Some Contributions to Computational Physics

Brian A. Freno
Sandia National Laboratories

Texas A&M University
September 24, 2024

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Biography: Brian Freno

Education: BS, MS, PhD in Aerospace Engineering at Texas A&M University

Work Experience:

Oct. 2015 – Present	Sandia National Laboratories
June 2014 – Sept. 2015	Halliburton
Summers 2012 & 2013	NASA Marshall Space Flight Center
Summers 2007 & 2008	Lockheed Martin Missiles and Fire Control
Summers 2005 & 2006	Standard Aero

Research Areas: reduced-order modeling, code verification, machine learning, computational fluid dynamics, and computational electromagnetics

Publications: primary author of several journal articles and one patent

Service:

- Associate fellow of AIAA, member of ASME & SIAM
- Associate editor for the ASME Journal of VVUQ
- Serve on AIAA Fluid Dynamics Technical Committee
- Adjunct professor in Texas A&M Department of Aerospace Engineering
- Conference session organizer, program & journal reviewer, mentor, recruiter

Machine-Learning Error Models for Approximate Solutions to Parameterized Systems of Nonlinear Equations

Brian A. Freno
Kevin T. Carlberg
Sandia National Laboratories

Motivation

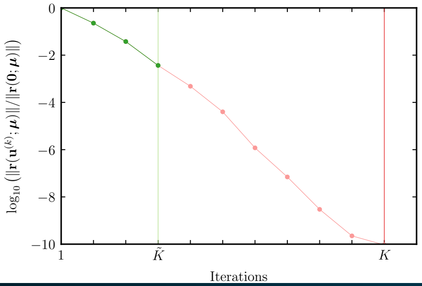
- Many-query problems can impose a formidable computational burden
- **Solution approximations** can exchange fidelity for speed
- Need to quantify the error

Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely terminate iterations
- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements
- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions

Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely terminate iterations
- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements
- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions

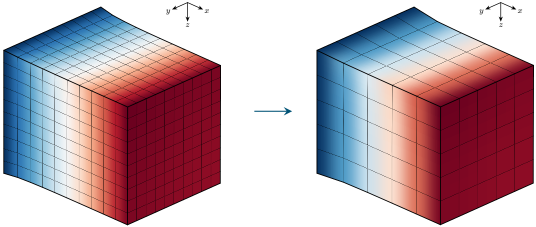


Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely terminate iterations

- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements

- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions



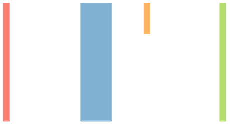
Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely terminate iterations

- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements

- **Reduced-order models:** Approximate solution with a linear combination of $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ basis functions

$$\tilde{\mathbf{u}}(\boldsymbol{\mu}) = \boldsymbol{\Phi}_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}) + \bar{\mathbf{u}}$$



Parameterized Systems of Nonlinear Equations

- Parameterized systems of nonlinear equations

$$\mathbf{r}(\mathbf{u}(\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbf{0}$$

  - $\mathbf{r} : \mathbb{R}^{N_\mathbf{u}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_\mathbf{u}}$ residual, nonlinear in at least $\mathbf{u}(\boldsymbol{\mu})$
  - $\mathbf{u} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_\mathbf{u}}$ state (solution vector)
  - $\boldsymbol{\mu} \in \mathcal{D}$ parameters in parameter domain $\mathcal{D} \subseteq \mathbb{R}^{N_{\boldsymbol{\mu}}}$

- Scalar-valued quantity of interest

$$s(\boldsymbol{\mu}) := g(\mathbf{u}(\boldsymbol{\mu}))$$

  - $s : \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}$ quantity of interest
  - $g : \mathbb{R}^{N_\mathbf{u}} \rightarrow \mathbb{R}$ quantity of interest functional

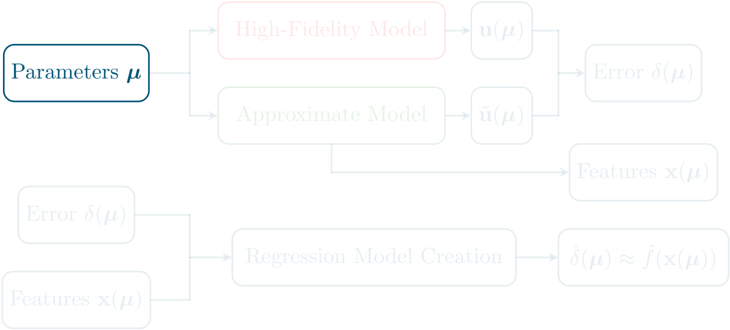
Error Model Construction Steps

1) **Feature engineering**
   - Cheaply computable features $\mathbf{x}$ from approximate model
   - Informative of the error – construct low-noise-variance model
   - Low dimensional (small $N_{\mathbf{x}}$) such that less training data are needed

2) **Regression-function approximation**
   - Construct $\hat{f}$ using regression methods from machine learning
   - Approximate mapping from features $\mathbf{x}$ to error δ using a training set

Summary

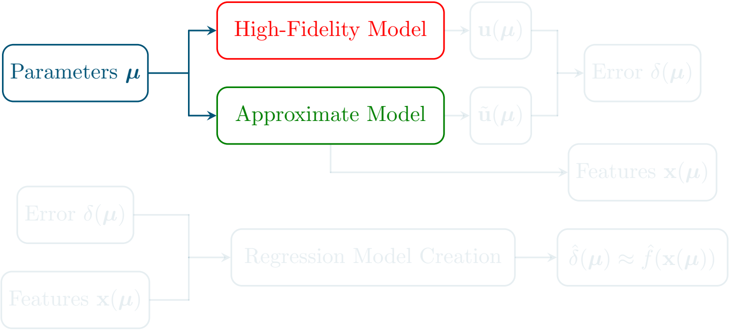
Training



Application

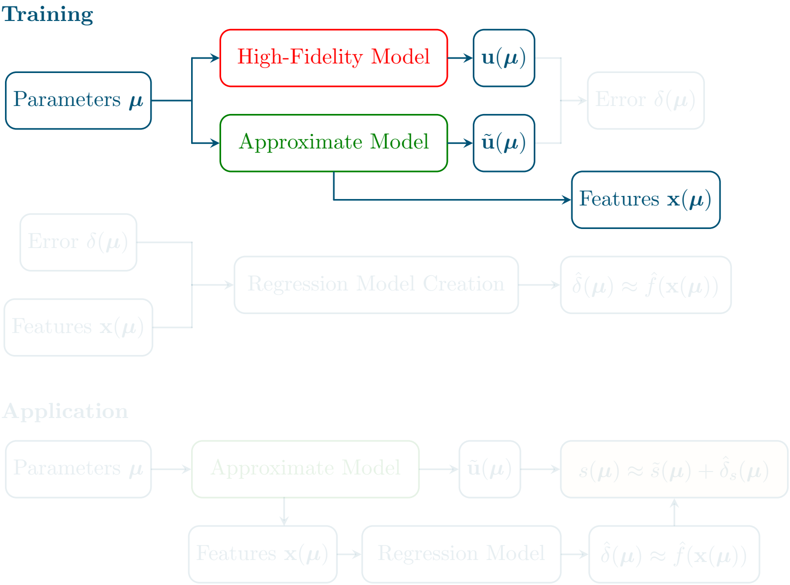
Summary

Training



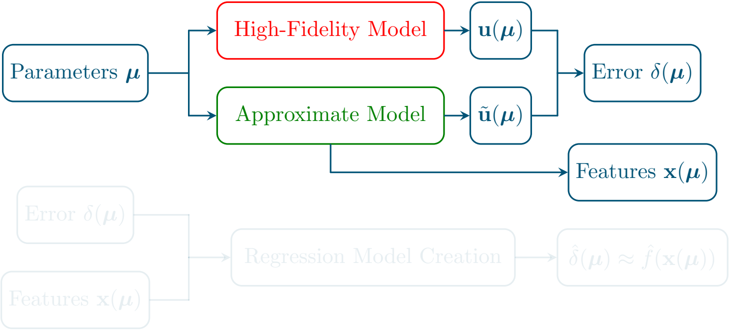
Application

Summary


Training
Parameters $\boldsymbol{\mu}$
High-Fidelity Model
$\mathbf{u}(\boldsymbol{\mu})$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
Error $\delta(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Error $\delta(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Regression Model Creation
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$
Application
Parameters $\boldsymbol{\mu}$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
$s(\boldsymbol{\mu}) \approx \tilde{s}(\boldsymbol{\mu}) + \hat{\delta}_s(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Regression Model
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$

Summary

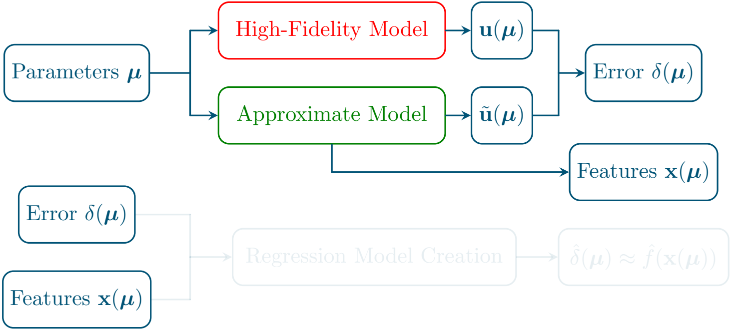
Training



Application

Summary

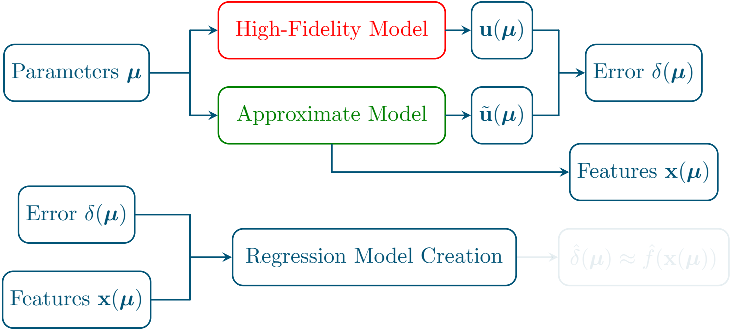
Training



Application

Summary

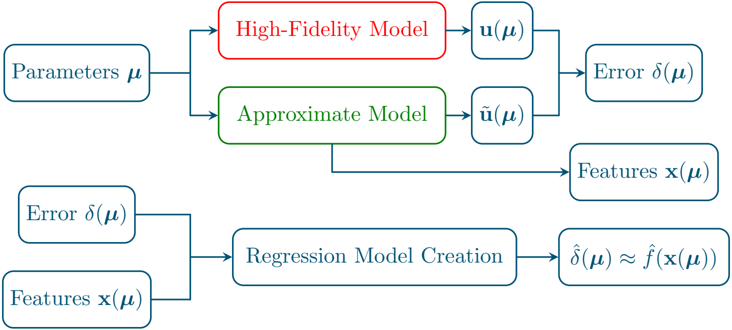
Training



Application

Summary

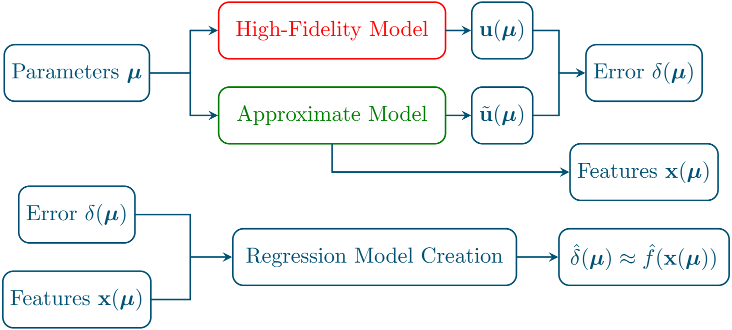
Training



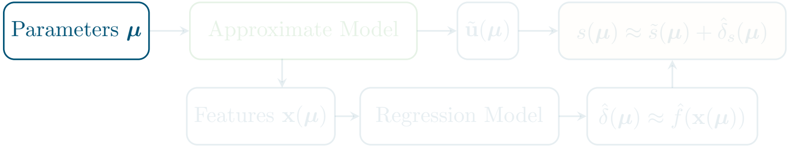
Application

Summary

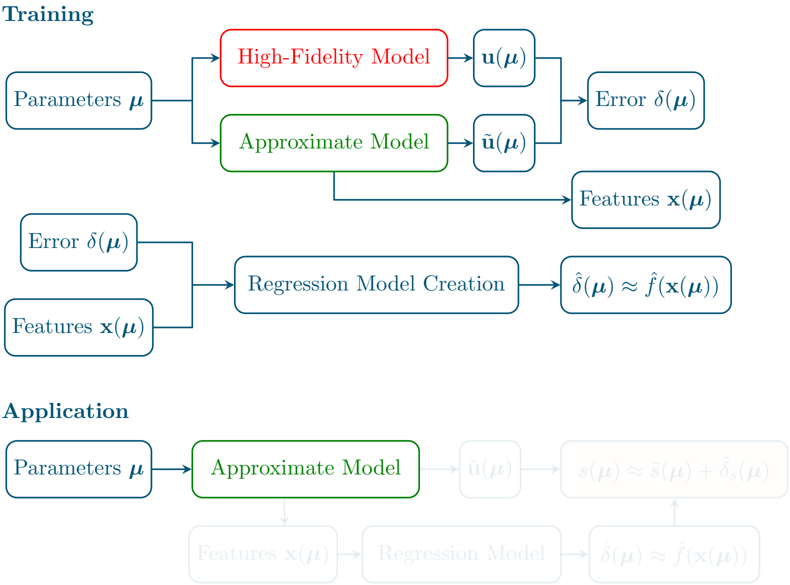
Training



Application

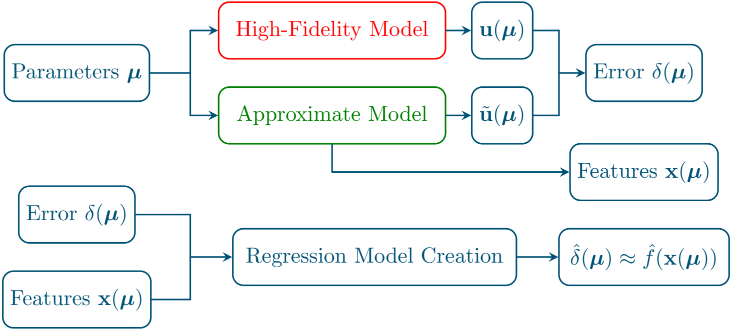


Summary

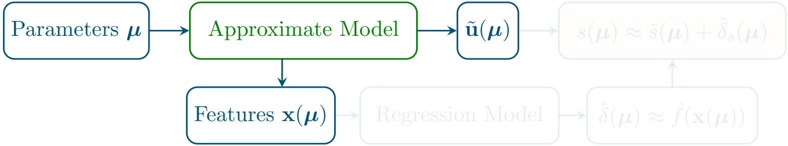

Training
High-Fidelity Model
$\mathbf{u}(\boldsymbol{\mu})$
Parameters $\boldsymbol{\mu}$
Error $\delta(\boldsymbol{\mu})$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Error $\delta(\boldsymbol{\mu})$
Regression Model Creation
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$
Features $\mathbf{x}(\boldsymbol{\mu})$
Application
Parameters $\boldsymbol{\mu}$
Approximate Model
$\tilde{\mathbf{u}}(\boldsymbol{\mu})$
$s(\boldsymbol{\mu}) \approx \tilde{s}(\boldsymbol{\mu}) + \hat{\delta}_s(\boldsymbol{\mu})$
Features $\mathbf{x}(\boldsymbol{\mu})$
Regression Model
$\hat{\delta}(\boldsymbol{\mu}) \approx \hat{f}(\mathbf{x}(\boldsymbol{\mu}))$

Summary

Training

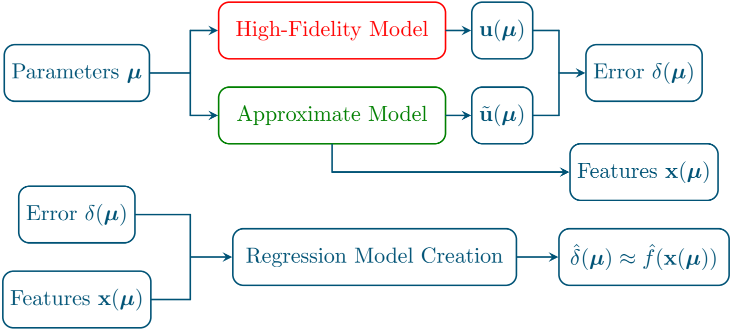


Application

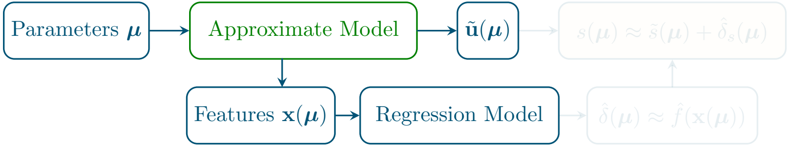


Summary

Training

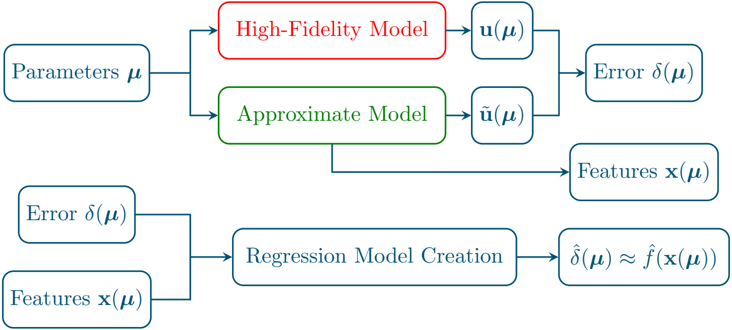


Application

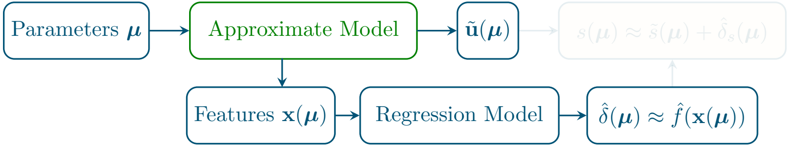


Summary

Training

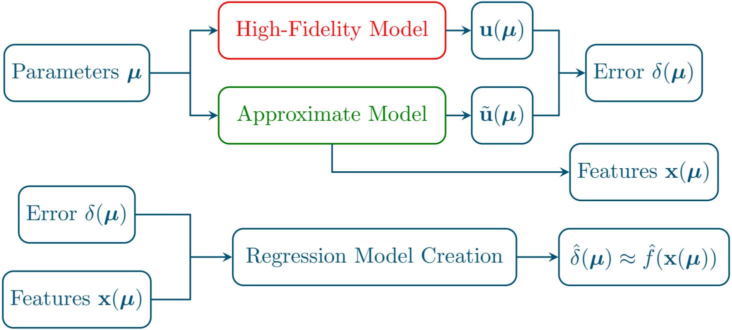


Application

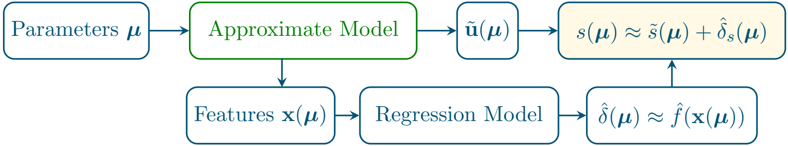


Summary

Training



Application



Dual-Weighted Residual

Approximate residual about approximate solution $\tilde{\mathbf{u}}$:

$$\mathbf{r}(\mathbf{u}(\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbf{0} = \underbrace{\mathbf{r}(\tilde{\mathbf{u}}(\boldsymbol{\mu});\boldsymbol{\mu})}_{\mathbf{r}(\boldsymbol{\mu})} + \underbrace{\frac{\partial \mathbf{r}}{\partial \mathbf{v}}(\tilde{\mathbf{u}}(\boldsymbol{\mu});\boldsymbol{\mu})}_{\mathbf{J}(\boldsymbol{\mu})} \underbrace{(\mathbf{u}(\boldsymbol{\mu}) - \tilde{\mathbf{u}}(\boldsymbol{\mu}))}_{\mathbf{e}(\boldsymbol{\mu})} + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

Rearrange to approximate state-space error: $\mathbf{e}(\boldsymbol{\mu}) = -\mathbf{J}(\boldsymbol{\mu})^{-1}\mathbf{r}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$ (1)

Approximate quantity of interest about $\tilde{\mathbf{u}}$: $s(\boldsymbol{\mu}) = \tilde{s}(\boldsymbol{\mu}) + \frac{\partial g}{\partial \mathbf{v}}(\tilde{\mathbf{u}}(\boldsymbol{\mu}))\mathbf{e}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$

Combine with state-space error approximation (1):

$$\delta_s(\boldsymbol{\mu}) = \underbrace{-\frac{\partial g}{\partial \mathbf{v}}(\tilde{\mathbf{u}}(\boldsymbol{\mu}))\mathbf{J}(\boldsymbol{\mu})^{-1}}_{\mathbf{y}(\boldsymbol{\mu})^T\text{: dual or adjoint}} \mathbf{r}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

Dual-weighted residual d is weighted sum of residual elements:

$$d(\boldsymbol{\mu}) := \mathbf{y}(\boldsymbol{\mu})^T\mathbf{r}(\boldsymbol{\mu}) = \sum_{i=1}^{N_\mathbf{u}} y_i(\boldsymbol{\mu})r_i(\boldsymbol{\mu})$$

Drawbacks to using the Dual-Weighted Residual

- **Computational Cost:** requires solving $N_{\mathbf{u}}$ linear equations
- **Implementation:** requires Jacobian – not always available

Nonetheless, structure provides insight into quantity-of-interest error

Feature Engineering: Parameters

$$\mathbf{x}(\boldsymbol{\mu}) = \boldsymbol{\mu}$$

- The mapping $\boldsymbol{\mu} \mapsto \delta(\boldsymbol{\mu})$ is deterministic, but often complex
  - Can be oscillatory for ROMs since $\delta(\boldsymbol{\mu}) \approx 0$ when $\boldsymbol{\mu} \in \mathcal{D}_{\text{Train}}^{\text{ROM}}$
- Inspired by 'multifidelity correction' methods for optimization

  Alexandrov et al., 2001; Gano et al., 2005; Eldred et al., 2004

Feature Engineering: Dual-Weighted Residual

$$\mathbf{x}(\boldsymbol{\mu}) = d(\boldsymbol{\mu}) := \mathbf{y}(\boldsymbol{\mu})^T \mathbf{r}(\boldsymbol{\mu})$$

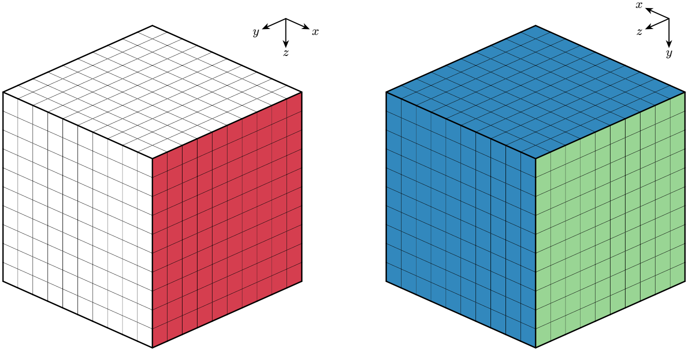
- First-order approximation of QoI error $\delta_s(\boldsymbol{\mu})$
- Small number ($N_{\mathbf{x}} = 1$) of high-quality features
- High computational cost and significant implementation effort

Feature Engineering: Parameters and Residual (Approximations)

$$\mathbf{x}(\boldsymbol{\mu}) = [\boldsymbol{\mu}; \mathbf{r}(\boldsymbol{\mu})]$$

- DWR is weighted sum of residual vector elements $d(\boldsymbol{\mu}) := \mathbf{y}(\boldsymbol{\mu})^T \mathbf{r}(\boldsymbol{\mu})$
- Avoids implementation and costs associated with dual vector $\mathbf{y}(\boldsymbol{\mu})$
- Large number ($N_{\mathbf{x}} = N_{\boldsymbol{\mu}} + N_{\mathbf{u}}$) of low-quality features
- Approaches to reduce number of features and improve quality
  - Use $m_{\mathbf{r}} \ll N_{\mathbf{u}}$ principal component coefficients: $\hat{\mathbf{r}}(\boldsymbol{\mu})$
  - Sample $n_{\mathbf{r}} \ll N_{\mathbf{u}}$ elements of residual: $\mathbf{P}\mathbf{r}(\boldsymbol{\mu})$, where $\mathbf{P} \in \{0, 1\}^{n_{\mathbf{r}} \times N_{\mathbf{u}}}$
  - Use $m_{\mathbf{r}} \ll N_{\mathbf{u}}$ gappy principal component coefficients: $\hat{\mathbf{r}}_{\mathrm{g}}(\boldsymbol{\mu})$

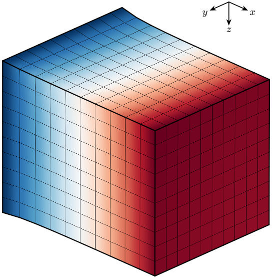
Cube: Reduced-Order Modeling

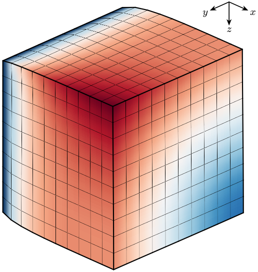


- Applied traction (Neumann boundary condition)
- Planar constraint (Dirichlet boundary condition)
- Complete constraint (Dirichlet boundary condition)
- Node of interest

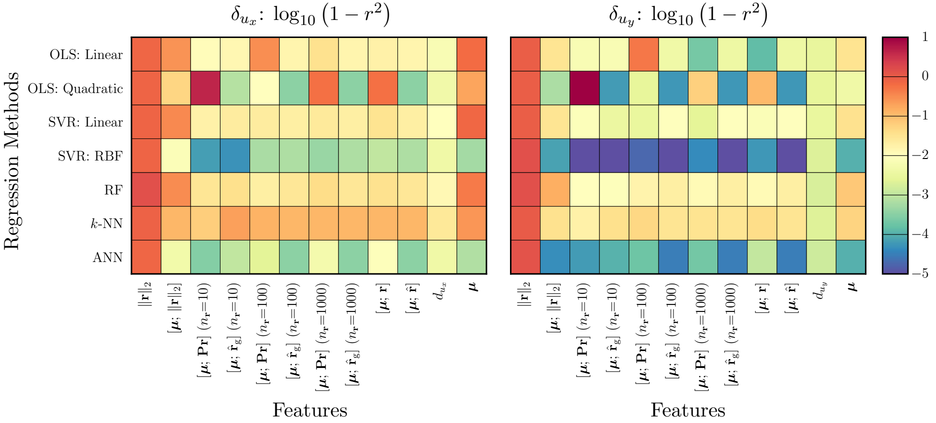
Cube: Overview

- $N_{\mathbf{u}} = 3410$ – deliberately small to compute $d(\boldsymbol{\mu})$ and use $\mathbf{r}(\boldsymbol{\mu})$
- $N_{\boldsymbol{\mu}} = 3$ parameters: $\boldsymbol{\mu} = [E; \nu; t]$
  - $E \in [75, 125]$ GPa, $\nu \in [0.20, 0.35]$, $t \in [40, 60]$ GPa
- 8 HF runs $\rightarrow$ up to $m_{\mathbf{u}} = 8$ ROM basis vectors (2 used – 99.49%)

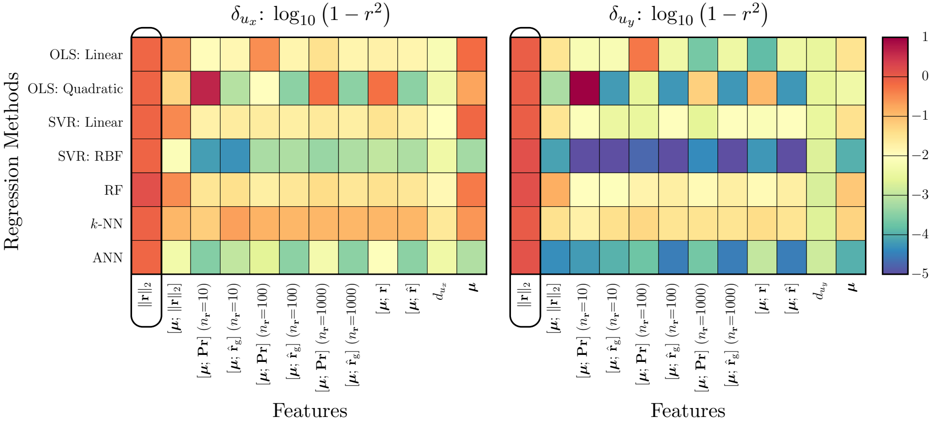




Cube: Variance Unexplained for QoI Error Prediction

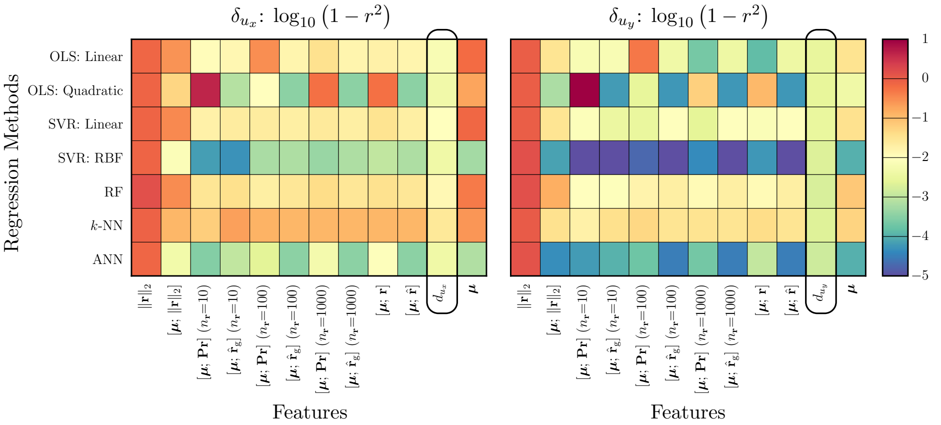


Cube: Variance Unexplained for QoI Error Prediction



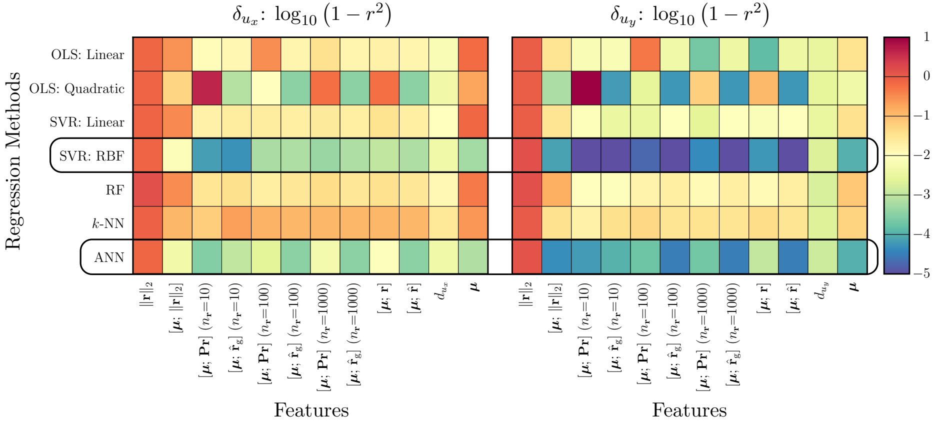
- $\|\mathbf{r}\|_2$ yields highest variance unexplained

Cube: Variance Unexplained for QoI Error Prediction



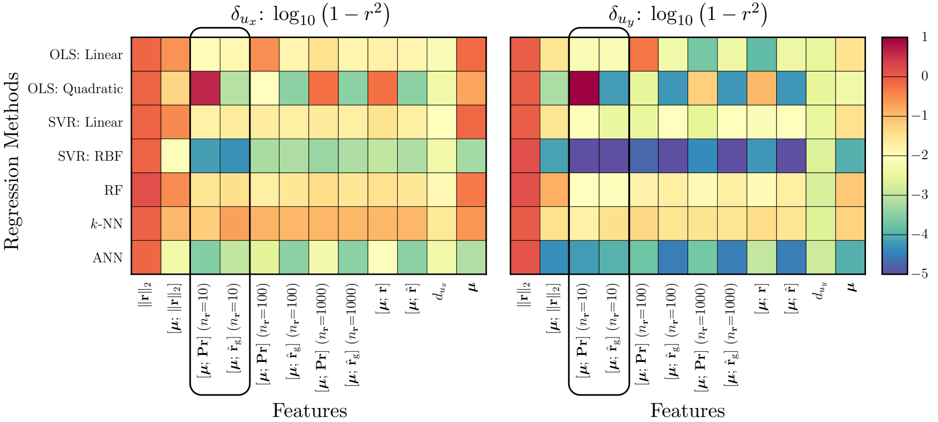
- $\|\mathbf{r}\|_2$ yields highest variance unexplained
- d_{u_x} and d_{u_y} yield moderate variance unexplained, but are costly

Cube: Variance Unexplained for QoI Error Prediction



- $\|\mathbf{r}\|_2$ yields highest variance unexplained
- d_{u_x} and d_{u_y} yield moderate variance unexplained, but are costly
- SVR: RBF and ANN yield lowest variance unexplained

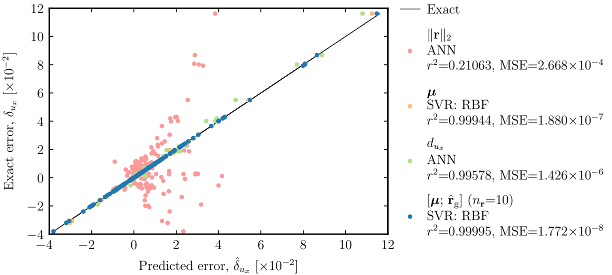
Cube: Variance Unexplained for QoI Error Prediction

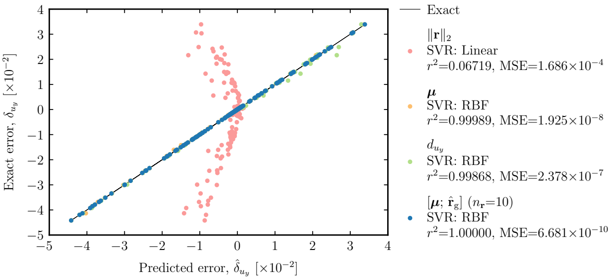


- $\|\mathbf{r}\|_2$ yields highest variance unexplained
- d_{u_x} and d_{u_y} yield moderate variance unexplained, but are costly
- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_\mathrm{g}]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_\mathbf{u} = 3410$)

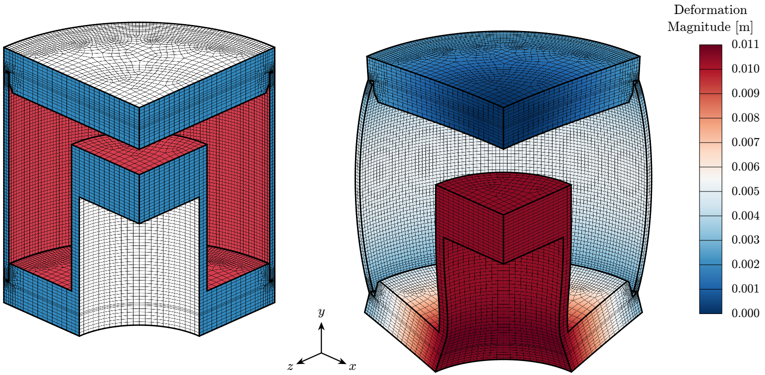
Cube: QoI Error Predictions

- Our method beats previous state-of-the-art methods with $r^2 > 0.9999$ in both cases





Predictive Capability Assessment Project: Reduced-Order Modeling

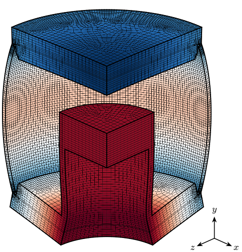


- Applied pressure (Neumann boundary condition)
- Planar constraint (Dirichlet boundary condition)
- Complete constraint (Dirichlet boundary condition)
- Nodes of interest

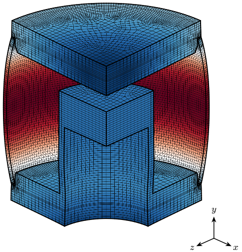
PCAP: Overview

- $N_{\mathbf{u}} = 274,954$ for quarter of domain
- $N_{\boldsymbol{\mu}} = 3$ parameters: $\boldsymbol{\mu} = [E;\, \nu;\, p]$
  - $E \in [50,\ 100]$ GPa, $\nu \in [0.20,\ 0.35]$, $p \in [6,\ 10]$ GPa
- 30 parameter training instances for regression model

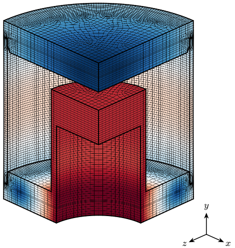
PCAP: Basis Vectors



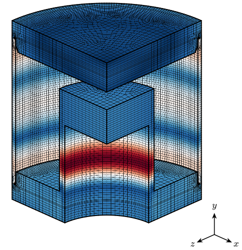
1: 85.03%



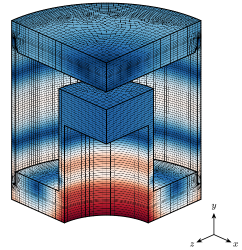
2: 95.69%



3: 99.35%

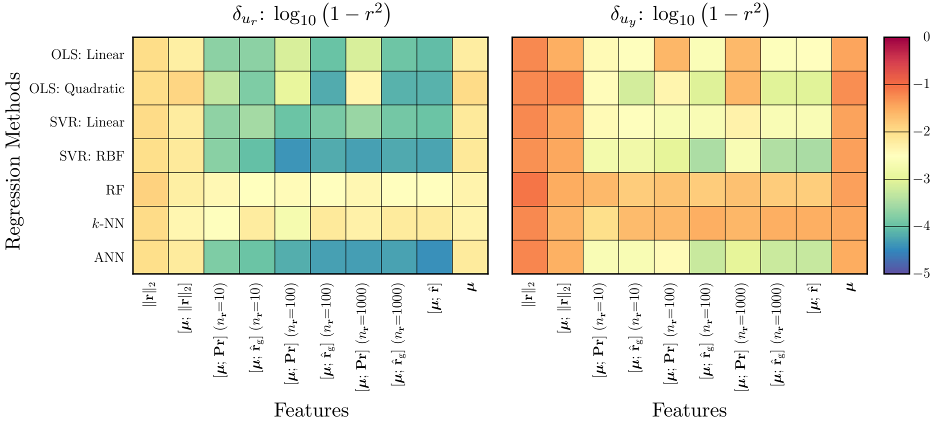


4: 99.77%

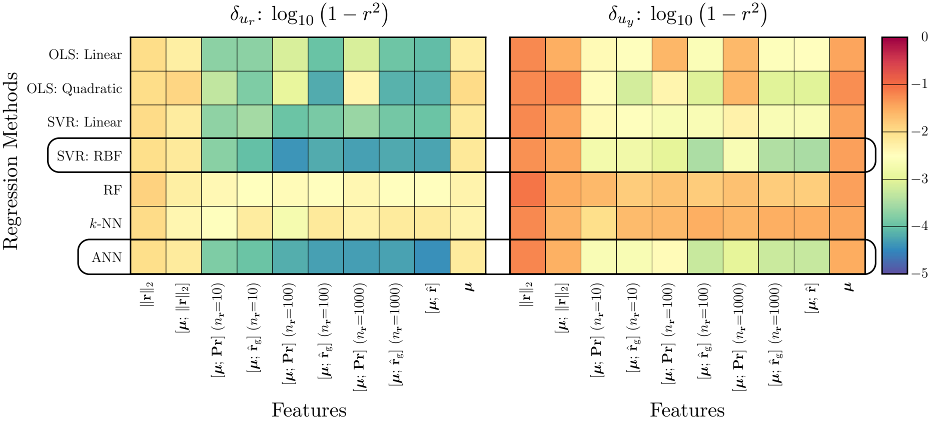


5: 99.90%

PCAP: Variance Unexplained for QoI Error Prediction

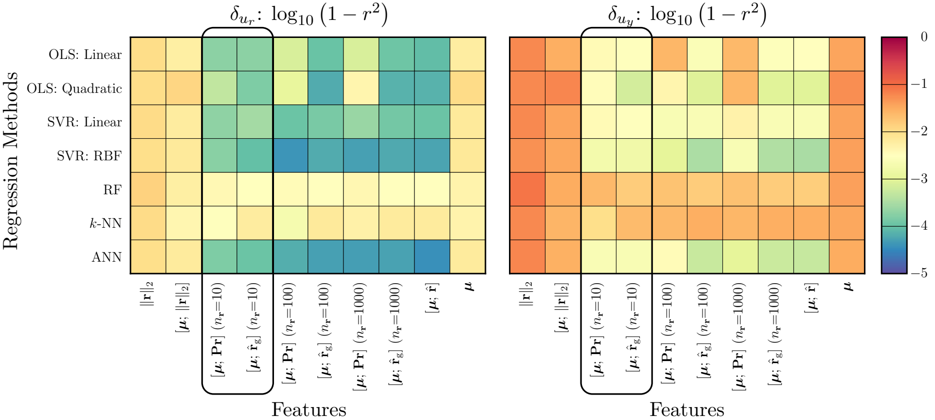


PCAP: Variance Unexplained for QoI Error Prediction



- SVR: RBF and ANN yield lowest variance unexplained

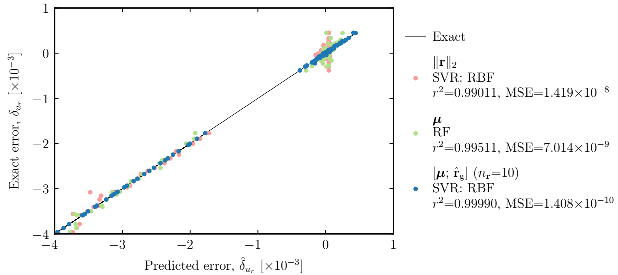
PCAP: Variance Unexplained for QoI Error Prediction

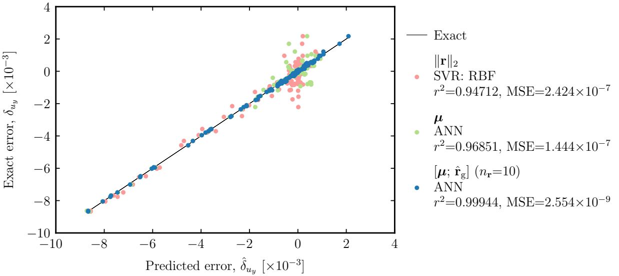


- SVR: RBF and ANN yield lowest variance unexplained
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_\mathrm{g}]$ and $[\boldsymbol{\mu}; \mathbf{Pr}]$ yield low variance unexplained with only 10 samples (compared to $N_\mathbf{u} = 274,954$)

PCAP: QoI Error Predictions

- Our method beats previous state-of-the-art methods with $r^2 > 0.9994$ in both cases





Summary

- Accurately computed error from approximate solutions
- $r^2 > 0.996$ for all experiments
- Only used about 13 features

Code-Verification Techniques for Computational Physics

Verification and Validation

Credibility of computational physics codes requires verification and validation

- **Validation** assesses how well models represent physical phenomena
  - Compare computational results with experimental results
  - Assess suitability of models, model error, and bounds of validity

- **Verification** assesses accuracy of numerical solutions against expectations
  - *Solution verification* estimates numerical error for particular solution
  - *Code verification* assesses correctness of numerical-method implementation

Code verification is the focus of this part

Importance of Code Verification

Consequences of incorrectly implemented numerical methods:

- Wasted computational expense:
  - Numerical errors **decrease slower** with mesh refinement than they should
  - Numerical errors **do not decrease** with mesh refinement
- Numerical solutions **differ moderately** from exact solutions
- Numerical solutions **differ significantly** from exact solutions
- **Application failure** from incorrect numerical results
- Consequences may not be obvious

Benefits of code verification:

- Assess suitability and correctness of numerical methods
- Compare accuracy of different algorithms
- Set expectations and estimate error for solution verification

Code Verification

Code verification assesses correctness of numerical-method implementation

- Continuous equations are numerically discretized

$$\mathbf{r}(\mathbf{u}) = \mathbf{0} \quad \rightarrow \quad \mathbf{r}_h(\mathbf{u}_h) = \mathbf{0}$$

- Discretization error is introduced in solution

$$\mathbf{e} = \mathbf{u}_h - \mathbf{u}$$

- Discretization error should decrease as discretization is refined

$$\lim_{h\rightarrow 0} \mathbf{e} = \mathbf{0}$$

- More rigorously, should decrease at an expected rate

$$\|\mathbf{e}\| \approx C h^p$$

- Measuring error requires exact solution – usually unavailable

Manufactured Solutions: Overview

Manufactured solutions are popular alternative

- Manufacture an arbitrary solution $\mathbf{u}_{\text{MS}}$
- Insert manufactured solution into continuous equations to get residual term

$$\mathbf{r}(\mathbf{u}_{\text{MS}}) \neq \mathbf{0}$$

- Add residual term to discretized equations

$$\mathbf{r}_h(\mathbf{u}_h) = \mathbf{r}(\mathbf{u}_{\text{MS}})$$

to coerce solution to manufactured solution

$$\mathbf{u}_h \to \mathbf{u}_{\text{MS}}$$

Manufactured Solutions: An Example

Consider Laplace's equation in 1D $\quad r(u) = \frac{\partial^2 u}{\partial x^2} = 0$

Discretized by finite differences

$$r_h(u_h) = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = 0$$

Manufacture the arbitrary solution $u_{\mathrm{MS}}(x) = \sin(\pi x)$, such that

$$r(u_{\mathrm{MS}}) = \frac{\partial^2 u_{\mathrm{MS}}}{\partial x^2} = -\pi^2 \sin(\pi x)$$

Solve $r_h(u_h) = r(u_{\mathrm{MS}})$, such that

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = -\pi^2 \sin(\pi x_i)$$

The discretization error is $e = u_h - u_{\mathrm{MS}}$

Code-Verification Techniques for Hypersonic Reacting Flows in Thermochemical Nonequilibrium

Brian A. Freno
Brian R. Carnes
V. Gregory Weirs
Sandia National Laboratories

Governing Equations: n_s Species in Vibrational Nonequilibrium

Conservation of mass, momentum, and energy:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot\mathbf{F}_c(\mathbf{u}) = -\nabla\cdot\mathbf{F}_p(\mathbf{u}) + \nabla\cdot\mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

where

$$\mathbf{u} = \begin{Bmatrix} \boldsymbol{\rho} \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$\boldsymbol{\rho} = \{\rho_1, \dots, \rho_{n_s}\}^T$, $\quad \dot{\mathbf{w}} = \{\dot{w}_1, \dots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$\rho = \sum_{s=1}^{n_s} \rho_s$, $\quad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s}\bar{R}T$, $\quad \mathbf{e}_v = \{e_{v_1}, \dots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v}: translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s}T + e_{v_s} + h_s^o\right)$$

Governing Equations: n_s Species in Vibrational Nonequilibrium

Conservation of mass, momentum, and energy:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

where

$$\mathbf{u} = \begin{Bmatrix} \boldsymbol{\rho} \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

Multiple species

$\boldsymbol{\rho} = \{\rho_1, \dots, \rho_{n_s}\}^T$,

$\rho = \sum_{s=1}^{n_s} \rho_s$,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s}\bar{R}T$,

$\dot{\mathbf{w}} = \{\dot{w}_1, \dots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$\mathbf{e}_v = \{e_{v_1}, \dots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v} : translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s}T + e_{v_s} + h_s^o\right)$$

Governing Equations: n_s Species in Vibrational Nonequilibrium

Conservation of mass, momentum, and energy:

Local time derivative

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

where

$$\mathbf{u} = \begin{Bmatrix} \boldsymbol{\rho} \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$\boldsymbol{\rho} = \{\rho_1, \dots, \rho_{n_s}\}^T$, $\quad \dot{\mathbf{w}} = \{\dot{w}_1, \dots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$\rho = \sum_{s=1}^{n_s} \rho_s$, $\quad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s} \bar{R} T$, $\quad \mathbf{e}_v = \{e_{v_1}, \dots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v}: translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s} T + e_{v_s} + h_s^o\right)$$

Governing Equations: n_s Species in Vibrational Nonequilibrium

Conservation of mass, momentum, and energy:

Convective flux gradient

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

where

$$\mathbf{u} = \begin{Bmatrix} \boldsymbol{\rho} \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$\boldsymbol{\rho} = \{\rho_1, \dots, \rho_{n_s}\}^T$, $\quad \dot{\mathbf{w}} = \{\dot{w}_1, \dots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$\rho = \sum_{s=1}^{n_s} \rho_s$, $\quad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s} \bar{R} T$, $\quad \mathbf{e}_v = \{e_{v_1}, \dots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v}: translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s} T + e_{v_s} + h_s^o\right)$$

Governing Equations: n_s Species in Vibrational Nonequilibrium

Conservation of mass, momentum, and energy:

Pressure flux gradient

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

where

$$\mathbf{u} = \begin{Bmatrix} \boldsymbol{\rho} \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$\boldsymbol{\rho} = \{\rho_1, \ldots, \rho_{n_s}\}^T$, $\quad \dot{\mathbf{w}} = \{\dot{w}_1, \ldots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$\rho = \sum_{s=1}^{n_s} \rho_s$, $\quad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s}\bar{R}T$, $\quad \mathbf{e}_v = \{e_{v_1}, \ldots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v}: translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s}T + e_{v_s} + h_s^o\right)$$

Governing Equations: n_s Species in Vibrational Nonequilibrium

Conservation of mass, momentum, and energy:

Diffusive flux gradient

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

where

$$\mathbf{u} = \begin{Bmatrix} \boldsymbol{\rho} \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$\boldsymbol{\rho} = \{\rho_1, \ldots, \rho_{n_s}\}^T$, $\quad \dot{\mathbf{w}} = \{\dot{w}_1, \ldots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$\rho = \sum_{s=1}^{n_s} \rho_s$, $\quad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s} \bar{R}T$, $\quad \mathbf{e}_v = \{e_{v_1}, \ldots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v}: translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s} T + e_{v_s} + h_s^o\right)$$

Governing Equations: n_s Species in Vibrational Nonequilibrium

Conservation of mass, momentum, and energy:

Thermochemical source term

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

where

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$\boldsymbol{\rho} = \{\rho_1, \ldots, \rho_{n_s}\}^T$, $\quad \dot{\mathbf{w}} = \{\dot{w}_1, \ldots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$\rho = \sum_{s=1}^{n_s} \rho_s$, $\quad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s} \bar{R}T$, $\quad \mathbf{e}_v = \{e_{v_1}, \ldots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v}: translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s}T + e_{v_s} + h_s^o\right)$$

Scope of Code Verification

Conservation of mass, momentum, and energy:

Non-diffusive flux gradients

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

Spatial discretization

where

$$\mathbf{u} = \begin{Bmatrix} \boldsymbol{\rho} \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \boldsymbol{\rho}\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$\boldsymbol{\rho} = \{\rho_1, \dots, \rho_{n_s}\}^T$, $\quad \dot{\mathbf{w}} = \{\dot{w}_1, \dots, \dot{w}_{n_s}\}^T$: mass production rates per volume,

$\rho = \sum_{s=1}^{n_s} \rho_s$, $\quad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s}$: mixture vibrational energy per mass,

$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s} \bar{R} T$, $\quad \mathbf{e}_v = \{e_{v_1}, \dots, e_{v_{n_s}}\}^T$: vibrational energies per mass,

Q_{t-v} : translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s} T + e_{v_s} + h_s^o\right)$$

Scope of Code Verification

Conservation of mass, momentum, and energy:

Thermochemical source term

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) = -\nabla \cdot \mathbf{F}_p(\mathbf{u}) + \nabla \cdot \mathbf{F}_d(\mathbf{u}) + \mathbf{S}(\mathbf{u}),$$

Implementation

where

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \rho\mathbf{v}^T \\ \rho\mathbf{v}\mathbf{v}^T \\ \rho E\mathbf{v}^T \\ \rho e_v\mathbf{v}^T \end{bmatrix}, \quad \mathbf{F}_p(\mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ p\mathbf{I} \\ p\mathbf{v}^T \\ \mathbf{0}^T \end{bmatrix}, \quad \mathbf{F}_d(\mathbf{u}) = \begin{bmatrix} -\mathbf{J} \\ \boldsymbol{\tau} \\ \left(\boldsymbol{\tau}\mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T\mathbf{h}\right)^T \\ \left(-\mathbf{q}_v - \mathbf{J}^T\mathbf{e}_v\right)^T \end{bmatrix},$$

$$\mathbf{S}(\mathbf{u}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T\dot{\mathbf{w}} \end{Bmatrix},$$

$$\boldsymbol{\rho} = \{\rho_1, \dots, \rho_{n_s}\}^T, \qquad \dot{\mathbf{w}} = \{\dot{w}_1, \dots, \dot{w}_{n_s}\}^T : \text{ mass production rates per volume},$$

$$\rho = \sum_{s=1}^{n_s} \rho_s, \qquad e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s} : \text{ mixture vibrational energy per mass},$$

$$p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s} \bar{R} T, \qquad \mathbf{e}_v = \left\{e_{v_1}, \dots, e_{v_{n_s}}\right\}^T : \text{ vibrational energies per mass},$$

Q_{t-v} : translational–vibrational energy exchange,

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho}\left(c_{\mathcal{V}_s} T + e_{v_s} + h_s^o\right)$$

2D Supersonic Flow using a Manufactured Solution

- Two-dimensional domain: $(x, y) \in [0,\ 1]\ \mathrm{m} \times [0,\ 1]\ \mathrm{m}$
- Boundary conditions:
  - Supersonic inflow ($x = 0$ m)
  - Supersonic outflow ($x = 1$ m)
  - Slip wall (tangent flow) ($y = 0$ m & $y = 1$ m)
- 5 nonuniform meshes: $25 \times 25 \rightarrow 400 \times 400$
- Solution consists of small, smooth perturbations to uniform flow:

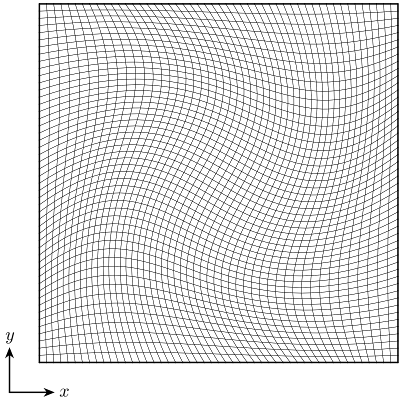
$$\rho(x,y) = \bar{\rho}\left[1 - \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\right],$$

$$u(x,y) = \bar{u}\left[1 + \epsilon \sin\left(\tfrac{1}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\right],$$

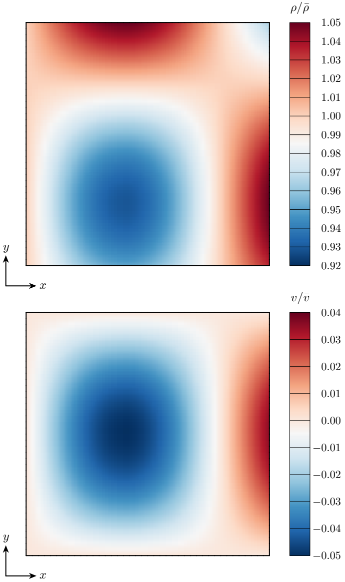
$$v(x,y) = \bar{v}\left[\ \ - \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y)\qquad\qquad\right)\right],$$

$$T(x,y) = \bar{T}\left[1 + \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\right],$$

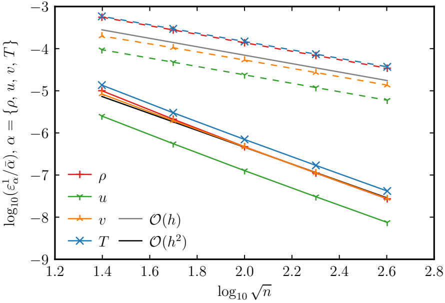
$\bar{\rho} = 1\ \mathrm{kg/m^3}$, $\bar{T} = 300$ K, $\bar{M} = 2.5$, $\epsilon = 0.05$

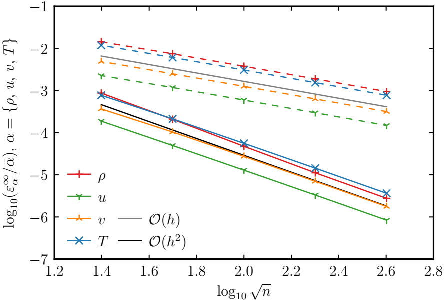


2D Supersonic Flow using a Manufactured Solution



2D Supersonic Flow using a Manufactured Solution





	First-order accurate				Second-order accurate			
	Original boundary conditions				Corrected boundary conditions			
Mesh	ρ	u	v	T	ρ	u	v	T
1–2	0.9420	0.9409	0.9721	0.9628	2.0623	1.9188	1.8174	1.8598
2–3	0.9850	0.9902	0.9910	0.9874	2.1304	1.9450	1.9221	1.9280
3–4	0.9960	1.0002	0.9924	0.9952	2.0902	1.9603	1.9671	1.9586
4–5	0.9989	1.0009	0.9959	0.9984	2.0128	1.9823	1.9860	1.9809

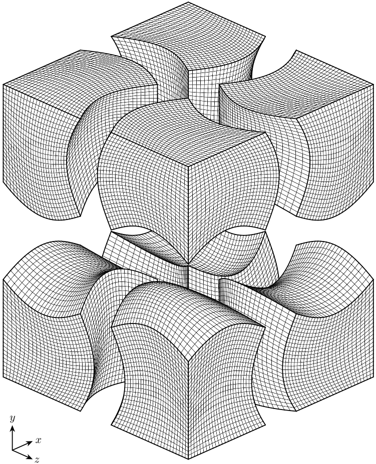
Observed accuracy p using L^∞-norms of the error

3D Supersonic Flow using a Manufactured Solution

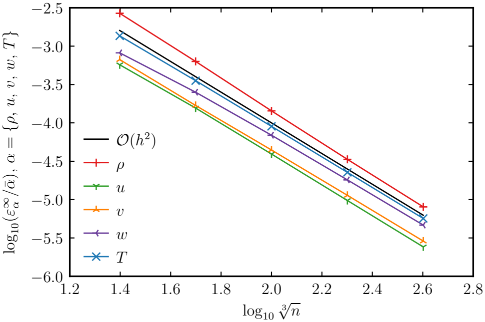
- Three-dimensional domain: $(x, y, z) \in [0,\ 1]\ \text{m} \times [0,\ 1]\ \text{m} \times [0,\ 1]\ \text{m}$
- Boundary conditions:
  - Supersonic inflow ($x = 0$ m)
  - Supersonic outflow ($x = 1$ m)
  - Slip wall (tangent flow)
    ($y = 0$ m, $y = 1$ m, $z = 0$ m, $z = 1$ m)
- 5 nonuniform meshes:
  $25 \times 25 \times 25 \rightarrow 400 \times 400 \times 400$
- Solution consists of small, smooth perturbations to uniform flow:

$$\rho(x,y,z) = \bar{\rho}\left[1 - \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\left(\sin(\pi z) + \cos(\pi z)\right)\right],$$
$$u(x,y,z) = \bar{u}\left[1 + \epsilon \sin\left(\tfrac{1}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\left(\sin(\pi z) + \cos(\pi z)\right)\right],$$
$$v(x,y,z) = \bar{v}\left[\ \ - \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y)\right)\left(\sin(\pi z) + \cos(\pi z)\right)\right],$$
$$w(x,y,z) = \bar{w}\left[\ \ - \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\left(\sin(\pi z)\right)\right],$$
$$T(x,y,z) = \bar{T}\left[1 + \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\left(\sin(\pi z) + \cos(\pi z)\right)\right],$$

$\bar{\rho} = 1\ \text{kg/m}^3$, $\bar{T} = 300$ K, $\bar{M} = 2.5$, $\epsilon = 0.05$



3D Supersonic Flow using a Manufactured Solution



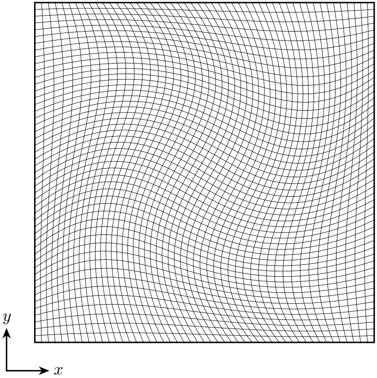
Mesh	ρ	u	v	w	T
1–2	2.0849	1.8731	1.9841	1.7039	1.9404
2–3	2.1406	1.9923	1.9295	1.8621	1.9774
3–4	2.0990	2.0115	1.9623	1.9349	1.9922
4–5	2.0585	2.0100	1.9820	1.9571	1.9964

Observed accuracy p using L^∞-norms of the error

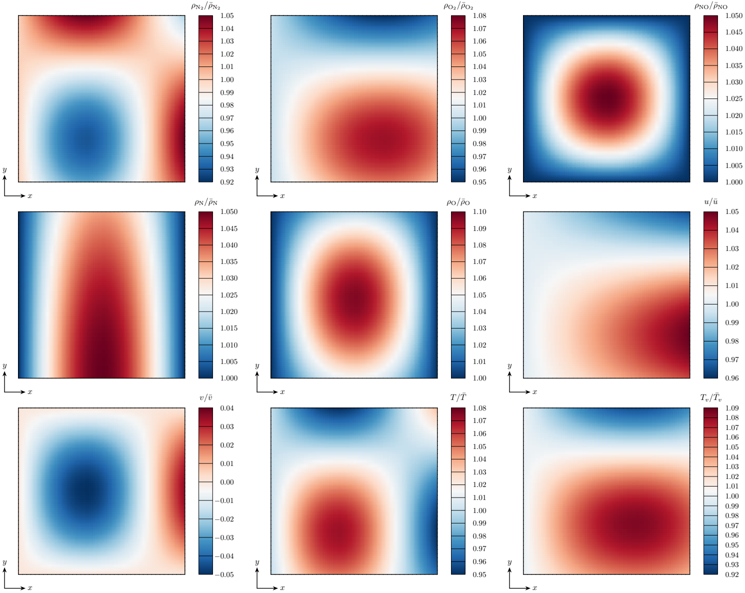
5-Species, 17-Reactions Inviscid Flow in Chemical Nonequilibrium

- Two-dimensional domain: $(x, y) \in [0,\ 1]$ m $\times$ $[0,\ 1]$ m
- Same boundary conditions
- 7 nonuniform meshes: $25 \times 25 \to 1600 \times 1600$
- Solution consists of small, smooth perturbations to uniform flow

$$
\begin{aligned}
\rho_{\mathrm{N}_2}(x,y) &= \bar\rho_{\mathrm{N}_2}\left[1 - \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\right],\\
\rho_{\mathrm{O}_2}(x,y) &= \bar\rho_{\mathrm{O}_2}\left[1 + \epsilon \sin\left(\tfrac{3}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\right],\\
\rho_{\mathrm{NO}}(x,y) &= \bar\rho_{\mathrm{NO}}\left[1 + \epsilon \sin(\pi x)\left(\sin(\pi y)\right)\right],\\
\rho_{\mathrm{N}}(x,y) &= \bar\rho_{\mathrm{N}}\left[1 + \epsilon \sin(\pi x)\left(\cos\left(\tfrac{1}{4}\pi y\right)\right)\right],\\
\rho_{\mathrm{O}}(x,y) &= \bar\rho_{\mathrm{O}}\left[1 + \epsilon \sin(\pi x)\left(\sin(\pi y) + \cos\left(\tfrac{1}{4}\pi y\right)\right)\right],\\
u(x,y) &= \bar u\left[1 + \epsilon \sin\left(\tfrac{1}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\right],\\
v(x,y) &= \bar v\left[- \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y)\right)\right],\\
T(x,y) &= \bar T\left[1 + \epsilon \sin\left(\tfrac{5}{4}\pi x\right)\left(\sin(\pi y) + \cos(\pi y)\right)\right],\\
T_v(x,y) &= \bar T_v\left[1 + \epsilon \sin\left(\tfrac{3}{4}\pi x\right)\left(\sin\left(\tfrac{5}{4}\pi y\right) + \cos\left(\tfrac{3}{4}\pi y\right)\right)\right]
\end{aligned}
$$

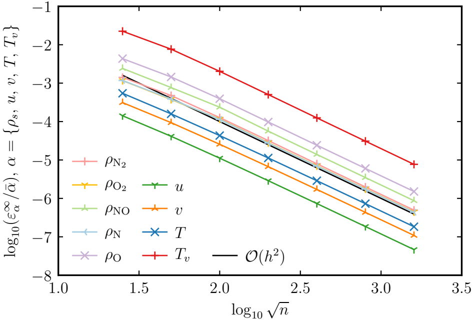


Five-Species Inviscid Flow in Chemical Nonequilibrium



2D Hypersonic Flow in Thermal Nonequilibrium using a Manufactured Solution

Variable	Value	Units
$\bar{\rho}_{\mathrm{N_2}}$	0.0077	kg/m^3
$\bar{\rho}_{\mathrm{O_2}}$	0.0020	kg/m^3
$\bar{\rho}_{\mathrm{NO}}$	0.0001	kg/m^3
$\bar{\rho}_{\mathrm{N}}$	0.0001	kg/m^3
$\bar{\rho}_{\mathrm{O}}$	0.0001	kg/m^3
$\bar{T}$	5000	K
$\bar{T}_v$	1000	K
$\bar{M}$	8	
ϵ	0.05	



Mesh	$\rho_{\mathrm{N_2}}$	$\rho_{\mathrm{O_2}}$	ρ_{NO}	ρ_{N}	ρ_{O}	u	v	T	T_v
1–2	1.5659	1.6370	1.6555	1.6046	1.5869	1.7742	1.7337	1.7814	1.5545
2–3	1.9067	1.6944	1.6986	1.7598	1.8819	1.8916	1.8701	1.8768	1.9150
3–4	1.9868	2.0475	2.0698	2.0477	2.0110	1.9488	1.9357	1.9349	2.0082
4–5	2.0074	1.9941	2.0138	1.9936	2.0089	1.9752	1.9684	1.9672	2.0168
5–6	2.0062	1.9939	2.0004	1.9935	2.0061	1.9879	1.9843	1.9836	2.0111
6–7	2.0037	1.9965	1.9994	1.9962	1.9955	1.9940	1.9922	1.9918	2.0063

2D MMS, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Observed accuracy p using L^∞-norms of the error

Verification Techniques for Thermochemical Source Term

- $\mathbf{S}(\mathbf{u}) = \left[\dot{\mathbf{w}};\, \mathbf{0};\, 0;\, Q_{t-v} + \mathbf{e}_v^T \dot{\mathbf{w}}\right]$ is algebraic
  - $\mathbf{S}(\mathbf{u})$ computed by same code for both sides of $\mathbf{r}_h(\mathbf{u}_h) = \mathbf{r}(\mathbf{u}_{\text{MS}})$
  - Manufactured solutions will **not** detect implementation errors

Verification Techniques for Thermochemical Source Term

- $\mathbf{S}(\mathbf{u}) = \left[\dot{\mathbf{w}};\, \mathbf{0};\, 0;\, Q_{t-v} + \mathbf{e}_v^T \dot{\mathbf{w}}\right]$ is algebraic
  - $\mathbf{S}(\mathbf{u})$ computed by same code for both sides of $\mathbf{r}_h(\mathbf{u}_h) = \mathbf{r}(\mathbf{u}_{\text{MS}})$
  - Manufactured solutions will **not** detect implementation errors

- Compute $Q_{t-v}(\boldsymbol{\rho}, T, T_v)$, $\mathbf{e}_v(\boldsymbol{\rho}, T, T_v)$, and $\dot{\mathbf{w}}(\boldsymbol{\rho}, T, T_v)$
  - For single-cell mesh when initialized to $\{\boldsymbol{\rho},\, T,\, T_v\}$ with no velocity
  - For many values of $\{\boldsymbol{\rho},\, T,\, T_v\}$
  - Compare with independently developed code
  - Perform convergence studies on distribution and difference

Verification Techniques for Thermochemical Source Term

- $\mathbf{S}(\mathbf{u}) = \left[\dot{\mathbf{w}};\, \mathbf{0};\, 0;\, Q_{t-v} + \mathbf{e}_v^T \dot{\mathbf{w}}\right]$ is algebraic
  - $\mathbf{S}(\mathbf{u})$ computed by same code for both sides of $\mathbf{r}_h(\mathbf{u}_h) = \mathbf{r}(\mathbf{u}_{\text{MS}})$
  - Manufactured solutions will **not** detect implementation errors
- Compute $Q_{t-v}(\boldsymbol{\rho}, T, T_v)$, $\mathbf{e}_v(\boldsymbol{\rho}, T, T_v)$, and $\dot{\mathbf{w}}(\boldsymbol{\rho}, T, T_v)$
  - For single-cell mesh when initialized to $\{\boldsymbol{\rho},\, T,\, T_v\}$ with no velocity
  - For many values of $\{\boldsymbol{\rho},\, T,\, T_v\}$
  - Compare with independently developed code
  - Perform convergence studies on distribution and difference
- For each query, compute symmetric relative difference

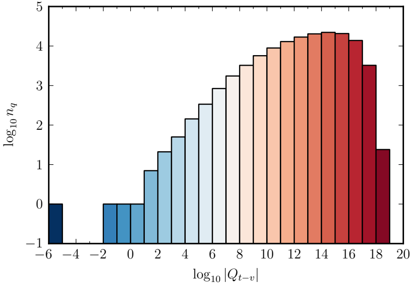
$$\delta_\beta = 2\frac{|\beta_{\texttt{SPARC}} - \beta'|}{|\beta_{\texttt{SPARC}}| + |\beta'|}$$

$$\beta = \left\{Q_{t-v},\, e_{v_{\text{N}_2}},\, e_{v_{\text{O}_2}},\, e_{v_{\text{NO}}},\, \dot{w}_{\text{N}_2},\, \dot{w}_{\text{O}_2},\, \dot{w}_{\text{NO}},\, \dot{w}_{\text{N}},\, \dot{w}_{\text{O}}\right\}$$

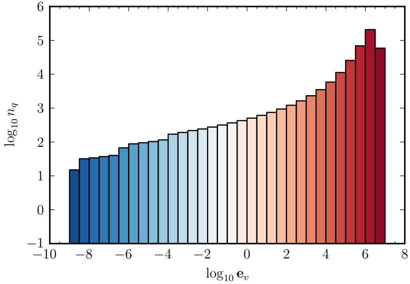
Distributions of $Q_{t-v}(\boldsymbol{\rho}, T, T_v)$, $\mathbf{e}_v(\boldsymbol{\rho}, T, T_v)$, and $\dot{\mathbf{w}}(\boldsymbol{\rho}, T, T_v)$

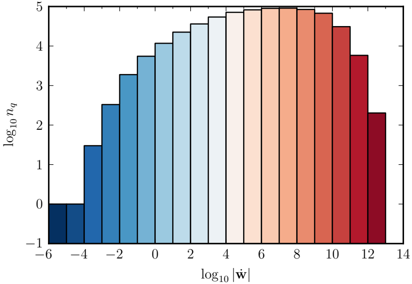
Variable	Minimum	Maximum	Units	Spacing
ρ_{N_2}	10^{-6}	10^{1}	kg/m^3	Logarithmic
ρ_{O_2}	10^{-6}	10^{1}	kg/m^3	Logarithmic
ρ_{NO}	10^{-6}	10^{1}	kg/m^3	Logarithmic
ρ_{N}	10^{-6}	10^{1}	kg/m^3	Logarithmic
ρ_{O}	10^{-6}	10^{1}	kg/m^3	Logarithmic
T	100	15,000	K	Linear
T_v	100	15,000	K	Linear

Ranges and spacings for Latin hypercube samples

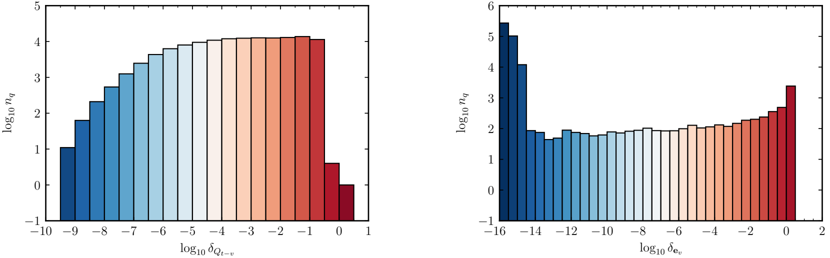


Distribution of absolute values for $n_{\mathcal{S}} = 2^{17} = 131,072$

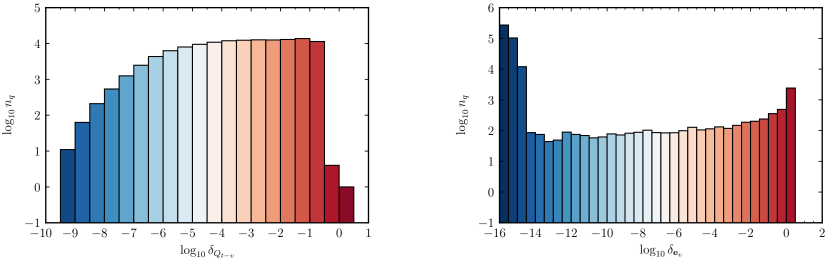




Original Nonzero Relative Differences in Q_{t-v} and $\mathbf{e}_v$

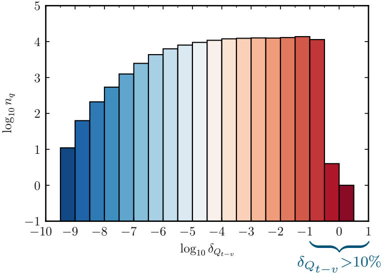


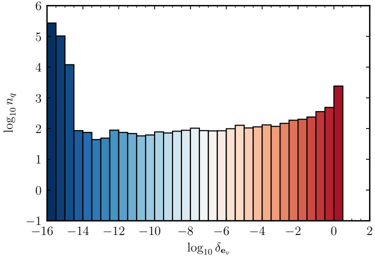
Original Nonzero Relative Differences in Q_{t-v} and $\mathbf{e}_v$



- Relative differences are not near machine precision

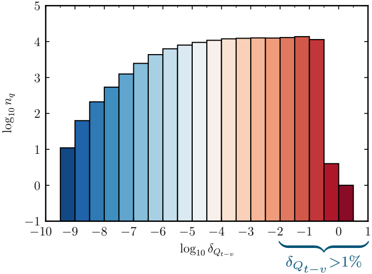
Original Nonzero Relative Differences in Q_{t-v} and $\mathbf{e}_v$

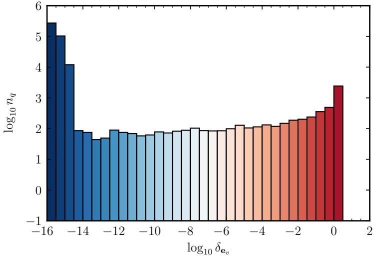




- Relative differences are not near machine precision
- $\delta_{Q_{t-v}} > 10\%$ in 8.7% of simulations

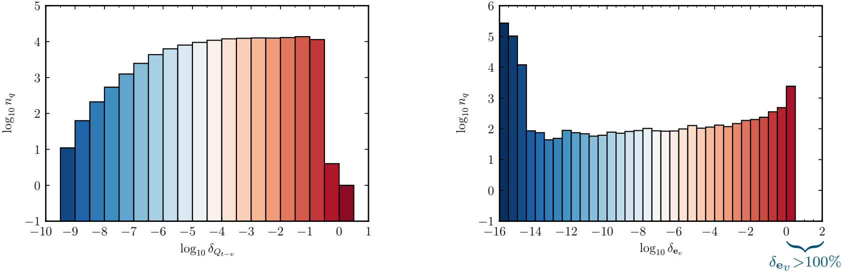
Original Nonzero Relative Differences in Q_{t-v} and $\mathbf{e}_v$





- Relative differences are not near machine precision
- $\delta_{Q_{t-v}} > 10\%$ in 8.7% of simulations
- $\delta_{Q_{t-v}} > 1\%$ in 29% of simulations

Original Nonzero Relative Differences in Q_{t-v} and $\mathbf{e}_v$



- Relative differences are not near machine precision
- $\delta_{Q_{t-v}} > 10\%$ in 8.7% of simulations
- $\delta_{Q_{t-v}} > 1\%$ in 29% of simulations
- $\delta_{\mathbf{e}_v} > 100\%$ for some simulations

Causes of Large Relative Differences in Q_{t-v} and $\mathbf{e}_v$

Two causes:

Causes of Large Relative Differences in Q_{t-v} and $\mathbf{e}_v$

Two causes:

- **Incorrect lookup table values** for vibrational constants
  - Introduced error in Q_{t-v} for all simulations

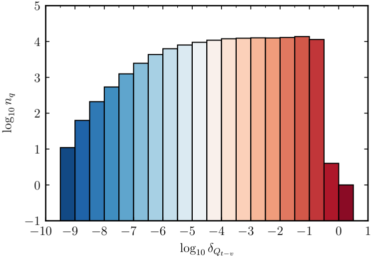
Causes of Large Relative Differences in Q_{t-v} and $\mathbf{e}_v$

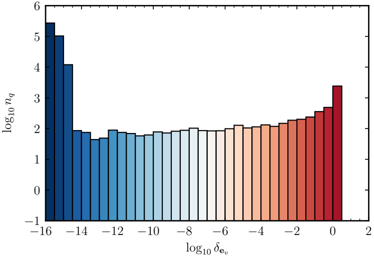
Two causes:

- **Incorrect lookup table values** for vibrational constants
  - Introduced error in Q_{t-v} for all simulations
- **Loose convergence criteria** for computing T_v from ρe_v
  - Introduced errors in Q_{t-v}, $\mathbf{e}_v$, and $\dot{\mathbf{w}}$ for low values of T_v

Corrected Nonzero Relative Differences in Q_{t-v} and $\mathbf{e}_v$

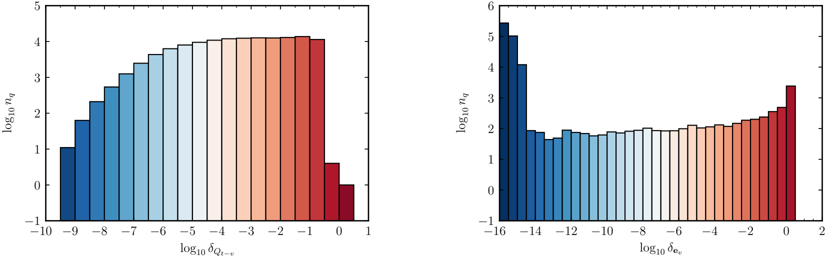
Original lookup table and convergence criteria



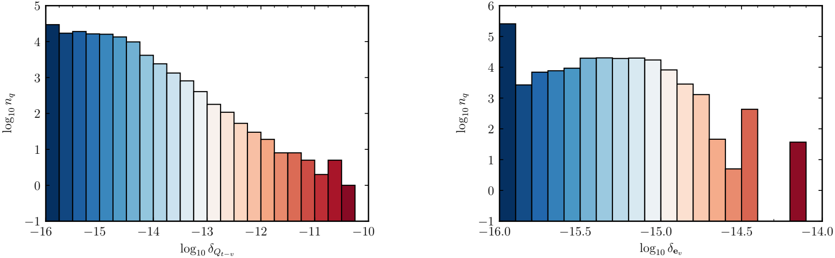


Corrected Nonzero Relative Differences in Q_{t-v} and $\mathbf{e}_v$

Original lookup table and convergence criteria



Corrected lookup table and tighter convergence criteria



Nonintrusive Manufactured Solutions for Non-Decomposing Ablation in Two Dimensions

Brian A. Freno
Brian R. Carnes
Victor E. Brunini
Neil R. Matula
Sandia National Laboratories

Governing Equations

- Ablative processes are important in many scientific & engineering problems
  - Glacial erosion, fire protection, medical procedures, and industrial manufacturing processes
  - Ablative materials used as sacrificial heat shields for weapons, rockets, and hypersonic reentry vehicles
- Temperature of ablating material modeled by heat equation

$$\rho c_p(T)\frac{\partial T}{\partial t} - \nabla \cdot (k(T)\nabla T) = 0$$

ρ: constant density, c_p: specific heat capacity, k: thermal conductivity

  - Insulated boundary conditions at non-ablating boundaries ($\partial T/\partial n = 0$)
  - Heat flux at ablating boundary due to convection, radiation, recession
- Recession rate intricately depends on temperature, pressure, space, time

Manufactured Solutions from Manufactured Parameters

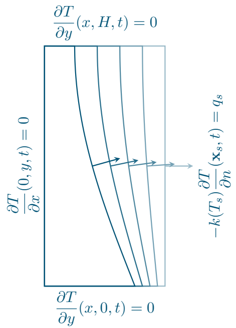
Approach: $\mathbf{r}(\mathbf{u}; \boldsymbol{\mu}_{\text{MP}}) = \mathbf{0}$

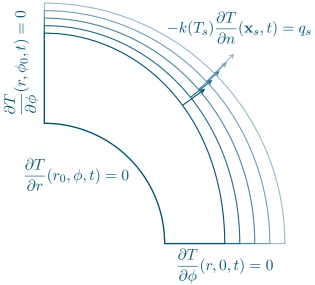
- Optionally transform the heat equation via Kirchhoff transformation
- Analytically solve equation by separating time and space dependencies
- Satisfy non-ablating boundary conditions
- Satisfy ablating boundary conditions
  - Manufacture parameters used to model recession rate
  - Manufacture functional dependencies using elementary functions
  - Parameter dependencies appear as external data (e.g., lookup tables)

Benefits:

- Similar capabilities to traditional manufactured solutions
- Only need to modify external data to obtain a nontrivial known solution
- Negligible implementation effort – no need to modify code

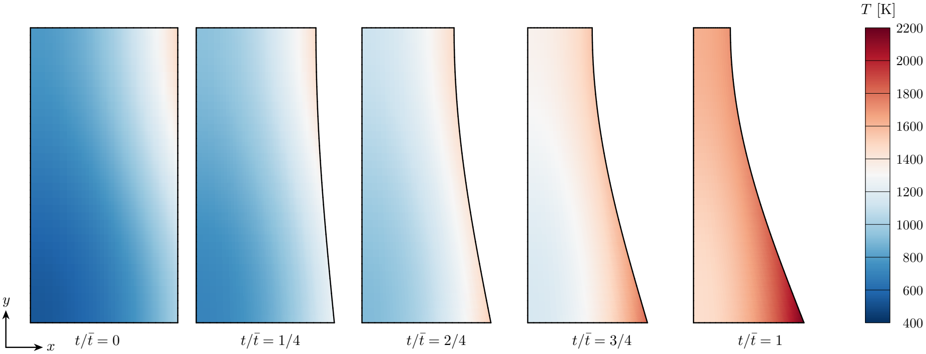
Numerical Examples



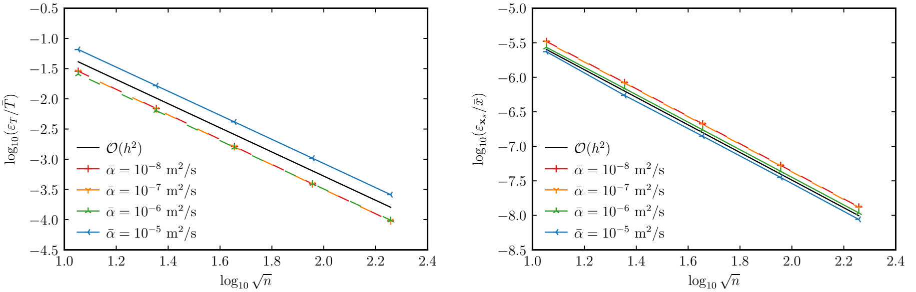


- Demonstrate methodology on two problems: Cartesian and polar
- Spatial domain discretized with $\mathcal{O}(h^2)$ finite elements
- Backward Euler time integration is $\mathcal{O}(h)$
- Each discretization doubles elements in each dimension, quarters time step
- Piecewise linear interpolation of tabulated data is $\mathcal{O}(h^2)$ – double samples

Cartesian Coordinates: Temperature and Recession ($\bar{\alpha} = 10^{-5}$ m^2/s)



Cartesian Coordinates: Error Norms

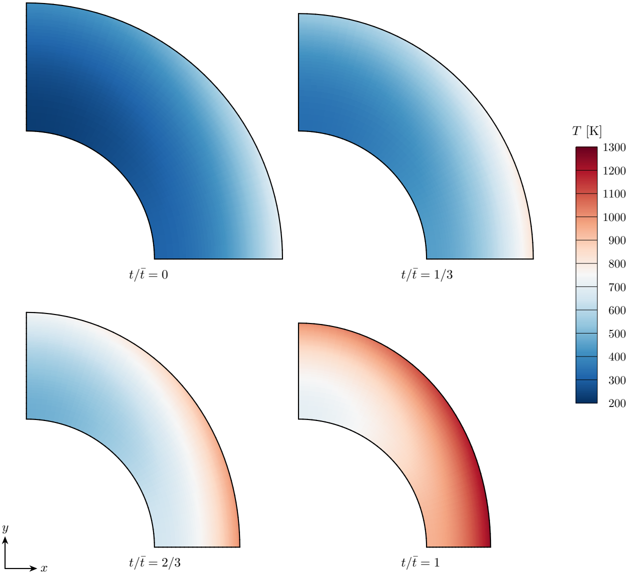


- 5 discretizations:

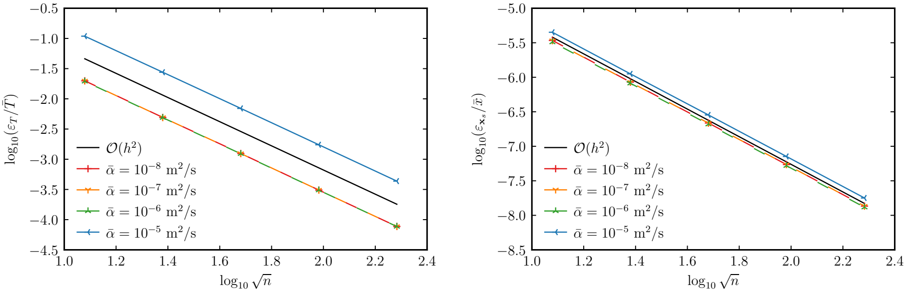
  8×16 elements, $\Delta t = 0.2$ s $\rightarrow$ 128×256 elements, $\Delta t = 0.78125$ ms

- Nondimensionalized by $T_0 = 1$ K and $x_0 = 1$ m
- n is number of elements
- Error norms in both plots are $\mathcal{O}(h^2)$, as expected

Polar Coordinates: Temperature and Recession ($\bar{\alpha} = 10^{-5}$ m^2/s)



Polar Coordinates: Error Norms



- 5 discretizations:

  8×18 elements, $\Delta t = 0.2$ s $\quad\rightarrow\quad$ 128×288 elements, $\Delta t = 0.78125$ ms
- Nondimensionalized by $T_0 = 1$ K and $x_0 = 1$ m
- n is number of elements
- Error norms in both plots are $\mathcal{O}(h^2)$, as expected

Manufactured Solutions for the Method-of-Moments Implementation of the Electromagnetic Integral Equations

Brian A. Freno
Neil R. Matula
Justin I. Owen
William A. Johnson
Sandia National Laboratories

Electromagnetic Integral Equations

- Numerically models electromagnetic scattering and radiation problems
- Relate electric surface current to incident electric and/or magnetic field
  - EFIE: compute electric surface current from incident electric field
  - MFIE: compute electric surface current from incident magnetic field
  - CFIE: linear combination of EFIE and MFIE
- Discretize surface of electromagnetic scatterer with elements
- Evaluate 4D reaction integrals over 2D test and source elements
- Contain singular integrands when test and source elements are near
- Express surface current in terms of vector-valued basis functions

Electromagnetic Potentials

In time-harmonic form, $\mathbf{E}^{\mathcal{S}}$ and/or $\mathbf{H}^{\mathcal{S}}$ computed from $\mathbf{J}$ and $\mathbf{M}$

Scattered electric field $$\mathbf{E}^{\mathcal{S}}(\mathbf{x}) = -\left(j\omega\mathbf{A}(\mathbf{x}) + \nabla\Phi(\mathbf{x}) + \frac{1}{\epsilon}\nabla\times\mathbf{F}(\mathbf{x})\right)$$

Scattered magnetic field $$\mathbf{H}^{\mathcal{S}}(\mathbf{x}) = \frac{1}{\mu}\nabla\times\mathbf{A}(\mathbf{x})$$

Magnetic vector potential $$\mathbf{A}(\mathbf{x}) = \mu\int_{S'}\mathbf{J}(\mathbf{x}')G(\mathbf{x},\mathbf{x}')dS'$$

Electric scalar potential $$\Phi(\mathbf{x}) = \frac{j}{\epsilon\omega}\int_{S'}\nabla'\cdot\mathbf{J}(\mathbf{x}')G(\mathbf{x},\mathbf{x}')dS'$$

Electric vector potential $$\mathbf{F}(\mathbf{x}) = \epsilon\int_{S'}\mathbf{M}(\mathbf{x}')G(\mathbf{x},\mathbf{x}')dS'$$

Green's function $$G(\mathbf{x},\mathbf{x}') = \frac{e^{-jkR}}{4\pi R}, \qquad R = |\mathbf{x}-\mathbf{x}'|$$

Singularity when $R \to 0$

$\mathbf{J}$ and $\mathbf{M}$ are electric and magnetic surface current densities

$S' = S$ is surface of scatterer, $k = \omega\sqrt{\mu\epsilon}$ is wavenumber, μ and ϵ are permeability and permittivity

Electromagnetic Integral Equations

Compute $\mathbf{J}$ from

- incident electric field $\mathbf{E}^{\mathcal{I}}$ $\quad \big(\mathbf{n} \times (\mathbf{E}^{\mathcal{S}} + \mathbf{E}^{\mathcal{I}}) = \mathbf{0}\big)$ (EFIE)
- incident magnetic field $\mathbf{H}^{\mathcal{I}}$ $\quad \big(\mathbf{n} \times (\mathbf{H}^{\mathcal{S}} + \mathbf{H}^{\mathcal{I}}) = \mathbf{J}\big)$ (MFIE)

Discretize surface with triangles, approximate $\mathbf{J}$ with RWG basis functions:

$$\mathbf{J}_h(\mathbf{x}) = \sum_{j=1}^{n_b} J_j \boldsymbol{\Lambda}_j(\mathbf{x})$$

Project equation onto basis functions: $a(\mathbf{J}_h, \boldsymbol{\Lambda}_i) = b\big(\mathbf{E}^{\mathcal{I}}, \mathbf{H}^{\mathcal{I}}, \boldsymbol{\Lambda}_i\big)$

In matrix–vector form, solve for $\mathbf{J}^h$:

$$\mathbf{Z}\mathbf{J}^h = \mathbf{V}$$

$Z_{i,j} = a(\boldsymbol{\Lambda}_j, \boldsymbol{\Lambda}_i)$, Impedance matrix $\quad J_j^h = J_j$, Current vector $\quad V_i = b\big(\mathbf{E}^{\mathcal{I}}, \mathbf{H}^{\mathcal{I}}, \boldsymbol{\Lambda}_i\big)$ Excitation vector

Error Sources in the Electromagnetic Integral Equations

Isolate and measure 3 sources of numerical error:

- **Domain discretization**: Representation of curved surfaces with planar elements
  - Second-order error for curved surfaces, no error for planar surfaces
  - Error reduced with curved elements
- **Solution discretization**: Representation of solution or operators
  - Common in solution to differential, integral, and integro-differential equations
  - Finite number of basis functions to approximate solution
  - Finite samples queried to approximate underlying equation operators
- **Numerical integration**: Quadrature
  - Analytical integration is usually not possible
  - For well-behaved integrands,
    - Expect integration error at least same order as solution-discretization error
    - Less rigorously, error should decrease with more quadrature points
  - For (nearly) singular integrands, monotonic convergence is not assured

Manufactured Surface Current

Continuous equations: $r_i(\mathbf{J}\ \ ) = a(\mathbf{J}\ \ , \boldsymbol{\Lambda}_i) - b\big(\mathbf{E}^{\mathcal{I}}, \mathbf{H}^{\mathcal{I}}, \boldsymbol{\Lambda}_i\big) = 0$

Discretized equations: $r_i(\mathbf{J}_h) = a(\mathbf{J}_h, \boldsymbol{\Lambda}_i) - b\big(\mathbf{E}^{\mathcal{I}}, \mathbf{H}^{\mathcal{I}}, \boldsymbol{\Lambda}_i\big) = 0$

Method of manufactured solutions modifies discretized equations:

$$\mathbf{r}(\mathbf{J}_h) = \mathbf{r}(\mathbf{J}_{\text{MS}}),$$

where $\mathbf{J}_{\text{MS}}$ is manufactured solution and $\mathbf{r}(\mathbf{J}_{\text{MS}})$ is computed exactly

Modified discretized equations: $a(\mathbf{J}_h, \boldsymbol{\Lambda}_i) = a(\mathbf{J}_{\text{MS}}, \boldsymbol{\Lambda}_i)$

Can be implemented via $\mathbf{E}^{\mathcal{I}}$ and $\mathbf{H}^{\mathcal{I}}$ if $b\big(\mathbf{E}^{\mathcal{I}}, \mathbf{H}^{\mathcal{I}}, \boldsymbol{\Lambda}_i\big) = a(\mathbf{J}_{\text{MS}}, \boldsymbol{\Lambda}_i) = V_i$:

$$\mathbf{E}^{\mathcal{I}} = \frac{j}{\omega\epsilon}\int_{S'}\left[k^2\mathbf{J}_{\text{MS}}(\mathbf{x}')G(\mathbf{x},\mathbf{x}') - \nabla'\cdot\mathbf{J}_{\text{MS}}(\mathbf{x}')\nabla' G(\mathbf{x},\mathbf{x}')\right]dS'$$

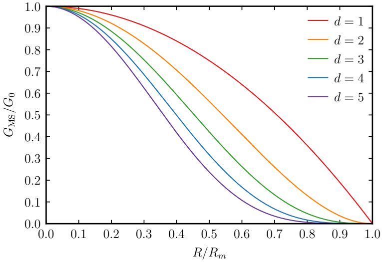
$$\mathbf{H}^{\mathcal{I}} = \frac{1}{2}\mathbf{J}_{\text{MS}}\times\mathbf{n} - \int_{S'}\left[\mathbf{J}_{\text{MS}}(\mathbf{x}')\times\nabla' G(\mathbf{x},\mathbf{x}')\right]dS'$$

Manufactured Green's Function

Integrals with G cannot be computed analytically or, when $R \to 0$, accurately

Inaccurately computing integrals on either side contaminates convergence studies

Manufacture Green's function: $G_{\text{MS}}(R) = G_0\left(1 - \frac{R^2}{R_m^2}\right)^d$, $R_m = \max\limits_{\mathbf{x},\mathbf{x}' \in S} R$ and $d \in \mathbb{N}$



Reasoning:

1) Even powers of R permit integrals to be computed analytically for many $\mathbf{J}_{\text{MS}}$
2) G_{MS} increases when R decreases, as with actual G

Error Metrics

Solution-Discretization Error

- Error due to basis-function approximation of solution: $\mathbf{J}_h(\mathbf{x}) = \sum_{j=1}^{n_b} J_j \mathbf{\Lambda}_j(\mathbf{x})$
- Error metric: $\mathbf{e}_\mathbf{J} = \mathbf{J}^h - \mathbf{J}_n$
- Avoid numerical-integration error → integrate exactly (G_{MS})

Numerical-Integration Error

- Error due to quadrature integral evaluation $(\cdot)^q$ on both sides of equations
- Error metrics: $e_a = \mathbf{J}_n^H(\mathbf{Z}^q - \mathbf{Z})\mathbf{J}_n$, $\quad e_b = \mathbf{J}_n^H(\mathbf{V}^q - \mathbf{V})$
- Cancel solution-discretization error using basis functions

Solution-Discretization Error: Solution Uniqueness

For terms with G_{MS}, $\mathbf{Z}$ is practically singular $\rightarrow$ infinite solutions for $\mathbf{J}^h$

Choose $\mathbf{J}^h$ closest to $\mathbf{J}_n$ (J_{n_j}: $\mathbf{J}_{\text{MS}}$ from $T_j^+ \rightarrow T_j^-$) that satisfies $\mathbf{Z}\mathbf{J}^h = \mathbf{V}$

Compute pivoted QR factorization of $\mathbf{Z}^H$ to determine rank

Express $\mathbf{J}^h$ in terms of basis $\mathbf{Q}$:

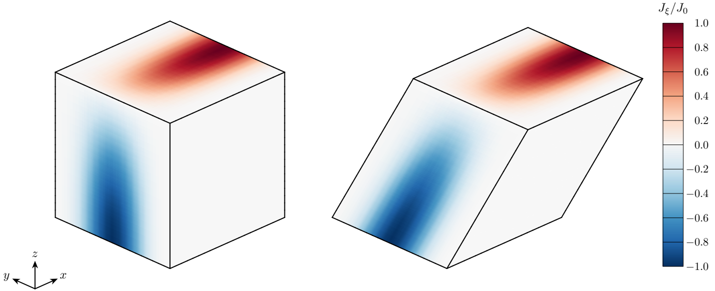
$$\mathbf{J}^h = \mathbf{Q}_1 \mathbf{u} + \mathbf{Q}_2 \mathbf{v}$$

$\mathbf{u}$: coefficients that satisfy $\mathbf{Z}\mathbf{J}^h = \mathbf{V}$
$\mathbf{v}$: coefficients that bring $\mathbf{J}^h$ closest to $\mathbf{J}_n$, given $\mathbf{u}$

Compute $\mathbf{v}$ by minimizing

- $\|\mathbf{e}_\mathbf{J}\|_2$: closed-form solution
  may require finer meshes when measuring $\|\mathbf{e}_\mathbf{J}\|_\infty$
- $\|\mathbf{e}_\mathbf{J}\|_\infty$: more expensive (linear programming)
  does not require finer meshes when measuring $\|\mathbf{e}_\mathbf{J}\|_\infty$

Manufactured Surface Current $\mathbf{J}_{\mathrm{MS}}$ for Cube and Rhombic Prism



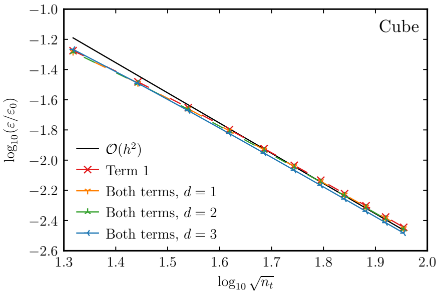
Manufactured surface current $J_\xi(\xi,\eta) = J_0 \begin{cases} \sin\left(\dfrac{\pi\xi}{2L}\right)\sin^3\left(\dfrac{\pi\eta}{L}\right), & \text{for } \mathbf{n}\cdot\mathbf{e}_y = 0 \\ 0, & \text{for } \mathbf{n}\cdot\mathbf{e}_y \neq 0 \end{cases}$

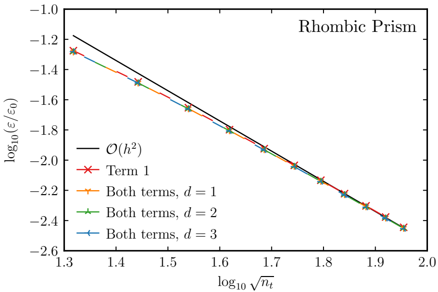
For $\mathbf{J}_{\mathrm{MS}}(\mathbf{x}) = J_\xi(\xi,\eta)\mathbf{e}_\xi$, with $J_0 = 1$ A/m and $L = 1$ m

Surface-fixed coordinate system:

- $\eta = y \in [0,\, 1]$ m
- $\xi \in [0,\, 4]$ m is perpendicular to η, wraps around surfaces for which $\mathbf{n}\cdot\mathbf{e}_y = 0$
- ξ begins at $x = 0$ m and $z = 1$ m for cube and $x = z = \sqrt{2}/2$ m for rhombic prism

Solution-Discretization Error: $\varepsilon = \|\mathbf{e_J}\|_\infty$, Term 1 and Both Terms



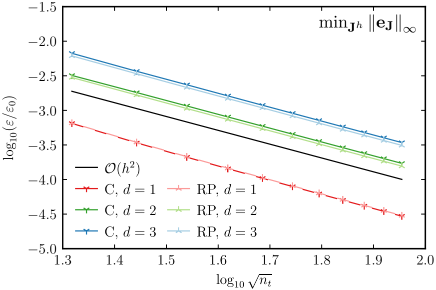


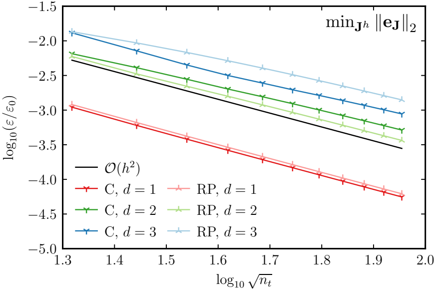
- Consider MFIE terms together and separately:

$$a(\mathbf{J}_h, \mathbf{\Lambda}_i) = \underbrace{\frac{1}{2}\int_S \mathbf{\Lambda}_i(\mathbf{x})\cdot\mathbf{J}_h(\mathbf{x})dS}_{\text{Term 1}} + \underbrace{\int_S \mathbf{\Lambda}_i(\mathbf{x})\cdot\left(\mathbf{n}(\mathbf{x})\times\int_{S'}\left[\nabla' G_{\text{MS}}(\mathbf{x},\mathbf{x}')\times\mathbf{J}_h(\mathbf{x}')\right]dS'\right)dS}_{\text{Term 2}}$$

- Consider different values of d in $G_{\text{MS}}(\mathbf{x},\mathbf{x}') = G_0\left(1-\frac{R^2}{R_m^2}\right)^d$
- For Term 1 and both terms, matrix is nonsingular – optimization not needed
- Errors converge at expected rates

Solution-Discretization Error: $\varepsilon = \|\mathbf{e_J}\|_\infty$, Term 2



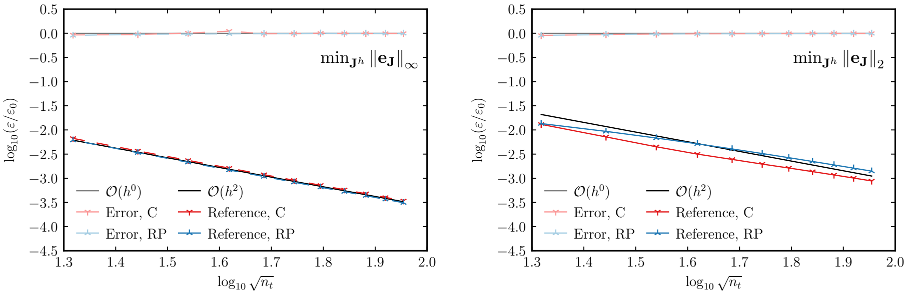


- Singular matrix requires optimization

	$\min_{\mathbf{J}^h} \|\mathbf{e_J}\|_\infty$		$\min_{\mathbf{J}^h} \|\mathbf{e_J}\|_2$	
Mesh	C	RP	C	RP
1–2	2.0800	2.0653	2.0811	1.2935
2–3	2.0141	2.0529	2.1055	1.4193
3–4	2.0303	2.0193	1.9159	1.5150
4–5	2.0196	2.0163	1.6421	1.5847
5–6	2.0061	2.0242	1.6677	1.6372
6–7	2.0133	2.0158	1.5800	1.6779
7–8	2.0113	2.0167	1.6282	1.7104
8–9	2.0037	2.0122	1.6664	1.7369
9–10	2.0086	2.0117	1.6974	1.7589
10–11	2.0053	2.0118	1.7231	1.7776

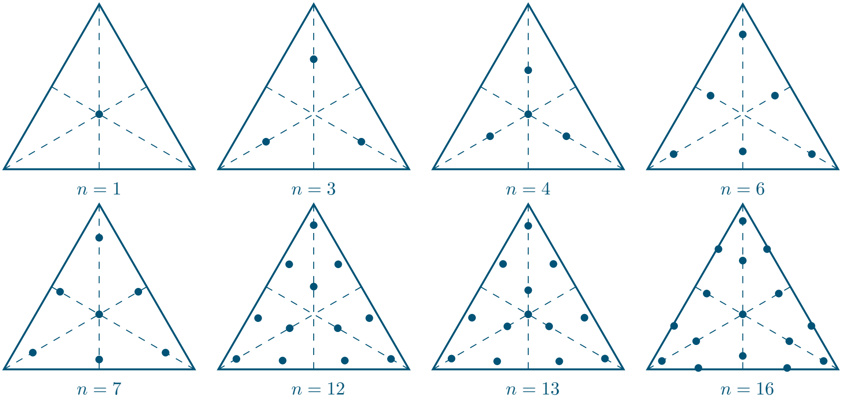
- $\|\mathbf{e_J}\|_2$ requires finer meshes to observe $\mathcal{O}(h^2)$
- $\|\mathbf{e_J}\|_\infty$ does not require finer meshes

Solution-Discretization Error: $\varepsilon = \|\mathbf{e_J}\|_\infty$ for Coding Error ($d = 3$)



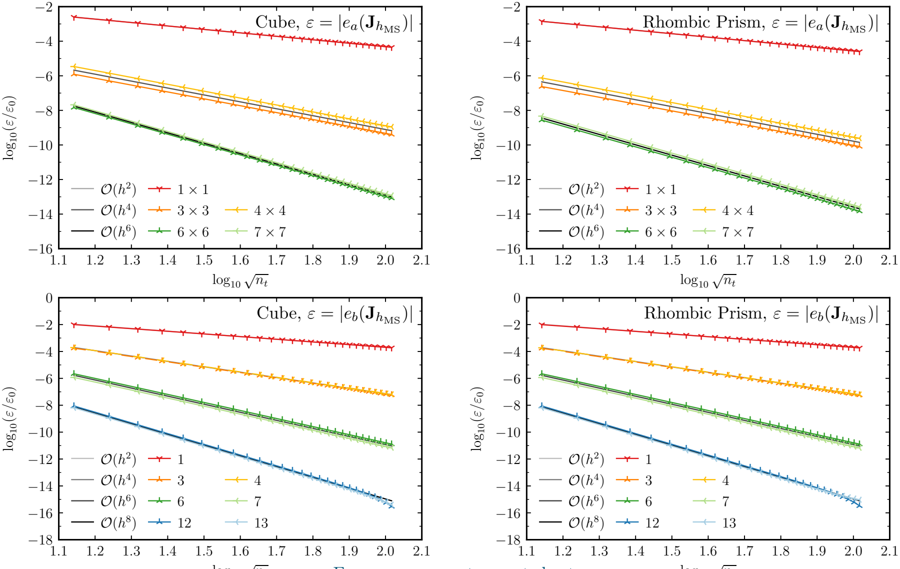
- Will selecting an optimal solution reveal a coding error?
- Manufacture a coding error: increase magnitude of diagonal elements of $\mathbf{Z}$ by 1%
- **Both** optimization norms **reveal** the $\mathcal{O}(1)$ **coding error**

Numerical-Integration Error: Polynomial Quadrature Rules



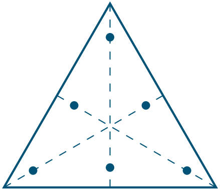
n	1	3	4	6	7	12	13	16
Max. integrand degree	1	2	3	4	5	6	7	8
Convergence rate	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^8)$	$\mathcal{O}(h^8)$	$\mathcal{O}(h^{10})$

Numerical-Integration Error ($d = 3$)

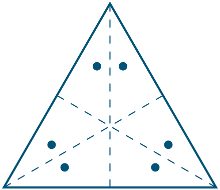


Errors converge at expected rates

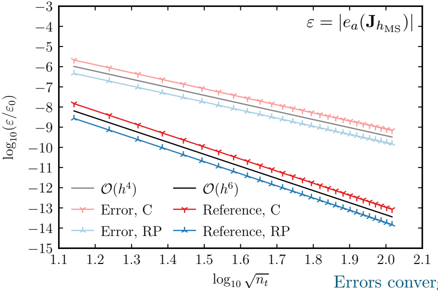
Numerical-Integration Error: Coding Error ($d = 3$)

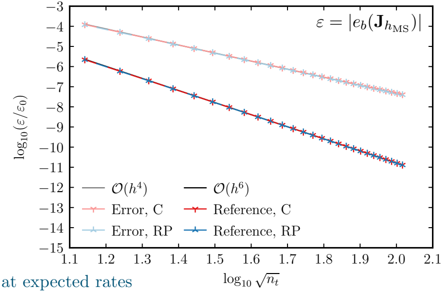


Optimal 6-point rule
Maximum polynomial degree: 4



Suboptimal 6-point rule
Maximum polynomial degree: 3





Errors converge at expected rates

Symmetric Triangle Quadrature Rules for Arbitrary Functions

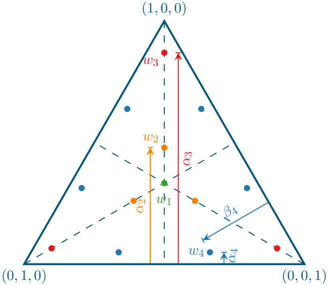
Brian A. Freno
William A. Johnson
Brian F. Zinser
Salvatore Campione
Sandia National Laboratories

Quadrature

- Quadrature rules for polynomials unreliably integrate singular integrands
- Developed 2 approaches to compute rules for arbitrary function sequences

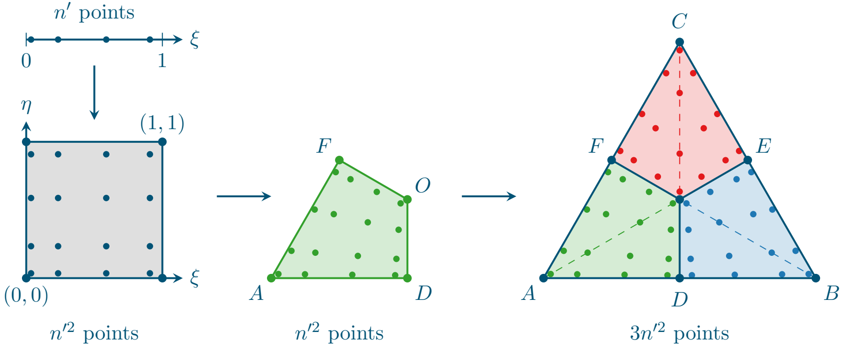
Approach 1: Optimization for Moderate Number of Functions

- Goal is to efficiently integrate polynomials and singularities
- Compute points & weights through optimization – nonlinear least squares
- This approach uses polynomial rules as a baseline
- Replace higher polynomial degrees with singular functions



Approach 2: Quadrilateral Subdomains

- In multiple dimensions, number of integrable functions not straightforward
- Computation is expensive and multiple solutions exist
- For large n_f, we employ n'-point 1D rules that integrate 1D function sequences, such that $n = 3n'^2$

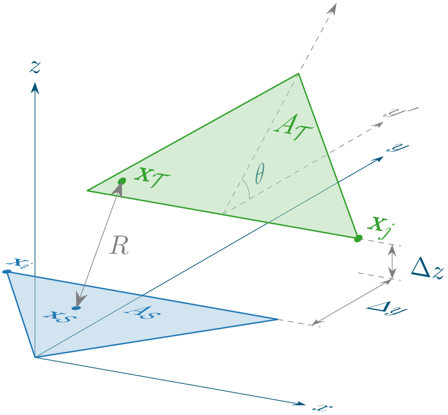


Numerical Examples

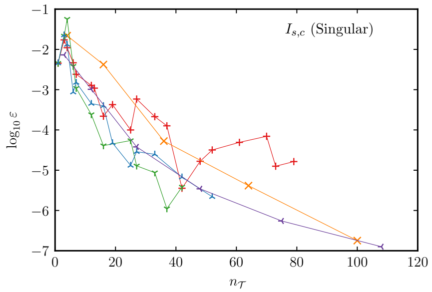
Electric scalar potential ($k = 2\pi$):

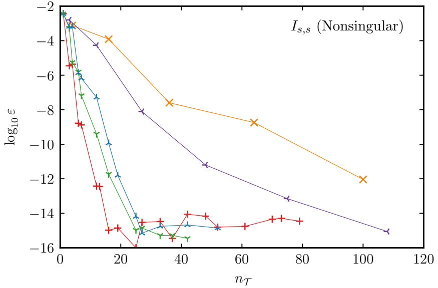
$$I_{s,c} = \int_{A_\mathcal{T}} \int_{A_\mathcal{S}} \frac{\cos(2\pi R)}{R} dA_\mathcal{S} dA_\mathcal{T}$$

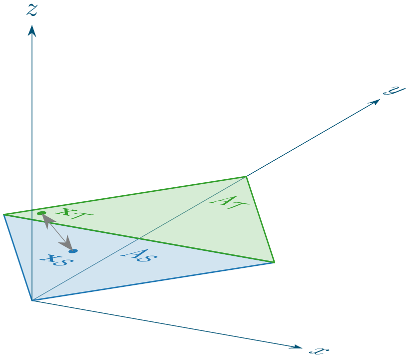
$$I_{s,s} = \int_{A_\mathcal{T}} \int_{A_\mathcal{S}} \frac{\sin(2\pi R)}{R} dA_\mathcal{S} dA_\mathcal{T}$$



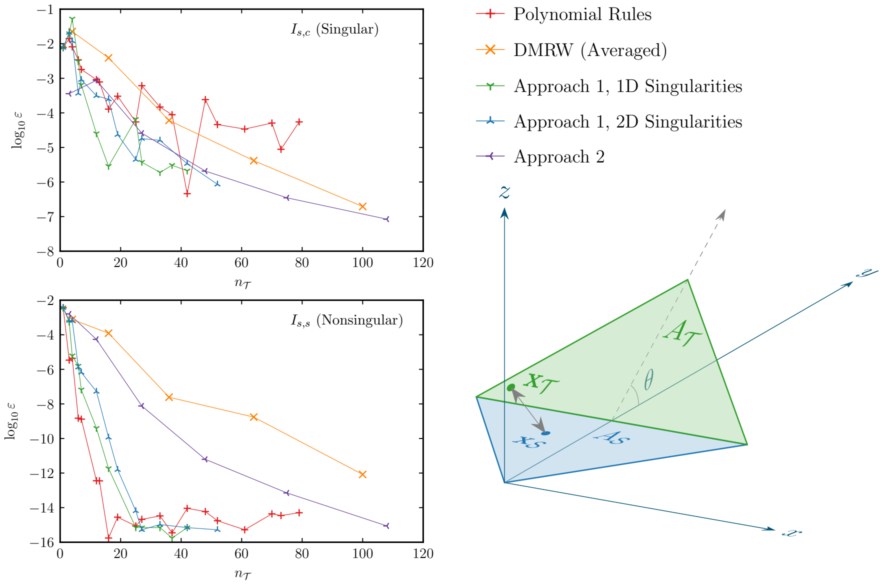
Case 1: Scalar potential, singular interaction, $\theta = 0°$, $\Delta y = 0$, and $\Delta z = 0$



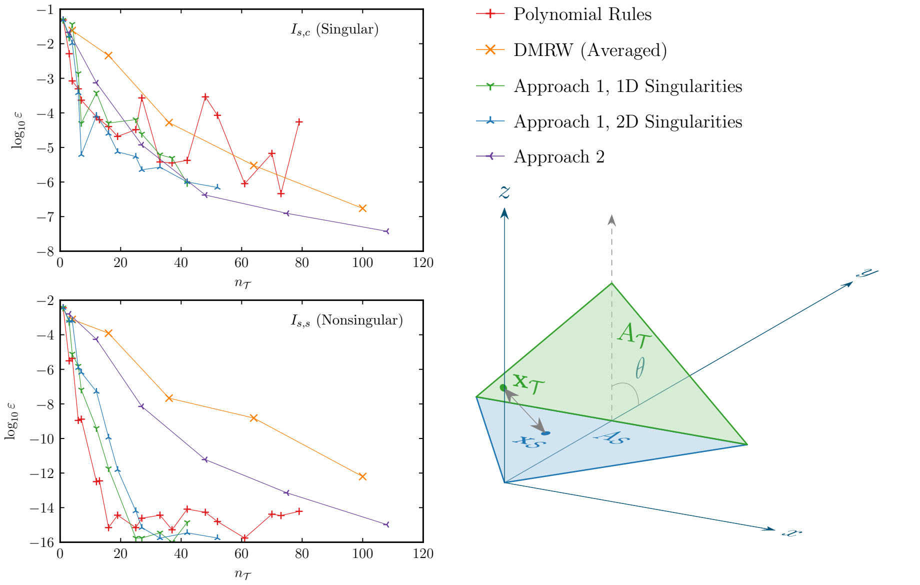




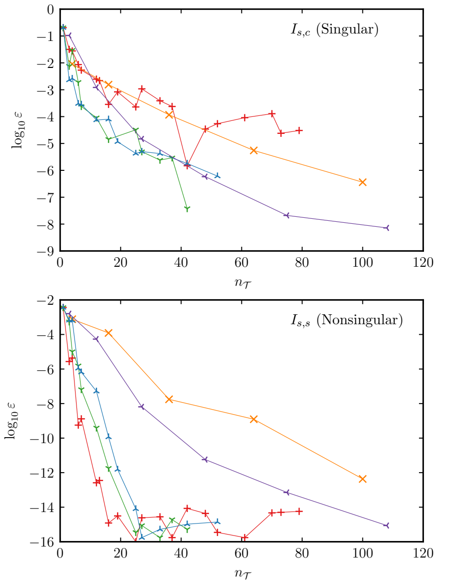
Case 2: Scalar potential, singular interaction, $\theta = 45°$, $\Delta y = 0$, and $\Delta z = 0$

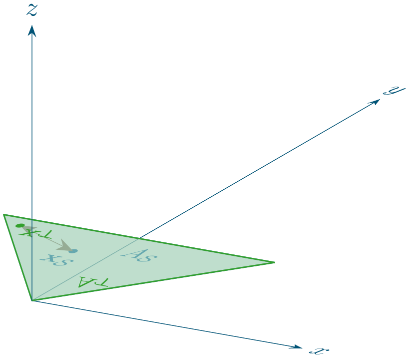


Case 3: Scalar potential, singular interaction, $\theta = 90°$, $\Delta y = 0$, and $\Delta z = 0$

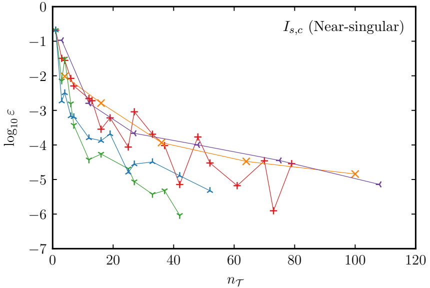


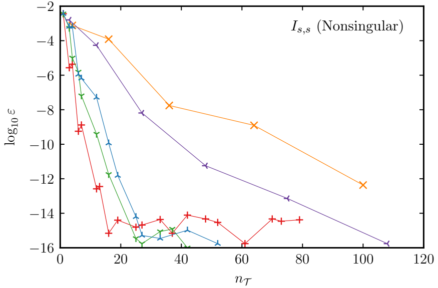
Case 4: Scalar potential, singular interaction, $\theta = 180°$, $\Delta y = 0$, and $\Delta z = 0$

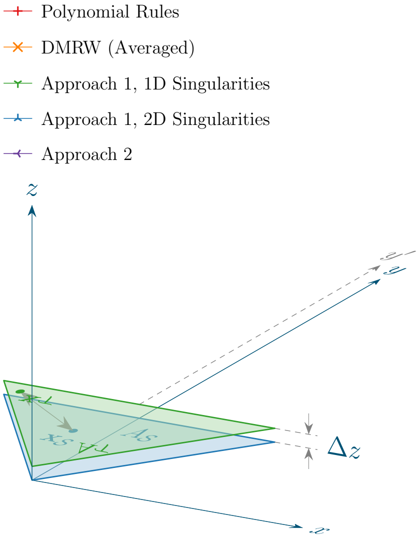




Case 5: Scalar potential, near-singular interaction, $\theta = 180°$, $\Delta y = 0$, and $\Delta z = \delta_z$

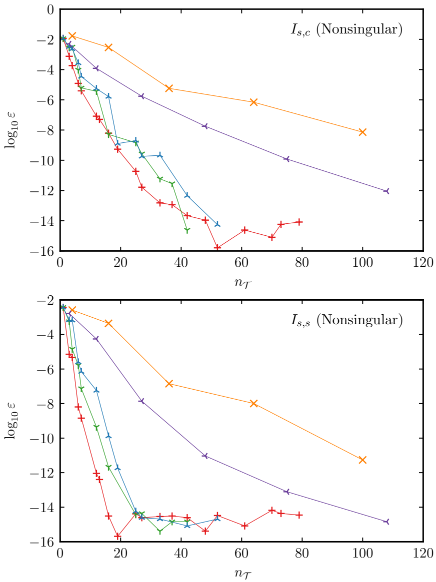




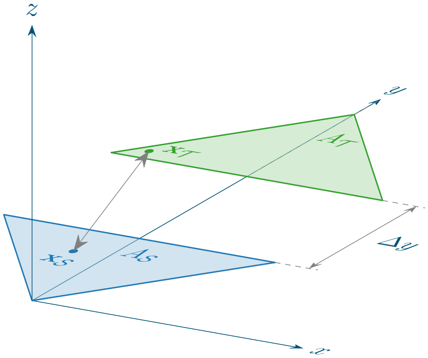


$\delta_z = 1/200$ of maximum edge length

Case 6: Scalar potential, far interaction, $\theta = 0°$, $\Delta y = \delta_y$, and $\Delta z = 0$



- Polynomial Rules
- DMRW (Averaged)
- Approach 1, 1D Singularities
- Approach 1, 2D Singularities
- Approach 2



$\delta_y \approx 1.25 \times$ (maximum edge length)

Summary

- Code verification plays important role in establishing simulation credibility
- Manufactured solutions are effective for verifying discretizations
  - Exercise features of interest
  - Effectively identify issues
- Code verification is less straightforward for integral equations
  - Requires some creativity

Additional Information

- B. Freno, K. Carlberg Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations
  Computer Methods in Applied Mechanics and Engineering (2019) `arXiv:1808.02097`
- B. Freno, B. Carnes, V. Weirs Code-Verification techniques for hypersonic reacting flows in thermochemical nonequilibrium
  Journal of Computational Physics (2021) `arXiv:2007.14376`
- B. Freno, B. Carnes, N. Matula Nonintrusive manufactured solutions for ablation
  Physics of Fluids (2021)
- B. Freno, B. Carnes, V. Brunini, N. Matula Nonintrusive manufactured solutions for non-decomposing ablation in two dimensions
  Journal of Computational Physics (2022) `arXiv:2110.13818`
- B. Freno, N. Matula, W. Johnson Manufactured solutions for the method-of-moments implementation of the EFIE
  Journal of Computational Physics (2021) `arXiv:2012.08681`
- B. Freno, N. Matula, J. Owen, W. Johnson Code-verification techniques for the method-of-moments implementation of the EFIE
  Journal of Computational Physics (2022) `arXiv:2106.13398`
- B. Freno, N. Matula Code verification for practically singular equations
  Journal of Computational Physics (2022) `arXiv:2204.01785`
- B. Freno, N. Matula Code-verification techniques for the method-of-moments implementation of the MFIE
  Journal of Computational Physics (2023) `arXiv:2209.09378`
- B. Freno, N. Matula Code-verification techniques for the method-of-moments implementation of the CFIE
  Journal of Computational Physics (2023) `arXiv:2302.06728`
- B. Freno, N. Matula, R. Pfeiffer, E. Dohme, J. Kotulski Manufactured solutions for an electromagnetic slot model
  Journal of Computational Physics (2024) `arXiv:2406.14573`
- B. Freno, W. Johnson, B. Zinser, S. Campione Symmetric triangle quadrature rules for arbitrary functions
  Computers & Mathematics with Applications (2020) `arXiv:1909.01480`
- B. Freno, W. Johnson, B. Zinser, D. Wilton, F. Vipiana, S. Campione
  Characterization and integration of the singular test integrals in the method-of-moments implementation of the EFIE
  Engineering Analysis with Boundary Elements (2021) `arXiv:1911.02107`