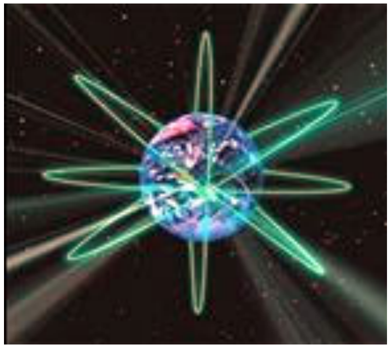
NE-255 Nuclear Reactor Theory

Lecture 23: Numerical Solution for Integral Transport Equation

Jasmina Vujic
Professor
Department of Nuclear Engineering
University of California, Berkeley

November 20, 2014



Time Dependent NTE without Delayed Neutrons

Rate of change = neutron sources rate – neutron losses rate

$$\frac{1}{v}\frac{\partial}{\partial t}\phi(\underline{r},E,\underline{\Omega},t) =$$

$$\int_0^\infty dE' \int_{4\pi} d\underline{\Omega}' \Sigma_s(\underline{r},E' \to E,\underline{\Omega} \bullet \underline{\Omega}')\phi(\underline{r},E',\underline{\Omega}',t) +$$

In-scattering neutron source rate

$$+\frac{\chi(E)}{4\pi}\int_0^\infty dE' \int_{4\pi} d\underline{\Omega}' \nu(E')\Sigma_f(\underline{r},E')\phi(\underline{r},E',\underline{\Omega}',t) +$$

Fission neutron source rate

$$+Q_{ext}(\underline{r},E,\underline{\Omega},t) -$$

External neutron source rate

$$-\Sigma_t(\underline{r},E)\phi(\underline{r},E,\underline{\Omega},t) -$$

Collision rate loss rate

$$-\underline{\Omega} \bullet \nabla\phi(\underline{r},E,\underline{\Omega},t)$$

Leakage rate loss rate

Time-Independent (Steady-State) Energy-dependent NTE

$$\Sigma_t(\underline{r},E)\phi(\underline{r},E,\underline{\Omega}) + \underline{\Omega} \bullet \nabla\phi(\underline{r},E,\underline{\Omega}) =$$
$$\int_0^\infty dE' \int_{4\pi} d\underline{\Omega}' \Sigma_s(\underline{r},E' \rightarrow E,\underline{\Omega} \bullet \underline{\Omega}')\phi(\underline{r},E',\underline{\Omega}') +$$
$$+\frac{\chi(E)}{4\pi}\int_0^\infty dE' \int_{4\pi} d\underline{\Omega}' \nu(E')\Sigma_f(\underline{r},E')\phi(\underline{r},E',\underline{\Omega}') +$$
$$+Q_{ext}(\underline{r},E,\underline{\Omega})$$

Time-independent NTE in purely absorbing medium:

$$\Sigma_a(\underline{r},E)\phi(\underline{r},E,\underline{\Omega}) + \underline{\Omega} \bullet \nabla\phi(\underline{r},E,\underline{\Omega}) = Q_{ext}(\underline{r},E,\underline{\Omega})$$

Time-independent NTE in vacuum:

$$\underline{\Omega} \bullet \nabla\phi(\underline{r},E,\underline{\Omega}) = 0$$

One-speed (monoenergetic) steady-state NTE

- **All neutrons have the same energy, E_0**

$$\phi(\underline{r},E,\underline{\Omega}) = \phi(\underline{r},\underline{\Omega})\delta(E-E_0); \quad \Sigma_s(\underline{r},E'\to E,\underline{\Omega}\bullet\underline{\Omega}') = \Sigma_s(\underline{r},\underline{\Omega}\bullet\underline{\Omega}')\delta(E-E_0)$$

$$\int_{-\infty}^{\infty} dE f(E)\delta(E-E_0) = f(E_0)$$

- **Assume that all neutrons sources are monoenergetic and isotropic**

$$Q_{ext}(\underline{r},E,\underline{\Omega}) = \frac{1}{4\pi}Q_{ext}(\underline{r},)\delta(E-E_0)$$

- **Now, integrate the NTE over energy to get monoenergetic steady-state NTE:**

$$\boxed{\begin{aligned}&\Sigma_t(\underline{r})\phi(\underline{r},\underline{\Omega}) + \underline{\Omega}\bullet\nabla\phi(\underline{r},\underline{\Omega}) = \\ &\qquad \int_{4\pi} d\underline{\Omega}'\Sigma_s(\underline{r},\underline{\Omega}\bullet\underline{\Omega}')\phi(\underline{r},\underline{\Omega}') + \frac{1}{4\pi}\int_{4\pi} d\underline{\Omega}'\nu\Sigma_f(\underline{r})\phi(\underline{r},\underline{\Omega}') + \frac{1}{4\pi}Q_{ext}(\underline{r})\end{aligned}}$$

Integral Form of Neutron Transport Equation in Infinite Medium

The optical length:

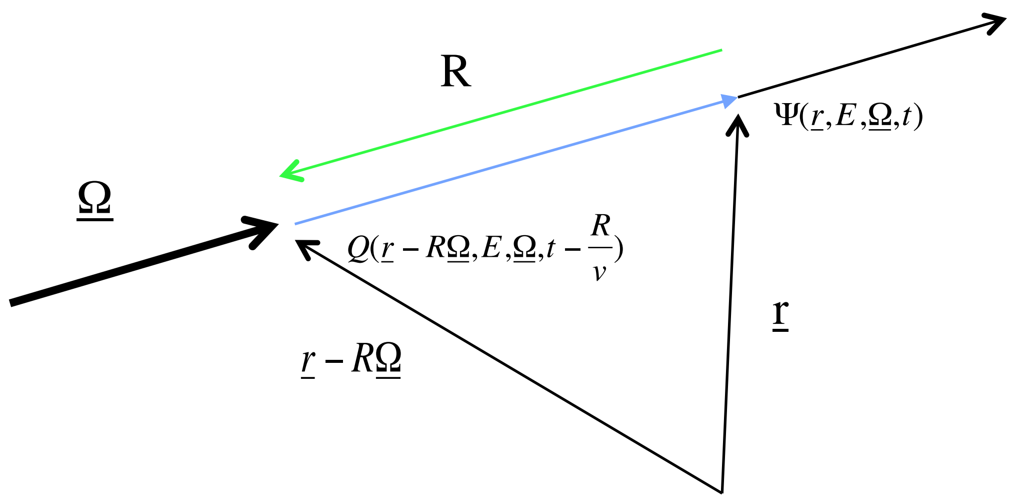
$$\tau(\underline{r},\underline{r}-R\underline{\Omega},E)=\int_0^R dR'\,\Sigma_t(\underline{r}-R'\underline{\Omega},E)$$

The source:

$$Q(\underline{r},E,\underline{\Omega},t)=\int_0^\infty dE'\int_{4\pi} d\Omega'\Sigma_s(r,E'\to E,\underline{\Omega}'\bullet\underline{\Omega})\Psi(\underline{r},E',\underline{\Omega}',t)+\frac{\chi_p(E)}{4\pi}\int_0^\infty dE'\nu(E')\Sigma_f(r,E')\Phi(\underline{r},E',t)+Q_{ext}(\underline{r},E,\underline{\Omega},t)$$

The flux:

$$\Psi(\underline{r},E,\underline{\Omega},t)=\int_0^\infty \exp\left[-\int_0^R dR'\,\Sigma_t(r-R'\Omega,E)\right]Q(\underline{r}-R\underline{\Omega},E,\underline{\Omega},t-\frac{R}{v})dR$$



Integral Form of Neutron Transport Equation in Finite Medium

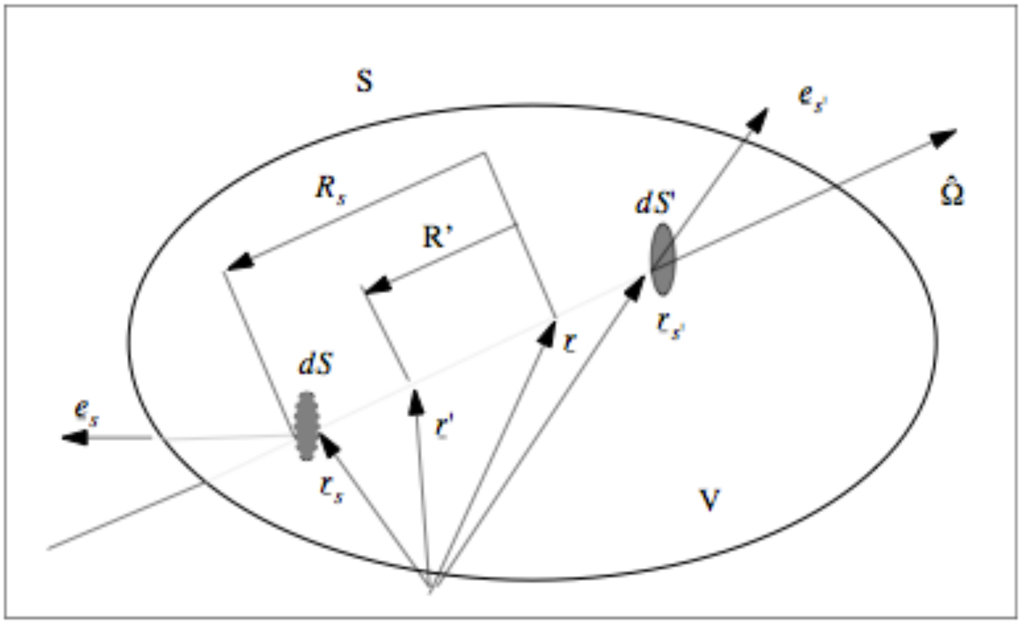


FIGURE XII.11. Coordinates characterizing neutron transport in 3-D

$$r' = r - R'\Omega\,,\; r_s = r - R_s\Omega\,,$$

$$\tau(r, r') = \tau(r, r - R'\Omega) = -\int_0^{R'} \Sigma_t(r - R''\Omega)dR''\,,$$

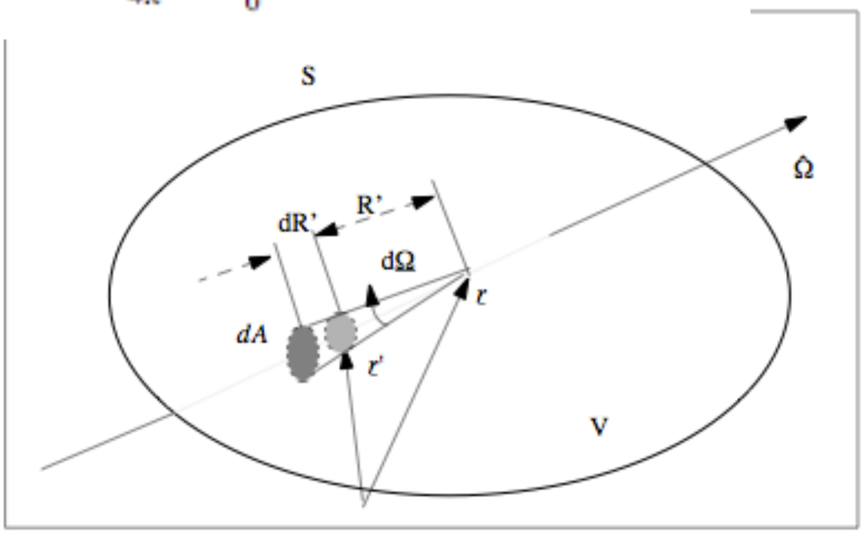
$$\tau(r, r_s) = \tau(r, r - R_s\Omega) = -\int_0^{R_s} \Sigma_t(r - R''\Omega)dR''$$

Used in the MOC

$$\boxed{\psi(r, \Omega) = \psi(r_s, \Omega)\exp[-\tau(r, r_s)] + \int_0^{R_s} Q(r', \Omega)\exp[-\tau(r, r')]dR'}$$

Integral Form of Neutron Transport Equation for Scalar Flux – Vacuum Boundary Conditions

$$\phi(\underline{r}) = \int_{4\pi} d\Omega \int_0^{R_s(\Omega)} dR' Q(\underline{r}', \Omega)\exp[-\tau(\underline{r}, \underline{r}')]$$



$$\underline{r}' = \underline{r} - R'\Omega$$

$$dV' = dR'dA$$

$$d\Omega = \frac{dA}{(R')^2}$$

$$dV' = dx'dy'dz' = d\underline{r}'$$

$$d\Omega dR' = \frac{d\underline{r}'}{(R')^2} = \frac{d\underline{r}'}{|\underline{r} - \underline{r}'|^2}$$

$$Q(\underline{r}', \Omega) = \frac{Q(\underline{r}')}{4\pi}$$

$$\phi(\underline{r}) = \int_V d\underline{r}'\left(Q(\underline{r}', \Omega)\frac{\exp[-\tau(\underline{r}, \underline{r}')]}{|\underline{r} - \underline{r}'|^2}\right)$$

$$\boxed{\phi(\underline{r}, E) = \int_V d\underline{r}'\left(Q(\underline{r}', E)\frac{\exp[-\tau(\underline{r}, \underline{r}', E)]}{4\pi|\underline{r} - \underline{r}'|^2}\right)}$$

$$Q(\underline{r}', E) = \int_0^\infty dE'\Sigma_s(\underline{r}', E' \to E)\phi(\underline{r}', E') + \chi(E)\int_0^\infty d(E'\nu\Sigma_f(\underline{r}', E')\phi(\underline{r}', E') + Q_{ext}(\underline{r}', E))$$

Integral Form of Neutron Transport Equation for Scalar Flux – General Boundary Conditions

$$\phi(r) = \int_{4\pi} d\Omega \psi(r_s, \Omega)\exp[-\tau(r, r_s)] + \int d\Omega \int_0^{R(\Omega),} dR' Q(r', \Omega)\exp[-\tau(r, r')]$$

$$d\Omega = \frac{dS|e_s \bullet \Omega|}{|r - r_s|^2}$$

$$\boxed{\phi(r) = \int_V dr' \left(Q(r', \Omega)\frac{\exp[-\tau(r, r')]}{|r - r'|^2}\right) + \int_S \frac{dS}{|r - r_s|^2} J^{in}(r_s, \Omega) exp[-\tau(r, r_s)],}$$

Used in CP

$$r' = r - R\Omega, \quad r_s = r - R_s\Omega, \quad \Omega = \frac{r - r'}{|r - r'|},$$

$$J^{in}(r_s, \Omega) = |e_s \bullet \Omega|\psi(r_s, \Omega), \text{ with } e_s \bullet \Omega < 0$$

$$\boxed{J^{out}(r_{s'}) = \int_V \frac{dr'}{|r - r_{s'}|^2}(e_{s'} \bullet \Omega) Q(r', \Omega) exp[-\tau(r, r_s)] \quad + \int_S \frac{dS}{|r_s - r_{s'}|^2}(e_{s'} \bullet \Omega) J^{in}(r_s, \Omega) exp[-\tau(r_{s'}, r_s)],}$$

Method of Characteristics (MOC) vs. Collision Probability (CP) Method

- **If a region consist of a homogeneous material with known sources that are uniform over that region, the following equation**

$$\psi(r,\Omega) = \psi(r_s,\Omega)\exp[-\tau(r,r_s)] + \int_0^{R_s} Q(r',\Omega)\exp[-\tau(r,r')]dR'$$

- **Simplifies to**

$$\psi = \psi_{in}\exp(-\Sigma_t s) + \frac{Q}{\Sigma_t}\left(1-\exp(-\Sigma_t s)\right)$$

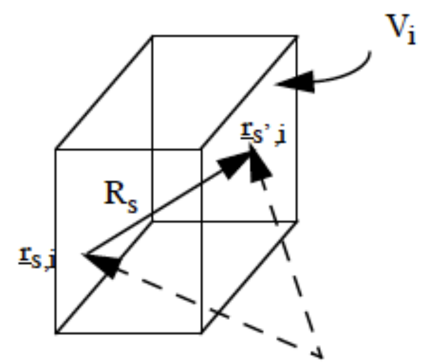
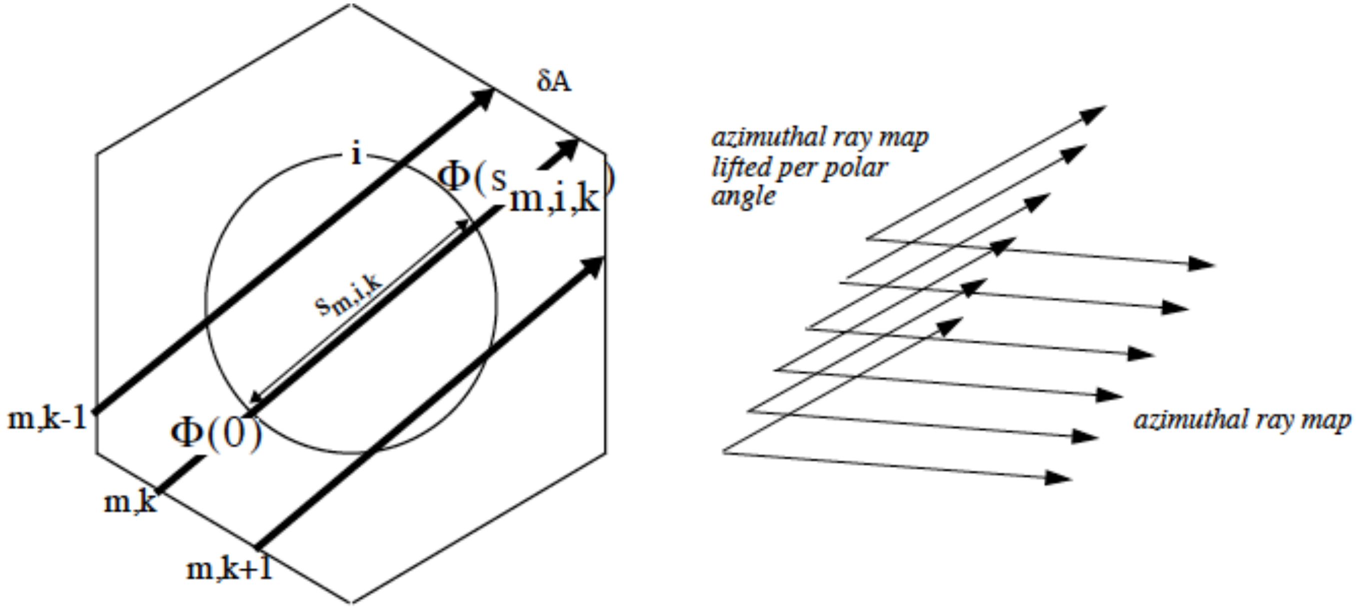


FIGURE XVIII.1. Characteristics line through volume Vi

The MOC Algorithm

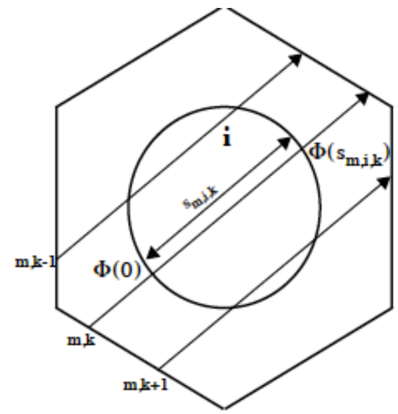


$$\boxed{\Phi_{g,m,i,k}(s_{m,i,k}) = \Phi_{g,m,i,k}(0)e^{-\Sigma_{g,i}s_{m,i,k}} + \frac{Q_{g,i}}{\Sigma_{t,g,i}}(1 - e^{-\Sigma_{g,i}s_{m,i,k}})} \qquad (3.1)$$

MOC Algorithm

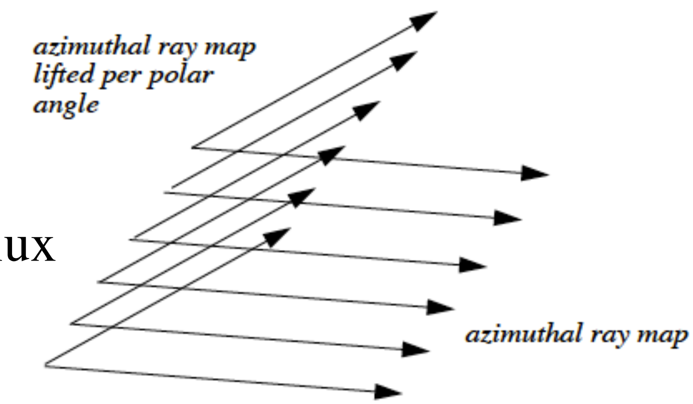
- Generate a set of rays for each direction.
- Solve ODE for a given segment:

$$\varphi_{l,m,k}^{out,g,n} = \varphi_{l,m,k}^{in,g,n} \exp\left(-\frac{\Sigma_g^n s_{l,k}^n}{\sin\theta_m} \right) + \frac{\tilde{q}_{l,m}^{g,n}}{\Sigma_g^n} \left[1 - \exp\left(-\frac{\Sigma_g^n s_{l,k}^n}{\sin\theta_m} \right) \right]$$



- Compute average angular flux for a given segment.

$$\overline{\varphi}_{l,m,k}^{g,n} = \frac{1}{s_{l,k}^n \Sigma_g^n} \left(\varphi_{l,m,k}^{in,g,n} - \varphi_{l,m,k}^{out,g,n} + \tilde{q}_{l,m}^{g,n} s_{l,k}^n \right)$$



- Integrate over angle and ray to get the scalar flux of a given region:

$$\phi_g^n = \frac{1}{V_n} \sum_m w_m \sum_l w_l \sum_{k \in n} \Delta_R^l s_{l,k}^n \left(\frac{\varphi_{l,m,k}^{in,g,k} - \varphi_{l,m,k}^{out,g,n}}{\Sigma_g^n s_{l,k}^n / \sin\theta_m} + \frac{\tilde{q}_m^{g,n}}{\Sigma_g^n} \right)$$

FIGURE XVIII.2. Angular discretization in MOC

CP Algorithm in 2D

Spatial discretization:

$$A = \sum_{i=1}^{N_r} A_i, \qquad L = \sum_{\alpha=1}^{N_b} L_\alpha$$

Spatial averaging:

$$\Phi_{g,i} = \frac{1}{A_i}\int_{A_i} dA\,\Phi_g(\vec{r})$$

$$Q_{g,i'} = \frac{1}{A_{i'}}\int_{A_{i'}} dA' \int_{4\pi} d\vec{\Omega} Q_g(\vec{r}', \hat{\Omega}), \qquad \Sigma_{t,g,i} = \frac{1}{A_i \Phi_{g,i}}\int_{A_i} dA\,\Sigma_{t,g}(\vec{r})\,\Phi_g(\vec{r})$$

$$J^{in}_{g,\alpha'} = \frac{1}{L_{\alpha'}}\int_{\alpha'} dL' \int_{\hat{n}\bullet\hat{\Omega}<0} d\hat{\Omega} J^{in}_g(\vec{r}_{L'}, \hat{\Omega}), \qquad J^{out}_{g,\alpha} = \frac{1}{L_\alpha}\int_{L_\alpha} dL \int_{\hat{n}\bullet\hat{\Omega}>0} d\hat{\Omega} J^{out}_g(\vec{r}_{L}, \hat{\Omega})$$

Assumptions: Isotropic scattering, uniform and isotropic fission source, P_0 incoming flux.

CP Algorithm in 2D

$$A_i \Sigma_{t,g,i} \Phi_{g,i} = \sum_{i'} A_{i'} Q_{g,i'} P_g (A_i \leftarrow A_{i'}) + \sum_{\alpha'} L_{\alpha'} J^{in}_{g,\alpha'} P_g (A_i \leftarrow L_{\alpha'})$$

$$L_\alpha J^{out}_{g,\alpha} = \sum_{i'} A_{i'} Q_{g,i'} P_g (L_\alpha \leftarrow A_{i'}) + \sum_{\alpha'} L_{\alpha'} J^{in}_{g,\alpha'} P_g (L_\alpha \leftarrow L_{\alpha'})$$

$$J^{in}_{g,\alpha} = J^0_{g,\alpha} + \beta_{g \to g,\alpha} J^{out}_{g,\alpha} + \sum_{g' \neq g} \beta_{g' \to g,\alpha} J^{out}_{g',\alpha}$$

where

$$g = 1, 2, \ldots, N_g; \qquad i = 1, 2, \ldots, N_r; \qquad \alpha = 1, 2, \ldots, N_b$$

The collision, escape and transmission probabilities are calculated by combination of analytical and numerical integration, and the eigenvalue equations are solved using some iterative method.

Probability Integrals in CP

Volume-to-Volume Collision Probabilities:

$$P_g(A_i \leftarrow A_{i'}) = \frac{\Sigma_{t,g,i}}{A_{i'}} \int_{A_i} dA \int_{A_{i'}} \frac{dA'}{2\pi|\vec{r} - \vec{r}'|} Ki_1\left[\tau_g(\vec{r}, \vec{r}')\right]$$

Surface-to-Volume Collision Probabilities:

$$P_g(A_i \leftarrow L_{\alpha'}) = \frac{4\Sigma_{t,g,i}}{L_{\alpha'}} \int_{A_i} dA \int_{L_{\alpha'}} \frac{dL'|\hat{n}' \bullet \hat{\Omega}_{xy}|}{2\pi|\vec{r} - \vec{r}_{L'}|} Ki_2[\tau_g(\vec{r}, \vec{r}_{L'})]$$

Probability Integrals in CP

Escape Probabilities

$$P_g(L_\alpha \leftarrow A_{i'}) = \frac{1}{A_{i'}} \int_{L_\alpha} dL \int_{A_{i'}} \frac{dA'(\hat{n} \bullet \hat{\Omega}_{xy})}{2\pi|\vec{r}_L - \vec{r}'|} Ki_2\left[\tau_g(\vec{r}_L, \vec{r}')\right]$$

Transmission Probabilities

$$P_g(L_\alpha \leftarrow L_{\alpha'}) = \frac{4}{L_{\alpha'}} \int_{L_\alpha} dL \int_{L_{\alpha'}} \frac{dL'(\hat{n} \bullet \hat{\Omega}_{xy})|\hat{n}' \bullet \hat{\Omega}_{xy}|}{2\pi|\vec{r}_L - \vec{r}_{L'}|} Ki_3[\tau_g(\vec{r}_L, \vec{r}_{L'})]$$

where the Bickley-Nayler functions are defined as

$$Ki_n(\tau) = \int_0^{\frac{\pi}{2}} d\theta (\sin\theta)^{n-1} exp\left(-\frac{\tau}{\sin\theta}\right)$$

Explicit Form of Collision Probability Integrals in 3D

- $P_{j\to i}$ **is the probability that a neutron born in region j will suffer its next collision in the region i.**
- $P^{bnd}_{k\to i}$ **is the probability that a neutron isotropically emitted from the boundary part k will suffer its next collision in the region i.**
- **According to mathematical shape of the integral equation kernel, collision probabilities can be cast in the following form:**
  - $$P_{j\to i} = \frac{\Sigma_{t,i}}{V_j}\int_{V_i} d^3r \int_{V_j} d^3r'\,\frac{exp[-\tau(\boldsymbol{r},\boldsymbol{r}')]}{4\pi|\boldsymbol{r}-\boldsymbol{r}'|^2}$$
  - $$P^{bnd}_{k\to i} = \frac{\Sigma_{t,i}}{\Gamma_k}\int_{V_i} d^3r \int_{\partial V_k} d^2r_0'\,\frac{exp[-\tau(\boldsymbol{r},\boldsymbol{r}_0')]}{\pi|\boldsymbol{r}-\boldsymbol{r}_0'|^2}\left|\boldsymbol{n}\frac{\boldsymbol{r}-\boldsymbol{r}_0'}{|\boldsymbol{r}-\boldsymbol{r}_0'|}\right|$$
- **By analogy one can derive escape and transmission probabilities:**
  - $$P^{esc}_{i\to k} = \frac{1}{V_i}\int_{\partial V_k} d^2r_0 \int_{V_i} d^3r'\,\frac{exp[-\tau(\boldsymbol{r}_0,\boldsymbol{r}')]}{4\pi|\boldsymbol{r}_0-\boldsymbol{r}'|^2}\left|\boldsymbol{n}\frac{\boldsymbol{r}'-\boldsymbol{r}_0}{|\boldsymbol{r}'-\boldsymbol{r}_0|}\right|$$
  - $$P^{trm}_{k\to \ell} = \frac{1}{\Gamma_k}\int_{\partial V_\ell} d^2r_0 \int_{\partial V_k} d^2r_0'\,\frac{exp[-\tau(\boldsymbol{r}_0,\boldsymbol{r}_0')]}{\pi|\boldsymbol{r}_0-\boldsymbol{r}_0'|^2}\left|\boldsymbol{n}\frac{\boldsymbol{r}_0-\boldsymbol{r}_0'}{|\boldsymbol{r}_0-\boldsymbol{r}_0'|}\right|\left|\boldsymbol{n}'\frac{\boldsymbol{r}_0-\boldsymbol{r}_0'}{|\boldsymbol{r}_0-\boldsymbol{r}_0'|}\right|$$

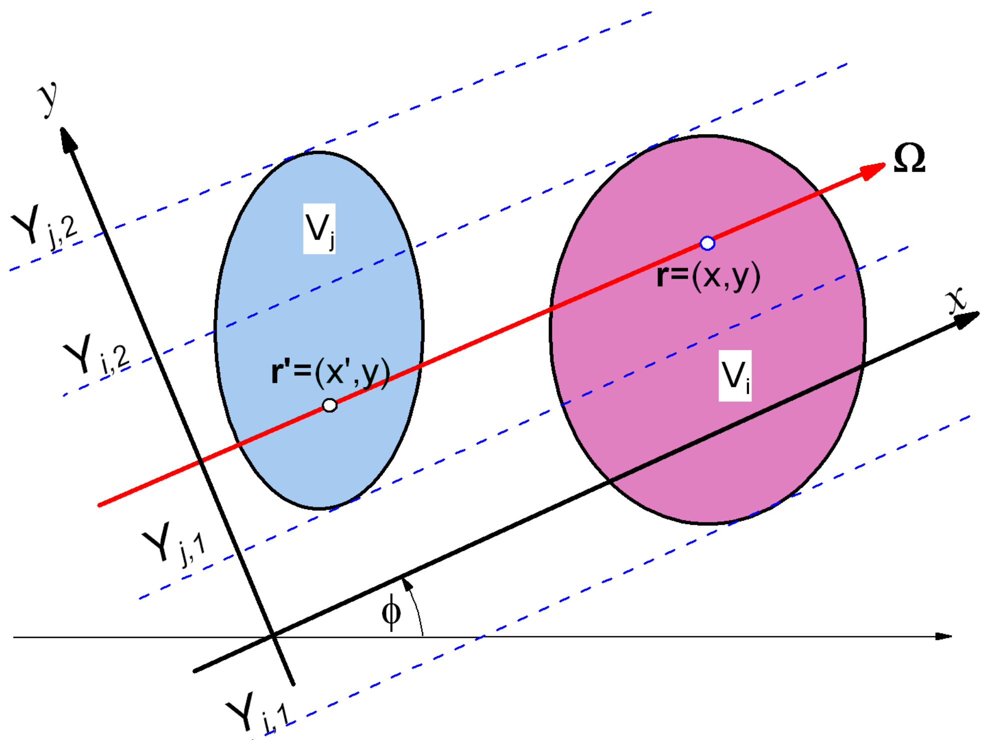
Reciprocity and Conservation Relations

- **Since the kernels are symmetric, it is easy to show that the following reciprocity relations hold:**
  - $V_i \Sigma_{t,i} P_{i\to j} = V_j \Sigma_{t,j} P_{j\to i}$
  - $V_i \Sigma_{t,i} P^{esc}_{i\to k} = \frac{1}{4} \Gamma_k P^{bnd}_{k\to i}$
  - $\Gamma_k P^{bnd}_{k\to \ell} = \Gamma_\ell P^{bnd}_{\ell\to k}$
- **Consider a neutron born in the region *i*. It is certain that it will either collide in some region *j* or escape from the volume *V* (reach the boundary ∂*V*). Thus, the sum of related probabilities equals unity.**
  - $$\sum_{i=1}^{n} P_{i\to j} + \sum_{k=1}^{m} P^{esc}_{i\to k} = 1$$
- **Similarly, it is certain that a neutron emitted from the boundary part *k* will either collide in some region *i* or pass through the volume *V* and reach the boundary ∂*V*.**
  - $$\sum_{i=1}^{n} P^{bnd}_{k\to i} + \sum_{\ell=1}^{m} P^{trm}_{k\to \ell} = 1$$

Five-Fold Integral Expression

$$P_{j\to i} = \frac{\Sigma_{t,i}}{4\pi V_j}\int_0^{2\pi} d\phi \int_{y_{min}}^{y_{max}} dy \int_{x_i-\Delta x_i}^{x_i} dx \int_{x'_j-\Delta x'_j}^{x'_j} dx' \int_0^{\pi} d\theta\, e^{-t(x,x')/\sin\theta}$$

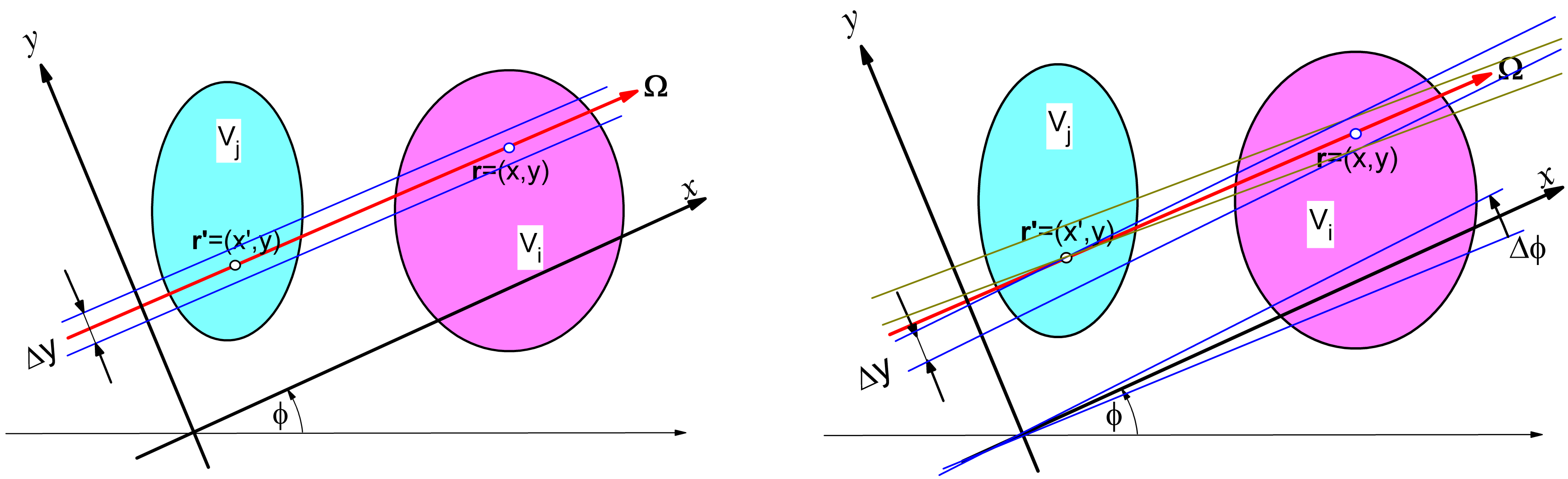
- **Each integral in the collision probability expression is over an independent variable, so that the order of integration can be changed as above.**
- **Let the x-axis is parallel to the projection of $\mathbf{\Omega}$ onto horizontal plane so that x-y Cartesian coordinate system will rotate with azimuth angle ϕ.**
- **Integration along direction $\mathbf{\Omega}$ is carried over parallel directions that intersects both regions V_i and V_j. The limits of y integration are $y_{min}=\min(Y_{i,1},Y_{j,1})$ and $y_{max}=\max(Y_{i,2},Y_{j,2})$**



Two-Fold Numerical Integration

Because of complicated geometric shapes that may occur, integration with respect to azimuth angle ϕ and Cartesian coordinate y cannot be carried out analytically. Instead, it can be evaluated numerically.

$$\int_0^{2\pi} d\phi \int_{y_{min}}^{y_{max}} dy\ f(\phi, y) \approx \sum_{k=1}^{K} w_k \sum_{\ell=1}^{L} w_\ell\ f(\phi_k, y_\ell)$$



Three-Fold Analytical Integration

$$P_{j\to i} = \frac{\Sigma_{t,i}}{4\pi V_j} \sum_{k=1}^{K} w_k \sum_{\ell=1}^{L} w_\ell \int_{x_i-\Delta x_i}^{x_i} dx \int_{x_j'-\Delta x_j'}^{x_j'} dx' \int_0^\pi d\theta\, e^{-t(x,x')/\sin\theta}$$

$$\int_0^\pi d\theta\, e^{-t(x,x')/\sin\theta} = Ki_1(t(x,x'))$$

$Ki_1(t)$ **Bickley function of the first order**

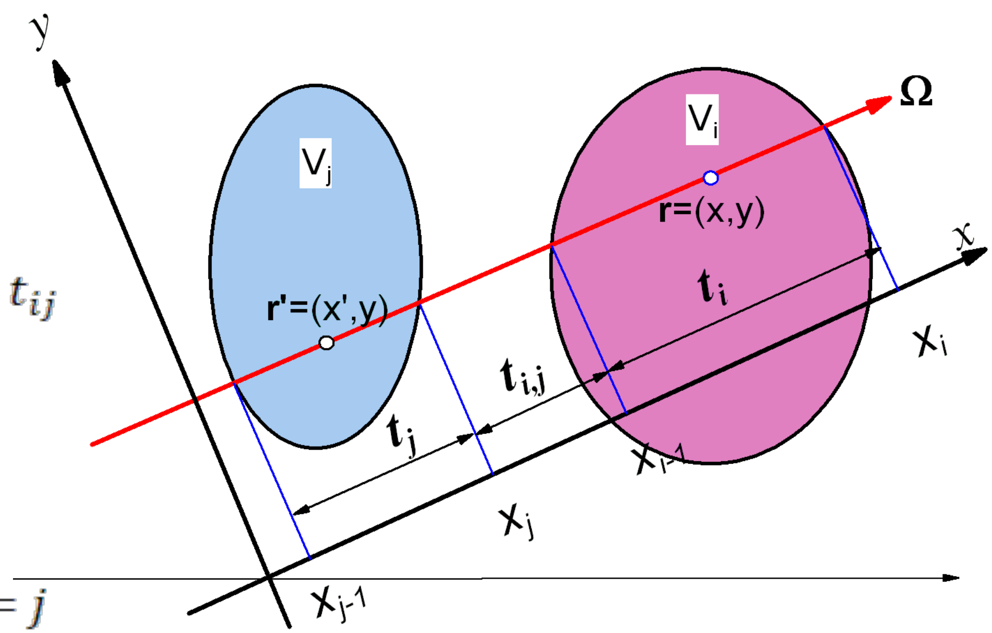
$$t(x,x') = \Sigma_{t,i}(x - x_{i-1}) + \Sigma_{t,j}(x_{j-1} - x') + t_{ij}$$

$$\int_{x_i-\Delta x_i}^{x_i} dx \int_{x_j'-\Delta x_j'}^{x_j'} dx'\; Ki_1(t(x,x')) =$$

$$= \Sigma_{t,i}\Delta x_i - [Ki_3(0) - Ki_3(t_i)], \qquad for\;\; i = j$$

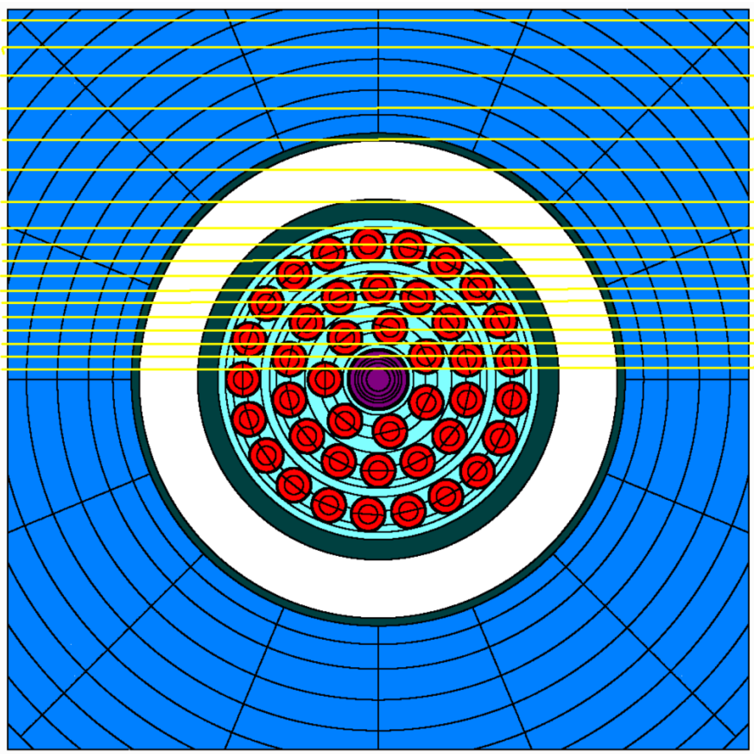
$$= Ki_3(t_{ij}) - Ki_3(t_{ij}+t_i) - Ki_3(t_{ij}+t_j) + Ki_3(t_{ij}+t_i+t_j) \qquad for\;\; i \neq j$$

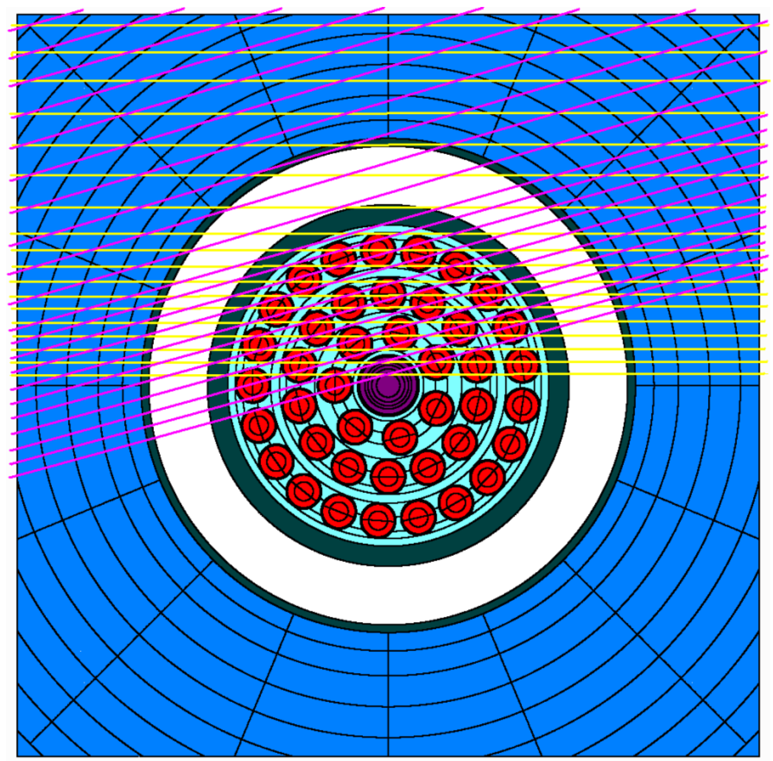
$$\lim_{\Delta y\to 0}\left(\sum_{\ell=1}^{L} w_\ell\; \Delta x_i(y = y_\ell)\right) = V_i$$



Numerical Integration by Ray Tracing

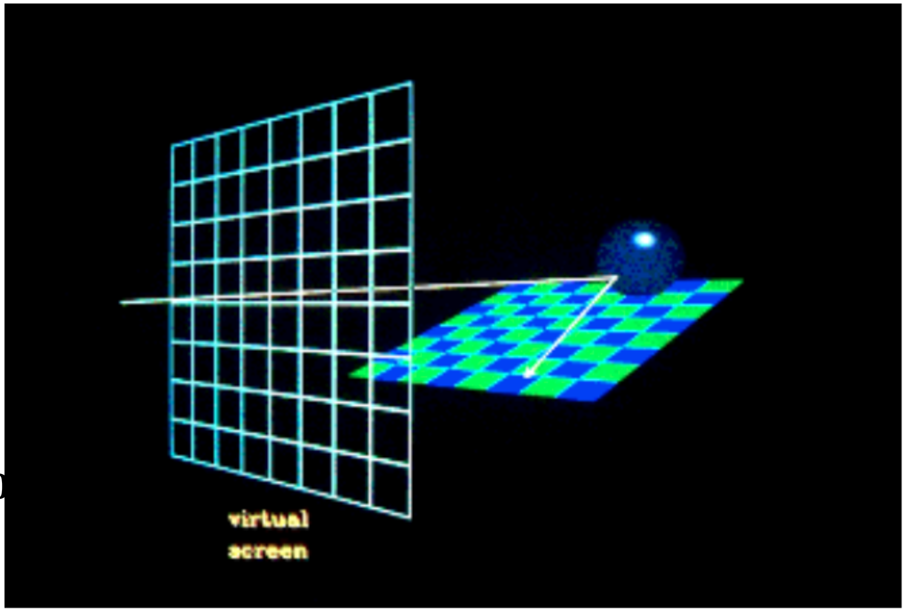
$$P_{j\to i} = \frac{\Sigma_{t,i}}{4\pi V_j} \sum_{k=1}^{K} w_k \sum_{\ell=1}^{L} w_\ell \begin{cases} \{Ki_3(t_{ij}) - Ki_3(t_{ij}+t_i) - Ki_3(t_{ij}+t_j) + Ki_3(t_{ij}+t_i+t_j)\} \\ \{\Sigma_{t,i}\Delta x_i - [Ki_3(0) - Ki_3(t_i)]\} \quad for\ i=j \end{cases}$$





Ray Tracing

- Ray tracing is an illumination based method for object visualization in computer graphics.
- A collimated beam of rays of light is traced from the eye back through the image plane into the scene.
- Each ray is tested against all objects in the scene to determine if it intersects any object.
- Mathematical equation of the ray is coupled with object surface equation to get ray/object intersections.
- Object related surface(s) are specified by solid modeling technique.
- In addition, ray tracing handles shadows, multiple specular reflections, and texture mapping.



Ray Tracing in Reactor Physics Calculations

- A variety of solid modeling methods have been developed in 1980's.
- Constructive Solid Geometry (CSG) uses Boolean operators to combine objects.
  - Largely accepted in computer graphics
  - Computer Aided Design and Manufacturing (CAD/CAM).
- R-function method of Rvachev specifies objects by means of analytic functions.
  - Any complex object the boundary of which consists of parts of analytic surfaces can be specified by a single continuous and differentiable function.
  - In addition to solid modeling, R-functions can be used to solve boundary value problems.

Analytic Domains

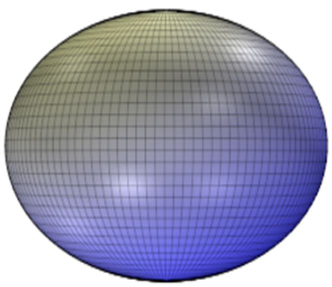
- Suppose an equation $\omega(\mathbf{r}) = 0$ is valid if and only if the point r is on a surface that represents the boundary ∂V of a closed spatial domain V.
- By a proper choice of the sign, the function $\omega(\mathbf{r})$ can be used to determine whether a given point r is inside or outside the domain, or on its boundary.

$$\omega(\boldsymbol{r}) \begin{cases} > 0\,, & \boldsymbol{r} \in V \backslash \partial V\,, \quad \textit{inside the domain} \\ = 0\,, & \boldsymbol{r} \in \partial V\,, \quad \textit{on the boundary} \\ < 0\,, & \boldsymbol{r} \notin V\,, \quad \textit{outside the domain} \end{cases}$$

- Such a spatial domain is referred to as analytic domain.

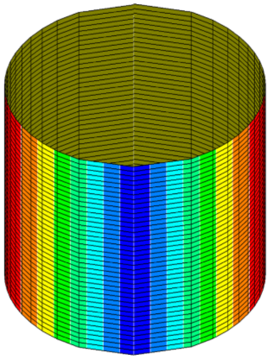
Algebraic Surfaces

- A particular class of surfaces are algebraic surfaces that are specified by algebraic equations.
- Owing to simplicity of mathematical expressions, this type of surfaces is of particular interest in engineering applications.
- Characteristic representatives are:
  - Planar surfaces specified by linear (first order) equation.
  - Quadrics specified by quadratic (second order) equation.



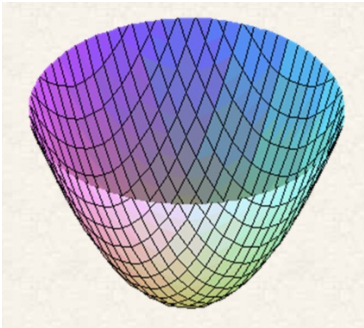
$$1-\frac{x^2}{a^2}-\frac{y^2}{b^2}-\frac{z^2}{c^2}=0$$

Ellipsoid



$$1-\frac{x^2}{a^2}-\frac{y^2}{b^2}=0$$

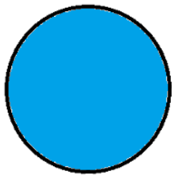
Elliptic Cylinder



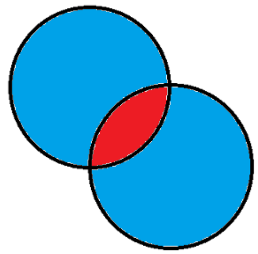
$$-\frac{x^2}{a^2}-\frac{y^2}{b^2}+2z=0$$

Elliptic Paraboloid

Set Operators
Illustrated by Euler Diagrams

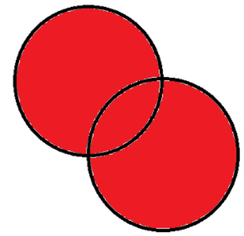


Closed Domain V



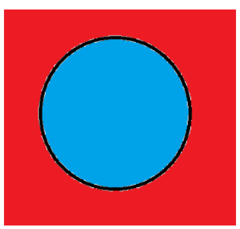
$V_1 \cap V_2$

Intersection



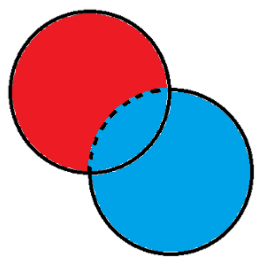
$V_1 \cup V_2$

Union



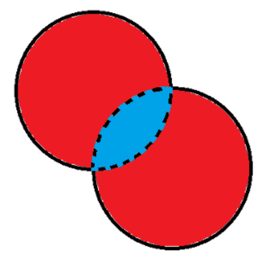
$\overline{V}$

Complement



$V_1 \setminus V_2$

Difference



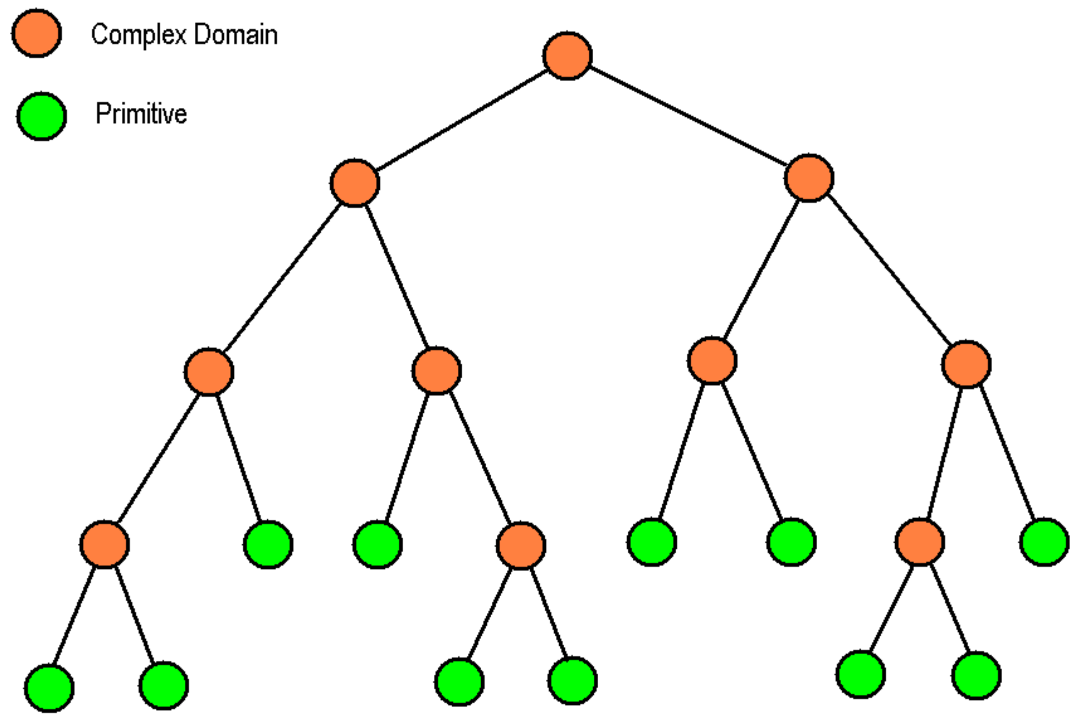
$V_1 \Delta V_2$

Symmetric Difference

Set Operator Relationships

- The following relationships hold:
  - $V_1 \cap V_2 = \overline{\overline{V_1} \cup \overline{V_2}}$
  - $V_1 \cup V_2 = \overline{\overline{V_1} \cap \overline{V_2}}$
  - $V_1 \setminus V_2 = V_1 \cap \overline{V_2}$
  - $V_1 \Delta V_2 = (V_1 \cap \overline{V_2}) \cup (V_2 \cap \overline{V_1})$
- For modeling purposes it is sufficient to use only two operators:
  - Complement, and
  - Intersection or Union.
- Other operators can be expressed as above.

Binary Tree Algorithm



Set operator expression of a solid can be represented as a binary tree.

- The simplest built-in domains (cube, cylinder, etc.) are referred to as primitives.
- Branching nodes represent a set operation on two previously defined domains.

Set Membership Classification

- **Starting from some simple domains, using set operators one can easily build complex domains.**
- **However, set operators cannot give an answer to set membership classification (point inclusion) problem.**
- **To resolve this problem, one needs to construct a three-valued function *T*(*r*) with following properties:**

$$T(\boldsymbol{r}) = \begin{cases} 2\,, & \boldsymbol{r} \in V \backslash \partial V, \qquad \textit{inside the domain} \\ 1\,, & \boldsymbol{r} \in \partial V, \qquad \textit{on the boundary} \\ 0\,, & \boldsymbol{r} \notin V, \qquad \textit{outside the domain} \end{cases}$$

- **For an analytic domain described by ω(*r*) = 0, *T*(*r*) can be easily constructed as** $T(\boldsymbol{r}) = 1 + sign\ \omega(\boldsymbol{r})$

- **What about semi-analytic (complex) domains?**

Three-Valued Operators for Semi-Analytic Domains

- **Association of three-valued operators and set operators**
  - *Complement:* $\overline{V} \Rightarrow \overline{T}$ *Negation (NOT)*
  - *Intersection:* $V_1 \cap V_2 \Rightarrow T_1 \wedge T_2$ *Conjunction (AND)*
  - *Union:* $V_1 \cup V_2 \Rightarrow T_1 \vee T_2$ *Disjunction (OR)*

- **Definition of three-valued operators**
  - Negation
  - Conjunction
  - Disjunction

$$\overline{T}(\boldsymbol{r}) = 2 - T(\boldsymbol{r})$$

$$T_1 \wedge T_2 = \min(T_1, T_2)$$

$$T_1 \vee T_2 = \max(T_1, T_2)$$

T_1	T_2	$\overline{T_1}$	$T_1 \wedge T_2$	$T_1 \vee T_2$
2	2	0	2	2
1	2	1	1	2
0	2	2	0	2
2	1		1	2
1	1		1	1
0	1		0	1
2	0		0	2
1	0		0	1
0	0		0	0

Ray/Solid Intersection

- **General algorithm:**
  - For each ray, determine intersection points with all surfaces.

    Spatial coordinates in surface equation are replaced with straight-line parametric equations in order to determine the distance from ray origin to the intersection point.
  - Reorder intersection points by distance, from nearest to farthest.

    It is a simple computer operation.
  - Check each ray/surface intersection point whether it belongs to a solid boundary or not.

    For each analytic domain the related analytic function is evaluated at the intersection point, then either three-valued function or R-function is calculated according to set operator expression and tested against 1 or 0, respectively.

Intersection with a Planar Surface

- **The problem reduces to the solution of a linear algebraic equation.**
- **The solution (the distance *s* between ray origin and intersection point) takes the following form:**

$$s = \frac{\omega(\boldsymbol{r}_0)}{\Omega_x \cos\alpha_x + \Omega_y \cos\alpha_y + \Omega_z \cos\alpha_z}$$

- **If the denominator is equal to zero, there is no intersection, and two particular cases are possible:**
  - $\omega(\boldsymbol{r}_0) \neq 0, \quad the\ ray\ is\ parallel\ to\ the\ surface$
  - $\omega(\boldsymbol{r}_0) = 0, \quad the\ ray\ is\ on\ the\ surface$

Intersection with a Quadric

- **The problem reduces to the solution of a quadratic algebraic equation.**
- **There are two solutions in the form:**

$$s_{1/2} = -\frac{1}{P}\left(Q \pm \sqrt{Q^2 + PF}\right) \quad \text{where} \quad F = \omega(\boldsymbol{r}_0)$$

- **Depending on P, Q and F, following cases may occur:**

$$Q^2 + PF \begin{cases} < 0 & \text{No intersection} \\ = 0 \begin{cases} P = 0 \begin{cases} F \neq 0 & \text{No intersection} \\ F = 0 & \text{Ray on surface, no intersection} \end{cases} \\ P \neq 0 & \text{Double intersection point} \end{cases} \\ > 0 \begin{cases} P = 0 \text{ and } Q = 0 & \text{One intersection} \\ P \neq 0 & \text{Two intersections} \end{cases} \end{cases}$$

Speeding-Up Ray Tracing Calculations

- **Depending on the field of application, the number of rays may be very high.**
  - **About one million of primary rays (one ray per pixel) is necessary for each frame in computer graphics.**
  - **Tens of millions of histories (initial rays) are usually applied in Monte Carlo simulations.**
- **For a straight forward ray-tracing algorithm, the computing time would be proportional to the product of the number of rays and the number of objects.**
- **Dealing with a large number of objects, the ray tracing calculations may become extremely time consuming.**

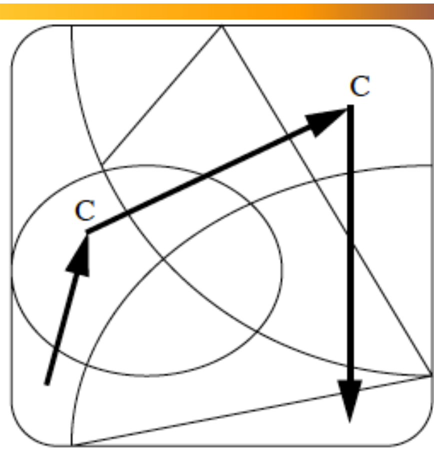
Accelerating Techniques

- Various approaches have been used to speed-up the ray tracing (faster machines, special hardware, more efficient algorithms, etc.).
- The general goal is to reduce the number of ray/surface intersections.
- An efficient way to do this is to subdivide the space into a number of sub-spaces.
  - Artificial nested dissection into octants.
  - Hierarchical bounding volumes each of which encompass parts of the space with many details.

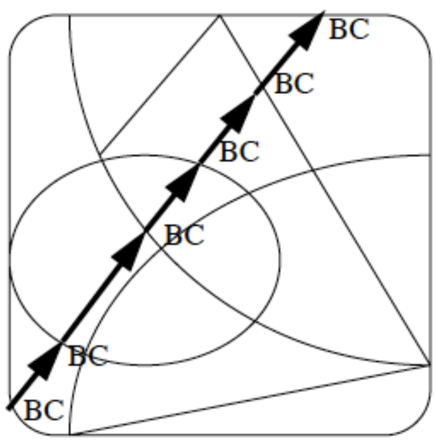
Ray-Tracing Acceleration in Reactor Physics Model

- **According to the design of the current reactors, the nested subdivision into bounding volumes seems to be very efficient technique for speeding-up ray-tracing calculations in reactor physics problems.**
  - **The reactor vessel on a whole is a bounding volume that contains almost all details of primary interest.**
  - **Reactor lattice consists of a number of assemblies (PWR) or fuel channels (CANDU).**
  - **Each assembly can be subdivided into a rectangular or hexagonal mesh of pin cells.**
  - **In a channel type reactor, the pressure tube encompasses the fuel bundles.**

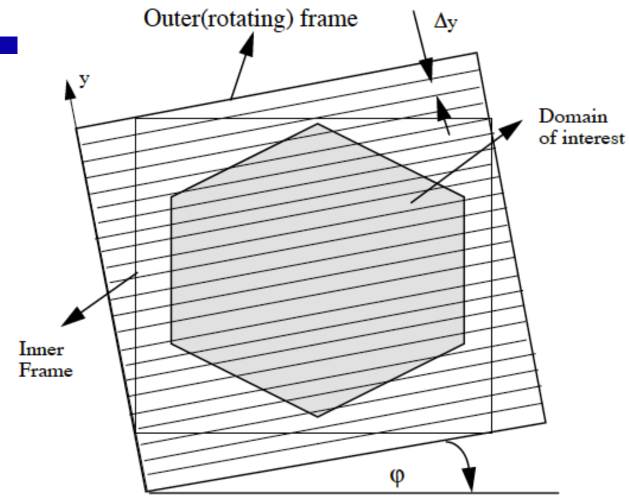
Ray Tracing in GTRAN2/MAGGENTA/MOCHA VS. Monte Carlo



a) Monte Carlo ray tracing



b) CP or MOC ray tracing



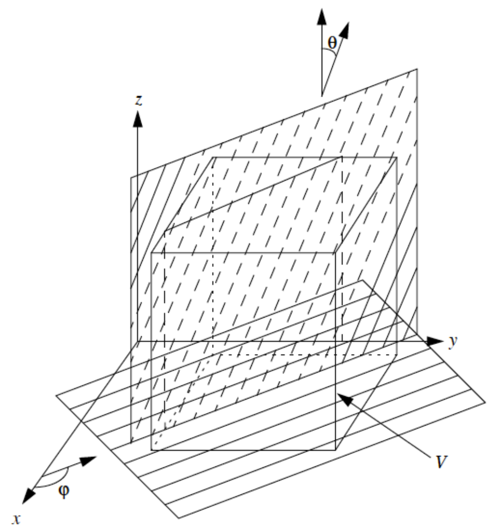


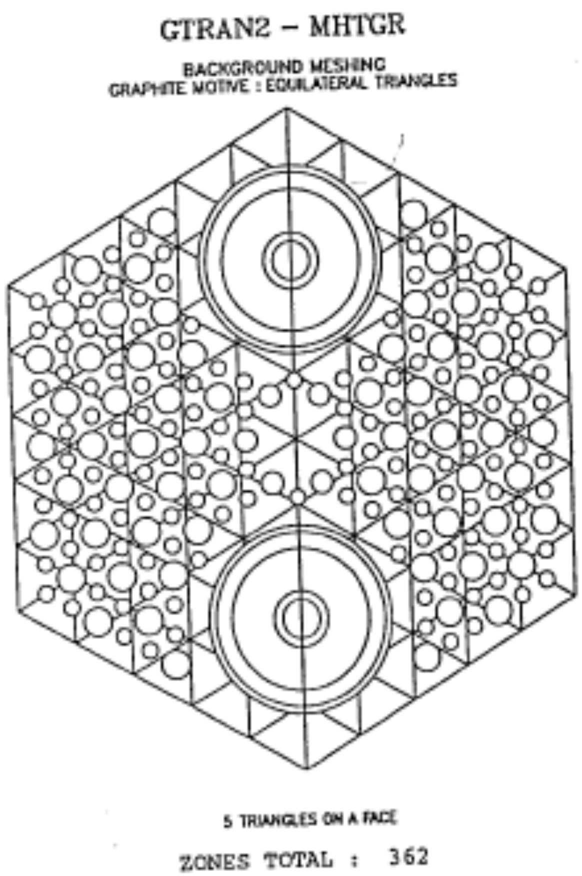
Figure 11 - 3D Parallel Integration Lines

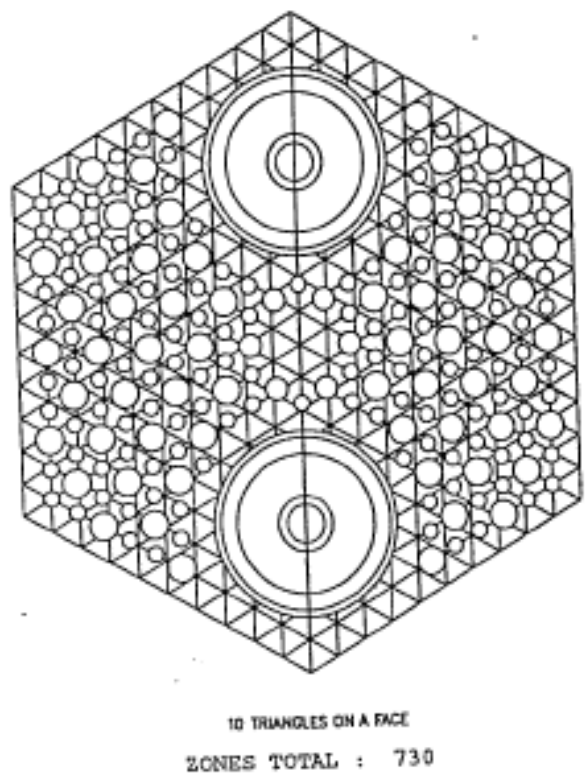
37

Monte Carlo Ray Tracing vs. MOC and CP Ray Tracing

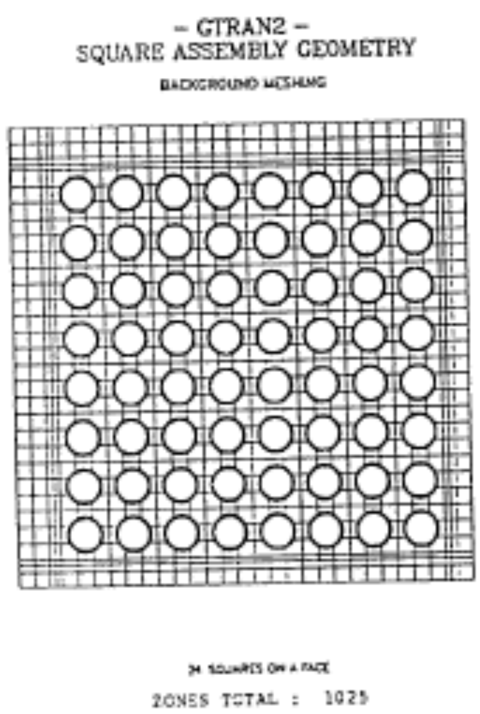
- The same use combinatorial geometry to describe the general problem and all regions.
- Thus, MOC and CP could treat various geometries without any changes to their equations and algorithms.
- The same ray tracing, different approach:
- MC – do ray tracing to the first boundary/collision point, and then do the physics.
- In MOC and CP, the rays, interception points, and track lengths could be pre-calculated in CP and MOC and reused as long as dimensions do not change.
- However, due to the flat flux/flat source approximation, CP and MOC do need more meshes that MC.
- The multigroup cross sections (transport corrected) could be calculated by MC for MOC and CP.
- Mesh generator needed.

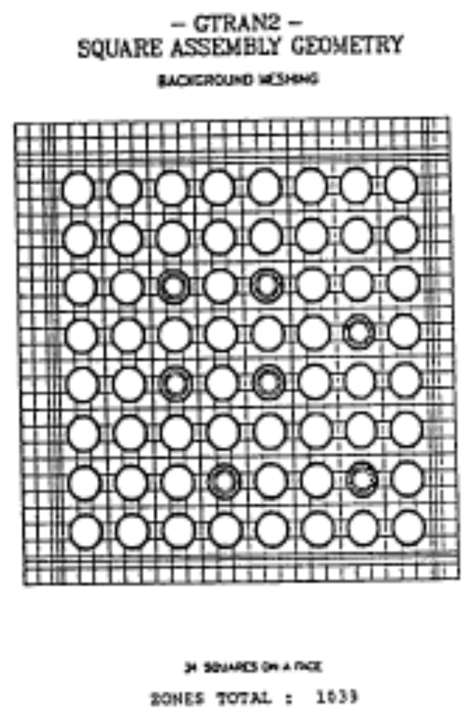
CP or MOC Meshing Needed for the MHTGR Assembly

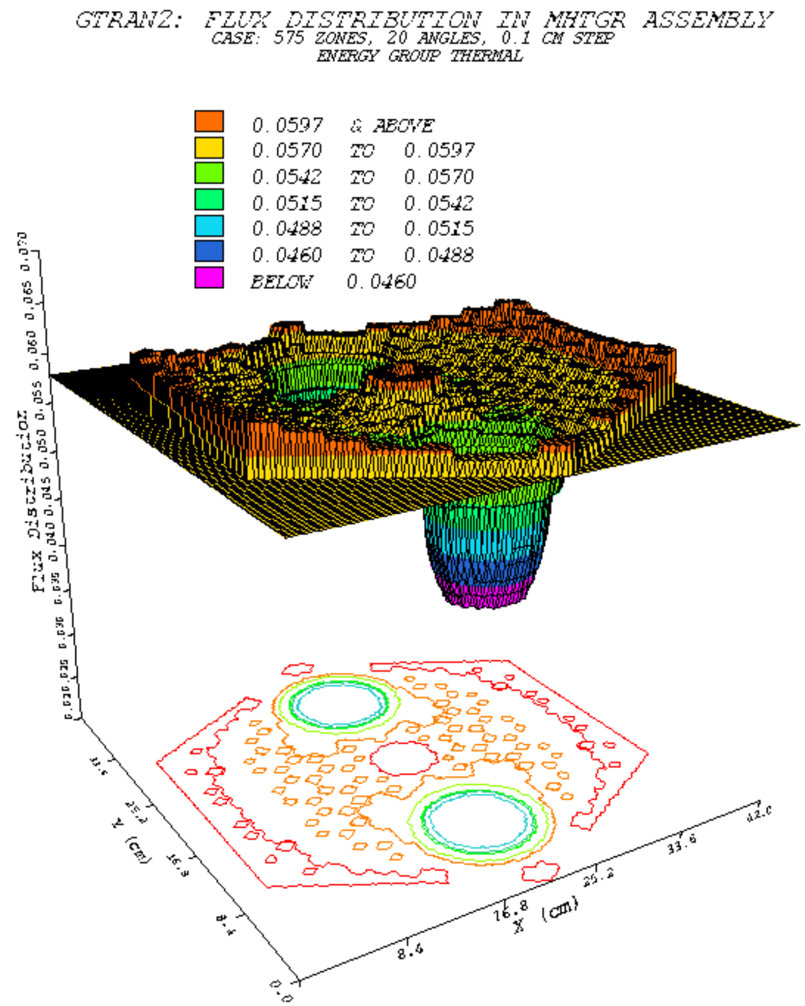


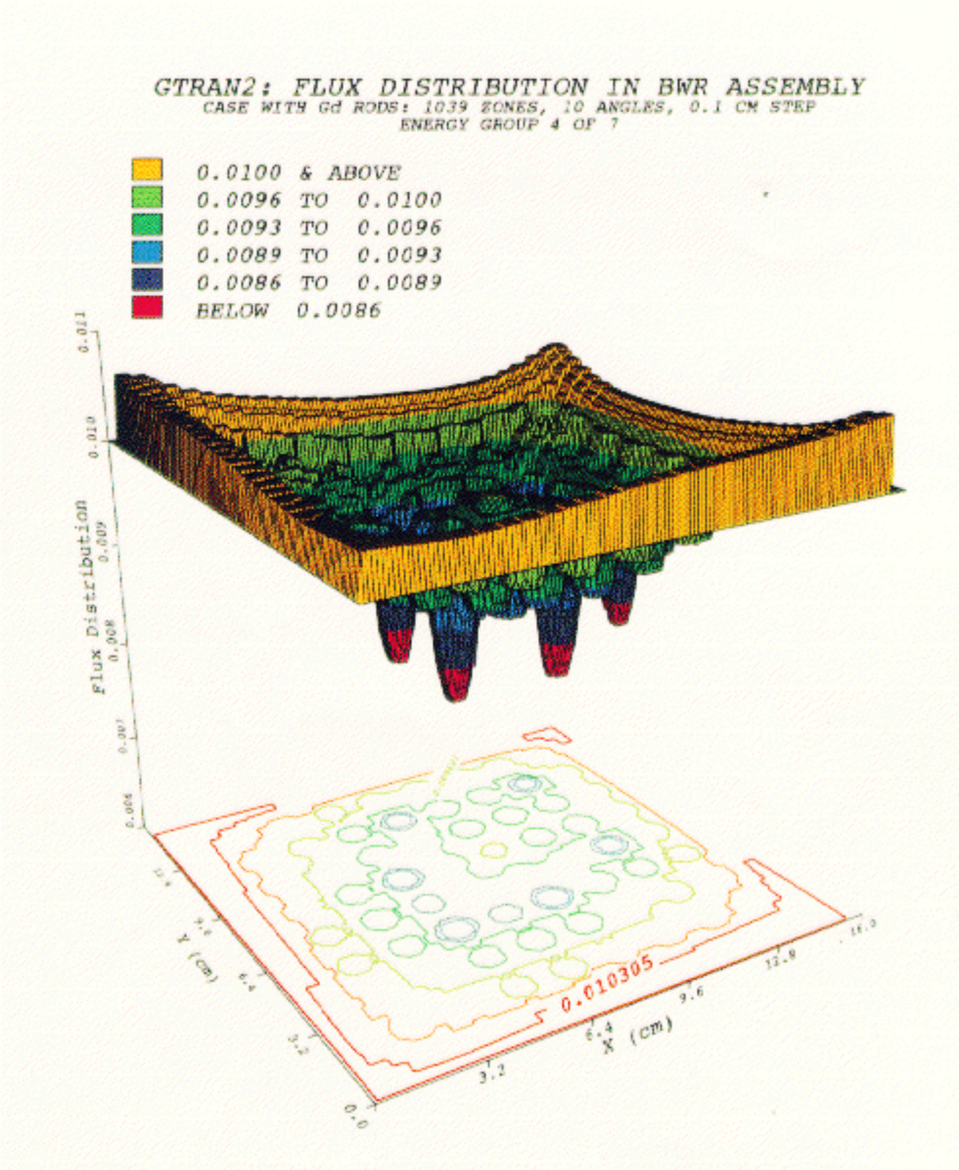


CP and MOC Meshing Needed for the BWR Assembly






GTRAN2: FLUX DISTRIBUTION IN MHTGR ASSEMBLY
CASE: 575 ZONES, 20 ANGLES, 0.1 CM STEP
ENERGY GROUP THERMAL
0.0597 & ABOVE
0.0570 TO 0.0597
0.0542 TO 0.0570
0.0515 TO 0.0542
0.0488 TO 0.0515
0.0460 TO 0.0488
BELOW 0.0460
Flux Distribution
Y (cm)
X (cm)


GTRAN2: FLUX DISTRIBUTION IN BWR ASSEMBLY
CASE WITH Gd RODS: 1039 ZONES, 10 ANGLES, 0.1 CM STEP
ENERGY GROUP 4 OF 7
0.0100 & ABOVE
0.0096 TO 0.0100
0.0093 TO 0.0096
0.0089 TO 0.0093
0.0086 TO 0.0089
BELOW 0.0086
Flux Distribution
0.010305
Y (cm)
X (cm)

Reactor Physics Computational Methods

- **Diffusion Theory**
  - discretized space
  - linearly anisotropic direction
  - discretized energy (few-group)
- **Deterministic Transport**
  - discretized space
  - discretized direction (discrete ordinates) or functional expansion of direction (spherical harmonics)
  - discretized energy (multi-group)
- **Monte Carlo**
  - continuous spatial representation
  - continuous direction representation
  - continuous energy representation

increasing accuracy

increasing run-time

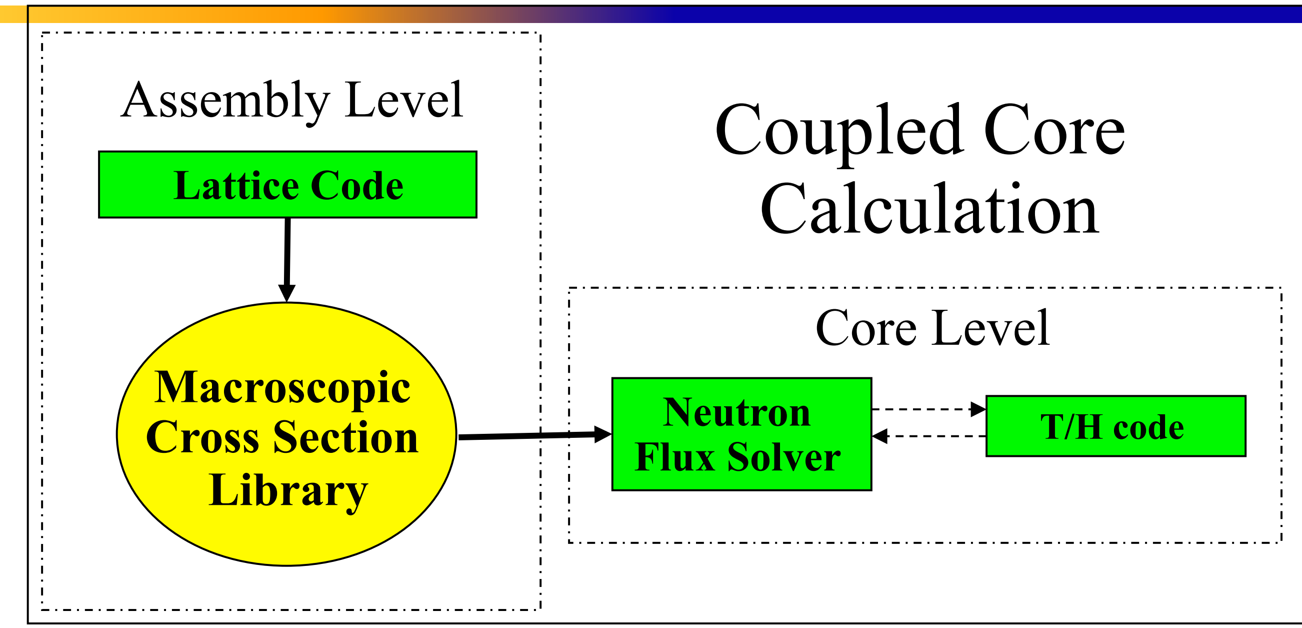
Methods used to perform 3D core calculations

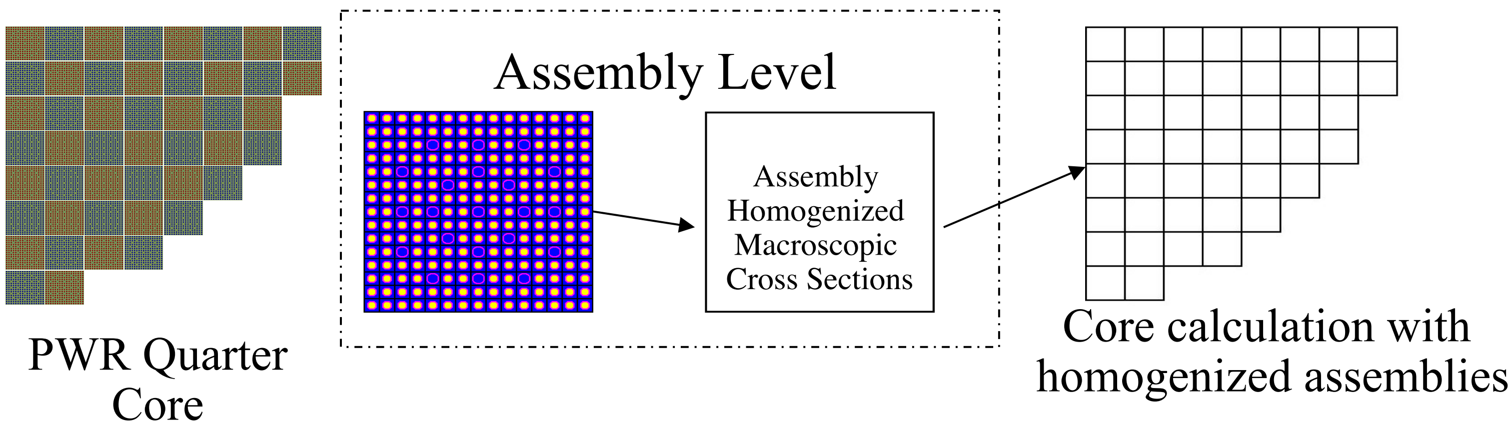
"Cheap" Calculations

<table>
<tr><th colspan="3">Method</th><th>Codes Examples</th><th>Advantages</th><th>Drawbacks</th></tr>
<tr><td colspan="3">Deterministic Methods (Diffusion Theory)</td><td>PARCS</td><td>Fast</td><td>Homogenization
Accuracy requires fine meshing</td></tr>
<tr><td rowspan="3">Deterministic Methods (Transport Theory)</td><td colspan="2">Differential Formulation
Discrete Ordinates</td><td>THREEDANT
TORT</td><td>Less expensive</td><td>Ray Effect
Negative scalar flux</td></tr>
<tr><td rowspan="2">Integral Formulation</td><td>Collision Probability Method</td><td>DRAGON
APOLLO
GTRAN2</td><td>Easy to handle 3D</td><td>Expensive</td></tr>
<tr><td>Method of Characteristics</td><td>DRAGON
DeCART,
MOCHA</td><td>Accurate</td><td>Expensive</td></tr>
<tr><td colspan="3">Stochastic Method (Monte Carlo)</td><td>MCNP
SERPENT</td><td>Very Accurate</td><td>Computationally expensive
No transient capability</td></tr>
</table>

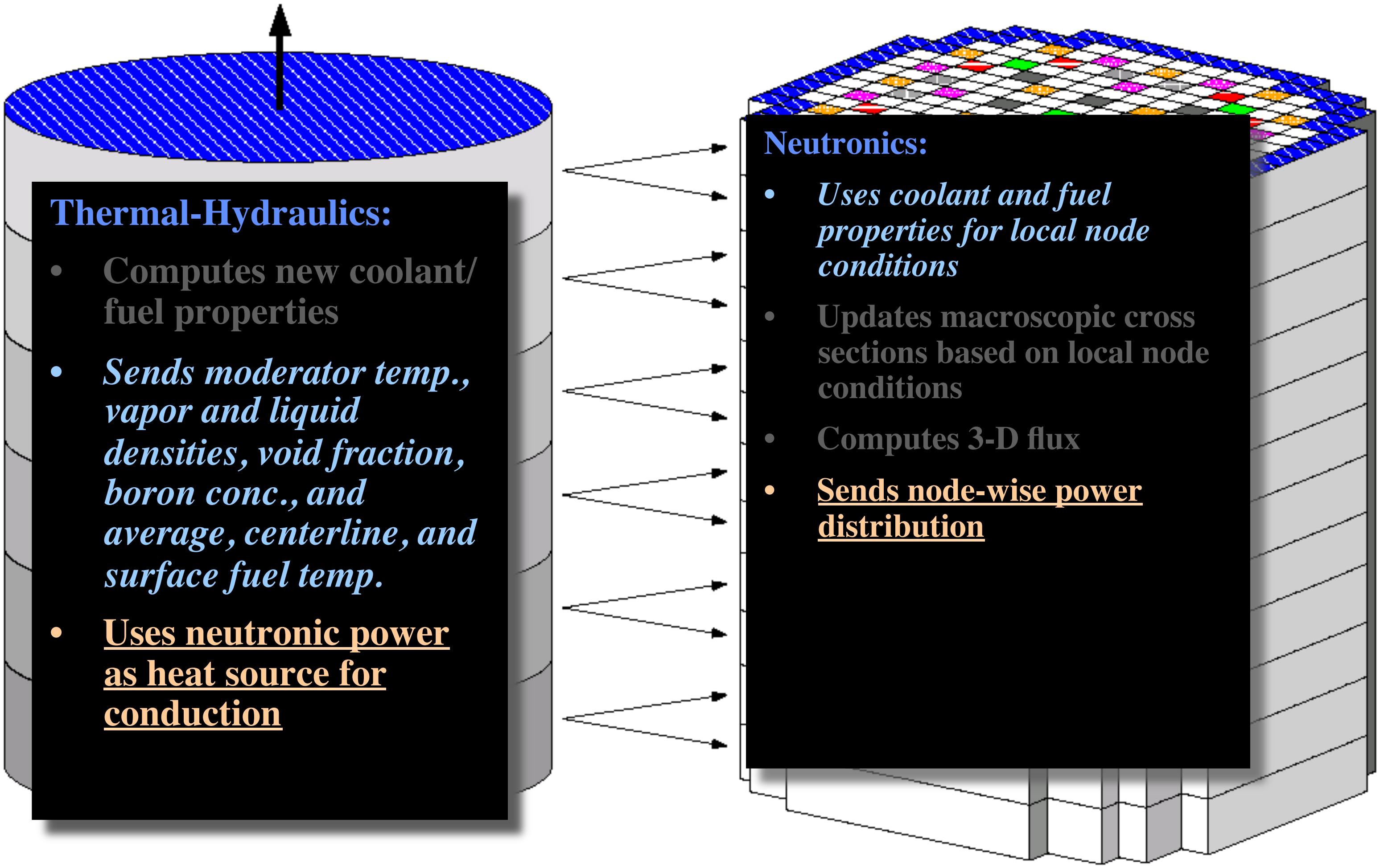
Accuracy

Current methodology for core calculation

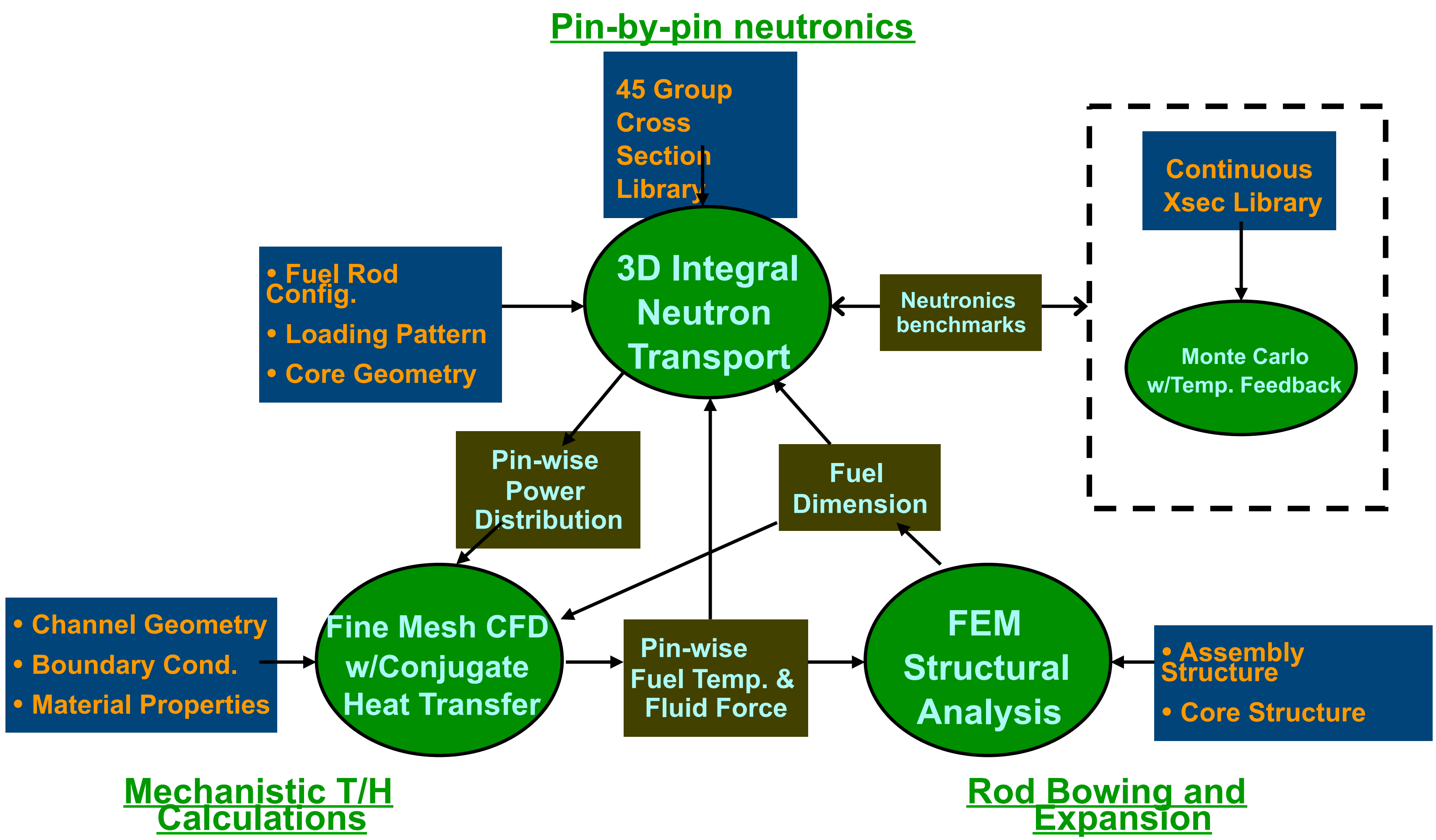




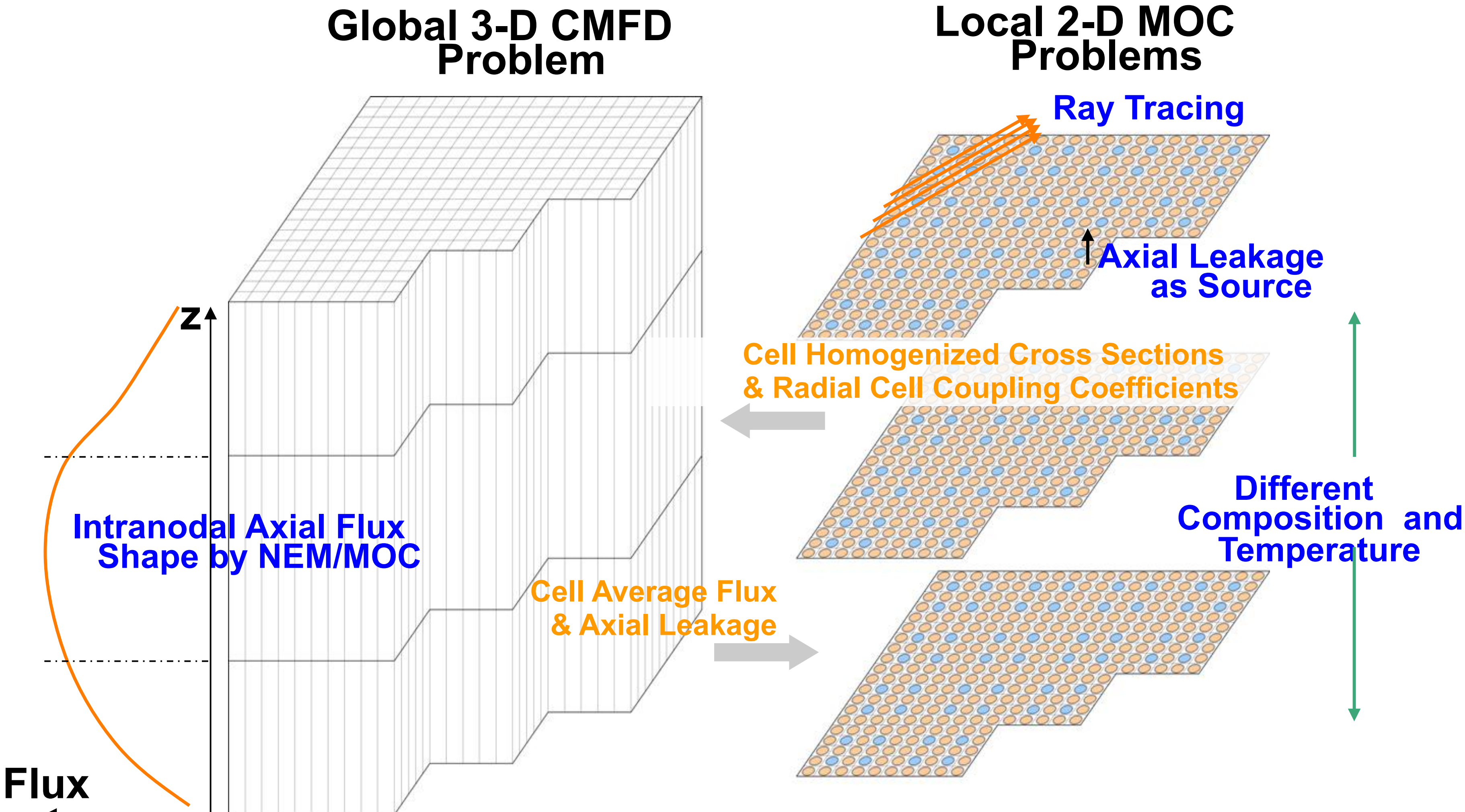
Code coupling: Spatial Coupling



The Numerical Nuclear Reactor Software System
(T. Downar)



Decompose the 3D Problem into 2D/1D



CMFD – Coarse Mesh Finite Difference, NEM – Nodal Expansion Method, MOC – Method of Characteristics

Current Status of Monte Carlo Methods

- MC codes model “exact” geometries (curved surfaces) and ”exact” physics (no multigroup approximation).
- MC codes are most efficient when limited information is aought (a reaction rate, an eigenvalue – k), and are least efficient when global information is sought.
- MC solutions have statistical errors but no truncation errors.
- MC codes are expensive to run and require significant user input
- Research is being done to develop more efficient non-analog (variance reduction) techniques.
- Research is being done to develop more reliable statistical techniques for estimating MC sampling errors.

Current status of Deterministic (S_N) Methods

- S_N codes have a limited (by expanding) capability of modeling complex geometries.
- These codes automatically produce global solutions.
- S_N solutions have truncation errors, but no statistical errors.
- S_N codes are less expensive to run that MC codes and require less user input.
- Current research topics include:
  - Fixing the ray effects
  - Developing noncartesian spatial grid
  - Developing more efficient iterative methods.

Integral Form of Neutron Transport Equation for Scalar Flux – General Boundary Conditions

$$\phi(r) = \int_{4\pi} d\Omega \psi(r_s, \Omega) \exp[-\tau(r, r_s)] + \int_{4\pi} d\Omega \int_0^{R(\Omega)} dR' Q(r', \Omega) \exp[-\tau(r, r')]$$

$$d\Omega = \frac{dS|e_s \bullet \Omega|}{|r - r_s|^2}$$

$$\phi(r) = \int_V dr' \left(Q(r', \Omega) \frac{\exp[-\tau(r, r')]}{|r - r'|^2} \right. + \int_S \frac{dS}{|r - r_s|^2} J^{in}(r_s, \Omega) exp[-\tau(r, r_s)],$$

$$r' = r - R\Omega, \quad r_s = r - R_s\Omega, \quad \Omega = \frac{r - r'}{|r - r'|},$$

$$J^{in}(r_s, \Omega) = |e_s \bullet \Omega| \psi(r_s, \Omega), \text{ with } e_s \bullet \Omega < 0$$

$$J^{out}(r_{s'}) = \int_V \frac{dr'}{|r - r_{s'}|^2} (e_{s'} \bullet \Omega) Q(r', \Omega) exp[-\tau(r, r_s)] \qquad + \int_S \frac{dS}{|r_s - r_{s'}|^2} (e_{s'} \bullet \Omega) J^{in}(r_s, \Omega) exp[-\tau(r_{s'}, r_s)],$$