International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)
Sun Valley, Idaho, USA, May 5-9, 2013, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2013)

NOVEL FINE FLUX INTEGRATION METHODS FOR THE DIABOLO SIMPLIFIED TRANSPORT SOLVER IN COCAGNE

COUYRAS David, FÉVOTTE François and PLAGNE Laurent
EDF R&D
david.couyras@edf.fr, francois.fevotte@edf.fr, laurent.plagne@edf.fr

ABSTRACT

As part of its activity, EDF R&D is developing a new nuclear core simulation code named COCAGNE. This code relies on DIABOLO, a Simplified P_N (SP_N) method to compute the neutron flux inside the core for k_{eff} eigenvalue problems. In order to complete complex simulations involving a large number of successive eigenvalue calculations within acceptable CPU times, the DIABOLO solver uses computational meshes that contain a restricted number of cells (e.g. 4×4 cells per assembly). Solutions obtained on these relatively coarse computational meshes are then spatially interpolated to obtain physical quantities such as neutron production rates on finer meshes (e.g. pin-by-pin power production). This paper describes two novel methods, named *Strawhat* and *Poisson*, that allow the neutron flux to be integrated over fine mesh cells. Unlike the original method used in COCAGNE, these new methods compute accurate fine flux integrals trough the interpolation of the current DOFs of DIABOLO's mixed dual RT_k finite elements. Assessed on a set of 3-D PWR realistic core configurations, these methods are shown to bring about significant improvements over the original integration scheme: for smooth enough flux distribution, the *Poisson* method improves the pin-by-pin production rates accuracy by as much as one order of magnitude.

Key Words: SP_N (Simplified Transport); Fine Flux Integration; Fine Power Reconstruction; DIABOLO.

1. INTRODUCTION

As operator of nuclear power plants, EDF performs many simulations of nuclear reactor cores in the processes of either reactor operation or design. Currently under development at EDF R&D, the COCAGNE nuclear reactor core simulation system aims at meeting this need. As a convenient trade-off between accuracy and numerical complexity, the Simplified P_N (SP_N) approximation provided by the DIABOLO* solver [1] is frequently used in neutron transport simulations.

In order to complete complex simulations involving a large number of successive eigenvalue calculations within acceptable CPU times, the DIABOLO solver uses computational meshes that contain a limited number of cells. Based on relatively coarse computational meshes, DIABOLO's neutron flux solutions are spatially interpolated on finer meshes to evaluate physical quantities such as power production rates at the pin-cell level. This paper describes two novel methods dedicated to this interpolation process, that produces neutron flux integrals on fine mesh cells. Unlike the original method, which only uses the *flux* Degrees of Freedom (DOFs) of DIABOLO's RT_k mixed finite elements discretization scheme, the two methods introduced in this paper are based on the *current* DOFs. At the present time the proposed new fine integration methods are restricted to SP_N eigenvalue calculations with null flux boundary conditions.

The context of this paper is described in section 2, which introduces the 2-step approach used for the EDF calculation scheme. Section 3 briefly presents the SP_N equations and their numerical solution. Based on

*DIABOLO is the new name of the COCAGNE platform SP_N solver.

these equations, section 4 describes the numerical derivation of the two novel flux integration methods and section 5 illustrates them on a simple analytical case. Section 6 details the experimental set-up used to study the numerical accuracy of the fine flux integrals and the production rates as a function of the computation CPU time. Two realistic 3-D PWR loading patterns are used to assess the efficiency of the fine flux integration (FFI) methods. Section 7 concludes the paper.

2. THE 2-STEP EDF CALCULATION SCHEME

The future EDF calculation scheme is based on a standard 2-step approach. The first step consists in computing few-group cross-section libraries thanks to assembly calculations performed with a 2-D deterministic transport code. The second step deals with simplified transport (SP_N) 3-D calculations of the whole reactor core using the data computed at the first stage.

The assembly calculation stage consists in solving the transport equation with a fine discretization in space and energy. The transport calculations are based on the 2-D deterministic LWR assembly calculation scheme REL2005 [2] which is validated by the CEA. The 2-D lattice code used for this purpose is APOLLO2 [3] developed by the CEA with the support of EDF and AREVA. The multi-group cross-section library associated with the REL2005 scheme uses the SHEM 281-group energy mesh [4]. Isotopic cross-sections gathered in the multi-group library used by APOLLO2 are mainly derived from the JEFF3.1 punctual evaluations.

The assembly calculation produces cross-sections which are collapsed to two energy groups and spatially homogenized. These cross-sections are then used in the 3-D core calculation, which consists in solving the simplified transport (SP_N) equations with two energy groups and a coarse Cartesian spatial mesh. This resolution is performed by the DIABOLO solver in the COCAGNE platform.

For the sake of simplicity, only full assembly homogenization will be discussed here, but any homogenization coarser than the pin-by-pin assembly structure requires a "dehomogenization" process in order to reach pin-by-pin production rates for core calculation. Pin-by-pin reaction rates are reconstructed using the following formula:

$$\tau_i^f = \Sigma_i^f \; I_i^f \; A_i^f ,$$

where superscripts f and subscripts i indicate that these quantities are computed on cell i of the *fine* pin-by-pin mesh, τ is a reaction rate, Σ is the associated macroscopic cross-section. A_i^f is a form factor resulting from the assembly calculation and representing the amplitude of the flux in pin i within an assembly. I_i^f is produced by the SP_N calculation and represents the integral of the flux over the fine mesh cell i.

The objective of this paper is to develop more accurate methods for the computation of the fine flux integrals I_i^f which appear in the formula.

3. THE SP_N METHOD

In the following, fine flux integration parameters will be denoted as follows:

$\mathcal{M}^c$	coarse computational mesh for the SP_N solver,
Φ^c and J^c	arrays storing degrees of freedom (DOFs) for the flux and current solutions to the SP_N RT_k calculation on $\mathcal{M}^c$,
φ^c and $\mathbf{j}^c$	functions associated to Φ^c and J^c in the RT_k basis,
$\mathcal{M}^f$	fine mesh on which flux integration will be performed.

More generally, the following conventions will be used throughout this document:
– lower-case letters for functions (e.g. φ and $\mathbf{j}$ for the flux and the current),
– upper-case letters for arrays of discrete values (e.g. Φ and J for vectors of DOFs),
– double-stroked letters for matrices (e.g. $\mathbb{A}$).

The Simplified P_N method described in [5] leads to the solution of a system of $N+1$ equations which can be seen as $(N+1)/2$ coupled diffusion equations where the vector functions $\mathbf{j}_i$ represent the current unknowns and the scalar functions φ_i represent the flux unknowns:

$$\begin{cases} \dfrac{2i+1}{4i+1}\,\mathrm{div}\,\mathbf{j}_i(\mathbf{r}) + \Sigma_{a,2i}\varphi_i(\mathbf{r}) = S_i^{\varphi}(\mathbf{r}) - \dfrac{2i}{4i+1}\,\mathrm{div}\,\mathbf{j}_{i-1}(\mathbf{r}), \\ \dfrac{2i+1}{4i+3}\,\overrightarrow{\mathrm{grad}}\,\varphi_i(\mathbf{r}) + \Sigma_{a,2i+1}(\mathbf{r})\mathbf{j}_i(\mathbf{r}) = \mathbf{S}_i(\mathbf{r}) - \dfrac{2i+2}{4i+3}\,\overrightarrow{\mathrm{grad}}\,\varphi_{i+1}(\mathbf{r}). \end{cases} \tag{1}$$

In our implementation, each diffusion system is solved spatially by using the *mixed dual* finite element RT_k described in [6]. With this element, we get a continuous current approximation of order $k+1$ and a discontinuous flux approximation, of order k, which is well suited to describe strong flux variations at interfaces. In this paper, computations are made with RT_0, RT_1 and RT_2 elements in a 3-D Cartesian mesh. If we exclude outcoming current Degrees of Freedom (DOFs), these elements have respectively 4, 32, and 108 DOFs per cell. For the RT_0 element, there are 1 DOF for the scalar unknown and 3 incoming DOFs for the vector unknowns as shown in Figure 1.

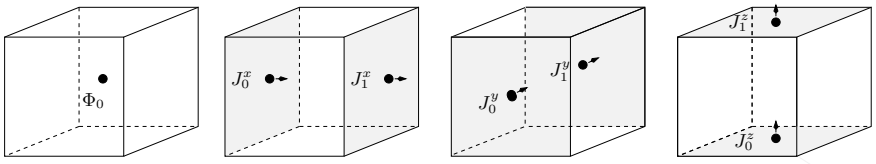


Figure 1: **Degrees of freedom for the RT_0 element.**

Flux and current basis are chosen in order to obtain a very sparse matrix system without direction coupling term in matrix $\mathbb{A}$:

$$\begin{pmatrix} \mathbb{A} & -\mathbb{B} \\ \mathbb{B}^T & \mathbb{T} \end{pmatrix} \begin{pmatrix} J \\ \Phi \end{pmatrix} = \begin{pmatrix} S_J \\ S_\Phi \end{pmatrix}. \tag{2}$$

4. FINE FLUX INTEGRATION METHODS

Denoting by φ^f an approximated neutron flux, the relevant quantities in our study are the fine flux integrals

$$I_i^f = \int_{\mathcal{M}_i^f} \varphi^f(\mathbf{r})\, d\mathbf{r}, \quad \forall \mathcal{M}_i^f \in \mathcal{M}^f, \tag{3}$$

where the mesh cells in $\mathcal{M}^f$ are denoted by $\mathcal{M}_i^f$, and $\mathbf{r} = (x, y, z)$ is the space variable. All the methods presented in the following aim at defining a fine flux approximation φ^f as accurate as possible to be injected into eq. (3).

4.1. Reference Method

Fine flux integrals I_i^f should ideally be computed using the exact solution φ to the $\mathrm{SP_N}$ problem. For the sake of simplicity, we consider in the following only $\mathrm{SP_1}$ (*i.e.* diffusion) problems of the form:

$$\begin{cases} \dfrac{1}{D}\,\mathbf{j} + \nabla\varphi = 0, & (4) \\ \mathrm{div}(\mathbf{j}) + \Sigma\,\varphi = s_\varphi, & (5) \end{cases}$$

where D denotes the diffusion coefficient, Σ represents the total macroscopic cross-section, and s_φ denotes the flux source term.

The exact flux being difficult to obtain, a practical way to obtain a flux approximation consistent with the required spatial resolution consists in solving the discrete $\mathrm{SP_N}$ system on a spatial mesh at least as fine as the pin-by-pin mesh $\mathcal{M}^f$:

$$\begin{cases} \mathbb{A}_k^f\, J_{spn,rt_k}^f - \mathbb{B}_k^f\, \Phi_{spn,rt_k}^f = 0, & (6) \\ \mathbb{B}_k^f\, J_{spn,rt_k}^f + \mathbb{T}_k^f\, \Phi_{spn,rt_k}^f = S_{\varphi,k}^f, & (7) \end{cases}$$

where $\mathbb{A}_k^f$, $\mathbb{B}_k^f$ and $\mathbb{T}_k^f$ are the standard $\mathrm{SP_N}$ matrices on fine mesh $\mathcal{M}^f$ with finite elements $\mathrm{RT_k}$, Φ_{spn,rt_k}^f and J_{spn,rt_k}^f respectively represent the arrays of DOFs for the flux and the current. The function φ_{spn,rt_k}^f associated to Φ_{spn,rt_k}^f can then be used in equation (3), yielding reference values for fine flux integrals I_i^f. This is the method used as a reference in our study.

4.2. *Direct* Integration Method

The *Direct Integration* (DI) method was originally used in COCAGNE. It consists in using the natural flux decomposition on the $\mathrm{RT_k}$ basis to compute averaged values on fine meshes. Using the above notations, this corresponds to taking $\varphi_{DI}^f = \varphi^c$ in eq. (3).

For example, if the $\mathrm{SP_N}$ resolution was done with an $\mathrm{RT_0}$ discretization, the flux is naturally represented by a staircase function on the mesh cells; this piecewise constant approximation is used to compute fine flux integrals I_i^f (see fig. 2c).

However, such a method does not appear to be very optimal as far as the $\mathrm{RT_k}$ mixed finite elements discretization is concerned. With an $\mathrm{RT_k}$ discretization, fluxes are indeed developed in a basis of order k polynomials, whereas the basis used for currents contains order $k+1$ polynomials. Only using flux DOFs in the fine flux integration method therefore completely ignores a large quantity of data contained in the current DOFs.

4.3. Methods based on Current Projection

Following the above discussion, it seems that more accurate fine flux integrals could be obtained by interpolating currents on the fine mesh in order to gain one polynomial order in the flux representation. Methods presented in section 4.3 are based on this idea.

We define the fine current $\mathbf{j}^f$ as the L^2 orthogonal projection of $\mathbf{j}^c$ on the fine mesh $\mathcal{M}^f$. In this context, the fine flux reconstruction method now aims at using $\mathbf{j}^f$ to produce as good an approximation of φ^f as possible. As already noted in section 4.1, the $(\varphi^f, \mathbf{j}^f)$ pair would ideally be as close as possible to the solution of $\mathrm{SP_N}$ problem (4)–(5).

It is worth noting that, when discretizing these equations using $\mathrm{RT_0}$ finite elements on fine mesh $\mathcal{M}^f$, the flux is projected on a basis of cell-wise constant functions. In this case, coefficients stored in the DOFs array Φ^f_{spn,rt_0} therefore exactly correspond to fine flux integrals I^f_i. The fine flux integration problem can thus be seen as the approximated solution of the $\mathrm{SP_N}$ $\mathrm{RT_0}$ problem on fine mesh $\mathcal{M}^f$:

$$\begin{cases} \mathbb{A}^f_0\, J^f - \mathbb{B}^f_0\, \Phi^f = 0, & (8) \\ {}^t\mathbb{B}^f_0\, J^f + \mathbb{T}^f_0\, \Phi^f = S^f_{\varphi,0}, & (9) \end{cases}$$

where notations are consistent with those of equations (6)–(7). For the sake of readability, when not explicitly mentioned it will always be assumed in the following that the RT order is 0, and indices will be dropped.

Since fine currents have been obtained through coarse currents projection on fine mesh $\mathcal{M}^f$, and are therefore not considered unknowns of the problem, it is clear that the above system – which now only has flux unknowns – is overdetermined, meaning that some of the equations have to be dropped in order to find a solution. This can be done by only solving the current equation (8)[†]. We will thus consider the following fine flux equation:

$$\mathbb{B}^f\, \Phi^f = \mathbb{A}^f\, J^f \tag{10}$$

where $\mathbb{A}$ and $\mathbb{B}$ are the $\mathrm{SP_N}$ $\mathrm{RT_0}$ matrices, J^f is the projected current on $\mathrm{RT_0}$ basis functions on fine mesh $\mathcal{M}^f$ and is considered to be known (through the current projection), and Φ^f is the unknown flux, identified to fine flux integrals I^f_i since it is developed on $\mathrm{RT_0}$ basis functions.

However, equation (10) still remains overdetermined: its number of unknowns is the number of flux DOFs, while its number of equations is the number of current DOFs. It will thus need yet another transformation – and approximation – to be solved. While there exist numerous ways of eliminating constraints in this system, we chose to investigate two methods which are described in the following paragraphs. Before entering into more details about these methods, we can sum up current projection-based methods by the following steps: (i) compute J^f by projecting coarse current j^c on the $\mathrm{RT_0}$ basis associated to fine mesh $\mathcal{M}^f$, (ii) build matrices $\mathbb{A}^f$ and $\mathbb{B}^f$ associated to the $\mathrm{RT_0}$ discretization of the $\mathrm{SP_N}$ problem on $\mathcal{M}^f$, (iii) compute a fine flux array Φ^f approximately fulfilling equation (10). Coefficients stored in this array will then be identified to fine flux integrals I^f.

[†]This could also be seen as the integration of a weak Fick's Law: $\mathbf{j} = -D\,\nabla\varphi$, allowing the neutron flux to be deduced from the current.

4.3.1. *Poisson* Method

A first way of transforming linear system (10) consists in left-multiplying it by ${}^t\mathbb{B}^f$, yielding the square system

$$ {}^t\mathbb{B}^f \mathbb{B}^f \Phi^f_{poisson} = {}^t\mathbb{B}^f \mathbb{A}^f J^f, \tag{11}$$

where matrix ${}^t\mathbb{B}^f\mathbb{B}^f$ exactly corresponds to the standard discretization of the Laplace operator in dimension D with a $(2D+1)$-point stencil, with homogeneous Dirichlet boundary conditions. This is of notable interest since this system thus corresponds to the discrete Poisson equation (hence the method name), for which very fast and robust resolution methods [7] are available.

The main characteristic of this method is its smoothing effect. Indeed, provided that the right-hand side is sufficiently smooth, obtained fluxes correspond to the RT_0 discretization of a twice weakly differentiable function in all directions (figure 2e).

However, left-multiplying equation (10) by ${}^t\mathbb{B}^f$ can raise problems, insofar as this matrix has non-zero nullity. The solution of equation (11) is thus not necessarily solution to problem (10). However, the kernel of matrix ${}^t\mathbb{B}^f$ is composed of constant vectors, so that homogeneous Dirichlet boundary conditions should actually enforce in practice the correctness of the solution. Further work should be done to check the validity of the method for other types of boundary conditions.

4.3.2. *StrawHat* Method

Another way of transforming linear system (10) consists in taking advantage of the structure of the matrices involved. Indeed, for a spatial problem with D dimensions, matrices $\mathbb{A}$ and $\mathbb{B}$ can be decomposed into directional blocks:

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_x & & \\ & \mathbb{A}_y & \\ & & \mathbb{A}_z \end{pmatrix}, \qquad {}^t\mathbb{B} = \begin{pmatrix} {}^t\mathbb{B}_x & {}^t\mathbb{B}_y & {}^t\mathbb{B}_z \end{pmatrix}, \qquad {}^t\mathbb{B}_d = \begin{pmatrix} 1 & & \\ -1 & \ddots & \\ & \ddots & 1 \\ & & -1 \end{pmatrix},$$

where blocks $\mathbb{A}_d$ are of size $(N_d+1)\times(N_d+1)$ and blocks ${}^t\mathbb{B}_d$ of size $(N_d+1)\times N_d$, denoting by N_d the number of mesh cells in direction d.

The *StrawHat* method aims at determining "directional" fluxes Φ^f_d fulfilling relation

$$ {}^t\mathbb{B}_d \Phi^f_d = \mathbb{A}^f J^f_d. \tag{12}$$

This system is still overdetermined, but possesses in practice a unique solution. Indeed, the form of ${}^t\mathbb{B}_d$ implies that its last row is the sum of all others. Likewise, the last coefficient of right-hand side $\mathbb{A}^f J^f_d$ is equal to the sum of all others due to the boundary conditions of SP_N problem (8)–(9). Ignoring the last equation of this system thus does not modify its solution while making it trivially invertible. Once again, further work is needed to study the validity of the method for other boundary conditions.

In the *StrawHat* method, the fine flux is defined as the arithmetic average of directional fluxes:

$$\Phi^f_{strawhat} = \frac{1}{D}\sum_{d=1}^{D} \Phi^f_d.$$

Since J_d^c was originally developed on a basis of cell-wise linear functions in direction d and staircase-shaped in other directions on the coarse mesh, each directional flux Φ_d^f will also have the same shape. Thanks to this directional averaging, the result exhibits the characteristic shape shown on figure 2d, which reminded us of a straw hat, hence the name[‡].

5. METHODS COMPARISON ON AN ANALYTICAL BENCHMARK

We present in this section a comparison of the fine flux integration above on a numerical 2-D benchmark with homogeneous materials and sinusoidal source term (this problem corresponds to Poisson's equation):

$$\forall (x,y) \in [0,1]^2, \quad D(x,y) = 1, \quad s_\varphi(x,y) = \sin(\pi\, x)\ \sin(\pi\, y).$$

All computations presented here were produced using a GNU/Octave[§] script, which solves the SP_1, RT_0 equations. A 10×10 cells mesh was used as the coarse mesh $\mathcal{M}^c$, while fine integrals were computed on a 20×20 cells fine mesh $\mathcal{M}^f$. Results are presented on figure 2. Fine flux integrals produced by the methods described above are presented in figures 2c, 2d and 2e. A fine SP_1 calculation (fig. 2b) is used as reference to evaluate errors on these fine integrals, denoted by $e_X^\varphi = I_X^f - I_{\mathrm{SP}_1}^f$ for method X. A 1-D cross-section of the error field is plotted on figure 2f, and relative L^∞ and L^2 norms of these errors are presented on the following table:

	DI	*StrawHat*	*Poisson*
$\|e^\varphi\|_{rel,\infty}$	8.2%	4.3%	0.69%
$\|e^\varphi\|_{rel,2}$	11%	5.6%	0.68%

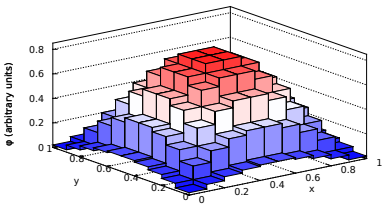
Although this benchmark problem is purely numerical, a few interesting conclusions can already be drawn from these results. First, the *Direct* Integration method seems less accurate than the StrawHat method, which is itself less accurate than the *Poisson* method. Overall the *Poisson* method reduces errors on fine flux integrals by one order of magnitude, both in L^2 and L^∞ norms.

Another interesting fact to notice is the difference between coarse and fine SP_N calculations (figs. 2a and 2b). Coarse computation results are not spatially converged, which leads to inexact neutron balances. Indeed, the flux integral over the entire domain varies by as much as 0.1% between the fine and coarse SP_N calculations on our benchmark. Fine flux integration methods based on the coarse calculation have no means to correct this error in neutron balance; their accuracy will thus be limited ultimately by this incorrect balance in the coarse calculation.

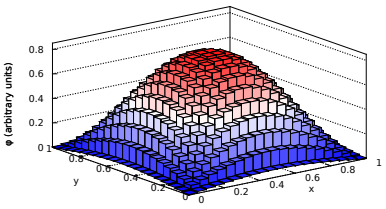
This allows us to better understand the shape of fine integral errors. Since fine flux integrals produced by the *Direct* Integration and *StrawHat* methods are much less smooth than the exact solution, errors are dominated by strong oscillations corresponding to interpolation errors. However, the *Poisson* method generates much smoother results, which in this case almost eliminates interpolation errors. Errors on the fine flux integrals produced by the *Poisson* method are thus dominated by balance errors coming from the coarse calculation, which are much smoother.

[‡]The straw hat shape is more easily recognized when using a piecewise linear representation, as opposed to the piecewise constant plot on figure 2d.

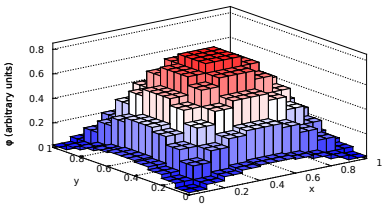
[§]www.gnu.org/software/octave/



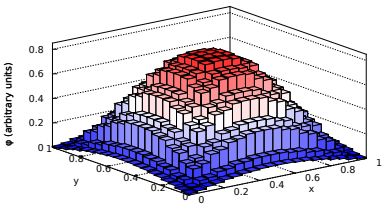
(a) Coarse SP_1 RT_0 calculation



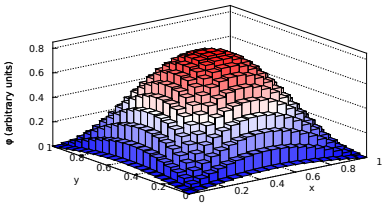
(b) Fine SP_1, RT_0 calculation, used as reference



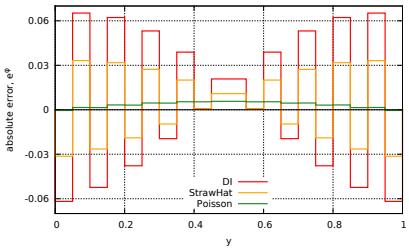
(c) Fine flux integrals with the *Direct* Integration method



(d) Fine flux integrals with the *StrawHat* method



(e) Fine flux integrals with the *Poisson* method



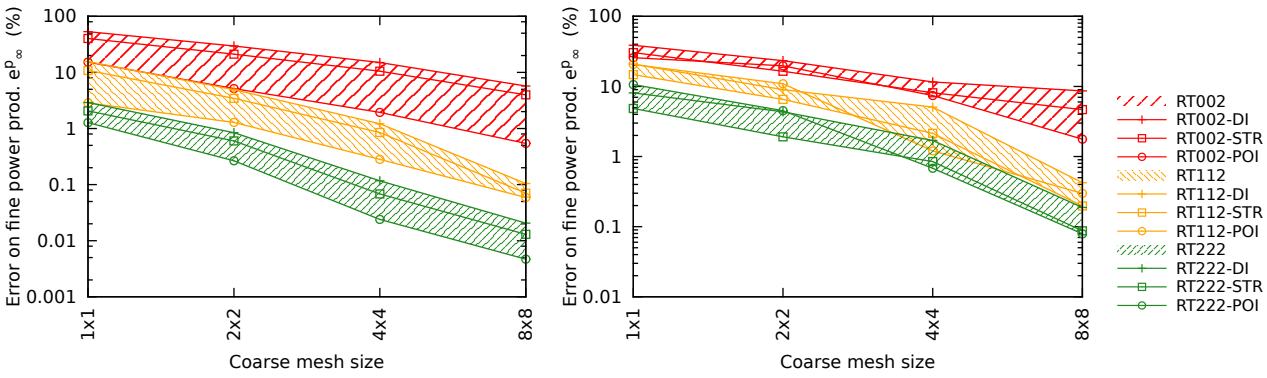
(f) 1-D cross section of the errors on fine flux integrals

Figure 2: **Methods comparison on a numerical benchmark: Poisson's equation with sinusoidal source term Errors presented on figure are obtained on a 1-D cross-section of the domain for $x = 0.5$.**

6. SPATIAL CONVERGENCE ANALYSIS

In this section we evaluate the efficiency of the two novel integration methods *Poisson* and *Strawhat* compared to the original *Direct* scheme for two different realistic 3-D PWR core configurations. The efficiency of fine flux integration (FFI) methods can be best assessed using a spatial convergence analysis of the SP_N solver. Indeed, a coarse SP_N calculation followed by an optimal FFI method should ideally produce results of as good accuracy as a finer SP_N calculation using a less elaborate FFI method. It is therefore interesting to measure the impact of FFI methods on measured convergence speeds of the SP_N solver.

All results presented below have been obtained using the same experimental set-up. All SP_N solvers are fed with a set of 2-group cross-sections which have been homogenized at the assembly level, on a $17 \times 17 \times 41$ physical mesh. Fine flux integrals are computed pin-by-pin, *i.e.* on a fine mesh $\mathcal{M}^f$ containing 17×17 cells per assembly in radial directions. The fine mesh thus contains a total of $289 \times 289 \times 41$ cells.



(a) UOX loading pattern

(b) UOX/MOX loading pattern

Figure 3: **Spatial convergence of the SP_N and FFI methods on 3-D computations.**

Reference results are obtained through an SP_1 RT_2 calculation on a reference mesh $\mathcal{M}^{ref}$ containing 32×32 cells per assembly in radial directions (it is thus finer than $\mathcal{M}^f$). Errors produced by calculations on coarser meshes are evaluated by comparison to these reference results, and based on fine power production rates, which is one of the most important physical quantities at the local scale:

$$e^p_\infty = \frac{\|P^f - P^{f,ref}\|_\infty}{\|P^{ref}\|_\infty}, \quad \text{with} \quad P^f_i = \sum_g \kappa \, \Sigma_f(g) \, I^f_i(g), \tag{13}$$

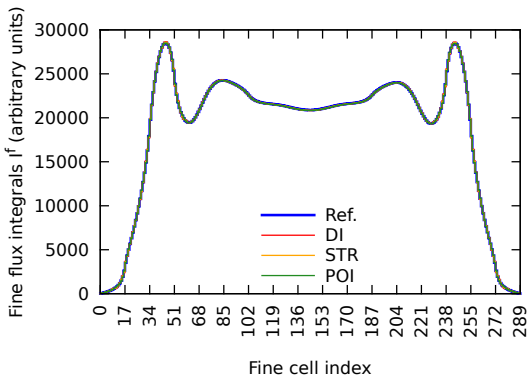
where P^f is the fine power on mesh $\mathcal{M}^f$, computed using multigroup power production cross-sections $\kappa \, \Sigma_f$ and fine flux integrals I^f_i.

All computations are run with RT_2 elements for the axial discretization (along direction z). However, three different RT_k orders will be compared for the discretization along radial directions (x and y). Solver configurations with different RT_k orders per direction will be denoted by RT_{klm} (where k, l and m are respectively the discretization order along x, y and z axes). On another hand, each of these RT_k configurations can be associated to any of the three FFI methods presented above. This gives a total of nine different solver configurations, which are all compared against the reference computation.

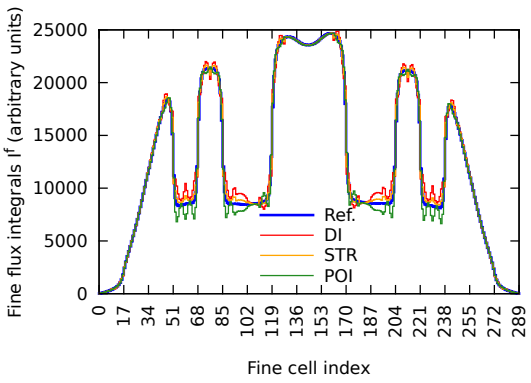
6.1. Solvers Accuracy as a Function of the Discretization

Figure 3 shows the relative error on fine production rates e^p_∞ as a function of the computational mesh, for the nine solver configurations. Three colored envelopes are used to group the computations corresponding to the same RT_k element. Within each colored area, three curves correspond each to a FFI method: *Direct* integration, *StrawHat* or *Poisson* (respectively abridged DI, STR and POI in the legend)

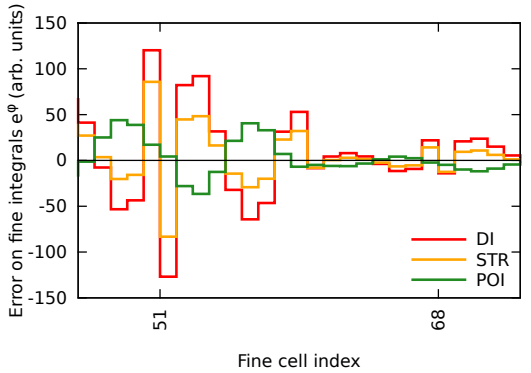
Figure 3a shows the results on the first PWR core configuration, which corresponds to a realistic UOX loading pattern. For all DIABOLO coarse computation meshes, the accuracy increases with the RT_k order. This is an expected result since the DOFs number increases with the RT_k orders. For this UOX loading pattern case, both *Poisson* and *Strawhat* integration methods improve the flux integrals accuracy compared to the original *Direct* scheme. The *Poisson* results are particularly impressive since this method often improves the pin-by-pin production rates accuracy by as much as one order of magnitude with respect to the original method.



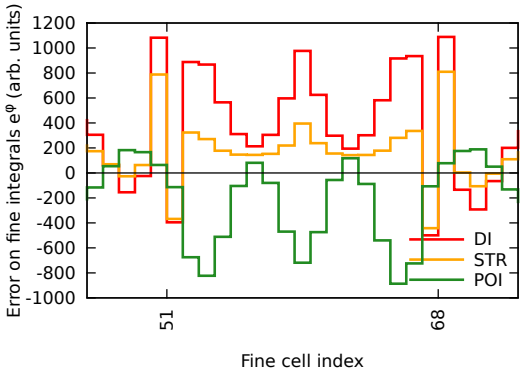
(a) UOX core: 1-D cross section of fine flux integrals



(b) UOX/MOX core: 1-D cross section of fine flux integrals



(c) UOX core: 1-D cross section of errors on I^f



(d) UOX/MOX core: 1-D cross section of errors on I^f

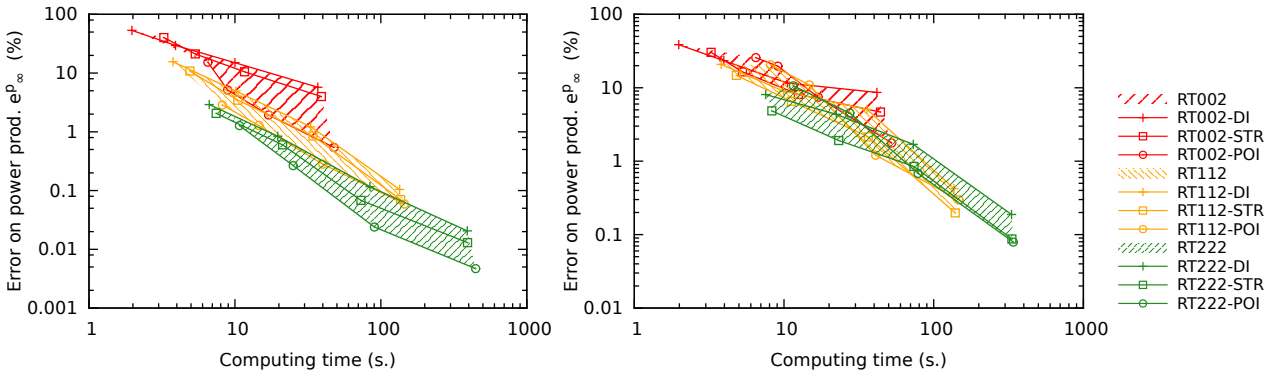
Figure 4: **1-D cross sections of Fine Flux Integrals I^f in the UOX and UOX/MOX cores. The two bottom graphs show errors on fine flux integrals, zoomed around an UOX/MOX interface.**

These very satisfactory results match those of section 5, where *Poisson* integration method was shown to be very efficient for smooth enough flux distribution.

The second core configuration, presented on figure 3b, corresponds to a UOX/MOX loading pattern. Considering the original *Direct* scheme curves, one can see that the accuracy also increases with the $\mathrm{RT_k}$ order for every considered coarse computational meshes. However, unlike the UOX case, the *Poisson* method fails to improve the *Direct* integrals accuracy for this UOX/MOX loading pattern. This behavior modification is explained by figure 4, which shows the flux evolution along a line crossing middle of the PWR core ($Z = N_Z/2$, $Y = N_Y/2$), obtained with $\mathrm{RT_{222}}$ calculations on a computational mesh containing 2×2 cells per assembly. The flux evolution exhibits strong discontinuities at the assembly interfaces that lead the *Poisson* method to produce sharp oscillations that spoil the fine flux integrals accuracy. In such cases, the *Strawhat* method usually outperforms both *Poisson* and *Direct* schemes. A maximal flux discontinuity threshold could probably be used for an *a priori* selection of the best suited integration method.

6.2. Solvers Accuracy as a Function of Computing Time

In order to prescribe guidelines for the solvers use in industrial settings, it is useful to consider the solvers accuracy as a function of computing time instead of the discretization: if a user is willing to wait for a given amount of time, which solver will produce the most accurate answer within this time frame?



(a) UOX loading pattern

(b) UOX/MOX loading pattern

Figure 5: **Accuracy of the SP_N solver and FFI methods as a function of the computing time.**

When trying to optimize the accuracy / cost ratio of an iterative solver, a crucial point is the choice of a stopping criterion for the iterative system: too large a stopping criterion will produce an inaccurate solution very fast, whereas too small a stopping criterion will uselessly iterate for a long time without improving the quality of the solution in the same proportions. In the following study, computing times were measured using an *optimal* stopping criterion ε_{opt}, defined as follows:

$$\varepsilon_{opt,\infty} = \max \left\{ \varepsilon \text{ such that } \forall\ \varepsilon' \leq \varepsilon, \frac{|e^p_\infty(\varepsilon') - e^p_\infty(0^+)|}{|e^p_\infty(0^+)|} < M \right\},$$

where $e^p_\infty(\varepsilon)$ denotes the error produced by a calculation using ε as stopping criterion, measured in L^∞ norm. $e^p_\infty(0^+)$ denotes the minimal error which would be obtained with an arbitrarily small stopping criterion, and M is an arbitrary threshold which has been set to 10% in our study. Defined in this way, the optimal stopping criterion is the largest possible value which ensures that the loss of precision with respect to the best possible precision is smaller than M.

Figure 5 presents the relative error on fine power production rates e^p_∞ as a function of computing times, following the same conventions as figure 3. Results corresponding to the UOX loading pattern (figure 5a) are very interesting: for a given CPU time, RT_2 results are always better than RT_1 ones which which always outperform the RT_0 computations. In this case, RT_2 is always the best choice except if a strong CPU time constraint excludes this discretization scheme for the coarsest $(1\times1\times1)$ mesh refinement. In other words, the accuracy increases faster than the CPU time with growing RT_k orders. Moreover, for any given RT_k order, we observe the same FFI methods hierarchy as in the spatial convergence study: the *Poisson* method outperforms *StrawHat*, which in turn presents a better accuracy / cost ratio than the *Direct* integration method.

Figure 5b however exhibits a very different behaviour on the UOX/MOX core configuration. As seen in the previous paragraph (figure 3b), increasing the RT_k order in this case does not improve the solution accuracy as much as for the UOX loading pattern. It might then happen that for a given computing time, RT_1 results are more accurate than the RT_2 calculation using the same FFI method. The same phenomenon can be observed between FFI methods for the same RT_k order: depending on the required accuracy and acceptable CPU time, the best FFI method changes. There is no clear prescription in this case, but it seems that two fundamental results remain valid: RT_1 calculations are always more accurate than RT_0 ones for a given CPU time, and the *StrawHat* method always outperforms the *Direct* integration in terms of accuracy / cost ratio.

7. CONCLUSION

DIABOLO is a Cartesian SP_N solver based on mixed dual RT_k finite elements. It is used to compute the neutron flux inside nuclear cores for k_{eff} eigenvalue problems. In order to complete complex simulations involving a large number of successive eigenvalue calculations within acceptable CPU times, DIABOLO uses computational meshes that contain a restricted number of cells (e.g. 4×4 cells per assembly). Solutions obtained on these relatively coarse computational meshes are then spatially interpolated to obtain physical quantities such as neutron production rates on finer meshes (e.g. pin-by-pin power production). This paper has described two novel methods, named *Strawhat* and *Poisson*, that allow the neutron flux to be integrated over the fine mesh cells. Unlike the original method used in COCAGNE, these new methods compute accurate fine flux integrals through the interpolation of the current DOFs of the DIABOLO's mixed dual RT_k finite elements. Assessed on two realistic 3-D PWR core configurations, these methods have been shown to bring about significant improvements over the original integration scheme. For the first PWR core configuration which corresponds to an UOX loading pattern, the *Poisson* method improves the pin-by-pin production rates accuracy by as much as one order of magnitude. For the second PWR core configuration, which corresponds to an UOX/MOX loading pattern with strong flux discontinuities at the assembly interfaces, the *Strawhat* method always outperforms *Poisson* in terms of accuracy / cost ratio and often reduces the maximal error on the pin-by-pin production rates by a factor two with respect to the original scheme.

In order to improve the usability of *Poisson* and *Strawhat*, an automatic selection of the best suited integration method depending on the core characteristics is an important issue to be addressed in the future. Additionally, we plan to extend the scope of the two novel methods to other kinds of boundary conditions (e.g. symmetry). Finally, the experimental assessment of the *Poisson* method should be extended to non-uniform computation grids in order to improve its results for strongly discontinuous flux distributions.

REFERENCES

[1] W. Kirschenmann, L. Plagne, A. Ponçot, and S. Vialle. "Parallel SPN on multi-core CPUs and many-core GPUs." *Transport Theory and Statistical Physics*, **vol. 39(2-4)**, pp. 255–281 (2011)

[2] J. Vidal, O. Litaize, D. Bernard, A. Santamarina, C. Vaglio-Gaudard, and R. Tran. "A new modeling of LWR assemblies using the APOLLO2 code package." *Proceedings of Mathematics and Computation, Supercomputing for Nuclear Applications, M&C+SNA 2007*. Monterey, CA, USA (2007)

[3] R. Sanchez, I. Zmijarevic, M. Coste-Delclaux, E. Masiello, S. Santandrea, E. Martinolli, L. Villate, N. Schwartz, and N. Guler. "Apollo2 year 2010." *Nuclear Engineering and Technology*, **vol. 42**, pp. 474–499 (2010)

[4] N. Hfaiedh and A. Santamarina. "Determination of the optimized SHEM mesh for neutron transport calculations." *Proceedings of Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Application, M&C 2005*. Avignon, France (2005)

[5] E. Gelbard. "Simplified Spherical Harmonics Equations and Their Use in Shielding Problems." Tech. rep., WAPD-T-1182 (Rev. 1), Westinghouse Electric Corp. Bettis Atomic Power Lab., Pittsburgh (1961)

[6] J. Lautard, D. Schneider, and A. Baudron. "Mixed Dual Methods for Neutronic Reactor Core Calculations in the CRONOS System." *Proc. Int. Conf. Mathematics and Computation, Reactor Physics and Environmental Analysis of Nuclear Systems, Madrid, Spain*, pp. 814–826 (1999)

[7] L. Plagne and J.-Y. Berthou. "Tensorial basis spline collocation method for Poisson's equation." *Journal of Computational Physics*, **vol. 157(2)**, pp. 419–440 (2000)