SAND2011-3052C

Verification of Radiation Transport Codes with Unstructured Meshes

Shawn D. Pautz
Sandia National Laboratories
Albuquerque, NM, U.S.

M&C 2011
Rio de Janeiro, Brazil
May 12, 2011

Sandia National Laboratories

Outline

- General verification definitions and techniques
- Special considerations for unstructured meshes
- Application to Sceptre transport code
- Results
- Conclusions

Sandia National Laboratories

V&V Definitions

Verification: Does the computer code faithfully implement the intended physical model?

"Are we solving the equations correctly?"

Validation: Is the physical model adequate for its intended use?

"Are we solving the correct equations?"

V&V Mappings

reality

Physical model $\qquad$ Validation

$$Du = q$$

Discretization $\qquad$ Verification

$$D_h u_h = q_h$$

Implementation $\qquad$ Verification

$$D_h^* u_h^* = q_h^*$$

Sandia National Laboratories

Verification techniques: method of manufactured solutions

General equation to solve:

$$Du = q$$

Classic verification approach:

Set q → derive u; compare u_h to u

MMS verification approach:

Set u → derive q; compare u_h to u

Verification techniques: method of manufactured solutions

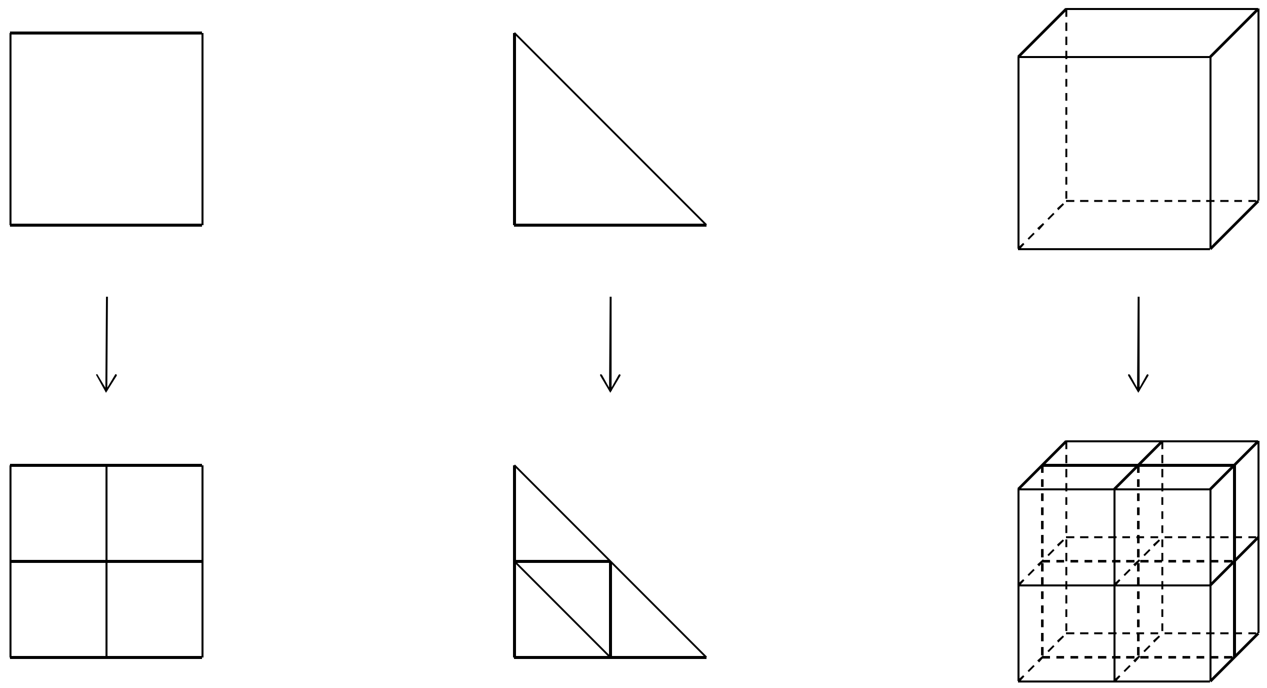
The method of manufactured solutions is used to verify error convergence rates.

$$\varepsilon_h = u_h - u = ch^p + c'h^{p+1} + \cdots$$

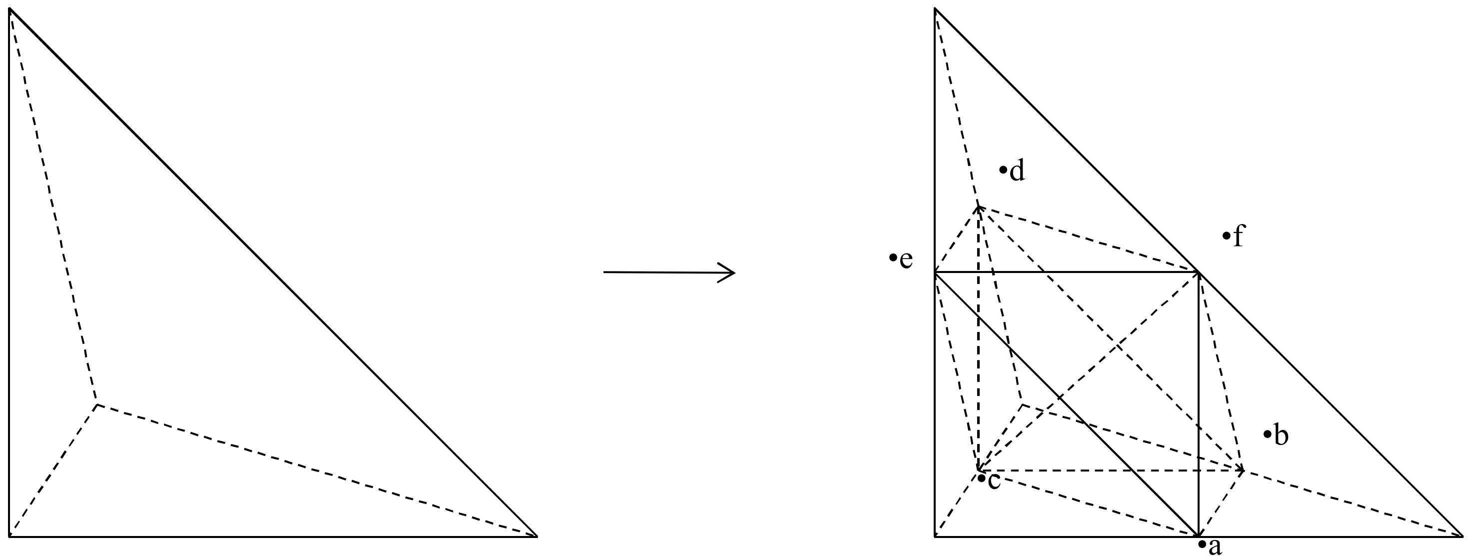
$$\rightarrow \quad p = \log_2\left(\frac{\varepsilon_h}{\varepsilon_{h/2}}\right)$$

We compare the observed convergence rate to the theoretical rate on a series of meshes to verify the code.

Unstructured meshes: refinement rules



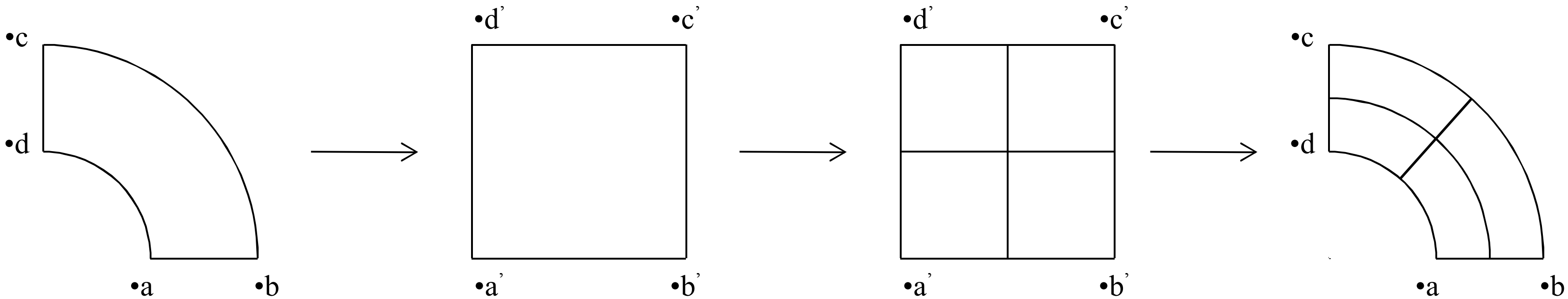
Unstructured meshes: refinement rules (tetrahedra)



The shortest diagonal *ad*, *be*, or *cf* should be used to subdivide each tetrahedron. Failure to do so can result in tetrahedra of increasingly poor quality.

Sandia National Laboratories

Mesh refinement rules for curved elements

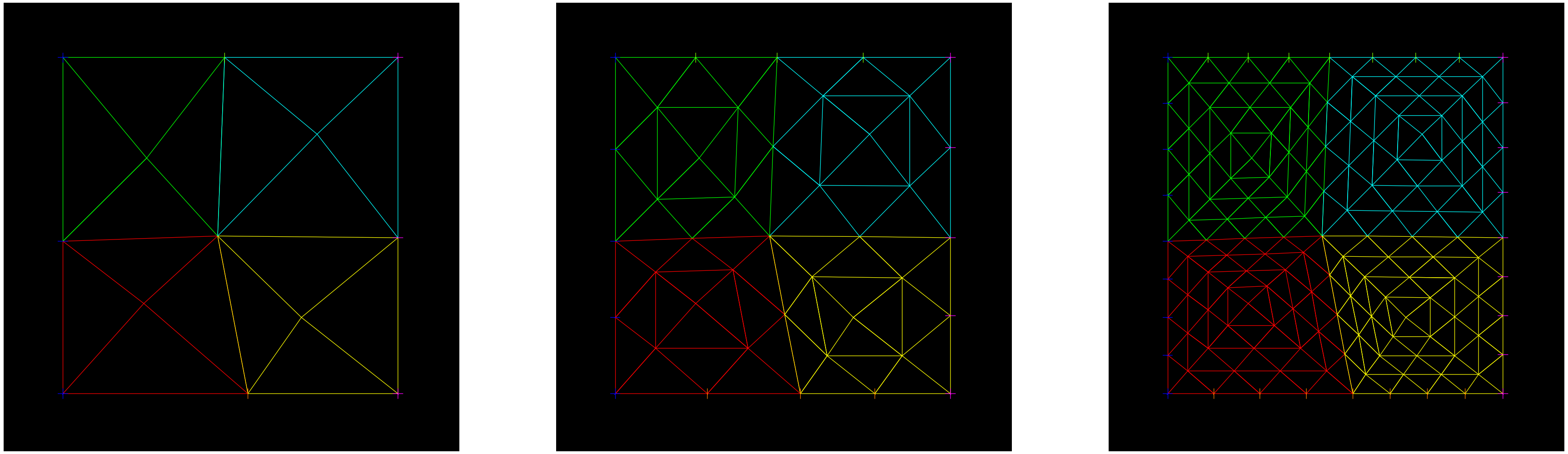


Element refinement should occur in interpolation space (i.e. "master element").

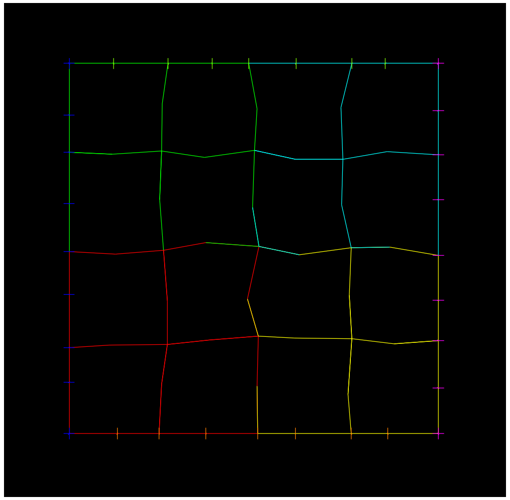
- ensures valid elements
- preserves Jacobians

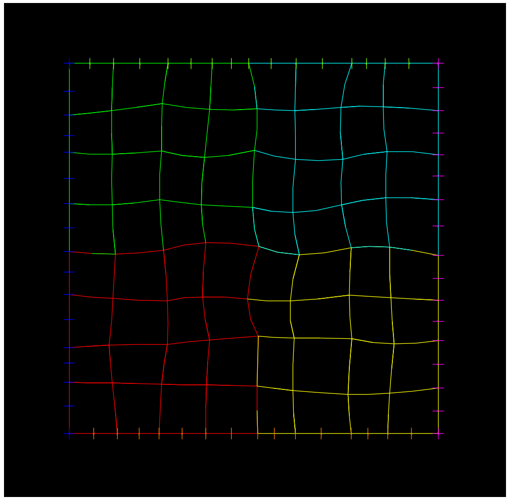
Sandia National Laboratories

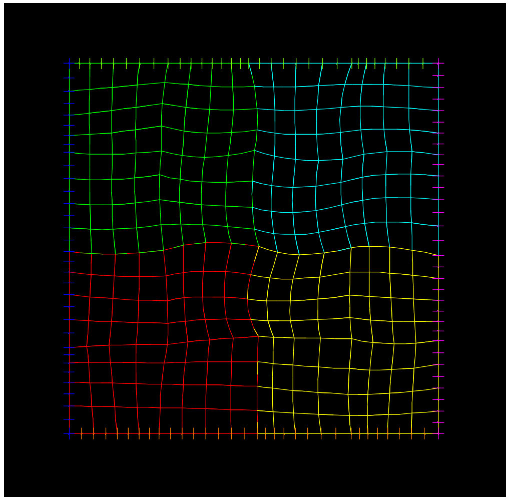
Example: triangular mesh refinement



Example: quadrilateral mesh refinement







Forms of Boltzmann Transport Equation

First-order: $\left[\Omega \cdot \nabla + \sigma_t\right]\psi(r,\Omega) = M\Sigma D\psi(r,\Omega) + Q(r,\Omega)$

Second-order:

$$\left[-\Omega \cdot \nabla \Re^{-1} \Omega \cdot \nabla + \Re\right]\psi(r,\Omega) = Q(r,\Omega) - \Omega \cdot \nabla\left[\Re^{-1} Q(r,\Omega)\right]$$

The two continuous forms above are equivalent.

Discretizations, however, yield different properties:

- Solutions
- Solvers

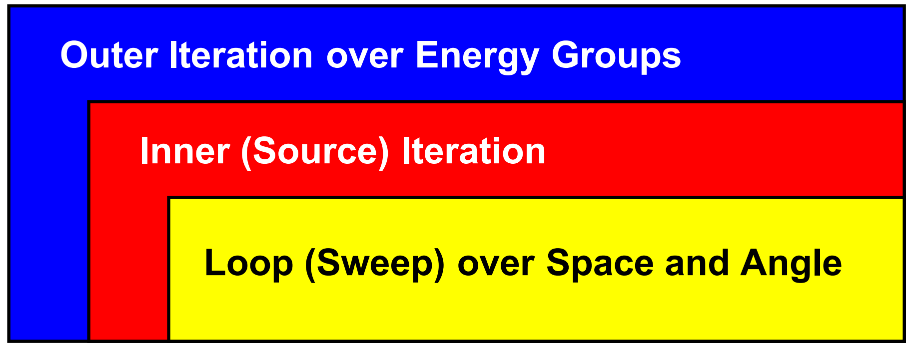
SCEPTRE Code Description

SCEPTRE: Sandia's Computational Engine for Particle Transport for Radiation Effects

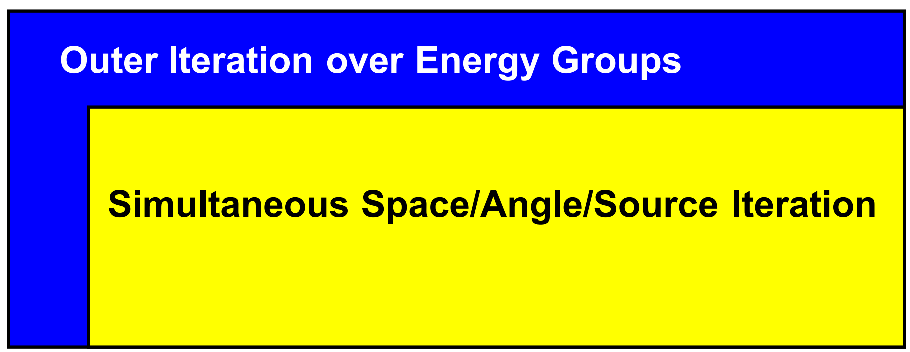
- Linear steady-state deterministic Boltzmann transport solver
- Discrete ordinates (S_n) in angle
- Multigroup in energy
- Discontinuous FEM (1st order) and continuous FEM (2nd order) on unstructured meshes
- Second-order variants
  - Even/odd parity flux (EOPF)
  - Self-adjoint angular flux (SAAF)
  - Least-squares (LS)

Solvers for Boltzmann Transport Equation in Sceptre

First-order:



Second-order:



Sandia National Laboratories

Error metrics used by Sceptre

L-norm: $$\left[\int d\Omega \int dV \left|\psi^{n}\right|\right]^{1/n}$$

H-norm: $$\left[\int d\Omega \int dV \left\|\nabla \psi\right\|^{n}\right]^{1/n}$$

Streaming-norm: $$\left[\int d\Omega \int dV \left|\Omega \cdot \nabla \psi\right|^{n}\right]^{1/n}$$

Used to determine the difference between a computational result and a reference solution.

Test strategy

We generate tests in two different categories:

- "Exact" tests: The solution can be exactly computed/represented regardless of problem refinement (also called "patch" tests)
- "Inexact" tests: The discrete solution will contain discretization error, which hopefully decreases with problem refinement.

Exactly solvable problems (prediction)

	1	x	y	z	xy	xz	yz	xyz	x^2	y^2	z^2	x^2y	x^2z	y^2x	y^2z	z^2x	z^2y	x^2yz	y^2xz	z^2xy
edge2	SU	SU																		
edge3	SU	SU							S											
tri3	SU	SU	SU																	
tri6	SU	SU	SU		S				S	S										
quad4	SU	SU	SU		S															
quad8	SU	SU	SU		S				S	S		S		S						
tet4	SU	SU	SU	SU																
tet10	SU	SU	SU	SU	S	S	S		S	S	S									
hex8	SU	SU	SU	SU	S	S	S	S												
hex20	SU	SU	SU	SU	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S

S: Structured meshes

U: Unstructured meshes

Jacobians are generally non-constant in unstructured meshes, resulting in some finite element integrals that cannot be exactly computed with standard quadratures.

(In)Exactly solvable problems, 2D results
h = 0.25, structured mesh, isotropic scattering

Element Type	Solver	L_{inf} error norm					
		1	x	x^2	xy	x^3	x^2y
tri3	Sn-1st	7.79×10^{-14}	3.70×10^{-14}	**1.06×10^{-2}**	**5.04×10^{-3}**	**2.73×10^{-2}**	**1.26×10^{-2}**
tri6	Sn-1st	7.74×10^{-13}	7.85×10^{-13}	7.43×10^{-13}	6.60×10^{-13}	**9.47×10^{-4}**	**2.79×10^{-4}**
quad4	Sn-1st	2.55×10^{-15}	1.67×10^{-15}	**1.18×10^{-2}**	1.22×10^{-15}	**3.13×10^{-2}**	**1.04×10^{-2}**
quad8	Sn-1st	1.05×10^{-14}	7.55×10^{-15}	7.33×10^{-15}	4.44×10^{-15}	**1.05×10^{-3}**	3.44×10^{-15}
tri3	Sn-EOP	2.29×10^{-14}	4.90×10^{-14}	**2.07×10^{-2}**	**5.32×10^{-3}**	**5.20×10^{-2}**	**2.46×10^{-2}**
tri6	Sn-EOP	8.06×10^{-14}	1.10×10^{-13}	1.14×10^{-13}	1.98×10^{-13}	**2.33×10^{-3}**	**5.61×10^{-4}**
quad4	Sn-EOP	3.66×10^{-15}	7.12×10^{-14}	**1.50×10^{-2}**	7.26×10^{-14}	**4.18×10^{-2}**	**1.31×10^{-2}**
quad8	Sn-EOP	5.63×10^{-14}	1.38×10^{-13}	1.13×10^{-13}	1.92×10^{-13}	**1.80×10^{-3}**	1.56×10^{-13}

(In)Exactly solvable problems, 2D results
h = 0.25, unstructured mesh, isotropic scattering

Element Type	Solver	L_{inf} error norm					
		1	x	x^2	xy	x^3	x^2y
tri3	Sn-1st	1.03×10^{-13}	5.27×10^{-14}	**1.40×10^{-2}**	**6.83×10^{-3}**	**3.22×10^{-2}**	**1.60×10^{-2}**
tri6	Sn-1st	1.53×10^{-12}	1.05×10^{-12}	**7.40×10^{-3}**	**2.52×10^{-3}**	**2.14×10^{-2}**	**5.32×10^{-3}**
quad4	Sn-1st	4.11×10^{-15}	2.66×10^{-15}	**2.09×10^{-2}**	**6.12×10^{-3}**	**4.03×10^{-2}**	**1.85×10^{-2}**
quad8	Sn-1st	4.64×10^{-14}	2.76×10^{-14}	**3.04×10^{-2}**	**1.75×10^{-2}**	**5.72×10^{-2}**	**3.26×10^{-2}**
tri3	Sn-EOP	3.67×10^{-14}	7.75×10^{-14}	**2.34×10^{-2}**	**9.23×10^{-3}**	**5.39×10^{-2}**	**2.43×10^{-2}**
tri6	Sn-EOP	7.41×10^{-14}	9.03×10^{-14}	**6.83×10^{-3}**	**3.25×10^{-3}**	**1.97×10^{-2}**	**6.95×10^{-3}**
quad4	Sn-EOP	9.88×10^{-14}	6.87×10^{-14}	**3.34×10^{-2}**	**9.51×10^{-3}**	**6.12×10^{-2}**	**2.77×10^{-2}**
quad8	Sn-EOP	6.76×10^{-14}	9.76×10^{-14}	**1.57×10^{-2}**	**7.64×10^{-3}**	**3.60×10^{-2}**	**1.36×10^{-2}**

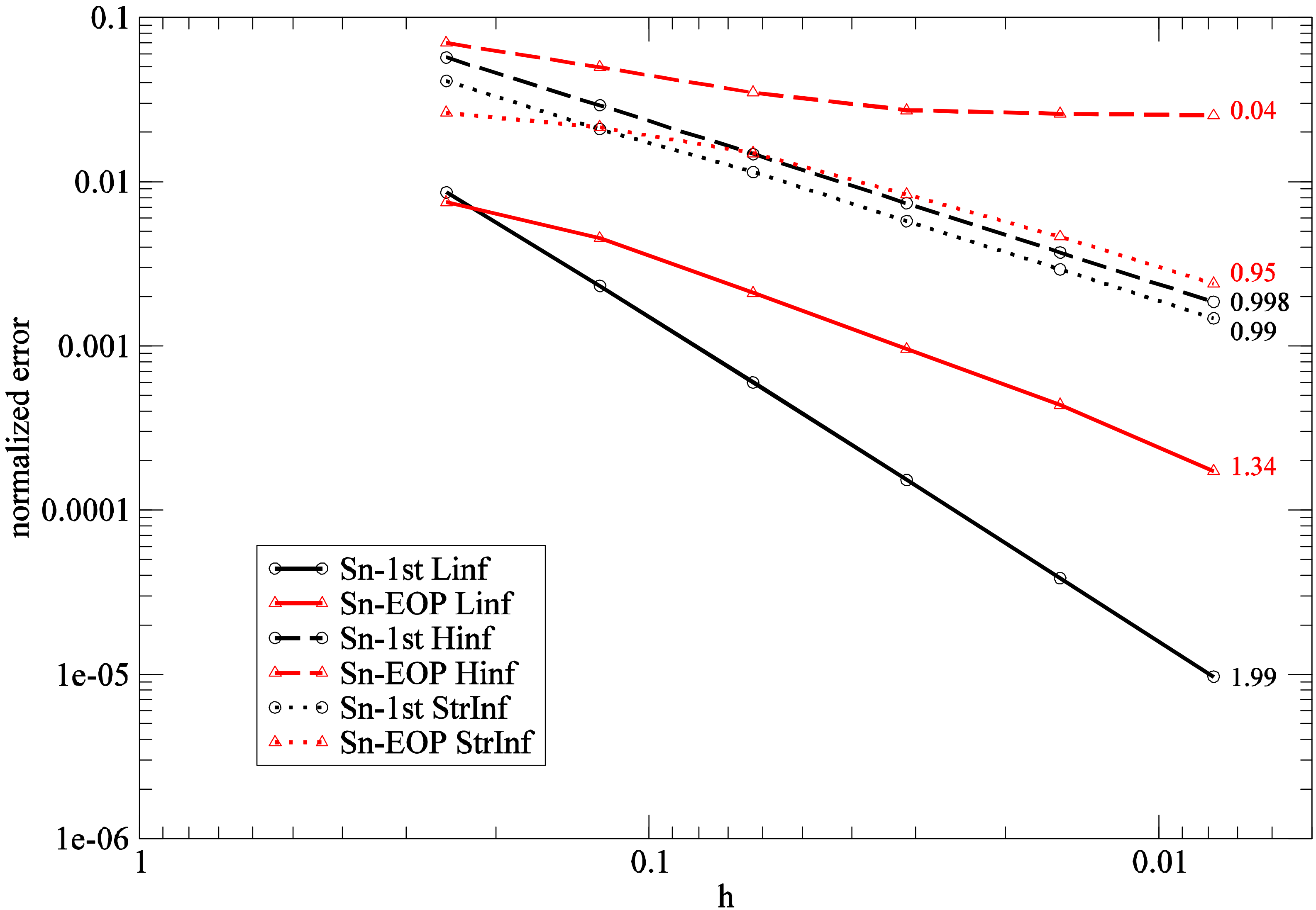
(In)Exactly solvable problems, 3D results
h = 0.25, structured mesh, isotropic scattering

Element Type	Solver	L_{inf} error norm							
		1	x	x^2	xy	x^3	x^2y	xyz	x^2yz
tet4	Sn-1st	1.05×10^{-12}	5.01×10^{-13}	**9.81×10^{-3}**	**5.74×10^{-3}**	**2.53×10^{-2}**	**1.85×10^{-2}**	**8.57×10^{-3}**	**1.62×10^{-2}**
tet10	Sn-1st	3.08×10^{-12}	1.48×10^{-12}	1.14×10^{-12}	1.08×10^{-12}	**9.20×10^{-4}**	**4.36×10^{-4}**	**1.88×10^{-4}**	**7.82×10^{-4}**
hex8	Sn-1st	5.88×10^{-15}	4.88×10^{-15}	**1.33×10^{-2}**	5.22×10^{-15}	**3.46×10^{-2}**	**1.17×10^{-2}**	4.33×10^{-15}	**1.03×10^{-2}**
hex20	Sn-1st	6.11×10^{-14}	3.64×10^{-14}	4.25×10^{-14}	3.97×10^{-14}	**1.13×10^{-3}**	2.32×10^{-14}	2.24×10^{-14}	1.97×10^{-14}
tet4	Sn-EOP	2.87×10^{-13}	2.22×10^{-14}	**1.87×10^{-2}**	**9.42×10^{-3}**	**4.94×10^{-2}**	**2.65×10^{-2}**	**1.82×10^{-2}**	**3.67×10^{-2}**
tet10	Sn-EOP	6.80×10^{-13}	2.26×10^{-13}	1.90×10^{-13}	1.72×10^{-13}	**1.91×10^{-3}**	**6.91×10^{-4}**	**3.26×10^{-4}**	**1.21×10^{-3}**
hex8	Sn-EOP	9.17×10^{-14}	1.03×10^{-14}	**3.29×10^{-2}**	1.39×10^{-14}	**8.99×10^{-2}**	**3.08×10^{-2}**	9.27×10^{-15}	**2.88×10^{-2}**
hex20	Sn-EOP	8.46×10^{-14}	6.06×10^{-14}	5.64×10^{-14}	4.45×10^{-14}	**3.62×10^{-3}**	5.20×10^{-14}	4.76×10^{-14}	3.55×10^{-14}

(In)Exactly solvable problems, 3D results
h = 0.25, unstructured mesh, isotropic scattering

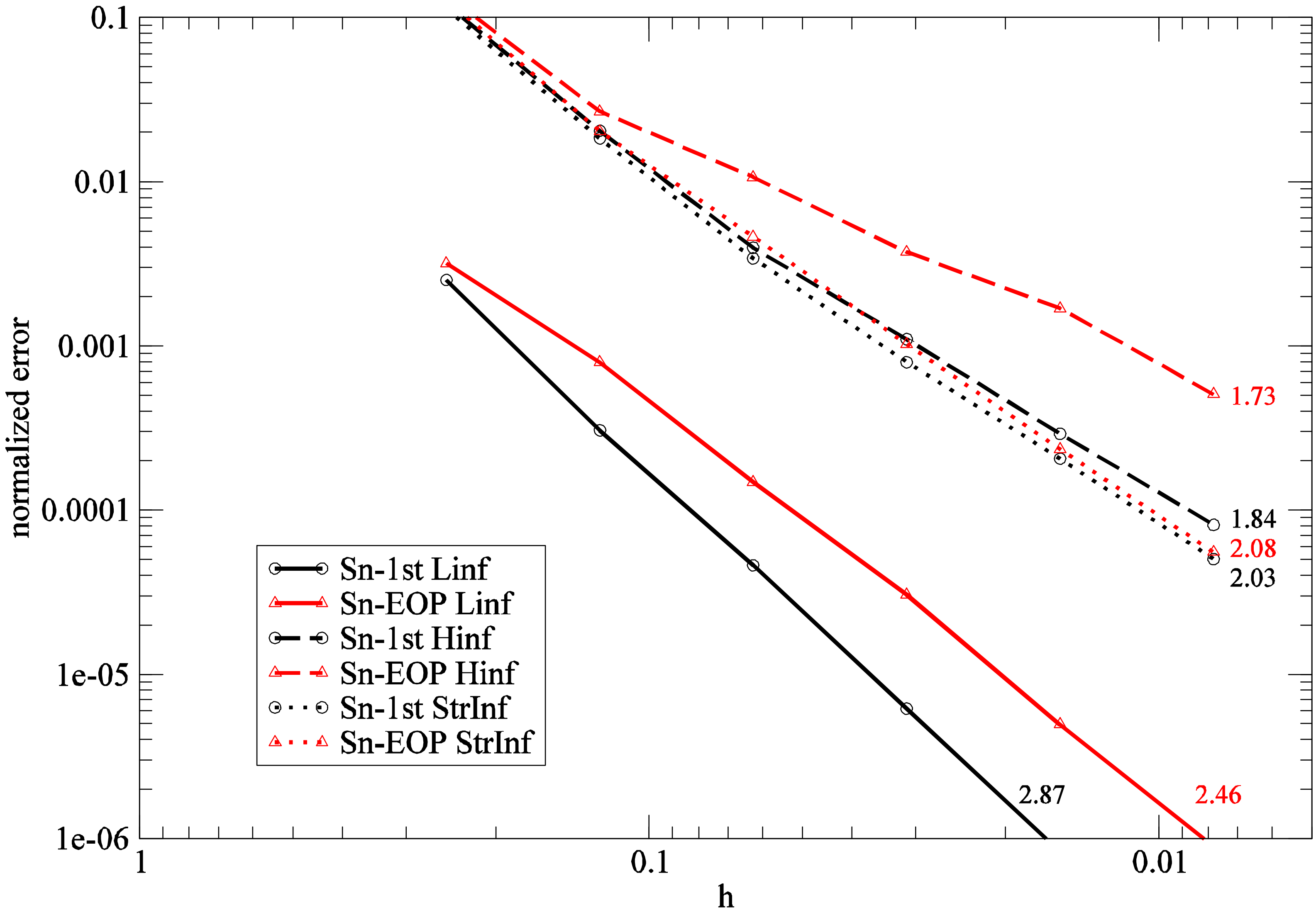
Element Type	Solver	L_{inf} error norm							
		1	x	x^2	xy	x^3	x^2y	xyz	x^2yz
tet4	Sn-1st	1.20×10^{-12}	5.82×10^{-13}	**1.88×10^{-2}**	**9.14×10^{-3}**	**4.89×10^{-2}**	**1.94×10^{-2}**	**1.13×10^{-2}**	**1.87×10^{-2}**
tet10	Sn-1st	4.20×10^{-12}	2.25×10^{-12}	**9.15×10^{-4}**	**6.30×10^{-4}**	**2.70×10^{-3}**	**1.63×10^{-3}**	**1.03×10^{-3}**	**1.36×10^{-3}**
hex8	Sn-1st	1.08×10^{-14}	9.77×10^{-15}	**2.35×10^{-2}**	**7.66×10^{-3}**	**6.15×10^{-2}**	**2.08×10^{-2}**	**1.01×10^{-2}**	**1.66×10^{-2}**
hex20	Sn-1st	6.38×10^{-14}	5.60×10^{-14}	**5.27×10^{-3}**	**3.14×10^{-3}**	**1.03×10^{-2}**	**4.20×10^{-3}**	**4.36×10^{-3}**	**5.26×10^{-3}**
tet4	Sn-EOP	4.40×10^{-13}	3.31×10^{-13}	**2.33×10^{-2}**	**1.53×10^{-2}**	**5.54×10^{-2}**	**3.24×10^{-2}**	**2.66×10^{-2}**	**4.91×10^{-2}**
tet10	Sn-EOP	1.41×10^{-13}	1.53×10^{-13}	**1.17×10^{-3}**	**9.17×10^{-4}**	**4.09×10^{-3}**	**1.67×10^{-3}**	**1.29×10^{-3}**	**1.94×10^{-3}**
hex8	Sn-EOP	3.55×10^{-13}	2.88×10^{-13}	**4.79×10^{-2}**	**9.16×10^{-3}**	**1.25×10^{-1}**	**4.55×10^{-2}**	**1.30×10^{-2}**	**5.14×10^{-2}**
hex20	Sn-EOP	1.10×10^{-13}	6.18×10^{-14}	**6.86×10^{-3}**	**6.07×10^{-3}**	**1.62×10^{-2}**	**6.78×10^{-3}**	**5.85×10^{-3}**	**7.28×10^{-3}**

Error metrics for tri3 meshes



Sandia National Laboratories

Error metrics for tri6 meshes



Why have we done all this?

The Sceptre code, of course, is completely bug-free. It always has been. Our testing is merely to demonstrate that fact to skeptics. ☺

Hypothetically speaking, though, if there had been bugs/issues with Sceptre…

…here's a partial list of what we would have found.

- 2^{nd}-order methods have inherent degradation of convergence order
- Element quadratures lose accuracy when Jacobians are non-constant, but do not affect convergence order
- MMS test generation incorrectly calculated gradients
- Inconsistent definitions of face rotations/mirroring
- Some solvers used 0 rather than previous iterate solutions for outer iteration
- Some solvers used incorrect scattering cross section functor

Future verification work

- Multigroup treatment
- Angular refinement (p-refinement)
- Discontinuous boundary conditions or solutions
- Reflecting/periodic boundary conditions

Conclusions

- Verification of transport codes with structured meshes is quite valuable and has been previously reported
- Verification with unstructured meshes introduces additional complexities
- We have successfully applied numerous verification tests with unstructured meshes to the Sceptre code
- Such testing helps to identify both code and algorithm issues
- The end result is a high and sustainable degree of confidence in the verified code

Sandia National Laboratories