Criticality Calculations

Criticality Calculations - Part I

- Criticality Overview
- Criticality Estimators
- KCODE & KSRC Cards
- Examples & Output

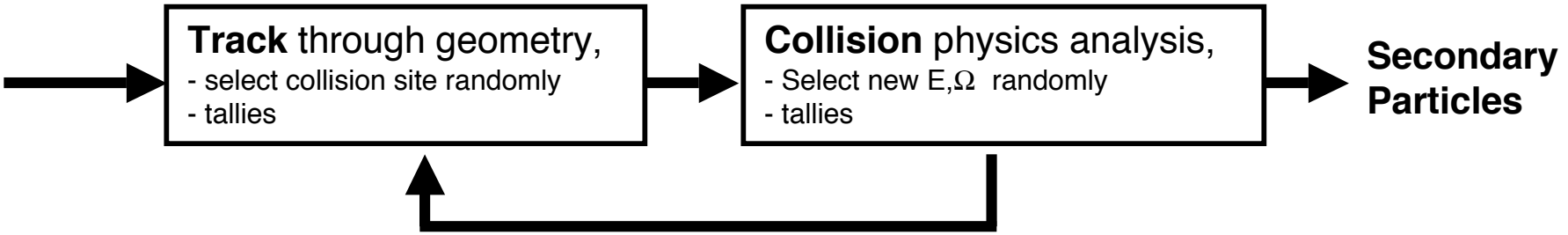
Criticality Calculations - Part II

- Convergence
- Plotting
- Examples & Plots
- Continue Runs
- Burnup

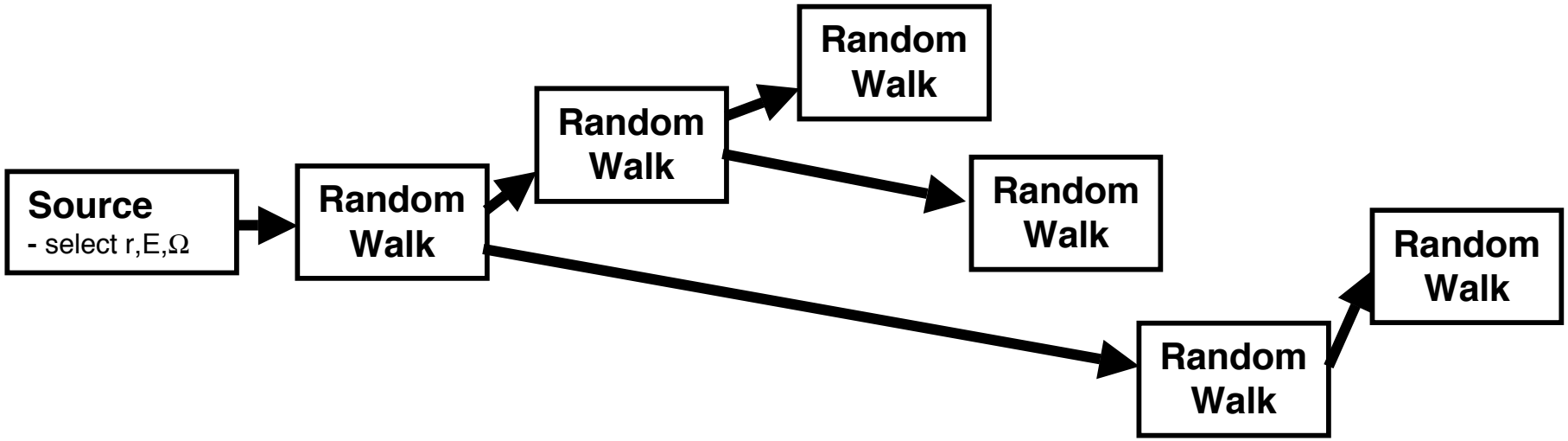
Criticality Overview

Particle Histories

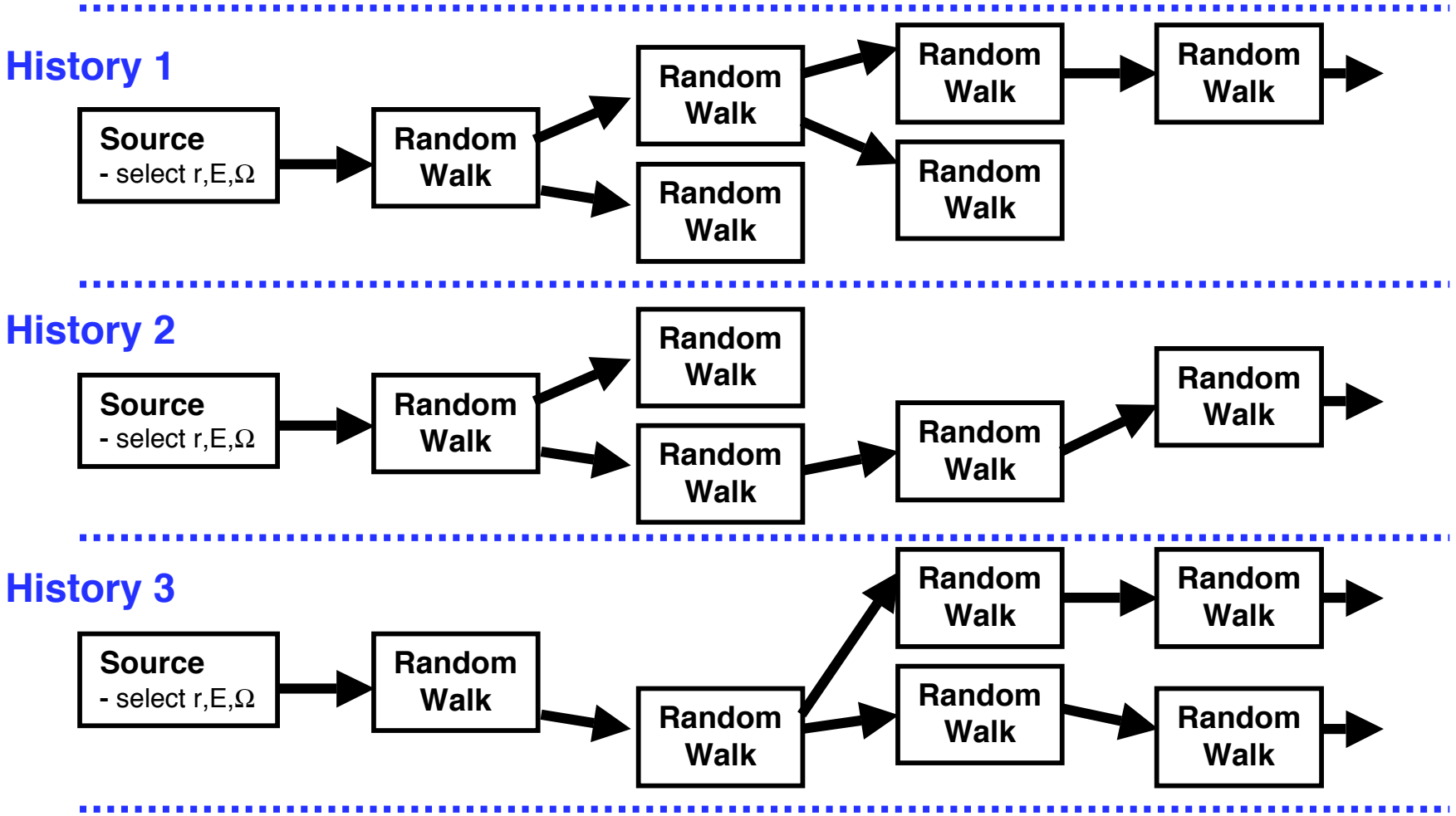
- **Random Walk for particle**



- **Particle History**



Fixed-source Monte Carlo Calculation



K-eigenvalue equation

$$\left[\vec{\Omega}\cdot\nabla+\Sigma_T(\vec{r},E)\right]\Psi(\vec{r},E,\vec{\Omega}) = \iint \Psi(\vec{r},E',\vec{\Omega}')\Sigma_S(\vec{r},E'\rightarrow E,\vec{\Omega}\cdot\vec{\Omega}')d\vec{\Omega}'dE'$$
$$+ \quad \frac{1}{K_{eff}}\cdot\frac{\chi(E)}{4\pi}\iint \nu\Sigma_F(\vec{r},E')\Psi(\vec{r},E',\vec{\Omega}')d\vec{\Omega}'dE'$$

where

K_{eff} = k-effective, eigenvalue for fundamental mode

$\Psi(\vec{r},E,\vec{\Omega})$ = angular flux, for fundamental k-eigenmode

$\vec{\Omega}\cdot\nabla\Psi(\vec{r},E,\vec{\Omega})$ = **loss** term, leakage

$\Sigma_T(\vec{r},E)\,\Psi(\vec{r},E,\vec{\Omega})$ = **loss** term, collisions

$\iint \Psi(\vec{r},E',\vec{\Omega}')\Sigma_S(\vec{r},E'\rightarrow E,\vec{\Omega}\cdot\vec{\Omega}')d\vec{\Omega}'dE'$ = **gain** term, scatter from E',Ω' into E,Ω

$\frac{1}{K_{eff}}\cdot\frac{\chi(E)}{4\pi}\iint \nu\Sigma_F(\vec{r},E')\Psi(\vec{r},E',\vec{\Omega}')d\vec{\Omega}'dE'$ = **gain** term, production from fission

⇒ Jointly find K_{eff} and $\Psi(r,E,\Omega)$ such that equation balances

K_{eff} Eigenvalue Equations

- K-eigenvalue equation

$$\left[\vec{\Omega}\cdot\nabla+\Sigma_T(\vec{r},E)\right]\Psi(\vec{r},E,\vec{\Omega}) = \iint \Psi(\vec{r},E',\vec{\Omega}')\Sigma_S(\vec{r},E'\rightarrow E,\vec{\Omega}\cdot\vec{\Omega}')d\vec{\Omega}'dE'$$
$$+ \quad \frac{1}{K_{eff}}\cdot\frac{\chi(E)}{4\pi}\iint \nu\Sigma_F(\vec{r},E')\Psi(\vec{r},E',\vec{\Omega}')d\vec{\Omega}'dE'$$

- This is a **static** equation, an **eigenvalue problem** for K_{eff} and Ψ without time-dependence
- **K_{eff} is called the effective multiplication factor**
- **Criticality**

Supercritical:	**$K_{eff} > 1$**
Critical:	**$K_{eff} = 1$**
Subcritical:	**$K_{eff} < 1$**

- **Never use K_{eff} and Ψ to model time-dependent problems.**
- **K_{eff}-eigenvalue problems can be solved by Monte Carlo methods**

K-eigenvalue equation

- **Use operator (or matrix) form to simplify notation**

$$(L + T)\Psi = S\Psi + \frac{1}{K_{eff}} M\Psi$$

where

L = leakage operator S = scatter-in operator

T = collision operator M = fission multiplication operator

- **Rearrange**

$$(L + T - S)\Psi = \frac{1}{K_{eff}} M\Psi$$

$$\Psi = \frac{1}{K_{eff}} \cdot (L + T - S)^{-1} M\Psi$$

$$\Psi = \frac{1}{K_{eff}} \cdot F\Psi$$

⇒ This eigenvalue equation will be solved by power iteration

Power Iteration

Eigenvalue equation

$$\Psi = \frac{1}{K_{eff}} \cdot F\Psi$$

1. **Assume that k_{eff} and Ψ on the right side are known for iteration n, solve for Ψ on left side (for iteration n+1)**

$$\Psi^{(n+1)} = \frac{1}{K_{eff}^{(n)}} \cdot F\Psi^{(n)}$$

Note: This requires solving the equation below for $\Psi^{(n+1)}$, with $K_{eff}^{(n)}$ and $\Psi^{(n)}$ fixed

$$(L + T - S)\Psi^{(n+1)} = \frac{1}{K_{eff}^{(n)}} M\Psi^{(n)}$$

2. **Then, compute $K_{eff}^{(n+1)}$**

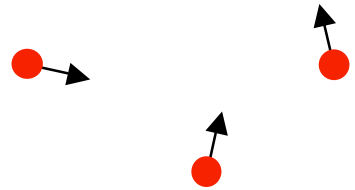
$$K_{eff}^{(n+1)} = K_{eff}^{(n)} \cdot \frac{\int M\Psi^{(n+1)} d\vec{r}}{\int M\Psi^{(n)} d\vec{r}}$$ (other norms could be used)

Power Iteration

- Power iteration procedure:

1. **Initial guess for K_{eff} and Ψ**
   $K_{eff}^{(0)}$, $\Psi^{(0)}$

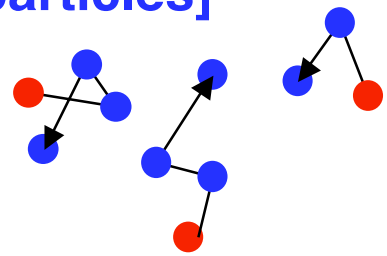
Source points for $\Psi^{(0)}$



2. **Solve for $\Psi^{(n+1)}$ [Monte Carlo random walk for N particles]**

$$\Psi^{(n+1)} = \frac{1}{K_{eff}^{(n)}} \cdot F\Psi^{(n)}$$

Source points for $\Psi^{(n+1)}$

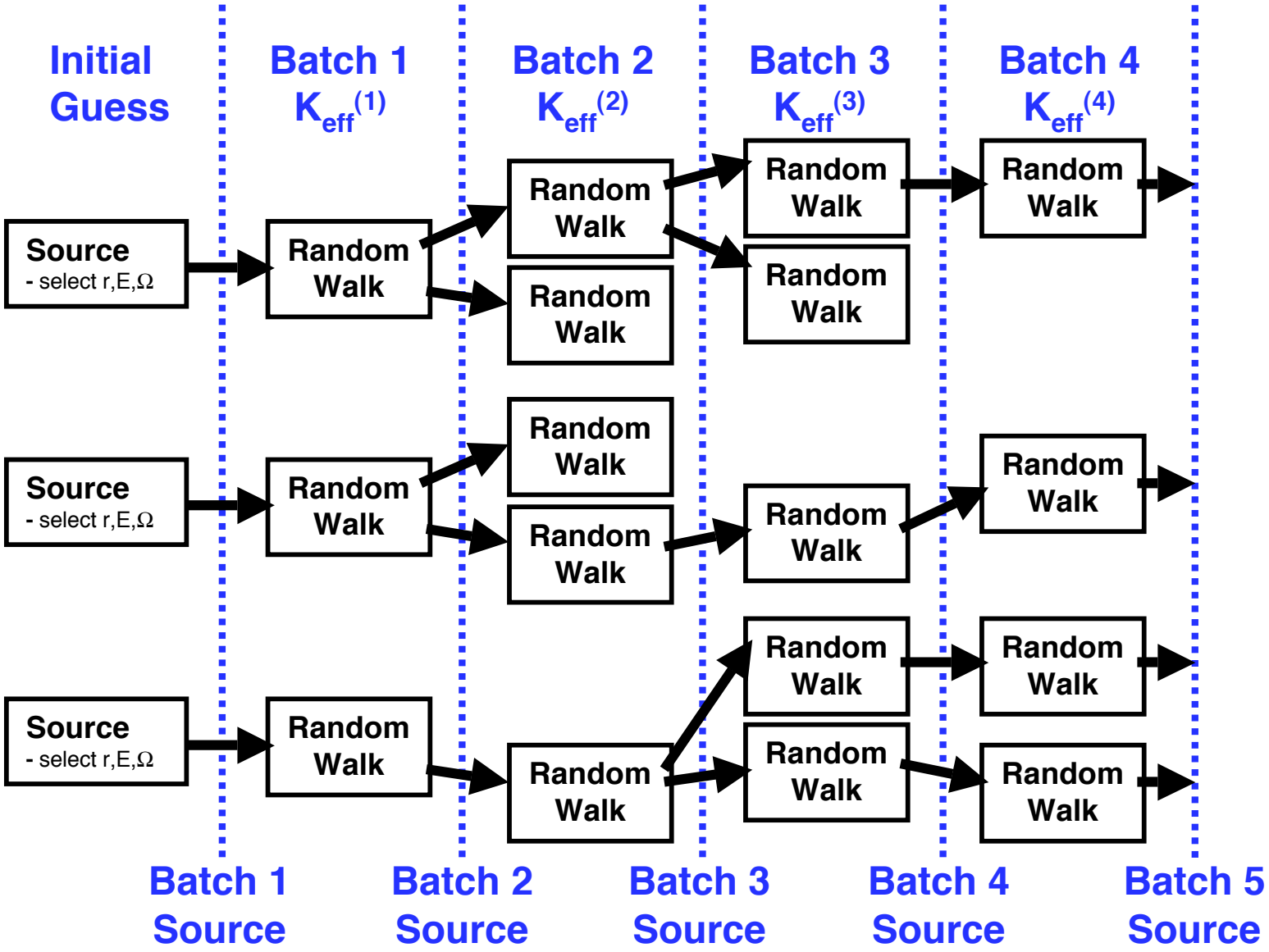


3. **Compute new K_{eff}**

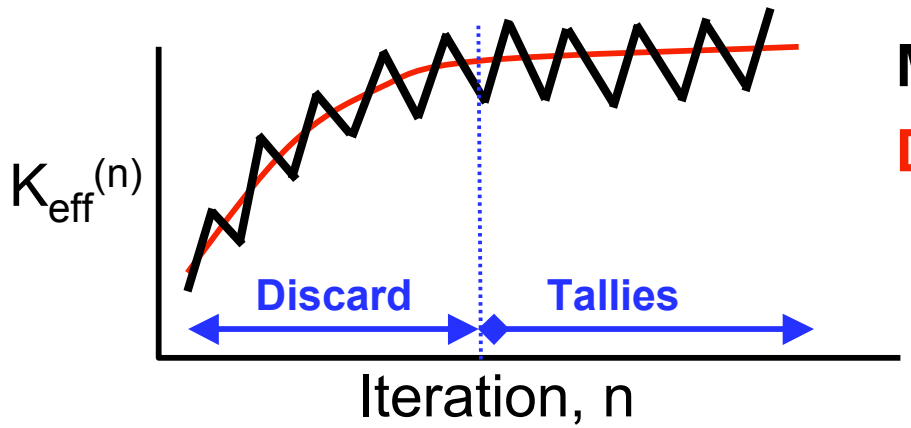
$$K_{eff}^{(n+1)} = K_{eff}^{(n)} \cdot \frac{\int M\Psi^{(n+1)} d\vec{r}}{\int M\Psi^{(n)} d\vec{r}}$$

4. **Repeat 1-3 until both $K_{eff}^{(n+1)}$ and $\Psi^{(n+1)}$ have converged**

Monte Carlo Eigenvalue calculation



Power Iteration



Monte Carlo

Deterministic (S_n)

- Guess an initial source distribution
- Iterate until converged (How do you know ???)
- Then
  - **For S_n code: done, print the results**
  - **For Monte Carlo: start tallies, keep running until uncertainties small enough**

- Convergence? Stationarity? Bias? Statistics?

MCNP Criticality Flow

- **Histories are run in batches of N particles**
- **Spatial distribution:**
  - Initial distribution of fission source sites from KSRC, SDEF, or SRCTP card
  - After first batch, fission source sites taken from previous batch
- **Total weight in each batch is N**
- **For each batch, 3 estimates of Keff are made for the batch**
  - Keff **track-length** estimator
  - Keff **collision** estimator
  - Keff **absorption** estimator
- **At the end of the problem, the batch estimates are combined into 7 overall estimates (only the last one matters):**
  - 3 cumulative Keff estimates using track-length, collision, & absorption
  - 3 cumulative Keff estimates using pairs
  - **1 overall combined cumulative estimate based on all data**

K-effective Estimators

Background on Monte Carlo Estimators

- **Pathlength estimator for flux**
  - Flux ≡ total pathlength traveled by all neutrons per unit volume per unit time
  - For flux in a cell,

$$\phi \approx \frac{1}{W \cdot V} \sum_{\substack{\text{all flights} \\ \text{in cell}}} \mathbf{d_k} \cdot \mathbf{wgt}$$

V = cell volume

W = total starting weight

- **Collision estimator for flux**
  - Collision rate = $\Sigma_T \phi$, so ϕ = [collision rate]/ Σ_T
  - For flux in cell,

$$\phi \approx \frac{1}{W \cdot V} \sum_{\substack{\text{all collisions} \\ \text{in cell}}} \frac{\mathbf{wgt}}{\Sigma_T}$$

- **Absorption estimator for flux**
  - Absorption rate = $\Sigma_A \phi$, so ϕ = [absorption rate]/ Σ_A
  - For flux in cell,

$$\phi \approx \frac{1}{W \cdot V} \sum_{\substack{\text{all absorptions} \\ \text{in cell}}} \frac{\mathbf{wgt}}{\Sigma_A}$$

Single-cycle Keff Estimators

Neutron production rate = $\nu\Sigma_F \phi$

W = total weight starting cycle n

- **Pathlength estimator for Keff, for cycle n**

$$K^{(n)}_{path} = \left(\sum_{\substack{\text{all} \\ \text{flights}}} wgt_j \cdot d_j \cdot \nu\Sigma_F \right) \Big/ W$$

- **Collision estimator for Keff, for cycle n**

$$K^{(n)}_{collision} = \left(\sum_{\substack{\text{all} \\ \text{collisions}}} \frac{wgt_j}{\Sigma_T} \cdot \nu\Sigma_F \right) \Big/ W$$

- **Absorption estimator for Keff, for cycle n**

$$K^{(n)}_{absorption} = \left(\sum_{\substack{\text{all} \\ \text{absorptions}}} \frac{wgt_j}{\Sigma_A} \cdot \nu\Sigma_F \right) \Big/ W$$

Cumulative Eigenvalue Estimators

- **7 estimators of overall Keff -- $K_{p\text{-}c\text{-}a}$ is best**

1. K_{path} = average over all active cycles of $K^{(n)}_{path}$
2. $K_{collision}$ = average over all active cycles of $K^{(n)}_{collision}$
3. $K_{absorption}$ = average over all active cycles of $K^{(n)}_{absorption}$
4. $K_{p\text{-}c}$ = combined average over all active cycles of $K^{(n)}_{path}$ and $K^{(n)}_{collision}$, including correlation
5. $K_{p\text{-}a}$ = combined average over all active cycles of $K^{(n)}_{path}$ and $K^{(n)}_{absorption}$, including correlation
6. $K_{c\text{-}a}$ = combined average over all active cycles of $K^{(n)}_{collision}$ and $K^{(n)}_{absorption}$, including correlation
7. **$K_{p\text{-}c\text{-}a}$ = combined average over all active cycles of $K^{(n)}_{path}$, $K^{(n)}_{collision}$, and $K^{(n)}_{absorption}$, including correlation**

Confidence Intervals

- **Confidence interval:**

  Range that contains the true Keff with some specified probability

- **68% Confidence interval**

$$K_{eff} - \sigma \leq K_{true} \leq K_{eff} + \sigma$$

- **95% Confidence interval**

$$K_{eff} - 2\sigma \leq K_{true} \leq K_{eff} + 2\sigma$$

- **99% Confidence interval**

$$K_{eff} - 2.6\sigma \leq K_{true} \leq K_{eff} + 2.6\sigma$$

- **To get better confidence interval (smaller σ), run more histories (more cycles)**

$$\sigma \propto \frac{1}{\sqrt{N}}$$

KCODE & KSRC Cards

MCNP Criticality Input

- **Control for K-effective calculations**

  KCODE card

  – Number of particles per cycle
  – Initial guess for Keff
  – Number of initial cycles to skip
  – Total number of cycles to run

- **Guess for initial source** **(only use 1 of these)**

  KSRC card

  – Can specify any number of x,y,z points for initial location of fission neutrons

  SRCTP file

  – Can use a file of source points from a previous calculation

  SDEF card

  – Can specify source points should be sampled from a volume (eg, sphere, cylinder, box, etc.)

KCODE Card

KCODE N kest ndiscard ntotal

N = number of particles per cycle,

Typical: 1K - 2K for testing

5K - 100K for production

kest = initial guess for Keff, usually 1.0

ndiscard = number of inactive cycles to discard before beginning tallies

ntotal = total number of cycles to run, should be > ndiscard+100

(see manual for other optional entries)

Initial Source & Energy Spectrum

- **Can specify source points with KSRC or SDEF (not both)**

```
ksrc  0 0 0   .5 .5 .25   .1 .1 .1   [etc]
```

  - For KSRC, source points are reused as needed to get starting locations for all the particles in the initial cycle

```
sdef    x=d1   y=d2   z=d3
si1    -5.    5.          $ extent for x
sp1     0     1           $ uniform in x
si2    15.   75.          $ extent for y
sp2     0     1           $ uniform in y
si3     0.  100.          $ extent for z
sp3     0     1           $ uniform in z
```

- **For the initial cycle only, neutron starting energy is sampled from a Watt fission spectrum. (Other cycles use actual (n,f) data & energy distributions.)**

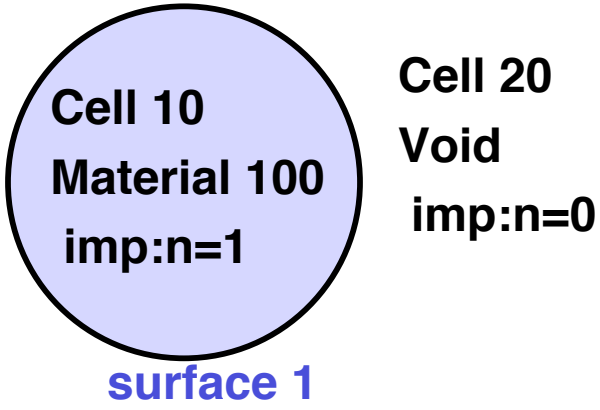
$$p(E) = Ce^{-E/a}\sinh\sqrt{bE}, \qquad a = 0.965 \text{ MeV}, \quad b = 2.29 \text{ MeV}^{-1}$$

Examples

Problem g1

Godiva critical -- using KSRC & surfaces

- Bare, high-enriched uranium sphere
- Sphere radius = 8.741 cm
- Material density =18.74 g/cm3
- Nuclide Wgt-fraction ZAID
  - U235 94.73 92235
  - U238 5.27 92238



(1) Create & edit file "g1"

(2) Add title, cell cards, surface cards, data cards

- surface card (sphere at origin): # so radius

also use these data cards:

- kcode 1000 1.0 10 50
- ksrc 0 0 0

(3) Plot the geometry: mcnp i=g1 ip

(4) Run the problem: mcnp i=g1

(5) Rerun the problem: mcnp i=g1

Problem g1

File g1

```
Godiva critical - using ksrc & surfaces
c
c  CELL CARDS
10    100    -18.74    -1
20    0                 1

c  SURFACE CARDS
1     so   8.741

c  DATA CARDS
imp:n  1   0
m100   92235  -94.73       92238  -5.27
c
kcode   1000   1.0   10   50
ksrc  0  0  0
```

Commands

mcnp5 i=g1 ip **<-- process input & plot geometry**

mcnp5 i=g1 **<-- run the problem...**

Source ?

Files created ?

Comments - g1

- **KSRC 0 0 0**
  - isotropic point source at (0,0,0)
  - used only for the source guess for initial cycle, ignored after that

- **KCODE 1000 1.0 10 50**
  - start 1000 particles
  - run 50 cycles, throw out the first 10
  - initial guess for Keff = 1.0

- **imp:n 1 0**
  - cell 1 has importance 1, cell 2 has importance 0
  - could put this information on cell cards instead:

```
c  CELL CARDS
10   100   -18.74  -1      imp:n 1
20   0                1     imp:n 0
```

- **cleanup:**

  rm out* src* run* com*

Problems g3, g4, g5

Same as problem g1, but:

g3: Use KSRC card to start points at center

g4: Use SDEF card to start points uniformly in sphere

g5: Use srctp file from g4, don't use KSRC or SDEF

For all: use 5000 neutrons/cycle, skip 50, run 150 total

Problem g3

```
Godiva - using KSRC
c
c CELL CARDS
10     100  -18.74  -1
20     0             1

c SURFACE CARDS
1     so 8.741

c DATA CARDS
kcode 5000 1.0 50 150
ksrc  0. 0. 0.
imp:n 1 0
m100     92235 -94.73   92238 -5.27
```

Problem g4

```
Godiva - using SDEF
c
c CELL CARDS
10     100   -18.74   -1
20     0                1

c SURFACE CARDS
1      so 8.741

c DATA CARDS
kcode 5000 1.0 50 150
Sdef   pos= 0 0 0    rad=d1
Si1    0     8.741
Sp1    -21    2      $ density ~ r**2
imp:n 1 0
m100     92235 -94.73    92238 -5.27
```

Problem g5

Assuming that problem g4 created source tape file srctw,

```
mcnp5  I=g5   src=srctw

Godiva - using srctp, no ksrc or sdeff
c
c CELL CARDS
10    100  -18.74  -1
20    0             1

c SURFACE CARDS
1    so 8.741

c DATA CARDS
kcode 5000 1.0 50 150   $ 50 could be changed to 0 !
imp:n 1 0
m100    92235 -94.73   92238 -5.27
```

Examine output file
For KCODE calculation
.....

Random Numbers

- **Random numbers**
  - **MCNP is a Monte Carlo code & uses random numbers to sample from probability densities**
  - **If a calculation is repeated using the same input & cross-section libraries, bit-for-bit identical results will be obtained.**

- **How can you repeat a calculation using different random numbers?**

rand gen= k seed= n

gen= k **- choose the random number generator**
- gen=1 for default 48-bit generator, period ~10^{14}
- gen= 2, 3, 4, 5, 6 for 63-bit generators, period ~10^{18}

seed= n **- set the initial random seed for the problem to n**
- n should be an odd integer, 18 digits or fewer

Example

- Pick a problem, change the random seed, rerun the problem, compare answers.
- Results should differ, but should agree within statistics.
- If calculations are run using different random seeds, the results are statistically independent & may be averaged together. (Using weights proportional to $1/\sigma^2$)

```
Godiva - using KSRC
1     1  -18.74  -1
2     0            1

1     so 8.741

kcode 5000 1.0 50 150
ksrc  0. 0. 0.
imp:n 1 0
m1     92235 -94.73   92238 -5.27
rand   seed= 123456789
```

Criticality Calculations - Part I

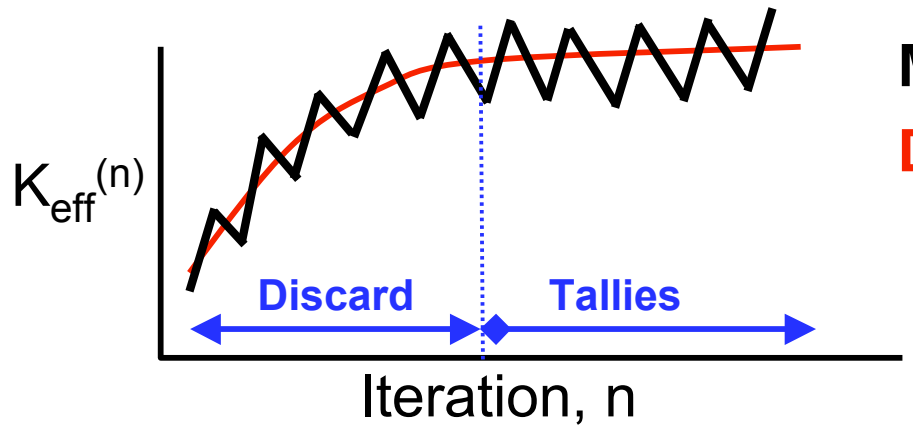
- Criticality Overview
- Criticality Estimators
- KCODE & KSRC Cards
- Examples & Output

Criticality Calculations - Part II

- Convergence
- Plotting
- Examples & Plots
- Continue Runs
- Burnup

Convergence

Power Iteration



Monte Carlo

Deterministic (S_n)

- Guess an initial source distribution
- Iterate until converged (How do you know ???)
- Then
  - **For S_n code: done, print the results**
  - **For Monte Carlo: start tallies, keep running until uncertainties small enough**

- Convergence? Stationarity? Bias? Statistics?

K-eigenvalue equation

Transport equation, for K-eigenvalue problems

- **Use operator (or matrix) form to simplify notation**

$$(L + T)\Psi = S\Psi + \frac{1}{K_{eff}} M\Psi$$

where **L = leakage operator** **S = scatter-in operator**
T = collision operator **M = fission multiplication operator**

- **Rearrange**

$$(L + T - S)\Psi = \frac{1}{K_{eff}} M\Psi$$

⇒ This eigenvalue equation will be solved by **power iteration**

$$(L + T - S)\Psi^{(n+1)} = \frac{1}{K_{eff}^{(n)}} M\Psi^{(n)}$$

Power Iteration - Convergence

- **Expand Ψ in terms of eigenfunctions $u_j(r,E,\Omega)$**

$$\Psi = \sum_{j=0}^{\infty} a_j \vec{u}_j = a_0\vec{u}_0 + a_1\vec{u}_1 + a_2\vec{u}_2 + a_3\vec{u}_3 + ...$$

$$\int \vec{u}_j \vec{u}_k dV = \delta_{jk} \qquad a_j = \int \Psi \cdot \vec{u}_j dV$$

$$\vec{u}_j = \frac{1}{k_j} F \cdot \vec{u}_j \qquad k_0 > k_1 > k_2 > \ldots > 0 \qquad k_0 \equiv k_{effective}$$

- **Expand the initial guess in terms of the eigenmodes**

$$\Psi^{(0)} = \sum_{j=0} a_j^{(0)} \vec{u}_j$$

- **Substitute expansion for $\Psi^{(0)}$ into power iteration equation**

$$\Psi^{(n+1)} = \frac{1}{K^{(n)}} F \cdot \Psi^{(n)} = \frac{1}{k^{(n)}} \cdot \frac{1}{k^{(n-1)}} \cdots \frac{1}{k^{(0)}} \cdot F^n \cdot \Psi^{(0)}$$

$$= \left[\prod_{m=0}^{n} \frac{k_0}{K^{(m)}} \right] \cdot a_0^{(0)} \cdot \left[\vec{u}_0 + \sum_{j=1} \left(\frac{a_j^{(0)}}{a_0^{(0)}} \right) \cdot \left(\frac{k_j}{k_0} \right)^{n+1} \cdot \vec{u}_j \right]$$

Power Iteration - Convergence

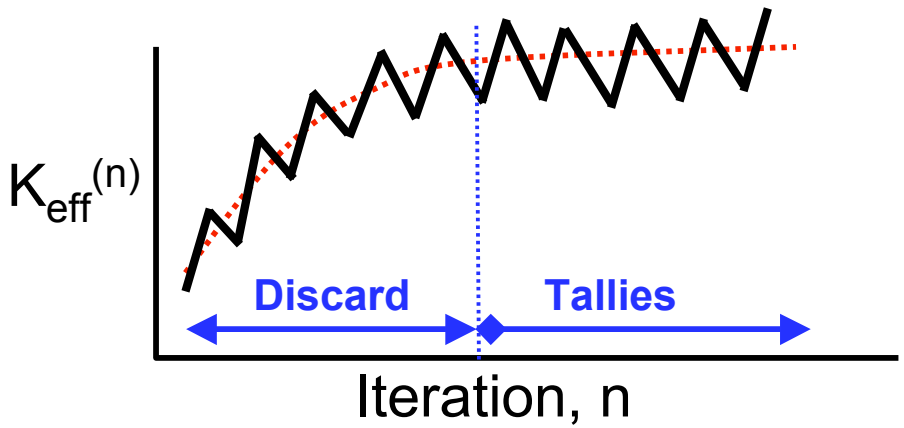
$$\Psi^{(n+1)} \approx [\text{constant}] \cdot \left[\vec{u}_0 + \left(\frac{a_1^{(0)}}{a_0^{(0)}} \right) \cdot \left(\frac{k_1}{k_0} \right)^{n+1} \cdot \vec{u}_1 + \ldots \right]$$

$$K^{(n+1)} \approx k_0 \cdot \left[1 + \left(\frac{a_1^{(0)}}{a_0^{(0)}} \right) \cdot \left(\frac{k_1}{k_0} \right)^{n} \cdot \left(\frac{k_1}{k_0} - 1 \right) \cdot G_1 + \ldots \right]$$

- Because $k_0 > k_1 > k_2 > \ldots$, all of the red terms vanish as $n \to \infty$
  - $\Psi^{(n+1)} \to$ constant $\cdot\ u_0$
  - $K^{(n+1)} \to k_0$
- After the initial transient, error in $\Psi^{(n)}$ is dominated by first mode
  - (k_1 / k_0) is called the dominance ratio, DR or ρ
  - Errors in the source distribution $\Psi^{(n)}$ die off as $\sim (DR)^n$
- For problems with a high dominance ratio (e.g., DR ~ .99), the error in K_{eff} may be small, since the factor (k_1/k_0 - 1) is small.
  - K_{eff} may appear converged, even if the source distribution is not converged

Keff Calculations

- Initial cycles of a Monte Carlo K-effective calculation should be discarded, to avoid contaminating results with errors from initial guess
  - **How many cycles should be discarded?**
  - **How do you know if you discarded enough cycles?**

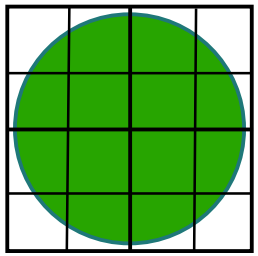


- Analysis of the power iteration method shows that Keff is not a reliable indicator of convergence **-- K_{eff} can converge faster than the source shape**

- Based on concepts from information theory, **Shannon entropy of the source distribution** is useful for characterizing the convergence of the source distribution

Shannon Entropy of the Fission Source

- **Divide the fissionable regions of the problem into N_S spatial bins**
  - **Spatial bins should be consistent with problem symmetry**
  - **Typical choices: -- 1 bin for each assembly**
    -- regular grid superimposed on core
  - **Use dozens or hundreds of bins, not thousands**



- **During the random walks for a cycle, tally the fission source points in each bin**
  - **Provides a discretized approximation to the source distribution**
  - **$\{ p_J, \quad J=1,N_S \}$**

- **Shannon entropy of the source distribution**

$$H(S) = -\sum_{J=1}^{N_S} p_J \cdot \ln_2(p_J), \quad \text{where} \quad p_J = \frac{\text{(\# source particles in bin J)}}{\text{(total \# source particles in all bins)}}$$

Shannon Entropy of the Fission Source

- **Shannon entropy of the source distribution**

$$H(S) = -\sum_{J=1}^{N_S} p_J \cdot \ln_2(p_J), \quad \text{where} \quad p_J = \frac{\text{(\# source particles in bin J)}}{\text{(total \# source particles in all bins)}}$$

  - **$0 \leq H(S) \leq \ln_2(N_S)$**
  - **For a <u>uniform</u> source distribution, $H(S) = \ln_2(N_S)$**
  - **For a <u>point</u> source (in a single bin), $H(S) = 0$**

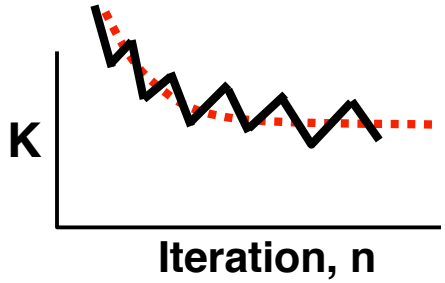
- **$H(S^{(n)})$ provides a single number to characterize the source distribution for iteration n (no physics!)**

$\Rightarrow$ As the source distribution converges in 3D space, a line plot of $H(S^{(n)})$ vs. n (the iteration number) converges

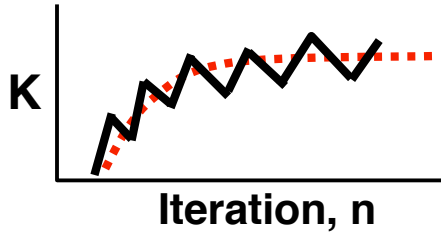
Typical K-effective convergence patterns

- **Higher mode error terms die out as $(k_1 / k_0)^n$, for n iterations**
  - **k_1 is the eigenvalue of the first higher mode, $k_0 = k_{eff}$**
  - **k_1 / k_0 is called the Dominance Ratio**

- **When initial guess is concentrated in center of reactor, initial K_{eff} is too high** (underestimates leakage)



- **When initial guess is uniformly distributed, initial K_{eff} is too low** (overestimates leakage)



- **The Sandwich Method uses 2 K_{eff} calculations - one starting too high & one starting too low. Both calculations should converge to the same result.**

K_{eff} & H_{src} Convergence

Example - Reactor core (Problem inp24)

$K^{(n)}$ vs cycle

20

H(fission source)

K_{eff}

80

K_{eff} & H_{src} Convergence

Example - Loosely-coupled array of spheres (Problem test4s)

$K^{(n)}$ vs cycle

75

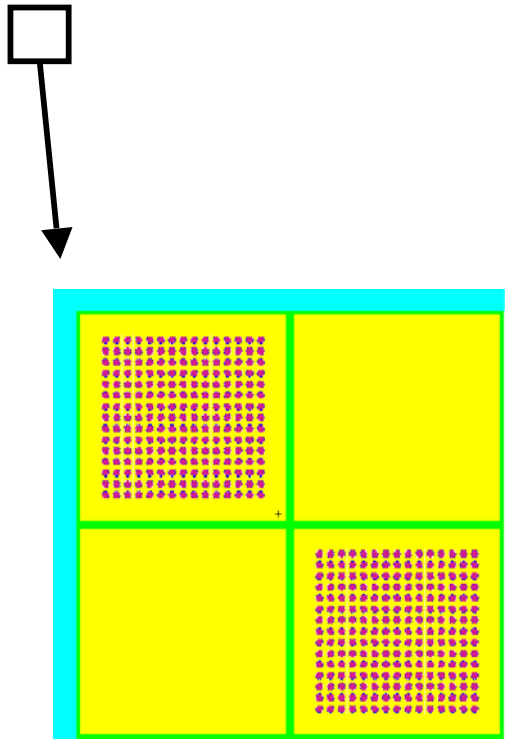
H(fission source)

K_{eff}

85

K_{eff} & H_{src} Convergence

Example - Fuel Storage Vault (Problem OECD_bench1)



$K^{(n)}$ vs cycle

20 ?

H(fission source)

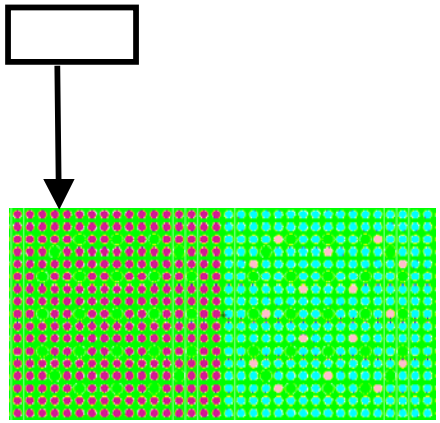
2000

K_{eff} & H_{src} Convergence

Example - PWR 1/4-Core (Napolitano)

$K^{(n)}$ vs cycle

25

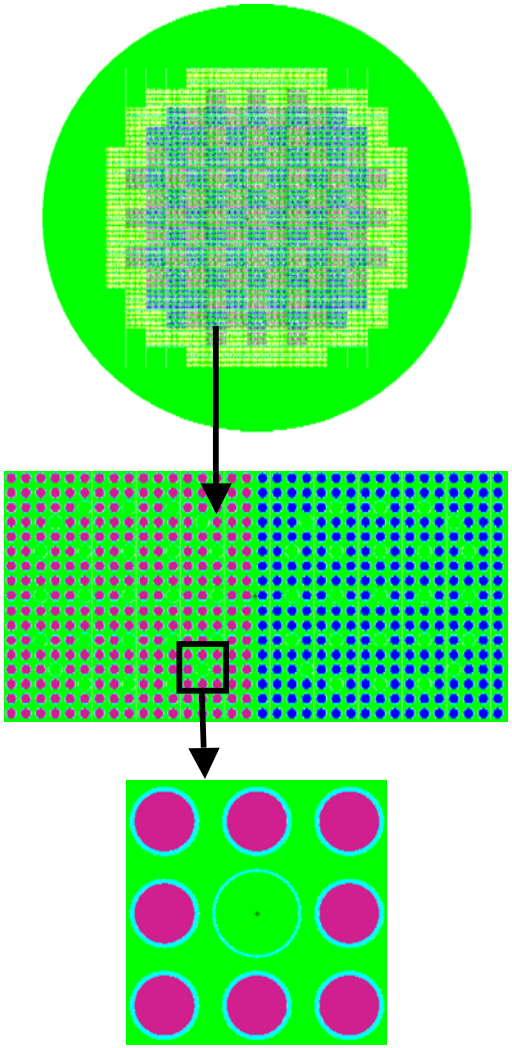


H(fission source)

50

K_{eff} & H_{src} Convergence

Example - 2D PWR (Ueki)



$K^{(n)}$ vs cycle

25

H(fission source)

50

Guidance on Computing H_{src}

Source Entropy & MCNP5

- **Grid for computing H_{src}**
  - **User can specify a rectangular grid in input**

    hsrc n_x x_{min} x_{max} n_y y_{min} y_{max} n_z z_{min} z_{max}

    example: hsrc 5 0. 100. 5 0. 100. 1 -2. 50.

  - **If hsrc card is absent, MCNP5 will choose a grid based on the fission source points, expanding it if needed during the calculation**

- **MCNP5 prints H_{src} for each cycle**

- **MCNP5 can plot H_{src} vs cycle**

- **Convergence check at end of problem**
  - **MCNP5 computes the average H_{src} and its population variance σ_H^2 for the last half of the cycles**
  - **Then, finds the first cycle where H_{src} is within the band $\langle H_{src} \rangle \pm 2\sigma_H$**
  - **Then, checks to see if at least that many cycles were discarded**

H_{src} Convergence vs Number of Spatial Bins

- For large number of bins, H_{src} approaches uniform upper limit
- Use 10s or 100s of bins, not 1000s or more

10000 bins

1000 bins

100 bins

OECD bench3

H_{src} and 2D vs 3D Spatial Bins

- For 3D problems, using a 2D bin layout for H_{src} may incorrectly assess convergence
- Important to use 3D bin layout for 3D problems

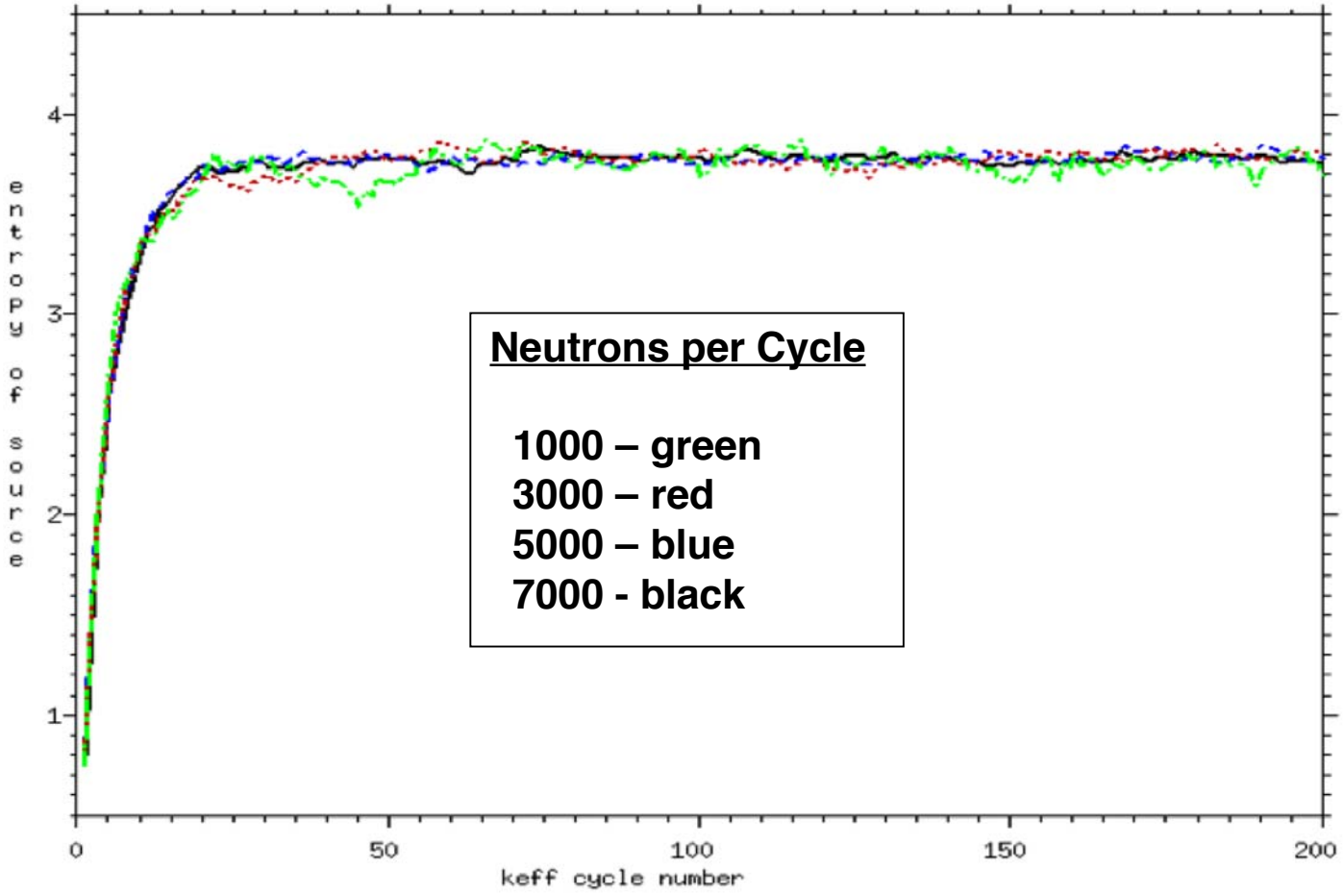
3D bin layout

2D bin layout

inp24, 3D 1/4 core PWR

H_{src} Convergence vs Neutrons per Cycle

- H_{src} converges to the same value, regardless of neutrons/cycle
- More neutrons/cycle ⇒ less noise in plot



Convergence Testing Using H_{src}

Example 1 - New Output

- Information on mesh used for calculating H_{src}

```
comment.
comment. entropy of the fission source distribution will be computed
comment.
comment. the mesh for source entropy is based on the site coordinates
comment.   using  4 x  4 x  4 =    64  mesh cells
comment.
comment.     Xmin=  2.2973E+01     Xmax=  3.6787E+02
comment.     Ymin=  5.3433E+01     Ymax=  1.8017E+02
comment.     Zmin= -1.1753E+01     Zmax=  1.1701E+01
comment.
comment. the mesh will be automatically expanded if necessary to
comment.   encompass the entire fission source distribution
comment.

cycle    k(col)       ctm   entropy    active     k(col)    std dev
    1   1.35561      0.00  1.58E+00
    2   1.12276      0.01  1.77E+00
  . . . . .
```

Example 1 - New Output (cont'd)

- At end of run, information on convergence, based on H_{src}

```
  . . . . .
  129    1.10703       0.17   2.43E+00         99     1.06324     0.00502
  130    1.14460       0.17   2.40E+00        100     1.06405     0.00503
source distribution to file srctt            cycle =   130
run terminated when     130 kcode cycles were done.
comment.
comment. Average fission-source entropy for the last half of cycles:
comment.       H=  3.02E+00  with population std.dev.=  4.97E-01
comment.
comment.
comment. Cycle   47 is the first cycle having fission-source
comment.   entropy within 1 std.dev. of the average
comment.   entropy for the last half of cycles.
comment.   At least this many cycles should be discarded.
warning.
warning. The fission-source entropy for the first active cycle, cycle=  31,
warning.   is NOT within 1 standard deviation of the average
warning.   source entropy for the last half of cycles.
warning.
warning. You should consider rerunning the problem,
warning.   discarding more initial cycles.
warning.
```

Example 2 - New Output

- At end of run, information on convergence, based on H_{src}

```
. . . . .
  174   1.15010       0.20  2.76E+00        99    1.11162    0.00292
  175   1.13930       0.21  2.73E+00       100    1.11190    0.00291
source distribution to file srctu           cycle =   175
run terminated when     175 kcode cycles were done.

comment.
comment. Average fission-source entropy for the last half of cycles:
comment.      H=  2.66E+00  with population std.dev.=  2.21E-01
comment.
comment.
comment. Cycle   36 is the first cycle having fission-source
comment.   entropy within 1 std.dev. of the average
comment.   entropy for the last half of cycles.
comment.   At least this many cycles should be discarded.
comment.
comment. Source entropy convergence check passed.
comment.
```

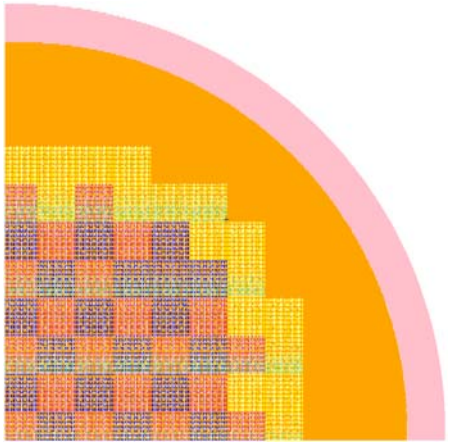
Example

- For Watts-Bar Unit #1, 3D whole-core model

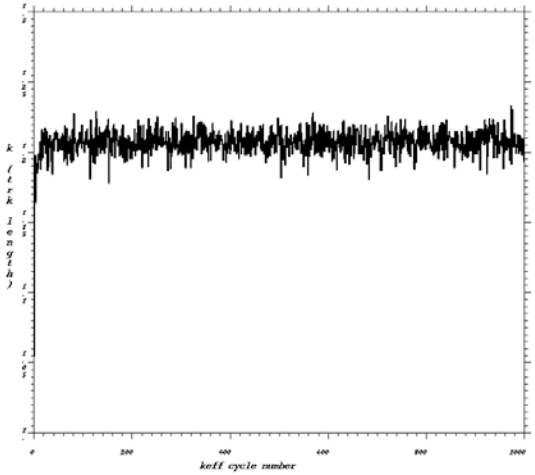
```
c
c   Mesh tally for assembly powers    (flux*fission*Q)
fmesh104:n  geom=xyz  origin= -161.25 -161.25 -194.492
            imesh=161.25 iints=15
            jmesh=161.25 jints=15
            kmesh=170.508 kints=1
fm104  -1.0  0  -6 -8       $ NOTE: 0 = use actual materials
c
c  Mesh tally for detailed fast & thermal flux
fmesh204:n  geom=xyz  origin= -161.25 -161.25 -194.492
            imesh=161.25 iints=150
            jmesh=161.25 jints=150
            kmesh=170.508 kints=1
            emesh  .625e-6  20.
```

Example (cont'd)

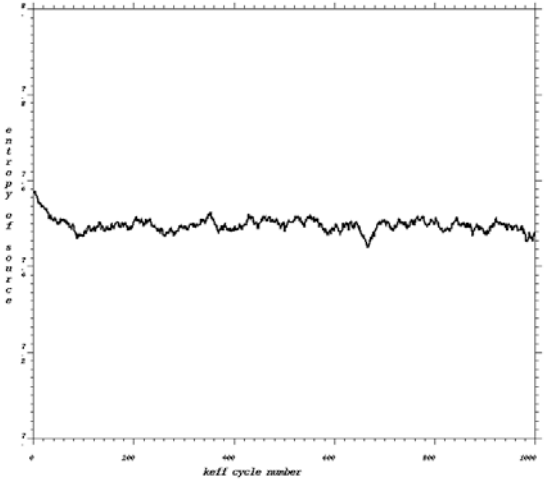
Geometry Model (1/4)



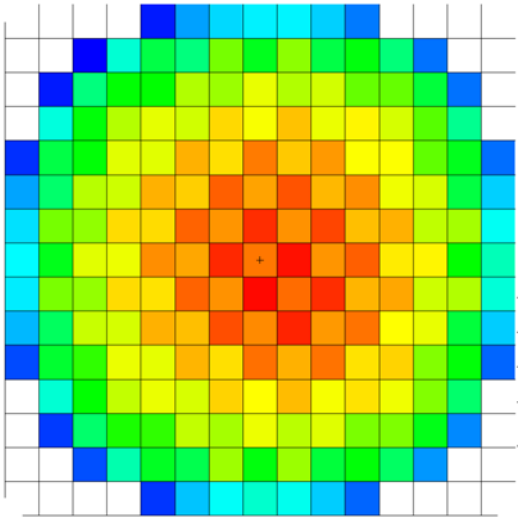
K vs cycle



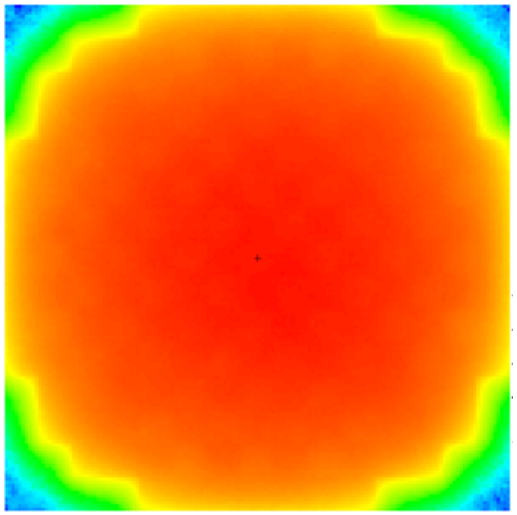
H_{src} vs cycle



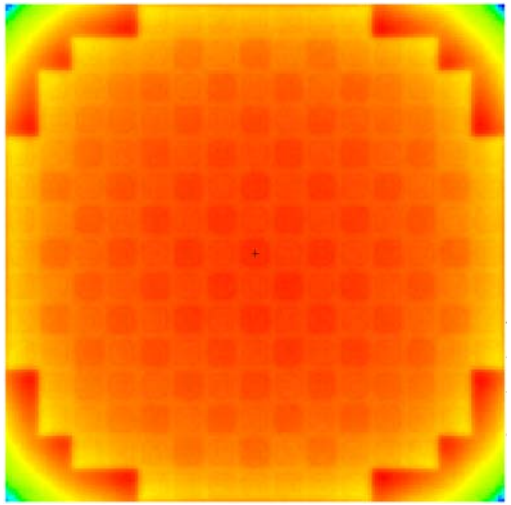
Assembly Powers



Fast Flux



Thermal Flux



Plotting

Criticality Output and Plotting

- **Plot the geometry**
- **Make a trial run**
  - 50-100 cycles
  - 1000-2000 neutrons/cycle
- **Examine plots of:**
  - Keff vs cycle number
  - Shannon entropy of fission distribution vs cycle
- **Fix the KCODE card:**
  - Make sure that enough initial cycles are discarded
  - Change neutrons/cycle to 5K or more
- **Run calculation until statistics are small enough**
- **Look at convergence plots again, to verify…**

Criticality Output and Plotting

- **To invoke the MCNP tally plotter:**
  - While problem is running, type **CTL-C**, then **m**, or
  - After run complete, type: **mcnp5 r=runtpe z**
- **Commands for the tally plotter:**

  mcplot> kcode n

kcode 1	collision estimate of single-cycle K vs cycle
kcode 2	absorption estimate of single-cycle K vs cycle
kcode 3	**track-length estimate of single-cycle K vs cycle**
kcode 11-13	same as 1, 2, 3, but cumulative averages vs cycle
kcode 6	**Shannon entropy of fission distribution vs cycle**
kcode 16	**Combined trk/abs/col ave. K vs cycle, with std dev**
kcode 17	Combined trk/abs/col ave K vs cycles skipped (what if)

- **Can plot several quantities together**

  mcplot> **kcode 1 coplot kcode 2 coplot kcode 3**
- **See MCNP5 Manual, Volume II, Appendix B**

Plotting Exercise

- **Make keff plots from Problem g1 RUNTPE**

```
mcnp5 r=runtpe  Z
mcplot>  kcode 3
mcplot>  kcode 6
mcplot>  kcode 16
mcplot>  KCODE 1 coplot KCODE 2 coplot KCODE 3
```

- **MPLOT in INP File produces plots during execution**
  - Add this data card to **g1**, then rerun

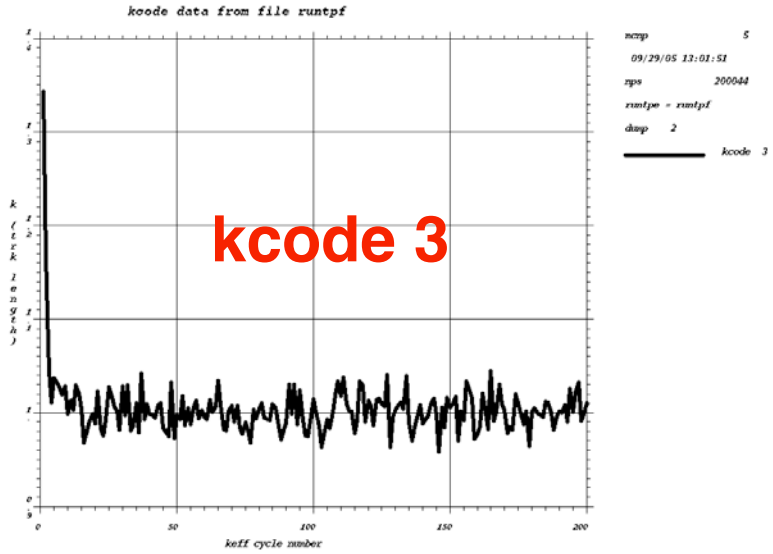
    …

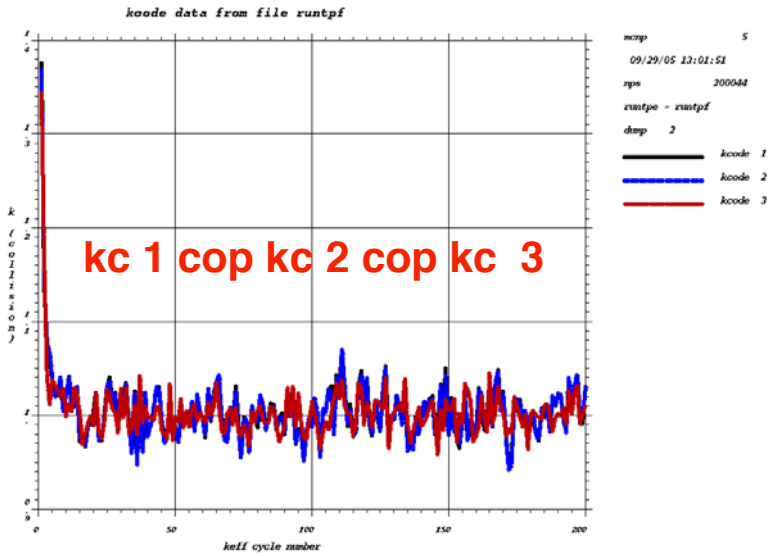
    kcode 5000 1.0 10 150

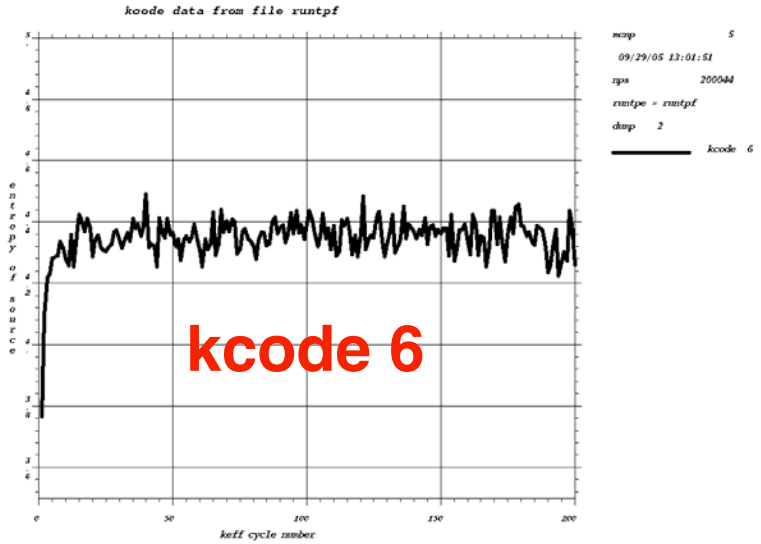
    mplot kcode 3 freq 5

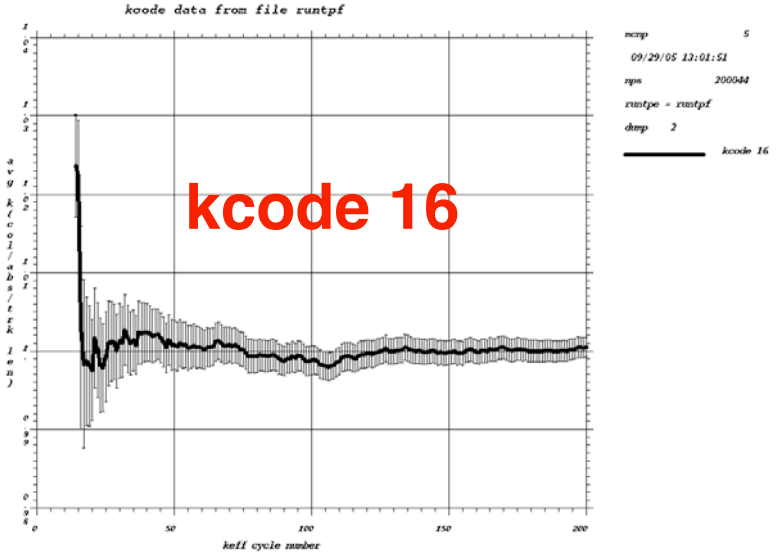
- **See MCNP5 Manual, Volume II, Appendix B**

Godiva Example









Cautions and Suggestions

- **You MUST make sure that enough initial cycles are discarded to ensure convergence of Keff & the fission source distribution**
- **Bias**
  - If you run with 10s or 100s of neutrons/cycle, there will be a bias in Keff.
  - Bias is proportional to 1/N, so using more neutrons/cycle reduces the bias.
  - **Use 5K or more neutrons/cycle for production runs so that bias is negligible.**
- **Correlation**
  - Tallies for one cycle are correlated to the previous (and next)
  - Cycle-to-cycle correlation may cause estimated relative errors may be low.
  - Can check on this by examining "batching" results in MCNP output.
- **If anything looks suspicious,**
  - Run more cycles
  - Start over with more neutrons/cycle
  - Rerun the calculation with a new random number seed.

Cautions and Suggestions

- Use more neutrons/cycle, not fewer…
- For systems with high dominance ratio (ie, close to 1), may need to discard 100s or 1000s of initial cycles. For typical LWR's, probably need to discard ~50 initial cycles.
- Should run at least 100 cycles after convergence, to ensure that statistical analysis is reliable
- For reactors, initial source guess should cover most of core. Not necessary to worry about fuel vs non-fuel.
- For criticality safety, make sure initial source guess has some source points in each fissionable region.

CONTINUE Runs

MCNP C [INP=inpfile RUN=runfile]

Continues from RUNTPE “runfile” using input file “inpfile”

Special input file to change parameters

Optional Message Block & blank
Continue
Data Cards

Example:

```
Continue
KCODE 5000   1 50 150
```

Legal Continue Run Data

KCODE (total)
NPS (total)
NPS -1
FQ
DD
CTME (more)
PRDMP
PRINT
MPLOT
DBCN
LOST
IDUM, RDUM, ZA, ZB, ZC

Burnup

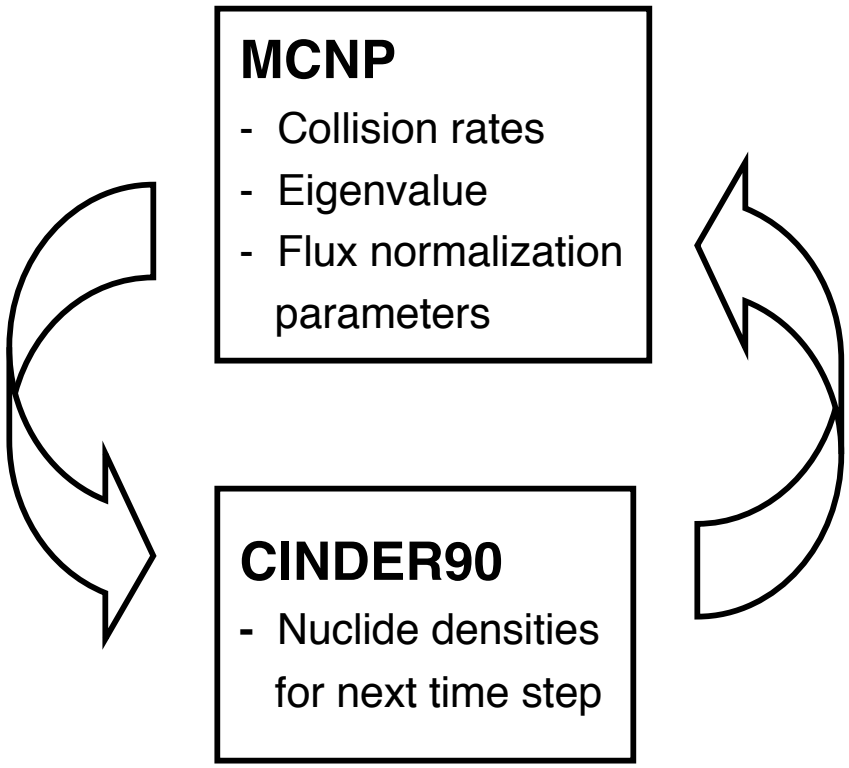
(MCNPX)

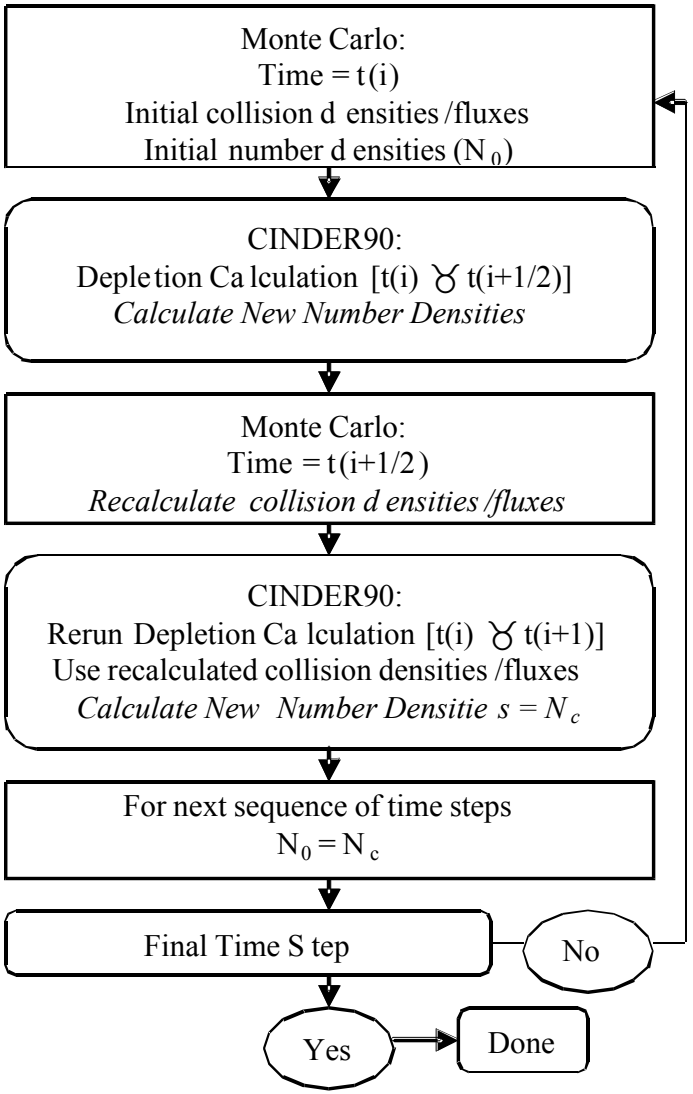
What is Depletion Analysis?

- **During reactor operation, the isotopic concentration of the reactor material (fuel/coolant/clad/shielding) will change as isotopes consume neutrons and undergo various nuclear reactions**
  - (n,f), (n,gamma), (n,alpha), (n,beta), (n,p), etc.

- **Changes in the isotopic concentration over time will result in changes in reactor parameters**
  - Flux, reactivity, power distribution, shutdown margin, poison concentration, rod height, …
  - These reactor parameters limit reactor operation characteristics, therefore it is necessary to accurately calculate these values at many time steps

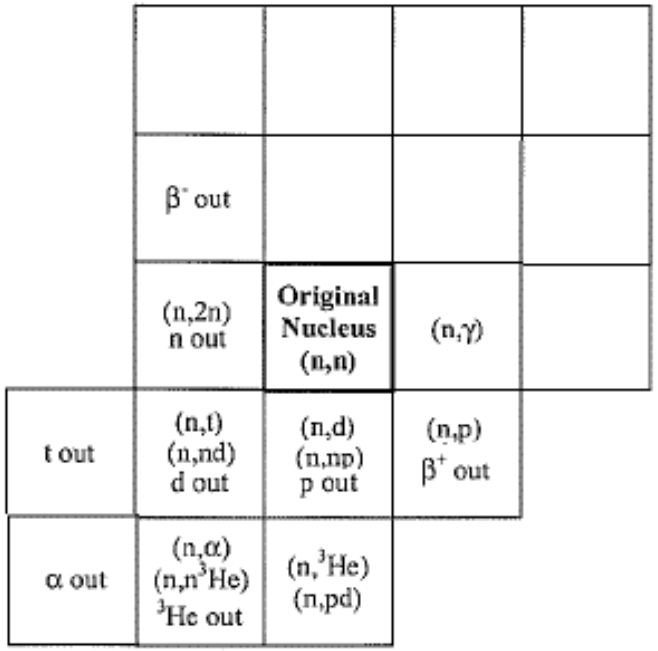
Burnup Implementation





Isotopes Tracked

- **MCNP currently only tracks depletion information for certain isotopes**
  - Materials listed on material card(s)
  - Nuclides selected from a fission product tier
    - **Tier 1.** (default) Zr-93, Mo-95, Tc-99, Ru-101, Xe-131, Cs-133, Cs-137, Ba-138, Pr-141, Nd-143, Nd-145
    - **Tier 2.** Nuclides contained in the fission product array that are included in the MCNPX 2.5.0 XSDIR
    - **Tier 3.** All isotopes contained in the fission product array
  - Nuclides created from the isotope generator algorithm
- **CINDER90 tracks isotope concentrations for 3400 isotopes**
  - Only those isotopes utilized in the transport calculation contain isotope abundance data in the output file



n = neutron
p = proton
d = deuteron
t = triton
α = alpha particle
β⁻ = beta minus (negative electron)
β⁺ = beta plus (positron)
ε = electron capture

BURN CARD

```
BURN   TIME=  T1  T2 T3   ...
       PFRAC=  F1  F2  F3 ...
       POWER= P
       MAT=   ±M1   ±M2   ±M3 ...
       OMIT=  J1  N1   I11  I12 ...   J2  N2  I21  I22 ...
       AFMIN=  A1  A2
       BOPT=  B1  B2  B3
       MATMOD=
       MATVOL=  V1  V2 ...
```

Burn Card must be placed before material cards!

Ti = duration of burn step i (days). Default is one time step of one day.

Fi = fraction of POWER (0-1). Default is 100% POWER (1.0)

P = power level (MW). Default is 1.0 MW.

Mi = list of material numbers to include in the burn. If the BURN card is utilized then a burn material MUST be specified.

BURN CARD (continued)

Ji = ith material for which to omit nuclides Ii1, Ii2, etc.

Ni = number of omitted nuclides listed for the ith material.

Ii1, Ii2, ... = 1st, 2nd, etc. omitted nuclide for the ith material.

B1 = Q value multiplier. Default is 1.0.

B2 = mn

m = 0/1/2 = fission product tiers 1/2/3

n = 1/2/3/4 = print output by decreasing mass/decreasing activity/decreasing specific activity/increasing ZAID

Example: B2 = 14 tier 2, print by increasing zaid

B3 = Models option

-1 Fatal error if models are used in the problem

1 Runs with models

A1 = atom fraction minimum below which the atom fraction is set to zero. Default is 1.0e-10.

A2 = decay chain convergence criteria. Default =1e-10.

V = total repeated volume of material M

BURN CARD (continued)

MATMOD=

$$NT,_1TS,_1NM,_1MT_1,_1K_1,_1Z_1^1,_1C_1^1,_1Z_1^2,_1C_1^2,\ldots,_1Z_1^{K_1},_1C_1^{K_1},$$
$$\ldots,_1MT_n,_1K_n,_1Z_n^1,_1C_n^1,_1Z_n^2,_1C_n^2,\ldots,_1Z_n^{K_n},_1C_n^{K_n},$$
$$_jTS,_jNM,_jMT_1,_jK_1,_jZ_1^1,_jC_1^1,_jZ_1^2,_jC_1^2,\ldots,_jZ_1^{K_1},_jC_1^{K_1},$$
$$\ldots,_jMT_n,_jK_n,_jZ_n^1,_jC_n^1,_jZ_n^2,_jC_n^2,\ldots,_jZ_n^{K_n},_jC_n^{K_n},$$

NT = Number of time steps (1 through l)

$_jTS$ = Time step (1..j) for which to manually change nuclide concentrations of material MT_i. Enter "1" for 2nd, etc.

$_jNM$ = Number of materials at time step "j" that incur nuclide concentration changes

$_jMT_n$ = nth material number for which to manually change nuclides at time step "j". Positive value indicates atom/wt. fraction and negative value indicates atom/gram densities.

$_jK_n$ = number of nuclides to manually change for the ith material

$_jZ_n^{K_n}$ = 1st, 2nd,.. Knth nuclide of the MTith material at time step "j" for which a concentration will be specified. List as a ZAID value.

$_jC_n^{K_n}$ = concentration for the nth isotope in material MTi at time step "j". Enter positive values for atom fractions or atom densities, and enter a negative value for wt. fractions or gram densities. See sign of MTi to specify either fraction or density

BURN CARD - MATMOD (continued)

MATMOD=

$$NT,_1TS,_1NM,_1MT_1,_1K_1,_1Z_1^1,_1C_1^1,_1Z_1^2,_1C_1^2,\ldots,_1Z_1^{K_1},_1C_1^{K_1},$$
$$\ldots,_1MT_n,_1K_n,_1Z_n^1,_1C_n^1,_1Z_n^2,_1C_n^2,\ldots,_1Z_n^{K_n},_1C_n^{K_n},$$
$$_jTS,_jNM,_jMT_1,_jK_1,_jZ_1^1,_jC_1^1,_jZ_1^2,_jC_1^2,\ldots,_jZ_1^{K_1},_jC_1^{K_1},$$
$$\ldots,_jMT_n,_jK_n,_jZ_n^1,_jC_n^1,_jZ_n^2,_jC_n^2,\ldots,_jZ_n^{K_n},_jC_n^{K_n},$$

$_jZ_n^{K_n}$ = 1st, 2nd,.. Knth nuclide of the MTith material at time step "j" for which a concentration will be specified. List as a ZAID value.

$_jC_n^{K_n}$ = concentration for the nth isotope in material MTi at time step "j". Enter positive values for atom fractions or atom densities, and enter a negative value for wt. fractions or gram densities. See sign of MTi to specify either fraction or density

Example Problem - b1 (mcnpx)

Problem b1

Burnup Default Cross Section Option

Copy case g1 and rename to b1

(1) Add a burn card

(2) Add a volume card for material 100

(3) Use 2 time steps (time durations 10 100 days)

(4) Burn at 5 MW

(5) Burn material 100 at 100% power

(6) Use the BOPT keyword (BOPT=1.0,4)

(7) Use the OMIT keyword (OMIT=100,2,90234,91232)

(8) Run the problem using MCNPX, not MCNP5

Burn Card must be placed before material cards!

Problem b1 (mcnpx)

File b1

```
Godiva critical - using burnup
c
c  CELL CARDS
10     100     -18.74     -1   vol=2797.5
20     0                   1

c  SURFACE CARDS
1      so    8.741

c  DATA CARDS
imp:n  1     0
BURN    TIME=10  100
        PFRAC=1.0  1.0
        POWER=5
        MAT=100
        BOPT=1.0 4
        OMIT=100,2,90234,91232
m100    92235  -94.73       92238  -5.27
c
kcode    1000    1.0    10    50
ksrc  0   0   0
```

Examine output file
For BURN calculation
.....

Burnup - Cautions and Suggestions

- **For each time step, you must run enough neutrons/cycle, discard enough cycles to guarantee convergence, and run enough active cycles, so that the uncertainties on fluxes and reaction rates are acceptably small in every depletion region.**
- An **omit** keyword will be required to omit nuclides generated by the isotope generator algorithm, that do not contain a transport cross sections, yet will be tracked in CINDER90
- Must use at least fission product tier 2 in order model majority of important fission products
- In order to track entire decay chains in transport, you must include nuclides at a density of 1e-36 that are not included by the isotope generator algorithm
- Use time steps that account for buildup and decay of highly absorbing isotopes
  - Fission product poison buildup
  - Burnable poison burnout
- Only group together materials incurring similar power distributions
- Section a larger burn material into smaller burn materials if the power gradient across the large burn material is determined to be too significant to assume a homogenized burnup

Burnup Equation (background)

$$\frac{dN_m(r,t)}{dt} = -N_m(r,t)\beta_m + \overline{Y}_m + \sum_{k \neq m} N_k(r,t)\gamma_{k \to m}$$

$$\beta_m = \lambda_m + \sum_j \int \sigma_{m,j}(E)\Phi(r,E,t)dE$$ = loss coefficient

$$\gamma_{k \to m} = \sum_{m \neq k} L_{km}\lambda_k + \sum_{m \neq k}\sum_j \int Y_{km,j}(E)\sigma_{k,j}(E)\Phi(r,E,t)dE$$ = gain coefficient

$\overline{Y}_m$ = continuous feed/removal coefficient

$Y_{km,j}(E)$ = reaction yield leading to production of isotope m from isotope k

λ_m = decay constant for isotope m

L_{km} = branching ratio for the decay of isotope k into isotope m

$\sigma_{m,j}(E)$ = absorption reaction cross section for reaction type j for isotope m

$\Phi(r,E,t)$ = scalar spatial, energy and time dependent flux

CINDER90 Method (background)

- **The set of coupled differential equations is reduced to a set of linear differential equations**
  - Linear chains are created for each isotope transmutation path

$$\frac{d\mathrm{N}_i}{dt} = \bar{Y}_i + \mathrm{N}_{i-1}(t)\gamma_{i-1} - \mathrm{N}_i(t)\beta_i$$

  - The solutions of each linear chain determines a partial nuclide density

$$\mathrm{N}_n(t) = \sum_{m=1}^{n}\prod_{k=m}^{n-1}\gamma_k\left\{\bar{Y}_m\left[\frac{1}{\prod_{l=m}^{n}\beta_l} - \sum_{j=m}^{n}\frac{e^{-\beta_{jt}}}{\prod_{i=m,\neq j}^{n}\left(\beta_i-\beta_j\right)}\right] + \mathrm{N}_m^0\sum_{j=m}^{n}\frac{e^{-\beta_{jt}}}{\prod_{i=m,\neq j}^{n}\left(\beta_i-\beta_j\right)}\right\}$$

CINDER90 Method (background)

– Partial nuclide densities are summed to determine the total nuclide density

$$N_m = \sum_{j=1}^{M} \mathrm{N}_{i,j}$$

– Each transmutation path for each nuclide is followed until it is determined that significant amounts of progeny are no longer created

$$P_m(t) = \int_0^t N_m(t')\beta_m dt'$$