COUNTERING NEUTRON CLUSTERING IN MONTE CARLO WITH A NEUTRON SOURCE INJECTION

P. Cosgrove[1], **M.A. Kowalski**[1], **E. Shwageraus**[1], **and G.T. Parks**[1]
[1]Department of Engineering, University of Cambridge
Trumpington Street, Cambridge, CB2 1PZ
pmc55@cam.ac.uk

ABSTRACT

Neutron clustering is a recently identified problem with Monte Carlo eigenvalue calculations which can produce significantly erroneous results. Previous work by Sutton & Mittal (2017) considered neutron clustering as a problem of maintaining 'genetic diversity' within the neutron population. This paper proposes reducing the extent of neutron clustering by replacing fission neutrons in the source bank with uncorrelated neutrons, sampled from a uniform source distribution – effectively adding new neutron genealogies to the population. The efficacy of the method is demonstrated on a number of simple problems, showing improved behaviour of the Shannon entropy and neutron centre-of-mass. Although currently limited in scope, this paper intends to provide a route to reducing clustering effects in more general problems.

KEYWORDS: Monte Carlo, clustering, correlation

1. INTRODUCTION

Monte Carlo eigenvalue solvers are increasingly popular within nuclear engineering and are regularly applied to the simulation of full-core reactor problems. However, as well as their computational expense, Monte Carlo methods suffer from a number of difficulties. These include determining and accelerating fission source convergence, under-estimation of errors, and, more recently, neutron clustering [1].

Neutron clustering is problematic for Monte Carlo in that it can cause significantly erroneous solutions to be obtained, while also being difficult to detect [2,3]. The most straightforward solution to neutron clustering is increasing the number of particles simulated per generation. However, for large problems, or even those which simply have extreme aspect ratios (e.g., a fuel pin), sufficiently large population sizes might be impractical. This could be due to i) the memory footprint of simulating many millions of particles per generation, and ii) the nature of the power iteration algorithm necessitating that a large number of generations be simulated to adequately converge the source – given that source convergence is nearly independent of the number of particles per generation, this can incur unnecessary computational expense. That said, the latter of these difficulties can be partially mitigated through source convergence techniques, such as fission matrix or response matrix acceleration, among others [4,5].

Sutton & Mittal took a novel approach to the analysis of clustering by explicitly considering the Monte Carlo generational algorithm, rather than a continuous-time representation [6]. Their work demonstrated the role that different neutron 'lineages' play in clustering: over generations, initially uncorrelated families of neutrons die out, such that, in the case of extreme clustering, all neutrons may eventually be descended from one neutron in the initial source guess. By itself, this problem is not significant provided that said family of neutrons has spread itself across the geometry of interest. If not, the problem will be under-sampled and the results will be biased.

The present work follows on from Sutton & Mittal by proposing a partial remedy to clustering, namely introducing neutrons into the simulation which are not descended from the initial source guess. Doing so at regular intervals should limit the extent to which neutrons becomes correlated. However, the particle injection shape must be chosen from a sufficiently representative distribution to prevent introducing significant bias to the simulation. This paper will demonstrate some preliminary applications of the injection algorithm to some relatively simple problems with further developments pursued in future work.

2. THEORY

Although dealing with an idealised Monte Carlo algorithm, Sutton & Mittal highlight that the extent of correlation between neutrons in an eigenvalue simulation increases over generations [6]. This arises from the need to ensure a constant population size while allowing for neutron multiplication – inevitably some neutrons will produce 2 or 3 offspring which feature in the subsequent generation while others will produce none due only to population normalisation. Sutton & Mittal considered a homogeneous, reflective problem. For more general heterogeneous problems with vacuum boundaries, this accumulation of correlation will be further accelerated through the introduction of another neutron loss mechanism, namely, leakage.

Clustering and correlation are not inter-changeable: in a small geometry which can be traversed by a particle family in a few generations, clustering is unlikely to occur appreciably even if all particles in the simulation descend from only one particle in the initial source guess. However, for large problems, requiring many particle generations to traverse, the accumulation of correlation drives clustering, as demonstrated by Sutton and Mittal [6].

For the idealised case they consider, an expression is presented describing how the fraction of uncorrelated pairs of particles varies with generations, g, with a fixed population of particles, N [6]:

$$\psi_{\text{unc}}(g) = \left(1 - \frac{1}{N}\right)^{g-1} \tag{1}$$

From this equation, the only way to decrease the accumulation of correlation over generations is to increase the population size. However, an alternative technique is available: removing particles from the population and replacing them with particles chosen from some suitable distribution – henceforth known as particle injection. In the frame of population genetics, this is analogous to introducing new lineages into the population.

Particle injection is a procedure to be performed between cycles, acting on the fission bank. After the fission bank is normalised to contain N particles (or fission sites, more exactly), some fraction, f, of said particles would be removed and discarded. This would be done by sampling from the fission bank uniformly. These particles would then be replaced by an equal number of particles chosen from some appropriate spatial distribution, D. Ideally, D should be the problem's eigenvector. Otherwise, some more approximate distribution could be used, but the particles injected and their descendants should be 'inactive', so to speak. That is, like in the inactive cycles of a Monte Carlo simulation, they do not contribute to statistical estimates until they have settled into the eigenvector. This is done such as to minimise the bias introduced by adding new particles. This inactive period would end after the injected particle's lineage, L_p, has existed for some chosen number of generations, L. Finally, it may be the case that particles are injected only infrequently, say, once every q generations. A modified Monte Carlo algorithm demonstrating particle injection is shown in Algorithm 1.

```
Input: $N, G, f, q, L, D$
Sample $N$ particles from a uniform distribution
Place sampled particles in bank $P$
for $g = 1, ..., G$ do
    for $p = 1, ..., N$ do
        if $L_p < L$ then
            Do not score statistics
            $L_p = L_p + 1$
        end
        Simulate $p$
        Store particles from fission in bank $P$
    end
    Normalise population
    if $\mod(g, q) = 0$ then
        Kill $\lfloor Nf \rfloor$ particles in $P$
        Sample $\lfloor Nf \rfloor$ particles from $D$
        For each sampled particle set $L_p = 0$
        Place sampled particles in $P$
    end
end
```

Algorithm 1: Modified Monte Carlo algorithm with particle injection

Modifying Eqn. (1) to account for particle injection, a heuristic understanding of its benefits can be obtained. If, every generation, a fraction, f, of particles is removed and replaced, this is equivalent to removing and replacing the fraction $\kappa = \frac{f(fN-1)}{N-1}$ of all particle pairs. The modified version of Eqn. (1) after the first generation and accounting for particle injection then becomes:

$$\psi_{\text{unc}}(g) = \left[(1-\kappa)\left(1-\frac{1}{N}\right)\right]^{g-1} + \kappa \sum_{j=0}^{g-1} \left[(1-\kappa)\left(1-\frac{1}{N}\right)\right]^{j} \tag{2}$$

Simplifying further by defining $r := (1-\kappa)\left(1-\frac{1}{N}\right)$, one notes that the resulting expression is a geometric series such that one may write:

$$\psi_{\text{unc}}(g) = r^{g-1} + \kappa\left(\frac{1-r^{g-1}}{1-r}\right) \tag{3}$$

When $f = 0$, Eqn. (1) is recovered. Furthermore, in the limit of many generation, the fraction of uncorrelated particles reaches a minimum value of $\frac{\kappa}{1-r}$, bounding the extent of correlation between particles. Similar, less convenient expressions can be derived if particle injection occurs more infrequently.

Previous works have demonstrated some effective methods for detecting neutron clustering in simple problems. Nowak et al. [7] use the Shannon entropy to detect clustering in uniform, homogeneous problems. In such cases, where clustering is not occurring to any significant extent, the Shannon entropy should reach its maximum value given that the neutron population should be distributed uniformly. Likewise, for symmetric problems the neutron centre-of-mass CoM) was used to quantify clustering: this value should be stationary at the geometry's centre. However, as the degree of clustering increases, the CoM is expected to oscillate about this point more violently [7].

3. NUMERICAL EXPERIMENTS AND RESULTS

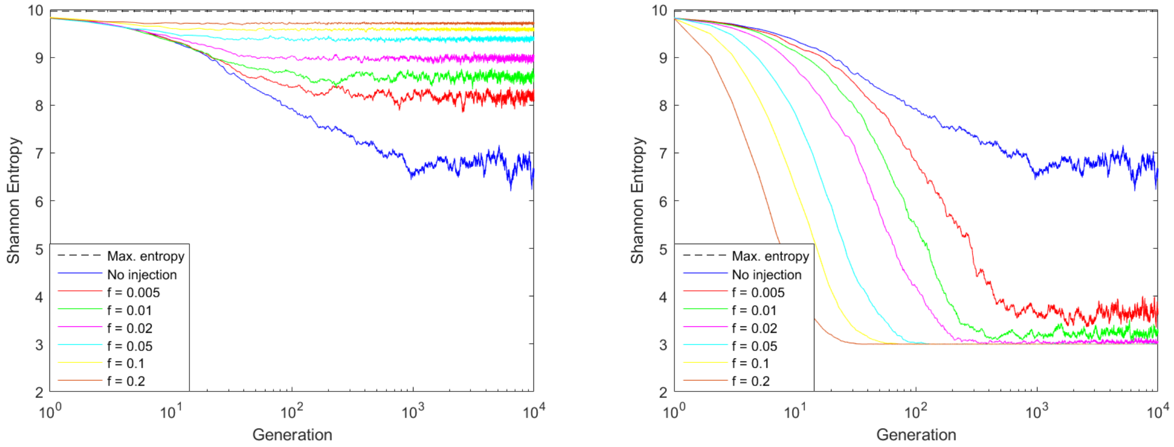
The different parameters in Algorithm 1 allow for a number of possible trade-offs in terms of correlation and bias. For example, the closer D is to the eigenvector, the less bias will be introduced and the fewer inactive generations will be required to negate this. Likewise, the more frequently particles are injected and the greater the fraction of particles that are replaced, the lesser will be the correlation between particles at the cost of increased bias for a fixed D and L. Given the number of parameters that can be investigated, this paper only serves as a preliminary study for some simple cases. In particular, the authors hope to demonstrate the ability of this modified algorithm to produce accurate results while minimising the effects of clustering.

The investigations will proceed in three parts. The first will demonstrate the algorithm for a toy problem, namely, Sutton and Mittal's homogeneous, reflective box, demonstrating the improvement in the Shannon Entropy and the effect of the choice of D. The implementation is then extended to a full-fidelity problem, simulating a 3D PWR pin with uniform composition and coolant density along its length with reflective boundaries. Both of these demonstrations are straightforward in that the problem eigenvector is also a uniform distribution (at least along one direction for the PWR pin), inducing negligible bias. A more difficult problem will then be considered: the same PWR pin but with vacuum boundaries such that the injection distribution is no longer the eigenvector such that the effect of introducing particles for some inactive period must be considered.

The box problem is simulated using a simple MATLAB model. The PWR pins are simulated using the University of Cambridge-developed Monte Carlo code SCONE. This code was designed to be highly-modifiable and object-oriented, allowing for easy modification of the solver of interest without requiring changes in the supporting subroutines. Furthermore, the code is able to apply different treatments to different segments of a particle population with ease, e.g., simulating particles with both a multi-group and continuous energy treatment simultaneously, or, in this case, handling active and inactive particles simultaneously. SCONE will be more fully described in another paper presented during this conference.

3.1. Homogeneous, reflective box using a pseudo-eigenvalue simulation

This problem is precisely that described in [6]. It is a homogeneous cube of length 4 m with reflective boundaries. A pseudo-eigenvalue problem is simulated in that a number, N, of neutrons



(a) Uniform injection **(b) Point source injection**

Figure 1: Shannon entropy of the box problem when replacing various fractions of the population each generation according to two different spatial distribution

are tracked and each is absorbed. A new generation is generated by selecting the absorption sites uniformly, with replacement, until N particles have been chosen for the next generation. The eigenvalue is irrelevant to the functioning of the algorithm, and there is no distinction between active and inactive cycles. The particles simulated are pseudo-neutrons in a single energy group. The neutronic properties of the medium are provided in the original paper [6].

Here an $10 \times 10 \times 10$ mesh is placed across the problem to obtain the Shannon entropy. Given the eigenvector of the problem is a uniform distribution, the Shannon entropy should approach the maximum value of $\log_2(1000) \approx 9.966$ when clustering is minimised. To investigate the effect of varying injection size, 10,000 pseudo-particles are simulated for 10,000 generations while the percentage of injected particles is varied from 0 to 20% per generation. To demonstrate the necessity of choosing a good approximation of the eigenvector to inform the injection location, this experiment is repeated with both a uniform distribution, and a point source (delta) distribution at the centre of the box. Given the relatively small number of particles as compared to the number of Shannon entropy bins, the expected maximum Shannon entropy will be somewhat less than the maximum, as discussed by Nowak et al. [7]. The results are shown in Fig. 1.

When using the uniform distribution, the anticipated results are attained: injecting particles reduces clustering, causing the Shannon entropy to approach its maximum value. Replacing increasingly large fractions of the population per generation only improves these results. Even so, replacing only a relatively small fraction of particles appears to improve the ultimate Shannon entropy substantially. Of course, this is an idealised problem with an exactly known eigenvector – Fig. 1(b) demonstrates clearly that a poorly chosen eigenvector can prove detrimental. Therefore, fresh neutron lineages alone are not inherently advantageous without accounting for the source shape.

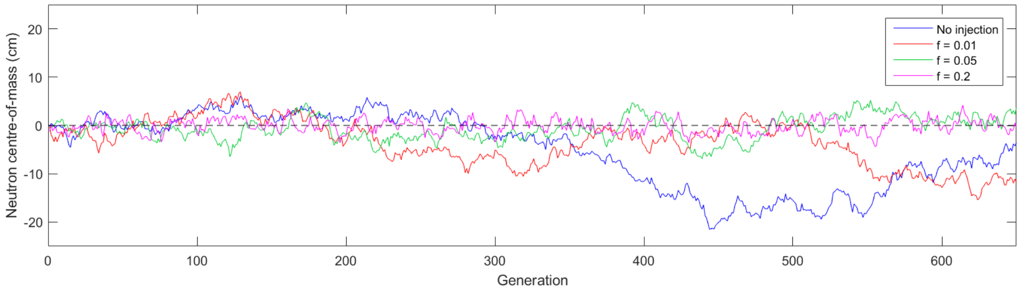


Figure 2: Neutron centre-of-mass for the reflected PWR pin when varying the particle injection fraction

Table 1: Results for the reflective PWR pin

Simulation	k_{eff}	σ_k**(pcm)**	**CoM (cm)**	σ_{CoM} **(cm)**
Reference	1.21005	5	-0.11	0.25
f = 0	1.20989	17	-5.23	7.61
f = 0.01	1.20974	16	-3.55	4.71
f = 0.05	1.20981	17	-0.63	2.47
f = 0.2	1.20932	17	-0.11	1.43

3.2. PWR pin with an axially reflective boundary

For a more realistic problem, a 3D PWR pin was simulated. The pin had a pitch of 1.26 cm and a height of 3.66 m. The fuel had a radius of 0.4095 cm and neither cladding nor fuel-cladding gap were modelled. In this case, all boundaries were reflective. The pin was made of 4.5% enriched UO_2 and a water coolant with a uniform density of 0.75 g/cm^3. In this problem, the axial component of the source eigenvector is known – a uniform distribution, as before. Each simulation used 20,000 particles per generation, with 500 active generations and 150 inactive. A reference simulation with minimal clustering was performed for comparison, using 2,000,000 particles per generation over 60 active and 100 inactive generations. To estimate the reduction in clustering, the neutron CoM along the axial direction was calculated during each generation, as was done by Nowak et al. [7]. The results are shown in Fig. 2 and Table 1. The amplitude of the CoM oscillation decreases, as expected, with an increased particle injection fraction as clustering is successfully reduced without substantial bias when injecting fewer particles. The f = 0.2 case, however, does appear slightly biased, presumably due to neglecting the radial component of the source eigenvector.

3.3. PWR pin with an axially vacuum boundary

The same problem as before is simulated, but with the axial boundary condition set to vacuum. Hence, the eigenvector for this problem is significantly more complicated than the previous, challenging the injection algorithm. In fact, investigations varying the injection fraction and number

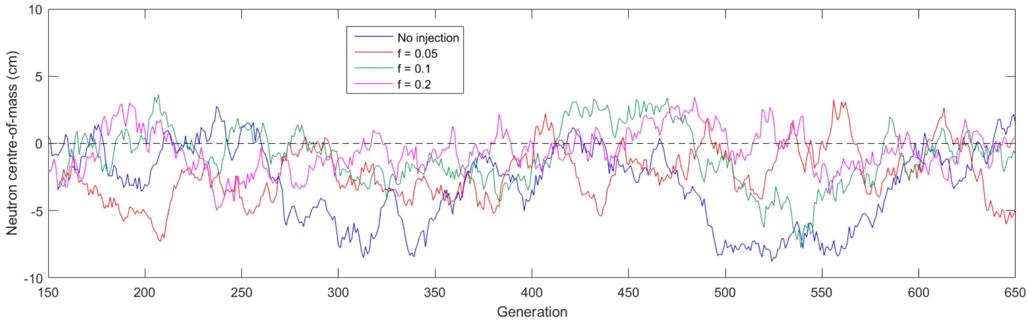


Figure 3: Neutron centre-of-mass for the vacuum-boundary PWR pin when varying the particle injection fraction

of inactive cycles have produced significantly biased results: if few inactive cycles for injected particles are used, the estimate of the eigenpair differs significantly from the true eigenpair. On the other hand, using many inactive cycles for injected particles reduces the active population, degrading statistical estimates. Therefore, a better approximation of the eigenvector was necessary.

To estimate the eigenvector, this paper used the dominant eigenvector calculated from a 1D fission matrix, discretised into 15 uniform-width bins and calculated for the last 100 inactive cycles, before being fixed during the active [4]. The discretisation of the resulting eigenvector will induce some bias. Furthermore, the fission matrix will also be subject to biases due to statistics and clustering. This latter problem may be partially alleviated in future by computing an accurate estimate of the eigenvector at the beginning of the simulation by some other semi-deterministic means, e.g., the response matrix method [5]. It was found that, even using the fission matrix, a substantial bias was still introduced to the simulation - this would presumably be quenched with a finer fission matrix discretisation. Doing so, however, would necessitate simulating more particle histories to avoid deteriorating the statistics per mesh bin. Hence, neutrons were only injected once every 5 generations and left inactive for the same duration. The results are show in Fig. 3 and Table 2. The graph in Fig. 3 begins at generation 150 as this is where particles are first injected during the simulations.

The CoM results are less stark than for the reflective pin. Nevertheless, particle injection does lead to a reduction in the CoM oscillation over the course of the simulation, with a small but noticeable bias in the eigenvalue only beginning to emerge when f = 0.2.

4. CONCLUSIONS

It has been shown, at least for simple problems, that particle injection can mitigate some of the effects of clustering, such as CoM oscillation. On the other hand, where the eigenvector is non-trivial, care must be taken to avoid biasing the simulation. This paper is a preliminary examination of the injection algorithm which will be followed by more thorough parameter studies in future, using more particle generations, different means of estimating the source eigenvector, and problems more strongly afflicted by clustering effects.

Table 2: Results for the vacuum-boundary PWR pin

Simulation	k_{eff}	σ_k(pcm)	CoM (cm)	σ_{CoM} (cm)
Reference	1.20572	9	-0.19	0.23
f = 0	1.20581	18	-2.15	3.27
f = 0.05	1.20602	18	-2.28	1.97
f = 0.1	1.20579	19	-0.97	2.14
f = 0.2	1.20631	19	-0.54	1.57

ACKNOWLEDGEMENTS

This work was carried out with the financial support of the UK Engineering and Physical Sciences Research Council (EPSRC) through the Imperial College, University of Cambridge and Open University Centre for Doctoral Training in Nuclear Energy under grant EP/L015900/1.

REFERENCES

[1] E. Dumonteil, F. Malvagi, A. Zoia, A. Mazzolo, D. Artusio, C. Dieudonné, and C. De Mulatier. "Particle clustering in Monte Carlo criticality simulations." *Annals of Nuclear Energy*, **volume 63**, pp. 612–618 (2014).

[2] E. Dumonteil, G. Bruna, F. Malvagi, A. Onillon, and Y. Richet. "Clustering and traveling waves in the Monte Carlo criticality simulation of decoupled and confined media." *Nuclear Engineering and Technology*, **volume 49**(6), pp. 1157–1164 (2017).

[3] P. Cosgrove, E. Shwageraus, and G. Parks. "Neutron clustering as a driver of Monte Carlo burn-up instability." *Annals of Nuclear Energy*, p. 106991 (2019).

[4] S. Carney, F. Brown, B. Kiedrowski, and W. Martin. "Theory and applications of the fission matrix method for continuous-energy Monte Carlo." *Annals of Nuclear Energy*, **volume 73**, pp. 423–431 (2014).

[5] J. Leppänen. "Acceleration of fission source convergence in the Serpent 2 Monte Carlo code using a response matrix based solution for the initial source distribution." *Annals of Nuclear Energy*, **volume 128**, pp. 63–68 (2019).

[6] T. M. Sutton and A. Mittal. "Neutron clustering in Monte Carlo iterated-source calculations." *Nuclear Engineering and Technology*, **volume 49**(6), pp. 1211–1218 (2017).

[7] M. Nowak, J. Miao, E. Dumonteil, B. Forget, A. Onillon, K. S. Smith, and A. Zoia. "Monte Carlo power iteration: Entropy and spatial correlations." *Annals of Nuclear Energy*, **volume 94**, pp. 856–868 (2016).