LA-UR-15-29020
Approved for public release; distribution is unlimited.

Title: GPU Acceleration of Mean Free Path Based Kernel Density Estimators for Monte Carlo Neutronics Simulations

Author(s): Burke, TImothy P.
Kiedrowski, Brian C.
Martin, William R.
Brown, Forrest B.

Intended for: report on summer internship at LANL, Monte Carlo R&D work
Report

Issued: 2015-11-19

GPU ACCELERATION OF MEAN FREE PATH BASED KERNEL DENSITY ESTIMATORS FOR MONTE CARLO NEUTRONICS SIMULATIONS

Timothy P. Burke, Brian C. Kiedrowski, William R. Martin, Forrest B. Brown

1 INTRODUCTION

Kernel Density Estimators (KDEs) are a non-parametric density estimation technique that has recently been applied to Monte Carlo radiation transport simulations [1,2]. Kernel density estimators are an alternative to histogram tallies for obtaining global solutions in Monte Carlo tallies. With KDEs, a single event, either a collision or particle track, can contribute to the score at multiple tally points with the uncertainty at those points being independent of the desired resolution of the solution. Thus, KDEs show potential for obtaining estimates of a global solution with reduced variance when compared to a histogram. Previously, KDEs have been applied to neutronics for one-group reactor physics problems [1] and fixed source shielding applications [2]. However, little work was done to obtain reaction rates using KDEs.

Previously, the Mean Free Path (MFP) based KDE was introduced that is capable of obtaining accurate estimates of reaction rates for reactor physics problems in 1-D slab geometries in continuous energy and 2-D one-group problems with linear material interfaces. However, the MFP KDE was not extended to 2-D geometries with non-planar surfaces [3]. This paper introduces a new form of the MFP KDE that is capable of handling general geometries. Furthermore, extending the MFP KDE to 2-D problems in continuous energy introduces inaccuracies to the solution. An ad-hoc solution to these inaccuracies is introduced that produces errors smaller than 4% at material interfaces.

Additionally, While KDEs produce smoother results compared to histograms, it comes at a cost of increased computation time. For every particle event, a kernel function must be evaluated for every tally point within the support range of the event. Furthermore, tallying to points in materials outside of where the particle event occurred requires the look up of additional cross section information. Both of these facts can make the KDE tally routine the most expensive portion of the Monte Carlo simulation. Since the KDE requires the calculation of multiple quantities for every particle event, it is well suited for computation on a Graphics Processing Unit (GPU). In an attempt to reduce run times, the KDE tally is exported to the GPU during the transport process. The KDE is applied to tallies in two 2-D pincell problems as well as two quarter-assembly problems. Speedups are problem dependent, and range between 1.6 and 13.8 for the problems studied in this paper.

2 BACKGROUND & THEORY

2.1 Distance-Based KDE

Previously, the multivariate distance-based collision KDE for scalar flux was developed [1]

$$\hat{\phi}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}}{\Sigma_t(\mathbf{X}_{i,c},E)}\prod_{l=1}^{d}\frac{1}{h_l}k\left(\frac{x_l - X_{l,n}}{h_l}\right), \tag{1}$$

where N is the number of histories, c_i is the number of collisions in history n, $x_l - X_{l,n}$ is the distance between the tally point at location $\mathbf{x}$ and the location of sample n in dimension l, Σ_r is the cross section of the reaction rate of interest, Σ_t is the total cross section, k is the univariate kernel function, and h_l is the bandwidth in dimension l. This estimator was extended to compute reaction rates [4], resulting in the collision KDE for reaction rates:

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}\Sigma_r(\mathbf{x},E)}{\Sigma_t(\mathbf{X}_{i,c},E)}\prod_{l=1}^{d}\frac{1}{h_l}k\left(\frac{x_l - X_{l,n}}{h_l}\right), \tag{2}$$

where $\Sigma_r(\mathbf{x},E)$ is macroscopic cross section of reaction rate r at the tally point at energy E. The kernel function k is generally a symmetric Probability Density Function (PDF) and has the properties

$$\int k(u)\mathrm{d}u = 1, \quad \int uk(u)\mathrm{d}u = 0, \quad \text{and} \quad \int u^2k(u)\mathrm{d}u = k_2 \neq 0. \tag{3}$$

The performance of the KDE is heavily dependent upon the bandwidth. An optimal bandwidth for general KDEs is discussed in depth by Silverman and is defined as the bandwidth that minimizes the Mean Integrated Square Error (MISE), the sum of the integrated square bias and the integrated variance [5]. The KDE obtains a biased estimate of the underlying PDF $f(x)$ by estimating

$$\hat{f}(x) = \int_{-\infty}^{\infty} k\left(\frac{x-y}{h}\right)f(y)dy. \tag{4}$$

The bias introduced by the kernel approximation in Eq. (4) is

$$\begin{aligned}\mathrm{bias}[\hat{f}(x)] &= f(x) - \hat{f}(x) &(5)\\ &= f(x) - f(x)\int_{-\infty}^{\infty}k(u)du - hf'(x)\int_{-\infty}^{\infty}uk(u)du + h^2f''(x)\int_{-\infty}^{\infty}u^2k(u)du + \ldots &(6)\end{aligned}$$

Using the kernel properties in Eq. 3, the bias reduces to

$$\mathrm{bias}[\hat{f}(x)] = h^2f''(x)k_2 + O(h^3). \tag{7}$$

Furthermore, the variance can be approximated as [5]

$$\mathrm{var}\left[f(x)\right] = \frac{1}{Nh}f(x)\int k(u)^2du. \tag{8}$$

Finding a bandwidth h that minimizes the MISE results in an optimal bandwidth

$$h_{MISE} = k_2^{-2/5} \left(\int K(u)^2 du \right)^{1/5} \left(\int f''(x)^2 dx \right)^{-1/5} n^{-1/5}. \tag{9}$$

Additionally, if more information about the underlying distribution is known or can be calculated beyond integral quantities then the Mean Square Error (MSE), the sum of the square bias and variance at a point in the PDF, can be minimized to obtain a MSE-optimal bandwidth

$$h_{MSE} = k_2^{-2/5} \left(\int K(u)^2 du \right)^{1/5} f(x)^{1/5} f''(x)^{-2/5} n^{-1/5}. \tag{10}$$

Since the optimal bandwidth is dependent upon the distribution being used, either an estimate of $f''(x)$ must be used or an assumption must be made about the distribution of $f(x)$ in order to approximate $f''(x)$. The most common assumption is to approximate $f(x)$ as the normal distribution and use the moments of the estimated distribution as parameters in the optimal bandwidth [5]. Applying the assumption that $f(x)$ is normal, MISE-optimal bandwidths can be computed using

$$h_{MISE,l} = \left(\frac{4}{(2+d)N} \right)^{1/(4+d)} \sigma_l, \tag{11}$$

where d is the number of dimensions used in the KDE tally, N is the number of samples, and σ_l is the standard deviation of the distribution of data in dimension l. For flux distributions, σ_l can be estimated using a collision estimator via

$$\sigma_l = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \frac{w_{i,l}}{\Sigma_t(X_{i,l})} \left(X_{i,l}\right)^2 - \left(\frac{1}{N} \sum_{i=1}^{N} X_{i,l} \frac{w_{i,l}}{\Sigma_t(X_{i,l})} \right)^2}. \tag{12}$$

2.2 MFP KDE

The distance-based KDE has difficulty estimating reaction rates at material interfaces when the materials exhibit different total macroscopic cross sections. This is especially the case in reactor physics problems when estimating the absorption reaction rate near control rods or when attempting to capture the rim effect in fuel pins. Because of this, the MFP KDE was developed in 1-D and for a 2-D tally in 1-D geometries [3]. The collision MFP KDE in 1-D is

$$\phi(x) = \sum_{i=1}^{N} \sum_{c=1}^{C_i} \frac{w_{i,c}}{h_x} k \left(\frac{\int_x^{X_{i,c}} \Sigma_t(x') \mathrm{d}x'}{h_x} \right). \tag{13}$$

The optimal bandwidth for the MFP KDE is calculated using Eq. (11), but with σ_l calculated by

$$\sigma_{\mathrm{MFP},l} = \sigma_l \overline{\Sigma}_t \tag{14}$$

with

$$\overline{\Sigma}_t = \frac{\int_0^\infty \int_R \Sigma_t(\mathbf{x}, E)\phi(\mathbf{x}, E)\mathrm{d}V\mathrm{d}E}{\int_0^\infty \int_\Omega \phi(\mathbf{x}, E)\mathrm{d}V\mathrm{d}E}, \tag{15}$$

where $\int_R \mathrm{d}V$ describes an 3-D integral over the KDE region R. The concept of a KDE region will be described in more detail in a later section.

The 2-D MFP KDE was derived for slab geometry in 1-D using a multivariate KDE that is a product of univariate kernels and is defined as

$$\phi(x) = \sum_{i=1}^{N}\sum_{c=1}^{C_i} \frac{w_{i,c}}{h_x} k\left(\frac{\int_{\mathbf{x}}^{\mathbf{X}_{i,c}} \Sigma_t(\mathbf{x})\mathrm{d}S\frac{\Delta X}{\Delta R}}{h_x}\right) \frac{\int_{\mathbf{x}}^{\mathbf{X}_{i,c}} \Sigma_t(\mathbf{x})\mathrm{d}S}{h_y \Delta R} k\left(\frac{\int_{\mathbf{x}}^{\mathbf{X}_{i,c}} \Sigma_t(\mathbf{x})\mathrm{d}S\frac{\Delta Y}{\Delta R}}{h_y}\right), \tag{16}$$

where ΔR is the distance between the tally point and the collision site and ΔX and ΔY are the distances between the tally point and the collision site in the x and y directions, respectively. This form of the MFP KDE does not account for geometries with non-planar surfaces exactly. Thus, another form of the MFP KDE using a multivariate kernel that is radially symmetric was derived to account for these geometries. The distance-based multivariate Epanechnikov kernel [5] is defined as

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2\sqrt{5}^d} C_d^{-1}(d+2)(1-\frac{1}{5}\mathbf{x}^{\mathrm{T}}\mathbf{x}), & \mathbf{x}^{\mathrm{T}}\mathbf{x} < 5 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where $\mathbf{x} = (x_1, \ldots, x_d)^{\mathrm{T}}$ and C_d is the volume of the d-dimensional sphere: $C_1 = 2$, $C_2 = \pi$, $C_3 = 4\pi/3$, etc. The multivariate Epanechnikov kernel cited in literature does not normally contain extra factors of $1/5$, but they are included here so the multivariate Epanechnikov kernel is equivalent to the univariate version in 1-D. The multivariate KDE [6] applied to Monte Carlo collision tallies results in the multivariate collision KDE

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i} \frac{w_{i,c}\Sigma_r(\mathbf{x}, E)}{\Sigma_t(\mathbf{X}_{i,c}, E)} \frac{1}{|\mathbf{H}|} K\left(\mathbf{H}^{-1}\left[\mathbf{x} - \mathbf{X}_{i,c}\right]\right), \tag{18}$$

where $|\mathbf{H}|$ is the determinant of the $d \times d$ symmetric positive definite bandwidth matrix. For uncorrelated data, $\mathbf{H}$ is a diagonal matrix comprised of the elements $h_1, \ldots, h_d$. The multivariate kernel must still satisfy the basic kernel properties:

$$\int K(\mathbf{u})\mathrm{d}\mathbf{u} = 1 \quad \text{and} \quad \int \mathbf{u}K(\mathbf{u})\mathrm{d}\mathbf{u} = 0. \tag{19}$$

In order to have the kernel argument be a function of the number of MFPs between the collision site and tally point the normalization coefficient $C(\mathbf{x})$ needs to be determined such that the kernel satisfies the properties in Eq. (19),

$$\int C(\mathbf{x})K\left(\mathbf{H}^{-1}\left[\mathbf{\Omega}\int_{\mathbf{X_n}}^{\mathbf{x}} \Sigma_t(\mathbf{x}')dS\right]\right)\mathrm{d}\mathbf{x} = 1, \tag{20}$$

where $\mathbf{X_n}$ represents the location of the tally point, $\mathbf{x}$ represents the collision location, and $\mathbf{\Omega}$ is a unit vector depicting the angle between the collision site and tally point projected onto the Cartesian

axes. The quantity in brackets, $\mathbf{\Omega}\int_{\mathbf{X_n}}^{\mathbf{x}}\Sigma_t(\mathbf{x}')dS$, is a vector describing the number of MFPs between the tally point and the collision site in x, y, and z. Finding a coefficient $C(\mathbf{x})$ such that Eq. (20) is satisfied begins with examining the first kernel property in units of space:

$$\int \frac{1}{|\mathbf{H}|}K(\mathbf{H}^{-1}\left[\mathbf{x}-\mathbf{X_n}\right])\mathrm{d}\mathbf{x}=1, \tag{21}$$

Changing this integral to spherical coordinates, shifting the system so it is centered about r_n, and applying a change of variables $v=r$ yields

$$\int_{-1}^{1}\int_{0}^{2\pi}\int_{0}^{\infty}\frac{1}{|\mathbf{H}|}K(\mathbf{H}^{-1}\left[v\mathbf{\Omega}\right])v^2\mathrm{d}v\mathrm{d}\theta\mathrm{d}\mu=1. \tag{22}$$

Another substitution is performed to change the kernel argument to be the number of MFPs between the collision site and tally location: let

$$v=\int_0^r\Sigma_t(r'\mathbf{\Omega})\mathrm{d}r \quad \text{and} \quad \mathrm{d}v=\Sigma_t(r\mathbf{\Omega})\mathrm{d}r. \tag{23}$$

Inserting this substitution into Eq. (22) produces

$$\int_{-1}^{1}\int_{0}^{2\pi}\int_{0}^{\infty}\frac{\Sigma_t(r\mathbf{\Omega})}{|\mathbf{H}|}\left(\int_0^r\Sigma_t(r'\mathbf{\Omega})\mathrm{d}r\right)^2K\left(\mathbf{H}^{-1}\left[\mathbf{\Omega}\int_0^r\Sigma_t(r'\mathbf{\Omega})\mathrm{d}r\right]\right)\mathrm{d}r\mathrm{d}\theta\mathrm{d}\mu=1. \tag{24}$$

Equation (24) provides insight into producing normalization coefficients for the multivariate MFP KDE. The normalization coefficient $C(\mathbf{x})$ is found by equating the integrands of Eqs. (24) and (20) in spherical coordinates, thus yielding

$$C(r,\mathbf{\Omega})=\frac{\Sigma_t(r\mathbf{\Omega})}{|\mathbf{H}|}\frac{\left(\int_0^r\Sigma_t(r'\mathbf{\Omega})\mathrm{d}r\right)^2}{r^2}. \tag{25}$$

The quantity $\left(\int_0^r\Sigma_t(r'\mathbf{\Omega})\mathrm{d}r\right)/r$ is the average cross section between the tally point and collision site, and does not cause instabilities when it is evaluated near $r=0$. Switching back to Cartesian coordinates, we get the d-dimensional normalization coefficient

$$C(\mathbf{x})=\frac{\Sigma_t(\mathbf{x})}{|\mathbf{H}|}\frac{\left(\int_{\mathbf{x}}^{\mathbf{X}_{i,c}}\Sigma_t(\mathbf{x}')\mathrm{d}S\right)^{d-1}}{||\mathbf{X}_{i,c}-\mathbf{x}||^{d-1}}. \tag{26}$$

Using this normalization coefficient, the multivariate MFP KDE is defined as

$$\hat{f}(\mathbf{x})=\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}\Sigma_r(\mathbf{x},E)}{|\mathbf{H}|}\frac{\left(\int_{\mathbf{x}}^{\mathbf{X}_{i,c}}\Sigma_t(\mathbf{x}')\mathrm{d}S\right)^{d-1}}{||\mathbf{X}_{i,c}-\mathbf{x}||^{d-1}}K\left(\mathbf{H}^{-1}\left[\mathbf{\Omega}\int_{\mathbf{x}}^{\mathbf{X}_{i,c}}\Sigma_t(\mathbf{x}')\mathrm{d}S\right]\right) \tag{27}$$

The normalization coefficient defined for the multivariate MFP KDE in Eq. (27) is identical to that obtained using the MFP KDE defined using a product of univariate kernels in Eq. (16). The only difference between the two equations is the shape of the kernel function. In 2-D, the kernel function

in Eq. (27) has a support region defined by an ellipse while the support region of the kernel function in Eq. (16) is defined by a rectangle. As such, the two estimators should produce similar results, as is demonstrated in the results section.

While the MFP KDE accurately accounts for material heterogeneities and severe cross section differences, it adds another factor of complexity when computing a particle event's contribution to the score at a tally point by requiring knowledge of the number of MFPs between the particle event and a tally point. Not only does this preclude the use of the track-length MFP KDE, but it increases the computational burden of the KDE. To reduce this computational burden, an approximation to the MFP KDE was developed that does not require the calculation of the exact number of mean free paths between the particle event and tally point. This approximate MFP KDE (aMFP KDE) estimates the number of MFPs between a particle event and a tally point by only using the cross section at the tally point to compute the number of MFPs between the tally point and particle event. The aMFP KDE for a reaction rate r is defined as

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}\Sigma_r(\mathbf{x},E)}{\Sigma_t(\mathbf{X}_{i,c},E)}\prod_{l=1}^{d}\frac{\Sigma_t(\mathbf{x},E)}{h_l}k\left(\frac{\Sigma_t(\mathbf{x},E)\left(X_{l,i,c}-x_l\right)}{h_l}\right). \tag{28}$$

This aMFP KDE can be thought of as the distance-based KDE in Eq. (2) with the bandwidth being modified by the inverse of the total macroscopic cross section of the material that the tally point resides in.

While the aMFP KDE and MFP KDE perform well in 1-D problems, spikes occur in 2-D problems. These spikes occur when a neutron undergoes a collision at resonance energies. With the bandwidth being modified by the inverse of the total macroscopic cross section, a collision at resonance energies will reduce the support region of the kernel to a localized area. Since the kernel function must integrate to 1, a large score is contributed to a small area. This results in spikes occurring in 2-D problems, where the result at one tally point may be twice that of the neighboring tally point less than a millimeter away. These spikes will be demonstrated in the 2-D portion of the results section.

Currently there exists no appealing way to handle these spikes. One method is to specify a minimum bandwidth such that if the ratio of h/Σ_t is below some user-specified $h_{\min}$ then the MFP kernel reverts to a distance-based kernel with $h = h_{\min}$. This works well for areas away from material interfaces, but it causes a bias at material interfaces whose magnitude is dependent on the value of $h_{\min}$ [7]. Another method is to use another form of the aMFP KDE, termed the fractional aMFP KDE, defined as

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{c_i}\frac{w_{i,c}\Sigma_r(\mathbf{x},E)}{\Sigma_t(\mathbf{X}_{i,c},E)}\prod_{l=1}^{d}\frac{\Sigma_t(\mathbf{x},E)^{1/d}}{h_l}k\left(\frac{\Sigma_t(\mathbf{x},E)^{1/d}\left(X_{l,i,c}-x_l\right)}{h_l}\right). \tag{29}$$

The motivation for the fractional aMFP KDE comes from the MFP KDE working well in 1-D tallies where the normalization coefficient is one factor of the macroscopic total cross section. Modifying the method for adjusting the spatial bandwidths in multiple dimensions such that the normalization coefficient is just one factor of the macroscopic total cross section effectively handles the spikes

seen in 2-D tallies. However, this fractional aMFP KDE also produces a bias at material interfaces. For problems demonstrated in this paper, this bias is less than 1% for fission reaction rates and less than 8% for absorption reaction rates. While fractional powers of the cross section do not often appear in the transport equation, it is important to note that this alternate form of adjusting the bandwidth does not affect the physics of the simulation; it is just an alternative form of defining an adaptive bandwidth.

3 GPU ACCELERATION

3.1 Motivation for GPUs

Graphics processing units are commonly found on high performance computing (HPC) machines as a means of increasing performance without adding additional compute nodes. While it is difficult for existing Monte Carlo neutron transport codes to take advantage of GPUs without re-writing large portions of the code, it is possible to leverage GPUs through heterogeneous computing. Rather than use GPUs for the bulk of the transport routine, it is possible to export the compute-intensive portions of the Monte Carlo algorithm onto GPUs. For KDEs, it is not unlikely for the bulk of the run time to be spent on the tally process. Significantly more floating-point operations are required for KDEs for each collision or particle track than a histogram tally since a single event can contribute to the scores of multiple tally points. As such, exporting the KDE onto the currently unused GPUs can have significant improvements in run times.

3.2 GPU Algorithm

The GPU KDE algorithm is implemented using CUDA C and uses C Bindings to link the CUDA C code to the main Fortran program in OpenMC. The algorithm is designed to hide the cost of copying memory from the host (CPU) to the device (GPU) and have the device compute KDE scores while the host continues transporting particles to collect tally information. Rather than tally scores directly, the host collects information in one of two sets of sample arrays during the transport process. The host calculates all the necessary cross section information and stores it as well as all necessary collision information in the first set of sample arrays. Once the host has stored a pre-set number of samples (50,000 collisions for this paper), the host copies the data asynchronously to the device. The host then puts the KDE GPU kernel into the same CUDA stream as the memory copy so the kernel will launch as soon as the memory has finished copying from the host to the device. The host then immediately returns and begins transporting particles and populating a second set of sample arrays.

Once the GPU has received the first array, it re-arranges data for better memory coalescence and the KDE GPU kernel is launched. The kernel uses 64 threads per block, with one block per tally point. Each block loops over the collisions in the sample array and computes each collision's contribution to the score at that tally point with each thread in a block handling a different collision. Once the host has finished filling the second array, it asynchronously sends the data to the device and waits until the first set of sample arrays has been received by the device before re-filling the first set of arrays. This process repeats until the end of the batch, when the partial set of sample arrays is sent

to the device for computation of scores. Once this has finished, the device sends back the tally data to the host and the host uses the normal MPI processes for combining tally data across multiple processors.

To increase the speed of this algorithm, a nearest neighbor list (NNL) was created on the GPU similar to that on the CPU. Rather than loop over all collisions, the finite support of the KDE kernel necessitates only looping over collisions that fall within one KDE kernel support region of the tally point. The NNL uses a mesh to divide the simulation domain into bins equal to the maximum kernel support length in each dimension. Each collision is assigned a key based on on its position in the neighborhood mesh. The index of each particle (the particle's initial location in the collision array) is sorted based on its key value using the CUDA UnBound (CUB) library's radix sort routine. Collisions and their cross section information are then rearranged based on their key and index values so that collisions occurring within the same neighborhood bin are located next to one another in memory.

To improve performance of the NNL, a maximum support region was defined for the aMFP KDE. The bandwidth for the aMFP KDE in Eq. (28) is effectively the bandwidth of the distance-based KDE multiplied by the average cross section in that region divided by the cross section at the tally point. Thus, if the cross section at a tally point is smaller than the average cross section in that region then the bandwidth becomes larger. This has a large impact on the performance of the aMFP KDE since the neighborhood list must now be searched beyond 1 bin in each direction. However, it is possible to manipulate the bandwidth such that no more than 1 kernel support length is searched in each direction. This is done by limiting the value of the cross section in the argument of the kernel function and the normalization coefficient, denoted as $\Sigma_{t,k}(x)$ in the following equation, such that

$$\Sigma_{t,k}(x) = \begin{cases} \Sigma_t(x) & \Sigma_t(x) > \overline{\Sigma}_t \\ \overline{\Sigma}_t & \text{otherwise} \end{cases} \tag{30}$$

Using Eq. (30) ensures that if the cross section is below the average cross section for a region, the aMFP kernel reverts to the distance-based kernel. This approximation does not adversely affect the accuracy of the simulation, since it only limits the size of the bandwidth. Using a smaller bandwidth rather than a larger bandwidth will reduce the bias in the simulation while increasing the variance in the results. Thus, using Eq. (30) will increase the figure of merit for the aMFP KDE since the decrease in run times is accompanied by only a slight increase in the variance of the solution. Speedups obtained using the maximum support region are problem dependent, but a speedup of 2.1 and a figure of merit increase by a factor of 1.5 and 2.1 for the flux and fission distributions, respectively, are obtained for the 2-D boxcell problem with 60×60 tally points. With the use of the NNL with the maximum support region, each block of threads loop over the collisions in the neighborhood bins within one support region of the tally point rather than over all collisions in the sample array. The speedup obtained by using the NNL is again problem dependent, but for 120×120 tally points in the 2-D boxcell problem a speedup of 29 is obtained.

3.3 GPU Optimization

Several improvements were made throughout the design of the GPU KDE compute kernel to reduce compute times. Kernel compute times for the various optimization iterations for the 2-D boxcell problem with 120×120 tally points are shown in Table I. The focal points for optimization were driven by the GPU's Single Instruction Multiple Thread (SIMT) architecture. The threads on a GPU are broken up into groups of 32 threads called a warp with all threads in a warp executing the same instruction. When a thread requires data from global memory, the warp reads either 32, 64, or 128 bytes of memory at once, depending on what the threads in the warp require. For example, if each thread in a warp needs to operate on a single precision floating-point (4 byte) number, then the warp needs to load in 128 bytes of memory. If the data the warp requires is sequentially located in memory, then this data load is coalesced into one read from global memory. However, if the data is randomly located in memory, then it could take up to 32 separate reads from global memory. Since each read from global memory costs 400 to 800 cycles on GPUs with compute capability 2.x, minimizing the number of loads from global memory is crucial to achieving high performance. While it is impossible to eliminate all reads from global memory, the GPU is capable of hiding some of this latency through warp scheduling. When one warp requires memory from global memory the streaming multiprocessor can switch to a different warp and execute arithmetic instructions on those threads while the initial warp is loading data from global memory. This effectively hides the latency, but only if there are a sufficient number of arithmetic operations on other warps. If there are too many global memory reads for the amount of arithmetic required in each warp then the GPU kernel is limited by the time it takes to read data from global memory. Thus, reducing the number of global memory loads can have a significant impact on the performance of the algorithm.

One design decision that is impacted by the coalesced data reading pattern of GPUs is using an array of structures (AoS) versus a Structure of Arrays (SoA). When using an AoS, the data of the structure is placed sequentially in memory. If a thread only requires one piece of data from that structure, then efficiency is lost based on the size of the structure. If the structure contains enough data, then it is possible for each thread in the warp to require a separate read from global memory when each thread is accessing a variable from sequentially located structures. The usual particle data required to for tallies in a Monte Carlo code includes position, previous position, angle, energy, cell, material, and weight. For example, if a KDE tally only requires the x position of a collision for a given computation, then an extra 10 floating-point values and 2 integers exist in memory between one collision's x position and the x position of the collision accessed by the next thread. This extra data between the x positions of particles causes extra reads from global memory, hampering the code's performance. This is eliminated by using a SoA. Rather than have an array of particle structures, a single structure, called a particle list, is created that contains separate arrays for a collision's data. For example, the collision's x, y, and z positions are all located in separate arrays. Therefore, a calculation of the distance between a collision site and a tally point in the x direction will only require two reads from global memory if the x positions are stored as double precision floating-point values. Converting the collision data and cross section information into a SoA results in a speedup of 40%.

Additionally, the shared memory on the GPU was leveraged to further reduce GPU kernel compute

times. From the NVIDIA Programming Guide [8], "shared memory is expected to be a low-latency memory near each processor core (much like an L1 cache)." Commonly used variables are copied to shared memory and results for a tally point are collected by each thread in shared memory before being reduced and added to the results array stored in global memory. Using shared memory results in a 20% reduction of kernel compute times.

Another optimization is to change tally variables from double precision to single precision floating-point values. Changing to single precision has two effects. First, it reduces the size of memory required by each thread. Using the previous example where all threads in a block require the x position of different collisions, using single precision data for the collision position allows all of the x positions for the 32 threads in a block to be loaded in one read from global memory rather than the two reads necessary with double precision data. Furthermore, the GPU has a faster clock speed for single precision floating-point operations. For the NVIDIA Tesla M2090, the clock speed for single precision floating-point operations is double that of double precision operations. This discrepancy is even greater in newer graphics cards, with the NVIDIA Tesla K80 having 8.74 Tflops for single precision floating point values versus 2.91 Tflops for double precision floating point values at peak performance [9]. For the KDE GPU algorithm, switching from double precision to single precision floating-point values further reduces the kernel compute time by approximately 30%. This reduction in precision does not come at a cost in accuracy for the problems studied here as the single precision results agree with those obtained using double precision to six significant digits.

Table I. KDE GPU kernel compute times for 2-D boxcell problem with 120×120 tally points

Design Iteration	Kernel Compute Time (ms)
Array of Structures	56.3
Structure of Arrays	35.6
Shared Memory Improvements	27.8
Single Precision	19.6
Single Precision - Fractional MFP	20.2
Shared Memory Improvements without NNL	815

Future design implementations could improve the speed of both the CPU and GPU tally process. The current GPU algorithm scales linearly with the number of tally points in the problem. This will limit the algorithm's application to larger, more complex problems. The algorithm could be improved so that it also scales with tally point density rather than the number of tally points, similar to the algorithm on the CPU. Another area for potential speedup is to reduce the number of cross sections computed per collision. Since collisions in one material region can contribute to the score in another material region, the cross sections of all materials are calculated after every collision. This could be improved upon by using another neighbor list such that only cross sections of materials within one support region of the collision are calculated. This would reduce the overhead for both the CPU and GPU versions of the KDE tally. Additionally, it may be possible to export the track-length KDE (TL KDE) to the GPU as well. While the TL KDE tally requires more computation per particle track, it would be more difficult to sort the particle tracks for memory coalescence. As such, the TL KDE is less well-suited for computation on the GPU.

Another area for improvement is to use GPUs with CUDA compute capability 3.5 or higher. These newer GPUs have a Multi Process Service (MPS) which allows commands sent to the device from several MPI processes to be handled simultaneously. In earlier GPUs, including the M2090, each MPI processes would have a separate CUDA context. Kernels from one CUDA context cannot run on the GPU at the same time as kernels from another context. Thus, with the current algorithm design, there is no way to have compute kernels overlap with one another. This inherently reduces efficiency since a kernel is not using all of the GPU's resources towards the end of the GPU kernel calculation. The MPS fixes this problem by having all MPI processes under one context, handled by the MPS. Thus, multiple kernels could be run at the same time, improving the efficiency of the algorithm.

4 RESULTS

4.1 Analytic Solutions in 1-D Slab Geometry

The exponential distribution is commonly encountered in Monte Carlo neutron transport. To test the capability of KDEs to capture distributions in neutron transport problems a simple 1-D, homogeneous purely absorbing slab with a mono-directional mono-energetic beam of neutrons incident on the left face of the slab was modeled. Since the underlying distribution is known, both the MISE-optimal and MSE-optimal bandwidths from Eqs. (9-10) can be computed exactly from the reference solution. Even though the flux distribution is being estimated, optimal bandwidths are calculated using the collision reaction rate distribution. This is because bandwidths that are optimal for estimating the scalar flux distribution are not equivalent to those used to estimate the collision density. Since collision estimators contribute score to the scalar flux with samples obtained from the collision density, it is reasonable to apply bandwidths that are optimized for the collision density and apply weights to the resulting density to obtain estimates of the scalar flux. For a collision density of $f(x) = \Sigma_t(x) e^{-\int_0^x \Sigma_t(x')dx'}$, the MISE-optimal and MSE-optimal bandwidths are

$$\begin{aligned} h &= \left(\int_0^\infty \Sigma_t(x)^6 e^{-2\int_0^x \Sigma_t(x')dx'} dx \right)^{-1/5} N^{-1/5} k_2^{-2/5}, \text{ and} \\ h(x) &= \Sigma_t(x)^{-1} e^{\frac{1}{5}\int_0^x \Sigma_t(x')dx'} n^{-1/5} k_2^{-2/5}, \end{aligned} \tag{31}$$

respectively.

The flux distribution was estimated from 10,000 samples using a collision histogram tally, MISE KDE tally, and MSE KDE tally. The total macroscopic cross section was set at 0.5 cm^{-1}, with 100 histogram bins and tally points placed on a structured grid from 0-10 cm. The boundary kernel method [10] is used at tally points with support regions that overlap the boundaries at x = 0 cm and x = 10 cm. The tally results are shown in Fig. 1.

As seen in Fig. 1, the KDEs are better at capturing the underlying distribution, with the MISE KDE and MSE KDE having similar performance. The L2 norms for the histogram, MISE KDE, and MSE KDE are 0.1095, 0.0367, and 0.0368 respectively. Thus, KDEs are able to accurately capture a basic distribution essential to neutron transport problems.

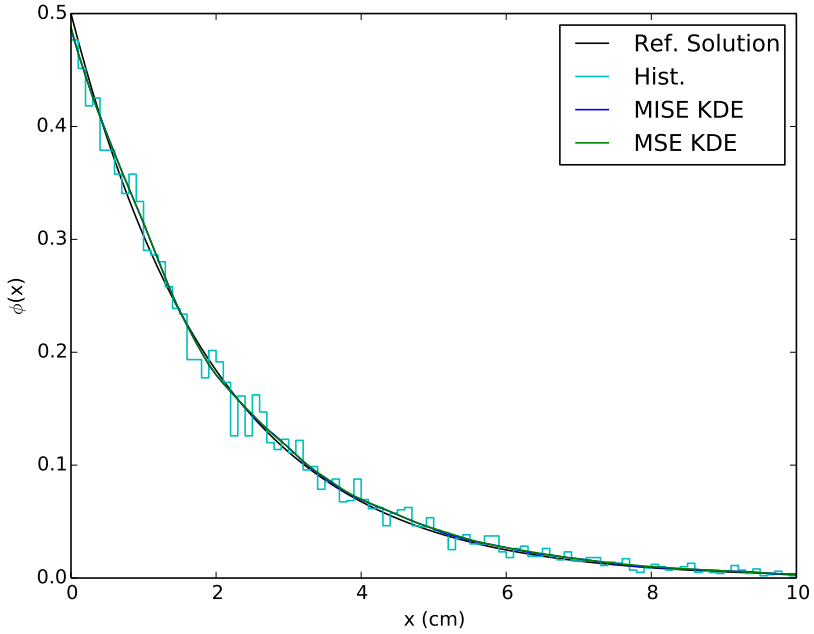


Figure 1. Comparison between histograms, KDEs, and reference solution for exponential distribution.

However, the smoothing capabilities of KDEs can be detrimental when the distribution being estimated has a discontinuous first derivative, another common feature in nuclear engineering problems. Extending the previous 1-D purely absorbing slab problem to include two materials with different macroscopic cross sections showcases this issue. The 2-material slab problem consists of a slab of material with $\Sigma_t = 1$ cm^{-1} ranging from 0 cm to 1 cm and a slab of material with $\Sigma_t = 5$ cm^{-1} ranging from 1 cm to 10 cm. The reference solution and results from the MISE KDE, MSE KDE, and histogram are shown in Fig. 2. The results show that the MSE KDE and MISE KDE accurately estimate the score away from the material interface, however they under-predict the score near the material interface. The MISE KDE spreads the score too far into the optically-thick slab, thus under-predicting the score at the interface and over-predicting the score in the optically-thick slab. The MSE KDE under-predicts the score in the optically-thin slab more so than the MISE KDE, however it is better able to capture the flux at the interface as well as the steeper gradient in the optically-thick slab.

To better account for the material heterogeneity, a change of variables can be done so that the distribution is smooth in the new phase space. Once a smooth distribution is obtained, it can be transformed back to the original phase space to obtain the kink in the distribution at the material interface. Let

$$u = \int_0^x \Sigma_t(x')\mathrm{d}x' \quad \text{and} \quad \mathrm{d}u = \Sigma_t(x)\mathrm{d}x, \tag{32}$$

Thus the underlying distribution becomes $f(u)du = e^{-u}du$. The MISE-optimal and MSE-optimal

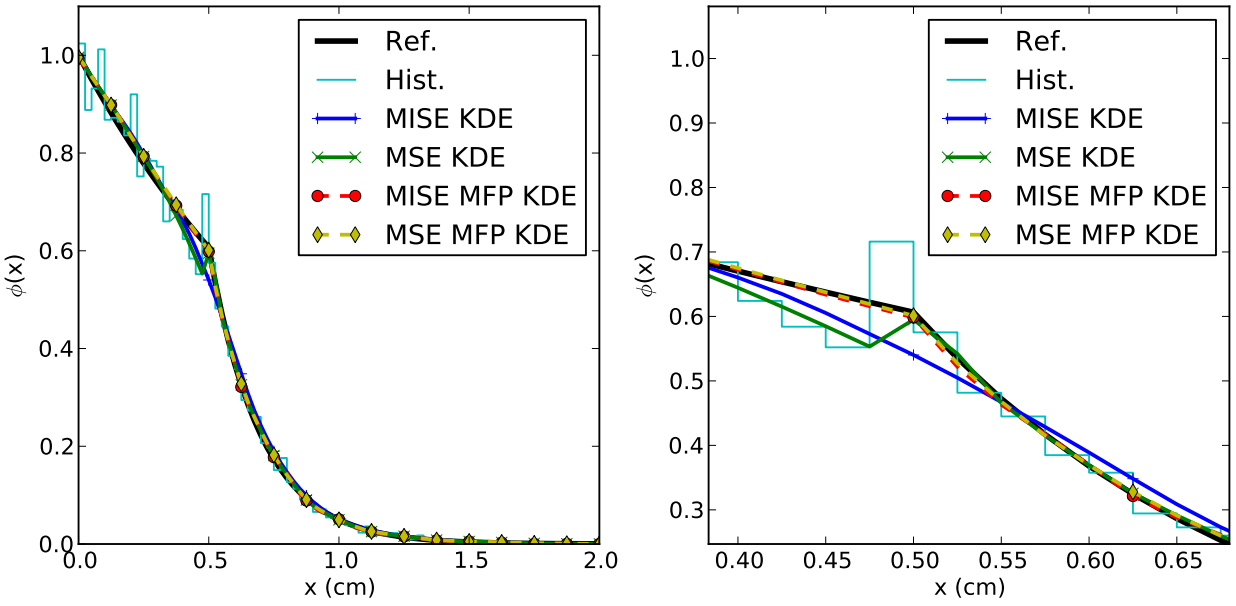
bandwidths are now

$$\begin{aligned} h &= \frac{1}{2} N^{-1/5} k_2^{-2/5} \\ h(u) &= e^{u/5} k_2^{-2/5} N^{-1/5}, \end{aligned} \tag{33}$$

respectively. The collision MFP KDEs for flux are defined as

$$\hat{\phi}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h} k \left(\frac{\int_x^{x_i} \Sigma_t(x') dx'}{h} \right). \tag{34}$$

Figure 2 shows the reference solution and the estimates from the histogram, MISE KDE, MSE KDE, MISE MFP KDE, and MSE MFP KDE. Figure 2 shows that the MISE MFP KDE and the

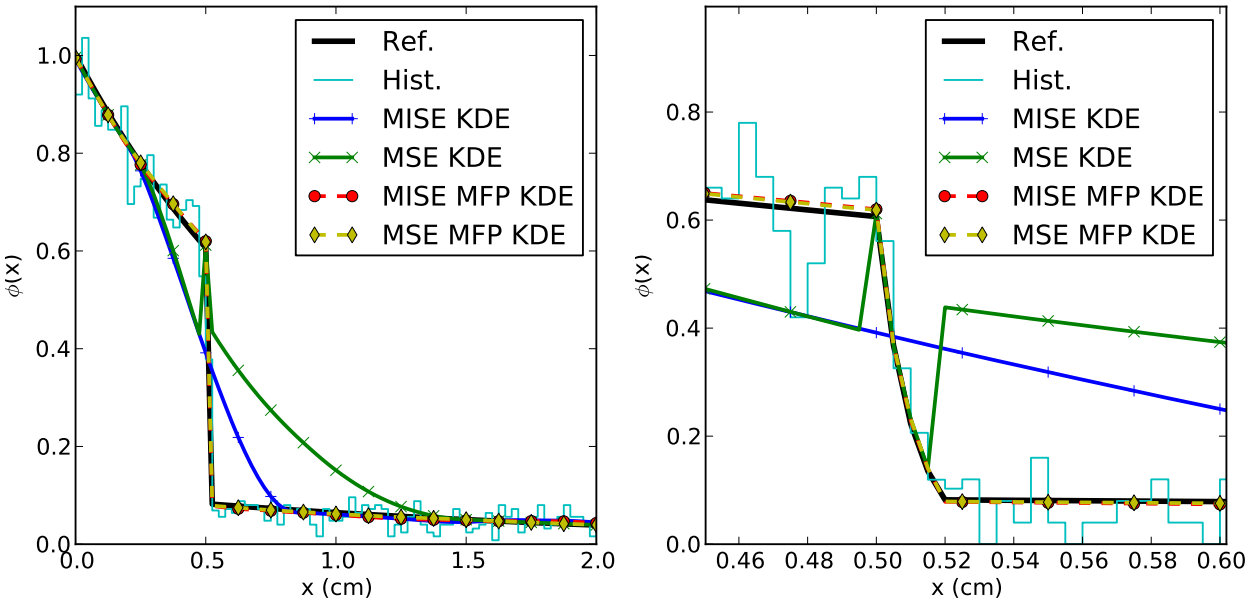


(a) Whole problem. **(b) Close-up of material interface.**

Figure 2. Comparison between histogram, MSE KDE, MISE KDE, MSE MFP KDE, MISE MFP KDE, and reference solution for the heterogeneous 1-D slab test problem.

MSE MFP KDE both accurately capture the flux at the material interface. The MFP KDEs are capable of capturing the discontinuous first derivative in the density due to their change of variable in the kernel argument. Furthermore, the MFP KDEs are as accurate as the distance-based MISE and MFP KDEs away from the material interface.

To further test these KDEs, a 1-D heterogeneous problem with a thin slab of strongly absorbing material is modeled. The 3-material slab problem consists of a slab of material with $\Sigma_t = 1$ cm^{-1} ranging from 0 cm to 0.5 cm, a slab of material with $\Sigma_t = 100$ cm^{-1} ranging from 0.5 cm to 0.52 cm, and a slab of material with $\Sigma_t = 0.5$ cm^{-1} from 0.52 cm to 10 cm. This problem was run with 10,000 particles with 80 histogram bins and KDE tally points placed from 0 cm to 2 cm. The results



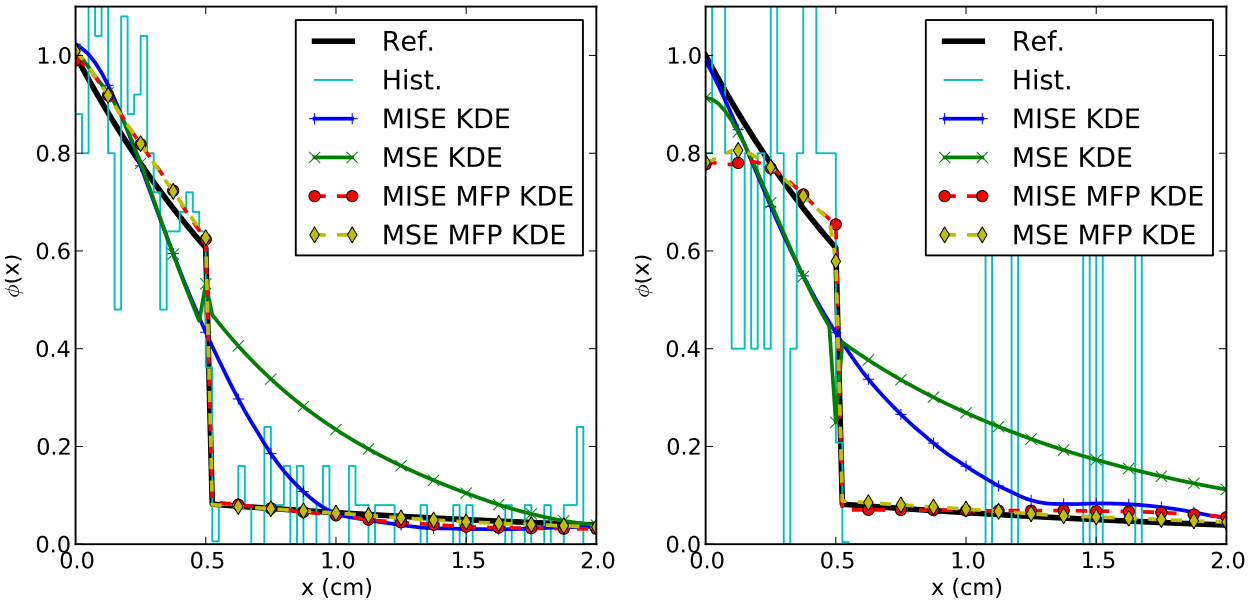
(a) Whole problem. **(b) Close-up of thin, strong absorber.**

Figure 3. Comparison of KDE methods and histogram to analytical solution for a 1-D, 3-material problem with a thin, strong absorber.

for this problem as well as a close-up of the results at the material interface with the number of bins and tally points increased to 400 are shown in Fig. 3.

Figure 3 shows that the histogram estimate has significant variance throughout the majority of the problem and the distance-based KDEs show substantial bias at the material interface with under-prediction as large as 30% prior to the strong absorber and over-predictions over 400% after the strong absorber. The distance-based KDEs spread the flux past the strong absorber, causing an under-prediction in the flux prior to the strong absorber and an over-prediction after the strong absorber. Conversely, the MFP KDEs show excellent agreement with the reference solution throughout the problem with reduced variance compared to the histogram. Looking closer at the material interface in Fig. 3b shows that the MFP KDEs accurately capture steep flux gradients through thin, strong absorbers. Additionally, the MSE KDE accurately captures the steep gradient in the strong absorber even though it produces poor estimates in neighboring materials near the interface and produces a discontinuous distribution.

To showcase the potential utility of KDEs, the 3-material slab problem was run with 1000 particles and 100 particles, the results of which are depicted in Fig. 4. Figure 4 shows that the MFP KDE is able to capture the flux profile with significantly less variance than the histogram, even when significantly fewer particles are run. Thus, KDEs show potential in obtaining global solutions in problems that are under-sampled regardless of the desired resolution of the result.



(a) 1000 Particles **(b) 100 Particles**

Figure 4. 1-D 3-material problem with 1000 and 100 particles.

4.2 1-D Slab With Thin, Strong Absorber

The transportation of scores across a strong absorber can also be seen in continuous energy k-eigenvalue problems for the aMFP KDE. If the problem has a strong, thin absorber then the collisions can "transport" scores across the strong absorber when the aMFP KDE is used, artificially increasing the result on the other side of the strong absorber while decreasing the score locally as was seen in the simple 1-group problems in Figs. 2-4. The problem is depicted in Fig. 5 and is composed of two slabs of fuel with water between them with a 0.1 cm thick strong absorber separated from the fuel with 10 cm of water in between. The fuel is UO_2 and the absorber is comprised of B_4C with a fictitious density of 120 g/cm^3. The density of the absorber was artificially increased in order to create a problem that presents the previously described issues for the aMFP KDE. The problem was run in a local version of OpenMC [11] with 100,000 particles per batch, 1000 batches with 60 inactive batches. Results were collected at 10,000 histogram bins placed uniformly from 13 cm to 14 cm with KDE tally points placed at the center of the bins. Results for the track-length MFP KDE and aMFP KDE and the histogram reference solution are shown in Fig. 6. The results for the collision KDE agree with the track-length KDE and are omitted for clarity. Close-ups of the left edge of the strong absorber and the right edge of the strong absorber are shown in Fig. 7.

Figures 6 and 7 show that the same issue occurs in the continuous energy problem, however it is more localized to the region near the material interface. The inaccuracies caused by using the aMFP KDE are more localized to the region around the strong absorber compared to the 1-D 1-group results in Fig. 3 due to there being more samples used in this problem, causing a reduction in the size of the bandwidth. While the flux is underpredicted before the strong absorber by as much as 25%

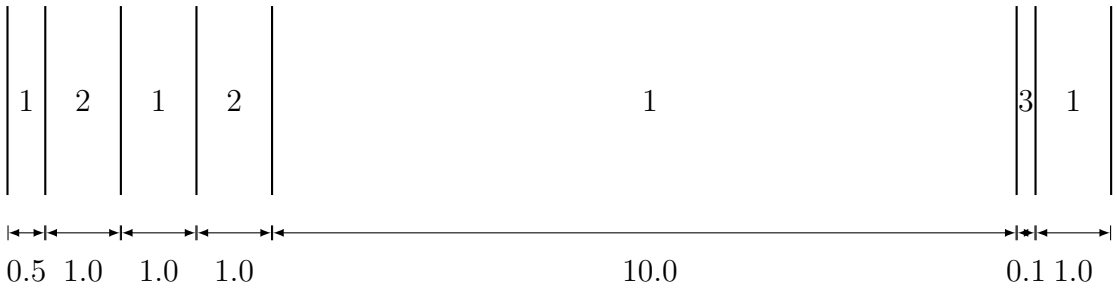


Figure 5. Graphic depicting problem with a thin, strong absorber. Regions labeled 1, 2, and 3 contain water, fuel, and strong absorber, respectively. The slab widths are shown underneath the geometry.

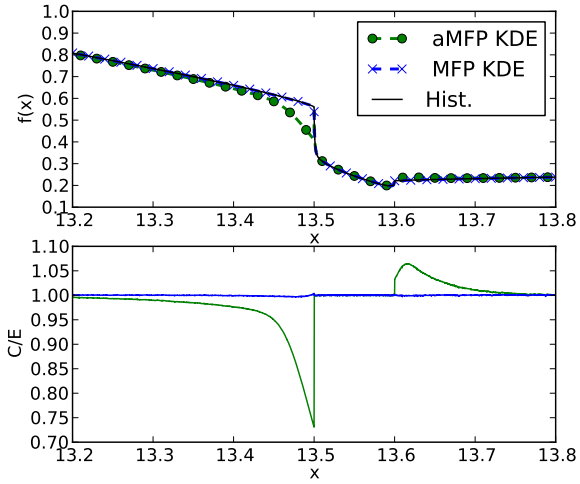


Figure 6. Flux comparison between aMFP KDE, MFP KDE, and reference histogram near the strong absorber of the thin, strong absorber problem.

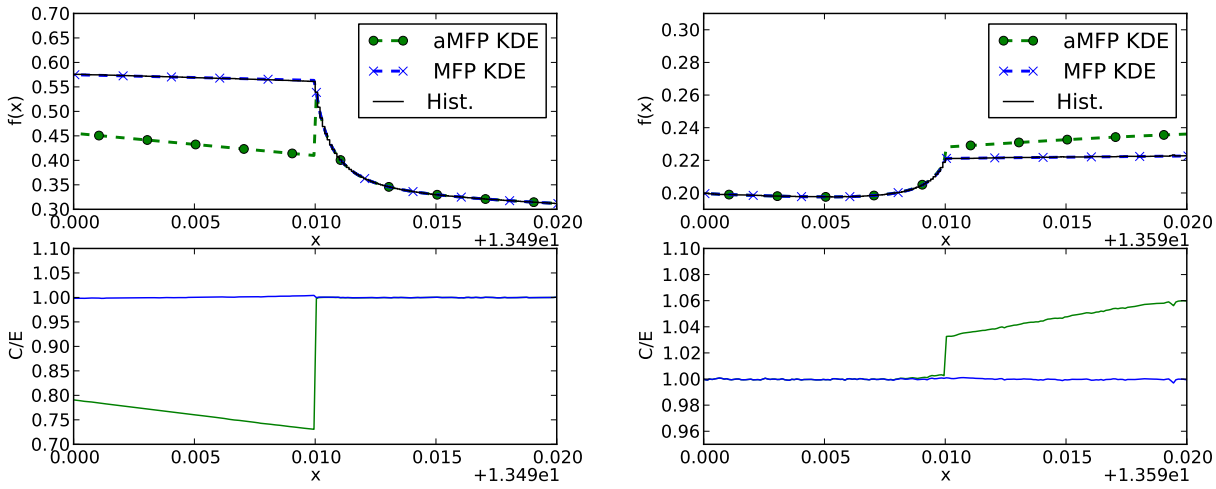
and overpredicted after the strong absorber by approximately 6%, the aMFP KDE correctly predicts the flux inside the strong absorber. The MFP KDE, on the other hand, is capable of capturing the steep gradients without any loss in accuracy.

It is possible to adjust the aMFP KDE such that it is better at capturing the distribution in regions near strong absorbers. A reduction in the bandwidth is necessary to prevent the score from spreading across the strong absorber. This can be done by using the maximum cross section in the problem to calculate the number of MFPs between the collision site and tally point in the water surrounding the strong absorber. This results in a kernel defined as

$$k(u)du = \frac{\max(\Sigma_t(x))}{h} k\left(\frac{\max(\Sigma_t(x))(X-x)}{h}\right) \mathrm{d}x. \tag{35}$$

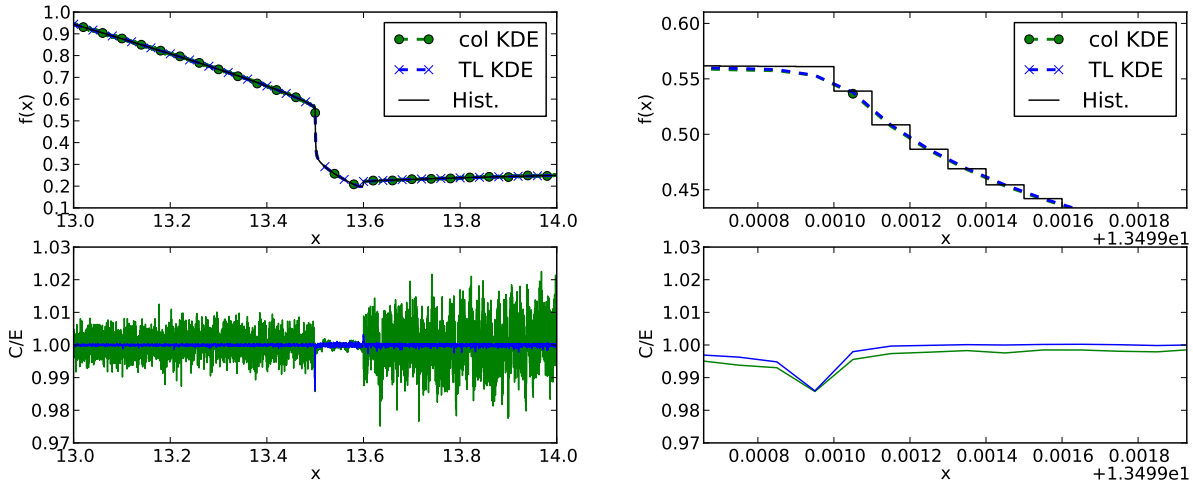
Figure 8 shows the results from the problem with the thin, strong absorber, but the kernel in the KDE is defined using Eq. (35).

Figure 8 shows that using the maximum cross section in the problem to compute the number of



(a) Left side of strong absorber **(b) Right side of strong absorber**

Figure 7. Comparison between TL MFP KDE, TL aMFP KDE, and histogram reference solution for the thin, strong absorber problem near the material interface.



(a) Whole problem. **(b) Close-up of left side of strong absorber.**

Figure 8. Flux comparison between aMFP collision and track-length KDE using the kernel described in Eq. (35) and the reference histogram for the thin, strong absorber problem.

MFPs between the particle track or collision site and the tally point produces results with reduced bias near the strong absorber. The maximum bias is approximately 1.5% at the left edge of the strong absorber, pictured in Fig. 8b. However, this reduction in bias comes at a cost of increased variance. Since the kernel described in Eq. (35) is used at every tally point in the problem, the relative uncertainty in the KDE solution in the water increases by a factor of 6 to 9. While it is possible to restrict the use of the alternate kernel to regions near the strong absorber, those regions will still have increased variance. It is important to note that while the aMFP KDE fails to capture the flux at material interfaces in these 1-D heterogeneous problems with strong, thin absorbers,

Table II. Cross sections for one-group problems.

	Σ_t	Σ_a	Σ_s	Σ_f	ν
Moderator	$3.264\,00 \times 10^{-1}$	$9.792\,00 \times 10^{-2}$	$2.284\,80 \times 10^{-1}$	0	0
Fuel	$6.528\,00 \times 10^{-1}$	$1.566\,72 \times 10^{-1}$	$4.961\,28 \times 10^{-1}$	$1.305\,60 \times 10^{-1}$	2.7
Absorber	$9.792\,00 \times 10^{-1}$	$9.652\,80 \times 10^{-1}$	$1.392\,00 \times 10^{-2}$	0	0

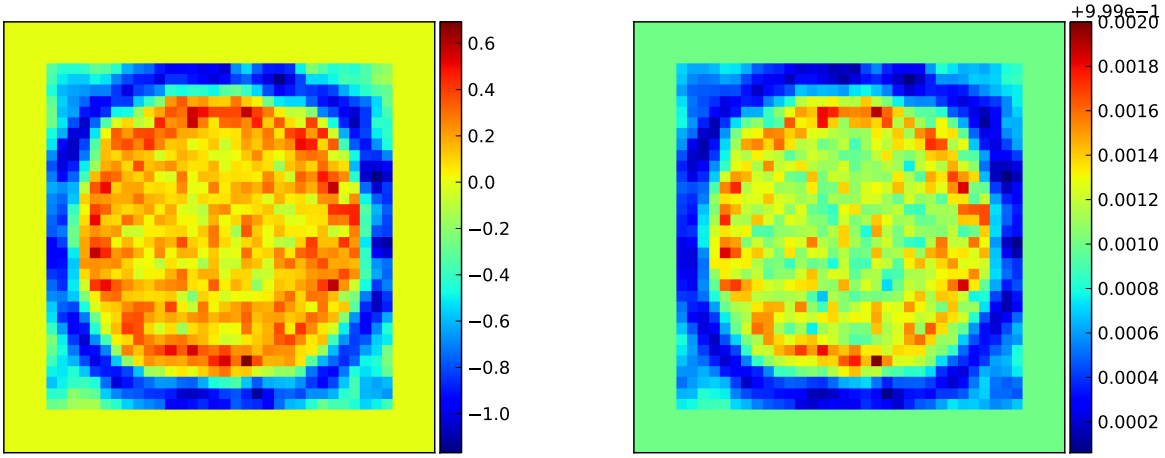
most problems in neutron transport do not contain materials that produce flux gradients as severe as this. The problem with a strong, thin absorber is shown as a potential limitation of the aMFP KDE, and the user should be aware that it exists.

4.3 2-D One-Group

The multivariate MFP KDE is tested in a 2-D 1-group pincell problem and is compared against a reference histogram and the 2-D MFP KDE derived for 1-D slab geometry. The pincell is comprised of a cylinder of UO_2 with radius 0.603 cm surrounded by water with a lattice pitch of 1.875 cm and reflecting boundary conditions. The boundary kernel method is used at tally points whose kernels overlap with the external boundaries, and is used for all other problems detailed in this paper. One-group cross sections used in this problem are shown in Table II. Separate KDE regions are defined for the fuel and water. Flux results were collected on a structured grid with 40 histogram bins in each dimension with KDE tally points placed at the center of the histogram bins. The simulations were run with 660 batches, 60 inactive batches with 60,000 particles per batch. The fuel and water are defined as separate KDE regions. The multivariate MFP KDE reverts to using a product of univariate kernels in order to use the boundary kernel method for tally points within one support region of the problem boundary. This does not affect the accuracy of the multivariate MFP KDE results as there are no material interfaces in that region. The results obtained using the 2-D multivariate MFP KDE in Eq. (27) are compared to those obtained using the MFP KDE formulated as a product of univariate kernels via Eq. 16 in Fig. 9. The difference between the results in units of mean free paths is pictured on the left while the C/E values, where C represents the multivariate MFP KDE values, are shown on the right.

The first thing to note in Fig. 9 is that the two estimators agree exactly at tally points within one kernel support region of the external boundary since they both use the boundary kernel method with the multivariate kernel expressed as a product of univariate kernels in each dimension for tally points in that region. Additionally, Fig. 9 shows that there is little difference between the two versions of the 2-D MFP KDE, with the maximum difference being less than 0.1%. This is expected, since the normalization coefficient for the kernel used in Eq. (16) is the same as that of the kernel function used in Eq. (27). Even so, there is a pattern of error between the two results, with the two estimators disagreeing by 1σ around the material interface. The kernel functions have slightly different support regions, with the multivariate MFP KDE in Eq. (27) being radially symmetric while the MFP KDE in Eq. (16) is symmetric over the coordinate axes. As such, the MFP KDE in Eq. (16) does not integrate to 1 in geometries with non-planar surfaces, causing this pattern of error.

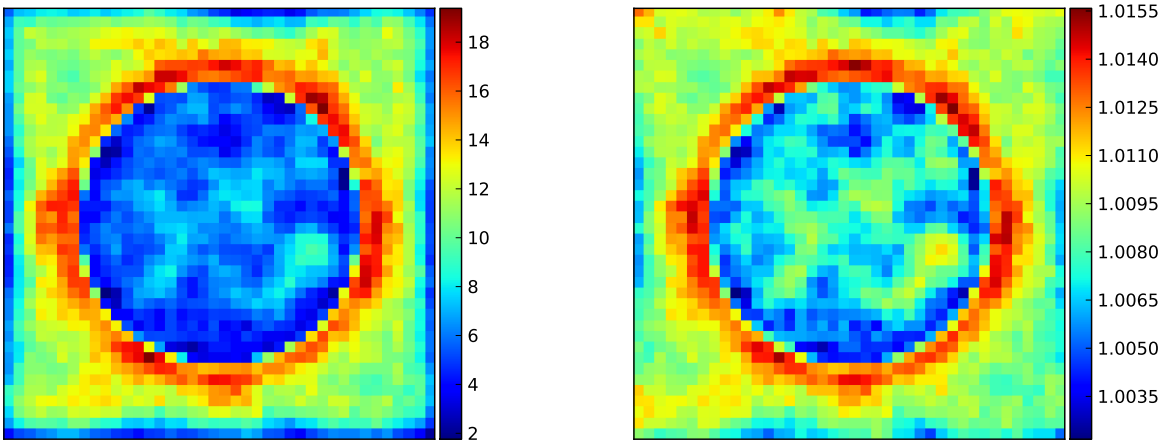
The C/E comparison and the difference between the multivariate MFP KDE and the reference



(a) Difference between results in units of number of σ **(b) C/E values**

Figure 9. Comparison of flux results obtained using the multivariate MFP KDE in Eq. 27 and the MFP KDE defined as a product of univariate kernels in Eq. 16 for a one-group pincell problem.

track-length histogram tally in units of the number of standard deviations are shown in Fig. 10. Figure 10 shows that the multivariate MFP KDE contains a slight bias throughout most of the



(a) Difference between results in units of σ **(b) C/E values**

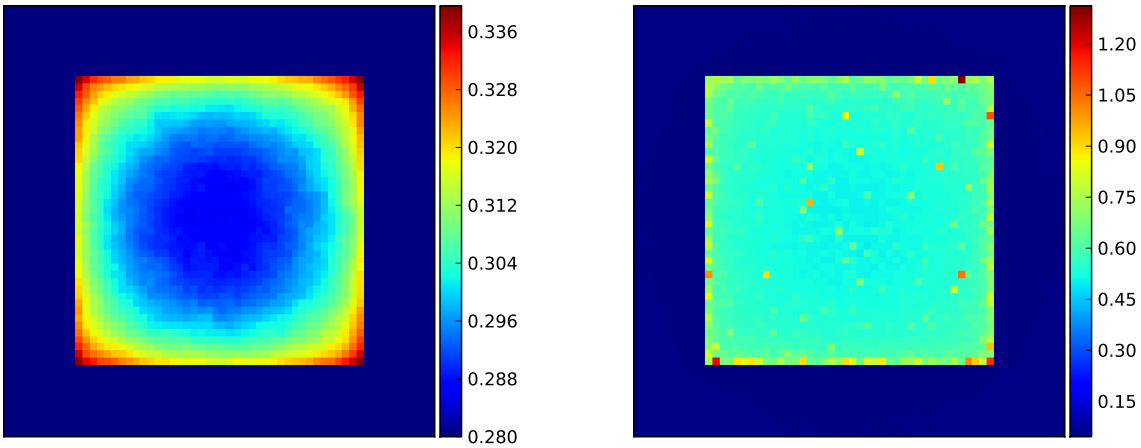
Figure 10. Comparison of flux results obtained using the multivariate MFP KDE and the reference track-length histogram for a one-group pincell problem.

problem, with results disagreeing by as much as 18σ. While the multivariate MFP KDE is biased compared to the histogram, the maximum difference between the results is less than 1.5% at any point. This bias comes from two sources. First, the comparison is between point-wise quantities and

volume-averaged quantities. In regions where the gradient changes, like in the water surrounding a fuel pin, comparing the value at the center of a bin to the average quantity in the bin becomes a poor approximation. Additionally, the KDE results naturally contain a bias by smoothing the results and spreading scores from regions of high concentration to those of low concentration. This is seen through the over prediction of the score in the water surrounding the fuel in Fig. 10. This bias is less apparent in continuous energy problems since the flux distributions have shallower gradients in continuous energy pincell problems. However, it is still unclear as to why the KDE overpredicts the score at every point in this problem compared to the collision histogram tally. Even so, with results that disagree by less than 1.5% at any point, the multivariate MFP KDE still obtains a suitably accurate representation of the underlying distribution.

4.4 2-D Continuous Energy

To test the accuracy of the aMFP KDE in continuous energy, the aMFP KDE is used to estimate flux and reaction rates in a 2-D boxcell problem. The problem consists of a square of 3% enriched UO_2 with a side length of 1.25 cm surrounded by water with a lattice pitch of 1.875 cm with reflecting boundary conditions. Separate KDE regions are defined for the fuel and each slab of water surrounding the fuel, making a total of 5 KDE regions for the 2-D boxcell problem. Figure 11 shows the fission and absorption reaction rate distributions obtained from the aMFP KDE on a structured mesh of 60×60 tally points placed over the problem domain with the simulation run using 200,000 particles per batch, 200 batches and 100 inactive batches. As seen in Fig. 11, the



(a) Fission **(b) Absorption**

Figure 11. Fission and absorption reaction rates estimated using aMFP KDE for the 2-D boxcell problem.

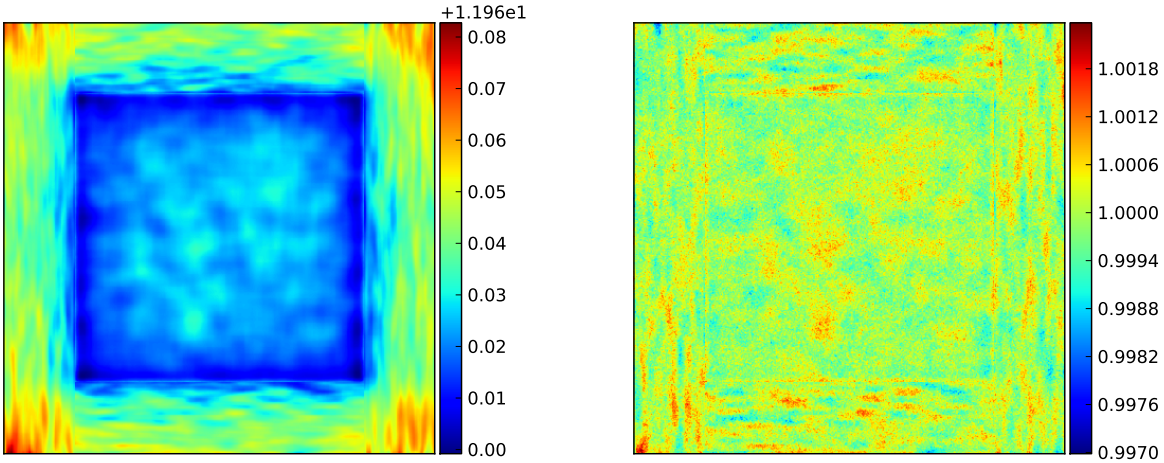
absorption reaction rate contains spikes where the result at one tally point is over twice that at a neighboring tally point less than a millimeter away. However, the fission reaction rate does not suffer from these spikes. This difference is due to the behavior of the absorption cross section in U-238 and the fission cross section in U-235. While the fission reaction rate in U-235 does exhibit

resonances, the maximum resonance is approximately 685 barns at 8.78 eV. Furthermore, the fission cross section at 0.4 eV is approximately 115 barns and follows a $1/v$ behavior. As such, the fission distribution has appreciable scores from both fission resonances and fission in the thermal energy range. On the other hand, The absorption reaction rate in U-238 exhibits several large resonances, the largest being approximately 7,000 barns at 6.67 eV. Since the largest absorption resonance is an order of magnitude above that of the fission resonance, collisions at absorption resonances contribute scores that are 100 times larger over an area that is 100 times smaller than collisions at fission resonances. This, coupled with the low thermal absorption cross section of less than 4 barns over the thermal energy range, means that the contributions to the absorption reaction rate density is dominated from collisions at resonances.

For example, for the 2-D boxcell problem with 200,000 collisions per batch with the bandwidth estimated using data from the previous batch, the bandwidths and support lengths in the fuel are approximately 0.33 mm and 0.74 mm, respectively, in each direction with an average cross section of 0.44 cm^{-1}. When an absorption resonance is encountered where the total cross section is 7,000 barns, the macroscopic cross section for UO_2 is approximately 161 cm^{-1}, thus reducing the spatial bandwidth and support length to 9×10^{-4} mm and 2×10^{-3} mm, respectively, in each direction with a KDE normalization coefficient of approximately 1.2×10^{8} cm^{-2}. If the total cross section is equal to the absorption cross section in the resonance, then this scenario would allow for one collision to contribute a score of over 10^{7} cm^{-2} to the 2-D reaction rate density at a single tally point with no score being contributed to any neighboring tally point. For 2×10^{7} active neutrons in a tally, a single collision at a resonance within the support range of a tally point can increase the final 2-D absorption reaction rate density of a tally point by over 0.5 cm^{-2}. This is seen directly in Fig. 11, where the reaction rate density at several tally points is over twice that of their neighboring tally points.

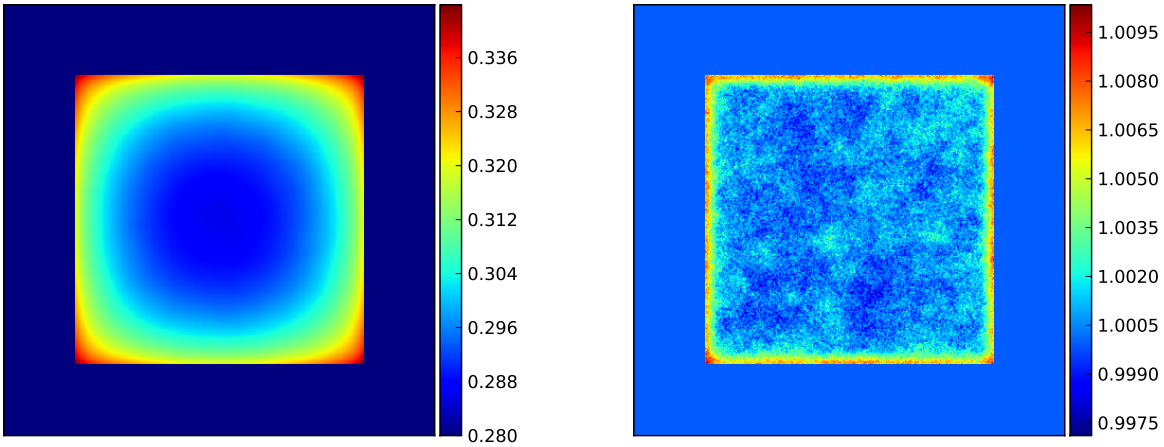
To mitigate this problem, the fractional aMFP KDE was created in Eq. (29). In the scenario previously described, using the fractional aMFP KDE increases the support length to 0.17 mm in each dimension and reduces the normalization coefficient to 17,500 cm^{-2}. This effectively eliminates the spikes seen in the absorption distribution, however it introduces additional the bias at the material interface. To estimate this bias, the fractional aMFP KDE is compared to a TL histogram tally on a 480×480 mesh using 200,000 particles per batch with 2,000 batches and 100 inactive batches. The flux, fission, and absorption reaction rates and the C/E (KDE/histogram) results are shown Figs. 12-14, respectively.

Figures 12-14 show that the fractional aMFP KDE is capable of capturing distributions without spikes. However, some bias exists in the fission and absorption reaction rates. While the maximum difference between the flux distributions is less than 0.4%, the maximum difference in the fission distribution is 1% and the maximum difference in the absorption distribution is 8%. While some of this bias is due to comparing a point-wise quantity to a volume-average quantity, a portion of it is also due to the use of the fractional aMFP KDE. Increasing the resolution of the histogram at the material interface would decrease this error, however it would not eliminate it. Further study is required to determine a more accurate estimate of the bias introduced from the fractional aMFP KDE.



(a) Flux **(b) Flux C/E**

Figure 12. Flux estimated from fractional aMFP KDE and the C/E comparison with a track-length histogram tally.



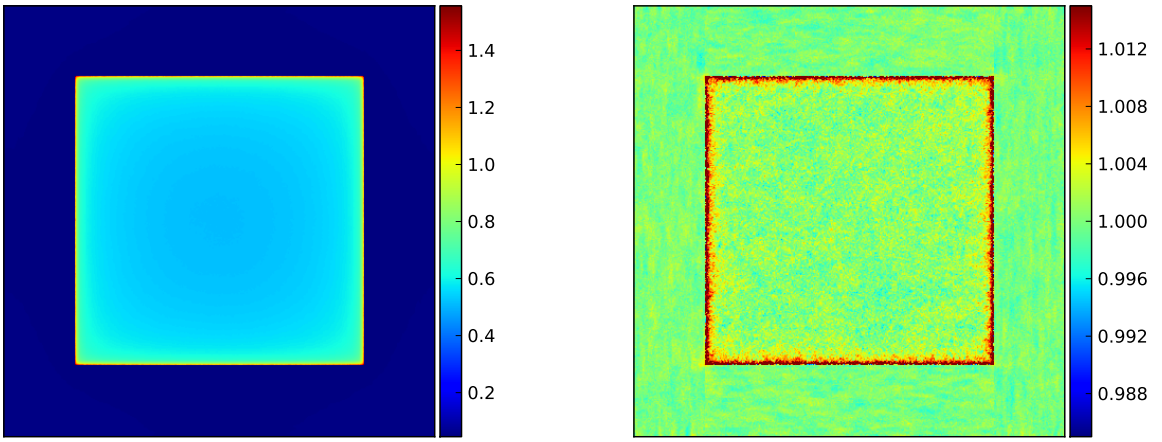
(a) Fission **(b) Fission C/E**

Figure 13. Fission distribution estimated from fractional aMFP KDE and the C/E comparison with a track-length histogram tally.

4.5 GPU Acceleration of 2-D Problems in Continuous Energy

2-D Boxcell

Since the performance of the GPU and CPU algorithms are problem-dependent, several problems were tested and their speedup analyzed. The first problem analyzed is the 2-D boxcell problem



(a) Absorption **(b) Absorption C/E**

Figure 14. Absorption distribution estimated from fractional aMFP KDE and the C/E comparison with a track-length histogram tally.

detailed in Section 4.4. Results are obtained using the fractional aMFP KDE, collision histogram, and track-length histogram on a structured grid of 60×60 tally points and bins with 200,000 particles per batch, 200 batches and 100 inactive batches. The fractional aMFP KDE results are computed using two Tesla M2090 GPUs with 16 MPI processes on two eight-core Intel Xeon E5-2670 processors while the histogram results use 16 MPI processes on the CPU. The fission distributions obtained using the fractional aMFP KDE and the collision histogram are shown in Figure 15. Figures Of Merit (FOM) are calculated using

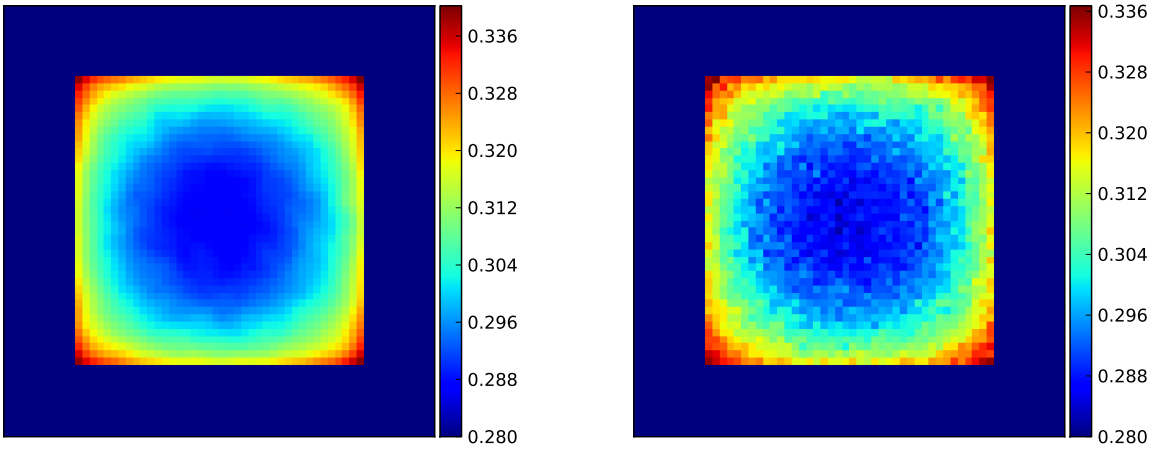
$$\text{FOM} = \frac{1}{\frac{1}{N}\sum_i^N \left(\frac{\sigma_i}{\phi_i}\right)^2 T}, \tag{36}$$

where N is the number of histogram bins or tally points, σ_i/ϕ_i is the relative uncertainty at tally point or bin i, and T is the time spent in the active batches. The ratio of the FOM for the fractional aMFP KDE to the histogram estimators for the flux, fission, and absorption reaction rates in the 2-D boxcell problem are given in Table III.

Table III. Ratios of the FOM for the fractional aMFP collision KDE to the collision and track-length histogram for the 2-D boxcell problem.

FOM Ratio	Flux	Fission	Absorption
KDE/Col Mesh	5.7	10.1	1.2
KDE/TL Mesh	0.28	0.80	0.36

Table III shows that the fractional aMFP KDE shows a favorable FOM compared to the collision histogram for all distributions, while the track-length histogram still has a superior FOM compared

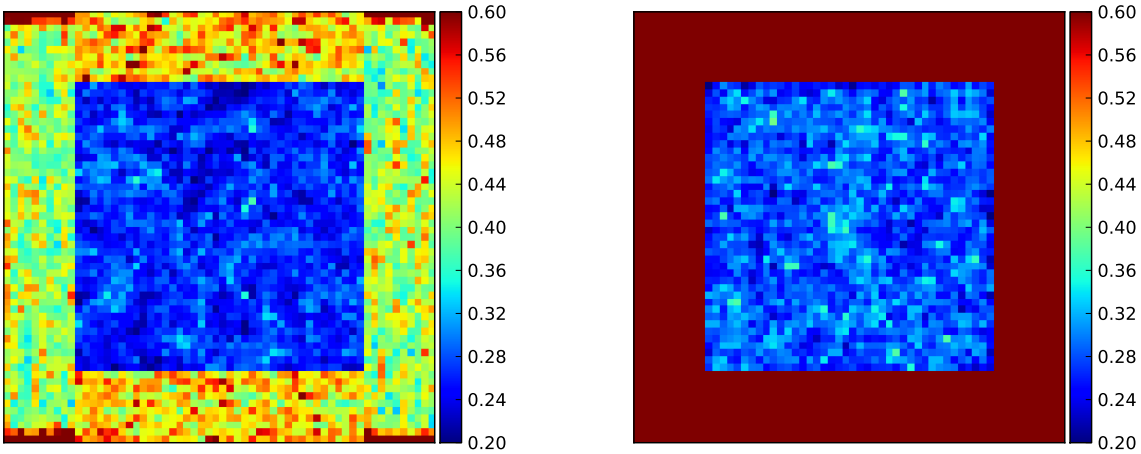


(a) Fractional aMFP KDE **(b) Collision histogram**

Figure 15. Fission reaction rates obtained using the fractional aMFP KDE and the collision histogram.

to the fractional aMFP collision KDE. The fission distributions in Fig. 15 show that the fractional aMFP KDE produces a smoother distribution compared to the collision histogram tally. This smoother distribution has lower variances, thus causing the improved FOM. Also noted in Table III is that the FOM changes for each distribution. This is because the distributions themselves change. The absorption distribution shows worse performance since the majority of the score is concentrated at the material interface due to resonance absorptions. As discussed previously, when a neutron undergoes a collision at a resonance the bandwidth for the KDE decreases. From Eq. (8), reducing the bandwidth increases the variance. Thus, the areas with the largest scores in the absorption distribution will have increased variances due to this reduction in bandwidth for the most important collisions. The difference between the FOM for the flux and fission distributions can be seen in the ratio of the relative uncertainties between the fractional aMFP KDE results and the collision histogram in Fig. 16. As seen in Fig. 16, the ratio of the relative uncertainty in the fission distribution is lower on average compared to the flux distribution for non-zero scores. This is due to how the KDE regions are prescribed. The bandwidth in each KDE region is dependent upon the standard deviation of the flux distribution via Eq. (11). Splitting a given distribution into smaller KDE regions will generate smaller standard deviations, thus reducing the size of the bandwidth. As such, the water cells in the 2-D boxcell problem generally have smaller bandwidths than the distribution in the fuel, causing increased variance in the water compared to the fuel and thus causing the fission distribution to have a better FOM than the flux distribution.

Speedups obtained by using the GPU are computed by comparing run times obtained using the GPUs to run times obtained with an equivalent amount of MPI processes without using GPUs. For the 2-D boxcell problem with 60×60 tally points and 16 MPI processes, using the GPUs results in a speedup of 3.9. For the same problem with 120×120 tally points, a speedup of 9.3 is achieved.



(a) Flux **(b) Fission**

Figure 16. Ratio of relative uncertainties between the fractional aMFP KDE and the collision histogram tally for the 2-D boxcell problem.

Assembly of Boxcells

To test the fractional aMFP KDE and the GPU-accelerated code on a more difficult problem, a quarter-assembly of 16×16 boxcells was modeled. Again, axis-aligned material interfaces are used in order to compare reaction rates to those obtained using histogram tallies. The problem is depicted in Fig. 17 and consists of squares of 3% enriched UO_2 with side length 0.78 cm and 0.6 mm thick Zircaloy 4 cladding surrounded by water with a lattice pitch of 1.26 cm. Absorbers of B_4C replace several of the fuel squares and are shown in black in Fig. 17. Two separate KDE regions are defined for each boxcell: one for the fuel or absorber and another for the surrounding cladding and water. The simulation was run with 200,000 particles per batch, 2,000 total batches with 100 inactive batches. Results are compared to a collision histogram tally and are obtained on a 168×168 structured grid of bins with tally points placed at the center of the bins. Flux, fission, and absorption reaction rates for the KDE tally as well as the C/E comparison with the collision histogram are shown in Figs. 18-20. Figures 18-20 show that the fractional aMFP KDE agrees with the collision histogram throughout most of the flux and fission distributions, however significant disagreement does occur in the absorption distribution. The flux comparison in Fig. 18 shows a maximum difference of 4% occurring in the area around the control rod in the upper left corner where the flux is at a minimum while the majority of the remaining distribution agrees within 1%. It is hypothesized that this disagreement originates from the strong flux gradients around the control rods. Since there is only one mesh bin within the cladding, a strong gradient may strongly influence the flux and reaction rate comparison between the point-wise KDE result and the volume-average histogram result within the cladding. The fission distribution comparison in Fig. 19 shows a maximum difference of 2.2% with no pattern of error. It is likely that this error would be reduced by increasing the number of active particles to reduce uncertainties. The absorption distribution in Fig. 20 shows significant differences, with some regions as high as 14%. Even so, the error is limited to

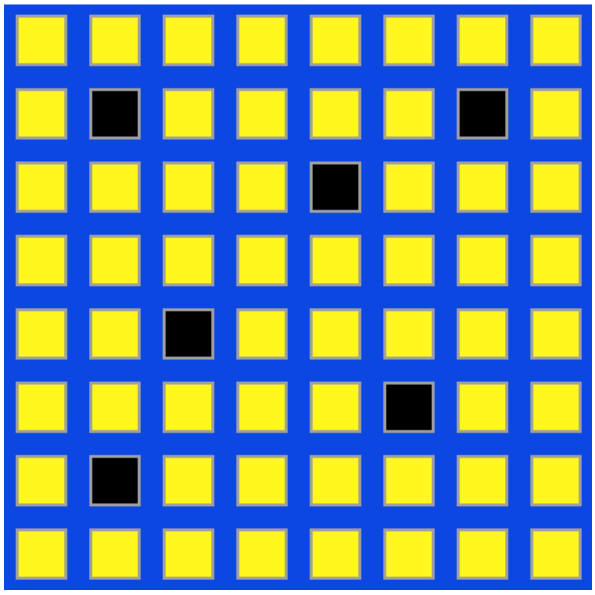
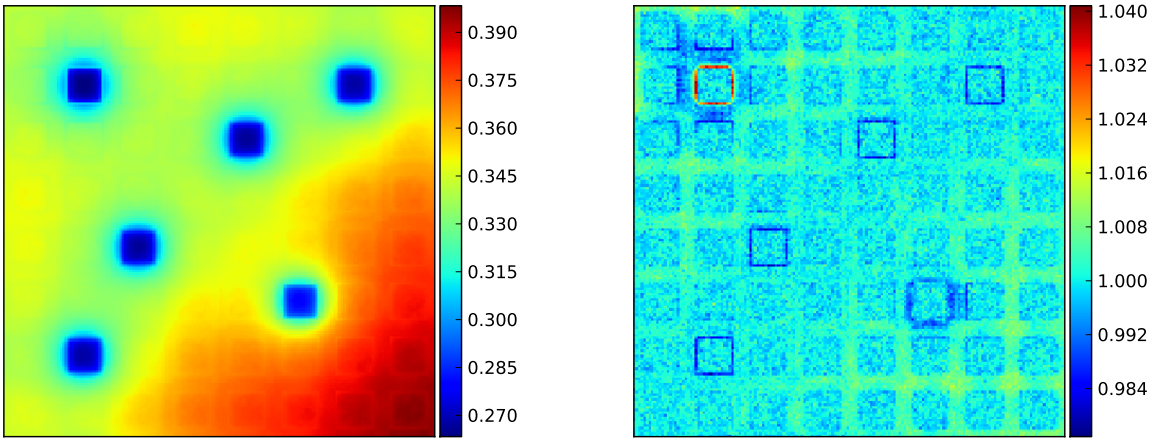


Figure 17. Depiction of the assembly of boxcells.

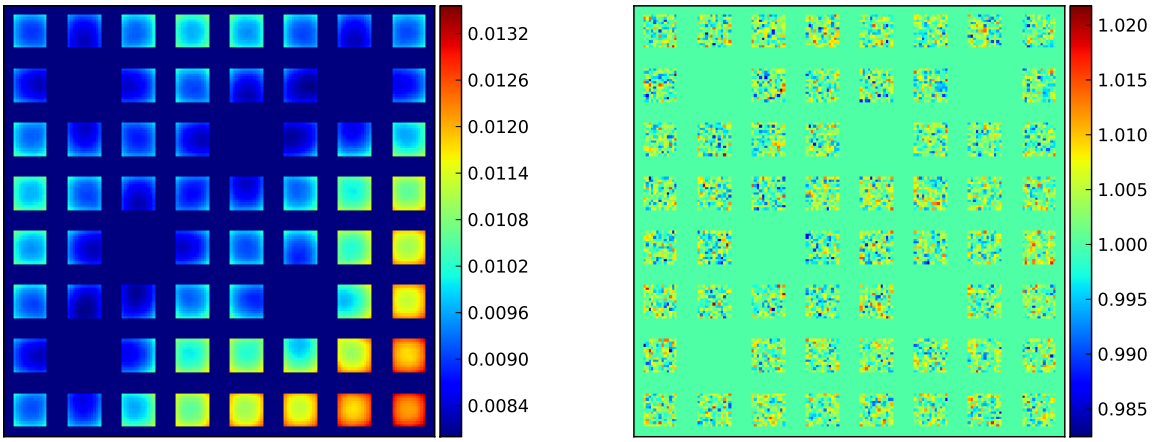


(a) Flux **(b) Flux C/E**

Figure 18. Flux estimated from fractional aMFP KDE and C/E comparison for assembly of boxcells.

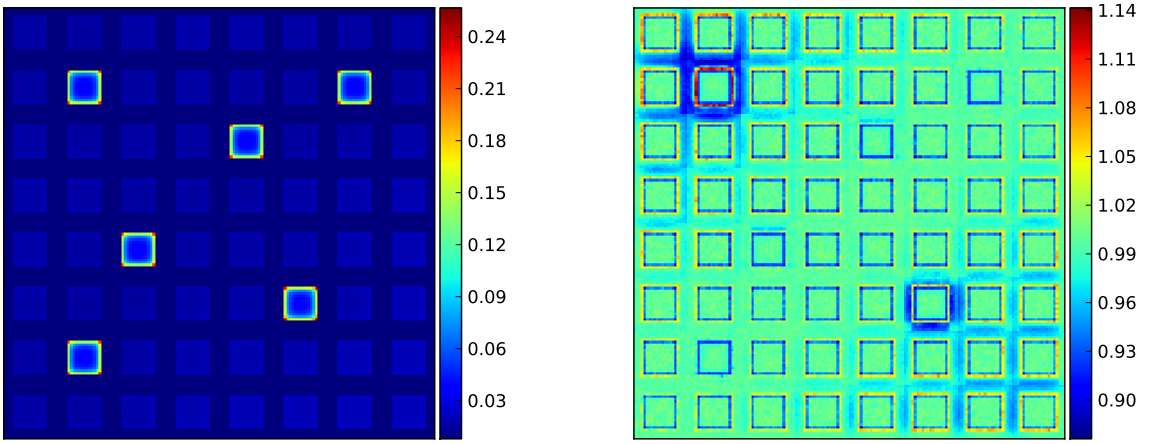
the cladding and water surrounding the fuel and control rods. This error may be a direct result of modeling the water and cladding in a boxcell as a single KDE region. This effectively increases the bandwidth in these regions, thus increasing the bias in the estimator. Further analysis will be performed to determine the exact cause of this discrepancy.

Using 16 MPI processes, a speedup of 13.8 is obtained by exporting the KDE tally to the GPUs. With the use of the GPUs, KDE to collision histogram FOM ratios of 2.6, 1.0, and 0.5 are obtained



(a) Fission **(b) Fission C/E**

Figure 19. Fission estimated from fractional aMFP KDE and C/E comparison for assembly of boxcells.



(a) Absorption **(b) Absorption C/E**

Figure 20. Absorption estimated from fractional aMFP KDE and C/E comparison for assembly of boxcells.

for the flux, fission, and absorption distributions, respectively. The decrease in FOM for the fission distribution compared to the flux distribution is due to using a single KDE region for the water and cladding regions in each boxcell. This increases the bandwidths used for tally points in the water and clad, thus decreasing the variance in those regions compared to that in the fuel regions. Additionally, FOM are lower for the assembly of boxcells versus the single boxcell problem due to adding Zircaloy 4 and B_4C to the list of materials. The KDE calculates all cross sections in the

problem after every collision, so adding more materials will result in longer run times. For the assembly of boxcells problem, approximately 75% of the active tally time spent looking up and storing cross sections in all materials after every collision. Future work will include reducing the burden of looking up cross sections for scoring to materials beyond where the collision occurred.

Assembly of Pincells

To demonstrate the capability of the fractional aMFP KDE to capture reaction rates in geometries with non-planar, a quarter-assembly of pincells, depicted in Fig. 21, was modeled using the same layout as the assembly of boxcells. Each pincell is comprised of a cylinder of 3% enriched UO_2 with a pin diameter of 0.7 cm and a lattice pitch of 1 cm. One KDE region is assigned to each pincell. The simulation was run with 200,000 particles per batch, 2,000 total batches with 100 inactive batches. Reference histogram results were collected on a structured grid of 120×120 bins with KDE tally points placed at the center of each bin. The flux obtained using the fractional aMFP KDE and its C/E comparison to a collision histogram tally is shown in Fig. 22 while the fission and absorption reaction rates are shown in Fig. 23. Figure 22 shows that the fractional aMFP KDE is

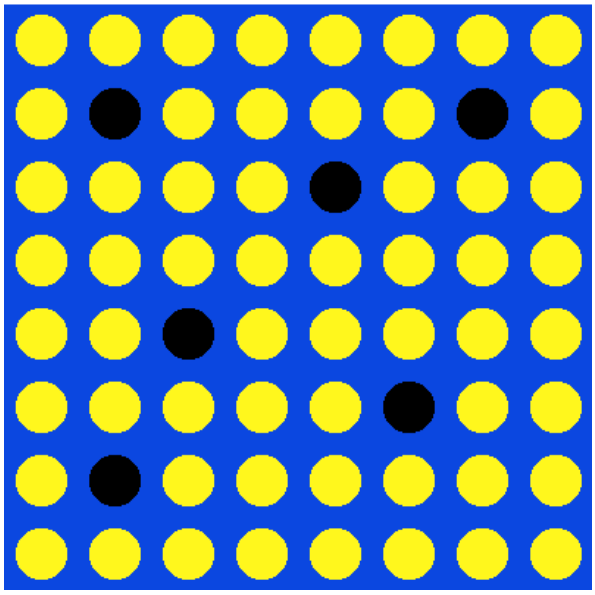


Figure 21. Depiction of the assembly of pincells.

capable of accurately capturing the flux distribution in geometries with non-planar surfaces with differences of less than 1% for all tally points when compared to a collision histogram tally. While differences do exist in the flux around the absorbers, this is likely due to comparing point-wise KDE quantities to volume-average histogram quantities in regions where the gradient of the flux is changing and not a true fault of the KDE. Additionally, the FOM for the fractional aMFP KDE is 3.3 times higher than that of the collision histogram tally. Furthermore, smooth distributions are obtained for the fission and absorption distributions, shown in Fig. 23.

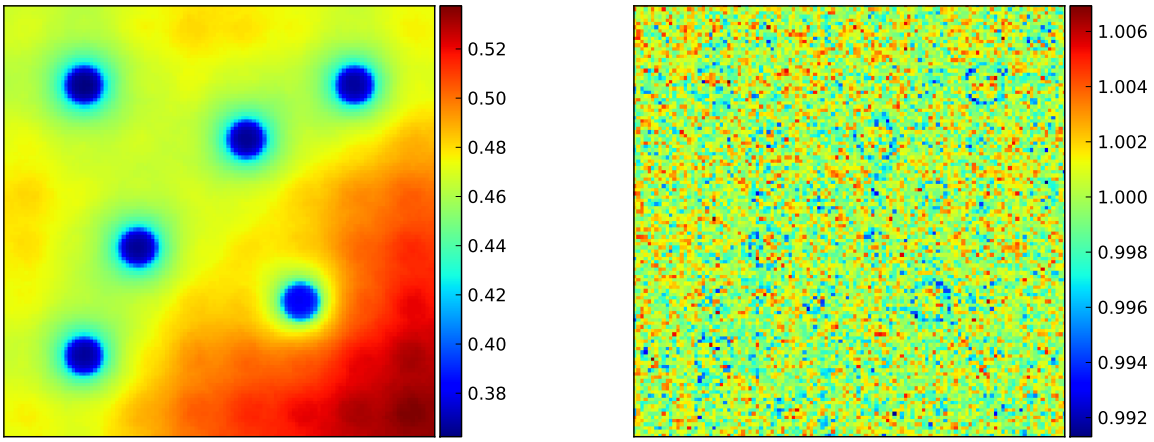
Speedups and runtime statistics for the assembly of pincells with 120×120 tally points and 240×240 tally points are given in Tables IV and V, respectively. Table IV shows that the speedup from

Table IV. Simulation speeds for 2-D pincell assembly problem with 120×120 tally points.

	N MPI Proc.	Active	Inactive	Speedup vs N MPI Proc.
GPU Single Precision	1	10063	17341	1.641599
	2	19410	33513	1.682267
	4	37008	63657	1.858391
	8	66971	117209	1.977529
	16	117615	221870	1.787326
GPU Double Precision	1	10015	17304	1.633768
	2	19491	33704	1.689288
	4	36833	63902	1.849603
	8	66918	118339	1.975964
	16	116889	221237	1.776294
CPU Double Precision	1	6130	17555	-
	2	11538	34157	-
	4	19914	64529	-
	8	33866	119041	-
	16	65805	242964	-

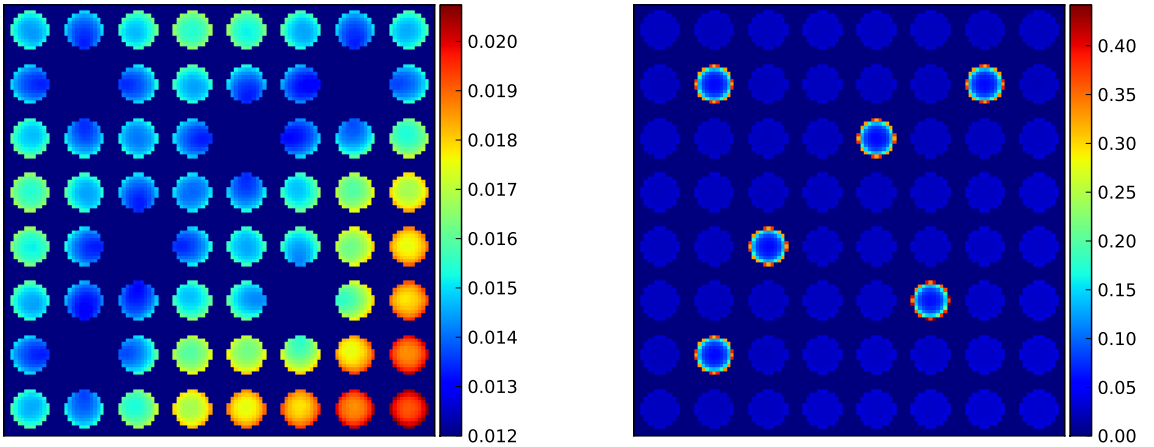
Table V. Simulation speeds for 2-D pincell assembly problem with 240×240 tally points.

	N MPI Proc.	Active	Inactive	Speedup vs N MPI Proc.
GPU Single Precision	1	11591	17560	4.288198
	2	22357	34106	4.657708
	4	41818	64269	4.728403
	8	74568	116962	5.086147
	16	98245	227637	3.728605
GPU Double Precision	1	11448	17676	4.235294
	2	22252	34073	4.635833
	4	41442	64897	4.685889
	8	73157	120479	4.989905
	16	71824	227637	2.725872
CPU Double Precision	1	2703	17676	-
	2	4800	34073	-
	4	8844	64897	-
	8	14661	120479	-
	16	26349	227637	-



(a) Flux **(b) Flux C/E**

Figure 22. Flux distribution from the fractional aMFP KDE and C/E comparison to collision histogram for the assembly of pincells.



(a) Fission **(b) Absorption**

Figure 23. Fission and absorption distributions estimated from fractional aMFP KDE for the assembly of pincells.

using the GPU algorithm for 120×120 tally points is relatively constant versus the number of MPI processes, with a final speedup of 1.8 with 16 MPI processes. This speedup is lower than that of previous problems due to the lower tally point density. The maximum speedup achievable by using GPUs occurs when the entire KDE tally can be computed on the GPUs without the CPU having to wait on the GPU. Thus, if the KDE tally is less expensive on the CPU, then the GPU speedup will be smaller as long as the GPUs are not already saturated with work. The relatively constant

speedup as well as the similar speedups obtained by the single and double precision GPU tallies are an indication that each GPU is able to process all tally arrays sent to it before the CPU sends the next batch of tally arrays. In this scenario the KDE tally is essentially free with the exception of looking up additional cross sections after every collision. This lookup of additional cross sections is the source of the majority of the discrepancy between the active and inactive particle rates in Table IV. Table V shows that increasing the tally point density results in an increase in speedup to 3.7 for 16 MPI processes and single precision GPU tallying. The decline in speedup when switching from 8 to 16 MPI processes as well as the increase in speed up when switching from double to single precision shows that the GPUs are saturated with work when using 16 MPI processes. In fact, the active particle calculation rate decreases when switching from 8 to 16 MPI processes with double precision GPU tallying. This shows that there is a maximum problem complexity that the GPUs are capable of handling before adding more processors per GPU becomes detrimental.

5 CONCLUSIONS

A new multivariate MFP KDE formulation that is capable of handling geometries with non-planar surfaces without approximation has been introduced. Even though this multivariate MFP KDE uses the same normalization coefficient as that of the 2-D MFP KDE formed using a product of univariate kernels, there is a slight bias in the results that is eliminated by using the multivariate MFP KDE. Even with this new multivariate MFP KDE formulation, the approximate MFP KDE is still attractive as a means of reducing computation time by eliminating the need to conduct ray tracing. However, spikes appear in results of the multivariate MFP KDE and the aMFP KDE in 2-D tallies in continuous energy. The use of the fracitonal aMFP KDE eliminates these spikes, however it also introduces additional bias at material interfaces. Even so, the fractional aMFP KDE is capable of capturing the fission distribution in a quarter-assembly problem with less than 2.2% error at any tally point.

Additionally, the fractional aMFP KDE tally was successfully accelerated using GPUs. Speedups are problem dependent, ranging from 1.8 to 13.8 for the problems described in this paper. Figures of merit for the fractional aMFP KDE are generally favorable compared to collision histogram tallies for flux and fission distributions, however the fractional aMFP KDE may not be favorable for absorption distributions. Also, the track-length histogram tally still produces better figures of merit for all distributions than the fractional aMFP KDE with GPU acceleration. This is not surprising, as the aMFP KDE requires cross section information for materials beyond where the collision occurred. Furthermore, the tallies thus far have been conducted on a structured mesh; extending applications to an unstructured mesh would increase the histogram tally complexity without affecting the KDE tally. Thus, its possible for the fractional aMFP KDE to have preferable figures of merit for tallies on an unstructured mesh.

Furthermore, the MFP KDE and aMFP KDE were tested in 1-D problems with thin, strong absorbers. The MFP KDE is capable of capturing the sharp change in flux gradients in problems with thin, strong absorbers, however the aMFP KDE failed to capture the flux distribution near the strong absorber in these problems. An alternative form of the aMFP KDE that uses the maximum cross section in the problem to moderate the bandwidth was created that allows the aMFP KDE to capture

these sharp changes in flux near thin, strong absorbers, reducing the error from 25% to 1.5% at a cost of increased variance in the estimate of the distribution outside of the strong absorber. Even so, the problem modeled with a thin, strong absorber in continuous energy is an unphysical representation of a reactor physics problem, and this issue has not appeared in more physical reactor physics problems studied thus far.

Future work includes creating a MFP KDE that is capable of handling multivariate densities without creating spikes. One potential solution is to use a MFP KDE that is specific to the geometry around the tally point such that only one factor of the cross section is used in the normalization coefficient. Furthermore, the GPU algorithm will be modified to scale with tally point density rather than linearly with tally points. Additionally, the feasibility of exporting track-length KDE tallies to the GPU will be investigated.

6 ACKNOWLEDGEMENTS

This material is based upon work supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1256260 and by the US DOE/NNSA Advanced Scientific Computing program.

7 REFERENCES

[1] K. Banerjee, "Kernel Density Estimator Methods for Monte Carlo Radiation Transport," Ph.D. Thesis, University of Michigan (2010).

[2] K. L. Dunn, "Monte Carlo Mesh Tallies Based on a Kernel Density Estimator Approach," Ph.D. Thesis, University of Wisconsin–Madison (2014).

[3] T. P. Burke, B. C. Kiedrowski, and W. R. Martin, "Mean Free Path Based Kernel Density Estimators for Capturing Edge Effects in Reactor Physics Problems," *M&C*, Nashville, TN, April 19-23, 2015.

[4] T. P. Burke, B. C. Kiedrowski, and W. R. Martin, "Flux and Reaction Rate Kernel Density Estimators in OpenMC," *Trans. Am. Nucl. Soc.*, **109**, (2013).

[5] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, UK (1986).

[6] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, John Wiley and Sons, Inc., Hoboken, NJ (1992).

[7] T. P. Burke, B. C. Kiedrowski, and W. R. Martin, "Extending Flux and Reaction Rate Kernel Density Estimators with Mean Free Path Based Bandwidths to 2-D," 2014, Los Alamos Unclassified Report, LA-UR-14-28469.

[8] "CUDA C Programming Guide," 2015, http://docs.nvidia.com/cuda/cuda-c-programming-guide.

[9] "Tesla GPU Accelerators For Servers," 2015, http://www.nvidia.com/object/tesla-servers.html.

[10] M. C. Jones, "Simple Boundary Correction for Kernel Density Estimation," *Statistics and Computing*, **3**, pp. 135–146 (1993).

[11] P. K. Romano and B. Forget, "The OpenMC Monte Carlo Particle Transport Code," *Ann. Nucl. Energy*, **51**, pp. 274–281 (2013).