

MOCFE: Ray Tracing Methods for Reactor Analysis

M. A. Smith, D. Kaushik, C. H. Lee, A. Martin-Lafleche, et. al.
Argonne National Laboratory



Organization of the Presentation

- Neutronics overview and solution methodology
- Parallelization Strategy and Success (and failures)
- Method of characteristics and parallelization thereof
- Parallel performance of MOCFE and MOCFE-bone
- Summary and future work



Neutronics=Neutron & Gamma Transport

- Fission process produces neutrons and gammas
 - 95% of heat is deposited at the site of fission events
 - 5% heat is gamma heating of everything in the domain
- So...predicting reactor behavior requires an idea of the fission rate distribution
- Boltzmann neutral particle transport equation
 - Eulerian form is most appropriate
 - Solution is the particle density as a function of position
 - Multiplied by the particle velocity and called “flux”
 - Have developed Stochastic (Monte Carlo) methods
 - **Have numerous deterministic methodologies**



The “Steady State” Boltzmann Transport Equation

$$\hat{\Omega} \cdot \vec{\nabla} \psi(\vec{r}, \hat{\Omega}, E) + \Sigma_t(\vec{r}, E) \psi(\vec{r}, \hat{\Omega}, E) = \iint \Sigma_s(\vec{r}, \hat{\Omega}' \rightarrow \hat{\Omega}, E' \rightarrow E) \psi(\vec{r}, \hat{\Omega}', E') d\Omega' dE' \\ + \frac{1}{k} \chi(\vec{r}, E) \iint v(\vec{r}, E') \Sigma_f(\vec{r}, E') \psi(\vec{r}, \hat{\Omega}', E') d\Omega' dE' \\ + S(\vec{r}, \hat{\Omega}, E)$$

$\psi(\vec{r}, \hat{\Omega}, E)$

The neutron flux (neutron density multiplied by speed)

$\Sigma_t(\vec{r}, E)$

The total probability of interaction in the domain

$\Sigma_s(\vec{r}, \hat{\Omega}' \rightarrow \hat{\Omega}, E' \rightarrow E) d\Omega' dE'$

The scattering transfer kernel

$\chi(\vec{r}, E) \quad v(\vec{r}, E) \quad \Sigma_f(\vec{r}, E)$

The steady state multiplicative fission source

$S(r, \hat{\Omega}, E)$

A generic source

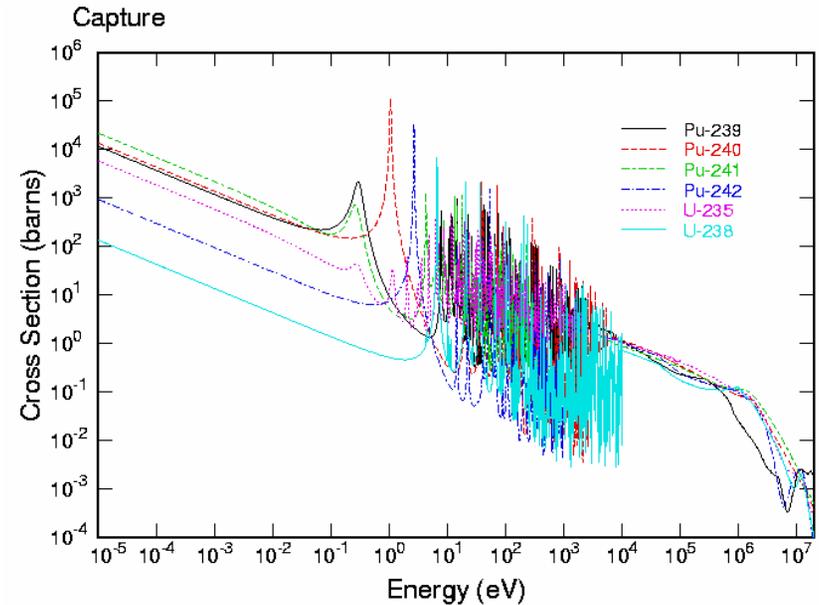
k

The multiplication eigenvalue



Deterministic Formulation = Multigroup

- To date the following has been tried in energy
 - Multigroup (histogram)
 - Polynomial expansions
 - Finite element expansions
- The gain in accuracy relative to the cost associated with the scheme indicates the multigroup method is the best
- Anything other than a continuous energy approach is a “fit” for a given reactor type
 - Must produce a multigroup library



$$\begin{aligned}
 \hat{\Omega} \cdot \vec{\nabla} \psi_g(\vec{r}, \hat{\Omega}) + \Sigma_{t,g}(\vec{r}) \psi_g(\vec{r}, \hat{\Omega}) &= \sum_{g'=1}^G \int \Sigma_{s,g' \rightarrow g}(\vec{r}, \hat{\Omega}' \rightarrow \hat{\Omega}) \psi_{g'}(\vec{r}, \hat{\Omega}') d\Omega' \\
 &+ \frac{1}{k} \chi_g(\vec{r}) \sum_{g'=1}^G \int v_{g'}(\vec{r}) \Sigma_{f,g'}(\vec{r}) \psi_{g'}(\vec{r}, \hat{\Omega}') d\Omega' \\
 &+ S_g(\vec{r}, \hat{\Omega})
 \end{aligned}$$



Neutronics Problem Size

$$\hat{\Omega} \cdot \vec{\nabla} \psi(\vec{r}, E, \hat{\Omega}) + \Sigma_t(\vec{r}, E) \psi(\vec{r}, \hat{\Omega}, E) = Q(\vec{r}, \hat{\Omega}, E)$$

$$Q(\vec{r}, \hat{\Omega}, E) = \int \int \Sigma_s(\vec{r}, \hat{\Omega}' \rightarrow \hat{\Omega}, E' \rightarrow E) \psi(\vec{r}, \hat{\Omega}', E') d\hat{\Omega}' dE' + \frac{1}{k} F(\vec{r}, \hat{\Omega}, E) + \dots$$

Nuclear Reactors are very large, complex systems

➤ Geometry is estimated to require 10-100 billion FEs

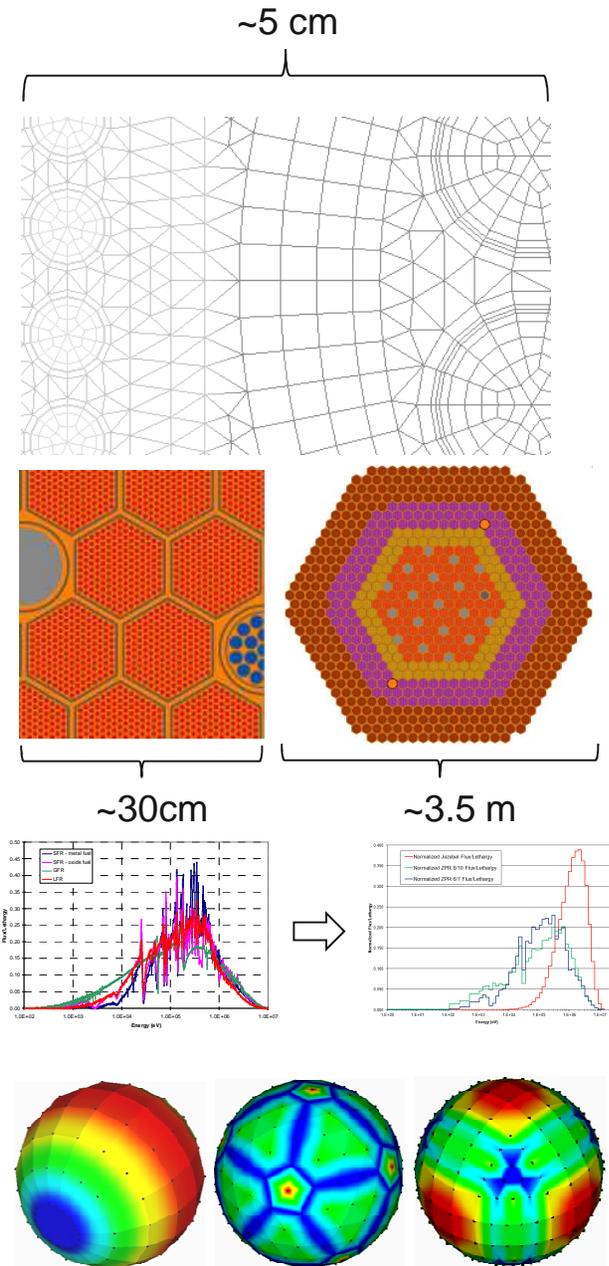
The energy grid is discretized into a multi-group grid turning one differential equation into G differential equations

$$\hat{\Omega} \cdot \vec{\nabla} \psi_g(\vec{r}, \hat{\Omega}) + \Sigma_{t,g}(\vec{r}) \psi_g(\vec{r}, \hat{\Omega}) = Q_g(\vec{r}, \hat{\Omega})$$

A discrete ordinates approximation breaks the spherical surface into a series of angular directions which creates N differential equations

$$\hat{\Omega}_n \cdot \vec{\nabla} \psi_g(\vec{r}, \hat{\Omega}_n) + \Sigma_t(\vec{r}) \psi_g(\vec{r}, \hat{\Omega}_n) = Q_g(\vec{r}, \hat{\Omega}_n)$$

For those of you counting, that is **G·N coupled differential equations** to solve on a mesh with $\sim 10^{10}$ elements. Or a **+10¹⁴ DOF problem**



Solving the Eigenvalue Problem

$Ax = \lambda x$ Cast the transport equation as a pseudo matrix-vector operation

T = streaming/collision/scattering

F = fission operator

$$\psi = \left[\psi_1(\vec{r}, \hat{\Omega}_1) \quad \psi_1(\vec{r}, \hat{\Omega}_2) \quad \cdots \quad \psi_G(\vec{r}, \hat{\Omega}_N) \right]^T$$

$$T\psi \leftrightarrow \hat{\Omega}_n \cdot \vec{\nabla} \psi_g(\vec{r}, \hat{\Omega}_n) + \Sigma_{t,g}(\vec{r})\psi_g(\vec{r}, \hat{\Omega}_n) - \sum_{g'=1}^G \sum_{n'=1}^N \Sigma_{s,g' \rightarrow g}(\vec{r}, \hat{\Omega}_{n'} \rightarrow \hat{\Omega}_n) \psi_{g'}(\vec{r}, \hat{\Omega}_{n'}) w_{n'}$$

$$F\psi \leftrightarrow \chi_g(\vec{r}) \sum_{g'=1}^G \nu_{g'}(\vec{r}) \Sigma_{f,g'}(\vec{r}) \sum_{n'=1}^N \psi_{g'}(\vec{r}, \hat{\Omega}_{n'}) w_{n'}$$

$$T\psi = \frac{1}{k} F\psi$$

Standard eigenvalue notation:

$$k \cdot \psi = T^{-1} F \psi$$

$$x = F \psi$$

$$k \cdot x = FT^{-1}x = Ax$$

$$Ax = \lambda x$$





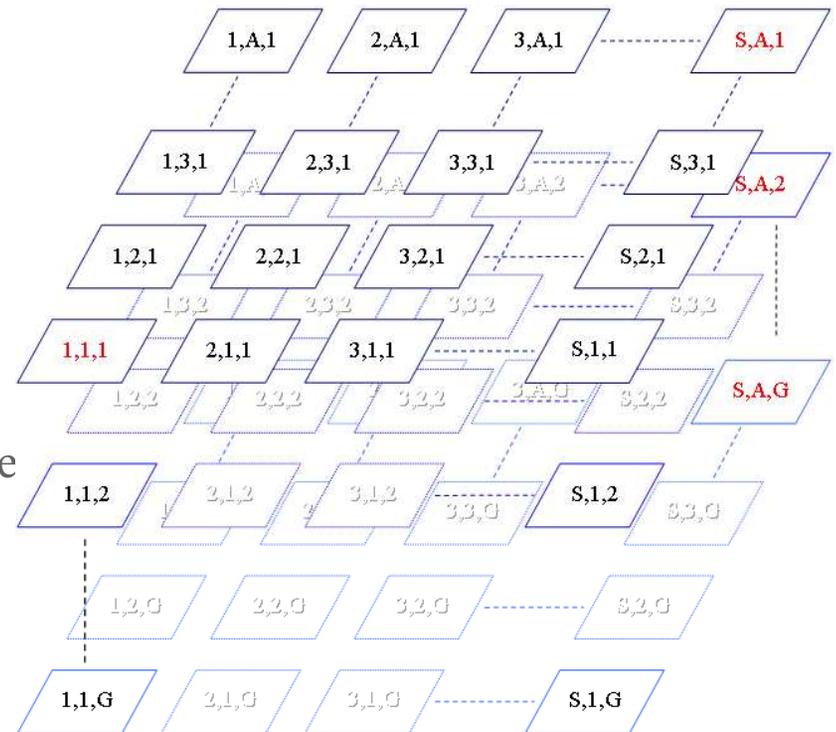
Organization of the Presentation

- Neutronics overview and solution methodology
- **Parallelization Strategy and Success** (and failures)
- Method of characteristics and parallelization thereof
- Parallel performance of MOCFE and MOCFE-bone
- Summary and future work



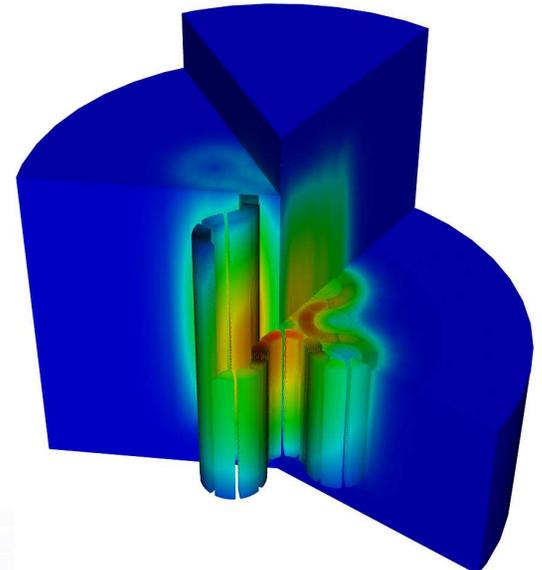
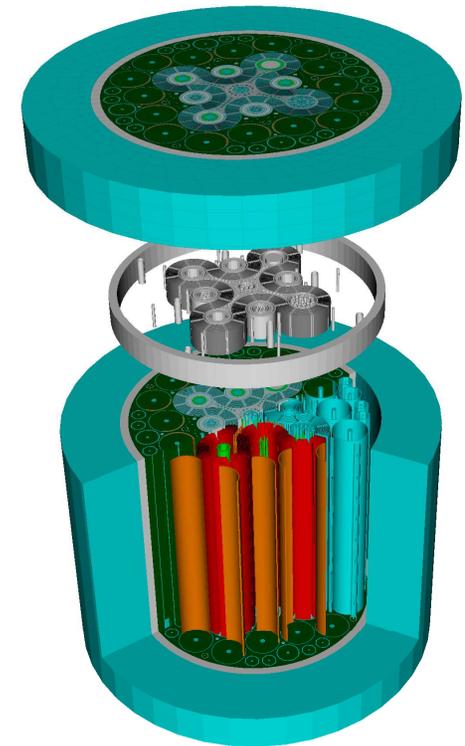
Deterministic Parallelism?

- Vector space: space-angle-energy
 - 10 billion elements so must use space
 - 200-500 angles
 - 200-500 energy groups
- 40,000-250,000 DOF per element?
 - Per process memory restriction
- Parallel transport studied for well over a decade
 - LANL, LLNL, ORNL, Sandia, and others
- Production tools?
 - Most are structured grid and use KBA
 - a pipeline for sweeping: overload processor
 - Unstructured mesh capabilities?
 - Very few tools, only one available
 - Parallel capabilities?
 - Few hundred processors
 - Most efficient on small machine
 - All focus use spatial domain decomposition



Parallelism in ANL Neutronics

- Lessons learned at ANL
 - Parallelization in space-angle-energy is necessary
 - Routinely use space-angle decomposition
 - Angle decomposition for S_N is generally excellent
 - 95% on supercomputers, 75% on lesser machines
 - Energy decomposition is least studied
 - Ineffective with fewer than 15 groups per process
 - Typical parallelization strategies are not necessarily best idea
- Capabilities
 - PN2ND: 500-3,000 processors (2007)
 - **SN2ND: 1,000-300,000+ processors (2008-present)**
 - MOCFE: 1,000-100,000+ processors (2010)



SN2ND Parallel Performance in 2008-2009

- Strong spatial scaling of 94% on Blue Gene/P

Total Cores	Vertices/ Process	Total Time (seconds)	Parallel Efficiency
8,192	7,324	2,402	100%
16,384	3,662	1,312	92%
24,576	2,441	873	92%
32,768	1,831	637	94%

Mainly measurement of PETSc

Strong angle scaling
 ~75% at 1 angle per process
 ~90% at 4 angles per process

Diffusion equation needs further partitioning

- Weak angle scaling of 76% 294,912 cores of Blue Gene/P

Total Cores	4 π Angles	Total Time (seconds)	Weak Scaling
32,768	32	579	100%
73,728	72	572	101%
131,072	128	581	100%
163,840	160	691	84%
294,912	288	763	76%

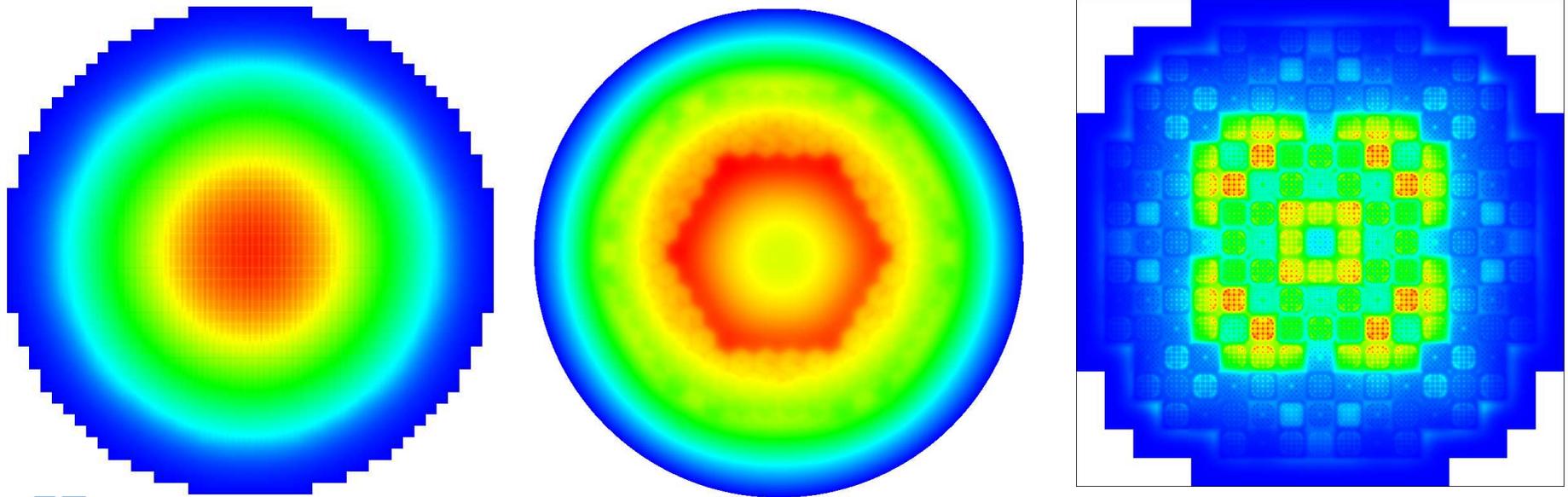
Weak scaling in space means mesh refinement for us

Refinement of bad aspect ratio elements produced >95%



Problems with SN2ND

- Has significant difficulty with whole core 3D geometries
 - Failed to solve full core PWR and VHTR
 - Condition number dramatically degraded as mesh size shrunk
 - “Void” regions had to be replaced with cladding or graphite
 - Simple test cases indicate more issues with accuracy
- **Is still a focus of ongoing research**
 - Used routinely on variety of problems older tools cannot solve
 - Least squares formulation appears to fix outstanding issues





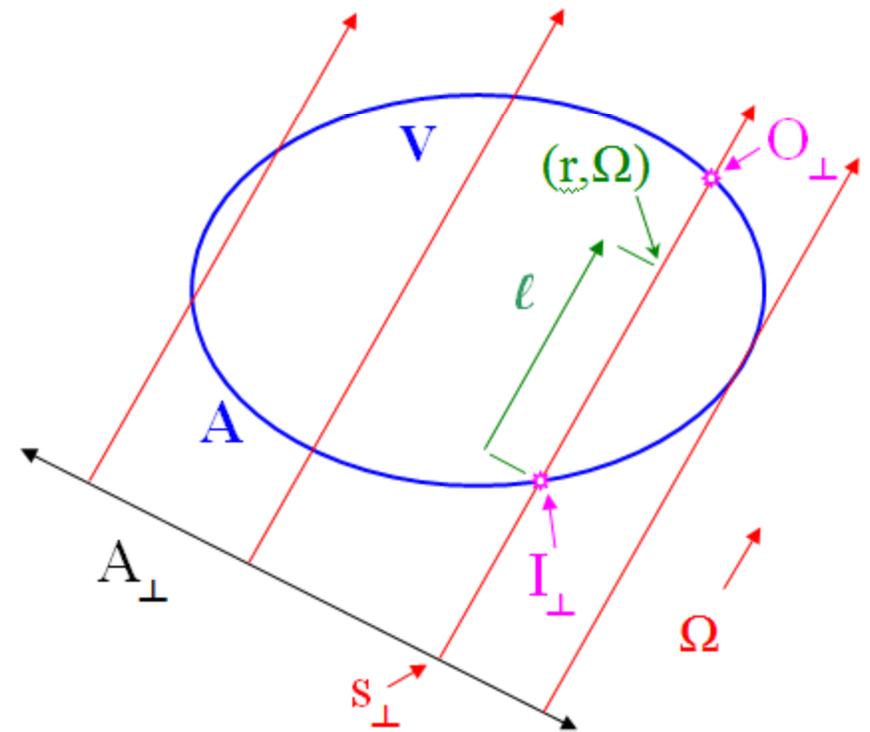
Organization of the Presentation

- Neutronics overview and solution methodology
- Parallelization Strategy and Success (and failures)
- **Method of characteristics and parallelization thereof**
- Parallel performance of MOCFE and MOCFE-bone
- Summary and future work

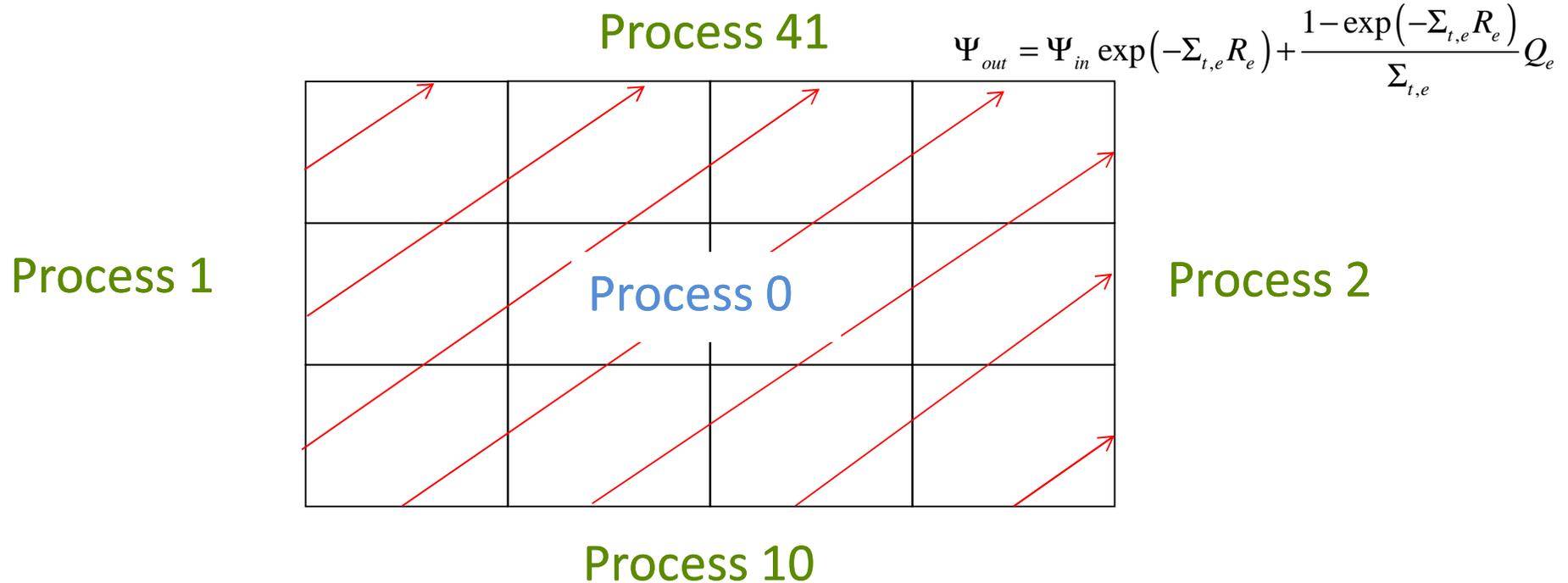


Method of Characteristics

- Fundamentally a discrete ordinates methodology
 - For each direction, characteristic lines are drawn through the domain
 - Turns a single differential equation by direction into $O(10^6)$ equations
 - Uses Rectangle rule to merge into integral relation for domain
 - The flux solution is propagated from the entry to the exit along each trajectory
 - Generally uses flat spatial sources where higher order projections are expensive
- Studied because it resolves SN2ND issues
 - Treats “low density” regions
 - No condition number problems
 - Very low overhead per element
- Outstanding Issues
 - **Not an easy path to extreme computing capability**
 - A multi-grid preconditioner concept appears possible but not proven
 - Adaptation to mesh deformation requires redrawing the trajectories
 - Other miscellaneous and minor stuff



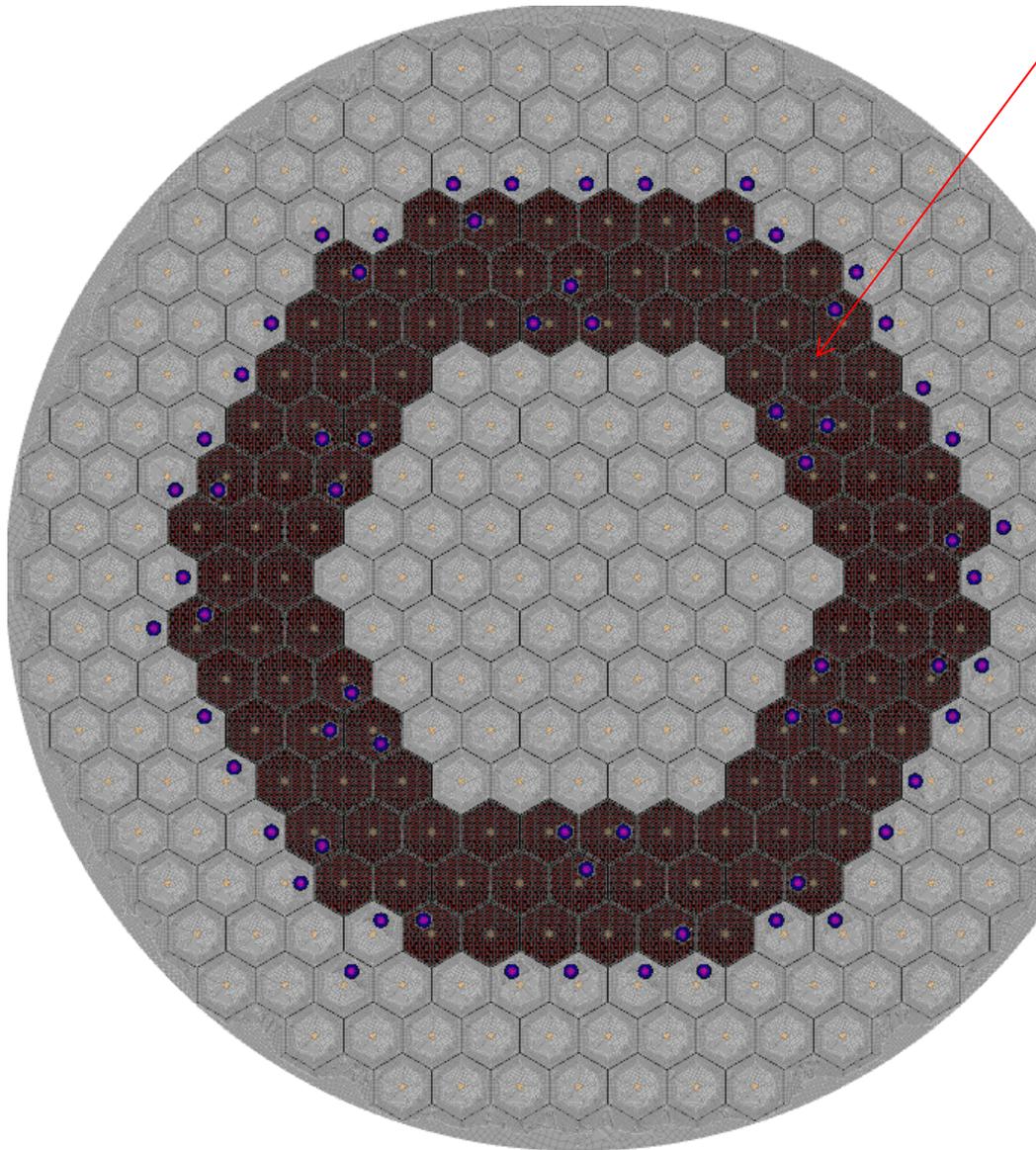
Design of the Mini-app: MOCFE-bone



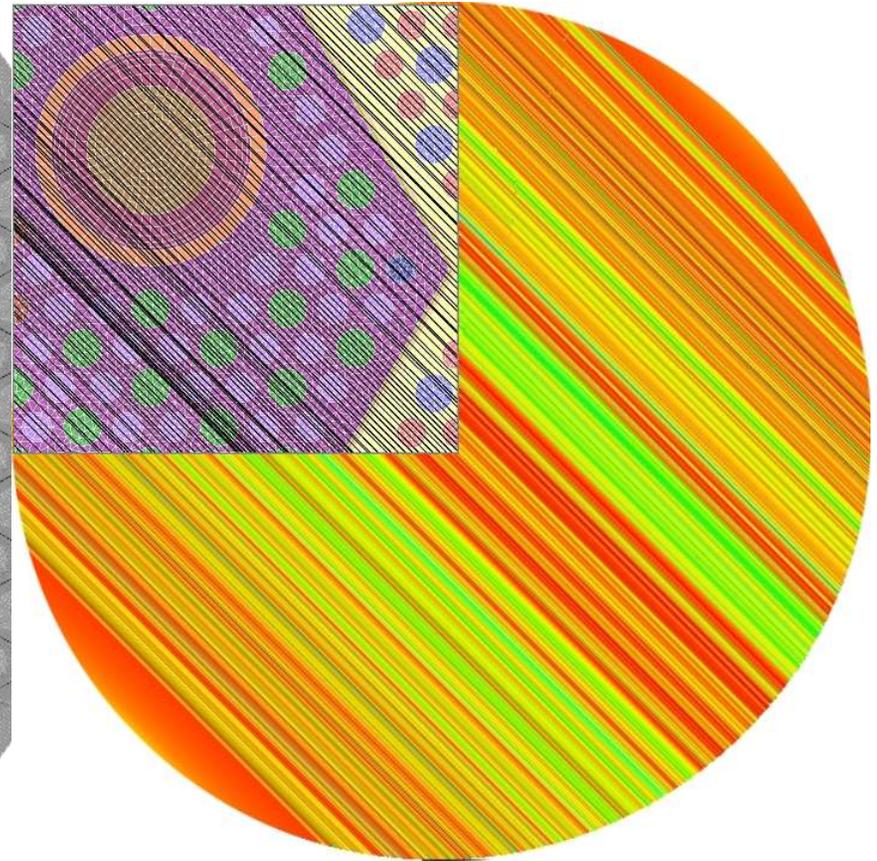
- The number of intersections per trajectory ranges from 1 to 6.
- The communication requires
 - 3 moments from rank 0 to rank 2 3 moments from rank 1 to rank 0
 - 4 moments from rank 0 to rank 41 4 moments from rank 10 to rank 0
- The load balance is dictated by the number of intersections per process
 - We can emulate a load imbalance by varying the number of intersections per process
 - The communication is inherently imbalanced but we can vary the domain volume to make it worse



Example MOCFE Characteristic Lines



Much higher mesh density

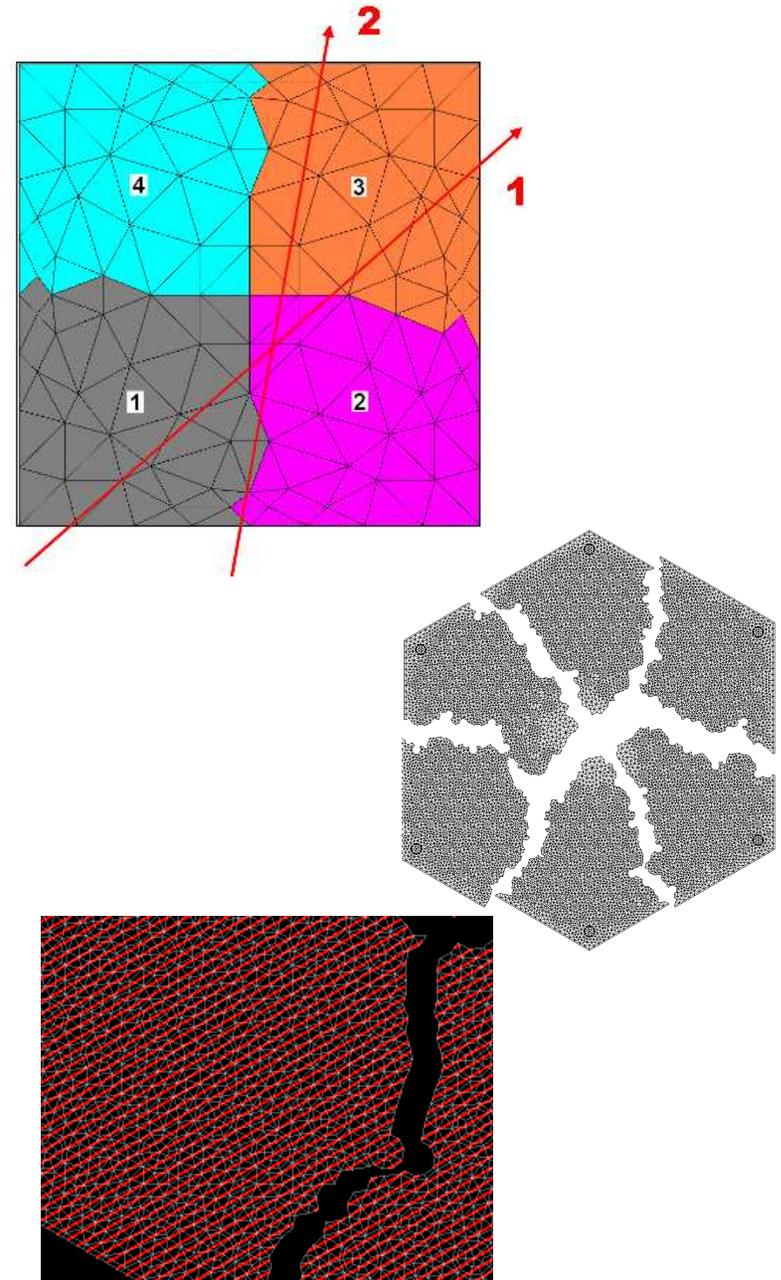


Trajectory density changes by direction!



MOC parallelization strategies

- Line by line sweep from the incident surface to the exiting surface for all discrete ordinate angular directions
- Option 1: Sequenced sweep
 - Yields severe latency issues (downwind processors wait)
 - Optimal domain decomposition = reentrant trajectories
 - Iterative source requires numerous sweeps
 - Direction dependent sweeping pattern
 - **Conclusion: (real) bad idea**
- Option 2: Increase vector space
 - Nearest neighbor communication and no latency
 - Per process communication scales
 - Per process memory increases gradually
 - Depends on trajectory density
 - **Conclusion: result depends on preconditioner**





Organization of the Presentation

- Neutronics overview and solution methodology
- Parallelization Strategy and Success (and failures)
- Method of characteristics and parallelization thereof
- **Parallel performance of MOCFE and MOCFE-bone**
- Summary and future work



Prototype MOCFE Scalability

Angle Decomposition	Strong Scaling for "D" Sub-domains per Process				
	D = 1	D = 2	D = 3	D = 4	D = 5
1	1.00	1.00	1.00	1.00	1.00
2	0.98	0.98	0.98	0.99	0.99
4	0.91	0.94	0.95	0.96	0.96
8	0.90	0.93	0.95	0.94	0.94
16	0.57	0.47	1.28	1.10	0.56

Spatial Decomposition	Elements per Process	Scaling for D subdomains per process			
		D=1	D=2	D=3	D=4
128	19531	1.00	1.00	1.00	1.00
256	9765	1.13	1.01	0.85	0.93
512	4882	1.39	0.87	0.78	1.04
1024	2441	1.14	0.98	0.68	0.96
2048	1220	1.15	0.82	0.51	0.78

Angles 2π	MPI processes	Trajectories	GMRES	Scaling
16	4,096	2,679,622	9733	1.00
32	8,192	5,359,244	9852	0.98
64	16,384	10,718,488	9275	1.00
128	32,768	21,436,976	9266	0.99
256	65,536	42,873,952	9826	0.93

Space Proc.	Total Elements	Trajectories	Time (sec)	Millions of Intersections	GMRES	Scaling	Modified Scaling
512	622,669	304,780	317	125	4611	1.00	1.00
2056	2,500,079	316,018	228	251	6038	1.39	0.74
3005	3,653,794	324,918	233	310	6546	1.36	0.59
5620	6,922,297	332,156	162	430	5478	1.96	0.61

Strong and Weak Scaling

- Check option of using several sub-domains (D=?) per process
- Weak angle scaling fixed with 512 D=3 spatial partitioning and used Legendre refinements
- Weak spatial scaling doesn't account for work related to trajectory operations hence "modified"

Observations

- Strong angle scaling
 - Good
 - Full parallelization not wise
- Strong spatial scaling
 - Obvious load balance issues
- Weak angle scaling
 - Excellent as expected
- Weak spatial scaling
 - Good, but hard to measure



MOCFE-bone Preliminary Scalability Results

- Direct comparison is difficult to get
 - 10x10x12 cm problem broken in to 32³ structured elements (not a real problem)
 - Solution algorithm is different (MGS GMRES versus GS & WGS GMRES)
 - Number of “GMRES” iterations can vary dramatically
 - Cross sections are made up in MOCFE-bone
 - Trajectory area is optimized in MOCFE, not so much in MOCFE-bone
- However, many parts are the same (if not identical)
 - Communication work per GMRES iteration
 - Multiplications per GMRES iteration
- Conclusion: Scalability per Iteration
 - The differences are comm load imbalance in structured (bone) versus unstructured

Nodes	MOCFE (0.01 ΔA)					MOCFE-bone	
	Millions of Intersections	Time	Scaling	GMRES	Scaling/Iter	0.01 ΔA Scaling/iter	0.003 ΔA Scaling/iter
1	0.64	403	–	59	–	–	–
2	0.64	265	0.76	93	1.20	1.05	1.05
4	1.08	236	0.72	111	1.36	0.92	1.04
8	1.09	127	0.68	125	1.44	1.10	1.12
16	1.13	87	0.51	174	1.51	1.06	1.09
32	1.11	60	0.37	250	1.55	1.05	1.06
64	1.17	34	0.34	270	1.55	1.01	0.97



Conclusions and Overall Assessment

- Good points
 - MOCFE using GS & WGS GMRES demonstrated scalability
 - Good strong and weak scalability with respect to angle
 - Acceptable strong and weak scalability with respect to space
 - Combined should allow effective use of a large parallel machine
 - Approach resolves the per process memory and work requirements
 - Load balancing is primary hang up for good performance
 - Neither space nor angle decomposition alone are panaceas
- MOCFE-bone
 - Representative of MOCFE per iteration work/communication
 - Memory sizes are easily controllable
 - Demonstrated similar scalability/iteration on 32768 nodes of BlueGene/P
- Problem points
 - Use of exponentials should be replaced by table lookup approach
 - Will impact ratio of work/communication
 - Preconditioner development
 - MOCFE requires a preconditioner which is not obvious at this time
 - There are questions with regard to multi-core feasibility of MOC



Questions

