

LA-UR-13-21067

Approved for public release;  
distribution is unlimited.

<i>Title:</i>	SNAP: SN (Discrete Ordinates) Application Proxy - Brief Description
<i>Author(s):</i>	Zerr, Robert J. Baker, Randal S.
<i>Intended for:</i>	Documentation intended to be distributed with/without developing proxy application SNAP.



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# SNAP: SN (Discrete Ordinates) Application Proxy, Rev 1

## Brief Description

Joseph Zerr (rzerr@lanl.gov) and Randal Baker (rsb@lanl.gov)

Los Alamos National Laboratory

Computational Physics and Methods, CCS-2

Updated: 26 March 2013

A new proxy application is being developed to model the performance of a modern discrete ordinates neutral particle transport application. SNAP may be considered an update to Sweep3D<sup>1</sup>, intended for hybrid computing architectures. It is modeled off the Los Alamos National Laboratory code PARTISN. PARTISN solves the linear Boltzmann transport equation (TE), a governing equation for determining the number of neutral particles (e.g., neutrons and gamma rays) in a multi-dimensional phase space. SNAP itself is not a particle transport application; SNAP incorporates no actual physics in its available data, nor does it use numerical operators specifically designed for particle transport. Rather, SNAP mimics the computational workload, memory requirements, and communication patterns of PARTISN. The equation it solves has been composed to use the same number of operations, use the same data layout, and load elements of the arrays in approximately the same order. Although the equation SNAP solves looks similar to the TE, it has no real world relevance.

The solution to the time-dependent TE is a “flux” function of seven independent variables: three spatial (3-D spatial mesh), two angular (set of discrete ordinates, directions in which particles travel), one energy (particle speeds binned into “groups”), and one temporal. PARTISN, and therefore SNAP, uses domain decomposition over these dimensions to coherently distribute the data and the tasks associated with solving the equation. The parallelization strategy is expected to be the most efficient compromise between computing resources and the iterative strategy necessary to converge the flux.

The iterative strategy is comprised of a set of two nested loops. These nested loops are performed for each step of a time-dependent calculation, wherein any particular time step requires information from the preceding one. No parallelization is performed over the temporal domain. However, for time-dependent calculations two copies of the unknown flux must be stored, each copy an array of the six remaining dimensions. The outer iterative loop involves solving for the flux over the energy domain with updated information about coupling among the energy groups. Typical calculations require tens to hundreds of groups, making the energy domain suitable for threading with the node’s (or nodes’) provided accelerator. The inner loop involves sweeping across the entire spatial mesh along each discrete direction of the angular domain. The spatial mesh may be immensely large. Therefore, SNAP spatially decomposes the problem across nodes and communicates needed information according to the KBA<sup>2</sup> method. KBA is a transport-specific application of general parallel wavefront methods. Nested threads, spawned by energy group threads, are available to perform “mini KBA” sweeps by concurrently operating on cells lying on the same diagonal of spatial sub-domains already decomposed across the distributed memory architecture (i.e., different MPI ranks). Lastly, although KBA efficiency is improved by pipelining operations according to the angle, current chipsets operate best with vectorized operations. During a mesh sweep, SNAP operations are vectorized over angles to take advantage of the modern hardware.

SNAP shall be tested with problem sizes that accurately reflect the types of calculations PARTISN frequently handles. The spatial domain shall be decomposed to 2,000–4,000 cells per node (MPI rank). Each node will own all the energy groups and angles for that group of cells; typical calculations feature 30–70 energy groups and as few as 100 to as many as 2,000 angles. Moreover, sufficient memory must be provided to store two full copies of the solution vector for time-dependent calculations. The preceding parameters assume current trends in available per-core memory. Significant advances or detriments affecting this assumption shall require reconsideration of appropriate parameters per compute node.

---

<sup>1</sup> “LANL: CCS-3: Performance and Architecture: Software,” Available at <http://www.ccs3.lanl.gov/PAL/software.shtml>,” Last accessed 11/30/2012 (2006).

<sup>2</sup> R.S. Baker and K.R. Koch, “An  $S_n$  Algorithm for the Massively Parallel CM-200 Computer,” *Nuclear Science and Engineering*, 128, 312 (1998).