Computational Particle Transport theory and Achieving Solutions in Real Time.

Dr. Alireza Haghighat
Professor and Director
Nuclear Engineering Program
Virginia Tech
haghighat@vt.edu
https://nuclear.ncr.vt.edu

Presented at the Colloquium of the Faculty of Math and Physics, JSI, Slovenia, Jan 14, 2019

Why real time?

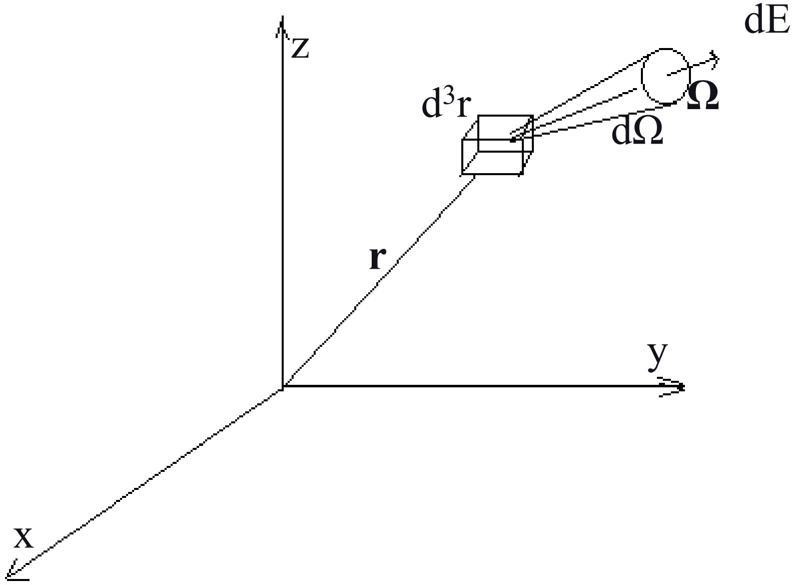
- For ***monitoring and in-situ detection***, accurate solutions are needed in real-time.

- For design and analysis of nuclear systems, 1000’s of accurate simulations are needed in real time. This is especially important for the design of ***advanced nuclear system***

- ***Image reconstruction for medical*** application can benefit from real-time simulations; this can lead to an ***improved image quality, better diagnosis, and reduced radioactivit***y!

Particle Transport

- **Determine the expected number of particles in a phase space ($\boldsymbol{d^3rdEd\Omega}$) at time $\boldsymbol{t}$:**



$$n(\vec{r}, E, \hat{\Omega}, t)d^3rdEd\Omega$$

- There are several physical quantities (e.g., flux, current and reaction rate) are defined based on the number density.

Particle Transport Simulation

- Deterministic Methods
    - Using the Boltzmann equation and its approximations

- Statistical Monte Carlo Methods
    - Simulate particle transport on a computer using random numbers to sample from probability density functions associated with physical events

Deterministic - Linear Boltzmann Equation (LBE) for *neutral particle transport*

Intro-differential form of LBE

streaming

collision

$$\hat{\Omega}.\nabla\Psi(\vec{r},E,\hat{\Omega})+\sigma(\vec{r},E)\Psi(\vec{r},E,\hat{\Omega})=$$

scattering

$$\int_{0}^{\infty}dE'\int_{4\pi}d\Omega'\sigma_{s}(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega})\Psi(\vec{r},E',\hat{\Omega})+$$

fission

Independent source

$$\frac{\chi(E)}{4\pi}\int_{0}^{\infty}dE'\int_{4\pi}d\Omega'\upsilon\sigma_{f}(\vec{r},E')\Psi(\vec{r},E',\hat{\Omega}')+S(\vec{r},E,\hat{\Omega})$$

Integral form of LBE

$$\psi(\vec{r},E,\widehat{\Omega})=\int_{0}^{R}d\ell Q(\vec{r}-\ell\widehat{\Omega})e^{-\tau_{E}(\vec{r},\vec{r}-\ell\widehat{\Omega})}+\psi(\vec{r}-R\widehat{\Omega},E,\widehat{\Omega})e^{-\tau_{E}(\vec{r},\vec{r}-R\widehat{\Omega})}$$

LBE Numerical Formulations

- Over the past 60 years, various numerical formulations have been derived for LBE. Table below provides an overview of more popular techniques used for handling different variables

Variable	LBE form	Methods	Comments
Angle	Integro-diff & Integral	Expansion of angular-dependent functions in terms of $P_\ell(\mu)$& $Y_{\ell,m}(\mu,\varphi)$	Solves for moments of angular fluxes
		Discrete ordinates (Sn)	LBE is solved for discrete ordinates; ordinates and associated weights to be determined
Space	Integro-diff	Finite volume, Finite element, nodal	LBE solved for average angular flux over mesh volumes; Differencing schemes (fitting functions) are needed
	Integral	Method of Characteristics (MOC)	LBE solved along characteristic rays
Energy	Integro-diff & integral	Multigroup	Flux weighted; flux & importance weighted; CPXSD methodology

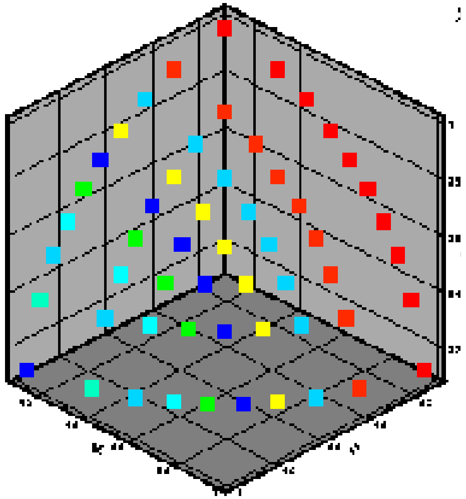
Discrete Ordinates

- LBE is written for a discrete set of angles with associated weights, $\{\Omega_n, w_n\}$,

$$\widehat{\Omega}_n \cdot \nabla\Psi\left(\vec{r}, E, \widehat{\Omega}_n\right) + \sigma_t(\vec{r}, E)\, \Psi\left(\vec{r}, E, \widehat{\Omega}_n\right) = q\left(\vec{r}, E, \widehat{\Omega}_n\right), \qquad \text{for } n = 1, N(N+2)$$

- Directions
  - generated by considering **90° rotational invariance, e.g.,** $\mu_i = \eta_i = \xi_i,\ i = 1, N$; leading to $\frac{N(N+2)}{8}$ directions per octant
  - Their weights are obtained by **preserving the moments** of direction cosines, e.g., $\int_{-1}^{1} du \mu^m = \frac{2}{m+1},\ or\ \sum_{n=1}^{N(N+2)} w_n \mu_n^m = \frac{2}{m+1},$ for even m's

S



NOTE: Typical quadrature order: for reactor shielding is S_8 (80 directions) or above, and for medical application **S_{24} (624 directions)** or above

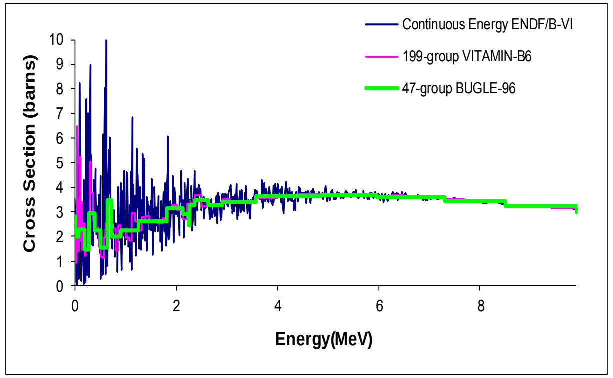
Multigroup LBE

- **Integrate the LBE** over energy groups

$$\int_{E_g}^{E_{g-1}} dE\,\widehat{\Omega}\cdot\nabla\psi\big(\vec{r},E,\widehat{\Omega}\big)+\int_{E_g}^{E_{g-1}} dE\,\sigma_t(\vec{r},E)\psi\big(\vec{r},E,\widehat{\Omega}\big)=$$

$$\int_{E_g}^{E_{g-1}} dE\int_0^{\infty} dE'\int_{4\pi} d\Omega'\sigma_s(\vec{r},E'\rightarrow E,\mu_0)\psi\big(\vec{r},E',\widehat{\Omega}'\big)+$$

$$\int_{E_g}^{E_{g-1}} dE\,\frac{\chi(E)}{4\pi k}\int_0^{\infty} dE'\int_{4\pi} d\Omega'\sigma_f(\vec{r},E'\,)\psi\big(\vec{r},E',\widehat{\Omega}'\big)$$



and obtain ***average group cross sections*** by

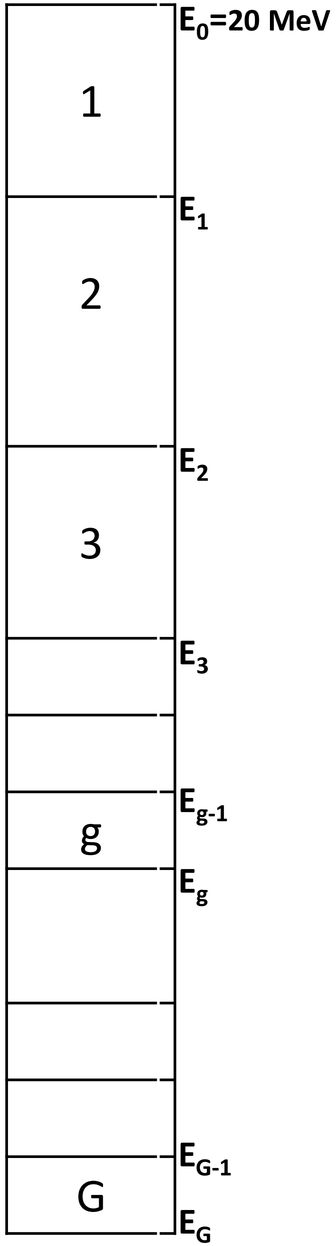
$$\sigma_{t,g}(\vec{r})=\frac{\int_{E_g}^{E_{g-1}} dE\int_{4\pi} d\Omega\,\sigma_t(\vec{r},E)\psi\big(\vec{r},E,\widehat{\Omega}\big)}{\int_{E_g}^{E_{g-1}} dE\int_{4\pi} d\Omega\,\psi\big(\vec{r},E,\widehat{\Omega}\big)}=\frac{\int_{E_g}^{E_{g-1}} dE\,\sigma_t(\vec{r},E)\phi(\vec{r},E\,)}{\phi_g(\vec{r})}$$

Hence,

$$\int_{E_g}^{E_{g-1}} dE\,\sigma_t(\vec{r},E)\phi(\vec{r},E\,)=\sigma_{t,g}(\vec{r})\phi_g(\vec{r})$$

Typical number of energy groups for shielding problems is **50.**

Energy groups



Spatial Finite-volume Technique

- The LBE is integrated over discretized volumes, e.g., collision term in a Cartesian geometry reduces to:

$$\int_{\Delta V_{ijk}} d^3r\sigma_{t,g}(\vec{r})\psi_{n,g}(\vec{r}) = \sigma_{t,g,ijk}\bar{\psi}_{n,g,ijk}\Delta x_i\Delta y_j\Delta z_k$$

$$\bar{\psi}_{n,g,ijk} = \frac{\int_{\Delta V_{ijk}} d^3r\psi_{n,g}(\vec{r})}{\Delta x_i\Delta y_j\Delta z_k}$$

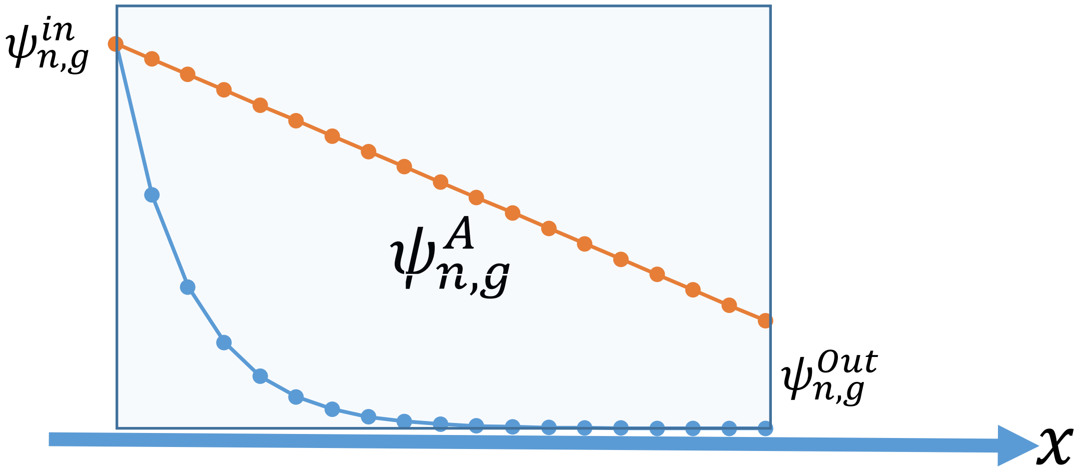
Hence, the 3-D multigroup, finite-volume, Sn equations are given by

$$\frac{\mu_n}{\Delta x_i}\left(\psi_{n,g}^{Out} - \psi_{n,g}^{in}\right) + \frac{\eta_n}{\Delta y_j}\left(\psi_{n,g}^{L} - \psi_{n,g}^{R}\right) + \frac{\xi_n}{\Delta z_k}\left(\psi_{n,g}^{T} - \psi_{n,g}^{B}\right) + \sigma_{t,g}\psi_{n,g}^{A} = q_{n,g}^{A}$$

- There are 7 unknowns and only one equation, there 3 BC's, and 3 auxiliary equations are needed to express the relation *between the boundaries and cell-average angular fluxes along x, y, or z axes.*

Auxiliary Equations - Differencing Schemes

- A “differencing scheme” refers spatial distribution of the angular flux (along x, y, or z) within a mesh cell, e.g., **along x axis**



- Desired characteristics of a differencing scheme

  Accurate

  Positive

  No unphysical oscillations

  Cheap (time & memory)

Examples for Differencing schemes (linear fit)

x-axis $$\psi_{n,g,ijk} = a_{n,g,ijk}\psi_{n,g,i-\frac{1}{2},j,k} + (1 - a_{n,g,ijk})\psi_{n,g,i+\frac{1}{2},j,k}$$

y-axis $$\psi_{n,g,ijk} = b_{n,g,ijk}\psi_{n,g,i,j-\frac{1}{2},k} + (1 - b_{n,g,ijk})\psi_{n,g,i,j+\frac{1}{2},k}$$

z-axis $$\psi_{n,g,ijk} = c_{n,g,ijk}\psi_{n,g,i,j,k-\frac{1}{2}} + (1 - c_{n,g,ijk})\psi_{n,g,i,j,k+\frac{1}{2}}$$

Linear Diamond (LD)

$$a_{n,g,ijk} = b_{n,g,ijk} = c_{n,g,ijk} = \frac{1}{2} \quad \Longrightarrow \quad \psi_A = \psi_{gijk}^{(n)} = \frac{\frac{2\mu_n}{\Delta x}\psi_{x\text{ in}} + \frac{2\eta_n}{\Delta y}\psi_{\text{y in}} + \frac{2\xi_n}{\Delta z}\psi_{\text{z in}} + Q_{gijk}^{(n)}}{\sigma_{ijk} + \frac{2\mu_n}{\Delta x} + \frac{2\eta_n}{\Delta y} + \frac{2\xi_n}{\Delta z}}$$

Directional $\boldsymbol{\theta - weighted\ (DTW)}$* (e.g., for *a* coefficient along *x; b & C coefficients have similar formulations)

$$a = 1 - \frac{\frac{|\mu_m|}{\Delta x}\Psi_{in,x} + \theta(\mu_m)\left(\frac{|\eta_m|}{\Delta y}\Psi_{in,y} + \frac{|\xi_m|}{\Delta z}\Psi_{in,z}\right) + q_A}{\left(\sigma + 2\frac{|\eta_m|}{\Delta y} + 2\frac{|\xi_m|}{\Delta z}\right)\Psi_{in,x}} \quad \Longrightarrow \quad \psi_A = \frac{q_A + \frac{|\mu_m|}{a\Delta x}\psi_{\text{in }x} + \frac{|\eta_m|}{b\Delta y}\psi_{\text{in }y} + \frac{|\xi_m|}{c\Delta z}\psi_{\text{in }z}}{\left(\frac{|\mu_m|}{a\Delta x} + \frac{|\eta_m|}{b\Delta y} + \frac{|\xi_m|}{c\Delta z} + \sigma\right)}$$

Examples for Differencing schemes (Exponential fit)

(Predicted – Corrected scheme)

$$\psi_{n,g}(x,y,z) = a_0 e^{\frac{\lambda_i P_1(x)}{|\mu_n|}} e^{\frac{\lambda_j P_1(y)}{|\eta_n|}} e^{\frac{\lambda_k P_1(z)}{|\xi_n|}}$$

Where,

$$P_1(u) = \frac{2u}{\Delta u} - 1, for\ \ 0 \le u \le \Delta u, \mathrm{u} \equiv x, y, or\ z$$

Exponential Directional Weighted (EDW)

$$\Psi_A = \left(\exp\left(\frac{2\lambda_i}{|\mu_m|}\right)-1\right)\left(\exp\left(\frac{2\lambda_j}{|\eta_m|}\right)-1\right)\left(\exp\left(\frac{2\lambda_k}{|\xi_m|}\right)-1\right)$$

$$\cdot\frac{1}{\beta}\left(q_{m,i,j,k} + \frac{|\mu_m|}{\Delta x_i}\Psi_{in,x} + \frac{|\eta_m|}{\Delta y_j}\Psi_{in,y} + \frac{|\xi_m|}{\Delta z_k}\Psi_{in,z}\right)$$

And, for example the cell boundary angular flux is given by

$$\Psi_{out,x} = \Psi_A \frac{2\lambda_i}{|\mu_m|}\left(1-\exp\left(\frac{2\lambda_i}{|\mu_m|}\right)\right)^{-1}$$

Similar equations are given for y and z axes

β coefficient formulation

$$\begin{aligned}\beta = {} & \frac{2\lambda_i}{\Delta x}\left(\exp(\frac{2\lambda_i}{|\mu_m|})\right)\left(\exp(\frac{2\lambda_j}{|\eta_m|})-1\right)\left(\exp(\frac{2\lambda_k}{|\xi_m|})-1\right)+\\ & \frac{2\lambda_j}{\Delta y}\left(\exp(\frac{2\lambda_j}{|\eta_m|})\right)\left(\exp(\frac{2\lambda_i}{|\mu_m|})-1\right)\left(\exp(\frac{2\lambda_k}{|\xi_m|})-1\right)+\\ & \frac{2\lambda_k}{\Delta z}\left(\exp(\frac{2\lambda_k}{|\xi_m|})\right)\left(\exp(\frac{2\lambda_i}{|\mu_m|})-1\right)\left(\exp(\frac{2\lambda_j}{|\eta_m|})-1\right)+\\ & \sigma\left(\exp(\frac{2\lambda_i}{|\mu_m|})-1\right)\left(\exp(\frac{2\lambda_j}{|\eta_m|})-1\right)\left(\exp(\frac{2\lambda_k}{|\xi_m|})-1\right)\end{aligned}$$

Hybrid Differencing
Adaptive Differencing Strategy (ADS)

This strategy allows for the use of different differencing schemes in different regions depending on problem physics and meshing

LD → DTW → EDW (or EDI)

Implemented in PENTRAN

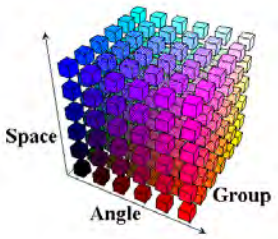
Discretized LBE

- Requires large memory, e.g.,
  - Considering: $80\ ordinates, 50\ groups, and\ 100 * 100 * 100\ spatial\ meshes,$
  - There are $4x10^9$ unknowns and therefore 16 GB of memory is needed to save only the angular flux distribution
- Therefore,

  Parallel algorithms are needed for Memory partitioning and multitasking

Developed PENTRAN^{TM} (Parallel Environment Neutral-particle TRANsport)

(G. Sjoden & A. Haghighat, 1996)



- ***Solves the integro-differential LBE***
  - ANSI FORTRAN F77/f90 with MPI library, over 33,000 lines
  - Industry standard FIDO input
- **Solves 3-D Cartesian, multigroup, anisotropic transport problems**
  - Forward and adjoint mode
  - Fixed source, criticality eigenvalue problems
- **Parallel processing algorithms (using the MPI libraries)**
  - Full **phase-space decomposition**: Parallel in angle, energy, and spatial variables
  - Parallel I/O, Partitioned memory
- **Unique numerical formulations:**
  - **Sn method** with different angular quadrature sets
  - **Finite volume** with the Adaptive Differencing Strategy
  - Different **iteration** schemes
  - Different **acceleration** schemes

Application of PENTRAN^{TM}

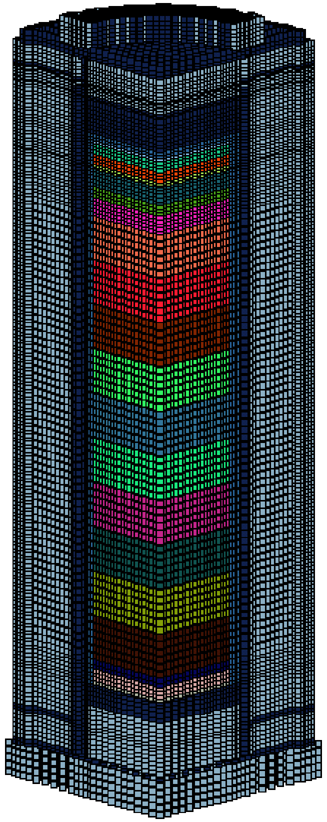
- Kobayashi 3-D Benchmarks
- VENUS-3 Benchmark facility
- BWR Core-Shroud
- Pulsed Gamma Neutron Activation Analysis (PGNAA) device
- X-Ray room
- Time-of-Flight (TOF)
- Spent fuel storage cask
- C5G7 MOX Criticality Benchmark
- UFTR Water tank Characterization
- SNM Detection
- PENBURN application
- PENTRAN-MP application
- PENTRAN-CRT application

NOTE

- Has benchmarked the performance and accuracy of PENTRAN>
- Achieved accurate solutions for complex large problems, but require significant number of computer cores (100’s or 1000’s) and computer time (hours and days)
- Additionally, discretization may lead to inaccuracy of the models

Spent Fuel Storage Cask Dimensions

- Height ~ 610 cm
- Shell O.D. ~340 cm
- Shell I.D. ~187 cm
- Empty Weight 269,000 lbs (55.3 MT)
- Max. Loaded Weight 358,000 lbs (162.4 MT)



'Large' Model

CASK library
(22n, 18g)

17 Materials

318,426 fine meshes (1000 coarse meshes) (40 z-levels)

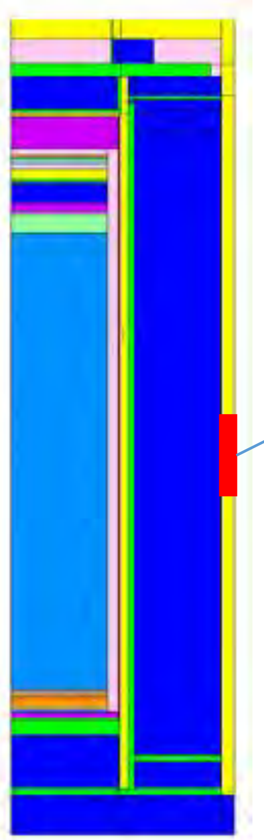
$P_{3,}$ S_{12} (168 directions)

1.48 GB per processor; 8 processors (~12 GB Total)

VT³G

VirginiaTech Invent the Future®

Accuracy and Performance of PENTRAN



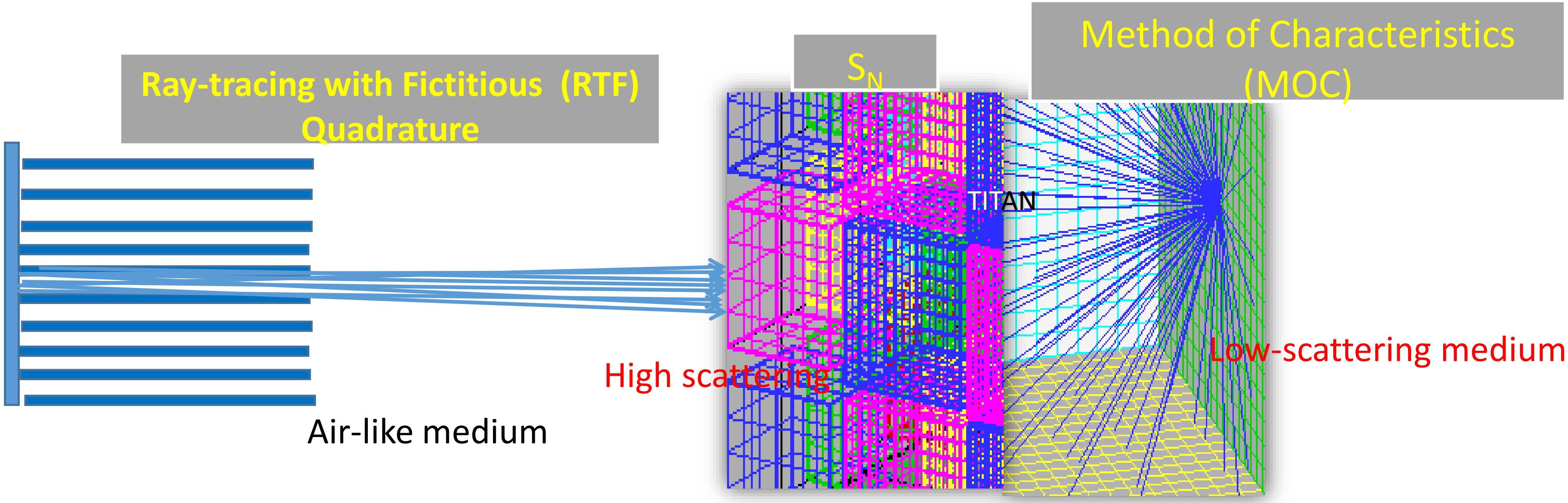
Reference Monte Carlo (Multigroup)	$\frac{PENTRAN}{Reference}$
1.25E-04 (1.30%)	1.04
1.50E-04 (1.18%)	1.04
1.39E-04 (1.23%)	1.04
1.00E-04 (1.46%)	1.04

Type equation here.

Model	# CPU	FOM	Run Time (hrs)	Values/ Hour	Speedup
Unbiased MCNP Multigroup*	8	0.46	362	19	-
PENTRAN 'Large' Model	8		165	42,100	2.2

Developed the TITAN hybrid Code system for coupling different forms of LBE
(C. Yi & A. Haghighat, 2005)

- Hybrid deterministic-deterministic: Allows for the use of different forms of LBE to be used in different coarse meshes of a problem
- Currently, it includes Sn, Method of Characteristics (MOC) & Ray-tracing with Fictitious quadrature set (RTF)

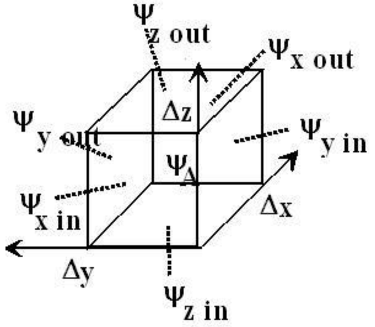


Ray-tracing with Fictitious (RTF) Quadrature
S_N
Method of Characteristics (MOC)
TITAN
Low-scattering medium
High scattering
Air-like medium

TITAN Sn-MOC Algorithm

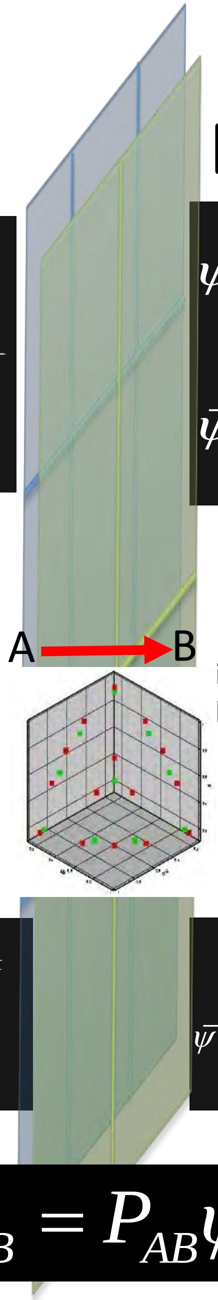
Sn

$$\psi_{gijk}^{(n)} = \frac{\frac{2\mu_n}{\Delta x}\psi_{x\,\mathrm{in}} + \frac{2\eta_n}{\Delta y}\psi_{\mathrm{y\,in}} + \frac{2\xi_n}{\Delta z}\psi_{\mathrm{z\,in}} + Q_{gijk}^{(n)}}{\sigma_{ijk} + \frac{2\mu_n}{\Delta x} + \frac{2\eta_n}{\Delta y} + \frac{2\xi_n}{\Delta z}}$$



E.g., LD Scheme;

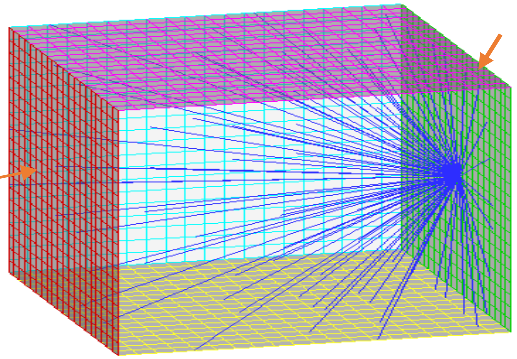
$$\psi_{\mathrm{x\,out}} = 2\psi_{\mathrm{x\,in}} - \psi_{gijk}^{(n)}$$
$$\psi_{\mathrm{y\,out}} = 2\psi_{\mathrm{y\,in}} - \psi_{gijk}^{(n)}$$
$$\psi_{\mathrm{z\,out}} = 2\psi_{\mathrm{z\,in}} - \psi_{gijk}^{(n)}$$



Block-oriented MOC

$$\psi_{gink}^{out} = \psi_{gink}^{in} e^{-\sigma_{gi}s_{ink}} + \frac{Q_{gin}}{\sigma_{gi}}(1 - e^{-\sigma_{gi}s_{ink}})$$

$$\bar{\psi}_{gink} = \frac{Q_{gin}}{\sigma_{gi}} + \frac{\psi_{gink}^{in} - \psi_{gink}^{out}}{s_{ink}\sigma_{gi}}$$



Outgoing boundary

incoming boundary

$$\bar{\psi}_{gin} = \frac{\sum_k \bar{\psi}_{gink}(\delta A_{ink}s_{ink})}{\sum_k (\delta A_{ink}s_{ink})} = \frac{Q_{gin}}{\sigma_{gi}} + \frac{1}{\sigma_{gi}}\frac{\sum_k \delta A_{ink}\Delta_{gink}}{\sum_k (\delta A_{ink}s_{ink})}$$

$$\psi_B = P_{AB}\psi_A$$

TITAN Benchmarking and Application

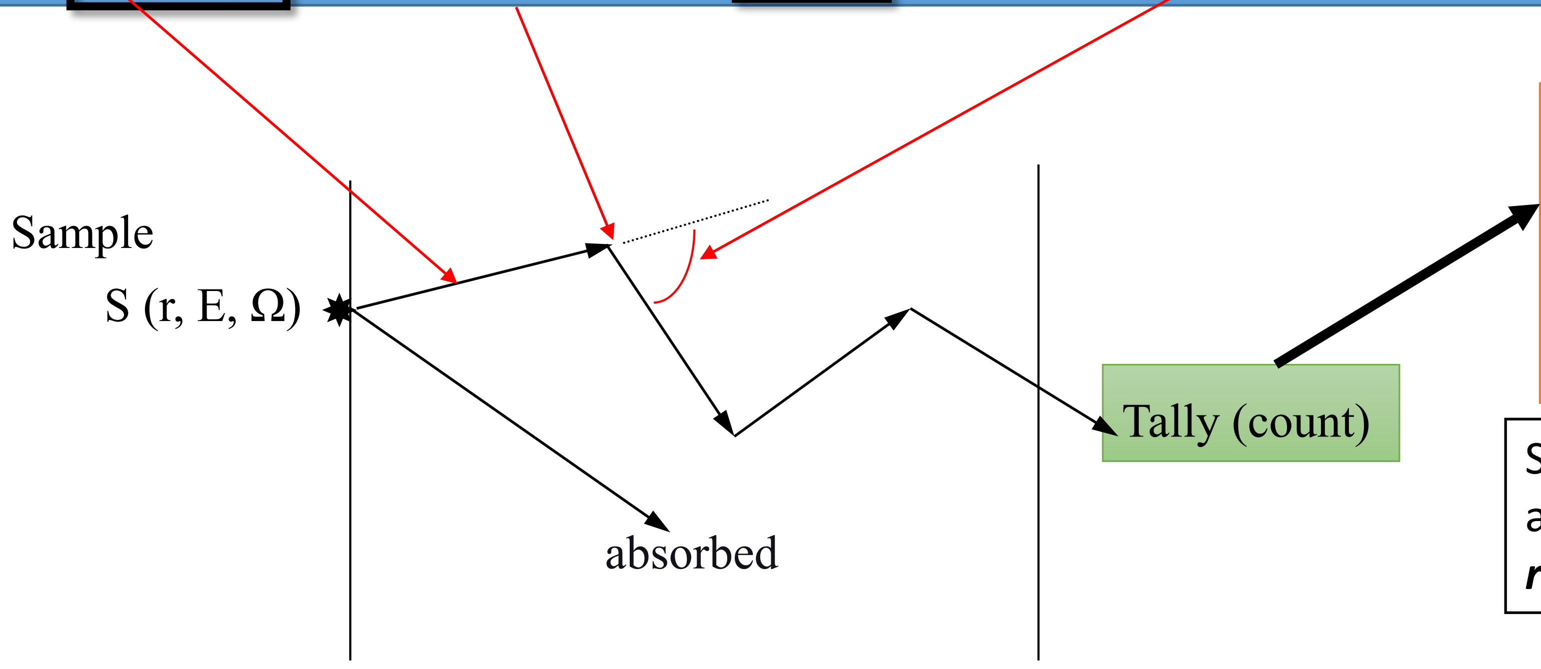
- OECD/NEA Benchmarks
  - C5G7 MOX
  - Kobayashi
  - 3-D parameter space
  - VENUS-2

- Applications
  - Adjoint calculation for the AIMS active interrogation simulation tool
  - *mPower* reactor core and external modeling
  - Modeling of a penetration duct in a nuclear reactor
  - Benchmarking the multigroup SDM (subgroup decomposition method) algorithm (developed by Georgia Tech)
  - Medical applications (e.g., CT and SPECT)

Monte Carlo Method

- Monte Carlo method can be considered as a method for performing a particle transport experiment on a computer.
- By relating the random variables associated with an event to random numbers (ξ), random sampling is performed, e.g., for simplified particle transport, there 3 events:

1) Path-length $\boxed{r = \frac{-\ln \xi}{\Sigma_t}}$ 2) Type of interaction, $\boxed{\xi \le \frac{\Sigma_s}{\Sigma_t}}$ 3) Scattering angle (isotropic scattering) $\boxed{\mu_0 = 2\xi - 1}$



$$\bar{C} \pm S_{\bar{c}}$$

and

$$R_{\bar{c}} = \frac{S_{\bar{c}}}{\bar{C}} = \frac{S_c}{\bar{C}\sqrt{N}}$$

Significant CPU is needed to achieve a ***statistically reliable*** results

Remarks - Monte Carlo methods

- In a Monte Carlo simulation, there **is no need for discretization** of angle, space , and energy

- To estimate a reliable count, it is necessary to ***discretize the independent variables***

- **To estimate a statistically reliable count, significant number of particle histories have to be simulated**

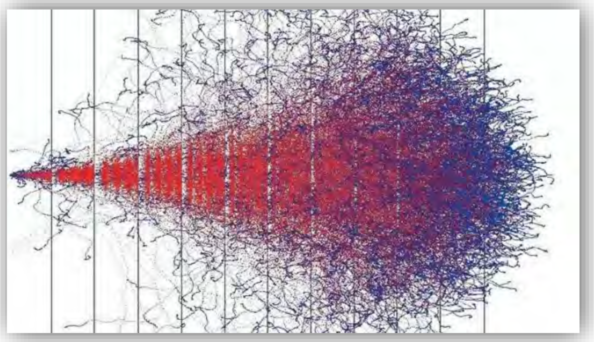
- Since particle histories are independent, **parallel computing** can be very effective

- But, to achieve detail solutions, it is necessary to employ effective and automated **variance reduction (VR) techniques**

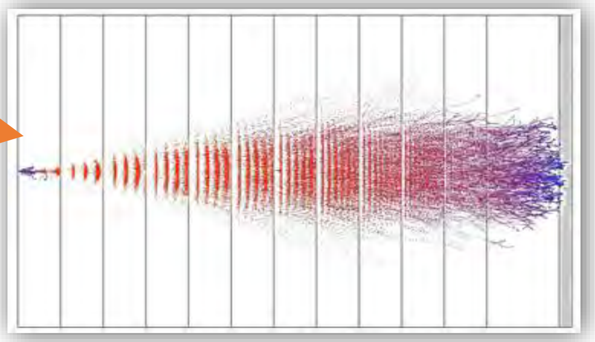
- To devise an effective VR, it is necessary to determine the **detector importance function** that provides information on the **importance of a particle to a response/objective**.

- The **importance function** can be obtained form the equation of balance of importance of a neutron:

Electron transport from a linear accelerator to a patient skin; ADIES code used



With VR



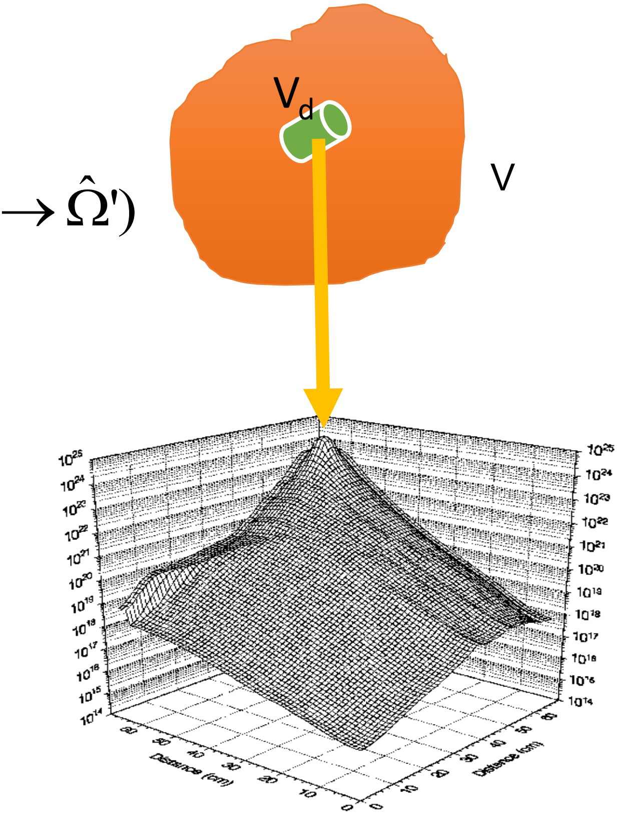
Particle Importance

Equation of balance of **Importance** of a particle to detector with cross-section (σ_d) is given below:

$$H^*\psi^* = \sigma_d \qquad \text{in V}$$

Where,

$$H^* = -\hat{\Omega}\cdot\nabla + \sigma_t(\vec{r},E) - \int_0^\infty dE' \int_{4\pi} d\Omega' \sigma_s(\vec{r},E \to E',\hat{\Omega}\to\hat{\Omega}')$$



By forming a commutation relation between the “forward” and importance equations with a vacuum boundary condition, the detector response is expressed by

$$R = \langle \psi^* q \rangle$$

a sample importance function distribution (ψ^*) around a detector

NOTE:

- Above equation indicates at that $\psi^* q$ represent the contribution of a source within a phase space to the detector;
- Why not use the importance function in the Monte Carlo VR?

Hybrid Method - CADIS Methodology & A^{3}MCNP

(J . Wagner & A. Haghighat, 1997)

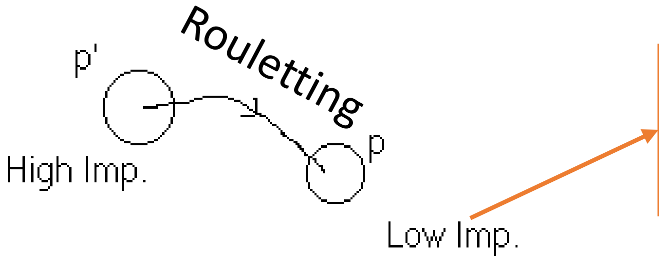
We developed the CADIS (consistent Adjoint Driven Importance Sampling) methodology :

CADIS uses *a **3-D S$_N$ importance function** distribution for*

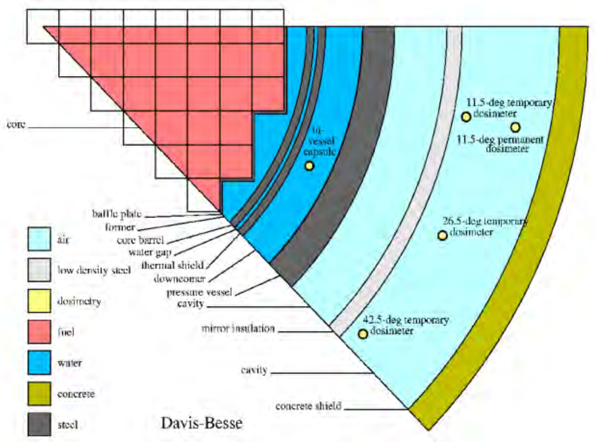
- *Source biasing*

$$\boxed{\hat{q}(p) = \frac{\psi^*(p)q(p)}{R}}$$

- *Transport biasing with splitting/rouletting rechnique*



$$\boxed{w(p) = w(p')\frac{\psi^*(p')}{\psi^*(p)}}$$



- *consistent manner, within the weight-window technique* **.**

The CADIS methodology was implemented into MCNP, to develop A^{3}MCNP (Automated Adjoint Accelerated MCNP) code system

Application of A^3MCNP

- PWR Cavity dosimetry

  For determination of neutron interaction rates with dosimetry materials placed at the reactor cavity, and estimation of fluence at the reactor pressure vessel

- BWR Core Shroud

  Determination of neutron and gamma fields at the reactor pressure vessel

- Storage cask

  Determination of neutron and gamma fields at the cask's outside surface

Significant speedups are obtained for obtaining highly reliable counts at a few regions, e.g.,

Problem	Speedup
PWR cavity dosimetry	50000
BWR core shroud	2000
Cask surface dose	140

My Journey - Particle Transport Algorithms Development

How about Real Time?

Year	Methodology	Computer Code	Wall-clock Time	Former & Current Students
2007	Hybrid MC-Dtrm (electon)	ADIES	Months Days 100's 1000's Cores	Dr. B. Dionne
2005	Hybrid Dtrm-Dtrm	TITAN		Drs. C. Yi & Walters
1997	Hybrid MC-Dtrm	A^{3}MCNP		Drs J. Wagner & Shedlock
1996	3-D Parallel	PENTRAN		Drs. Sjoden & Kucukboyaci
1992	2-D Vector & Parallel Dtrm		Years Months	Drs. M. Hunter, R. Mattis & B. Petrovic
1989	1-D Parallel Dtrm		A few cores	
1986	1-D vector Dtrm			

VT³G

VirginiaTech Invent the Future®

New Paradigm – Multi-stage, Response-function Transport (MRT) Methodology

(A. Haghighat, K. Royston & W. Walters, 20014)

- ***MRT Methodology****
  - Based on problem physics partition a problem into **stages** (sub-problems),
  - For each stage employ response method and/or adjoint function methodology
  - Pre-calculate response-function or adjoint-function using an accurate and fast transport code
  - Solve a linear system of equations to couple all the stages in *real-time*

*Haghighat, A., K. Royston, and W. Walters, “MRT Methodologies for Real-Time Simulation of Nonproliferation and Safeguards Problems,” Annals of Nuclear Energy, pp. 61-67, 2015.

Application of the *MRT Methodology*

- **Nondestructive testing:** Optimization of the Westinghouse's PGNNA active interrogation system for detection of RCRA (Resource Conversation and Recovery Act) (e.g., lead, mercury, cadmium) in waste drums (partial implementation of MRT; 1999)[1]

- **Nuclear Safeguards:** Monitoring of spent fuel pools for detection of fuel diversion (funded by LLNL); Developed INSPCT-s code system (2007)[2]

- **Nuclear nonproliferation:** Active interrogation of cargo containers for simulation of special nuclear materials (SNMs) (2013) (in collaboration with GaTech); developed the AIMS (Active Interrogation for Monitoring Special-nuclear-materials) code system (2013)[3,4]

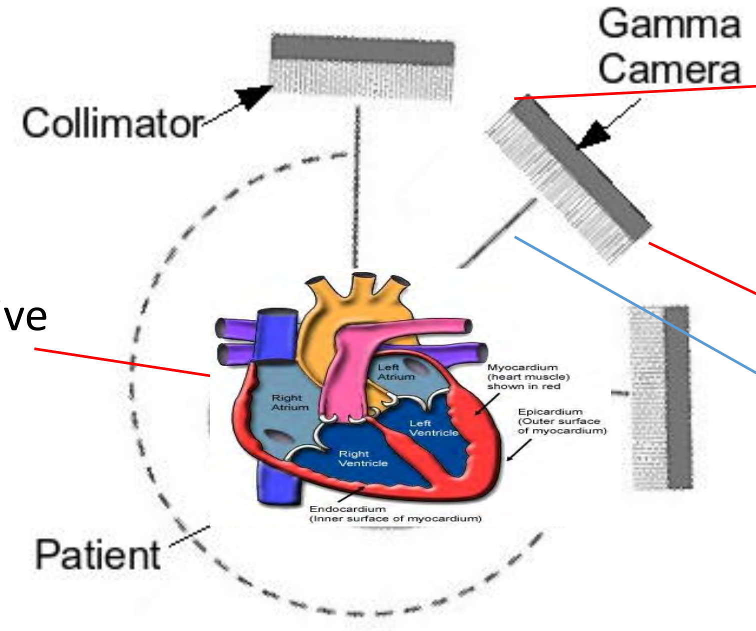
- **Image reconstruction for SPECT (Single Photon Emission Computed Tomography):** Real-time simulation of an SPECT device for generation of project images using an MRT methodology and Maximum Likelihood Estimation Maximization (MLEM); Developed the TITAN-IR code system (filed for a patent, June 2015)[5,6]

- Simulation, Monitoring, and safeguards of nuclear systems: developed the RAPID (Real-time Analysis for Particle-transport and In-situ Detection) code system (2014) & more recently a Virtual Reality System for RAPID, VRS-RAPID (Sept 2017) (filed for patents for both concepts, Oct 2017)[7-14]

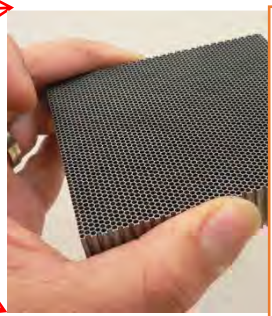
MRT - Simulation of SPECT (Single Photon Emission Computed Tomography) & Development of an Image Reconstruction Algorithm

SPECT

- Functional imaging modality



Inject a
radioactive
material



Typical dimensions include

- Hole diameter ~0.18 cm
- Septa thickness ~0.02 cm
- Length ~3.3 cm
- Acceptance Angle ~1.6°

A projection angle

Issue

- Limited image quality and spatial resolution

Goal

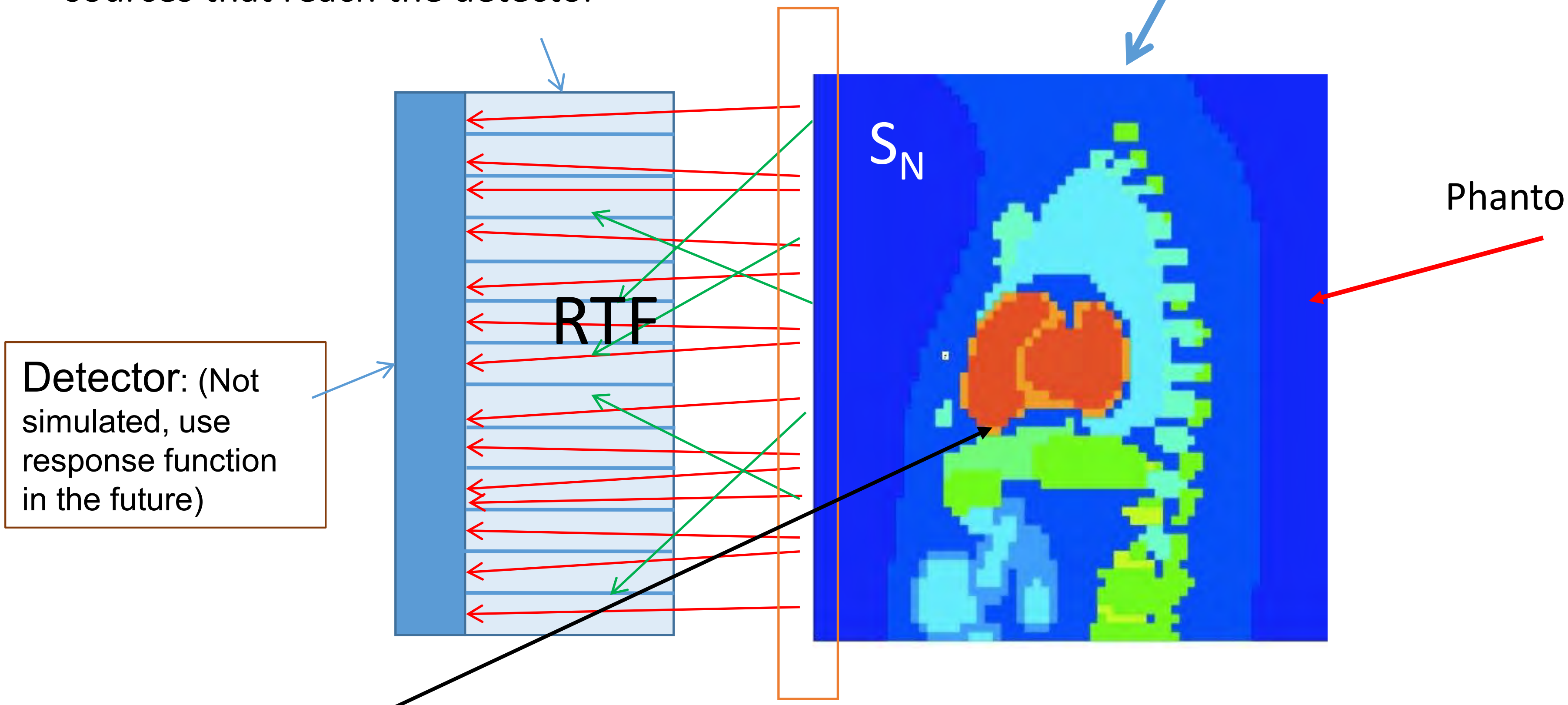
- Improving the image quality
- Reducing radioactive uptake

Hybrid formulation for SPECT simulation

- Used the hybrid Sn and RTF formulation

Stage 3: determination of count of sources that reach the detector

Stage 1 (simulated once)



Detector: (Not simulated, use response function in the future)

Stage 2 – Identify the intersection of rays emanating from the collimators with the phantom surface

In **Stage 4**, using ML-EM, one can obtain a new source

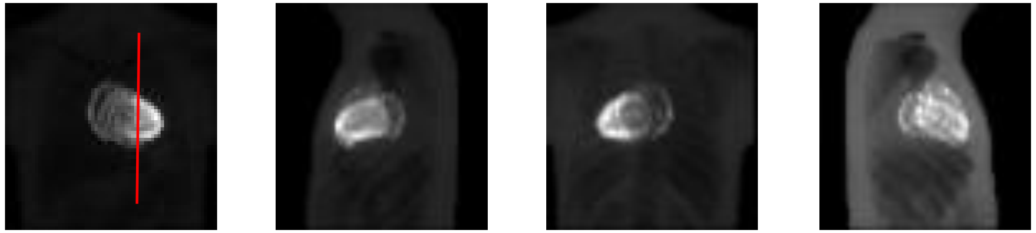
T

Benchmarking TITAN SPECT Projection Images

Comparison with the SIMIND reference code system using

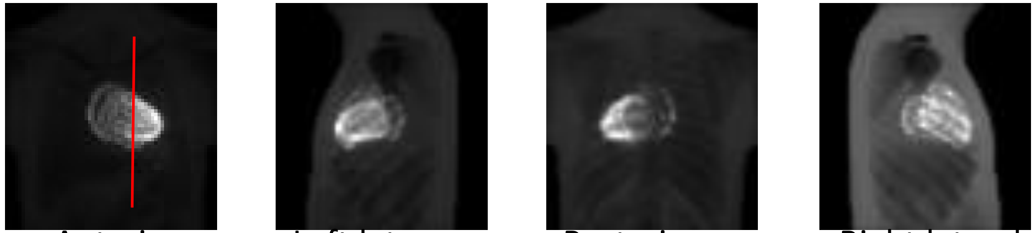
NURBS-based cardiac-torso (NCAT) phantom with Tc-99m (140 keV)

SIMIND generated projection images

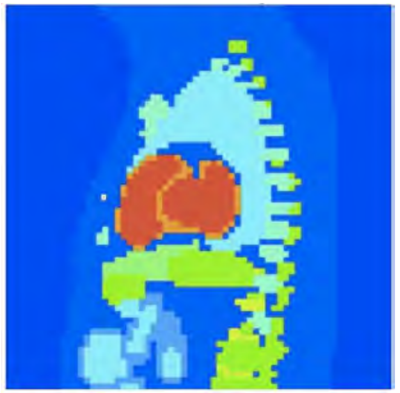


Anterior Left lateral Posterior Right lateral

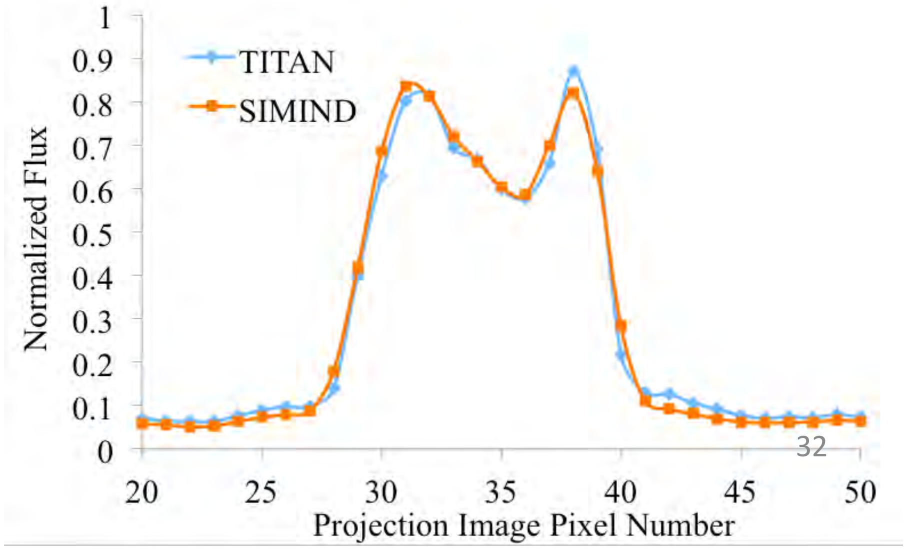
TITAN generated projection images



Anterior Left later Posterior Right lateral



Sagittal slice of NCAT phantom



TITAN SPECT Simulation Parallel Performance

Number of Processors	Number of projection images	Total time with serial projection images (sec)	Total time with parallel projection images (sec)
16	1	28.5	29.0
16	90	183.1	38.1
16	180	340.9	49.3

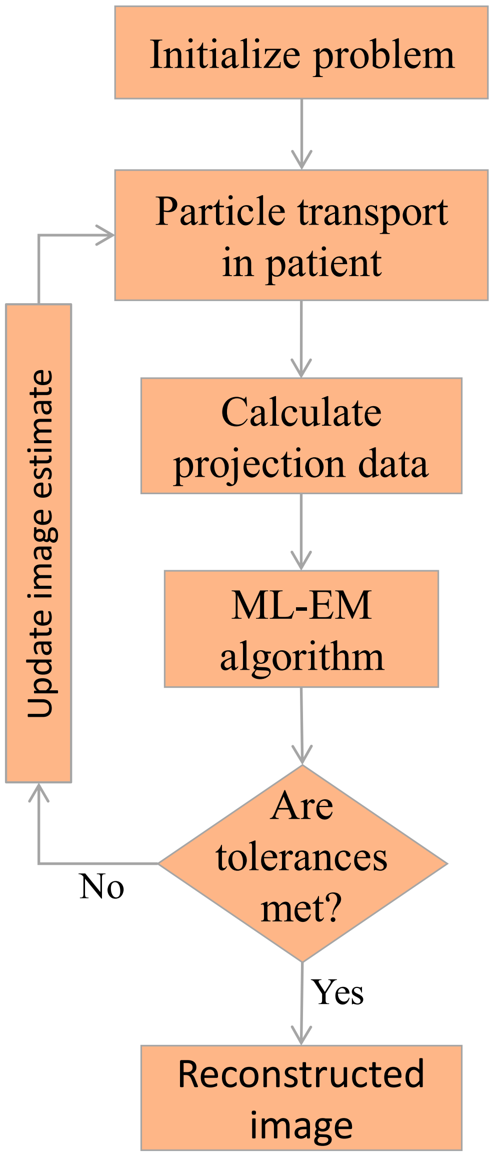
Royston *et al.*, *Progress in Nuclear Science and Technology*, 2, 2011

Iterative Deterministic Image-Reconstruction for SPECT (DRS)

- Projection data calculated by deterministic transport code
- Particle transport fully modeled in patient for forward projection
- Detailed system matrix never needs to be created
- Backprojection uses simple system matrix

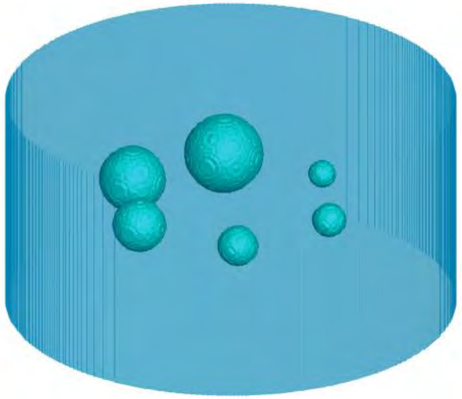
$$\hat{\lambda}_b^{(i+1)} = \frac{\hat{\lambda}_b^{(i)}}{\sum_{d=1}^{D} p_{b,d}} \sum_{d=1}^{D} \frac{n_d^*}{\hat{n}_d^{(i)}} p_{b,d}, \quad b = 1, \ldots, B$$

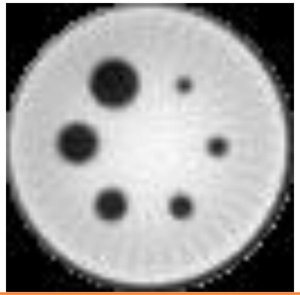
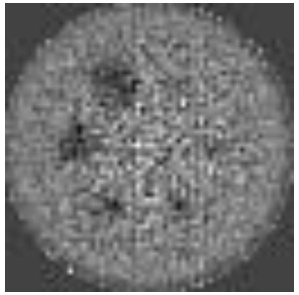
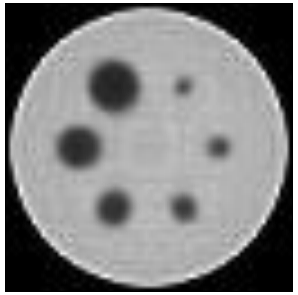
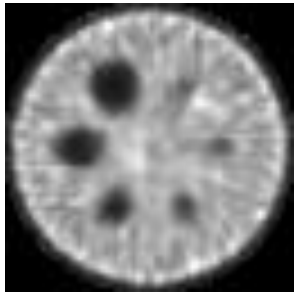
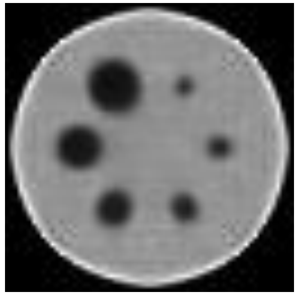
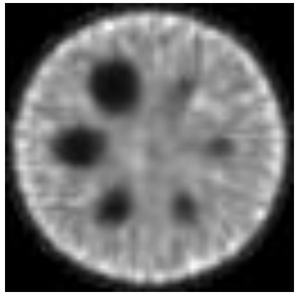
- A script was developed to allow anyone to use this method with any code that creates projection data for a given source distribution
- Implemented into the TITAN-IR



Example - TITAN-IR with *Fast Back Projection & Iterative ML-EM*

Jaszczak Cold Sphere Phantom



Algorithm	Noiseless, no collimator blur	Noisy, with blur (SE-LEHR)
FBP		
ML-EM with SM only		
TITAN-IR		

RAPID Formulation

- It is based on the MRT approach
  - Based on the problem physics and objective, i.e., RAPID writes the LBE differently for eigenvalue, subcritical, fixed-source problems, and detector response using response matrix/coefficient/function formulations
  - RAPID pre-calculates these response matrix/coefficient/functions by performing detailed continuous energy Monte Carlo simulation (using MCNP or Serpent):
    - For pre-calculation, RAPID uses the *p*RAPID module using a multi-layer approach (Filed for patent, application no. US 62/582,709)
  - RAPID solves the transport problem in real time using a linear system of equation

Eigenvalue Monte Carlo – Based on the Fission Matrix (FM)

- LBE formulation in operator form is expressed by

$$H\psi(\bar{p}) = \frac{1}{k}F\psi(\bar{p})$$

Where, $\bar{p} = (\bar{r}, E, \widehat{\Omega})\backslash$

$$H = \widehat{\Omega}\cdot\nabla + \sigma_t(\bar{r}, E) - \int_0^\infty dE' \int\limits_{4\pi} d\Omega'\, \sigma_s(\bar{r}, E' \rightarrow \mathrm{E}, \mu_0); \qquad F = \frac{\chi(E)}{4\pi}\int_0^\infty dE' \int\limits_{4\pi} d\Omega'\, \nu\sigma_f(\bar{r}, E')$$

$$\psi(\bar{p}) = \frac{1}{k}H^{-1}F\psi(\bar{p})$$

$$\psi(\bar{p}) = \frac{1}{k}H^{-1}\chi\tilde{F}\psi(\bar{p})$$

Where, $\tilde{F} = \frac{1}{4\pi}\int_0^\infty dE' \int_{4\pi} d\Omega'\, \nu\sigma_f(\bar{r}, E')$, then multiply by $\tilde{F}$ *operator*

$$\left(\tilde{F}\psi(\bar{p})\right) = \frac{1}{k}\left(\tilde{F}H^{-1}\chi\right)\left(\tilde{F}\psi(\bar{p})\right)$$

$$S(\bar{p}) = \frac{1}{k}AS(\bar{p})$$

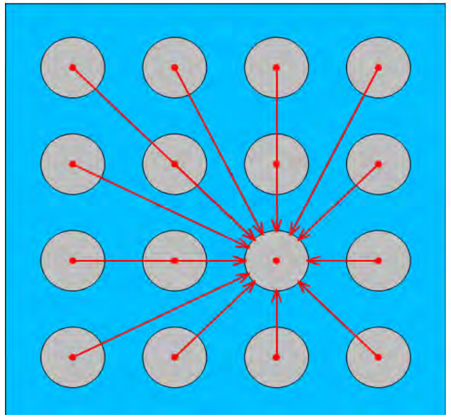
Where,

$$S = \tilde{F}\psi\,, \quad A = \tilde{F}H^{-1}\chi, \quad \& \quad \tilde{F} = \frac{1}{4\pi}\int_0^\infty dE' \int_{4\pi} d\Omega'\, \nu\sigma_f(\bar{r}, E')$$

Discretized FM formulation

- **Eigenvalue** formulation

$$F_i = \frac{1}{k}\sum_{j=1}^{N} a_{i,j}F_j$$



  - k is eigenvalue
  - F_j is fission source, S_j is fixed source in cell j
  - $a_{i,j}$ is the number of fission neutrons produced in cell i due to a fission neutron born in cell j.

- $a_{i,j}$'s are pre-calculated by a series of *fixed-source transport (e.g., Monte Carlo)* calculations
- Above equation is solved iteratively in a very short time
- NOTE: the *coefficient matrix is highly sparse and there is geometric similarity and symmetry (filed for a patent,* no. US 62/582,709*)*

Detector Response Function (DRF)

- *Standard detector response formulation*

$$R_i = \sum_j \sum_g \sigma_{j,g} \phi_{j,g} V_j$$

Where $\phi_{i,g}$ is obtained from by first LBE solved for flux in the cask

$$H\psi_C = F\psi_C + \mathrm{S}$$

Then, sum of fission source & independent source is used to solve for flux using

$$H\psi = S_f + S$$

- *Novel Detector Response Function (DRF) Methodology*

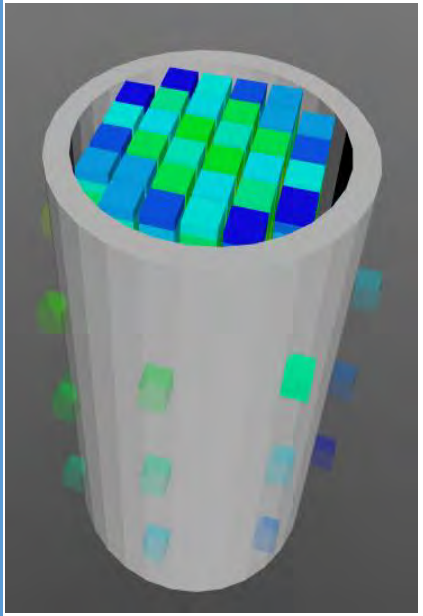
Response at detector i is calculated by

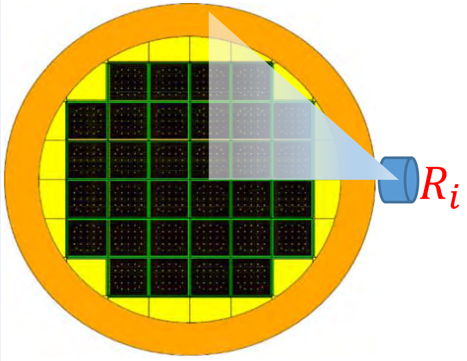
$$R_i = \sum_j \alpha_{ij} F_j + \beta_{ij} S_j$$

Where, $F_j$and S_j refer to fission and independent sources, respectively, and $\alpha_{ij}$and β_{ij}refer to DRF coefficients that have to **pre-calculated** ,

$$\alpha_{ij} = \frac{\sum_{V_i \epsilon V_d} \sum_g V_i \sigma_{i,g} \phi_{i,g}}{\sum_{V_j \epsilon V_{fis}} \sum_g V_j S_{i,g}^{fis}}$$

$$\beta_{ij} = \frac{\sum_{V_i \epsilon V_d} \sum_g V_i \sigma_{i,g} \phi_{i,g}}{\sum_{V_j \epsilon V_{fis}} \sum_g V_j S_{i,g}^{indp}}$$





*b*RAPID - Time-dependent (Fuel Burnup)

Standard Method

1. **Calculate Macroscopic Cross Sections:**

$$\Sigma(\underline{r}, E) = N\sigma(\underline{r}, E)$$

2. **Transport Calculation:** solve for multi-group, scalar flux for each mesh cell k

$$\phi_{g,k}$$

3. **Calculate One-group Cross-sections & Fluxes**:

$$\sigma_{ij,k} = \frac{\sum_g \sigma_{ij,g,k}\phi_{g,k}}{\sum_g \phi_{g,k}}$$

4. **Solve Depletion Equations:** for each burnable region k, update number densities

$$\frac{dN_j}{dt} = \sum_{i \neq j} \lambda_{ij} Ni - \lambda_j N_j \quad \text{for} \quad j = 1, ..., n.$$

5. **Repeat Steps 1-4 for all Burnup Steps**

1. **Pre-calculation:** to obtain FM coefficients and irradiated fuel properties (one time) Run *p*RAPID
2. **Define Burnup Fission Matrix**: via interpolation of database FM coefficients

$$a_{i,j}^b = \frac{(p - p_{ip-1})a_{i,j}(ip, t) + (p_{ip} - p)a_{i,j}(ip-1, t)}{p_{ip} - p_{ip-1}}$$

3. **Solve FM Eigenvalue Equations**: for fission source distribution

$$F_i^n = \frac{1}{k}\sum_j a_{i,j}^b F_j^n$$

F_i^n Fission Source for step *n*

k Eigenvalue

3. **Calculate Next-step Power Density Distribution**: per burnable region *i* (assembly-wise, axially-dependent)

$$p_i^{n+1} = (\bar{P}^{n+1} N_b) f_i^n$$

p^{n+1} Assembly-wise, axially-dependent power density at step *n+1*

$\bar{P}^{n+1}$ Average power density (user input)

N_b Number of burnable regions

f_i^n Normalized Fission Source

4. **Calculate Next-step Time:**
5. **Calculate Irradiated Fuel Properties:**
6. **Repeat Steps 2-5 for Each Burnup Step**

Interpolate from database

Standard Kinetic Transport Equations

- Time-dependent **flux** equation (no external source):

$$\frac{1}{v}\cdot\frac{\partial\psi(\vec{p},t)}{\partial t}+\sigma_t(\vec{p},t)\psi(\vec{p},t)+\widehat{\Omega}\cdot\nabla\psi(\vec{p},t)-\int dp'\sigma_s(\vec{p}'\to\vec{p})\psi(\vec{p}',t)=$$

$$(1-\beta)\frac{\chi_p(E)}{4\pi}\int dp'\nu\sigma_f(\vec{p}',t)\psi(\vec{p}',t)+\sum_{f=1}^{N_f}\frac{\chi_{d,f}(E)}{4\pi}\lambda_d C_d(\vec{r},t)$$

- Time-dependent **DNP concentration** equation for family f:

$$\frac{dC_{d,f}(\vec{r},t)}{dt}=-\lambda_{d,f}C_{d,f}(\vec{r},t)+\beta_f\int dp\,\nu\sigma_f(\vec{p}',t)\psi(\vec{p}',t)$$

*t*RAPID Time-dependent (Kinetics)
The TFM (Transient FM) matrices

- We define **4 fission matrices**, based on the type of the neutron that induces fission and of the one generated by fission:
  - $\boldsymbol{G_{pp}}$, is the fission matrix for *prompt neutrons* generated by a fission induced by a *prompt neutron*
  - $\boldsymbol{G_{pd}}$, is the fission matrix for *delayed neutrons* generated by a fission induced by a *prompt neutron*
  - $\boldsymbol{G_{dp}}$, is the fission matrix for *prompt neutrons* generated by a fission induced by a *delayed neutron*
  - $\boldsymbol{G_{dd}}$, is the fission matrix for *delayed neutrons* generated by a fission induced by a *delayed neutron*

tRAPID Formulation

- Similar to the steady-state FM, the TFM equations are obtained by **recasting the transport equation into matrix form.**

- The TFM equations (with 1 delayed neutron family), at time t_k, take the following form:

$$\boldsymbol{S}_{\boldsymbol{p}}^{(\boldsymbol{k})} = \sum_{k'=0}^{k} \bar{\bar{G}}^{(k-k')}{}_{pp} \cdot \boldsymbol{S}_{\boldsymbol{p}}^{(\boldsymbol{k'})} + \sum_{k'=0}^{k} \bar{\bar{G}}^{(k-k')}{}_{dp} \cdot \lambda_d \boldsymbol{C}_{\boldsymbol{d}}^{(\boldsymbol{k'})}$$

Time integral of prompt neutrons generated by prompt neutrons

Time integral of prompt neutrons generated by delayed neutrons

$$\left(\frac{d\boldsymbol{C}_{\boldsymbol{d}}}{dt}\right)^{(\boldsymbol{k})} = -\lambda_d \boldsymbol{C}_{\boldsymbol{d}}^{(\boldsymbol{k})} + \sum_{k'=0}^{k} \bar{\bar{G}}^{(k-k')}{}_{pd} \cdot \boldsymbol{S}_{\boldsymbol{p}}^{(\boldsymbol{k'})} + \sum_{k'=0}^{k} \bar{\bar{G}}^{(k-k')}{}_{pd} \cdot \lambda_d \boldsymbol{C}_{\boldsymbol{d}}^{(\boldsymbol{k'})}$$

DNP decay term

Time integral of DNP generated by prompt neutrons

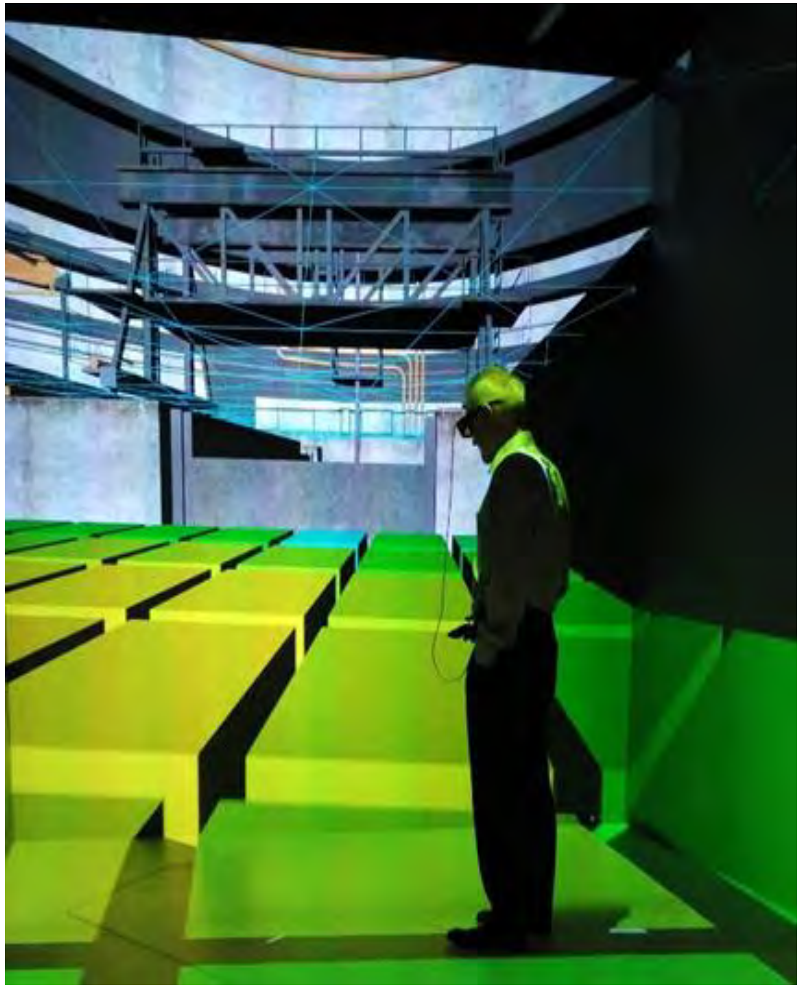
Time integral of DNP generated by delayed neutrons

VRS Rapid

Web Application

Web Application

- RAPID is incorporated into a *Web application*, referred to as the Virtual Reality System (VRS) for RAPID*.
- VRS-RAPID provides a collaborative Virtual Reality environment for a user to build models, perform simulation, and view 3-D diagrams in an interactive mode.
- 3-D diagrams can be projected onto a virtual system environment (e.g., a pool) for further analysis and training purposes.
- Additionally, VRS-RAPID outputs can be coupled with an immersive facility such as the VT's HyberCube System, as shown in this figure.



**Filed a disclosure application to the VT-IP Office, March 2, 2018*

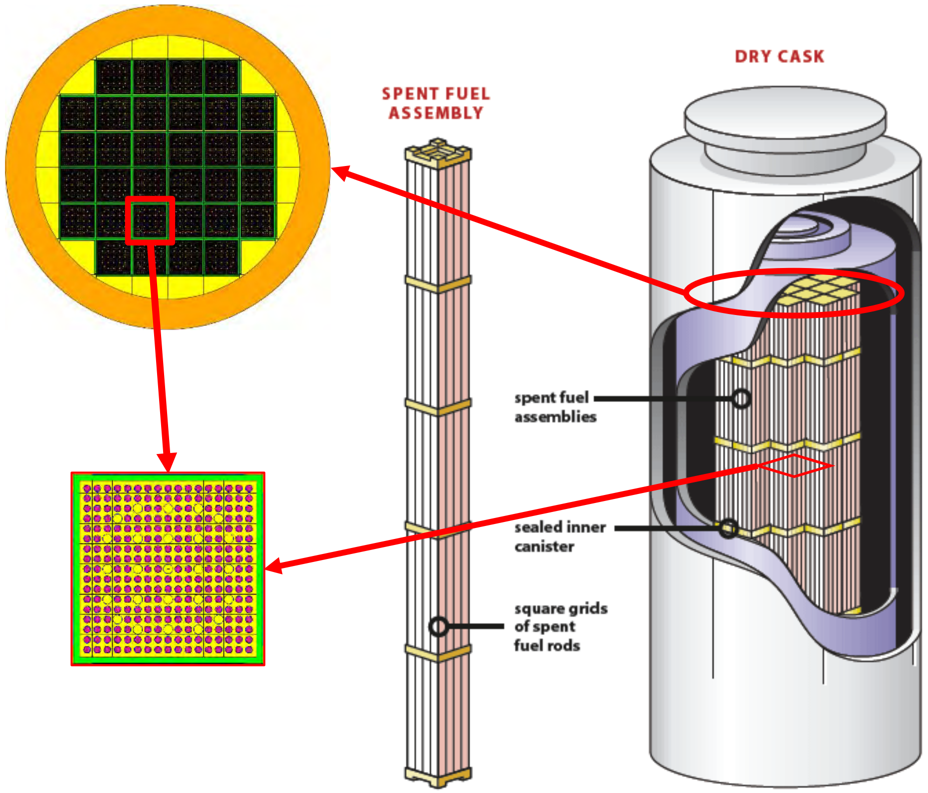
RAPID Benchmarking studies

Performed numerous computational benchmark and two experimental benchmark as follows:

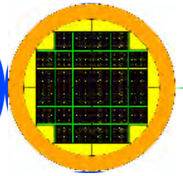
Spent fuel pool	**Different arrays from 2x2 to 9x9, assemblies of different burnups and cooling times, with axially-dependent burnups**
US Naval Academy subcritical reactor	Performed measurements both inside and outside the reactor, and performed detailed computational analyses
Spent fuel cask	GBC-32 benchmark & its variations (assemblies of different burnups & cooling times, with axial burnup distribution
Reactor core	Several large PWR problems, based on the NEA/OECD Monte Carlo Performance Benchmark Problem (considering different enrichment, different axial moderator temperature)
Detector Response for spent fuel pool/cask	Inspection of a pool (placement of a detector on top of the fuel) Inspection of a cask (placement of a detector on the surface of a cask)
3-D burnup for RAPID (bRAPID)	A smaller size (5x5) version of the OECD/NEA Monte Carlo Performance Benchmark; various combinations of specific power and irradiation times are studied
Time-dependent RAPID (tRAPID)	Performed detailed analyses using the Flattop experimental facility

Spent fuel cask
GBC-32 Cask Computational Benchmark

- **Geometry**
  - 32 Fuel assemblies
  - Stainless steel (SS304) cylindrical canister
  - Inter-assembly Boral absorber panels
  - Height of the canister: 470.76 cm
- **Fuel assembly**
  - 17x17 Optimized Fuel Assembly (OFA)
  - 25 instrumentation guides
  - Fresh UO_2 4% wt. enriched fuel pins
  - Active height: 365.76 cm
- **Benchmarked** against the MCNP & Serpent code systems
- **Pre-calculations**
  - Similar to the Pool calculations, except for a 17x17 array of pins, i.e., requiring 819 calculations, each 20 min in parallel (with 56 cores, about 5 hours)



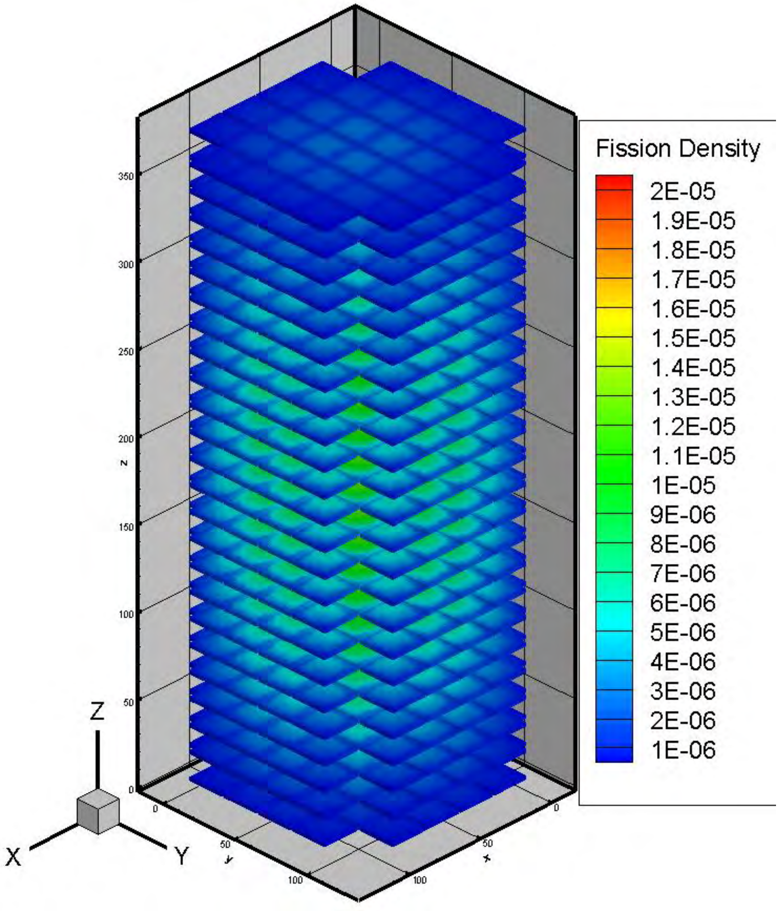
RAPID vs. MCNP – FULL Cask model



#assemblies = 32; # pins = 264; #axial levels = 24; # tallies = **202,752**

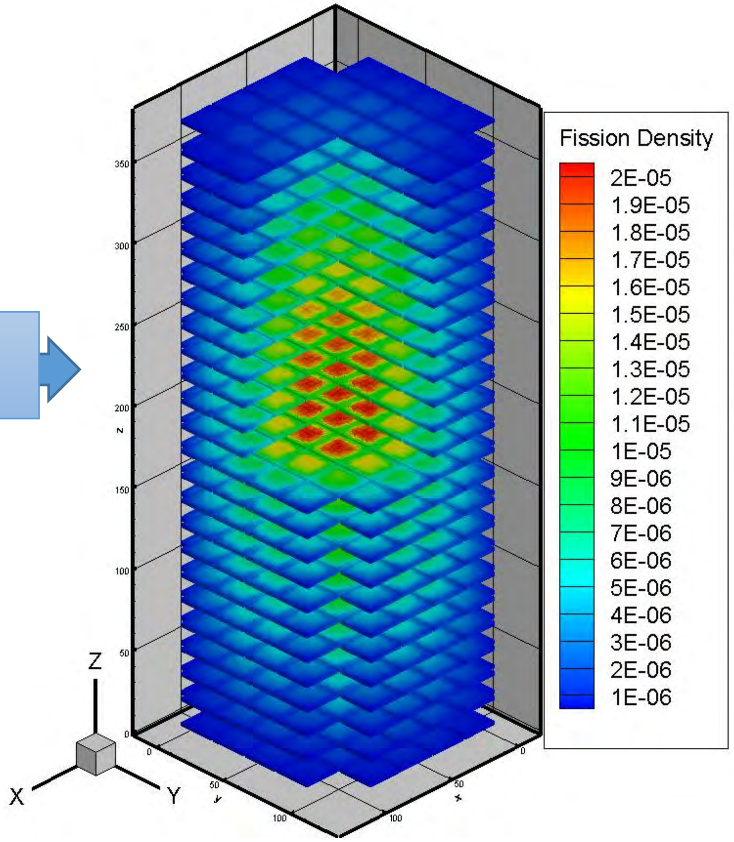
Case	MCNP	RAPID
k_{eff}	1.14545 (± 1 pcm)	1.14590
Relative Difference	-	39 pcm
Fission density rel. uncertainty	1.15%	-
Fission density relative diff.	-	1.56%
Computer	16 cores	1 core
Time	13,767 min (9.5 days)	0.585 min (35 seconds)
Speedup	-	23,533

Calculated Axially-Dependent, Pin-wise Fission Density in the GBC-32 Benchmark Using RAPID



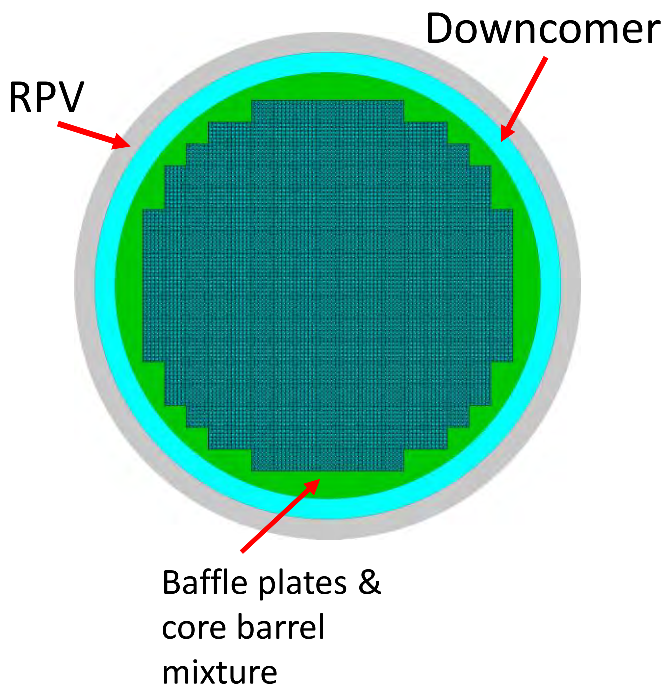
Inside view

With a quarter Blanked



Reactor Core - RAPID vs. SERPENT – PWR*

- RAPID has been applied to several large PWR problems, based on the NEA/OECD Monte Carlo Performance Benchmark Problem.
    - 241 assemblies; 17x17 lattice; 264 fuel pins + 25 control/instrumentation tubes per assembly; 24 GWd/MT+ chemical shim
    - 100 axial levels
    - 6.4 million cells



- FM coefficients are pre-calculated using the SERPENT Monte Carlo code for different core configurations

*W. Walters, N. Roskoff, and A. Haghighat, "The RAPID Fission Matrix Approach to Reactor Core Criticality Calculations," Accepted for publication in the Journal Nuclear Science and Engineering (June, 2018)

Sample Result
RAPID vs. SERPENT – Keff & Compuation Time

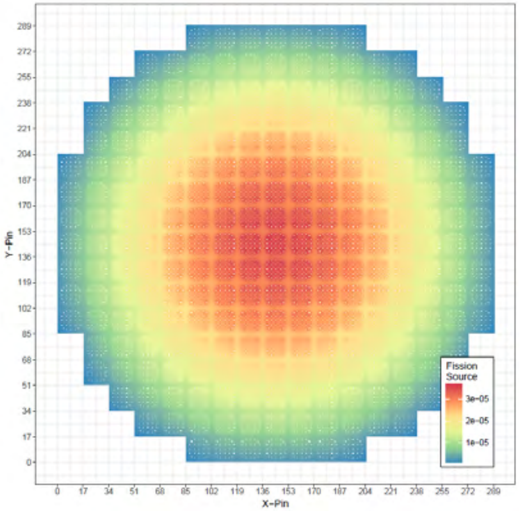
Method	Keff	Relative Diff
Serpent	$1.000855 \pm 1.0\,pcm$	-
RAPID	$1.000912 \pm 1.4\,pcm$	5.3 pcm

Method	# Cores	CPU (hrs)	Speedup
Serpent	20	1000	-
RAPID*	1	0.23	4348

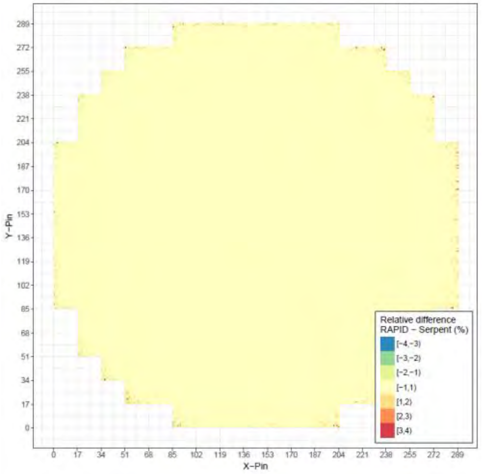
*Pre-calculation requires about 700 CPU hrs

Sample Result
RAPID vs. SERPENT –Fission Density

Pin-wise Fission Source



RAPID-Serpent Difference



Pin-wise fission source

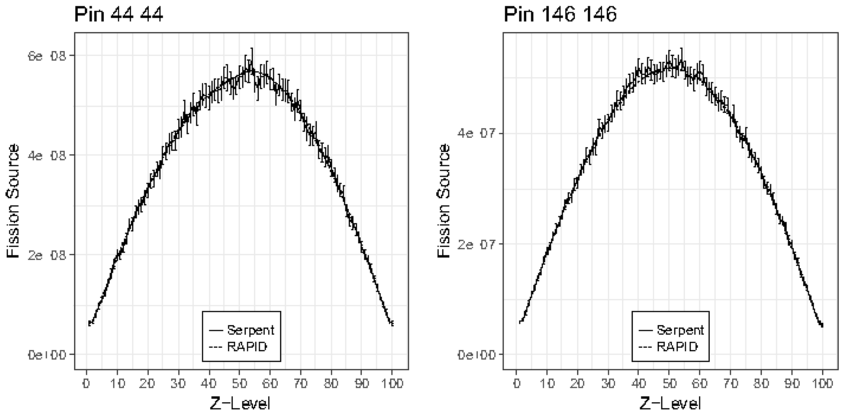
RMS error : 2.18%

1-σ Uncertainty

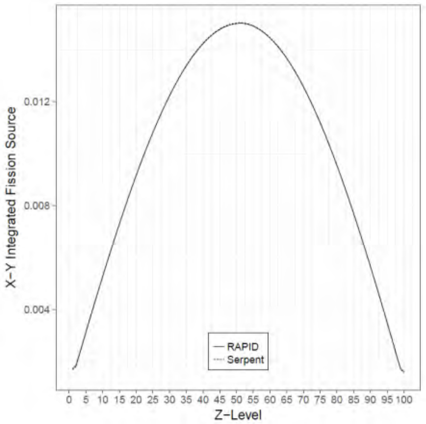
Type	Serpent	RAPID
RMS	2.03%	0.21%
Max	10.16%*	3.64%

*has not converged!

Fission density along single pins



Axial Fission density



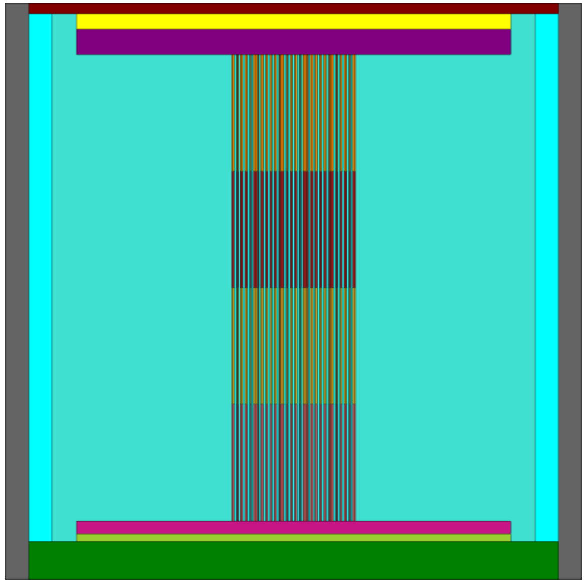
VT³G

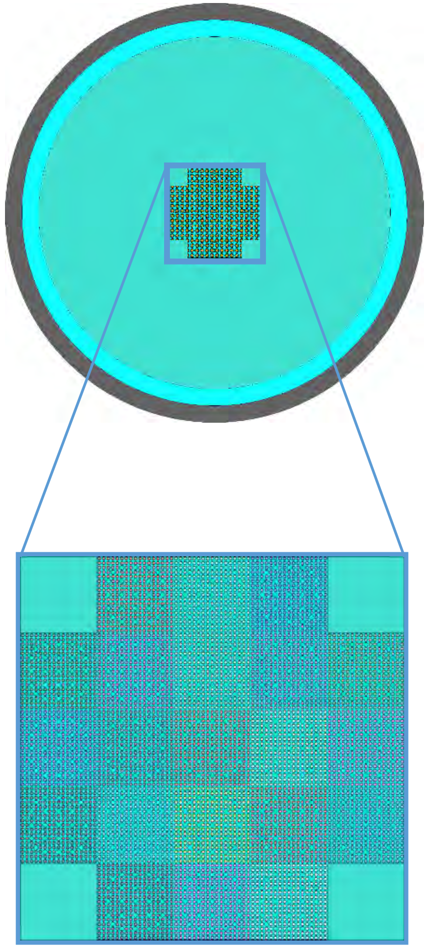
VirginiaTech
Invent the Future®

A version of OECD/NEA Monte Carlo Performance Benchmark with 25 Assemblies, 17x17 fuel lattice

- 5x5 Mini Core w/ 4 axial zones
  - 84 burnable regions

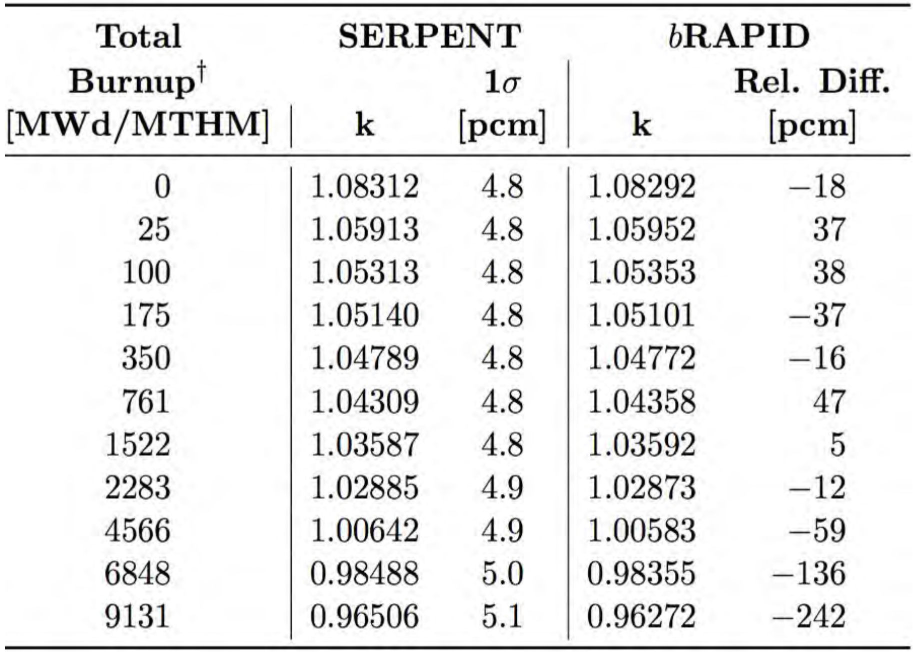
- Burn this model with:
  - Power density = 25 kW/kg
  - Irradiation Time = 1 year (10 steps)
  - Total Burnup = 9,131 MWd/MTHM





Comparison of RAPID and Serpent fuel burnup calculational results

Eigenvalue

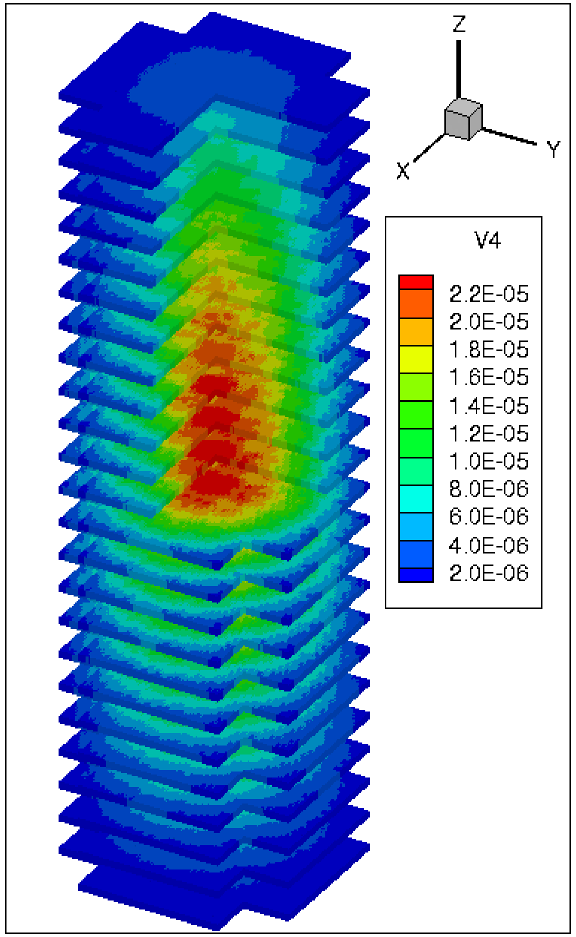


Total Burnup† [MWd/MTHM]	SERPENT k	SERPENT 1σ [pcm]	*b*RAPID k	*b*RAPID Rel. Diff. [pcm]
0	1.08312	4.8	1.08292	−18
25	1.05913	4.8	1.05952	37
100	1.05313	4.8	1.05353	38
175	1.05140	4.8	1.05101	−37
350	1.04789	4.8	1.04772	−16
761	1.04309	4.8	1.04358	47
1522	1.03587	4.8	1.03592	5
2283	1.02885	4.9	1.02873	−12
4566	1.00642	4.9	1.00583	−59
6848	0.98488	5.0	0.98355	−136
9131	0.96506	5.1	0.96272	−242

Time & Computers

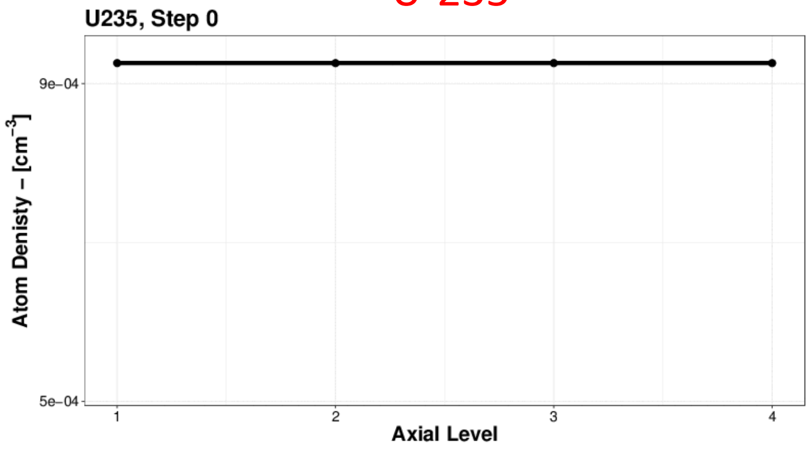
Method	No. Proc.	Time	Speedup
Serpent	32	4.3 d	--
*b*RAPID	1	21 min	294

Fission neutron variations

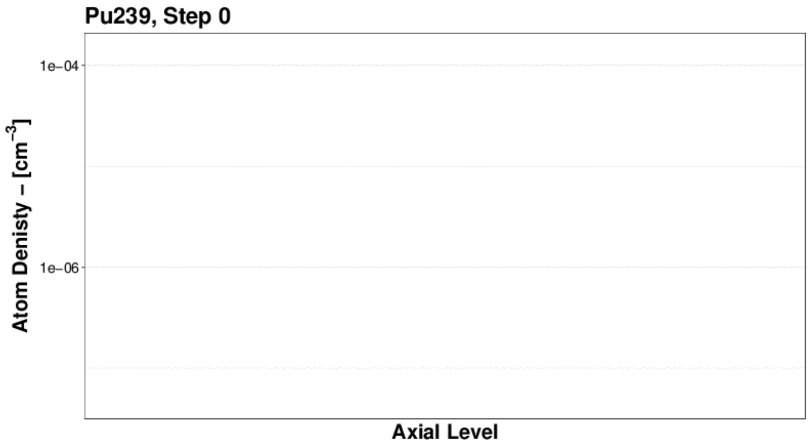


Axial fuel isotopic burnup

U-235

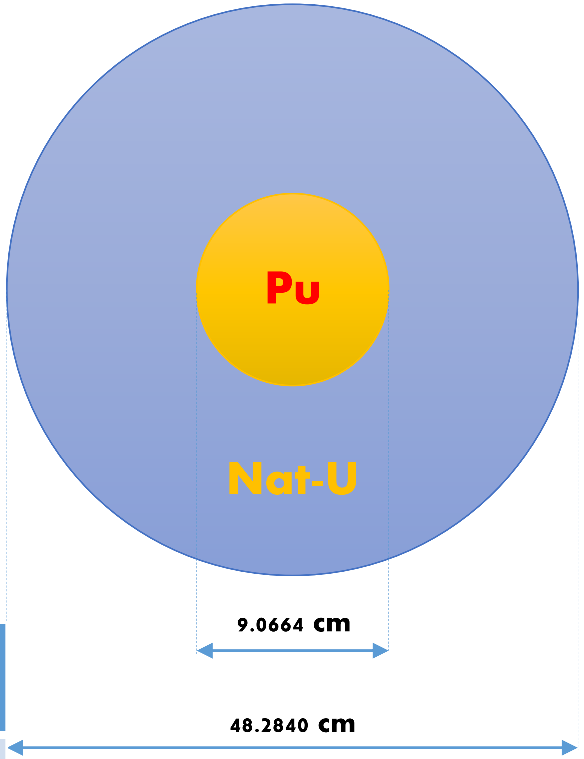


Pu239



The ”Flattop” criticality benchmark

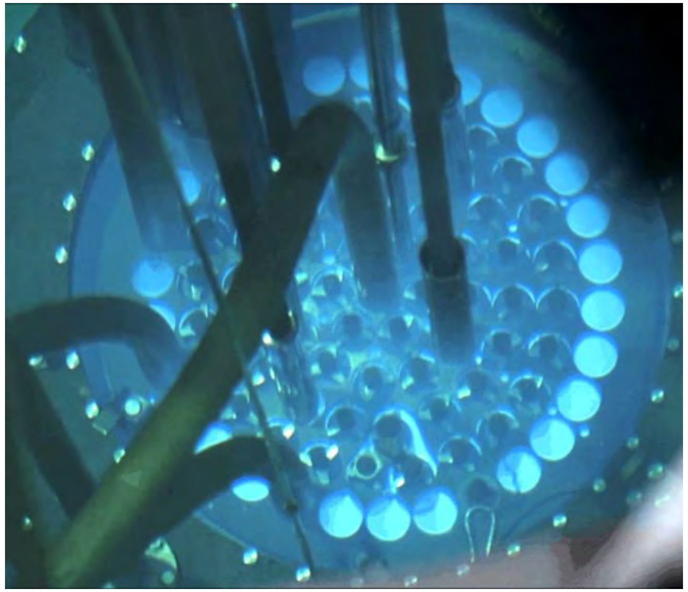
- NEA/OECD Criticality Benchmark: PU-MET-FAST-006
- Plutonium sphere surrounded by natural uranium reflector
- **Fast** reflected critical reactor



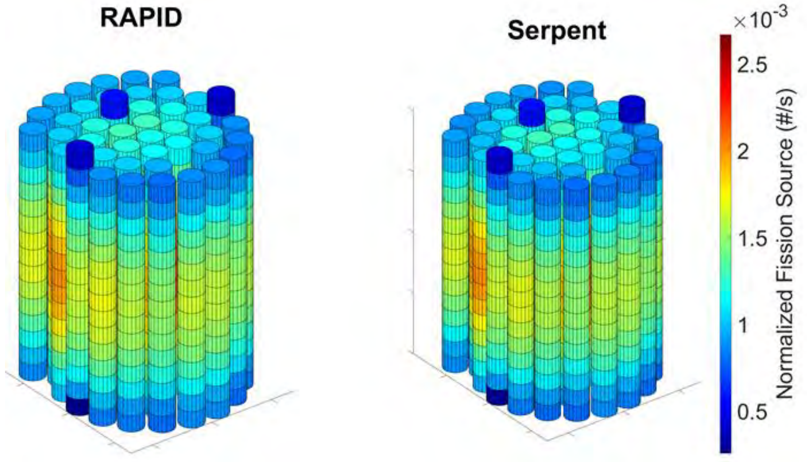
Comparison of k-effective and α_{Rossi} (RAPID vs. Experiment)

Case	k_{eff}	β_{eff}	l_{eff}	Λ_{eff}	α_{Rossi}
*t*RAPID	$0.99907 \pm 5\ pcm$	0.00278	$13.18\ ns$	$13.23\ ns$	$-0.210\ \mu s^{-1}$
Experiment	$1.0000 \pm 300 pcm$	/	/	/	$-0.214\ \mu s^{-1}$
Relative Difference	$-93\ pcm$	/	/	/	1.9%

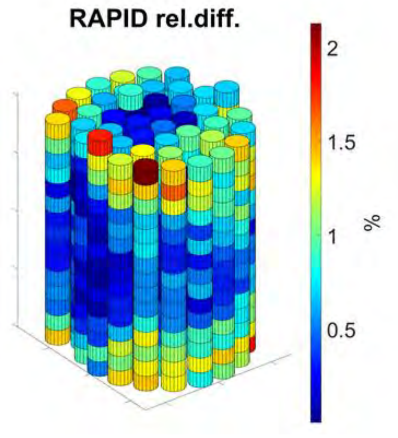
Jozef Stefan Institute's TRIGA



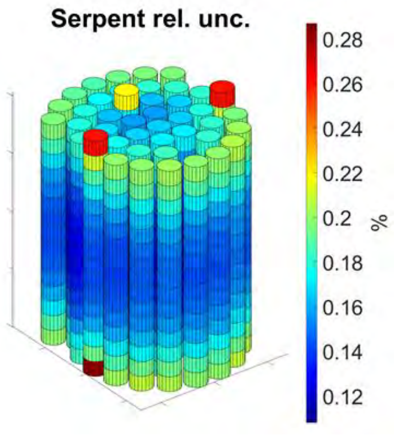
Pin-wise Fission Neutrons



Relative Difference RAPID vs. Serpent



Relative Uncertainty of the Serpent Results



Model	k_{eff}	**Rel. Diff. (pcm)**
Reference (Experimental)	$1.00460\ (560\ pcm)$	-
Serpent	$1.00908\ (6\ pcm)$	$446\ pcm$
RAPID	$1.00668\ (2\ pcm)$	$207\ pcm$

Model	# Computer Cores	Wall-clock time (s)	Speedup
Serpent	32	12730	-
RAPID	1	6	2120

My Journey - Particle Transport Algorithms Development

Year	Methodology	Computer Code	Wall-clock Time	Former & Current Students
2018	MRT	*t*RAPID	Smart phone 1 Core Minutes Seconds	V. Mascolino
2018	MRT	*b*RAPID		Dr. N. Roskoff
2017	VRS for RAPID	*VRS*-RAPID		V. Mascolino
2016	MRT	RAPID		Drs. Walters & Roskoff
2015	MRT	TITAN-IR		Dr. K. Royston
2013	MRT	AIMS		Drs. Royston &Walters
2009	MRT	INSPCT-S		Dr. W. Walters
2007	Hybrid MC-Dtrm (electon)	ADIES	Months Days 100’s 1000’s Cores	Dr. B. Dionne
2005	Hybrid Dtrm-Dtrm	TITAN		Drs. C. Yi & Walters
1997	Hybrid MC-Dtrm	A^3MCNP		Drs J. Wagner & Shedlock
1996	3-D Parallel	PENTRAN		Drs. Sjoden & Kucukboyaci
1992	2-D Vector & Parallel Dtrm		Years Months	Drs. M. Hunter, R. Mattis & B. Petrovic
1989	1-D Parallel Dtrm		A few cores	
1986	1-D vector Dtrm			

Thanks!

Questions?