DRAGON Theory for 3-D CANDU Problems

G. Marleau

Institut de génie nucléaire

École Polytechnique de Montréal

ÉCOLE POLYTECHNIQUE MONTRÉAL

Summary 1

1. Introduction to the Collision Probability Method
2. 3–D Collision Probability Calculations
3. 3–D DRAGON Examples
4. Solving the Collision Probability Equations
5. Condensation and Homogenization Techniques
6. Managing a DRAGON Execution
7. Discussion and Conclusion

ÉCOLE POLYTECHNIQUE MONTRÉAL

Introduction to the CP Method 1

Contents

- The Transport Equation
- The Collision Probability Technique
- Boundary Conditions
- Cross Sections Considerations

The Transport Equation 1

The transport equation is a neutron balance equation

$$\mathcal{L}(\vec{r},E,\vec{\Omega}) = \mathcal{Q}(\vec{r},E,\vec{\Omega})$$

- $\mathcal{L}$ represents neutron lost from the system:

$$\mathcal{L} = \vec{\Omega}\cdot\vec{\nabla}\Phi(\vec{r},E,\vec{\Omega}) + \Sigma(\vec{r},E)\Phi(\vec{r},E,\vec{\Omega})$$

- $\mathcal{Q}$ represents neutron created in the system

$$\mathcal{Q}_s = \int dE' d^2\Omega \Sigma_s(\vec{r},E' \to E,\vec{\Omega}' \to \vec{\Omega})\Phi(\vec{r},E',\vec{\Omega}')$$

$$\mathcal{Q}_f = \chi(\vec{r},E)\int dE' d^2\Omega \nu\Sigma_f(\vec{r},E')\Phi(\vec{r},E',\vec{\Omega}')$$

The Transport Equation 2

Multigroup transport equation

$$\left[\vec{\Omega}\cdot\vec{\nabla}+\Sigma^g(\vec{r})+D^g(\vec{r})B^2\right]\Phi^g(\vec{r},\vec{\Omega})=Q_s^g(\vec{r},\vec{\Omega})+\frac{1}{k}Q_f^g(\vec{r},\vec{\Omega})$$

Scattering source $Q_s^g(\vec{r},\vec{\Omega})$

$$Q_s^g(\vec{r},\vec{\Omega})=\sum_{h=1}^{G}\int d^2\Omega'\Sigma_s^{hg}(\vec{r},\vec{\Omega}'\to\vec{\Omega})\Phi^h(\vec{r},\vec{\Omega}')$$

Fission source $Q_f^g(\vec{r},\vec{\Omega})$

$$Q_f^g(\vec{r})=\chi^g\sum_{h=1}^{G}\nu\Sigma_f^h(\vec{r})\int d^2\Omega'\Phi^h(\vec{r},\vec{\Omega}')$$

The Transport Equation 3

Transport equation in the absence of external sources is an eigenvalue problem:

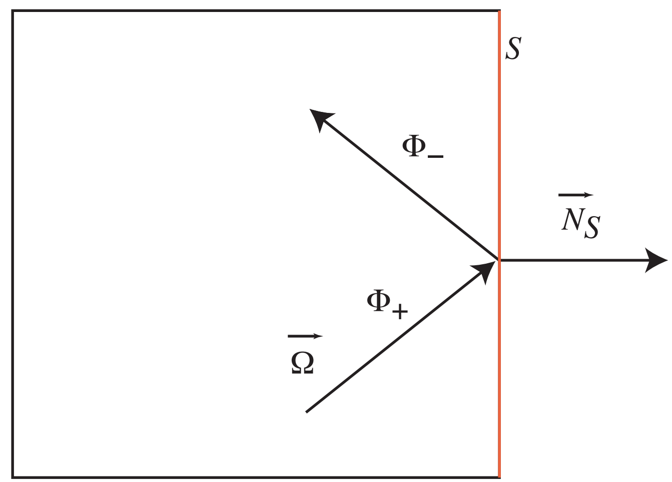
1. k eigenvalue with imposed leakage ($D^g(\vec{r})B^2$ fixed):
   - k indicates how the fission rate should be modified to make the system critical (reach a non-trivial solution to the transport equation)
2. Buckling eigenvalue with imposed k:
   - $D^g(\vec{r})B^2$ represent the amount of leakage required to make the system critical

The Transport Equation

Boundary conditions are required to close the system

- Albedo conditions

$$\phi_-(\vec{r}_S, \vec{\Omega} - 2(\vec{N}_S \cdot \vec{\Omega})) = \beta(\vec{r}_S, \vec{\Omega})\phi_+(\vec{r}_S, \vec{\Omega})$$

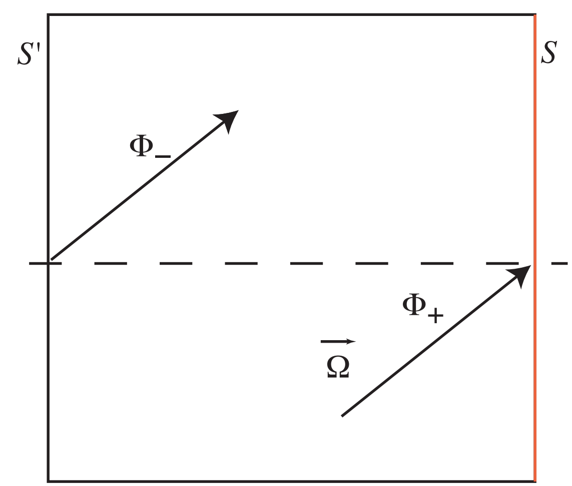


$\beta(\vec{r}_S, \vec{\Omega}) = 0$ for void BC
$\beta(\vec{r}_S, \vec{\Omega}) = 1$ for reflective BC

The Transport Equation 5

- Periodic conditions

$$\phi_-(\vec{r}_S, \vec{\Omega}) = \phi_+(\vec{r}_{S'}, \vec{\Omega})$$



The Transport Equation

Integral transport equation (case without leakage)

- Flux at a point $\vec{r}$ due to neutrons created at any point $\vec{r}\,' = \vec{r} - R\vec{\Omega}$ surrounding it

$$\left[-\frac{d}{dR} + \Sigma^g(\vec{r} - R\vec{\Omega})\right]\Phi^g(\vec{r} - R\vec{\Omega}, \vec{\Omega}) = Q^g(\vec{r} - R\vec{\Omega}, \vec{\Omega})$$

- Integrate transport equation over R and $\vec{\Omega}$

$$\phi^g(\vec{r}) = \int d^2\Omega \; e^{-\tau^g(R_S)}(\vec{\Omega}\cdot\vec{N}_-)\Phi^g_-(\vec{r}\,'_S, \vec{\Omega})\Theta(\vec{r}, \vec{r}\,'_S, \vec{\Omega})$$
$$+ \int d^2\Omega \int_0^R e^{-\tau^g(R')} Q^g(\vec{r}\,', \vec{\Omega})\Theta(\vec{r}, \vec{r}\,', \vec{\Omega}) dR'$$

The Transport Equation

Definitions

$$\phi^g(\vec{r}) = \int d^2\Omega \Phi^g(\vec{r}, \vec{\Omega})$$

$$\tau^g(R) = \int_O^R \Sigma^g(\vec{r} - R'\vec{\Omega}) dR'$$

$$\Theta(\vec{r}, \vec{r}\,', \vec{\Omega}) = \begin{cases} 1 & \text{if } \vec{r} = \vec{r}\,' + R'\vec{\Omega} \\ 0 & \text{otherwise} \end{cases}$$

and $\Phi^g_-(\vec{r}\,'_S, \vec{\Omega})$ is the incoming angular flux on surface S

The Transport Equation 8

Equation for the outgoing flux $\phi_+(\vec{r}_S)$ at S

$$\phi_+^g(\vec{r}_S) = \int d^2\Omega(\vec{\Omega}\cdot\vec{N}_+)\int_0^R e^{-\tau^g(R')}Q^g(\vec{r}\,',\vec{\Omega})\Theta(\vec{r}_S,\vec{r}\,',\vec{\Omega})dR'$$

$$+\int d^2\Omega(\vec{\Omega}\cdot\vec{N}_+)(\vec{\Omega}\cdot\vec{N}_-)\;e^{-\tau^g(R_S)}\Phi_-^g(\vec{r}\,'_S,\vec{\Omega})\Theta(\vec{r}_S,\vec{r}\,'_S,\vec{\Omega})$$

where

$$\phi_+^g(\vec{r}_S) = \int d^2\Omega\;\Phi_+^g(\vec{r}_S,\vec{\Omega})(\vec{\Omega}\cdot\vec{N}_+)$$

The Transport Equation

CP approximations

- Divide domain into N_V regions of volume V_i where the cross sections sources are independent of $\vec{r}$ and $\vec{\Omega}$

$$\Sigma^g(\vec{r}) = \Sigma^g_j \qquad \text{for } \vec{r} \in V_j$$

$$Q^g(\vec{r}, \vec{\Omega}) = \frac{1}{4\pi} q^g(\vec{r}) = q^g_j \qquad \text{for } \vec{r} \in V_j$$

(the source is assumed isotropic)

- Divide the external boundary S into N_S surfaces of area S_α and assume angular flux constant on these surfaces

$$\Phi^g_-(\vec{r}_S, \vec{\Omega}) = \frac{1}{4\pi} \phi^g_{\alpha,-}$$

The Transport Equation

$$\int d^2\Omega \int_0^R e^{-\tau^g(R')} Q^g(\vec{r}\,', \vec{\Omega})\Theta(\vec{r}, \vec{r}\,', \vec{\Omega}) dR' =$$

$$q_j^g \int_{V_i} e^{-\tau^g(R')} \Theta(\vec{r}, \vec{r}\,', \vec{\Omega}) \frac{d^3 r}{4\pi |\vec{r} - \vec{r}\,'|^2}$$

$$\int d^2\Omega \; e^{-\tau^g(R_S)} \Phi_-^g(\vec{r}\,'_S, \vec{\Omega})\Theta(\vec{r}, \vec{r}\,'_S, \vec{\Omega}) =$$

$$\phi_{\alpha,-}^g \int_{S_\alpha} e^{-\tau^g(R_S)} \Theta(\vec{r}, \vec{r}\,'_S, \vec{\Omega}) \frac{d^2 r}{4\pi |\vec{r} - \vec{r}\,'|^2}$$

The Transport Equation

Transport equations in CP form

$$\phi_i^g = \sum_{\alpha=1}^{N_S} p_{i\alpha}^g \phi_{-,\alpha}^g + \sum_{j=1}^{N_V} p_{ij}^g q_j^g$$

$$\phi_{+,\alpha}^g = \sum_{\beta=1}^{N_S} p_{\alpha\beta}^g \phi_{-,\beta}^g + \sum_{j=1}^{N_V} p_{\alpha j}^g q_j^g$$

where

$$\phi_i^g = \frac{1}{V_i} \int_{V_i} d^3r \, \phi^g(\vec{r})$$

$$\phi_{+,\alpha}^g = \frac{1}{S_\alpha} \int_{S_\alpha} d^2r \, \phi^g(\vec{r}_S)$$

The Transport Equation

Four types of probabilities

$$\tilde{p}^g_{ij} = V_i p^g_{ij} = \int_{V_i}\int_{V_j} \frac{e^{-\tau^g(R)}}{4\pi R^2}\Theta_i\Theta_j d^3r' d^3r$$

$$\tilde{p}^g_{i\alpha} = V_i p^g_{i\alpha} = \int_{V_i}\int_{S_\alpha} \frac{e^{-\tau^g(R_S)}}{4\pi R_S^2}(\vec{\Omega}\cdot\vec{N}_-)\Theta_i\Theta_\alpha d^3r' d^2r$$

$$\tilde{p}^g_{\alpha i} = \frac{S_\alpha}{4} p^g_{\alpha i} = \int_{S_\alpha}\int_{V_i} \frac{e^{-\tau^g(R)}}{4\pi R^2}(\vec{\Omega}\cdot\vec{N}_+)\Theta_\alpha\Theta_i d^2r' d^3r$$

$$\tilde{p}^g_{\alpha\beta} = \frac{S_\alpha}{4} p^g_{\alpha\beta} = \int_{S_\alpha}\int_{S_\beta} \frac{e^{-\tau^g(R_S)}}{4\pi R_S^2}(\vec{\Omega}\cdot\vec{N}_-)(\vec{\Omega}\cdot\vec{N}_+) \times\Theta_\alpha\Theta_\beta d^2r d^2r'$$

The Transport Equation

Symmetry relations

$$
\begin{aligned}
V_i p_{ij}^g &= V_j p_{ji}^g \\
4V_i p_{i\alpha}^g &= S_\alpha p_{\alpha i}^g \\
S_\alpha p_{\alpha\beta}^g &= S_\beta p_{\beta\alpha}^g
\end{aligned}
$$

Conservation properties

$$
\begin{aligned}
\sum_{\alpha=1}^{N_\alpha} p_{i\alpha}^g + \sum_{j=1}^{N_j} p_{ij}^g \Sigma_j^g &= 1 \\
\sum_{\beta=1}^{N_\beta} p_{\alpha\beta}^g + \sum_{i=1}^{N_i} p_{\alpha i}^g \Sigma_i &= 1
\end{aligned}
$$

Boundary Conditions 1

Surface flux approximation:

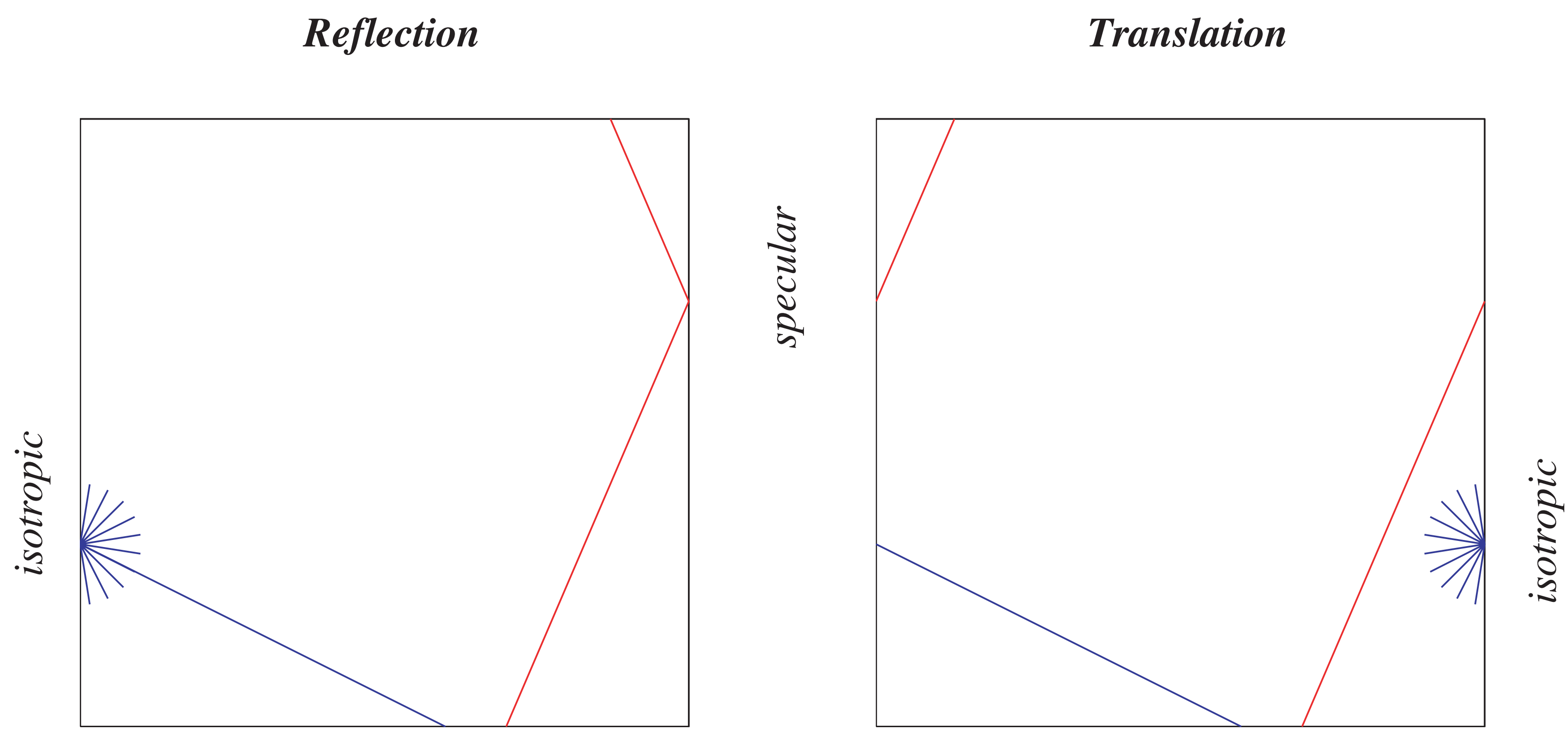
- Incoming angular flux on outer surfaces assumed to be independent of $\vec{\Omega}$

Comments

- Outgoing angular flux on outer surfaces integrated over $\vec{\Omega}$
- Angular flux not used at region interfaces
- Approximation for incoming angular flux exact for vacuum BC
- Approximation for incoming angular flux leads to large errors in surface flux when flux is not isotropic

Boundary Conditions 2

Illustration of approximate BC



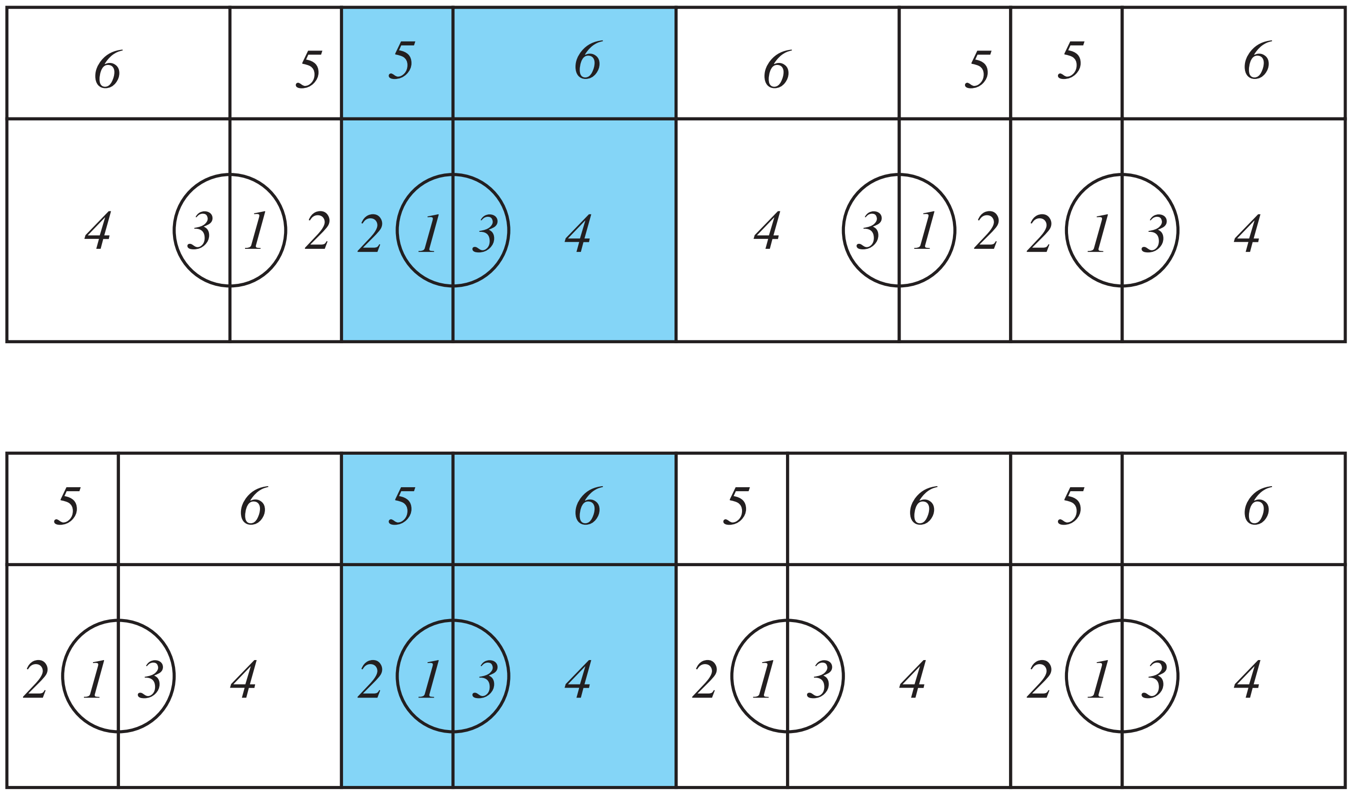
Boundary Conditions 3

Recommendations to reduce errors due to approximate use of BC for a fixed direction

- No special treatment for 2 vacuum BC
- Unfold cell once for 1 vacuum and 1 reflection BC
- Multiply unfold cell for 2 reflection or periodic cell, apply approximate BC on final surfaces and consider results in cell located far from these surfaces.

Boundary Conditions

Example of cell unfolding in direction X



Boundary Conditions

Simplifying CP equations using approximate BC

- Assume

$$\vec{J}_-^g = \mathbf{A}^g \vec{J}_+^g$$

- Final transport equation

$$\vec{\phi}^g = \mathbf{P}_{c,vv}^g \vec{q}^g$$

with the complete collision probability matrix $\mathbf{P}_{c,vv}^g$:

$$\mathbf{P}_{c,vv}^g = \left(\mathbf{P}_{vv}^g + \mathbf{P}_{vs}^g ((\mathbf{A}^g)^{-1} - \mathbf{P}_{ss}^g)^{-1} \mathbf{P}_{sv}^g \right)$$

Cross Sections Consideration 1

Two types of multigroup cross-section database can be read by DRAGON

- Mixture macroscopic cross-section
- Isotope microscopic cross-section that contains itself a macroscopic cross-section database

Cross Sections Consideration 2

Minimum cross-section requirements for each mixture m

- The multigroup total cross section Σ_m^g
- The isotropic component of the multigroup scattering cross section $\Sigma_{m,s,0}^{h\to g}$ defined as

$$\Sigma_{m,s,0}^{h\to g} = \int_{4\pi} d^2\Omega^2 \Sigma_{m,s}^{h\to g}(\vec{\Omega}\,' \to \vec{\Omega}) P_0(\vec{\Omega}\,' \cdot \vec{\Omega})$$

- The product of the average neutron emitted per fission with the multigroup fission cross section $\nu\Sigma_{m,f}^g$
- The multigroup fission spectrum χ_m^g

Cross Sections Consideration 3

The linearly isotropic component of the multigroup scattering cross section $\Sigma_{m,s,1}^{h\to g}$

$$\Sigma_{m,s,1}^{h\to g} = \int_{4\pi} d^2\Omega^2 \Sigma_{m,s}^{h\to g}(\vec{\Omega}\,' \to \vec{\Omega}) P_1(\vec{\Omega}\,' \cdot \vec{\Omega})$$

- Required only if B_1 leakage method is used

Cross Sections Consideration 4

The transport correction $\Sigma^g_{m,\text{tc}}$

- The transport calculations are performed using transport corrected total ($\tilde{\Sigma}^g_m$) and scattering ($\tilde{\Sigma}^{h\to g}_{m,s,0}$) cross sections

$$\tilde{\Sigma}^g_m = \Sigma^g_m - \Sigma^g_{m,\text{tc}}$$

$$\tilde{\Sigma}^{h\to g}_{m,s,0} = \Sigma^{h\to g}_{m,s,0} - \delta^{gh}\Sigma^g_{m,\text{tc}}$$

- Takes partially into account the linearly anisotropic scattering contributions

Cross Sections Consideration 5

For contribution of multi-neutron production reactions such as

$$\mathsf{X}_A^Z + \mathsf{n}_1^0 \rightarrow \mathsf{X}_{A-1}^Z + 2\mathsf{n}_1^0 + \gamma$$

The scattering cross section must be corrected to take into account this effect

$$\tilde{\Sigma}_{m,s,0}^{h\rightarrow g} = \Sigma_{m,s,0}^{h\rightarrow g} + 2\delta^{gh}\Sigma_{m,(\mathsf{n},2\mathsf{n})}^{g}$$

where $\Sigma_{(\mathsf{n},2\mathsf{n})}^{g}$ is the macroscopic cross section associated with the reaction

Cross Sections Consideration 6

Macroscopic cross section data base can be created using

- from the input file using the `MAC:` module
- from a GOXS file using the `MAC:` module
- from a microscopic library using the `LIB:` module
- from the homogenization and condensation module `EDI:`
- from a WIMS-AECL execution using the information available on `TAPE16` (side-step method)
- from a HELIOS execution

Microscopic libraries 1

Many formats can be processed by DRAGON including

- WIMS–AECL format
- MATXS format
- WIMD-D4 format

In DRAGON resonance self-shielding calculations are preformed using the Stamm'ler method

3–D CP Calculations

Contents

- Collision Probabilities in 3-D
- Numerical Quadrature and Tracking
- Collision Probability Integration
- Neutron Conservation and CP Normalization

3–D CP Calculations 2

Recall CP approximations

- Divide domain into N_V regions of volume V_i
- Assume total cross sections constant inside each region
- Assume sources constant inside each region
  - This has an impact on the selection of the spatial mesh
- Assume sources isotropic inside each region
  - This may lead to problem when scattering is highly anisotropic.

3–D CP Calculations

Recall BC approximations

- Divide the external boundary S into N_S surfaces of area S_α
- Assume flux constant on external surfaces
  - This has an impact on the selection of the spatial mesh
- Assume flux isotropic on external surfaces
  - This may lead to problems when flux is highly anisotropic near external boundaries

Collision Probabilities in 3-D 1

Collision probability definition

$$\tilde{p}^g_{ij} = V_i p^g_{ij} = \int_{V_i}\int_{V_j} \frac{e^{-\tau^g(R)}}{4\pi R^2}\Theta_i\Theta_j d^3r' d^3r$$

- Spherical coordinates for d^3r integral

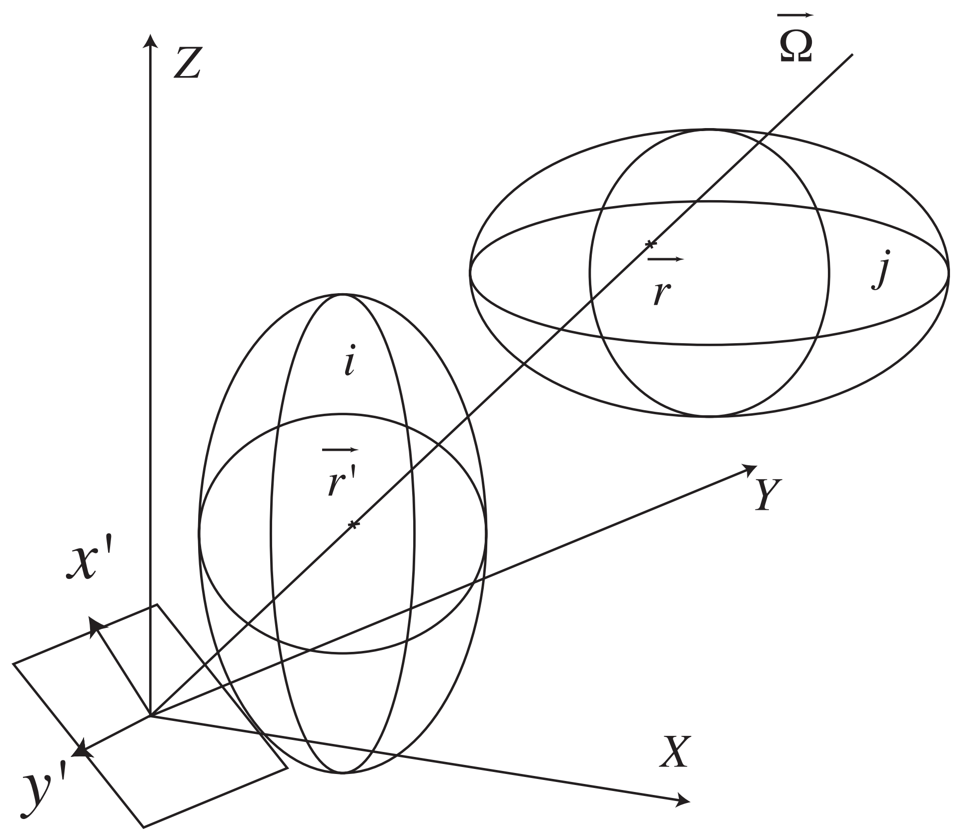
$$\int_{V_j} d^3r\Theta_j = \int_{4\pi} d^2\Omega \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} R^2 dR\Theta_j$$

- Cartesian coordinates for d^3r' integral

$$\int_{V_i} d^3r'\Theta_i = \int dx' \int dy' \int dR'\Theta_i$$

Collision Probabilities in 3-D 2

General 3–D geometry for collision probability integration



Collision Probabilities in 3-D 3

Final form for collision probability integration

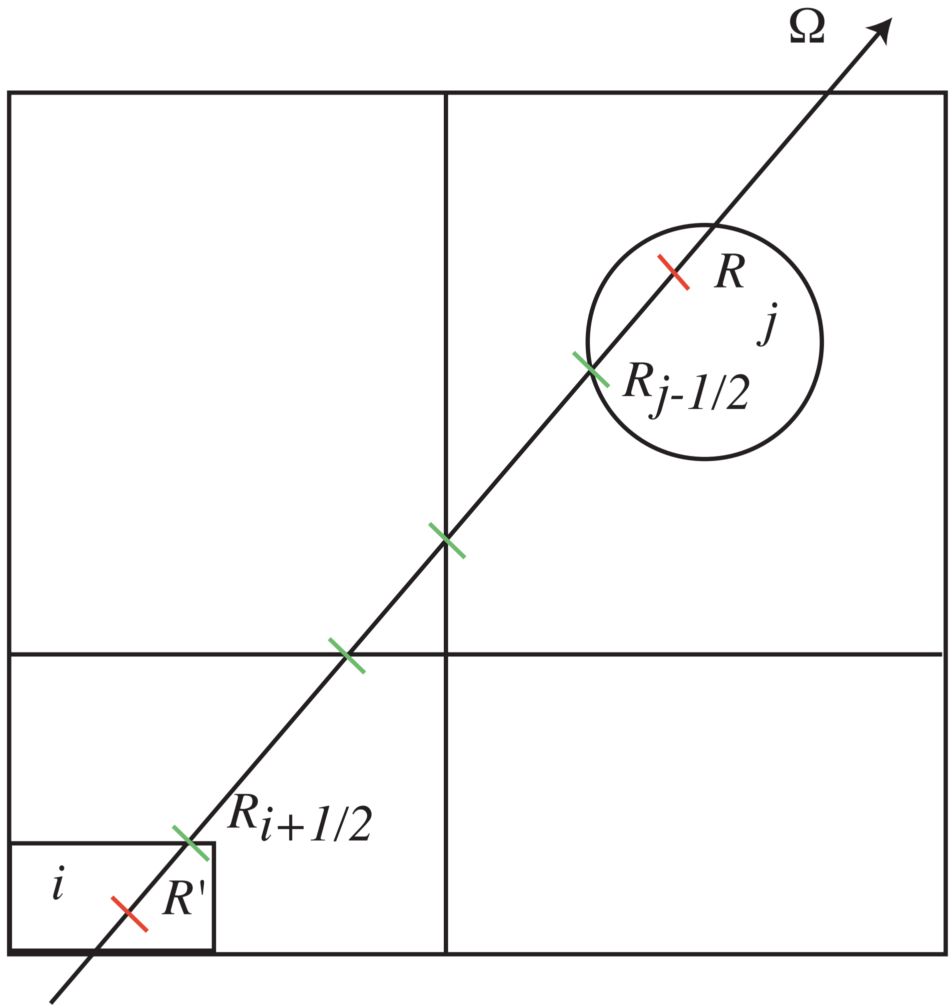
$$\tilde{p}_{ij}^g = \int_{4\pi} \frac{d^2\Omega}{4\pi} \int dx' \int dy' \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} dR' \int_{R_{j-\frac{1}{2}}}^{R_{j+\frac{1}{2}}} dR \, e^{-\tau^g(R',R)} \Theta_i \Theta_j$$

$$\tau^g = (R_{i+\frac{1}{2}} - R')\Sigma_i^g + \sum_{k=i+1}^{j-1} \Delta R_k \Sigma_k^g + (R - R_{j-\frac{1}{2}})\Sigma_j^g$$

with $\Delta R_k = R_{k+\frac{1}{2}} - R_{k-\frac{1}{2}}$.

Collision Probabilities in 3-D

Notation for optical path



Collision Probabilities in 3-D 5

After integration over R' and R, one obtains

$$\tilde{p}_{ij}^g = \frac{1}{4\pi\Sigma_i^g\Sigma_j^g}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_j$$
$$\times \left[1-\exp\left(-\tau^g_{i-\frac{1}{2},i+\frac{1}{2}}\right)\right]\exp\left(-\tau^g_{i+\frac{1}{2},j-\frac{1}{2}}\right)$$
$$\times \left[1-\exp\left(-\tau^g_{j-\frac{1}{2},j+\frac{1}{2}}\right)\right]$$

with

$$\tau^g_{i\pm\frac{1}{2},j\pm\frac{1}{2}} = \Sigma_i^g(R_{i+\frac{1}{2}}-R_{i\pm\frac{1}{2}}) + \tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j^g(R_{j\pm\frac{1}{2}}-R_{j-\frac{1}{2}})$$

Collision Probabilities in 3-D 6

Case where $\Sigma_i^g = 0$

$$\tilde{p}_{ij}^g = \frac{1}{4\pi\Sigma_j^g} \int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_j$$
$$\times \Delta R_i \exp(-\tau^g_{i+\frac{1}{2},j-\frac{1}{2}}) \left[1 - \exp(-\tau^g_{j-\frac{1}{2},j+\frac{1}{2}})\right]$$

Case where $\Sigma_i^g = \Sigma_j^g = 0$

$$\tilde{p}_{ij}^g = \frac{1}{4\pi} \int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_j \Delta R_i \Delta R_j \exp(-\tau^g_{i+\frac{1}{2},j-\frac{1}{2}})$$

Collision Probabilities in 3-D 7

For $\tilde{p}^g_{ii}$, one obtains after integration over R' and R

$$\tilde{p}^g_{ii} = \frac{1}{2\pi\left(\Sigma^g_i\right)^2}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_i \times \left[\tau^g_{i-\frac{1}{2},i+\frac{1}{2}} - \left(1 - \exp(-\tau^g_{i-\frac{1}{2},i+\frac{1}{2}})\right)\right]$$

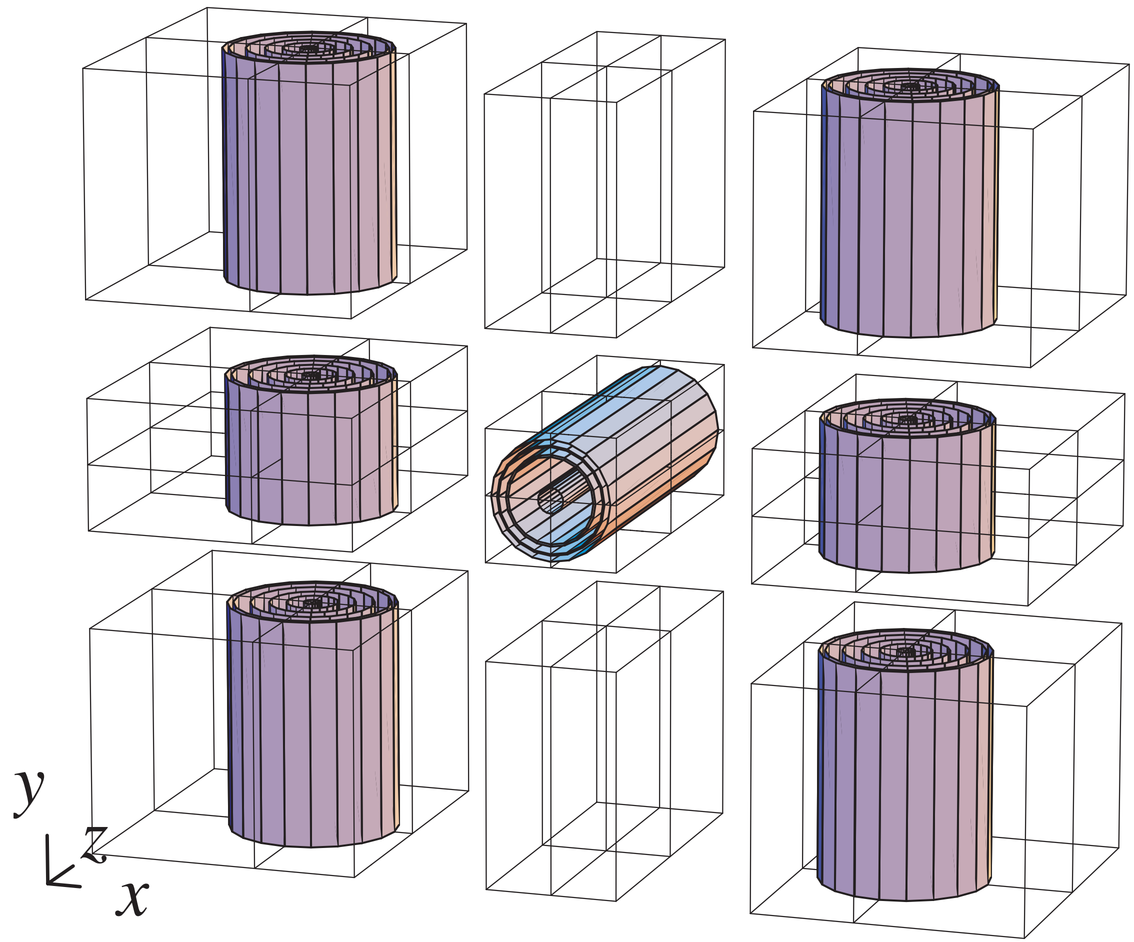
For $\Sigma^g_i = 0$ this is simplified to

$$\tilde{p}^g_{ii} = \frac{1}{4\pi}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_i \left(\Delta R_i\right)^2$$

Similar relations are obtained for $\tilde{p}^g_{i\alpha}$ and $\tilde{p}^g_{\alpha\beta}$

Quadrature and Tracking 1

3-D Adjuster model in DRAGON



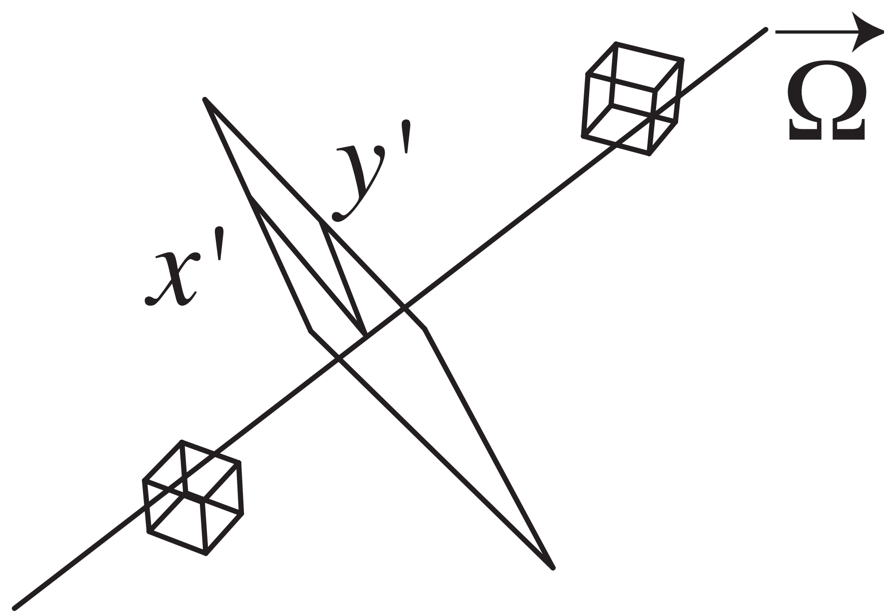
Quadrature and Tracking 2

Angular quadrature

- S_n type EQ$_N$ quadrature with $4N_\Omega(N_\Omega+2)/8$ angular directions $\vec{\Omega}_{1,i}$, $\vec{\Omega}_{2,i}$, $\vec{\Omega}_{3,i}$ and $\vec{\Omega}_{4,i}$
- Global quadrature weight $W_\Omega = 2/(N_\Omega(N_\Omega+2))$
- Tracking in lower half sphere only
- Number of tracking quadrant automatically reduced for symmetric cell
- Select as many angles as possible (Neutrons travel on a straight line)

Quadrature and Tracking 3

Cartesian surface quadrature



Quadrature and Tracking 4

Cartesian quadrature

- Identify the radius h_+ of the smallest sphere including the geometry
- Select a tracking density ρ_p and define the line spacing δ

$$\delta = \frac{2h_+}{N_p} \qquad N_p = (2\sqrt{\rho_p}h_+) + 1$$

- Integration line l_{mn} passes through

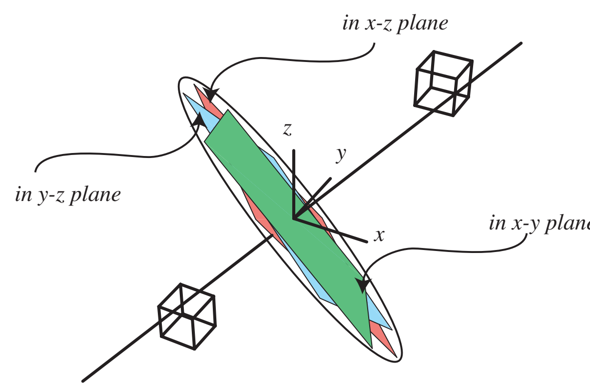
$$u_m^x = \left(\frac{2m-1}{2}\right)\delta \qquad u_n^y = \left(\frac{2n-1}{2}\right)\delta$$

- Integration weight $W_p = \delta^2$

Quadrature and Tracking 5

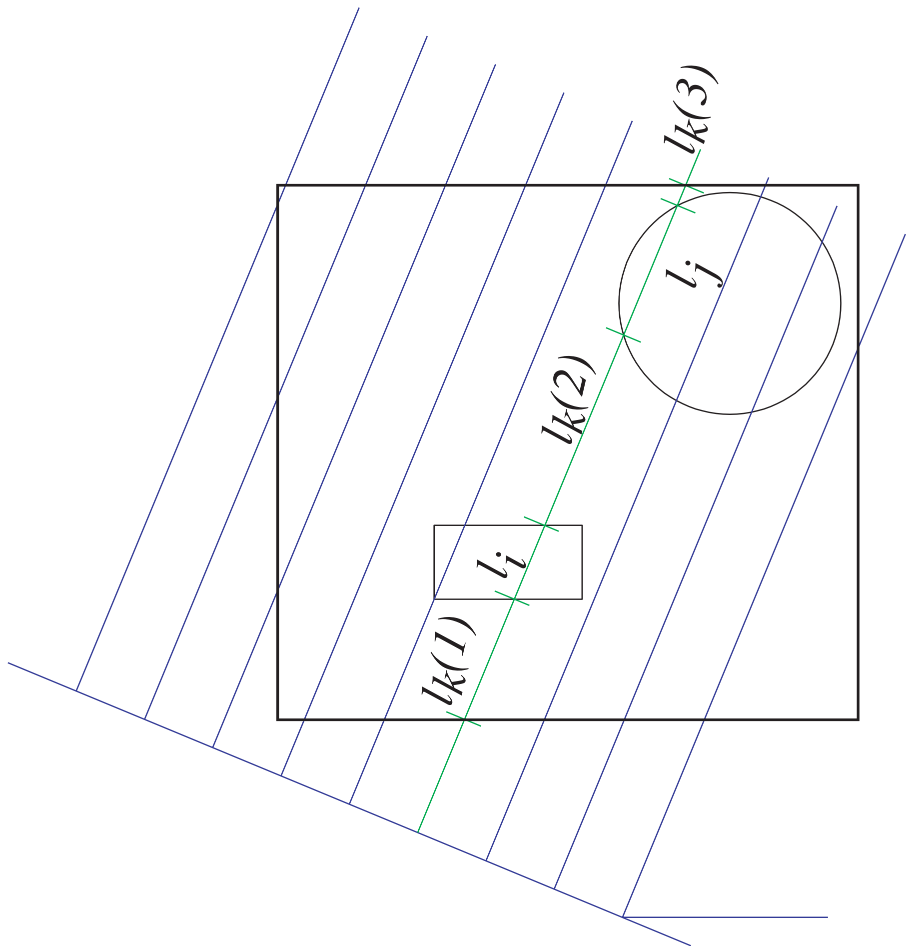
Comments on Cartesian surface quadrature

- In Dragon 3 different planes are selected for each given spatial direction



- Select tracking density as dense as possible
  - Each region must be touched by a maximum number of lines

DRAGON tracking example



Quadrature and Tracking 7

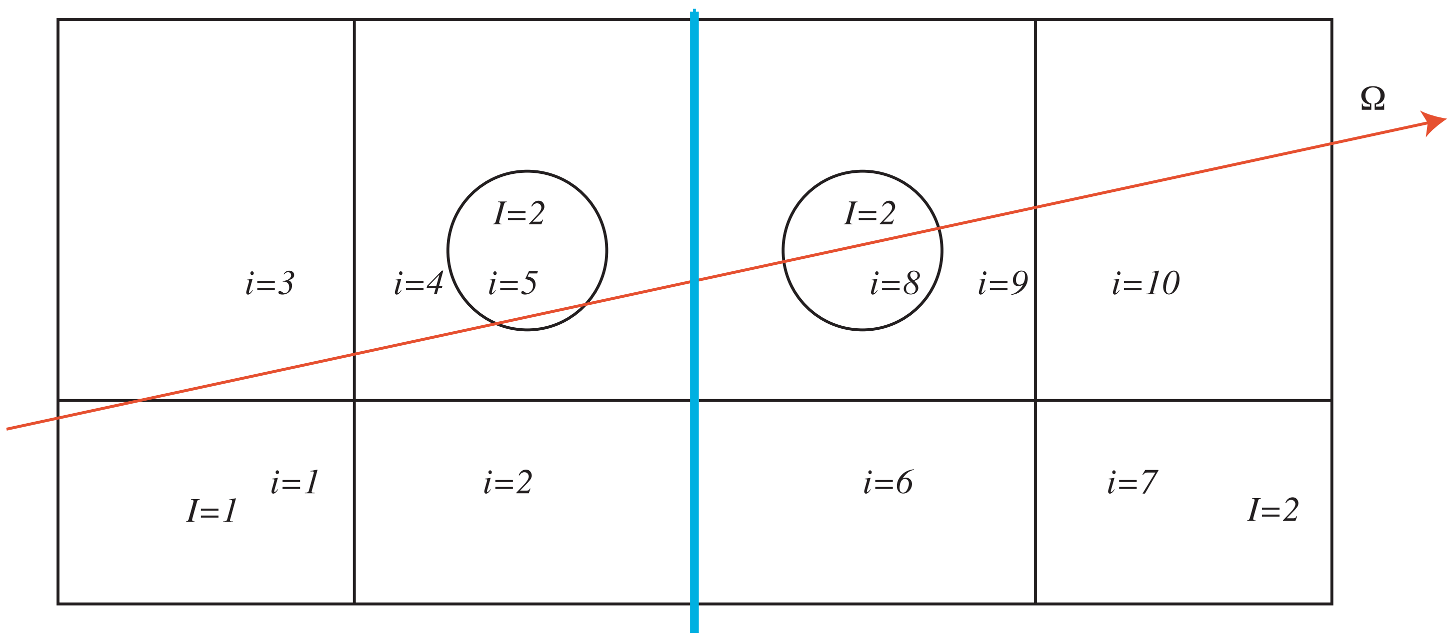
DRAGON tracking of a line $l_{m,n}$

1. Follow the tracking line as it travels through the cell
2. Identify each region i and surface s uniquely
3. Identify final region number I associated with set of regions i (a flux region)
4. Identify final surface number S associated with sets of surfaces s
5. Identify external surfaces and regions i crossed by line
6. Compute distance $\tilde{l}_{i,m,n}$ the neutron travels in each region
7. Store information on temporary tracking file

Quadrature and Tracking

Case of symmetric cells (mirror reflection on one side of the cell)

- Unfold the cell according to symmetry



- Regions in unfolded cells are originally assigned new region numbers

Quadrature and Tracking 9

Angle selection for symmetric cells

- For symmetry with respect to a (y, z) plane, track only in quadrants corresponding to directions $\vec{\Omega}_{1,i}$ and $\vec{\Omega}_{3,i}$
- For symmetry with respect to a (z, x) plane, track only in quadrants corresponding to directions $\vec{\Omega}_{1,i}$ and $\vec{\Omega}_{2,i}$
- For symmetry with respect to a (x, y) plane, track only in all quadrants

Quadrature and Tracking 10

Post treatment of tracking file

- Assign to all spatial region i its final flux region number combining track segments as required
- For each direction, normalize tracks using

$$l_{i,m,n} = \left(\frac{V_i}{\tilde{V}_i}\right) \tilde{l}_{i,m,n}$$

where

$$\tilde{V}_i = \frac{W_p}{3} \sum_{m=1}^{N_p} \sum_{n=1}^{N_p} \tilde{l}_{i,m,n}$$

Quadrature and Tracking 11

Comments on storage requirements for tracking

- Maximum number of line segments tracks d_t

$$d_t < 6N(N_\Omega(N_\Omega + 2))\rho_p h_+^2$$

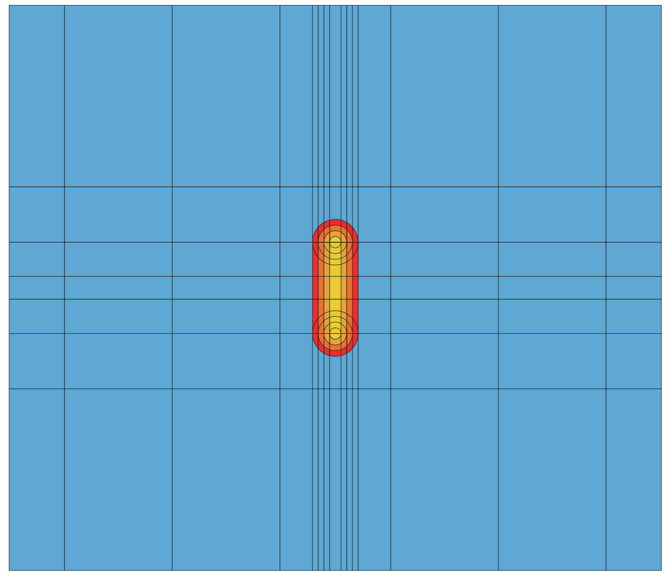
- For$N = 1000$, $h_+ = 50$, $\rho_p = 20$ t/cm^2 and $N_\Omega = 8$

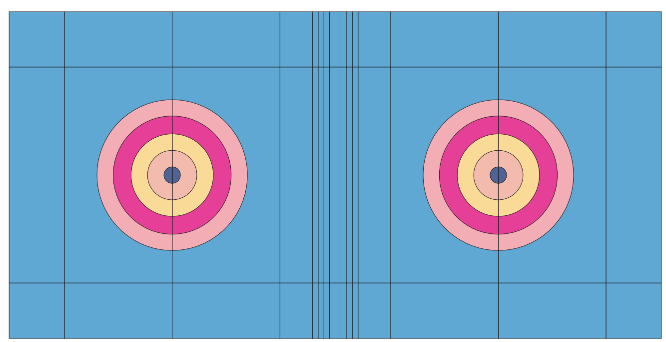
$$d_t < 20 \times 10^9$$

tracks segments.

Quadrature and Tracking

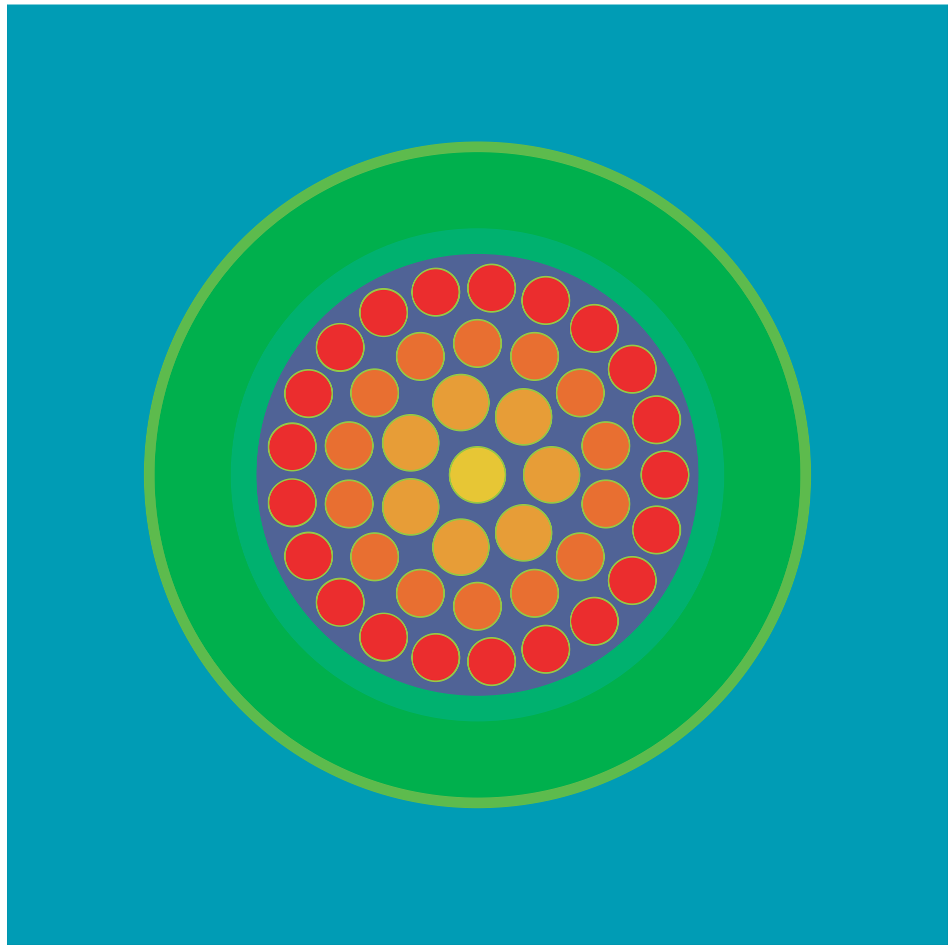
ACR control rod model in DRAGON





Quadrature and Tracking

3-D cluster analysis not currently allowed in DRAGON



CP Integration

For each energy group

- Read a line from tracking file
- Scan this line and add contribution to $\tilde{p}^g_{ii}$

$$\frac{1}{2}(\Sigma^g_i)^2\tilde{p}^g_{ii} = \frac{1}{2}(\Sigma^g_i)^2\tilde{p}^g_{ii} + \sum_n W_n \sum_{m\in i}\left(\tau^g_{i,n,m} - \kappa^g_{i,n,m}\right)$$

where $W_n = W_\Omega W_p/3$ and

$$\kappa^g_{i,n,m} = \left(1 - \exp\left[-\tau^g_{i,n,m}\right]\right)$$

$$\tau^g_{i,n,m} = \Sigma^g_i l_{i,n,m} \qquad \text{with } m \in i$$

CP Integration 2

- Scan this line a second time and add contributions to $\tilde{p}^g_{ij}$

$$\begin{aligned}\Sigma^g_i \Sigma^g_j \tilde{p}^g_{ij} =& \Sigma^g_i \Sigma^g_j \tilde{p}^g_{ij} \\ &+ \sum_n W_n \sum_{m\in i} \sum_{m'\in j} \kappa^g_{i,n,m} \kappa^g_{n,m+1,m'-1} \kappa^g_{j,n,m'}\end{aligned}$$

using

$$\kappa^g_{n,m,m'} = \prod_{l=m}^{m'} \exp\left[-\tau^g_{i,n,l}\right]$$

CP Integration

Finish CP calculations

- Only the contributions with $m < m'$ has been considered
- Symmetrize p_{ij}^g using

$$\tilde{p}_{ij}^g = \tilde{p}_{ij}^g + \tilde{p}_{ji}^g$$

CP Normalization 1

Compute errors on CP conservation rules

$$R_j^g = \Sigma_j^g V_j - \sum_{\alpha=1}^{N_\alpha} \frac{S_\alpha}{4} \Sigma_i^g p_{\alpha j}^g - \sum_{i=1}^{N_i} \Sigma_j^g \Sigma_i^g V_i p_{ij}^g$$

$$R_\beta^g = \frac{S_\beta}{4} - \sum_{\alpha=1}^{N_\alpha} \frac{S_\alpha}{4} p_{\alpha\beta}^g - \sum_{i=1}^{N_i} \Sigma_i^g V_i p_{i\beta}^g$$

CP Normalization 2

Diagonal Normalization

$$p^g_{D,ii} = p^g_{ii} - \frac{R^g_i}{(\Sigma^g_i)^2 V_i}$$

$$p^g_{D,\alpha\alpha} = p^g_{\alpha\alpha} - \frac{4R^g_\alpha}{S_\alpha}$$

- May result in non-physical negative probabilities
- Cannot be applied to problems involving voided zones

CP Normalization 3

HELIOS Type Normalization

$$p^g_{H,ij} = (w^g_i + w^g_j) p^g_{ij} \qquad p^g_{H,\alpha\alpha} = (w^g_\alpha + w^g_\beta) p^g_{\alpha\alpha}$$

- Apply conservation laws to above relation
- Solve resulting system for w^g using an iterative process
- Does not lead to negative probabilities and works for void regions
- Default option in DRAGON

CP Normalization

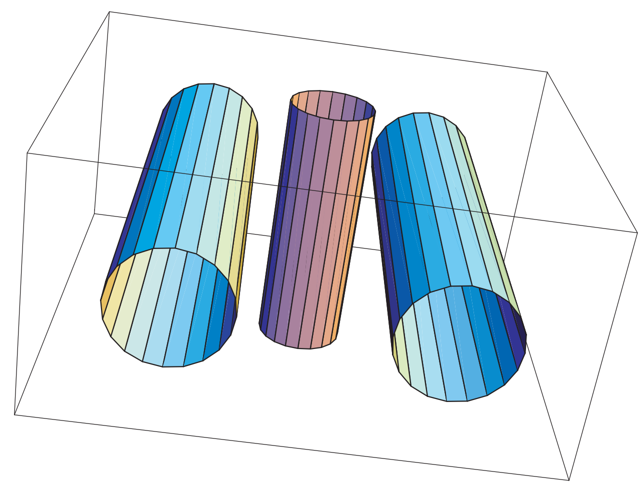
Comments on storage requirements for CP matrices

- Number of elements per groups is N^2
- Memory space required for execution is about $5N^2$
- Total disk space required for storage of G group CP is GN^2

3–D DRAGON Examples

Contents:

- Geometry.
- Collision Probability Integration and Tracking.
- Region Merging.



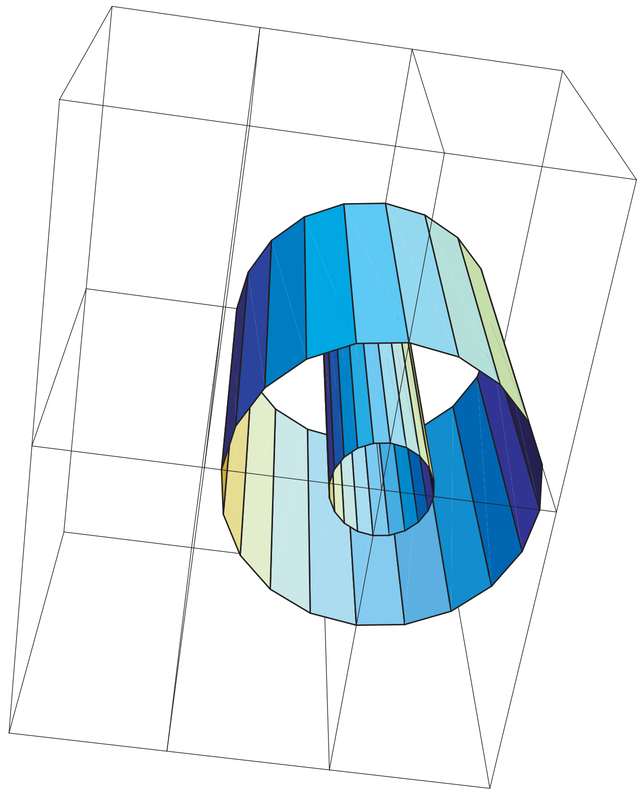
3–D DRAGON Examples 2

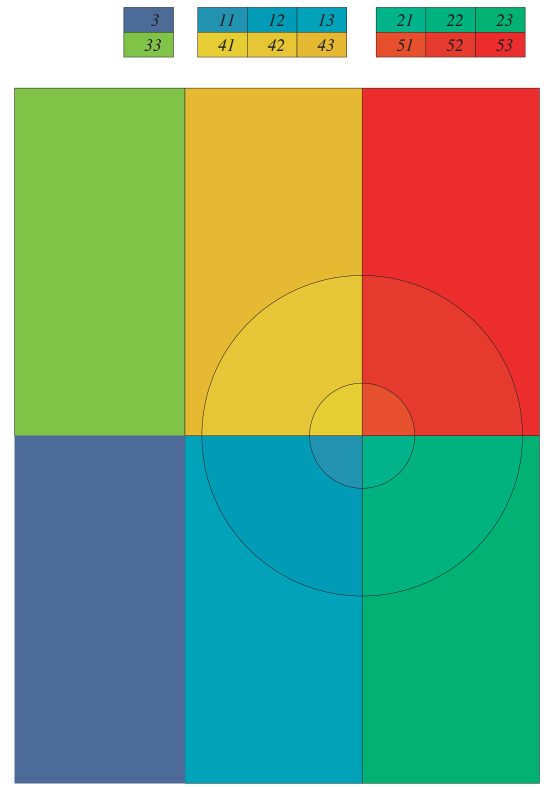
3–D Geometry restrictions in DRAGON:

- Cartesian mesh in each direction must extend to the whole geometry.
- Cluster option not permitted.
- A single cylinder per cell.
- Cylinders cannot intersect other than axially.
- Cylinders must extend to the whole geometry.
- Cylinders are by default centered in the cell.
  They can be displaced using the `OFFCENTER` option.
- Mixtures are specified radially, then in x, y and z.
- Mixtures are specified even in location that do not exists.

3–D DRAGON Examples

A Simple 3–D Cell:





3–D DRAGON Examples 4

Mixture specification for simple 3–D cell:

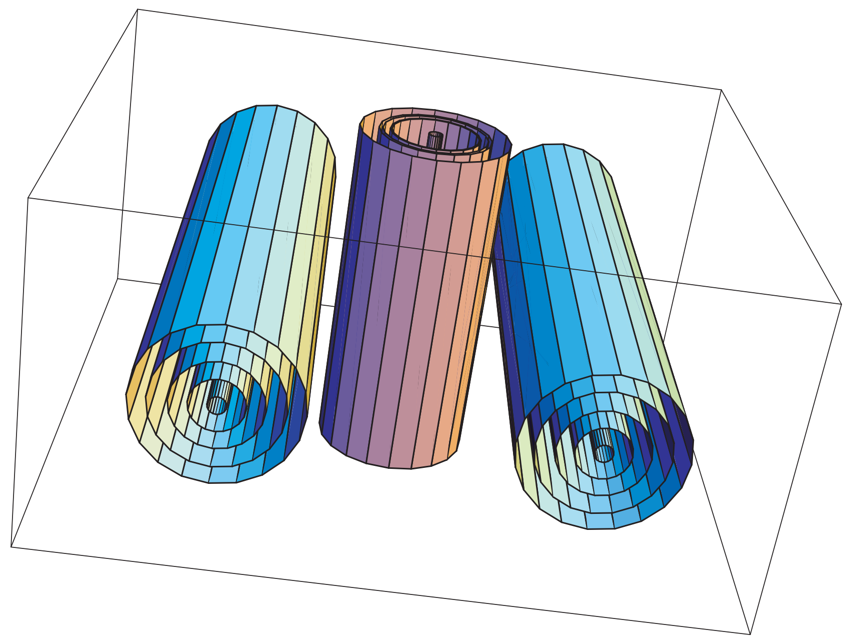
```
TMPGEO    := GEO:   :: CAR3D 1 1 1
  CELL     FC1B
  X- REFL X+ REFL Y- REFL Y+ REFL Z- REFL Z+ REFL
  ::: FC1B := GEO: CARCELZ 2 3 2 1
      MESHX       <<MXLP>>   <<MXL>>  <<C14CN>>  <<MXYD>>
      MESHY       <<MYLP>>  <<CALLCN>> <<PYLP>>
      MESHZ       0.0       49.5
      OFFCENTER  <<FC1XD>> <<FC1YD>>
      RADIUS  0.0  <<RF2>>   <<RCT>>
      MIX          1  2  3
                 11 12 13
                 21 22 23
                 31 32 33
                 41 42 43
                 51 52 53  ;
  ;
```

Note: mixtures 1, 2, 31 and 32 not used.

3–D DRAGON Examples 5

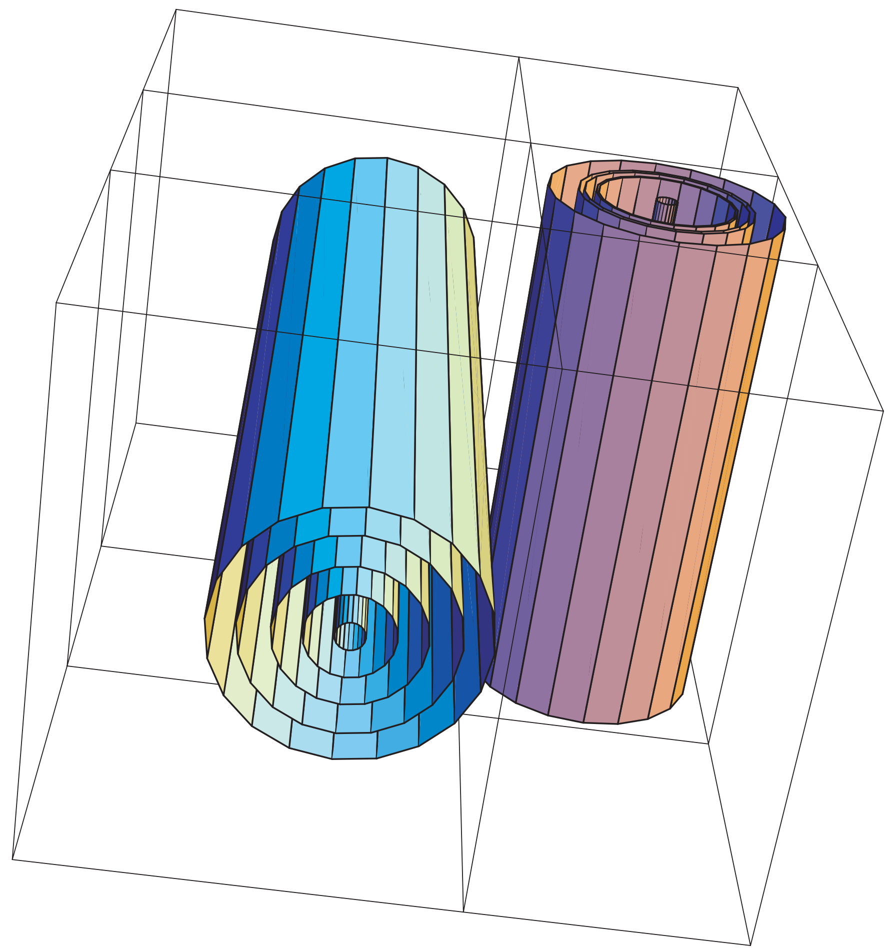
CANDU adjuster rod simulation:

- 5 region annular fuel (including PT and CT).
- 6 region annular adjuster rod.



3–D DRAGON Examples 6

Coarse mesh geometry definition for CANDU adjuster:



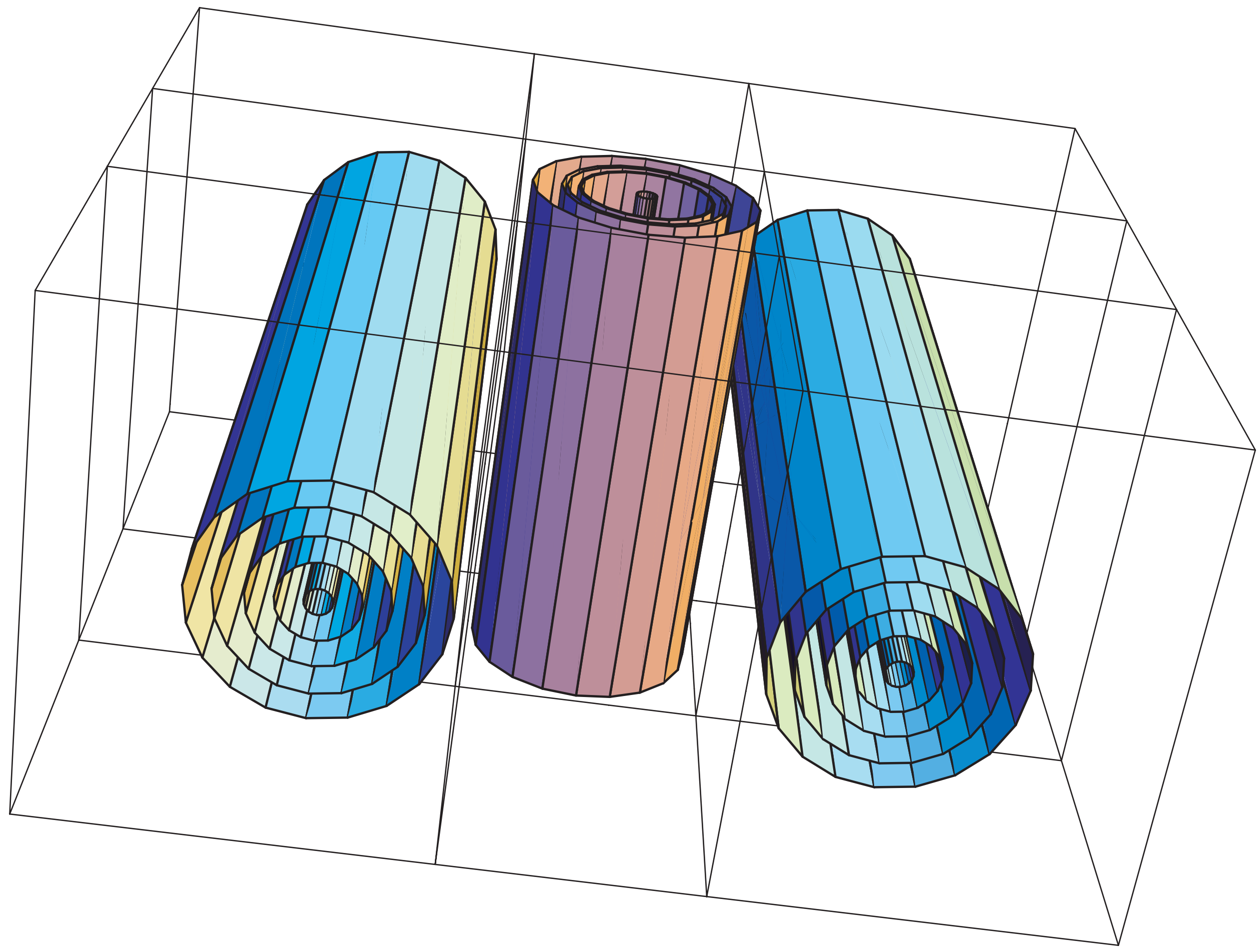
3–D DRAGON Examples

DRAGON geometry for CANDU adjuster rod:

```
TMPGEO   := GEO:  :: CAR3D 2 1 1
  CELL    FC1B  AD1T
  X- REFL X+ SYME Y- REFL Y+ SYME Z- REFL Z+ SYME
  ::: FC1B := GEO: CARCELZ 5 1 1 3
      MESHX  0.0000 21.5750 MESHY  0.0000 28.5750
      MESHZ  0.0000 17.7650 31.7650 49.5300
      OFFCENTER  3.5 0.0
      RADIUS 0.0 0.7222  2.1603  3.6007  5.1689  6.5875
      MIX  1 2 3 4 5 6  1 2 3 4 5 6  1 2 3 4 5 6  1 2 3 4 5 6 ;
  ::: AD1T := GEO: CARCELY 6 1 1 3
      MESHX 21.5750 35.5750 MESHY  0.0000 28.5750
      MESHZ  0.0000 17.7650 31.7650 49.5300
      RADIUS 0.0 0.5770  3.6781  3.8100  4.4450  4.7520  6.3776
      MIX 7 8 9 10 11 12 1  7 8 9 10 11 12 1
          7 8 9 10 11 12 1  7 8 9 10 11 12 1 ;
  ;
```

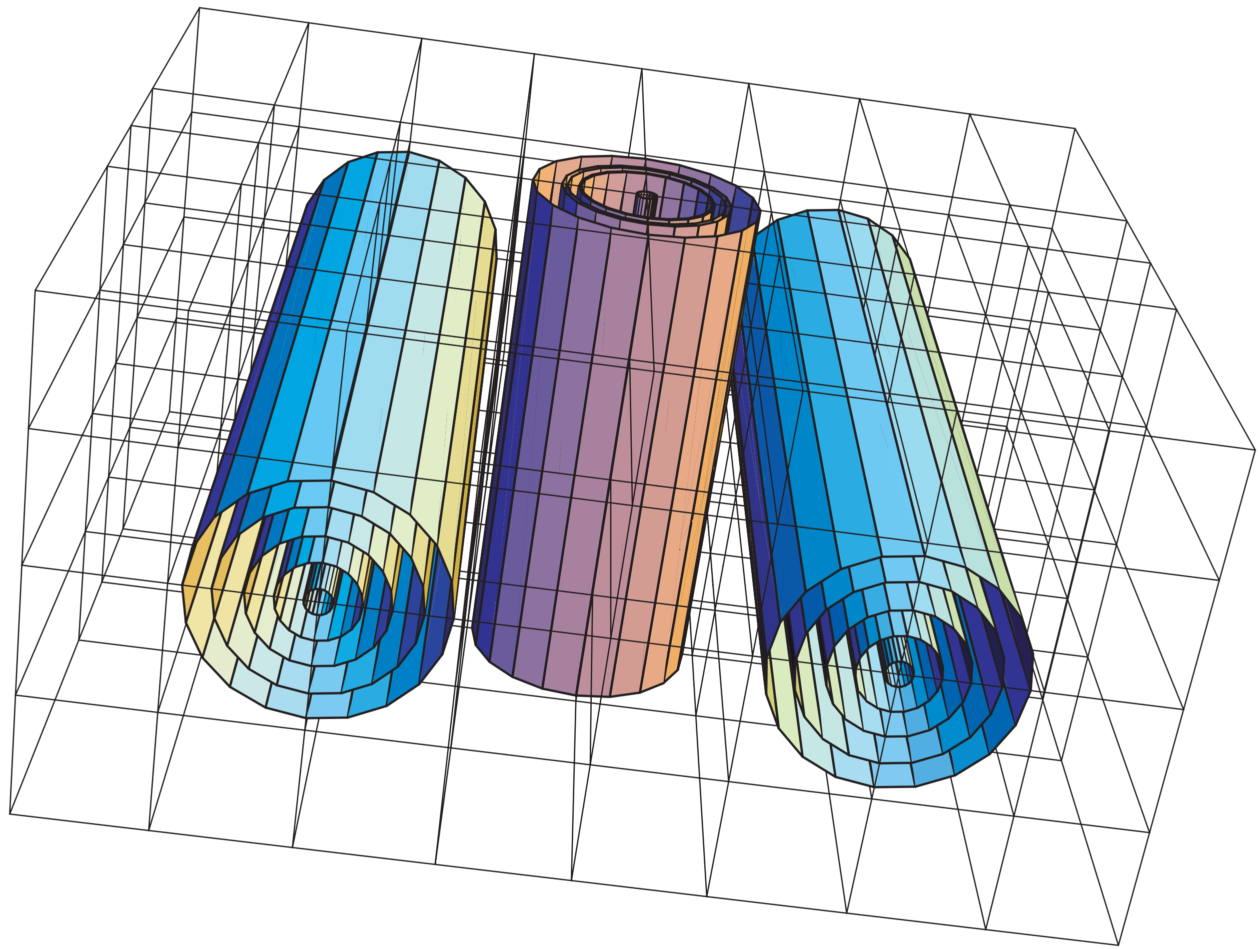
3–D DRAGON Examples 8

Coarse mesh CANDU adjuster rod after unfolding:



3–D DRAGON Examples 9

Fine mesh CANDU adjuster rod after unfolding:



3–D DRAGON Examples 10

DRAGON geometry for CANDU adjuster rod:

```
TMPGEO    := GEO:   :: CAR3D 2 1 1
  CELL     FC1B  AD1T
  X- REFL X+ SYME Y- REFL Y+ SYME Z- REFL Z+ SYME
  ::: FC1B := GEO: CARCELZ 5 1 1 3
      MESHX  0.0000 21.5750 SPLITX 3 MESHY  0.0000 28.5750 SPLITY 3
      MESHZ  0.0000 17.7650 31.7650 49.5300
      OFFCENTER  3.5 0.0
      RADIUS 0.0 0.7222  2.1603  3.6007  5.1689  6.5875
      MIX  1 2 3 4 5 6  1 2 3 4 5 6  1 2 3 4 5 6  1 2 3 4 5 6 ;
  ::: AD1T := GEO: CARCELY 6 1 1 3
      MESHX 21.5750 35.5750 SPLITX 2 MESHY  0.0000 28.5750 SPLITY 3
      MESHZ  0.0000 17.7650 31.7650 49.5300 SPLITZ 2
      RADIUS 0.0 0.5770  3.6781  3.8100  4.4450  4.7520  6.3776
      MIX 7 8 9 10 11 12 1  7 8 9 10 11 12 1
          7 8 9 10 11 12 1  7 8 9 10 11 12 1 ;
  ;
```

3–D DRAGON Examples 11

Exact boundary conditions:

- `VOID`: applied at the explicit boundary of the cell or assembly.
- `SYME`: applied at the center of the cells closest to the explicit assembly boundary specified.
- `DIAG`: applied at the center of the cells closest to the explicit assembly boundary specified.
- `SSYM`: applied at the explicit boundary of the cell or assembly.

3–D DRAGON Examples

Approximate boundary conditions:

- `REFL`: applied at the explicit boundary of the cell or assembly. Exact specular option not available in 3–D.
- `TRAN`: applied at the explicit boundary of the cell or assembly. Exact specular option not available in 3–D.
- `ALBE`: applied at the explicit boundary of the cell or assembly. Exact specular option not available.

3–D DRAGON Examples

Region identification for single cell:

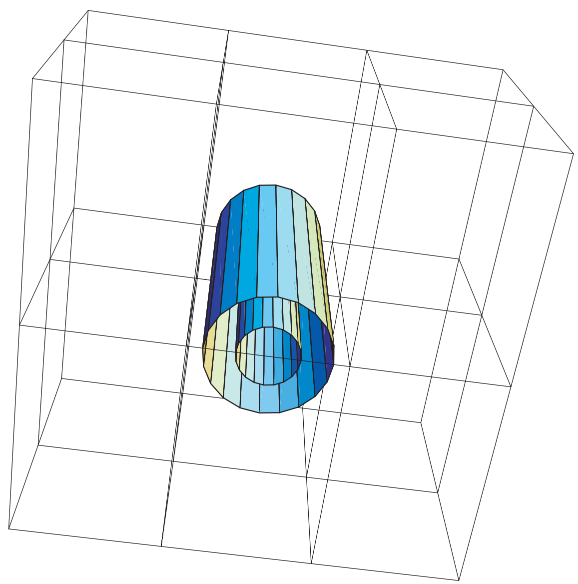
- radially outward in a cell.
- from lower to upper x location in a cell.
- from lower to upper y location in a cell.
- from lower to upper z location in a cell.

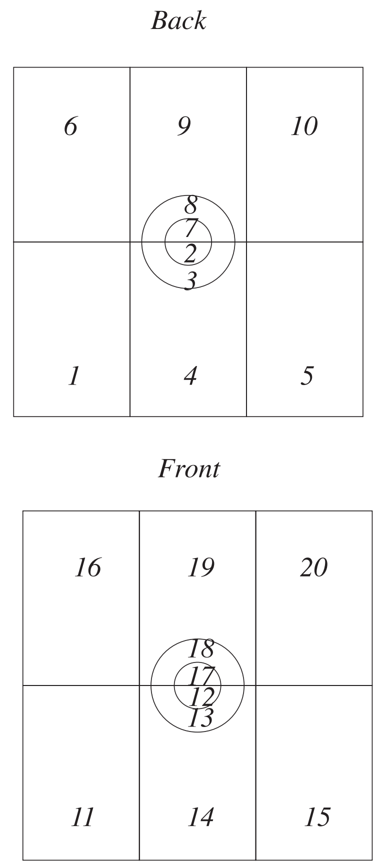
Region identification for assembly of cells:

- Inside each cell as above.
- from lower to upper x cell location in the assembly.
- from lower to upper y cell location in the assembly.
- from lower to upper z cell location in the assembly.

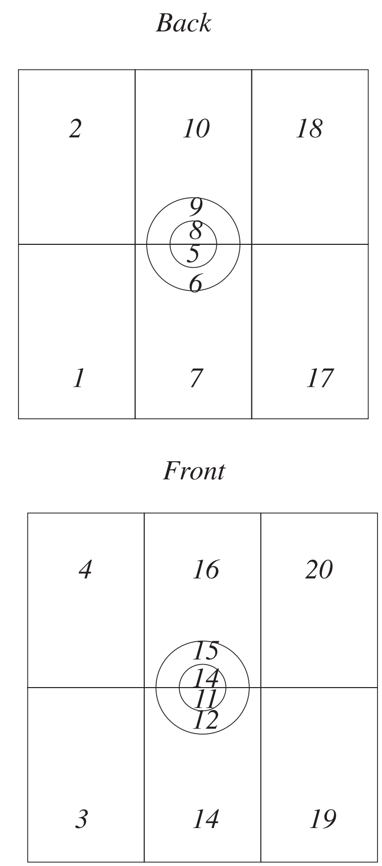
3–D DRAGON Examples

Region identification for cells and assemblies:





one cell



three cells assembly

3–D DRAGON Examples 15

Region identification for one cell problem:

```
                        ----------------------------------------
                          (   1)       16  (   1)       19  (   1)       20
                                   ABSENT  (   1)       18           ABSENT
                                   ABSENT  (   1)       17           ABSENT
PLANE - 2               ----------------------------------------
                          (   1)       11  (   1)       14  (   1)       15
                                   ABSENT  (   1)       13           ABSENT
                                   ABSENT  (   1)       12           ABSENT
                        ----------------------------------------
                          (   1)        6  (   1)        9  (   1)       10
                                   ABSENT  (   1)        8           ABSENT
                                   ABSENT  (   1)        7           ABSENT
PLANE - 1               ----------------------------------------
                          (   1)        1  (   1)        4  (   1)        5
                                   ABSENT  (   1)        3           ABSENT
                                   ABSENT  (   1)        2           ABSENT
                        ----------------------------------------
```

3–D DRAGON Examples

Region identification for three cells assembly:

```
                 ---------------------------------------
                   (   1)      4  (   1)      16  (   1)      20
                          ABSENT  (   1)      15          ABSENT
                          ABSENT  (   1)      14          ABSENT
PLANE - 2        ---------------------------------------
                   (   1)      3  (   1)      13  (   1)      19
                          ABSENT  (   1)      12          ABSENT
                          ABSENT  (   1)      11          ABSENT
                 ---------------------------------------
                   (   1)      2  (   1)      10  (   1)      18
                          ABSENT  (   1)       9          ABSENT
                          ABSENT  (   1)       8          ABSENT
PLANE - 1        ---------------------------------------
                   (   1)      1  (   1)       7  (   1)      17
                          ABSENT  (   1)       6          ABSENT
                          ABSENT  (   1)       5          ABSENT
                 ---------------------------------------
```

3–D DRAGON Examples 17

Quadrature selection:

- As many angles as possible:
  $\rightarrow$ neutron travels on a straight line.
- Tracking density must be as dense as possible:
  $\rightarrow$ to touch as often as possible each region and surface.
- For CANDU reactivity devices `TRAK TISO 8 25`:
  $\rightarrow$ 10 angles per quadrant.
  $\rightarrow$ 3×25 tracks per cm^2.
- Integration lines are renormalized using ratio of approximate to exact volumes.

3–D DRAGON Examples

Comments on storage requirements:

- Size of tracking file linear in N:

$$d_t \propto \rho h_+^2 N$$

  for $h_+ = 50$ cm, $\rho = 20$ t/cm^2 and $N = 1000$ regions:
  $\rightarrow d_t = 600$ Mb.

- Size of CP matrix quadratic in N:

$$d_a \propto N^2 G$$

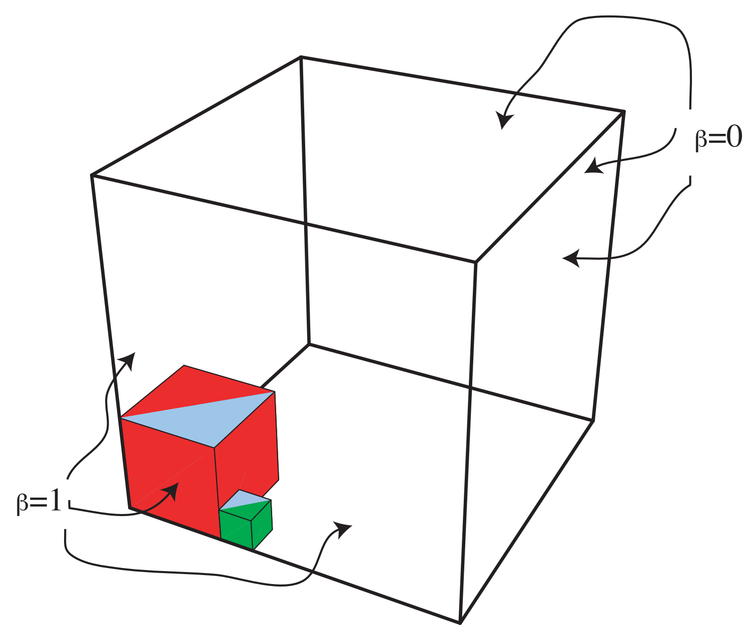
  for $G = 89$ groups and $N = 1000$ regions:
  $\rightarrow d_a = 356$ Mb.

- Use `XSM_FILE` for ASMPIJ data structure.

Region Merging 1

Example of storage requirements for a simple 3–D problem.

- Total volume 1 liter ($V = 10^3$ cm^3).
- Central fissile region is red ($V = 27$ cm^3).
- Strong absorber is green ($V = 1$ cm^3).

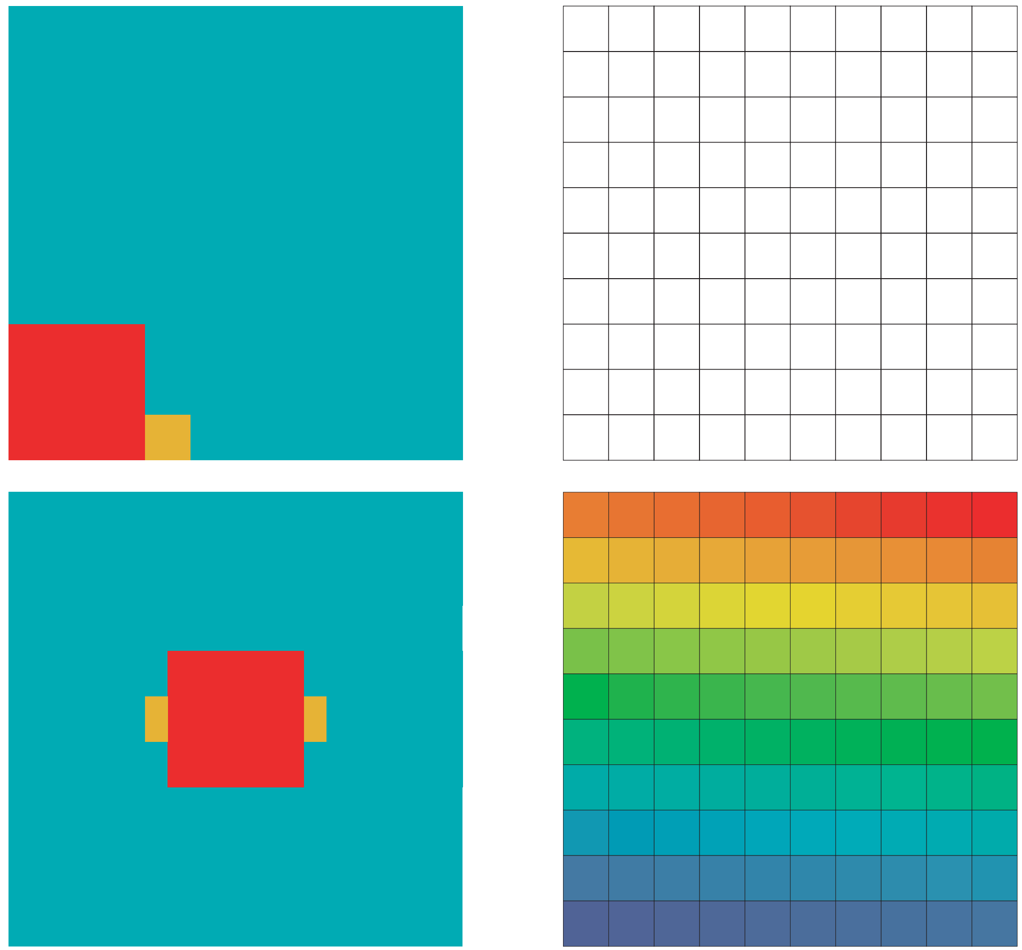


Region Merging 2

- Region with strong absorber:
  - Try to avoid using approximate boundary conditions.
  - Fine mesh discretization is required.
- Region with fission:
  - Try to avoid using approximate boundary conditions.
  - Medium to fine mesh discretization is required.
- For moderator region
  - Fine to coarse mesh discretization is required.

Region Merging

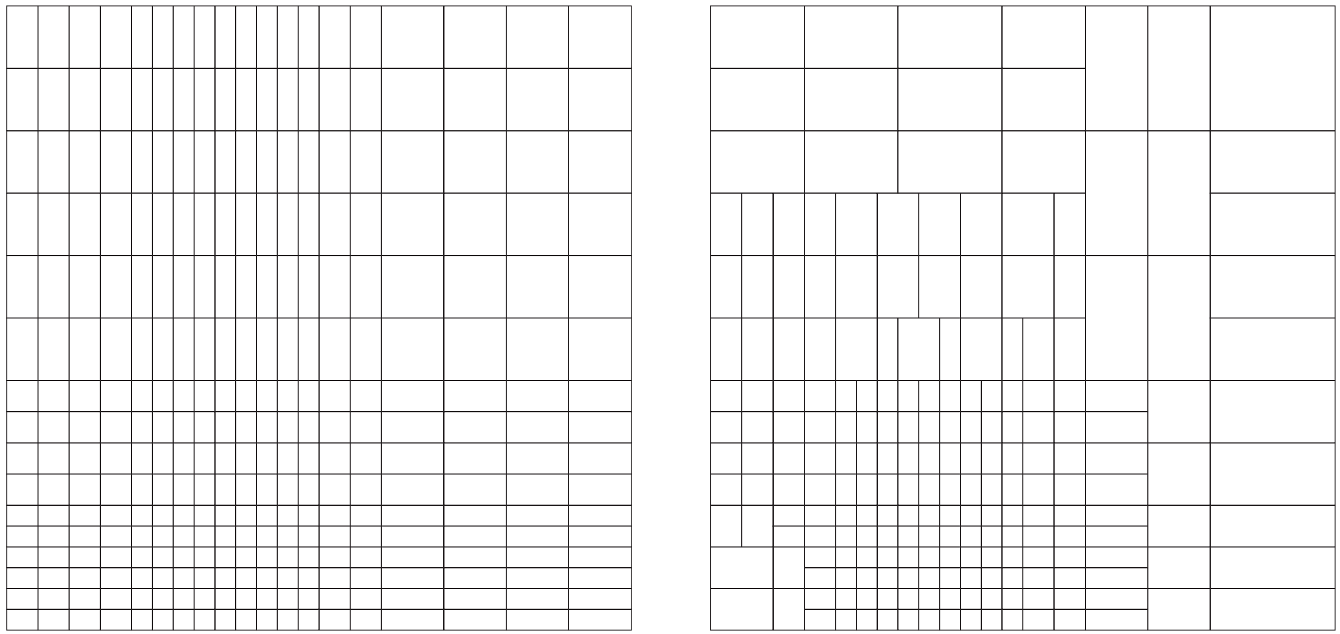
Uniform mesh for simple 3–D problem:



Note: $x - y$ and $x - z$ planes are identical.

Region Merging 4

Non-uniform meshes in DRAGON:



- $N = 19 \times 16 \times 16 = 4864$ regions, $d_a = 100$ Mb/groups for uniform mesh.
- $N \approx 1500$ regions, $d_a = 9$ Mb/groups for non-uniform mesh.

Region Merging

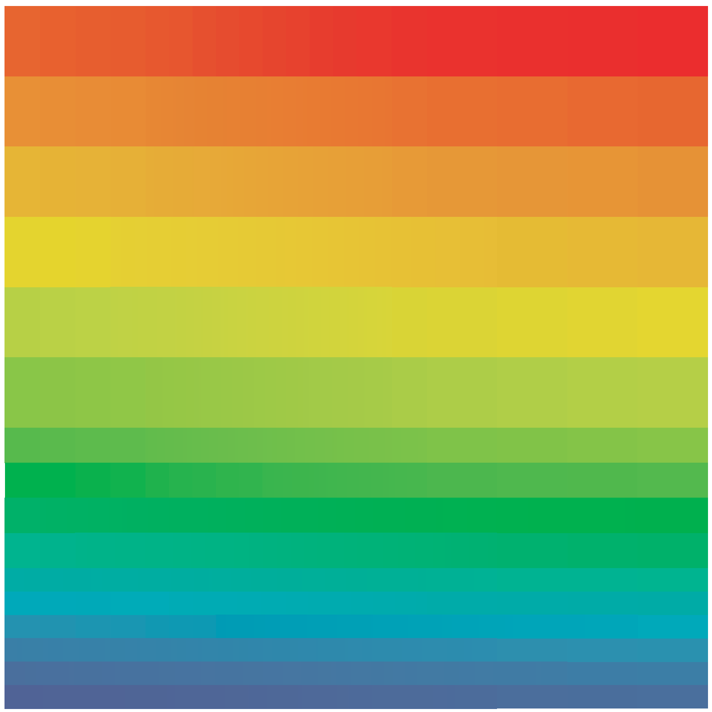
Using the `MRG:` module:

```
TMPV2 TMPTR2 := MRG: TMPVOL TMPTRK    ::
  REGI    1   1   2   3   4   5   6   7   8
          9  10  11  12  13  14  15  16  17  17
          1   1   2  18  19  20  21  22  23
         24  25  26  27  28  29  30  16  17  17
         31  31  32  33  34  35  36  37  38
         39  40  41  42  43  44  45  46  47  47
         31  31  32  48  49  50  51  52  53
         54  55  56  57  58  59  60  46  47  47
         61  62  63  64  65  66  67  68  69
         70  71  72  73  74  75  76  77  78  78
  ...
 ;
```

Region Merging

Region numbering for first (out of 16) z-plane:

Color by region for 304 regions



Color by region for 216 regions

Solving the CP Equations

Contents

- The Power Iteration
- The Multigroup Iteration
- Leakage Models

Solving the CP Equations 2

The multigroup transport equation has the form

$$
\begin{aligned}
\vec{\phi} =& \mathbf{P}_{c,vv}(\vec{q}_s + \frac{1}{k}\vec{q}_f) \\
\vec{q}_s =& \boldsymbol{\Sigma}_s \vec{\phi} \\
\vec{q}_f =& \chi\nu\boldsymbol{\Sigma}_f \vec{\phi}
\end{aligned}
$$

- $\vec{\phi}$ is a $N \times G$ dimensional vector
- $\mathbf{P}_{c,vv}$ is the multigroup CP matrix
  - Diagonal in energy, full in space.
- $\chi\nu\boldsymbol{\Sigma}_f$ is a matrix for neutron production by fission and $\boldsymbol{\Sigma}_s$ is the scattering matrix
  - Diagonal in space, full in energy

Solving the CP Equations 3

We can decompose the scattering matrix as

$$\Sigma_s = \Sigma_{d,s} + \Sigma_{u,s} + \Sigma_{w,s}$$

with

- $\Sigma_{u,s}$ the up-scattering matrix (lower triangular in energy)
- $\Sigma_{d,s}$ the down-scattering matrix (upper triangular in energy)
- $\Sigma_{w,s}$ the within-group scattering matrix (diagonal in energy)

Solving the CP Equations 4

Defining $\mathbf{W}$ the scattering modified CP matrix

$$\mathbf{W} = (\mathbf{I} - \mathbf{P}_{c,vv}\mathbf{\Sigma}_{w,s})^{-1}\mathbf{P}_{c,vv}$$

the transport equation now becomes

$$\vec{\phi} = \mathbf{W}(\mathbf{\Sigma}_{d,s}\vec{\phi} + \mathbf{\Sigma}_{u,s}\vec{\phi} + \frac{1}{k}\vec{q}_f)$$

Assuming

- $\vec{q}_f/k$ is fixed and $\mathbf{\Sigma}_{w,s} = 0$

The above equation can be solved directly from group $g = 1$ to $g = G$

Solving the CP Equations 5

The general equation can be solve using two iteration processes

- The Power iteration
  illustrated by solving the transport problem with $\Sigma_{w,s} = 0$
- The multigroup iteration
  illustrated by solving the transport problem with $\vec{q}_f / k$ fixed

The Power Iteration 1

- Assume $\Sigma_{u,s} = 0$
- Write an iterative group by broup solution to the transport problem as

$$\vec{\phi}(l) = \mathbf{W}\left(\mathbf{\Sigma}_{d,s}\vec{\phi}(l) + \frac{\chi}{k(l-1)}\nu\mathbf{\Sigma}_f\vec{\phi}(l-1)\right)$$

$$k(l) = \sum_{g=1}^{G}\sum_{i=1}^{N} V_i\chi_i^g \sum_{h=1}^{G}\nu\Sigma_{f,i}^h\phi_i^h(l)$$

with $\vec{\phi}(0)$ a known arbitrary flux distribution and

$$k(0) = \sum_{g=1}^{G}\sum_{i=1}^{N} V_i\chi_i^g \sum_{h=1}^{G}\nu\Sigma_{f,i}^h\phi_i^h(0)$$

The Power Iteration 1

The iteration process is repeated until

$$k(l) - k(l-1) < \epsilon_1$$

$$\left| \frac{\vec{\phi}(l)}{k(l)} - \frac{\vec{\phi}(l-1)}{k(l-1)} \right| < \epsilon_2 \left| \frac{\vec{\phi}(l)}{k(l)} \right|$$

are both satisfied

- The parameters ϵ_1 and ϵ_2 can be defined independently in DRAGON

The Multigroup Iteration 1

- Assume $\vec{q}_f/k = \vec{q}$ is fixed
- Solve group-by-group this fixed source problem using a Gauss-Seidel strategy

$$\vec{\phi}(l) = \mathbf{W}\left(\mathbf{\Sigma}_{d,s}\vec{\phi}(l) + \mathbf{\Sigma}_{u,s}\vec{\phi}(l-1) + \vec{q}\right)$$

- Iterate until

$$\left|\frac{\vec{\phi}^g(l) - \vec{\phi}^g(l-1)}{\vec{\phi}^g(l)}\right| < \epsilon_3$$

The Multigroup Iteration 2

Multigroup rebalancing technique

- Neutron conservation states that for a converged solution $\phi_i^g(l)$

$$\sum_{i=1}^{N} \Sigma_i^g V_i \phi_i^g(l) = \sum_{i=1}^{N} R_i^g V_i q_i^g(l)$$
$$+ \sum_{i=1}^{N} R_i^g V_i \left(\sum_{h=1}^{g} \Sigma_s^{h \to g} \phi_i^h(l) + \sum_{h=g+1}^{G} \Sigma_s^{h \to g} \phi_i^h(l) \right)$$

with $R_i^g = 1 - \sum_{i=1}^{N} \Sigma_j^g p_{c,ij}^g$

The Multigroup Iteration 3

- The multigroup problem we effectively solve yields

$$\sum_{i=1}^{N} \Sigma_i^g V_i \phi_i^g(l) = \sum_{i=1}^{N} R_i^g V_i q_i^g(l)$$

$$+ \sum_{i=1}^{N} R_i^g V_i \left(\sum_{h=1}^{g} \Sigma_s^{h\to g} \phi_i^h(l) + \sum_{h=g+1}^{G} \Sigma_s^{h\to g} \phi_i^h(l-1) \right)$$

The Multigroup Iteration 4

To restore conservation at each iteration

- Use $\tilde{\phi}_i^g = \alpha^g \phi_i^g$ and assume $\tilde{\phi}_i^g$ satisfies conservation relations, then α^g must satisfy

$$\sum_{h=1}^{G} M^{h\to g} \alpha^h = q^g$$

$$M^{h\to g} = \sum_{i=1}^{N} R_i^g V_i \left(\Sigma_i^h \delta_{gh} - \Sigma_s^{h\to g} \right) \phi_i^h$$

$$q^g = \sum_{i=1}^{N} R_i^g V_i q_i^g$$

- Solve for $\alpha^g(k)$ and rebalance flux

The Multigroup Iteration 5

Add relaxation parameter to the Gauss–Seidel iteration scheme

- For appromimate solution $\vec{\Gamma}(l)$

$$\vec{\Gamma}(l) = \mathbf{W}\left(\mathbf{\Sigma}_{d,s}\vec{\Gamma}(l) + \mathbf{\Sigma}_{u,s}\vec{\Gamma}(l-1) + \vec{q}\right)$$

- Define an improved flux distribution for the next iteration using

$$\begin{aligned}\vec{\phi}(l) =& \vec{\Gamma}(l) + \omega(l)\vec{\Delta}(l)\\ \vec{\Delta}(l) =& \vec{\Gamma}(l) - \vec{\phi}(l-1)\end{aligned}$$

and $\omega(l)$ will be computed using a variational procedure

The Multigroup Iteration 6

Select $\omega(l)$ in such a way that $\vec{\phi}(l)$ minimizes the tranport functional

$$\mathcal{F}[\vec{\phi}] =\frac{1}{2}\vec{\phi}^T\mathbf{Z}^T\mathbf{Z}\vec{\phi} - \vec{\phi}^T\mathbf{Z}^T\mathbf{W}\vec{q}$$

$$\mathbf{Z} =[\mathbf{I} - \mathbf{W}\left(\mathbf{\Sigma}_{d,s} + \mathbf{\Sigma}_{u,s}\right)]$$

This yield

$$\omega(l) = - \frac{[\vec{\Delta} - \mathbf{W}\vec{S}_1(l)]^T[\vec{\Gamma} - \mathbf{W}\vec{S}_2(l)]}{[\vec{\Delta} - \mathbf{W}\vec{S}_1(l)]^T[\vec{\Delta} - \mathbf{W}\vec{S}_1(l)]}$$

$$\vec{S}_1(l) =(\mathbf{\Sigma}_{d,s} + \mathbf{\Sigma}_{u,s})\vec{\Delta}(l)$$

$$\vec{S}_2(l) =(\vec{q} + \mathbf{\Sigma}_{d,s}\vec{\Gamma}(l) + \mathbf{\Sigma}_{u,s}\vec{\Gamma}(l))$$

Leakage Models

In 3–D, the transport equation can be solved in DRAGON using the B_0 and B_1 leakage models

- Both of these models are based on the following factorization of the flux

$$\Phi^g(\vec{r},\vec{\Omega}) \approx \Psi^g(\vec{r},\vec{\Omega}) \exp(i\vec{B}\cdot\vec{r})$$

- Transport equation with leakage

$$\vec{\Omega}\cdot\vec{\nabla}\Psi^g(\vec{r},\vec{\Omega}) + [\Sigma^g(\vec{r}) + i\vec{B}\cdot\vec{\Omega}]\Psi^g(\vec{r},\vec{\Omega}) = Q_s^g(\vec{r},\vec{\Omega}) + \frac{1}{k_{\mathit{eff}}}Q_f^g(\vec{r})$$

- In general we will assume that k_{eff}=1

Leakage Models 2

- For an infinite homogeneous media the scalar flux and vector current are related to each other according to

$$\vec{\Omega}\Psi^g(\vec{\Omega}) = \vec{J}^g(\vec{\Omega}) = -iD^g\vec{B}\Psi^g(\vec{\Omega})$$

with D^g is an homogeneous diffusion coefficient

- Apply to heterogeneous systems

$$\vec{\Omega}\cdot\vec{\nabla}\Psi^g(\vec{r},\vec{\Omega}) + [\Sigma^g(\vec{r}) + D^g B^2]\Psi^g(\vec{r},\vec{\Omega}) = Q_s^g(\vec{r},\vec{\Omega}) + Q_f^g(\vec{r})$$

- Find the homogeneous diffusion coefficient compatible with this heterogeneous problem

Leakage Models 3

- Assume an heterogeneous solution is known for $B^2 = 0$
- Use this solution to define an equivalent infinite homogeneous problem

$$\Sigma^g \Psi^g(\vec{\Omega}) + i\vec{B} \cdot \vec{J}^g(\vec{\Omega}) = Q_s^g(\vec{\Omega}) + Q_f^g$$

  where the cross sections and sources are homogenized using the heterogeneous flux
- Solve the homogeneous problem for D^g and B
- Insert in heterogeneous transport equation and obtain an improved solution.
- Repeat until the iterative procedure is converged

Leakage Models

Solving the homogeneous problem (B_1 model)

- Use a 2 terms expansion for the scattering cross section in Legendre polynomials

$$\Sigma_s^{h\rightarrow g}(\vec{\Omega}' \rightarrow \vec{\Omega}) = \Sigma_{s,0}^{h\rightarrow g} + 3\Sigma_{s,1}^{h\rightarrow g}\vec{\Omega}\cdot\vec{\Omega}\,'$$

- Define

$$\psi^g = \int d^2\Omega \Psi^g(\vec{\Omega})$$

$$\vec{j}^g = \int d^2\Omega \vec{\Omega}\,' \Psi^g(\vec{\Omega})$$

Leakage Models 5

- Insert into the homogeneous transport equation, and integrate to obtain

$$
\begin{aligned}
\psi^g =& \alpha^g \sum_h (\Sigma_{s,0}^{h\to g} + \chi^g \nu \Sigma_f^h)\psi^h + 3\beta^g \sum_h \Sigma_{s,1}^{h\to g} \frac{\vec{B}\cdot\vec{j}^h}{iB^2} \\
\vec{j}^g =& \beta^g \sum_h \left[(\Sigma_{s,0}^{h\to g} + \chi^g \nu \Sigma_f^h) \frac{\vec{B}\psi^h}{iB^2} + 3\Sigma^g \Sigma_{s,1}^{h\to g} \frac{\vec{j}^h}{B^2} \right] \\
\alpha^g =& \frac{1}{B} \text{arctan} \left(\frac{B}{\Sigma^g} \right) \\
\beta^g =& 1 - \Sigma^g \alpha^g
\end{aligned}
$$

Solve for B, ψ^g and $\vec{j}^g$ and compute $D^g = i\vec{B}\cdot\vec{j}^g / B^2 \psi^g$

Leakage Models

The B_0 homogeneous problem

- Assume $\Sigma_{s,1}^{h\to g} = 0$ and obtain

$$\psi^g = \alpha^g \sum_h (\Sigma_{s,0}^{h\to g} + \chi^g \nu \Sigma_f^h)\psi^h$$

$$\vec{j}^g = \beta^g \frac{\vec{B}}{iB^2} \sum_h (\Sigma_{s,0}^{h\to g} + \chi^g \nu \Sigma_f^h)\psi^h$$

The homogeneous diffusion coefficient is then given by

$$D^g = \frac{\beta^g}{\alpha^g}$$

Condensation and Homogenization 1

Contents

Condensation and Homogenization 2

Condensation and homogenization techniques in DRAGON are based on the following assumptions

- Reaction rates are physically meaningfull and should be preserved by the condensation/homogenization procedure

$$R_i = \sum_g V_i \phi_i^g \Sigma_i^g = V_i \phi_i \Sigma_i$$

$$R^g = \sum_i V_i \phi_i^g \Sigma_i^g = V \phi^g \Sigma^g$$

- The eigenvalue is physically meaningfull and should be preserved by the condensation/homogenization procedure

Condensation Technique 1

Condensed transport equation (macrogroup K that includes $g \in G_K$)

$$\vec{\Omega} \cdot \vec{\nabla} \sum_{g \in G_K} \Phi^g(\vec{r}, \vec{\Omega}) + \sum_{g \in G_K} \Sigma^g(\vec{r}) \Phi^g(\vec{r}, \vec{\Omega}) =$$

$$\sum_{g \in G_K} [Q_s^g(\vec{r}, \vec{\Omega}) + \frac{1}{k} Q_f^g(\vec{r})]$$

Few group version of the same equation is

$$\vec{\Omega} \cdot \vec{\nabla} \Phi^K(\vec{r}, \vec{\Omega}) + \Sigma^K(\vec{r}) \Phi^K(\vec{r}, \vec{\Omega}) = [Q_s^K(\vec{r}, \vec{\Omega}) + \frac{1}{k} Q_f^K(\vec{r}, \vec{\Omega})]$$

It should reproduce condensed multigroup results

Condensation Technique 2

The condensation procedure that satisfies our requirements

$$\phi_i^K = \sum_{g \in G_K} \phi_i^g$$

$$\Sigma_i^K = \frac{1}{\phi_i^K} \sum_{g \in G_K} \Sigma_i^g \phi_i^g$$

$$\Sigma_{s,i}^{L \to K} = \frac{1}{\phi_i^L} \sum_{h \in G_L} \sum_{g \in G_K} \Sigma_{s,i}^{h \to g} \phi_i^h$$

$$\chi_i^K = \sum_{g \in G_K} \chi_i^g$$

$$\nu\Sigma_{f,i}^K = \frac{1}{\phi_i^K} \sum_{g \in G_K} \nu\Sigma_{f,i}^h \phi_i^g$$

Condensation Technique 3

Multiplying CP transport equation by $\Sigma_i^g V_i$ and summing over all regions i yields

$$\sum_i \Sigma_i^g V_i \phi_i^g = \sum_i V_i [Q_{s,i}^g + \frac{1}{k} Q_{f,i}^g]$$

The equivalent transport equation in a homogeneous infinite cell is

$$\hat{\Sigma}^g V \hat{\phi}^g = V [\hat{Q}_{s,i}^g + \frac{1}{k} \hat{Q}_{f,i}^g]$$

The homogenized and homogeneous transport equations are identical if one selects a flux-volume homogenization technique

Condensation Technique 4

Flux-volume homogenization technique

$$
\begin{aligned}
\hat{\phi}^g &= \frac{1}{V} \sum_i V_i \phi_i^g \\
\hat{\Sigma}^g &= \frac{1}{V \hat{\phi}^g} \sum_i V_i \Sigma_i^g \phi_i^g \\
\hat{\Sigma}_s^{h \to g} &= \frac{1}{V \hat{\phi}^h} \sum_i V_i \Sigma_{s,i}^{h \to g} \phi_i^h \\
\hat{\nu} \hat{\Sigma}_f^g &= \frac{1}{V \hat{\phi}^g} \sum_i V_i \nu \Sigma_{f,i}^g \phi_i^g \\
\hat{\chi}^g &= \frac{1}{V \sum_h \hat{\nu} \hat{\Sigma}_f^g \hat{\phi}^g} \sum_i \chi_i^g V_i \sum_h \nu \Sigma_{f,i}^g \phi_i^h
\end{aligned}
$$

Condensation Technique

Flux-volume homogenization fails if

- The cell is finite (a cell with leakage) and

$$\sum_{j=1}^{N_j} p_{ij}^g \Sigma_j^g \neq 1$$

- Partial cell homogenization cell is considered

Partial Cell Homogenization 1

The heterogeneous N region transport equation homogenized over M regions takes the form

$$\sum_{i\in M_I} V_i\Sigma_i^g\phi_i^g = \sum_{i\in M_I}\sum_J\sum_{j\in M_J} p_{ji}^g(\Sigma^g)[Q_{s,i}^g + \frac{1}{k}Q_{f,i}^g]$$

The M region heterogeneous transport equation takes the from

$$V_I\hat{\Sigma}_I^g\hat{\phi}_I^g = \sum_J \hat{p}_{JI}^g(\hat{\Sigma}^g)[Q_{s,J}^g + \frac{1}{k}Q_{f,J}^g]$$

where $\hat{P}_{JI}^g(\hat{\Sigma}^g)$ indicates that the CP are computed using homogenized cross sections

Partial Cell Homogenization 2

We need

$$\sum_{i \in M_I} V_i \Sigma_i^g \phi_i^g = V_I \Sigma_I^g \phi_I^g$$

and

$$\sum_J \hat{p}_{JI}^g(\hat{\Sigma}^g)[Q_{s,J}^g + \frac{1}{k} Q_{f,J}^g] =$$

$$\sum_{i \in M_I} \sum_J \sum_{j \in M_J} p_{ji}^g(\Sigma^g)[Q_{s,i}^g + \frac{1}{k} Q_{f,i}^g]$$

to be simultaneously true

Partial Cell Homogenization

The flux-volume homogenization method is not longer adequate because

- There is no simple relation between $\hat{p}^g_{JI}(\hat{\Sigma}^g)$ and $p^g_{ji}(\Sigma^g)$

The alternative here is to use a non-linear process

- Consider a flux-volume homogenization for ϕ^g_I and Σ^g_I
- Redefine the homogeneous flux $\hat{\phi}^g_I$ and cross sections $\hat{\Sigma}^g_I$ as follows

$$\hat{\phi}^g_I = \frac{1}{\mu^g_I}\phi^g_I \qquad \hat{\Sigma}^g_I = \mu^g_I \Sigma^g_I$$

Partial Cell Homogenization 4

- Determine the SPH factors μ_I^g numerically in such a way that

$$\sum_J \hat{p}^g_{JI}(\hat{\Sigma}^g)[Q^g_{s,J} + \frac{1}{k}Q^g_{f,J}] =$$

$$\sum_{i\in M_I}\sum_J\sum_{j\in M_J} p^g_{ji}(\Sigma^g)[Q^g_{s,i} + \frac{1}{k}Q^g_{f,i}]$$

is true

- The definition of the SPH factors automatically ensures

$$\sum_{i\in M_I} V_i\Sigma_i^g\phi_i^g = V_I\Sigma_I^g\phi_I^g = V_I\tilde{\Sigma}_I^g\tilde{\phi}_I^g$$

Microscopic Cross Section 1

The macroscopic cross section associated with a material is simply the sum over all isotopes of the isotopic macroscopic cross section Σ_I namely

$$\Sigma_i^g = \sum_I \Sigma_{I,i}^g$$

where

$$\Sigma_{I,i}^g = N_{I,i} \sigma_I^g$$

with $N_{I,i}$, the concentration of isotope I in region i

- The homogenization and condensation procedure described above remain valid for $\Sigma_{I,i}^g$

Microscopic Cross Section 2

Since the final concentration of isotope I in the cell is given by:

$$N_I = \frac{1}{V} \sum_i N_{I,i} V_i$$

we can define the equivalent homogenized microscopic cross section as:

$$\hat{\sigma}_I^K = \frac{\mu_I^K}{N_I V \phi^K} \sum_{i \in M_I} \sum_{g \in G_K} N_{I,i} V_i \sigma_I^g \phi_i^g$$

where the microscopic cross sections now become dependent on the spatial position

Managing a DRAGON Execution 1

Contents

- Input file formats.
- Data structure formats.
- Working with variables.
- Conditional execution and loops.
- Working with procedures.
- Flow chart in DRAGON input decks

Managing a DRAGON Execution 2

Input file format

- 72 columns, free format instruction ends by `;`
- Comments `*` or `!`
- `MODULE` and objects declarations
- Sequence of calls to modules

```
(list of output objects)  :=  GEO:  (list of input objects)  ::
                                      (data input)           ;
```

- `END:␣;` statement
- `QUIT␣"LIST"␣.` end compilation

Managing a DRAGON Execution 3

Data structure formats.

- `LINKED_LIST` Memory access
- `XSM_FILE` Direct-access file
- `SEQ_BINARY` Tracking information mainly
- `SEQ_ASCII` Machine independent format
- `DIR_ACCESS` XS library file

Managing a DRAGON Execution 4

Variable types.

- INTEGER (signed) Numbers
- REAL (signed) Decimal numbers with E or .
- DOUBLE (signed) Decimal numbers with D and .
- STRING 72 character long, enclosed in " "
- LOGICAL = $True_L or $False_L

Variable names are case sensitive.

Managing a DRAGON Execution 5

Assign or Evaluate variables

REAL	(variable names)	:=	(value)	;
EVALUATE	(variable names)	:=	(value)	;
ECHO	(variable names)	;		

Variable in data input deck.

- `<< . >>` access the content of a variable `send`
- `>> . <<` put a value into a variable `recover`

Managing a DRAGON Execution 6

Operations on variables.

- Reverse Polish Notation
  (value) (operator) (value) $\Leftrightarrow$ (value) (value) (operator)
- Arithmetic operations + - * / **
  Ex : `delta := b 2 **  4. a c * * -`
- Unary operations COS SQRT ABS NOT LN
  Ex : `delta := delta SQRT`
- Relational operations < > <> <= + -
  Ex : `condition := a b <=`
- Operations on `STRING` variables + -
- NO mixed mode operations

Managing a DRAGON Execution 7

IF/THEN/ELSE statement

```
IF      (condition)  THEN  (statements)
ELSEIF  (condition)  THEN  (statements)
                     ELSE  (statements)
ENDIF      ;
```

WHILE and REPEAT statement

```
WHILE (condition) DO        | REPEAT
      (modif condition)     |     (modif condition)
      (statements)          |     (statements)
ENDWHILE ;                  | UNTIL (condition) ;
```

Working with procedures

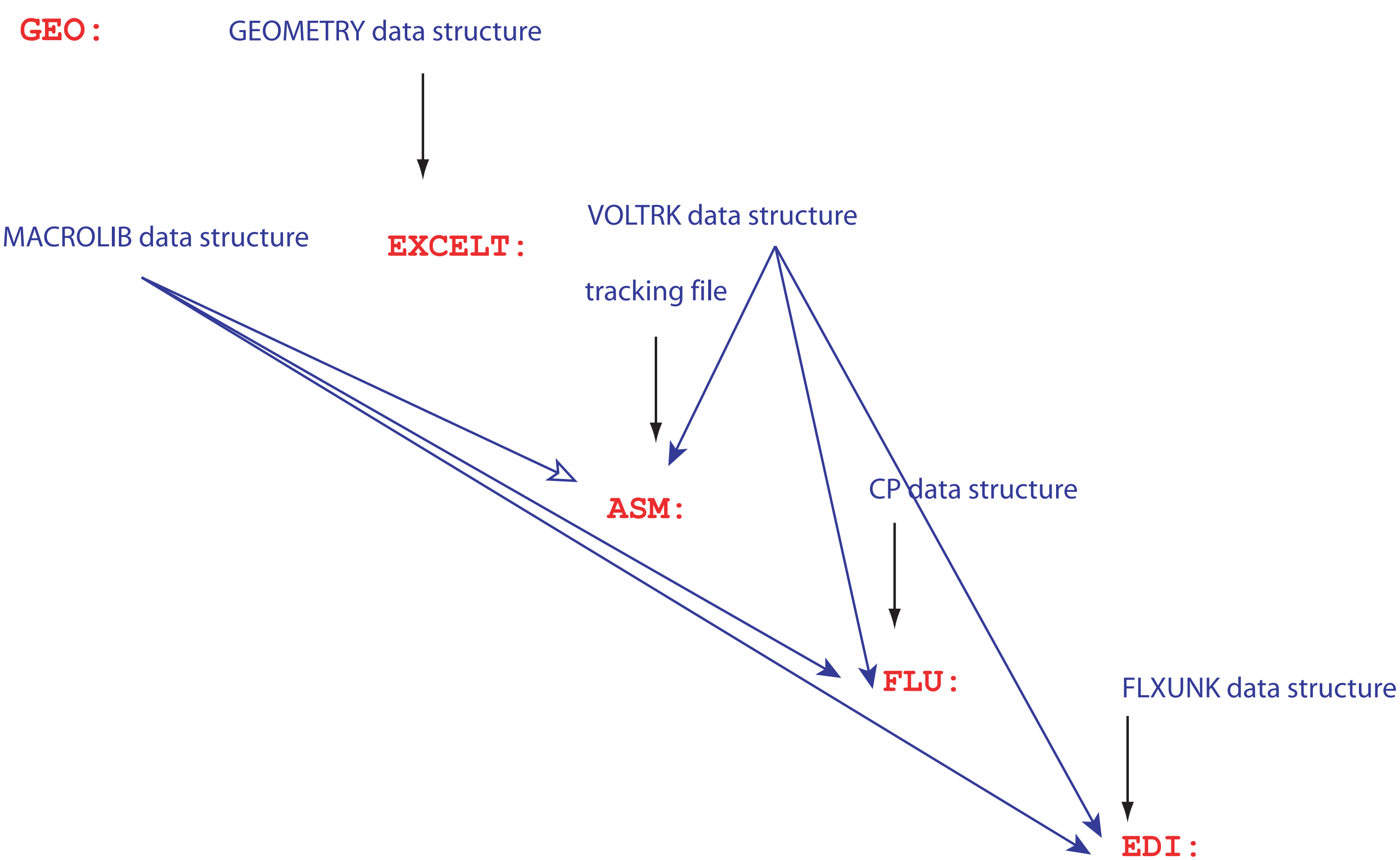
Calling Part: Main File	Called Part: Procedure File procedure name.c2m
`PROCEDURE (procedure name) ;`	

`(output objects) := (procedure name) (input objects) ::`	`PARAMETER (output objects) (input objects) ::`
	`:: LINKED_LIST (object name) ;`
	`SEQ_ASCII (object name) ;`
	`;`

(data input)	(same data input list)
`<< .>>` send	`:: >>.<< ;` recover
`>>.<<` recover	`::<< .>> ;` send
`;`	`END: ;`

Managing a DRAGON Execution

Flow chart in DRAGON input decks



Managing a DRAGON Execution 10

Input cross sections

- Macroscopic library
  - `MAC:` data input
  - `EDI:` DRAGON calculations
  - `MODULE:` other transport codes, WIMS TAPE16 ...
  - `LIB:` microscopic library

$\Rightarrow$ object input in `ASM:` or `FLU:` and `EDI:`

Managing a DRAGON Execution 11

Tracking validation

- Set relatively low line densities and angle number
- Verify volume and surface integration errors
- `EXCELT:` Useful print levels `EDIT` *iprint* :
  - *iprint* = 0 no printing
  - *iprint* = 1 (default) geometric information and echo of data input
  - *iprint* = 2 tracking error on volumes and surfaces
  - *iprint* = 5 surface and region numbering and description, cell by cell and then global by plane in 3D

Managing a DRAGON Execution 12

```
*----
*  Use           : Perform 3-D reactivity device analysis
*  Procedures    :
*     DevGeo      ->  generate 3-D geometry
*     DevEva      ->  solve 3-D transport problem
*  Input files   :
*     MACRO       ->  macrolib for reactivity devices
*  Output files  :
*     Results     ->  Edition results
*----
PROCEDURE    DevGeo DevMac   ;
MODULE       EDI:  DELETE:    BACKUP:    END:   ;
SEQ_ASCII    MACRO Results                      ;
LINKED_LIST Volumes Macrolib Edition Fluxes  PIJ ;
SEQ_BINARY   Tracks                             ;
Macrolib := MACRO    ;
STRING      DevType   := "ADJ1"     ;
STRING      DevLocation    ;
```

Managing a DRAGON Execution 13

```
EVALUATE  DevLocation := "IN"    ;
Volumes Tracks  := DevGeo  ::  <<DevType>> <<DevLocation>> ;
PIJ      := ASM:  Macrolib  Volumes Tracks ;
Fluxes   := FLU:  PIJ Macrolib Volumes  ::  TYPE B B1 PNL ;
Edition := EDI:   Fluxes Macrolib Volumes      ::
  COND 0.626 MERGE COMP SAVE                  ;
PIJ Volumes Tracks Fluxes  := DELETE: PIJ Volumes Tracks Fluxes ;
EVALUATE  DevLocation := "OUT"  ;
Volumes Tracks  := DevGeo  ::  <<DevType>> <<DevLocation>> ;
PIJ      := ASM:  Macrolib  Volumes Tracks ;
Fluxes   := FLU:  PIJ Macrolib Volumes  ::  TYPE B B1 PNL ;
Edition := EDI:  Edition Fluxes Macrolib Volumes  :: SAVE ;
PIJ Volumes Tracks Fluxes  := DELETE: PIJ Volumes Tracks Fluxes ;
Results  := Edition                     ;
Edition Macrolib   := DELETE: Edition Macrolib ;
END:                                    ;
QUIT "LIST" .
```

Managing a DRAGON Execution 14

```
*----
*   Input and output structures and variables
*   Volumes          : output LINKED_LIST containing geometry analysis
*   Tracks           : output sequential binary file containing
*                      integration lines
*   Device           : string variable for yype of devices
*                      "ADJn" -> for adjuster rods (Type n = 1,6)
*   DevLocation       : string variable for device Location
*                      "IN"  -> Device and guide tube in
*                      "OUT" -> Device out and guide tube in
*----
PARAMETER   Volumes Tracks  ::
    ::: LINKED_LIST Volumes            ;
    ::: SEQ_BINARY  Tracks             ;  ;
STRING    Device                       ;
INTEGER   DevLocation                  ;
:: >>Device<<  >>DevLocation<<         ;
```

Managing a DRAGON Execution 15

```
MODULE          GEO:  EXCELT:  DELETE:       ;
LINKED_LIST     GEOMETRY                     ;
REAL        RA1 RA2 RA3 RA4 RA5 RA6 RAM  ;
IF       Device "ADJ1" = THEN
  EVALUATE  RA1    RA2    RA3    RA4    RA5    RA6      :=
            0.577 3.678 3.810 4.445 4.752 6.378    ;
ELSEIF   Device "ADJ2" = THEN
  EVALUATE  RA1    RA2    RA3    RA4    RA5    RA6      :=
            0.649 3.723 3.810 4.445 4.752 6.378    ;
ENDIF ;
INTEGER  NbReg   NbAngles := 48 8    ;
REAL     TrkDens          := 25.0    ;
```

Managing a DRAGON Execution 16

```
INTEGER   MM   MA1  MA2  MA3  MA4  MA5  MA6 ;
IF  DevLocation "IN" = THEN
  EVALUATE    MM    MA1  MA2  MA3  MA4  MA5  MA6 :=
              11    12   13   14   15   16   17  ;
ELSE
  EVALUATE    MM    MA1  MA2  MA3  MA4  MA5  MA6 :=
              11    11   11   11   11   11   17  ;
ENDIF ;
```

Managing a DRAGON Execution 17

```
GEOMETRY   := GEO:  :: CAR3D 2 1 2
  CELL    FC1B  MD1B  FC1T  AD1T
  X- REFL X+ SYME  Y- REFL Y+ SYME   Z- REFL Z+ SYME
  ::: FC1B := GEO: CARCELZ 5 3 4 1
   ... ;
  ::: MD1B := GEO: CAR3D     2 4 1
   ... ;
  ::: FC1T := GEO: CARCELZ 5 3 4 2
   ... ;
  ::: AD1T := GEO: CARCELY 6 2 4 2
   ... ;
;
Volumes Tracks := EXCELT: GEOMETRY ::
  MAXR  <<NbReg>> TRAK TISO <<NbAngles>> <<TrkDens>> ;
GEOMETRY := DELETE: GEOMETRY ;
QUIT "LIST" .
```

Conclusions

Some comments and warning on the CP method

- The sources are assumed constant inside each region
  - Select an adequate spatial discretization
    This may lead to a large number of region (CP is proportionnal to N^2)
    Some regions may be very small causing problem with tracking
- Select a problem that is not too heterogeneous

Conclusions 2

- The angular flux on each external surface are assumed constant and isotropic
  - Try to get rid of external surfaces with re-entrant angular flux
  - Select a model where the region of interest is far from the external surfaces