INL/CON-24-77668-Revision-0

Changing the World's Energy Future

Coarse Mesh Finite Difference Acceleration for Pebble Tracking Transport in Griffin

April 2024

Yaqi Wang, Namjae Choi, Joshua Thomas Hanophy

Idaho National Laboratory

INL/CON-24-77668-Revision-0

Coarse Mesh Finite Difference Acceleration for Pebble Tracking Transport in Griffin

Yaqi Wang, Namjae Choi, Joshua Thomas Hanophy

April 2024

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Coarse Mesh Finite Difference Acceleration for Pebble Tracking Transport in Griffin

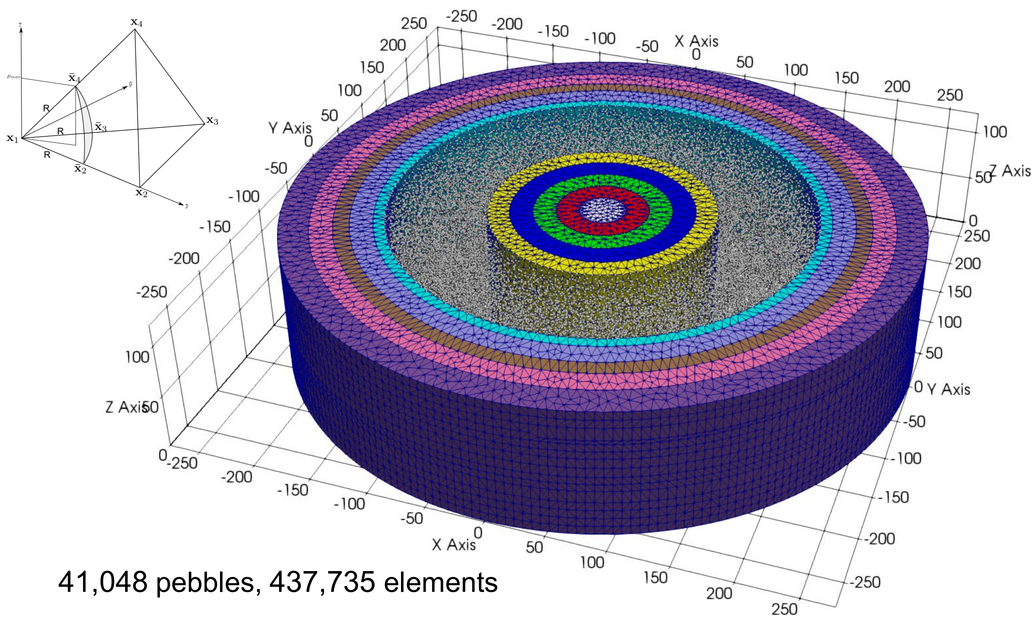
Yaqi Wang, Joshua Hanophy, Namjae Choi

International Conference on Physics of Reactors (PHYSOR 2024)

Outline

- What is PTT (pebble packing transport)?
- To enable the transport solver (CMFD accelerated Richardson iteration) in Griffin for PTT with DFEM (discontinuous finite element method) –SN (discrete ordinates method):
  - Residual evaluation,
  - Transport update,
  - CMFD acceleration.
- Numerical results of the transport solver.

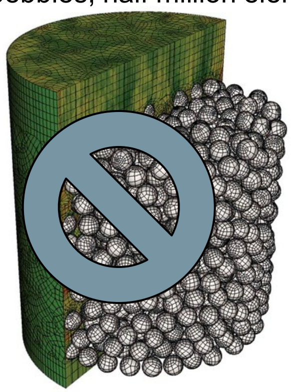
What is PTT?



41,048 pebbles, 437,735 elements

- PTT stands for pebble tracking transport.
- First implemented with DFEM-SN in Rattlesnake at the end of FY 2017 and published at PHYSOR2018.
- In the pebble packing region, a mesh node represents a pebble. A tetrahedron element is formed with five pieces: four pebbles on the vertices and one gap in-between.
- Cross sections are homogenized on pebbles. A tetrahedron element may have five sets of macroscopic cross sections. Pebbles are assumed as a perfect sphere.

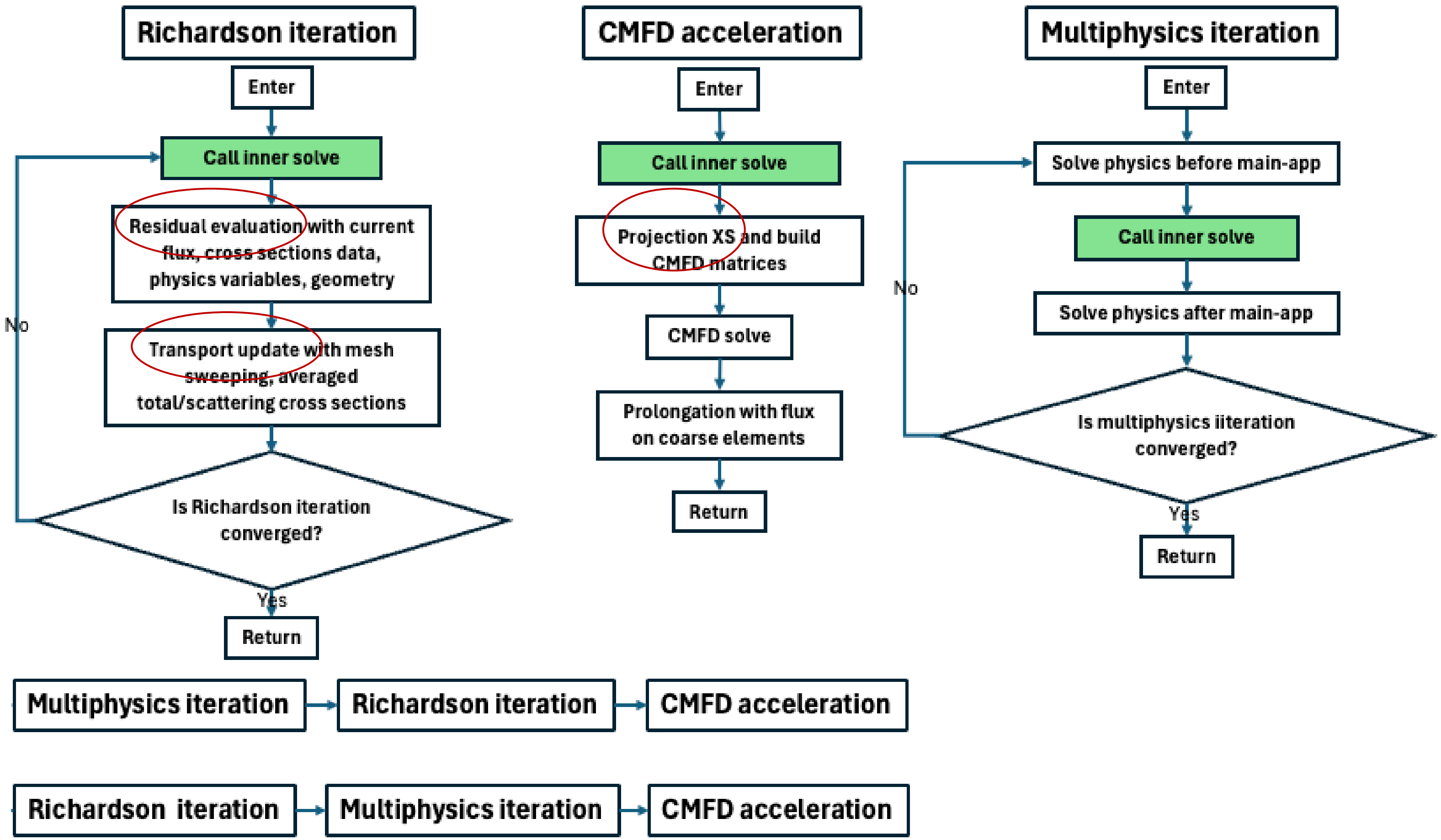
1568 pebbles, half million elements



- To remove the region homogenization of the pebble-bed domain in traditional methods.
- To enable direct transport calculations with pebble tracking without meshing individual pebbles.
- About 1% error on powers of individual pebbles can be achieved with pebble-homogenized broad cross sections generated with MC.

CMFD (Coarse Mesh Finite Difference) Accelerated Richardson Iteration in Griffin

- The algorithm for CMFD accelerated Richardson iteration with multiphysics:

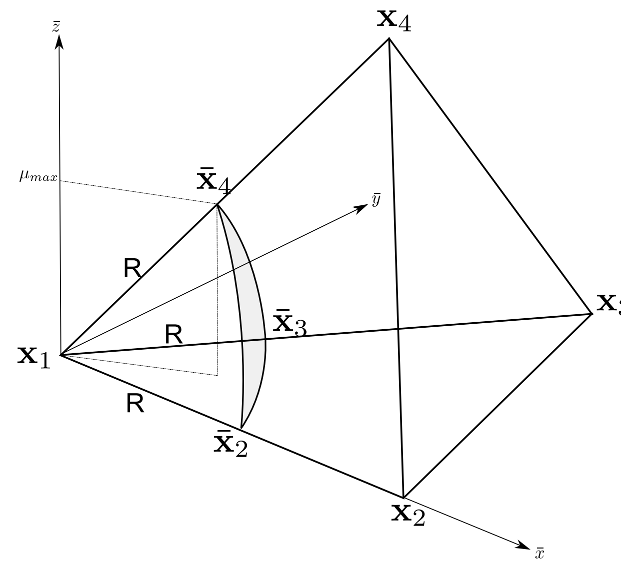


Element Mass Matrix for PTT

$$(\mathbf{\Psi}^*, \mathbf{\Sigma}_t \mathbf{\Psi})_{\mathcal{D}} = \sum_{g=1}^{G} \sum_{e \in \mathcal{D}} \sum_{m=1}^{M} w_m \sum_{i=1}^{N(e)} \Psi^*_{g,e,m,i} \sum_{j=1}^{N(e)} \Psi_{g,e,m,j} \boxed{\int_e \Sigma_{t,g}(x) b_i(x) b_j(x) dx}$$

- $\int_e \Sigma_{t,g}(x) b_i(x) b_j(x) dx = \sum_{k=1}^{4} \Sigma_{t,g,k} \int_{P_k} b_i(x) b_j(x) dx + \Sigma_{t,g,0} \left(\int_e b_i(x) b_j(x) dx - \sum_{k=1}^{4} \int_{P_k} b_i(x) b_j(x) dx \right)$

- How the integration is done on the partial pebbles around vertices can be found in the paper:
  - Shape functions are polynomials;
  - $\Sigma_t(x)$ is assumed constant within a pebble and in the gap;
  - Elemental matrices ONLY depend on mesh and are pre-calculated and stored.
- We can do similar treatment to the scattering/fission terms.



Residual Evaluation and Transport Update with DFEM-SN

- The final algebraic equation:
  - $L\Psi = S\Psi + \frac{1}{k}F\Psi$
- Residual of a solution Ψ is defined as
  - $R(\Psi) \equiv S\Psi + \frac{1}{k}F\Psi - L\Psi$ (implemented previously)
- Transport update
  - $\Psi = \Psi + {\color{red}U}R(\Psi)$
  - Update operator ${\color{red}U}$ can be as simple as L^{-1} with a sweeper (implemented previously),
  - Or more complicated as $(L - S)^{-1}$ done with a matrix-free multigroup iterative solver, which calls the mesh sweeper, with residual as the source.
  - S can be an approximation, for example, only containing the isotropic part.

Meshless Consistent CMFD Setup/Projection/Solve/Prolongation

- Coarse elements (elements with the same coarse element ids):

$$V_E = \sum_{e\in E}(1,1)_e, \quad \vec{x}_E = \frac{\sum_{e\in E}(1,\vec{x})_e}{V_E}, \quad \vec{x}_{E,E'} = \frac{\sum_{s\in\Gamma_{E,E'}}(1,\vec{x})_s}{\sum_{s\in\Gamma_{E,E'}}(1,1)_s}, \quad \vec{n}_{E,E'} = \frac{\sum_{s\in\Gamma_{E,E'}}(1,\vec{n}_e)_s}{\sum_{s\in\Gamma_{E,E'}}(1,1)_s}, \quad h_{E,E'} = (\vec{x}_{E,E'} - \vec{x}_E)\cdot\vec{n}_{E,E'}$$

- On coarse element side:

$$J^{\text{out}}_{E\to E',g} = \sum_{s\in\Gamma_{E,E'}}\sum_{g\in p}\sum_{\vec{\Omega}\cdot\vec{n}_E>0}(1,|\vec{\Omega}\cdot\vec{n}_E|\Psi_{p,d})_s,$$

$$\hat{D}_{E,E',g} = \frac{J^{\text{out}}_{E\to E',g} - J^{\text{out}}_{E'\to E,g} - \kappa_g(\Phi_{E',g}-\Phi_{E,g})}{\Phi_{E',g}+\Phi_{E,g}},$$

$$\kappa_{E,E',g} = \max\left(\frac{1}{\frac{h_{E,E'}}{D_{E,g}}+\frac{h_{E',E}}{D_{E',g}}},\frac{1}{4}\right). \qquad D_{E,g} = \min\left(\frac{1}{3\Sigma_{E,t,g}}, D_{\max}\right),$$

- On boundary: $\kappa_{E,S,g} = \frac{J^{\text{out}}_{E\to S,g}}{\Phi_{E,g}}.$

- On element:

$$\Phi_{E,p} = \frac{\sum_{e\in E}\sum_{g\in p}(1,\Phi_g)_e}{V_E}, \qquad \Sigma_{E,t,p} = \frac{\sum_{e\in E}\sum_{g\in p}(1,\Sigma_{t,g}\Phi_g)_e}{V_E\Phi_{E,p}},$$

$$\Sigma_{E,s,p'\to p} = \frac{\sum_{e\in E}\sum_{g'\in p'}\sum_{g\in p}(1,\Sigma_{s,0,g'\to g}\Phi_{g'})_e}{V_E\Phi_{E,p'}},$$

$$\nu\Sigma_{E,f,p} = \frac{\sum_{e\in E}\sum_{g\in p}(1,\nu\Sigma_{f,g}\Phi_g)_e}{V_E\Phi_{E,p}},$$

$$\chi_{E,p} = \frac{\sum_{e\in E}\sum_{g\in p}\sum_{g'=1}^{G}(1,\chi_g\nu\Sigma_{f,g'}\Phi_{g'})_e}{\sum_{e\in E}\sum_{g'=1}^{G}(1,\nu\Sigma_{f,g'}\Phi_{g'})_e},$$

- Solve: $J_{E,E',g} = \kappa_{E,E',g}(\phi_{E',g}-\phi_{E,g}) + \hat{D}_{E,E'g}(\phi_{E',g}+\phi_{E,g}), \; J_{E,S,g} = \kappa_{E,S,g}\phi_{E,g}$

  $$\mathbb{A}\boldsymbol{\phi} = \frac{1}{k}\mathbb{B}\boldsymbol{\phi}$$

  – Uses PETSc/SLEPc with A and B explicitly assembled.

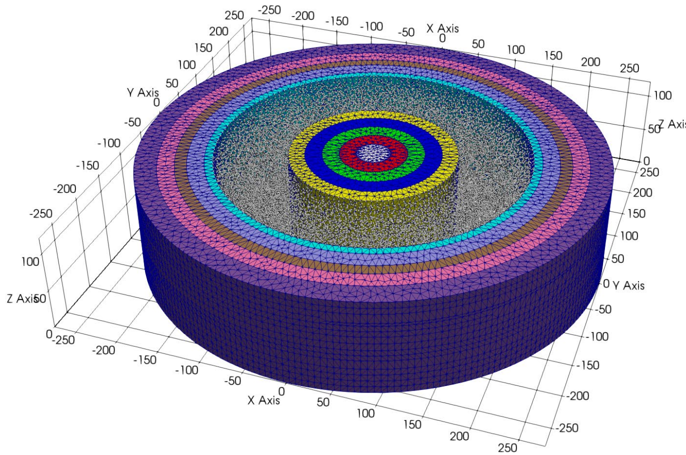
- Prolongation:

$$\Psi_{e,p,d}(\vec{x}) = \frac{\phi_{E,g}}{\Phi_{E,g}}\Psi_{e,p,d}(\vec{x})$$

*Meshless means these formula do not care whether coarse elements are regular and only require which coarse element a fine element belongs to.

Results with a Simplified Randomly-Packed PBR

- Total 41,048 pebbles. Packing fraction is 0.51 (relatively low) with PEBBLES.
- Total 444,729 tetrahedra. 78,250 node points.
- Reflecting boundary condition on top and bottom, vacuum on the outer radius.
- 11-group cross sections are pre-generated with Serpent.



- Total 15 cross section sets: five for the inner reflector, five for the pebble bed, five for the outer reflector, based on the distance to the center line.
- Level-symmetric S4 quadrature is used (24 streaming directions).
- Calculation can be one on INL Sawtooth with 2 nodes in 20min. But all CPU results are gathered with 24 nodes with 24 CPU cores on each node.

Coarse Element ID and Coarse Groups

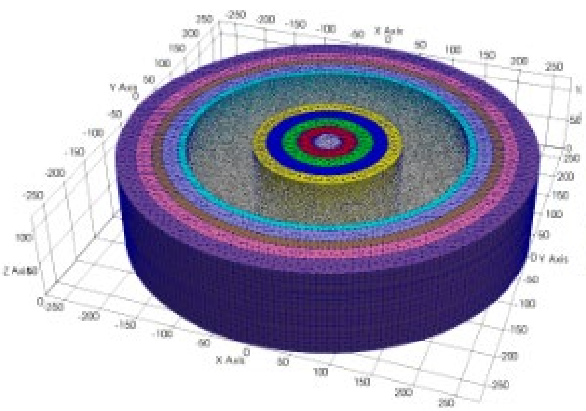


Figure 1. A simplified PBR.

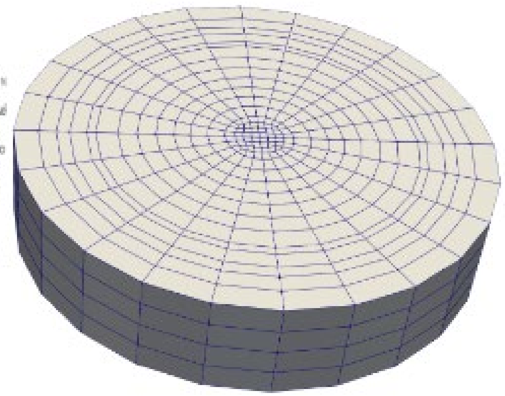


Figure 2. Coarse mesh.

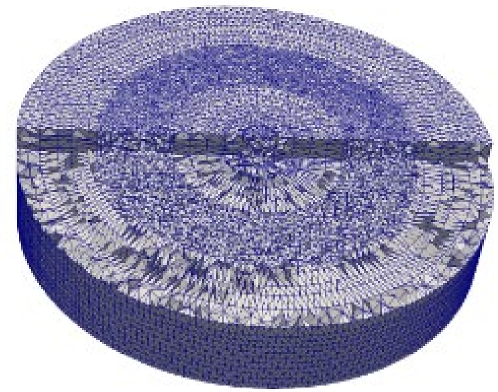


Figure 3. Mesh with first 174 coarse elements removed.

- Coarse element ids of all element are assigned.
- Results with the coarse element id being, both *the element id* and *coarse element id assigned through the coarse mesh* will be presented.
- Number of groups for CMFD: 11 or 3.

Group index	Upper boundary (MeV)	Lower boundary (MeV)	Coarse group index
1	[Serpent default]	3.32870E+00	1
2	3.32870E+00	1.15620E-01	1
3	1.15620E-01	3.48110E-03	1
4	3.48110E-03	1.32700E-04	1
5	1.32700E-04	8.10003E-06	2
6	8.10003E-06	6.25000E-07	2
7	6.25000E-07	2.09610E-07	2
8	2.09610E-07	7.64970E-08	3
9	7.64970E-08	4.73020E-08	3
10	4.73020E-08	2.00100E-08	3
11	2.00100E-08	[Serpent default]	3

Results with different polynomial order

- Fine-mesh finite difference diffusion acceleration; 11-group CMFD;
- Few other settings for the multigroup iterative solver in the transport update: Scattering truncation 0; 4 inner iteration with GMRes .

p	Δk_{eff} (pcm)	N	Wall time (s)	Residual grind time (μs)	Sweeping grind time (μs)	Total Sweeps (7*N)
0	8503	12	76.0	0.818	0.467	84
1	52.9	16	122.1	0.514	0.239	112
2	1.2	15	316.5	0.621	0.313	140
3	0	27	1330.5	1.169	0.540	189

- k-eff of p=0 indicates significant homogenization error.
- k-eff convergence with respect to p is fast. We recommend p=2 for typical PBR analysis.
- Grind time (total time divided by the number of calls, the number of total DoFs, multiplied by the number of processors) is about a micro second.
- Paper contains detailed break-down on the wall time.

Effects of coarse mesh and coarse energy groups

	N	Wall time (s)	Time in CMFD (s)
Fine mesh, 11-group	20	316.5	73.9
Fine mesh, 3-group	24	314.2	49.3
Coarse mesh, 11-group	23	254.2	4.4

- Coarse mesh or coarse energy groups in CMFD result into slightly more Richardson iterations (more residual evaluations and transport sweeps) with less time in CMFD.
- For this test problem with 11 groups and 24 streaming directions, coarse mesh with 11-groups performs slightly better.

Conclusions

- PTT with CMFD is implemented and verified in Griffin.
- Results with PTT compare well with Serpent references for a simplified pebble bed reactor.
- PTT does not require fundamental changes to the current transport codes, i.e. most existing solving techniques, post-processing, mesh generation and cross section preparation can be reused.
- CMFD can significantly accelerate PTT calculations making calculation done in minutes for one single eigenvalue calculation.

Questions?

Conclusions and Future Works

- PTT with CMFD is implemented and verified in Griffin.
- Results with PTT compare well with Serpent references for a simplified pebble bed reactor.
- PTT does not require fundamental changes to the current transport codes, i.e. most existing solving techniques, post-processing, mesh generation and cross section preparation can be reused.
- CMFD can significantly accelerate PTT calculations.

- Pebble tracking depletion.
- Online cross section with machine learning.
- Mesh generation.
- Transient.

Weak Form for the Transport Equation on the Mesh

- We use the weak form with DFEM-SN, one-group, k-eigenvalue problem, isotropic scattering and vacuum boundary. The idea can be extended to general multigroup transport equations.
- Find solution $\Psi(x,\vec{\Omega})$, such that

$$b(\Psi^*,\Psi) = \frac{1}{k} f(\Psi^*,\Psi), \forall \Psi^* \in W,$$

$$b(\Psi^*,\Psi) \equiv -\left(\Psi^*,\vec{\Omega}\cdot\nabla\Psi\right)_{\mathcal{D}} + (\Psi^*,\Sigma_t\Psi)_{\mathcal{D}} - \langle [\![\Psi^*]\!],\Psi^-\rangle_{\Gamma_i} + \langle \Psi^*,\Psi\rangle_{\partial\mathcal{D}}^+ -\left(\Psi^*,\frac{1}{4\pi}\Sigma_s\Phi\right)_{\mathcal{D}},$$

$$f(\Psi^*,\Psi) \equiv \left(\Psi^*,\frac{1}{4\pi}\Sigma_f\Phi\right)_{\mathcal{D}}.$$

- Details on the notation can be found in the paper.
- Solution on each element (with partial pebbles on its vertices) is expanded with polynomials.
- Only the terms with cross sections need to be implemented differently.
- Break-down of $(\Psi^*,\Sigma_t\Psi)_{\mathcal{D}}$ term:

$$(\Psi^*,\Sigma_t\Psi)_{\mathcal{D}} = \sum_{e\in\mathcal{D}}\sum_{m=1}^{M} w_m \sum_{i=1}^{N(e)} \Psi^*_{e,m,i} \sum_{j=1}^{N(e)} \Psi_{e,m,j} \boxed{\int_e \Sigma_t(x) b_i(x) b_j(x) dx}$$

How Do We Generate the Mesh for PTT?

- Use a DEM code to generate all pebble locations in a file.
- Use a Fortran script to draw the geometry in a PLC (Piecewise Linear Complexes) file and include nodes of all pebble locations to form a final node file.
- Let TetGen process the PLC file to generate the final mesh for the entire geometry. (No new nodes should be inserted in the pebble packing region.)

- Drawbacks:
  - No control on how TetGen inserts *Steiner points* on the interface between pebble packing region and the static region, and how TetGen connects nodes to form tetrahedra.
  - Every time geometry changes, we have to modify the Fortran script.
  - Users must lean a DEM code to generate a packing manually.
  - The static region has to be meshed with tets.
- In the future: We hope to have a dedicated MOOSE mesh generator.

How Do We Generate Cross Sections for PTT?

- We use Monte Carlo to do reference calculations to generate multigroup (<30) macroscopic cross sections with fresh pebbles.
- Pebbles are grouped into clusters. Pebbles in one cluster have the same cross sections.
  - It appears that the number of clusters is small for making k-effective error in few hundreds pcm.
- No thermal feedback and no pebble tracking depletion with PTT in Griffin yet.

- In the future:
  - On-line cross section capability that can handle temperature dependency with depleted pebbles.
  - DEM codes can be used to generate pebble follow pattern during depletion without actually changing the mesh, or without moving node locations.
  - Demonstrate both equilibrium core, running-in, and transient calculations with PTT.

Results

p	NCE	NCG	L_{max}	N_{inner}	Inner	Δk (pcm)	CPU-time T1	T2	T3	T4	T5	T6	T	N1	N2
3	444,729	11	0	4	GMRes	0.0	416.3	128.7	598.3	102.7	8.2	541.7	1330.5	27	189
2	444,729	11	0	1	GMRes	1.1	114.4	56.7	191.4	158.6	8.5	74.3	475.0	45	180
2	444,729	11	0	4	GMRes	1.1	89.3	25.3	124.9	73.9	8.1	78.5	316.5	20	140
2	444,729	11	0	8	GMRes	1.1	104.9	18.8	132.7	62.5	8.0	76.1	308.7	15	165
2	444,729	11	0	12	GMRes	1.2	132.8	17.7	164.9	51.4	8.1	69.4	322.2	14	210
2	444,729	11	0	1	SI	0.9	66.2	128.0	233.1	367.8	8.2	69.6	747.9	102	102
2	444,729	11	0	4	SI	1.1	94.3	46.6	158.3	140.7	7.9	70.0	416.0	37	148
2	444,729	11	0	8	SI	1.1	127.2	31.2	172.5	90.3	8.0	69.2	373.3	25	200
2	444,729	11	0	12	SI	1.1	158.0	26.6	203.8	77.8	8.1	69.3	389.8	21	252
2	444,729	11	1	1	GMRes	1.1	104.7	51.5	179.9	153.6	8.0	75.2	456.5	41	164
2	444,729	11	1	4	GMRes	1.1	93.3	26.6	134.6	77.7	8.3	69.3	321.2	21	147
2	444,729	11	1	8	GMRes	1.1	111.3	20.1	145.8	65.8	9.5	74.8	322.7	16	176
2	444,729	11	1	12	GMRes	1.2	142.2	18.9	178.7	56.1	8.2	71.8	341.7	15	225
2	444,729	11	2	1	GMRes	1.1	109.2	59.6	201.1	146.4	8.1	74.6	470.2	41	164
2	444,729	11	2	4	GMRes	1.1	97.4	26.6	159.3	84.4	8.5	76.2	359.3	21	147
2	444,729	11	2	8	GMRes	1.1	116.2	20.2	160.4	59.7	8.0	69.4	326.0	16	176
2	444,729	11	2	12	GMRes	1.2	149.6	18.9	197.1	60.8	8.1	69.3	364.8	15	225
1	444,729	11	0	1	GMRes	52.9	22.2	11.7	38.6	99.6	8.8	8.8	174.2	28	112
1	444,729	11	0	4	GMRes	52.8	21.8	6.7	31.8	59.1	8.2	8.0	122.1	16	112
1	444,729	11	2	1	GMRes	52.8	25.3	11.7	46.8	100.5	8.5	8.1	187.2	28	112
1	444,729	11	2	4	GMRes	52.8	26.6	7.1	46.6	62.7	8.1	7.8	140.1	17	119
1	444,729	11	2	8	GMRes	52.8	34.8	5.9	49.9	52.2	8.0	7.7	137.2	14	154
1	444,729	11	2	12	GMRes	52.9	43.8	5.4	60.1	47.8	8.0	7.8	146.9	13	195
0	444,729	11	0	1	GMRes	-8503.1	5.7	2.5	9.4	51.6	7.7	0.5	81.6	15	60
0	444,729	11	0	4	GMRes	-8503.1	8.0	2.0	11.1	42.2	8.0	0.5	76.0	12	84
0	444,729	11	0	8	GMRes	-8503.0	12.5	2.0	15.9	42.0	8.6	0.5	79.2	12	132
0	444,729	11	0	12	GMRes	-8503.0	17.3	2.0	20.9	42.3	8.1	0.5	84.2	12	180
0	444,729	11	2	1	GMRes	-8502.9	7.5	2.5	12.8	52.6	8.1	0.5	86.5	15	60
0	444,729	11	2	4	GMRes	-8503.0	11.2	2.2	16.8	46.0	8.0	0.5	84.0	13	91
0	444,729	11	2	8	GMRes	-8503.1	16.1	2.0	22.3	42.3	8.1	0.5	92.9	12	132
0	444,729	11	2	12	GMRes	-8503.1	22.1	2.0	29.2	41.6	8.2	0.5	92.3	12	180
2	444,729	3	0	1	GMRes	0.6	176.4	86.1	294.2	120.9	8.3	71.8	547.9	68	272
2	444,729	3	0	4	GMRes	1.1	105.5	30.3	155.7	49.3	8.0	69.3	314.2	24	168
2	444,729	3	0	8	GMRes	1.2	139.2	25.2	176.5	35.8	7.9	77.8	329.6	20	220
2	444,729	3	0	12	GMRes	1.2	193.3	25.1	231.9	35.7	7.9	69.5	376.3	20	300
2	1,392	11	0	1	GMRes	-22.2	163.1	84.5	275.8	11.4	0.16	69.3	406.9	63	252
2	1,392	11	0	4	GMRes	-5.5	103.9	29.0	144.7	4.4	0.22	71.7	254.2	23	161
2	1,392	11	0	8	GMRes	-1.5	146.6	26.5	185.4	3.9	0.15	75.7	296.8	21	231
2	1,392	11	0	12	GMRes	-0.4	201.6	26.3	241.8	3.9	0.15	69.5	347.0	21	315

- 24 nodes on Sawtooth, 24 CPUs per node, Level-Symmetric S4, fixed convergence check.

- Δk roughly depends only on polynomial order.
- k is 1.24345 with p=3 while the reference Serpent value is 1.24181.
- T1 is the CPU time in all mesh sweepings; Sweeping grind time is about 0.32 μs.
- T2 is the time in all residual evaluations; Residual grid time is about 0.62 μs.
- T3 is the time in all transport updates including the time in residual evaluations, mesh sweepings, scattering source evaluation, etc.; T3 > T1+T2.
- T4 is the time in CMFD projection/solve/prolongation;
- T5 is the time in CMFD initial setup; It depends only on NCE.
- T6 is the time in partial matrix evaluation; It depends only on polynomial order p.
- T is the total wall time.
- N1: The number of Richardson iterations
- N2: The number of total mesh sweepings

- PJFNK solver takes about 1460.8s with 8 power iterations. We see 4 times reduction on CPU time.
- Moderate inner iterations with scattering truncation with GMRes is preferred.
- Coarse mesh can reduce the CMFD time although it requires more Richardson iterations than fine mesh.