Synergies of Acceleration Methodologies for Whole-Core N/TH-Coupled Steady-State and Transient Computations

René van Geemert[1]

AREVA NP – An AREVA and Siemens Company

Abstract

AREVA NP's transient code PANBOX has been applied since the early 1980s for pursuing accurate whole-core N/TH-coupled transient computations, and has served licencing needs of several customers in the nuclear industry for many years now. Being part of the CASCADE-3D package, it features a clear link with the fuel cycle simulation program PRISM, from which it acquires core data and with which it shares the steady-state flux solver module FLUXS. In order to allow flux solution CPU-time frames of merely seconds (steady-state) and minutes (transients) on present-day 2GHz processors, numerical convergence acceleration, enabled by carefully implemented multi-level rebalancing and asymptotic extrapolation procedures, is a default standard in FLUXS as well as in the PANBOX-specific transient flux solver module FLUXT. At rebalancing level, robust preconditioned Krylov subspace methods are used for enforcing a very fast error decay, thereby enabling the optimum acceleration effect that is achievable within the bounds of the chosen hierarchy of grid levels. The computational impact due to rebalancing follows from an effected substantial suppression, at low computational cost, of low frequency nonfundamental mode components (contained in the not-yet-converged nodal diffusion iterand), whose presence would otherwise decay extremely slowly. Asymptotic extrapolation is used as an auxiliary measure, which effects a suppression of slowly decaying higher modal components as well, though with a comparatively less dramatic impact. In present-day PANBOX practise, asymptotic extrapolation is automatically ignited only when a clearly asymptotic error shape is detected in spite of a systematic use of rebalancing between full core nodal diffusion sweeps. Results are presented for cases featuring different core conditions, from which a clear confirmation emerges that the two acceleration measures are complementary. A particular benefit arises for tedious cases where asymptotic extrapolation does cover slowly converging modes that are still untouched by coarse mesh rebalancing.

Keywords: steady-state and transient whole-core computations, nodal expansion method, multi-level coarse mesh rebalancing, asymptotic extrapolation, Krylov subspace methods

1. Introduction

1.1 The PANBOX N/TH-coupled Steady-state and Transient Core Simulation Tool

AREVA NP's N/TH-coupled core steady-state and transient program PANBOX [1], which is embedded in the CASCADE-3D integral core simulation package [2] and features COBRA 3-CP [3] as thermal-hydraulics module, enables accurate whole-core three-dimensional N/TH-coupled transient computations within practical time frames. Its steady-state and transient numerical neutron physics models are characterized by a nodal diffusion approach based on the NEM (Nodal

[1]Correspondence address: AREVA NP GmbH, Freyeslebenstrasse 1, 91058 Erlangen (Germany). Tel. 0049-9131-1896745, fax 0049-9131-1894045, e-mail: rene.vangeemert@areva.com

Expansion Method) methodology [4,5]. The latter is a well-known and consolidated numerical basis for enabling accurate as well as computationally efficient whole-core steady-state and transient computations for applications in the nuclear industry. Its iterations are usually defined, for common size reactors, at the level of several thousands of nodes with, in the *absence* of an efficient acceleration methodology, a typically very poor error decay between successive whole-core NEM nodal diffusion sweeps (in particular for cases featuring a high degree of in-core heterogeneity). Therefore, realistic three-dimensional nodal whole-core transient computations would, without efficient acceleration, lead to CPU time requirements that would still be inacceptable on the linear timescale of industrial activity. In the CASCADE-3D integral core simulation package, this problem is avoided through the default use of *coarse mesh rebalancing* [6,7,8,9] and *asymptotic extrapolation techniques* [8] in the multi-group steady-state and transient flux solution modules *FLUXS* (embedded in PANBOX and PRISM) and *FLUXT* (embedded in PANBOX only).

This paper presents information on the observed impact of coarse mesh rebalancing and asymptotic extrapolation for acceleration of steady-state and transient whole-core computations pursued with PANBOX. Results are presented for cases featuring different core conditions, from which a clear confirmation emerges that the two acceleration measures are complementary. Furthermore, a steady-state example will be shown which is characterized by a combination of high in-core heterogeneity and cold reactor conditions, for which asymptotic extrapolation does cover slowly converging modes that are still untouched by coarse mesh rebalancing, thereby effecting a noticeable additional acceleration effect.

1.2 Coarse Mesh Rebalancing in PANBOX and PRISM

The basic details of coarse mesh rebalancing for NEM-like equations have been described in a number of previously released publications [9,10]. The applicability of the specific rebalancing approach adopted in FLUXS and FLUXT is conveniently *invariant* with respect to the exact setup of the NEM equations, as long as a clear nodal balance equation equivalent to the diffusion equation (thus featuring nodal fluxes, incurrents and outcurrents) can be identified and used as a basis for restriction to a coarser space-energy mesh. This means that the available FLUXS/T rebalancing routines can be used regardless of whether one solves second- or higher-order polynomial NEM equations [4,5], or semi-analytical [11] variants of the NEM equation (with the latter being related to ANM/AFEN [12,13]). All of these NEM-variants are available in FLUXS and FLUXT (and hence in PANBOX and PRISM as well), with the semi-analytical variant being the recommended standard for steady-state and transient production computations.

Coarse mesh rebalancing, which was proposed originally as a ”variational acceleration technique” for basic diffusion iterations by Wachpress [6] in the 1960s, provides a suppression of otherwise slowly decaying low-frequency nonfundamental mode components in the not-yet-converged diffusion iterand. This is realized through a systematic multiplicative correction of nodal fluxes and interface outcurrents, prior to each full core NEM diffusion sweep, with iteratively obtained ratios between coarse mesh rebalanced and prebalanced fluxes [7]. These ratios, conventionally referred to as *driving factors*, emerge from restarted preconditioned Krylov subspace procedures defined at the computationally substantially cheaper coarse mesh equation level and serve to push down the long wavelength / lower frequency nonfundamental mode components of the NEM residual to a very significant extent. The acceleration effect realized in this way depends on the numerical

proximity of the highest coarse mesh level in the multi-level hierarchy to the full-core NEM diffusion level. Because of the multiplicative rather than additive nature of the correction, coarse mesh rebalancing clearly belongs to the class of *nonlinear acceleration approaches* [14]. As reported in subsection 4.1 of this document, speed-up factors higher than 10, compared to unaccelerated computations, are common for typical steady-state and transient calculations at AREVA NP.

Analogous nonlinear acceleration approaches, which are roughly similar in numerical nature to the ones applied in FLUXS/T multi-level rebalancing, have in fact also been applied to whole-core (typically two-dimensional) *multi-group transport computations* [15,16]. Here, the coarse mesh restriction usually consists of a reduction in angular information and/or projecting the transport equation towards a coarse-group diffusion-like equation level of substantially lower dimensionality. Thus, the computational cost of pursuing several coarse mesh iterations is similarly negligible compared to the cost of a single whole-core transport sweep. Substantial CPU-time reductions were reported for computationally demanding two-dimensional whole-core MOC calculations in CASMO-4E [17] and APOLLO [18].

1.3 Asymptotic Extrapolation as a Auxiliary Acceleration Mechanism for Nodal Steady-State and Transient Whole-Core Computations

Whereas rebalancing provides the principal convergence acceleration effect for both steady-state and transient computations, occasional asymptotic extrapolations can still, under certain conditions, provide substantially more than merely a second-order additional acceleration effect. In particular for cases where convergence is still relatively slow in spite of using multi-level rebalancing with well-converged driving factors (characterized for example by in-core heterogeneity combined with cold reactor conditions), the additional use of occasional asymptotic extrapolations (ignited upon automated detections of a clearly asymptotic error shape) can provide a significant additional convergence boost. This document will report an example of this in subsection 4.1.

Furthermore, asymptotic extrapolation serves as an *auxiliary acceleration* mechanism for computationally more lengthy transient computations. Cases where this acceleration backup was activated automatically at a certain point were usually characterized by a conclusion *a posteriori* that the pursued rebalancing iterations had not been pursued to sufficient convergence, due to which the pursued prolongations effected only part of the maximum possible suppression of the slowly decaying low-frequency mode components in the iterands. These scenarios (an example will be shown in subsection 4.1) typically led to detected asymptotic phases in a nonoptimal error decay for which the automated asymptotic extrapolation measure provided a partial backup remedy. However, with the present-day availability in FLUXS/T of application-tailored heavily preconditioned Krylov subspace methods (these are discussed in subsection 2.4), which enable a very fast convergence of the coarse mesh equations (as a substitute for older, more classical and less robust source iteration methods), these particular scenarios have been practically eliminated. Nevertheless, the added value of asymptotic extrapolation remains high for heterogeneous core computations where the rebalancing procedure does not catch the slowly converging mode-components in the iterand sufficiently in spite of well-converged driving factors. Being an auxiliary mechanism which is activated only upon detection of asymptotic error decay according to strict criteria, the use of the asymptotic error shape detector in PANBOX is therefore still recommended for all steady-state and transient computations.

2. Nonlinear Acceleration by Multi-level Rebalancing in FLUXS/T

2.1 FLUXS/T coarse mesh equations

The coarse mesh equations in PANBOX/PRISM follows directly from the standard NEM representation of the diffusion equation and its spatial and energy-group integration over a Cartesian volume V_m with intranodal dimensions a_x^m, a_y^m, a_z^m. This leads to a projection of neutronic disbalance towards a coarser mesh level where it can subsequently be suppressed quasi-artificially by the definition and cheap iterative solution of multiplicative correction factors (for nodal fluxes and outcurrents) that convert the coarse mesh disbalance into a coarse mesh balance. In all available NEM-variants, a multi-group nodal balance equation of the form

$$\left(\Sigma_{ag}^m + \sum_{g' \neq g}^{G} \Sigma_{g'g}^m \right) \phi_g^m + \sum_{m'(m)} \frac{1}{a_{mm'}} \left(j_g^{m' \leftarrow m} - j_g^{m \leftarrow m'} \right) = \frac{1}{\kappa_{eff}} \chi_g \sum_{g'=1}^{G} \Sigma_{fg'}^m \phi_{g'}^m + \sum_{g' \neq g}^{G} \Sigma_{gg'}^m \phi_{g'}^m + Q_g^m \qquad \forall\, g \tag{1}$$

with the meanings of the different symbols corresponding to notations common in the reactor physics literature, can be identified for either a steady-state or a transient computation. The sum $\sum_{m'(m)}$ indicates a sum over all Cartesian neighbour nodes directly adjacent to node m. For steady-state computations on self-sustaining systems (e.g. critical reactors), the isotropic source term Q_g^m is usually zero, so that effectively a *large sparse eigensystem* is solved (optionally with a criticality search for enforcing $\kappa_{eff} = 1$). For transients, the computation becomes numerically equivalent to solving a slightly subcritical system with $1/\kappa_{eff}$ replaced by a fixed unity multiplier instead of an eigenvalue, and a nonzero source term Q_g^m.

The NEM-methodology supplies numerically convenient relationships between nodal flux and interface-averaged in- and outcurrents [4,5] which, combined with Eq.(1), provide a tractable basis for the iterative solution of the whole-core steady-state and transient fluxes and interface currents. In order to enable a high efficiency in the arrangement of the iteration loops, as well as a numerical advantage through the realization of a Gauss-Seidel setup, FLUXS and FLUXT use a chessboard *red-black* ordering of spatial nodes such that incurrents of red nodes can be associated directly with outcurrents of black nodes, and vice versa. This means that the NEM-sweeps in FLUXS and FLUXT typically consist of successive sweeps over the red and black part of the spatial grid.

Depending on the order of the applied intranodal expansion, higher-order expansion coefficients are co-solved through weighted residual (WR) procedures [4,5], and self-evidently these co-determine the fluxes and interface currents as the outcome of the integral solution process. Nevertheless, they do not explicitly appear in Eq.(1) as long as this equation features merely incurrents and outcurrents explicitly, without either of them being implicitly eliminated (which would then introduce higher-order coefficients as source terms in the nodal balance equation). This means that Eq.(1) constitutes a NEM option-invariant basis from which a generically applicable rebalancing approach follows straightforwardly.

2.2 Restriction, Rebalancing and Prolongation

The default rebalancing equation in PANBOX features the definition of driving factors per nodal volume, to be used for corrective multiplication (often referred to as *prolongation* [9,14]) of 3D NEM nodal fluxes and nodal interface outcurrents. The driving factor array $\underline{\mathrm{d}}$ is computed as the solution of a lower-dimensional dynamic nodal balance equation resulting from restricting the different terms in the not-yet-balanced NEM equation to a coarsened space-energy mesh. This restriction typically involves a volume-integration of different process rates and the computation of Cartesian surface-integrated internodal coupling terms (connected to partial interface currents), followed by a summation over all energy groups. The result is a simplified equation of substantially lower dimensionality that yet captures [7] most of the otherwise slowly decaying nonfundamental low-frequency 3D modal components in the NEM iterand.

Following the corrective multiplication of the NEM fluxes and currents with the obtained driving factors, the effected improvement in flux error decay rate depends on the numerical proximity of the highest-level rebalancing equation with respect to the NEM equation. In the absence of acceleration, these modes will damp out only very slowly, since their associated eigenvalues are close to the fundamental eigenvalue. Due to a typically very strong nonlinear acceleration effect effected by rebalancing, a usually substantial convergence speed-up results. The default rebalancing equation can be derived directly by volume integration over coarse space node k (this node being, for the first rebalancing level, usually the same as the node over which the NEM polynomials are defined) and summing over all energy groups g (thereby eliminating spectral scattering terms that systematically add up to zero), leading to the projected equation

$$\left(\mathrm{R_k} + \sum_{\mathrm{k'(k)}} \mathrm{J_{k'\leftarrow k}} \right) \mathrm{d_k} - \sum_{\mathrm{k'(k)}} \mathrm{J_{k\leftarrow k'}}\, \mathrm{d_{k'}} = \frac{1}{\kappa}\, \mathrm{F_k}\, \mathrm{d_k} + \mathrm{Q_k} \tag{2}$$

with k'(k) indicating a sum over directly adjacent neighbours of k, κ the inverse of the coarse mesh eigenvalue and the rebalancing operators derived from restricted NEM equation terms as

$$\begin{aligned}
\mathrm{R_k} &= \textstyle\sum_{\mathrm{m\in k}} \mathrm{V_m} \sum_{\mathrm{g=1}}^{\mathrm{G}} \Sigma_{\mathrm{ag}}^{\mathrm{m}}\, \phi_{\mathrm{g}}^{\mathrm{m}} \\
\mathrm{F_k} &= \textstyle\sum_{\mathrm{m\in k}} \mathrm{V_m} \sum_{\mathrm{g=1}}^{\mathrm{G}} \nu\Sigma_{\mathrm{fg}}^{\mathrm{m}}\, \phi_{\mathrm{g}}^{\mathrm{m}} \\
\mathrm{Q_k} &= \textstyle\sum_{\mathrm{m\in k}} \mathrm{V_m} \sum_{\mathrm{g=1}}^{\mathrm{G}} \mathrm{Q_g^m} \\
\mathrm{J_{k\leftarrow k'}} &= \textstyle\sum_{\mathrm{m,m'}\in\Omega_{\mathrm{k\cup k'}}} \frac{\mathrm{V_m}}{\mathrm{a_{m,m'}}}\, \mathrm{j_{g,m\leftarrow m'}}
\end{aligned} \tag{3}$$

In general operator notation, Eq.(2) can be written as

$$\mathcal{M}\underline{\mathrm{d}} = \frac{1}{\kappa}\mathcal{F}\underline{\mathrm{d}} + \underline{\mathrm{Q}} \tag{4}$$

It is the driving factor array $\underline{\mathrm{d}}$ which, as an iteratively solved vector, pushes the convergence of the NEM iterand in a nonlinear way through enforced minimization of the coarse mesh-projected residual associated with Eq.(2). As argued already in subsection 2.1, the computational cost of pursuing iteration steps at rebalancing level is typically very small compared to one whole-core NEM red-black sweep. Prior to the next whole-core NEM sweep, the multi-group fluxes and outcurrents are multiplied correctively with the obtained driving factors (which is a nonlinear acceleration measure),

$$\mathbf{P}\,[\,\mathrm{d_m}\,]\;:\;\begin{pmatrix}\phi_{\mathrm{g}}^{\mathrm{m}}\\ \\ \mathrm{j}_{\mathrm{g}}^{\mathrm{m}'\leftarrow\mathrm{m}}\end{pmatrix}\;\longrightarrow\;\begin{pmatrix}\mathrm{d_m}\;\phi_{\mathrm{g}}^{\mathrm{m}}\\ \\ \mathrm{d_m}\;\mathrm{j}_{\mathrm{g}}^{\mathrm{m}'\leftarrow\mathrm{m}}\end{pmatrix}\;\;\forall\;\mathrm{g}\,,\;\mathrm{m}'(\mathrm{m}) \tag{5}$$

In case of higher NEM variants, the higher-order NEM coefficients are co-multiplied as well. The flux error decay rate enhancement effected by the prolongation usually depends on the realized degree of coarse mesh residual minimization at rebalancing level and on whether a highly homogeneous (typically steady-state) or highly heterogeneous (typically transient) case is computed.

Whereas for most of the more homogeneous cases one can observe a rather early saturation of NEM error decay rate enhancement vs. coarse mesh residual minimization, this is quite different for transients featuring higher degrees of in-core heterogeneity. The latter require a substantially stronger coarse mesh residual minimization for reaching the maximum flux error decay rate enhancement that can be reached within the the bounds of the chosen rebalancing concept. This applies in particular for lower NEM error regimes where nodal process rate disbalances are small already, such that the driving factors will in fact be very close to unity. Specifically here, it pays off to have a robust Krylov rebalancing iteration approach that guarantees a fast coarse mesh residual decay also in such lower error regimes. Self-evidently, $\underline{\mathrm{d}}$ will converge to a flat unity distribution upon final convergence of the nodal diffusion iterands.

In subsection 2.3, the FLUXS/T *multi-level* rebalancing approach, which typically provides the best computational efficiency for steady-state whole-core computations, will be discussed briefly. According to this scenario, several coarse mesh levels are used, and for typical steady-state cases merely few Gauss-Seidel iteration steps per level suffice for effecting the maximum possible prolongation-induced flux error decay enhancement.

In subsection 2.4, the concept of applying restarted preconditioned Krylov (i.e. Bi-CGSTAB [19,20]) source iteration is described, which offers the ability of enforcing a coarse mesh rebalancing residual as low as necessary to get an optimum acceleration of NEM whole-core transient computations. With the former being a more classical approach which works well enough in practise, the latter corresponds to a more recent implementation which offers the advantage of guaranteeing a significantly faster asymptotic error decay. This is particularly advantageous in the lower error regimes where the error decay effected by Gauss-Seidel iterations typically tends to lose momentum in spite of occasional extrapolations. The restarted preconditioned Krylov (Bi-CGSTAB) source iteration option is, as will be argued in section 2.4 and shown in section 4.1, therefore indeed the preferred option for pursuing whole-core transient computations with excellent convergence accuracy within mimimized computational time frames.

2.3 Multi-level Rebalancing

With the spatial grid in the coarsened space-energy mesh being usually the same as the spatial grid for the NEM iterations (with the main dimension reduction associated with the implicit elimination of currents and the integration over all energy groups), and the error decay ratio of unaccelerated Gauss-Seidel rebalancing restarted source iterations typically still being poor at this level as well, it can certainly pay off to also project this equation to a further coarsened space-energy mesh (this time featuring a genuine coarsening of the spatial grid), leading to the iterative solution of a lower-dimensional driving factor array $\underline{\mathcal{D}}$ for prolongation (=multiplicative correction) of the finer array $\underline{\mathrm{d}}$. This approach leads to a *multi-level V-cycle design*, featuring a hierarchy of M + 1 rebalancing levels and driving factor arrays $\underline{\mathrm{d}}, \underline{\mathcal{D}}^{(\mathrm{L})}, \mathrm{L} = 1, ..., \mathrm{M}$, which is known to be very effective for suppressing higher modal components in the long wavelength range at very low computational cost. In FLUXS/T, the number of applied rebalancing levels is determined through an automated procedure, with a hierarchy of up to six coarse mesh levels (the lowest one corresponding to axial rebalancing) being common for most of the relevant whole-core steady-state and transient computations. Whereas the finest-level rebalancing equation can be written in terms of Eq.(2), restriction to coarser grid levels leads to a hierarchy of M coarser grid levels L (L=1,...,M) for which the rebalancing equations can be written as

$$\left(\mathrm{R}_{\mathrm{K}}^{(\mathrm{L})} + \sum_{\mathrm{K}'(\mathrm{K})} \mathrm{J}_{\mathrm{K}'\leftarrow\mathrm{K}}^{(\mathrm{L})} \right) \mathcal{D}_{\mathrm{K}}^{(\mathrm{L})} - \sum_{\mathrm{K}'(\mathrm{K})} \mathrm{J}_{\mathrm{K}\leftarrow\mathrm{K}'}^{(\mathrm{L})} \mathcal{D}_{\mathrm{K}'}^{(\mathrm{L})} = \frac{1}{\kappa^{(\mathrm{L})}} \mathrm{F}_{\mathrm{K}}^{(\mathrm{L})} \mathcal{D}_{\mathrm{K}}^{(\mathrm{L})} + \mathrm{Q}_{\mathrm{K}}^{(\mathrm{L})} \tag{6}$$

with

$$\begin{aligned}
\mathrm{R}_{\mathrm{K}}^{(\mathrm{L})} &= \textstyle\sum_{\mathrm{k}\in\mathrm{K}} \mathrm{V}_{\mathrm{k}} \mathrm{R}_{\mathrm{k}}^{(\mathrm{L}-1)} \mathcal{D}_{\mathrm{k}}^{(\mathrm{L}-1)} \\
\mathrm{F}_{\mathrm{K}}^{(\mathrm{L})} &= \textstyle\sum_{\mathrm{k}\in\mathrm{K}} \mathrm{V}_{\mathrm{k}} \mathrm{F}_{\mathrm{k}}^{(\mathrm{L}-1)} \mathcal{D}_{\mathrm{k}}^{(\mathrm{L}-1)} \\
\mathrm{Q}_{\mathrm{K}}^{(\mathrm{L})} &= \textstyle\sum_{\mathrm{k}\in\mathrm{K}} \mathrm{V}_{\mathrm{k}} \mathrm{Q}_{\mathrm{k}}^{(\mathrm{L}-1)} \mathcal{D}_{\mathrm{k}}^{(\mathrm{L}-1)} \\
\mathrm{J}_{\mathrm{K}\leftarrow\mathrm{K}'}^{(\mathrm{L})} &= \textstyle\sum_{\mathrm{k},\mathrm{k}'\in\Omega_{\mathrm{K}\cup\mathrm{K}'}} \frac{\mathrm{V}_{\mathrm{k}}}{\mathrm{a}_{\mathrm{k},\mathrm{k}'})} \mathrm{J}_{\mathrm{k}\leftarrow\mathrm{k}'} \mathcal{D}_{\mathrm{k}'}^{(\mathrm{L}-1)}
\end{aligned} \tag{7}$$

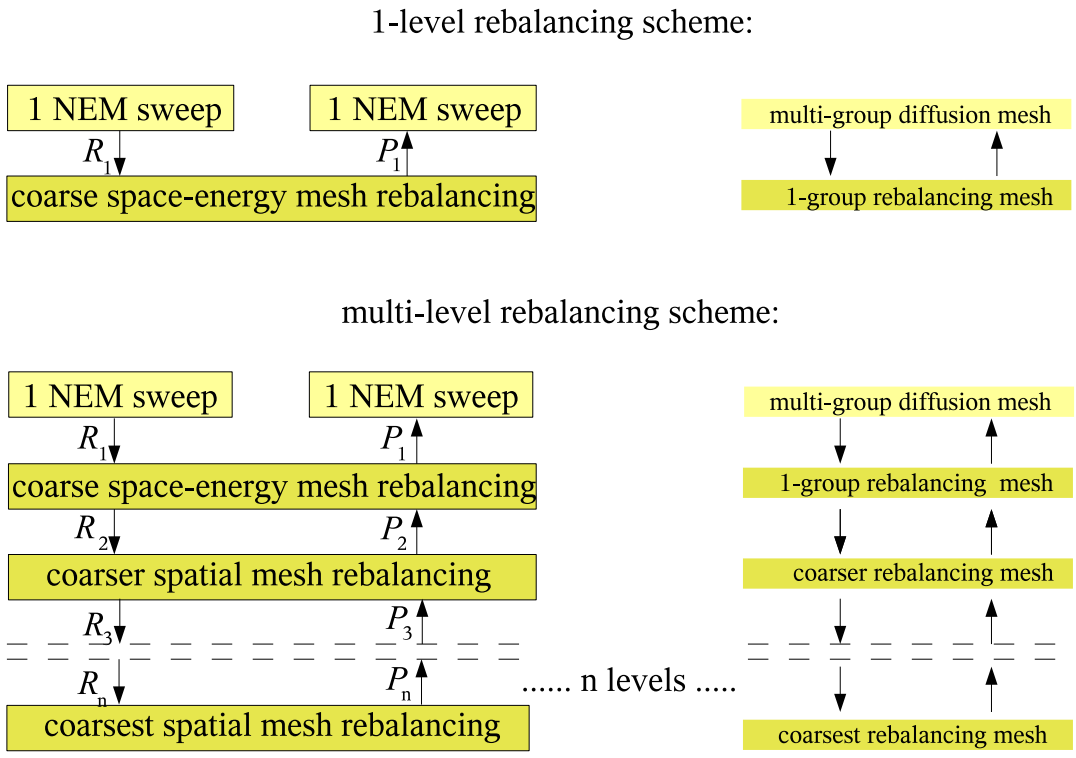
and with $\underline{\mathcal{D}}^{(0)} \equiv \underline{\mathrm{d}}$. The default V-cycle approach is characterized by a downward part where usually merely one iterative rebalancing step per level L is pursued prior to the restriction to the next coarsened space energy mesh level L+1, until the lowest, coarsest rebalancing level has been reached. At the lowest (and computationally cheapest) rebalancing level, as many iterations as needed for reaching a very low coarse mesh residual are pursued, followed by the upward part of the V-cycle. In this upward part, again a few iterative steps are pursued per lower level L+1 prior to the *prolongation* of the higher, less coarse level L iterand which, with coarse nodes K comprising less coarse nodes k, can be written as

$$\mathcal{D}_{\mathrm{k}}^{(\mathrm{L})} := \mathcal{D}_{\mathrm{K}}^{(\mathrm{L}+1)} \mathcal{D}_{\mathrm{k}}^{(\mathrm{L})} \quad , \ \forall\, \mathrm{k} \in \mathrm{K} \tag{8}$$

Subsequently, a few iteration steps are pursued at level L before prolongating the iterand at the next, finer level L-1, and so on. The custom of pursuing a few instead of merely one iterative step

per rebalancing level is recommended for the upward part of the rebalancing cycle, as prolongation from a coarser mesh to a less coarse mesh usually also excites [7] modest disturbances in the high frequency eigenmode range. These are subsequently smoothed out very quickly by pursuing a few iteration steps per rebalancing level prior to prolongation of the iterand at the next level in the multigrid hierarchy, thereby fulfilling the necessity of a smooth coarse mesh residual [19]. Self-evidently, as the NEM-iterand converges to the exact solution of the NEM transient equation for the time step in question, all driving factor arrays gradually converge to flat unity distributions. The rebalancing multi-level cycle featuring interlevel restrictions R_n and prolongations P_n (illustrated in Fig.1), is concluded with the prolongation of the nodal fluxes and interface currents (and higher NEM coefficients) of all energy groups according to Eq.(5), using the driving factors obtained at the highest rebalancing level, prior to the next whole-core NEM diffusion sweep. The total computational cost associated with a full rebalancing cycle is very low compared to a single whole-core NEM-sweep. Furthermore, the lower the rebalancing level, the lower the grid dimension of the associated coarse rebalancing mesh, and hence the lower also the computational cost connected with restrictions, rebalancing iterations and prolongations. Applying merely a single NEM-sweep between rebalancing cycles usually leads to the best overall computational performance.

Fig.1. Illustration of rebalancing cycles over multiple grid levels.



In N/TH-coupled steady-state or transient computations, periodic cross-section updates occur during the N-TH iterations, through periodic calls to the TH-module COBRA 3-CP between variable numbers of whole-core nodal diffusion NEM sweeps.

From a mathematical physics viewpoint, the rebalancing cycle approach is a fast way for the effects of disbalances, occurring on one part of the core and influencing the other parts through coupled neutron physics, to propagate numerically through the whole system as part of the common coupled physics interplay during the iterative solution process. Similar multigrid acceleration approaches for computing not only the fundamental but also higher 3D modes (typically related with mechanical stability studies) are of relevance in aviation and car industry (i.e. the design of planes and cars), where usually sparse eigensystems of even much higher dimensions follow from highly detailed finite element models of the pursued designs [20].

This multi-level coarse mesh rebalancing (MLCMR) approach has been available for acceleration of CASCADE-3D (i.e. PRISM and PANBOX) steady-state and transient whole-core computations for more than two decades now. Different rebalancing variants are offered, with differences pertaining to the definition of the highest rebalancing grid, the number of rebalancing levels, the rebalancing iteration method and the number of iterative steps pursued per rebalancing level. For example, a mere two-level scheme is available as well, featuring only the NEM level plus a first-and-only rebalancing level. For this two-level scheme, iterations are pursued using either a source iteration method or a preconditioned Krylov subspace method, with a variable stopping criterium related to the current NEM error.

2.4 Application of preconditioned Krylov subspace methods for rebalancing iterations

Since the palettes of relevant cases include transients featuring substantial dynamic local gradients in volumetric material composition (such as rod drop or rod ejection cases), the benefit of applying Bi-CGSTAB (Bi-Conjugate Gradient Stabilized) [21,22] at the level of our rebalancing iterations, with the specific aim of improving computational efficiency specifically for such dynamic heterogeneity scenarios, was investigated and confirmed. The Bi-CGSTAB algorithm belongs to the general class of Krylov subspace methods [22] which incorporates also the Conjugate Gradient (CG) and the Generalized Minimized Residual (GMRES) method. This algorithm is one of the later additions to the class of Krylov subspace methods, as it was published in 1992 and is specifically tailored for robust, efficient and fast solution of *nonsymmetric* sparse linear systems.

The Bi-CGSTAB algorithm can just as easily be to put into code as the more classical Conjugate Gradient algorithm. It is therefore not surprising that a number of other reactor simulation tools [23,24,25] already incorporate its use for robust and fast solution of the inner part of a typical outer-inner iteration structure for solving multi-dimensional diffusion or transport equations, though typically not as part of a synergy with multi-level coarse mesh rebalancing the way it was recently done in FLUXS/T for PANBOX/PRISM applications.

The principal benefits of the effected synergy between multi-level coarse mesh rebalancing and Bi-CGSTAB, in comparison with the combination of coarse mesh rebalancing and more classical iteration approaches at rebalancing level (i.e. Gauss-Seidel restarted source iterations with occasional asymptotic extrapolations) clearly consisted of further improved numerical stability and significantly faster convergence of the driving factors in dynamic heterogeneity scenarios. This hence led to systematically better converged driving factor arrays for these cases, resulting finally in substantially lower numbers of required V-cycles per transient time step and thus a significant CPU-time reduction for the total whole-core three-dimensional transient computations.

While the Bi-CGSTAB algorithm for solving a nonsymmetric linear system $\mathbf{A}$x=b can basically be used as a cook-book recipe which, as mentioned above, can be put into code very easily, the principal challenges are related to the requirement of having a *well-preconditioned* linear system $\tilde{\mathbf{A}}\mathrm{x} = \tilde{\mathrm{b}}$ in order to enable Bi-CGSTAB to offer an optimum efficiency. Since a good preconditioning typically means an explicit or implicit strong diagonal dominance for the preconditioned operator $\tilde{\mathbf{A}}$, the art is to find a (usually implicit) preconditioned form that realizes an optimum trade-off between (i) an effected sufficient degree of diagonal dominance and (ii) a limited computational penalty associated with the systematic implicit preconditioning measure per step in the Bi-CGSTAB algorithm. For our rebalancing form, a number of algebraic manipulations led to an well-preconditioned implicit system featuring the above-mentioned optimum trade-off, which enabled the setup of a heavily-preconditioned outer-inner iterative system equation of the form

$$\tilde{\mathbf{\Theta}}\ \underline{\mathrm{d}}^{(\mathrm{n})} = \underline{\tilde{\mathrm{s}}}^{(\mathrm{n})} + \underline{\mathrm{q}} \tag{9}$$

with the source term $\underline{\tilde{\mathrm{s}}}^{(\mathrm{n})}$ determined by $\underline{\tilde{\mathrm{s}}}^{(\mathrm{n})} = \tilde{\mathbf{S}}\ \underline{\mathrm{d}}^{(\mathrm{n}-1)}$. This preconditioned form features the property that most of the coarse-mesh-projected neutron physics (i.e. absorption, migration, production through fission) is already concentrated in the nonsingular implicit operator $\tilde{\mathbf{\Theta}}$ on the left (which consists of a fixed sequence of a limited number of operations for enhancement of diagonal dominance) and with correspondingly small restarted (and implicitly preconditioned) source terms $\underline{\tilde{\mathrm{s}}}$ determined by the small-norm operator $\tilde{\mathbf{S}}$. Self-evidently, the source term $\underline{\mathrm{q}}$ vanishes for steady-state cases. For the linear inner systems $\tilde{\mathbf{\Theta}}\ \underline{\mathrm{d}} = \underline{\tilde{\mathrm{s}}} + \underline{\mathrm{q}}$, the Bi-CGSTAB procedure is systematically truncated after merely a few steps, yielding an improved $\underline{\mathrm{d}}$. Following the corresponding update of the preconditioned source $\underline{\tilde{\mathrm{s}}}$, this limited Bi-CGSTAB procedure is repeated until the source $\underline{\tilde{\mathrm{s}}}$ (and, correspondly, the driving factor array $\underline{\mathrm{d}}$) is converged. With most of the coarse-mesh-projected neutron physics being concentrated already in the nonsingular implicit operator $\tilde{\mathbf{\Theta}}$, typically very few outer iterative steps are needed for acquiring a practically exact solution of the rebalancing equation. The final combination of few needed outer iterative steps and systematically few inner iterative steps leads to a very high computational efficiency. The Bi-CGSTAB sequence, truncated after N steps (with N maximally 4 in our applications), can be written as

$$\begin{array}{ll}
1. & \mathrm{r}_0 := \tilde{\mathrm{s}} - \tilde{\mathbf{\Theta}}\ \mathrm{d}_0 \quad , \quad \mathrm{r}^* = \mathrm{r}_0 \\
2. & \mathrm{p}_0 := \mathrm{r}_0 \ , \\
3. & \mathrm{do\ i} = 0, 1, ..., \mathrm{N} \\
4. & \alpha_\mathrm{i} := \frac{(\mathrm{r}^*, \mathrm{r}_\mathrm{i})}{(\mathrm{r}^*, \tilde{\mathbf{\Theta}}\ \mathrm{p}_\mathrm{i})} \ , \\
5. & \mathrm{s}_\mathrm{i} := \mathrm{r}_\mathrm{i} - \alpha_\mathrm{i} \tilde{\mathbf{\Theta}}\ \mathrm{p}_\mathrm{i} \ , \\
6. & \omega_\mathrm{i} := \frac{(\tilde{\mathbf{\Theta}}\ \mathrm{s}_\mathrm{i}, \mathrm{s}_\mathrm{i})}{(\tilde{\mathbf{\Theta}}\ \mathrm{s}_\mathrm{i}, \tilde{\mathbf{\Theta}}\ \mathrm{s}_\mathrm{i})} \ , \\
7. & \mathrm{d}_{\mathrm{i}+1} := \mathrm{d}_\mathrm{i} + \alpha_\mathrm{i} \mathrm{p}_\mathrm{i} + \omega_\mathrm{i} \mathrm{s}_\mathrm{i} \ , \\
8. & \mathrm{r}_{\mathrm{i}+1} := \mathrm{s}_\mathrm{i} - \omega_\mathrm{i} \tilde{\mathbf{\Theta}}\ \mathrm{s}_\mathrm{i} \ , \\
9. & \beta_\mathrm{i} := \frac{(\mathrm{r}^*, \mathrm{r}_{\mathrm{i}+1})}{(\mathrm{r}^*, \mathrm{r}_\mathrm{i})} \ \frac{\alpha_\mathrm{j}}{\omega_\mathrm{j}} \ , \\
10. & \mathrm{p}_{\mathrm{i}+1} := \mathrm{r}_{\mathrm{i}+1} + \beta_{\mathrm{i}+1} \ (\mathrm{p}_\mathrm{i} - \omega_\mathrm{i} \tilde{\mathbf{\Theta}}\ \mathrm{p}_\mathrm{i}) \\
11. & \mathrm{end\ do}
\end{array} \tag{10}$$

From Eq.(10) it is obvious that the Bi-CGSTAB can be just as easily programmed as a standard Conjugate Gradient (CG) iteration, having a calculational sequence which is computationally almost the same as the one for a Conjugate Gradient approach, which was known already 40 years earlier ! An important computational difference is that Bi-CGSTAB features two matrix-vector multiplications per step, contrary to merely one for the CG sequence. With the system apparently featuring not so dramatic departures from symmetry, use of a CG sequence (which is primarily designed for symmetric systems) proved to be practically similarly adequate with regard to observed convergence rate at rebalancing level. However, since Bi-CGSTAB is clearly designed to cover nonsymmetric systems optimally as well, the latter is the preferred approach.

3. Asymptotic Extrapolation in PANBOX and PRISM

Upon detection of asymptotic behaviour (also when systematically applying coarse mesh rebalancing), the nodal diffusion iterations in PANBOX and PRISM can be given an additional acceleration boost by the application of *asymptotic extrapolation*. The implementation of the asymptotic extrapolation technique in FLUXS/T was inspired by an earlier development pursued by Bennewitz and Wagner for the MEDIUM code [8], and was refined later by Böer [26] for being compatible with arbitrary variants of the NEM diffusion equation formulations.

3.1 A Brief Modal Analysis of Asymptotic Extrapolation

A modal analysis, leading to the generic asymptotic extrapolation formula, typically starts with the assumption that the diffusion iterand has started developing in an asymptotic way, i.e. with a practically constant error decay ratio which is determined by the lowest frequency/longest wavelength component of the remaining higher modes in the diffusion iterand. With this converged decay ratio, this higher-mode part will eventually damp out, if the iterations are pursued long enough. However, if the error decay ratio is only marginally smaller than 1, (like 0.96 or higher), very many additional whole-core diffusion sweeps would be required for establishing sufficient convergence when relying on power decay only.

With the application of extrapolation at this point, the condition being that asymptotic behaviour has been detected, one can realize a coarse filtering effect that realizes a significant suppression, in the diffusion iterand, of the dominant asymptotic error shape. The added value of asymptotic extrapolation is that, in this way, an additional eigenspectral range will be caught which obviously was not covered by systematically applied rebalancings (an example of this will be shown in section 4.1). The modal analysis starts with denoting the error $\underline{\epsilon}^{(\mathrm{n})}$ of the n^{th} diffusion iterand $\underline{\Psi}^{(\mathrm{n})}$ (which is a composite of nodal fluxes, outcurrents and higher-order nodal expansion coefficients) with regard to the exact solution $\underline{\Psi}_0$, as an expansion in terms of fundamental and higher nodal diffusion modes,

$$\underline{\epsilon}^{(\mathrm{n})} \;=\; \underline{\Psi}^{(\mathrm{n})} \;-\; \underline{\Psi}_0 \;=\; \sum_{\mathrm{m}=1}^{\infty} \mathrm{c}_{\mathrm{m}}^{(\mathrm{n})}\, \underline{\Psi}_{\mathrm{m}} \tag{11}$$

with the eigenmodes $\underline{\Psi}_{\mathrm{m}}$ being associated with eigenvalues λ_{m} (with $\lambda_0 < \lambda_1 < \lambda_2 < \ldots$) with regard to the operators that define the successive iterative steps. The decay factor $|\mathrm{c}_{\mathrm{m}}^{(\mathrm{n}+1)}/\mathrm{c}_{\mathrm{m}}^{(\mathrm{n})}|$ for the renormalized presence of the m^{th} mode in the iterand thus follows from

$$\left| \frac{c_m^{(n+1)}}{c_m^{(n)}} \right| \cong \frac{\lambda_0}{\lambda_m} \tag{12}$$

When applying coarse mesh rebalancing systematically, one can assume that the modal components with eigenvalues very much closest to λ_0 will be significantly suppressed already, depending on the eigenspectral proximity of the highest rebalancing level to the diffusion level.

The first higher mode that is *not* adequately covered by rebalancing is denoted here with $\underline{\Psi}_\eta$. Common whole-core steady-state cases that converge reasonably well through rebalancing are typically characterized by $\lambda_\eta > \frac{125}{100}\lambda_0$, meaning that average diffusion error decay ratios of 0.8 or less occur, and that no more than between 25 and 50 whole core nodal diffusion sweeps are necessary for establishing a fast flux error of less than 10^{-5}, starting from flat initial guesses for the diffusion iterand. In case of the IAEA-3D1 full core steady-state benchmark, for example, between 27 and 40 diffusion sweeps (depending on which NEM variant is chosen) are necessary for reaching a fast flux error of less than 10^{-5}, compared to more than 800 sweeps if no rebalancing is applied.

This may however be worse for less common cases (an example of which is shown in subsection 4.1) where perhaps $\lambda_\eta \cong \frac{103}{100}\lambda_0$, and thus more than 200 full core diffusion sweeps would be required in spite of a systematic use of rebalancing with well-converged driving factors. As soon as the error decay ratio for the successive nodal diffusion iterands starts behaving asymptotically, the shape of the higher modal expansion part $\sum_{m=\eta}^{\infty} c_m \underline{\Psi}_m$ can be expected to have boiled down to a more or less stable form which is a sum of the remaining longer wavelength/low frequency nonfundamental modes. Now, using the approximation

$$\underline{\epsilon}^{(n+1)} - \underline{\epsilon}^{(n)} = \underline{\Psi}^{(n+1)} - \underline{\Psi}^{(n)} \cong (\bar{\sigma} - 1) \underbrace{\sum_{m=\eta}^{\infty} c_m^{(n)} \underline{\Psi}_m}_{\underline{\Psi}^{(n)} - \underline{\Psi}_0} = (\bar{\sigma} - 1) \left[\underline{\Psi}^{(n)} - \underline{\Psi}_0 \right] , \tag{13}$$

with $\bar{\sigma}$ the observed asymptotic error reduction ratio, the asymptotic approximation $\underline{\Psi}_{\mathrm{ASYX}}$ for the exact fundamental mode $\underline{\Psi}_0$ can be derived as follows:

$$\begin{aligned} \underline{\Psi}_{\mathrm{ASYX}} = \underline{\Psi}^{(n)} - \frac{1}{\bar{\sigma} - 1} \left[\underline{\Psi}^{(n+1)} - \underline{\Psi}^{(n)} \right] &= \underline{\Psi}^{(n)} + \frac{1}{1 - \bar{\sigma}} \left[\underline{\Psi}^{(n+1)} - \underline{\Psi}^{(n)} \right] \\ &= \frac{1}{1 - \bar{\sigma}} \underline{\Psi}^{(n+1)} - \frac{\bar{\sigma}}{1 - \bar{\sigma}} \underline{\Psi}^{(n)} \\ &= \bar{\theta} \underline{\Psi}^{(n+1)} - [\bar{\theta} - 1] \underline{\Psi}^{(n)} \end{aligned} \tag{14}$$

with the resulting scalar extrapolation factor $\bar{\theta} = \frac{1}{1-\bar{\sigma}} = \frac{1}{1-\frac{\lambda_0}{\lambda_\eta}}$. Using this expression for $\bar{\theta}$ and reordering Eq.(14) by writing

$$\bar{\theta}\,[\underline{\Psi}^{(\mathrm{n})} - \underline{\Psi}^{(\mathrm{n}+1)}] \;=\; \bar{\theta} \sum_{\mathrm{m}=\eta}^{\infty} \left(1 - \frac{\lambda_0}{\lambda_\mathrm{m}}\right) \mathrm{c}_\mathrm{m}^{(\mathrm{n})} \underline{\Psi}_\mathrm{m} \;=\; \mathrm{c}_\eta^{(\mathrm{n})} \underline{\Psi}_\eta \;+\; \bar{\theta} \sum_{\mathrm{m}=\eta+1}^{\infty} \left(1 - \frac{\lambda_0}{\lambda_\mathrm{m}}\right) \mathrm{c}_\mathrm{m}^{(\mathrm{n})} \underline{\Psi}_\mathrm{m}$$

$$= \underbrace{\mathrm{c}_\eta^{(\mathrm{n})} \underline{\Psi}_\eta + \sum_{\mathrm{m}=\eta+1}^{\infty} \mathrm{c}_\mathrm{m}^{(\mathrm{n})} \underline{\Psi}_\mathrm{m}}_{\underline{\Psi}^{(\mathrm{n})} \,-\, \underline{\Psi}_0} \;-\; \bar{\theta} \underbrace{\sum_{\mathrm{m}=\eta+1}^{\infty} \left(1 - \frac{\lambda_\eta}{\lambda_\mathrm{m}}\right) \mathrm{c}_\mathrm{m}^{(\mathrm{n})} \underline{\Psi}_\mathrm{m}}_{\text{"unidentified higher frequencies"}}$$

$$= \underline{\Psi}^{(\mathrm{n})} \;-\; \underline{\Psi}_0 \;-\; \bar{\theta} \sum_{\mathrm{m}=\eta+1}^{\infty} \left(1 - \frac{\lambda_\eta}{\lambda_\mathrm{m}}\right) \mathrm{c}_\mathrm{m}^{(\mathrm{n})} \underline{\Psi}_\mathrm{m} \quad , \qquad (15)$$

it becomes obvious that the asymptotic extrapolation approximation does in fact ignore the legitimately remaining higher frequency residual $\sum_{\mathrm{m}=\eta+1}^{\infty}(1 - \frac{\lambda_\eta}{\lambda_\mathrm{m}})\ \mathrm{c}_\mathrm{m}^{(\mathrm{n})} \underline{\Psi}_\mathrm{m}$ in this modal analysis equation. Now, one can expect that, after the iterative solution procedure has progressed, lower frequency modes with λ_m not much higher than λ_η will be clearly dominant. It is especially the range of lower frequency modes with λ_m a bit more distant from λ_η (for which $|1 - \frac{\lambda_\eta}{\lambda_\mathrm{m}}|$ will be a bit larger), which can somewhat degrade the acceleration effect of an asymptotic extrapolation. With values $\bar{\sigma}$ >0.95 not being uncommon for more heterogeneous cases where asymptotic extrapolation is ignited, the extrapolation factor $\bar{\theta}$ can become quite large. Therefore, the extrapolation measure can possibly cause a noticeable *reintroduction* of the unwanted higher modal residual $\sum_{\mathrm{m}=\eta+1}^{\infty}(1 - \frac{\lambda_\eta}{\lambda_\mathrm{m}})\ \mathrm{c}_\mathrm{m}^{(\mathrm{n})} \underline{\Psi}_\mathrm{m}$ (due to the magnification effected by the multiplication with $\bar{\theta}$). This reintroduction hence can lead to a high-frequency disturbance in the extrapolated iterand approximation as a manageable collateral damage that will usually decay rather quickly again, i.e. after merely a few additional iterations. Typically, following an asymptotic extrapolation, the solution error first goes up and subsequently goes down with a slope significantly steeper than that of the asymptotic error decay prior to the application of the extrapolation. After a number of diffusion sweeps following the extrapolation, asymptotic behaviour will manifest itself gradually again, but at a substantially lower error level of the solution. It is the trade-off between the initial error increase and the following accelerated error decay which determines the effect of the pursued extrapolation. In any case, expert knowledge-based limitations have been engineered on the use and asymptotic extrapolation in FLUXS/T, which prevent the potential occurrence of numerical instabilities. An example is shown in subsection 4.1.

4. Results

For illustrating the impact of different acceleration options in PANBOX for cases of relevance in industrial licencing computations, two transient cases (corresponding to a rod drop scenario and a rod ejection followed by a scram, for different reactor cores, respectively) are shown, followed by a steady-state case featuring stuck rods combined with cold reactor conditions. The latter is a typical example of a case where the rebalancing procedure does not catch the slowly converging mode-components in the iterand sufficiently in spite of well-converged driving factors, and where the additional use of asymptotic extrapolation gives a substantial performance benefit.

4.1 Rod drop and rod ejection transient cases with different acceleration optional settings

A number of representative and relevant cases are presented here which were computed by PANBOX on an Opteron 250 (2GHz) processor, with the following different acceleration optional settings:

i) No acceleration (i.e. plain power iteration through repeated nodal diffusion sweeps)

ii) Asymptotic extrapolation (ASYX) for accelerating the nodal diffusion iterations

iii) Multi-level rebalancing (MLCMR) with few iterations per rebalancing level

iv) Acceleration settings (iii) and (ii) combined: MLCMR + ASYX

v) 1-level Krylov rebalancing (CMR1-K)

vi) Acceleration settings (v) and (ii) combined: CMR1-K + ASYX

vii) 2-level Krylov rebalancing (CMR2-K)

viii) Acceleration settings (vii) and (ii) combined: CMR2-K + ASYX

In Table I, the CPU-times needed by an Opteron 250 (2GHz) processor for computing these different cases, which are described and discussed below, are listed with regard to their dependence on different acceleration settings. For the two transients, the CPU-time associated with the preceding N-TH steady-state computation (typically also including pin power reconstruction and hot subchannel analysis) is not included. Furthermore, Figs. 2 and 3 show the required numbers of whole-core transient diffusion sweeps per time step on a *logarithmic* scale for the two presented transients. The scale being logarithmic means that visible differences in the vertical positioning of the displayed curves translate into a relevant CPU-time difference listed in Table I. Figure 4 shows the flux error decay as a function of acceleration setup for a tedious steady-state case which is discussed below as well.

The first case (”core 1 rod ejection”) consists of a reactor transient induced by the ejection of a single control rod from lower stop and 94 % power within 0.1 s, which is followed by a scram at 0.41 s. The transient was computed with high full core accuracy up to t=3.0 s; with the applied time control step mechanism, this added up to 381 transient time step whole-core diffusion computations. The transient nodal diffusion calculation was based on a 2-group semi-analytical NEM approach with quadratic transverse leakage, for a reactor core that adds up to more than 8000 nodal volumes in the NEM diffusion iterations (with 177 radial assemblies). Self-evidently, the core power first shows a rapid decrease, followed by a drastic power decrease, as of 0.41 s, due to the imposed scram. When looking at Fig.2, it becomes clear that the MLCMR approach with ASYX (setting (iv)) provides a substantial performance improvement compared to MLCMR only. However, when simply guaranteeing exact solutions at rebalancing level through use of the preconditioned Krylov approach (with a merely marginal performance difference between CMR1-K and CMR2-K), the number of ASYX occurrences decreases dramatically during the transient computation. For this case, ASYX in fact becomes practically redundant.

The second case (”core 2 rod drop”) consists of a reactor transient induced by dropping all control rods within 3.75 s, starting at t=1.04 s and 100 % power. The 2-group transient computations are pursued with polynomial NEM-M2B2 [4,5] at the level of more than 4000 spatial nodes (with 193 assemblies). The transient is computed up to 5.0 seconds; with the applied time control step mechanism, this added up to 133 transient time step whole-core diffusion computations. Obviously, the core power decreases immediately and dramatically as a result of the rod drop action. When looking at Fig.3, it becomes similarly clear that the MLCMR approach with ASYX (setting (iv)) provides a substantial performance improvement compared to MLCMR only, as was the case for the previous transient as well. Also here, when simply guaranteeing exact solutions at rebalancing level through use of the preconditioned Krylov approach, the number of ASYX occurrences decreased substantially, though not as dramatically as for the previous transient. Nevertheless a second-order computational advantage can be observed when co-using ASYX, in particular in combination with the 2-level Krylov rebalancing scheme CMR2-K.

The third case (”core 3 cold steady state + SR”) consists of a steady-state computation under cold conditions with a few stuck control rods, which gives rise to a high degree of in-core heterogeneity that turns out to give a very tedious convergence behaviour. For this example, more than 200 full core diffusion sweeps were required in spite of a systematic use of rebalancing with well-converged driving factors, when not using asymptotic extrapolation. It is here that the periodic asymptotic extrapolation (ASYX) measure was able to give a *substantial* additional convergence boost. It is intesting to observe here, however, that for asymptotic extrapolation to be the most effective, a very accurate solution of the rebalancing equations, and thereby a solid separation between ”CMR-covered modes” and the rest of the eigenspectrum, is apparently *mandatory* for enabling a good result induces by ASYX. Clearly, the multi-level scheme MLCMR, which is known not to solve rebalancing equations exactly per level, enables only a degraded additional acceleration effect to be realized by asymptotic extrapolation. However, the Krylov 1- and 2-level approaches CMR1-K (1-level rebalancing) and CMR2-K (2-level rebalancing), both of whom use preconditioned Bi-CGSTAB to enforce practically exact solutions at rebalancing level, clearly enable a noticeably higher benefit from the additional use of asymptotic extrapolation. For cases like this one, ASYX as an auxiliary acceleration mechanism can hence provide substantially more than merely a second-order additional acceleration effect.

Table I. Required CPU-times (seconds) on an Opteron 250 (2GHz) processor with regard to their dependence on different acceleration settings for pursuing the specified computations.

Acc. mode	core 1 rod ejection	core 2 rod drop	core 3 cold steady state + SR
no acc.	–	1502	58.3
ASYX only	7484	1012	53.4
MLCMR	850	150	11.5
MLCMR + ASYX	792	125	11.1
CMR1-K	655	113	11.8
CMR1-K + ASYX	658	111	9.4
CMR2-K	650	112	11.9
CMR2-K + ASYX	619	107	6.2

Fig.2. Required numbers of whole-core transient diffusion sweeps for case ”core 1 rod ejection”.

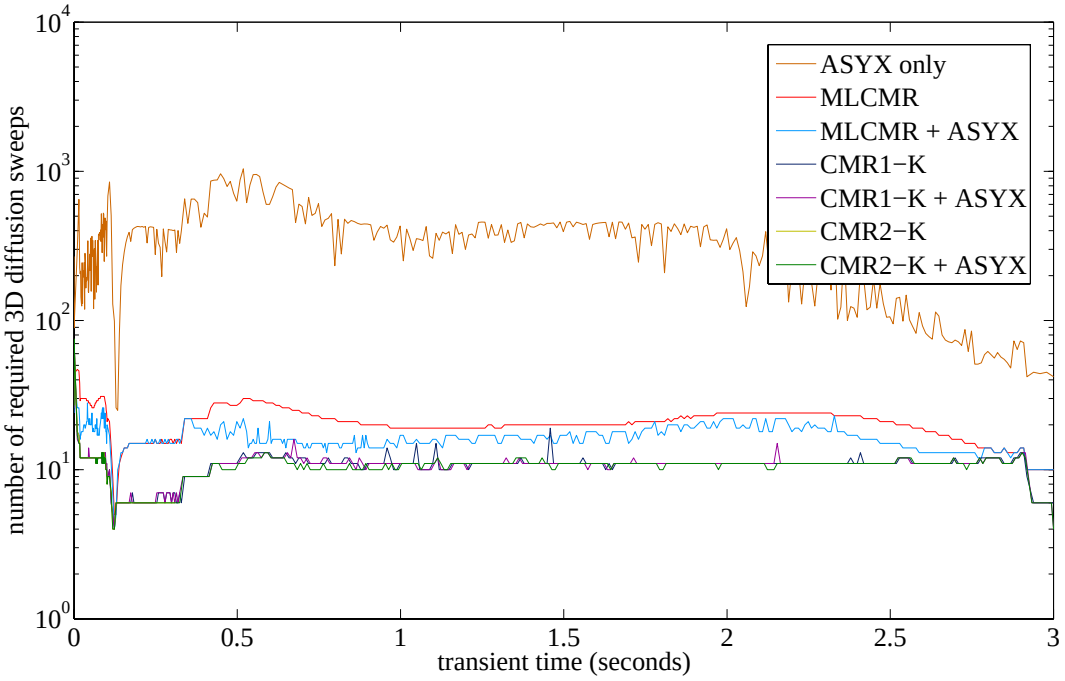


Fig.3. Required numbers of whole-core transient diffusion sweeps for case ”core 2 rod drop”.

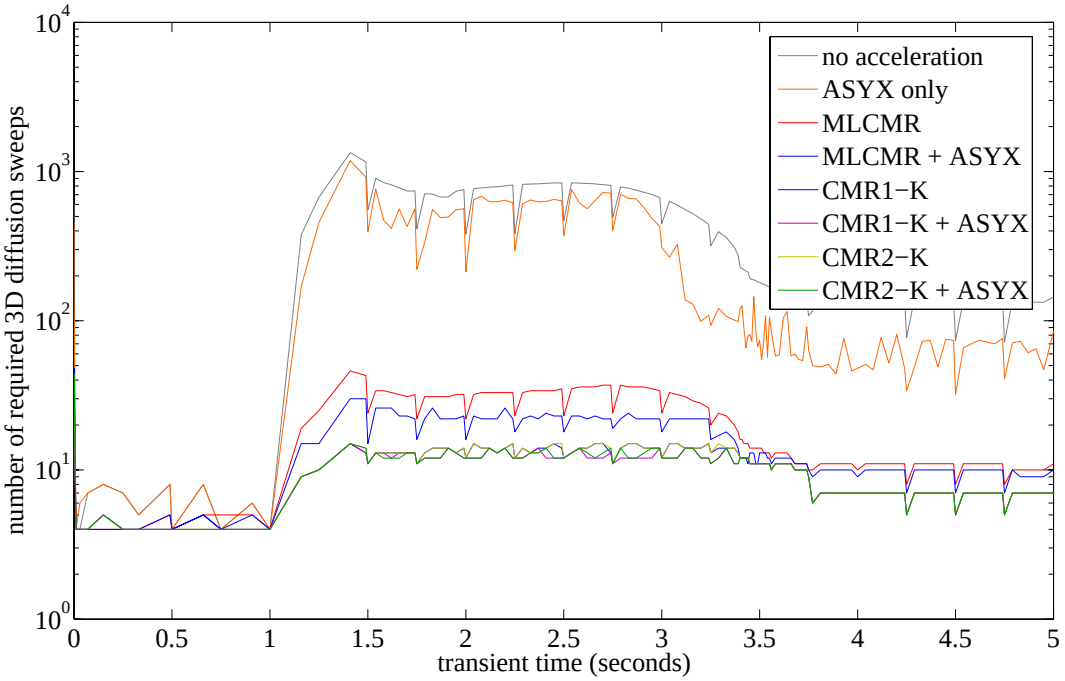
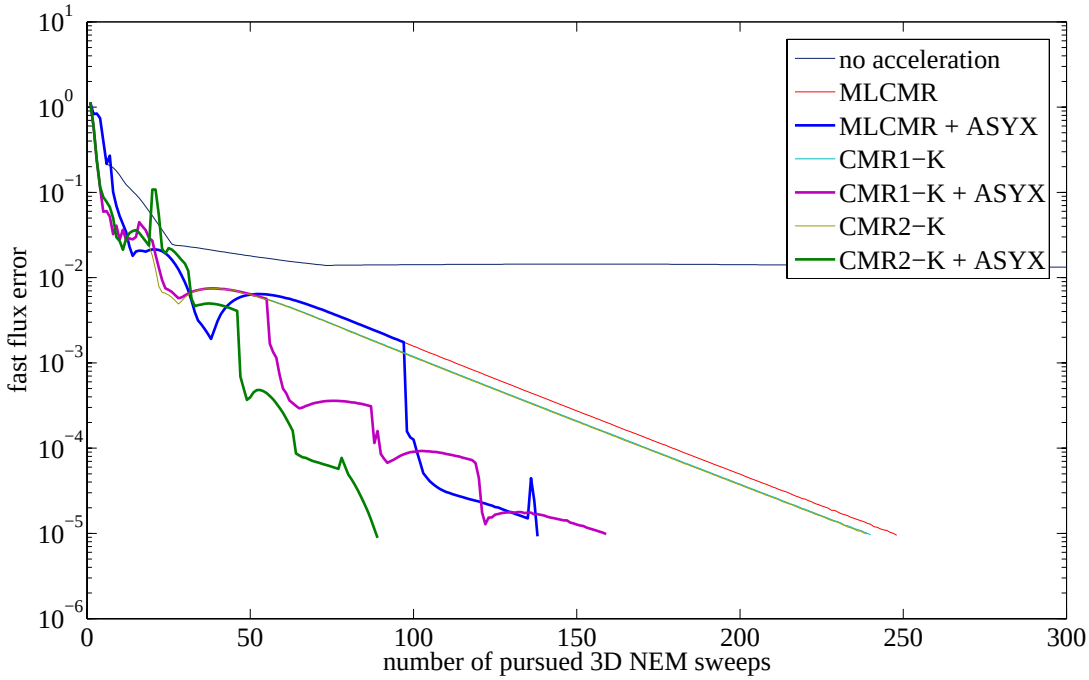


Fig.4. Convergence plots for case ”core 3 cold steady state + SR”.



5. Conclusions

A clear overall message is that the introduction of preconditioned Krylov subspace approaches at rebalancing level, which serve to provide highly accurate solutions of the rebalancing equations at the different rebalancing grid levels, gave rise to a substantially improved all-round behaviour of FLUXS/T, particularly also in conjunction with the use of asymptotic extrapolation as an additional acceleration measure. The presented results further provide clear arguments in favour of the adopted convention of using coarse mesh rebalancing and asymptotic extrapolation as complementary acceleration mechanisms within the same whole-core flux calculation, with a particular benefit especially for cases where rebalancing and asymptotic extrapolation cover different parts of the higher modal spectrum to be suppressed. Using more than one rebalancing level, when combined with Krylov methods at rebalancing level, still provides the best overall computational performance, through a powerful synergy of multi-grid suppression of low-frequency higher modes combined with an additionally low residual in the higher modal range through the use of Krylov subspace iteration. With its carefully implemented acceleration mechanisms, PANBOX and PRISM offer a high computational performance and hence a high degree of industrial practicality, which enables demanding steady-state and transient whole-core computations to be pursued with high accuracy and within sufficiently short time frames for AREVA NP’s customers in the nuclear industry.

Acknowledgements

With the exception of the recently implemented preconditioned Krylov subspace approaches at rebalancing level, the current status of FLUXS/T and PANBOX is the result of more than two decades of code and methodology development work pursued by R.E. Böer and H. Finnemann (both of whom are retired now). Furthermore, the author wishes to acknowledge research contributions provided by PhD-student Maxime Havet, connected to a recent implementation of Bi-CGSTAB [21,22] as a valuable added component of the preconditioned Krylov rebalancing equation solver in FLUXS/T.

References

1) R. Böer, R. Böhm, H. Finnemann, R. Müller, "The Coupled Neutronics and Thermal-Hydraulics Code System PANBOX for PWR Safety Analysis", *Atomenergie/Kerntechnik* **57** pp.49-54 (1992).

2) R.G. Grummer et al., "Siemens Integrated Code System CASCADE-3D for Core Design and Safety Analysis", *Proceedings PHYSOR 2000*, Pittsburgh USA, 2000.

3) J.W. Jackson, N.E. Todreas, "COBRA IIIC/MIT-2: A Digital Computer Program for Steady State and Transient Thermal-Hydraulic Analysis of Rod Bundle Nuclear Fuel Elements", Energy Laboratory Report MIT-EL 81-018, Massachusetts Institute of Technology, USA (1981).

4) H. Finnemann, F. Bennewitz, M. Wagner, "Interface current techniques for multidimensional reactor calculations", *Atomkernenergie (ATKE)* **30** (1977).

5) H. Finnemann, W. Gundlach, "Space-time kinetics code IQSBOX for PWR and BWR", *Atomkernenergie (ATKE)* **37** (1981).

6) E. Wachspress, "Iterative Solution of Elliptic Systems", Prentice Hall, Englewood Cliffs (1966).

7) S. Nakamura, "Computational Methods in Engineering and Science", chapter 8 (Wiley & Sons, 1977).

8) F. Bennewitz, A. Müller, M.R. Wagner, "MEDIUM – Erstellung eines 3-dimensionalen Abbrandprogrammes für H_2O-moderierte Druckwasserreaktoren", *Kraftwerk Union Programmentwicklungsbericht Nr. 674/74*, Erlangen, Germany (August 19, 1974).

9) H. Finnemann, R. Böer, R. Müller, Y.I. Kim, "Adaptive Multi-level Techniques for the Solution of Nodal Transport Equations", *Proceedings Third European Conference on Multigrid Methods*, Bonn, Germany (1990).

10) D. S. Kim, N.Z. Cho, "Acceleration of Three-Dimensional AFEN Nodal Codes via Coarse Group Rebalance and Direct Matrix Inverse", *Proceedings M&C 1999*, Madrid, Spain (1999).

11) R. Böer, H. Finnemann, "Fast Analytical Flux Reconstruction Method for Nodal Space-Time Nuclear Reactor Analysis", *Annals of Nuclear Energy* **19**, No.10-12, pp.617-628 (1992).

12) H.C. Lee, K.Y. Chung, C.H. Kim, "Unified Nodal Method for Solution to the Space-Time Kinetics Problems", *Nuclear Science and Engineering* **147**/3, pp.275-291 (2004).

13) D. Lee, T.J. Downar, Y. Kim, "A Nodal and Finite Difference Hybrid Method for Pin-by-Pin Heterogeneous Three-Dimensional Light Water Reactor Diffusion Calculations", *Nuclear Science and Engineering* **146**/3, pp.319-339 (2004).

14) W. Hackbusch, "Multi-Grid Methods and Applications", Springer, Berlin (1985).

15) W.F. Miller, "Generalized Rebalance: A Common Framework for Transport Acceleration Methods", *Nuclear Science and Engineering* **65** (1978).

16) G.R. Cefus, E.W. Larsen, "Stability Analysis of Coarse Mesh Rebalancing", *Nuclear Science and Engineering* **105** (1990).

17) K.S. Smith, J.D. Rhodes, "Full-Core, 2-D, LWR Core Calculations with CASMO-4E", *Proceedings PHYSOR 2002* (Oktober 7-10), Seoul, Korea (2002).

18) G. Grassi, "A Nonlinear Spatial Multi-Grid Acceleration for the Method of Characteristics in Unstructured Meshes", *Proceedings M&C 2005*, Avignon (2005).

19) U. Trottenberg, C. Oosterlee, A. Schüller, "MULTIGRID", *'Academic Press* (2001).

20) S.Y. Fialko, "An Aggregation Multilevel Iterative Solver with Shift Acceleration for Eigenvalue Analysis of Large-Scale Structures", *Computer Methods in Mechanics (CMM) 2003*, Gliwice, Poland (2003).

21) H.A. van der Vorst, "Bi-CGSTAB: a Fast and Smoothly Converging Variant of Bi-CG for the solution of nonsymmetric linear systems", SIAM J. Sci. Stat. Comput. **13**(2), pp.631-644 (1992).

22) Y. Saad, "Iterative Methods for Sparse Linear Systems", 2^{nd} edition, *Society for Industrial and Applied Mathematics (SIAM)* (2003).

23) J.E. Morel, "Basic Krylov Methods with Application to Transport", *Proceedings M&C 2005*, Avignon (2005).

24) D.Y.F. Kastanya, P.J. Turinsky, "Development and Implementation of a Newton-BICGSTAB Iterative Solver in the FORMOSA-B BWR Core Simulator Code", *Nuclear Science and Engineering* **150**/1, pp.56-71 (2005).

25) J. Gan, Y. Xu, T.J. Downar, "A Matrix-Free Newton Method for Coupled Neutronics Thermal-Hydraulics Reactor Analysis", *Proceedings M&C 2003*, Gatlinburg USA (2003).

26) R. Böer, *private communication* (2005).