Verification of the Cardinal Multiphysics Solver for 1-D Coupled Heat Transfer and Neutron Transport

Lewis I. Gross[1], April J. Novak[2,3], Patrick Shriwise[3,1], and Paul P.H. Wilson[1]

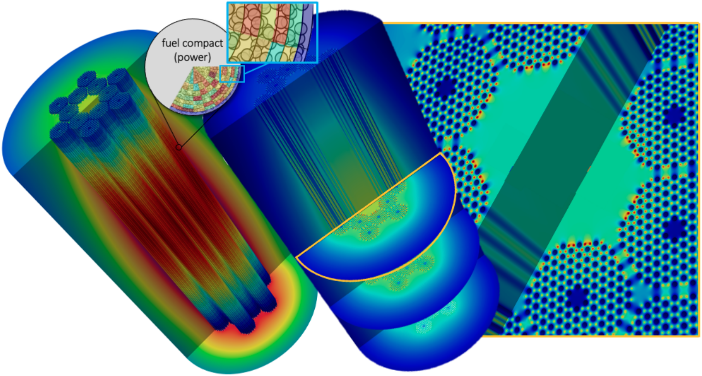
[1] University of Wisconsin-Madison [2] University of Illinois Urbana-Champaign [3] Argonne National Lab

August 15, 2022

WISCONSIN

Outline

Modern Multiphysics Simulation and the Importance of V&V



- Full-core multiphysics model of an HTGR using Cardinal: OpenMC power (left) and MOOSE solid temperature (right) [1].

Modern Multiphysics Simulation and the Importance of V&V

Modern Multiphysics Simulation and the Importance of V&V

- Many new emerging multiphysics tools.

Modern Multiphysics Simulation and the Importance of V&V

- Many new emerging multiphysics tools.
- All software tools require Verification and Validation (V&V).

Modern Multiphysics Simulation and the Importance of V&V

- Many new emerging multiphysics tools.
- All software tools require Verification and Validation (V&V).
- Analytical benchmarks allow measurement of true error.

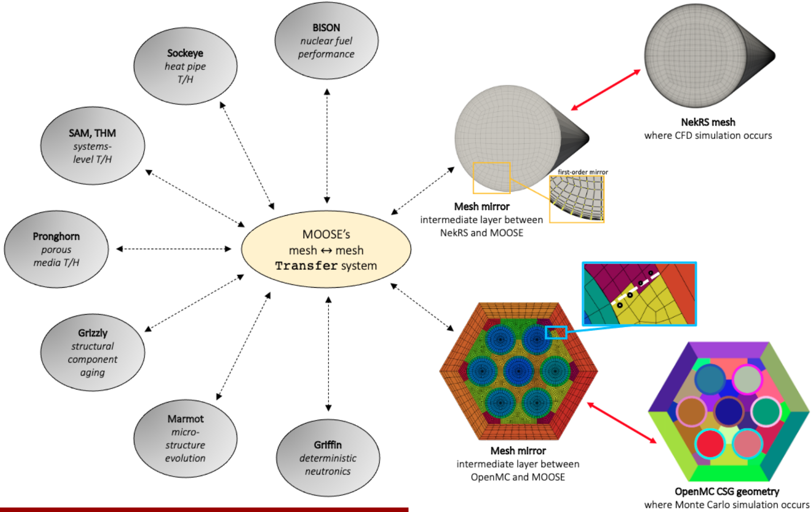
Modern Multiphysics Simulation and the Importance of V&V

- Many new emerging multiphysics tools.
- All software tools require Verification and Validation (V&V).
- Analytical benchmarks allow measurement of true error.
- Greisheimer and Kooreman presented a 1-D analytical benchmark featuring coupled heat transfer and neutron transport [2].

Modern Multiphysics Simulation and the Importance of V&V

- Many new emerging multiphysics tools.
- All software tools require Verification and Validation (V&V).
- Analytical benchmarks allow measurement of true error.
- Greisheimer and Kooreman presented a 1-D analytical benchmark featuring coupled heat transfer and neutron transport [2].
- Cardinal [1] our software choice to model this benchmark couples OpenMC [3] neutronics and NekRS [4] CFD into the MOOSE framework [5].

Cardinal's connection to the MOOSE Framework



Analytical Benchmark

Analytical Benchmark

- This analytical benchmark includes

Analytical Benchmark

- This analytical benchmark includes
  - S_2 transport: particles restricted to the $\pm x$ direction.

Analytical Benchmark

- This analytical benchmark includes
  - S_2 transport: particles restricted to the $\pm$x direction.
  - Doppler-broadening

$$\sigma_t(x) = \sigma_{t,0}\sqrt{\frac{T_0}{T(x)}} \tag{1}$$

Analytical Benchmark

- This analytical benchmark includes
  - S_2 transport: particles restricted to the $\pm$x direction.
  - Doppler-broadening

$$\sigma_t(x) = \sigma_{t,0}\sqrt{\frac{T_0}{T(x)}} \tag{1}$$

  - 1-D thermal expansion

$$\rho(x) = \rho_0\sqrt{\frac{T_0}{T(x)}} \tag{2}$$

Analytical Benchmark

- This analytical benchmark includes
  - S_2 transport: particles restricted to the $\pm$x direction.
  - Doppler-broadening

$$\sigma_t(x) = \sigma_{t,0}\sqrt{\frac{T_0}{T(x)}} \quad (1)$$

  - 1-D thermal expansion

$$\rho(x) = \rho_0\sqrt{\frac{T_0}{T(x)}} \quad (2)$$

  - Linear temperature dependence for thermal conductivity

$$\kappa(x) = \kappa_0 T(x) \quad (3)$$

Analytical Benchmark

- This analytical benchmark includes
  - S_2 transport: particles restricted to the $\pm$x direction.
  - Doppler-broadening

$$\sigma_t(x) = \sigma_{t,0}\sqrt{\frac{T_0}{T(x)}} \tag{1}$$

  - 1-D thermal expansion

$$\rho(x) = \rho_0\sqrt{\frac{T_0}{T(x)}} \tag{2}$$

  - Linear temperature dependence for thermal conductivity

$$\kappa(x) = \kappa_0 T(x) \tag{3}$$

- Using (1) and (2) gives a Doppler-broadened, macroscopic, total cross section that accounts for changes in density due to temperature as

$$\Sigma_t(x) = \frac{\rho_0\sigma_{t,0}N_A}{A}\frac{T_0}{T(x)} \equiv \Sigma_{t,0}\frac{T_0}{T(x)} \tag{4}$$

- where $\sigma_{t,0}$ is the total microscopic cross section at T_0, N_A is Avogadro's number, and A is the mass number of the medium.

Analytical Benchmark

- Based on 1-D S_2 transport, the neutron flux $\phi(x)$ is governed by

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right]+\Sigma_t(x)\left(\lambda-1\right)\phi(x)=0 \tag{5}$$

- with $\lambda \equiv \left(\frac{1}{k_{eff}}\frac{\nu\Sigma_f}{\Sigma_t}+\frac{\Sigma_s}{\Sigma_t}\right)$ [2].

Analytical Benchmark

- Based on 1-D S_2 transport, the neutron flux $\phi(x)$ is governed by

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right]+\Sigma_t(x)\,(\lambda-1)\,\phi(x)=0 \tag{5}$$

- with $\lambda \equiv (\frac{1}{k_{eff}}\frac{\nu\Sigma_f}{\Sigma_t}+\frac{\Sigma_s}{\Sigma_t})$ [2].
- The conduction equation governs energy conservation in the slab:

$$\frac{d}{dx}\left[\kappa(T)\frac{dT(x)}{dx}\right]+q\Sigma_t(x)\phi(x)=0 \qquad \text{WITH}$$

$$-\kappa(T)\frac{dT}{dx}\bigg|_{\pm\frac{L}{2}}=\pm h\left[T(\pm\frac{L}{2})-T_0\right] \tag{6}$$

- where κ is the thermal conductivity, q is the energy released **per reaction**, Σ_t is the total macroscopic cross section, and h is the heat transfer coefficient.

System Domain, Differential Equations, and Boundary Conditions

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi}{dx}\right] + (\lambda - 1)\Sigma_t(x)\phi(x) = 0$$

$$\frac{d}{dx}\left[T(x)\frac{dT}{dx}\right] + \frac{q\Sigma_{t,0}T_0}{\kappa_0}\frac{\phi(x)}{T(x)} = 0$$

Neutrons leaving do not re-enter

x = - ½L

Convection with heat sink at T_0

Neutrons leaving do not re-enter

x = + ½L

Convection with heat sink at T_0

$$\pm T\left(\pm\frac{L}{2}\right)\left.\frac{dT}{dx}\right|_{\pm\frac{L}{2}} + \frac{h}{\kappa_0}\left[T\left(\pm\frac{L}{2}\right) - T_0\right] = 0$$

$$\pm\left.\frac{d\phi}{dx}\right|_{\pm\frac{L}{2}} + \Sigma_t\left(\pm\frac{L}{2}\right)\phi\left(\pm\frac{L}{2}\right) = 0$$

System Domain, Differential Equations, and Boundary Conditions

Neutrons leaving do not re-enter

$x = -\frac{1}{2}L$

Convection with heat sink at T_0

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi}{dx}\right] + (\lambda - 1)\Sigma_t(x)\phi(x) = 0$$

$$\frac{d}{dx}\left[T(x)\frac{dT}{dx}\right] + \frac{q\Sigma_{t,0}T_0}{\kappa_0}\frac{\phi(x)}{T(x)} = 0$$

Neutrons leaving do not re-enter

$x = +\frac{1}{2}L$

Convection with heat sink at T_0

$$\pm T\left(\pm\frac{L}{2}\right)\frac{dT}{dx}\bigg|_{\pm\frac{L}{2}} + \frac{h}{\kappa_0}\left[T\left(\pm\frac{L}{2}\right) - T_0\right] = 0$$

$$\pm\frac{d\phi}{dx}\bigg|_{\pm\frac{L}{2}} + \Sigma_t\left(\pm\frac{L}{2}\right)\phi\left(\pm\frac{L}{2}\right) = 0$$

- The fundamental assumption (ansatz) of [2]: $T(x) = f\phi(x)$.

System Domain, Differential Equations, and Boundary Conditions

Neutrons leaving do not re-enter

x = - ½L

Convection with heat sink at T_0

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi}{dx}\right]+(\lambda-1)\Sigma_t(x)\phi(x)=0$$

$$\frac{d}{dx}\left[T(x)\frac{dT}{dx}\right]+\frac{q\Sigma_{t,0}T_0}{\kappa_0}\frac{\phi(x)}{T(x)}=0$$

Neutrons leaving do not re-enter

x = + ½L

Convection with heat sink at T_0

$$\pm T\left(\pm\frac{L}{2}\right)\frac{dT}{dx}\bigg|_{\pm\frac{L}{2}}+\frac{h}{\kappa_0}\left[T\left(\pm\frac{L}{2}\right)-T_0\right]=0$$

$$\pm\frac{d\phi}{dx}\bigg|_{\pm\frac{L}{2}}+\Sigma_t\left(\pm\frac{L}{2}\right)\phi\left(\pm\frac{L}{2}\right)=0$$

- The fundamental assumption (ansatz) of [2]: $T(x)=f\phi(x)$.
- This imposes two constraints that determine h and $\sigma_{t,0}$. The solution:

$$\phi(x)=\phi(0)\sqrt{1-\frac{(\lambda-1)P^2x^2}{L^2q^2\phi^2(0)}} \tag{7}$$

- where P is the slab power and L is the slab equilibrium length.

OpenMC Model

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.
  - Y and Z-planes reflective BCs to simulate infiniteness.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.
  - Y and Z-planes reflective BCs to simulate infiniteness.
- Macroscopic XS library via OpenMC's `XSData` class. It uses the macroscopic XS to account for changes in the XS due to temperature-density feedback on the equilibrium mesh instead of modeling thermal expansion via mesh deformation.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.
  - Y and Z-planes reflective BCs to simulate infiniteness.
- Macroscopic XS library via OpenMC's `XSData` class. It uses the macroscopic XS to account for changes in the XS due to temperature-density feedback on the equilibrium mesh instead of modeling thermal expansion via mesh deformation.
- Mesh deformation is possible in Cardinal via DAGMC [6].

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.
  - Y and Z-planes reflective BCs to simulate infiniteness.
- Macroscopic XS library via OpenMC's `XSData` class. It uses the macroscopic XS to account for changes in the XS due to temperature-density feedback on the equilibrium mesh instead of modeling thermal expansion via mesh deformation.
- Mesh deformation is possible in Cardinal via DAGMC [6].
- XS library has data for every integer temperature between 308 K and 358 K and rounds to nearest for a lookup between two data points.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.
  - Y and Z-planes reflective BCs to simulate infiniteness.
- Macroscopic XS library via OpenMC's `XSData` class. It uses the macroscopic XS to account for changes in the XS due to temperature-density feedback on the equilibrium mesh instead of modeling thermal expansion via mesh deformation.
- Mesh deformation is possible in Cardinal via DAGMC [6].
- XS library has data for every integer temperature between 308 K and 358 K and rounds to nearest for a lookup between two data points.
- XS library uses one energy group from 0 to 20 MeV.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.
  - Y and Z-planes reflective BCs to simulate infiniteness.
- Macroscopic XS library via OpenMC's `XSData` class. It uses the macroscopic XS to account for changes in the XS due to temperature-density feedback on the equilibrium mesh instead of modeling thermal expansion via mesh deformation.
- Mesh deformation is possible in Cardinal via DAGMC [6].
- XS library has data for every integer temperature between 308 K and 358 K and rounds to nearest for a lookup between two data points.
- XS library uses one energy group from 0 to 20 MeV.
- S_2 patch to restrict particle birth direction and scattering direction to only $\pm$x.

OpenMC Model

- Slab geometry divided into N cells, $N = [5, 10, 25, 50, 100, 250, 500, 1000]$.
- Equilibrium length $L = 106.47$ cm in the x-direction, 1 cm in the y-direction, 1 cm in the z-direction.
  - Internal X-planes use transmissive BCs and boundary X-planes use vacuum BCs.
  - Y and Z-planes reflective BCs to simulate infiniteness.
- Macroscopic XS library via OpenMC's `XSData` class. It uses the macroscopic XS to account for changes in the XS due to temperature-density feedback on the equilibrium mesh instead of modeling thermal expansion via mesh deformation.
- Mesh deformation is possible in Cardinal via DAGMC [6].
- XS library has data for every integer temperature between 308 K and 358 K and rounds to nearest for a lookup between two data points.
- XS library uses one energy group from 0 to 20 MeV.
- S_2 patch to restrict particle birth direction and scattering direction to only $\pm$x.
- Tallies flux, kappa-fission heating rate, and k-eigenvalue.

MOOSE Heat Conduction Model

MOOSE Heat Conduction Model

- Mesh with identical dimensions as OpenMC model. Allows 1:1 feedback between single physics solves.

MOOSE Heat Conduction Model

- Mesh with identical dimensions as OpenMC model. Allows 1:1 feedback between single physics solves.
- MOOSE accepts heating rate in each mesh element from OpenMC and solves for temperature distribution.

MOOSE Heat Conduction Model

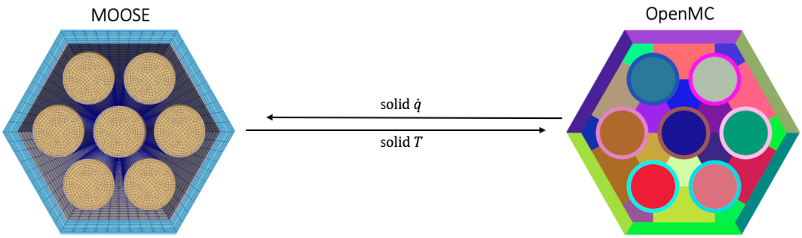
- Mesh with identical dimensions as OpenMC model. Allows 1:1 feedback between single physics solves.
- MOOSE accepts heating rate in each mesh element from OpenMC and solves for temperature distribution.
- Convective boundary conditions at end points with heat sink at $T_0 = 293$ K.

MOOSE Heat Conduction Model

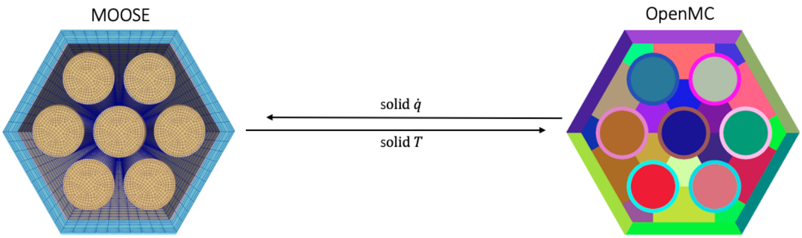
- Mesh with identical dimensions as OpenMC model. Allows 1:1 feedback between single physics solves.
- MOOSE accepts heating rate in each mesh element from OpenMC and solves for temperature distribution.
- Convective boundary conditions at end points with heat sink at $T_0 = 293$ K.
- Jacobi Free Newton Krylov solver: 10^{-7} absolute tolerance and 10^{-9} relative tolerance.

Coupling, Data Mapping, and Convergence Criteria

Coupling, Data Mapping, and Convergence Criteria

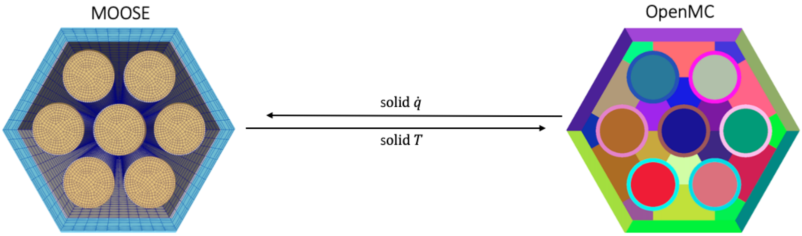


Coupling, Data Mapping, and Convergence Criteria



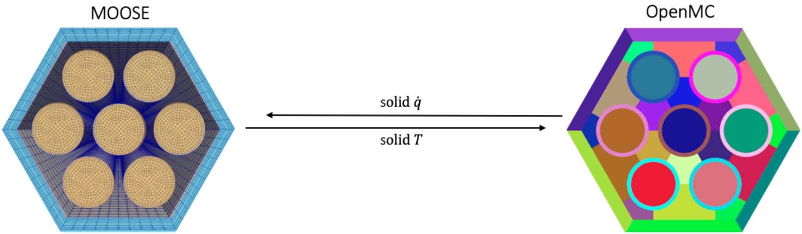
- 200 Picard Iterations. Did not use steady state detection, though MOOSE has this capability. Robbins-Monro relaxation assisted tally statistics [7].

Coupling, Data Mapping, and Convergence Criteria



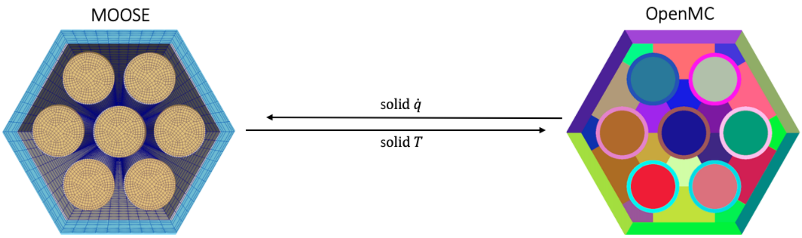
- 200 Picard Iterations. Did not use steady state detection, though MOOSE has this capability. Robbins-Monro relaxation assisted tally statistics [7].
- *k*-eigenvalue simulation used 50,000 particles per batch, 50 inactive batches and 100 active batches.

Coupling, Data Mapping, and Convergence Criteria



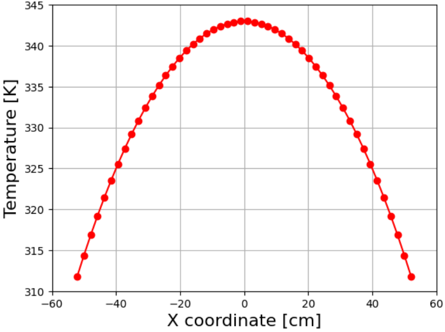
- 200 Picard Iterations. Did not use steady state detection, though MOOSE has this capability. Robbins-Monro relaxation assisted tally statistics [7].
- k-eigenvalue simulation used 50,000 particles per batch, 50 inactive batches and 100 active batches.
    - Shannon Entropy study confirmed this sufficient following criteria from [8].

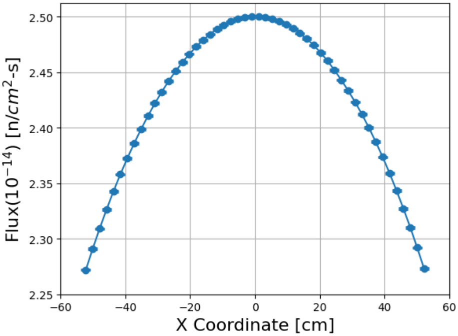
Coupling, Data Mapping, and Convergence Criteria



- 200 Picard Iterations. Did not use steady state detection, though MOOSE has this capability. Robbins-Monro relaxation assisted tally statistics [7].
- k-eigenvalue simulation used 50,000 particles per batch, 50 inactive batches and 100 active batches.
  - Shannon Entropy study confirmed this sufficient following criteria from [8].
- Final transport solve with converged temperature used 250,000 particles per batch.

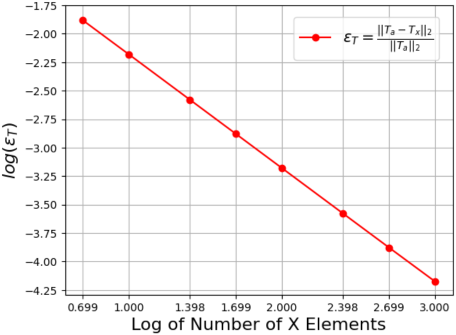
Outputs and Comparisons

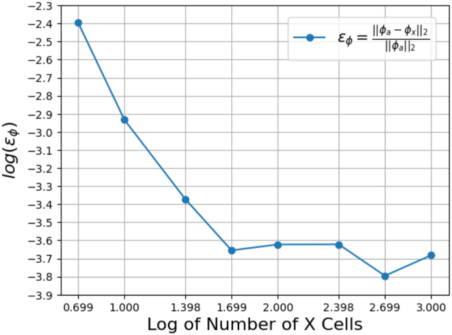




- Numerical solutions for 50 mesh elements. On the right, error bars show the relative error of the flux, which are nearly smaller than the circular marker sizes.

Solution L_2 Error Norms





- Error norms as a function of heat conduction mesh element count and OpenMC cell count, respectively.

Eigenvalue comparisons across each spatial discretization

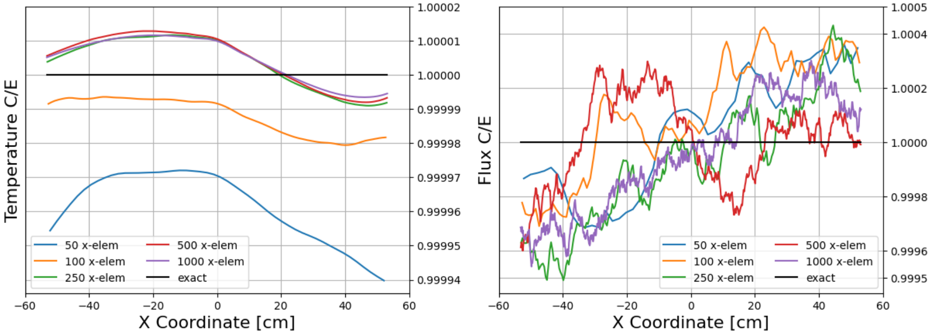
Resolution	k_{eff}	(numerical - analytical) [pcm]
analytical	0.29557	-
5	0.29624 $\pm$ 0.00003	67 $\pm$ 3
10	0.29581 $\pm$ 0.00004	24 $\pm$ 4
25	0.29563 $\pm$ 0.00004	6 $\pm$ 4
50	0.29553 $\pm$ 0.00004	-4 $\pm$ 4
100	0.29557 $\pm$ 0.00003	0 $\pm$ 3
250	0.29561 $\pm$ 0.00004	4 $\pm$ 4
500	0.29561 $\pm$ 0.00004	4 $\pm$ 4
1000	0.29558 $\pm$ 0.00004	1 $\pm$ 4

Eigenvalue comparisons across each spatial discretization

Resolution	k_{eff}	(numerical - analytical) [pcm]
analytical	0.29557	-
5	0.29624 ± 0.00003	67 ± 3
10	0.29581 ± 0.00004	24 ± 4
25	0.29563 ± 0.00004	6 ± 4
50	0.29553 ± 0.00004	-4 ± 4
100	0.29557 ± 0.00003	0 ± 3
250	0.29561 ± 0.00004	4 ± 4
500	0.29561 ± 0.00004	4 ± 4
1000	0.29558 ± 0.00004	1 ± 4

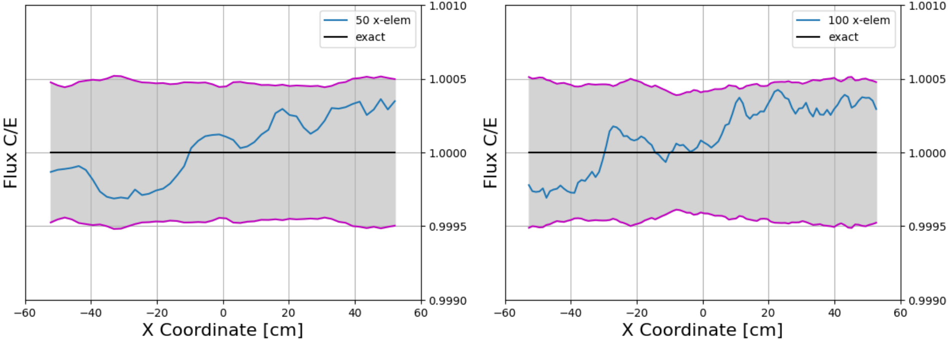
- k_{eff} is a system-wide parameter, so it converges much faster than flux and is not as dependent on number of cells.

Computed to Expected Ratios



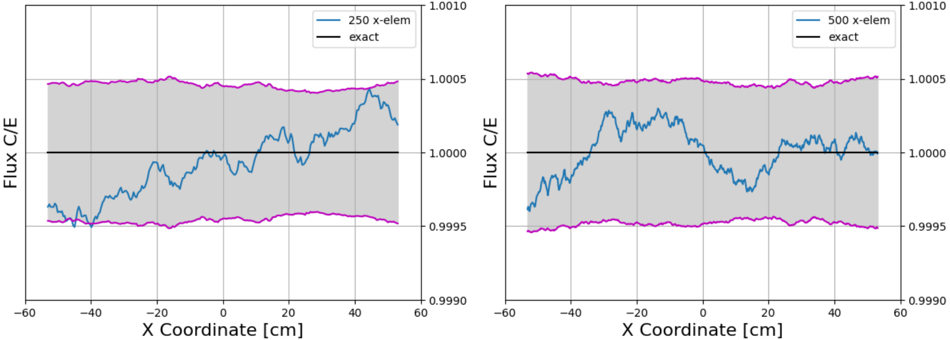
- C/E for fine cases ($N = 50, 100, 250, 500, 1000$). Note the scales of the y-axes - the temperature is everywhere being predicted to within 0.006% and flux is everywhere being predicted to within 0.05%.

Individual Flux C/E with 2σ Error Bars for Fine Cases



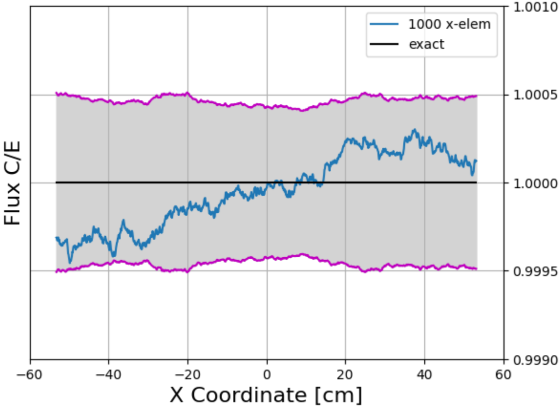
- C/E in blue with 2σ error bars (gray bounded by purple). 50 and 100 cells.

Individual Flux C/E with 2σ Error Bars for Fine Cases



- C/E in blue with 2σ error bars (gray bounded by purple). 250 and 500 cells

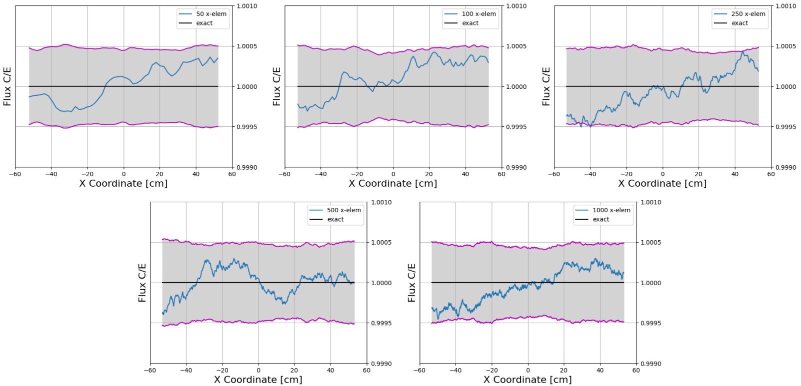
Individual Flux C/E with 2σ Error Bars for Fine Cases



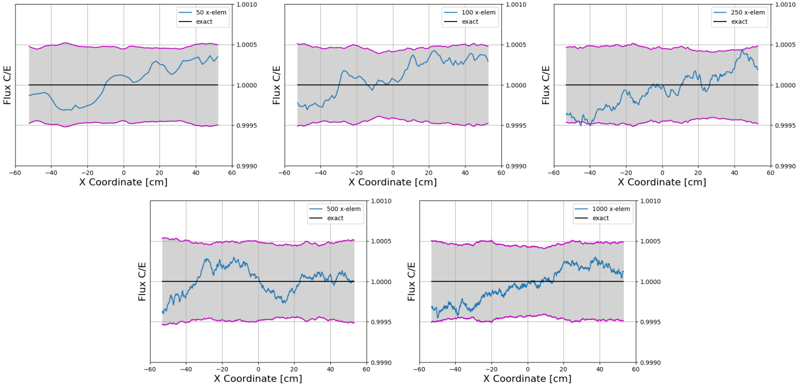
- C/E in blue with 2σ error bars (gray bounded by purple). 1000 cells.

Why do the error bars appear on the same order despite cell refinement?

Why do the error bars appear on the same order despite cell refinement?

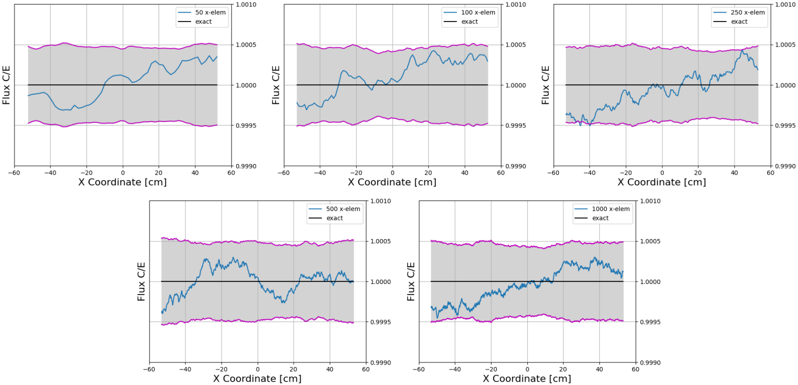


Why do the error bars appear on the same order despite cell refinement?



- MFP $\in [239, 278]$ cm ($L = 106.47$ cm). Spatially uniform birth distribution.

Why do the error bars appear on the same order despite cell refinement?



- MFP $\in [239, 278]$ cm ($L = 106.47$ cm). Spatially uniform birth distribution.
- Nearly all points fall within 2σ (95% Confidence Interval), meaning that Cardinal is computing the correct flux within statistical uncertainty.

Acknowledgements

- Benchmark authors: David P. Greisheimer and Gabriel Kooreman
- Co-authors: April J. Novak, Patrick Shriwise, Paul P.H. Wilson
- OpenMC, Cardinal, and MOOSE teams!

Questions?

- OpenMC website: https://openmc.org/
- OpenMC repository: https://github.com/openmc-dev/openmc
- Cardinal website: https://cardinal.cels.anl.gov
- Cardinal repository: https://github.com/neams-th-coe/cardinal.git
- MOOSE website:https://mooseframework.inl.gov/
- MOOSE repository: https://github.com/idaholab/moose

Bibliography

[1] A.J. Novak et al. "Coupled Monte Carlo and Thermal-Fluid Modeling of High Temperature Gas Reactors Using Cardinal". In: *Annals of Nuclear Energy* 177 (2022), p. 109310. DOI: 10.1016/j.anucene.2022.109310.

[2] D.P. Griesheimer and G. Kooreman. "Analytical Benchmark Solution for 1-D Neutron Transport Coupled with Thermal Conduction and Material Expansion". In: *Proceedings of M&C*. Pittsburgh, Pennsylvania, 2022.

[3] P.K. Romano et al. "OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development". In: *Annals of Nuclear Energy* 82 (2015), pp. 90–97. DOI: 10.1016/j.anucene.2014.07.048.

[4] P. Fischer et al. *NekRS, a GPU-Accelerated Spectral Element Navier-Stokes Solver*. arXiv:2104.05829. Apr. 2021.

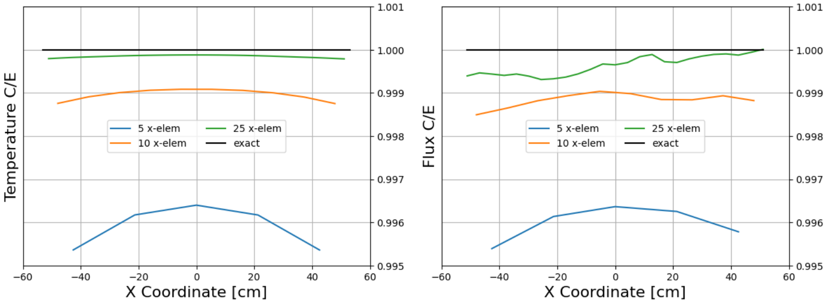
[5] Alexander D. Lindsay et al. "2.0 - MOOSE: Enabling massively parallel multiphysics simulation". In: *SoftwareX* 20 (2022), p. 101202. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2022.101202.

[6] A.J. Novak et al. "Multiphysics Coupling of OpenMC CAD-Based Transport to MOOSE using Cardinal and Aurora". In: *Proceedings of M&C*. Niagara Falls, Ontario, Canada, 2023.

[7] J. Dufek and W. Gudowski. “Stochastic Approximation for Monte Carlo Calculation of Steady-State Conditions in Thermal Reactors”. In: *Nuclear Science and Engineering* 152 (2006), pp. 274–283. DOI: 10.13182/NSE06-2.

[8] F.B. Brown. “On the Use of Shannon Entropy of the Fission Distribution for Assessing Convergence of Monte Carlo Criticality Calculations”. In: *Proceedings of PHYSOR*. Vancouver, British Columbia, Canada, 2006.

Coarse C/E results



- C/E for coarse cases ($N = 5, 10, 25$). The coarse cases' errors are a few orders of magnitude larger than the fine cases. A significant improvement in agreement can be seen between each coarse case.

Ensuring Validity of the Fundamental Ansatz

- Taking the heat conduction ODE

$$\frac{d}{dx}\left[\kappa(T(x))\frac{dT(x)}{dx}\right]+q\Sigma_t(x)\phi(x)=0 \tag{8}$$

- and using the thermal conductivity and cross section temperature dependence

$$\frac{d}{dx}\left[\kappa_0 T(x)\frac{dT(x)}{dx}\right]+q\Sigma_{t,0}\frac{T_0}{T(x)}\phi(x)=0 \tag{9}$$

- Taking the neutron transport ODE

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right]+\Sigma_t(x)\left(\lambda-1\right)\phi(x)=0 \tag{10}$$

- and inserting cross section temperature dependence gives

$$\frac{d}{dx}\left[\frac{T(x)}{\Sigma_{t,0}T_0}\frac{d\phi(x)}{dx}\right]+\Sigma_{t,0}\frac{T_0}{T(x)}\left(\lambda-1\right)\phi(x)=0 \tag{11}$$

- these two equations are very close, and after some re-arranging, they look even closer

$$\frac{d}{dx}\left[T(x)\frac{dT(x)}{dx}\right]+\frac{q\Sigma_{t,0}}{\kappa_0}\frac{T_0}{T(x)}\phi(x)=0 \qquad \text{AND}$$
$$\frac{d}{dx}\left[\frac{T(x)}{\Sigma_{t,0}T_0}\frac{d\phi(x)}{dx}\right]+\Sigma_{t,0}\frac{T_0}{T(x)}(\lambda-1)\phi(x)=0 \quad (12)$$

- Applying the ansatz $T(x)=f\phi(x)$ gives

$$\frac{d}{dx}\left[f^2\phi(x)\frac{d\phi(x)}{dx}\right]+\frac{q\Sigma_{t,0}}{\kappa_0}\frac{T_0}{f}=0 \qquad \text{AND}$$
$$\frac{d}{dx}\left[\frac{f\phi(x)}{\Sigma_{t,0}T_0}\frac{d\phi(x)}{dx}\right]+\Sigma_{t,0}\frac{T_0}{f}(\lambda-1)=0 \quad (13)$$

$$\frac{d}{dx}\left[\phi(x)\frac{d\phi(x)}{dx}\right]+\frac{q\Sigma_{t,0}}{\kappa_0}\frac{T_0}{f^3}=0 \qquad \text{AND}$$
$$\frac{d}{dx}\left[\phi(x)\frac{d\phi(x)}{dx}\right]+\left(\Sigma_{t,0}\frac{T_0}{f}\right)^2(\lambda-1)=0 \quad (14)$$

Ensuring Validity of the Fundamental Ansatz

- In order to make the ansatz hold, this implies that

$$\frac{q\Sigma_{t,0}}{\kappa_0}\frac{T_0}{f^3} = \left(\Sigma_{t,0}\frac{T_0}{f}\right)^2(\lambda - 1) \qquad \text{OR} \qquad \Sigma_{t,0} = \frac{q}{(\lambda-1)\kappa_0 T_0 f} \tag{15}$$

which is a condition for the total cross section based on system parameters.

- A similar process of matching coefficients must be applied to the boundary conditions to gain a condition for the heat transfer coefficient. Though, at this point, realize that

$$\frac{d}{dx}\left[\phi(x)\frac{d\phi(x)}{dx}\right] + \left(\Sigma_{t,0}\frac{T_0}{f}\right)^2(\lambda - 1) = 0 \tag{16}$$

- is a separable ODE that can be solved for an analytical solution. The result for the heat transfer coefficient is given by

$$h\left(\sqrt{\frac{L(\lambda-1)}{\kappa_0 P}} - \frac{2T_0}{P}\right) = 1 \tag{17}$$

- Going from the steady-state, mono-energetic, 1-D neutron transport equation to the ODE that describes neutron transport for this benchmark:

$$\mu\frac{\partial\psi(x,\mu)}{\partial x}+\Sigma_t(x)\psi(x,\mu)=\int_{-1}^{1}\frac{1}{2}\big[\Sigma_s(x)+\frac{\nu\Sigma_f(x)}{k_{eff}}\big]\psi(x,\mu')d\mu' \tag{18}$$

For now, lump the fission term into the scattering cross section to get

$$\mu\frac{\partial\psi(x,\mu)}{\partial x}+\Sigma_t(x)\psi(x,\mu)=\int_{-1}^{1}\frac{1}{2}\Sigma_s(x)\psi(x,\mu')d\mu' \tag{19}$$

Define the scalar flux and the magnitude of current

$$\phi(x)=\int_{-1}^{1}\psi(x,\mu)d\mu \qquad \text{AND} \qquad J(x)=\int_{-1}^{1}\mu\psi(x,\mu)d\mu. \tag{20}$$

Considering S_2 transport means restricting the angular cosine to $\mu=\pm1$:

$$\psi(x,\mu)=\psi(x,-1)\delta(\mu+1)+\psi(x,1)\delta(\mu-1) \tag{21}$$

(sometimes denoted $\psi^+\equiv\psi(x,1)\delta(\mu-1)$ and $\psi^-\equiv\psi(x,-1)\delta(\mu+1)$)

Now carrying out the integral definitions with S_2 quantities gives

$$\phi(x) = \int_{-1}^{1} \left[\psi(x,-1)\delta(\mu+1) + \psi(x,1)\delta(\mu-1)\right] d\mu = \psi(x,-1) + \psi(x,1) \tag{22}$$

and

$$J(x) = \int_{-1}^{1} \mu \left[\psi(x,-1)\delta(\mu+1) + \psi(x,1)\delta(\mu-1)\right] d\mu = \psi(x,1) - \psi(x,-1) \tag{23}$$

Evaluating (19) at $\mu = \pm 1$ gives

$$-\frac{\partial \psi(x,-1)}{\partial x} + \Sigma_t(x)\psi(x,-1) = \frac{1}{2}\Sigma_s(x)\phi(x); \tag{24}$$

$$\frac{\partial \psi(x,1)}{\partial x} + \Sigma_t(x)\psi(x,1) = \frac{1}{2}\Sigma_s(x)\phi(x). \tag{25}$$

Adding (24) and (25) gives

$$-\frac{\partial \psi(x,-1)}{\partial x} + \frac{\partial \psi(x,1)}{\partial x} + \Sigma_t(x)\left(\psi(x,-1) + \psi(x,1)\right) = \Sigma_s(x)\phi(x) \tag{26}$$

Deriving the Benchmark ODE for $\phi(x)$

The results for $\phi(x)$ and $J(x)$ can simplify (26)

$$\frac{dJ(x)}{dx} + \Sigma_t(x)\phi(x) = \Sigma_s(x)\phi(x) \tag{27}$$

Subtracting (24) and (25) gives

$$-\frac{\partial\psi(x,-1)}{dx} - \frac{\partial\psi(x,1)}{dx} + \Sigma_t(x)\left(\psi(x,-1) - \Sigma_t(x)\psi(x,1)\right) = 0 \tag{28}$$

Which can be transformed with similar tricks to

$$\frac{d\phi(x)}{dx} + \Sigma_t(x)J(x) = 0 \qquad \text{OR} \qquad J(x) = -\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx} \tag{29}$$

Since $\frac{dJ(x)}{dx}$ appears in (27), we can take the derivative of both sides of (29) and substitute it in

$$\frac{dJ(x)}{dx} = -\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right] \tag{30}$$

Now the equation that only depends on $\phi(x)$ is given by

$$-\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right] + \Sigma_t(x)\phi(x) = \Sigma_s(x)\phi(x) \tag{31}$$

Deriving the Benchmark ODE for $\phi(x)$

At this point, we "un-lump" the scattering cross section to write out the fission term

$$-\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right]+\Sigma_t(x)\phi(x)=\left[\Sigma_s(x)+\frac{\nu\Sigma_f}{k_{eff}}\right]\phi(x) \tag{32}$$

$$-\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right]+\Sigma_t(x)\left[1-\frac{\Sigma_s(x)+\frac{\nu\Sigma_f}{k_{eff}}}{\Sigma_t}\right]\phi(x)=0 \tag{33}$$

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right]+\Sigma_t(x)\left[\frac{\Sigma_s(x)+\frac{\nu\Sigma_f}{k_{eff}}}{\Sigma_t}-1\right]\phi(x)=0 \tag{34}$$

Now define

$$\lambda\equiv\frac{\Sigma_s(x)+\frac{\nu\Sigma_f}{k_{eff}}}{\Sigma_t} \tag{35}$$

giving the final result:

$$\frac{d}{dx}\left[\frac{1}{\Sigma_t(x)}\frac{d\phi(x)}{dx}\right]+\Sigma_t(x)\left(\lambda-1\right)\phi(x)=0 \tag{36}$$

The next task is to apply boundary conditions so that $\phi(x)$ can be specified. In discrete ordinates with $\mu = \pm 1$ (S_2), we use the vacuum boundary condition. The angular flux for positive angular cosines is zero at the left boundary and is zero for negative angular cosines at the right boundary. Using the previous results for $\phi(x)$ and $J(x)$ at the boundaries gives

$$\phi(x = \frac{L}{2}) = \psi(x = \frac{L}{2}, \mu = -1) + \psi(x = \frac{L}{2}, \mu = 1) \qquad \text{AND}$$
$$\phi(x = -\frac{L}{2}) = \psi(x = -\frac{L}{2}, \mu = -1) + \psi(x = -\frac{L}{2}, \mu = 1) \quad (37)$$

and

$$J(x = \frac{L}{2}) = -\psi(x = \frac{L}{2}, -1) + \psi(x = \frac{L}{2}, 1) \qquad \text{AND}$$
$$J(x = -\frac{L}{2}) = -\psi(x = -\frac{L}{2}, -1) + \psi(x = -\frac{L}{2}, 1) \quad (38)$$

Now, terms can be crossed out due to vacuum boundaries. This gives that

$$\phi(x = \frac{L}{2}) = \psi(x = \frac{L}{2}, \mu = 1) \qquad \text{AND} \qquad \phi(x = -\frac{L}{2}) = \psi(x = -\frac{L}{2}, \mu = -1) \tag{39}$$

$$J(x = \frac{L}{2}) = \psi(x = \frac{L}{2}, \mu = 1) \qquad \text{AND} \qquad J(x = -\frac{L}{2}) = -\psi(x = -\frac{L}{2}, \mu = -1) \tag{40}$$

Using this with (29) gives the desired boundary conditions

$$J(x = \pm\frac{L}{2}) = \pm\phi(x = \pm\frac{L}{2}) \tag{41}$$

And the boundary conditions of interest are now

$$\left.\frac{d\phi}{dx}\right|_{x=\pm\frac{L}{2}} \pm \Sigma_t(x = \pm\frac{L}{2})\phi(x = \pm\frac{L}{2}) = 0 \tag{42}$$