as,

$$\epsilon = \left|\tilde{\mathsf{A}}_{\text{analytic}} - \tilde{\mathsf{A}}_{\text{numerical}}\right|. \tag{1}$$

Numerical Integration for the Speed of Sound

Defining Error and Order of Accuracy

- Error for the speed of sound is defined as,

$$\epsilon = \left| \tilde{\mathrm{A}}_{\text{analytic}} - \tilde{\mathrm{A}}_{\text{numerical}} \right|. \tag{1}$$

- If the numerical simulation was perfect, there is no error,

$$0 = \left| \tilde{\mathrm{A}}_{\text{analytic}} - \tilde{\mathrm{A}}_{\text{numerical}} \right|. \tag{2}$$

Numerical Integration for the Speed of Sound

Defining Error and Order of Accuracy

- Error for the speed of sound is defined as,

$$\epsilon = \left|\tilde{\text{A}}_{\text{analytic}} - \tilde{\text{A}}_{\text{numerical}}\right|. \tag{1}$$

- If the numerical simulation was perfect, there is no error,

$$0 = \left|\tilde{\text{A}}_{\text{analytic}} - \tilde{\text{A}}_{\text{numerical}}\right|. \tag{2}$$

- Solving for $\bar{\text{A}}_{\text{numerical}}$ and expanding with a Taylor Series gives,

$$\tilde{\text{A}}_{\text{numerical}} = \tilde{\text{A}}_{\text{analytic}} + \mathcal{O}\left(\Delta r^{\alpha}\right) + H.O.T. \tag{3}$$

Numerical Integration for the Speed of Sound

Defining Error and Order of Accuracy

- Error for the speed of sound is defined as,

$$\epsilon = \left| \tilde{\text{A}}_{\text{analytic}} - \tilde{\text{A}}_{\text{numerical}} \right|. \tag{1}$$

- If the numerical simulation was perfect, there is no error,

$$0 = \left| \tilde{\text{A}}_{\text{analytic}} - \tilde{\text{A}}_{\text{numerical}} \right|. \tag{2}$$

- Solving for $\bar{\text{A}}_{\text{numerical}}$ and expanding with a Taylor Series gives,

$$\tilde{\text{A}}_{\text{numerical}} = \tilde{\text{A}}_{\text{analytic}} + \mathcal{O}\left(\Delta r^{\alpha}\right) + H.O.T. \tag{3}$$

α is the formal order-of-accuracy (OOA) of the numerical scheme which dictates the expected rate of convergence as grid spacing is decreased.

Verification of Numerical Integration

- In *Swirl*, the analytic speed of sound is,

$$\widetilde{A}_{\text{analytic}} = 1 - (\gamma - 1) \int_{\tilde{r}}^{1} \frac{M_\theta}{\tilde{r}} d\tilde{r} \tag{4}$$

Verification of Numerical Integration

- In *Swirl*, the analytic speed of sound is,

$$\widetilde{A}_{\text{analytic}} = 1 - (\gamma - 1)\int_{\tilde{r}}^{1} \frac{M_\theta}{\tilde{r}} d\tilde{r} \tag{4}$$

- However, the integration is done with trapezoidal rule,

$$\int_{\tilde{r}}^{1} \frac{M_\theta}{\tilde{r}} d\tilde{r} \approx \frac{\Delta\widetilde{r}}{2}\left[\frac{M_{\theta,1}^2}{\tilde{r}_1} + \frac{M_{\theta,n}^2}{\tilde{r}_n} + 2\Sigma_{i=1}^{n-2}\frac{M_{\theta,i}^2}{\tilde{r}}\right] + \mathcal{O}\left(\Delta r^2\right) \tag{5}$$

Verification of Numerical Integration

- In *Swirl*, the analytic speed of sound is,

$$\widetilde{A}_{\text{analytic}} = 1 - (\gamma - 1) \int_{\tilde{r}}^{1} \frac{M_\theta}{\tilde{r}} d\tilde{r} \tag{4}$$

- However, the integration is done with trapezoidal rule,

$$\int_{\tilde{r}}^{1} \frac{M_\theta}{\tilde{r}} d\tilde{r} \approx \frac{\Delta \widetilde{r}}{2} \left[\frac{M_{\theta,1}^2}{\tilde{r}_1} + \frac{M_{\theta,n}^2}{\tilde{r}_n} + 2\Sigma_{i=1}^{n-2} \frac{M_{\theta,i}^2}{\tilde{r}} \right] + \mathcal{O}\left(\Delta r^2\right) \tag{5}$$

Code Verification by the MMS

To verify the numerical integration an analytic function is chosen i.e. manufactured for M_θ

Verification of Numerical Integration

Manufacturing Solutions

- A Python library was written to generate summations of *tanh* symbolically for M_θ and then converted to FORTRAN for code compatibility.

Verification of Numerical Integration

Manufacturing Solutions

- A Python library was written to generate summations of *tanh* symbolically for M_θ and then converted to FORTRAN for code compatibility.
- Once plugged into Eqn (4), the speed of sound is

$$\tilde{A}_{MS} = \frac{1}{24}\left(\begin{array}{c} \tanh\left(\frac{\tilde{r}}{3} - \frac{1}{3}\right) + \tanh\left(\frac{\tilde{r}}{3} - \frac{31}{120}\right) + \\ \tanh\left(\frac{\tilde{r}}{3} - \frac{11}{60}\right) + \tanh\left(\frac{\tilde{r}}{3} - \frac{13}{120}\right) + \tanh\left(\frac{\tilde{r}}{3} - \frac{1}{30}\right) \end{array}\right) + \frac{853}{880} \tag{6}$$

Verification of Numerical Integration

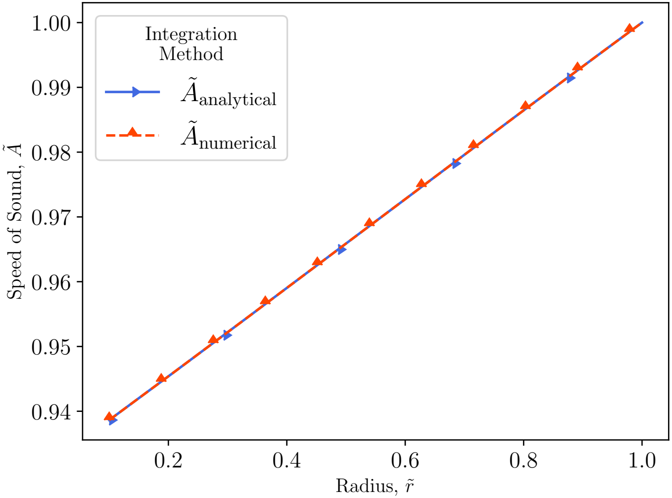
Manufacturing Solutions

- A Python library was written to generate summations of *tanh* symbolically for M_θ and then converted to FORTRAN for code compatibility.
- Once plugged into Eqn (4), the speed of sound is

$$\tilde{A}_{MS} = \frac{1}{24}\left(\begin{array}{c}\tanh\left(\frac{\tilde{r}}{3}-\frac{1}{3}\right)+\tanh\left(\frac{\tilde{r}}{3}-\frac{31}{120}\right)+\\ \tanh\left(\frac{\tilde{r}}{3}-\frac{11}{60}\right)+\tanh\left(\frac{\tilde{r}}{3}-\frac{13}{120}\right)+\tanh\left(\frac{\tilde{r}}{3}-\frac{1}{30}\right)\end{array}\right)+\frac{853}{880} \tag{6}$$

This is one possible analytic solution to the speed of sound, but this solution is *manufactured*, hence the subscript *MS*

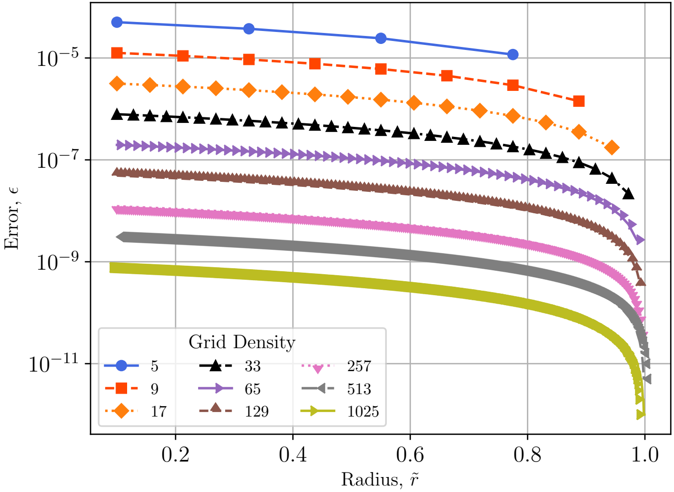
Verification of Numerical Integration



Visually looks identical..

The first step of the OOA test is to compute the error

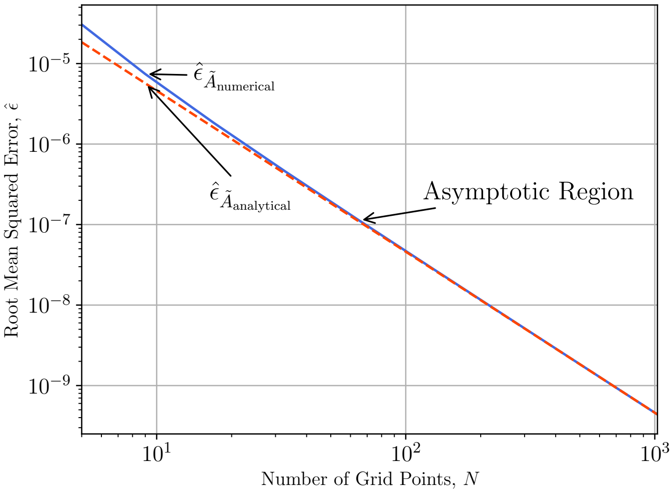
Verification of Numerical Integration



Value of error indicates accuracy

The root mean square of the error, (L_2 norm) is then used to compute the *observed* OOA

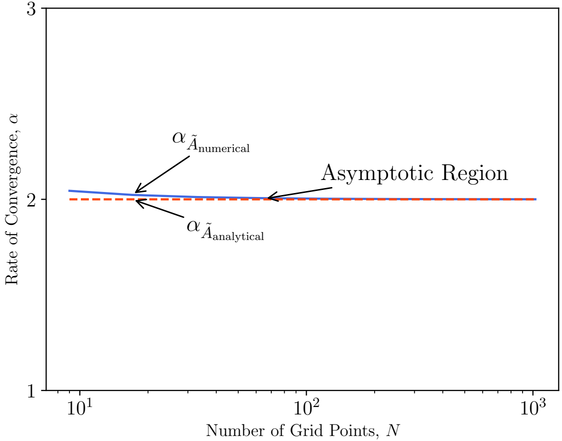
Verification of Numerical Integration



A straight line of expected slope is superimposed

Convergence starts in the asymptotic region at $\approx$ 100 grid points

Verification of Numerical Integration



The observed rate of convergence (OOA) approaches the formal value

The numerical integration is now verified via MMS

Section 3

Verification of Numerical Differentiation

Verification of Numerical Differentation and Matrix Construction

The Governing Equations

- A given mean flow is used to establish an eigenvalue problem with the Linearized Euler Equations

$$[A]\chi = \lambda[B]\chi \tag{7}$$

where χ and λ contain the eigenvalues (axial wavenumbers) and vectors (radial pressure modes)

Verification of Numerical Differentation and Matrix Construction

The Governing Equations

- A given mean flow is used to establish an eigenvalue problem with the Linearized Euler Equations

$$[A]\chi = \lambda[B]\chi \tag{7}$$

  where χ and λ contain the eigenvalues (axial wavenumbers) and vectors (radial pressure modes)

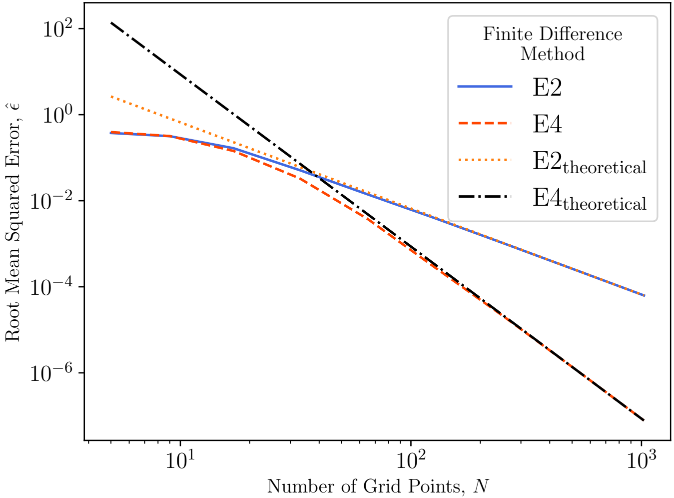
- *A* requires finite difference approximations of the mean flow are needed for the *A* matrix construction.

Verification of Numerical Differentation and Matrix Construction

The Governing Equations

- A given mean flow is used to establish an eigenvalue problem with the Linearized Euler Equations
$$[A]\chi = \lambda[B]\chi \tag{7}$$
where χ and λ contain the eigenvalues (axial wavenumbers) and vectors (radial pressure modes)
- *A* requires finite difference approximations of the mean flow are needed for the *A* matrix construction.
- Manufactured solutions were generated using a summation of tangents to verify OOA of a second and fourth order scheme

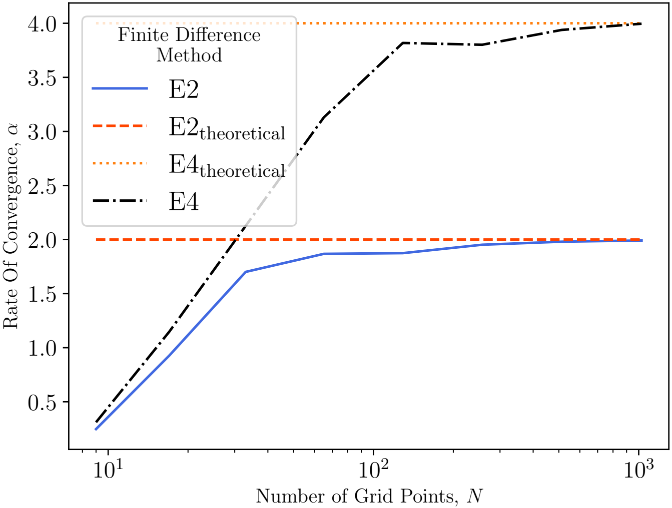
Verification of Numerical Differentation and Matrix Construction



Two straight lines of expected slope for each scheme is superimposed

Similarly, convergence is observed the asymptotic region at ≈ 100 grid points

Verification of Numerical Differentation and Matrix Construction



The observed rate of convergence (OOA) approaches the formal value

The finite differencing scheme and matrix construction have been verified with MMS

Section 4

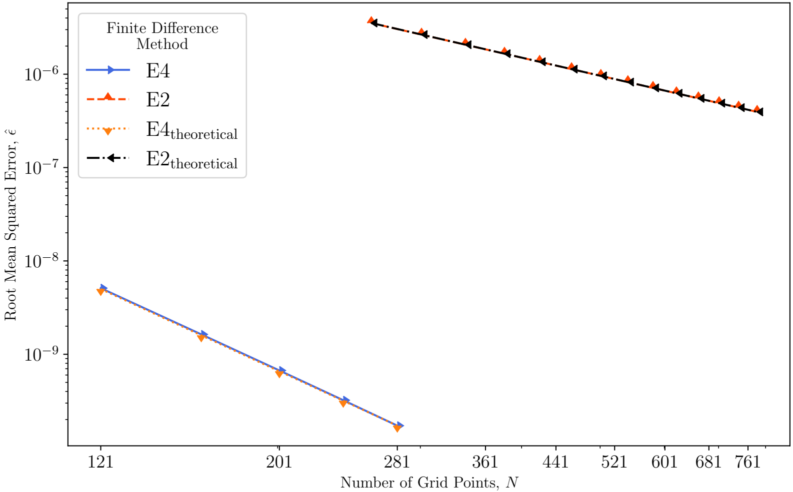
Verification of Eigenvalues and vectors

Verification of Eigenvalues and vectors

Verification by the Method of Exact Solutions

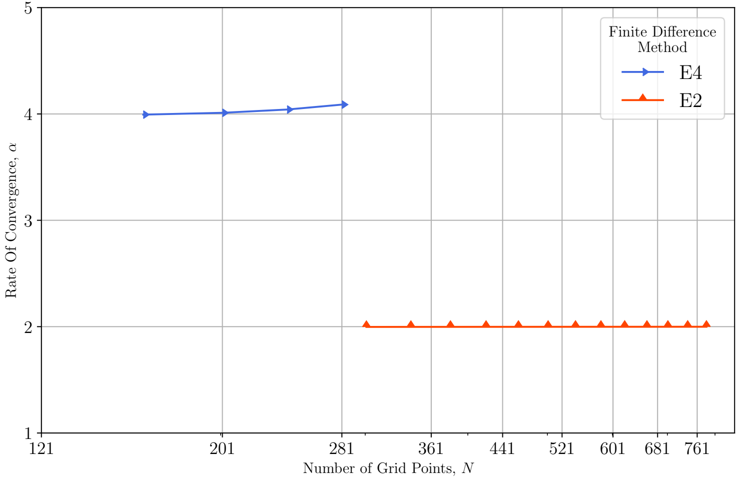
- There are exact solutions for the modal content in simplified mean flows.
- Code verification by the MES was performed using a uniform axial flow in a hard-walled annular duct

Eigenvalue (Axial Wavenumber) Comparison



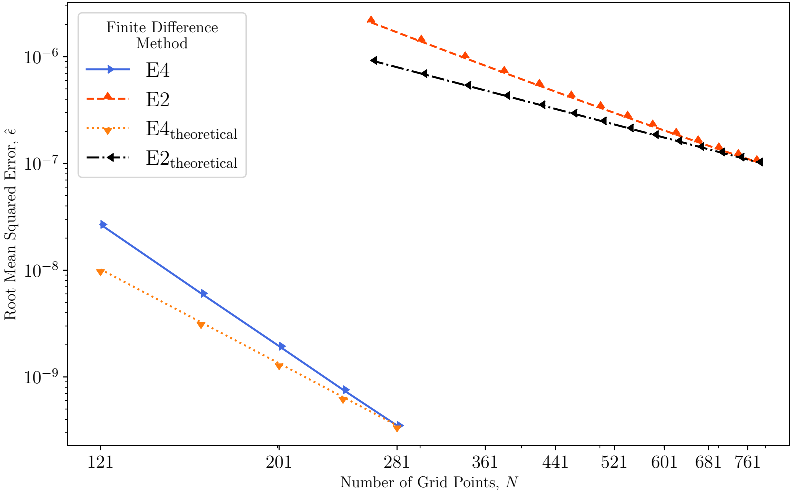
Both fourth and second order schemes perform as expected. The rate of decrease in error coincides with the exponent of the leading error terms

Eigenvalue (Axial Wavenumber) Comparison



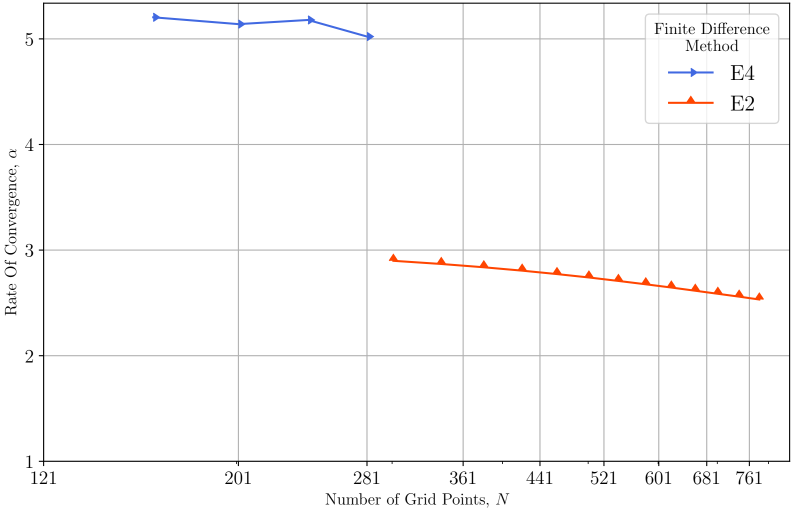
Both fourth and second order schemes perform as expected. The rate of decrease in error coincides with the exponent of the leading error terms

Eigenvector (Radial Pressure Mode) Comparison



The imposed lines are not parallel to the computed norms..

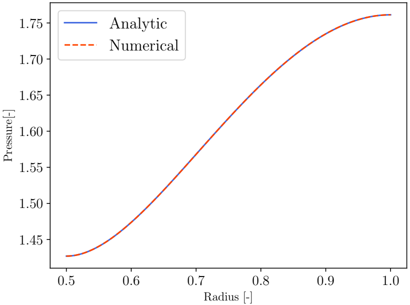
Eigenvector (Radial Pressure Mode) Comparison

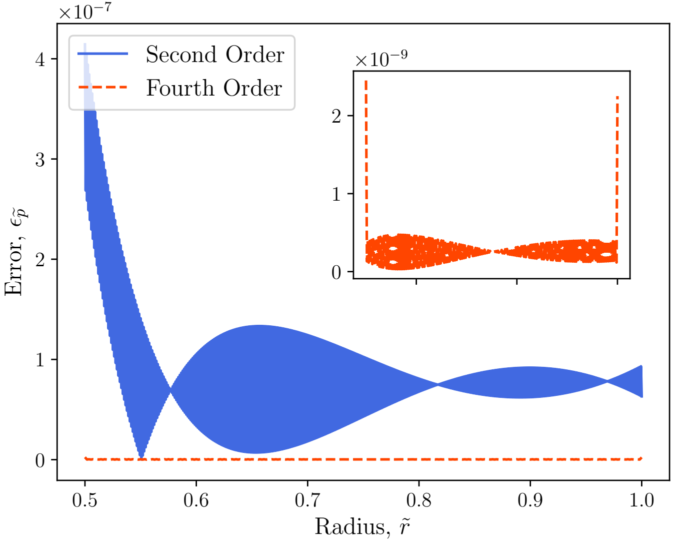


The computed slopes are higher than expected

Is *Swirl* converging to the right solution?

Examining the Error in the Radial Pressure Mode





Nonphysical oscillations and boundary errors are dominating the solution

Applying Filters to the Radial Pressure Modes

Will filters/artificial dissipation yield the expected OOA?

- Filters (also known as artificial dissipation operators) can eliminate the unresolved oscillations in the solution

Applying Filters to the Radial Pressure Modes

Will filters/artificial dissipation yield the expected OOA?

- Filters (also known as artificial dissipation operators) can eliminate the unresolved oscillations in the solution
- The higher the order of the filter, the less impact it will have on the original solution

Applying Filters to the Radial Pressure Modes

Will filters/artificial dissipation yield the expected OOA?

- Filters (also known as artificial dissipation operators) can eliminate the unresolved oscillations in the solution
- The higher the order of the filter, the less impact it will have on the original solution
- Eriksson proposed a method for conservative operators near boundaries (tested by Kennedy and Carpenter)

Applying Filters to the Radial Pressure Modes

Will filters/artificial dissipation yield the expected OOA?

- Filters (also known as artificial dissipation operators) can eliminate the unresolved oscillations in the solution
- The higher the order of the filter, the less impact it will have on the original solution
- Eriksson proposed a method for conservative operators near boundaries (tested by Kennedy and Carpenter)
- Rigby proposed a method where the operators were designed to match the performance of the interior instead

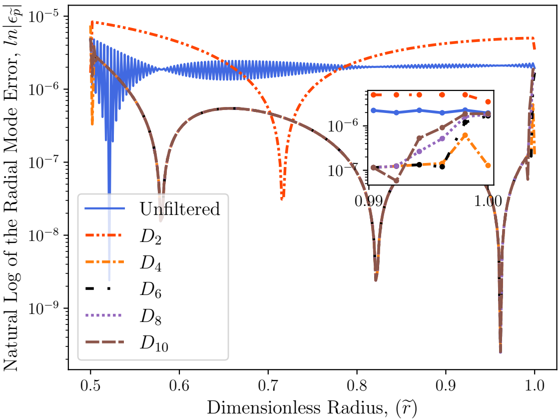
Applying Filters to the Radial Pressure Modes

Will filters/artificial dissipation yield the expected OOA?

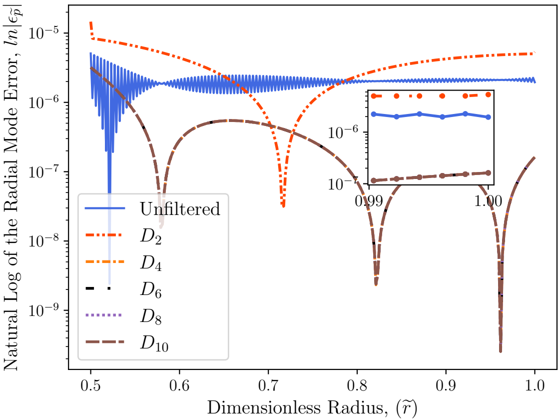
- Filters (also known as artificial dissipation operators) can eliminate the unresolved oscillations in the solution
- The higher the order of the filter, the less impact it will have on the original solution
- Eriksson proposed a method for conservative operators near boundaries (tested by Kennedy and Carpenter)
- Rigby proposed a method where the operators were designed to match the performance of the interior instead

The OOA was recomputed after applying Kennedy and Carpenter (K&C) and Rigby filters ($D2 - D10$)

Eigenvector (Radial Pressure Mode) Error With Various Orders of Filters - Second Order Study



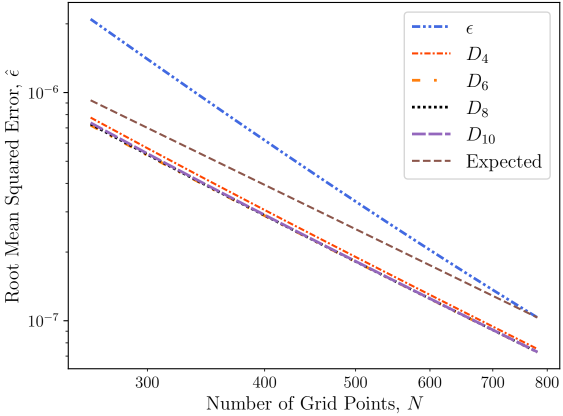
Kennedy and Carpenter



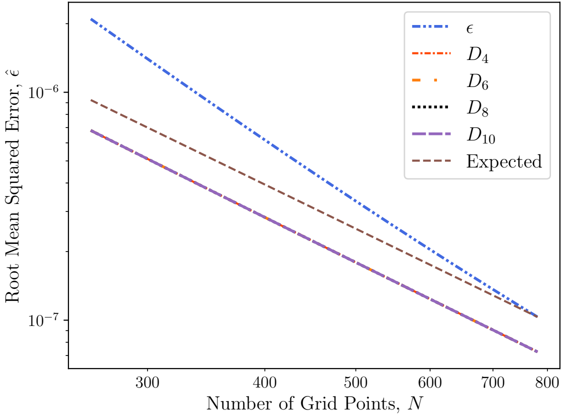
Rigby

Both filters mitigate oscillations, Rigby is more aggressive on the boundaries

Eigenvector (Radial Pressure Mode) Error With Various Orders of Filters - Second Order Study

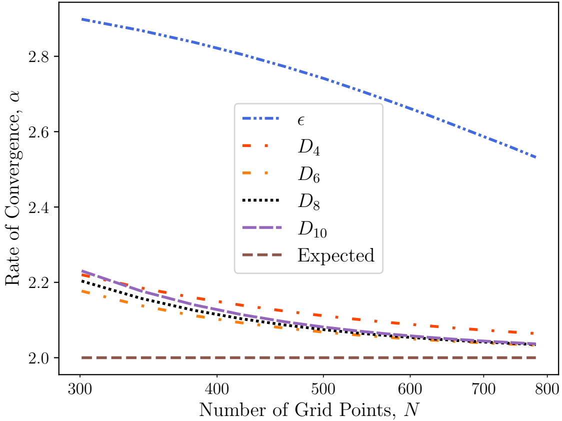


Kennedy and Carpenter

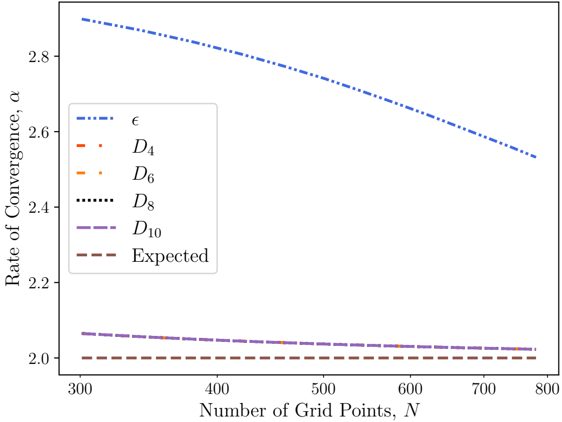


Rigby

Eigenvector (Radial Pressure Mode) Error With Various Orders of Filters - Second Order Study

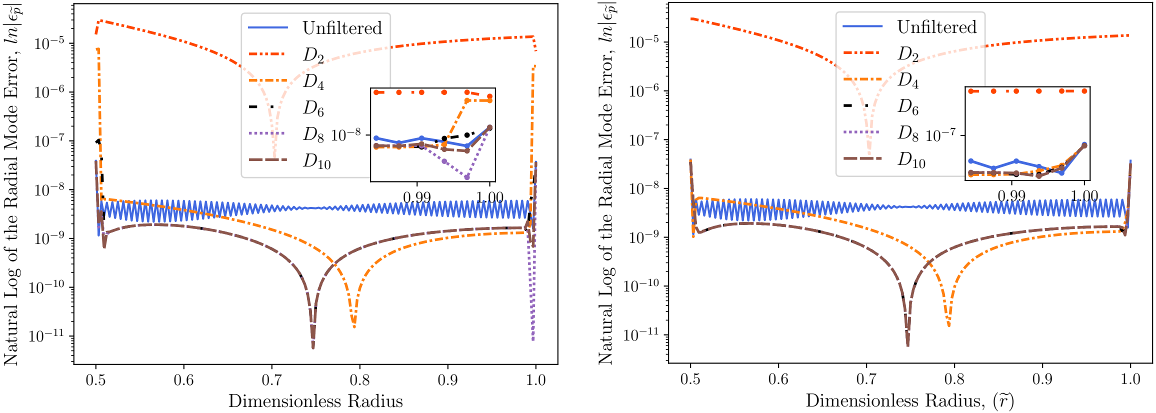


Kennedy and Carpenter



Rigby

Eigenvector (Radial Pressure Mode) Error With Various Orders of Filters - Fourth Order Study

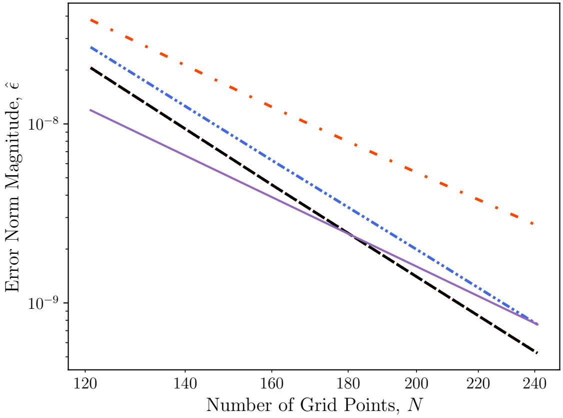


Kennedy and Carpenter

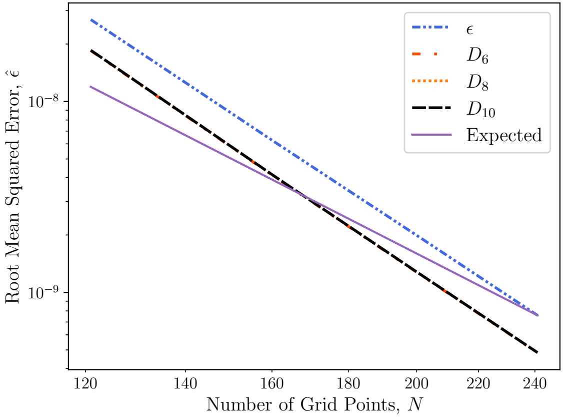
Rigby

Both filters mitigate oscillations, Rigby is more aggressive on the boundaries

Eigenvector (Radial Pressure Mode) Error With Various Orders of Filters - Fourth Order Study



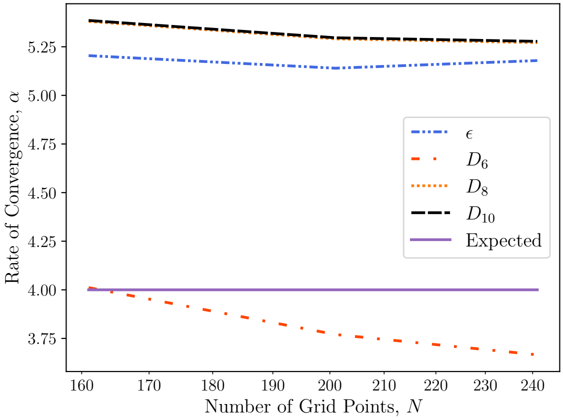
Kennedy and Carpenter



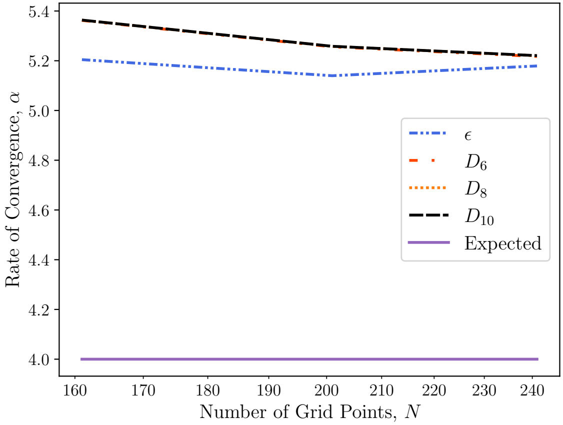
Rigby

Rigby operators show the most promise for an artificial dissipation operator

Eigenvector (Radial Pressure Mode) Error With Various Orders of Filters - Fourth Order Study



Kennedy and Carpenter



Rigby

Rigby operators show the most promise for an artificial dissipation operator

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

- Provides a way of “manufacturing” solution

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

- Provides a way of "manufacturing" solution
- For *Swirl*, MMS can't be used to verify the eigenvalue/vectors

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

- Provides a way of “manufacturing” solution
- For *Swirl*, MMS can’t be used to verify the eigenvalue/vectors

Pros and Cons of MES

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

- Provides a way of “manufacturing” solution
- For *Swirl*, MMS can’t be used to verify the eigenvalue/vectors

Pros and Cons of MES

- If solutions are available, they can be used for OOA testing

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

- Provides a way of “manufacturing” solution
- For *Swirl*, MMS can’t be used to verify the eigenvalue/vectors

Pros and Cons of MES

- If solutions are available, they can be used for OOA testing
- For *Swirl*, MES showed issues that MMS can’t due to boundary condition agreement being a requirement for the MS

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

- Provides a way of “manufacturing” solution
- For *Swirl*, MMS can’t be used to verify the eigenvalue/vectors

Pros and Cons of MES

- If solutions are available, they can be used for OOA testing
- For *Swirl*, MES showed issues that MMS can’t due to boundary condition agreement being a requirement for the MS
- Exact solutions are limited to idealized flows

Conclusions and Future Work

In summary, MMS/MES offers a means for robust code verification and assessing the performance of various integration, differencing, and artificial dissipation schemes and their BC implementation.

Pros and Cons of MMS

- Provides a way of "manufacturing" solution
- For *Swirl*, MMS can't be used to verify the eigenvalue/vectors

Pros and Cons of MES

- If solutions are available, they can be used for OOA testing
- For *Swirl*, MES showed issues that MMS can't due to boundary condition agreement being a requirement for the MS
- Exact solutions are limited to idealized flows

By using both MMS/MES on multiple code components, a path to improvements to the numerical approximations and BC implementation are outlined

Questions?

References I

K. A. Kousen. Eigenmode analysis of ducted flows with radially dependent axial and swirl components. In *CEAS/AIAA Joint Aeroacoustics Conference, 1 st, Munich, Germany*, pages 1085–1094, 1995.