A MULTIGRID METHOD FOR SPATIAL NEUTRON KINETICS

Nguyen Thai Sinh, William J. Garland

Department of Engineering Physics, McMaster University
1280 Main St W, Hamilton, Ontario L8S 4L7
nguyets@mcmaster.ca; garlandw@mcmaster.ca

Abstract

A multigrid solver is developed for dealing with the spatial neutron kinetics problem in a nuclear reactor. Neutron kinetics is modeled by the multigroup diffusion equations, dicretized by using a cell-centered finite volume technique in space, and a fully implicit scheme in time. Coarse grid equations are derived by using an additive correction method, allowing for easy coarsening of the grids without having to know their geometry. Even with a point Gauss-Seidel as a smoother and a simple V-cycle, the multigrid solver is shown to have the convergence rate being not only fast but also weakly dependent on the mesh size of discretization. Further improvement of the multigrid components is expected to make the solver optimal.

Introduction

The control and safety analysis of a nuclear reactor strongly relies on the prediction of transient behaviour of the reactor system under both normal operating conditions and accident situations. The analysis of such transients has been traditionally based on numerical simulation of coupled neutron kinetics and thermalhydraulics. While most reactor transient codes (e.g. CATHENA [1] or RELAP5 [2]) now employ the modern thermalhydraulic models that reflect the up-to-date knowledge of governing phenomena, these codes still depend on simplified neutron kinetics models (i.e. point reactor or dimensionally reduced models) whose results tend to be not only inaccurate but also non-conservative for many important cases of accident analyses [3]. The incorporation of a full 3D core model of neutron kinetics into the reactor system code will allow "best-estimate" simulations of reactor dynamics but still requires an extensive computation [4]. The purpose of this work is to investigate an efficient method for accurate solution of the spatial neutron kinetics problem.

It is commonly agreed that the kinetics of neutrons in a nuclear reactor is adequately represented by the multigroup neutron diffusion equations, together with the 6-group delayed neutron precursor equations [5]. Typically, finite difference methods are the simplest and most direct approach to the numerical solution of such a system of parabolic partial differential equations. The primary advantage of the finite difference method is that differential equations can be straightforwardly expressed in difference form to obtain algebraic expressions. Also, the method is shown to converge to the exact solution in the limit of infinitely fine mesh spacing. However, to obtain an acceptable accuracy, the method must work with a very fine grid, giving rise to an extremely large number of discretized equations to solve at each time step. Moreover, to avoid numerical instability and inconsistency, an implicit scheme for time integration must be used. Basic numerical

methods, direct and iterative, are quite inefficient for inverting such a large algebraic system in 3D geometry.

The inefficiency of the finite difference methods has led to the development of various coarse-mesh methods [6] for dealing with the space problem of reactor kinetics, among which the nodal methods [7] have received the greatest acceptance among the reactor physics community. Nodal methods utilize a large mesh size in order to reduce the number of equations to be solved and thus increase the computational efficiency. However, in addition to the great complexity in mathematical derivation and difficulty in error analysis, it is difficult to accelerate the convergence for the solution of the nodal discretized equations [8]. As a result, the nodal kinetics computation is rather time-consuming for spatial kinetics problems of practical interest.

Multigrid methods [9] are among the fastest iterative methods known today for solving a large, sparse algebraic system arising from the discretization of partial differential equations. The idea of all multigrid methods is to compute using a set of grids of different scales to eliminate error components (which can be thought as waves) of different wavelengths. Many usual iterative methods are very efficient at damping short-wave components on a given grid but not as efficient at damping the long-wave components, which, however, turn out to be short waves on a coarse enough related grid. In this sense, the multigrid is essentially a technique that transfers the error between related grids and smoothes its components on appropriate grids. The greatest property of the multigrid is that, unlike the usual iterative methods, the multigrid offers a convergence rate that is not deteriorated with increasing size of the discretized system.

Although multigrid methods now become a quite standard numerical method in many engineering problems, there have been not many applications to reactor kinetics found in the literature. As noted above, nodal methods currently predominate in reactor kinetics computations but the choice of the nodal unknowns (the node flux and leakage) makes it difficult to apply the multgrid to the solution of a quite complex nodal discretized system. With the traditional multigrid, known as the geometric multigrid, it is not easy to form the coarse grid equations (i.e. the grid coarsening). The algebraic multigrid [9], whose grid coarsening is not based on the grid geometry, can be a remedy for this problem but it is more expensive and, generally, slower than the geometric multigrid. Even with the finite differencing discretization (which is more suitable for multigrid application) there are still difficulties encountered in grid coarsening due to great material heterogeneity of the reactor core [10]. A very practical procedure in reactor calculations is to homogenize separate but regular core regions (cells) such as fuel assemblies or reflectors, in which the material properties are treated in an average sense. Due to a strong difference and/or discontinuity of the properties from cell to cell, it is easier to refine the grid (i.e. to divide each cell) rather than to coarsen it (i.e. to group two or more cells). As a result, the coarsest grid still has a relatively large number of equations to solve for.

In this work, we discretize the kinetics equations by using a cell-centered finite volume technique in space and a fully implicit scheme in time. A multigrid solver is developed to solve the discretized system at each time step. An additive correction multigrid [11] is

used to generate coarse-grid equations by summing up the corresponding residual equations on a finer grid. In fact, the coarse-grid equation system is exactly the same in form as the fine-grid system but has almost 2^3 times fewer equations. This coarsening method allows for reaching the non-trivial coarsest grid that has only a single node without taking into account the coarse-grid properties. A Gauss-Seidel iterative method is used as a smoother. Numerical experiments have been performed for the McMaster Nuclear Reactor core discretized with different mesh spacing. The preliminary results show that, even with the simplest V-cycle, the convergence rate of the multigrid solver is much faster than that of the same Gauss-Seidel method on the single fine grid. Moreover, while the latter triples in the number of iterations when the grid size doubles, the multigrid solver only slightly increases this number.

Another problem in solving a large algebraic system is associated with memory storage and computation work per iteration. This problem is not so critical nowadays because it can be effectively handled by the implementation of parallel computation on a multiprocessor computer system. By using an appropriate smoother method (for example, the Jacobi or Red-Black Gauss-Seidel method) the multigrid solver is suitable for such parallel computation.

As our goal is to construct a multigrid solver that has a convergence rate independent of the fine grid size, it is still required to improve the performance of each multigrid component.

Neutron Kinetics Equations

Assume that the kinetics of neutrons in a reactor core is adequately represented by the multigroup diffusion equations with delayed neutron effects taken into account. As a result, we have a system of parabolic partial differential equations to solve for the neutron energy group fluxes ϕ_g (g=1,2,...,G) and the delayed neutron precursor group concentrations C_i (i=1,2,...,N) in space and time $(\vec{r}, t)$ [6]:

$$\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\vec{r},t) = \nabla\cdot D_g(\vec{r},t)\nabla\phi_g(\vec{r},t) - \Sigma_{tg}(\vec{r},t)\phi_g(\vec{r},t) + \sum_{g'=1}^{G}\Sigma_{sg'g}(\vec{r},t)\phi_{g'}(\vec{r},t) + (1-\beta)\chi_g^p\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r},t)\phi_{g'}(\vec{r},t) + \sum_{i=1}^{N}\lambda_i C_i(\vec{r},t)\chi_{gi}^d + S_g^{ext}(\vec{r},t) \quad (1)$$

$$\frac{\partial}{\partial t}C_i(\vec{r},t) = -\lambda_i C_i(\vec{r},t) + \beta_i\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r},t)\phi_{g'}(\vec{r},t) \quad (2)$$

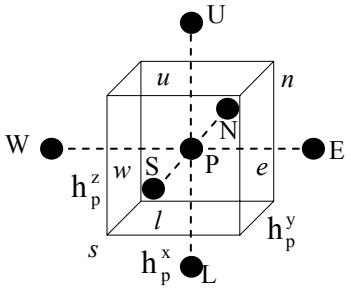
Here, S_g^{ext} is the external neutron source. The neutron group constants are denoted by: v – speed; D – diffusion coefficient; Σ_t – total cross-section; $\nu\Sigma_f$ – fission yield; $\Sigma_{sg'g}$ – scattering cross-section from group g' into group g; χ^p and χ^d – spectra of prompt and delayed neutrons, respectively. The delayed neutron group constants are: λ_i – decay constant and β_i – fraction yield, $\beta = \sum_{i=1}^{N}\beta_i$.

Appropriate initial and boundary conditions are required to make the system closure, such that:

$$\phi_g(\vec{r},0) = \phi_{0g}(\vec{r});\ C_i(\vec{r},0) = C_{i0}(\vec{r}) \quad \text{and} \quad \phi_g(\vec{r}_s,t) = 0 \quad \forall t \geq 0,\ \vec{r}_s \in \text{boundary}$$

Discretization

For spatial discretization, the finite cell-centered difference method in 3D Cartesian geometry is used. Thus, the computational domain, the reactor core extended to its boundary, is divided into a number of small rectangular boxes to form a regular grid of nodes. The boundary is assumed to be an outer surface at an extrapolation length from the fuelled core region. Then, the kinetics equations are integrated over each node volume $V_p = h_p^x h_p^y h_p^z$ using the box integration technique [12]. For a given node P, we define the node average flux of energy group g: $\phi_p^g \equiv \frac{1}{V_P}\iiint_{V_p}\phi^g dV$, and the node average precursor concentration of delayed group i: $C_{pi} \equiv \frac{1}{V_P}\iiint_{V_p} C_i dV$.



Then, the LHS terms become:

$$\frac{1}{V_P}\iiint_{V_p}\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\vec{r},t)dV = \frac{1}{v_g}\frac{\partial}{\partial t}\phi_p^g(t)$$

$$\frac{1}{V_P}\iiint_{V_P}\frac{\partial}{\partial t}C_i dV = \frac{\partial}{\partial t}C_{pi}(t)$$

The diffusion term in the RHS is computed, first, by applying the Gauss's Divergence Theorem:

$$\frac{1}{V_P}\iiint_{V_P}\nabla\cdot D\nabla\phi dV = \frac{1}{V_P}\iint_S D\nabla\phi\cdot d\vec{S} = \sum_f\left(D\frac{\partial\phi}{\partial r}\right)\Big|_f h_p^r,$$

where f is any from the six faces (w, e, s, n, u, l) of the cubic node; r is a coordinate x, y or z when the face f is, respectively, (w or e), (s or n) and (u or l). Next, the gradient of flux at a node face is approximated by a finite difference, for example at face w:

$$\left(D\frac{\partial\phi}{\partial x}\right)\Big|_w \cong D_w\frac{\phi_P-\phi_W}{\Delta_{PW}}, \quad \text{with } D_w = \frac{\Delta_{PW}}{\frac{h_P^x}{2D_P}+\frac{h_W^x}{2D_W}};\ \Delta_{PW} = \frac{h_p^x}{2}+\frac{h_W^x}{2}$$

It is noted here that, for acceptable accuracy of this approximation, the grid spacing or the node size h must be less than the smallest diffusion length [12], which is about 2 cm in thermal reactors. The diffusion term can be written as following:

$$\frac{1}{V_P}\iiint_{V_P}\nabla\cdot D\nabla\phi dV = \left[\sum_{nb\subset P}a_{nb}^g\phi_{nb}^g - \left(\sum_{nb\subset P}a_{nb}^g\right)\phi_P^g\right]$$

where 'nb' (neighbouring to P) stands for W, E, N, S, U, L, i.e. the nodes respectively West, East, North, South, Upper, Lower of the given node P; and, the coefficients a_{nb}^{g} are $a_{nb}^{g} = \dfrac{D_f^g}{\Delta_{P \div nb} h_P^r}$ (f is the face dividing node P and node 'nb').

The remaining terms in the RHS are simply the product of averaged values of the flux and group constant within the node. We finally arrive at the following equation system for an inner node P (which is not on the boundary):

$$\frac{1}{v_g}\frac{d}{dt}\phi_p^g = \sum_{nb\subset P} a_{nb}^g \phi_{nb}^g - \left(\sum_{nb\subset P} a_{nb}^g + \Sigma_t^g\right)\phi_P^g + \sum_{g'=1}^{G}\left[\Sigma_s^{g'g} + (1-\beta)\chi_p^g \nu\Sigma_f^{g'}\right]\phi_p^{g'} + \sum_{i=1}^{N}\lambda_i \chi_{di}^g C_{pi} + S_p^g \qquad (3)$$

$$\frac{d}{dt}C_{pi} = -\lambda_i C_{pi} + \beta_i \sum_{g'=1}^{G} \nu\Sigma_f^{g'}\phi_P^{g'} \qquad (4)$$

It is convenient to form boundary nodes so that they all have a zero width in the direction to a nearby inner node and, hence, they all have zero volume. Equations for boundary nodes are derived directly from the boundary conditions: $\phi_p^g = 0$ for P on the boundary.

We now discretize equations (3) and (4) in time. Assume that the material properties of the reactor core are unchanged during a rather short interval of time Δt, over which the kinetics equations are temporally integrated. If we use the fully implicit scheme, then all fluxes and precursor concentrations in the right hand sides of equations (3) and (4) will be taken at time $t+\Delta t$. Using an approximation $\dfrac{d}{dt}f(t) \cong \dfrac{f^{t+\Delta t} - f^t}{\Delta t}$ for the $\phi_p^g(t)$ and $C_{pi}(t)$, we can first compute $C_{pi} \equiv C_{pi}^{t+\Delta t}$ (omitting $t+\Delta t$ for short):

$$C_{pi} = \frac{C_{pi}^t + \Delta t \beta_i \sum_{g'=1}^{G} \nu\Sigma_f^{g'}\phi_p^{g'}}{1+\lambda_i \Delta t} \qquad (5)$$

and then $\phi_p^g \equiv \phi_p^{g(t+\Delta t)}$, given in matrix notation by:

$$[A_P][\Phi_P] = \sum_{nb\subset P}[A_{nb}][\Phi_{nb}] + [B_p] \qquad (6)$$

where $[\Phi_0] = \{\phi_0^g\}$ $[B_p] = \{b_p^g\}$, $g = 1, 2, ..., G$

$$[A_P] = \begin{bmatrix} a_p^1 - a_p^{11} & -a_p^{21} & \dots & -a_p^{G1} \\ -a_p^{12} & a_p^2 - a_p^{22} & \dots & -a_p^{G2} \\ \dots & \dots & \dots & \dots \\ -a_p^{1G} & -a_p^{2G} & \dots & a_p^G - a_p^{GG} \end{bmatrix} \qquad [A_{nb}] = \begin{bmatrix} a_{nb}^1 & & & \\ & a_{nb}^2 & & \\ & & \dots & \\ & & & a_{nb}^G \end{bmatrix}$$

with $$a_p^g = \left[\frac{1}{v_g \Delta t} + \sum_{nb \subset P} a_{nb}^g + \Sigma_t^g \right]; \qquad a_p^{g'g} = \Sigma_s^{g'g} + \left[(1-\beta)\chi_p^g + \sum_{i=1}^{N} \frac{\lambda_i \chi_{di}^g \beta_i \Delta t}{1+\lambda_i \Delta t} \right] \nu \Sigma_f^{g'}$$

and $$b_p^g = \sum_{i=1}^{N} \frac{\lambda_i \chi_{di}^g C_{pi}^t}{1+\lambda_i \Delta t} + S_p^g + \frac{\phi_p^{g,t}}{v_g \Delta t}$$

The boundary and initial conditions are given by:

- for fluxes $\phi_{P=B}^g = 0, \ \forall g, t>0; \qquad \phi_p^g(0) = \phi_{0p}^g, \quad \forall P, g$

- for precursors $C_{pi}(0) = \frac{\beta_i}{\lambda_i} \sum_{g'=1}^{G} \nu \Sigma_f^{g'} \phi_{0p}^{g'}, \quad \forall P, i$

Iterative Solution of Discretized Equations

At each time step, the algebraic system (6) is formed and solved for ϕ_p^g by using some linear iterative solver such as the Gauss-Seidel (GS). A good initial guess can be the solution at the previous time step, i.e. $\phi_p^{g(0)} = \phi_p^{g,t}$. The following stopping criterion is frequently used:

$$\frac{\sum_p \sum_{g=1}^{G} | R_p^{g(m+1)} |}{\sum_p \sum_{g=1}^{G} | R_p^{g(0)} |} < \varepsilon \qquad (7)$$

where the residual $[R_p]^{(m)} \equiv \{ R_p^{g(m)} \}^T$ after the m-th iteration is calculated as:

$$[R_p]^{(m)} = \sum_{nb \subset p} [A_{nb}][\Phi_{nb}]^{(m)} + [B_p] - [A_p][\Phi_p]^{(m)} \qquad (8)$$

In system (6), there is not only spatial coupling of the fluxes (by means of diffusion) but also the inter-group coupling (by means of scattering and fission) taking place. One can use a two-loop procedure similar to the power method [12] for iterating system (6). The outer loop is to update the source term including both scattering and fission sources, and the inner loop is to compute the spatial fluxes at a given guessed source. It can be observed that there is no need to accurately compute the fluxes while in the inner loop since the source term is still not accurate itself. It appears that both outer and inner loops can be combined in just one. Also, instead of moving from group to group and then from node to node in each group, we can now move from node to node and solve G equations

for G group fluxes in each node. If G is small (2 to 4 is typically for a thermal reactor), a direct method such as the Gaussian elimination can be used with small cost. The modified procedure as described is suitable for multigrid and parallel computation as well.

Multigrid Solver

Consider a large algebraic system that arises from the discretization of a set of partial differential equations on some computational grid with the mesh size $h_0=(h_x, h_y, h_z)$:

$$A\phi = b \qquad (\text{or} \qquad A_0\phi_0 = b_0) \tag{9}$$

where A and b are an N×N matrix and an N-size vector of the coefficients obtained from discretization, ϕ is an N-size column vectors of the unknowns. Typically, basic iterative methods such as GS or SOR are used to solve such a system with an arbitrary initial guess. However, it is observed that while the convergence rate is high enough at very first several iterations, it slows down quickly, and thus a large number of iterations increasing with N are required for the solution to converge.

Let $\tilde{\phi}$ be an approximation to the exact solution ϕ of the above system. We define the residual as:

$$r = b - A\tilde{\phi} = Ae, \qquad \text{where } e = \phi - \tilde{\phi} \text{ is the error} \tag{10}$$

It is easily seen that solving the algebraic system (9) for ϕ with initial guess $\tilde{\phi}$ is equivalent to solving the residual system (10) for the error e and then correcting the solution as $\phi = \tilde{\phi} + e$.

The error can be expanded in a Fourier series of sine modes of different frequencies, i.e. waves of different wavelengths. It is found from mathematical analysis that the iterative methods like GS/SOR reduce only the high frequency mode that has the half wavelength of order of the mesh size h [9]. That is, instead of reducing the error these methods actually only smooth it (hence their name is 'smoother'). The low frequency modes with larger half wavelengths are not effectively damped out. The idea of the multigrid is to send these remaining low frequency modes to a set of coarse enough grids to reduce them there by using a cheap iterative method. There are quite good reasons to do so: First, a coarser grid always has fewer nodes, hence, fewer unknowns; consequently, it is less expensive to solve for the error on coarse grids. Second, more importantly, the remaining error component modes, though smooth on the fine grid, become more oscillated on coarser grids; consequently, the newly-appearing high frequency modes could be effectively damped out by iteration. Another view may be that, in each step a normal iterative method propagates disturbances that occur at a given grid point to the points in its vicinity only, while with a multigrid the same disturbances manage to spread out all over the grid just in one step.

Let $A_1\phi_1 = b_1$ be an algebraic system on a coarse grid with the mesh size $h_1 > h_0$. Let R and P be linear operators of restriction and prolongation used to transfer information

between grids, namely, $b_1 = Rr_0$ and $e_0 = P\phi_1$. To construct a multigrid solver is essentially to determine the main components: R, P and A_1. R and P are usually chosen, while A_1 can be determined by one of the following ways: (i) discretize the original equations on the grid with mesh size h_1, or (ii) compute directly $A_1 = RA_0P$. The first approach would require a detailed knowledge of coarse grid properties and hence is very difficult to be used for a physical space of strongly heterogeneity as a reactor core. One could imagine how difficult it would be to determine, say, the diffusion coefficient for the whole core. The second approach, depending on the predefined R and P, could be quite complicated and costly in computing A_1 due to the complexity of A_0 for the multidimensional multigroup kinetics equations. In this work, we will derive the coarse grid equations $A_1\phi_1 = b_1$ based on the additive correction method [11].

Let I, J, K be the inner sizes, i.e. excluding the boundary nodes, of the fine grid (FG) on which the original equations are spatially discretized. The grid size, i.e. the total number of inner nodes, on FG is I×J×K. Each node is denoted by three-integer index i, j, k (i=1,2,...,I; j=1,2,...,J; k=1,2,...,K). Let CG_ℓ (ℓ=1,2,...,L) be a set of coarse grids generated from FG and having the grid size successively reduced approximately twice until the coarsest grid that has only one node. That is,

$CG_0 \equiv FG$	$I_0 = I,\ J_0 = J,\ K_0 = K$
CG_ℓ	$I_\ell = [(I_{\ell-1}+1)/2],\ J_\ell = [(J_{\ell-1}+1)/2],\ K_\ell = [(K_{\ell-1}+1)/2]$
CG_L	$I_L = J_L = K_L = 1$

The algebraic equations in each of these CG nodes are derived as follow. Starting from the corner node (i,j,k) = (1,1,1) on FG, in one direction, say, i, line by line we sum up the equations (6) in every two consecutive nodes and put the resulting equation in one corresponding node on CG. If the total number of FG nodes in the line is odd, then the equation in the last FG node is kept unchanged for the last node on CG. Then, the process is repeated in the second and, finally, third directions. That is, a block of up to 8 FG nodes (i to i+1; j to j+1; k to k+1) will form one CG node (i_1=[(i+1)/2], j_1=[(j+1)/2], k_1=[(k+1)/2]).

Now, assume that after the m-th iteration, the fluxes at the (m+1)-th iteration can be calculated by:

$$\Phi_{ijk} \equiv \Phi_{ijk}^{(m+1)} = \Phi_{ijk}^{(m)} + E_{ijk}, \quad E_{ijk} = \{ e_{ijk}^{g} \} \text{ - the error vector}$$

It is obvious that if the solution converges, then $E_{ijk} \to 0$. Substituting this expression into (6) with the residual replaced by (8), we get the residual equation for the error:

$$A_p^{ijk} E_{ijk} = A_W^{ijk} E_{i-1,jk} + A_E^{ijk} E_{i+1,jk} + A_S^{ijk} E_{ij-1,k} + A_N^{ijk} E_{ij+1,k} + A_U^{ijk} E_{ijk-1} + A_L^{ijk} E_{ijk+1} + R_{ijk}$$

Suppose that all fluxes in the block have the same error and denote it as $\Psi_p \equiv \Psi_{i_1 j_1 k_1} = E_{i+a,j+b,k+c}$ for a=0,1; b=0,1; c=0,1. Summing up all the residual equations in the block and noting that $\Psi_W \equiv \Psi_{i_1-1,j_1k_1} = E_{i-1,...}$, $\Psi_E \equiv \Psi_{i_1+1,j_1k_1} = E_{i+1,...}$, etc. give

$$A_p^{i_1 j_1 k_1} \Psi_p = \sum_{nb} A_{nb}^{i_1 j_1 k_1} \Psi_{nb} + B_{i_1 j_1 k_1} \tag{11}$$

where $B_{i_1 j_1 k_1} = \sum_{block} R_{ijk}$ ≡ sum of all residuals in the block

$A_{nb}^{i_1 j_1 k_1} = \sum_{nb/CG} A_{nb}^{ijk}$ ≡ sum of A_{nb} in those nodes that remain facing to 'nb' of the newly-created CG node

$A_p^{i_1 j_1 k_1} = \sum_{block} A_p^{ijk} - \sum_{nb \backslash CG} A_{nb}^{ijk}$ ≡ sum of all A_P in the block minus sum of all A_{nb} other than on faces of the CG node

In case a CG node is formed of fewer than 8 nodes, namely, 4, 2 or 1, the above formulas for CG-equation coefficients can still be used, simply replacing the FG coefficient with zero whenever any of its index gets out of its bounds. Equations (11) have the same form as of equations (6) but for fewer nodes on the CG. They can be used as coarse grid equations for correction of solution on the FG. By doing the same for the remaining CG_ℓ with $CG_{\ell-1}$ in the role of FG, we will have all needed CG equations. The FG and CG equations can be formally written in general matrix notation as:

$$A_\ell \phi_\ell = b_\ell, \qquad \ell = 0,1,\ldots,L \tag{12}$$

The algorithm for iteration of (12) is given as below:

At each time step $t > 0$:

- *Calculate* A_ℓ ($\ell = 0,1,..., L$) *and* b_0
- *On the fine grid* ($\ell = 0$):
  - *Guess* $\phi^{(0)} = \phi^t$
  - *Iterate until* ϕ *converges*: $\phi^{(m+1)} = MG(0, \phi^{(m)}, m_1, m_2, m_c)$
  - $\phi^{t+\Delta t} = \phi^{(m+1)}$

Repeat for next time step $t = t+\Delta t$

where the multigrid solver MG(…) is given by

MG(ℓ, ϕ_ℓ^*, m_1, m_2, m_c){

if(ℓ=L){ *Solve* $A^L\phi^L = b^L$ }

else{

- *Smooth* m_1 *times on* $A_\ell\phi_\ell = b_\ell$ *with the initial guess* ϕ_ℓ^*
- *Calculate the residual* $r_\ell = b_\ell - A_\ell\phi_\ell$
- *Transfer the residual* $b_{\ell+1} = Rr_\ell$
- *Call* m_c *times* $MG(\ell+1, \phi_{\ell+1}^* = 0, m_1, m_2, m_c)$
- *Correct the solution* $\phi_\ell = \phi_\ell + P\phi_{\ell+1}$
- *Smooth* m_2 *times on* $A_\ell\phi_\ell = b_\ell$

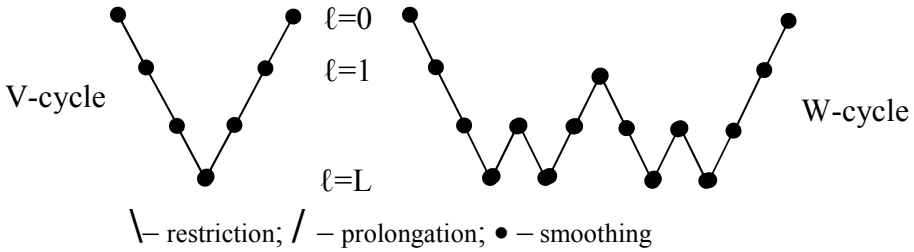
}

return ϕ^ℓ

}

In this algorithm, R adds the residual in a given grid to the source term on the next coarser grid. P simply takes the coarse grid solution as correction for the solution on the

finer grid. m_1 and m_2 are the numbers of pre- and post-smoothing sweeps; m_c is the cycling strategy number: $m_c = 1$ for V-cycle and $m_c = 2$ for W-cycle as:



The convergence rate of an MG scheme may also depend on the choice of the smoother, as well as the numbers of pre- and post-smoothing sweeps.

Numerical Experiments

As a model for numerical experiments, we use the McMaster Nuclear Reactor core, with 4 neutron groups and 6 groups of delayed neutron precursors. We first choose the basic grid with the node size on order of a fuel cell. This basic grid would be the same as a computational grid for nodal discretization if a nodal method were used. We can further refine this basic grid to have as many nodes as we want. Due to limitation of memory of our PC, we can only run the multigrid solver on the finest grid of the mesh size equal to half that of the basic grid. This limitation can be overcome by parallel computation on a multiprocessor system.

We have tried the GS as a smoother and V-cycle in order to compare their convergence rates. At a given time step, the number of iterations required for the solution to converge with the same stopping criteria is given in Table 1.

Table 1. Convergence rate vs. grid size

#iterations / Method	Basic grid 10×13×12 = 1,560 nodes	Refined grid 20×26×24 = 12,480 nodes
GS	120	384
MG-V(1,1)	18	23

Time step: $\Delta t = 10^{-4}$s Stopping criterion: $\varepsilon = 10^{-5}$

It can be seen that MG-V, the simplest, is much better than GS (although one iteration of MG-V(1,1) takes 4-5 times work as much as a single GS iteration does). The advantage is more obvious with the increasing grid size. When the number of grid nodes doubles, the convergence rate of GS decreases more than 3 times while that of MG changes slightly. The MG is evidently a fast iterative solver for large algebraic system. But the point here is that, the convergence rate of MG is independent of the grid size (although our MG solver is still not optimal yet and thus requires further improvement). It is expected that, with larger grid size, e.g. of millions of nodes, its convergence rate does not deteriorate

considerably. It can also be expected that, if any of nodal methods were used, the nodal discretized system would be more complex than the simple system (6) and hence would require more computational time to solve. This leads us to believe that the method based on finite differencing discretization and MG solver could be better than the nodal approach in many aspects, except the time step (Δt must be reduced with smaller mesh spacing). Perhaps, we should use some method for lengthening the time step such as the improved quasistatic method [3] in conjunction with an MG-solver for computing the flux shape.

Conclusion

The multigrid solver, presented in this paper, is shown to be an efficient and accurate method for numerical solution of the 3D neutron kinetics equations. While the finite differencing discretization with fine mesh is used to obtain an acceptable accuracy, the multigrid can solve the large system of discretized equations in just a small number of iterations that are weakly dependent on the system size. Further improvement of the multigrid components is required for the solver to become optimal, i.e. its convergence rate is independent of the grid size.

References

1. Hanna B.N. (1997), “CATHENA: A thermalhydraulic code for CANDU analysis”, *Nuclear Engineering and Design* **180**, 113-131.
2. The RELAP5 Code Development Team (1995), “RELAP5/MODE3 code manual”, NUREG/CR-5535.
3. Ott K.O. and Neuhold R.J. (1985), “Nuclear Reactor dynamics”, American Nuclear Society.
4. Jackson C.J., Cacuci D.G. and Finnemann H.B. (1999), “Dimensionally adaptive neutron kinetics for multidimensional reactor safety - I: New features of RELAP5/PANBOX”, *Nuclear Science and Engineering* **131**, 143-163.
5. Henry A.F. (1975), “Approximations that make space dependent kinetics computations more efficient”, Proc. The Joint NEACRP/CSNI Specialists’ Meeting on New Development in Three-Dimensional Neutron Kinetics and Review of Kinetics Benchmark Calculations, Garching, Germany (April, 1975).
6. Sutton T.M. and Aviles B.N. (1996), “Diffusion theory methods for spatial kinetics calculations”, *Progress in Nuclear Energy* **30**, 119-182.
7. Lawrence R.D. (1986), “Progress in nodal methods for the solution of the neutron diffusion and transport equations”, *Progress in Nuclear Energy* **17**, 271-301.
8. Moulton J.D. (1996), “Nodal methods: analysis, performance and fast iterative solvers”, PhD Thesis, University of British Columbia.
9. Wesseling P. (1992), “An introduction to multigrid methods”, John Wiley & Sons.

10. Scheilchl R. (2000), "Parallel solvers for transient multigroup neutron diffusion equations", *Int. J. Numer. Meth. Eng.* **47**, 1751-1771.
11. Hutchinson B.R. and Raithby G.D. (1986), "A multigrid method based on the additive correction strategy", *Numer. Heat Transfer* **9**, 511-537.
12. Wachspress E.L. (1966), "Iterative solution of elliptic systems and applications to the neutron diffusion equations of reactor physics", Prentice-Hall.