LA-UR-06-7094

Title: Monte Carlo Eigenvalue Calculations

Author(s): Forrest Brown

Intended for: Monte Carlo lectures

LA-UR-06–7094

Monte Carlo Eigenvalue Calculations

Forrest Brown

Monte Carlo Codes (X-3-MCC)
Los Alamos National Laboratory

`fbrown@lanl.gov`

Abstract

LA-UR-06–7094

Monte Carlo Eigenvalue Calculations

F Brown, X-3-MCC

This talk will cover 4 aspects of Monte Carlo eigenvalue calculations:

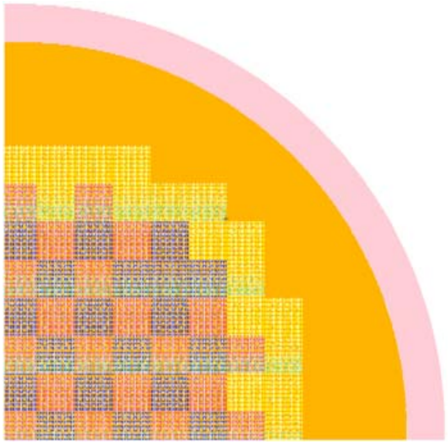
1. Formulation of the k- and alpha-eigenvalue equations from the time-dependent linear Boltzmann transport equation
2. The power iteration method for solving the equations & its convergence behavior
3. The use of Shannon entropy of the fission source distribution for assessing convergence
4. A novel application of Wielandt's method to accelerate the convergence.

Monte Carlo Eigenvalue Calculations

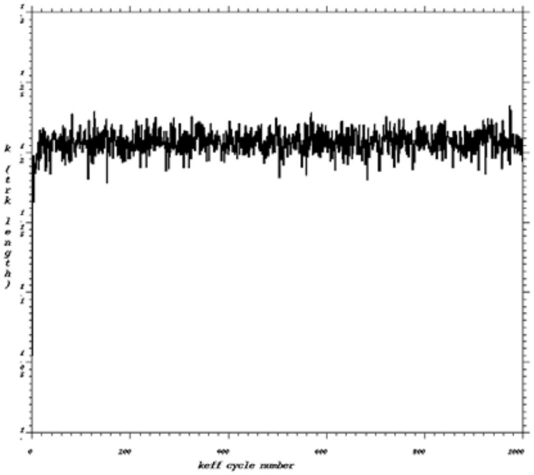
- K- and α-Eigenvalue Equations
- Power Iteration & Convergence
- Shannon Entropy for Convergence Analysis
- Wielandt Acceleration

Reactor Analysis with Monte Carlo

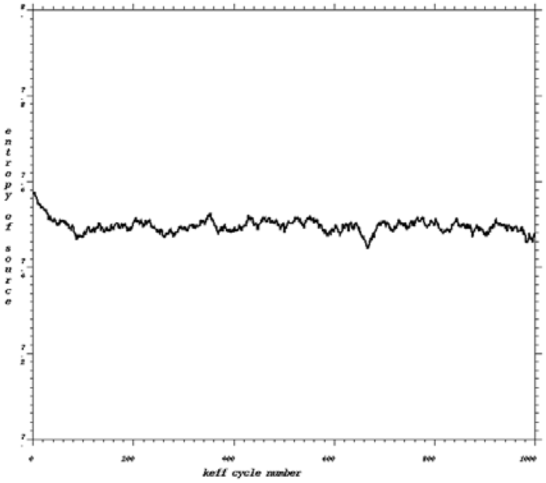
Geometry Model (1/4)



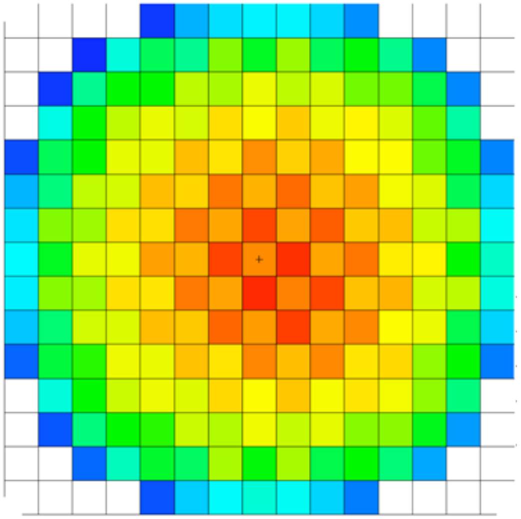
K vs cycle



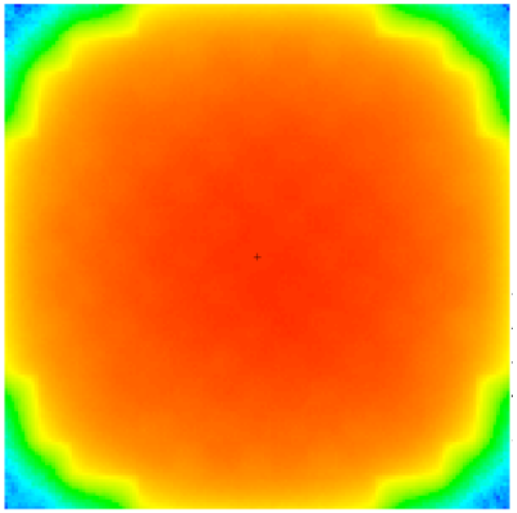
H_{src} vs cycle



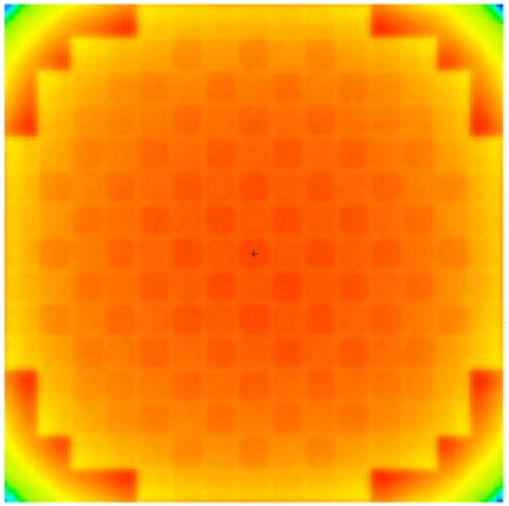
Assembly Powers



Fast Flux



Thermal Flux



K- and α- Eigenvalue Equations

Time-dependent Transport

- **Time-dependent linear Boltzmann transport equation for neutrons, with prompt fission source & external source**

$$
\begin{aligned}
\frac{1}{v}\frac{\partial\psi(\vec{r},E,\vec{\Omega},t)}{\partial t} = \quad & Q(\vec{r},E,\vec{\Omega},t) \;+\; \iint \psi(\vec{r},E',\vec{\Omega}',t)\,\Sigma_S(\vec{r},E' \to E,\vec{\Omega}\cdot\vec{\Omega}',t)\,d\vec{\Omega}'dE' \\
& + \frac{\chi(\vec{r},E,t)}{4\pi}\iint \nu\Sigma_F(\vec{r},E',t)\,\psi(\vec{r},E',\vec{\Omega}',t)\,d\vec{\Omega}'dE' \\
& - \left[\,\vec{\Omega}\cdot\nabla \;+\; \Sigma_T(\vec{r},E,t)\,\right]\cdot\psi(\vec{r},E,\vec{\Omega},t)
\end{aligned}
$$

Without material motion corrections

$$
\frac{1}{v}\frac{\partial\psi(\vec{r},E,\vec{\Omega},t)}{\partial t} = Q \;+\; [S+M]\cdot\psi \;-\; [L+T]\cdot\psi
$$

- **This equation can be solved directly by Monte Carlo, assuming:**
  - Each neutron history is an IID trial (independent, identically distributed)
  - All neutrons must see same probability densities in all of phase space
  - **Usual method: geometry & materials fixed over solution interval Δt**

Time-dependent Transport

$$\frac{1}{v}\frac{\partial\psi(\vec{r},E,\vec{\Omega},t)}{\partial t} = Q + [S+M]\cdot\psi - [L+T]\cdot\psi$$

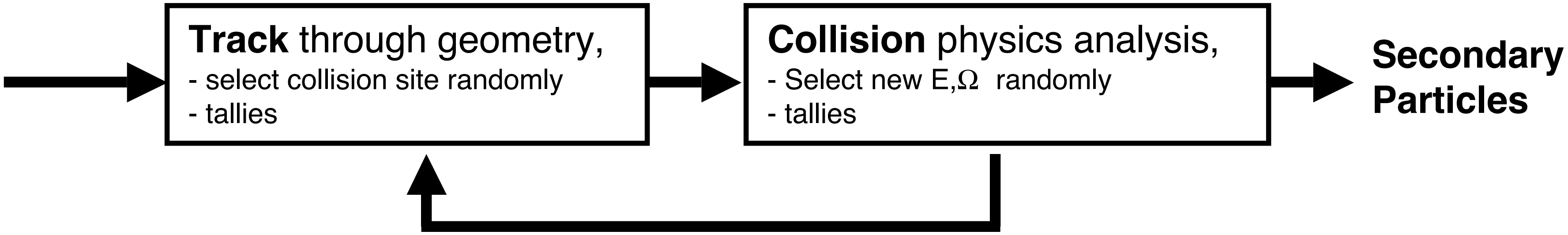
- **Monte Carlo solution (over Δt, with fixed geometry & materials)**
  - Simulate time-dependent transport for a neutron history
  - If fission occurs, bank any secondary neutrons.
  - When original particle is finished, simulate secondaries till done.
  - Tallies for time bins, energy bins, cells, …
- **At time t, the overall neutron level is** $N(t) = \iiint_{\vec{r},E,\hat{\Omega}} \frac{\psi(\vec{r},E,\hat{\Omega},t)}{v} d\vec{r}dEd\hat{\Omega}$
- **Alpha can be defined by: $N(t) = N(0)\, e^{\alpha t}$**

$$\alpha \approx \frac{\ln N_2 - \ln N_1}{t_2 - t_1}$$

This is the "dynamic alpha", NOT an eigenvalue !

Particle Histories

- **Random Walk for particle**

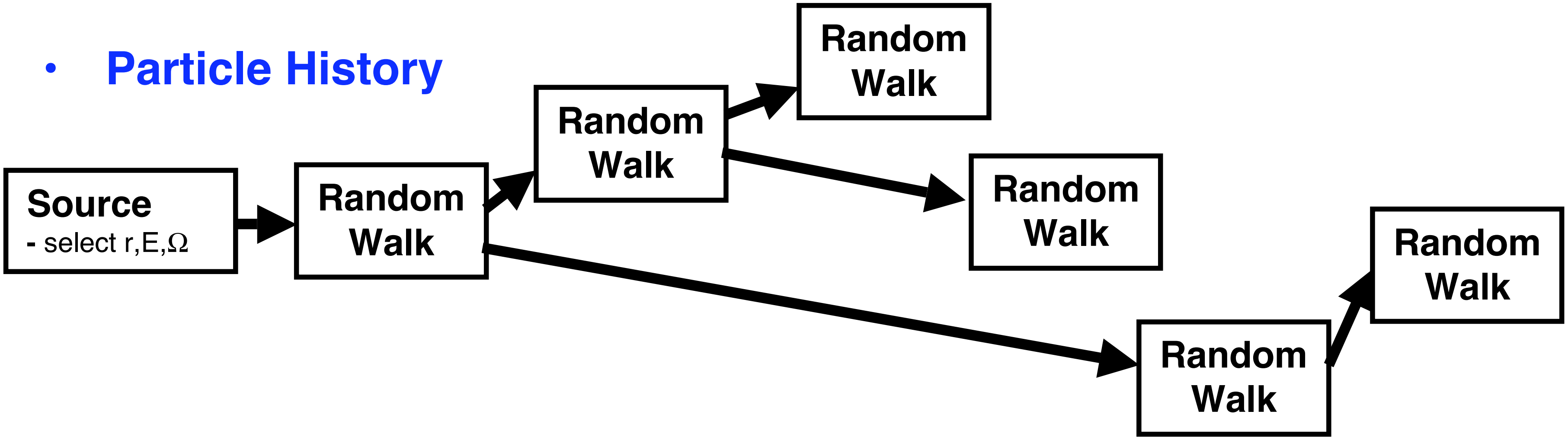


Track through geometry,
- select collision site randomly
- tallies

Collision physics analysis,
- Select new E,Ω randomly
- tallies

Secondary Particles

- **Particle History**



Source
- select r,E,Ω

Random Walk

Random Walk

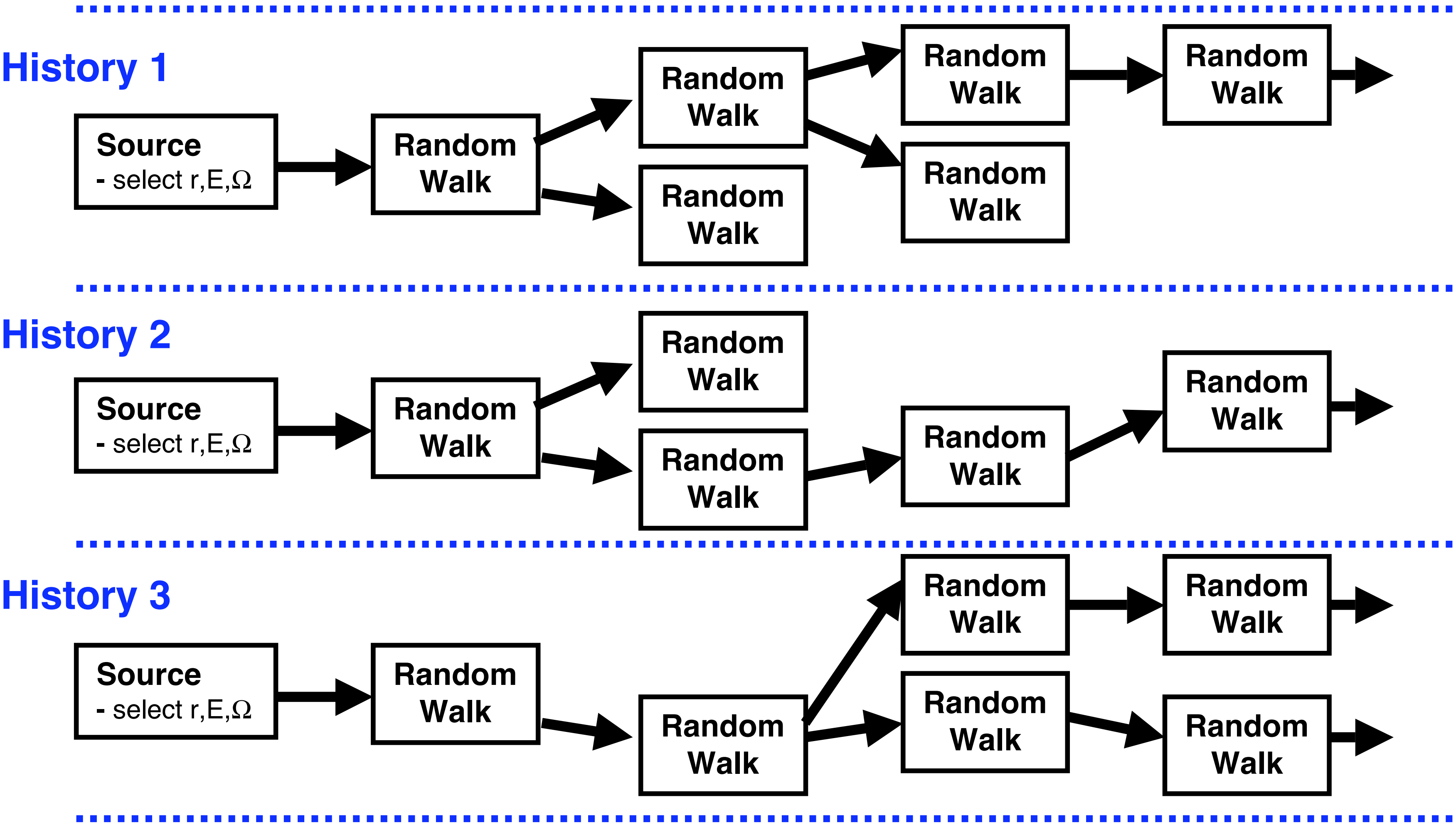
Random Walk

Random Walk

Random Walk

Random Walk

Fixed-source Monte Carlo Calculation



History 1

Source
- select r,E,Ω

Random Walk

Random Walk

Random Walk

Random Walk

Random Walk

Random Walk

History 2

Source
- select r,E,Ω

Random Walk

Random Walk

Random Walk

Random Walk

Random Walk

History 3

Source
- select r,E,Ω

Random Walk

Random Walk

Random Walk

Random Walk

Random Walk

Random Walk

Alpha Eigenvalue Equations

- For problems which are separable in space & time, it may be advantageous to solve a **static eigenvalue problem**, rather than a fully time-dependent problem
- **Assume:**
  1. **Fixed geometry & materials**
  2. **No external source:** **$Q(r,E,\Omega,t) = 0$**
  3. **Separability:** **$\psi(r,E,\Omega,t) = \Psi_\alpha(r,E,\Omega)\, e^{\alpha t}$,**

- Substituting ψ into the time-dependent transport equation yields

$$\left[\vec{\Omega}\cdot\nabla + \Sigma_T(\vec{r},E) + \frac{\alpha}{v}\right]\Psi_\alpha(\vec{r},E,\vec{\Omega}) = \iint \Psi_\alpha(\vec{r},E',\vec{\Omega}')\,\Sigma_S(\vec{r},E'\to E,\vec{\Omega}\cdot\vec{\Omega}')\,d\vec{\Omega}'dE'$$

$$+ \frac{\chi(E)}{4\pi}\iint \nu\Sigma_F(\vec{r},E')\,\Psi_\alpha(\vec{r},E',\vec{\Omega}')\,d\vec{\Omega}'dE'$$

- This is a **static** equation, an **eigenvalue problem for α and Ψ_α** without time-dependence
- α is often called the time-eigenvalue or time-absorption
- α-eigenvalue problems can be solved by Monte Carlo methods

Keff Eigenvalue Equations

- Another approach to creating a **static eigenvalue problem** from the time-dependent transport equation is to **introduce K_{eff}, a scaling factor on the multiplication (ν)**
- **Assume:**
  1. **Fixed geometry & materials**
  2. **No external source:** **$Q(r,E,\Omega,t) = 0$**
  3. **$\partial\psi/\partial t = 0$:** **$\nu \Rightarrow \nu/k_{eff}$**

- **Setting $\partial\psi/\partial t = 0$ and introducing the K_{eff} eigenvalue gives**

$$\left[\vec{\Omega}\cdot\nabla + \Sigma_T(\vec{r},E)\right]\Psi_k(\vec{r},E,\vec{\Omega}) = \iint \Psi_k(\vec{r},E',\vec{\Omega}')\Sigma_S(\vec{r},E'\to E,\vec{\Omega}\cdot\vec{\Omega}')d\vec{\Omega}'dE'$$

$$+ \quad \frac{1}{K_{eff}}\cdot\frac{\chi(E)}{4\pi}\iint \nu\Sigma_F(\vec{r},E')\Psi_k(\vec{r},E',\vec{\Omega}')d\vec{\Omega}'dE'$$

- This is a **static** equation, an **eigenvalue problem for** K_{eff} **and Ψ_k** without time-dependence
- K_{eff} is called the effective multiplication factor
- K_{eff} and Ψ_k should **never** be used to model time-dependent problems.
- K_{eff}-eigenvalue problems can be solved by Monte Carlo methods

Comments on K_{eff} and α Equations

- **Criticality**

Supercritical:	$\alpha > 0$	or	$K_{eff} > 1$
Critical:	$\alpha = 0$	or	$K_{eff} = 1$
Subcritical:	$\alpha < 0$	or	$K_{eff} < 1$

- **K_{eff} vs. α eigenvalue equations**
  - $\boldsymbol{\Psi_k(r,E,\Omega) \neq \Psi_\alpha(r,E,\Omega)}$, except for a critical system
  - α eigenvalue & eigenfunction used for time-dependent problems
  - K_{eff} eigenvalue & eigenfunction used for reactor design & analysis
  - Although $\alpha = (K_{eff}-1)/\Lambda$, where Λ = lifetime,

    there is **no** direct relationship between $\Psi_k(r,E,\Omega)$ and $\Psi_\alpha(r,E,\Omega)$

- K_{eff} eigenvalue problems can be solved **directly** using Monte Carlo

- α eigenvalue problems are solved by Monte Carlo **indirectly** using a series of K_{eff} calculations

Comments on K_{eff} and α Equations

K equation $[\, L + T \,]\, \Psi_k \;=\; [S \;+\; 1/k\ M \,]\, \Psi_k$

α equation $[\, L + T + \alpha/v \,]\, \Psi_\alpha \;=\; [S \;+\; M \,]\, \Psi_\alpha$

- The factor 1/k changes the relative level of the fission source
- The factor α/v changes the absorption & neutron spectrum
  - For $\alpha > 0$, more absorption at low E ⇒ harder spectrum
  - Double-density Godiva, average neutron energy causing fission:

    k calculation: 1.30 MeV

    α calculation: 1.68 MeV
- For separable problems, $\psi(r,E,\Omega,t) = \Psi_\alpha(r,E,\Omega)\, e^{\alpha t}$
- No similar equation for k, since not used for time-dependence

Power Iteration

&

Convergence

K-eigenvalue equation

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

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 <u>power iteration</u>

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

Power Iteration

Diffusion Theory or Discrete-ordinates Transport

1. Initial guess for K_{eff} and Ψ

   $K_{eff}^{(0)}, \quad \Psi^{(0)}$

2. Solve for $\Psi^{(n+1)}$

   Inner iterations over space or space/angle to solve for $\Psi^{(n+1)}$

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

3. Compute new K_{eff}

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

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

Monte Carlo

1. Initial guess for K_{eff} and Ψ

   $K_{eff}^{(0)}, \quad \Psi^{(0)}$

2. Solve for $\Psi^{(n+1)}$

   Follow particle histories to solve for $\Psi^{(n+1)}$

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

   During histories, save fission sites to use for source in next iteration

3. Compute new K_{eff}

   During histories for iteration (n+1), estimate $K_{eff}^{(n+1)}$

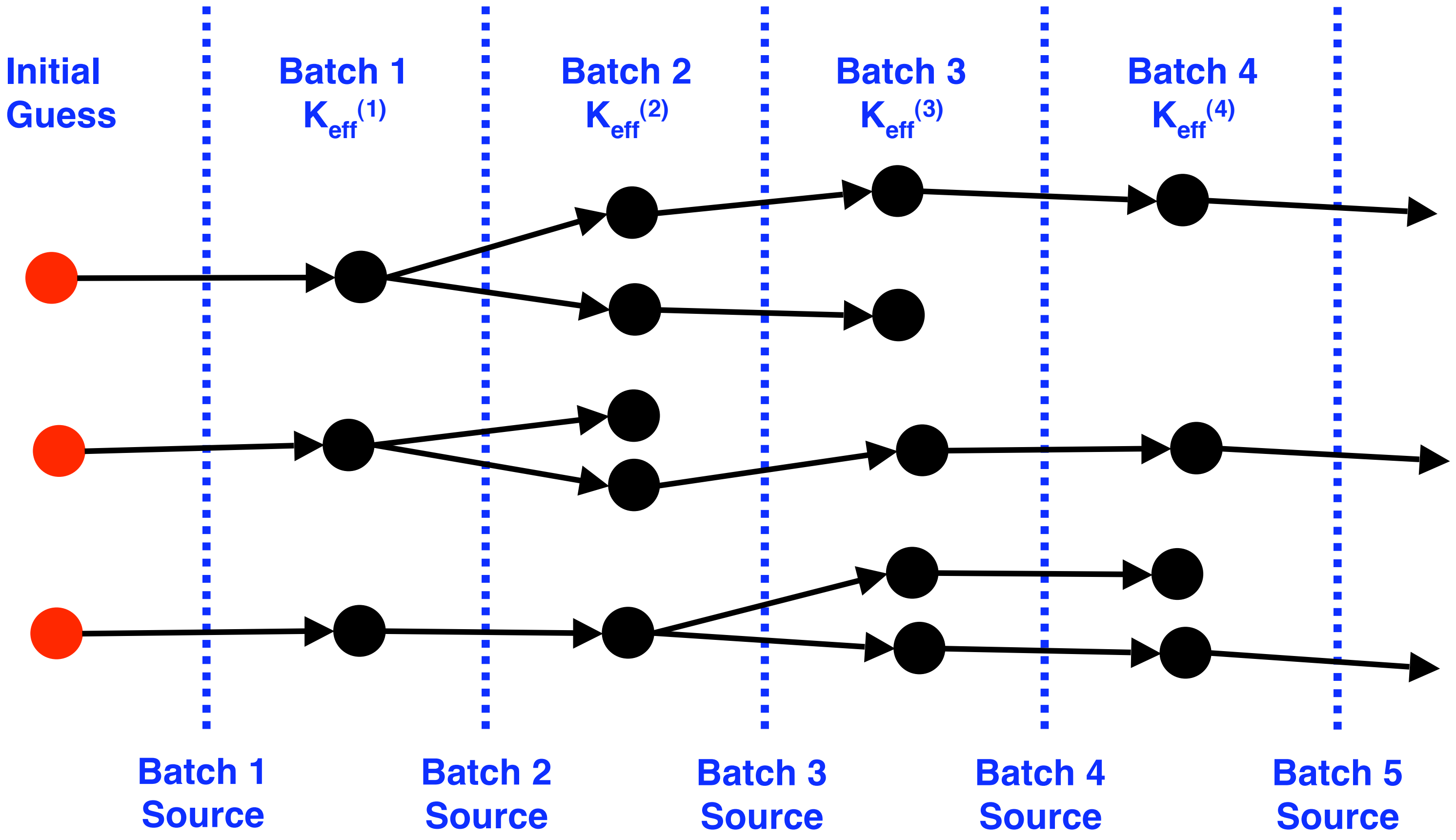
$$K_{eff}^{(n+1)} = K_{eff}^{(n)} \cdot \frac{\int M\Psi^{(n+1)} d\vec{r}}{\int M\Psi^{(n)} d\vec{r}}$$

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

5. Continue iterating, to compute tallies

Power Iteration

- Power iteration for Monte Carlo k-effective calculation



● Source particle generation

● Monte Carlo random walk

→ Neutron

α-Eigenvalue Calculations

- Eigenvalue equation with **both** K_{eff} & α
  - α **is a fixed number**, not a variable
  - Find the k-eigenvalue as function of α, **K(α)**

$$\left[\ \vec{\Omega}\cdot\nabla + \Sigma_T(\vec{r},E) + \frac{\alpha}{v}\ \right]\Psi(\vec{r},E,\vec{\Omega}) = \iint \Psi(\vec{r},E',\vec{\Omega}')\Sigma_S(\vec{r},E'\rightarrow E,\vec{\Omega}\cdot\vec{\Omega}')d\vec{\Omega}'dE'$$

$$+ \quad \frac{1}{K_{eff}}\cdot\frac{\chi(E)}{4\pi}\iint \nu\Sigma_F(\vec{r},E')\Psi(\vec{r},E',\vec{\Omega}')d\vec{\Omega}'dE'$$

- Note: If $\alpha < 0$
  - Real absorption plus time absorption could be negative
  - Move α/v to right side to prevent negative absorption,
  - -α/v term on right side is treated as a delta-function source

- Select a fixed value for α
- Solve the K-eigenvalue equations, with fixed time-absorption α/v
- Select a different α and solve for a new Keff
- Repeat, searching for value of α which results in Keff = 1

K- and α-Eigenvalue Calculations

- **K-eigenvalue solution**

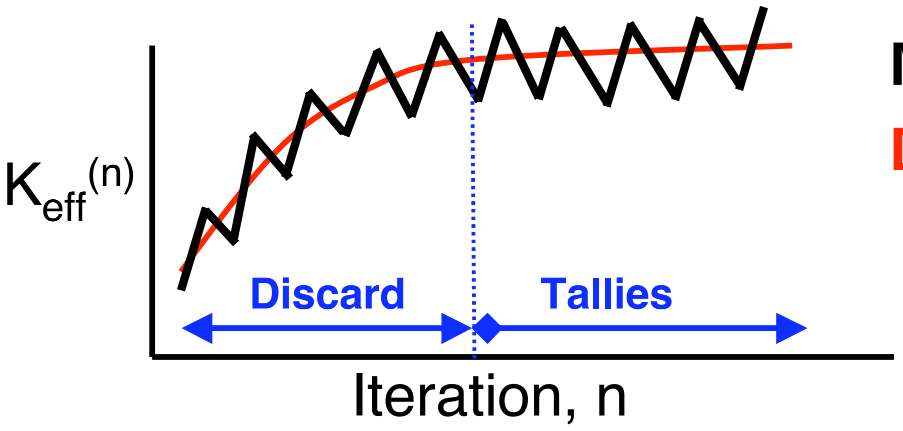
```
Loop for Power Iteration for K
•           Loop over neutrons in cycle
•           •           neutron history
•           • • •                 Monte Carlo
• • •
```

- **α-eigenvalue solution**

```
Loop for α search iterations
•           Loop for Power Iteration for K
•           •           Loop over neutrons in cycle
•           •           •           neutron history
•           •           • • •                 Monte Carlo
•           • • •
• • •
```

⇒ Find K(α), then solve for α that gives K(α)=1

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?**

Power Iteration – Convergence

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

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

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

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

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

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

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

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

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

Power Iteration – Convergence

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

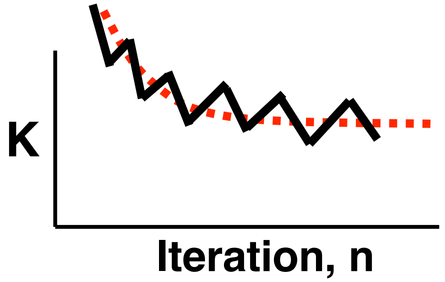
$$K^{(n+1)} \approx k_0 \cdot \left[1 + \left(\frac{a_1^{(0)}}{a_0^{(0)}} \right) \cdot \left(\frac{k_1}{k_0} \right)^{n} \cdot \left(\frac{k_1}{k_0} - 1 \right) \cdot G_1 + \ldots \right]$$

- **Because $k_0 > k_1 > k_2 > \ldots$, all of the red terms vanish as n→∞**
  - **$\Psi^{(n+1)} \rightarrow$ constant · u_0**
  - **$K^{(n+1)} \rightarrow k_0$**
- **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 $\Psi^{(n)}$ die off as ~ $(DR)^n$**
- **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**

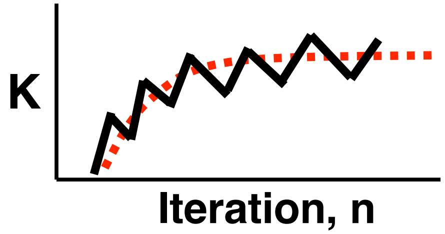
Typical K-effective convergence patterns

- **Higher mode error terms die out as (k_J / k_0)^n, for n iterations**

- **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)

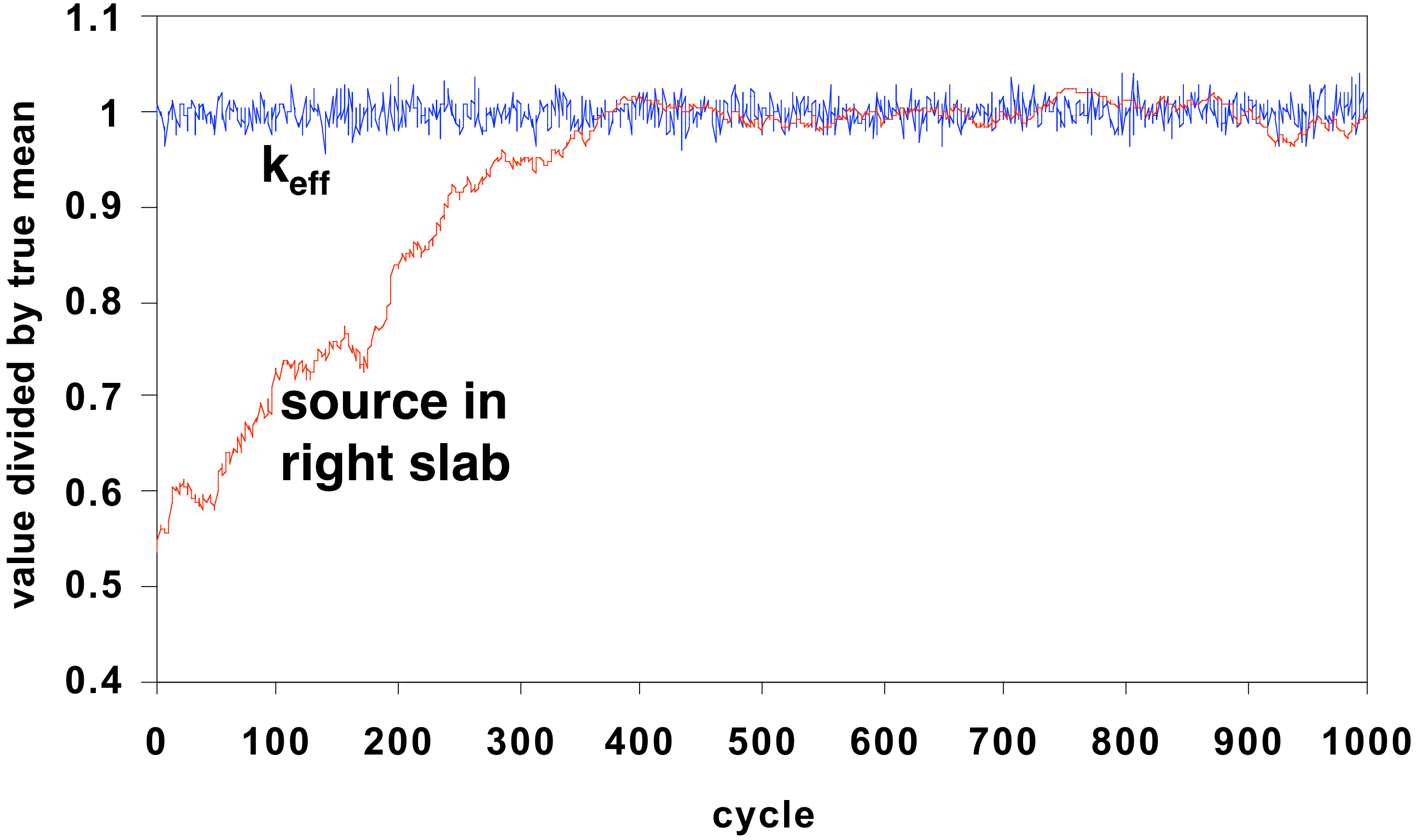


- The **Sandwich Method** uses 2 K_{eff} calculations - one starting too high & one starting too low. Both calculations should converge to the same result.

Power Iteration – Convergence

- **Keff is an integral quantity – converges faster than source shape**

Keff calculation for 2 nearly symmetric slabs, with Dominance Ratio = .9925



Shannon Entropy of the Fission Source Distribution For Assessing Convergence

The Challenge

- In the old days, when people used Monte Carlo just to compute K-effective, plots of k_{cycle} vs cycle were adequate to judge convergence

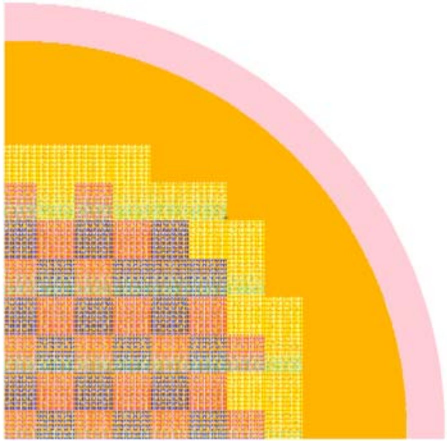
- Today, for computing power distributions & localized reaction rates, <u>new tools are needed to judge local convergence of source distribution</u>

  - K-effective converges before the source distribution converges

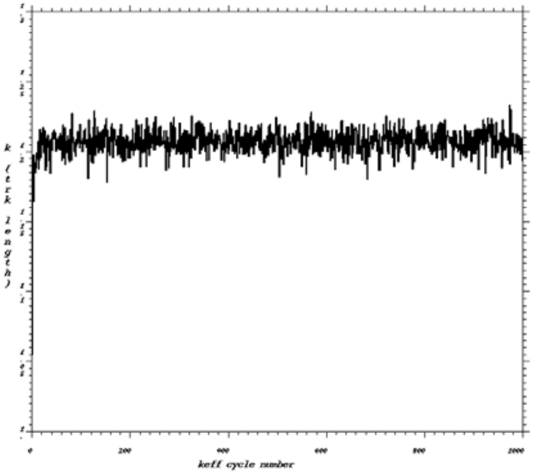
  - How do you tell if a 3D distribution has converged ?

Source Distribution Convergence

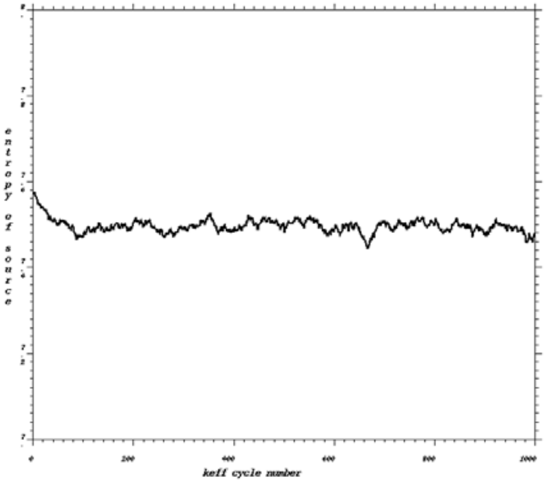
Geometry Model (1/4)



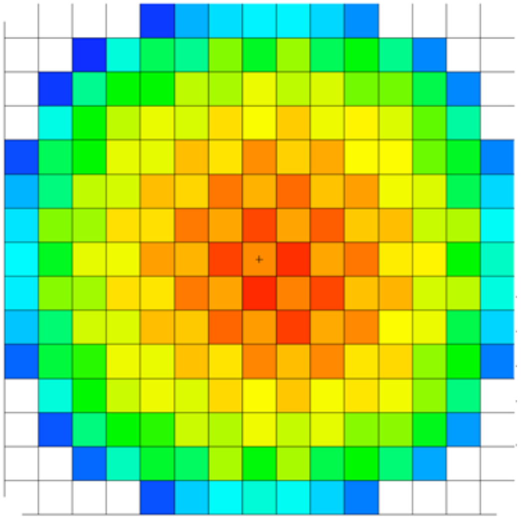
K vs cycle



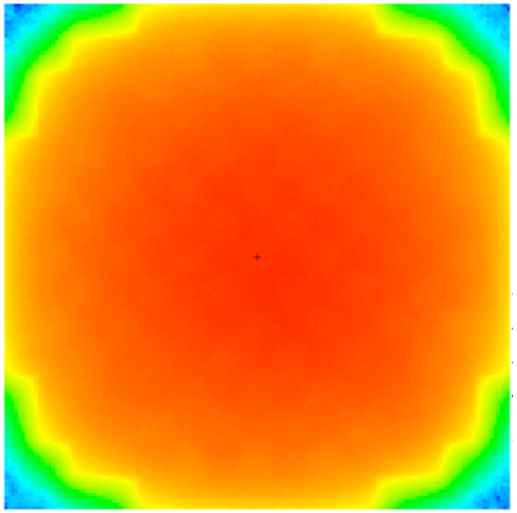
H_{src} vs cycle



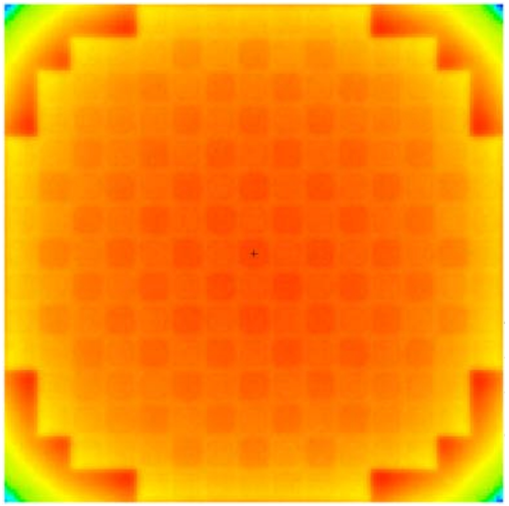
Assembly Powers



Fast Flux

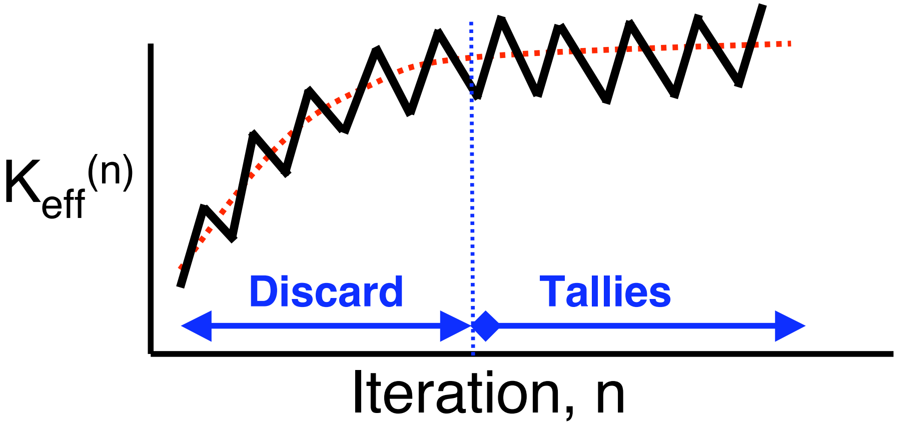


Thermal Flux



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?**

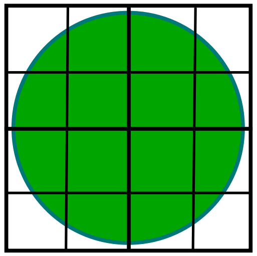


- **Analysis of the power iteration method shows that Keff is not a reliable indicator of convergence** — K_{eff} can converge faster than the source shape

- **Based on concepts from information theory (not physics), Shannon entropy of the source distribution is useful for characterizing the convergence of the source distribution**

K_{eff} Calculations – Stationarity Diagnostics

- 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**

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

K_{eff} Calculations – Stationarity Diagnostics

- **Shannon entropy of the source distribution**

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

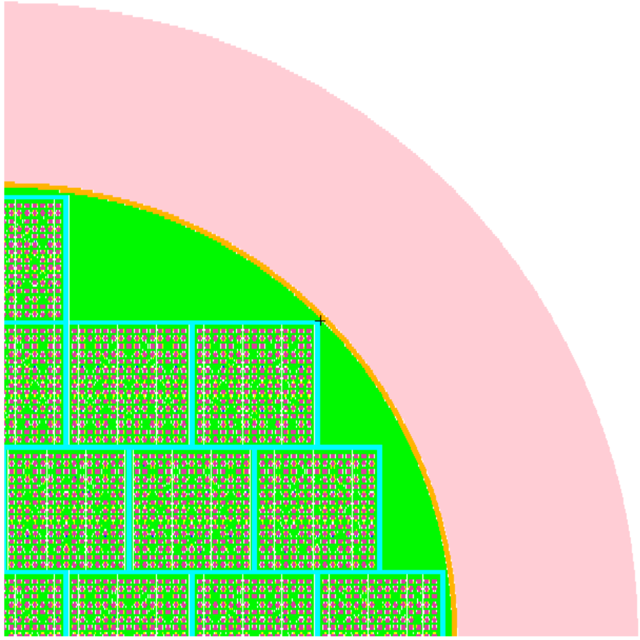
  - **$0 \ \leq \ H(S) \ \leq \ \ln_2(\ N_S\ )$**
  - **For a uniform source distribution, $\quad H(S) = \ln_2(\ N_S\ )$**
    since $p_1 = p_2 = \ldots = p_{Ns} = 1/N_S$
  - **For a point source (in a single bin), $\quad H(S) = 0$**

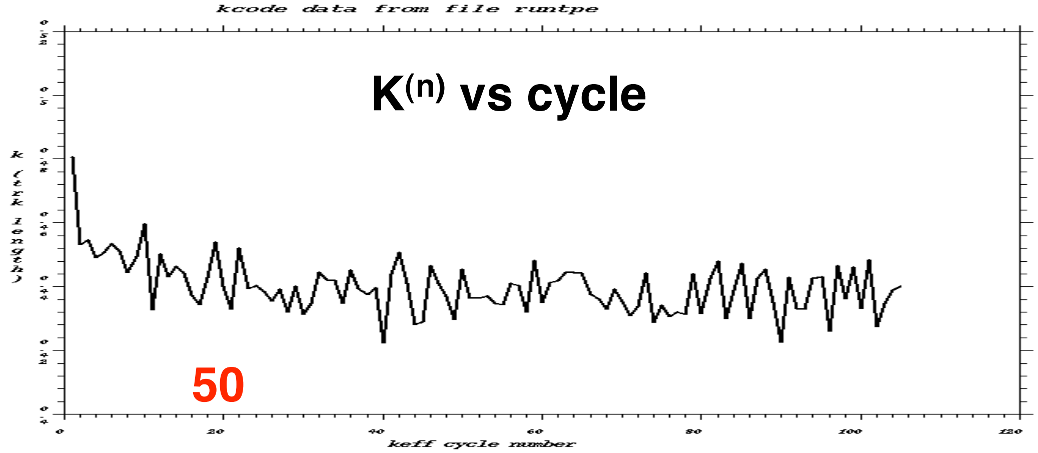
- **$H(S^{(n)})$ provides a single number to characterize the source distribution for iteration n (no physics!)**

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

K_{eff} Calculations – Stationarity Diagnostics

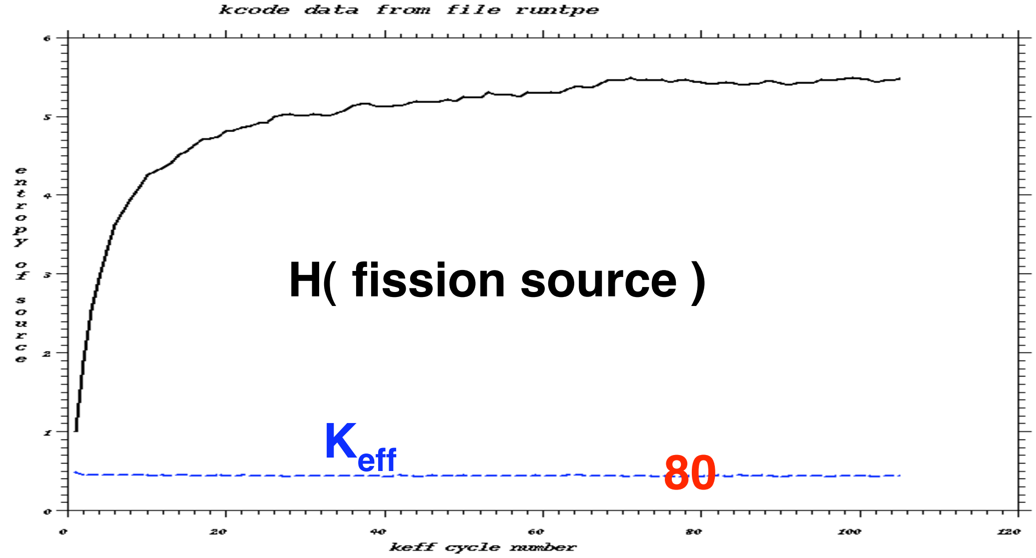
- **Example – Reactor core (Problem inp24)**





$K^{(n)}$ vs cycle

50



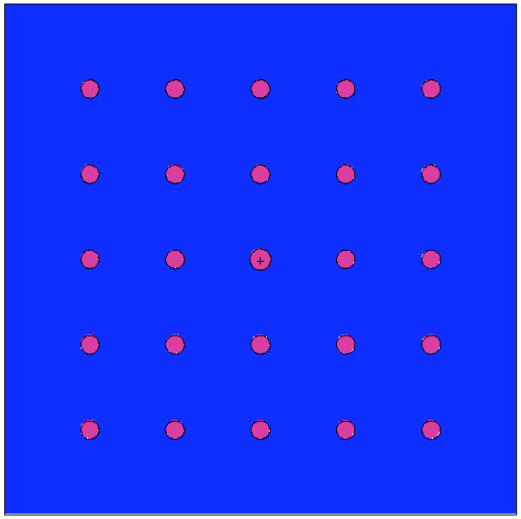
H(fission source)

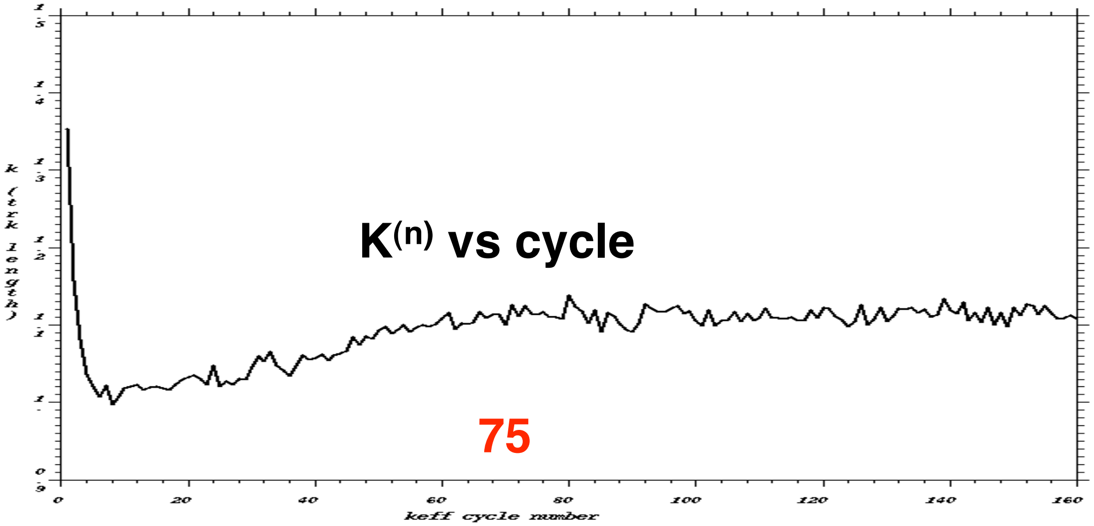
K_{eff}

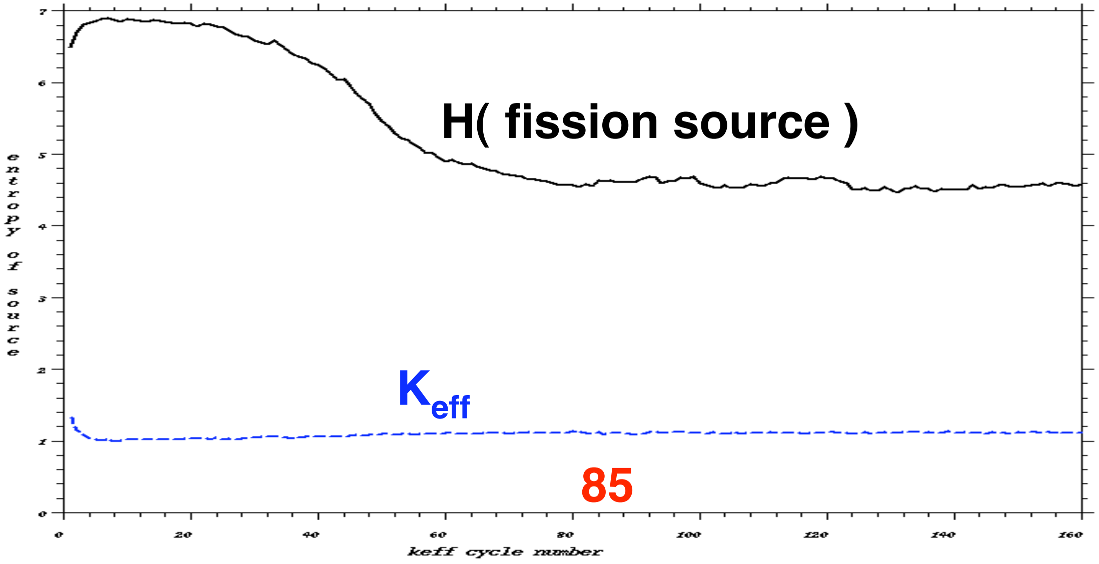
80

K_{eff} Calculations – Stationarity Diagnostics

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

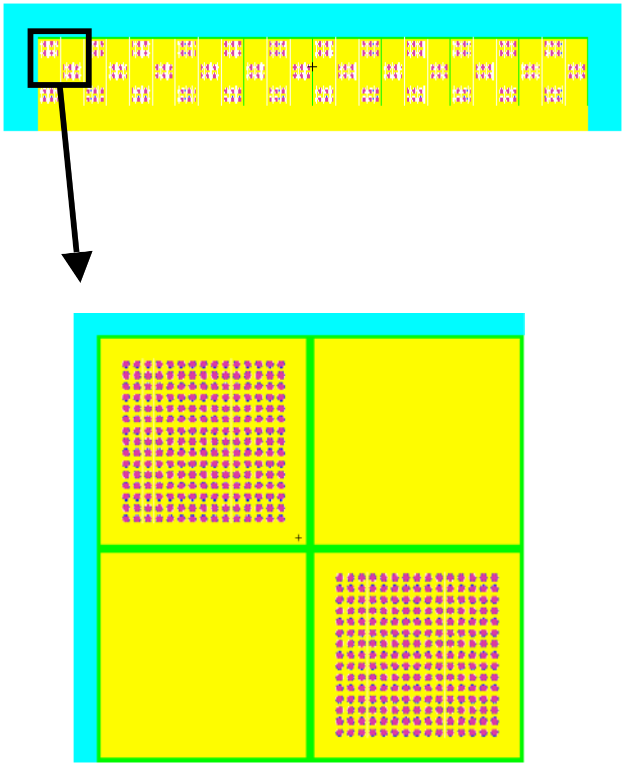


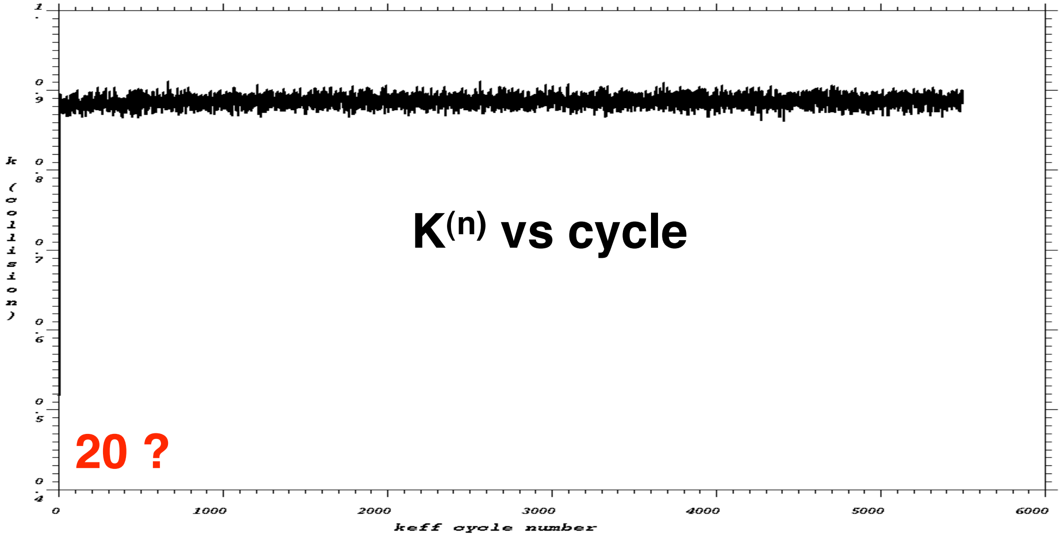


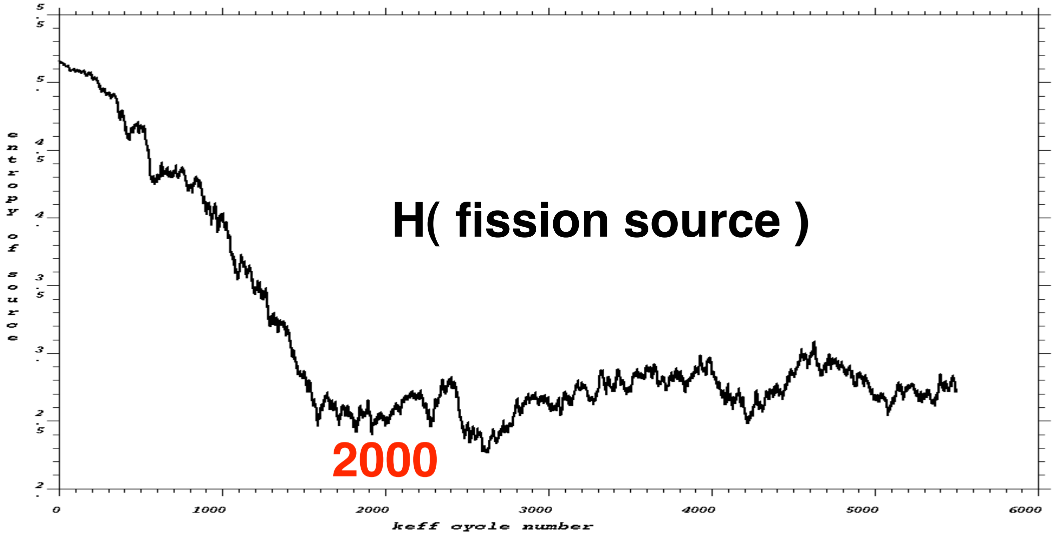


K_{eff} Calculations – Stationarity Diagnostics

- Example – Fuel Storage Vault (Problem OECD_bench1)

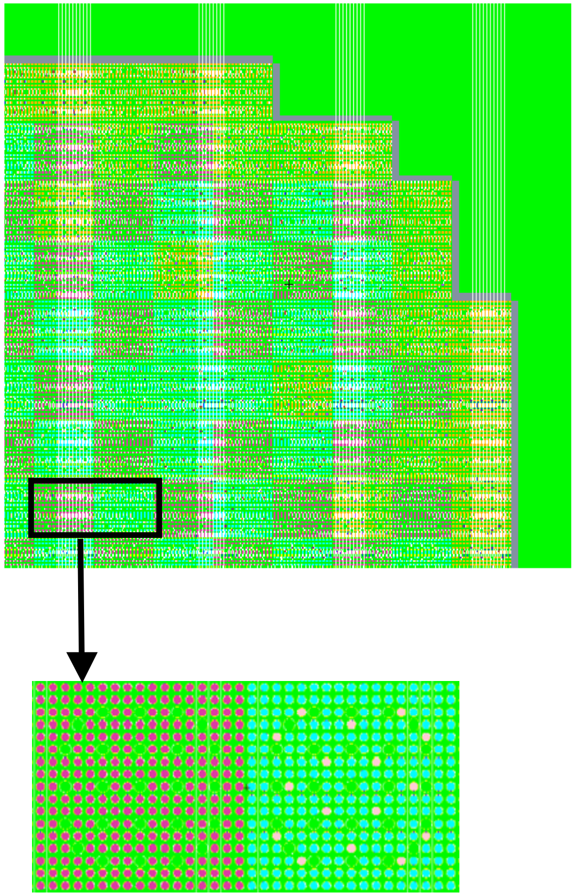


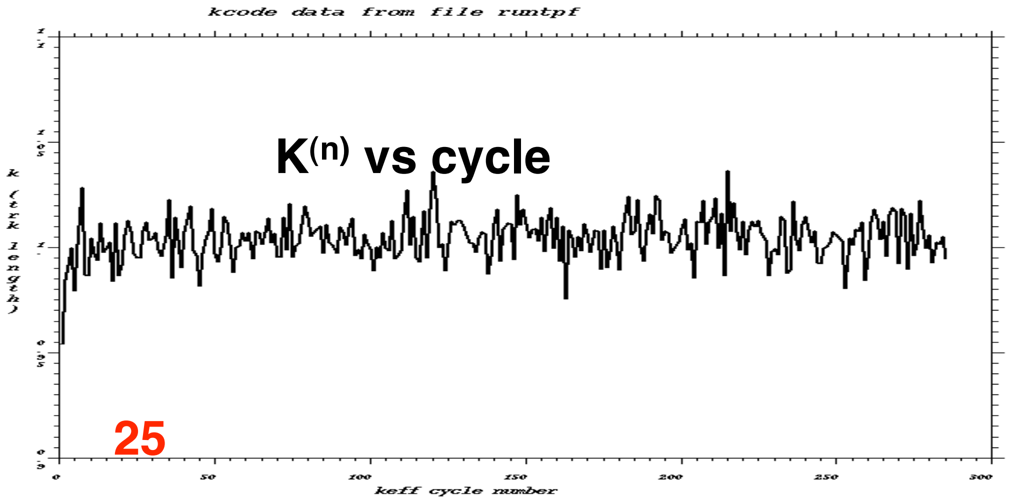


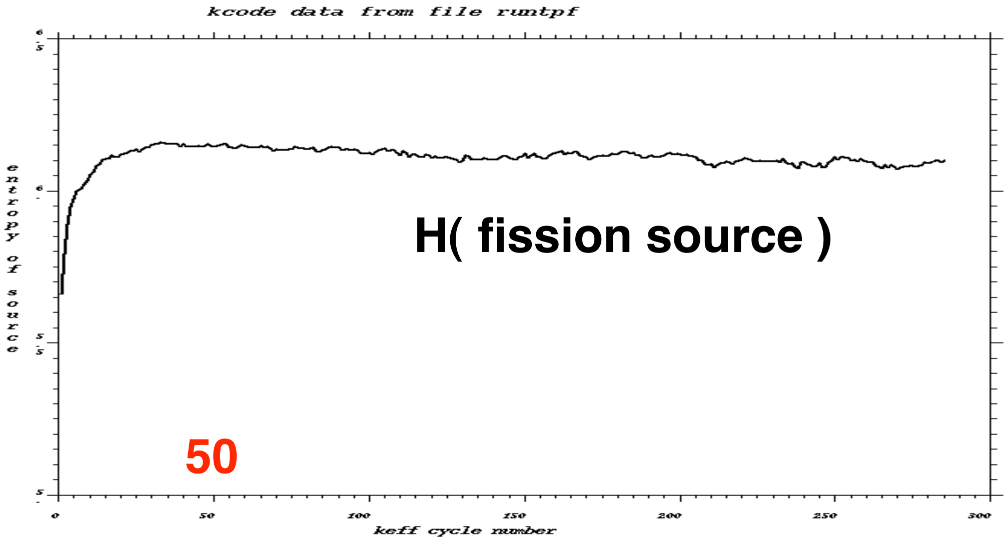


K_{eff} Calculations – Stationarity Diagnostics

- Example – PWR 1/4-Core   (Napolitano)

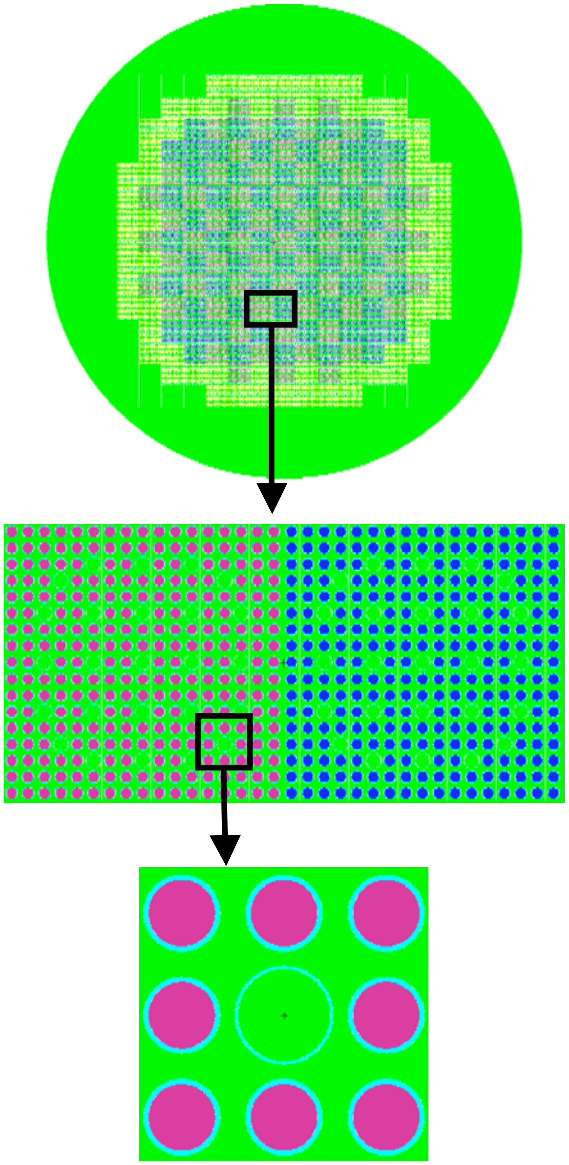


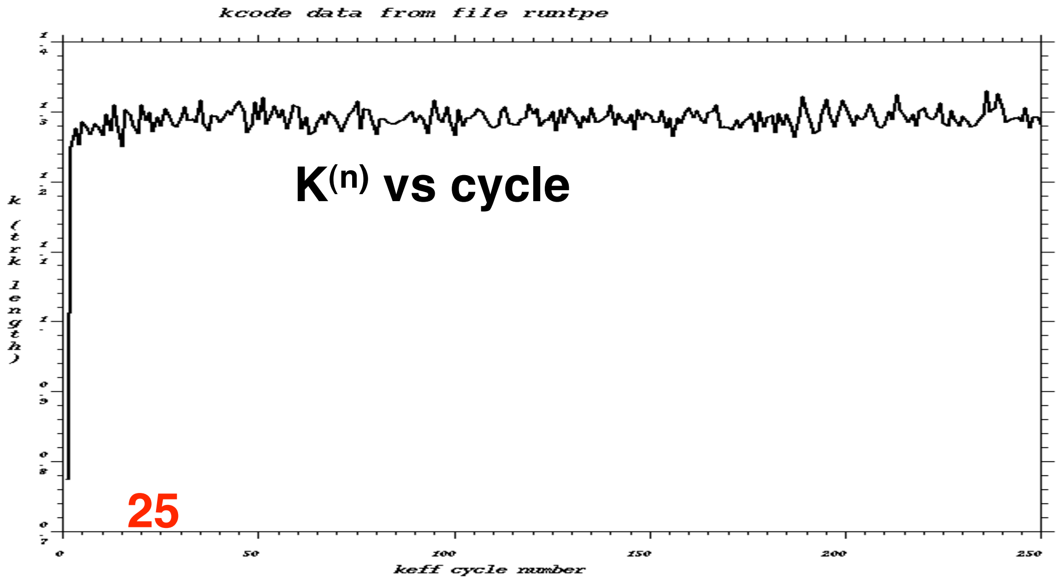




K_{eff} Calculations – Stationarity Diagnostics

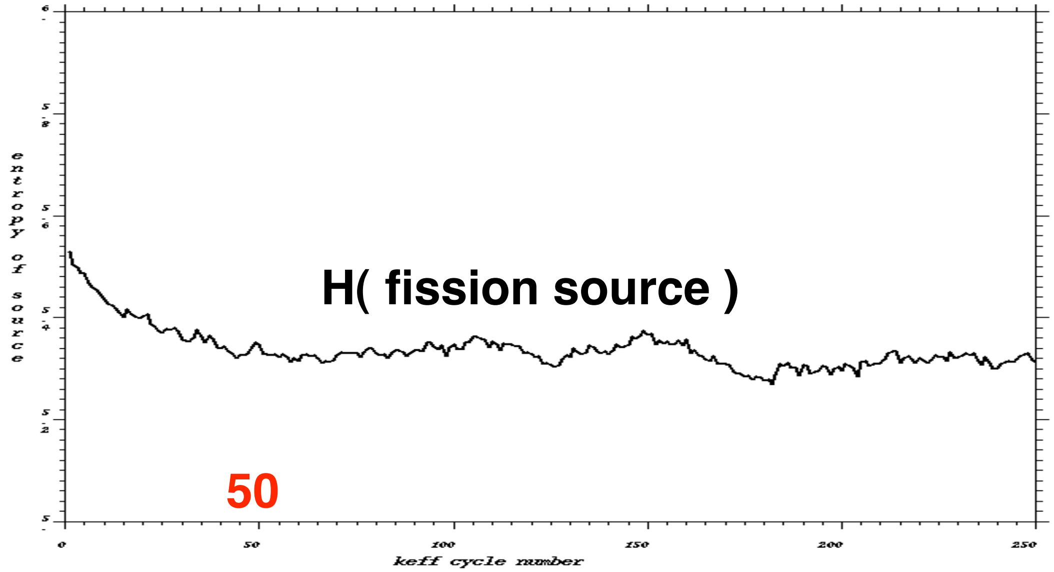
- **Example – 2D PWR (Ueki)**





K(n) vs cycle

25



H(fission source)

50

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 $<H_{src}> \pm 2\sigma_H$
  - Then, checks to see if at least that many cycles were discarded

Summary

- Local errors in the source distribution decay as $(k_J/k_0)^n$
  - Higher eigenmodes die out rapidly, convergence dominated by k_1/k_0
  - High DR → slow convergence
  - High DR → large correlation → large error in computed variances
- Errors in K_{eff} decay as $(k_J/k_0 - 1) * (k_J/k_0)^n$
  - High DR → $k_J/k_0 \sim 1$ → small error
- K_{eff} errors die out faster than local source errors
  - K_{eff} is an integral quantity – positive & negative fluctuations cancel
- Shannon entropy of the fission source distribution (H_{src}) is an effective diagnostic for source convergence
  - Now part of standard MCNP5 (beginning with version 1.40, November 2005)
  - Basis for initial source convergence tests — more are coming

⇒ If local tallies are important (e.g., assembly power, pin power, …), examine convergence using H_{src} - not just K_{eff} convergence

Wielandt Acceleration

Wielandt Method

- **Basic transport equation for eigenvalue problems**

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

L = loss to leakage

T = loss to collisions

S = gain from scatter-in

M = gain from fission multiplication

- Define a fixed parameter $\mathbf{k_e}$ such that $\mathbf{k_e > k_0}$ (k_0 = exact eigenvalue)

- **Subtract** $\frac{1}{k_e} M\Psi$ **from each side of the transport equation**

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

- **Solve the modified transport equation by power iteration**

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

Wielandt Method

- **Power iteration for modified transport equation**

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

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

$$\Psi^{(n+1)} = \tfrac{1}{\tilde{K}^{(n)}} \cdot \tilde{F}\Psi^{(n)}$$

$$\text{where} \quad \tilde{K}^{(n)} = (\tfrac{1}{K_{eff}^{(n)}} - \tfrac{1}{k_e})^{-1} \quad \text{or} \quad K_{eff}^{(n)} = (\tfrac{1}{\tilde{K}^{(n)}} + \tfrac{1}{k_e})^{-1}$$

- **How to choose k_e**
  - **k_e must be larger than k_0** (but, don't know k_0!)
  - **k_e must be held constant for all of the histories in a batch, but can be adjusted between batches**
    - Typically, guess a large initial value for k_e, such as k_e=5 or k_e=2
    - Run a few batches, keeping k_e fixed, to get an initial estimate of K_{eff}
    - Adjust k_e to a value slightly larger than the estimated K_{eff}
    - Run more batches, possibly adjusting k_e if the estimated K_{eff} changes

Wielandt Method

- **Convergence**
  - Eigenfunctions for the Wielandt method are same as for basic power iteration
  - Eigenvalues are shifted:

$$\tilde{k}_J = \left[\frac{1}{k_J} - \frac{1}{k_e}\right]^{-1} \qquad\qquad k_e > k_0 > k_1 > \ldots$$

  - Expand the initial guess, substitute into Wielandt method, rearrange to:

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

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

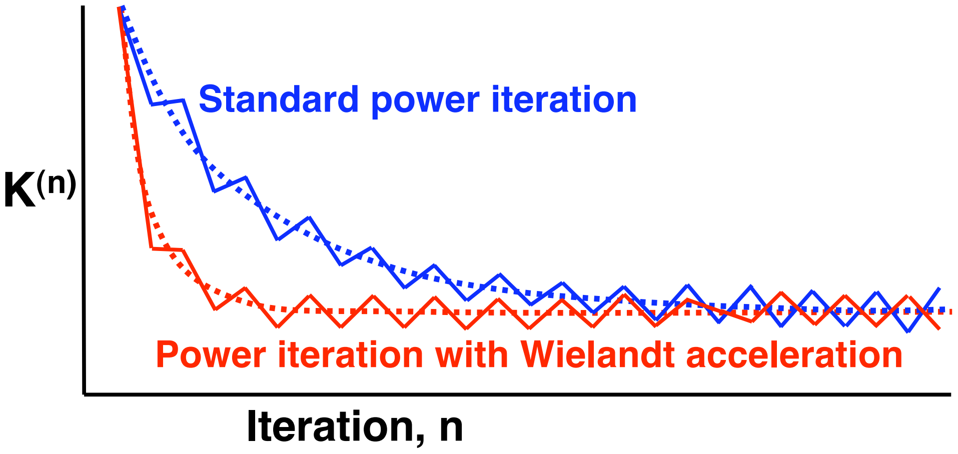
  - **Additional factor $(k_e - k_0)/(k_e - k_1)$ is less than 1 and positive, so that the red terms die out faster than for standard power iteration**

Wielandt Method

- The **dominance ratio** for this modified power iteration is

$$DR' = \frac{\tilde{k}_1}{\tilde{k}_0} = \frac{[\frac{1}{k_1} - \frac{1}{k_e}]^{-1}}{[\frac{1}{k_0} - \frac{1}{k_e}]^{-1}} = \frac{k_e - k_0}{k_e - k_1} \cdot \frac{k_1}{k_0} = \frac{k_e - k_0}{k_e - k_1} \cdot DR$$

  - Since $k_e > k_0$ and $k_0 > k_1$, **DR' < DR**
  - DR of Wielandt method is always **smaller** than standard power iteration
- **Wielandt acceleration improves the convergence rate of the power iteration method for solving the k-eigenvalue equation**



⇒ **Weilandt method converges at a faster rate than power iteration**

Wielandt Method

- **Monte Carlo procedure for Wielandt acceleration**

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

- **For standard Monte Carlo (power iteration) in generation n+1**
  - When a collision occurs, the expected number of fission neutrons produced is

$$n_F = \left\lfloor wgt \cdot \frac{\nu\Sigma_F}{\Sigma_T} \cdot \frac{1}{K^{(n)}} + \xi \right\rfloor$$

  - **Store n_F copies of particle in the fission-bank for the next generation (n+2)**

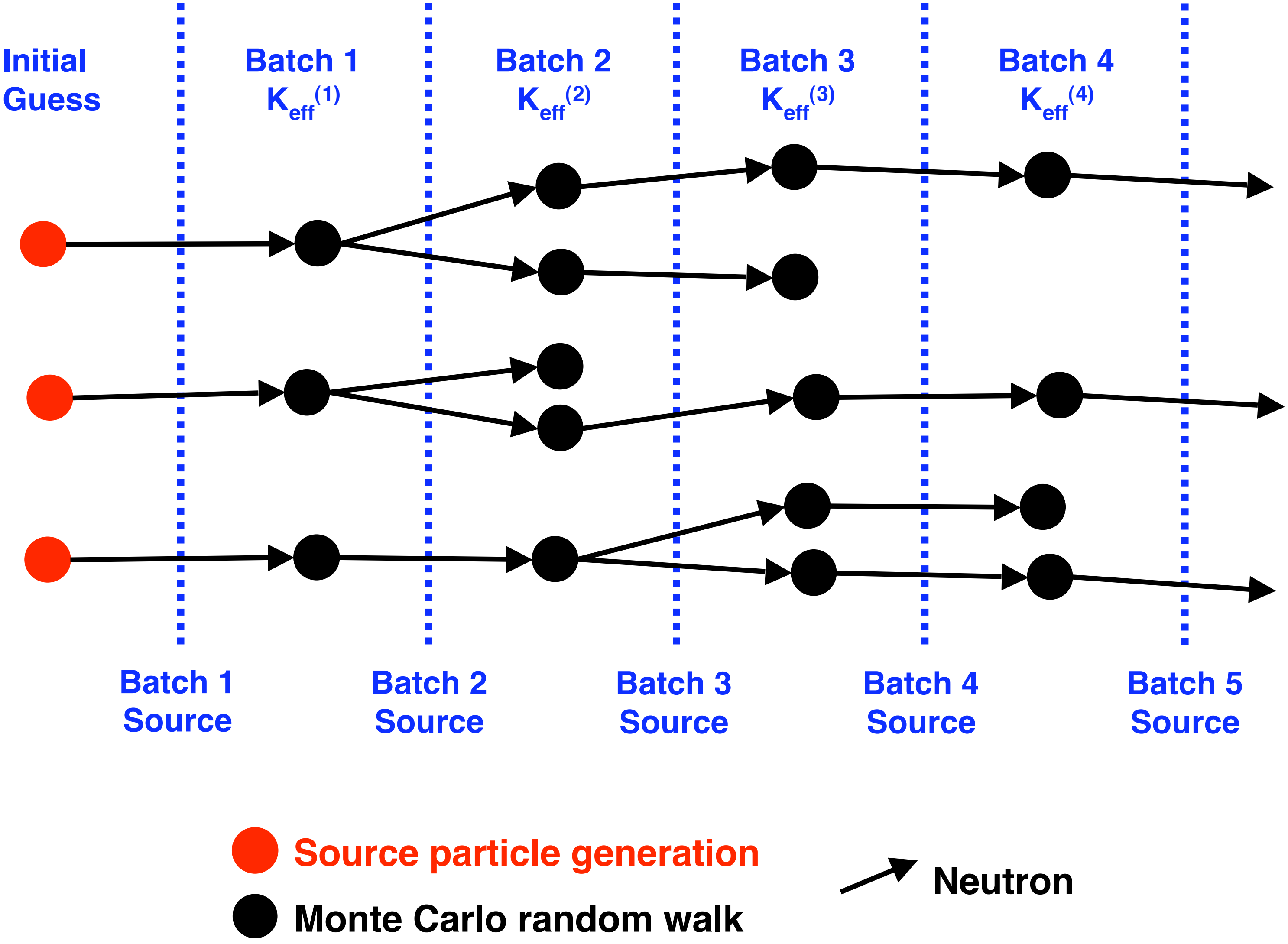
- **For Monte Carlo Wielandt method in generation n+1**
  - When a collision occurs, compute **2** expected numbers of fission neutrons

$$n'_F = \left\lfloor wgt \cdot \frac{\nu\Sigma_F}{\Sigma_T} \cdot \left(\frac{1}{K^{(n)}} - \frac{1}{k_e} \right) + \xi \right\rfloor \qquad n'_e = \left\lfloor wgt \cdot \frac{\nu\Sigma_F}{\Sigma_T} \cdot \frac{1}{k_e} + \xi \right\rfloor$$

  - Note that $E[\, n'_F + n'_e \,] = E[\, n_F \,]$
  - **Follow n'_e copies of the particle in the current generation (n+1)**
  - **Store n'_F copies of particle in the fission-bank for the next generation (n+2)**

Wielandt Method

- Power iteration for Monte Carlo k-effective calculation



Initial Guess

Batch 1 $K_{eff}^{(1)}$

Batch 2 $K_{eff}^{(2)}$

Batch 3 $K_{eff}^{(3)}$

Batch 4 $K_{eff}^{(4)}$

Batch 1 Source

Batch 2 Source

Batch 3 Source

Batch 4 Source

Batch 5 Source

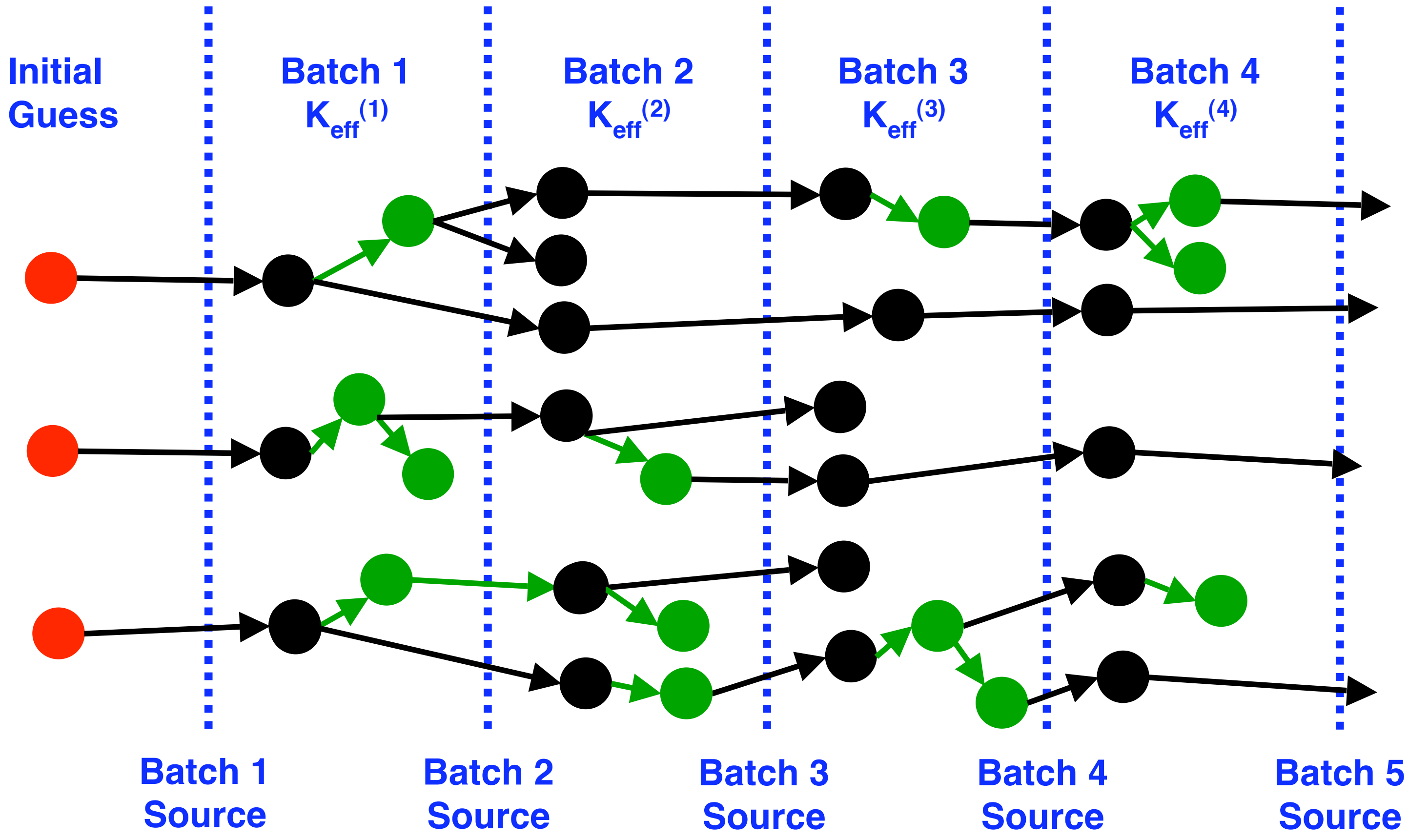
● Source particle generation

● Monte Carlo random walk

Neutron

Wielandt Method

- <u>Wielandt method</u> for Monte Carlo k-effective calculation



Initial Guess

Batch 1 $K_{eff}^{(1)}$

Batch 2 $K_{eff}^{(2)}$

Batch 3 $K_{eff}^{(3)}$

Batch 4 $K_{eff}^{(4)}$

Batch 1 Source

Batch 2 Source

Batch 3 Source

Batch 4 Source

Batch 5 Source

🔴 Source particle generation

⚫ Monte Carlo random walk

➚ ➚ Neutron

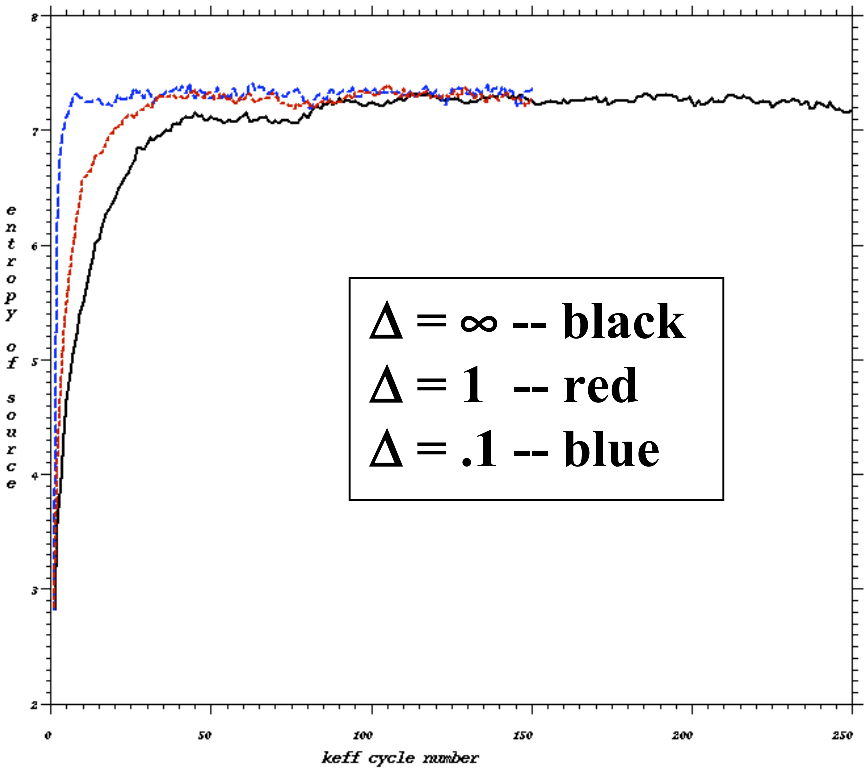
🟢 Additional Monte Carlo random walks within generation due to Wielandt method

MCNP5 Testing of Wielandt's Method

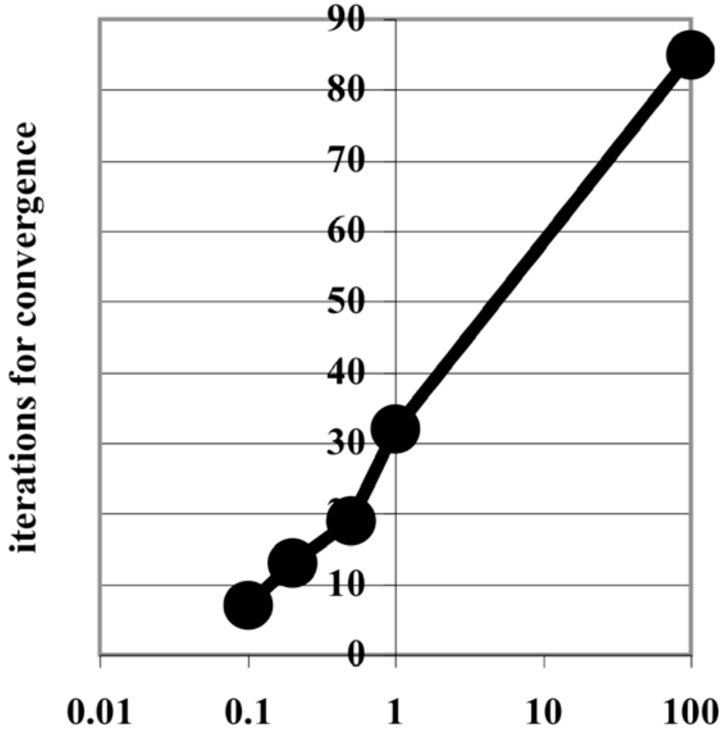
- **Wielandt shift parameter**

$$K_e^{(n+1)} = K^{(n)}_{collision} + \Delta$$

Convergence of H_{src} vs Δ



Iterations for convergence



Wielandt Method

Summary

- Wielandt Method has a lower DR than power iteration
  - Faster convergence rate than power iteration ⇒ fewer iterations
  - Some of the particle random walks are moved from the next generation into the current generation ⇒ more work per iteration
  - Same total number of random walks ⇒ no reduction in CPU time
- Advantages
  - Reduced chance of false convergence for very slowly converging problems
  - Reduced inter-generation correlation effects on variance
  - Fission source distribution spreads more widely in a generation (due to the additional particle random walks), which should result in more interactions for loosely-coupled problems
- Wielandt method will be included in next version of MCNP5

References

F. B. Brown, "Fundamentals of Monte Carlo Particle Transport," LA-UR-05-4983, available at http://mcnp.lanl.gov (2005).

Monte Carlo k-effective Calculations

J. Lieberoth, "A Monte Carlo Technique to Solve the Static Eigenvalue Problem of the Boltzmann Transport Equation," *Nukleonik* **11**,213 (1968).

M. R. Mendelson, "Monte Carlo Criticality Calculations for Thermal Reactors," *Nucl. Sci Eng.* **32**, 319–331 (1968).

H. Rief and H. Kschwendt, "Reactor Analysis by Monte Carlo," *Nucl. Sci. Eng.*, **30**, 395 (1967).

W. Goad and R. Johnston, "A Monte Carlo Method for Criticality Problems," *Nucl. Sci. Eng.* **5**, 371–375 (1959).

Superhistory Method

R.J. Brissenden and A.R. Garlick, "Biases in the Estimation of Keff and Its Error by Monte Carlo Methods," *Ann. Nucl. Energy*, Vol 13, No. 2, 63–83 (1986).

Wielandt Method

T Yamamoto & Y Miyoshi, "Reliable Method for Fission Source Convergence of Monte Carlo Criticality Calculation with Wielandt's Method", *J. Nuc. Sci. Tech.*, **41**, No. 2, 99–107 (Feb 2004).

S Nakamura, Computational Methods in Engineering and Science, R. E. Krieger Pub. Company, Malabar, FL (1986).