Accelerated Solution of One-Dimensional Neutron Transport on General Purpose Graphics Processing Unit *

List of authors:

E. D'Azevedo †

B. Vacaliuc, J. Munro Jr. ‡

Z. Bell §

T. Evans, G. Davidson ¶

This talk will address the accelerated solution of the time independent Boltzmann equation for modeling neutron transport in one-dimensional spherical geometry using the discrete ordinates method on general purpose graphics processing unit (GPGPU). These have several hundred cores and offer potential for very high performance. However, software for GPGPU has to be reformulated to take advantage of the massive parallelism. The transport sweep is a key computational kernel but is difficult to parallelize. Mathematically, the sweep operation is equivalent to the solution of a sparse lower triangular system. Closer examination of the sparsity pattern (see Figure 1 below) suggests the sweep operation can be decomposed as a sequence of solutions to narrow banded lower triangular matrices and sparse matrix multiplies of the off-diagonal entries. Zhang [3] et al. considered the solution of a tridiagonal system on GPGPU using a hybrid algorithm of cyclic reduction, parallel cyclic reduction, and recursive doubling. The cyclic reduction technique can also be used on solving block lower bidiagonal or narrow banded lower triangular systems. We present results on the implementation of cyclic reduction for solving block bidiagonal systems on GPGPU and multi-core machines and the overall impact on the solution of the neutron transport equation.

The multigroup approximation partitions the energy space into G discrete groups and solves the subproblem for one group at a time. The discrete ordinates (S_N) method is a finite element collocation scheme in angle. By representing the scattering source in a Legendre expansion, the resulting equation can be written as [2, Chapter 1]

$$\frac{\mu_m}{r^2}\frac{\partial}{\partial \mu_m}[r^2\psi] + \frac{1}{r}\left\{\frac{\partial}{\partial \mu}[(1-\mu^2)\psi]\right\}_m + \sigma\psi_m = Q_m(r) \tag{1}$$

$$Q_m(r) = \sum_{\ell=0}^{L} \frac{2\ell+1}{4\pi}\sigma_{s\ell}P_\ell(\mu_m)\phi_\ell(r) + q_m(r) \tag{2}$$

†Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831.

‡Measurement Science and Systems Engineering Division, Oak Ridge National Laboratory.

§Global Nuclear Security Technology Division, Oak Ridge National Laboratory.

¶Reactor and Nuclear Systems Division, Oak Ridge National Laboratory

where the angular flux $\psi_m \equiv \psi(r, \mu_m)$ and moments of the angular flux (ϕ_ℓ) are computed by the quadrature formula

$$\phi_\ell(r) = 2\pi \int_{-1}^{1} P_\ell(\mu)\psi(r,\mu)d\mu \approx \sum_{m=1}^{N} P_\ell(\mu_m)\psi_m w_m.$$

With an appropriate spatial discretization, the solution of Eq. (1) with a given Q_m can proceed as a wavefront or sweep algorithm [2]. Starting from the inflow boundary condition, the solution in each cell depends upon the entering flux from the upwind inflow direction of the adjacent neighboring cell. When expressed as matrix operations, Eqs. (1) and (2) can be written as

$$\mathbf{L}\Psi = \mathbf{Q}, \quad \mathbf{Q} = \mathbf{MS}\Phi + \mathbf{q}, \quad \Phi = \mathbf{D}\Psi \tag{3}$$

where $\mathbf{L}$ is the streaming-plus-collision transport operator, $\mathbf{M}$ is the moment-to-discrete operator, $\mathbf{S}$ is the scattering operator and $\mathbf{D}$ maps the angular flux to Legendre moments. Note that $\mathbf{L}^{-1}$ represents a transport sweep. With the appropriate ordering of the variables, the operator $\mathbf{L}$ can be represented as a sparse lower triangular matrix. Figure 1 is a sparsity plot of the transport operator. In the neutron transport community, Eq. (3) is commonly solved using a fixed-point iteration known as *source iteration*

$$\Phi^{k+1} = \mathbf{DL^{-1}}(\mathbf{MS}\Phi^k + \mathbf{Q}). \tag{4}$$

Source iteration is stable but can suffer from slow convergence if the material is highly scattering and optically thick. Krylov methods, such as GMRES or BICGSTAB, may also be used to solve the system

$$(\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS})\Phi = \tilde{\mathbf{Q}}, \quad \tilde{\mathbf{Q}} = \mathbf{DL}^{-1}\mathbf{Q}\ . \tag{5}$$

The transport sweep operation $\mathbf{L}^{-1}$ is a key computational kernel since the action of the matrix-vector multiply is computed as needed. The main insight is that the transport sweep $\mathbf{L}^{-1}$ can be computed as a sequence of solutions to narrow banded lower triangular systems and sparse matrix multiplications of the off-diagonal entries.

Cyclic reduction (CR) [1] for solving a block bidiagonal lower triangular system exposes more parallelism but performs more work. For a system with n blocks, each a $k \times k$ matrix, CR takes $2\log_2(n)$ stages, and $O(3nk^3+6nk^2)$ operations, whereas the classical sequential (CS) algorithm requires n stages and $O(3nk^3)$ operations. If there are $(n/2)$ processors, the ratio of time for CR over CS is roughly $(3k+2)\log_2(n)/n$. The greater parallelism coupled with the availability of more computational cores makes GPGPU approach plausible. At each forward stage, the method reduces (in parallel) the system to another block bidiagonal system of variables with even index. On the backward stage, once the even variables are known, the odd variables can be computed in parallel. We illustrate the method on a 6×6 problem,

$$\begin{bmatrix} b_1 & & & & & \\ a_2 & b_2 & & & & \\ & a_3 & b_3 & & & \\ & & a_4 & b_4 & & \\ & & & a_5 & b_5 & \\ & & & & a_6 & b_6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix}. \tag{6}$$

The method orders the odd and even variables separately and then forms the Schur's complement for the even variables,

$$\begin{bmatrix} b_2 & & \\ -a_4 b_3^{-1} a_3 & b_4 & \\ & -a_6 b_5^{-1} a_5 & b_6 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ x_6 \end{bmatrix} = \begin{bmatrix} d_2 \\ d_4 \\ d_6 \end{bmatrix} - \begin{bmatrix} a_2 b_1^{-1} d_1 \\ a_4 b_3^{-1} d_3 \\ a_6 b_5^{-1} d_5 \end{bmatrix} . \tag{7}$$

The reordering can be recursively repeated on the smaller system. Once the even variables are known, the odd variables can be computed in parallel as

$$x_1 = b_1^{-1} d_1, \quad x_3 = b_3^{-1}(d_3 - a_3 x_2), \quad x_5 = b_5^{-1}(d_5 - a_5 x_6). \tag{8}$$

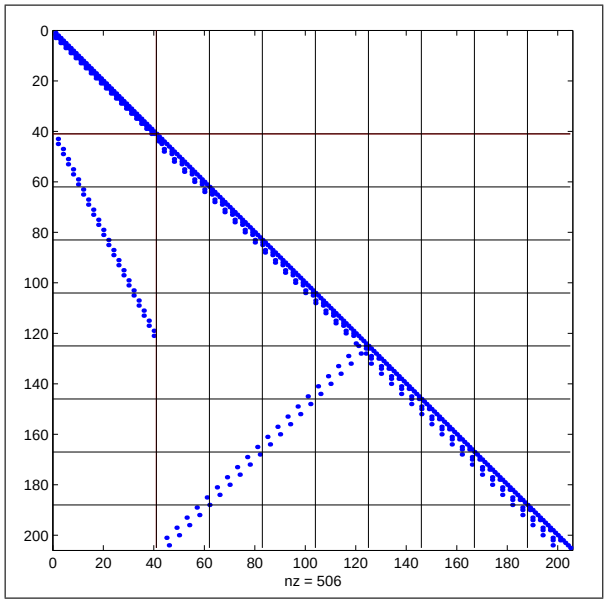

Figure 1: Sparsity pattern of transport sweep operator with 4 quadrature directions and 20 spatial cells and one moment.

References

[1] R. Hockney, "A fast direct solution of Poisson's equation using Fourier analysis" *Journal of the ACM*, 12(1):95-113, 1965.

[2] E. Lewis and W. Miller Jr., "Computational methods of neutron transport", American Nuclear Society, La Grange Park, Ill., USA, 1993.

[3] Y. Zhang Y. and J. Cohen J. and J. Owens , "Fast Tridiagonal Solvers on the GPU", *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP 2010)*, January 2010.