

Piecewise Diffusion Synthetic Acceleration scheme for neutron transport simulations in optically thick systems

François Févotte^{a,*}

^a*EDF Lab Paris-Saclay – 7, Boulevard Gaspard Monge – 91120 Palaiseau, France*

Abstract

The method of discrete ordinates (S_N) is a popular choice for the solution of the neutron transport equation. It is however well known that it suffers from slow convergence of the scattering source in optically thick and diffusive media, such as pressurized water nuclear reactors (PWR). In reactor physics applications, the S_N method is thus often accompanied by an acceleration algorithm, such as Diffusion Synthetic Acceleration (DSA). With the recent increase in computational power, whole core transport calculations have become a reasonable objective. It however requires using large computers and parallelizing the transport solver. Due to the elliptic nature of the DSA operator, its parallelization is not straightforward. In this paper, we present an acceleration operator derived from DSA, but defined in a piecewise way such that its parallel implementation is straightforward. We mathematically show that, for optically thick enough media, this Piecewise Diffusion Synthetic Acceleration (PDSA) scheme preserves the good properties of DSA. This conclusion is supported by numerical experiments.

Keywords: DSA, Diffusion Synthetic Acceleration, Parallelization, Fourier analysis

1. Introduction

The simulation of neutron transport phenomena in nuclear reactor cores requires the solution of the Boltzmann Transport Equation (BTE). We focus in this paper on the simulation of Pressurized Water Reactors (PWR). For such reactors, the geometry and materials used make the domain optically thick and diffusive, meaning *(i)* that the core size represents a large number of neutron mean free paths, and *(ii)* that scattering represents a large fraction of neutron-matter interactions. In such cases, the diffusion equation is often considered a good enough alternative to the BTE, which is why most industrial calculations rely on a 2-step scheme:

*Corresponding author

Email address: francois.fevotte@edf.fr (François Févotte)

1. the BTE is solved in 2D, at the scale of an assembly, with relatively fine spatial and energetic discretization. This calculation produces homogenized and condensed cross-sections;
2. these homogenized and condensed cross-sections are fed to a 3D diffusion or Simplified Transport (SP_N) calculation, which is performed at the scale of the reactor core and uses a relatively coarse spatial and energetic discretization.

Such a scheme has the advantage of involving neutron transport calculations only at the scale of the fuel assembly and for two spatial dimensions. In such calculations, energy is traditionally discretized using the multigroup formalism, and the angular variable is handled by the discrete ordinates (S_N) method. Various methods can be used to discretize the spatial operators, but we will not consider such details in this paper. As was uncovered by early adopters of the S_N formalism, this method suffers from a major problem in optically thick diffusive media: the classical Source Iteration (SI) scheme converges very slowly in this case. To remedy this issue, the Diffusion Synthetic Acceleration (DSA) scheme has been proposed as early as the late 1970s [1, 2, 3], and remains one of the most popular acceleration schemes today, especially for Cartesian geometries.

However, the approximations induced by the use of such 2-step schemes need to be assessed, which is why full core 3D neutron transport solvers are still needed. We focus here on the solution of the 3D stationary BTE, which is one of the most important building blocks for state-of-the-art 3D whole-core criticality calculations. Even though it ignores the time variable, the 3D stationary BTE is still set in a 6-dimensional phase space (3 for space, 2 for travel direction and 1 for energy). Its discretization at the scale of the full reactor core therefore quickly produces very large problems of size on the order of 10^{10} to 10^{12} degrees of freedom, whose solution has remained largely out of reach before the early 2010s [4, 5, 6], when big enough supercomputers became available, along with numerical methods able to efficiently harness them.

Devising and implementing parallel methods able to efficiently solve the transport equation for such large problems is in itself no easy task, the major difficulty lying in the fact that the hyperbolic nature of the transport equations implies dependencies between cells. However, another practical difficulty arises, in the case of optically thick geometries, from the need for an acceleration scheme that *(i)* accelerates the Source Iteration scheme, and *(ii)* can be efficiently parallelized using the same data distribution as the transport solver.

A first technique consists in keeping the traditional DSA scheme, and parallelizing it alongside the transport solver. This presents the advantage of reusing the same whole-core diffusion solvers as the second step mentioned above. However, industry-grade neutron diffusion solvers are generally sequential, and the elliptic nature of the diffusion equation makes their parallelization a challenging task. Although efficient parallel diffusion solvers can be implemented [7, 8], the induced code complexity is often considered a heavy price to pay. The same is also true in the case of alternate acceleration methods such as Coarse-Mesh

Finite Differences [9], which are also elliptic in nature and thus difficult to parallelize.

Other techniques consist in departing from the standard Source Iteration & DSA scheme. For example, DENOVO uses a Krylov solver [4], which converges faster than the traditional multi-group Gauss-Seidel algorithm and angular Source Iteration and alleviates the need for an acceleration scheme. Such a Krylov solver can still be further preconditioned, for example using multi-grid methods in energy [10]. While very efficient, the implementation of such techniques makes the reference neutron transport code share few software components, or even algorithms, with the industrial diffusion code. This, once again, makes the development, maintenance and verification price heavy to pay for the industry.

In this paper, we introduce the Piecewise Diffusion Synthetic Acceleration scheme (PDSA), a new acceleration method for parallel neutron transport calculations, specifically designed to minimize the development effort and reuse as much code as possible from existing diffusion solvers. Indeed, the scheme is defined in such a way that any code implementing transport iterations accelerated by a DSA operator (with consistent spatial discretization schemes), can be transformed in a PDSA implementation at practically no programming cost. We will focus here on the definition of PDSA, and on the proof that it converges at the continuous level, along with simple 1D numerical experiments. We show in a companion paper [11] how this has been implemented in EDF's COCAGNE [12] platform, which features a diamond-difference S_N transport solver, accelerated by an SP_1 solver using mixed dual Raviart-Thomas (RT_k) finite elements [13, 6].

The remainder of this paper is organized as follows: in the following part, we briefly describe PDSA. We then proceed to a Fourier analysis in part 3: we review the standard unaccelerated transport Source Iteration, as well as DSA. Then, we Fourier analyze the proposed PDSA scheme. We show in particular how it can be seen as a perturbation of standard DSA, and derive conditions under which the perturbation is small enough that convergence properties of DSA are not lost. In part 4, we assess the validity of the theory by performing a few numerical experiments in 1D. We finally make a few concluding notes in part 5.

2. Description of the Piecewise Diffusion Synthetic Acceleration

In this section, we briefly describe and introduce the Piecewise Diffusion Synthetic Acceleration scheme. The focus here is on the definition of the scheme, while part 3.3 will be devoted to the analysis of its properties.

2.1. Standard DSA

As underlined in the introduction, PDSA is defined as a perturbation to the DSA scheme. We thus start with recalling the standard equations of DSA.

Let us consider the following time-independent, one-group neutron transport equation with isotropic scattering:

$$\forall \boldsymbol{\Omega} \in S^2, \forall \mathbf{r} \in \mathcal{D},$$

$$\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma(\mathbf{r}) \psi(\mathbf{r}, \boldsymbol{\Omega}) = Q(\mathbf{r}) + \frac{\Sigma_s(\mathbf{r})}{4\pi} \int_{S_2} d\boldsymbol{\Omega}' \psi(\mathbf{r}, \boldsymbol{\Omega}'), \quad (1)$$

with void boundary conditions to model a full core:

$$\forall \mathbf{r} \in \partial\mathcal{D}, \forall \boldsymbol{\Omega} \in S^2 \text{ such that } \boldsymbol{\Omega} \cdot \mathbf{n}(\mathbf{r}) < 0,$$

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = 0. \quad (2)$$

In the equation above, $\psi(\mathbf{r}, \boldsymbol{\Omega})$ denotes the neutron flux at position \mathbf{r} and in direction $\boldsymbol{\Omega}$. The total and scattering cross-sections are denoted by Σ and Σ_s respectively, and Q is a source term coming from outer iterations. The spatial domain is denoted by \mathcal{D} , and its boundary by $\partial\mathcal{D}$. The normal vector to this boundary is \mathbf{n} , so that the boundary condition above states that no flux enters the domain.

In this context, traditional DSA is defined as follows. In a first stage, the streaming operator is inverted:

$$\boldsymbol{\Omega} \cdot \nabla \psi_{\ell+\frac{1}{2}}(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma(\mathbf{r}) \psi_{\ell+\frac{1}{2}}(\mathbf{r}, \boldsymbol{\Omega}) = Q(\mathbf{r}) + \Sigma_s(\mathbf{r}) \phi_\ell(\mathbf{r}), \quad (3)$$

where inner iteration index ℓ was introduced, along with the scalar flux

$$\phi_\ell(\mathbf{r}) = \frac{1}{4\pi} \int_{S_2} d\boldsymbol{\Omega} \psi_\ell(\mathbf{r}, \boldsymbol{\Omega}).$$

In a second stage, an approximate diffusion operator is solved

$$\operatorname{div} \left(\frac{1}{3\Sigma} \nabla \tilde{\phi}_{\ell+1} \right) + \Sigma \tilde{\phi}_{\ell+1} = \Sigma_s \left(\phi_{\ell+\frac{1}{2}} - \phi_\ell \right). \quad (4)$$

While the original void boundary conditions (2) can be retained for transport equation (3), they have no meaning for diffusion equation (4) whose unknown is a scalar flux. They are thus replaced with a boundary condition operating on ϕ , which can be either of Robin or Dirichlet type:

$$\tilde{\phi}(\mathbf{r}) + \frac{2}{3\Sigma_t(\mathbf{r})} \mathbf{n}(\mathbf{r}) \cdot \nabla \tilde{\phi}(\mathbf{r}) = 0, \quad \forall \mathbf{r} \in \partial\mathcal{D}, \quad (5a)$$

or

$$\tilde{\phi}(\mathbf{r}) = 0, \quad \forall \mathbf{r} \in \partial\mathcal{D}, \quad (5b)$$

where $\mathbf{n}(\mathbf{r})$ denotes the vector normal to boundary at position $\mathbf{r} \in \partial\mathcal{D}$. In the latter case of Dirichlet boundary conditions, the position of the boundary might be artificially shifted by an extrapolation length.

At the end of a Diffusion Synthetic Accelerated iteration, the scalar flux is updated as follows:

$$\phi_{\ell+1} = \phi_{\ell+\frac{1}{2}} + \tilde{\phi}_{\ell+1}.$$

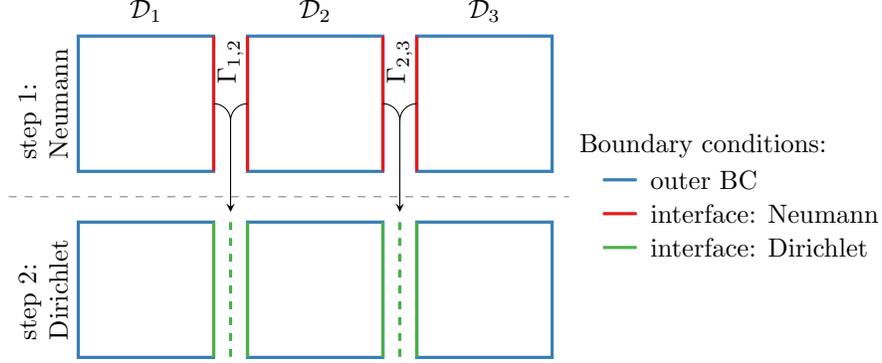


Figure 1: Schematic presentation of the PDSA scheme in a 3-subdomain case

2.2. Piecewise DSA

The PDSA scheme described in this paper aims at replacing system (4)–(5) by an operator which is more local and easier to solve in parallel. Figure 1 illustrates the construction of PDSA on domain \mathcal{D} , which has been partitioned into

$$\mathcal{D} = \cup_{i=1}^N \mathcal{D}_i,$$

with $N = 3$ in the figure. In the following, we temporarily drop iteration indices ℓ to simplify the notations.

In a first step, called the Neumann diffusion problem in the following, a flux correction $\tilde{\phi}_N^i$ is computed as the solution to equation (4) in each subdomain \mathcal{D}_i . Boundary condition (5) is considered for the outer boundary $\partial\mathcal{D} \cap \partial\mathcal{D}_i$. However, at inner interfaces between subdomains, an homogeneous Neumann boundary condition is used:

$$\nabla \tilde{\phi}_N^i(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) = 0, \quad \forall \mathbf{r} \in \Gamma_i = \partial\mathcal{D}_i \setminus \partial\mathcal{D}. \quad (6)$$

The second step, hereafter called the Dirichlet diffusion problem, differs from the first only with respect to the boundary conditions at the interface. A flux correction $\tilde{\phi}_D^i$ is computed as the solution to equation (4) in each subdomain, but in this case an inhomogeneous Dirichlet boundary condition is used at the interface between subdomains: for any two subdomains \mathcal{D}_i and \mathcal{D}_j sharing a common interface $\Gamma_{i,j}$,

$$\tilde{\phi}_D^i(\mathbf{r}) = \tilde{\phi}_D^j(\mathbf{r}) = \frac{1}{2} \left(\gamma_{\Gamma_{i,j}}(\tilde{\phi}_N^i) + \gamma_{\Gamma_{i,j}}(\tilde{\phi}_N^j) \right), \quad \forall \mathbf{r} \in \Gamma_{i,j}.$$

In the equation above, $\gamma_{\Gamma_{i,j}}$ denotes the trace function on the Γ interface, so that the value at the interface is computed as the half-sum of values coming from both subdomains at first step.

The solution to this second step is used to update the scalar flux at the end of a PDSA iteration:

$$\phi_{\ell+1}(\mathbf{r}) = \phi_{\ell+\frac{1}{2}}(\mathbf{r}) + \tilde{\phi}_{\mathcal{D},\ell+1}(\mathbf{r}), \quad \forall \mathbf{r} \in \mathcal{D}_i, \quad \forall i.$$

2.3. Discussion: rationale, advantages and limits of PDSA, previous experimental results

As mentioned in introduction, parallelizing the resolution of the diffusion equation is a rather difficult task. However, as shown by the very classical Fourier analysis of Source Iteration, the effectiveness of DSA comes from its ability to handle slowly oscillating error modes. This means that it should be possible to build effective acceleration operators which do not really solve diffusion problems: all that is needed is that slowly oscillating modes are correctly handled; other modes can be handled by Source Iteration.

The worst error mode being constant both in space and angle, it has a zero gradient everywhere. In order to compute it from a set of local diffusion problems on subdomains, it is therefore adequate to set Neumann boundary conditions at the interfaces between subdomains. This is the rationale for the first PDSA step. However, doing this produces flux correction terms which are discontinuous in the general case (for all non-constant error modes), causing the accelerated scheme to diverge. This is fixed by the second step, where the Dirichlet boundary condition ensures the continuity of the flux correction over the whole domain. The value imposed at interfaces between subdomains is determined so as to preserve global conservation for the flux correction term.

Although these boundary conditions at interfaces can not be correct, one can hope that, apart from a boundary layer near interfaces, their impact is not too significant. Therefore, PDSA is expected to be effective provided that the subdomains remain sufficiently optically thick that the introduction of the artificial boundary conditions at subdomain interfaces has minimal impact on the accuracy of the DSA correction.

Of course, this is no rigorous explanation, and a Fourier analysis of the proposed PDSA scheme should be conducted in order to study its effectiveness. This is the topic of section 3.3, in which it will be shown that, when subdomains are optically thick enough, the PDSA operator can accelerate Source Iteration. When this is the case, it features some advantages over standard DSA.

First and foremost, it is defined piecewise, which means it can adapt to any geometric domain partitioning used by the underlying transport solver. In a parallel context, there is only one point-to-point data exchange, so that the communication overhead can be considered low with respect to the computations performed within each subdomain.

Also, the two steps of PDSA are very similar problems. If an iterative solver is used to solve them, the second step can be initialized with the first to help it converge faster.

Moreover, if a DSA implementation is already available in a given neutron transport code (*i.e.* if a diffusion solver has already been developed, with a

consistent discretization scheme), PDSA can be implemented at almost no additional cost¹. This allows for easy parallelization of the acceleration scheme when parallelizing the transport solver.

Limitations of PDSA obviously lie in the conditions under which it accelerates Source Iteration. This limits the number of subdomains which can be defined for a given calculation. It should however be noted that section 3.3 gives indicators which can be computed beforehand to estimate the maximal number of subdomains allowed, or warn a user if the computation might not converge. Although this does not alleviate the limitation in the number of subdomains, it at least allows avoiding most common mistakes.

As mentioned in introduction, PDSA has already been implemented in the DOMINO solver of EDF’s industrial core-calculation platform COCAGNE [12]. Thanks to this implementation, the effectiveness of PDSA has already been tested in practice. The work presented in [14] shows that, for test cases with moderate optical thickness, PDSA loses some of the effectiveness of DSA but remains able to accelerate SI. On the Takeda [15] benchmark (model 1), 282 iterations are needed for SI to converge, whereas an optimized DSA implementation uses only 14 iterations to produce the same converged result. On this same test-case, the number of PDSA iterations ranges from 48 iterations for $2 \times 1 \times 1$ subdomains, to 118 iterations for $10 \times 10 \times 10$ subdomains.

On the other hand, for cases with larger optical thickness such as the 2-group, S_{12} critical calculation of a 900MW PWR [14, 11], experiments with DOMINO show that the effectiveness of PDSA is much closer to that of DSA. This test case requires 315 unaccelerated iterations to converge, while the number of accelerated iterations ranges from 56 for standard DSA, to 71 for PDSA with $4 \times 4 \times 4$ subdomains. This represents an increase of only 27% in the number of iterations, for a calculation which allows using 64 cluster nodes and 1536 cores.

The objective of the present paper is to present some theoretical background supporting these experimental findings, thus allowing to gain more confidence in PDSA.

2.4. Relationship to domain decomposition methods

It should be noted that the two diffusion steps (Neumann and Dirichlet) in PDSA correspond to the first iteration of a Domain Decomposition (DD) technique called the Dirichlet–Dirichlet algorithm in [16], or the Dirichlet preconditioned FETI method introduced in [17]. It is also related to Neumann–Neumann methods (which have been studied as early as [18]), in which each iteration defines the same two steps in the reverse order (the Dirichlet problem is solved first, and imposes a boundary condition to the Neumann problem).

All these domain decomposition techniques differ from the PDSA scheme proposed here, in that they are defined as iterative methods, *i.e.* it is proved

¹The only missing feature might be inhomogeneous Dirichlet boundary conditions, which are not always implemented in diffusion solvers.

that, whatever the value imposed on the interface at the first iteration, they converge to the solution of the diffusion problem when multiple iterations are performed. But nothing is said of the solution given after the first iteration. In our case, we impose a null current boundary condition in the Neumann step (6), and are able to prove that one iteration is enough to make PDSA convergent under some assumptions.

Should these conditions become too restrictive in practice, then a potential solution could be to add more Dirichlet–Dirichlet iterations, and turn PDSA into a full domain-decomposition technique. However, this increases the number of computations (and the number of communications in a parallel setup). In such a case, it would be interesting to compare the efficiency of the Dirichlet–Dirichlet method to other DD techniques such as the one described in [8]. Such a comparison should be performed in the specific case of DSA problems, since the diffusion solver is only required to attenuate some error modes in this context.

3. Fourier analysis

The 1D Fourier analysis is the primary tool used in the literature for the study of acceleration schemes [3]. In this section, we briefly review the well-known Fourier analysis of the standard Source Iteration and DSA schemes, before extending it to the proposed PDSA scheme.

We will perform this analysis for an homogeneous infinite 1D slab geometry, modeled by the finite spatial domain $\mathcal{D} = [0, L]$, with reflective boundary conditions. In this case, the neutron transport problem can be written as:

$$\begin{cases} \mu \frac{\partial \psi}{\partial x}(x, \mu) + \Sigma \psi(x, \mu) = \frac{\Sigma_s}{2} \int_{-1}^1 \psi(x, \mu') d\mu' + Q(x), \\ \psi(0, \mu) = \psi(0, -\mu), \\ \psi(L, \mu) = \psi(L, -\mu), \end{cases} \quad (7)$$

where notations are consistent with equation (1), except that in a 1D geometry, x represents the spatial variable and μ is the cosine of the angular direction. In the following, the scattering ratio will be denoted by $c = \frac{\Sigma_s}{\Sigma}$. It will be assumed to be strictly less than 1, in order for the transport problem to be well posed.

3.1. Source Iteration

The standard Source Iteration scheme is defined as:

$$\begin{cases} \mu \frac{\partial \psi_{\ell+1}}{\partial x}(x, \mu) + \Sigma \psi_{\ell+1}(x, \mu) = c \Sigma \phi_{\ell} + Q(x), \\ \psi_{\ell+1}(0, \mu) = \psi_{\ell+1}(0, -\mu), \\ \psi_{\ell+1}(L, \mu) = \psi_{\ell+1}(L, -\mu), \end{cases} \quad (8)$$

with the following notation for the scalar flux:

$$\phi_\ell(x) = \frac{1}{2} \int_{-1}^1 \psi_\ell(x, \mu') d\mu'.$$

The error after the ℓ^{th} iteration can be defined as $e_\ell = \psi(x, \mu) - \psi_\ell(x, \mu)$. This error satisfies an equation similar to (8), but with $Q(x) = 0$. Analyzing the convergence of the Source Iteration scheme towards ψ for an arbitrary Q source term is thus equivalent to studying the convergence towards 0 without the source term. In the following, we will thus consider $Q = 0$ and consider the flux ψ_ℓ to be an error term e_ℓ for which we will study the convergence towards 0.

The efficiency of the Source Iteration scheme is traditionally studied using a Fourier analysis. Assuming the (scalar) initial error to be given by

$$\phi_0(x) = \cos\left(\frac{\pi k x}{L}\right), \quad (9)$$

then the first iteration yields the angular flux

$$\psi_1(x, \mu) = \frac{c \mu \omega \sin(\omega \Sigma x) + c \cos(\omega \Sigma x)}{\mu^2 \omega^2 + 1},$$

where $\omega = \frac{\pi k}{\Sigma L}$ denotes the frequency of the initial error. After the first source iteration, the scalar flux is given by

$$\begin{aligned} \phi_{\text{SI}}(x) &= \frac{1}{2} \int_{-1}^1 \psi_1(x, \mu) d\mu \\ &= \rho_{\text{SI}}(\omega) \phi_0(x), \end{aligned}$$

where subscript SI denotes the Source Iteration scheme and

$$\rho_{\text{SI}}(\omega) = \frac{c \arctan \omega}{\omega}.$$

In other words, functions of the form given by ϕ_0 are eigenmodes of the Source Iteration operator, associated to eigenvalues ρ_{SI} . After ℓ source iterations, the scalar error is given by the expression:

$$\phi_{\text{SI}, \ell}(x) = \rho_{\text{SI}}^\ell(\omega) \phi_0(x).$$

The solid line in figure 2 presents the evolution of ρ_{SI} as a function of frequency ω . It shows that the spectral radius of the Source Iteration scheme is c , obtained for $\omega = 0$. In diffusive media, convergence can thus become arbitrarily slow. The slowest modes are defined by low frequencies ($\omega \ll 1$), and correspond to a weak spatial and angular dependence:

$$\psi_1(x, \mu) \underset{\omega \rightarrow 0}{\sim} c + O(\omega^2).$$

This shows that the Source Iteration scheme needs to be accelerated, and that the acceleration operator will be most effective if it handles slowly oscillating modes.

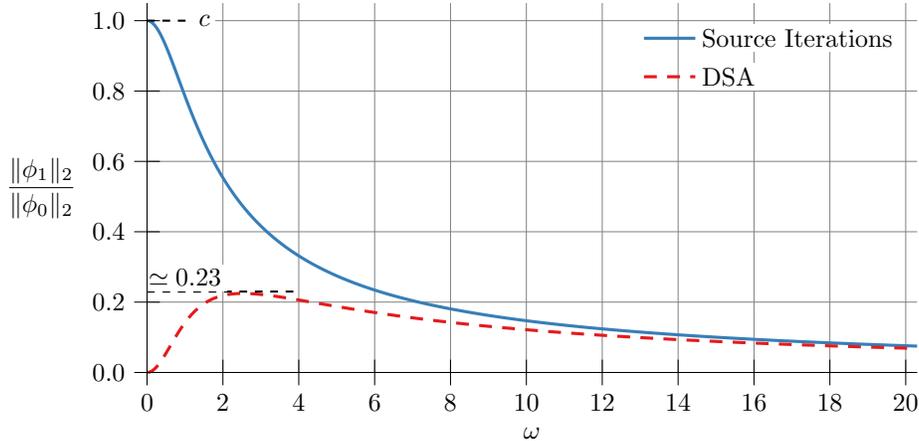


Figure 2: Amplification factors of the Source Iteration and DSA schemes as functions of frequency ω , in the diffusive case ($c = 1$).

3.2. Diffusion Synthetic Acceleration

In this section, we describe the Diffusion Synthetic Acceleration scheme, which can be used to improve the convergence properties of the Source Iteration scheme.

3.2.1. Diffusion problem

The accelerated scheme starts by a standard source iteration on a transport operator, as described in equation (8). Subtracting the Source Iteration equation from the exact transport equation (7) yields the equations satisfied by the iteration error $F_\ell(x, \mu)$ after ℓ iterations:

$$F_\ell(x, \mu) = \psi(x, \mu) - \psi_\ell(x, \mu).$$

Introducing $\psi_{\ell+1}$ in the right-hand side, and after rearrangement of the terms, the equations describing the error are given by:

$$\begin{cases} \mu \frac{\partial F_{\ell+1}}{\partial x}(x, \mu) + \Sigma F_{\ell+1}(x, \mu) &= c \Sigma [\phi(x) - \phi_\ell(x)] \\ &= c \Sigma f_{\ell+1}(x) + c \Sigma [\phi_{\ell+1}(x) - \phi_\ell(x)], \\ F_{\ell+1}(0, \mu) &= F_{\ell+1}(0, -\mu), \\ F_{\ell+1}(L, \mu) &= F_{\ell+1}(L, -\mu), \end{cases}$$

where we denoted by f_ℓ the scalar flux associated to error F_ℓ :

$$f_\ell(x) = \frac{1}{2} \int_{-1}^1 F_\ell(x, \mu') d\mu'.$$

This problem is of course as complicated to solve as the initial transport problem. The principle of DSA consists in replacing it with an approximated diffusion problem, whose solution is easier to compute. At the first iteration ($\ell = 0$), one thus computes the solution to the following problem:

$$\begin{cases} \frac{-1}{3\Sigma} f''(x) + (1 - c) \Sigma f(x) = c \Sigma [\phi_{\text{SI}}(x) - \phi_0(x)], \\ f'(0) = 0, \\ f'(L) = 0. \end{cases} \quad (10)$$

In the problem above, unknown f is supposed to be an approximation to the scalar flux associated to error F_1 :

$$f \simeq f_1.$$

We can show that the solution to this problem takes the form

$$f(x) = \rho_d(\omega) \phi_0(x),$$

where subscript d denotes that it comes from a diffusion calculation, and we have introduced

$$\rho_d(\omega) = \frac{c(3\rho_{\text{SI}}(\omega) - 3)}{\omega^2 - 3c + 3}.$$

Once again, this shows that functions of the form given by ϕ_0 are eigenmodes of the diffusion operator.

3.2.2. Flux correction

At the end of a diffusion-accelerated iteration, the scalar flux is given by

$$\begin{aligned} \phi_{\text{DSA}} &= \phi_{\text{SI}} + f \\ &= (\rho_{\text{SI}} + \rho_d) \phi_0 \\ &= \rho_{\text{DSA}} \phi_0, \end{aligned}$$

where subscripts DSA denote that the quantities are defined in the DSA scheme, and the eigenvalue associated to ϕ_0 for the whole iteration is denoted by

$$\rho_{\text{DSA}}(\omega) = \frac{\omega^2 \rho_{\text{SI}}(\omega) + 3\rho_{\text{SI}}(\omega) - 3c}{\omega^2 - 3c + 3}.$$

The dashed line of figure 2 presents, in the diffusive case ($c = 1$), the variation of ρ_{DSA} as a function of frequency ω . It shows that low frequency modes ($\omega \ll 1$) are associated with significantly lower eigenvalues in the DSA scheme than in the Source Iteration scheme. The spectral radius of the DSA iteration is approximately 0.23, obtained for $\omega \simeq 2.5$. This shows that DSA presents much more favorable convergence properties than the Source Iteration scheme (see for example [3] for a more thorough analysis of DSA).

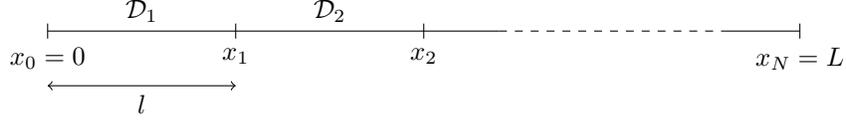


Figure 3: Partition in subdomains for the Piecewise Diffusion Synthetic Acceleration (PDSA) scheme.

3.3. Piecewise Diffusion Synthetic Acceleration (PDSA)

We now perform the same analysis, replacing standard DSA by the PDSA scheme introduced in section 2. Domain \mathcal{D} is partitioned in N subdomains without overlapping. In the remainder of this paper, the following notations will be used, as explained on figure 3:

$$\begin{aligned}
 l &= \frac{L}{N}, \\
 x_i &= i l, & 0 \leq i \leq N, \\
 \mathcal{D}_i &= [x_{i-1}, x_i], & 1 \leq i \leq N, \\
 t_i: \mathcal{D}_i &\rightarrow \mathcal{D}_1 & 1 \leq i \leq N. \\
 x &\mapsto x - x_{i-1},
 \end{aligned}$$

Each subdomain is a segment of length l . Translation t_i maps subdomain \mathcal{D}_i onto the reference subdomain $\mathcal{D}_1 = [0, l]$.

The Piecewise Diffusion Synthetic Acceleration is defined by the following steps:

1. a transport source iteration (8) is performed, yielding scalar flux ϕ_{SI} (it should be noted that this is a standard sweep over the entire domain, which may or may not be decomposed into subdomains at this stage);
2. a diffusion problem is solved in each subdomain, with outer boundary conditions coming from (10), and homogeneous Neumann conditions at interfaces between subdomains:

$$\begin{cases}
 \frac{-1}{3\Sigma} g''(x) + (1-c)\Sigma g(x) = c\Sigma [\phi_{\text{SI}}(x) - \phi_0(x)], \\
 g'(0) = g'(L) = 0, \\
 g'(x_i) = 0, & 1 \leq i \leq N-1.
 \end{cases} \quad (11)$$

3. a second diffusion problem is solved in each subdomain, again with outer boundary conditions from (10), but now with inhomogeneous Dirichlet conditions at the interfaces. The value set for the flux at the interfaces is obtained as the half sum of the interface values of the solutions in the

previous step:

$$\begin{cases} \frac{-1}{3\Sigma} h''(x) + (1-c)\Sigma h(x) = c\Sigma [\phi_{\text{SI}}(x) - \phi_0(x)], \\ h'(0) = h'(L) = 0, \\ h(x_i) = \frac{1}{2} [g^-(x_i) + g^+(x_i)], \quad 1 \leq i \leq N-1. \end{cases} \quad (12)$$

4. at the end of an iteration, the scalar flux is corrected using the solution of the second diffusion problem:

$$\phi_{\text{PDSA}} = \phi_{\text{SI}} + h.$$

3.3.1. Step 1: Neumann diffusion problem

We study here PDSA as a perturbation of the DSA scheme. We therefore consider the error introduced by the Neumann Diffusion problem (11), with respect to the global diffusion problem (10):

$$\delta = g - f.$$

Subtracting (10) to (11), rearranging the terms, and restricting it to \mathcal{D}_1 , we find that $\delta|_{\mathcal{D}_1}$ solves the following problem:

$$\begin{cases} \frac{-1}{3\Sigma} \delta''_{|\mathcal{D}_1}(x) + (1-c)\Sigma \delta_{|\mathcal{D}_1}(x) = 0, \\ \delta'_{|\mathcal{D}_1}(x_0) = -f'(x_0), \\ \delta'_{|\mathcal{D}_1}(x_1) = -f'(x_1). \end{cases}$$

$\delta|_{\mathcal{D}_1}$ can thus be defined as the linear combination

$$\delta|_{\mathcal{D}_1} = -f'(x_0) e_{\text{NN}}^1 - f'(x_1) e_{\text{NN}}^r,$$

where functions e_{NN}^1 and e_{NN}^r measure the error due to not knowing the real boundary values of f' respectively on the left and right extremities of \mathcal{D}_1 :

$$\begin{cases} \frac{d^2 e_{\text{NN}}^1}{dx^2}(x) - \alpha^2 e_{\text{NN}}^1(x) = 0, \\ \frac{de_{\text{NN}}^1}{dx}(0) = 1, \\ \frac{de_{\text{NN}}^1}{dx}(l) = 0, \end{cases} \quad \text{and} \quad \begin{cases} \frac{d^2 e_{\text{NN}}^r}{dx^2}(x) - \alpha^2 e_{\text{NN}}^r(x) = 0, \\ \frac{de_{\text{NN}}^r}{dx}(0) = 0, \\ \frac{de_{\text{NN}}^r}{dx}(l) = 1, \end{cases}$$

where parameter α is defined as

$$\alpha = \sqrt{3(1-c)} \Sigma.$$

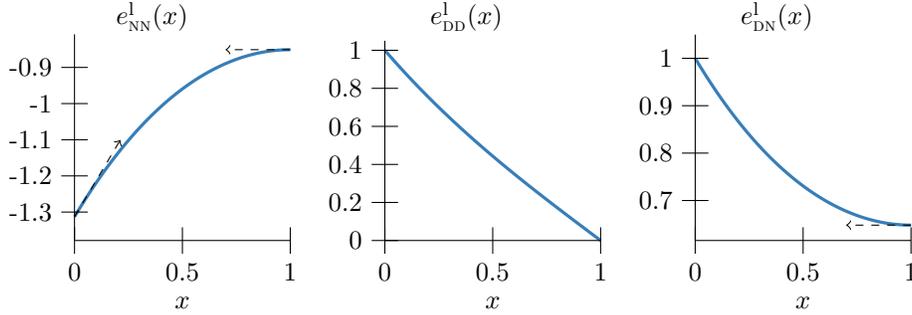


Figure 4: Auxilliary functions $e_{NN}^1(x)$, $e_{DD}^1(x)$ and $e_{DN}^1(x)$, for $l = 1$ and $\alpha = 1$

Solutions to these problems can be calculated analytically, and are linked by symmetry relations. Figure 4 show a plot of e_{NN}^1 in order to help visualize its behaviour.

$$e_{NN}^1(x) = -\frac{e^{-\alpha x} (e^{2\alpha x} + e^{2\alpha l})}{\alpha e^{2\alpha l} - \alpha} \quad \text{and} \quad e_{NN}^r(x) = -e_{NN}^1(l - x). \quad (13)$$

Finally, any subdomain \mathcal{D}_i can be mapped to \mathcal{D}_1 using translation t_i , which allows following the same line of reasoning to obtain:

$$\delta = -\sum_{i=1}^N \chi_i [f'(x_{i-1}) e_{NN}^1 + f'(x_i) e_{NN}^r] \circ t_i, \quad (14)$$

where χ_i denotes the indicator function for subdomain \mathcal{D}_i .

3.3.2. Step 2: Dirichlet diffusion problem

We now consider the Dirichlet Diffusion problem (12) as a perturbation of DSA, and define error

$$\varepsilon = f - h,$$

which follows equation

$$\begin{cases} \frac{-1}{3\Sigma} \varepsilon''(x) + (1 - c) \Sigma \varepsilon(x) = 0, \\ \varepsilon'(0) = \varepsilon'(L) = 0, \\ \varepsilon(x_i) = \varepsilon_i, \quad 1 \leq i \leq N - 1. \end{cases}$$

In the equation above, the value at subdomain interfaces is given by

$$\varepsilon_i = h(x_i) - f(x_i) = \frac{1}{2} [g^-(x_i) + g^+(x_i)] - f(x_i) = \frac{1}{2} [\delta^-(x_i) + \delta^+(x_i)].$$

Equation (14) yields

$$\begin{aligned}\delta^-(x_i) &= -f'(x_{i-1}) e_{\text{NN}}^1(l) - f'(x_i) e_{\text{NN}}^r(l), \\ \delta^+(x_i) &= -f'(x_i) e_{\text{NN}}^1(0) - f'(x_{i+1}) e_{\text{NN}}^r(0),\end{aligned}$$

and, noticing that terms evaluated at point x_i vanish thanks to symmetry relation (13),

$$\varepsilon_i = \frac{1}{2} \left[f'(x_{i+1}) - f'(x_{i-1}) \right] e_{\text{NN}}^1(l). \quad (15)$$

Following the same line of reasoning than for the Neumann diffusion problem, in each internal subdomain \mathcal{D}_i , $2 \leq i \leq N-1$, error $\varepsilon_{|\mathcal{D}_i}$ can be expressed as the linear combination

$$\varepsilon_{|\mathcal{D}_i} = [\varepsilon_{i-1} e_{\text{DD}}^1 + \varepsilon_i e_{\text{DD}}^r] \circ t_i, \quad (16)$$

where e_{DD}^1 and e_{DD}^r respectively measure errors stemming from not knowing the value h should take at the left and right extremities of the subdomain:

$$\left\{ \begin{array}{l} \frac{d^2 e_{\text{DD}}^1}{dx^2}(x) - \alpha^2 e_{\text{DD}}^1(x) = 0, \\ e_{\text{DD}}^1(0) = 1, \\ e_{\text{DD}}^1(l) = 0, \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \frac{d^2 e_{\text{DD}}^r}{dx^2}(x) - \alpha^2 e_{\text{DD}}^r(x) = 0, \\ e_{\text{DD}}^r(0) = 0, \\ e_{\text{DD}}^r(l) = 1. \end{array} \right.$$

As for the Neumann Diffusion problem, the solutions to these problems can be expressed analytically, and are linked by symmetry relations (*cf* Fig. 4):

$$e_{\text{DD}}^1(x) = -\frac{e^{-\alpha x} (e^{2\alpha x} - e^{2\alpha l})}{e^{2\alpha l} - 1} \quad \text{and} \quad e_{\text{DD}}^r(x) = e_{\text{DD}}^1(l-x).$$

Boundary subdomains \mathcal{D}_1 and \mathcal{D}_N must be handled specially, since they have mixed boundary conditions: on one of their extremities, the boundary condition is known exactly; the error only comes from not knowing the exact boundary condition on the other extremity. We can write

$$\varepsilon_{|\mathcal{D}_1} = \varepsilon_1 e_{\text{ND}}^r \quad \text{and} \quad \varepsilon_{|\mathcal{D}_N} = \varepsilon_{N-1} e_{\text{DN}}^1 \circ t_N, \quad (17)$$

where e_{DN}^1 and e_{ND}^r are given by

$$\left\{ \begin{array}{l} \frac{d^2 e_{\text{DN}}^1}{dx^2}(x) - \alpha^2 e_{\text{DN}}^1(x) = 0, \\ e_{\text{DN}}^1(0) = 1, \\ \frac{de_{\text{DN}}^1}{dx}(l) = 0, \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \frac{d^2 e_{\text{ND}}^r}{dx^2}(x) - \alpha^2 e_{\text{ND}}^r(x) = 0, \\ \frac{de_{\text{ND}}^r}{dx}(0) = 0, \\ e_{\text{ND}}^r(l) = 1. \end{array} \right.$$

As in previous cases, analytical and symmetric expressions can be found for these terms, of which Figure 4 shows a plot:

$$e_{\text{DN}}^1(x) = \frac{e^{-\alpha x} (e^{2\alpha x} + e^{2\alpha l})}{e^{2\alpha l} + 1} \quad \text{and} \quad e_{\text{ND}}^r(x) = e_{\text{DN}}^1(l - x).$$

It may be interesting to note here that e_{NN}^1 and e_{DN}^1 are identical functions, aside from a scaling factor, which tends to $-\alpha$ in the thick limit (when $l \rightarrow \infty$):

$$\begin{aligned} e_{\text{DN}}^1(x) &= \frac{e^{-\alpha x} (e^{2\alpha x} + e^{2\alpha l})}{e^{2\alpha l} + 1} \\ &= -\frac{e^{-\alpha x} (e^{2\alpha x} + e^{2\alpha l})}{\alpha e^{2\alpha l} - \alpha} \frac{\alpha (1 - e^{2\alpha l})}{1 + e^{2\alpha l}} \\ &= \frac{\alpha (1 - e^{2\alpha l})}{1 + e^{2\alpha l}} e_{\text{DN}}^1(x). \end{aligned}$$

3.3.3. Flux correction

After a PDSA iteration, the corrected scalar flux is given by

$$\phi_{\text{PDSA}} = \phi_{\text{SI}} + h = \phi_{\text{SI}} + f + \varepsilon = \rho_{\text{DSA}} \phi_0 + \varepsilon.$$

Unlike in standard DSA, ϕ_0 is not an eigenmode of the PDSA scheme. It is therefore more difficult to express error evolutions from one iteration to the next. It is however possible to state that

$$\frac{\|\phi_{\text{PDSA}}\|_2}{\|\phi_0\|_2} \leq \underbrace{\rho_{\text{DSA}} + \frac{\|\varepsilon\|_2}{\|\phi_0\|_2}}_{\rho_{\text{PDSA}}^{\max}}, \quad (18)$$

where $\rho_{\text{PDSA}}^{\max}$ denotes the upper bound of the amplification factor of the whole PDSA scheme.

Equations (16) and (17) yield

$$\varepsilon = \chi_1 \varepsilon_1 e_{\text{ND}}^r + \sum_{i=2}^{N-1} \chi_i [\varepsilon_{i-1} e_{\text{DD}}^1 + \varepsilon_i e_{\text{DD}}^r] \circ t_i + \chi_N \varepsilon_{N-1} e_{\text{DN}}^1 \circ t_N,$$

and

$$\begin{aligned} \|\varepsilon\|_2^2 &= \varepsilon_1^2 \|e_{\text{ND}}^r\|_2^2 + \sum_{i=2}^{N-1} \|\varepsilon_{i-1} e_{\text{DD}}^1 + \varepsilon_i e_{\text{DD}}^r\|_2^2 + \varepsilon_{N-1}^2 \|e_{\text{DN}}^1\|_2^2 \\ &\leq \varepsilon_1^2 \|e_{\text{ND}}^r\|_2^2 + 2 \sum_{i=2}^{N-1} \left(\varepsilon_{i-1}^2 \|e_{\text{DD}}^1\|_2^2 + \varepsilon_i^2 \|e_{\text{DD}}^r\|_2^2 \right) + \varepsilon_{N-1}^2 \|e_{\text{DN}}^1\|_2^2 \\ &\leq 5 \|e_{\text{DD}}^1\|_2^2 \sum_{i=1}^N \varepsilon_i^2. \end{aligned}$$

The last inequality was obtained by noticing that $\|e_{\text{ND}}^r\|_2^2 = \|e_{\text{DN}}^1\|_2^2 \leq 3 \|e_{\text{DD}}^1\|_2^2$ and $\|e_{\text{DD}}^1\|_2^2 = \|e_{\text{DD}}^r\|_2^2$.

Equation (15) also allows us to bound the error at interfaces

$$\begin{aligned} |\varepsilon_i| &\leq \frac{1}{2} \left(|f'(x_{i+1})| + |f'(x_{i-1})| \right) |e_{\text{NN}}^1(l)| \\ &\leq |e_{\text{NN}}^1(l)| \sup_x |f'(x)| \\ &\leq |\rho_d| |e_{\text{NN}}^1(l)| \sup_x |\phi_0'(x)| \\ &\leq |\rho_d| \frac{k\pi}{Nl} |e_{\text{NN}}^1(l)|, \end{aligned}$$

so that

$$\varepsilon_i^2 \leq \rho_d^2 \left(\frac{k\pi}{Nl} \right)^2 (e_{\text{NN}}^1(l))^2.$$

Combining previous results yields the following global bound:

$$\|\varepsilon\|_2 \leq \sqrt{5N} \rho_d \frac{k\pi}{Nl} |e_{\text{NN}}^1(l)| \|e_{\text{DD}}^1\|_2.$$

Noticing that, as soon as $k \neq 0$, $\|\phi_0\|_2 = \sqrt{\frac{L}{2}} = \sqrt{\frac{Nl}{2}}$, it follows that

$$\begin{aligned} \frac{\|\varepsilon\|_2}{\|\phi_0\|_2} &\leq \sqrt{\frac{2}{Nl}} \sqrt{5N} \rho_d \frac{k\pi}{Nl} |e_{\text{NN}}^1(l)| \|e_{\text{DD}}^1\|_2 \\ &\leq \underbrace{\sqrt{\frac{10}{3(1-c)}} \rho_d \omega}_{\tilde{\rho}_d(\omega)} \underbrace{\frac{\alpha |e_{\text{NN}}^1(l)| \|e_{\text{DD}}^1\|_2}{\sqrt{l}}}_R \end{aligned} \quad (19)$$

It should be mentioned that the first part of this expression, denoted by $\tilde{\rho}_d$, only depends on the scattering ratio c and the frequency ω . As shown by an asymptotic development and illustrated in figure 5, in the asymptotic limit when $c \rightarrow 1$, the maximum value of $\tilde{\rho}_d$ is approximately given by $\frac{1.26}{\sqrt{1-c}}$.

On the other hand, the second part of expression (19), denoted by R , can be expressed as

$$R(\theta) = \sqrt{\frac{2e^{6\theta} - 8\theta e^{4\theta} - 2e^{2\theta}}{\theta e^{8\theta} - 4\theta e^{6\theta} + 6\theta e^{4\theta} - 4\theta e^{2\theta} + \theta}},$$

where we defined the quantity

$$\theta = \alpha l = \sqrt{3(1-c)} \frac{\Sigma L}{N},$$

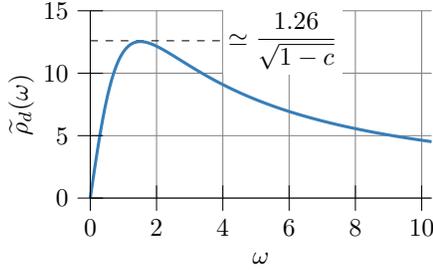


Figure 5: Evolution of factor $\tilde{\rho}_d$ with frequency ω , for $c = 0.99$.

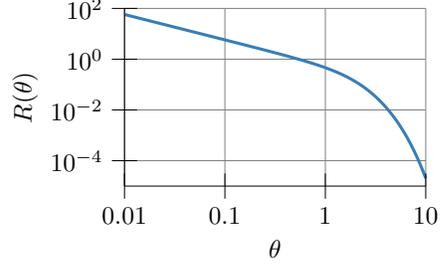


Figure 6: Evolution of factor R with optical thickness θ .

which is a dimensionless parameter depending only on the physical properties associated to the problem, and characterizes the optical thickness of a subdomain. Figure 6 presents the variation of factor R with optical thickness θ . As shown by asymptotic developments for small and large optical thicknesses, R is not bounded for small optical thicknesses, but converges extremely rapidly towards 0 when the optical thickness of the subdomains increases:

$$R(\theta) \underset{\theta \rightarrow 0}{\sim} \frac{1}{\sqrt{3}\theta} \quad \text{and} \quad R(\theta) \underset{\theta \rightarrow \infty}{\sim} \frac{\sqrt{2}e^{-\theta}}{\sqrt{\theta}}.$$

3.3.4. Convergence

As a conclusion, for any set of cross sections Σ and Σ_s , there exists a critical subdomain size \underline{l} such that

$$\forall l \geq \underline{l}, \quad \frac{\|\varepsilon\|_2}{\|\phi_0\|_2} < 1 - \rho_{\text{DSA}},$$

so that, from equation (18),

$$\frac{\|\phi_{\text{PDSA}}\|_2}{\|\phi_0\|_2} \leq \rho_{\text{PDSA}}^{\max} < 1, \quad (20)$$

and PDSA can accelerate the convergence of Source Iteration. Moreover, as the subdomain size l increases above the critical size, the efficiency of PDSA rapidly converges to that of standard DSA:

$$\frac{\|\phi_{\text{PDSA}}\|_2}{\|\phi_0\|_2} \xrightarrow{l \rightarrow \infty} \rho_{\text{DSA}}.$$

Conversely, since $\tilde{\rho}_d \xrightarrow{c \rightarrow 0} 0$, for any domain of fixed optical thickness $\tau = \Sigma L$, there exists a critical scattering ratio \bar{c} under which PDSA converges:

$$\forall c \leq \bar{c}, \quad \frac{\|\phi_{\text{PDSA}}\|_2}{\|\phi_0\|_2} \leq \rho_{\text{PDSA}}^{\max} < 1.$$

In practice, this limits the use of PDSA to cases which are optically thick enough for condition (20) to apply for the whole geometrical domain. In such cases, the condition also limits the maximal number of subdomains which can be used.

3.3.5. Special case: two-subdomain partition

In the special case where the domain is partitioned in two subdomains, the first Neumann diffusion step in PDSA yields, from equation (15) and boundary conditions from problem (10):

$$\varepsilon_1 = \frac{1}{2} \left[f'(0) - f'(L) \right] e_{\text{NN}}^1(l) = 0.$$

It follows that the second PDSA step, Dirichlet diffusion, yields the solution to the global DSA problem: $h = f$. In this case, PDSA is thus equivalent to a global DSA scheme.

4. Numerical results

In order to assess the validity of the above theory, we present in this section some numerical results.

These results were obtained using a very simple code, developed in Julia. We consider the time-independent, one-group Boltzmann transport equation with isotropic scattering, set in a homogeneous 1D slab geometry over the spatial domain $[0, L]$. In order to model a full reactor core, we impose void boundary conditions with zero incoming flux:

$$\begin{cases} \mu \frac{\partial \psi}{\partial x}(x, \mu) + \Sigma \psi(x, \mu) = \frac{\Sigma_s}{2} \int_{-1}^1 \psi(x, \mu') d\mu' + Q(x), \\ \psi(0, \mu) = 0 & \forall \mu > 0, \\ \psi(L, \mu) = 0 & \forall \mu < 0. \end{cases}$$

The solver uses the discrete-ordinates method to handle the angular dependency of the solution. The transport equation is spatially discretized using a standard diamond-differencing (DD) scheme. The diffusion equations used in the DSA and PDSA schemes are discretized using a P_1 finite-element method. Table 1 summarizes the main parameters used in the discretization.

In the following, we will set a unit-length domain ($L = 1$) and a linear source ($Q(x) = x$). The cases studied will vary only with respect to the material used in the geometry, which can be entirely characterized by its total and scattering cross-sections Σ and Σ_s . Equivalently, the problem may be characterized by its total optical thickness $\tau = \Sigma L$ and its absorption ratio $\epsilon = 1 - c = 1 - \frac{\Sigma_s}{\Sigma}$.

From the bounds discussed above, one may expect the PDSA scheme to converge when the optical thickness of each domain $\frac{\tau}{N}$ is large enough and ϵ is not too small.

Parameter	Value
Angular discretization	
Method	discrete ordinates
Angular quadrature	10 directions
Spatial discretization	
Transport	diamond-differencing
Acceleration	P_1 finite elements
Spatial mesh	1 000 meshes
Iteration control	
Max. number of outer iterations	10 000
Stopping criterion	$\Sigma_s \ \phi_{\ell+1} - \phi_\ell\ _\infty < 10^{-5}$

Table 1: Discretization parameters for numerical tests

4.1. Fourier analysis

Setting an initial flux of the form given by (9) and performing an iteration, one can perform a numerical Fourier analysis of the different schemes.

The results of such an analysis are presented in figure 7, in the case where $\tau = 10$ and $\epsilon = 0.1$. Unsurprisingly, the Source Iteration and DSA schemes behave similarly to figure 2. The behavior of PDSA is presented for different numbers of subdomains. As noted in paragraph 3.3.5, PDSA with two subdomains is exactly equivalent to standard DSA. Then, as the number of subdomains increases, larger and larger perturbations start to appear until the amplification factor exceeds 1 for 9 subdomains.

The dashed black line in figure 7 represents the theoretical bound on the amplification factor, as obtained using eqs. (18)–(19) in the case of 3 subdomains. It appears that this value effectively bounds the measured amplification factor, but is not very sharp.

However, such a problem being neither very optically thick nor very diffusive, it is not representative of the cases where PDSA would be applied in practice for PWR calculations. In order to show a behaviour when the optical thickness increases, figure 8 presents the same analysis for $\tau = 30$ and $\epsilon = 0.05$. In this case, both 2-domain and 3-domain PDSA are indistinguishable from standard DSA, and amplification factors for other numbers of subdomains are reduced as expected. The theoretical bound for PDSA(3) is still pessimistic, but stays in more acceptable limits.

4.2. Number of iterations

The practical interest of PDSA can be assessed in terms of the reduction in the number of iterations. Table 2 presents a comparison of the acceleration schemes on different problems. An “X” marks settings in which DSA does not converge.

The first two rows of the table (cases A and B) correspond to the two cases used for the Fourier analysis in the previous section. In case A, we can see that,

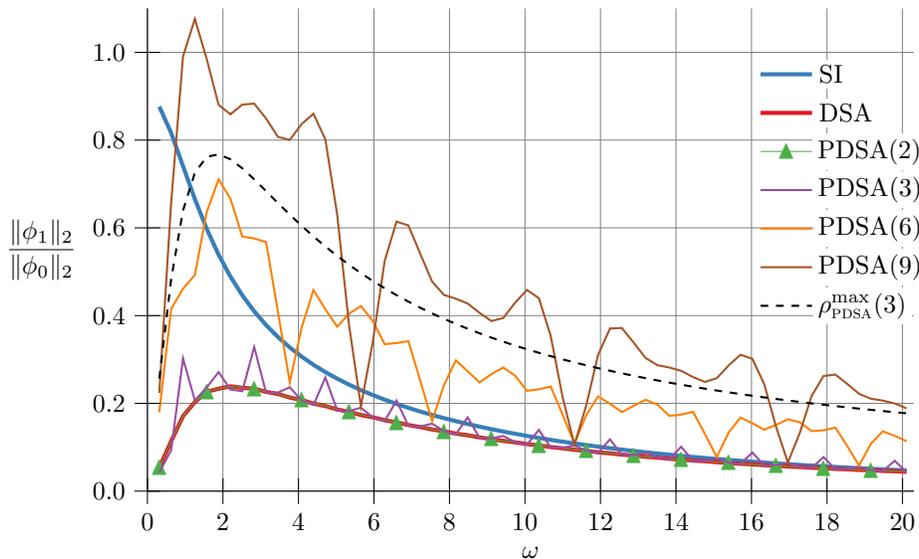


Figure 7: Measured amplification factors of various acceleration schemes, for the case where $\tau = 10$ and $\epsilon = 0.1$ (case A in table 2).

	τ	ϵ	SI	DSA	PDSA(3)	PDSA(4)	PDSA(6)	PDSA(9)
A	10	0.100	144	24	24	24	24	X
B	30	0.050	342	27	27	27	27	27
C	30	0.010	1399	34	34	34	56	X
D	30	0.005	2248	35	36	36	X	X
E	30	0.001	4351	38	X	X	X	X

Table 2: Iteration count of the various schemes for several cases.

as expected, 9-domain PDSA does not converge in the first case. However, although figure 7 showed rather degraded amplification factors for 6-domain PDSA, its iteration count is in practice not higher than for the standard DSA. Similar results occur for case B, in which all PDSA schemes exhibit no degradation of efficiency with respect to standard DSA. This is in contrast to figure 8, which evidenced a degradation of the amplification factor for 9-domain PDSA.

Cases C–E demonstrate the behaviour of the iteration count as ϵ decreases. Unsurprisingly, the number of source iterations increases with the scattering ratio. This is in contrast with the rather stable DSA iteration count. The PDSA schemes behave almost identically to DSA, until they reach a point where the number of iterations starts increasing. The scheme stops converging soon after this point.

This is more clearly shown on figure 9, which presents the variation of the iteration count with the scattering ratio. Reading the figure from right to left:

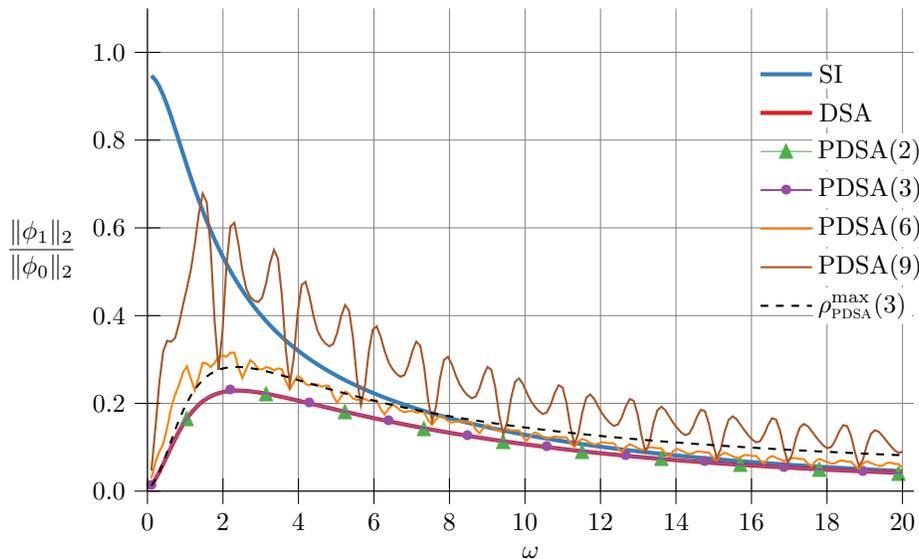


Figure 8: Measured amplification factors of various acceleration schemes, for $\tau = 30$ and $\epsilon = 0.05$ (case B in table 2).

as the scattering ratio increases, the number of source iterations increases. At the same time, the DSA iteration count stays more stable. PDSA behaves identically to DSA, until the scattering ratio approaches a critical value, at which its performance degrades very rapidly. This illustrates the existence of a critical scattering ratio \bar{c} , as mentioned in section 3.3.4.

The main conclusion to draw from this study is that, when it converges, PDSA almost always exhibits the same performance as standard DSA. The following part discusses the conditions under which PDSA does converge.

4.3. Convergence region

Figure 10 presents the convergence region of PDSA. For low scattering ratios and high optical depths (in the top right part of the figure), PDSA converges. Then, as the scattering ratio increases above the critical value \bar{c} , the scheme leaves the convergence region. On figure 10, dashed lines present the theoretical critical scattering ratio. That is, the dashed lines are level curves for which $\rho_{\text{PDSA}}^{\text{max}} = 1$. On the other hand, solid lines present the critical point at which the scheme is observed to start diverging in practice.

First, it is interesting to note that the theoretical value always bounds the practical one. In other words, for a given number of subdomains, the dashed line is always above the solid one. The overestimation of $\rho_{\text{PDSA}}^{\text{max}}$, observed in figures 7 and 8, manifests itself as a gap between the theoretical and observed values. In practice, theoretical bounds can help ensuring that PDSA will converge when $\rho_{\text{PDSA}}^{\text{max}} < 1$. However, if the theoretical bound goes above unity a practical

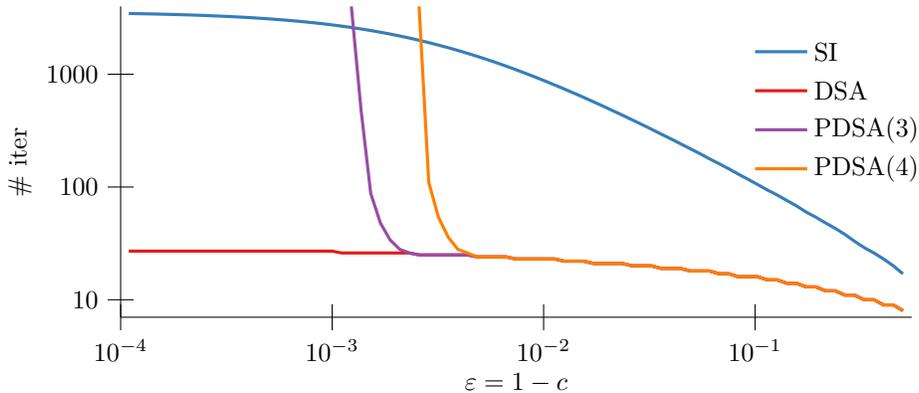


Figure 9: Iteration count as a function of the scattering ratio, for $\tau = 30$.

test should still be conducted, as PDSA might still very well converge. This is especially true in the limit of large optical thicknesses, where the overestimation of $\rho_{\text{PDSA}}^{\text{max}}$ appears to increase.

5. Conclusions

We presented in this paper a Piecewise Diffusion Synthetic Acceleration scheme (PDSA), which is specifically designed to be straightforwardly used in parallel contexts. The implementation of PDSA only requires having a standard neutron diffusion solver whose discretization is consistent with that of the neutron transport solver. In practice, and as explained in [11], starting from an initially sequential DSA-accelerated transport code, one only needs to take care of the parallelization of the transport solver; the parallel acceleration scheme comes at no practical development cost.

We showed that, although PDSA only approximates DSA, it converges for a class of problems which are optically thick enough. For this class of problems, we also showed that PDSA is in practice as efficient as standard DSA, in terms of the number of iterations. We presented an indicator, coming from 1D geometries but computable for any kind of 3D problem, allowing us to estimate *a priori* if the problem at hand is optically thick enough for PDSA to converge.

The limited accuracy of this indicator is the main shortcoming of this work. Simple 1D experiments show that PDSA performs in practice much better than the indicator would predict. The work presented in [11] draws similar conclusions for more complex, 3D, industrial calculations. In practice, the theoretical indicator presented here can be used to guarantee that the method will converge, but no practical conclusion can be drawn as to the divergence of the scheme. This might be because the bounds derived here are not tight enough to yield the sharp estimators that one would like to have in practice. Also, as source iterations advance, we might expect the DSA correction to be smoother

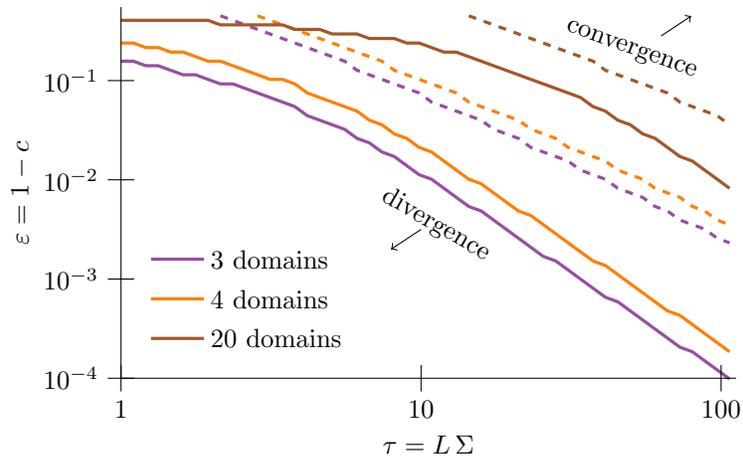


Figure 10: Boundary of the domain of convergence of PDSA: the scheme converges only for parameters which are above the curves. Theoretical limits are indicated by dashed lines, while solid lines indicate the experimentally measured limits.

and smoother, and the gradient of the correction to be closer and closer to zero. This phenomenon has not been accounted for here, although it could help reduce the error made in the first PDSA step. This should be the topic of further analyses and work. The applicability of these homogeneous, 1D results to the more useful case of heterogeneous, multi-dimensional calculations should also be investigated.

References

- [1] R. E. Alcouffe, Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations, *Nuclear Science and Engineering* 64 (1977) 344–355.
- [2] E. W. Larsen, Diffusion-synthetic acceleration methods for discrete-ordinates problems, *Transport Theory and Statistical Physics* 13 (1984) 107–126.
- [3] M. L. Adams, E. W. Larsen, Fast iterative methods for discrete-ordinates particle transport calculations, *Progress in Nuclear Energy* 40 (1) (2002) 3–159.
- [4] G. G. Davidson, T. M. Evans, J. J. Jarrell, R. N. Slaybaugh, Massively parallel, three-dimensional transport solutions for the k-eigenvalue problem, in: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2011)*, Brazil, 2011.

- [5] T. Courau, L. Plagne, A. Ponçot, G. Sjoden, Hybrid parallel code acceleration methods in full-core reactor physics calculations, in: ANS Reactor Physics Topical Meeting (PHYSOR), Knoxville, Tennessee, USA, 2012.
- [6] T. Courau, S. Moustafa, L. Plagne, A. Ponçot, DOMINO: A fast 3D cartesian discrete ordinates solver for reference PWR simulations and SPN validation, in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, 2013.
- [7] M. Barrault, B. Lathuilière, P. Ramet, J. Roman, Efficient parallel resolution of the simplified transport equations in mixed-dual formulation, *Journal of Computational Physics* 230 (5) (2011) 2004–2020.
- [8] E. Jamelot, P. Ciarlet Jr, Fast non-overlapping Schwarz domain decomposition methods for solving the neutron diffusion equation, *Journal of Computational Physics* 241 (2013) 445–463.
- [9] R. Lenain, E. Masiello, R. Sanchez, F. Damian, A parallel full core transport calculation based on domain decomposition method, in: Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo (SNA+MC), 2013.
- [10] R. Slaybaugh, T. Evans, G. Davidson, P. Wilson, Multigrid in energy preconditioner for Krylov solvers, *Journal of Computational Physics* 242 (2013) 405–419.
- [11] S. Moustafa, F. Févotte, M. Faverge, L. Plagne, P. Ramet, Efficient parallel solution of the 3D stationary Boltzmann transport equation for diffusive problems, submitted to *Journal of Computational Physics*, <https://arxiv.org/abs/1710.01536>.
- [12] A. Calloo, D. Couyras, F. Févotte, M. Guillo, COCAGNE: EDF new neutron core code for ANDROMÈDE calculation chain, in: International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C), Jeju, Korea, 2017.
- [13] A. Hébert, The search for superconvergence in spherical harmonics approximations, *Nuclear Science and Engineering* 154 (2006) 134.
- [14] S. Moustafa, Massively parallel cartesian discrete ordinates method for neutron transport simulation, Ph.D. thesis, Université de Bordeaux (Dec. 2015).
URL <https://tel.archives-ouvertes.fr/tel-01379686>
- [15] T. Takeda, H. Ikeda, Final report on the 3-D neutron transport benchmarks, Tech. rep., OECD/NEA Committee on Reactor Physics (Mar. 1991).
URL <http://www.nea.fr/html/science/docs/1990/neacrp-1-1990-330.pdf>

- [16] A. Toselli, O. Widlund, Domain Decomposition Methods – Algorithms and Theory, Springer Series in Computational Mathematics, Springer, 2000.
- [17] C. Farhat, J. Mandel, F.-X. Roux, Optimal convergence properties of the FETI domain decomposition method, *Computer Methods in Applied Mechanics and Engineering* 115 (3–4) (1994) 365–385.
- [18] R. Glowinski, M. F. Wheeler, Domain decomposition and mixed finite element methods for elliptic problems, in: *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, 1987, pp. 144–172.