M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

On The Acceleration Of The Source Iteration Method

Daniella Machado Schulz[a], Rudnei Dias da Cunha[b], Liliane Basso Barichello[b]

[a]*Postgraduate Program in Applied Mathematics, Mathematics and Statistics Institute, Universidade Federal do Rio Grande do Sul, Brazil*

[b]*Mathematics and Statistics Institute, Universidade Federal do Rio Grande do Sul, Brazil*

dani.schulz88@gmail.com, rudnei.cunha@ufrgs.br, lbaric@mat.ufrgs.br

Abstract - *We present a comparison on the performance exhibited by acceleration techniques of Source Iteration (SI), namely the Diffusion Synthetic Acceleration (DSA), the Epsilon-k algorithm, the Coarse-Mesh Finite Difference (CMFD). We also propose a new acceleration based on the use of the Analytical Discrete Ordinates (ADO) method to provide a better initial guess of the scalar and angular fluxes to SI.*
The results show that DSA or CMFD are the schemes of choice to accelerate numerical solutions of the typical transport problems we have chosen to present in this work.

I. INTRODUCTION

Solving analytically the neutron transport equation is often a hard task, hence a number of iterative schemes were developed in order to solve the problem approximately. Unfortunately, these methods may not converge fast enough for problems of practical interest. One of the most classical and fundamental iterative schemes proposed for solving transport problems is the Source Iteration (SI) method; however, as reported by Adams and Larsen [1], when the scattering ratio is close to one and particularly for the case of thick media, the convergence of the SI method may be very slow. Among the acceleration schemes available to increase the rate of convergence of SI, one of the most used is the Diffusion Synthetic Acceleration (DSA) scheme [2, 3, 4, 5]. In recent years, a classic nonlinear acceleration scheme of numerical iterative methods, Wynn's Epsilon algorithm [6, 7, 8] has been used for the same purpose [9, 10]. Another acceleration scheme is the Coarse-Mesh Finite Difference (CMFD) which is a two-level angular-spatial multigrid, employing a fine and a coarse mesh, where a transport sweep is carried on the fine mesh and the diffusion part of the transport equation is applied on the coarse mesh [11, 12, 13, 14, 15].

We consider here the one-speed, slab-geometry S_N transport problems with linearly anisotropic scattering given by

$$\mu_n \frac{\mathrm{d}}{\mathrm{d}x}\psi_n(x) + \sigma_t \psi_n(x) = \frac{1}{2}\sigma_{s0}\sum_{k=1}^{N}\psi_k(x)\omega_k + \frac{3}{2}\mu_n\sigma_{s1}\sum_{k=1}^{N}\mu_k\psi_k(x)\omega_k + \frac{Q(x)}{2} \quad (1)$$

where $0 < x < X$ is the spatial variable, $(\mu_n, \omega_n), 1 \le n \le N$ is an even-order Gauss Legendre quadrature set and $\psi_n(x) = \psi(x, \mu_n)$ is the angular flux of particles in the direction μ_n; σ_t is the total macroscopic cross section; σ_{s0} and σ_{s1} are the coefficients of the expansion of the phase function in Legendre polynomials; and Q is the interior source. Boundary conditions for Equation (1) are set accordingly, including vacuum and/or reflexive conditions at one or both ends of the domain.

The spatial variable in Equation (1) is discretized on a computational grid consisting of $j = 1, \ldots, M$ cells of width h, with $x_{j-1/2} < x_j < x_{j+1/2}$, $\psi_{n,j} = \psi_n(x_j)$ indicating the angular flux evaluated in the middle of the cell. Then we integrate Equation (1) between $x_{j-1/2}$ and $x_{j+1/2}$ to yield

$$\frac{\mu_n}{h_j}\left(\psi_{n,j+1/2} - \psi_{n,j-1/2}\right) + \sigma_{t_j}\psi_{n,j} = \frac{1}{2}\sigma_{s0_j}\sum_{k=1}^{N}\psi_{k,j}\omega_k + \frac{3}{2}\mu_n\sigma_{s1_j}\sum_{k=1}^{N}\mu_k\psi_{k,j}\omega_k + \frac{Q_j}{2} \quad (2)$$

where $\psi_{n,j} = (1/h_j)\psi_n(x_j)$ and $\psi_{n,j\pm1/2} = \psi_n\left(x_{j\pm1/2}\right)$. Upon writing Equation (2) in each j cell and using the specified boundary conditions we arrive at a system of N equations with $2N$ $\psi_{n,j}$ variables, which are defined here with the usual diamond difference approximation,

$$\psi_{n,j} = \frac{1}{2}\left(\psi_{n,j+1/2} + \psi_{n,j-1/2}\right). \quad (3)$$

Equation (2) may be solved using the Source Iteration (SI) method [1]. An initial approximation to the angular flux, $\psi^{(0)}_{n,j}$ is given by its specification at the boundaries; there is also the need to provide an initial guess $\phi^{(0)}_j$ to the scalar flux and to the current, $J^{(0)}_j$ in the right-hand-side of Equation (2), these two usually given as zero. In an iteration l, this provides a half-step approximation $\psi^{(l+1/2)}_{n,j}$ and these values are used to obtain the new values of $\phi^{(l+1)}_j$. Therefore, Equation (2) may be rewritten as

$$\frac{\mu_n}{h_j}\left(\psi^{(l+1/2)}_{n,j+1/2} - \psi^{(l+1/2)}_{n,j-1/2}\right) + \sigma_{t_j}\psi^{(l+1/2)}_{n,j} = \frac{1}{2}\sigma_{s0_j}\phi^{(l+1/2)}_j + \frac{3}{2}\mu_n\sigma_{s1_j}J^{(l+1/2)}_j + \frac{Q_j}{2}, l = 0, 1, \ldots, l_{\max} \quad (4)$$

and the Diamond Difference (DD) auxiliary equation [16]

$$\psi^{(l+1/2)}_{n,j} = \frac{1}{2}\left(\psi^{(l+1/2)}_{n,j+1/2} + \psi^{(l+1/2)}_{n,j-1/2}\right) \quad (5)$$

where

$$\phi^{(l+1/2)}_j = \sum_{k=1}^{N}\psi^{(l+1/2)}_{k,j}\omega_k \quad (6)$$

and

$$J_j^{(l+1/2)} = \sum_{k=1}^{N} \mu_k \psi_{k,j}^{(l+1/2)} \omega_k. \tag{7}$$

with $1 \leq j \leq M$. If DSA or CMFD is used to accelerate the convergence of SI, then $\phi_j^{(l+1)}$ is obtained by adding a correction term to $\phi_j^{(l+1/2)}$; otherwise, we set $\phi_j^{(l+1)} = \phi_j^{(l+1/2)}$. This iterative step or transport sweep is repeated until the maximum relative error of the angular flux taken across the cells satisfies the relationship

$$\max_{1 \leq j \leq M} \left| \frac{\phi_n^{(l+1)} - \phi_n^{(l)}}{\phi_n^{(l+1)}} \right| \leq \epsilon \ll 1. \tag{8}$$

II. ACCELERATION TECHNIQUES

As noted in [1, p. 24ff], SI may have a rather slow convergence if the scattering ratio $c = \sigma_{s0}/\sigma_t \approx 1$. It is that result that encourages one to use acceleration techniques to speed-up SI. We have elected to compare the use of Wynn's Epsilon-k algorithm, DSA, CMFD and a new technique that we have developed that we call ADO+SI.

1. Epsilon-k algorithm

We apply the Epsilon-k algorithm [6, 7, 8, 9] over the sequence of approximations to the scalar flux $\phi_j^{(l)}$. The Epsilon-k algorithm (for even values of k) is a general, nonlinear extrapolation that may be applied to a sequence of values to obtain a faster convergence to the limit of that sequence. As mentioned in [17], its convergence and acceleration properties are difficult to ascertain due to its nonlinearity, though for totally monotonic or totally oscillating sequences there are two theorems that guarantee its convergence.

Based on Wynn's notation, after k transport sweeps, we have a vector-valued sequence (each vector with J elements) of $(k+1)$ values $\varepsilon_0^{(0)} = \phi^{(0)}$, $\varepsilon_0^{(1)} = \phi^{(1)}$, …, $\varepsilon_0^{(k)} = \phi^{(k)}$ (note that we have dropped the j subscript on ϕ as we refer here to all ϕ grid elements). We wish to obtain $\varepsilon_k^{(0)}$ that will be the accelerated value of $\phi_j^{(l)}$; using the Epsilon-k equations, we arrive at

$$\delta = \varepsilon_{i+1}^{(s-1)} - \varepsilon_i^{(s-1)} \tag{9}$$

$$\varepsilon_i^{(s)} = \begin{cases} \varepsilon_{i+1}^{(s-2)} + \delta^{-1}, & |\delta| > 0 \\ 0, & otherwise \end{cases} \tag{10}$$

where $0 \leq i \leq k - s$, $1 \leq s \leq k$ and we overwrite $\phi_j^{(l)}$ with $\varepsilon_k^{(0)}$ after computing Equations (9)-(10). The Epsilon-k algorithm is applied to each $\phi_j^{(l)}$, each cell in turn. To aid in this, we store the ε_k values on a Fortran 95 array EPSILON of $k+1$ rows and $k+2$ columns (i.e. the array has $k^2 + 3k + 2$ elements in memory) with indices running from 0 to k and -1 to k respectively. Using a Fortran 95 notation, we initialize EPSILON$(0:k, -1)$ with 0 (once, outside the main SI iteration) and EPSILON$(0:k, 0)$ with a specific $\phi_j^{(l)}$. Then, we traverse this array with indices i for the rows and s for the columns (as defined above) in order to produce the required ε_k applying Equations (9)-(10). The desired $\varepsilon_k^{(0)}$ will be found in element $(0, k)$ of the array EPSILON. We note that this traversal is a triangular traversal of the array from column 0 towards column k, from rows k to 0; therefore there are $(k^2 + k)/2$ unused array elements. For the sake of speed of computation we do not use a compressed storage of the array (which would eliminate the unused elements and reduce the amount of memory required to store the array in memory); this penalty is acceptable for the maximum value of k we have used, 16, which leads to 306 elements in memory or, equivalently, 2.390625 kilobytes of memory using double precision (64 bits).

2. Diffusion Synthetic Acceleration

Another approach to reduce the number of iterations of SI is to use DSA [18, 19], [1, p. 52ff]. After each transport sweep using Equation (4) we obtain an approximation $\psi_{n,j}^{(l+1/2)}$. With these values at hand we write a related S_N problem in terms of the discrete correction

$$f_{n,j(\pm 1/2)} = \psi_{n,j(\pm 1/2)} - \psi_{n,j(\pm 1/2)}^{(l+1/2)}. \tag{11}$$

Equation (4) may be rewritten as

$$\frac{\mu_n}{h_j}\left(f_{n,j+1/2} - f_{n,j-1/2}\right) + \sigma_{t_j} f_{n,j} - \frac{1}{2}\sigma_{s0_j} \sum_{n=1}^{N} f_{n,j}\omega_n - \frac{3}{2}\mu_n \sigma_{s1_j} \sum_{n=1}^{N} \mu_n f_{n,j} \omega_n = \frac{1}{2}\sigma_{s0_j}\left(\phi_j^{(l+1/2)} - \phi_j^{(l)}\right) + \frac{3}{2}\mu_n \sigma_{s1_j}\left(J_j^{(l+1/2)} - J_j^{(l)}\right) \tag{12}$$

where

$$f_{n,j} = \frac{1}{2}\left(f_{n,j+1/2} - f_{n,j-1/2}\right) \tag{13}$$

Applying a P_1 approximation to Equations (12)-(13), we arrive at a tridiagonal system of $M+1$ linear equations on $F_j^{(l+1)}$ derived from the governing equation

$$-\frac{1}{3\sigma_{tr_{j+1}} h_{j+1}}(F_{j+3/2}^{(l+1)} - F_{j+1/2}^{(l+1)}) + \frac{1}{3\sigma_{tr_j} h_j}(F_{j+1/2}^{(l+1)} - F_{j-1/2}^{(l+1)}) + \frac{\sigma_{a_{j+1}} h_{j+1}}{4}(F_{j+3/2}^{(l+1)} + F_{j+1/2}^{(l+1)}) + \frac{\sigma_{a_j} h_j}{4}(F_{j+1/2}^{(l+1)} + F_{j-1/2}^{(l+1)}) = \frac{\sigma_{s0_{j+1}} h_{j+1}}{2}(\phi_{j+1}^{(l+1/2)} - \phi_{j+1}^{(l)}) + \frac{\sigma_{s0_j} h_j}{2}(\phi_j^{(l+1/2)} - \phi_j^{(l)}) \tag{14}$$

where $0 \leq j \leq M$, $\sigma_{a_j} = \sigma_{t_j} - \sigma_{s0_j}$, $\sigma_{tr_j} = \sigma_{t_j} - \sigma_{s1_j}$ and

$$F_j^{(l+1)} = \sum_{k=1}^{N} f_{k,j}\omega_k. \tag{15}$$

This system is solved using LAPACK's DGTSVX routine [20, p. 263ff] and then we update $\phi_j^{(l)}$ via

$$\phi_j^{(l+1)} = \phi_j^{(l+1/2)} + \frac{1}{2}\left(F_{j+1/2}^{(l+1)} + F_{j-1/2}^{(l+1)}\right). \tag{16}$$

Once $\phi_j^{(l+1)}$ have been computed on each cell, another iteration consisting of computing equations (4)-(7) and (14)-(16) is performed, until Equation (8) is satisfied.

3. Coarse-Mesh Finite Difference

As mentioned in [15], CMFD is similar to DSA in the sense that each high-order transport sweep is followed by the solution of a low-order diffusion problem. However CMFD is more akin to a traditional multigrid method used to solve an elliptic partial differential equation since that there are two easily distinguishable spatial grids, a fine one with M cells (as defined previously) and another, coarse grid with K cells, $K \leq M$. We will consider here for simplicity that $p = M/K$, the number of fine cells in a coarse cell, is constant but this may be generalized as given in [15]. Another two characteristics of CMFD are that its low-order solution (on the coarse grid) is a volume-averaged scalar flux and it is nonlinear. As analyzed in [13, 14, 15], CMFD may diverge if the coarse cells have widths more than 1 mean free path. Our implementation of CMFD follows that presented in [15] to which we refer the reader.

4. Hybrid analytical-iterative acceleration: ADO+SI

It is reasonable to expect that the SI iteration will converge faster if one has available a better approximation to the scalar flux than simply using, say, zero as an initial guess for it in the right-and-side of Equation (4). Recently, [21, pp. 180-181] has used such an approach to obtain via a synthetic diffusion equation such a better approximation for the initial scalar flux, and proceeding with the transport sweep thereafter.

We propose here the use of a new acceleration scheme that employs the Analytical Discrete Ordinates (ADO) method [22] and which has recently been extended to 2D domains [23]. This acceleration scheme consists of using ADO to obtain a initial (better) first estimate to the scalar and angular fluxes (and the current for linear anisotropic scattering problems) and then using these values into the SI iteration. Among the main advantages we seek using ADO is that we may use a smaller number of directions than that used in the SI iteration while still obtaining a good approximation to the scalar flux. Note that due to the ADO formulation [22] the use of N_{ADO} directions actually provide a solution as if using double the number of directions; we refer the reader to [24] for further details.

III. NUMERICAL EXPERIMENTS

We carried out a number of numerical experiments to compare the Epsilon-k, DSA, CMFD and ADO+SI accelerations to the SI method with diamond difference approximation, in terms of the number of iterations taken to achieve convergence (k^*); the wallclock execution time taken; and the maximum relative error (MRE) between the scalar flux obtained by SI and that by an acceleration scheme.

Our implementations were written in Fortran 95 using double-precision floating point arithmetic and were compiled under the GNU gfortran 5.4.0 with full optimization. The computer used has an Intel i5-6400T 2.20 GHz quad-core processor with 6 MB cache memory and 6 GB DDR3L-1600 SDRAM main memory. The execution times presented are the average of five different runs, timed with calls to a 1-millisecond resolution clock software function. To compare the performance exhibited by the accelerations, we define the speed-up as the ratio between the time taken for convergence of the acceleration technique and the time taken by SI. In our experiments we use the standard DD transport sweep in the SI iteration and the solution of linear systems of equations present in DSA, CMFD and ADO+SI is obtained using an optimized LAPACK compiled library.

The first experiment uses a homogeneous slab with $X = 100$ cm divided into $M = 2000$ cells, with $\sigma_t = 1.0$ cm^{-1}, $\sigma_{s0} = 0.995$ cm^{-1}, $Q = 0$ and isotropic unit incident flux at the boundaries, as given in [21, p. 181]. We have used $\epsilon = 10^{-6}$ in Equation (8) as the tolerance for convergence. As can be seen in Table I-a, the DSA acceleration provides the fastest solution, and the MRE values are all coherent with the tolerance used. It is possible to verify from the results that the time taken for convergence of SI, DSA and Epsilon-k are linearly dependent on N. Also note that the time per iteration of SI is smaller than that of DSA (approximately 0.69 times that of DSA for $N = 128$) since each SI iteration performs less floating-point iterations than each DSA iteration; however since the use of DSA provides the solution to the problem in a constant number of iterations for all values of N tested, the speed-up over SI increases for larger N. We note that the number of iterations required by SI and DSA to achieve convergence using the specified tolerance are in agreement with the theoretical upper bounds given in [1, p. 30].

If we look at the columns for CMFD for $p = 10$ and 20 in Table I-a we notice that it provides a solution with similar MREs to that presented by DSA and in this sense the solution has the same quality as that. Its execution time is, however, larger than DSA for all $N > 4$ but it is faster than the other two accelerations.

The Epsilon-k acceleration provided a faster solution than SI for almost all values of N (except for $N = 4$) but it was at least two orders of magnitude slower than DSA. It also provided MREs of a similar order of magnitude to those presented by DSA and CMFD.

As can be seen from the data on Table I-a, the ADO+SI acceleration (with $N_{ADO} = 4$) is at least twice as fast as the SI iteration while providing similar MRE values to the other accelerations presented. The ADO method for $N_{ADO} = 4$ needs 1.010(−03) seconds to compute and for the largest number of directions used ($N = 128$) this corresponds to less than 0.0001% of the execution time of ADO+SI.

In Figure 1 we show the effect of increasing the value of N_{ADO} on ADO+SI using the first experiment. Note that using $N_{ADO} = 4$ and $N_{ADO} = 8$ reduce the number of iterations compared to using $N_{ADO} = 2$. On the other hand, using either $N_{ADO} = 4$ or $N_{ADO} = 8$ produces almost the same number of iterations on the subsequent SI iteration and, since doubling N_{ADO} requires twice the amount of work on ADO, it is better to use the smallest value of N_{ADO}. The same behavior was found on the second and third experiments described below.

We note that our ADO+SI result for $N = 16$ compares favorably ($k^* = 823$) to that given in [21, p. 181, Table 1] ($k^* = 1097$), using their acceleration with DSA to provide the first initial guess; our ADO+SI provides a reduction of approximately 25% in the number of iterations. In terms of execution

a) Results for the first experiment.

	SI		DSA			Epsilon-8			Epsilon-16		
N	k^*	Time [s]	k^*	Time [s]	MRE	k^*	Time [s]	MRE	k^*	Time [s]	MRE
4	2158	2.00(−01)	9	5.20(−03)	1.87(−04)	381	4.40(−01)	1.95(−04)	38	9.06(−02)	2.60(−04)
8	2158	1.75(+00)	9	5.60(−03)	1.87(−04)	497	8.14(−01)	1.70(−04)	28	9.36(−02)	2.50(−04)
16	2158	3.46(+00)	9	6.60(−03)	1.87(−04)	180	5.06(−01)	1.72(−04)	92	5.30(−01)	1.84(−04)
32	2158	6.86(+00)	9	8.80(−03)	1.88(−04)	448	2.52(+00)	1.40(−04)	72	8.25(−01)	1.69(−04)
64	2158	13.39(+00)	9	1.40(−02)	1.88(−04)	219	2.39(+00)	2.14(−04)	222	4.67(+00)	2.04(−04)
128	2158	26.86(+00)	9	2.06(−02)	1.87(−04)	489	10.24(+00)	2.25(−04)	78	3.05(+00)	1.74(−04)

	SI		ADO+SI (N_{ADO} = 4)			CMFD (p = 10)			CMFD (p = 20)		
N	k^*	Time [s]	k^*	Time [s]	MRE	k^*	Time [s]	MRE	k^*	Time [s]	MRE
4	2158	2.00(−01)	980	1.12(−01)	6.79(−05)	14	2.60(−03)	1.87(−04)	24	3.40(−03)	1.88(−04)
8	2158	1.75(+00)	853	1.50(−01)	2.10(−04)	14	1.22(−02)	1.87(−04)	22	1.94(−02)	1.87(−04)
16	2158	3.46(+00)	823	2.34(−01)	2.71(−04)	15	2.56(−02)	1.87(−04)	22	3.62(−02)	1.87(−04)
32	2158	6.86(+00)	815	4.25(−01)	2.88(−04)	15	5.10(−02)	1.87(−04)	21	6.90(−02)	1.87(−04)
64	2158	13.39(+00)	813	7.94(−01)	2.93(−04)	15	9.30(−02)	1.87(−04)	21	1.30(−01)	1.87(−04)
128	2158	26.86(+00)	813	1.56(+00)	2.94(−04)	15	1.84(−01)	1.87(−04)	21	2.60(−01)	1.87(−04)

b) Results for the second experiment.

	SI		DSA			Epsilon-8			Epsilon-16		
N	k^*	Time [s]	k^*	Time [s]	MRE	k^*	Time [s]	MRE	k^*	Time [s]	MRE
64	42	6.00(−02)	13	2.50(−02)	1.57(−06)	4	3.78(−02)	1.67(−06)	2	4.18(−02)	9.92(−07)

	SI		ADO+SI (N_{ADO} = 4)			CMFD (p = 10)			CMFD (p = 20)		
N	k^*	Time [s]	k^*	Time [s]	MRE	k^*	Time [s]	MRE	k^*	Time [s]	MRE
64	42	6.00(−02)	35	5.54(−02)	1.91(−07)	13	1.74(−02)	1.53(−06)	13	1.70(−02)	1.51(−06)

c) Results for the third experiment.

	SI		DSA			Epsilon-8			Epsilon-16		
N	k^*	Time [s]	k^*	Time [s]	MRE	k^*	Time [s]	MRE	k^*	Time [s]	MRE
128	489	2.11(+00)	22	8.50(−02)	2.94(−05)	24	4.54(−01)	3.35(−05)	9	3.51(−01)	1.94(−05)

	SI		ADO+SI (N_{ADO} = 4)			CMFD (p = 10)			CMFD (p = 20)		
N	k^*	Time [s]	k^*	Time [s]	MRE	k^*	Time [s]	MRE	k^*	Time [s]	MRE
128	489	2.11(+00)	439	1.19(+00)	3.91(−07)	43	1.00(−01)	2.96(−05)	31	7.00(−02)	3.00(−05)

TABLE I. Results for the three experiments.

time, our implementation of their acceleration scheme needs 1.9083 seconds whereas the ADO+SI implementation obtains the solution in 0.2340 seconds (cf. Table I-a).

We present in Figure 2 the speed-ups between SI and the acceleration schemes (the second figure is a zoom of the first to better show the speed-up curves for the Epsilon-k algorithm and ADO+SI). They clearly show how much faster is DSA over SI; moreover, it can be seen that as N grows, DSA provides the solution in even less time. This is a consequence of the fact that the diffusion solve becomes insignificant in expense when compared to SI. On the other hand, Epsilon-8 and Epsilon-16 show an oscillating behavior. Also note that ADO+SI is consistently better than SI, by a factor of approximately 2 (see Table I-a).

The second experiment considers a linearly anisotropic, heterogeneous test problem given in [21, p. 182], with a slab of length $X = 30$ cm divided into five regions of different lengths, number of cells and total and scattering macroscopic cross sections and sources, as in Table II. The number of discrete

Region	Length	M	σ_t	σ_{s0}	σ_{s1}	$Q(x)$
1	8	800	0.5	0.2	0.1	0
2	5	200	1.2	0.8	0.2	0
3	10	1000	0.9	0.6	0.0	1
4	5	500	0.5	0.2	0.1	0
5	2	200	1.2	0.8	0.2	0

TABLE II. Description of the regions used in the second experiment.

directions used was $N = 64$ and vacuum boundary conditions were imposed on the boundaries at $x = 0$ cm and $x = 30$ cm. We have also used $\epsilon = 10^{-6}$ in Equation (8) as the tolerance for convergence. Table I-b shows the results obtained. In this problem, the MRE values are much better, being at least of the same order as ϵ. The ADO+SI performs better than SI but in here we have a much better performance by the Epsilon-k acceleration which we attribute to the fact that the ratios of σ_{s0} and σ_{s1} to σ_t are not close to 1. In this experiment, we have CMFD being the fastest acceleration and providing MREs of similar order than DSA.

The third experiment considers a linearly anisotropic, heterogeneous test problem given in [25] with a slab of length $X = 100$ cm divided into five regions of different lengths, number of cells and total and scattering cross sections and sources, as given in Table III. The number of discrete directions used

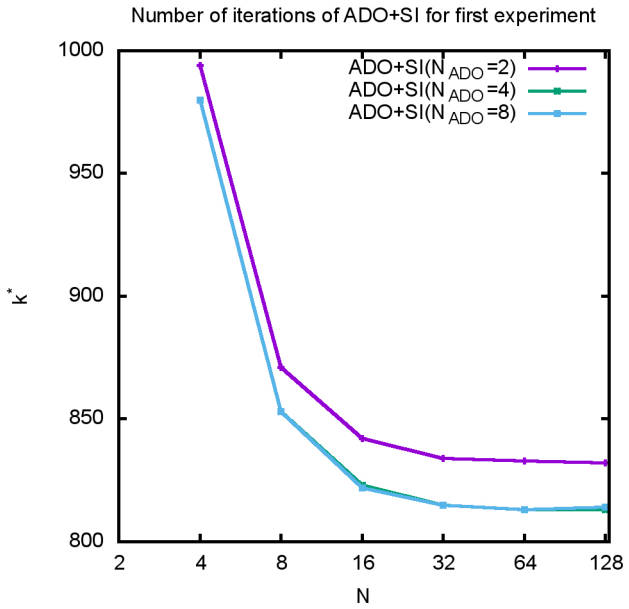

Fig. 1. Effect of increasing the value of N_{ADO} on ADO+SI.

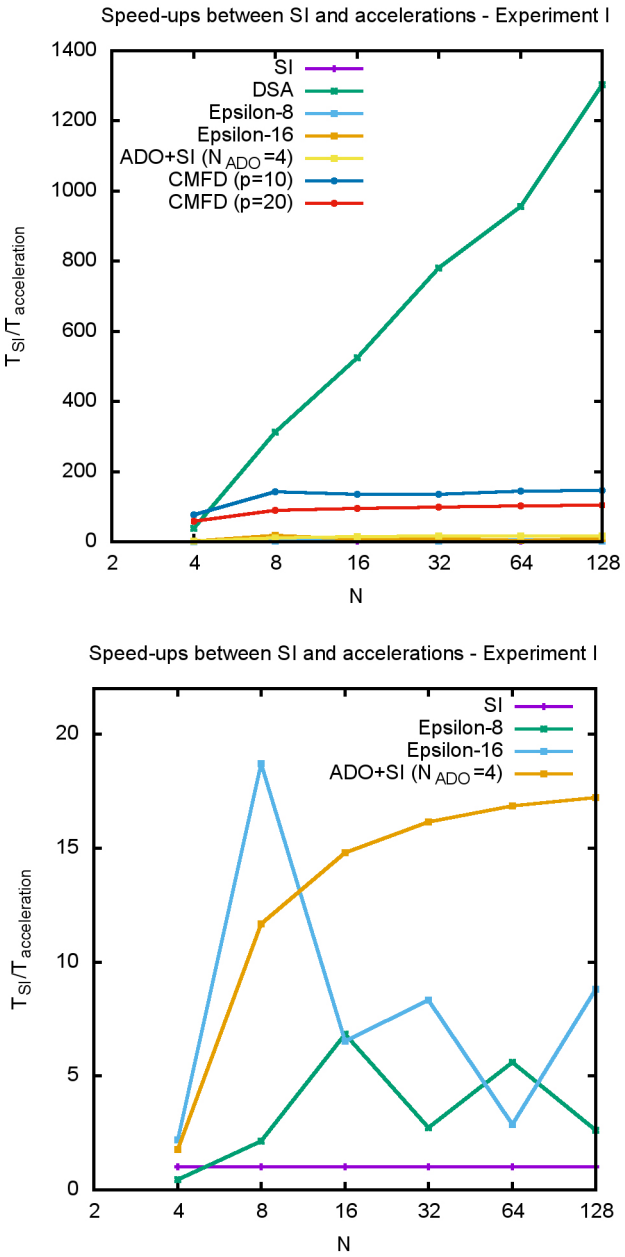

Fig. 2. Speed-ups between SI and the acceleration schemes.

Region	Length	M	σ_t	σ_{s0}	σ_{s1}	$Q(x)$
1	30	1500	1.0	0.97	0.32	0
2	5	250	0.9	0.8	0.267	2
3	22	1100	0.95	0.9	0.3	0
4	3	150	0.8	0.7	0.233	4
5	40	2000	1.0	0.97	0.32	0

TABLE III. Description of the regions used in the third experiment.

was $N = 64$ and vacuum boundary conditions were imposed on the boundaries at $x = 0$ cm and $x = 100$ cm. Again we have used $\epsilon = 10^{-6}$ as the tolerance for convergence. As can be seen in Table I-c, CMFD is again the fastest of the acceleration schemes, being over 30 times faster than SI. In this experiment, we have a similar behavior presented by the Epsilon-k and the ADO+SI accelerations. The MRE values are again acceptable within the tolerance used.

IV. CONCLUDING REMARKS

We have presented results obtained with the Diffusion Synthetic Acceleration, Wynn's Epsilon-k algorithm, Coarse-Mesh Finite Difference and a new hybrid analytical-numerical scheme (ADO+SI) to accelerate the rate of convergence of Source Iteration on three different problems. The results confirm the superior performance of DSA and CMFD over the Epsilon-k algorithm and ADO+SI as acceleration techniques in terms of execution time. We note that in some cases an acceleration technique provides a smaller MRE value, being either faster or slower than others.

We intend to continue this study analysing other acceleration techniques applied to SI in order to make other comparisons with regards to different problems, including the effects of the values of σ_t, σ_{s0} and σ_{s1} on the overall convergence of the methods, as well as with respect to the increase in the number of cells used in the discretization and the number of discrete directions used. We also intend to expand this study by considering two-dimensional spatial geometries and how these techniques would then perform. These comparisons will aid in the study of new acceleration techniques applied to SI which are currently under investigation.

ACKNOWLEDGMENTS

The authors acknowledge the financial support from the National Council for Research and Development (Brazil).

REFERENCES

1. M. L. ADAMS and E. W. LARSEN, "Fast iterative methods for discrete-ordinates particle transport calculations," *Progress in Nuclear Energy*, **40**, *1*, 3–159 (2002).
2. E. W. LARSEN, "Unconditionally Stable Diffusion Synthetic Acceleration Methods for the Slab Geometry Discrete-Ordinates Equations. Part I: Theory." *Nuclear Science and Engineering*, **82**, 47–63 (1982).
3. F. P. SANTOS, H. A. FILHO, and R. C. BARROS, "Coarse-mesh diffusion synthetic acceleration of the scattering source iteration scheme for one-speed slab-geometry discrete ordinates problems," *AIP Conference Proceedings*, **1558**, *1*, 2398–2401 (2013).
4. D. A. KNOLL, H. PARK, and K. SMITH, "Application of the Jacobian-Free Newton-Krylov Method to Nonlinear Acceleration of Transport Source Iteration in Slab Geometry," *Nuclear Science and Engineering*, **167**, 122–132 (2011).
5. L. QIN and L. WANG, "Implicit Asymptotic Preserving Method for Linear Transport Equations," https://arxiv.org/abs/1602.00746 (2016).
6. P. WYNN, "On a Device for Computing the $e_m(S_n)$ Transformation," *Mathematical Tables and Aids of Computation*, **10**, 91–96 (1956).
7. P. WYNN, "The epsilon algorithm and operational formulas of numerical analysis," *Mathematics of Computation*, **15**, 151–158 (1961).
8. P. R. GRAVES-MORRIS, D. E. ROBERTS, and A. SALAM, "The epsilon algorithm and related topics," *Journal of Computational and Applied Mathematics*, **122**, 51–80 (2000).
9. B. D. GANAPOL, *Analytical Benchmarks for Nuclear Engineering Applications – Case Studies in Neutron Transport Theory*, OECD Publications, Paris (2008).
10. P. PICCA, R. FURFARO, and B. GANAPOL, "Application of non-linear extrapolations for the convergence acceleration of source iteration," *Journal of Computational and Theoretical Transport*, **45**, *5*, 351–367 (2016).
11. K. S. SMITH, "Nodal Method Storage Reduction by Nonlinear Iteration," *Trans. Am. Nucl. Soc.*, **44**, *265* (1983).
12. Y. A. CHAO, "Coarse Mesh Finite Difference Methods and Applications," in "Proc. PHYSOR," American Nuclear Society (May 2000).
13. N. Z. CHO and C. J. PARK, " A Comparison of Coarse Mesh Rebalance (CMR) and Coarse Mesh Finite Difference (CMFD) Acceleration Methods for the Neutron Transport Calculations," Tech. Rep. NURAPT-2002-02, Nuclear Reactor Analysis and Particle Transport Laboratory, Korea Advanced Institute of Science and Technology (2003).
14. E. MASIELLO, "Analytical stability analysis of Coarse-Mesh Finite Difference method," in "Proc. PHYSOR," American Nuclear Society, Interlaken, Switzerland (September 2008).
15. E. W. LARSEN and B. W. KELLEY, "CMFD and Coarse-Mesh DSA," in "Proc. PHYSOR," American Nuclear Society, Knoxwille, Tennessee, USA (April 2012).
16. B. G. CARLSON and K. D. LATHROP, *Transport theory - the method of discrete ordinates*, Gordon & Breach, New York, In: H. Greenspan, C. N. Kelber, D. Okrent (Eds.) - Computing methods in reactor physics (1968).
17. C. BREZINSKI, "Convergence acceleration methods: The past decade," *Journal of Computational and Applied Mathematics*, **12&13**, 19–36 (1985).
18. R. E. AUCOUFFE, "A stable diffusion synthetic acceleration method for neutron transport equations," *Transactions of the American Nuclear Society*, **23**, 203 (1976).
19. R. E. AUCOUFFE, "The diffusion synthetic acceleration method applied to two-dimensional neutron transport

problems," *Transactions of the American Nuclear Society*, **27**, 346 (1977).

20. E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, and D. SORENSEN, *LAPACK Users' Guide*, SIAM, Philadelphia, third ed. (1999).
21. R. S. MANSUR, F. P. SANTOS, H. ALVES FILHO, and R. C. BARROS, "Diffusion synthetic methods for computational modeling of one-speed slab-geometry transport problems with linearly anisotropic scattering," *Progress in Nuclear Energy*, **73**, 179–187 (2014).
22. L. B. BARICHELLO and C. E. SIEWERT, "A discrete-ordinates solution for a non-grey model with complete frequency redistribution," *Journal of Quantitative Spectroscopy and Radiative Transfer*, **62**, 665–675 (1999).
23. C. B. PICOLOTO, A. TRES, R. D. DA CUNHA, and L. B. BARICHELLO, "Closed-form solutions for nodal formulations of two dimensional transport problems in heterogeneous media," *Annals of Nuclear Energy*, **86**, 65–71 (2015).
24. L. B. BARICHELLO, *Explicit Formulations for Radiative Transfer Problems*, CRC Press, Boca Raton, pp. 541–562, In: H. R. B. Orlande, O. Fudyin, D. Maillet, R. M. Cotta (Org.) - Thermal Measurements and Inverse Techniques (2011).
25. C. B. PAZINATTO, R. C. BARROS, and L. B. BARICHELLO, "Analytical adjoint discrete ordinates formulation for monoenergetic slab-geometry source-detector calculations," *Nuclear Energy Science and Tecnology*, **10**, *2*, 107–122 (2016).