Title: A Review of Monte Carlo Criticality Calculations - Convergence, Bias, Statistics

Author(s): Forrest B. Brown

A Review of Monte Carlo Criticality Calculations - Convergence, Bias, Statistics

Forrest B. Brown

Los Alamos National Laboratory, Los Alamos, NM, USA
The author & LANL greatly appreciate the inspiration & support from:

Ely Gelbard, who devoted much of his career to the theory & practice of Monte Carlo criticality calculations

Enrico Sartori, for his long-time support of Monte Carlo & various international expert study groups through the OECD / NEA Data Bank

US DOE Nuclear Criticality Safety Program

This review and related R&D work on Monte Carlo criticality calculations would not have been possible without them.
Review of Monte Carlo Criticality Calculations

- Introduction
- Power method for MC
- Convergence of $K_{eff}$ & fission source
- Bias in $K_{eff}$ & tallies
- Bias in confidence intervals
- Conclusions
Examples - Reactor Analysis with Monte Carlo

MIT
research reactor

ATR
(1/4 of geometry)

PWR
with TRISO fuel

VHTR

Accurate & explicit modeling at multiple levels

Pictures from mcnp plotter
Perspective

• Bigger, faster computers ➔ more Monte Carlo calculations ➔ better local statistics

• Principal uses of Monte Carlo have evolved:

  1960s: K-effective
  Today: K-effective, detailed 3D whole-core, depletion, reactor design parameters, …

➔ More important now than ever to address the fundamental theory & best practices for Monte Carlo criticality calculations

Longstanding problems with the fundamental theory:
– Convergence of $K_{\text{eff}}$ & source distribution
– Bias in $K_{\text{eff}}$ & tallies
– Bias in statistics on tallies
Power Method for Monte Carlo Criticality Calculations
K-eigenvalue equation

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

where

- \(L\) = leakage operator
- \(S\) = scatter-in operator
- \(T\) = collision operator
- \(M\) = fission multiplication operator

\cdot Rearrange

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

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

\[\Psi = \frac{1}{K_{\text{eff}}} \cdot F\Psi\]

\(\Rightarrow\) This eigenvalue equation will be solved by power iteration

\[\Psi^{(n+1)} = \frac{1}{K_{\text{eff}}^{(n)}} \cdot F\Psi^{(n)}\]
**Power Iteration**

**Diffusion Theory or Discrete-ordinates Transport**

Initial guess: \( \text{Keff}^{(0)}, \Psi^{(0)} \)

Outer iterations (n)
- Inner iterations to solve for \( \Psi^{(n+1)} \)
- \((L + T - S)\Psi^{(n+1)} = \frac{1}{\text{Keff}^{(n)}} M \Psi^{(n)}\)
- Solve linear equations or sweep through space/angle mesh
- Compute new Keff
  \[ K_{\text{eff}}^{(n+1)} = K_{\text{eff}}^{(n)} \cdot \frac{\int M \Psi^{(n+1)}}{\int M \Psi^{(n)}} \]
- Renormalize \( \Psi^{(n+1)} \)
- If converged \( \Rightarrow \) stop
- ...

**Monte Carlo**

Initial guess: \( \text{Keff}^{(0)}, \Psi^{(0)} \)

Outer iterations (n)
- Follow histories to solve for \( \Psi^{(n+1)} \)
  \[ (L + T - S)\Psi^{(n+1)} = \frac{1}{\text{Keff}^{(n)}} M \Psi^{(n)}\]
- During histories, save fission sites to use for source in next iteration
- Compute new Keff
  Tally \( \text{Keff}^{(n+1)} \) during histories
- ...
- Renormalize \( \Psi^{(n+1)} \)
- If converged \( \Rightarrow \) turn on tallies
- If statistics small enough \( \Rightarrow \) stop
- ...

Diffusion Theory or Discrete-ordinates Transport

Monte Carlo
• Power iteration for Monte Carlo k-effective calculation
Concerns

- **Assessing convergence of Keff & fission distribution**
  - Keff and fission distribution converge differently
  - Both should be converged before beginning tallies

- **Bias in Keff & tallies**
  - Power iteration requires renormalization every cycle
  - MC renormalization involves dividing by a stochastic quantity, which introduces bias in Keff & tallies

- **Bias in uncertainties on tallies**
  - MC codes ignore cycle-to-cycle correlation when computing statistics
  - MC codes give statistics that are too small

**This talk:**
- Brief description & explanation for each concern
- Illustrate magnitude using **realistic PWR quarter-core**
- Discuss practical approaches to avoid the problems
Example Problem

2D quarter-core PWR  (Nakagawa & Mori model)

• 48 1/4 fuel assemblies:
  – 12,738 fuel pins with cladding
  – 1206 1/4 water tubes for control rods or detectors

• Each assembly:
  – Explicit fuel pins & rod channels
  – 17x17 lattice
  – Enrichments: 2.1%, 2.6%, 3.1%

• Dominance ratio ~ .96

• 125 M active neutrons for each calculation
• ENDF/B-VII data, continuous-energy
• Tally fission rates in each quarter-assembly
Convergence of Source Distribution
Power Iteration Convergence

- Power iteration convergence is well-understood:
  
  \[ n = \text{cycle number}, \quad k_0, u_0 - \text{fundamental}, \quad k_1, u_1 - 1\text{st higher mode} \]

  \[
  \Psi^{(n)}(\vec{r}) = \tilde{u}_0(\vec{r}) + a_1 \cdot \rho^n \cdot \tilde{u}_1(\vec{r}) + \ldots
  \]

  \[
  k_{\text{eff}}^{(n)} = k_0 \cdot \left[ 1 - \rho^{n-1}(1 - \rho) \cdot g_1 + \ldots \right]
  \]

  - First-harmonic source errors die out as \( \rho^n, \quad \rho = k_1 / k_0 < 1 \)
  - First-harmonic \( K_{\text{eff}} \) errors die out as \( \rho^{n-1} (1 - \rho) \)
  - Source converges slower than \( K_{\text{eff}} \)

- Most codes only provide tools for assessing \( K_{\text{eff}} \) convergence

- MCNP5 also gives Shannon entropy of the source distribution, \( H_{\text{src}} \)
Source Convergence Diagnostic - $H_{src}$

- Divide the fissionable regions of the problem into $N_s$ spatial bins

- Shannon entropy of the source distribution

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

  - For a uniform source distribution, $H(S) = \ln_2(N_s)$
  - For a point source (in a single bin), $H(S) = 0$
  - For any general source, $0 \leq H(S) \leq \ln_2(N_s)$

$\Rightarrow$ As the source distribution converges in 3D space, a line plot of $H(S^{(n)})$ vs. $n$ (the iteration number) converges
• Use $K_{\text{eff}}$ vs cycle & $H_{\text{src}}$ vs cycle to assess convergence of both $K_{\text{eff}}$ and the fission distribution

• The number of cycles to converge is determined by:

  – Dominance ratio $\rho = k_1 / k_0$

  – Closeness of initial source guess to converged distribution
Source Convergence - Dominance Ratio

- Dominance ratio determines the rate of convergence
  \[ \rho > .9 \Rightarrow \text{many cycles to converge} \]
- To reduce the dominance ratio
  - Take advantage of problem symmetry & reflecting boundary, to eliminate some higher modes
    - PWR reactor example:
      - full core \( \rho \sim .98 \)
      - 1/2 core \( \rho \sim .97 \)
      - 1/4 core \( \rho \sim .96 \)
      - 1/8 core \( \rho \sim .94 \)
  - Use Wielandt method (when available) to increase the average number of generations per cycle, \( L \)
    - PWR 1/4 core example:
      - \( L = 1 \) \( \rho \sim .96 \)
      - \( L = 5 \) \( \rho \sim .83 \)
      - \( L = 10 \) \( \rho \sim .72 \)
      - \( L = 20 \) \( \rho \sim .57 \)
  - Smaller dominance ratio \( \Rightarrow \) fewer cycles to converge
Source Convergence - Initial Guess

• Better initial source guess $\Rightarrow$ fewer cycles to converge

• Typical
  
  – Point at center - terrible guess

  – Reactor:
    
    uniform in core region - good guess

  – Criticality Safety:
    
    points in each fissionable region, or
    uniform in each fissionable region - good guess
Convergence for Different Source Guesses

- $H_{src}$, initial source in center of center 1/4 assy
- $H_{src}$, initial source in center of diagonal 1/4 assys
- $H_{src}$, initial source uniform in core region

- $k_{eff}$, initial source in center of center 1/4 assy
- $k_{eff}$, initial source in center of diagonal 1/4 assys
- $k_{eff}$, initial source uniform in core region
Conclusions - Convergence

- If you are computing more than just $K_{\text{eff}}$ (e.g., local reaction rates, dose fields, fission distributions, heating distributions, etc.):
  - Should check both $k_{\text{eff}}$ and $H_{\text{src}}$ for convergence

- Use problem symmetry, if possible

- Use Wielandt method, when available

- Better initial source guess $\Rightarrow$ fewer cycles to converge
  - Reactor: uniform in core region
  - Criticality Safety: points in each fissionable region, or uniform in each fissionable region
Bias in Keff & Tallies
Bias in Keff

- **Power iteration is used for Monte Carlo Keff calculations**
  - For one cycle (iteration):
    - $M_0$ neutrons start
    - $M_1$ neutrons produced, $E[M_1] = K_{eff} \cdot M_0$
  - At end of each cycle, must **renormalize** by factor $\frac{M_0}{M_1}$
  - Dividing by stochastic quantity ($M_1$) introduces bias

- **Bias in Keff, due to renormalization**

\[
\text{bias in } K_{\text{eff}} = -\frac{\sigma_k^2}{K_{\text{eff}}} \cdot \left( \text{sum of lag-i correlation coeff's between batch K's} \right) \propto \frac{1}{M_0}
\]

Note: $\sigma_k^2 = \text{population variance}$; $\sigma_{K_{\text{eff}}}^2 = \frac{\sigma_k^2}{N}$

- **Run the reactor problem with different M (neutrons/cycle)**
  - 500, 1000, 5000, 10000, 20000
Bias in $K_{eff}$

- $N = \#$ cycles
- $M =$ neutrons/cycle
- $N \cdot M =$ constant for all calculations

Graph showing the relationship between $K_{eff}$ and $1/M$. The graph indicates that as $1/M$ increases, $K_{eff}$ decreases. At $M = 1000, 5000, 20000$, the bias in $K_{eff}$ is marked as 30 pcm.
### Bias in Fission Tallies

Percent errors in 1/4-assembly fission rates using 500 neutrons/cycle

<table>
<thead>
<tr>
<th></th>
<th>-0.5</th>
<th>-0.6</th>
<th>-0.2</th>
<th>-0.3</th>
<th>0.5</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.2</td>
<td>-0.7</td>
<td>-0.8</td>
<td>0.1</td>
<td>0.3</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>-0.5</td>
<td>-0.7</td>
<td>-0.7</td>
<td>0.0</td>
<td>0.3</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td>-0.4</td>
<td>-0.6</td>
<td>-0.5</td>
<td>0.0</td>
<td>-0.1</td>
<td>0.2</td>
<td>0.7</td>
</tr>
<tr>
<td>-0.7</td>
<td>-0.9</td>
<td>-0.8</td>
<td>-0.4</td>
<td>0.2</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>-0.6</td>
<td>-0.3</td>
<td>-0.7</td>
<td>-0.6</td>
<td>0.3</td>
<td>0.8</td>
<td>1.1</td>
</tr>
<tr>
<td>-0.5</td>
<td>-0.8</td>
<td>-1.0</td>
<td>-0.8</td>
<td>-0.5</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>-0.5</td>
<td>-0.9</td>
<td>-0.8</td>
<td>-1.0</td>
<td>-0.6</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>-0.9</td>
<td>-0.9</td>
<td>-1.1</td>
<td>-1.0</td>
<td>-0.9</td>
<td>-0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>-1.2</td>
<td>-1.3</td>
<td>-1.2</td>
<td>-1.0</td>
<td>-0.6</td>
<td>-0.5</td>
<td>-0.3</td>
</tr>
<tr>
<td>-1.3</td>
<td>-1.5</td>
<td>-1.0</td>
<td>-0.9</td>
<td>-0.7</td>
<td>-0.5</td>
<td>-0.6</td>
</tr>
<tr>
<td>-1.7</td>
<td>-1.5</td>
<td>-1.1</td>
<td>-1.1</td>
<td>-0.6</td>
<td>-0.5</td>
<td>-0.2</td>
</tr>
<tr>
<td>-1.5</td>
<td>-1.5</td>
<td>-1.4</td>
<td>-1.0</td>
<td>-1.1</td>
<td>-0.8</td>
<td>0.0</td>
</tr>
<tr>
<td>-1.6</td>
<td>-1.6</td>
<td>-1.2</td>
<td>-1.2</td>
<td>-0.6</td>
<td>-0.7</td>
<td>-0.4</td>
</tr>
</tbody>
</table>

RMS error = 1.1 %  
MCNP std deviations: .1% - .3%  
True std deviations: .3% - .8%

Errors of -1.7% to +3.2%  
Statistics ~ .1% to .3%

Reference: ensemble-average of 25 independent calculations, with 25 M neutrons each & 20K neutrons/cycle
Bias in Fission Tallies

Percent error in fission rates along diagonal
Bias in Keff & Tallies

- **Past work - eliminating bias**
  - **MacMillan**
    - Weight the tallies for each cycle \( n \) by
      \[
      W_n = \frac{\prod_{j=1}^{n-1} k_j}{K^{n-1}},
      \]
      where \( K = \left( \prod_{j=1}^{N} k_j \right)^{1/N} \), \( N = \) number of active cycles
    - Difficulty: Must save all tallies for all cycles, combine at end of problem
  - **Gast & Candelore**
    - Increase \( M \) (neutrons/cycle) each cycle by 10 neutrons
    - Difficulty: For finite number of cycles, bias still exists

- **Practical solution - use large \( M \) (neutrons/cycle)**
  - **Years ago**
    - Slow computers, \( M \sim 500 \) \( \Rightarrow \) bias could be a problem
  - **Today**
    - Fast computers, typically \( M \sim 10K \) or \( 100K \) \( \Rightarrow \) bias negligible
    - Large \( M \) gives more efficient parallel calculations
Bias in Keff & Tallies - Comments

• For reactor problem with 500 neutrons/cycle
  – Bias in Keff is ~ 30 pcm
  – Bias in the power distribution shows a significant tilt
  – Errors of -1.7 % to +3.2 % in power fractions
  – The bias is much larger than the MC uncertainties

• Bias in Keff & the fission distribution is smaller with 1000 neutrons per cycle, and negligible with 10,000 or more neutrons per cycle

• Practical solution - use large M (neutrons/cycle)
  – For M ~ 10K or more ⇒ bias negligible
  – Large M gives more efficient parallel calculations

• Wielandt's method also reduces bias
  – Reduces frequency of renormalizations, reduces correlation
Underprediction Bias in Confidence Intervals in Monte Carlo Keff Calculations
Tallies & Correlation

- MC eigenvalue calculations are solved by power iteration
  - A generation model is used in following neutron histories
  - Tallies from one generation (including K) are correlated with tallies in successive generations
  - The correlation is positive
Bias in $\sigma^2$

- For tally $X$, made $N$ times (for large $N$)

\[
\bar{X} = \frac{1}{N} \sum_{n=1}^{N} X_n
\]

\[
\tilde{\sigma}_X^2 = \frac{1}{N} \left( \sum_{n=1}^{N} X_n^2 - \bar{X}^2 \right)
\]

\[
\sigma_X^2 \approx \tilde{\sigma}_X^2 + \tilde{\sigma}_X^2 \cdot 2 \cdot \sum_{i=1}^{\infty} r_i
\]

\[
\text{True variance, including correlations}
\]

\[
r_i = \text{lag-i correlation coef. between } X_n \text{'s}
\]

- $(\text{True } \sigma^2) > (\text{computed } \sigma^2)$, since correlations are positive

\[
\frac{\text{True } \sigma_X^2}{\text{Computed } \sigma_X^2} = \frac{\sigma_X^2}{\tilde{\sigma}_X^2} \approx 1 + 2 \cdot \left( \text{sum of lag-i correlation coeff's between tallies} \right)
\]

Variance underprediction bias is independent of $N$ and $M$
Correlation vs Tallies

- MC codes ignore correlation in tallies when computing $\sigma^2$'s
- $\sigma^2$'s computed by MC codes are always too small

\[
\frac{\text{True } \sigma_{\chi}^2}{\text{Computed } \sigma_{\chi}^2} = 1 + 2 \cdot \left( \text{sum of lag-i correlation coeff's between tallies} \right)
\]

- The size of underprediction bias in $\sigma^2$'s depends on how tallies are performed:

  MCNP:
  - generation tallies for $K_{eff}$,
  - history tallies for everything else

  VIM, KENO, RACER, RCP, …:
  - generation tallies

  MCNP+Wielandt, MONK:
  - several generations

  Repeated MC runs, averaged:
  - all generations from each run

Larger    Correlation & Bias
None
### Bias in Uncertainties

True relative errors in 1/4-assembly fission rates, as multiples of calculated relative errors, $\frac{\sigma_{\text{TRUE}}}{\sigma_{\text{MCNP}}}$

<table>
<thead>
<tr>
<th></th>
<th>3.4</th>
<th>3.1</th>
<th>2.7</th>
<th>2.6</th>
<th>2.3</th>
<th>2.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>3.7</td>
<td>3.6</td>
<td>3.7</td>
<td>3.7</td>
<td>2.7</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>3.8</td>
<td>3.9</td>
<td>4.2</td>
<td>3.9</td>
<td>3.4</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>3.9</td>
<td>3.6</td>
<td>3.3</td>
<td>3.4</td>
<td>3.4</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>3.8</td>
<td>3.5</td>
<td>3.2</td>
<td>2.9</td>
<td>2.6</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td>3.2</td>
<td>3.5</td>
<td>2.6</td>
<td>3.0</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td>3.4</td>
<td>3.1</td>
<td>2.7</td>
<td>2.3</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>3.5</td>
<td>3.4</td>
<td>3.1</td>
<td>2.7</td>
<td>2.3</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>3.9</td>
<td>3.6</td>
<td>3.1</td>
<td>2.9</td>
<td>1.9</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>3.7</td>
<td>3.3</td>
<td>3.6</td>
<td>2.4</td>
<td>2.2</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>3.1</td>
<td>3.0</td>
<td>2.2</td>
<td>2.2</td>
<td>2.1</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>3.7</td>
<td>3.3</td>
<td>2.6</td>
<td>2.5</td>
<td>2.8</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>3.2</td>
<td>3.1</td>
<td>2.9</td>
<td>3.1</td>
<td>3.3</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td>3.0</td>
<td>3.6</td>
<td>3.4</td>
<td>3.5</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td>3.2</td>
<td>2.8</td>
<td>3.5</td>
<td>3.9</td>
<td>4.1</td>
</tr>
</tbody>
</table>

Average factor = 3.1

Calculated uncertainties are 1.7 to 4.7 times smaller than true uncertainties
Past Work - Bias in Uncertainties

\[ \frac{\text{True } \sigma_x^2}{\text{Computed } \sigma_x^2} = 1 + 2 \cdot \sum_{k=1}^{\infty} r_k \]

- **MacMillan (1973)** [similar approach by Gast in 1974]
  - Calculate \( r_1 \) for each tally (lag-1 inter-cycle correlation coefficient)
  - Assume dominance ratio \( \rho \) is known
  - Assume \( r_k \leq r_1 \cdot \rho^k \) for \( k=2,3,\ldots \)
  - Then,
    \[ \frac{\text{True } \sigma_x^2}{\text{Computed } \sigma_x^2} \leq 1 + \frac{2 \cdot r_1}{1 - \rho} \]
  - This factor can then be used to correct the computed \( \sigma \) for the tally

- **Difficulties:**
  - Only gives a conservative upper bound
  - Useless if \( \rho \) near 1.0
  - Requires extra storage for each tally
  - **Notoriously sensitive to noise** ....
  - Assumption for higher \( r_k \)'s may often be incorrect
  - Dominance ratio is usually not known
• Uncertainties computed by MC codes exhibit a bias due to inter-cycle correlation effects that are neglected in tallies

• Primarily affects local tally statistics, not K-effective statistics

• Computed uncertainties are always smaller than the true uncertainties for a tally

• Running more cycles or more neutrons per cycle does not reduce the biases

• Wielandt’s method can reduce or eliminate the underprediction bias in uncertainties (see next slide)
Reduced Uncertainty Bias, using Wielandt

Wielandt's method increases the fission chain-length in each cycle, and reduces inter-cycle correlations.

Run the problem using different amounts of Wielandt acceleration (different shift parameters) to get average chain-lengths of 5, 10, 20 generations per cycle.

Plot relative error in quarter-assemblies along diagonal.
Conclusions

New features for MCNP5 (soon)
  + Wielandt method
  + Dominance ratio calculation

Final remarks
Wielandt Method - Basics

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

\[
k_e = k_0 + \Delta, \quad \Delta > 0
\]

- Modify the transport equation & solve by power iteration

\[
(L + T - S - \frac{1}{k_e} M)\Psi^{(n)} = \left(\frac{1}{K_{\text{eff}}^{(n-1)}} - \frac{1}{k_e}\right)M\Psi^{(n-1)}
\]

- The dominance ratio for Wielandt method is always smaller than for power iteration

\[
\rho_{\text{Wielandt}} = \frac{k_e - k_0}{k_e - k_1} \cdot \rho_{\text{Power}} \quad \rho = \frac{k_1}{k_0} < 1, \quad k_e > k_0 > k_1 > ...
\]

\( \Rightarrow \) Wielandt method will converge in fewer iterations

\( \Rightarrow \) Reduces inter-cycle correlation, hence improves statistics
Wielandt Method - Generations vs Iterations

- Power method: one neutron generation per iteration
- Wielandt method: multiple neutron generations per iteration, varies for each starting neutron

\[ k_e = k_0 + \Delta \]

Average chain length,
\[ L = 1 + \frac{k_0}{\Delta} \]
Dominance Ratio Calculation in MCNP5

• **Fission matrix DR**
  – Can be determined *before convergence*
  – Sensitive to mesh size
  – Provides *approximate DR*
  – Useful for characterizing problem convergence
  – May be useful for automated convergence tests

• **Coarse Mesh Projection Method with time series analysis for DR**
  – Can only be used *after convergence*
  – Independent of mesh size
  – Provides *accurate DR*

• **Both methods for DR were added to test version of MCNP5**

• **Negligible extra CPU time for either method**
Best Practices - Summary

• To avoid bias in $K_{\text{eff}}$ & tally distributions, use 10K or more neutrons/cycle

• Always check convergence of both $K_{\text{eff}}$ & $H_{\text{src}}$

• Take advantage of problem symmetry, if possible

• Use a good initial source guess, uniform in fissionable regions

• Run at least a few hundred active cycles to allow codes adequate information to compute statistics

• Be aware that statistics on tallies from codes are underestimated, possibly make multiple independent runs
References

- **Monte Carlo Methods**
- **Monte Carlo k-effective Calculations**
- **Superhistory Method**
- **Wielandt Method**
- **Shannon entropy & convergence**
References

- **Bias, correlation, & statistics**
  - E.M. Gelbard and R.E. Prael, "Monte carlo Work at Argonne National Laboratory", in Proc. NEACRP Meeting of a Monte Carlo Study Group, ANL-75-2, Argonne National Laboratory, Argonne, IL (1974).
Abstract

A Review of Monte Carlo Criticality Calculations - Convergence, Bias, Statistics

Forrest B. Brown (LANL)

Monte Carlo criticality calculations have been performed for over 50 years for reactor physics and criticality safety applications. With today’s faster computers, these calculations are being carried out to greater precision (smaller uncertainties) in $k_{eff}$, and detailed distributions of power and reaction rates are being computed routinely. This paper provides a review of the fundamental theory of Monte Carlo criticality calculations, with guidance on practical methods for: (1) assuring convergence of both $k_{eff}$ and the source distribution, (2) minimizing the bias in $k_{eff}$ and reaction rate distributions, and (3) dealing with the underprediction bias in uncertainties for $k_{eff}$ and reaction rate distributions.
Introduction
Perspective

- Bigger, faster computers ➔ more Monte Carlo calculations
  ➔ better localized statistics

- Principal uses of Monte Carlo have evolved:
  1960s: K-effective
  1970s: K-effective, detailed assembly power
  1980s: K-effective, detailed 2D whole-core
  1990s: K-effective, detailed 3D whole-core
  2000s: K-effective, detailed 3D whole-core, depletion, reactor design parameters

➔ Recent Monte Carlo R&D focused on advanced methods for modeling, depletion, & design parameters

➔ More important now than ever to address the fundamental theory & best practices for Monte Carlo criticality calculations
Challenges in Monte Carlo Criticality Calculations

Current Monte Carlo codes can model almost any geometry, with continuous-energy cross-sections & collision physics

But …..

Longstanding problems with the fundamental theory:

1. Bias in Keff & tallies
2. Convergence of Keff & source distribution
3. Underprediction bias in confidence intervals
4. Lack of adjoint weighting for tallies
5. Determining adequate population size
6. Propagation of error (xsecs, depletion, etc.)
7. …..

Problems (1) - (4) have been addressed in the last few years. MCNP5 features exist, or are coming this year.