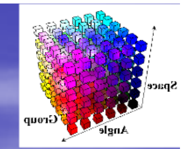
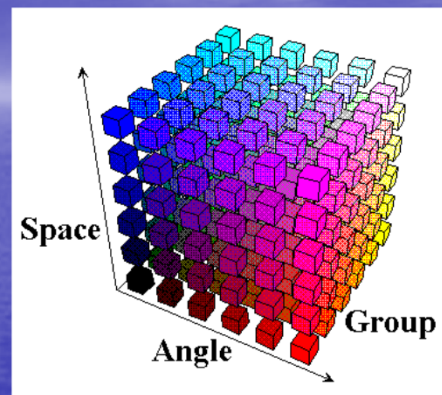




PENTRAN-CRT Code Workshop

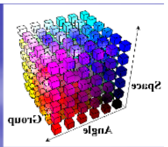
Colorado School of Mines

Golden, Colorado, March 2014

Georgia Tech

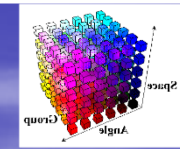
Nuclear and Radiological Engineering Program
George W. Woodruff School of Mechanical Engineering

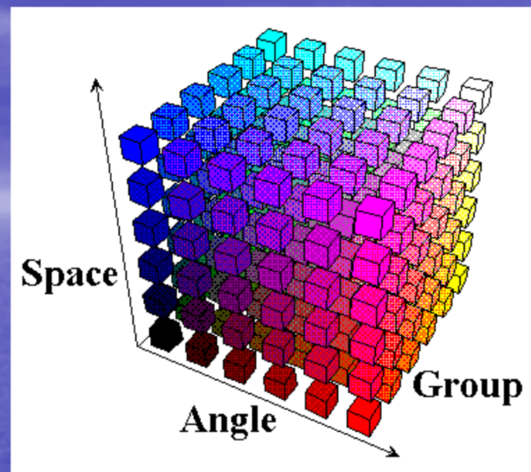
HSW TECH



Overview

- Code Overview – *G. Sjoden*
    - Features, Adaptive Differencing, Parallel Issues
- Code Examples and Applications –
- Large problems, Hybrid Methods, etc
    - Parallel Performance
- Problem examples
    - Parallel Scripts, PENDATA gathering
    - Output Details and analysis tools
- PENMSH
- Problem and Mesh Generation Tools
- Exercises and Exploration





PENTRAN Code Overview

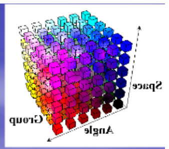
Presented by

Glenn E. Sjoden, Ph.D., P.E.

Georgia Tech

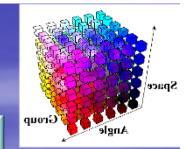
Nuclear and Radiological Engineering Program
George W. Woodruff School of Mechanical Engineering

HSW TECH



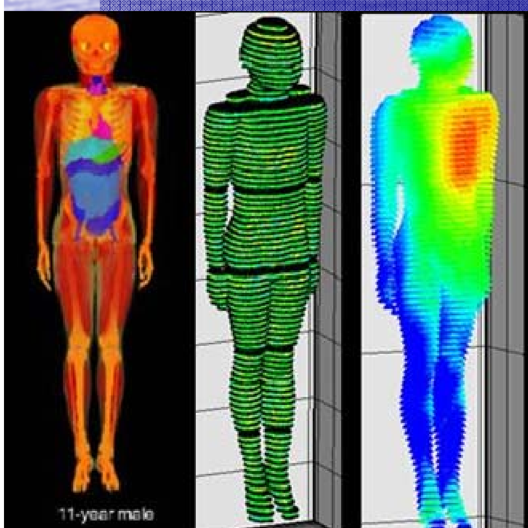
PENTRAN Code Update

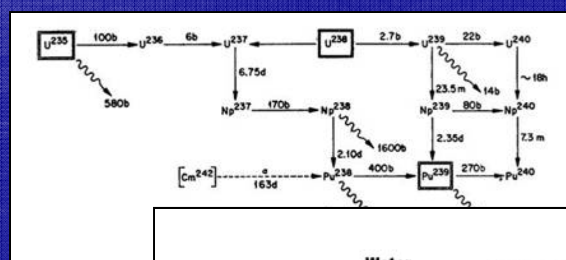
- To be available Free to Universities via RSICC at ORNL
  - http://www-rsicc.ornl.gov/
  - US Citizens and US Residents
- PENTRAN runs on ALL types of distributed parallel computers
  - Parallel decomposition in Angle, Energy, and Space
  - ANSI F90 standard code
  - Tested on up to 4096 Cores with excellent scaling
  - Has undergone significant benchmarking
  - Is under continuous enhancement
- Training is available
- In cooperative development between HSW Technologies LLC and Georgia Tech/NRE

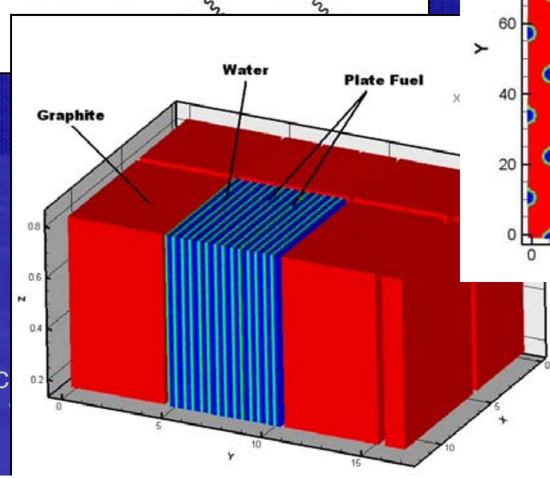


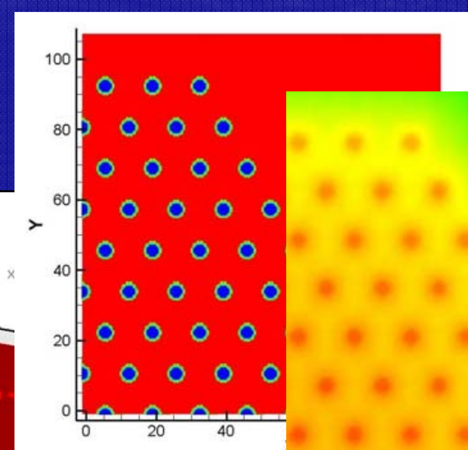
PENTRAN Parallel Code System

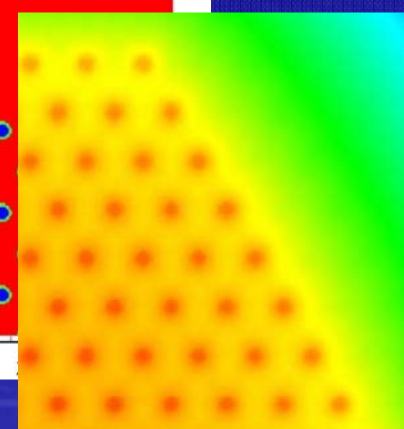
- PENMSHxp -- “PENMSH eXPress” by C. Yi and A. Haghighat
  - Automatic mesh and code deck generation
- “PENTRAN-CRT” Code with adaptive Sn and Characteistic Ray Trace Hybrid
  - Now Undergoing testing and benchmarking with TITAN code
- “PENTRAN-MP” Code Set for Whole Body Medical Physics Applications
  - Under further development and testing with Electron Dose Kernel (EDK) Methodology
- “PENBURN” Burnup Module for 3-D burnup/parallel chain processing
- Used for new methods in detection, characterization, reactor analysis
- Scalable memory & processing enables ‘extreme’ transport to >10^9 unknowns

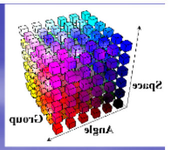






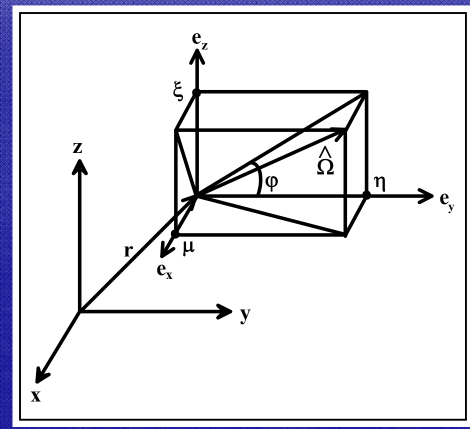






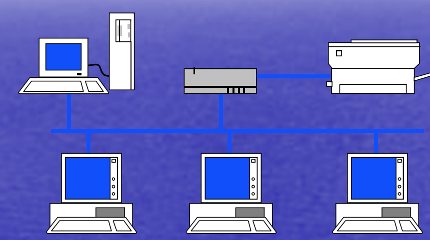
PENTRAN Overview

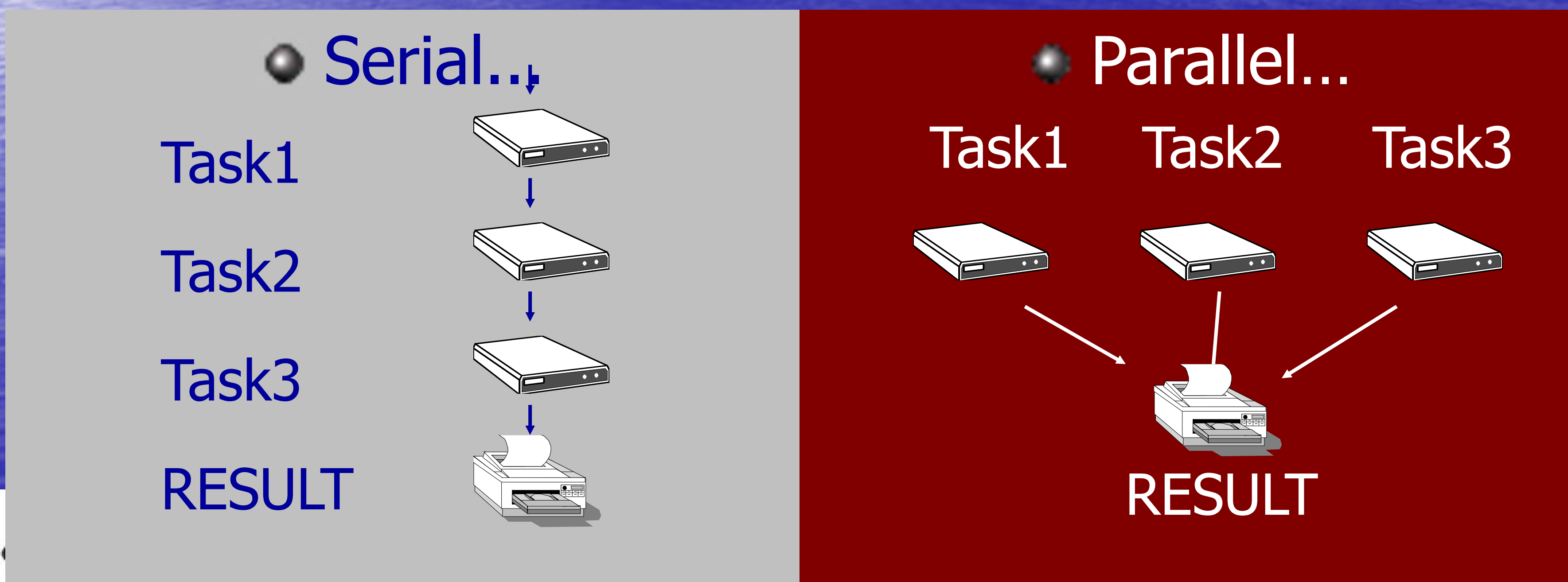
- Parallel Computing & MPI
- Boltzmann & Sn Transport
- PENTRAN^{TM} Code System
- Features
  - – Adaptive Differencing, Parallel
- Discussion and Summary
- Questions

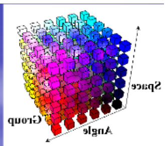


What is Parallel Computing?

- Computer with many processors
  - "Lumped" or *Network* "Distributed"
  - Divide problem up on processors







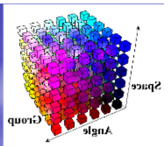
About MPI…

- MPI (Message Passing Interface) Library
  - Began with "MPI Vendor Forum" May 1994
    - Formalized in 1995
- For distributed memory machines (FORTRAN, C, other language variants support)
- ANSI-like "standard" in message passing
- Process com groups, decomposition topologies
- Port code directly to architectures running MPI
- Versions: MPICH, MPICH2, OPENMPI

About MPI… (2)

- A Minimum Set of MPI Calls (Fortran or C):

Function	*Explanation*
MPI_INIT	Initiate MPI
MPI_COMM_SIZE	Find out how many processes there are
MPI_COMM_RANK	Determine rank of the calling process
MPI_SEND	Send a message
MPI_RECV	Receive a message
MPI_FINALIZE	Terminate MPI

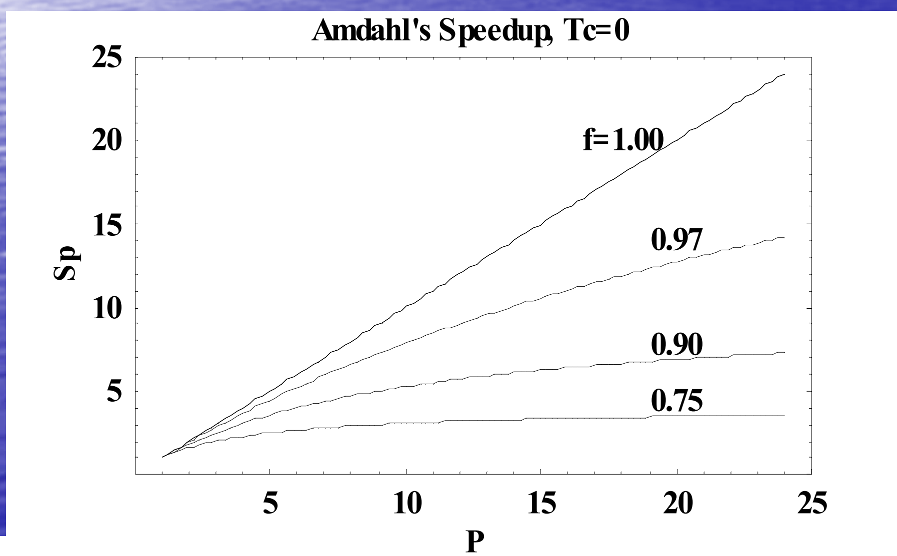


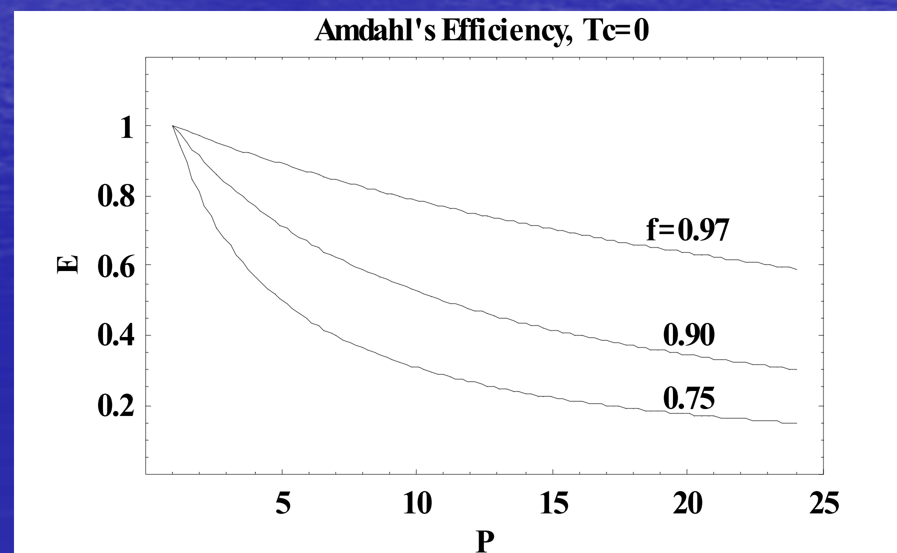
Parallel Job Setup/Compile

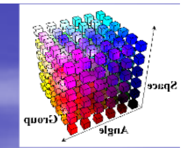
- Normal FORTRAN or C programming
- MPI libraries are included in each routine requiring them:
  - INCLUDE 'mpif.h', include 'mpi.h'
- Same code executed on each processor
- Logical coding controls "who does what"
- User responsible for:
  - start/stop/all synchronization
  - message passing calls/memory management

Amdahl's Law: Speedup is Limited

- (Parallel Code Fraction = f)
- $S_p = 1 / ((1-f) + f/p + T_c/T_s)$
- Speedup= $S_p = T_s / T_p$, Efficiency = $E = S_p / P$
- If f =0.8, then max S_p =5.0 *as P->Infinity*







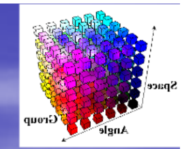
3-D Boltzmann Equation

$$(\mu \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y} + \xi \frac{\partial}{\partial z})\, \psi_g(x,y,z,\mu,\varphi) + \sigma_g\,(x,y,z)\, \psi_g(x,y,z,\mu,\varphi) =$$

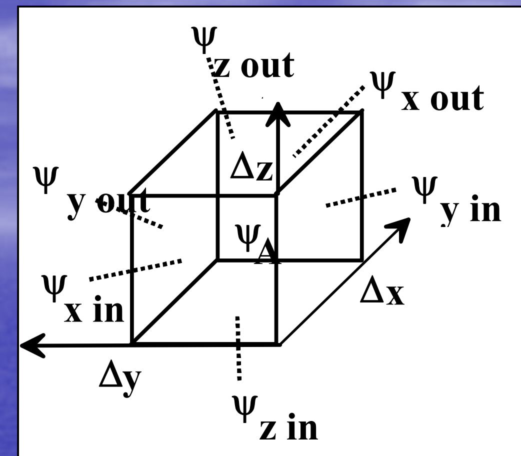
$$\sum_{g'=1}^{G} \sum_{l=0}^{L} (2l+1)\; \sigma_{s,g'\to g,l}(x,y,z)\, \{ P_l(\mu)\phi_{g',l}(x,y,z) + 2 \sum_{k=1}^{l} \frac{(l-k)!}{(l+k)!} P_l^k(\mu) \cdot$$

$$\left[\phi^k_{C\,g',l}(x,y,z)\cos(k\varphi) + \phi^k_{S\,g',l}(x,y,z)\sin(k\varphi) \right] \} + \frac{\chi_g}{k_0} \sum_{g'=1}^{G} \nu\sigma_{f g'}(x,y,z)\; \phi_{g',0}(x,y,z)$$

- Boltzmann Transport Equation
- Track particles traveling in different …
  - *directions*
  - over a range of *energies*
  - in different *spatial locations* in 3-D



Transport Theory



- **Boltzmann transport methods**
  - **Monte Carlo method**
  - **Discrete Ordinates (S_N) method**
  - **Each has specific …**

    ***Advantages* (+)**

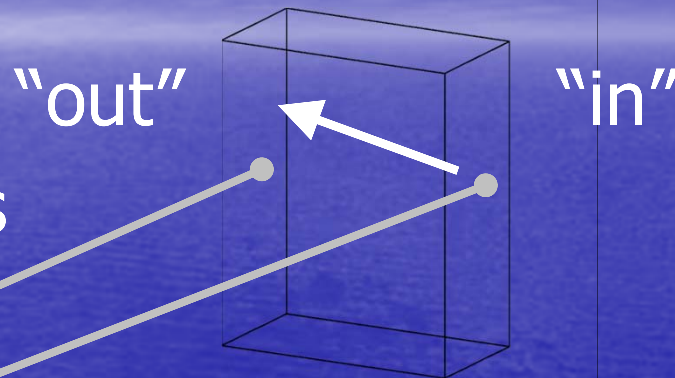
    ***Disadvantages* (-)**

    Issues

- **Both methods can take advantage of parallel processing!**

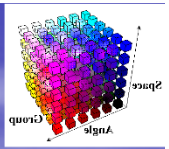
3-D Discrete Ordinates (Sn)

- Balance Equation…
- Surface Averaged Fluxes

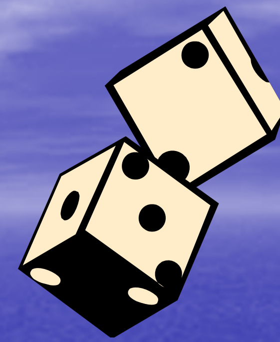


$$\frac{|\mu_m|}{\Delta x}(\psi_{\text{out}\,x} - \psi_{\text{in}\,x}) + \frac{|\eta_m|}{\Delta y}(\psi_{\text{out}\,y} - \psi_{\text{in}\,y}) + \frac{|\xi_m|}{\Delta z}(\psi_{\text{out}\,z} - \psi_{\text{in}\,z}) + \sigma\psi_A = q_A$$

$$\psi_{\text{out}\,x} = \frac{1}{\Delta y \Delta z} \int_0^{\Delta y} \int_0^{\Delta z} \psi_m(\Delta x, y, z)\, P_0(y)\, P_0(z)\, dy\, dz$$



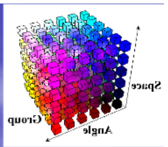
Monte Carlo



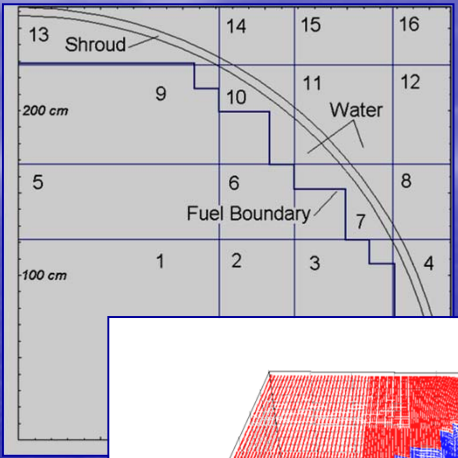
- Advantages
  - Traditional, straight forward
  - Geometrically precise
  - Robust particle physics--Continuous-Energy xsec
  - Parallelization obvious: particle histories
- **Disadvantages:**
  - Processing time, Non-analog variance reduction
  - Inevitable uncertainties and Central LT
  - “Global” solution difficult to obtain
  - Results can be limited in application

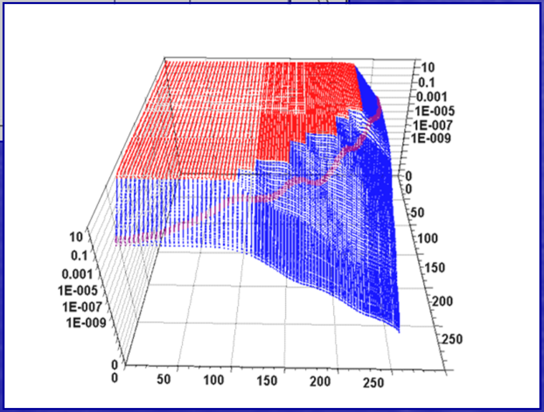
Georgia Tech

HSW TECH



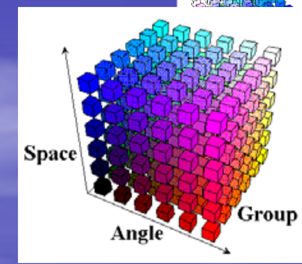
Discrete Ordinates



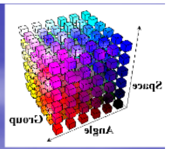


- **Advantages**
  - **Model entire geometry**
  - **Fast and accurate**
  - **Global flux distribution**
  - **Directly allows for burnup, etc**
- **Disadvantages:**
  - **Proper multi-group cross sections**
  - **Geometry discretized**
  - ***Memory*, storage, differencing scheme issues**
  - **Parallelization--coupling in *angle, energy, space***

PENTRAN Code System

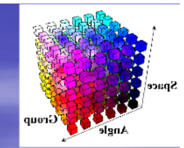


- *P*arallel *E*nvironment *N*eutral-particle *TRAN*sport
  - Two decades of continuous development
    - Introduced in 1996 by Sjoden and Haghighat
    - Parallel S_N in *angle*, *energy*, and *space* & *I/O*
    - ANSI FORTRAN 90, ~40k lines
    - Industry standard FIDO input
  - Solves 3-D Cartesian Sn Equations
    - Multigroup, anisotropic transport problems
  - Forward and adjoint mode
    - Fixed source
    - Criticality eigenvalue problems



Distributed Memory Programming Philosophy PENTRAN

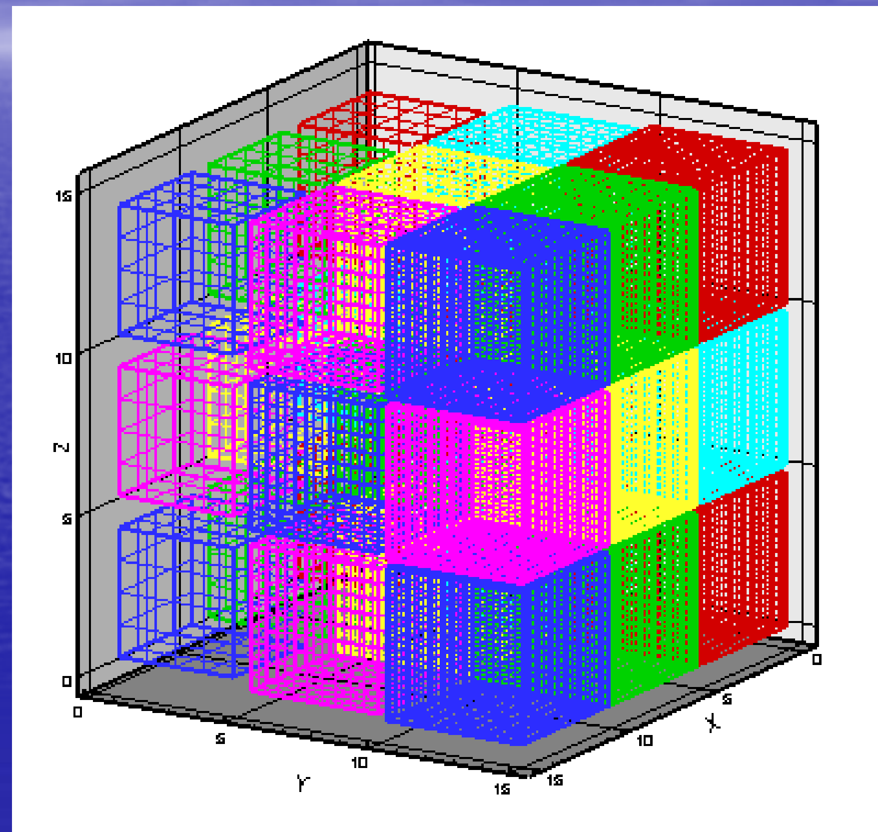
- **Programming Model:**
  - View message passing as a *last resort*
    - Minimize serialization, minimize barriers
  - Use a partitioned memory storage of data
    - Only way to *solve larger problems*
    - *Process mapping arrays* to determine "who's where"
- **Overall…**
  - Use of a coarse-grained code structure
  - Many flops performed before messages passed
  - Don't forget about *Amdahl's Law*



Coarse Mesh/Fine Mesh Scheme

Primary Two level meshing scheme for Cartesian geometry

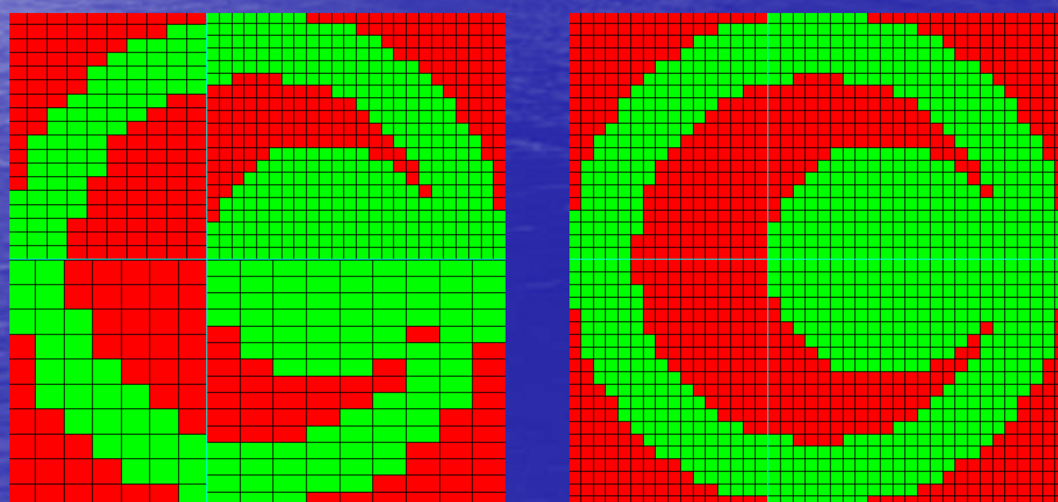
- Model divided into coarse meshes (CMs)
  - Each CM filled with fine mesh grids
  - "Block adaptive" fine mesh structure,
  - No requirements on surrounding CMs; Higher order coupling with TPMC
- Intermediate mesh level ("medium" grids) for "/-multigrid" nested iteration



3 x 3 x 3 coarse mesh Model

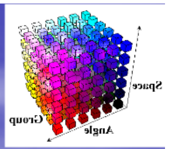
Block Adaptive Mesh Generation

- **PENMESH-XP**- automatically sets up problem
  - You define shapes, 3-D intervals, it does the rest…
  - Coarse Mesh Structure
  - Fully discontinuous variable meshing among coarse meshes



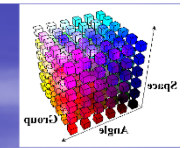
Georgia Tech

HSW TECH

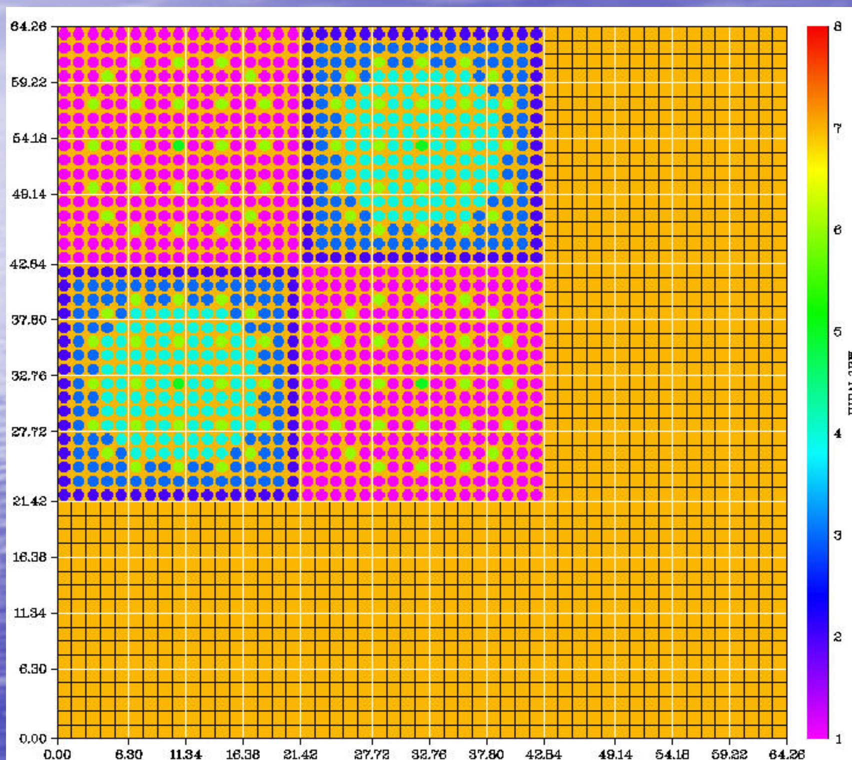


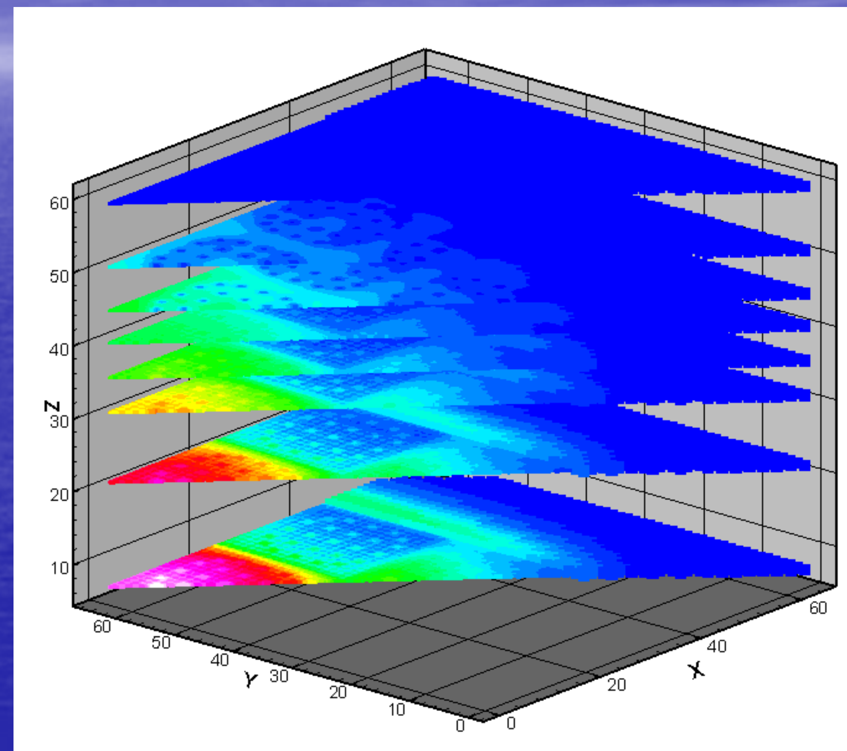
PENMSH-XP input: "Z-level" basis

File names	Description	PENMSH XP
penmsh.inp	**Meshing parameters**	**Required**
prbname#.inp	**Meshing per 'z level'**	**Required**
prbname.src	Fixed source grid	Optional
prbname.spc	Source spectrum	Optional
prbname.chi	Fission spectrum	Optional
prbname.mba	**Material balance**	**Optional**
prbname#.flx or *prb#.flx* or *.fjn* files	Group fluxes (*.flx*) or current (*.fjn*) generated by PENDATA from PENTRAN calculation	Optional

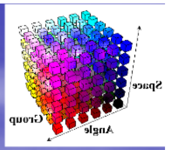


Sample Geometry Slice from PENMSH-XP





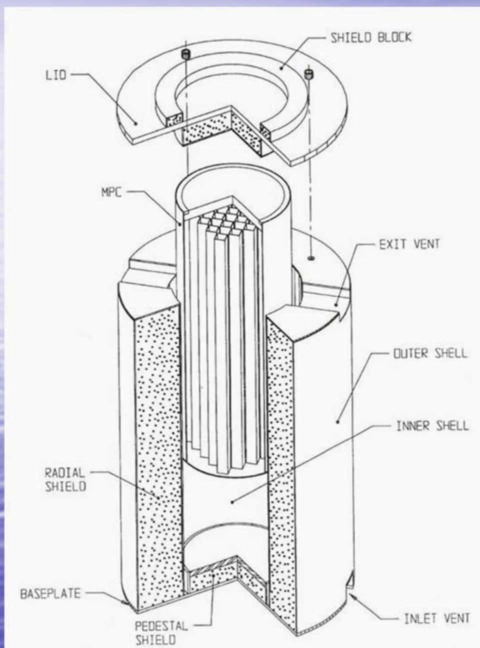
2-D Geometry, Flux Plots, nuclear reactor model

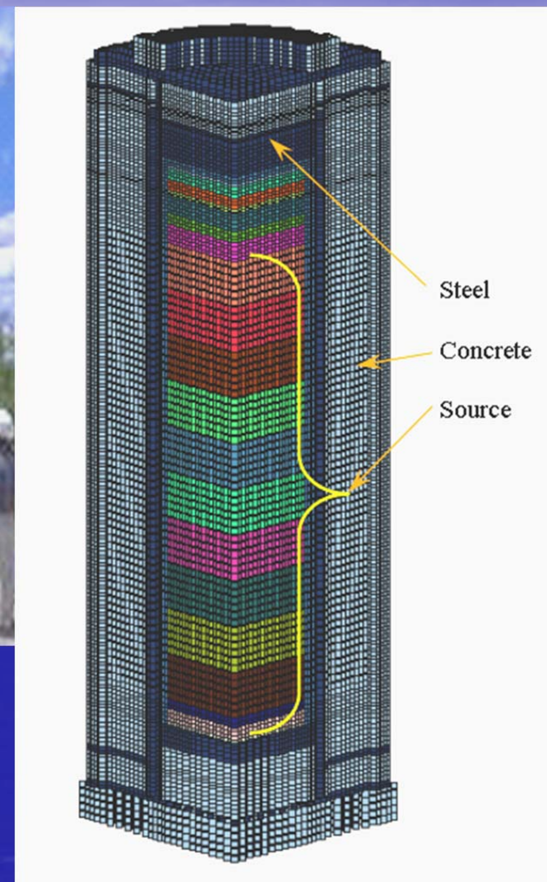


PENTRAN
Parallel Implementation

- Uses MPI Message Passing Interface library
- Performs all I/O in parallel by each processor
- Uses a local, partitioned memory for memory intensive arrays (angular fluxes, etc)
- *Memory Tuning* feature for optimum memory use
- Automatic scheduling based on a user-specified parallel *decomposition weighting vector*
- Optimized Processor MPI "Communicators"
  - Minimized messages, comm. processor "planes"
- Adaptive Differencing

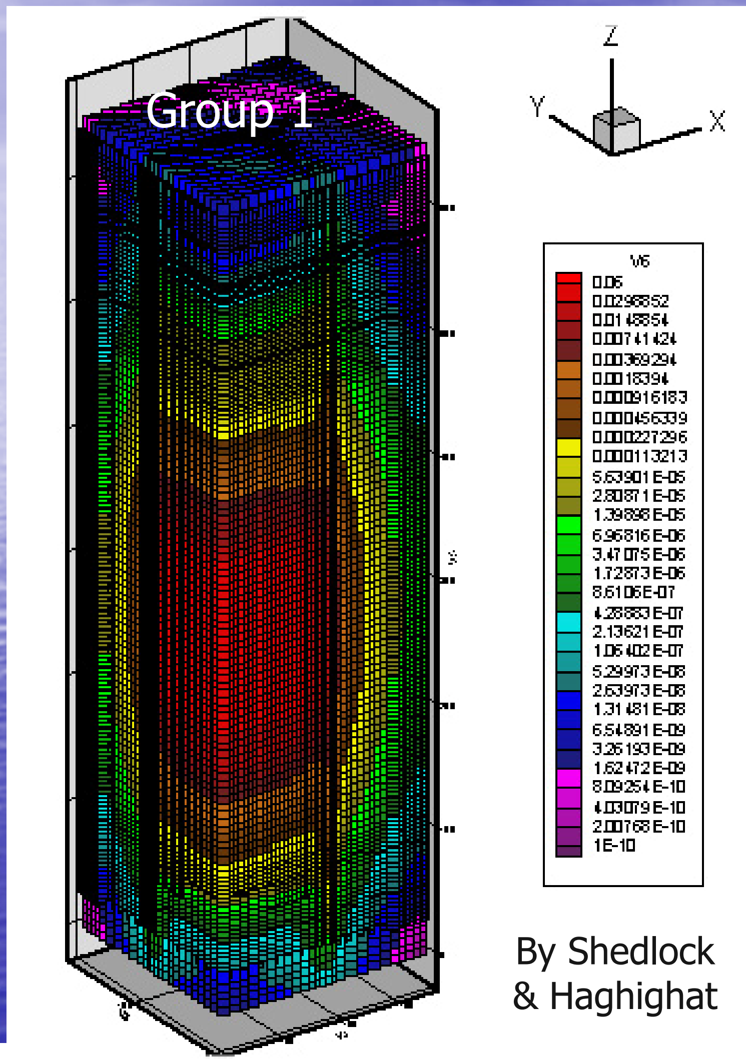
Memory Tuning Example: Fuel Cask Model



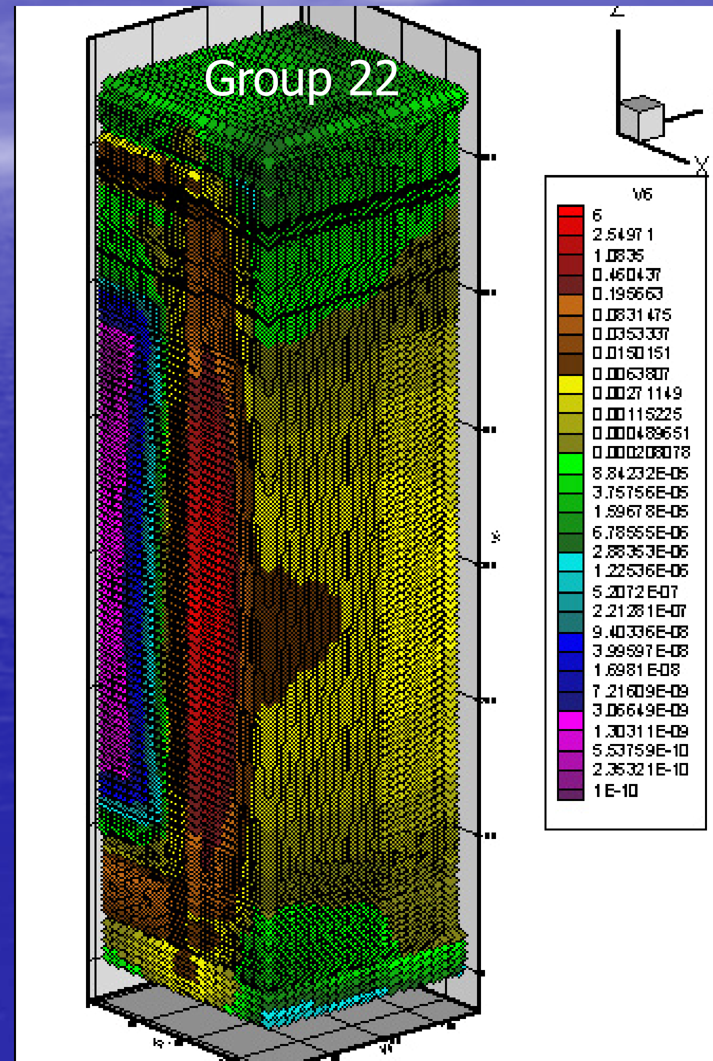


PENTRAN 3-D model of fuel storage cask, 1/4 symmetry, 40 Group P3-S12, 318,426 cells : ***2.14 Billion Eqns***

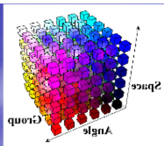
Cask 3-D Flux Distribution

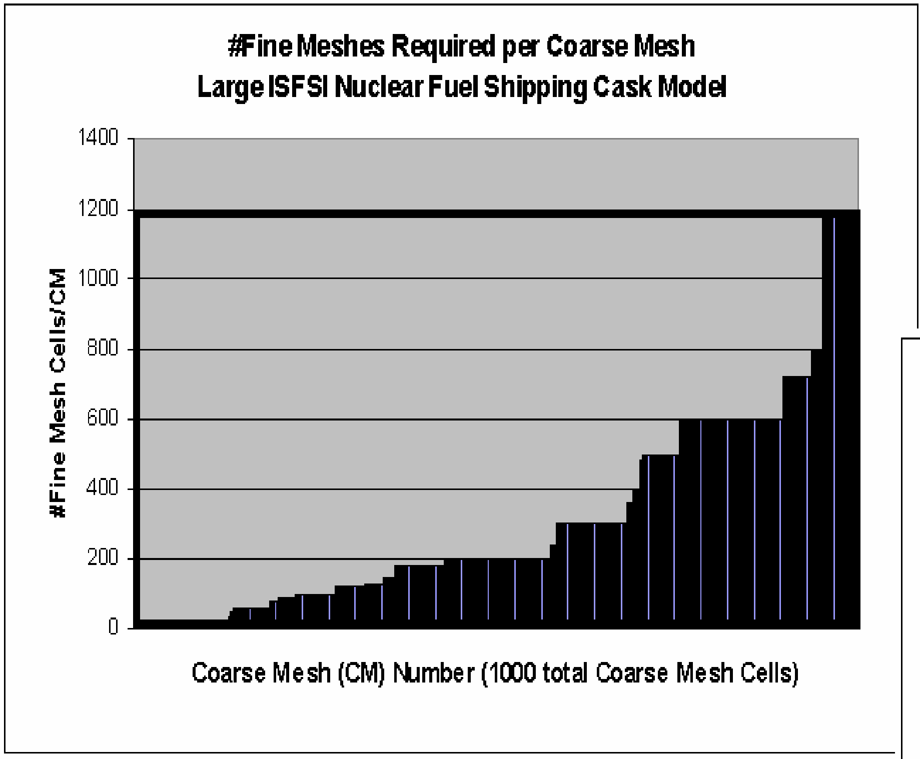


By Shedlock
& Haghighat

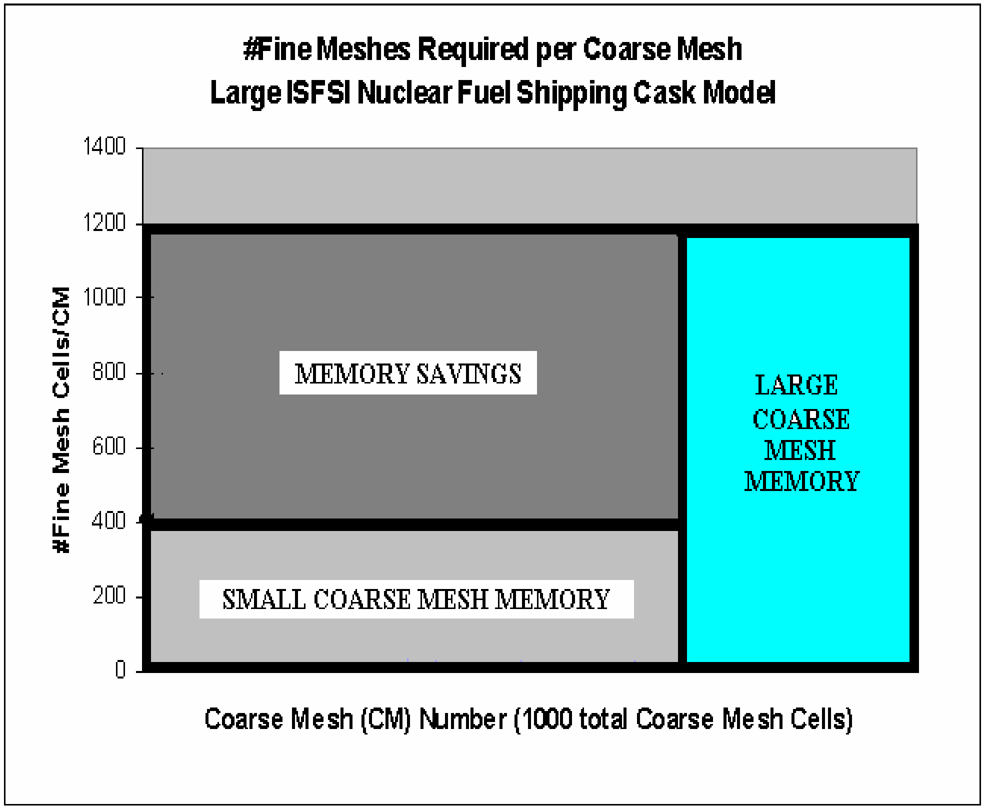


Two-Level Memory Tuning

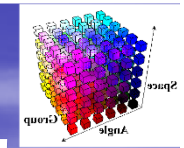




Original ISFSI Cask Model global upper limit memory specification wasteful



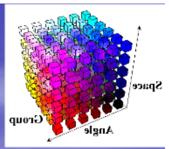
Optimized *Memory Tuning* Cask Model with Two-Level memory specification



3-Level Adaptive Differencing

- PENTRAN has 3 *adaptive differencing* levels
  - All schemes based on 0th spatial moment balance
  - Less parallel information to pass on a given mesh compared to higher order methods
  - *Level 1*: Diamond Zero (DZ)
    - Use DZ until a fixup is needed
  - *Level 2*: Directional Theta-Weighted (DTW)
    - Use until weight exceeds 0.95 (Petrovic and Haghighat)
  - *Level 3*: Exponential Directional-Iterative (EDI)
    - Intended for severe streaming (Sjoden; Sjoden and Haghighat)

Level 1: DZ

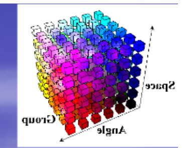


- Diamond Zero (DZ) Differencing

$$\psi_A = \frac{(\psi_{\mathrm{in\,u}} + \psi_{\mathrm{out\,u}})}{2} \qquad \text{where} \quad u \in \{x, y, z\}$$

+ Still main scheme in many production codes
+ Simple, fast, linear flux extrapolation
+ Good in source regions/slowly varying flux
– Non-linear “zero fixup” when exiting flux < 0
– Prone to unphysical solution oscillations
– Very ill-conditioned for parallel processing

Level 2: DTW



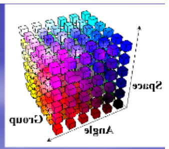
- Directional Theta Weighted Differencing

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

$$\psi_{\text{out } x} = 1/a(\psi_A + \psi_{\text{in } x}(a-1))$$

  + evolution from TW scheme (Rhoades & Engle) to a directional weighting by Petrovic and Haghighat
  + *positive*, with "proper" stable flux behavior
  + eliminates inherent oscillations
  + excellent in general, costs ~10%>DZ in 3-D
  – inaccurate with severe streaming, larger cells
    - equivalent to step scheme when cells are thick

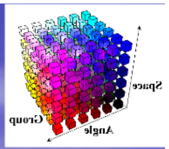
Level 3: EDI



- Exponential Directional Iterative Differencing

$$\psi_m(x,y,z) = a_o\ exp(\lambda_i\ P_1(x)/|\mu_m|)\ exp(\lambda_j\ P_1(y)/|\eta_m|)\ exp(\lambda_k\ P_1(z)/|\xi_m|)$$

+ DTW predictor provides λ coefficient guess for auxiliary
+ Grew from EDW scheme
+ EDI formulation corrects average, outgoing flux
+ Still a 0th order based (spatial moment) scheme
+ Inherent positivity, accuracy in severe streaming
+ Cost: up to 2x DTW in 3-D, depending on λ coeff iterations



Level 3: EDI

- Consider auxiliary partial derivatives

$$\frac{1}{\psi}\frac{\partial \psi}{\partial x} = \frac{2\lambda_i}{\Delta x\,|\,\mu\,|} \qquad \frac{1}{\psi}\frac{\partial \psi}{\partial y} = \frac{2\lambda_j}{\Delta y\,|\,\eta\,|} \qquad \frac{1}{\psi}\frac{\partial \psi}{\partial z} = \frac{2\lambda_k}{\Delta z\,|\,\xi\,|}$$

- Applying the balance equation along *x* yields

$$\iiint \left(\frac{\partial \psi}{\partial x} = \frac{2\lambda_i}{\Delta x \mu_m} \psi(x,y,z) \right) dx\, dy\, dz \quad \rightarrow \quad (\psi_{out} - \psi_{in}) = \frac{2\lambda_i}{\Delta x \mu_m} \psi_A$$

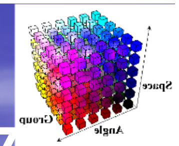
- Fixed point iteration formulations

$$\lambda_{i,I} = f(\lambda_{i,I-1}) = \frac{(\psi_{out\,\mathrm{x}}(\lambda_{i,I-1}) - \psi_{in\,\mathrm{x}}(\lambda_{i,I-1}))\,|\,\mu_m\,|}{2\psi_A(\lambda_{i,I-1})}$$

$$\lambda_{j,I} = g(\lambda_{j,I-1}) = \frac{(\psi_{out\,\mathrm{y}}(\lambda_{j,I-1}) - \psi_{in\,\mathrm{y}}(\lambda_{j,I-1}))\,|\,\eta_m\,|}{2\psi_A(\lambda_{j,I-1})}$$

$$\lambda_{k,I} = h(\lambda_{k,I-1}) = \frac{(\psi_{out\,\mathrm{z}}(\lambda_{k,I-1}) - \psi_{in\,\mathrm{z}}(\lambda_{k,I-1}))\,|\,\xi_m\,|}{2\psi_A(\lambda_{k,I-1})}$$

- Costs ~40%>DZ



EDI: NSE Vol 155, pp. 179-189 (Feb 2007)

NUCLEAR SCIENCE AND ENGINEERING: 155, 179–189 (2007)

An Efficient Exponential Directional Iterative Differencing Scheme for Three-Dimensional S_n Computations in *XYZ* Geometry

Glenn E. Sjoden*

University of Florida
Department of Nuclear and Radiological Engineering
202 NSB, P.O. Box 118300
Gainesville, Florida 32611-8300

Received December 23, 2005
Accepted April 28, 2006

Abstract—*A new exponential spatial differencing scheme based on zeroth spatial transport moments, the exponential directional iterative (EDI) S_n scheme for three-dimensional (3-D) Cartesian geometry, is presented. The EDI scheme is a logical extension of the positive, efficient exponential directional weighted (EDW) method used in the PENTRAN parallel S_n solver in an adaptive differencing strategy. The EDI scheme uses EDW-rendered exponential coefficients as initial values to begin a fixed-point iteration to refine exponential coefficients. Iterative refinement of these coefficients typically converged in fewer than four fixed-point iterations per ordinate, and yielded more accurate angular fluxes compared to other schemes tested. Overall, the EDI scheme is an order of magnitude more accurate than EDW, and two orders of magnitude more accurate than the legacy diamond zero (DZ) scheme for a given mesh. EDI is therefore a good candidate for a fourth-level scheme in the PENTRAN adaptive sequence. The 3-D Cartesian computational cost of EDI was ~20% more than EDW, and only ~40% more than DZ. Thus, EDI renders increased accuracy using zeroth spatial transport moments in a straightforward manner for any 3-D Cartesian code. More evaluation is ongoing to determine suitability in an upgraded adaptive differencing sequence algorithm in PENTRAN.*

I. INTRODUCTION

This paper introduces the new exponential directional iterative (EDI) scheme for evaluation as a new differencing algorithm for three-dimensional (3-D) Cartesian coordinates. Here, the derivation and advantages of the EDI scheme are presented as it is implemented into the PENTRAN parallel discrete ordinates code. An adaptive differencing algorithm currently in the PENTRAN parallel discrete ordinates solver uses either diamond zero (DZ), directional theta weighted (DTW), or exponential directional weighted (EDW) schemes, depending upon regional problem physics and angular flux gradient. As a result of this work, the EDI scheme may offer more accuracy in certain problems than the EDW scheme, and is a good candidate to consider as a fourth-level adaptive scheme.

Many S_n differencing schemes have been formulated over the years since the introduction of the discrete ordinates (S_n) method by Carlson.[1] Following some initial development and application of the S_n method, Lathrop emphasized that positivity and accuracy are often mutually exclusive issues with regard to spatial differencing schemes, especially in the early applications of the discrete ordinates method. He also noted the unique difficulties inherent in quasi-analytic formulations extended beyond slab geometry, where numerical overhead for some methods may become quite impractical, which has contributed to continued wide application of the DZ fixup scheme.[2] In many ways, Lathrop's observations

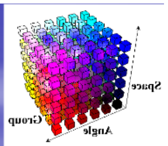
*E-mail: sjoden@ufl.edu

Non-Upgraded Schemes

- Scheme 4: EDW Scheme
  - Similar to EDI scheme, but exponential arguments fixed at initial guess from DTW without iteration
  - Not typically as accurate as EDI, but can be a compromise between speed and accuracy when regions are known to require exponential treatment
  - Is not part of the 1-2-3 adaptive differencing sequence and must be selected by the user

Non-Upgraded Schemes

- Scheme 5: CRT Scheme (Developed by Yi and Haghighat)
  - Characteristic Ray Trace (CRT) is a block-oriented transport solution method adapted for a coarse-mesh where the transport is solved along a characteristic ray projected across the coarse mesh coupled in flow to surrounding Sn meshes
  - Each set of characteristic rays are parallel to a direction of the angular quadrature from Sn.
  - Considers the whole coarse mesh as one region
    - The cross-section and angular source terms are constant throughout the region
  - CRT is **<u>not</u>** allowed for CMs containing fission or independent sources



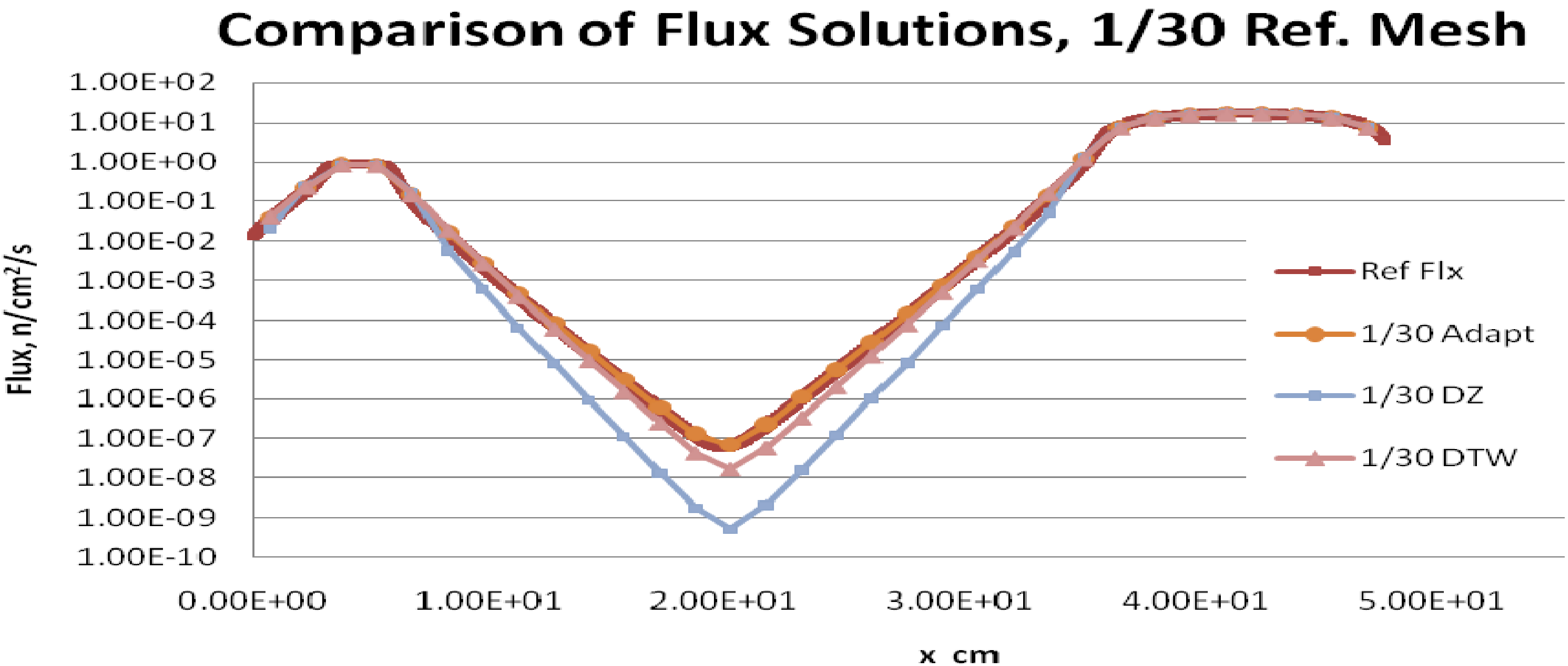
3-Level Adaptive Differencing: Selective Upgrade/Tuning Feature

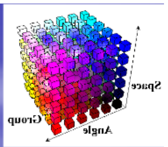
- *Level 1 → Level 2* : DZ to DTW Upgrade
  - Use DZ until a fixup is needed
- *Level 2 → Level 3* : DTW to EDI Upgrade
  - Use until DTW weight exceeds 0.95 (weights close to 1 indicate 'step' differencing)
    - Strategy effective for no or low-level source
    - For strong sources, a test is performed with *'qfratio'*
      - If *qfratio > 1.00 (default), DTW persists and we do not upgrade*

$$\frac{\hat{\Omega}\cdot\nabla\psi_g}{\sigma_g\psi_g}+1=\frac{(q_{sg}+q_{fg}+q_{ind\,g})}{\sigma_g\psi_g}=qfratio$$

  - No upgrade is performed if optical thickness of cells is less than 0.02 – for 'vanishingly thin cells' – stay with DTW

Numerical Testing: Simple Slab



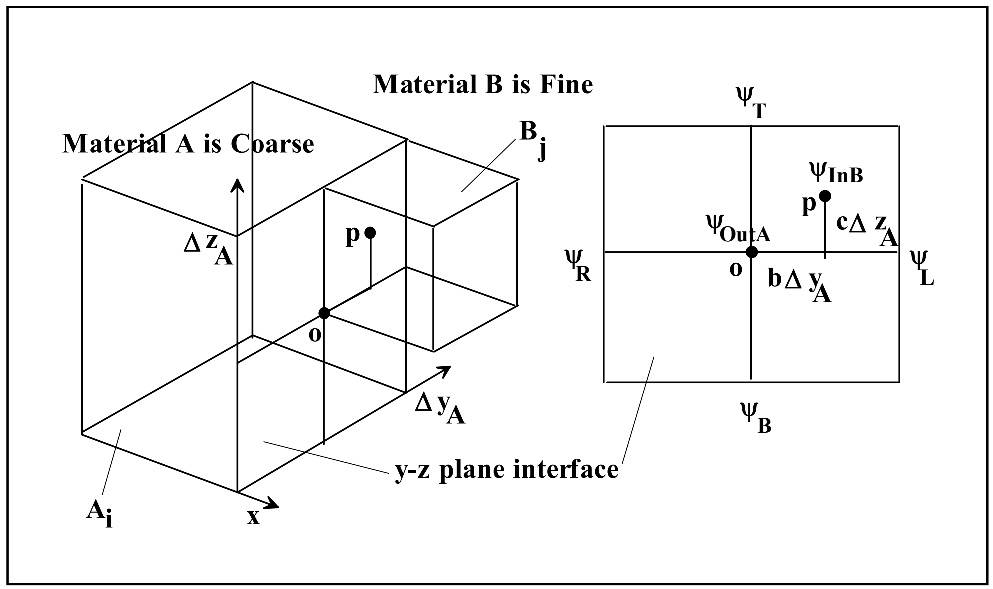


Other Numerical Treatments

- Uses a higher order mesh coupling scheme: *Taylor Projection Mesh Coupling (TPMC)*
- Accelerations…
  - Two-grid 3-D TPMC-coupled "/" multi-grid transport acceleration
  - PCR with a zoned rebalance acceleration
  - Preconditioning with REPRO
    - Use a previous model to accelerate computation
- Multigroup & One-level SI schemes

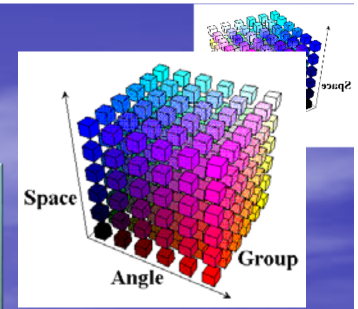
Taylor Projection Mesh Coupling

- **Coupling of Discontinuous grid densities**
  - Allow high definition in ROIs, parallel load balance
- **First Order Taylor projection of angular fluxes at interface between discontinuous grid surfaces**
  - Flow Step
  - Particle Balance Step
  - Important for "Coarse to Fine"



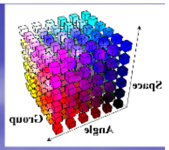
$$\psi_{\text{inB}} = \psi_{\text{outA}} + b\,\Delta y_A \frac{\partial \psi}{\partial y}\Big|_A + c\Delta z_A \frac{\partial \psi}{\partial z}\Big|_A + O(\Delta^2)$$

More Features

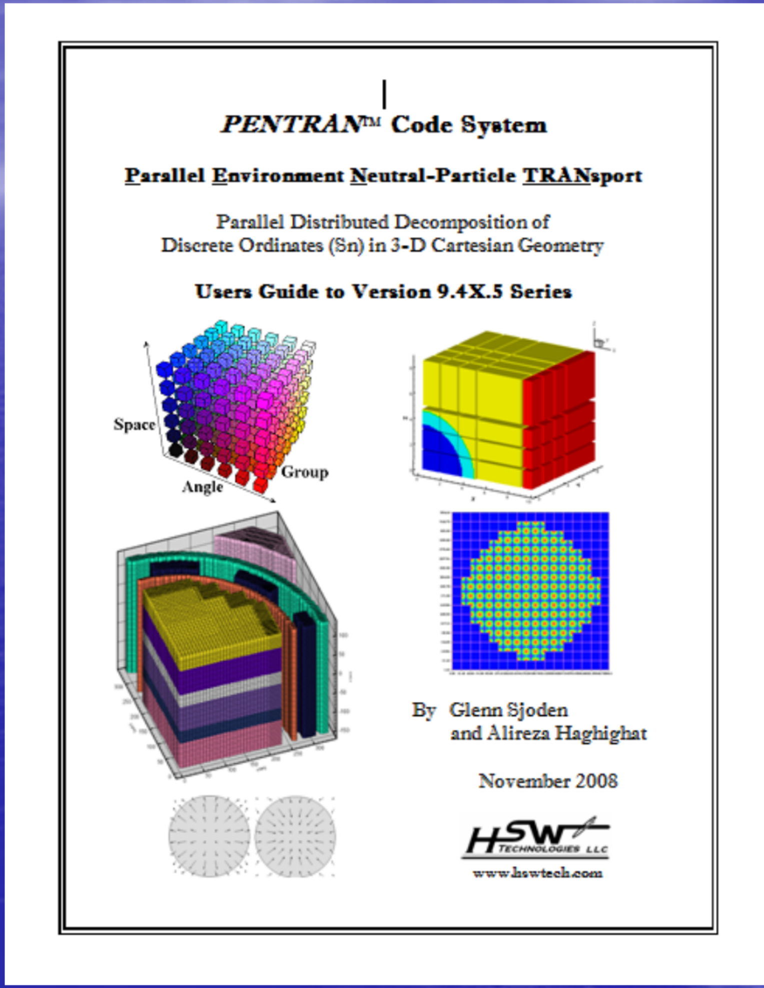


- Allows for automatic Red-Black or Block-Jacobi, Automatic load balancing
- Anisotropic scattering, arbitrary Pn moments
- 3-D angular quadratures
  - level symmetric through S_{20}, Legendre-Chebychev (Pn-Tn), arbitrary order
- Vacuum, reflective, group-albedo boundaries
- Volumetric sources, plane surface fluxes, currents
- PENDATA, other utilities…

PENTRAN



- Demonstrated 97% to 98% parallel fraction
  - Performance depends on problem, decomposition
  - Development focus on high accuracy & parallel efficiency
  - *Numerous applications over the past 14 years*
- Speedups of 50-nx100 are achievable
  - Demonstrated scalability
  - http://www.hswtech.com/pdfs/PENTRANmanual.pdf



PENTRAN™ **Code System**

Parallel Environment Neutral-Particle TRANsport

Parallel Distributed Decomposition of
Discrete Ordinates (Sn) in 3-D Cartesian Geometry

Users Guide to Version 9.4X.5 Series

By Glenn Sjoden
and Alireza Haghighat

November 2008

www.hswtech.com

Simple Example: hello.f

```fortran
      PROGRAM hello
c
c      This must be present in any routine using MPI calls
c
      INCLUDE 'mpif.h'
c
c      Declarations
c
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      IMPLICIT INTEGER (I-N)
      CHARACTER*12 messag,inmsg
      INTEGER nstat(MPI_STATUS_SIZE)
c
c      Initialize MPI
c
      CALL MPI_INIT (mpierr)
      CALL MPI_COMM_SIZE (MPI_COMM_WORLD, nproc, mpierr)
      CALL MPI_COMM_RANK (MPI_COMM_WORLD, me, mpierr)
c
c      Add +1 to process number (to shift ranks from 0,1 ... to 1,2)
c
      me=me+1
c
c      Set the tag (unique identifier) for this message set
c
      ntag=1
c      Set message length
c
      msglen=12
c
c      Send messages from processor 1 to all other processors
c
      IF(me.EQ.1) THEN
        messag='Hello, world'
        DO 10 i=2, nproc
          CALL MPI_SEND(messag,msglen,MPI_CHARACTER,(i-1),ntag,
     +      MPI_COMM_WORLD, mpierr)
10        CONTINUE
c
c        Assign inmsg variable on processor 1
c
        inmsg='Greetings'
      ELSE
c
c      Receive inmsg from processor 1
c
        CALL MPI_RECV(inmsg,msglen,MPI_CHARACTER,0,ntag,
     +    MPI_COMM_WORLD,nstat,mpierr)
      END IF
c
c      Write message on each node
c
      WRITE(6,*) 'node ', me,': ',inmsg
      CALL MPI_FINALIZE(ierr)
      END
```

Venus-3 Model

- PENMSH-XP code used to generate 3-D Cartesian Grid
- ~85,000 cells
- 26 Groups
- P3-S8 Discrete Ordinates

- Group 1 Flux Solution:

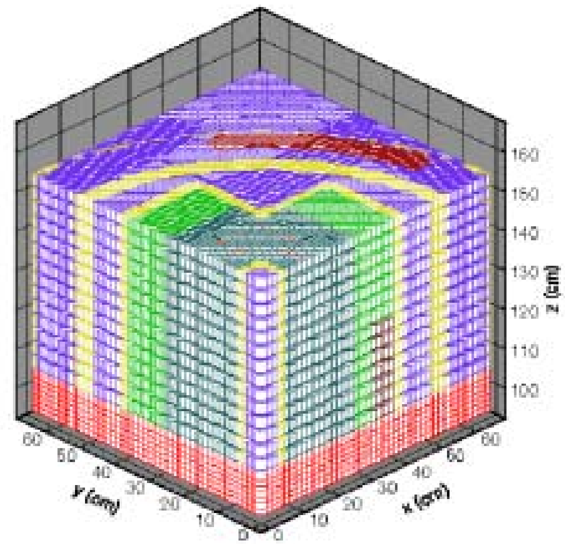


Figure 2: 3-D Mesh and Material distribution of PENTRAN Venus-3 Model (upper reflector not shown)

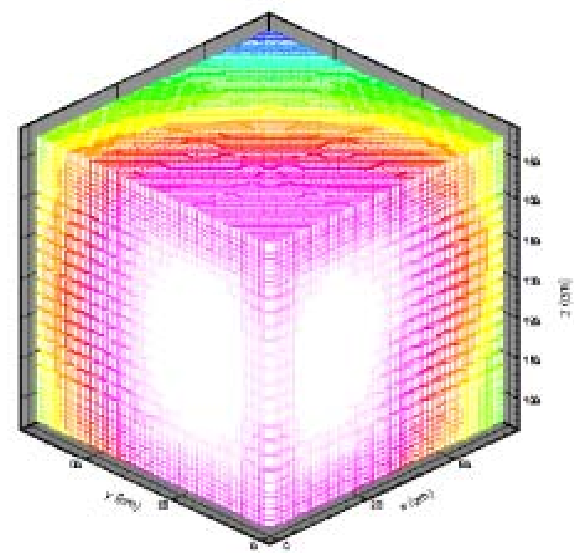
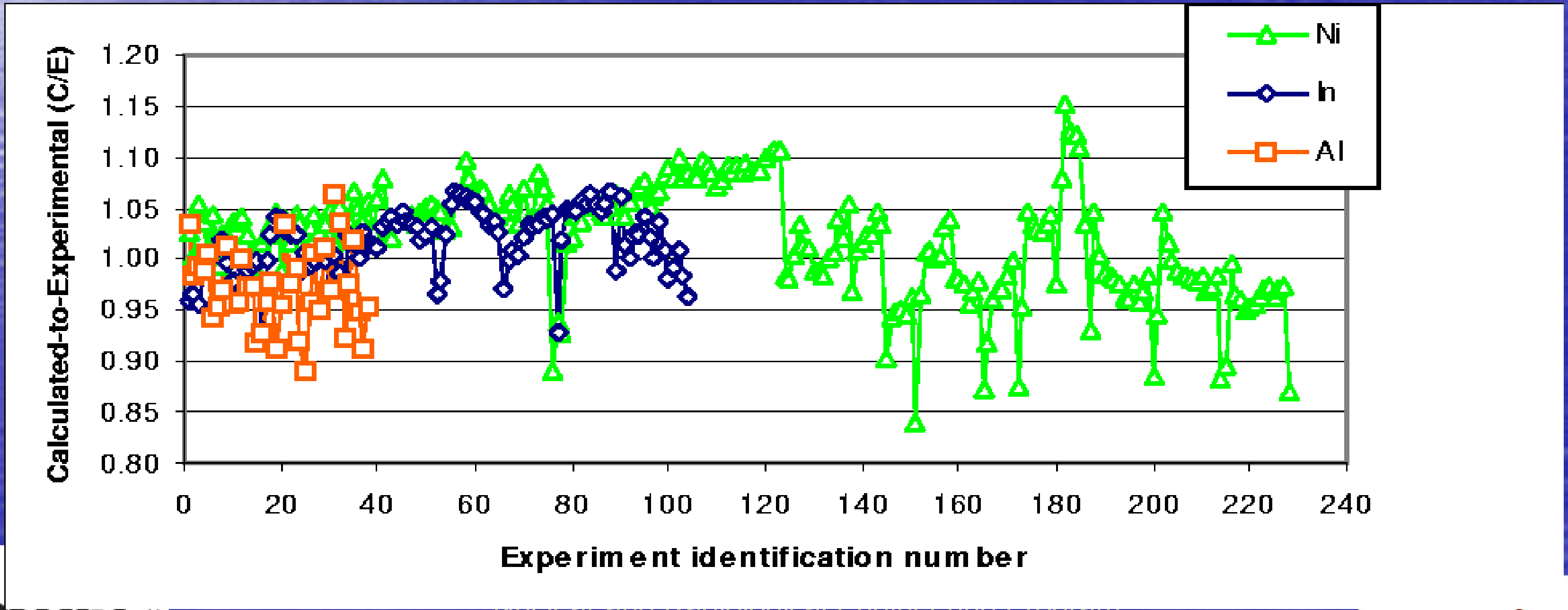


Figure 3: Group 1 Flux Distribution for PENTRAN Venus-3 Model

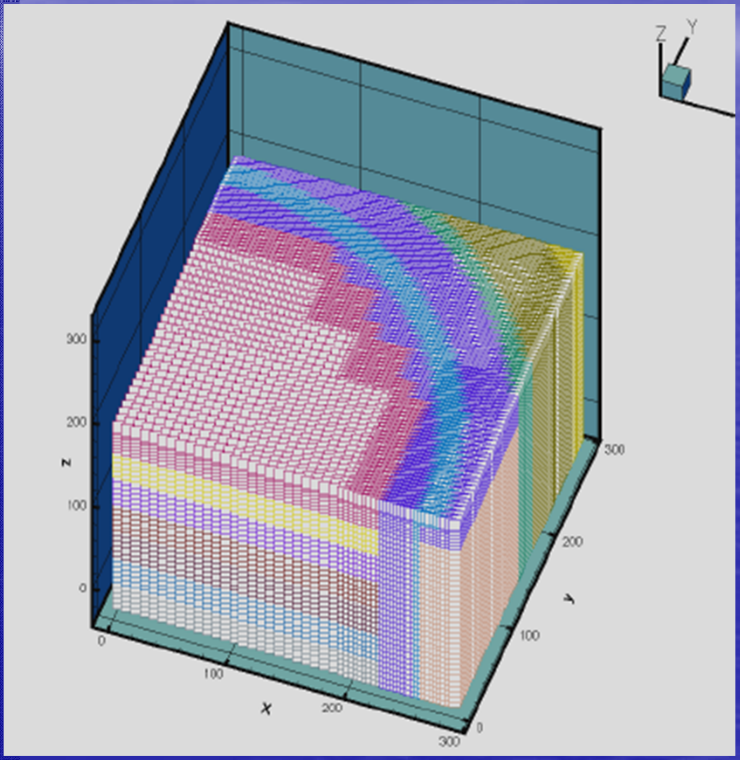
Venus-3 Results

- Compared 370 Measured Rxn Rates (Ni, In, Al dosimeters)
- vs Integral Rxn Rates computed from PENTRAN P3-S8 26 group-dependent fluxes.
- 95% C/E values +/-10%; 5% within +/-15% (near P/L rods).



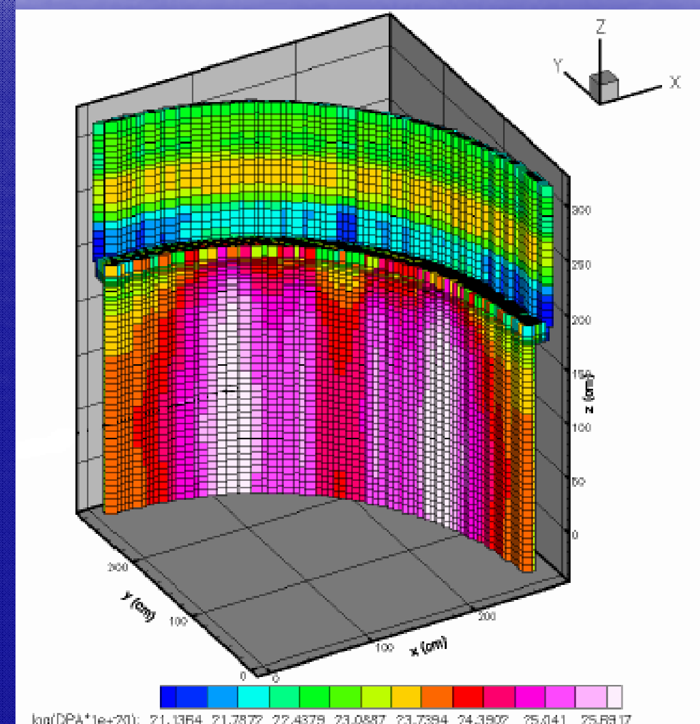
BWR Core Shroud Problem

- BWR reactor and Core Shroud Assembly…
- …with baffles, jet pumps, steam voids, etc.
- (Top) 67 Group P3-S8 coupled neutron-gamma calculation
- 265,264 fine mesh cells
- Solved in 12 hours on 48 IBM-SP2 processors… 8 processors angular, 6 processors spatial decomposition.



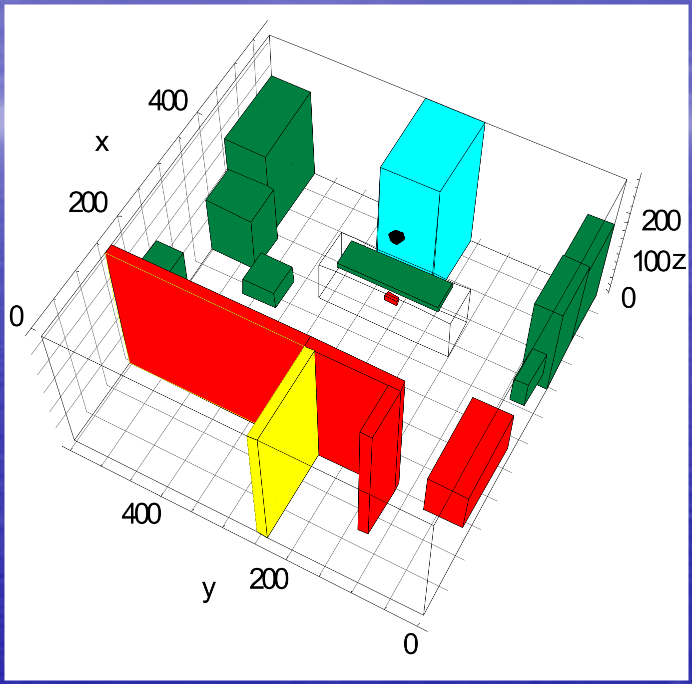
BWR Core Shroud Results

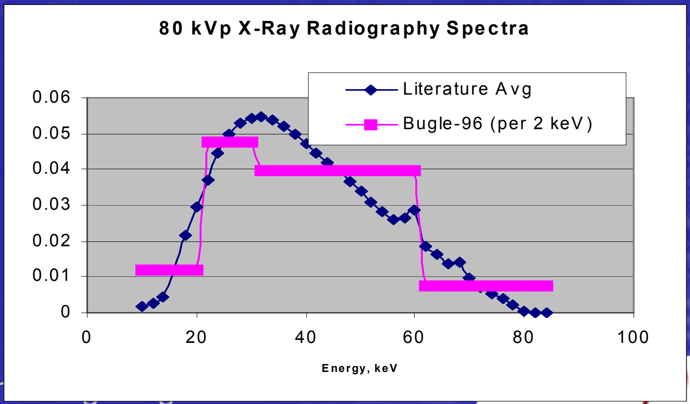
- Displacement per Atom (DPA) in the core shroud shows intense radiation damage where fuel is close to the shroud.
- Results were verified independently by Monte Carlo computations
- Multigroup PENTRAN (using BUGLE-96) values were within 5-15% of continuous energy MCNP values.
- (Kucukboyaci, et. al, 2000).



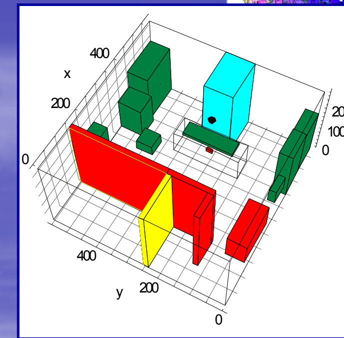
X-Ray Modeling …

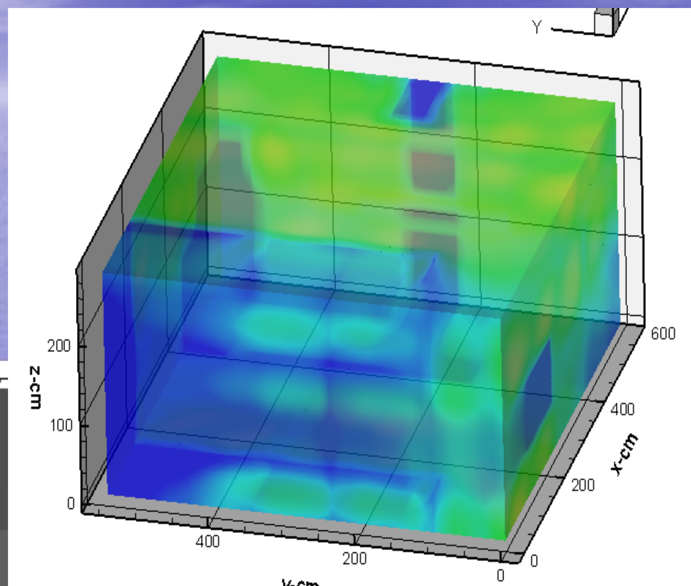
- 90 m^3 room discretized into ~131,000 3-D cells
- PENMSH™ code (8 "z-levels" floor to ceiling)
- BUGLE-96: last 4-group photon xsecs
- 80kV radiographic W-anode 32 mAs x-ray burst
  - Rotating anode water cooled
- Hybrid X-ray Spectrum
  - Characteristic X-rays
  - Bremsstrahlung continuum

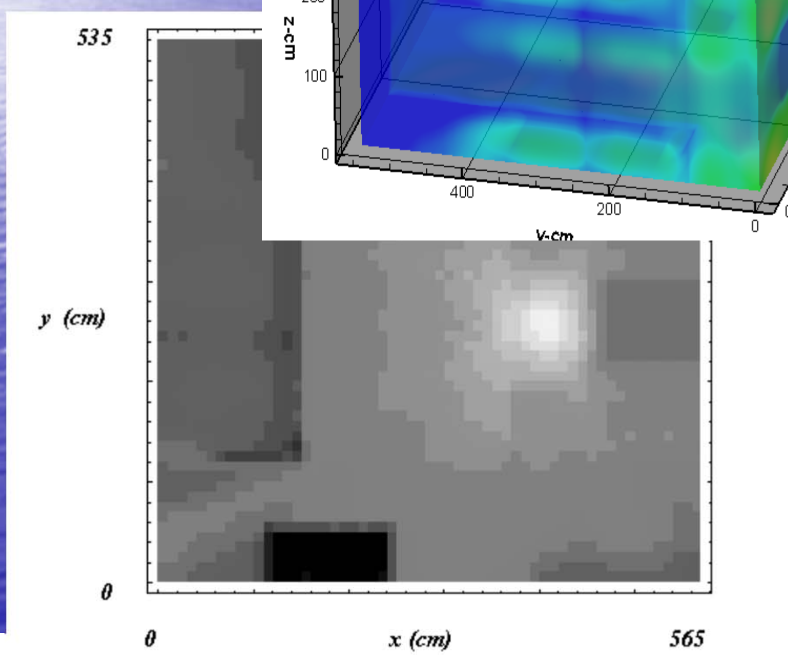


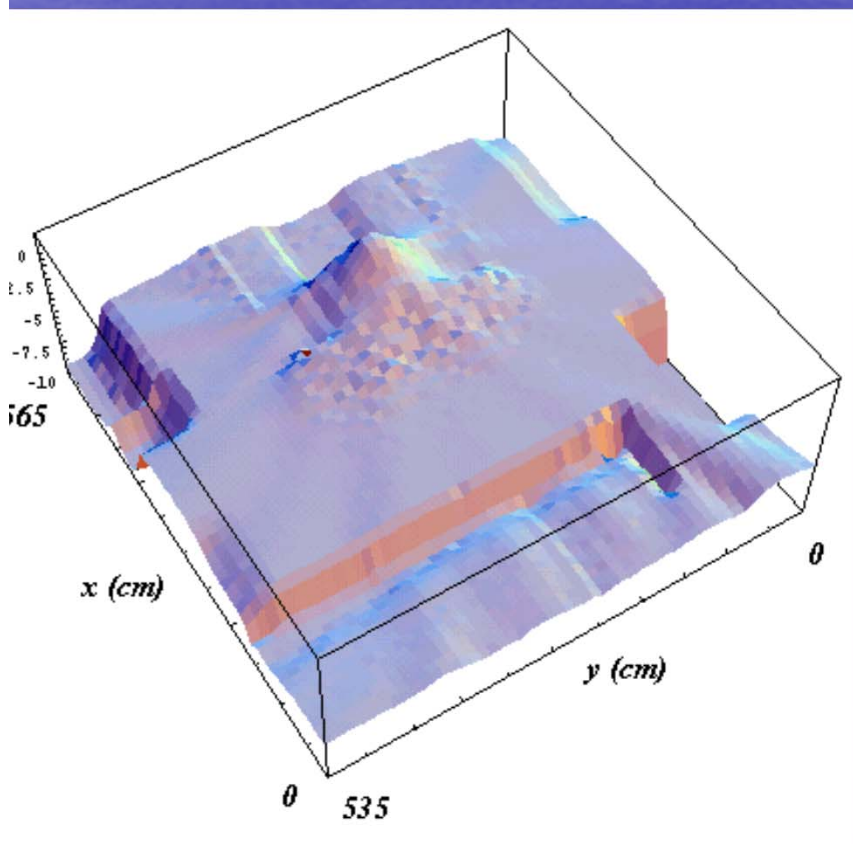


X-Ray Dose, z=100 cm

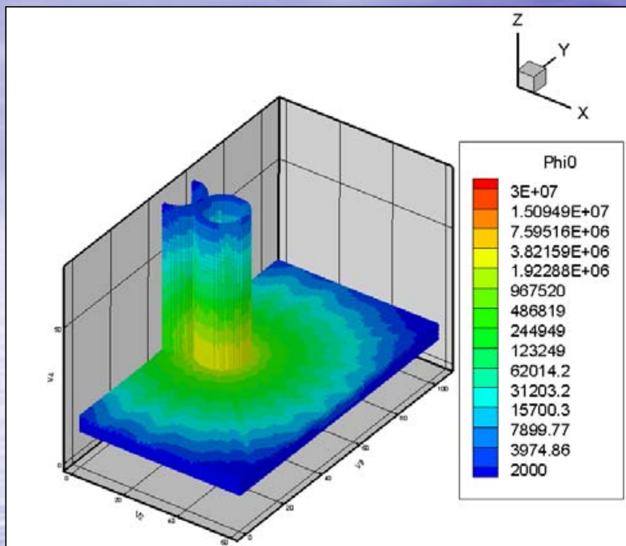




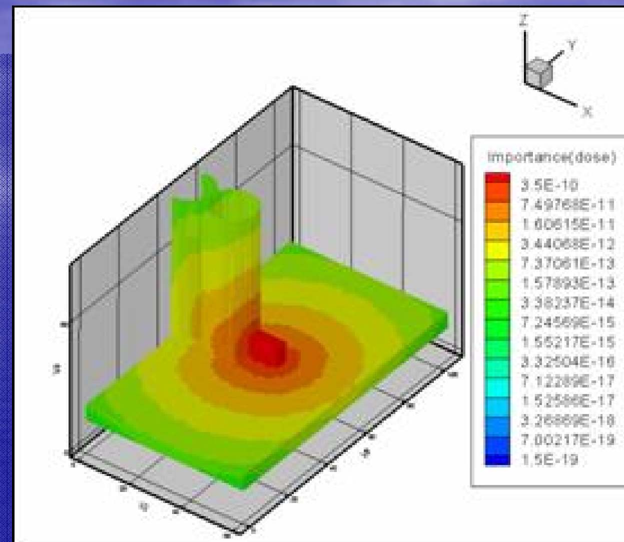




Gamma Transport Example

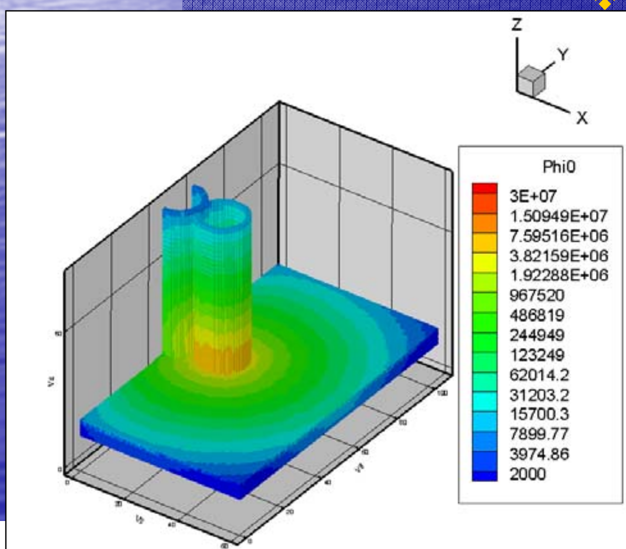


0.5 MeV forward

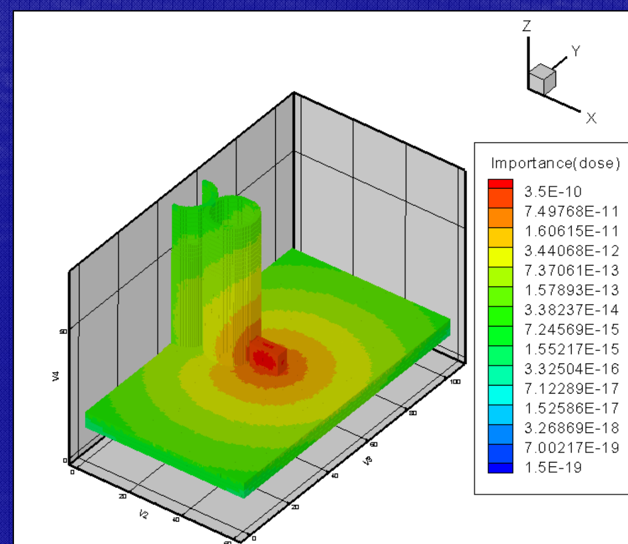


0.5 MeV adjoint

(relative to detector response)



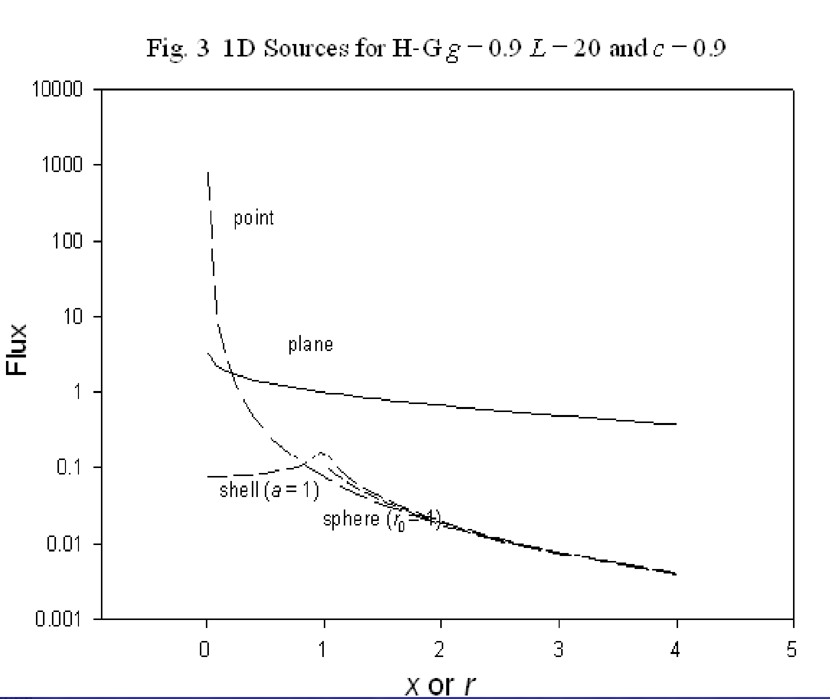
0.15 MeV forward

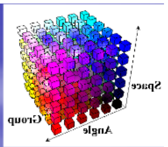


0.15 MeV adjoint

“TIEL” Benchmarks Tested the Numerics…

- Proposed by Ganapol
  - Ganapol, JBC-M&C Division, 2006, Source Series I
- Goal: provide a set of analytical benchmark problems, solutions
  - infinite media, Variety of sources
    - Plane, Sphere, Shell, Point…
- Well documented and solved using quasi-analytic methods
  - Fourier Transform inversion
  - Henyey-Greenstein (H-G) scattering kernel $\sigma_l = c\, g^{\,l}$, Up to L=24 Scattering Order
- Advantages
  - Can be unequivocally used as standards of comparison
  - Enable direct assessment of the quality of solutions and methodologies





Planar Source Problem

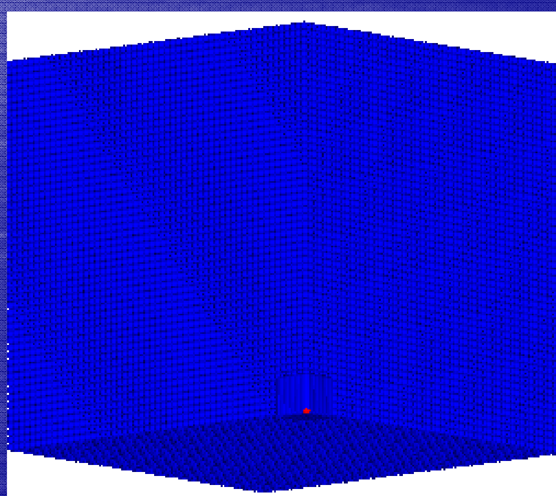
- Difficulties close in near source (also noted by Ganapol) , accuracy affected by extent of the boundary in the model ->
- Scalar flux solution at *x*=0.01 mfp (close in) for the plane source:
  - Effect of the S_N order for *c* =0.9, *g* =0.9, *L*=12, ε=10-3
  - Up to S_{64} (4224 directions/mesh)

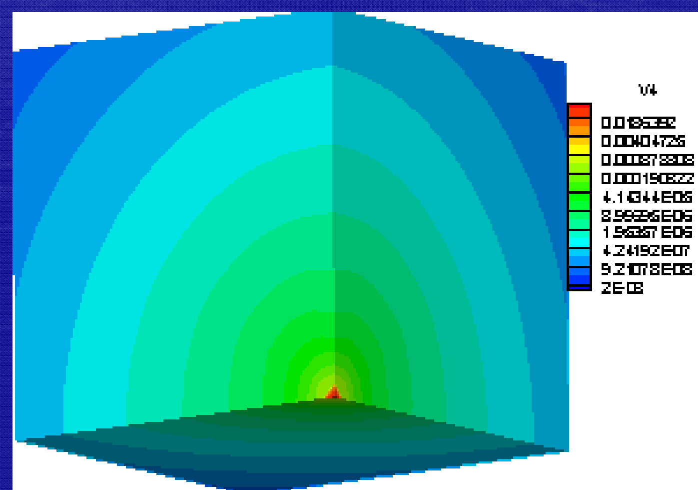
x (*mfp*)	PENTRAN-S_{34}	Ref[1]	Rel_diff (%)
0.01	**3.05371**	**3.29089**	7.20718
1.12	**0.9458**	**0.94499**	-0.08529
2.23	**0.62624**	**0.62603**	-0.03353
3.34	**0.45239**	**0.45228**	-0.02454
4.45	**0.33914**	**0.33908**	-0.01698
5.56	**0.25937**	**0.25937**	-0.00001
6.67	**0.20073**	**0.20081**	0.03984
7.78	**0.15651**	**0.15671**	0.13081
8.89	**0.12256**	**0.12295**	0.31233
10.00	**0.09618**	**0.09681**	0.65125

	S_{34}	S_{44}	S_{54}	S_{64}	Ref[1]
Scalar flux	3.05371	3.17748	3.2678	3.33902	3.29089
Rel_diff (%)	7.20718	3.44629	0.70175	-1.46259	-

Point Source

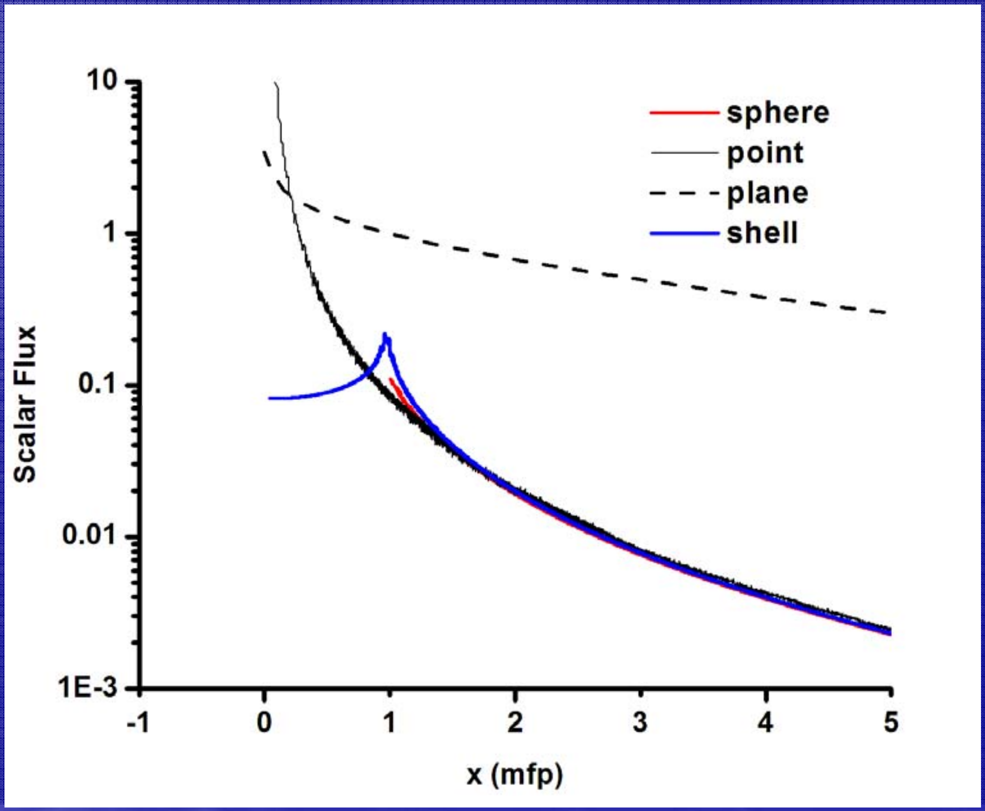
- The most challenging problem for a 3-D code
- Isotropic source in a highly anisotropic medium
  - Modeled as a 12 x 12 x 12 mfp box divided in 86,000 fine cells of 5 mm (each side) cells, except for center region (0.5 mm)
  - First coarse mesh: Source (origin) as one single fine mesh cell of 0.5 mm size

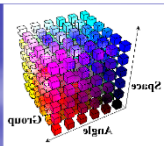




PENTRAN Solution Summary

- **Scalar flux solutions for the 1-D sources: c=0.9, g=0.9, L=12**

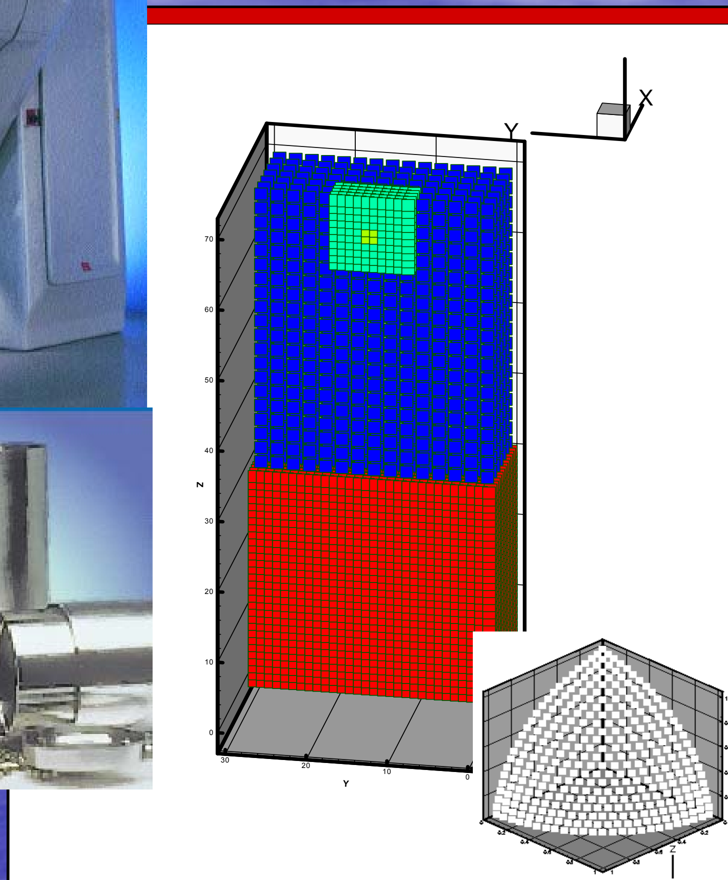




TIEL Benchmark Findings

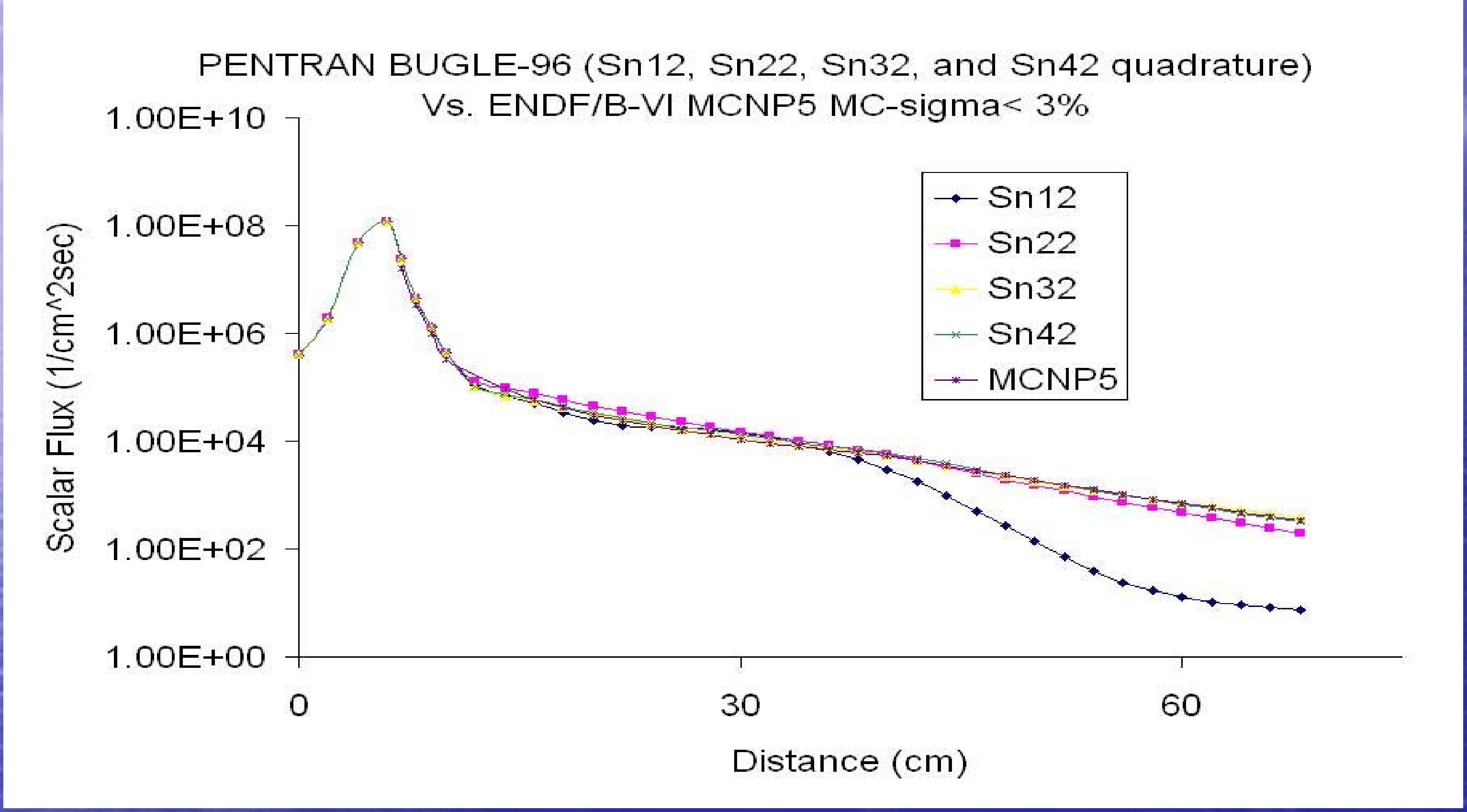
- PENTRAN demonstrated Good Agreement with benchmarks
  - Solutions accurate to < 0.7% difference
  - Demonstrated accuracy and asymptotic convergence
- Verified arbitrary $S_N P_L$ formulations in code
  - 3-D Legendre-Chebychev quadratures to N=64 (4224 dir) here
  - 3-D Anisotropic scattering to L=24
- Other Issues
  - EDI vs EDW differencing algorithms; EDI better adapts to various scattering situations
  - Verified Angular-Spatial Parallel numerics of PENTRAN
    - Importance of parallel computing to yield results in 3-D codes
  - Excellent learning exercise for students
  - *Confidence in solutions as we develop other supporting tools & methods*

Simplified Co-60 Problem

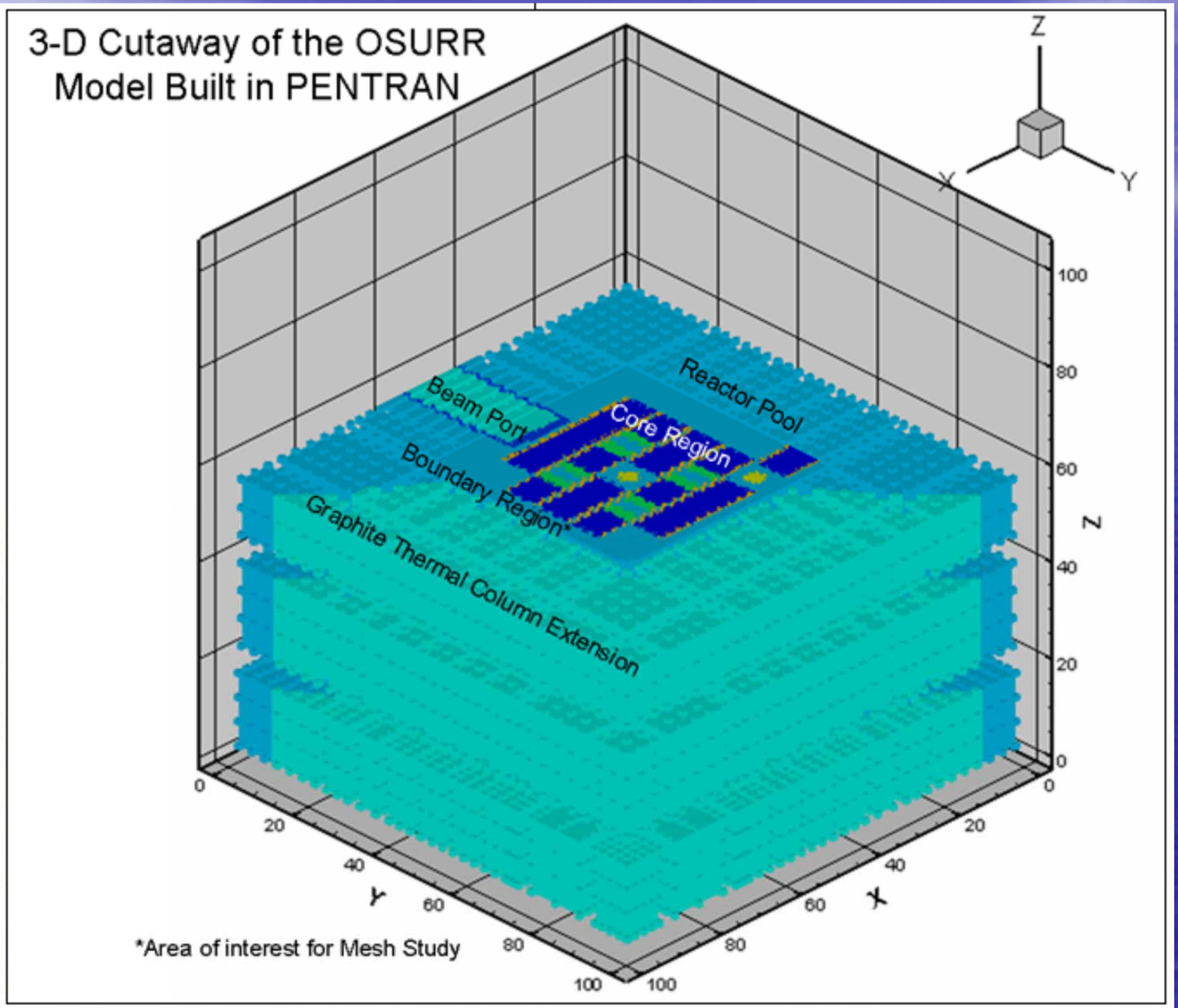


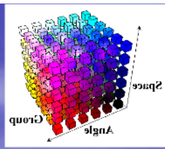
- Idealized model of a teletherapy device for benchmark applications
- This model consisted of four key components:
- source (2×2×2 cm3)
- lead source housing 4 cm thick, air layer
- homogeneous water phantom (30×30×30 cm3)
- P3-S42 Discrete Ordinates (1848 directions/ mesh/energy group)

Investigation of Angular Quadrature



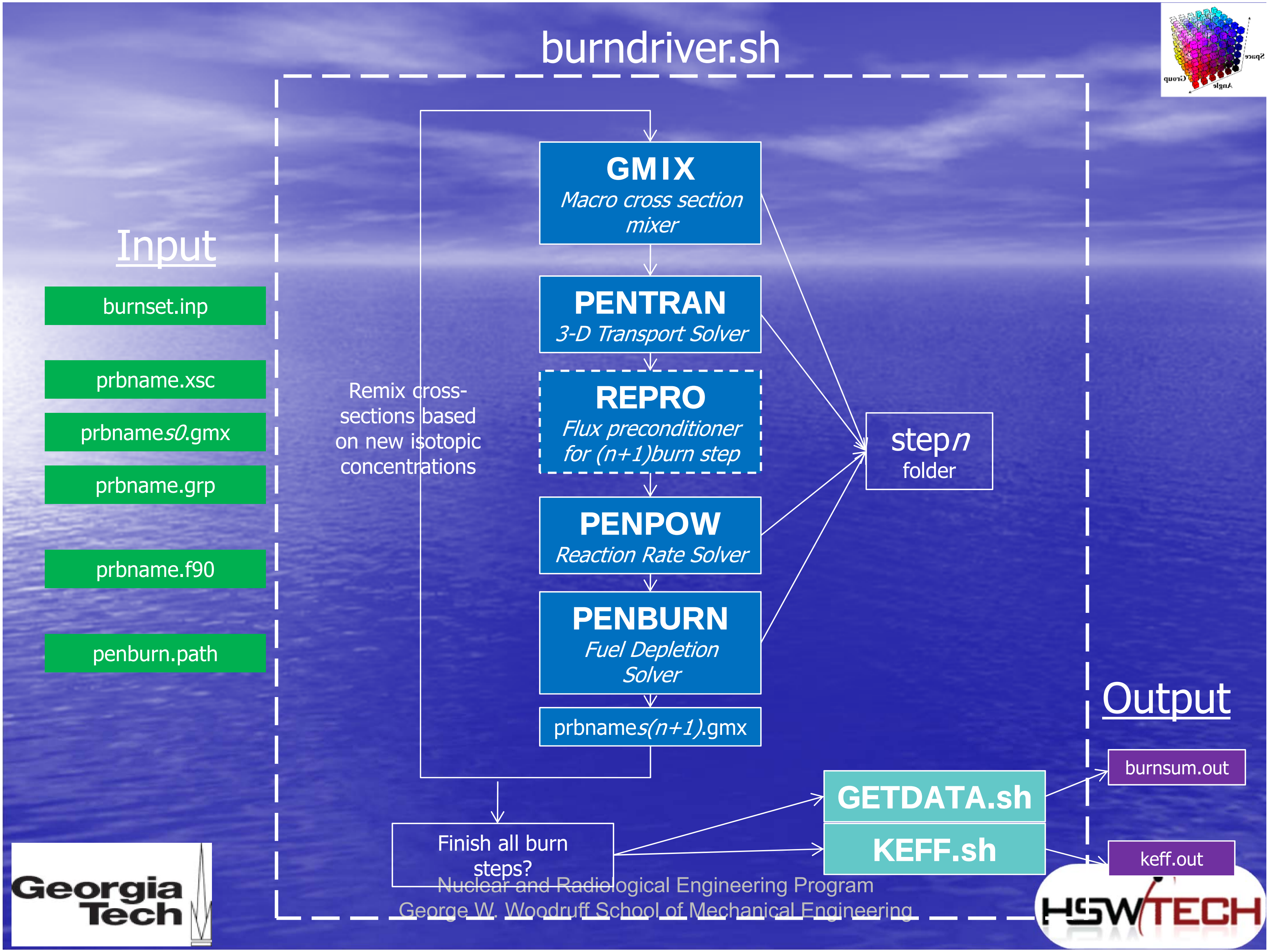
OSU Reactor Model

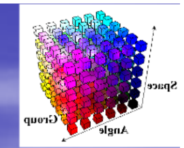




PENBURN with ‘BURNDRIVER’ methodology

- “Forward-Euler” approach - fluxes solved at the beginning of each time step are assumed to be constant over step.
  - Considerations are currently being made to consider a mid-step depletion/transport calculation to forecast mid-step fluxes.
- Infinite dilution cross-section treatment – microscopic cross sections are assumed to remain constant through burnup.
  - Dehart has shown this assumption begins to fail when in-growth reaches $1x10^{-3}$ atom/bn-cm.
  - Cross-section updates need to be considered when reaching this large of in-growth.

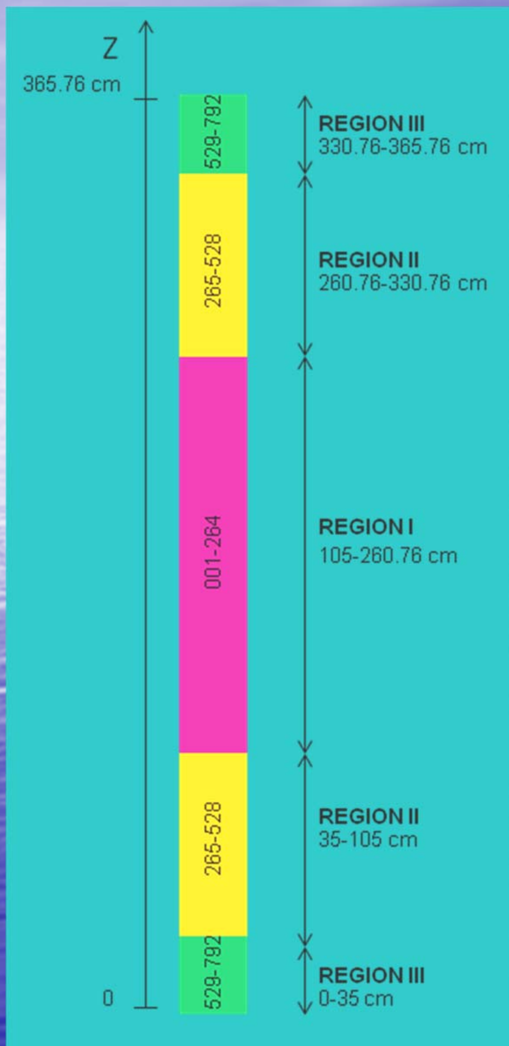

burndriver.sh
Input
burnset.inp
prbname.xsc
prbname*s0*.gmx
prbname.grp
prbname.f90
penburn.path
GMIX
Macro cross section mixer
PENTRAN
3-D Transport Solver
REPRO
Flux preconditioner for (n+1)burn step
PENPOW
Reaction Rate Solver
PENBURN
Fuel Depletion Solver
prbname*s(n+1)*.gmx
Remix cross-sections based on new isotopic concentrations
step*n* folder
Finish all burn steps?
GETDATA.sh
KEFF.sh
Output
burnsum.out
keff.out
Georgia Tech
HSW TECH
Nuclear and Radiological Engineering Program
George W. Woodruff School of Mechanical Engineering

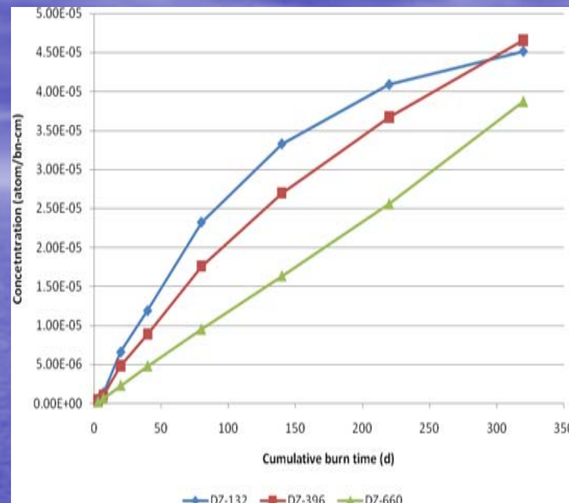


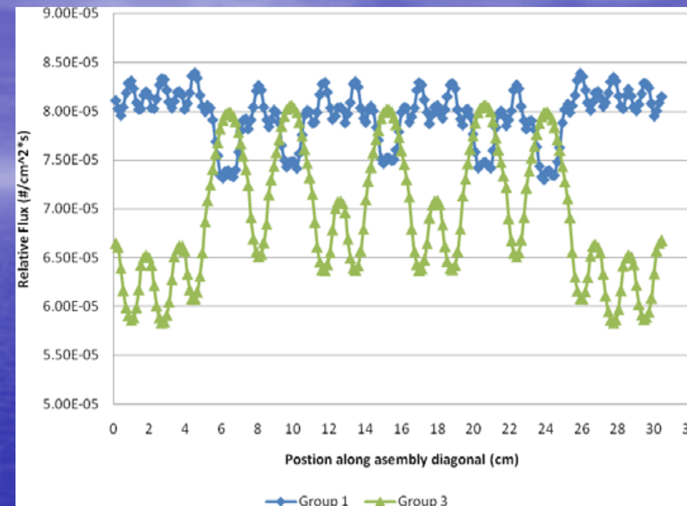
PENTRAN Westinghouse Assembly Calculation

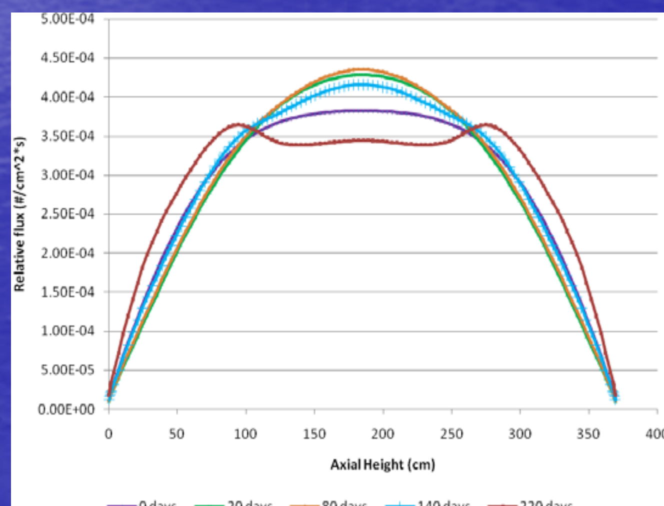
- Full length, 17x17 PWR assembly (w/out control rod fingerlings) is currently modeled in PENTRAN.
  - Active length: 365.76 cm
  - Assembly pitch: 21.5 cm
  - Total PENTRAN meshes: 4.5×10^6
- Homogenized fuel pins (fuel/gap/clad).
- Each fuel pin modeled as unique material
  - 3*264 fuel materials + 1 water material = 793 total
  - Allows for pin-by-pin depletion in PENBURN

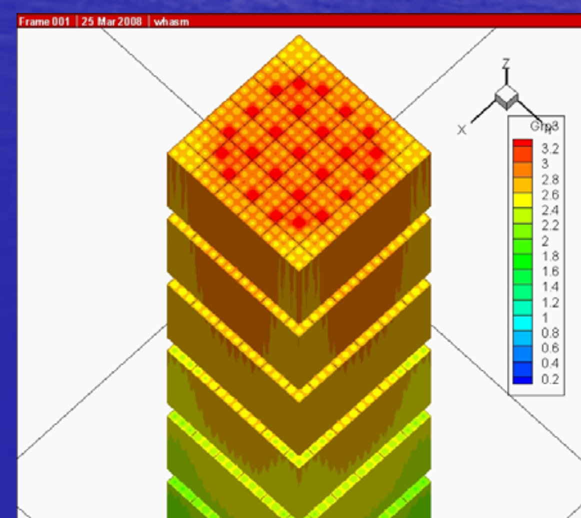
Westinghouse Assembly Model





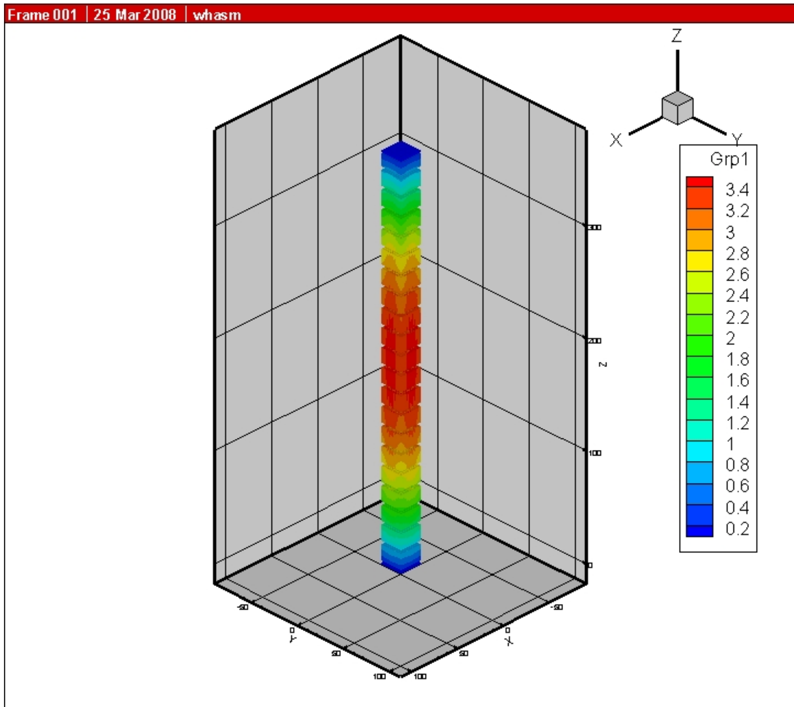




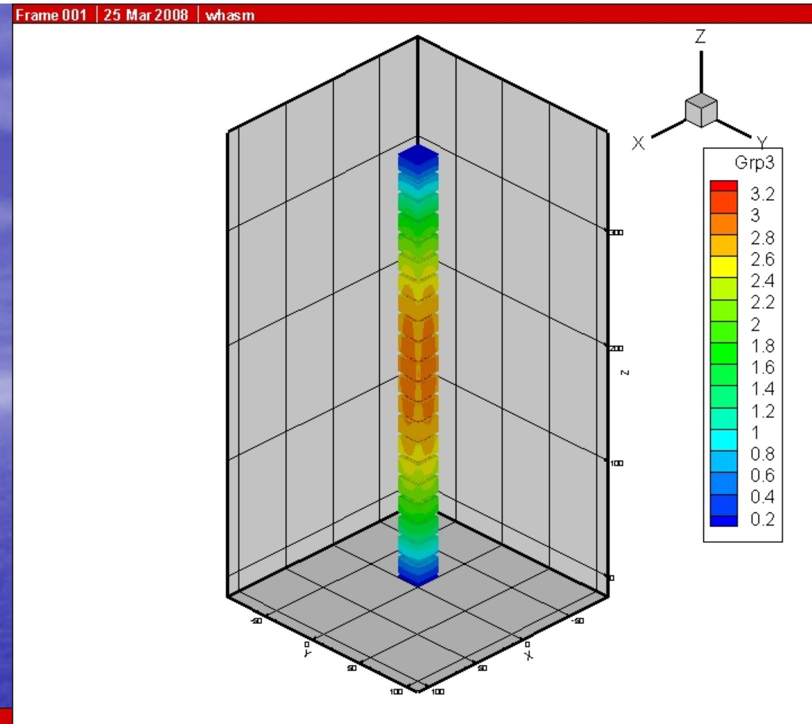


Group 3
(E <

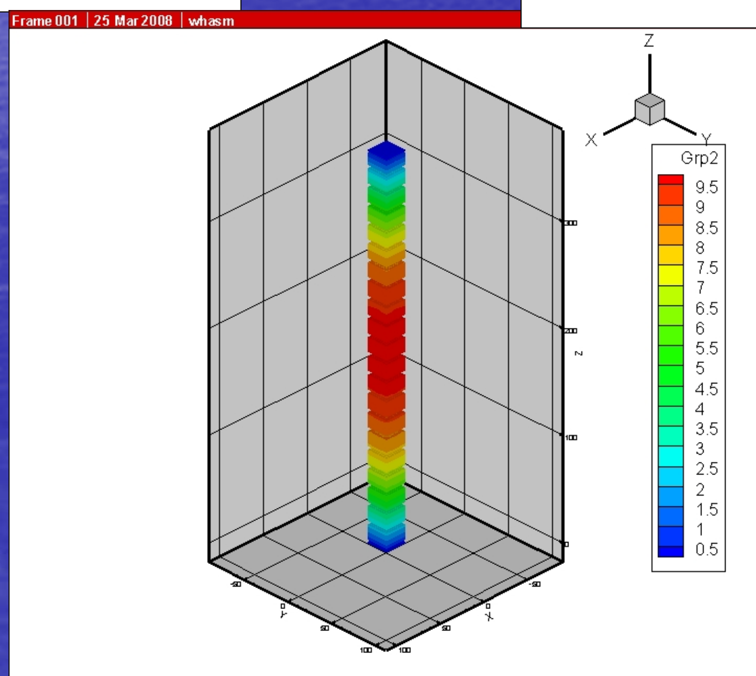
Nuclear and Radiological Engineering Program
George W. Woodruff School of Mechanical Engineering



Group 1
(1 MeV < E < 20 MeV)



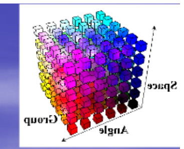
Group 3
(E < 0.625 eV)



Group 2 (0.625 eV<E<1 MeV)

SFCOMPO Database

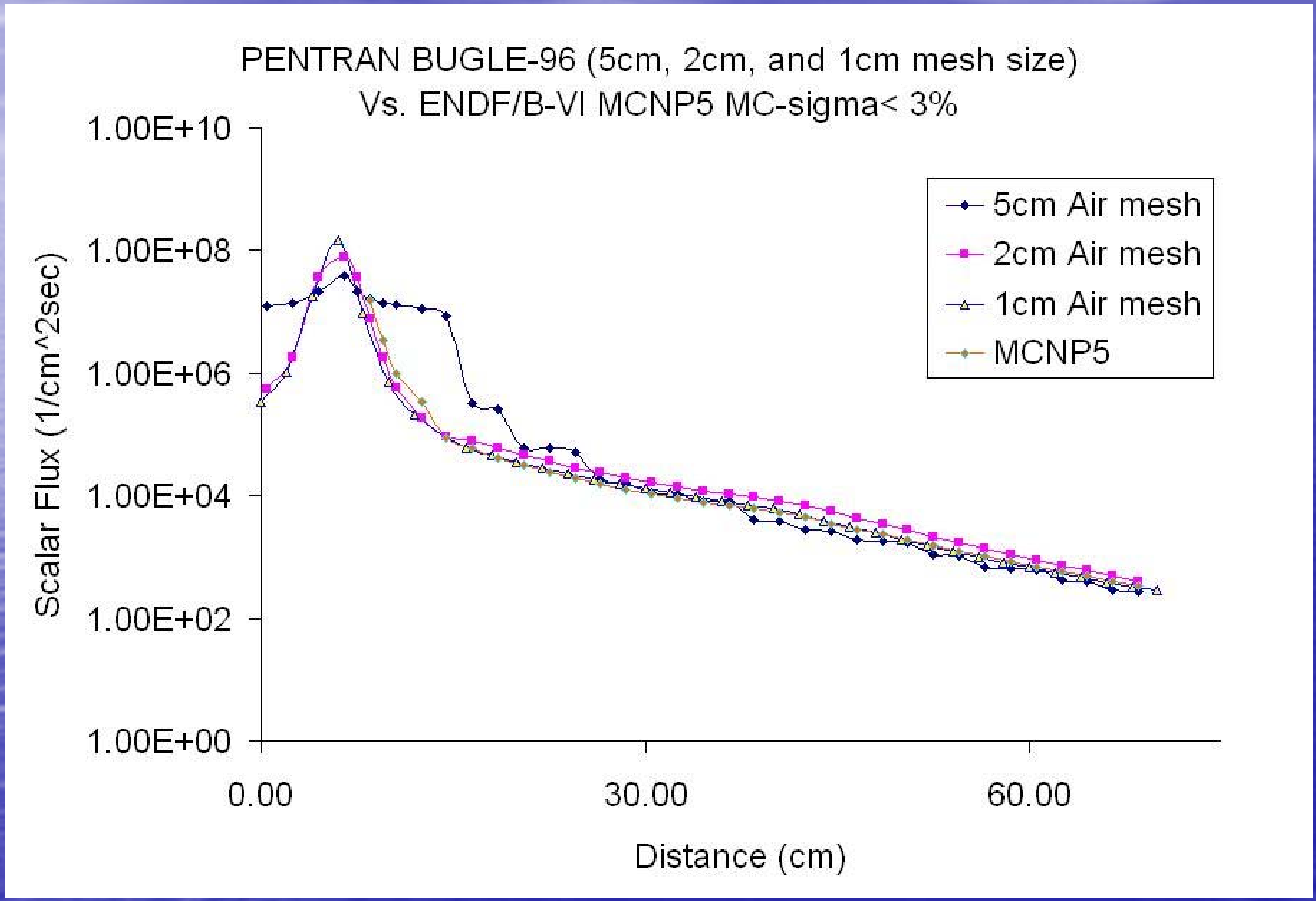
- SFCOMPO: Spent Fuel COMPosition Database
  - Found at *http://www.nea.fr/sfcompo/*
- Database which archives isotopic compositions of spent fuel assemblies pulled from power reactors.
- Determined from gamma spectrometry and chemical analysis.
- Originally developed by the JAERI Department of Fuel Cycle Safety Research's Fuel Cycle Safety Evaluation Laboratory and later transferred to the NEA.



SFCOMPO (cont'd)

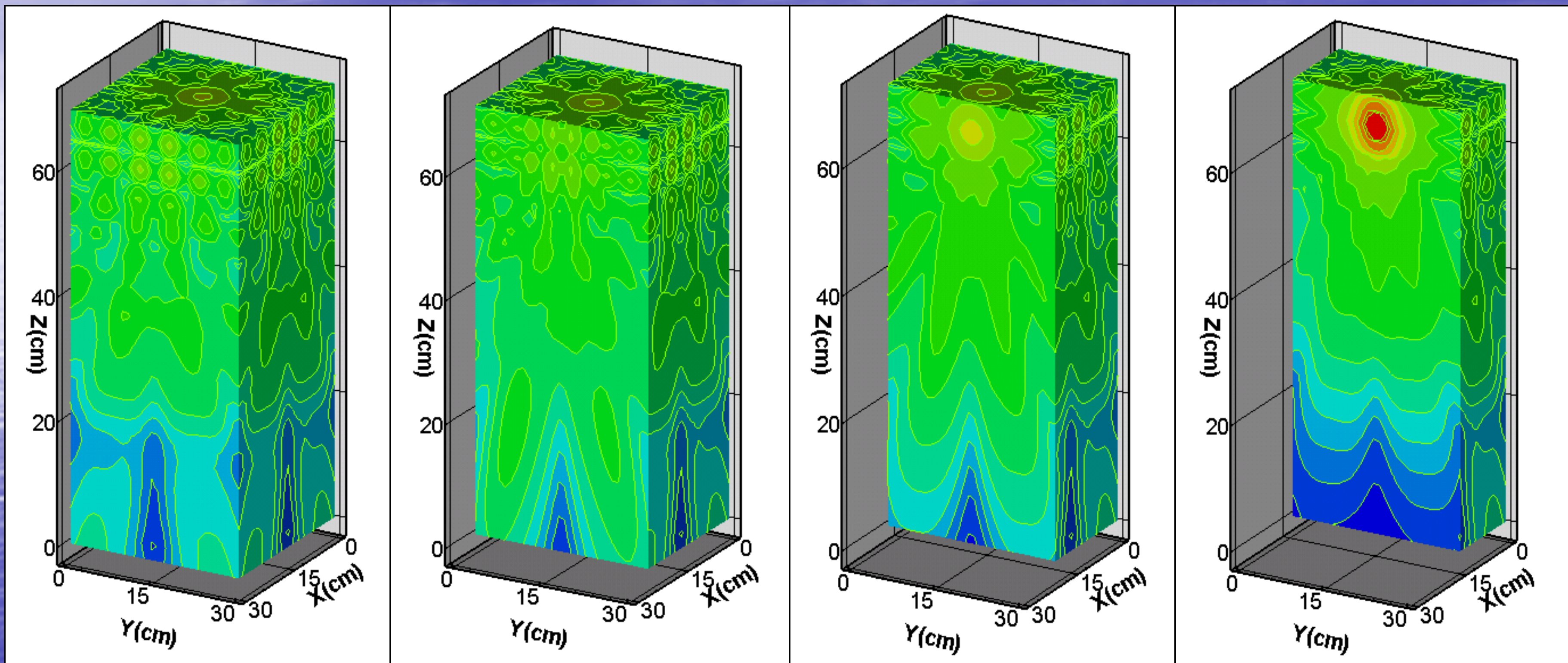
- Contains data for PWR and BWRs from Germany, Italy, and Japan.
- Provides operation history for fuel element, and even shows position in core at each irradiation step.
- Website includes search engine which allow the user to search based on a reactor name or range of burnup, initial enrichment, and cooling time after irradiation.
- Good alternative to comparisons with SCALE simulations.

Investigation of Mesh Interval



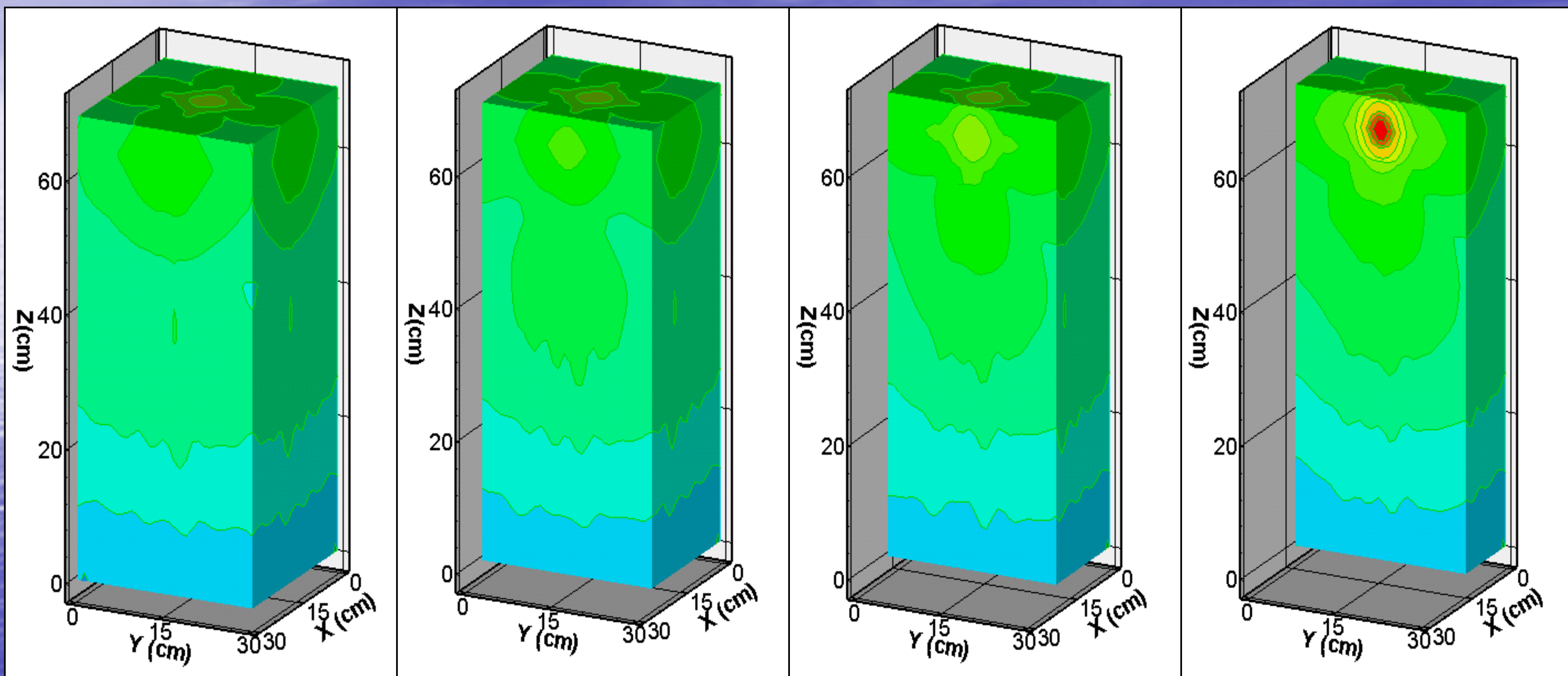
Mesh variations in the Co-60 model

Co-60 Device 3-D scalar flux distributions slices model at different x-levels with S12



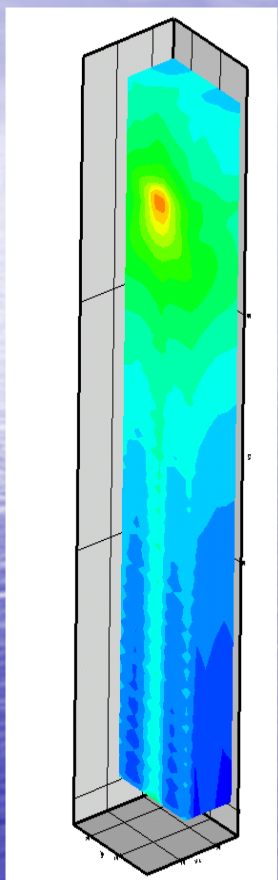
gamma-ray 3-D Group 1 scalar flux distribution for depth 0cm, 5cm, 10 cm and 15 cm, computed by PENTRAN with the BUGLE-96 gamma cross section library, (An S12 angular quadrature,168 directions, with P3 scattering anisotropy was used).

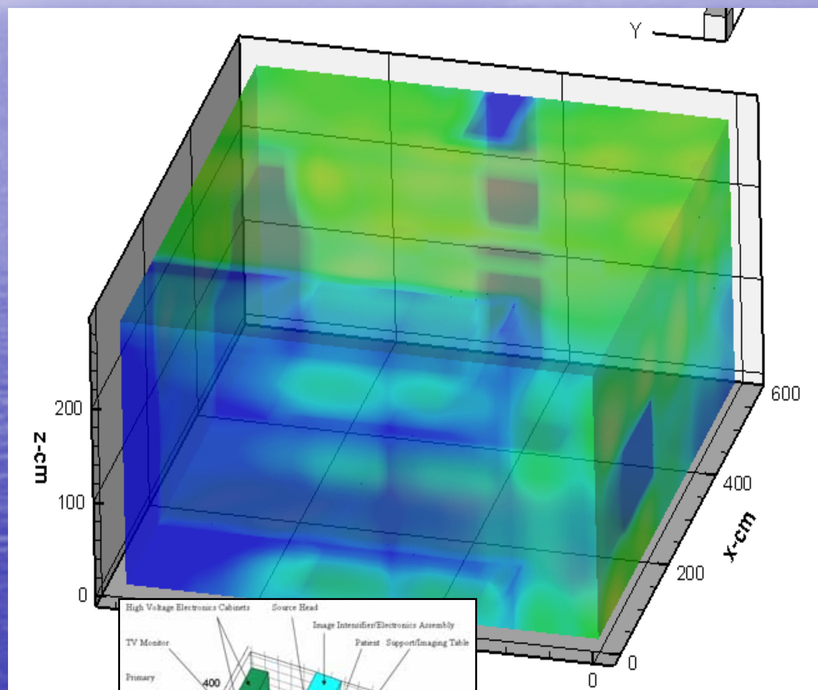
Co-60 Device 3-D scalar flux distributions slices model at different x-levels with S42

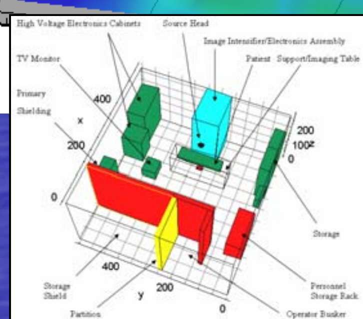


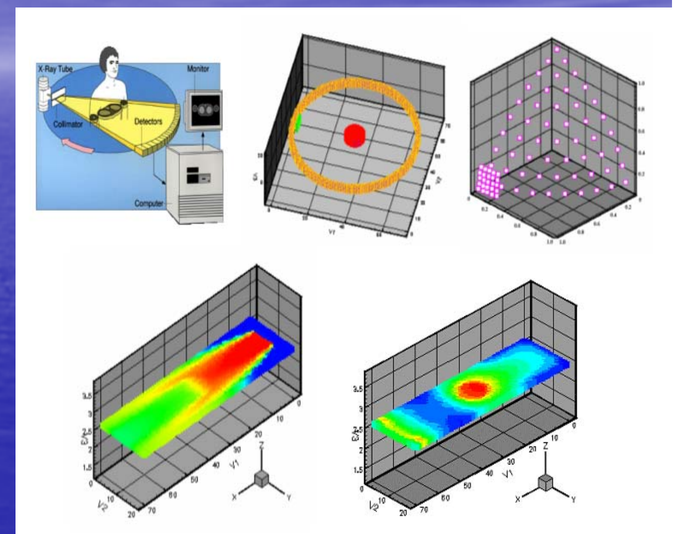
Gamma-ray 3-D Group 1 scalar flux distribution for x-level depths at 0cm, 5cm, 10 cm and 15 cm, computed by PENTRAN with the BUGLE-96 gamma cross section library; an S42 angular quadrature (1848 directions) with P3 scattering anisotropy as a reference case.

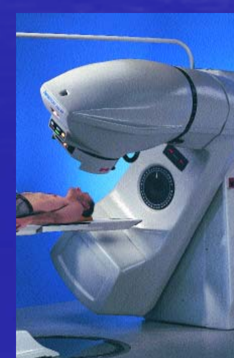
Significant efforts in Shielding calculations…

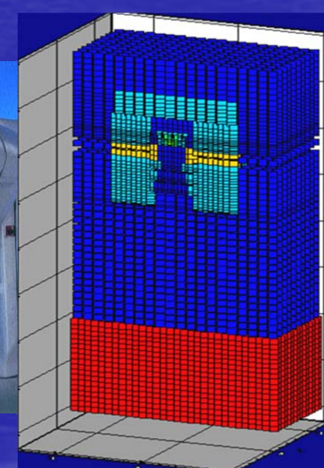












BURNDRIVER

- PENBURN sequence is now automated with the release of BURNDRIVER shell script.
- Script creates all necessary directories, moves inputs/outputs, executes programs, and creates output summary files.
- Requires 6 input files:
  1. **burnset.inp** 2. prbname.gmx
  3. penburn.path 4. prbname.xsc
  5. prbname.f90 6. prbname.grp

SCALE5.1 T-NEWT
Micro cross-section generator

DEV-XS
GMIXFORM and COLLAPSEFORM

PENMSHXP
PENTRAN geometry/input generator

prbname*s0*.gmx

prbname.grp

SCALFORM

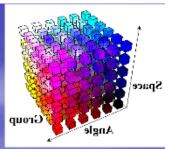
prbname.f90

prbname.xsc

penburn.path

burnset.inp

Sources of input files.



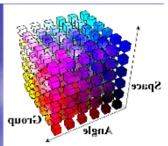
burnset.inp

```
/ problem name
whpin
/ # of processors for parallel PENTRAN run
8
/ Following 3 lines dedicated to PENPOW Problem Description
   Westinghouse Hetr Pin Burnup Study
   17 x 17 OFA Assembly
   Fuel Gap Clad Moderator Regions
/ number of fuel materials, no. nonburn nuclides
1
/ range of fuel material number: eg. 1 10
1 1
.
.
.
```

burnset.inp

```
/ Following 3 lines dedicated to PENBURN Problem Description
Westinghouse Hetr Pin Burnup Study
17 x 17 OFA Assembly
Fuel Gap Clad Moderator Regions
/ # of irradiation/cool steps
2
/ power, time, time unit (d), irrad (i)/cool(c), print step, print option, GMIX keyword
-30 1 d i 1 2 s1
-30 3 d i 1 2 s2
```

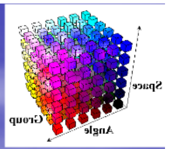
*Very familiar to penburn.inp



INTERP-XS

- Purpose: Creates a problem dependent microscopic cross-section file based on specified:
  - Avg. system energy (MeV)
  - Operating temperature (K)
  - Burnup (GWd/MTU)
- Microscopic cross-section libraries generated for three reactor systems:
  - Thermal: PWR WH fuel pin
  - Epithermal: Graphite moderated Magnox fuel pin
    - Loosely based off of UFCD II design
  - Fast: Clinch River Breeder Reactor fuel pin
    - Hexagonal lattice, Na cooled

NOTE: Spectra characterized by MCNP analysis of unit cell's neutron density distribution.



INTERP-XS (cont'd)

- System temperatures: 300, 650, 1000, 1350 K
- Libraries currently exist for fresh fuel concentrations
    - Libraries for mid-cycle fuel compositions are work in progress.
- 2 types of interpolation methods:
    - Linear interpolation
        - Option for using lethargy, ln(Eo/E), instead of energy, E