Efficient Aircraft Flutter Analysis Using Model Order Reduction With Error Estimation

Brandon Lowe and Prof. David Zingg
Centre for Research in Sustainable Aviation

University of Toronto Institute for Aerospace Studies

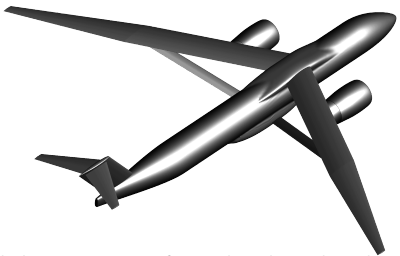


Institute for Aerospace Studies
UNIVERSITY OF TORONTO
Centre for Research in Sustainable Aviation

Advanced Modeling & Simulation (AMS) Seminar Series
NASA Ames Research Center, February 22nd, 2022

Motivation

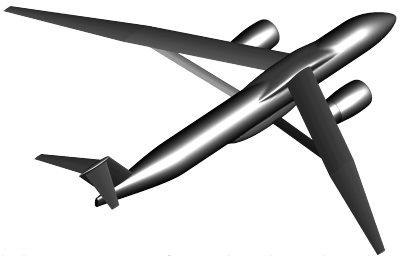
- Growing concerns about the **environmental sustainability** of the aviation industry has led to the consideration of **unconventional aircraft configurations**.
- Unconventional designs tend to include high-aspect-ratio wings and light-weight structures with increased flexibility, which may be more susceptible to **flutter**.
- Flutter analysis at transonic Mach number requires a **high-fidelity** aerodynamic model due to aerodynamic nonlinearities (phase lag in shock wave motion).



Example of high-aspect-ratio aircraft: strut-braced wing design by Timothy Chau

Motivation

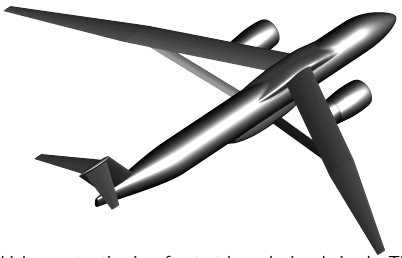
- Growing concerns about the **environmental sustainability** of the aviation industry has led to the consideration of **unconventional aircraft configurations**.
- Unconventional designs tend to include high-aspect-ratio wings and light-weight structures with increased flexibility, which may be more susceptible to **flutter**.
- Flutter analysis at transonic Mach number requires a **high-fidelity** aerodynamic model due to aerodynamic nonlinearities (phase lag in shock wave motion).



Example of high-aspect-ratio aircraft: strut-braced wing design by Timothy Chau

Motivation

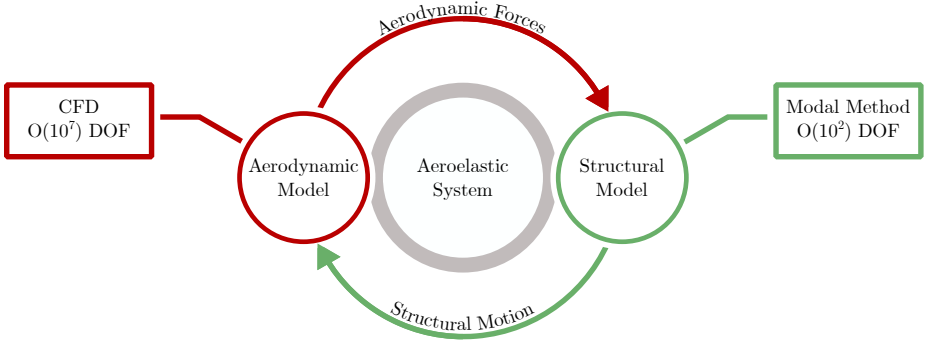
- Growing concerns about the **environmental sustainability** of the aviation industry has led to the consideration of **unconventional aircraft configurations**.
- Unconventional designs tend to include high-aspect-ratio wings and light-weight structures with increased flexibility, which may be more susceptible to **flutter**.
- Flutter analysis at transonic Mach number requires a **high-fidelity** aerodynamic model due to aerodynamic nonlinearities (phase lag in shock wave motion).



Example of high-aspect-ratio aircraft: strut-braced wing design by Timothy Chau

Motivation

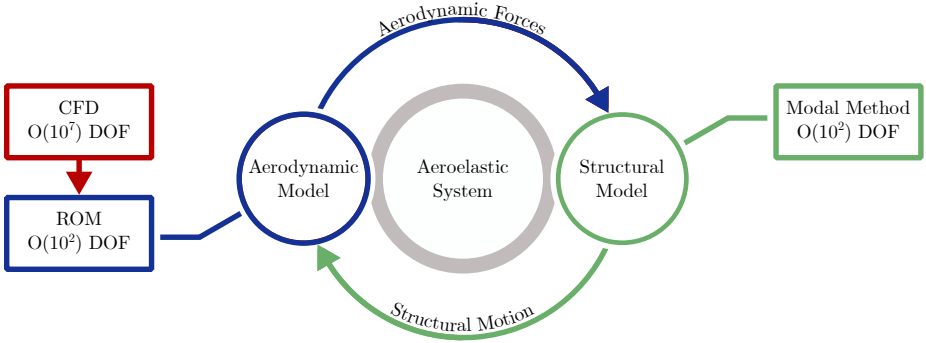
- Direct application of **computational fluid dynamics** (CFD) remains computationally expensive for flutter analysis due to the large number of degrees of freedom.
- A **reduced-order model** (ROM) can be used to approximate the output of the high-dimensional CFD model.
  - **Rapid**: computationally efficient to train and run
  - **Reliable**: efficient error estimate to quantify accuracy
  - **Automated**: minimal user knowledge



Motivation

- Direct application of **computational fluid dynamics** (CFD) remains computationally expensive for flutter analysis due to the large number of degrees of freedom.

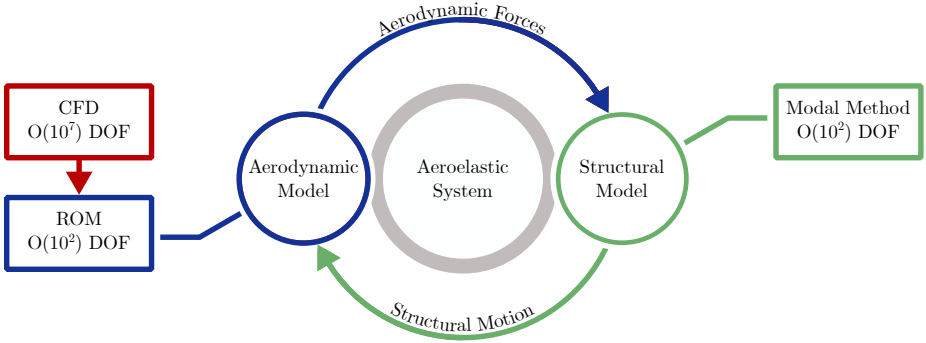
- A **reduced-order model** (ROM) can be used to approximate the output of the high-dimensional CFD model.
  - **Rapid**: computationally efficient to train and run
  - **Reliable**: efficient error estimate to quantify accuracy
  - **Automated**: minimal user knowledge



Motivation

- Direct application of **computational fluid dynamics** (CFD) remains computationally expensive for flutter analysis due to the large number of degrees of freedom.

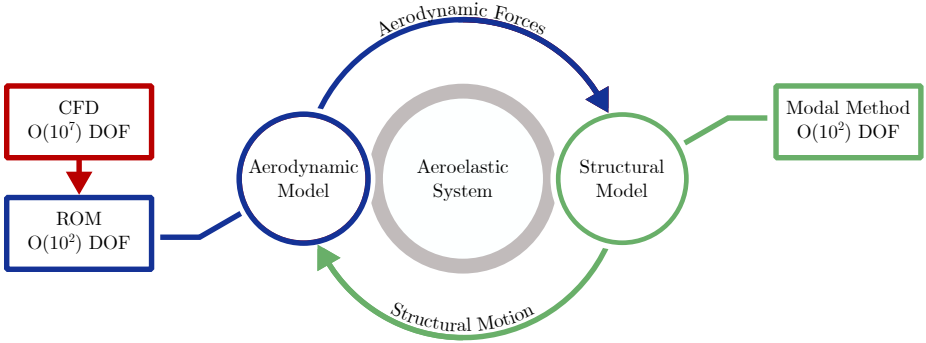
- A **reduced-order model** (ROM) can be used to approximate the output of the high-dimensional CFD model.
  - **Rapid**: computationally efficient to train and run
  - **Reliable**: efficient error estimate to quantify accuracy
  - **Automated**: minimal user knowledge



Motivation

- Direct application of **computational fluid dynamics** (CFD) remains computationally expensive for flutter analysis due to the large number of degrees of freedom.

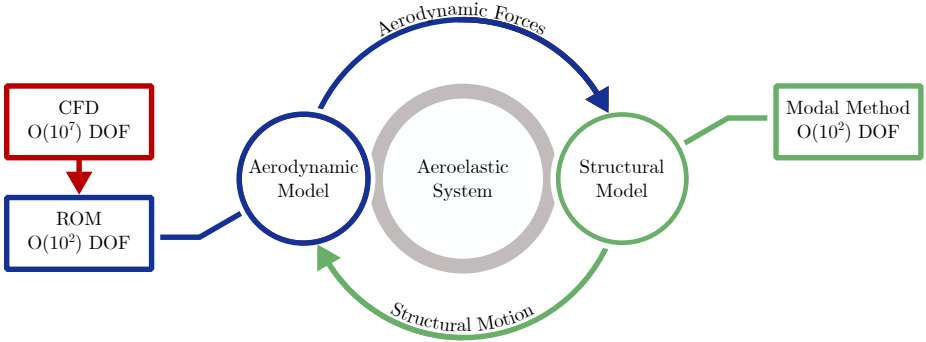
- A **reduced-order model** (ROM) can be used to approximate the output of the high-dimensional CFD model.
  - **Rapid**: computationally efficient to train and run
  - **Reliable**: efficient error estimate to quantify accuracy
  - **Automated**: minimal user knowledge



Motivation

- Direct application of **computational fluid dynamics** (CFD) remains computationally expensive for flutter analysis due to the large number of degrees of freedom.

- A **reduced-order model** (ROM) can be used to approximate the output of the high-dimensional CFD model.
  - **Rapid**: computationally efficient to train and run
  - **Reliable**: efficient error estimate to quantify accuracy
  - **Automated**: minimal user knowledge



Model Order Reduction

Two categories of ROMs.

- **Data-driven ROMs**: Constructed using input/output data

Advantages	Disadvantages
○ Easily applied to any computational model ○ Applied to experimental data	○ No relation between ROM and HDM states ○ No direct method for error estimation ○ Applicable only for parameter regions at which trained

- **Projection-based ROMs**: Constructed by projecting the high-dimensional operators onto a reduced space

Advantages	Disadvantages
○ Explicit relation between ROM and HDM states ○ Computable error estimates/bounds ○ Provides better approximations at untrained parameters[1]	○ Highly intrusive: requires access to HDM operators ○ High development times

HDM = High-Dimensional Model (CFD Model)

ROM = Reduced-Order Model

[1] *Problem dependent*

Model Order Reduction

Two categories of ROMs.

- **Data-driven ROMs**: Constructed using input/output data

Advantages	Disadvantages
○ Easily applied to any computational model ○ Applied to experimental data	○ No relation between ROM and HDM states ○ No direct method for error estimation ○ Applicable only for parameter regions at which trained

- **Projection-based ROMs**: Constructed by projecting the high-dimensional operators onto a reduced space

Advantages	Disadvantages
○ Explicit relation between ROM and HDM states ○ Computable error estimates/bounds ○ Provides better approximations at untrained parameters[1]	○ Highly intrusive: requires access to HDM operators ○ High development times

HDM = High-Dimensional Model (CFD Model)

ROM = Reduced-Order Model

[1] *Problem dependent*

Modeling Flutter



- Assume we have appropriate mathematical models for the structural and aerodynamic equations.
- After discretization in space, the coupled aerodynamic-structural model can be written as,

$$\overbrace{\mathbb{M}(V)}^{\text{mass}} \frac{d\boldsymbol{q}}{dt} = \overbrace{\boldsymbol{R}(\boldsymbol{q}, V)}^{\text{residual}} \qquad \begin{aligned} \boldsymbol{q} &= \text{aerodynamic + structural states,} \\ V &= \text{parameter: speed index} \end{aligned}$$

- Assume steady-state solution $\boldsymbol{q}_0$ is non-parametric (true for symmetric wings at 0° AoA),

$$\boldsymbol{R}(\boldsymbol{q}_0) = 0, \qquad \boldsymbol{q}_0 \neq \boldsymbol{q}_0(V)$$

- **Parameter sweep**: find the flutter speed index V_f such that $\text{Real}\{\lambda\} = 0$,

$$\mathbb{A}\boldsymbol{v} = \lambda \mathbb{M}\boldsymbol{v}, \qquad \begin{aligned} \mathbb{A} &= \mathbb{A}(V) \equiv \left.\frac{\partial \boldsymbol{R}(V)}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}_0} \\ \mathbb{M} &= \mathbb{M}(V) \end{aligned}$$



- Assume we have appropriate mathematical models for the structural and aerodynamic equations.

- After discretization in space, the coupled aerodynamic-structural model can be written as,

$$\overbrace{\mathbb{M}(V)}^{\text{mass}} \frac{d\boldsymbol{q}}{dt} = \overbrace{\boldsymbol{R}(\boldsymbol{q}, V)}^{\text{residual}}$$

$\boldsymbol{q} =$ aerodynamic + structural states,
$V =$ parameter: speed index

- Assume steady-state solution $\boldsymbol{q}_0$ is non-parametric (true for symmetric wings at 0° AoA),

$$\boldsymbol{R}(\boldsymbol{q}_0) = 0, \qquad \boldsymbol{q}_0 \neq \boldsymbol{q}_0(V)$$

- **Parameter sweep**: find the flutter speed index V_f such that $\mathrm{Real}\{\lambda\} = 0$,

$$\mathbb{A}\boldsymbol{v} = \lambda \mathbb{M}\boldsymbol{v}, \qquad \mathbb{A} = \mathbb{A}(V) \equiv \left.\frac{\partial \boldsymbol{R}(V)}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}_0}, \qquad \mathbb{M} = \mathbb{M}(V)$$



- Assume we have appropriate mathematical models for the structural and aerodynamic equations.

- After discretization in space, the coupled aerodynamic-structural model can be written as,

$$\overbrace{\mathbb{M}(V)}^{\text{mass}} \frac{d\boldsymbol{q}}{dt} = \overbrace{\boldsymbol{R}(\boldsymbol{q}, V)}^{\text{residual}} \qquad \begin{aligned} \boldsymbol{q} &= \text{aerodynamic + structural states,} \\ V &= \text{parameter: speed index} \end{aligned}$$

- Assume steady-state solution $\boldsymbol{q}_0$ is non-parametric (true for symmetric wings at 0° AoA),

$$\boldsymbol{R}(\boldsymbol{q}_0) = 0, \qquad \boldsymbol{q}_0 \neq \boldsymbol{q}_0(V)$$

- **Parameter sweep**: find the flutter speed index V_f such that $\text{Real}\{\lambda\} = 0$,

$$\mathbb{A}\boldsymbol{v} = \lambda \mathbb{M}\boldsymbol{v}, \qquad \mathbb{A} = \mathbb{A}(V) \equiv \left.\frac{\partial \boldsymbol{R}(V)}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}_0}$$
$$\mathbb{M} = \mathbb{M}(V)$$



- Assume we have appropriate mathematical models for the structural and aerodynamic equations.

- After discretization in space, the coupled aerodynamic-structural model can be written as,

$$\overbrace{\mathbb{M}(V)}^{\text{mass}} \frac{d\boldsymbol{q}}{dt} = \overbrace{\boldsymbol{R}(\boldsymbol{q}, V)}^{\text{residual}} \qquad \begin{aligned} \boldsymbol{q} &= \text{aerodynamic + structural states,} \\ V &= \text{parameter: speed index} \end{aligned}$$

- Assume steady-state solution $\boldsymbol{q}_0$ is non-parametric (true for symmetric wings at 0° AoA),

$$\boldsymbol{R}(\boldsymbol{q}_0) = 0, \qquad \boldsymbol{q}_0 \neq \boldsymbol{q}_0(V)$$

- **Parameter sweep**: find the flutter speed index V_f such that Real$\{\lambda\} = 0$,

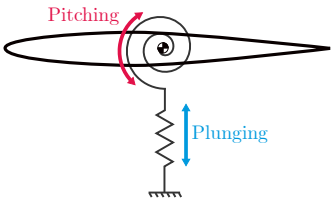
$$\mathbb{A}\boldsymbol{v} = \lambda \mathbb{M}\boldsymbol{v}, \qquad \begin{aligned} \mathbb{A} &= \mathbb{A}(V) \equiv \left.\frac{\partial \boldsymbol{R}(V)}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}_0} \\ \mathbb{M} &= \mathbb{M}(V) \end{aligned}$$

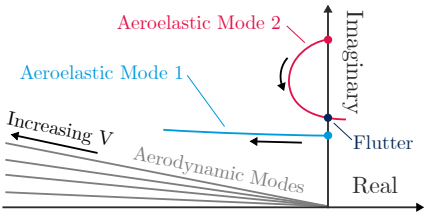


Aeroelastic Eigenvalues

Eigenvalues originating from the wind-off structural modes are important for flutter analysis.

- At low speed index V, we have two distinct groups of eigenvalues,
  - **Aeroelastic modes**: near the natural frequencies of the structure
  - **Aerodynamic modes**: near zero
- One of the aeroelastic modes will become unstable as the speed index V increases.
- We wish to find the flutter speed index V_f at which one of the aeroelastic modes crosses the imaginary axis.

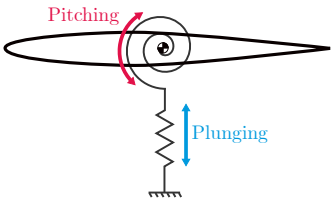


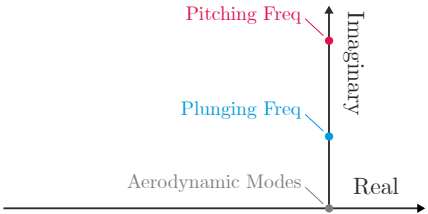


Aeroelastic Eigenvalues

Eigenvalues originating from the wind-off structural modes are important for flutter analysis.

- At low speed index V, we have two distinct groups of eigenvalues,
  - **Aeroelastic modes**: near the natural frequencies of the structure
  - **Aerodynamic modes**: near zero
- One of the aeroelastic modes will become unstable as the speed index V increases.
- We wish to find the flutter speed index V_f at which one of the aeroelastic modes crosses the imaginary axis.

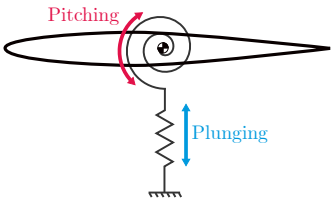


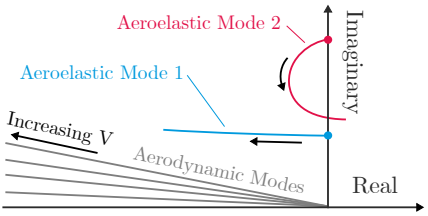


Aeroelastic Eigenvalues

Eigenvalues originating from the wind-off structural modes are important for flutter analysis.

- At low speed index V, we have two distinct groups of eigenvalues,
  - **Aeroelastic modes**: near the natural frequencies of the structure
  - **Aerodynamic modes**: near zero
- One of the aeroelastic modes will become unstable as the speed index V increases.
- We wish to find the flutter speed index V_f at which one of the aeroelastic modes crosses the imaginary axis.

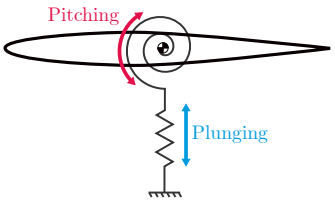


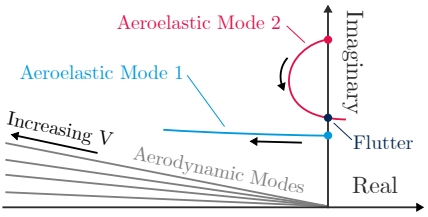


Aeroelastic Eigenvalues

Eigenvalues originating from the wind-off structural modes are important for flutter analysis.

- At low speed index V, we have two distinct groups of eigenvalues,
  - **Aeroelastic modes**: near the natural frequencies of the structure
  - **Aerodynamic modes**: near zero
- One of the aeroelastic modes will become unstable as the speed index V increases.
- We wish to find the flutter speed index V_f at which one of the aeroelastic modes crosses the imaginary axis.





High-Dimensional Aeroelastic Model



The aeroelastic model for flutter analysis is linearized about a nonlinear steady-state solution,

$$\boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a},0}) = 0, \qquad \begin{aligned} \boldsymbol{R}_{\mathrm{a}} &= \text{aerodynamic residual} \\ \boldsymbol{q}_{\mathrm{a}} &= \text{aerodynamic state} \end{aligned}$$

This equation is solved using an efficient parallel flow solver[1] previously developed at the University of Toronto:

- Newton-Krylov-Schur multiblock implicit flow solver,
- capable of solving both the Euler* and RANS equations,
- spatial discretization using 2^{nd}-order-accurate summation-by-parts operators,
- boundary conditions weakly imposed using simultaneous-approximation terms.

*Work in this dissertation has focused on applications of the Euler equations



[1] *Hicken, J. E., and Zingg, D. W., AIAA Journal, Vol. 46, No. 11, 2008, pp. 2773–2786.*

The aeroelastic model for flutter analysis is linearized about a nonlinear steady-state solution,

$$\boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a},0}) = 0, \qquad \begin{aligned} \boldsymbol{R}_{\mathrm{a}} &= \text{aerodynamic residual} \\ \boldsymbol{q}_{\mathrm{a}} &= \text{aerodynamic state} \end{aligned}$$

This equation is solved using an efficient parallel flow solver[1] previously developed at the University of Toronto:

- Newton-Krylov-Schur multiblock implicit flow solver,
- capable of solving both the Euler* and RANS equations,
- spatial discretization using 2^{nd}-order-accurate summation-by-parts operators,
- boundary conditions weakly imposed using simultaneous-approximation terms.

*Work in this dissertation has focused on applications of the Euler equations



[1] *Hicken, J. E., and Zingg, D. W., AIAA Journal, Vol. 46, No. 11, 2008, pp. 2773–2786.*

The aeroelastic model for flutter analysis is linearized about a nonlinear steady-state solution,

$$\boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a},0}) = 0, \qquad \begin{aligned} \boldsymbol{R}_{\mathrm{a}} &= \text{aerodynamic residual} \\ \boldsymbol{q}_{\mathrm{a}} &= \text{aerodynamic state} \end{aligned}$$

This equation is solved using an efficient parallel flow solver[1] previously developed at the University of Toronto:

- Newton-Krylov-Schur multiblock implicit flow solver,
- capable of solving both the Euler* and RANS equations,
- spatial discretization using 2^{nd}-order-accurate summation-by-parts operators,
- boundary conditions weakly imposed using simultaneous-approximation terms.

*Work in this dissertation has focused on applications of the Euler equations



[1] *Hicken, J. E., and Zingg, D. W., AIAA Journal, Vol. 46, No. 11, 2008, pp. 2773–2786.*

Nonlinear Flow Solver

The aeroelastic model for flutter analysis is linearized about a nonlinear steady-state solution,

$$\boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a},0}) = 0, \qquad \begin{aligned} \boldsymbol{R}_{\mathrm{a}} &= \text{aerodynamic residual} \\ \boldsymbol{q}_{\mathrm{a}} &= \text{aerodynamic state} \end{aligned}$$

This equation is solved using an efficient parallel flow solver[1] previously developed at the University of Toronto:

- Newton-Krylov-Schur multiblock implicit flow solver,
- capable of solving both the Euler* and RANS equations,
- spatial discretization using 2^{nd}-order-accurate summation-by-parts operators,
- boundary conditions weakly imposed using simultaneous-approximation terms.

*Work in this dissertation has focused on applications of the Euler equations



[1] *Hicken, J. E., and Zingg, D. W., AIAA Journal, Vol. 46, No. 11, 2008, pp. 2773–2786.*

The aeroelastic model for flutter analysis is linearized about a nonlinear steady-state solution,

$$\boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a},0}) = 0, \qquad \begin{aligned} \boldsymbol{R}_{\mathrm{a}} &= \text{aerodynamic residual} \\ \boldsymbol{q}_{\mathrm{a}} &= \text{aerodynamic state} \end{aligned}$$

This equation is solved using an efficient parallel flow solver[1] previously developed at the University of Toronto:

- Newton-Krylov-Schur multiblock implicit flow solver,
- capable of solving both the Euler* and RANS equations,
- spatial discretization using 2^{nd}-order-accurate summation-by-parts operators,
- boundary conditions weakly imposed using simultaneous-approximation terms.

*Work in this dissertation has focused on applications of the Euler equations



[1] *Hicken, J. E., and Zingg, D. W., AIAA Journal, Vol. 46, No. 11, 2008, pp. 2773–2786.*

This work applies the linearized semi-discrete Euler equations.

- The **semi-discretized nonlinear Euler equations** are

$$\frac{d\left(J^{-1}\boldsymbol{q}_{\mathrm{a}}\right)}{dt} + \boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a}}, \boldsymbol{q}_{\mathrm{s}}, \dot{\boldsymbol{q}}_{\mathrm{s}}) = \boldsymbol{0},$$

J^{-1} = metric Jacobian
$\boldsymbol{q}_{\mathrm{s}}$ = gen. structural coord.
$\dot{\boldsymbol{q}}_{\mathrm{s}}$ = gen. structural vel.

- Linearize with respect to the aerodynamic and structural states,

$$\boldsymbol{q}_{\mathrm{a}}(t) = \overbrace{\boldsymbol{q}_{\mathrm{a},0}}^{\text{steady}} + \overbrace{\boldsymbol{u}_{\mathrm{a}}(t)}^{\text{fluctuations}}, \qquad \boldsymbol{q}_{\mathrm{s}}(t) = \boldsymbol{q}_{\mathrm{s},0} + \boldsymbol{u}_{\mathrm{s}}(t), \qquad \dot{\boldsymbol{q}}_{\mathrm{s}}(t) = \dot{\boldsymbol{u}}_{\mathrm{s}}(t),$$

- **Aerodynamic Model**: high-dimensional linearized Euler equations,

$$\underbrace{J_0^{-1}}_{E_{\mathrm{a}}} \frac{d\boldsymbol{u}_{\mathrm{a}}}{dt} = \underbrace{\frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{q}_{\mathrm{a}}}}_{A_{\mathrm{a}}} \boldsymbol{u}_{\mathrm{a}} - \underbrace{\frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{q}_{\mathrm{s}}}}_{B_{\mathrm{a},1}} \boldsymbol{u}_{\mathrm{s}} - \underbrace{\left(\frac{\partial}{\partial \dot{\boldsymbol{q}}_{\mathrm{s}}}\left(\frac{dJ^{-1}}{dt}\right)\boldsymbol{q}_{\mathrm{a},0} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \dot{\boldsymbol{q}}_{\mathrm{s}}}\right)}_{B_{\mathrm{a},2}} \dot{\boldsymbol{u}}_{\mathrm{s}}.$$



This work applies the linearized semi-discrete Euler equations.

- The **semi-discretized nonlinear Euler equations** are

$$\frac{d\left(J^{-1}\boldsymbol{q}_{\mathrm{a}}\right)}{dt} + \boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a}}, \boldsymbol{q}_{\mathrm{s}}, \dot{\boldsymbol{q}}_{\mathrm{s}}) = \boldsymbol{0},$$

J^{-1} = metric Jacobian
$\boldsymbol{q}_{\mathrm{s}}$ = gen. structural coord.
$\dot{\boldsymbol{q}}_{\mathrm{s}}$ = gen. structural vel.

- Linearize with respect to the aerodynamic and structural states,

$$\boldsymbol{q}_{\mathrm{a}}(t) = \overbrace{\boldsymbol{q}_{\mathrm{a},0}}^{\text{steady}} + \overbrace{\boldsymbol{u}_{\mathrm{a}}(t)}^{\text{fluctuations}}, \qquad \boldsymbol{q}_{\mathrm{s}}(t) = \boldsymbol{q}_{\mathrm{s},0} + \boldsymbol{u}_{\mathrm{s}}(t), \qquad \dot{\boldsymbol{q}}_{\mathrm{s}}(t) = \dot{\boldsymbol{u}}_{\mathrm{s}}(t),$$

- **Aerodynamic Model**: high-dimensional linearized Euler equations,

$$\underbrace{J_0^{-1}}_{E_{\mathrm{a}}} \frac{d\boldsymbol{u}_{\mathrm{a}}}{dt} = \underbrace{\frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{q}_{\mathrm{a}}}}_{A_{\mathrm{a}}} \boldsymbol{u}_{\mathrm{a}} - \underbrace{\frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{q}_{\mathrm{s}}}}_{B_{\mathrm{a},1}} \boldsymbol{u}_{\mathrm{s}} - \underbrace{\left(\frac{\partial}{\partial \dot{\boldsymbol{q}}_{\mathrm{s}}}\left(\frac{dJ^{-1}}{dt}\right)\boldsymbol{q}_{\mathrm{a},0} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \dot{\boldsymbol{q}}_{\mathrm{s}}}\right)}_{B_{\mathrm{a},2}} \dot{\boldsymbol{u}}_{\mathrm{s}}.$$



This work applies the linearized semi-discrete Euler equations.

- The **semi-discretized nonlinear Euler equations** are

$$\frac{d\left(J^{-1}\boldsymbol{q}_{\mathrm{a}}\right)}{dt} + \boldsymbol{R}_{\mathrm{a}}(\boldsymbol{q}_{\mathrm{a}}, \boldsymbol{q}_{\mathrm{s}}, \dot{\boldsymbol{q}}_{\mathrm{s}}) = \boldsymbol{0},$$

J^{-1} = metric Jacobian
$\boldsymbol{q}_{\mathrm{s}}$ = gen. structural coord.
$\dot{\boldsymbol{q}}_{\mathrm{s}}$ = gen. structural vel.

- Linearize with respect to the aerodynamic and structural states,

$$\boldsymbol{q}_{\mathrm{a}}(t) = \overbrace{\boldsymbol{q}_{\mathrm{a},0}}^{\text{steady}} + \overbrace{\boldsymbol{u}_{\mathrm{a}}(t)}^{\text{fluctuations}}, \qquad \boldsymbol{q}_{\mathrm{s}}(t) = \boldsymbol{q}_{\mathrm{s},0} + \boldsymbol{u}_{\mathrm{s}}(t), \qquad \dot{\boldsymbol{q}}_{\mathrm{s}}(t) = \dot{\boldsymbol{u}}_{\mathrm{s}}(t),$$

- **Aerodynamic Model**: high-dimensional linearized Euler equations,

$$\underbrace{J_0^{-1}}_{E_{\mathrm{a}}} \frac{d\boldsymbol{u}_{\mathrm{a}}}{dt} = \underbrace{\frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{q}_{\mathrm{a}}}}_{A_{\mathrm{a}}} \boldsymbol{u}_{\mathrm{a}} - \underbrace{\frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{q}_{\mathrm{s}}}}_{B_{\mathrm{a},1}} \boldsymbol{u}_{\mathrm{s}} - \underbrace{\left(\frac{\partial}{\partial \dot{\boldsymbol{q}}_{\mathrm{s}}}\left(\frac{dJ^{-1}}{dt}\right)\boldsymbol{q}_{\mathrm{a},0} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \dot{\boldsymbol{q}}_{\mathrm{s}}}\right)}_{B_{\mathrm{a},2}} \dot{\boldsymbol{u}}_{\mathrm{s}}.$$



High-Dimensional Aeroelastic Model

- The (linearized) **governing set of equations** for aeroelastic analysis are:

  Structural Model $\quad \frac{d^2 \boldsymbol{u}_\mathrm{s}}{dt^2} = A_\mathrm{s} \boldsymbol{u}_\mathrm{s} + C_\mathrm{s} \boldsymbol{u}_\mathrm{a}$

  Trivial Equation $\quad \frac{d \boldsymbol{u}_\mathrm{s}}{dt} = \dot{\boldsymbol{u}}_\mathrm{s}$

  Aerodynamic Model (Linearized Euler) $\quad E_\mathrm{a} \frac{d \boldsymbol{u}_\mathrm{a}}{dt} = A_\mathrm{a} \boldsymbol{u}_\mathrm{a} + B_{\mathrm{a},1} \boldsymbol{u}_\mathrm{s} + B_{\mathrm{a},2} \frac{d \boldsymbol{u}_\mathrm{s}}{dt}$

- Coupling these equations leads to the monolithic aeroelastic system:

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & E_\mathrm{a} \end{bmatrix}}_{\mathbb{M}} \frac{d}{dt} \underbrace{\begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{a} \end{bmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{bmatrix} 0 & A_\mathrm{s} & C_\mathrm{s} \\ I & 0 & 0 \\ B_{\mathrm{a},2} & B_{\mathrm{a},1} & A_\mathrm{a} \end{bmatrix}}_{\mathbb{A}} \begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{a} \end{bmatrix}$$

  or equivalently

$$\mathbb{M} \frac{d \boldsymbol{u}}{dt} = \mathbb{A} \boldsymbol{u}$$

- The **stability** of the aeroelastic system can be determined by the eigenvalues problem:

$$\mathbb{A} \boldsymbol{v} = \lambda \mathbb{M} \boldsymbol{v}.$$



High-Dimensional Aeroelastic Model

- The (linearized) **governing set of equations** for aeroelastic analysis are:

Structural Model $\quad \frac{d^2\boldsymbol{u}_\mathrm{s}}{dt^2} = A_\mathrm{s}\boldsymbol{u}_\mathrm{s} + C_\mathrm{s}\boldsymbol{u}_\mathrm{a}$

Trivial Equation $\quad \frac{d\boldsymbol{u}_\mathrm{s}}{dt} = \dot{\boldsymbol{u}}_\mathrm{s}$

Aerodynamic Model (Linearized Euler) $\quad E_\mathrm{a}\frac{d\boldsymbol{u}_\mathrm{a}}{dt} = A_\mathrm{a}\boldsymbol{u}_\mathrm{a} + B_{\mathrm{a},1}\boldsymbol{u}_\mathrm{s} + B_{\mathrm{a},2}\frac{d\boldsymbol{u}_\mathrm{s}}{dt}$

- Coupling these equations leads to the monolithic aeroelastic system:

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & E_\mathrm{a} \end{bmatrix}}_{\mathbb{M}} \frac{d}{dt} \underbrace{\begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{a} \end{bmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{bmatrix} 0 & A_\mathrm{s} & C_\mathrm{s} \\ I & 0 & 0 \\ B_{\mathrm{a},2} & B_{\mathrm{a},1} & A_\mathrm{a} \end{bmatrix}}_{\mathbb{A}} \begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{a} \end{bmatrix}$$

or equivalently

$$\mathbb{M}\frac{d\boldsymbol{u}}{dt} = \mathbb{A}\boldsymbol{u}$$

- The **stability** of the aeroelastic system can be determined by the eigenvalues problem:

$$\mathbb{A}\boldsymbol{v} = \lambda\mathbb{M}\boldsymbol{v}.$$

- The (linearized) **governing set of equations** for aeroelastic analysis are:

Structural Model $\quad \frac{d^2 \boldsymbol{u}_\mathrm{s}}{dt^2} = A_\mathrm{s}\boldsymbol{u}_\mathrm{s} + C_\mathrm{s}\boldsymbol{u}_\mathrm{a}$

Trivial Equation $\quad \frac{d\boldsymbol{u}_\mathrm{s}}{dt} = \dot{\boldsymbol{u}}_\mathrm{s}$

Aerodynamic Model (Linearized Euler) $\quad E_\mathrm{a}\frac{d\boldsymbol{u}_\mathrm{a}}{dt} = A_\mathrm{a}\boldsymbol{u}_\mathrm{a} + B_{\mathrm{a},1}\boldsymbol{u}_\mathrm{s} + B_{\mathrm{a},2}\frac{d\boldsymbol{u}_\mathrm{s}}{dt}$

- Coupling these equations leads to the monolithic aeroelastic system:

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & E_\mathrm{a} \end{bmatrix}}_{\mathbb{M}} \frac{d}{dt} \underbrace{\begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{a} \end{bmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{bmatrix} 0 & A_\mathrm{s} & C_\mathrm{s} \\ I & 0 & 0 \\ B_{\mathrm{a},2} & B_{\mathrm{a},1} & A_\mathrm{a} \end{bmatrix}}_{\mathbb{A}} \begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{a} \end{bmatrix}$$

or equivalently

$$\mathbb{M}\frac{d\boldsymbol{u}}{dt} = \mathbb{A}\boldsymbol{u}$$

- The **stability** of the aeroelastic system can be determined by the eigenvalues problem:

$$\mathbb{A}\boldsymbol{v} = \lambda\mathbb{M}\boldsymbol{v}.$$



Model Order Reduction



Projection-Based Model Order Reduction

Here we show an overview of **projection-based model order reduction** for linear systems.

- Approximate the aerodynamics state using a **reduced basis** Φ:

$$\boldsymbol{u}_{\mathrm{a}}(t) \approx \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}}(t).$$

- This leads to the residual for the aerodynamic model,

$$\boldsymbol{R}_{\mathrm{ROM}} = E_{\mathrm{a}} \Phi \frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - A_{\mathrm{a}} \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}} - B_{\mathrm{a},1} \boldsymbol{u}_{\mathrm{s}} - B_{\mathrm{a},2} \frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}.$$

- Force the residual to be orthogonal to the test basis Ψ (Petrov-Galerkin Condition),

$$\Psi^T \boldsymbol{R}_{\mathrm{ROM}} = 0$$

$$\underbrace{\Psi^T E_{\mathrm{a}} \Phi}_{\tilde{E}_{\mathrm{a}}} \frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - \underbrace{\Psi^T A_{\mathrm{a}} \Phi}_{\tilde{A}_{\mathrm{a}}} \tilde{\boldsymbol{u}}_{\mathrm{a}} - \underbrace{\Psi^T B_{\mathrm{a},1}}_{\tilde{B}_{\mathrm{a},1}} \boldsymbol{u}_{\mathrm{s}} - \underbrace{\Psi^T B_{\mathrm{a},2}}_{\tilde{B}_{\mathrm{a},2}} \frac{d\boldsymbol{u}_{\mathrm{s}}}{dt} = 0.$$

- We obtain the **aerodynamic ROM**,

$$\tilde{E}_{\mathrm{a}} \frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} = \tilde{A}_{\mathrm{a}} \boldsymbol{u}_{\mathrm{a}} + \tilde{B}_{\mathrm{a},1} \boldsymbol{u}_{\mathrm{s}} + \tilde{B}_{\mathrm{a},2} \frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}, \qquad \text{with } \dim(\tilde{E}_{\mathrm{a}}, \tilde{A}_{\mathrm{a}}) \ll \dim(E_{\mathrm{a}}, A_{\mathrm{a}})$$



Projection-Based Model Order Reduction

Here we show an overview of **projection-based model order reduction** for linear systems.

- Approximate the aerodynamics state using a **reduced basis** Φ:

$$\boldsymbol{u}_{\mathrm{a}}(t) \approx \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}}(t).$$

- This leads to the residual for the aerodynamic model,

$$\boldsymbol{R}_{\mathrm{ROM}} = E_{\mathrm{a}}\Phi\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - A_{\mathrm{a}}\Phi\tilde{\boldsymbol{u}}_{\mathrm{a}} - B_{\mathrm{a},1}\boldsymbol{u}_{\mathrm{s}} - B_{\mathrm{a},2}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}.$$

- Force the residual to be orthogonal to the test basis Ψ (Petrov-Galerkin Condition),

$$\Psi^T \boldsymbol{R}_{\mathrm{ROM}} = 0$$

$$\underbrace{\Psi^T E_{\mathrm{a}}\Phi}_{\tilde{E}_{\mathrm{a}}}\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - \underbrace{\Psi^T A_{\mathrm{a}}\Phi}_{\tilde{A}_{\mathrm{a}}}\tilde{\boldsymbol{u}}_{\mathrm{a}} - \underbrace{\Psi^T B_{\mathrm{a},1}}_{\tilde{B}_{\mathrm{a},1}}\boldsymbol{u}_{\mathrm{s}} - \underbrace{\Psi^T B_{\mathrm{a},2}}_{\tilde{B}_{\mathrm{a},2}}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt} = 0.$$

- We obtain the **aerodynamic ROM**,

$$\tilde{E}_{\mathrm{a}}\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} = \tilde{A}_{\mathrm{a}}\boldsymbol{u}_{\mathrm{a}} + \tilde{B}_{\mathrm{a},1}\boldsymbol{u}_{\mathrm{s}} + \tilde{B}_{\mathrm{a},2}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}, \qquad \text{with } \dim(\tilde{E}_{\mathrm{a}}, \tilde{A}_{\mathrm{a}}) \ll \dim(E_{\mathrm{a}}, A_{\mathrm{a}})$$



Projection-Based Model Order Reduction

Here we show an overview of **projection-based model order reduction** for linear systems.

- Approximate the aerodynamics state using a **reduced basis** Φ:

$$\boldsymbol{u}_{\mathrm{a}}(t) \approx \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}}(t).$$

- This leads to the residual for the aerodynamic model,

$$\boldsymbol{R}_{\mathrm{ROM}} = E_{\mathrm{a}}\Phi\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - A_{\mathrm{a}}\Phi\tilde{\boldsymbol{u}}_{\mathrm{a}} - B_{\mathrm{a},1}\boldsymbol{u}_{\mathrm{s}} - B_{\mathrm{a},2}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}.$$

- Force the residual to be orthogonal to the test basis Ψ (Petrov-Galerkin Condition),

$$\Psi^T \boldsymbol{R}_{\mathrm{ROM}} = 0$$

$$\underbrace{\Psi^T E_{\mathrm{a}}\Phi}_{\tilde{E}_{\mathrm{a}}}\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - \underbrace{\Psi^T A_{\mathrm{a}}\Phi}_{\tilde{A}_{\mathrm{a}}}\tilde{\boldsymbol{u}}_{\mathrm{a}} - \underbrace{\Psi^T B_{\mathrm{a},1}}_{\tilde{B}_{\mathrm{a},1}}\boldsymbol{u}_{\mathrm{s}} - \underbrace{\Psi^T B_{\mathrm{a},2}}_{\tilde{B}_{\mathrm{a},2}}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt} = 0.$$

- We obtain the **aerodynamic ROM**,

$$\tilde{E}_{\mathrm{a}}\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} = \tilde{A}_{\mathrm{a}}\boldsymbol{u}_{\mathrm{a}} + \tilde{B}_{\mathrm{a},1}\boldsymbol{u}_{\mathrm{s}} + \tilde{B}_{\mathrm{a},2}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}, \qquad \text{with } \dim(\tilde{E}_{\mathrm{a}}, \tilde{A}_{\mathrm{a}}) \ll \dim(E_{\mathrm{a}}, A_{\mathrm{a}})$$



Projection-Based Model Order Reduction

Here we show an overview of **projection-based model order reduction** for linear systems.

- Approximate the aerodynamics state using a **reduced basis** Φ:

$$\boldsymbol{u}_{\mathrm{a}}(t) \approx \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}}(t).$$

- This leads to the residual for the aerodynamic model,

$$\boldsymbol{R}_{\mathsf{ROM}} = E_{\mathrm{a}}\Phi\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - A_{\mathrm{a}}\Phi\tilde{\boldsymbol{u}}_{\mathrm{a}} - B_{\mathrm{a},1}\boldsymbol{u}_{\mathrm{s}} - B_{\mathrm{a},2}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}.$$

- Force the residual to be orthogonal to the test basis Ψ (Petrov-Galerkin Condition),

$$\Psi^T \boldsymbol{R}_{\mathsf{ROM}} = 0$$

$$\underbrace{\Psi^T E_{\mathrm{a}}\Phi}_{\tilde{E}_{\mathrm{a}}}\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} - \underbrace{\Psi^T A_{\mathrm{a}}\Phi}_{\tilde{A}_{\mathrm{a}}}\tilde{\boldsymbol{u}}_{\mathrm{a}} - \underbrace{\Psi^T B_{\mathrm{a},1}}_{\tilde{B}_{\mathrm{a},1}}\boldsymbol{u}_{\mathrm{s}} - \underbrace{\Psi^T B_{\mathrm{a},2}}_{\tilde{B}_{\mathrm{a},2}}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt} = 0.$$

- We obtain the **aerodynamic ROM**,

$$\tilde{E}_{\mathrm{a}}\frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt} = \tilde{A}_{\mathrm{a}}\boldsymbol{u}_{\mathrm{a}} + \tilde{B}_{\mathrm{a},1}\boldsymbol{u}_{\mathrm{s}} + \tilde{B}_{\mathrm{a},2}\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt}, \qquad \text{with } \dim(\tilde{E}_{\mathrm{a}}, \tilde{A}_{\mathrm{a}}) \ll \dim(E_{\mathrm{a}}, A_{\mathrm{a}})$$



- The **reduced** linearized governing set of equations for aeroelastic analysis are:

Structural Model	$\frac{d^2\boldsymbol{u}_\mathrm{s}}{dt^2} = A_\mathrm{s}\boldsymbol{u}_\mathrm{s} + \overbrace{C_\mathrm{s}\boldsymbol{\Phi}\tilde{\boldsymbol{u}}_\mathrm{a}}^{\tilde{C}_\mathrm{s}}$
Trivial Equation	$\frac{d\boldsymbol{u}_\mathrm{s}}{dt} = \dot{\boldsymbol{u}}_\mathrm{s}$
Aerodynamic ROM	$\tilde{E}_\mathrm{a}\frac{d\tilde{\boldsymbol{u}}_\mathrm{a}}{dt} = \tilde{A}_\mathrm{a}\tilde{\boldsymbol{u}}_\mathrm{a} + \tilde{B}_{\mathrm{a},1}\boldsymbol{u}_\mathrm{s} + \tilde{B}_{\mathrm{a},2}\frac{d\boldsymbol{u}_\mathrm{s}}{dt}$

- Coupling these equations leads to the monolithic **aeroelastic ROM**:

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \tilde{E}_\mathrm{a} \end{bmatrix}}_{\tilde{\mathbb{M}}} \frac{d}{dt} \underbrace{\begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \tilde{\boldsymbol{u}}_\mathrm{a} \end{bmatrix}}_{\tilde{\boldsymbol{u}}} = \underbrace{\begin{bmatrix} 0 & A_\mathrm{s} & \tilde{C}_\mathrm{s} \\ I & 0 & 0 \\ \tilde{B}_{\mathrm{a},2} & \tilde{B}_{\mathrm{a},1} & \tilde{A}_\mathrm{a} \end{bmatrix}}_{\tilde{\mathbb{A}}} \begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \tilde{\boldsymbol{u}}_\mathrm{a} \end{bmatrix}$$

or equivalently

$$\tilde{\mathbb{M}}\frac{d\tilde{\boldsymbol{u}}}{dt} = \tilde{\mathbb{A}}\tilde{\boldsymbol{u}}$$

- Stability is determined by the eigenvalues from the reduced eigenproblem,

$$\tilde{\mathbb{A}}\tilde{\boldsymbol{v}} = \tilde{\lambda}\tilde{\mathbb{M}}\tilde{\boldsymbol{v}} \qquad \text{with } \dim(\tilde{\mathbb{A}}, \tilde{\mathbb{M}}) \ll \dim(\mathbb{A}, \mathbb{M})$$



- The **reduced** linearized governing set of equations for aeroelastic analysis are:

Structural Model $\quad \frac{d^2\boldsymbol{u}_\mathrm{s}}{dt^2} = A_\mathrm{s}\boldsymbol{u}_\mathrm{s} + \overbrace{C_\mathrm{s}\boldsymbol{\Phi}\tilde{\boldsymbol{u}}_\mathrm{a}}^{\tilde{C}_\mathrm{s}}$

Trivial Equation $\quad \frac{d\boldsymbol{u}_\mathrm{s}}{dt} = \dot{\boldsymbol{u}}_\mathrm{s}$

Aerodynamic ROM $\quad \tilde{E}_\mathrm{a}\frac{d\tilde{\boldsymbol{u}}_\mathrm{a}}{dt} = \tilde{A}_\mathrm{a}\tilde{\boldsymbol{u}}_\mathrm{a} + \tilde{B}_{\mathrm{a},1}\boldsymbol{u}_\mathrm{s} + \tilde{B}_{\mathrm{a},2}\frac{d\boldsymbol{u}_\mathrm{s}}{dt}$

- Coupling these equations leads to the monolithic **aeroelastic ROM**:

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \tilde{E}_\mathrm{a} \end{bmatrix}}_{\tilde{\mathbb{M}}} \frac{d}{dt} \underbrace{\begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \tilde{\boldsymbol{u}}_\mathrm{a} \end{bmatrix}}_{\tilde{\boldsymbol{u}}} = \underbrace{\begin{bmatrix} 0 & A_\mathrm{s} & \tilde{C}_\mathrm{s} \\ I & 0 & 0 \\ \tilde{B}_{\mathrm{a},2} & \tilde{B}_{\mathrm{a},1} & \tilde{A}_\mathrm{a} \end{bmatrix}}_{\tilde{\mathbb{A}}} \begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \tilde{\boldsymbol{u}}_\mathrm{a} \end{bmatrix}$$

or equivalently

$$\tilde{\mathbb{M}}\frac{d\tilde{\boldsymbol{u}}}{dt} = \tilde{\mathbb{A}}\tilde{\boldsymbol{u}}$$

- Stability is determined by the eigenvalues from the reduced eigenproblem,

$$\tilde{\mathbb{A}}\tilde{\boldsymbol{v}} = \tilde{\lambda}\tilde{\mathbb{M}}\tilde{\boldsymbol{v}} \qquad \text{with } \dim(\tilde{\mathbb{A}}, \tilde{\mathbb{M}}) \ll \dim(\mathbb{A}, \mathbb{M})$$

- The **reduced** linearized governing set of equations for aeroelastic analysis are:

Structural Model $\quad \frac{d^2\boldsymbol{u}_\mathrm{s}}{dt^2} = A_\mathrm{s}\boldsymbol{u}_\mathrm{s} + \overbrace{C_\mathrm{s}\boldsymbol{\Phi}\tilde{\boldsymbol{u}}_\mathrm{a}}^{\tilde{C}_\mathrm{s}}$

Trivial Equation $\quad \frac{d\boldsymbol{u}_\mathrm{s}}{dt} = \dot{\boldsymbol{u}}_\mathrm{s}$

Aerodynamic ROM $\quad \tilde{E}_\mathrm{a}\frac{d\tilde{\boldsymbol{u}}_\mathrm{a}}{dt} = \tilde{A}_\mathrm{a}\tilde{\boldsymbol{u}}_\mathrm{a} + \tilde{B}_{\mathrm{a},1}\boldsymbol{u}_\mathrm{s} + \tilde{B}_{\mathrm{a},2}\frac{d\boldsymbol{u}_\mathrm{s}}{dt}$

- Coupling these equations leads to the monolithic **aeroelastic ROM**:

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \tilde{E}_\mathrm{a} \end{bmatrix}}_{\tilde{\mathbb{M}}} \frac{d}{dt} \underbrace{\begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \tilde{\boldsymbol{u}}_\mathrm{a} \end{bmatrix}}_{\tilde{\boldsymbol{u}}} = \underbrace{\begin{bmatrix} 0 & A_\mathrm{s} & \tilde{C}_\mathrm{s} \\ I & 0 & 0 \\ \tilde{B}_{\mathrm{a},2} & \tilde{B}_{\mathrm{a},1} & \tilde{A}_\mathrm{a} \end{bmatrix}}_{\tilde{\mathbb{A}}} \begin{bmatrix} \dot{\boldsymbol{u}}_\mathrm{s} \\ \boldsymbol{u}_\mathrm{s} \\ \tilde{\boldsymbol{u}}_\mathrm{a} \end{bmatrix}$$

or equivalently

$$\tilde{\mathbb{M}}\frac{d\tilde{\boldsymbol{u}}}{dt} = \tilde{\mathbb{A}}\tilde{\boldsymbol{u}}$$

- Stability is determined by the eigenvalues from the reduced eigenproblem,

$$\tilde{\mathbb{A}}\tilde{\boldsymbol{v}} = \tilde{\lambda}\tilde{\mathbb{M}}\tilde{\boldsymbol{v}} \qquad \text{with } \dim(\tilde{\mathbb{A}}, \tilde{\mathbb{M}}) \ll \dim(\mathbb{A}, \mathbb{M})$$

Some important questions:

Q1 How do we construct the **reduced basis** Φ for accurate flutter analysis?

From the approximation of the aerodynamic state with the reduced basis,

$$\boldsymbol{u}_{\mathrm{a}}(t) \approx \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}}(t)$$

we see that it is important to have the **aerodynamic response at the flutter point** included in the column-span of Φ.

Q2 Can we estimate the **error in the flutter prediction** from the ROM efficiently without directly using the high-dimensional model?

In this presentation, I will discuss two approaches:

Approach 1: Proper Orthogonal Decomposition (POD) training

Constructs Φ, but does not estimate the error.

Approach 2: Novel error-based training

Constructs Φ and estimates the error.



Some important questions:

Q1 How do we construct the **reduced basis** Φ for accurate flutter analysis?

From the approximation of the aerodynamic state with the reduced basis,

$$\boldsymbol{u}_{\mathrm{a}}(t) \approx \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}}(t)$$

we see that it is important to have the **aerodynamic response at the flutter point** included in the column-span of Φ.

Q2 Can we estimate the **error in the flutter prediction** from the ROM efficiently without directly using the high-dimensional model?

In this presentation, I will discuss two approaches:

Approach 1: Proper Orthogonal Decomposition (POD) training

Constructs Φ, but does not estimate the error.

Approach 2: Novel error-based training

Constructs Φ and estimates the error.



Some important questions:

Q1 How do we construct the **reduced basis** Φ for accurate flutter analysis?

From the approximation of the aerodynamic state with the reduced basis,

$$\boldsymbol{u}_{\mathrm{a}}(t) \approx \Phi \tilde{\boldsymbol{u}}_{\mathrm{a}}(t)$$

we see that it is important to have the **aerodynamic response at the flutter point** included in the column-span of Φ.

Q2 Can we estimate the **error in the flutter prediction** from the ROM efficiently without directly using the high-dimensional model?

In this presentation, I will discuss two approaches:

Approach 1: Proper Orthogonal Decomposition (POD) training

Constructs Φ, but does not estimate the error.

Approach 2: Novel error-based training

Constructs Φ and estimates the error.



Approach 1:
Proper Orthogonal Decomposition



Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) has been widely used for flutter analysis[1]

Overview: Application of POD to construct a **reduced basis** Φ:

- assume we have a set of relevant solution "snapshots" stored in the matrix S,

$$S = \{\boldsymbol{u}_{\mathrm{a}}(t_1), \boldsymbol{u}_{\mathrm{a}}(t_2), ..., \boldsymbol{u}_{\mathrm{a}}(t_n)\}$$

- compute the **singular value decomposition** of the snapshot matrix S,

$$S = U\Sigma Y^T$$

- use the first n_{r} **left singular vectors** in U associated to the **largest singular values** to form the reduced basis,

$$\Phi = \{\phi_1, \phi_2, ..., \phi_{n_{\mathrm{r}}}\}$$



[1] *Hall, K., et al., 2000; Thomas, J., et al., 2003; Beran, P., et al., 2004; Kalashnikova, I., et al. 2013; Amsallem, D.,et al., 2016*

Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) has been widely used for flutter analysis[1]

Overview: Application of POD to construct a **reduced basis** Φ:

- assume we have a set of relevant solution "snapshots" stored in the matrix S,

$$S = \{\boldsymbol{u}_{\mathrm{a}}(t_1), \boldsymbol{u}_{\mathrm{a}}(t_2), ..., \boldsymbol{u}_{\mathrm{a}}(t_n)\}$$

- compute the **singular value decomposition** of the snapshot matrix S,

$$S = U\Sigma Y^T$$

- use the first n_{r} **left singular vectors** in U associated to the **largest singular values** to form the reduced basis,

$$\Phi = \{\phi_1, \phi_2, ..., \phi_{n_{\mathrm{r}}}\}$$



[1] *Hall, K., et al., 2000; Thomas, J., et al., 2003; Beran, P., et al., 2004; Kalashnikova, I., et al. 2013; Amsallem, D.,et al., 2016*

Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) has been widely used for flutter analysis[1]

Overview: Application of POD to construct a **reduced basis** Φ:

- assume we have a set of relevant solution "snapshots" stored in the matrix S,

$$S = \{\boldsymbol{u}_{\mathrm{a}}(t_1), \boldsymbol{u}_{\mathrm{a}}(t_2), ..., \boldsymbol{u}_{\mathrm{a}}(t_n)\}$$

- compute the **singular value decomposition** of the snapshot matrix S,

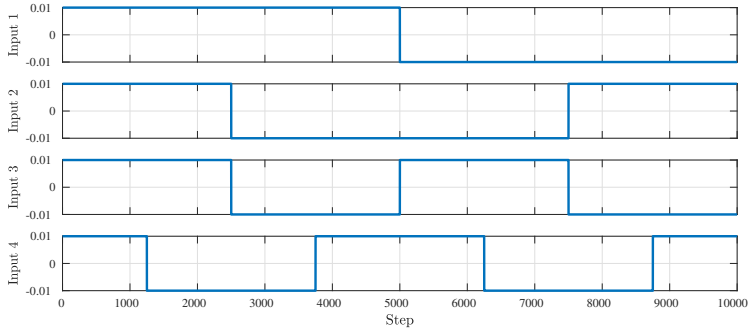
$$S = U\Sigma Y^T$$

- use the first n_{r} **left singular vectors** in U associated to the **largest singular values** to form the reduced basis,

$$\Phi = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_{n_{\mathrm{r}}}\}$$

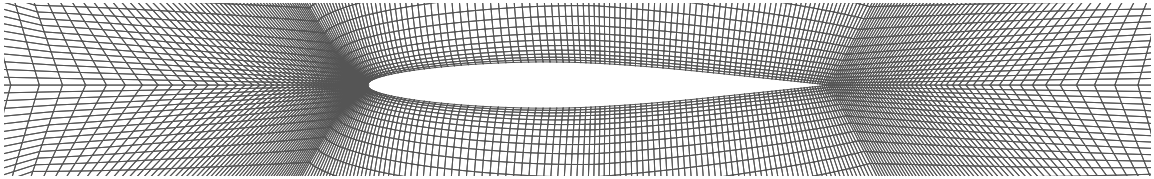


[1] *Hall, K., et al., 2000; Thomas, J., et al., 2003; Beran, P., et al., 2004; Kalashnikova, I., et al. 2013; Amsallem, D.,et al., 2016*

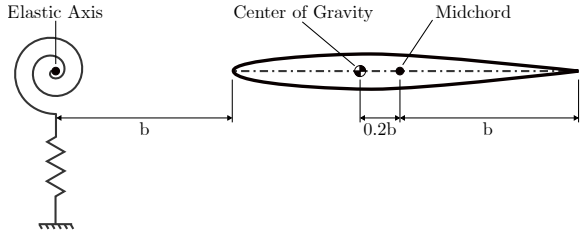


- Snapshots are collected during a **single unsteady flow simulation** during which the structural coordinates and velocities are excited.
- Structural coordinates and velocities are simultaneously excited using the **Walsh functions**[1].

[1] *Silva, W. A., Journal of Aircraft, Vol. 45, No. 4, 2008, pp. 1267–1274.*



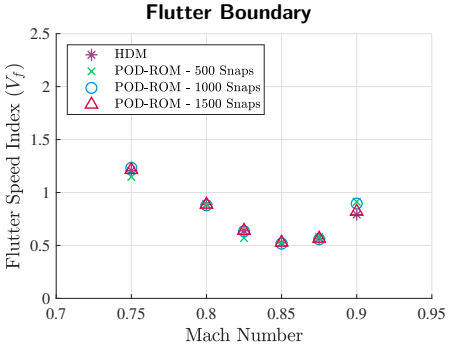
Two degree-of-freedom NACA 64A010 airfoil[1]

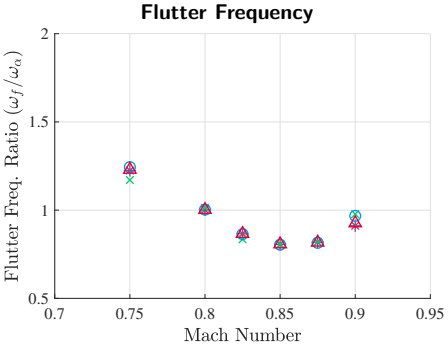


Natural Frequencies: 0.7134 5.3377

[1] *Isogai, Tech. Rep. TR-617T, National Aerospace Laboratory, 1980.*

POD Results





Number of Snapshots[1]	500	1000	1500
Average Time Required [Equ. Num. Steady Euler Flow Solves]	8.4	14	19
Average Relative Error	5.6%	3.4%	0.91%

[1] *All snapshots collected with a time step size of 0.2*

Error Estimation



Error Estimation

We use a **dual-weighted residual (DWR)-based** error estimator[1].

- The aeroelastic ROM gives an **approximate eigenvalue** $\tilde{\lambda}$ and **right eigenvector** $\check{\Phi}\tilde{\boldsymbol{v}}$.
- The error of the approximate eigenvalue can be estimated using the **left eigenvector** $\boldsymbol{v}^{\mathrm{du}}$,

$$\lambda - \tilde{\lambda} \approx \left(\boldsymbol{v}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\check{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\check{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

- Obtaining $\boldsymbol{v}^{\mathrm{du}}$ is computationally expensive. Instead we approximate it using a **reduced basis** $\check{\Phi}^{\mathrm{du}}$,

$$\boldsymbol{v}^{\mathrm{du}} \approx \check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- Obtain the approximate left eigenvector from a **dual ROM**,

$$\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{A}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{A}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}} = \tilde{\lambda}^{\mathrm{du}}\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{M}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{M}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- The error estimator used in this work is,

$$\lambda - \tilde{\lambda} \approx \eta \equiv \left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\check{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\check{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$



[1] *Bangerth W., Rannacher R., "Adaptive Finite Element Methods for Differential Equations", Chapter 7, 2003, p. 81-100.*

Error Estimation

We use a **dual-weighted residual (DWR)-based** error estimator[1].

- The aeroelastic ROM gives an **approximate eigenvalue** $\tilde{\lambda}$ and **right eigenvector** $\check{\Phi}\tilde{\boldsymbol{v}}$.
- The error of the approximate eigenvalue can be estimated using the **left eigenvector** $\boldsymbol{v}^{\mathrm{du}}$,

$$\lambda - \tilde{\lambda} \approx \left(\boldsymbol{v}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\tilde{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\tilde{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

- Obtaining $\boldsymbol{v}^{\mathrm{du}}$ is computationally expensive. Instead we approximate it using a **reduced basis** $\check{\Phi}^{\mathrm{du}}$,

$$\boldsymbol{v}^{\mathrm{du}} \approx \check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- Obtain the approximate left eigenvector from a **dual ROM**,

$$\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{A}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{A}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}} = \tilde{\lambda}^{\mathrm{du}}\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{M}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{M}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- The error estimator used in this work is,

$$\lambda - \tilde{\lambda} \approx \eta \equiv \left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\tilde{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\tilde{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

[1] *Bangerth W., Rannacher R., "Adaptive Finite Element Methods for Differential Equations", Chapter 7, 2003, p. 81-100.*

Error Estimation

We use a **dual-weighted residual (DWR)-based** error estimator[1].

- The aeroelastic ROM gives an **approximate eigenvalue** $\tilde{\lambda}$ and **right eigenvector** $\check{\Phi}\tilde{\boldsymbol{v}}$.
- The error of the approximate eigenvalue can be estimated using the **left eigenvector** $\boldsymbol{v}^{\mathrm{du}}$,

$$\lambda - \tilde{\lambda} \approx \left(\boldsymbol{v}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\check{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\check{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

- Obtaining $\boldsymbol{v}^{\mathrm{du}}$ is computationally expensive. Instead we approximate it using a **reduced basis** $\check{\Phi}^{\mathrm{du}}$,

$$\boldsymbol{v}^{\mathrm{du}} \approx \check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- Obtain the approximate left eigenvector from a **dual ROM**,

$$\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{A}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{A}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}} = \tilde{\lambda}^{\mathrm{du}}\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{M}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{M}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- The error estimator used in this work is,

$$\lambda - \tilde{\lambda} \approx \eta \equiv \left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\check{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\check{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

[1] *Bangerth W., Rannacher R., "Adaptive Finite Element Methods for Differential Equations", Chapter 7, 2003, p. 81-100.*

Error Estimation

We use a **dual-weighted residual (DWR)-based** error estimator[1].

- The aeroelastic ROM gives an **approximate eigenvalue** $\tilde{\lambda}$ and **right eigenvector** $\check{\Phi}\tilde{\boldsymbol{v}}$.
- The error of the approximate eigenvalue can be estimated using the **left eigenvector** $\boldsymbol{v}^{\mathrm{du}}$,

$$\lambda - \tilde{\lambda} \approx \left(\boldsymbol{v}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\check{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\check{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

- Obtaining $\boldsymbol{v}^{\mathrm{du}}$ is computationally expensive. Instead we approximate it using a **reduced basis** $\check{\Phi}^{\mathrm{du}}$,

$$\boldsymbol{v}^{\mathrm{du}} \approx \check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- Obtain the approximate left eigenvector from a **dual ROM**,

$$\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{A}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{A}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}} = \tilde{\lambda}^{\mathrm{du}}\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{M}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{M}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- The error estimator used in this work is,

$$\lambda - \tilde{\lambda} \approx \eta \equiv \left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\check{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\check{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

[1] *Bangerth W., Rannacher R., "Adaptive Finite Element Methods for Differential Equations", Chapter 7, 2003, p. 81-100.*

Error Estimation

We use a **dual-weighted residual (DWR)-based** error estimator[1].

- The aeroelastic ROM gives an **approximate eigenvalue** $\tilde{\lambda}$ and **right eigenvector** $\check{\Phi}\tilde{\boldsymbol{v}}$.
- The error of the approximate eigenvalue can be estimated using the **left eigenvector** $\boldsymbol{v}^{\mathrm{du}}$,

$$\lambda - \tilde{\lambda} \approx \left(\boldsymbol{v}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\tilde{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\tilde{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$

- Obtaining $\boldsymbol{v}^{\mathrm{du}}$ is computationally expensive. Instead we approximate it using a **reduced basis** $\check{\Phi}^{\mathrm{du}}$,

$$\boldsymbol{v}^{\mathrm{du}} \approx \check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- Obtain the approximate left eigenvector from a **dual ROM**,

$$\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{A}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{A}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}} = \tilde{\lambda}^{\mathrm{du}}\underbrace{\left(\check{\Psi}^{\mathrm{du}}\right)^{T}\mathbb{M}^{T}\check{\Phi}^{\mathrm{du}}}_{\tilde{\mathbb{M}}^{\mathrm{du}}}\tilde{\boldsymbol{v}}^{\mathrm{du}}$$

- The error estimator used in this work is,

$$\lambda - \tilde{\lambda} \approx \eta \equiv \left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^{H} \underbrace{\left(\mathbb{A}\tilde{\Phi}\tilde{\boldsymbol{v}} - \tilde{\lambda}\mathbb{M}\tilde{\Phi}\tilde{\boldsymbol{v}}\right)}_{\text{Eigenproblem Residual}}$$



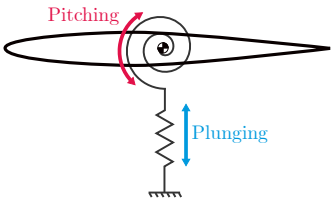
[1] *Bangerth W., Rannacher R., "Adaptive Finite Element Methods for Differential Equations", Chapter 7, 2003, p. 81-100.*

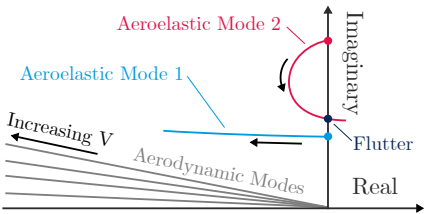
Approach 2: Error-Based Training



Idea: Construct the reduced basis Φ using eigenvectors associated to the aeroelastic modes.

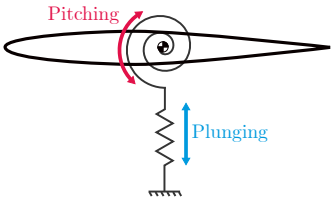
- Recall the aeroelastic modes originating from the wind-off structural modes are important for flutter analysis.
- What if we iteratively construct the reduced basis Φ using approximate eigenvectors associated to these aeroelastic modes?
- As we increase the speed index V, update the reduced basis Φ with new aeroelastic eigenvectors.
- Additionally, we can use the eigenvalue error estimator η to tell us when we need to update Φ to obtain a better eigenvalue approximation.

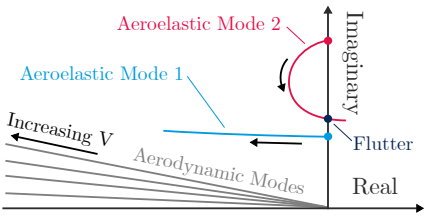




Idea: Construct the reduced basis Φ using eigenvectors associated to the aeroelastic modes.

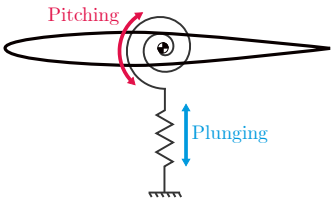
- Recall the aeroelastic modes originating from the wind-off structural modes are important for flutter analysis.
- What if we iteratively construct the reduced basis Φ using approximate eigenvectors associated to these aeroelastic modes?
- As we increase the speed index V, update the reduced basis Φ with new aeroelastic eigenvectors.
- Additionally, we can use the eigenvalue error estimator η to tell us when we need to update Φ to obtain a better eigenvalue approximation.

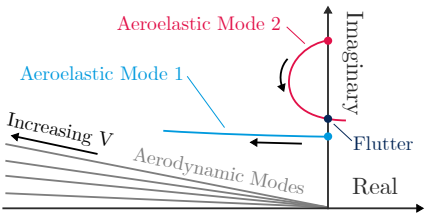




Idea: Construct the reduced basis Φ using eigenvectors associated to the aeroelastic modes.

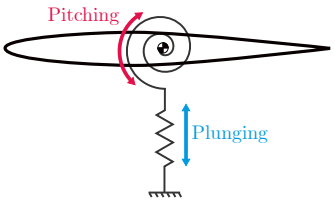
- Recall the aeroelastic modes originating from the wind-off structural modes are important for flutter analysis.
- What if we iteratively construct the reduced basis Φ using approximate eigenvectors associated to these aeroelastic modes?
- As we increase the speed index V, update the reduced basis Φ with new aeroelastic eigenvectors.
- Additionally, we can use the eigenvalue error estimator η to tell us when we need to update Φ to obtain a better eigenvalue approximation.

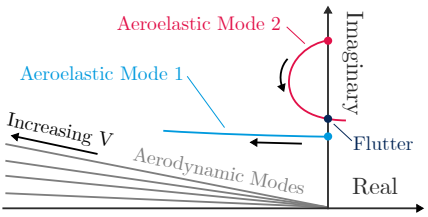




Idea: Construct the reduced basis Φ using eigenvectors associated to the aeroelastic modes.

- Recall the aeroelastic modes originating from the wind-off structural modes are important for flutter analysis.
- What if we iteratively construct the reduced basis Φ using approximate eigenvectors associated to these aeroelastic modes?
- As we increase the speed index V, update the reduced basis Φ with new aeroelastic eigenvectors.
- Additionally, we can use the eigenvalue error estimator η to tell us when we need to update Φ to obtain a better eigenvalue approximation.





STAGE 1

ROM Initialization

STAGE 2

Eigenvalue Tracking & ROM Updates

STAGE 3

Flutter Point Calculation



STAGE 1

ROM Initialization

STAGE 2

Eigenvalue Tracking & ROM Updates

STAGE 3

Flutter Point Calculation



Goal: Construct reduced bases Φ and Φ^{du} for the initial primal and dual aeroelastic ROMs.

- Initialize the ROM using snapshots from the high-dimensional linearized Euler equations in the frequency domain,

  Primal Snapshots $$(i\omega E_{\text{a}} - A_{\text{a}})\,\bar{\boldsymbol{u}}_{\text{a}} = (B_{\text{a},1} + i\omega B_{\text{a},2})\bar{\boldsymbol{u}}_{\text{s}}$$

  Dual Snapshots $$\left(i\omega E_{\text{a}} - A_{\text{a}}^T\right)\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}} = i\omega C_{\text{a}}^T\,\bar{\boldsymbol{u}}_{\text{s}}$$

  $\omega \leftarrow$ Structural Natural Frequency $\quad \bar{\boldsymbol{u}}_{\text{s}} \leftarrow$ Structural Mode Shapes

- Solve this at low values of speed index V_i to obtain set of initial snapshots,

$$S = \{\bar{\boldsymbol{u}}_{\text{a}}(V_1,\omega_1),...,\bar{\boldsymbol{u}}_{\text{a}}(V_i,\omega_j),...\},$$
$$S^{\text{du}} = \{\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}}(V_1,\omega_1),...,\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}}(V_i,\omega_j),...\}.$$

- Modified Gram-Schmidt (MGS) process to orthogonalize snapshot vectors and create the initial reduced bases,

  Primal Reduced Basis $$\Phi = \text{MGS}\{S\},$$

  Dual Reduced Basis $$\Phi^{\text{du}} = \text{MGS}\{S^{\text{du}}\}.$$



Goal: Construct reduced bases Φ and Φ^{du} for the initial primal and dual aeroelastic ROMs.

- Initialize the ROM using snapshots from the high-dimensional linearized Euler equations in the frequency domain,

  Primal Snapshots $$(i\omega E_{\text{a}} - A_{\text{a}})\,\bar{\boldsymbol{u}}_{\text{a}} = (B_{\text{a},1} + i\omega B_{\text{a},2})\bar{\boldsymbol{u}}_{\text{s}}$$

  Dual Snapshots $$\left(i\omega E_{\text{a}} - A_{\text{a}}^{T}\right)\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}} = i\omega C_{\text{a}}^{T}\,\bar{\boldsymbol{u}}_{\text{s}}$$

  $\omega \leftarrow$ Structural Natural Frequency $\qquad \bar{\boldsymbol{u}}_{\text{s}} \leftarrow$ Structural Mode Shapes

- Solve this at low values of speed index V_i to obtain set of initial snapshots,

$$S = \{\bar{\boldsymbol{u}}_{\text{a}}(V_1,\omega_1),...,\bar{\boldsymbol{u}}_{\text{a}}(V_i,\omega_j),...\},$$
$$S^{\text{du}} = \{\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}}(V_1,\omega_1),...,\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}}(V_i,\omega_j),...\}.$$

- Modified Gram-Schmidt (MGS) process to orthogonalize snapshot vectors and create the initial reduced bases,

  Primal Reduced Basis $$\Phi = \text{MGS}\{S\},$$

  Dual Reduced Basis $$\Phi^{\text{du}} = \text{MGS}\{S^{\text{du}}\}.$$



Goal: Construct reduced bases Φ and Φ^{du} for the initial primal and dual aeroelastic ROMs.

- Initialize the ROM using snapshots from the high-dimensional linearized Euler equations in the frequency domain,

  Primal Snapshots $$(i\omega E_{\text{a}} - A_{\text{a}})\,\bar{\boldsymbol{u}}_{\text{a}} = (B_{\text{a},1} + i\omega B_{\text{a},2})\bar{\boldsymbol{u}}_{\text{s}}$$

  Dual Snapshots $$\left(i\omega E_{\text{a}} - A_{\text{a}}^{T}\right)\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}} = i\omega C_{\text{a}}^{T}\,\bar{\boldsymbol{u}}_{\text{s}}$$

  $\omega \leftarrow$ Structural Natural Frequency $\qquad \bar{\boldsymbol{u}}_{\text{s}} \leftarrow$ Structural Mode Shapes

- Solve this at low values of speed index V_i to obtain set of initial snapshots,

$$S = \{\bar{\boldsymbol{u}}_{\text{a}}(V_1, \omega_1), ..., \bar{\boldsymbol{u}}_{\text{a}}(V_i, \omega_j), ...\},$$
$$S^{\text{du}} = \{\bar{\boldsymbol{u}}_{\text{a}}^{\text{du}}(V_1, \omega_1), ..., \bar{\boldsymbol{u}}_{\text{a}}^{\text{du}}(V_i, \omega_j), ...\}.$$

- Modified Gram-Schmidt (MGS) process to orthogonalize snapshot vectors and create the initial reduced bases,

  Primal Reduced Basis $$\Phi = \text{MGS}\{S\},$$

  Dual Reduced Basis $$\Phi^{\text{du}} = \text{MGS}\{S^{\text{du}}\}.$$



STAGE 1

ROM Initialization

STAGE 2

Eigenvalue Tracking & ROM Updates

STAGE 3

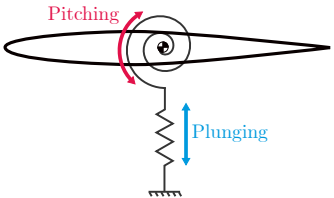
Flutter Point Calculation

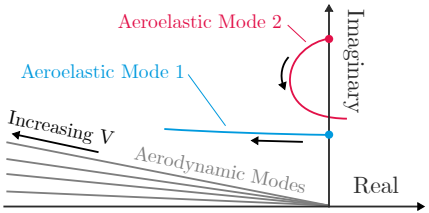


Eigenvalue Tracking and ROM Updates

Goal: Track aeroelastic eigenvalues for increasing speed index values, and train the ROM to reduce eigenvalue errors.

- **Eigenvalue Tracking**: Identify the aeroelastic eigenvalues from the ROM for increasing speed index values.
  - Assume we have the known aeroelastic eigenvalues ($\bar{\lambda}_j^{(n-1)}$ for $j = [1, \text{Num. Strct. Modes}]$).
  - Track these eigenvalues between speed index increments.
  - Tracking metrics: error estimation, magnitude difference, eigenvector correlation[1].
- **ROM Updates**: Update the ROM to obtain accurate eigenvalue approximations.
  - approximate the error in the eigenvalue using η,
  - if $\eta >$ user-set tolerance, update the ROM.

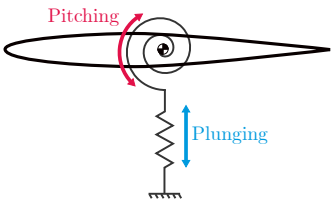


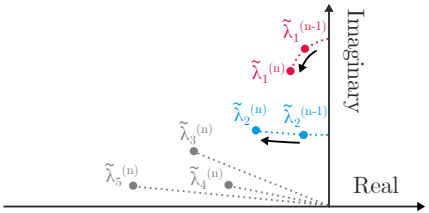


[1] *Lowe, B., and Zingg, D., AIAA Journal, Accepted.*

Goal: Track aeroelastic eigenvalues for increasing speed index values, and train the ROM to reduce eigenvalue errors.

- **Eigenvalue Tracking**: Identify the aeroelastic eigenvalues from the ROM for increasing speed index values.
  - Assume we have the known aeroelastic eigenvalues ($\tilde{\lambda}_j^{(n-1)}$ for $j = [1, \text{Num. Strct. Modes}]$).
  - Track these eigenvalues between speed index increments.
  - Tracking metrics: error estimation, magnitude difference, eigenvector correlation[1].
- **ROM Updates**: Update the ROM to obtain accurate eigenvalue approximations.
  - approximate the error in the eigenvalue using η,
  - if $\eta >$ user-set tolerance, update the ROM.

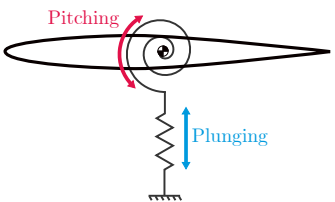


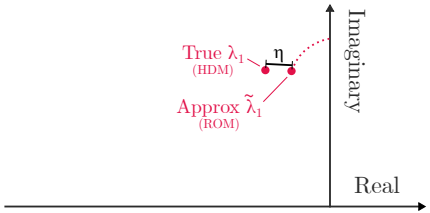


[1] *Lowe, B., and Zingg, D., AIAA Journal, Accepted.*

Goal: Track aeroelastic eigenvalues for increasing speed index values, and train the ROM to reduce eigenvalue errors.

- **Eigenvalue Tracking**: Identify the aeroelastic eigenvalues from the ROM for increasing speed index values.
  - Assume we have the known aeroelastic eigenvalues ($\bar{\lambda}_j^{(n-1)}$ for $j = [1, \text{Num. Strct. Modes}]$).
  - Track these eigenvalues between speed index increments.
  - Tracking metrics: error estimation, magnitude difference, eigenvector correlation[1].
- **ROM Updates**: Update the ROM to obtain accurate eigenvalue approximations.
  - approximate the error in the eigenvalue using η,
  - if $\eta >$ user-set tolerance, update the ROM.





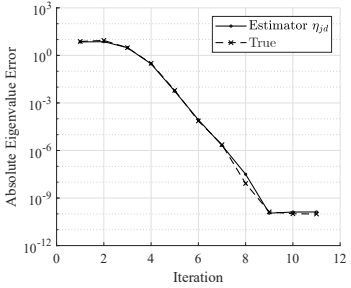
[1] *Lowe, B., and Zingg, D., AIAA Journal, Accepted.*

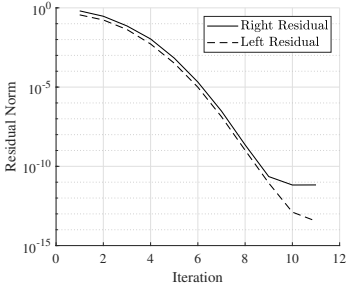
The **reduced bases** are updated using **approximate truth eigenvectors** from the high-dimensional eigenproblem:

$$\Phi \leftarrow \text{Right Eigenvectors}, \qquad \Phi^{\text{du}} \leftarrow \text{Left Eigenvectors}$$

Right and left truth eigenvector approximations are obtained using the **two-sided refined Jacobi-Davidson**[1] algorithm:

- converges to the eigenvalue nearest a specified target (use $\tilde{\lambda}$),
- simultaneously provides approximate left and right eigenvectors,
- new **dual-weighted residual-based error** stopping criterion η_{jd}.





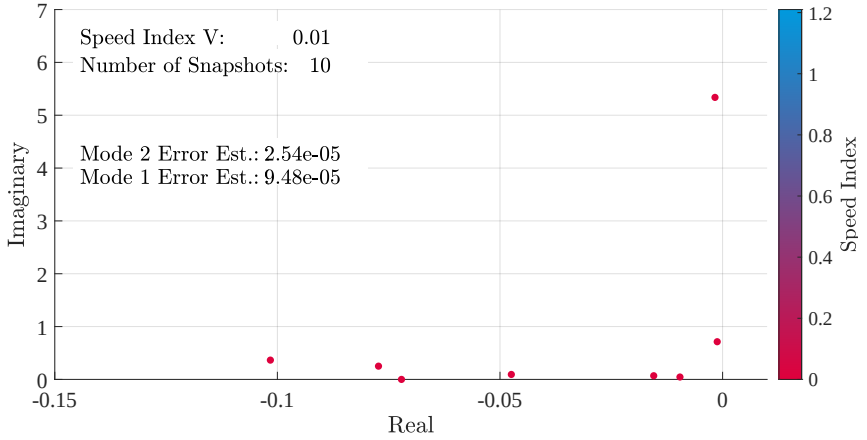
[1] *Hochstenbach, M. E. and Sleijpen, G. L., Linear Algebra and its Applications, Vol. 358, No. 1, 2003, pp. 145–172.*

Example Run

Example eigenvalue tracking and ROM updates:

Isogai NACA 64A010 test case (two-DOF airfoil) at Mach 0.75 and 0° angle-of-attack

Relative eigenvalue error tolerance $\eta/\tilde{\lambda} < 0.01$

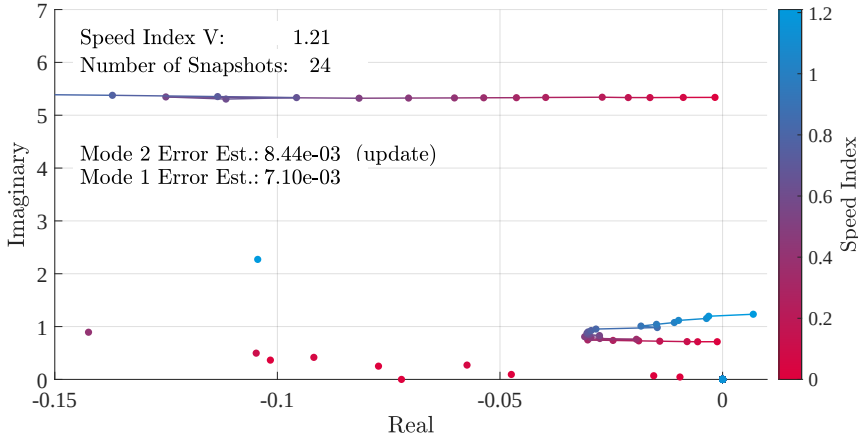


Example Run

Example eigenvalue tracking and ROM updates:

Isogai NACA 64A010 test case (two-DOF airfoil) at Mach 0.75 and 0° angle-of-attack

Relative eigenvalue error tolerance $\eta/\tilde{\lambda} < 0.01$



STAGE 1

ROM Initialization

STAGE 2

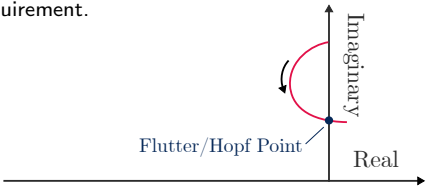
Eigenvalue Tracking & ROM Updates

STAGE 3

Flutter Point Calculation



Goal: Predict the flutter point of the HDM within the user-set accuracy requirement.



- We seek the **flutter speed index** $\tilde{V}_{\mathrm{f}}$ where $\tilde{\lambda} = i\tilde{\omega}_{\mathrm{f}}$ for the **aeroelastic ROM**.
- The **flutter point** of the aeroelastic ROM is determined using a direct Hopf point calculation method[1].
- To estimate the difference between the **ROM flutter point** $\tilde{V}_{\mathrm{f}}$ and the **HDM flutter point** V_{f}, we have developed the **error estimator**,

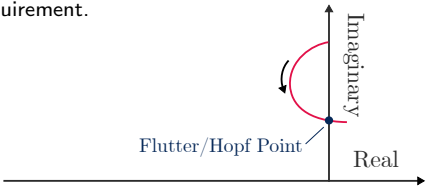
$$\left|V_{\mathrm{f}} - \tilde{V}_{\mathrm{f}}\right| \approx \eta_{\mathrm{f}} \equiv \overbrace{\left|\mathrm{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{v}^{\mathrm{du}}\right)^{H}\left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{v}\right\}\right|^{-1}}^{\approx\left|\frac{dV}{d\mathrm{Real}\{\lambda\}}\right|} \overbrace{\approx\frac{|\mathrm{Real}\{\lambda(\tilde{V}_{\mathrm{f}})\}|}{|\mathrm{Real}\{\eta\}|}}^{} .$$

- Iterate to converge; if $\eta_{\mathrm{f}} >$ user-set tolerance:
  1. apply the Jacobi-Davidson algorithm to approximate the left and right eigenvectors at $\tilde{V}_{\mathrm{f}}$,
  2. update the primal and dual ROMs,
  3. repeat Hopf point calculation and error estimation.

[1] *Griewank and Reddien, Journal of Numerical Analysis, Vol. 3, No. 3, 1983, p. 295-303.*

Goal: Predict the flutter point of the HDM within the user-set accuracy requirement.

- We seek the **flutter speed index** $\tilde{V}_\mathrm{f}$ where $\tilde{\lambda} = i\tilde{\omega}_\mathrm{f}$ for the **aeroelastic ROM**.
- The **flutter point** of the aeroelastic ROM is determined using a direct Hopf point calculation method[1].



- To estimate the difference between the **ROM flutter point** $\tilde{V}_\mathrm{f}$ and the **HDM flutter point** V_f, we have developed the **error estimator**,

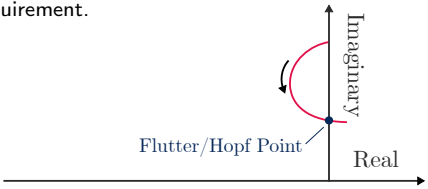
$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \eta_\mathrm{f} \equiv \overbrace{\left|\mathrm{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{v}^{\mathrm{du}}\right)^H\left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{v}\right\}\right|^{-1}}^{\approx\left|\frac{dV}{d\mathrm{Real}\{\lambda\}}\right|} \overbrace{\frac{|\mathrm{Real}\{\lambda(\tilde{V}_\mathrm{f})\}|}{|\mathrm{Real}\{\eta\}|}}^{\approx|\mathrm{Real}\{\lambda(\tilde{V}_\mathrm{f})\}|} .$$

- Iterate to converge; if $\eta_\mathrm{f} >$ user-set tolerance:
  1. apply the Jacobi-Davidson algorithm to approximate the left and right eigenvectors at $\tilde{V}_\mathrm{f}$,
  2. update the primal and dual ROMs,
  3. repeat Hopf point calculation and error estimation.

[1] *Griewank and Reddien, Journal of Numerical Analysis, Vol. 3, No. 3, 1983, p. 295-303.*

Goal: Predict the flutter point of the HDM within the user-set accuracy requirement.

- We seek the **flutter speed index** $\tilde{V}_\mathrm{f}$ where $\tilde{\lambda} = i\tilde{\omega}_\mathrm{f}$ for the **aeroelastic ROM**.
- The **flutter point** of the aeroelastic ROM is determined using a direct Hopf point calculation method[1].



- To estimate the difference between the **ROM flutter point** $\tilde{V}_\mathrm{f}$ and the **HDM flutter point** V_f, we have developed the **error estimator**,

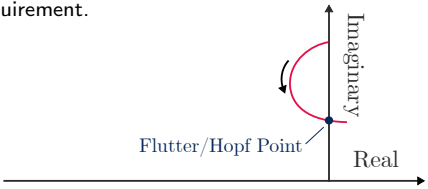
$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \eta_\mathrm{f} \equiv \overbrace{\left|\mathsf{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^H\left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{\boldsymbol{v}}\right\}\right|^{-1}}^{\approx\left|\frac{dV}{d\mathsf{Real}\{\lambda\}}\right|} \overbrace{\left|\mathsf{Real}\{\lambda(\tilde{V}_\mathrm{f})\}\right|}^{\approx|\mathsf{Real}\{\lambda(\tilde{V}_\mathrm{f})\}|} \Big/ \left|\mathsf{Real}\{\eta\}\right| \quad .$$

- Iterate to converge; if $\eta_\mathrm{f} >$ user-set tolerance:
  1. apply the Jacobi-Davidson algorithm to approximate the left and right eigenvectors at $\tilde{V}_\mathrm{f}$,
  2. update the primal and dual ROMs,
  3. repeat Hopf point calculation and error estimation.

[1] *Griewank and Reddien, Journal of Numerical Analysis, Vol. 3, No. 3, 1983, p. 295-303.*

Goal: Predict the flutter point of the HDM within the user-set accuracy requirement.

- We seek the **flutter speed index** $\tilde{V}_\mathrm{f}$ where $\tilde{\lambda} = i\tilde{\omega}_\mathrm{f}$ for the **aeroelastic ROM**.
- The **flutter point** of the aeroelastic ROM is determined using a direct Hopf point calculation method[1].



- To estimate the difference between the **ROM flutter point** $\tilde{V}_\mathrm{f}$ and the **HDM flutter point** V_f, we have developed the **error estimator**,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \eta_\mathrm{f} \equiv \overbrace{\left|\mathrm{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^H\left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{\boldsymbol{v}}\right\}\right|^{-1}}^{\approx\left|\frac{dV}{d\mathrm{Real}\{\lambda\}}\right|} \overbrace{\frac{|\mathrm{Real}\{\lambda(\tilde{V}_\mathrm{f})\}|}{|\mathrm{Real}\{\eta\}|}}^{\approx} .$$

- Iterate to converge; if $\eta_\mathrm{f} >$ user-set tolerance:
  1. apply the Jacobi-Davidson algorithm to approximate the left and right eigenvectors at $\tilde{V}_\mathrm{f}$,
  2. update the primal and dual ROMs,
  3. repeat Hopf point calculation and error estimation.

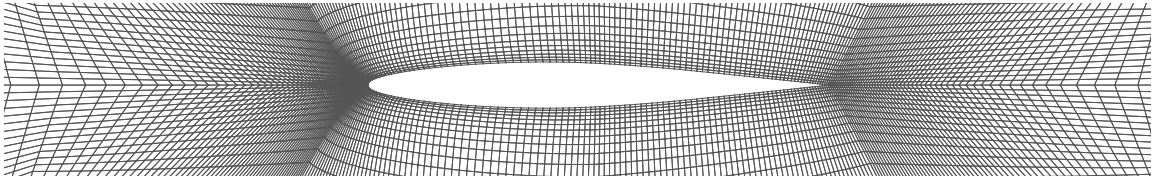
[1] *Griewank and Reddien, Journal of Numerical Analysis, Vol. 3, No. 3, 1983, p. 295-303.*

Results

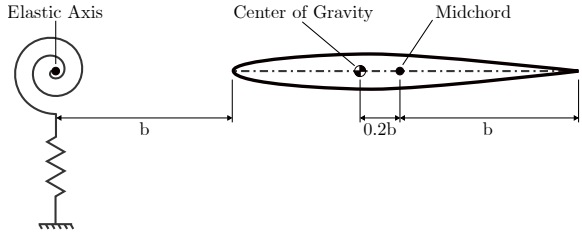
Approach 1: POD-Based ROM

Approach 2: Novel Error-Based ROM





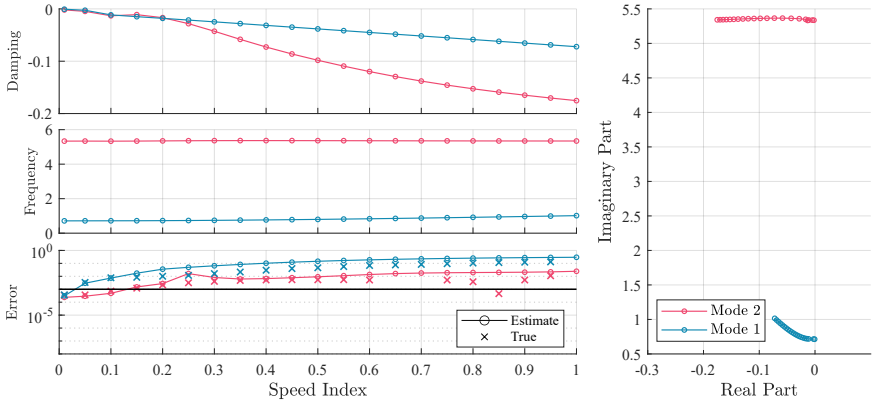
Two degree-of-freedom NACA 64A010 airfoil[1]



Natural Frequencies: 0.7134 5.3377

[1] *Isogai, Tech. Rep. TR-617T, National Aerospace Laboratory, 1980.*

Error-Based ROM: root locus convergence at Mach 0.85 and 0° angle-of-attack.



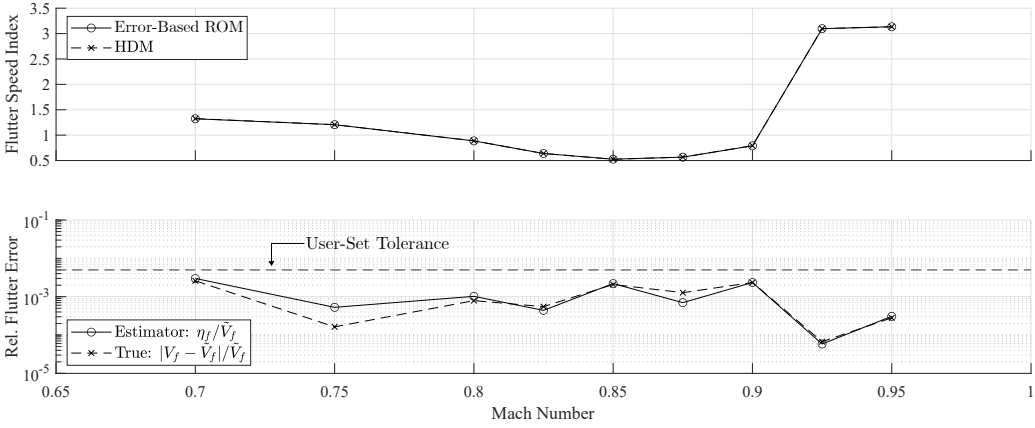
Number of Snapshots: 10

Error-Based ROM: Comparison between the flutter point error estimator η_{f} and the true flutter point error,

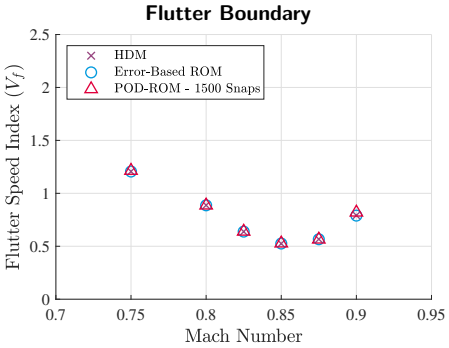
$$\eta_{\text{f}} \approx \left| V_{\text{f}} - \tilde{V}_{\text{f}} \right|$$

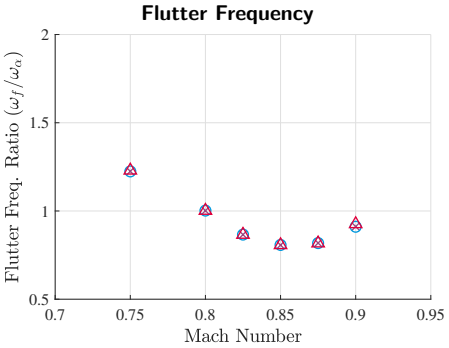
V_{f} = HDM Flutter Speed Index

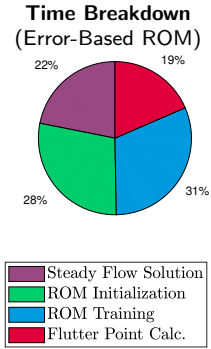
$\tilde{V}_{\text{f}}$ = ROM Flutter Speed Index



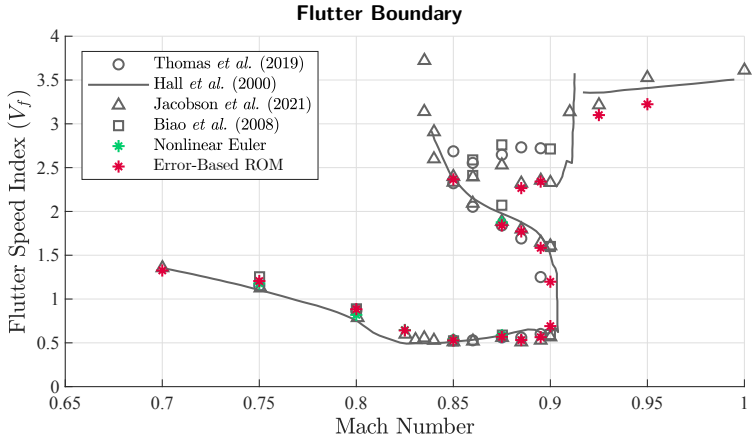
POD Results



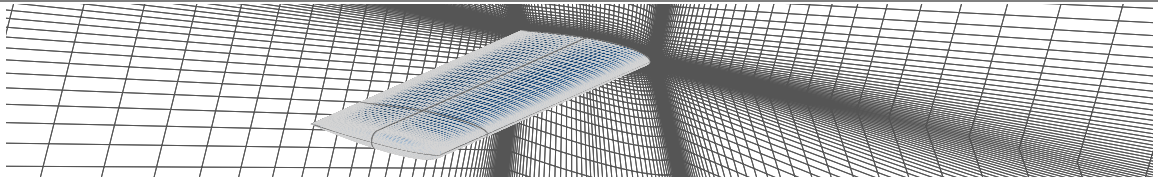




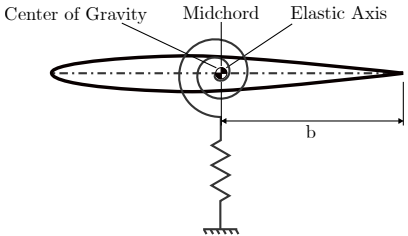
	POD-Based ROM			Error-Based ROM
	500 Snapshots	1000 Snapshots	1500 Snapshots	
Average Time Required [Equ. Num. Steady Euler Flow Solves]	8.4	14	19	4.5
Average Relative Error	5.6%	3.4%	0.91%	0.44%



Time for Single Flutter Point: 4.5 Nonlinear Steady Euler Flow Solutions



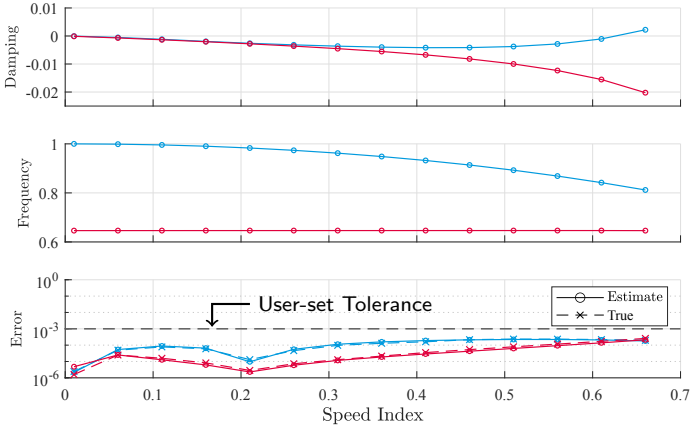
Two degree-of-freedom NACA 0012 rectangular wing[1]

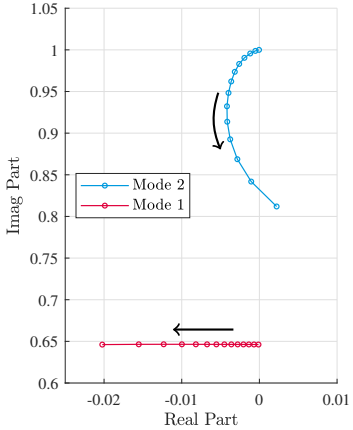


Natural Frequencies: $3.36\,\text{Hz}$ $5.20\,\text{Hz}$;

[1] *Rivera, J.R., et al., 33rd Structures, Structural Dynamics and Materials Conference, Texas, 1992. AIAA 1992-2396*

Error-Based ROM: root locus at Mach 0.67 and 0° angle-of-attack:



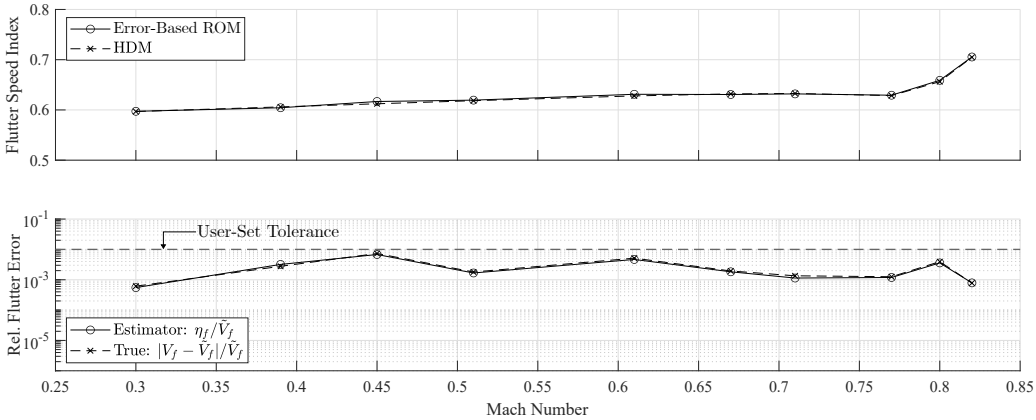


Error-Based ROM: Comparison between the flutter point error estimator η_{f} and the true flutter point error,

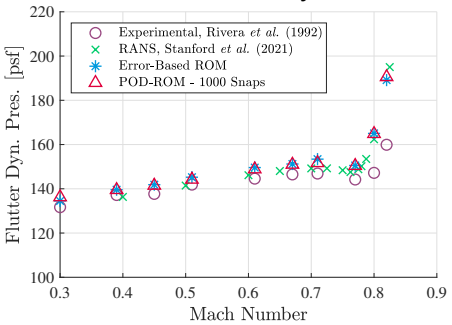
$$\eta_{\mathrm{f}} \approx \left| V_{\mathrm{f}} - \tilde{V}_{\mathrm{f}} \right|$$

V_{f} = HDM Flutter Speed Index

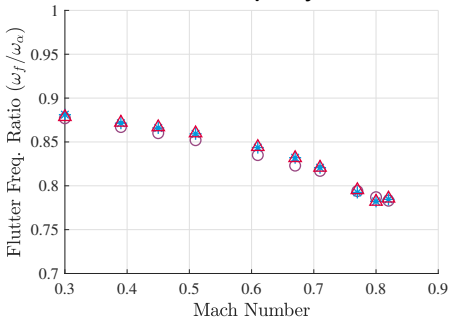
$\tilde{V}_{\mathrm{f}}$ = ROM Flutter Speed Index



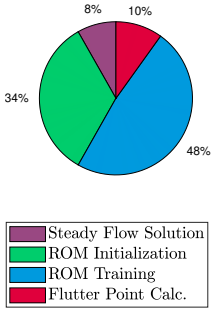
Flutter Boundary



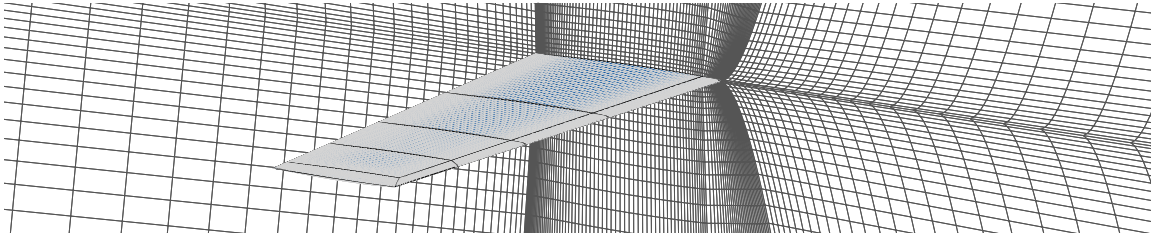
Flutter Frequency



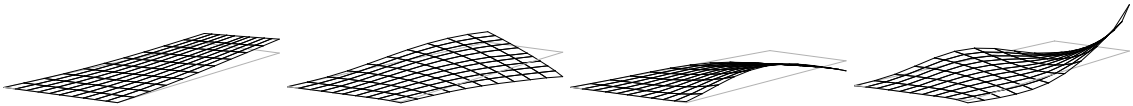
Time Breakdown (Error-Based ROM)



	POD-Based ROM			Error-Based ROM
	250 Snapshots	500 Snapshots	1000 Snapshots	
Average Time Required [Equ. Num. Steady Euler Flow Solves]	13	20	27	12
Average Error Relative to HDM	0.68%	0.57%	0.27%	0.87%



AGARD 445.6 wing weakened model[1]



Mode 1
9.5992 Hz

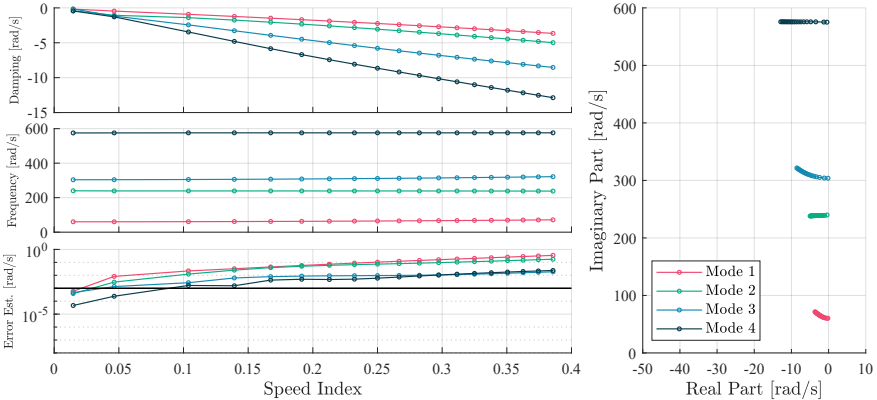
Mode 2
38.1650 Hz

Mode 3
48.3482 Hz

Mode 4
91.5448 Hz

[1] *Yates, E. C., Tech. Rep. N88-11202, NASA, 1987.*

Error-Based ROM: root locus convergence at Mach 0.901 and 0° angle-of-attack.



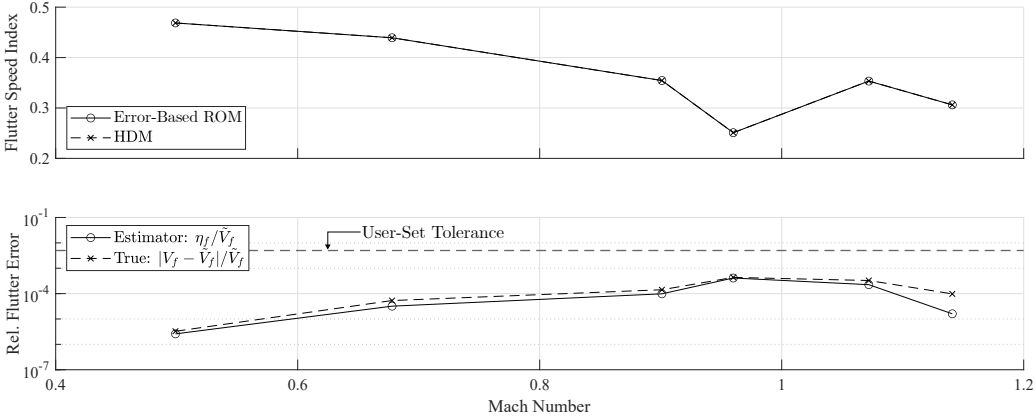
Number of Snapshots: 21

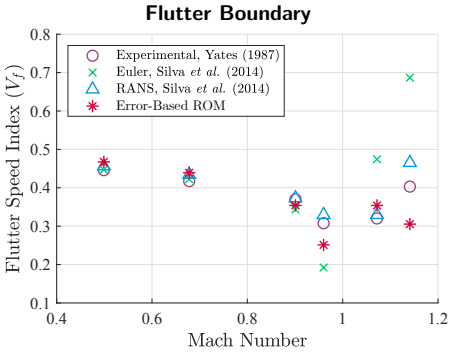
Error-Based ROM: Comparison between the flutter point error estimator η_{f} and the true flutter point error,

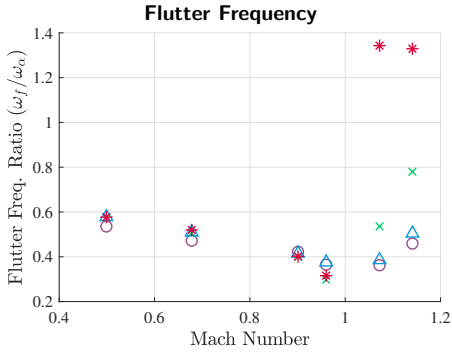
$$\eta_{\mathrm{f}} \approx \left| V_{\mathrm{f}} - \tilde{V}_{\mathrm{f}} \right|$$

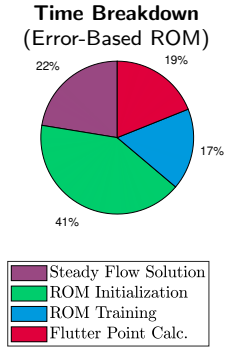
V_{f} = HDM Flutter Speed Index

$\tilde{V}_{\mathrm{f}}$ = ROM Flutter Speed Index

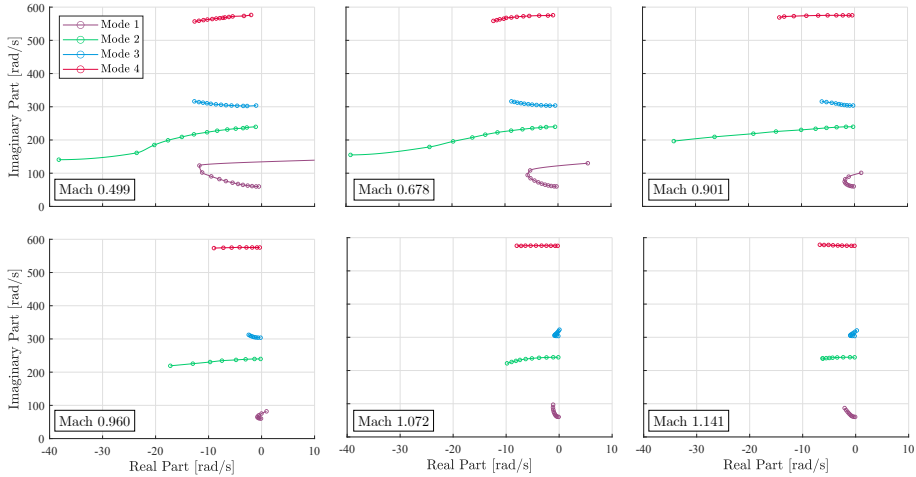








	POD-Based ROM 3000 Snapshots	Error-Based ROM
Average Time Required [Equ. Num. Steady Euler Flow Solves]	30	4.5
Average Error Relative to HDM	1.55%	0.13%


Mode 1
Mode 2
Mode 3
Mode 4
Imaginary Part [rad/s]
Real Part [rad/s]
Mach 0.499
Mach 0.678
Mach 0.901
Mach 0.960
Mach 1.072
Mach 1.141

Conclusion



Conclusion

We have presented a new method for ROM-based flutter analysis with error estimation.

- **Rapid**: The new ROM-based approach with error estimation is capable of efficiently predicting flutter. Flutter can be predicted in a computational time equivalent to 4.5 to 12 steady nonlinear inviscid flow evaluations for the test cases presented.
- **Reliable**: The eigenvalue and flutter point error estimators show good comparison with the true error.
- **Automated**: Flutter can be reliably predicted with selection of only the speed index step size, the eigenvalue error tolerance, and the flutter point error tolerance[1].

Future work:

- including static aeroelastic deflections,
- including viscous terms in the aerodynamic model,
- constraint for aerodynamic shape optimization.

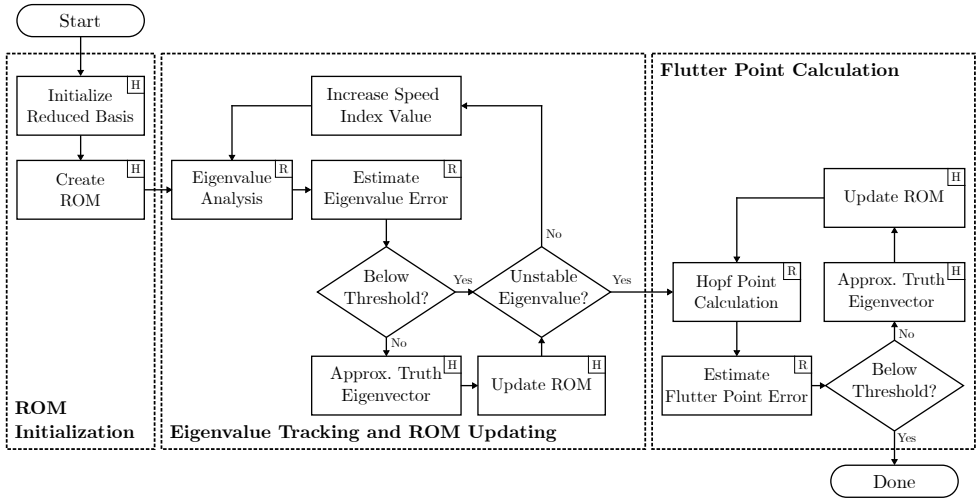
Financial support provided by:

BOMBARDIER NSERC CRSNG Ontario UNIVERSITY OF TORONTO

[1] *Initial snapshot parameters were fixed for all test cases presented.*

Extra Slides




Start
Initialize Reduced Basis
H
Create ROM
H
ROM Initialization
Increase Speed Index Value
Eigenvalue Analysis
R
Estimate Eigenvalue Error
R
Below Threshold?
Yes
No
Unstable Eigenvalue?
No
Yes
Approx. Truth Eigenvector
H
Update ROM
H
Eigenvalue Tracking and ROM Updating
Flutter Point Calculation
Update ROM
H
Hopf Point Calculation
R
Approx. Truth Eigenvector
H
No
Estimate Flutter Point Error
R
Below Threshold?
Yes
Done

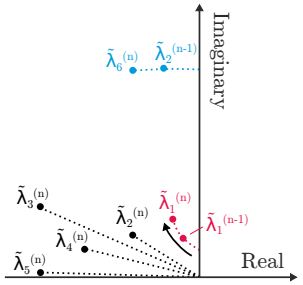
Eigenvalue Tracking

- At speed index $V^{(n-1)}$ we have the known aeroelastic eigenvalues $[\tilde{\lambda}_1^{(n-1)}, ..., \tilde{\lambda}_j^{(n-1)}, ...]$
- At speed index $V^{(n)}$ we have the eigenvalues $[\tilde{\lambda}_1^{(n)}, ..., \tilde{\lambda}_i^{(n)}, ...]$
- Calculate Tracking Metrics for all eigenpairs $(\tilde{\lambda}_i^{(n-1)}, \tilde{\lambda}_j^{(n)})$,

  $\sigma_{i,j}^{\text{err}}$ $\leftarrow$ Error-based tracking using left eigenvalue at $V^{(n-1)}$

  $\sigma_{i,j}^{\text{diff}}$ $\leftarrow$ Magnitude Difference $|\tilde{\lambda}^{(n-1)} - \tilde{\lambda}^{(n)}|$

  $\sigma_{i,j}^{\text{corr}}$ $\leftarrow$ Correlation metric of van Zyl



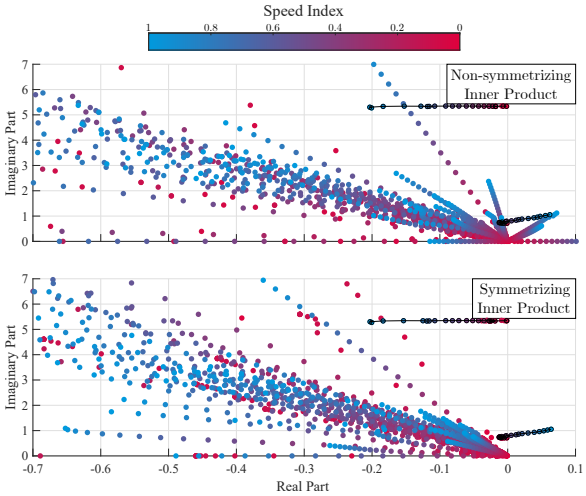
- Aggregate into an opinion pool,

$$p_{i,j} = (\sigma_{i,j}^{\text{err}})^{w_1} (\sigma_{i,j}^{\text{diff}})^{w_2} (\sigma_{i,j}^{\text{corr}})^{w_3}$$

- Projection-based ROMs can sometimes be **unstable**.
- In other word, the aerodynamic matrix pencil $(\tilde{A}_\mathrm{a}, \tilde{E}_\mathrm{a})$ has positive eigenvalues, where recall that the aerodynamic ROM is,

$$\tilde{E}_\mathrm{a}\frac{d\tilde{\boldsymbol{u}}_\mathrm{a}}{dt} = \tilde{A}_\mathrm{a}\boldsymbol{u}_\mathrm{a} + \tilde{B}_{\mathrm{a},1}\boldsymbol{u}_\mathrm{s} + \tilde{B}_{\mathrm{a},2}\frac{d\boldsymbol{u}_\mathrm{s}}{dt}$$

- Stability of the ROM can be ensured by using a **symmetrizing inner product** for the Petrov-Galerkin projection.[1]
- We use the Hessian of the mathematical entropy as the symmetrizing inner product.



[1] *Barone, M. F., et al., Journal of Computational Physics, Vol. 228, No. 6, 2009, pp. 1932–1946.*

- The flutter point is computed using the direct method for Hopf points by **Griewank and Reddien (1983)**,
- We seek the **dynamic pressure** $q_\infty^{\text{flutter}}$ where the real part of the eigenvalue equals zero for the **aeroelastic ROM**:

$$\tilde{\mathbb{A}}(q_\infty)\tilde{\boldsymbol{v}} = i\tilde{\omega}\tilde{\mathbb{M}}(q_\infty)\tilde{\boldsymbol{v}}.$$

- The residual and state vectors are,

$$\boldsymbol{R}^{\text{Hopf}} = \begin{bmatrix} \tilde{\mathbb{A}}\tilde{\boldsymbol{p}}_1 + \tilde{\omega}\tilde{\mathbb{M}}\tilde{\boldsymbol{p}}_2 \\ \tilde{\mathbb{A}}\tilde{\boldsymbol{p}}_2 - \tilde{\omega}\tilde{\mathbb{M}}\tilde{\boldsymbol{p}}_1 \\ \boldsymbol{c}^T\tilde{\boldsymbol{p}}_1 \\ \boldsymbol{c}^T\tilde{\boldsymbol{p}}_2 - 1 \end{bmatrix}, \qquad \boldsymbol{x}^{\text{Hopf}} = \begin{bmatrix} \tilde{\boldsymbol{p}}_1 \\ \tilde{\boldsymbol{p}}_2 \\ \tilde{\omega} \\ \tilde{V} \end{bmatrix},$$

- Solve for the flutter point using Newton's method,

$$\boldsymbol{x}_{n+1}^{\text{Hopf}} = \boldsymbol{x}_n^{\text{Hopf}} + \left(\frac{\partial \boldsymbol{R}^{\text{Hopf}}}{\partial \boldsymbol{x}^{\text{Hopf}}}\right)^{-1} \boldsymbol{R}^{\text{Hopf}}(\boldsymbol{x}_n^{\text{Hopf}}),$$



- Expand the true flutter point V_f in a Taylor series about the approximate flutter point $\tilde{V}_\mathrm{f}$,

$$V_\mathrm{f} = \tilde{V}_\mathrm{f} + \frac{dV}{d\lambda_\mathrm{r}}\left(0 - \lambda_\mathrm{r}\right) + \mathcal{O}\left(\lambda_\mathrm{r}^2\right), \qquad \text{where} \quad \lambda_\mathrm{r} \equiv \mathrm{Real}\{\lambda\}$$

- This is rearranged into,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \left|\frac{d\lambda_\mathrm{r}}{dV}\right|^{-1} \left|\lambda_\mathrm{r} - \overbrace{\tilde{\lambda}_\mathrm{r}}^{=0}\right|.$$

- Estimate these quantities,

$$\left|\lambda_\mathrm{r} - \tilde{\lambda}_\mathrm{r}\right| \approx \left|\mathrm{Real}\{\eta\}\right|, \qquad \frac{d\lambda_r}{dV} \approx \mathrm{Real}\left\{\left(\tilde{\Phi}^{\mathrm{du}}\tilde{v}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\tilde{\Phi}\tilde{v}\right\}.$$

- Error estimator,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \eta_\mathrm{f} \equiv \left|\mathrm{Real}\left\{\left(\tilde{\Phi}^{\mathrm{du}}\tilde{v}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\tilde{\Phi}\tilde{v}\right\}\right|^{-1} \left|\mathrm{Real}\{\eta\}\right|.$$



- Expand the true flutter point V_f in a Taylor series about the approximate flutter point $\tilde{V}_\mathrm{f}$,

$$V_\mathrm{f} = \tilde{V}_\mathrm{f} + \frac{dV}{d\lambda_\mathrm{r}}\left(0 - \lambda_\mathrm{r}\right) + \mathcal{O}\left(\lambda_\mathrm{r}^2\right), \qquad \text{where} \quad \lambda_\mathrm{r} \equiv \mathsf{Real}\{\lambda\}$$

- This is rearranged into,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \left|\frac{d\lambda_\mathrm{r}}{dV}\right|^{-1} \left|\lambda_\mathrm{r} - \overbrace{\tilde{\lambda}_\mathrm{r}}^{=0}\right|.$$

- Estimate these quantities,

$$\left|\lambda_\mathrm{r} - \tilde{\lambda}_\mathrm{r}\right| \approx \left|\mathsf{Real}\{\eta\}\right|, \qquad \frac{d\lambda_r}{dV} \approx \mathsf{Real}\left\{\left(\tilde{\Phi}^{\mathrm{du}}\tilde{v}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\tilde{\Phi}\tilde{v}\right\}.$$

- Error estimator,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \eta_\mathrm{f} \equiv \left|\mathsf{Real}\left\{\left(\tilde{\Phi}^{\mathrm{du}}\tilde{v}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\tilde{\Phi}\tilde{v}\right\}\right|^{-1} \left|\mathsf{Real}\{\eta\}\right|.$$



- Expand the true flutter point V_f in a Taylor series about the approximate flutter point $\tilde{V}_\mathrm{f}$,

$$V_\mathrm{f} = \tilde{V}_\mathrm{f} + \frac{dV}{d\lambda_\mathrm{r}}\left(0 - \lambda_\mathrm{r}\right) + \mathcal{O}\left(\lambda_\mathrm{r}^2\right), \qquad \text{where} \quad \lambda_\mathrm{r} \equiv \mathsf{Real}\{\lambda\}$$

- This is rearranged into,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \left|\frac{d\lambda_\mathrm{r}}{dV}\right|^{-1} \left|\lambda_\mathrm{r} - \overbrace{\tilde{\lambda}_\mathrm{r}}^{=0}\right|.$$

- Estimate these quantities,

$$\left|\lambda_\mathrm{r} - \tilde{\lambda}_\mathrm{r}\right| \approx \left|\mathsf{Real}\{\eta\}\right|, \qquad \frac{d\lambda_r}{dV} \approx \mathsf{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{\boldsymbol{v}}\right\}.$$

- Error estimator,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \eta_\mathrm{f} \equiv \left|\mathsf{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{\boldsymbol{v}}\right\}\right|^{-1} \left|\mathsf{Real}\{\eta\}\right|.$$



- Expand the true flutter point V_f in a Taylor series about the approximate flutter point $\tilde{V}_\mathrm{f}$,

$$V_\mathrm{f} = \tilde{V}_\mathrm{f} + \frac{dV}{d\lambda_\mathrm{r}}\left(0 - \lambda_\mathrm{r}\right) + \mathcal{O}\left(\lambda_\mathrm{r}^2\right), \qquad \text{where} \quad \lambda_\mathrm{r} \equiv \mathrm{Real}\{\lambda\}$$

- This is rearranged into,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \left|\frac{d\lambda_\mathrm{r}}{dV}\right|^{-1} \left|\lambda_\mathrm{r} - \overbrace{\tilde{\lambda}_\mathrm{r}}^{=0}\right|.$$

- Estimate these quantities,

$$\left|\lambda_\mathrm{r} - \tilde{\lambda}_\mathrm{r}\right| \approx \left|\mathrm{Real}\{\eta\}\right|, \qquad \frac{d\lambda_r}{dV} \approx \mathrm{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{\boldsymbol{v}}\right\}.$$

- Error estimator,

$$\left|V_\mathrm{f} - \tilde{V}_\mathrm{f}\right| \approx \eta_\mathrm{f} \equiv \left|\mathrm{Real}\left\{\left(\check{\Phi}^{\mathrm{du}}\tilde{\boldsymbol{v}}^{\mathrm{du}}\right)^H \left(\frac{d\mathbb{A}}{dV} - \tilde{\lambda}\frac{d\mathbb{M}}{dV}\right)\check{\Phi}\tilde{\boldsymbol{v}}\right\}\right|^{-1} \left|\mathrm{Real}\{\eta\}\right|.$$



Jacobi-Davidson Stopping Criterion

- The **Rayleigh Quotient** θ of the right eigenvector $\check{\boldsymbol{v}}_{\mathsf{jd}}$ approximates the eigenvalue:

$$\lambda \approx \theta = \frac{\check{\boldsymbol{v}}_{\mathsf{jd}}^H \mathbb{A} \check{\boldsymbol{v}}_{\mathsf{jd}}}{\check{\boldsymbol{v}}_{\mathsf{jd}}^H \mathbb{M} \check{\boldsymbol{v}}_{\mathsf{jd}}}$$

- Using the dual-weighted approach, we approximate the error in θ,

$$\lambda - \theta \approx \eta_{\mathsf{jd}} \equiv \left(\check{\boldsymbol{v}}_{\mathsf{jd}}^{\mathsf{du}}\right)^H \left(\mathbb{A}\check{\boldsymbol{v}}_{\mathsf{jd}} - \theta \mathbb{M}\check{\boldsymbol{v}}_{\mathsf{jd}}\right).$$

