Code-Verification Techniques for the Method-of-Moments Implementation of the Magnetic-Field Integral Equation

Brian A. Freno
Neil R. Matula
Sandia National Laboratories

ASME Verification, Validation, and Uncertainty Quantification Symposium
May 17–19, 2023

Outline

- Introduction
- The Method-of-Moments Implementation of the MFIE
- Code-Verification Approaches
- Numerical Examples
- Summary

Outline

- Introduction
  - Electromagnetic Integral Equations
  - Verification and Validation
  - Error Sources
  - This Work
- The Method-of-Moments Implementation of the MFIE
- Code-Verification Approaches
- Numerical Examples
- Summary

Electromagnetic Integral Equations

- Are commonly used to model electromagnetic scattering and radiation
- Relate electric surface current to incident electric and/or magnetic field
- 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

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* verifies correctness of numerical-method implementation

Code Verification

- Code verification most rigorously assesses rate at which error decreases
- Error requires exact solution – usually unavailable
- Manufactured solutions are popular alternative
  - Manufacture an arbitrary solution
  - Insert manufactured solution into governing equations to get residual term
  - Add residual term to equations to coerce solution to manufactured solution
- For integral equations, few instances of code verification exist
- Analytical differentiation is straightforward – analytical integration is not
- Therefore, manufactured source term may have its own numerical error

Error Sources in the Electromagnetic Integral Equations

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 not always 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

This Work

Isolate solution-discretization error

- Eliminate integration error by manufacturing solution and Green's function
- Select unique solution through optimization when equations are singular

Isolate numerical-integration error

- Cancel solution-discretization error using basis functions
- Eliminate solution-discretization error by avoiding basis functions

Address domain-discretization error

- Account for curvature – integrate over curved triangular elements
- Neglect curvature – integrate over planar triangular elements

Outline

- Introduction
- The Method-of-Moments Implementation of the MFIE
  - The Magnetic-Field Integral Equation
  - Variational Formulation
  - Discretization
- Code-Verification Approaches
- Numerical Examples
- Summary

The Magnetic-Field Integral Equation

In time-harmonic form, scattered magnetic field $\mathbf{H}^{\mathcal{S}}$ computed from current:

$$\mathbf{H}^{\mathcal{S}} = \frac{1}{\mu}\nabla \times \mathbf{A}$$

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

$\mathbf{J}$ is electric surface current density
$S' = S$ is surface of scatterer
μ and ϵ are permeability and permittivity of surrounding medium
G is the Green's function

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

where $R = |\mathbf{x} - \mathbf{x}'|$ and $k = \omega\sqrt{\mu\epsilon}$ is wave number

Singularity when $R \to 0$

The Magnetic-Field Integral Equation (continued)

Total magnetic field $\mathbf{H} = \mathbf{H}^{\mathcal{I}} + \mathbf{H}^{\mathcal{S}}$

Incident magnetic field $\mathbf{H}^{\mathcal{I}}$ induces surface current $\mathbf{J}$

Scattered magnetic field $\mathbf{H}^{\mathcal{S}}(\mathbf{x}) = \int_{S'} [\mathbf{J}(\mathbf{x}') \times \nabla' G(\mathbf{x}, \mathbf{x}')] dS'$

On surface S, $\mathbf{n} \times \mathbf{H} = \mathbf{J}$

Through principal value integration, compute $\mathbf{J}$ from $\mathbf{H}^{\mathcal{I}}$:

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

Variational Formulation

Project $\frac{1}{2}\mathbf{J} - \mathbf{n} \times \int_{S'} [\mathbf{J}(\mathbf{x}') \times \nabla' G(\mathbf{x}, \mathbf{x}')] dS' = \mathbf{n} \times \mathbf{H}^{\mathcal{I}}$ onto space $\mathbb{V}$

Space $\mathbb{V}$ contains vector fields tangent to S

Find $\mathbf{J} \in \mathbb{V}$, such that

$$a(\mathbf{J}, \mathbf{v}) = b\big(\mathbf{H}^{\mathcal{I}}, \mathbf{v}\big) \quad \forall \mathbf{v} \in \mathbb{V},$$

where

$$a(\mathbf{u}, \mathbf{v}) = \frac{1}{2}\int_S \bar{\mathbf{v}}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) dS - \int_S \bar{\mathbf{v}}(\mathbf{x}) \cdot \left(\mathbf{n}(\mathbf{x}) \times \int_{S'} [\mathbf{u}(\mathbf{x}') \times \nabla' G(\mathbf{x}, \mathbf{x}')] dS'\right) dS,$$
$$b(\mathbf{u}, \mathbf{v}) = \int_S \bar{\mathbf{v}}(\mathbf{x}) \cdot [\mathbf{n}(\mathbf{x}) \times \mathbf{u}(\mathbf{x})] dS$$

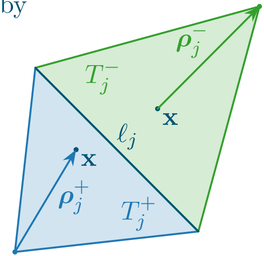
Rao–Wilton–Glisson Basis Functions

Discretize S with triangles and approximate $\mathbf{J}$ with basis-function representation:

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

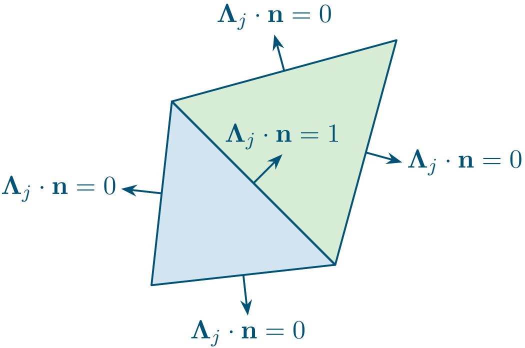
RWG basis functions defined for triangle pair by

$$\boldsymbol{\Lambda}_j(\mathbf{x}) = \begin{cases} \dfrac{\ell_j}{2A_j^+}\boldsymbol{\rho}_j^+, & \text{for } \mathbf{x} \in T_j^+ \\ \dfrac{\ell_j}{2A_j^-}\boldsymbol{\rho}_j^-, & \text{for } \mathbf{x} \in T_j^- \\ \mathbf{0}, & \text{otherwise} \end{cases}$$



ℓ_j: length of shared edge
A_j^+ and A_j^-: areas of triangles T_j^+ and T_j^- associated with $\boldsymbol{\Lambda}_j$
$\boldsymbol{\rho}_j^+$: vector from vertex of T_j^+ opposite of shared edge to $\mathbf{x}$
$\boldsymbol{\rho}_j^-$: vector to vertex of T_j^- opposite of shared edge from $\mathbf{x}$

Rao–Wilton–Glisson Basis Functions (continued)



RWG basis functions ensure

- $\mathbf{J}_h$ is tangent to elements
- $\mathbf{J}_h$ has no component normal to outer boundary of triangle pair

Along shared edge, component of $\mathbf{\Lambda}_j$ normal to edge is unity

- For edge shared by only 2 triangles, component of $\mathbf{J}_h$ normal to edge is J_j

Solution considered most accurate at edge midpoints

Discretized Problem

Find $\mathbf{J}_h \in \mathbb{V}_h$ (span of RWG basis functions), such that

$$a(\mathbf{J}_h, \mathbf{\Lambda}_i) = b(\mathbf{H}^{\mathcal{I}}, \mathbf{\Lambda}_i)$$

for $i = 1, \dots, n_b$

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

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

$Z_{i,j} = a(\mathbf{\Lambda}_j, \mathbf{\Lambda}_i)$, Impedance matrix

$J_j^h = J_j$, Current vector

$V_i = b(\mathbf{H}^{\mathcal{I}}, \mathbf{\Lambda}_i)$ Excitation vector

Outline

- Introduction
- The Method-of-Moments Implementation of the MFIE
- Code-Verification Approaches
  - Manufactured Surface Current and Green's Function
  - Solution-Discretization Error
  - Numerical-Integration Error
  - Domain-Discretization Error
- Numerical Examples
- Summary

Manufactured Surface Current

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

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

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

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

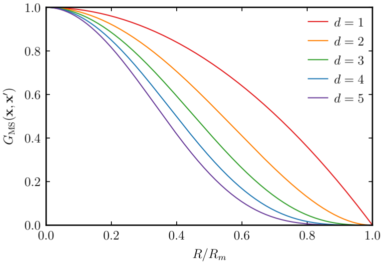
$$\mathbf{H}^{\mathcal{I}} = \frac{1}{2}\mathbf{J}_{\text{MS}} \times \mathbf{n} - \int_{S'} \left[\mathbf{J}_{\text{MS}}(\mathbf{x}') \times \nabla' G(\mathbf{x}, \mathbf{x}')\right] dS'$$

Manufactured Green's Function

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

Inaccurately computing $\mathbf{H}^{\mathcal{I}}$ contaminates convergence studies

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

Solution-Discretization Error

- Error due to basis-function approximation of solution: $\mathbf{J}_h(\mathbf{x}) = \sum_{j=1}^{n_b} J_j \mathbf{\Lambda}_j(\mathbf{x})$
- Measured with discretization error: $\mathbf{e}_\mathbf{J} = \mathbf{J}^h - \mathbf{J}_n$

$$\|\mathbf{e}_\mathbf{J}\| \leq C_\mathbf{J} h^{p_\mathbf{J}}$$

J_{n_j} : component of $\mathbf{J}_{\text{MS}}$ flowing from T_j^+ to T_j^-
$C_\mathbf{J}$: function of solution derivatives
h : measure of mesh size
$p_\mathbf{J}$: order of accuracy

- Compute $p_\mathbf{J}$ from $\|\mathbf{e}_\mathbf{J}\|$ across multiple meshes (expect $p_\mathbf{J} = 2$ for RWG)
- Avoid numerical-integration error contamination → integrate exactly (G_{MS})

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 $\mathbf{Z}\mathbf{J}^h = \mathbf{V}_{\text{MS}}$

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 $\mathbf{Z}\mathbf{J}^h = \mathbf{V}_{\text{MS}}$
$\mathbf{v}$: 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 $\|\mathbf{e}_\mathbf{J}\|_\infty$

Numerical-Integration Error

- Error due to quadrature evaluation of integrals on both sides of equation
- Measured by functionals

$$e_a(\mathbf{u}) = a^q(\mathbf{u},\mathbf{u}) - a(\mathbf{u},\mathbf{u}) \qquad e_b(\mathbf{u}) = b^q(\mathbf{H}_{\text{MS}}^{\mathcal{I}},\mathbf{u}) - b(\mathbf{H}_{\text{MS}}^{\mathcal{I}},\mathbf{u})$$

$$|e_a| \leq C_a h^{p_a} \qquad |e_b| \leq C_b h^{p_b}$$

a^q, b^q: quadrature evaluation of a and b
C_a, C_b: functions of integrand derivatives
p_a, p_b: order of accuracy of quadrature rules

- With multiple meshes, compute p_a and p_b from $|e_a|$ and $|e_b|$
- Avoid solution-discretization error contamination → cancel or eliminate it

Numerical-Integration Error: Solution-Discretization Error Avoidance

2 complementary approaches to avoiding solution-discretization error:

- Solution-discretization error cancellation

$$e_a(\mathbf{J}_{h_{\text{MS}}}) = a^q(\mathbf{J}_{h_{\text{MS}}}, \mathbf{J}_{h_{\text{MS}}}) - a(\mathbf{J}_{h_{\text{MS}}}, \mathbf{J}_{h_{\text{MS}}})$$
$$e_b(\mathbf{J}_{h_{\text{MS}}}) = b^q(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \mathbf{J}_{h_{\text{MS}}}) - b(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \mathbf{J}_{h_{\text{MS}}})$$

$\mathbf{J}_{h_{\text{MS}}}$ is the basis-function representation of $\mathbf{J}_{\text{MS}}$

- Solution-discretization error elimination

$$e_a(\mathbf{J}_{\text{MS}}) = a^q(\mathbf{J}_{\text{MS}}, \mathbf{J}_{\text{MS}}) - a(\mathbf{J}_{\text{MS}}, \mathbf{J}_{\text{MS}})$$
$$e_b(\mathbf{J}_{\text{MS}}) = b^q(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \mathbf{J}_{\text{MS}}) - b(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \mathbf{J}_{\text{MS}})$$

$e_a(\mathbf{J}_{h_{\text{MS}}})$ and $e_b(\mathbf{J}_{h_{\text{MS}}})$ are proportional to their influence on $\mathbf{e}_{\mathbf{J}} = \mathbf{J}^h - \mathbf{J}_n$

Domain-Discretization Error

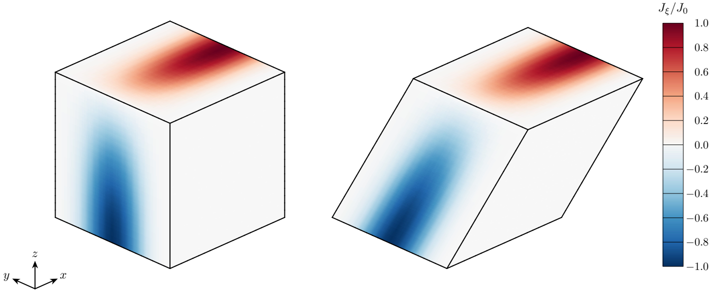
Triangular elements approximate curved S with faceted approximation S_h

- Accounting for curvature
  - Integrate over curved triangles that conform to S instead of planar triangles
  - Use solution-discretization error elimination approach
  - Assess curvature implementation and numerical integration
- Neglecting curvature
  - Use solution-discretization error cancellation approach
  - Assess numerical integration by computing integrals on S_h instead of S

Outline

- Introduction
- The Method-of-Moments Implementation of the MFIE
- Code-Verification Approaches
- Numerical Examples
  - No Curvature: Overview
  - No Curvature: Solution-Discretization Error
  - No Curvature: Numerical-Integration Error
  - Curvature: Overview
  - Curvature: Domain-Discretization Error
- Summary

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



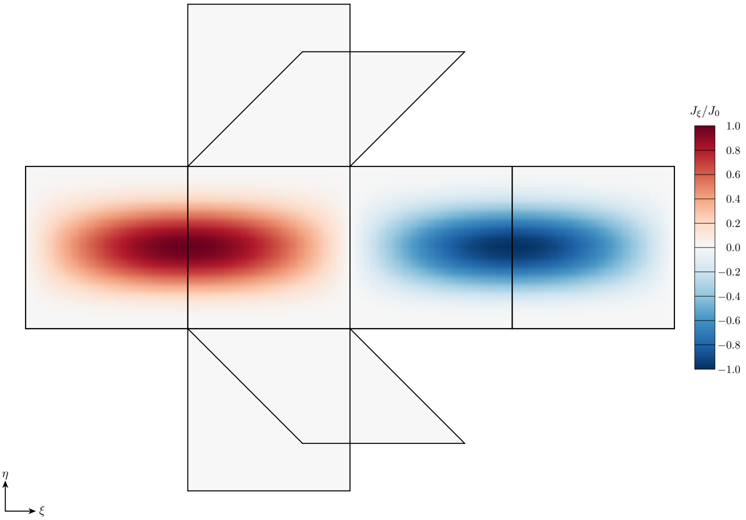
Manufactured surface current $J_\xi(\xi,\eta) = J_0 \begin{cases} \sin\left(\frac{\pi\xi}{2L}\right)\sin^3\left(\frac{\pi\eta}{L}\right), & \text{for } \mathbf{n}\cdot\mathbf{e}_y = 0 \\ 0, & \text{for } \mathbf{n}\cdot\mathbf{e}_y \neq 0 \end{cases}$

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

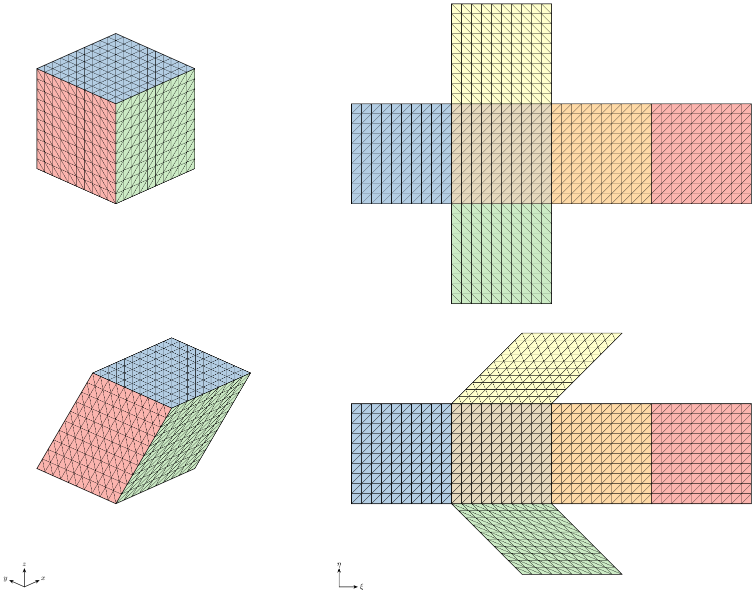
Surface-fixed coordinate system:

- $\eta = y \in [0,\, 1]$ 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

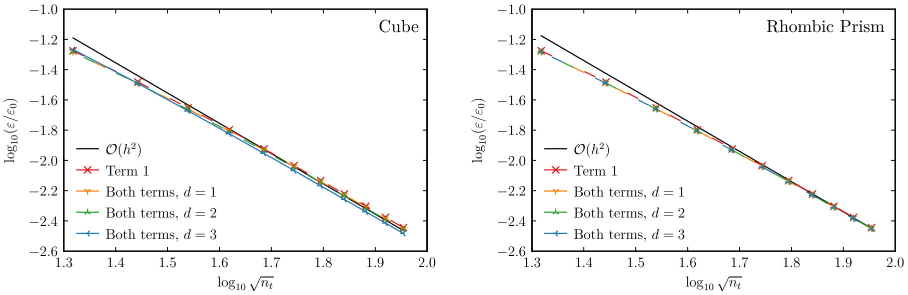
Manufactured Surface Current $\mathbf{J}_{\text{MS}}$ for Cube and Rhombic Prism



Cube and Rhombic Prism Meshes, with $n_t = 1200$



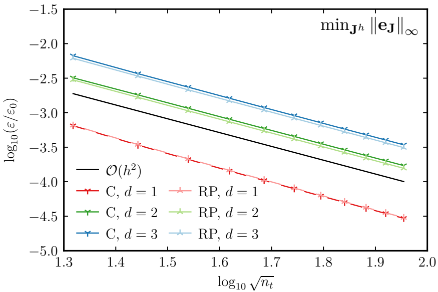
Solution-Discretization Error: $\varepsilon = \|\mathbf{e_J}\|_\infty$

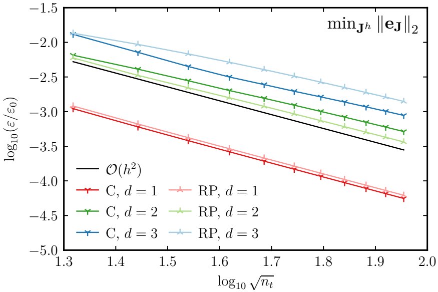


$$a(\mathbf{J}_h, \mathbf{\Lambda}_i) = \underbrace{\frac{1}{2}\int_S \mathbf{\Lambda}_i(\mathbf{x}) \cdot \mathbf{J}_h(\mathbf{x}) dS}_{\text{Term 1}} + \underbrace{\int_S \mathbf{\Lambda}_i(\mathbf{x}) \cdot \left(\mathbf{n}(\mathbf{x}) \times \int_{S'} \left[\nabla' G_{\text{MS}}(\mathbf{x}, \mathbf{x}')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

$$G_{\text{MS}}(\mathbf{x}, \mathbf{x}') = \left(1 - \frac{R^2}{R_m^2}\right)^d$$

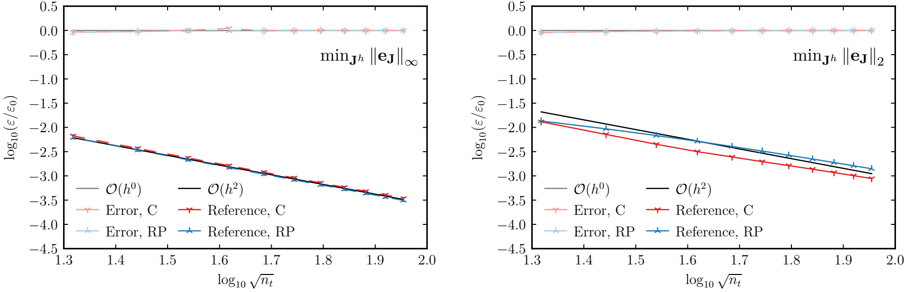
Solution-Discretization Error: $\varepsilon = \|\mathbf{e_J}\|_\infty$, Term 2





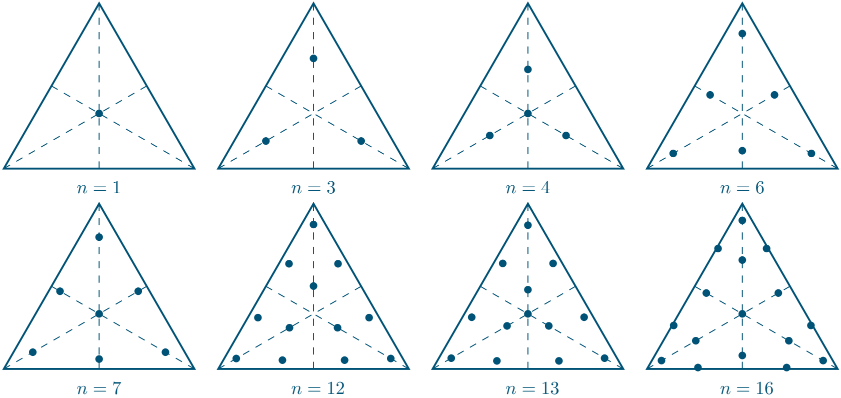
	$\min_{\mathbf{J}^h} \|\mathbf{e_J}\|_\infty$		$\min_{\mathbf{J}^h} \|\mathbf{e_J}\|_2$	
Mesh	C	RP	C	RP
1–2	2.0800	2.0653	2.0811	1.2935
2–3	2.0141	2.0529	2.1055	1.4193
3–4	2.0303	2.0193	1.9159	1.5150
4–5	2.0196	2.0163	1.6421	1.5847
5–6	2.0061	2.0242	1.6677	1.6372
6–7	2.0133	2.0158	1.5800	1.6779
7–8	2.0113	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

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



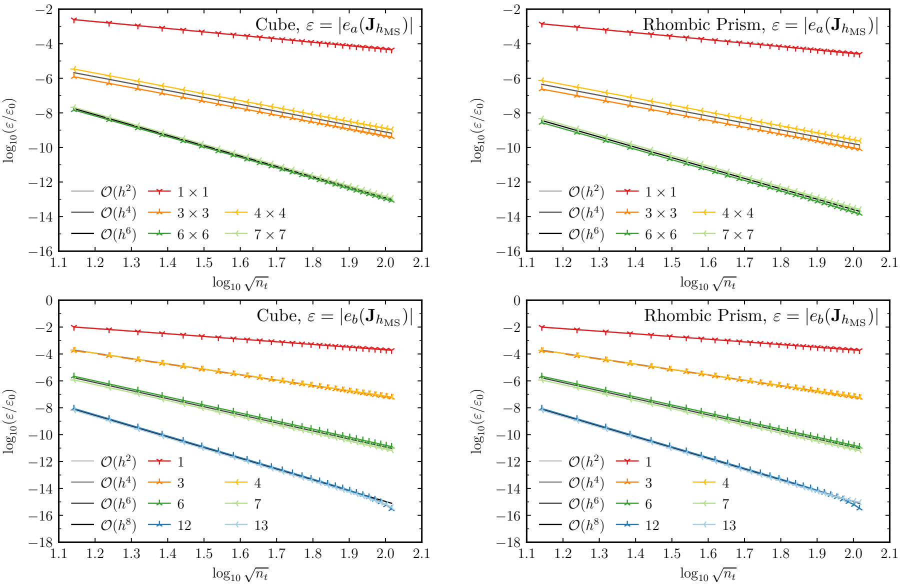
Both norms are able to detect the coding error

Numerical-Integration Error: Polynomial Quadrature Rules

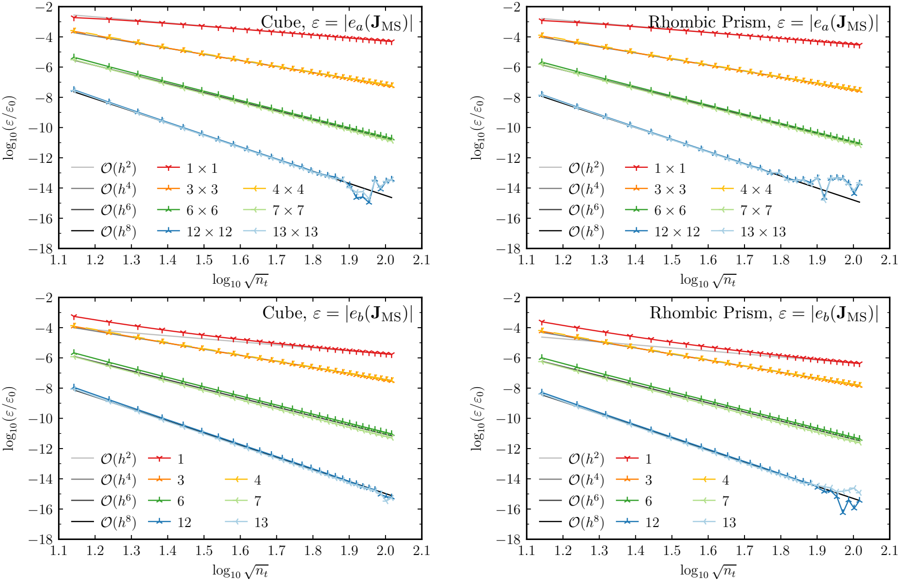


n	1	3	4	6	7	12	13	16
Max. integrand degree	1	2	3	4	5	6	7	8
Convergence rate	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^8)$	$\mathcal{O}(h^8)$	$\mathcal{O}(h^{10})$

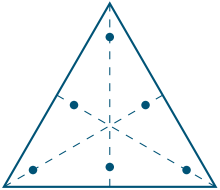
Numerical-Integration Error: Cancellation ($d = 3$)



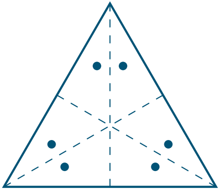
Numerical-Integration Error: Elimination ($d = 3$)



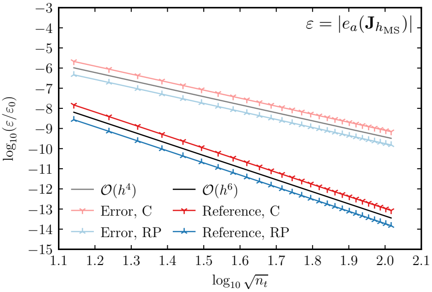
Numerical-Integration Error: Coding Error, Cancellation ($d = 3$)

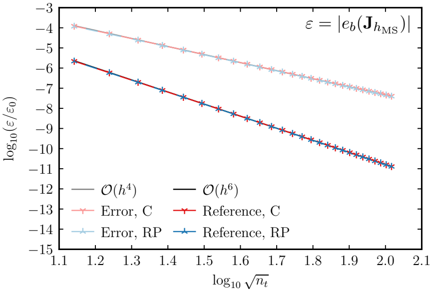


Maximum polynomial degree: 4

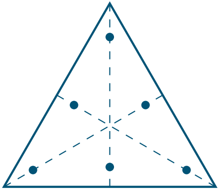


Maximum polynomial degree: 3

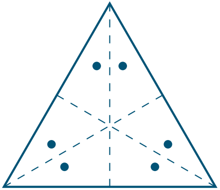




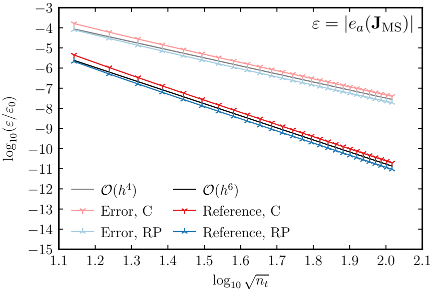
Numerical-Integration Error: Coding Error, Elimination ($d = 3$)

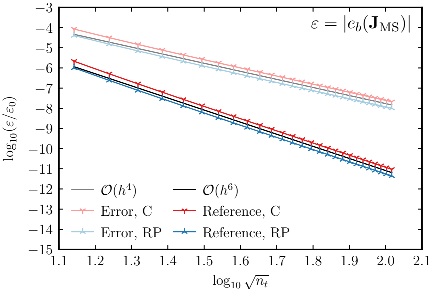


Maximum polynomial degree: 4

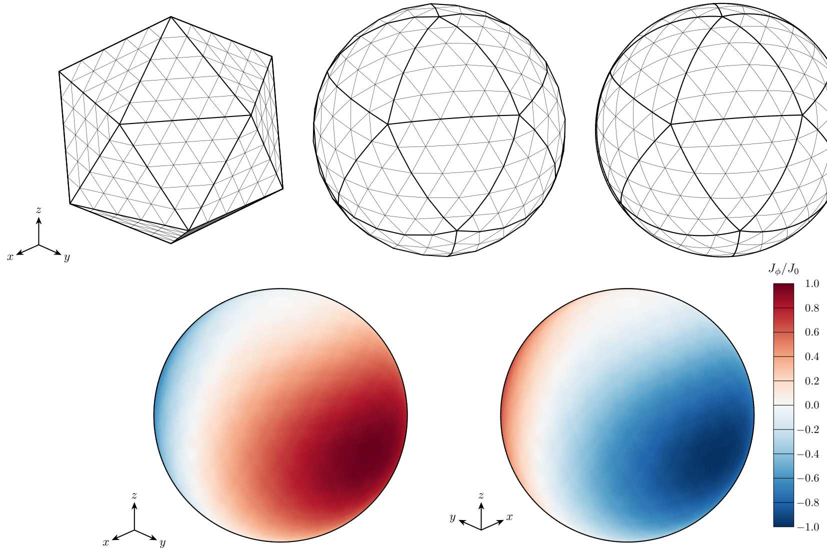


Maximum polynomial degree: 3



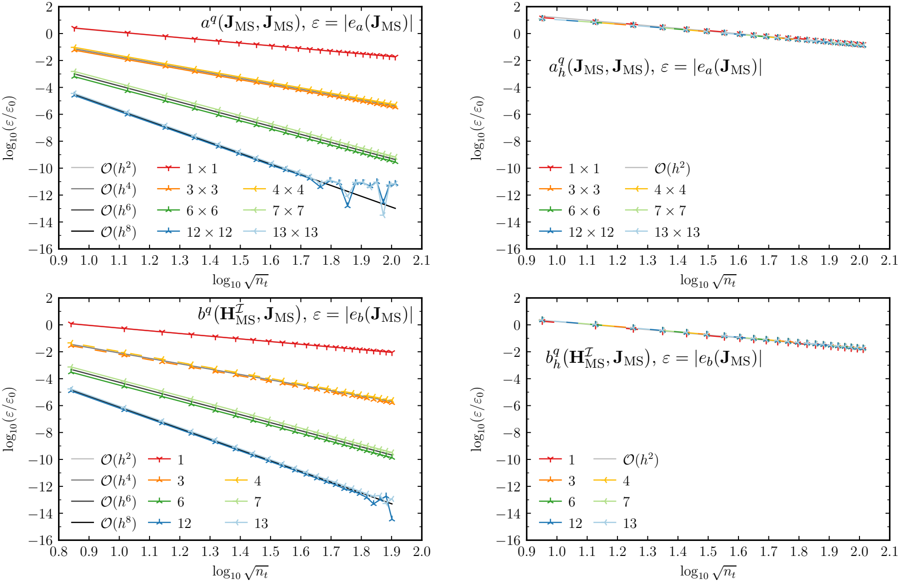


Manufactured Surface Current $\mathbf{J}_{\text{MS}}$ and Mesh ($n_t = 500$) for Sphere

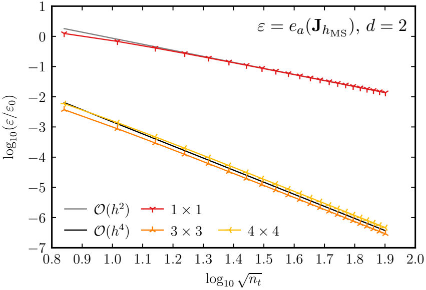


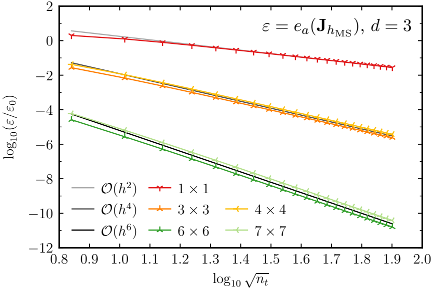
Manufactured surface current $\mathbf{J}_{\text{MS}}(\mathbf{x}) = J_\phi(\theta,\phi)\mathbf{e}_\phi = (J_0 \sin^2\theta \sin\phi)\mathbf{e}_\phi$ (ϕ around z)

Domain-Discretization Error: Elimination ($d = 3$)



Domain-Discretization Error: Cancellation, No Curvature





Outline

- Introduction
- The Method-of-Moments Implementation of the MFIE
- Code-Verification Approaches
- Numerical Examples
- Summary
  - Closing Remarks

Closing Remarks

3 error sources in MoM implementation of MFIE:

- **Solution-discretization error** – isolated
  - Integrate exactly
  - Optimize to select unique solution when equations are singular
- **Numerical-integration error** – isolated
  - Cancel solution-discretization error – uses basis functions
  - Eliminate solution-discretization error – does not use basis functions
- **Domain-discretization error** – addressed
  - Account for curvature – integrate over curved triangular elements
  - Neglect curvature – integrate over planar triangular elements

Additional Information

- 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`

Questions?

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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-NA0003525.

Outline

- Code-Verification Approaches
  - Solution-Discretization Error
  - Numerical-Integration Error
- Numerical Examples

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 $\mathbf{Z}\mathbf{J}^h = \mathbf{V}_{\text{MS}}$

Compute pivoted QR factorization of $\mathbf{Z}^H$:

$$\mathbf{Z}^H\mathbf{P} = [\mathbf{Q}_1,\, \mathbf{Q}_2]\begin{bmatrix}\mathbf{R}_1\\ \mathbf{0}\end{bmatrix} = \mathbf{Q}_1\mathbf{R}_1,$$

where $\mathbf{Z} \in \mathbb{C}^{n_b\times n_b}$, $\mathbf{Q}_1 \in \mathbb{C}^{n_b\times m_b}$, $\mathbf{Q}_2 \in \mathbb{C}^{n_b\times(n_b-m_b)}$, and $\mathbf{R}_1 \in \mathbb{C}^{m_b\times n_b}$

Numerically, pivoting facilitates determination of rank $m_b \leq n_b$ of $\mathbf{Z}$

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

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

$\mathbf{u} \in \mathbb{C}^{m_b}$: coefficients that satisfy $\mathbf{Z}\mathbf{J}^h = \mathbf{V}_{\text{MS}}$
$\mathbf{v} \in \mathbb{C}^{n_b-m_b}$: coefficients that bring $\mathbf{J}^h$ closest to $\mathbf{J}_n$, given $\mathbf{u}$

Solution-Discretization Error: Solution Uniqueness (continued)

L^2 Optimization

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{e}_\mathbf{J}\|_2 \\ \text{subject to} \quad & a(\mathbf{J}_h, \boldsymbol{\Lambda}_i) = b(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \boldsymbol{\Lambda}_i) \end{aligned}$$

- Closed-form solution: $\mathbf{J}^h = \mathbf{J}_n + \mathbf{Q}_1(\mathbf{u} - \mathbf{Q}_1^H \mathbf{J}_n)$, where $\mathbf{R}_1^H \mathbf{u} = \mathbf{P}^T \mathbf{V}$
- May require finer meshes to see expected rate when measuring $\|\mathbf{e}_\mathbf{J}\|_\infty$

L^∞ Optimization

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{e}_\mathbf{J}\|_\infty \\ \text{subject to} \quad & a(\mathbf{J}_h, \boldsymbol{\Lambda}_i) = b(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \boldsymbol{\Lambda}_i) \end{aligned}$$

- Linear programming problem – more expensive
- Does not require finer meshes to see expected rate when measuring $\|\mathbf{e}_\mathbf{J}\|_\infty$

Numerical-Integration Error: Relation to Discretization Error

- Relate $e_a(\mathbf{J}_{h_{\text{MS}}})$ to $\mathbf{e}_\mathbf{J}$ by solving

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{e}_\mathbf{J}\|_2 \\ \text{subject to} \quad & a^q(\mathbf{J}_h, \mathbf{J}_{h_{\text{MS}}}) = a(\mathbf{J}_{h_{\text{MS}}}, \mathbf{J}_{h_{\text{MS}}}) \end{aligned}$$

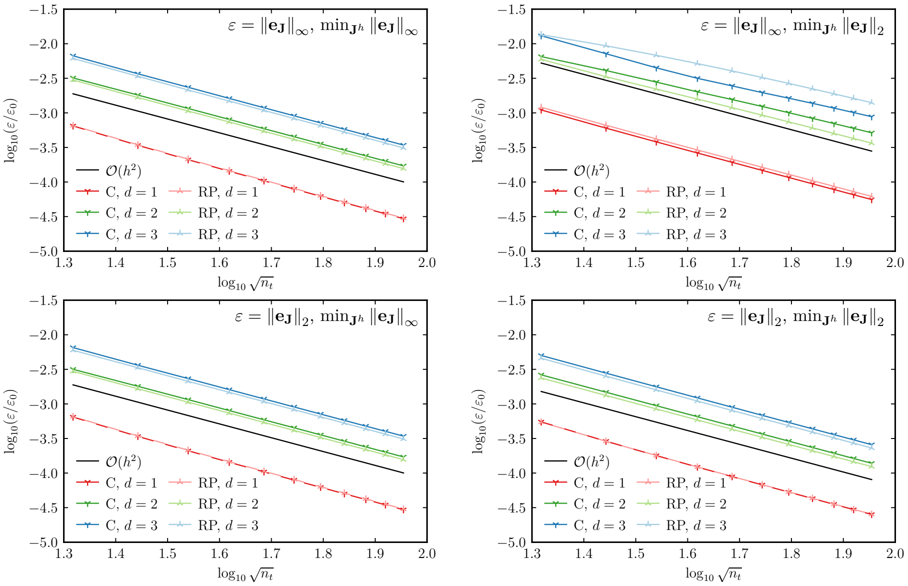
- Relate $e_b(\mathbf{J}_{h_{\text{MS}}})$ to $\mathbf{e}_\mathbf{J}$, by solving

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{e}_\mathbf{J}\|_2 \\ \text{subject to} \quad & b^q(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \mathbf{J}_h) = b(\mathbf{H}^{\mathcal{I}}_{\text{MS}}, \mathbf{J}_{h_{\text{MS}}}) \end{aligned}$$

Outline

- Code-Verification Approaches
- Numerical Examples
  - No Curvature: Solution-Discretization Error
  - No Curvature: Numerical-Integration Error
  - Curvature: Domain-Discretization Error

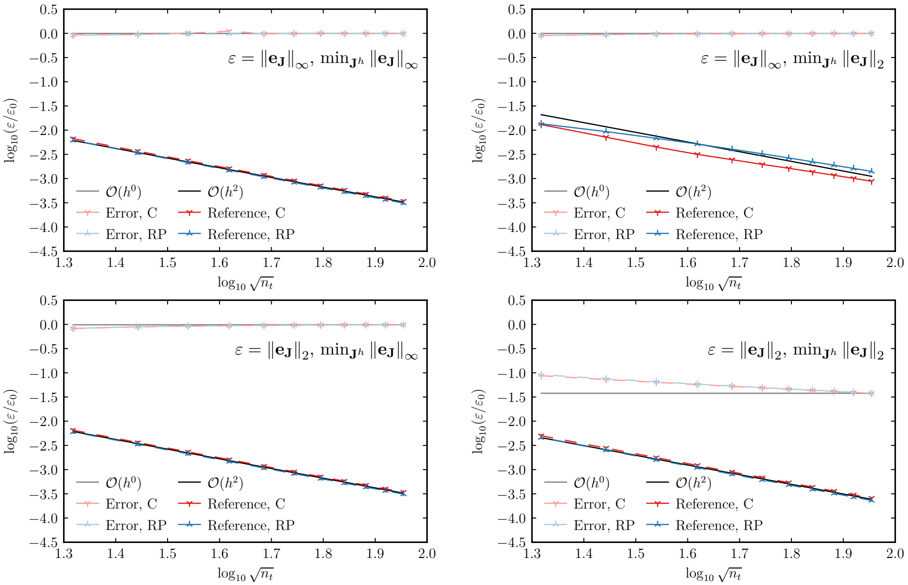
Solution-Discretization Error: $\varepsilon = \|\mathbf{e}_\mathbf{J}\|$, Term 2



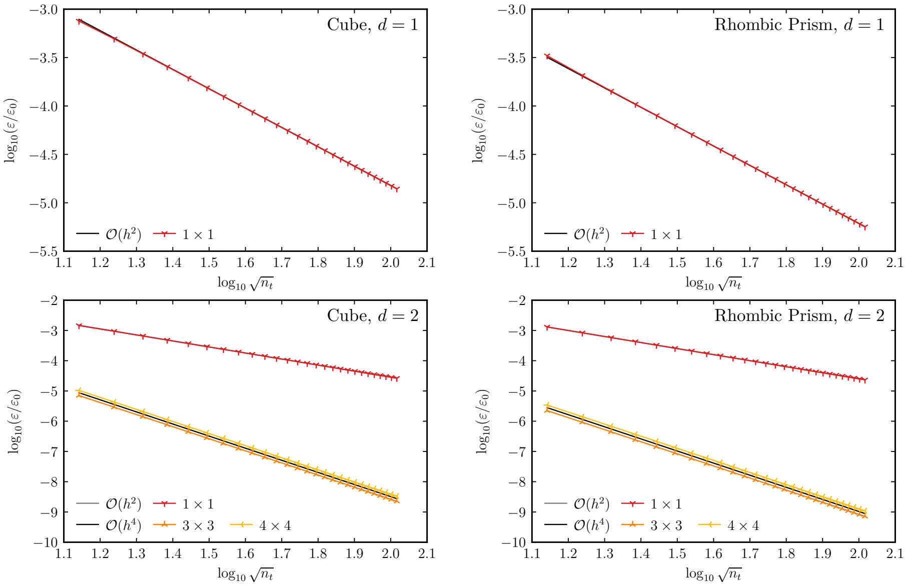
Solution-Discretization Error: Convergence Rates, Term 2

	$\varepsilon = \lVert \mathbf{e}_\mathbf{J} \rVert_\infty$				$\varepsilon = \lVert \mathbf{e}_\mathbf{J} \rVert_2$			
	$\min_{\mathbf{J}^h} \lVert \mathbf{e}_\mathbf{J} \rVert_\infty$		$\min_{\mathbf{J}^h} \lVert \mathbf{e}_\mathbf{J} \rVert_2$		$\min_{\mathbf{J}^h} \lVert \mathbf{e}_\mathbf{J} \rVert_\infty$		$\min_{\mathbf{J}^h} \lVert \mathbf{e}_\mathbf{J} \rVert_2$	
Mesh	C	RP	C	RP	C	RP	C	RP
1–2	2.0800	2.0653	2.0811	1.2935	2.0447	2.0229	2.0454	2.0763
2–3	2.0141	2.0529	2.1055	1.4193	1.9948	2.0323	2.0359	2.0499
3–4	2.0303	2.0193	1.9159	1.5150	2.0141	1.9999	2.0283	2.0372
4–5	2.0196	2.0163	1.6421	1.5847	2.0064	2.0093	2.0229	2.0297
5–6	2.0061	2.0242	1.6677	1.6372	2.0060	2.0102	2.0190	2.0246
6–7	2.0133	2.0158	1.5800	1.6779	2.0057	2.0097	2.0162	2.0211
7–8	2.0113	2.0167	1.6282	1.7104	2.0057	2.0122	2.0140	2.0184
8–9	2.0037	2.0122	1.6664	1.7369	1.9965	2.0076	2.0123	2.0163
9–10	2.0086	2.0117	1.6974	1.7589	2.0039	2.0067	2.0110	2.0146
10–11	2.0053	2.0118	1.7231	1.7776	2.0039	2.0094	2.0099	2.0133

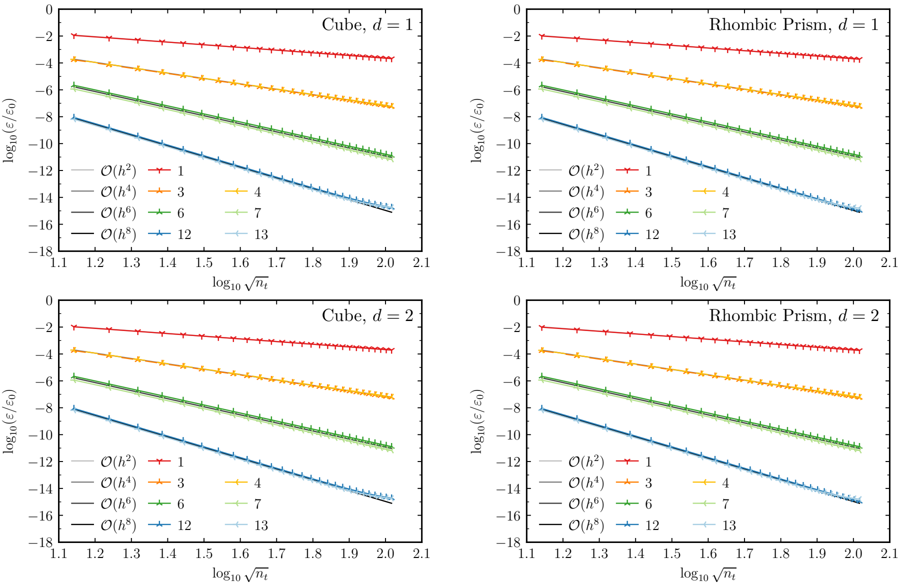
Solution-Discretization Error: $\varepsilon = \|\mathbf{e}_\mathbf{J}\|$ for Coding Error ($d = 3$)



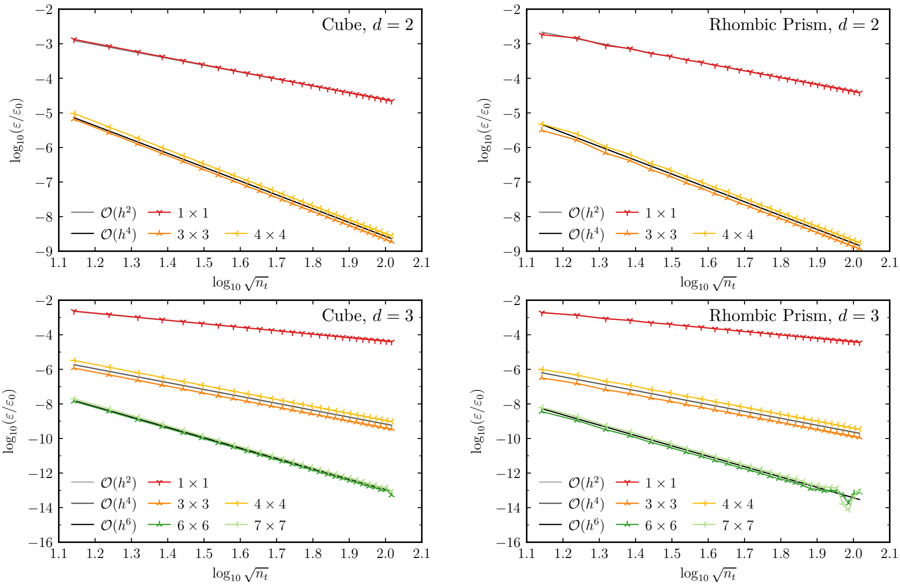
Numerical-Integration Error: $\varepsilon = |e_a(\mathbf{J}_{h_{\text{MS}}})|$ (Cancellation)



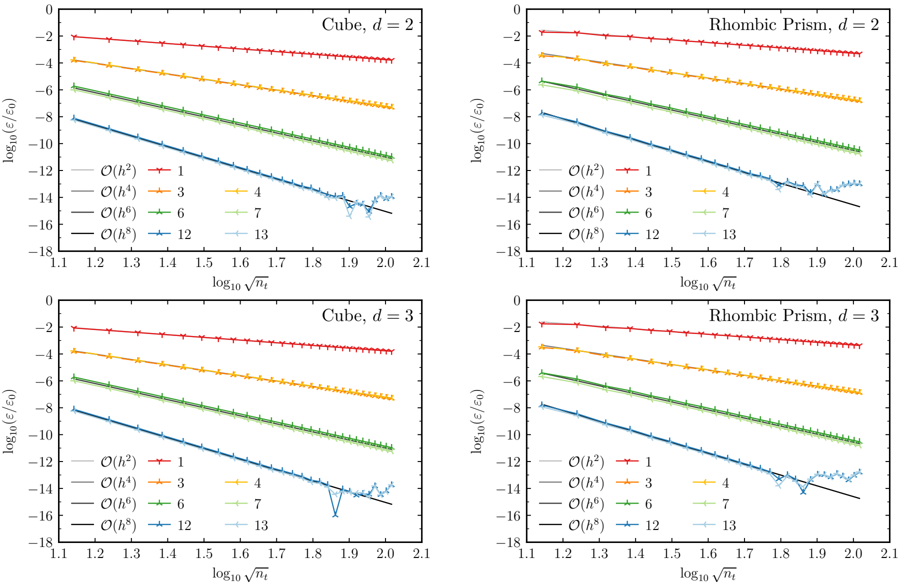
Numerical-Integration Error: $\varepsilon = |e_b(\mathbf{J}_{h_{\text{MS}}})|$ (Cancellation)



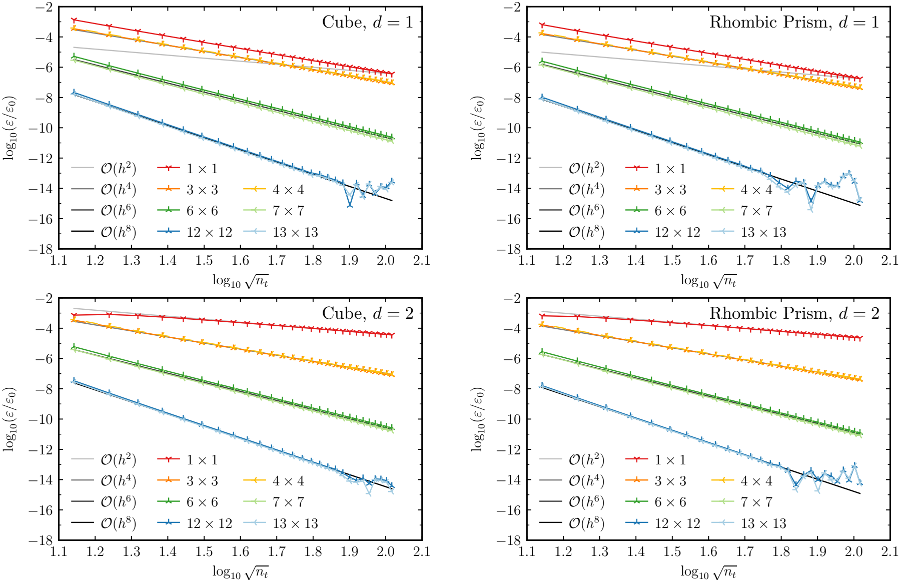
Numerical-Integration Error: $\varepsilon = \|\mathbf{e}_\mathbf{J}\|_\infty$ (Cancel/Opt s.t. $a^q = a$)



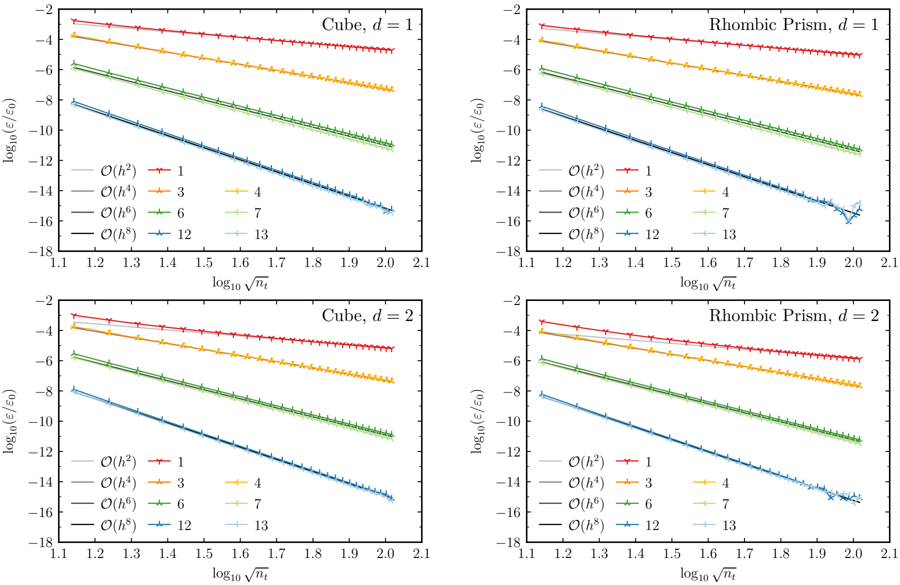
Numerical-Integration Error: $\varepsilon = \|\mathbf{e}_\mathbf{J}\|_\infty$ (Cancel/Opt s.t. $b^q = b$)



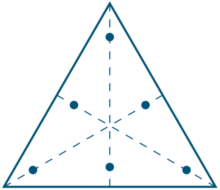
Numerical-Integration Error: $\varepsilon = |e_a(\mathbf{J}_{\mathrm{MS}})|$ (Elimination)



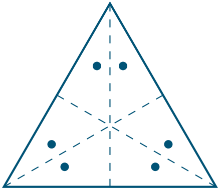
Numerical-Integration Error: $\varepsilon = |e_b(\mathbf{J}_{\text{MS}})|$ (Elimination)



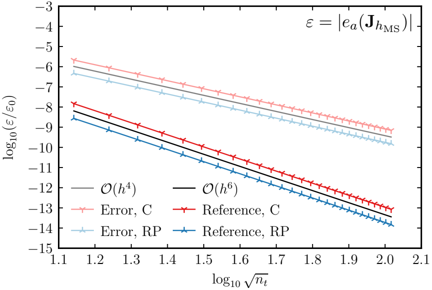
Numerical-Integration Error: Coding Error ($d = 3$)

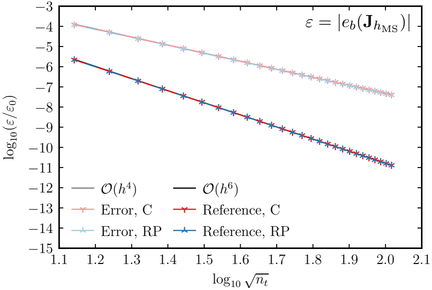


Maximum polynomial degree: 4

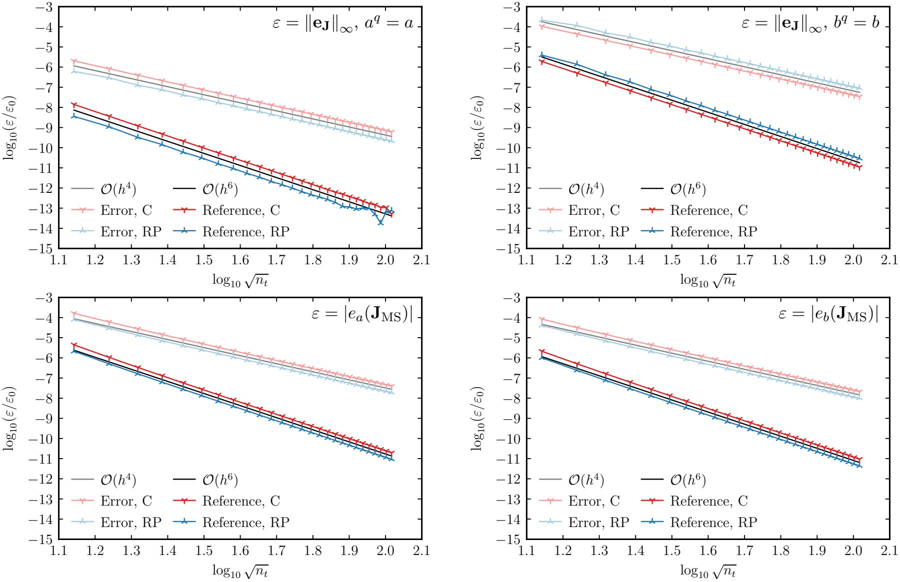


Maximum polynomial degree: 3

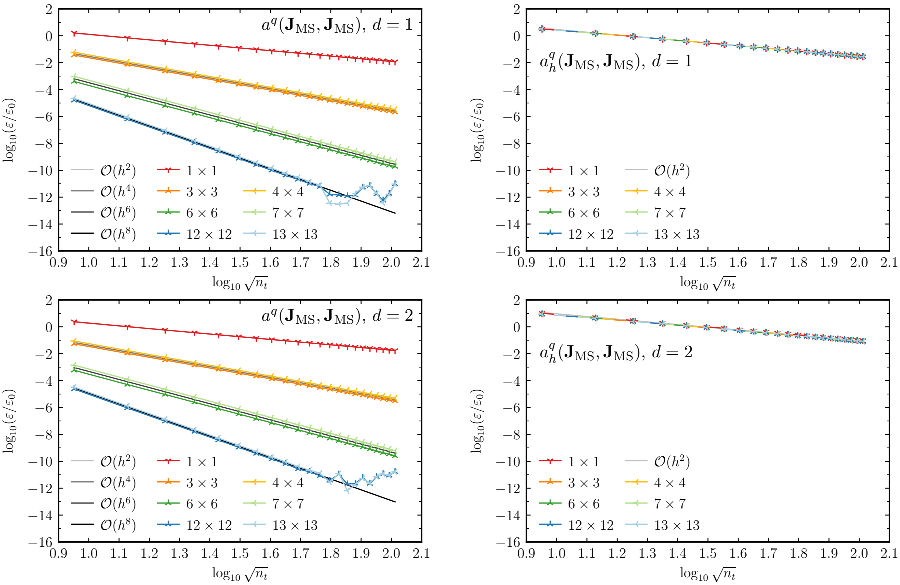




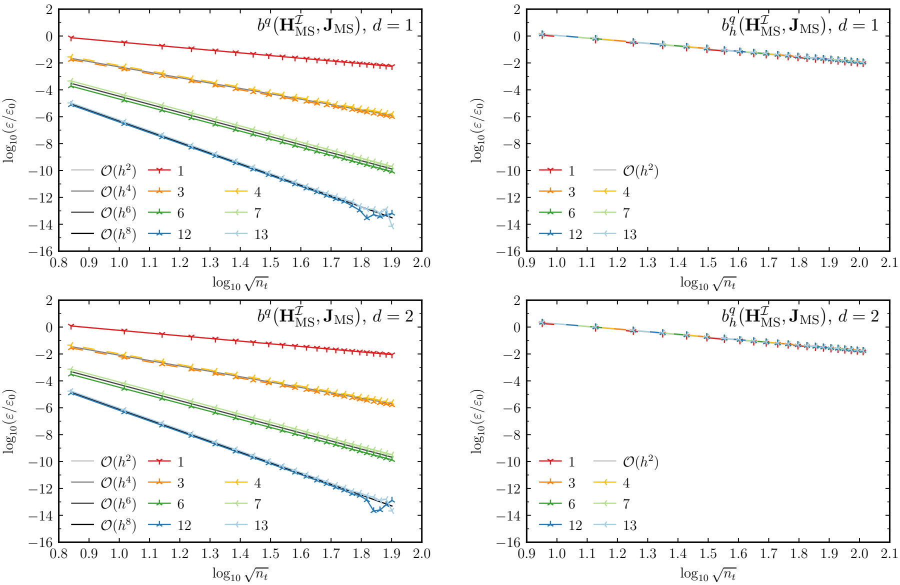
Numerical-Integration Error: Coding Error ($d = 3$) (cont.)



Domain-Discretization Error: $\varepsilon = |e_a(\mathbf{J}_{\text{MS}})|$ (Elimination)



Domain-Discretization Error: $\varepsilon = |e_b(\mathbf{J}_{\text{MS}})|$ (Elimination)



Domain-Discretization Error: Cancellation, No Curvature