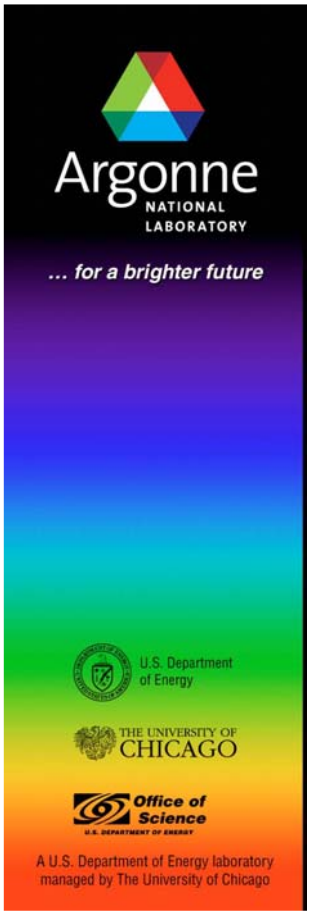


The Method of Characteristics for the Neutron Transport Equation

C. Rabiti¥, M.A. Smith, G. Palmiotti
Argonne, Nuclear Engineering Division

¥ *crabiti@anl.gov*

The Neutron Transport Equation

$$\left(\vec{\Omega}\cdot\vec{\nabla}+\Sigma_T\left(\vec{r},E\right)\right)\varphi\left(\vec{r},\vec{\Omega},E\right)=\int_0^\infty dE'\int_{4\pi}d\vec{\Omega}'\Sigma_s\left(\vec{r},\vec{\Omega}\leftarrow\vec{\Omega}',E\leftarrow E'\right)\varphi\left(\vec{r},\vec{\Omega}',E'\right)+$$

$$\frac{1}{K_{eff}}\chi\left(\vec{r},E\right)\int_0^\infty dE'\nu\left(\vec{r},E'\right)\Sigma_f\left(\vec{r},E'\right)\int_{4\pi}d\vec{\Omega}'\varphi\left(\vec{r},\vec{\Omega}',E'\right)$$

$\vec{r}$	: spatial coordinate	$\vec{\Omega}$	: angular coordinate
E	: energy	$\varphi\left(\vec{r},\vec{\Omega},E\right)$	: neutron angular flux
$\Sigma_T\left(\vec{r},E\right)$	: total cross section	$\Sigma_f\left(\vec{r},E\right)$	: fission cross section
$\nu\left(\vec{r},E\right)$	: number of neutron by fission	$\chi\left(\vec{r},E\right)$	: fission spectrum
K_{eff}	: fundamental eigenvalue		
$\Sigma_s\left(\vec{r},\vec{\Omega}\leftarrow\vec{\Omega}',E\leftarrow E'\right)$	: scattering kernel		

The Power Iteration Algorithm for the Steady State Solution

- The eigenvalue K_{eff} is computed by the power iteration method.

$$F[\varphi](\vec{r},E) = \chi(\vec{r},E)\int_0^{\infty} dE'\,\nu(\vec{r},E')\Sigma_f(\vec{r},E')\int_{4\pi} d\vec{\Omega}'\,\varphi(\vec{r},\vec{\Omega}',E')$$

$$\left[K_{eff}\right]^{h+1} = \frac{\left[K_{eff}\right]^{h}\int_{E_{\min}}^{E_{\max}} dE\int_V d\vec{r}\left[F[\varphi]\right]^{h+1}}{\int_{E_{\min}}^{E_{\max}} dE\int_V d\vec{r}\left[F[\varphi]\right]^{h}}$$

- This is the mostly widely used method because it has good convergence proprieties for the multigroup solution of the neutron transport equation.
- Several classes of acceleration methods are available such as Tchebyshev polynomial and rebalance.
- More recent investigations have considered an implicitly restarted Arnoldi method where the fundamental eigenvalue is not strictly isolated.

The Within Group Equation

$$\left(\vec{\Omega}\cdot\vec{\nabla}+\Sigma_{T,g}(\vec{r})\right)\varphi_g(\vec{r},\vec{\Omega}) = H_{g\to g}\left[\varphi_g\right](\vec{r},\vec{\Omega}) + S_g(\vec{r},\vec{\Omega})$$

φ_g : within group angular neutron flux

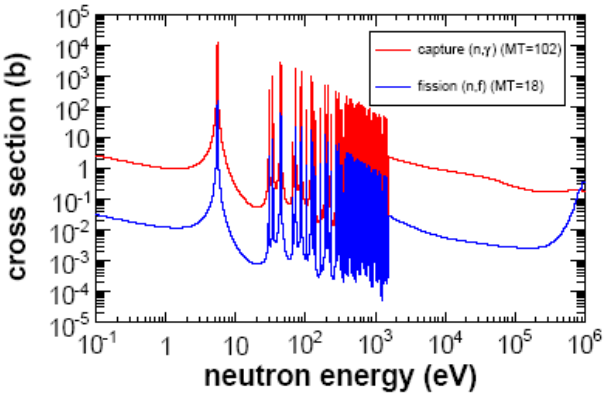
$$S_g = \sum_{\substack{g'=1\\ g'\neq g}}^{NG} H_{g'\to g}\left[\varphi_{g'}\right] + \frac{\chi_g(\vec{r})}{K_{eff}}\sum_{g'=1}^{NG} F_{g'}\left[\varphi_{g'}\right] \quad : \text{within group source}$$

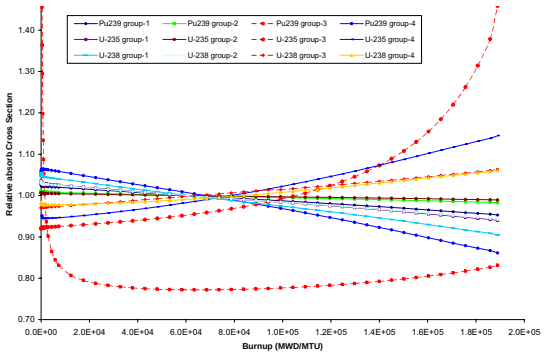
$$H_{g'\to g}[\ ] = \int_{4\pi} d\vec{\Omega}'\,\Sigma_{s,g'\to g}(\vec{r},\vec{\Omega}\leftarrow\vec{\Omega}')[\ ] \quad : \text{multi-group scattering kernel}$$

$$F_{g'}[\ ](\vec{r},\vec{\Omega},t) = \nu_{g'}(\vec{r})\Sigma_{f,g'}(\vec{r})\int_{4\pi} d\vec{\Omega}'[\ ] \quad : \text{multi-group fission operator}$$

Mesh Storage Setup I

- The primary problem with storing the mesh structure is the ability to model fuel depletion.
  - Thermal reactor neutron cross sections change substantially
  - Fast reactor neutron cross sections change significantly
- Additional problems arise with storage and evaluation of the cross section data at different levels of fuel depletion and temperature values (hence the focus on 10000 groups).
- For a given element of the mesh, the individual isotopic cross sections are merged to form macroscopic cross sections.
- The macroscopic cross sections define the source treatment necessary to solve the neutron transport problem (fission, void, etc…).





Mesh Storage Setup II

- Critical Assumptions
  - Thermal-hydraulic feedback
    - *Coarsening the temperature/density feedbacks does not dramatically alter the important power distribution.*
  - Fuel depletion
    - *Atom densities have very rapidly changing shapes in both space and time which are difficult to model with polynomials.*
    - *Given the temporal scale for fuel cycle operations (12-18 months), small inaccuracies in the local atom densities cannot realistically be accounted for even with the most complicated spatial-temporal functionalization.*
    - *Coarsening of the depletion zones is sufficient since the total fission rate (power level) will dominate the local solution and thus the coarse solution will still yield the average value predicted by a finer resolution method.*
- Mesh blocking is assumed to be sufficient for the current scope where each block is assumed to contain thousands of elements.
- Macroscopic cross sections are assigned to each block.

The Method of Characteristics: Introduction

- Assuming the flux is known inside of the scattering operator, the within group transport equation can be written in terms of linear characteristics curves.

$$\left(\vec{\Omega}\cdot\vec{\nabla}+\Sigma_{T,g}\left(\vec{r}\right)\right)\varphi_g\left(\vec{r},\vec{\Omega}\right)=Q_{g,s}\left(\vec{r},\vec{\Omega}\right)+S_g\left(\vec{r},\vec{\Omega}\right)$$

$$Q_{g,s}\left(\vec{r},\vec{\Omega}\right)=\int_{4\pi}d\vec{\Omega}'\Sigma_{s,g'\to g}\left(\vec{r},\vec{\Omega}\leftarrow\vec{\Omega}'\right)\left[\varphi_g\left(\vec{r},\vec{\Omega}'\right)\right]$$

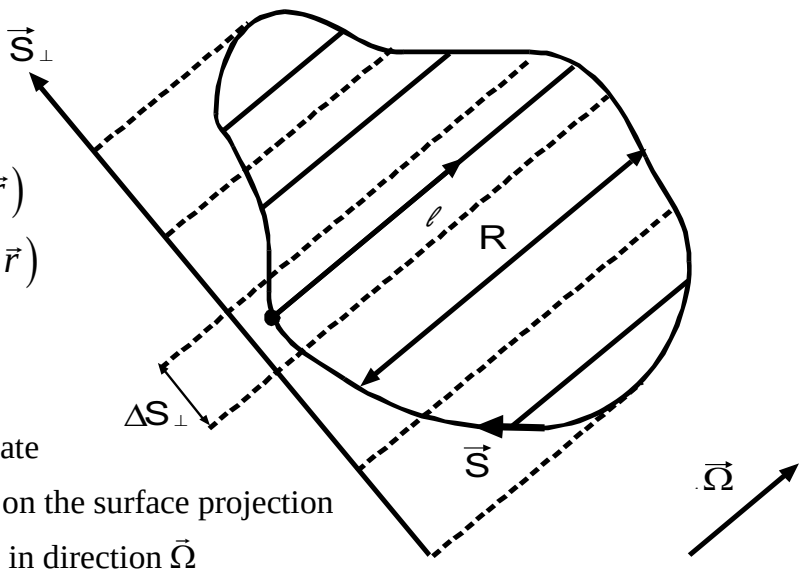
  - Along the characteristics curves the within group transport equation becomes a first order differential equation which is analytical.

$$\left(\frac{\partial}{\partial\ell}+{}^{i}\Sigma_T\right)\varphi\left(\ell,\vec{S},\vec{\Omega}\right)=Q_{g,s}\left(\ell,\vec{S},\vec{\Omega}\right)+S_g\left(\ell,\vec{S},\vec{\Omega}\right)$$

  - Consequently, it is possible to use an iterative solution that only acts upon the scattering source $Q_{g,s}\left(\ell,\vec{S},\vec{\Omega}\right)$.

The Characteristics Coordinate System

$$\begin{Bmatrix}\vec{r}\\ \vec{\Omega}\end{Bmatrix}\to\begin{cases}\ell=\ell\left(\vec{\Omega},\vec{r}\right)\\ \vec{S}=S\left(\vec{\Omega},\vec{r}\right)\\ \vec{\Omega}=\vec{\Omega}\end{cases}$$



$\vec{S}=$ surface coordinate

$\vec{S}_\perp=$ 2D coordinate on the surface projection

$\ell=$ distance from $\vec{S}$ in direction $\vec{\Omega}$

$\vec{\Omega}=$ angular coordinate

The Analytic Inversion of the Streaming Operator

- The domain is divided into finite elements or "cells" within which the cross sections are constant in space.
- The new reference system is introduced for each element.
- The transport equation is written in this new coordinate system.

$$\left(\frac{\partial}{\partial \ell}+{}^{i}\Sigma_{T}\right)\varphi\left(\ell,\vec{S},\vec{\Omega}\right)={}^{i}H\left[\varphi\right]\left(\ell,\vec{S},\vec{\Omega}\right)+{}^{i}S\left(\ell,\vec{S},\vec{\Omega}\right)$$

- By the introduction of the intrinsic source term: $Q^{i}={}^{i}H\left[\varphi\right]+{}^{i}S$ the transport equation is formally resolved in the form:

$$\varphi\left(\ell,\vec{S},\vec{\Omega}\right)=\varphi\left(0,\vec{S},\vec{\Omega}\right)e^{-{}^{i}\Sigma_{T}R}+\int_{o}^{\ell}d\ell' e^{-{}^{i}\Sigma_{T}\ell'}Q^{i}\left(\ell',\vec{S},\vec{\Omega}\right)$$

The Approximation of the Scattering and Source Term

- The scattering operator and the within group source are assumed to generate a flat contribution within each element.
- In order to simplify the presentation we assume isotropic (in angle) scattering: ${}^{i}\Sigma_{s}\left(\vec{\Omega}\leftarrow\vec{\Omega}'\right)\rightarrow{}^{i}\Sigma_{s,0}$.

The average scattering source is computed as follow:

$$\frac{1}{V_i}{}^{i}\Sigma_{s,0}\int_{V_i}d\vec{r}\int_{4\pi}d\vec{\Omega}'\varphi\left(\ell,\vec{S},\vec{\Omega}'\right)\approx{}^{i}\Sigma_{s,0}\sum_{n'=1}^{N\ \text{angle}}w_{n'}\frac{1}{V_i}\int_{V_i}d\vec{r}\,\varphi_{n'}\left(\ell,\vec{S}\right)=$$

$$={}^{i}\Sigma_{s,0}\sum_{n'=1}^{N\ \text{angle}}w_{n'}\frac{1}{V_i}\int_{S_{\perp}^{n'}}ds_2\int_{0}^{R(s_2)}d\ell\,\varphi\left(\ell,\vec{s},\vec{\Omega}_{n'}'\right)\approx$$

$$\approx{}^{i}\Sigma_{s,0}\sum_{n'=1}^{N\ \text{angle}}w_{n'}\frac{1}{V_i}\sum_{t_{n'}\cap V^{i}}w_{t_{n'}}^{\perp}\int_{0}^{R_{t_{n'}}}d\ell\,\varphi\left(\ell,t,\vec{\Omega}_{n'}'\right)={}^{i}\Sigma_{s,0}\sum_{n'=1}^{N\ \text{angle}}w_{n'}\frac{1}{V_i}\sum_{t_{n'}\cap V^{i}}w_{t_{n'}}^{\perp}\left\langle\varphi\right\rangle_{t_{n'}}^{i}$$

- Therefore:

$$Q^{i}\left(\vec{\Omega}\right)={}^{i}\Sigma_{s,0}\sum_{n'=1}^{N\ \text{angle}}w_{n'}\frac{1}{V_i}\sum_{t_{n'}\cap V^{i}}w_{t_{n'}}^{\perp}\left\langle\varphi\right\rangle_{t_{n'}}^{i}+{}^{i}S\left(\ell,\vec{S},\vec{\Omega}\right)$$

The Propagation Relation

- Assuming that the within group source is flat (and isotropic) within each element (in absence of external source it is only scattering).

$$Q^i = {}^i\Sigma_{s,0} \sum_{n'=1}^{N\ \text{angle}} w_{n'} \frac{1}{V_i} \sum_{t_{n'} \cap V^i} w^{\perp}_{t_{n'}} \left\langle \varphi \right\rangle^i_{t_{n'}} + {}^iS$$

- By means of the analytic integration of the transport equation is possible to write a relation bounding the incoming to the outgoing flux through the source term along a fixed trajectory.

$${}^{out}\varphi^i_{t_n} = {}^{in}\varphi^i_{t_n} e^{-{}^i\Sigma_T R^i_{t_n}} + \frac{1 - e^{-{}^i\Sigma_T R^i_{t_n}}}{{}^i\Sigma_T} Q^i$$

- Where

$${}^{out}\varphi^i_{t_n} = \varphi\left(R^i_{t_n}, \vec{S}, \vec{\Omega}\right) \qquad {}^{in}\varphi^i_{t_n} = \varphi\left(0, \vec{S}, \vec{\Omega}\right)$$

The Balance Equation

- An additional equation is required to relate the $\left\langle \varphi \right\rangle^i_{t_n}$ to the incoming and out going flux. The integral of the transport equation along a trajectory inside an element provides the needed relation.

$$\int_0^{R_{t_n}} d\ell \left[\left(\frac{\partial}{\partial \ell} + {}^i\Sigma_T \right) \varphi\left(\ell, \vec{S}, \vec{\Omega}\right) \right] = Q^i \int_0^{R_{t_n}} d\ell$$

$$\left\langle \varphi \right\rangle^i_{t_n} = \frac{{}^{in}\varphi^i_{t_n} - {}^{out}\varphi^i_{t_n} + R^i_{t_n} Q^i}{{}^i\Sigma_T}$$

Introduction to the Iteration Scheme

- In order to introduce the iterative scheme it is useful to recall the following relations and define the new quantities.

$$ {}^{out}\varphi^i_{t_n} = {}^{in}\varphi^i_{t_n} e^{-{}^i\Sigma_T R^i_{t_n}} + \frac{1-e^{-{}^i\Sigma_T R^i_{t_n}}}{{}^i\Sigma_T} Q^i $$

$$ \langle\varphi\rangle^i_{t_n} = \frac{{}^{in}\varphi^i_{t_n} - {}^{out}\varphi^i_{t_n} + R^i_{t_n} Q^i}{{}^i\Sigma_T} $$

$$ Q^i = \overbrace{{}^i\Sigma_{s,0} \underbrace{\frac{1}{V_i} \sum_{n'=1}^{N\ \text{angle}} \sum_{t_{n'} \cap V^i} \omega_{t_{n'}} \langle\varphi\rangle^i_{t_{n'}}}_{\langle\varphi_0\rangle^i}}^{Q^i_{s0}} + {}^iS_0 $$

Trajectory Sweep

- For all of the trajectories the outgoing flux is computed using a value of $\left[Q^i\right]^j$ the terms computed by the old $\left[\langle\varphi_0\rangle^i\right]^{j-1}$, where j is the iteration index.

$$ \left[{}^{out}\varphi^i_{t_n}\right]^j = \left[{}^{in}\varphi^i_{t_n}\right]^j e^{-{}^i\Sigma_T R^i_{t_n}} + \frac{1-e^{-{}^i\Sigma_T R^i_{t_n}}}{{}^i\Sigma_T} \left[Q^i\right]^j $$

- The contribution from this trajectory is added to the average flux on the region.

$$ \left[\langle\varphi\rangle^i_{t_n}\right]^j = \frac{\left[{}^{in}\varphi^i_{t_n}\right]^j - \left[{}^{out}\varphi^i_{t_n}\right]^j + R^i_{t_n}\left[Q^i\right]^j}{{}^i\Sigma_T} $$

$$ V_i\left[\langle\varphi\rangle^i\right]^j = V_i\left[\langle\varphi\rangle^i\right]^j + \omega_{t_n}\left[\langle\varphi\rangle^i_{t_n}\right]^j $$

- The outgoing flux is used as the incoming flux for the next cell along the trajectory.

$$ \left[{}^{out}\varphi^i_{t_n}\right]^j = \left[{}^{out}\varphi^{i+1}_{t_n}\right]^j $$

End of the Trajectory Sweep

- First, the average flux for each cell is computed and used for checking the convergence.

$$\left[\left\langle \varphi_0 \right\rangle^i \right]^j = \frac{V_i \left[\left\langle \varphi \right\rangle^i \right]^j}{V_i}$$

- If the convergence is not satisfied the new intrinsic source is computed.

$$\left[Q^i \right]^{j+1} = {}^i\Sigma_{s,0} \left[\left\langle \varphi_0 \right\rangle^i \right]^j + {}^iS_0$$

Limits of the MOC Method

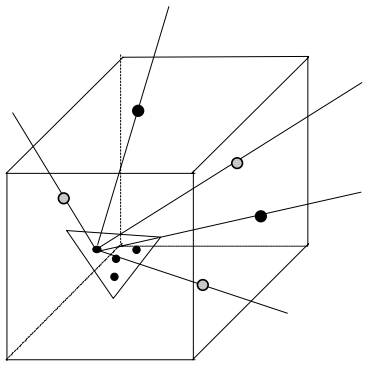
- The scattering source iteration will not converge when the ratio $\Sigma_{s,0}/\Sigma_T$ approaches 1.
- It is difficult to define coarse meshes for acceleration schemes in unstructured meshes.
- The ray tracing process in 3D becomes expensive for elements other than linear tetrahedral elements.
- The angular cubatures developed for MOC are mostly specialized for 2D problem.

MOC Parallel Computation

- The parallel implementation of MOC can be done easily with respect to the sweep process:
  - Each processor is assigned a set of rays.
  - The load balance can be done upfront since the number of intersections by trajectory is known before the sweeping process starts.
- The contribution to the average flux moments by each ray on each processor, at the end of each sweep, needs to be collected and distributed.
- Classical domain decomposition seems not to be an option since it would require the exchange of the angular flux solution for all the trajectories on each domain boundary.
- The domain decomposition would be preferably based upon the set of elements intersected by the set of rays assigned to each processor.

Ray Tracing Method I (part 1)

- Ray tracing is the process that determines for each trajectory the intersected elements and the respective intersection length.
- The ray tracing available in the first implementation of the code is based on tetrahedron meshes with linear interpolation descriptions.
- The first step in this ray tracing procedure is to define a quadrature on the external triangular surface that defines the set of starting trajectory points.



Ray Tracing Method I (part 2)

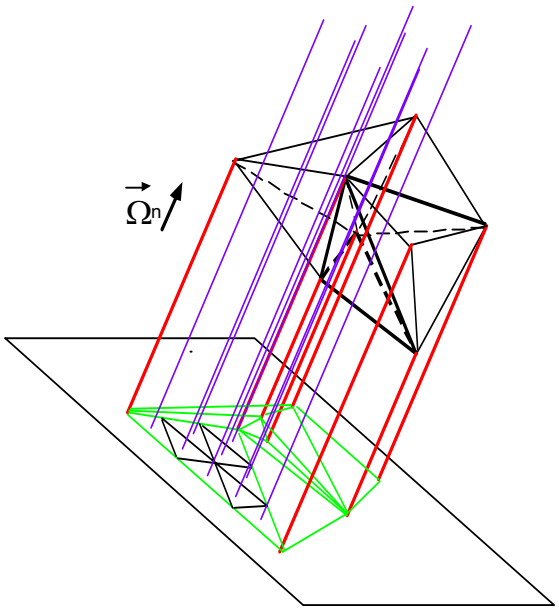
- Once that the direction and the starting point is known the ray tracing through the domain can proceed.
- The information needed for the ray tracing includes the tetrahedral vertices and the element surface to surface mapping.
- The problems of this approach are:
  - There is no guarantee that at least one trajectory for each direction will intersect each element.
    - *Such a failure violates the angular integration and thus renormalization factors by element are needed.*
  - The quadrature on the surface does not preserve the surface area on the outgoing boundary, thus renormalization is required for reflected boundary conditions.
- With respect to a parallel implementation this method shows a clear problem: even if a subset of starting points and directions can be assigned to each processor, the entire mesh structure and element surface to surface mapping is needed on all of the processors.

Ray Tracing Method II -Back Projection- (part 1)

- The back projection method is widely used in 2D calculation.
- This method has the following advantages:
  - All the elements are intersected at least once by one trajectory for each direction even if small heterogeneities in the mesh are present
  - The surface of each element is always preserved
  - The renormalization for reflected boundary conditions is eliminated
  - The accuracy of the average flux can be systematically improved in each element
- In 3D, the back projection method is quite expensive.

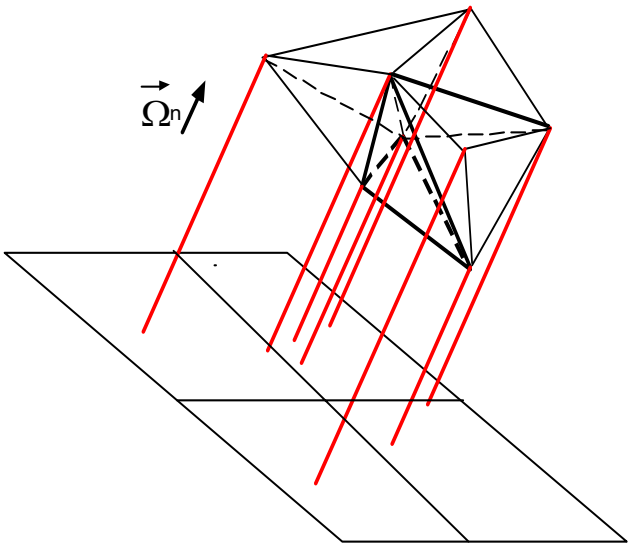
Ray Tracing Method II -Back Projection- (part 2)

- First all the vertices of the mesh are projected on a plane orthogonal to the trajectory set under consideration.
- The set of points obtained are triangulated (nlog(n) complexity).
- The surface quadrature is applied to each triangle defining the starting point on the plane.
- A ray casting process is needed for finding the starting positions of the rays on the external surface of the domain.



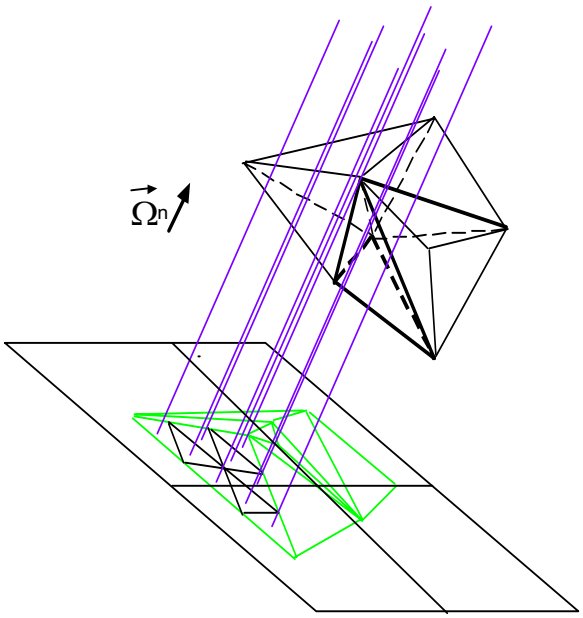
Ray Tracing Method II -Back Projection- (part 3)

- In a parallel environment a multigrid ray casting acceleration could be used.
- First a regular coarse grid (CG) mesh is built on the orthogonal plane.
- A set of elements is assigned to each processor which then determine in which CG mesh on the orthogonal plane the projection of the vertices lie.
- During this process each element is mapped to a corresponding CG mesh on the plane.



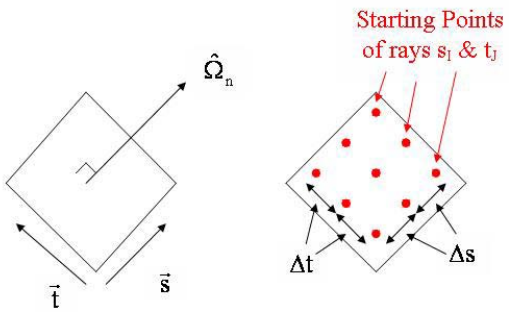
Ray Tracing Method II -Back Projection- (part 4)

- Each processor is then assigned one or more CG meshes on the plane.
- Each processor does the triangulation and the surface quadrature on the assigned CG meshes.
- The ray casting between the starting points on the plane and the domain surface has to take into account only the boundary elements mapped onto CG meshes assigned to the processor.
- The ray tracing starts and each processor needs only the mesh information for the elements mapped onto its CG meshes on the plane.
- This algorithm needs further investigation since the expense of the triangulation process across the CG meshes is not defined.



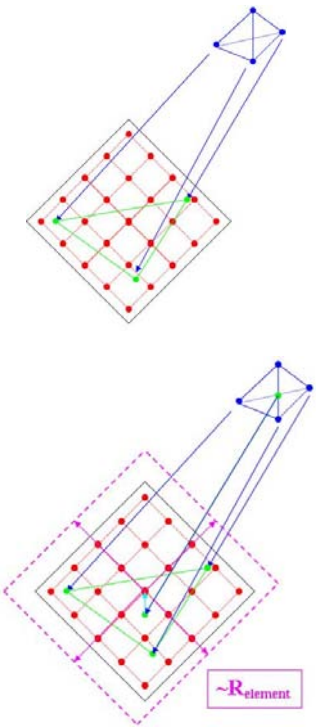
Ray Tracing Method III: (part 1)

- Backplane projection is used where an average ray spacing of Δs and Δt is assumed in directions s and t.
  - User defined or determined via min{$R_{element}$}
  - $R_{element}$ =Volume / min{$A_{element}$}
- The range of I,J are defined in terms of the “zero point” on the plane thus each processor will determine for each element and direction the I,J numbers associated with rays that pass through the element.



Ray Tracing Method III: (part 2)

- Exact projection is not applied.
  - More expensive due to need to triangulate set of points for each element surface.
  - Assigned elements may have convoluted surface.
- Instead, element centroid is projected to backplane and converted to an I,J point.
- $\sim R_{element}$ is used to define range of rays in I & J space.
  - All resulting points are checked for intersection (two possible intersections per element).
- For each element and direction, store:
  - I & J position in s and t directions
  - Of two intersections, store the distance from the midpoint to the backplane along with the distance between them.



Conclusion

- A first version of serial MOC neutron transport solver is in an advance debug status and the optimization process has been already started.
- The results shows that the computational time will be competitive with P_N method.
- The current ray tracing method has shown to be extremely fast in 3D tetrahedrons.
- Next steps will be:
  - Finish the optimization process of inner and outer and ray tracing
  - Benchmark the code
  - Test the new angular quadrature
  - Introduction of more finite elements like quadratic tetrahedrons and linear and quadratic hexahedrons
  - Anisotropic scattering kernel and sources
  - Parallel algorithm for the sweep iteration
  - Introduction of synthetic acceleration in angle and space
  - Parallel algorithm for the ray tracing

- Some slides with more info

Mesh Storage Setup III

- Mesh generator
  - Assumed to produce element connectivity and spatial grid of vertex points that defines the interpolation.
  - "Regions" are assigned to geometrical blocks in the domain (fuel pin)
- "Compositions" are defined listing the isotopes that are used to create macroscopic cross sections.
- Compositions are assigned to each region thereby allowing for individual definitions of atom densities along with temperature info for each block or "region."
- Mesh Level data structure
  - Global data constants :
    - *# mesh points, elements, blocks*
  - Global spatial vertex listing
  - Array of element block data structure
  - "Editing" information
- Element block data structure
  - Block data constants
    - *# elements, Region assignment*
    - *Element type & interpolation*
  - Spatial connectivity
  - Boundary condition information

Ray Tracing Method III: Ray Tracing for Each Processor (part 3)

- Given that all angles and intersecting rays (I,J) are treated on a single processor for each element, the normalization coefficients can be immediately determined and applied.
- The ray information below is stored for all elements on a given processor and sorted with respect to I then J.
- "master" processor consecutively requests non-assigned I,J ray number from each processor.
  - Rays are then assigned a global id number
  - All processors are given the unique global id number to update list.

ray #	1	2	3	4	…
I	1	2	1	2	…
J	1	1	1	2	…
Distance	100.0	100.2	120	114	…
Length	0.5	0.2	1.2	1.6	…
Element #	77734	77734	77735	77735	…