International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)
Saratoga Springs, New York, May 3-7, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2009)

SAND2009-1475C

PARALLEL DISCRETE ORDINATES METHODS IN THE SCEPTRE PROJECT

Shawn Pautz, Bill Bohnhoff, Clif Drumm and Wesley Fan
Sandia National Laboratories*
Albuquerque, NM 87185-1179
sdpautz@sandia.gov, wjbohnh@sandia.gov, crdrumm@sandia.gov, wcfan@sandia.gov

ABSTRACT

Although the transport of photons and electrons/positrons is described by the same Boltzmann transport equation, the cross sections are very different, resulting in very different solution convergence properties for the two particle types. The SCEPTRE project is a suite of deterministic codes for solving the linear steady-state Boltzmann transport equation, containing two very different solver approaches: a sweeps-based approach for solving the first-order transport equation that is efficient for photon transport, and a conjugate-gradients algorithm for solving the second-order transport equation that is efficient for electron/positron transport. The two solver algorithms are coupled through a common set of tools for handling the spatial finite elements methods, angular discrete-ordinates methods, scattering source terms, and input, pre- and post-processing capabilities. We evaluate the two solver algorithms by comparing solver run times for photon- and electron-transport problems, investigating properties such as parallel performance, finite-elements basis function type, preconditioning, and scaling with angular quadrature order and Legendre cross section expansion order.

Key Words: coupled photon-electron transport, finite elements, parallel

1. INTRODUCTION

SCEPTRE (Sandia's Computational Engine for Particle Transport for Radiation Effects) is a suite of deterministic codes for solving the linear steady-state Boltzmann transport equation. It solves the first- and/or second-order forms of the transport equation by means of the multigroup energy discretization, the discrete ordinates angular discretization, and finite element spatial discretization on unstructured meshes [1],[2]. Currently, SCEPTRE is primarily applied to predict the effects of x-rays and secondary electrons on cables and other electronic components. The high resolution needed for the accurate modeling of electron transport near conductor-dielectric boundary layers requires the use of large meshes and massively parallel computations.

The first-order solver in SCEPTRE uses discontinuous finite element differencing. The solution method employs conventional source iteration and a parallel wavefront (sweeping) algorithm on distributed meshes. The sweeping algorithm mostly respects the streaming dependency graph; some dependencies may be ignored during the sweep in order to avoid cyclic dependencies and to improve performance. This approach typically preserves the iteration count needed for

* Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DEAC04-94AL85000

solution convergence of a serial computation; there is little or no degradation in iterative effectiveness as the processor count increases. However, this approach also causes degradation in the scaling of each iteration as the number of processors increases, since it becomes increasingly difficult to keep each processor occupied with useful tasks. Although SCEPTRE can in principle be used with any mesh partitioning, to date we have used only conventional partitioning approaches, which further limits the parallel scaling of the first-order method.

The second-order solver in SCEPTRE uses continuous finite element differencing. This discretization yields a symmetric positive definite (SPD) matrix that couples all angular and spatial variables, permitting the use of a parallel conjugate gradient (PCG) solver and eliminating the need for source iteration. This approach, unlike that for the first-order form, yields good scalability of each iteration as the processor count increases. SCEPTRE includes three different second-order solver algorithms: even-odd parity flux (EOPF) [3], self-adjoint angular flux (SAAF) [4] and least-squares finite elements (LSFE) [5]. The three algorithms have different strengths and weaknesses, e.g. the LSFE method can be used for problems containing internal void regions without special treatment, and the problem size for EOPF method is half that of the other methods due to symmetry. SCEPTRE includes both S_n (discrete ordinates) and P_n (spherical harmonics) second-order solver algorithms, but only the S_n algorithm will be considered here since the P_n algorithm has not yet been extensively tested.

The iterative solution process of the first-order form is fundamentally different than that of the second-order form, resulting in a different set of strengths and weaknesses for each approach. In SCEPTRE the user can choose on a group-by-group basis whether to use the first-order solver or one of the second-order solvers. This is extremely beneficial for coupled photon-electron transport problems, since the two solvers have very different convergence properties for different particle types and different energy regimes. The solver that is best suited for solving each particle type at each energy can be chosen to provide the most efficient global solve. This is especially beneficial for multi-particle applications, where the cross sections may be radically different, e.g. the transport mean free path of 200-keV photons is 4-5 orders of magnitude larger than that of 200-keV electrons, and the scattering ratio of 200-keV photons is about 0.1, while that of 200-keV electrons is typically greater than 0.9.

In general, the first-order solver works better for streaming-dominated problems, such as photon transport, and the second-order solvers work better for problems with higher scattering ratios, such as electron transport. The convergence rate of the first-order solver is strongly influenced by the scattering ratio, while the convergence rate of the second-order solvers depends upon the condition number of the matrix resulting from the space-angle discretization of the problem. For the second-order solvers, effective preconditioning can greatly reduce the condition number of the matrix, resulting in drastic reduction in solver times, but further work remains to be done to arrive at optimal preconditioning. The convergence properties of the first- and second-order algorithms in many ways complement each other.

The first- and second-order transport algorithms share many components in common, including finite-elements tools, cross section handling, angular quadrature, angular moments, distributed source data, boundary conditions and transport field containers. The transport field containers and distributed source data are based on discontinuous finite elements. Since the second-order

solvers use continuous finite elements, a mapping is required from discontinuous data structures to continuous for the second-order solvers, and then back again to map the results to the discontinuous data structures for compatibility with the first-order solver. These mappings are performed seamlessly and efficiently with Trilinos tools [6].

2. PROBLEM DESCRIPTION

The problem considered is a two-dimensional cross section of a coaxial cable, with dimensions and materials as shown in Table I. The cable is small and relatively transparent to photons but not to electrons, as can be seen by looking at the typical cross sections shown in Tables IIa and IIb. The cross sections were computed by the CEPXS code [7], which is a physics code for computing multigroup-Legendre cross sections for use in deterministic and multigroup Monte Carlo radiation transport codes.

Table I. Coaxial cable specifications

Layer	Material	Outer Radius (cm)	Thickness (cm)
Center Conductor	iron	0.0478	0.0956
	copper	0.0594	0.0116
	silver	0.0606	0.0012
Dielectric	PTFE	0.15113	0.09053
Outer Conductor	copper	0.17907	0.02794

Table IIa. 200-kev photon cross sections

	iron	copper	silver	teflon
σ_t	1.17004×10^{0}	1.42261×10^{0}	3.16203×10^{0}	2.62917×10^{-1}
σ_{s0}	1.13938×10^{-1}	1.45778×10^{-1}	3.08826×10^{-1}	1.30645×10^{-2}
σ_{s1}	1.08554×10^{-1}	1.38100×10^{-1}	2.83913×10^{-1}	1.28180×10^{-2}
σ_{s2}	1.01109×10^{-1}	1.28283×10^{-1}	2.62529×10^{-1}	1.23410×10^{-2}
σ_{s3}	9.31516×10^{-2}	1.18137×10^{-1}	2.40030×10^{-1}	1.16613×10^{-2}
scattering ratio	0.0974	0.102	0.0977	0.0497

At present, the primary application of SCEPTRE is to predict the effect of x-rays and secondary electrons on cables and other electronic components. The result of interest in these types of calculations is the electron emission from the conductors and penetration into the dielectric materials. Because of the large magnitude of the electron cross sections and the large magnitude of the production of electrons in high atomic number materials, the gradient of the electron fluence near material interfaces is very large, so that a fine spatial resolution (typically sub micron) is needed near interfaces for acceptable accuracy. Ideally, it would be desirable to use a different mesh for different particle types and even for different energy regimes, with the spatial dependence of the mesh refinement driven by the spatial gradients in the solution. However, the

mapping of source terms in parallel from coarse to fine meshes is a difficult bookkeeping problem and is not currently implemented in SCEPTRE. This is not a serious problem for the first-order sweeps-based algorithm, but for the second-order algorithms, transporting photons on an over-refined mesh increases the condition number of the resulting matrix, resulting in slow convergence. Effective preconditioning of the matrix can substantially mitigate this effect, however.

Table IIb. 200-kev electron cross sections

	iron	**copper**	**silver**	**teflon**
σ_t	6.97961×10^4	8.34048×10^4	1.18091×10^5	1.04219×10^4
σ_{s0}	6.44678×10^4	7.75741×10^4	1.12025×10^5	8.58885×10^3
σ_{s1}	6.39921×10^4	7.69881×10^4	1.10984×10^5	8.54100×10^3
σ_{s2}	6.32079×10^4	7.60309×10^4	1.09359×10^5	8.45897×10^3
σ_{s3}	6.21969×10^4	7.48034×10^4	1.07343×10^5	8.34969×10^3
σ_{s4}	6.10023×10^4	7.33574×10^4	1.05020×10^5	8.21724×10^3
σ_{s5}	5.96529×10^4	7.17264×10^4	1.02443×10^5	8.06450×10^3
σ_{s6}	5.81699×10^4	6.99346×10^4	9.96485×10^4	7.89366×10^3
σ_{s7}	5.65695×10^4	6.80013×10^4	9.66656×10^4	7.70649×10^3
scattering ratio	0.924	0.930	0.949	0.824

Typical finite-elements meshes of the cross section of a coaxial cable are shown in Figs. 1a and 1b. Fig. 1a shows a quadrilateral mesh of the coaxial cable, refined to 2 μm near the conductor-dielectric interafaces. The triangular mesh is shown in Fig. 1b, which was obtained by splitting quadrilaterals in half. Creating a mesh in this way results in large aspect ratios. The aspect ratios can be decreased, with the result of increasing the size of the mesh.

3. RESULTS

This section compares the performance of the second-order SAAF algorithm and the first-order sweeps-based algorithm by comparing convergence iteration count and solver runtimes for a variety of properties, including finite-elements basis function type and mesh refinement, preconditioning, parallel performance, and scaling with angular quadrature order and Legendre cross section expansion order. Test problems were run on Sandia's Thunderbird cluster, with hardware environment of dual 3.6 GHz intel EM64T processors with 6 GB RAM. Thunderbird's high-speed message passing fabric is Infiniband. The Infiniband fabric is a two level CLOS topology with eight top-level core switches and 280 leaf switches (24 ports per leaf switch) that the compute nodes connect to. Each leaf switch has 16 downlinks (16 compute nodes per leaf switch) and 8 uplinks. Thus, the network is 2-to-1 oversubscribed in terms of raw number of links. SCEPTRE was compiled with optimization with GCC version 3.4.6 and Open MPI version 1.2.7.

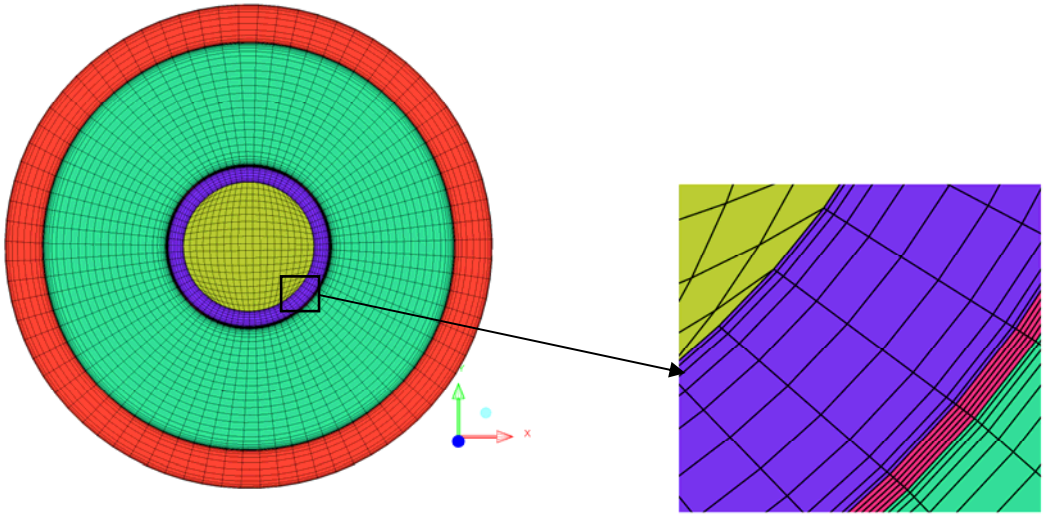

Figure 1a. Quadrilateral mesh of coaxial cable, showing close up near inner conductor-dielectric layer.

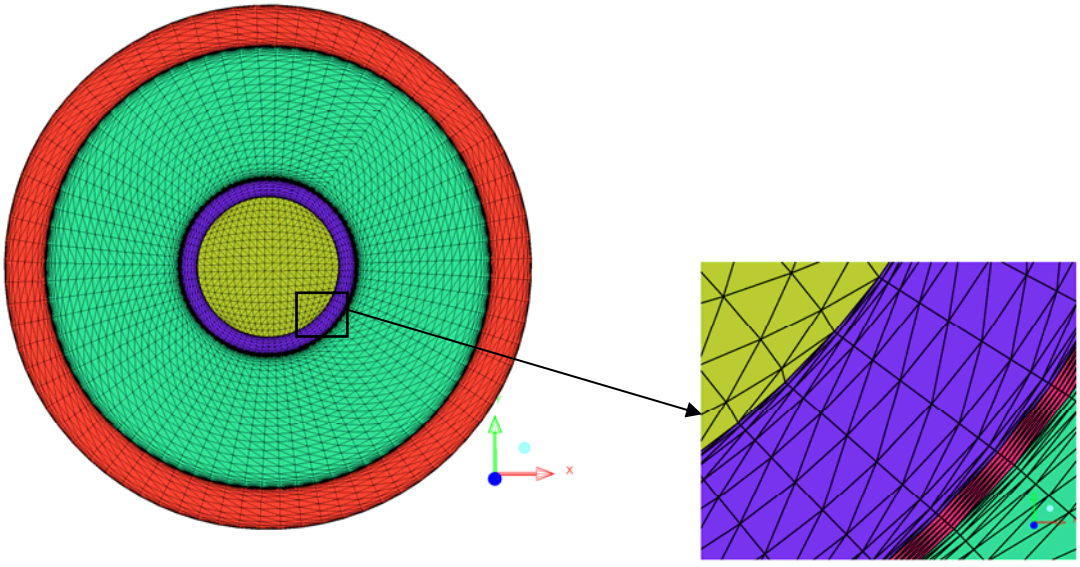

Figure 1b. Triangular mesh of coaxial cable, showing close-up near inner conductor-dielectric layer.

3.1 Finite-Element Basis Function Type and Mesh Refinement

Table IIIa shows results for the transport of 200-keV photons on the coaxial cable meshes shown in Figs. 1a and 1b. Linear (quad4 and tri3) and quadratic (quad8 and tri6) finite-element basis functions were compared. 8^{th} order angular quadrature and 3^{rd} order Legendre cross section expansion were used in the calculations, with convergence tolerance of 10^{-8}, which were run on four processors. The first- and second-order algorithms use different expressions of convergence tolerance: the SAAF solver uses the 2-norm of the point wise relative residual from the PCG iterations as the tolerance, and the first-order solver uses a norm of the solution difference between successive iterates. More work is needed to evaluate the effect of the convergence tolerance on the results of interest in the calculations. The SAAF calculations used a block-diagonal preconditioner, which is described in Sec. 3.2.

A number of interesting observations can be made from the results. The first-order algorithm scales roughly with the number of elements in the mesh, while the SAAF algorithm scales more directly with the number of nodes. The SAAF runtimes are much larger for quadratic basis functions (quad8 and tri6) as compared with linear basis functions, partly as a result of the increase in the number of nodes and partly as a result of increased cost of the preconditioning.

Table IIIa. Comparison of transport algorithms for various element types and mesh refinement for 200-keV photon transport

S_8P_3, tol=10^{-8}, 4 processors			SAAF (preconditioned)		First Order	
Elem Type	**Num Elems**	**Num Nodes**	**Num Iters**	**Solver Time (s)**	**Num Iters**	**Solver Time (s)**
quad4	4,845	4,884	6	8.61	6	7.45
quad8	4,845	14,612	"	47.5	"	12.2
tri3	9,690	4,884	"	8.37	"	14.1
tri6	9,690	19,457	"	55.4	"	17.8
Refined Meshes:						
quad4	19,380	19,457	6	67.1	6	35.0
tri3	38,760	19,457	"	53.5	"	59.0

The results for 200-keV electron transport are shown in Table IIIb. Linear (quad4 and tri3) and quadratic (quad8 and tri6) finite-element basis functions were compared. 12^{th} order angular quadrature and 7^{th} order Legendre cross section expansion were used in the calculations, with convergence tolerance of 10^{-8}, which were run on 32 processors. A higher-order Legendre cross section expansion is used for the electron transport calculations, due to the higher degree of anisotropy in the electron scattering physics. No preconditioning was used for the SAAF results, as an effective preconditioner for the electron transport has not yet been found, so that preconditioning has little effect or even a negative effect on convergence of electron groups.

The first-order algorithm iteration counts are much higher for electron transport, as compared with those for photon transport, as expected due to the higher scattering ratio of electron cross

sections. The first-order solver times are quite a bit higher for electron transport than for photon transport, even with using 32 processors instead of 4.

Like the photon-transport results, the first-order algorithm scales roughly with the number of elements in the mesh, while the SAAF algorithm scales more directly with the number of nodes. The SAAF runtimes are much larger for quadratic basis functions (quad8 and tri6) as compared with linear basis functions, partly as a result of the increase in the number of nodes and partly as a result of increased cost of the preconditioning.

Unlike the photon results, the SAAF algorithm performs better with a refined mesh with linear basis functions, rather than a coarser mesh with quadratic basis function. Further work remains to be done to determine computational efficiency as a function of accuracy of refining a mesh vs. using a higher basis function order.

Table IIIb. Comparison of transport algorithms for various element types and mesh refinement for 200-keV electron transport

$S_{12}P_7$, tol=10^{-8}, 32 processors			SAAF (unpreconditioned)		First Order	
Elem Type	**Num Elems**	**Num Nodes**	**Num Iters**	**Solver Time (s)**	**Num Iters**	**Solver Time (s)**
quad4	4,845	4,884	29	3.5	55	36.9
quad8	4,845	14,612	133	49.6	"	71.1
tri3	9,690	4,884	29	3.1	"	53.2
tri6	9,690	19,457	78	32.9	"	98.5
Refined Meshes:						
quad4	19,380	19,457	56	19.1	55	155.
tri3	38,760	19,457	57	16.5	"	234.

3.2 Preconditioning the Second-Order Transport Algorithm

As mentioned previously, modeling photon transport on an over-refined mesh, such as a mesh refined to accurately model electron transport, results in an ill-conditioned matrix and slow convergence for the second-order transport algorithms. Effective preconditioning can dramatically improve the convergence. We have found that for the photon transport, a block-diagonal preconditioner is very effective [8]. Rather than solving the unpreconditioned linear system

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1a}$$

a different linear system, $M^{-1}A$, with a lower condition number than the original linear system, A, is solved instead

$$M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{b}. \tag{1b}$$

An effective preconditioner, *M*, should be relatively easy to invert and a good approximation to the original linear system.

SCEPTRE contains several preconditioning options, including multilevel (ML) and incomplete factorization (IF), which need further investigation. We have found that an effective preconditioner for streaming-dominated problems, such as photon transport in small geometries, is the obtained by using the pure-absorption transport equation, which is nearly equivalent to using the diagonals of the blocks in the discretized transport linear system, *A*, as a preconditioner. This effectively replaces the full system solve, which scales as the number of directions squared, with a set of preconditioner linear systems, which scale linearly with the number of directions.

Implementing the block-diagonal preconditioner requires a procedure for inverting the preconditioning operator, *M*. There are several ways of solving the preconditioner (uncollided-flux) system, and the results for two different methods are shown here. One method is to use a PCG algorithm for each discrete direction. This method is fairly easy to implement, scales well with number of processors, but has the drawback that if *A* is ill-conditioned, *M* will also likely be ill-conditioned, resulting in slow convergence.

Another method is to use a direct solver to invert the preconditioner linear system for each discrete direction, and then to keep it for use in subsequent iterations. The KLU sparse LU factorization algorithm [10] is implemented in the Trilinos package and has been implemented in SCEPTRE. This approach results in very efficient convergence of the SAAF algorithm but is a serial implementation. Parallel direct solvers have not yet been implemented into SCEPTRE. Table IV compares the results of the SAAF algorithm with the first-order algorithm for several preconditioning options. Despite being limited to a serial implementation, the KLU option works quite well, resulting in iteration counts and run times comparable to the first-order algorithm.

Table IV. Effect of preconditioning of SAAF algorithm on 200-keV photon transport

S_8 P_3, tol=10^{-8}, quadratic-triangular (tri6) mesh								
num procs	SAAF						First order solver	
	no preconditioning		block-diagonal preconditioning				iters	time
			PCG		KLU (serial)			
	iters	time	iters	time	iters	time		
1	20,268		23	1,821.	6	113.	6	56.9
4	"	11,340.	"	866.	"	55.4	"	17.8

3.3 Scattering Order, Extended-Transport Correction and Scattering Ratio

The convergence rate of the first-order algorithm is strongly affected by the scattering ratio, i.e. the ratio of the scattering cross section with the total cross section. The convergence rate of the second-order algorithm, however, depends on the condition number of the system matrix and is insensitive to the scattering ratio. It is for this reason that the second-order algorithm performs well for electron transport problems, where the scattering ratio is large.

Table V shows the solver times for the first- and second-order algorithms for the model coaxial cable problem, for the quadratic-triangular (tri6) mesh, S_{16} angular quadrature, 32 processors, with convergence tolerance of 10^{-8}. Since the electron scattering is highly forward peaked, the Legendre-polynomial approximation of the angular dependence of the electron cross sections converge slowly, so that using an extended transport correction, or δ-function correction, of the cross sections improves accuracy of the angular dependence of the cross sections [10]. In this approach, a δ-function scattering term with magnitude equal to the scattering cross section moment with order one greater than the Legendre order of the cross section expansion, is subtracted from the total cross section and the scattering moments.

$$\sigma_t^{ETC} = \sigma_t - \sigma_{s,L+1} \tag{1a}$$

$$\sigma_{s,l}^{ETC} = \sigma_{s,l} - \sigma_{s,L+1} \tag{1b}$$

The extended transport correction reduces the effective scattering ratio, resulting in faster convergence of the first-order algorithm. The order of the Legendre expansion needed for a given application depends on accuracy considerations, but P_7 is not an unreasonable scattering order to consider. If the Legendre order is too large, unreasonably large angular quadrature order would be needed. The timings for the transport without the extended transport correction gives upper limits on the run times for large Legendre expansion orders.

Table V. Effect of Scattering Ratio on Solver Time for 200-keV Electron Transport

32 processors, S_{16} angular quadrature, tol=10^{-8}, quadratic-triangular (tri6) mesh						
P_n order	**Scattering ratio**				**Solver time (s)**	
	iron	copper	silver	teflon	SAAF	first order solver
With extended transport correction						
1	0.191	0.209	0.305	0.0662	96.5	25.5
3	0.394	0.420	0.536	0.169	96.2	52.3
7	0.643	0.666	0.753	0.372	97.1	193.4
Without extended transport correction						
1	0.924	0.930	0.949	0.824	139.	398.
3	"	"	"	"	131.	530.
7	"	"	"	"	117.	973.

3.2 Parallel Performance

Parallel performance results are shown in Tables VIa and VIb for photon and electron transport, respectively. 8th order angular quadrature and 3rd order Legendre cross section expansion were used in the photon-transport calculations, with convergence tolerance of 10^{-8}. Two measures of parallel performance are given in the tables: 1) a relative performance, which is parallel

efficiency relative to that of the previous number of processors, and 2) a cumulative performance, which is the parallel efficiency relative to the serial run. 12^{th} order angular quadrature and 7^{th} order Legendre cross section expansion were used in the photon-transport calculations, with convergence tolerance of 10^{-8}.

Table VIa. Parallel Performance for 200-keV Photon Transport

S_8 P_3, tol=10^{-8}, quadratic-triangular (tri6) mesh						
num procs	SAAF (preconditioned)			First order solver		
	solver time (s)	parallel performance		solver time (s)	parallel performance	
		rel	cum		rel	cum
1	1,846	-	1	56.9	-	1
2	1,703.	0.54	0.54	31.5	0.90	0.90
4	900.	0.95	0.51	17.8	0.88	0.80
8	446.	1.0	0.52	9.50	0.94	0.75
16	216.	1.0	0.53	5.15	0.92	0.69
32	112.	0.96	0.52	2.81	0.92	0.63

Table VIb. Parallel Performance for 200-keV Electron Transport

S_{12} P_7, tol=10^{-8}, linear-triangular (tri3) mesh						
num procs	SAAF			First order solver		
	solver time (s)	parallel performance		solver time (s)	parallel performance	
		rel	cum		rel	cum
1	72.3	-	1	1,073	-	1
2	40.4	0.89	0.89	660.	0.81	0.81
4	20.7	0.98	0.87	377.	0.88	0.71
8	10.7	0.97	0.84	197.	0.96	0.68
16	5.65	0.95	0.80	108	0.91	0.62
32	3.06	0.92	0.74	52.3	1.0	0.64

3.4 Angular Quadrature Order

The first-order solver time scales linearly with number of directions. The dependence of the second-order solver time on number of directions is more complicated. The full linear solver time scales as the number of directions squared, while the preconditioner system solver time scales linearly with the number of directions, so the effective dependence will be somewhere between linear and quadratic, in practice. p is the observed order of the dependence of the solver time on the number of S_n directions.

As expected, the observed p for the first-order algorithm is nearly 1. For the unpreconditioned second-order algorithm, the observed p is nearly 2, while for the photon calculation include

block diagonal preconditioning with the sparse direct solve of the preconditioner system, p is nearly 1, and for the block-diagonal preconditioner with PCG solve, p is between 1 and 2.

Table VIIa. 200-keV Photon Transport

32 procs, P_3 Scattering, tol=10^{-8}, quadratic-triangular (tri6) mesh										
S_n order	num dirs	SAAF (block-diagonal preconditioner)						First order solver		
		PCG			KLU (serial)					
		iters	time	p	iters	time (s)	p	iters	time (s)	p
4	12	23	20.8		6	13.2		6	0.92	
8	40	"	111.	1.39	"	45.1	1.02	"	3.1	1.01
12	84	"	261.	1.15	"	99.7	1.07	"	6.4	0.98
16	144	"	449.	1.01	"	out of memory		"	10.8	0.97

Table VIIb. 200-keV Electron Transport

32 procs, P_3 Scattering, tol=10^{-8}, linear-triangular (tri3) refined mesh							
S_n order	num dirs	2nd order solver			First order solver		
		iters	Solver time (s)	p	iters	Solver time (s)	p
4	12	60	0.59		26	9.45	
8	40	55	3.97	1.58	26	34.4	1.07
12	84	55	16.0	1.88	26	79.0	1.12
16	144	55	48.6	2.06	26	138.	1.03

4. SUMMARY

This article has presented preliminary comparisons of the performance of the first-order sweeps-base algorithm and the second-order PCG algorithm in the SCEPTRE project. Generally, the first-order algorithm performs better for photon transport, due to the small scattering ratio of photon cross sections, and the second-order algorithm performs better for electron transport, due to the efficiency of the PCG algorithm. More work is needed to incorporate acceleration methods into the first-order algorithm, and to develop preconditioning methods that work well for second-order algorithm for electron transport. Some conclusions that may be drawn from this study include: 1) the first-order algorithm is efficient for higher-order finite-element basis functions, while the second-order performs better for linear basis functions, 2) both algorithms scale well in parallel, 3) the first-order algorithm depends strongly on the order of Legendre expansion of the cross sections for electron transport, while the second-order algorithm is insensitive to it, 4) the first-order algorithm scales linearly with the number of S_n directions, while the unpreconditioned second-order algorithm scales with the number of directions squared.

REFERENCES

1. S. D. Pautz, "An Algorithm for Parallel Sn Sweeps on Unstructured Meshes," *Nucl. Sci. Eng.*, **140**, pp. 111-136 (2002).
2. C.R. Drumm and J. Lorenz, "Parallel FE Approximation of the Even/Odd-Parity Form of the Linear Boltzmann Equation," *Math. Comp. Modeling*, **31**, pp. 55 (2000).
3. J. J. Duderstadt and W. R. Martin, *Transport Theory*, Wiley-Interscience, New York, USA (1979).
4. J. E. Morel and J. M. McGhee, "A Self-Adjoint Angular Flux Equation," *Nucl. Sci. Eng.*, **132**, pp. 312-325 (1999).
5. B. Jiang, *The Least-Squares Finite Element Method*, Springer-Verlag, Berlin, Germany (1998).
6. Michael Heroux, et al., "An Overview of Trilinos," SAND2003-2927, Sandia National Laboratories report, Albuquerque, NM (2003).
7. L. J. Lorence, Jr., J. E. Morel, and G. D. Valdez, "Physics Guide to CEPXS: A Multigroup Coupled Electron-Photon Cross-Section Generating Code Version 1.0," SAND89-1685, Sandia National Laboratories report, Albuquerque, NM (1989).
8. C. R. Drumm and W. C. Fan, "Uncollided-Flux Preconditioning of the Conjugate Gradients Solution of the Transport Equation," *Proc. Int. Conf. Nuclear Mathematical and Computational Sciences*, Gatlinburg, TN, April 6-10 (2003).
9. T. A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, USA (2006).
10. J. E. Morel, "On the Validity of the Extended Transport Cross-Section Correction for Low-Energy Electron Transport," ," *Nucl. Sci. Eng.*, **71**, pp. 64-71 (1979).