

NE 255, Fa16
Acceleration and Preconditioning
November 1, 2016

We've talked about basic methods for inner, outer, and eigenvalue iterations. We've talked a bit about how and why some methods are faster than others as well as when some methods might struggle with converging quickly. We're next going to talk about the fundamental idea of preconditioning and acceleration methods. We'll then go into methods specifically used in Nuclear Engineering.

First recall that conditioning is one way to express the perturbation behavior of the mathematical problem. A *well-conditioned* problem is one in which all small perturbations of x lead to only small changes in $f(x)$. An *ill-conditioned* problem is one in which some small perturbation of x leads to a large change in $f(x)$. The condition number is a quantity used to express how well-conditioned a matrix or problem is. A small condition number corresponds to a well-conditioned problem, and vice versa.

The condition number of an $n \times m$ matrix \mathbf{A} is defined as

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| = \frac{\sigma_1}{\sigma_m}, \quad (1)$$

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$ are the singular values of \mathbf{A} . If \mathbf{A} is singular, its condition number is infinity.

A way to think about the condition number: the ratio $\frac{\sigma_1}{\sigma_m}$ can be interpreted as the eccentricity of the hyper-ellipse that is the image of an m -dimensional unit sphere under \mathbf{A} . When \mathbf{A} has a large condition number the largest principle semiaxis is much longer than the smallest principle semiaxis.

(again, taken from my dissertation)

Preconditioning

The new “grand challenge” problems facing the nuclear transport community are large and complex. Cutting edge methods are required to solve them. While we've discussed methods that enable the solution of such problems, low-cost preconditioners that can reduce the number of iterations needed for convergence will be invaluable. This is true for Krylov methods in particular because

the memory required and cost per iteration increase dramatically with the number of iterations. Krylov methods can converge very slowly for poorly conditioned systems.

The general idea of preconditioning is to transform the system of interest into another equivalent system that has more favorable properties, for example one with a smaller condition number. A preconditioner is a matrix that induces such a transformation by improving the spectral properties of the problem being solved. Let \mathbf{G} be a non-singular preconditioner, then $\mathbf{A}x = b$ can be transformed in the following ways:

$$\mathbf{G}^{-1}\mathbf{A}x = \mathbf{G}^{-1}b \quad \text{left preconditioning,} \quad (2)$$

$$\mathbf{A}\mathbf{G}^{-1}y = b, \quad x = \mathbf{G}^{-1}y \quad \text{right preconditioning, and} \quad (3)$$

$$\mathbf{G}_1^{-1}\mathbf{A}\mathbf{G}_2^{-1}y = \mathbf{G}_1^{-1}b, \quad \mathbf{G} = \mathbf{G}_1\mathbf{G}_2, \quad x = \mathbf{G}_2^{-1}y \quad \text{split preconditioning.} \quad (4)$$

If \mathbf{A} and/or \mathbf{G} are non-normal then each of the preconditioning constructs will likely give different behavior, though they will converge to the same answer because the matrices are similar and therefore have the same eigenvalues. Right preconditioning leaves the right hand side of the equation unaffected and does not change the norm of the residual, which is used for convergence testing in most iterative methods. Right preconditioning is usually preferred over left preconditioning for iterative solvers for this reason. The remaining discussion will be presented in right preconditioner format.

The matrix $\mathbf{A}\mathbf{G}^{-1}$ is typically not formed in practice. The preconditioner can be applied by using some method to solve $\mathbf{G}y = c \rightarrow y \approx \mathbf{G}^{-1}c$, or by otherwise implementing the action of \mathbf{G}^{-1} without ever explicitly forming and inverting \mathbf{G} . There are two extremes between which all other preconditioners lie: $\mathbf{G} = \mathbf{A}$, in which case the solution can be found directly, and $\mathbf{G} = \mathbf{I}$, which will have no effect.

Functionally, a good preconditioner should make the system easier to solve and result in faster convergence. It should also be cheap to construct and apply. These tend to be competing goals in that the easier a preconditioner is to construct and apply the less it typically does to improve convergence. A preconditioner is considered good if $\mathbf{A}\mathbf{G}^{-1}$ is not too far from normal and its singular values are clustered.

There are many different types of preconditioners, but they can be put into two general categories: **matrix-based** and **physics-based**. Matrix-based preconditioners rely entirely on the structure of the matrix \mathbf{A} regardless of the physics of the problem. That is, these methods do not change when the underlying problem changes. This can be a very useful property because matrix-based methods

are then broadly applicable and do not require any understanding of the physical problem. Extrapolation methods and incomplete factorizations are examples of matrix-based preconditioners.

Physics-based preconditioning uses knowledge about the physics of the problem in question to guide the creation of the preconditioner. This means that some methods only work with certain kinds of problems, and that the preconditioners may have to be tailored or adapted for different applications. However, such methods take advantage of knowing something about the problem and can be more effective than matrix-based methods for the range of problems for which they are intended. Rebalance and synthetic acceleration are examples of physics-based preconditioners.

Extrapolation Methods

Extrapolation methods were historically used to accelerate k -eigenvalue calculations. These tend to be based on simple iterative solvers or polynomial approximations. One step of a simple iterative method such as Jacobi, Gauss Seidel, or successive over relaxation (SOR) can be used at the outset of a problem to serve as a preconditioner. When applied to neutron transport, an overrelaxation method looks like $\mathbf{F}_{i+1} = \omega(\tilde{\mathbf{F}}_i - \mathbf{F}_i) + \mathbf{F}_i$ with $1 \leq \omega < 2$, where the unaccelerated new iterate for the fission source is designated $\tilde{\mathbf{F}}_i$.

Polynomial preconditioners create a matrix polynomial $\mathbf{G}^{-1} = p(\mathbf{A})$, where $p(\mathbf{A})$ serves as a polynomial approximation to \mathbf{A}^{-1} . Common polynomial choices are truncated Neumann series and Chebyshev polynomials. An advanced variation of this method is to determine the polynomial coefficients adaptively. When Chebyshev polynomials are applied to the transport equation, $\mathbf{F}_{i+1} = \mathbf{F}_i + \alpha_{i+1}(\tilde{\mathbf{F}}_i - \mathbf{F}_i) + \beta_i(\mathbf{F}_i - \mathbf{F}_{i-1})$, where α and β are iteration dependent. Neither of these extrapolation methods are widely used today as they have not been very successful in multi-dimensional problems. There are analogous versions of these methods for fixed source problems.

Rebalance

Rebalance methods are designed to accelerate iteration on the scattering or fission source by imposing a balance condition on the unconverged solution over either coarse or fine regions. If the region is coarse, this is a coarse-grid approximation preconditioner since fine-grid physics are excluded. If the region is fine, this is a local approximation preconditioner where short range effects are permitted and long range interactions are excluded.

In either case, the unaccelerated solution is multiplied by a constant in each region that satisfies the balance equation. The rebalance process adjusts the average amplitude of the flux over a region and the iteration adjusts the space-angle distribution. The notion is that these processes work in concert to eliminate all error modes simultaneously.

Coarse mesh rebalance (CMR) has been more successful than the fine version and is therefore used more often. In practice, choosing regions properly must be done with care. Traditional forms of CMR can be unstable for problems where the spatial mesh is large compared to the neutron mean-free-path and where the scattering ratio is close to unity.

Incomplete Factorizations

The idea is to partially factor the matrix \mathbf{A} such that the resulting matrices are sparse enough to take advantage of sparse solver methods, but close enough to \mathbf{A} to be valuable as preconditioners.

Incomplete Cholesky (IC) or incomplete LU (ILU) factorization have both been extensively used within the nuclear community. The Cholesky version is used when \mathbf{A} is symmetric and LU when non-symmetric. Factorizations typically destroy sparsity, resulting in matrices that are denser than the originals and making their use more costly. Different variations of incomplete factorization methods address this by permitting the new matrices to have values only in positions where \mathbf{A} has values, allowing a prescribed amount of fill in, or only saving entries greater than some tolerance.

Synthetic Acceleration

In synthetic acceleration a low-order approximation to the transport operator is used to accelerate the full transport problem. If a system is discretized with a high-order method it can be preconditioned with a lower-order approximation. The low-order method is often much sparser than the original, but still captures the general behavior of the problem. This is generally done with either the diffusion equation, giving diffusion synthetic acceleration (DSA), or a transport equation that is simpler than the one actually being solved, giving transport synthetic acceleration (TSA). These can be used for fission or fixed source problems.

Synthetic methods have at least two iteration stages. The simplest way to illustrate this is to use source iteration for the within-group solver. The first step is a transport sweep, giving:

$$\mathbf{L}\psi^{(l+\frac{1}{2})} = \mathbf{S}\psi^{(l)} + q, \quad l \geq 0. \quad (5)$$

Here l is the iteration index, \mathbf{L} is the transport operator, \mathbf{S} is the scattering matrix, and q is the external source. Equation (5) is subtracted from the exact equation to get an expression for an exact additive correction:

$$(\mathbf{L} - \mathbf{S})(\psi - \psi^{(l+\frac{1}{2})}) = \mathbf{S}(\psi^{(l+\frac{1}{2})} - \psi^{(l)}) . \quad (6)$$

This cannot be solved exactly, but the idea of synthetic methods is to find an $\mathbf{G}^{-1} \approx (\mathbf{L} - \mathbf{S})^{-1}$ that will be easier to evaluate than $(\mathbf{L} - \mathbf{S})^{-1}$. Equation (5) is solved by a high-order scheme and Equation (6) is solved with a low-order approximation. If the synthetic system converges, it must satisfy the original transport equation.

Original DSA formulations exhibited instability in some instances. In 1977 Alcouffe demonstrated with diamond difference that stability can be obtained by using a consistent spatial differencing scheme. The transport and diffusion equations cannot be discretized independently or the diffusion equation may not satisfy the original transport equation. The discretization of the diffusion equation must therefore be derived from the discretization of the transport equation such that they are consistent. Alcouffe's idea has since been extended to accelerate currents, to more dimensions, and to different spatial discretizations.

DSA can be implemented a few different ways, each of which performs a transport sweep and then solves a corrected diffusion equation. Different terms in the diffusion equation can be corrected for different problem types. DSA has been found to degrade in the presence of material discontinuities in addition to when it is not consistently derived. However, Warsa et al. found that when DSA is used as a preconditioner for Krylov iterative methods, it is effective even when only partially consistent.

To address the concerns about spatial discretization associated with DSA, a low-order transport solve could be used to find the correction term instead. Larsen and Miller developed a method that neglects scattering in the low-order transport equation, but this can cause instability in multi-dimensional problems. Ramone et al. proposed including some scattering by using a tunable parameter. A high-order transport equation is solved first, then a transport equation with a coarse quadrature and reduced scattering is solved to find the correction that is applied to the new iterate of the scalar flux. This algorithm can be viewed as Richardson iteration with a coarse operator preconditioner. Synthetic acceleration methods are widely used and actively researched in the transport community today.

DSA in Transport

DSA is typically used to precondition the within-group transport equation, $(\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS}_{gg})\phi_g = \mathbf{DL}^{-1}Q_g$, which works very well when using a Krylov method. High-frequency error modes are what often cause instability in DSA. Krylov iteration, like iterative methods in general, will rapidly damp such oscillatory modes. Eliminating the high-frequency error enables the removal of the consistency requirement since DSA is not likely to fail when those error modes are gone. This means DSA can be applied successfully for a variety of spatial discretizations.

The DSA-preconditioned one-group equation is:

$$(\mathbf{I} + \mathbf{PC}^{-1}\mathbf{RS})(\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS})\phi = (\mathbf{I} + \mathbf{PC}^{-1}\mathbf{RS})\mathbf{DL}^{-1}\bar{Q}. \quad (7)$$

Here \mathbf{C} is the diffusion operator; \mathbf{R} is the restriction operator that maps the transport solution onto the diffusion vector; and \mathbf{P} is the projection operator that maps the diffusion vector onto the transport solution. Most codes do not actually form these operators, instead they solve the diffusion equation and update the ϕ_{00} moments. In practice this means that for a Krylov iteration

$$\mathbf{C}z = \mathbf{RS}(\mathbf{I} - \mathbf{DL}^{-1}\mathbf{SM})v \quad (8)$$

is solved for each group. DSA has been found to be beneficial for diffusive problems with high scattering ratios that are close to being isotropic.

Multigrid Methods

Multigrid methods take advantage of the fact that the error in any solution guess, v for the system $\mathbf{A}u = b$, can be written as a combination of Fourier modes. [The algebraic error is given by $e = u - v$; since e cannot be calculated exactly (otherwise the answer would be known), the residual $r = b - \mathbf{A}v$ is used instead.] A Fourier mode is described by a wavenumber, k , which denotes the frequency of oscillation. The j th Fourier mode is

$$e_j = \sin\left(\frac{jk\pi}{n}\right), \quad 0 \leq j \leq n, \quad 1 \leq k \leq n-1. \quad (9)$$

There are k half sine waves that comprise e on the domain of the problem. The term e_k indicates an e with wavenumber k . The k th wave has $\frac{k}{2}$ full sine waves with a wavelength of $\frac{2}{k}$. Modes with wavenumbers in the range $1 \leq k \leq \frac{n}{2}$ are called low-frequency or smooth modes and those in $\frac{n}{2} \leq k \leq n-1$ are called high-frequency or oscillatory modes. Figure 1 shows what a few

different modes look like.

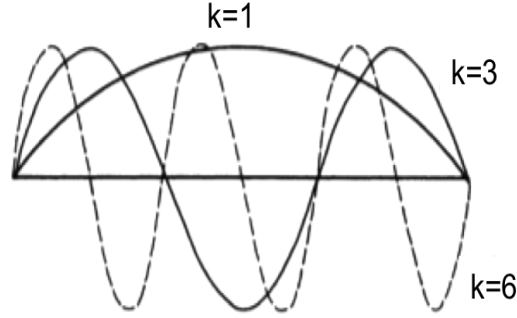


Figure 1: Three Fourier Modes

Iterative methods, also referred to as smoothers or relaxers, remove high frequency error components very quickly, but take a long time to remove the low frequency components.

The idea of multigrid methods is to take advantage of the smoothing effect by making smooth error look oscillatory so it can be removed more easily. Error that is low frequency on a fine grid can be mapped onto a coarser grid where it is oscillatory. A certain grid will be denoted as Ω^h where h is the grid spacing, and vectors on that grid are u^h . This change can be thought of as increasing the number of oscillations per grid point. Look at Figure 2 for an example. When $n = 12$, the maximum number of half-sine waves is 12, so $k = 4$ is relatively smooth. When mapped to $n = 6$, $k = 4$ is relatively oscillatory.

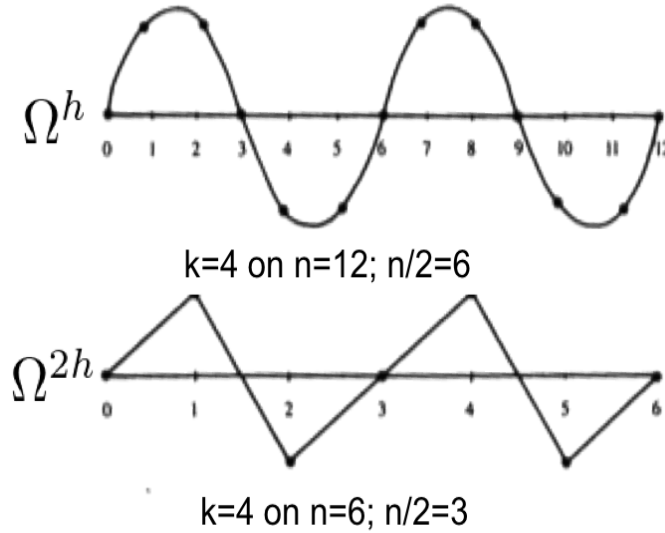


Figure 2: Relative Oscillation of a Fourier Mode Mapped Between Two Grids

Using this information, multigrid methods can be understood. The error is mapped from a fine grid to a coarser grid. A smoother is applied on the coarse grid to remove the newly oscillatory error components. The result is mapped back to the fine grid and used to correct the solution there. A few more relaxations are done back on the fine grid. That whole process is called a v-cycle. A call to this method is symbolized as $v^h \leftarrow \mathbf{G}(v^h, b^h)$ and is effectively doing $v^h \approx \mathbf{G}^{-1}b^h$. A v-cycle algorithm is shown here:

Relax ν_1 times on $\mathbf{A}^h u^h = b^h$ on the fine grid Ω^h using initial guess v^h .

Compute the residual $r^h = b^h - \mathbf{A}v^h$.

Using some restriction operator, \mathbf{R}_h^{2h} , restrict the residual to a coarser grid: $r^{2h} = \mathbf{R}_h^{2h}r^h$.

Solve the residual equation $\mathbf{A}^{2h}e^{2h} = r^{2h}$ on the coarse grid Ω^{2h} .

Using some prolongation operator, \mathbf{P}_{2h}^h , prolong (interpolate) the coarse grid error back to a finer grid: $e^h = \mathbf{P}_{2h}^h e^{2h}$.

Add the error to the fine grid guess: $v^h \leftarrow v^h + e^h$.

Relax ν_2 times on $\mathbf{A}^h u^h = b^h$ on Ω^h to get an improved solution.

There are many variations of how multigrid methods are put together. All methods, however, are built on the basic v-cycle correction scheme. A V-cycle is an extension of the v-cycle. Instead of only using two grids, many grids are used. The problem is restricted from grid to grid until it is on some grid which is coarse enough to directly invert the equations. Then the errors are prolonged back up the chain, continuously correcting on finer grids, until the finest grid is reached. Multigrid methods are differentiated by how many times the V-cycle is done and how many grids are used.

Multigrid methods can be thought of as stationary iterative schemes that can be used alone or as accelerators for other methods. In the past these methods were highly problem specific and only applied to second-order elliptic PDEs. Over time they have been extended to different problem types, geometries, and discretizations.

Two Grid and Transport Two Grid

The *two-grid acceleration method* developed by Adams and Morel was one of the earlier spatial multigrid methods. It is intended to accelerate convergence of the outer iterations for the transport equation when upscattering is present; the outer iteration method is Gauss Seidel and the method

is only applied to upscattering groups. The original work was done for slab geometries with linear discontinuous (LD) discretization.

Adams and Morel create a within group error equation by subtracting the GS equation from the transport equation. If the error in iteration k is ϵ^k and the l th moment of the residual is R_l^k , then for group g in 1-D this gives:

$$\mu \frac{\partial \epsilon_g^{k+1}(\mu)}{\partial x} + \Sigma_{t,g} \epsilon_g^{k+1}(\mu) = \sum_{l=0}^L \frac{2l+1}{4\pi} \left(\sum_{g'=1}^G \Sigma_{s,g' \rightarrow g,l} \epsilon_{g',l}^{k+1} + R_{g,l}^{k+1} \right) P_l(\mu) , \quad (10)$$

$$\epsilon_g^{k+1}(\mu) = \psi_g(\mu) - \psi_g^{k+1}(\mu) , \quad (11)$$

$$\epsilon_{g',l}^{k+1} = 2\pi \int_{-1}^1 \epsilon_{g'}^{k+1}(\mu') P_l(\mu') d\mu' , \quad (12)$$

$$R_{g,l}^{k+1} = \sum_{g'=g+1}^G \Sigma_{s,g' \rightarrow g,l} (\phi_{g',l}^{k+1} - \phi_{g',l}^k) . \quad (13)$$

The diffusion approximation is applied to Equation (10), giving a coarse grid equation (isotropic, one-group, diffusion equation). The fine grid is the S_N equations for the upscattering groups.

It is assumed that the zeroth moment of the error is a product of a spectral shape function, ξ_g , and a space-dependent modulation function, $E(x)$, giving

$$\epsilon_{g,0}^{k+1}(x) = E(x) \xi_g , \text{ and} \quad (14)$$

$$\sum_{g=1}^G \xi_g = 1 . \quad (15)$$

The spectral shape function corresponds to the slowest converging error mode, i.e. the mode to be eliminated. To find the shape function, Fourier analysis is performed on the GS iterative method. The zeroth moment of the cross sections is used to form the Fourier matrix, and then an eigenproblem can be formed. The spectral radius of the isotropic GS matrix is the eigenvalue, and the corresponding eigenvector is the shape function. Because the shape function is dependent on materials, one such calculation must be done for each material region. Note that by using the zeroth moment only the isotropic component of the solution is accelerated.

To restrict the residual, which is used as the source, to the coarse grid in the angular dimension, the anisotropic terms are truncated. To restrict to one energy group, the residual is simply summed in energy. The coarse grid equation is then solved on the coarse grid for the error. That error is prolonged to the fine grid and used to correct the flux there. The anisotropic terms are not recovered in the prolongation. The shape functions are used to expand the one group diffusion

solution (error) into multiple groups.

The coarse grid diffusion equations that result from Equations (10) and (14) differ slightly in form from the standard diffusion equation in that there is an extra term containing the gradient of the shape function. This is zero in homogeneous regions, but undefined at material interfaces. Adams and Morel found that neglecting the gradient term all together still gave good results for their test problems. This may not be true for more complex cases.

The two-grid method requires the diffusion operator to be consistent with the transport operator, just like in DSA. This requirement can be difficult to meet for multi-dimensional problems, particularly for some spatial discretizations.

The *Transport two-grid (TTG) acceleration scheme* is based on the one developed by Adams and Morel. The original two-grid method was modified by Evans et al. to make it applicable for multi-dimensional and multi-material problems by using a one-group transport equation instead of the diffusion equation. Adams and Morel showed that the slowest converging spatial modes are diffusive and can be exactly computed in the infinite homogeneous case. To preserve this, the TTG method gives the correct error estimation in that limit.

The cross sections used in TTG are therefore calculated to give the same energy-collapsed cross sections as the diffusion equation for the infinite homogeneous case, as seen in Equations (17) and (18). TTG solves a one-group transport equation for the low-order error in each GS iteration, where $[g1, g2]$ is the group range of the upscatter block:

$$\hat{\Omega} \cdot \nabla \psi_\epsilon + \bar{\Sigma} \psi_\epsilon = \frac{1}{4\pi} \bar{\Sigma}_s \phi_\epsilon + \frac{1}{4\pi} \bar{R}, \quad (16)$$

$$\bar{\Sigma} = \frac{1}{\sum_{g=g_1}^{g_2} \frac{1}{\bar{\Sigma}^g} \zeta^g}, \quad (17)$$

$$\bar{\Sigma}_s = \frac{1}{\sum_{g=g_1}^{g_2} \frac{1}{\bar{\Sigma}^g} \zeta^g} - \sum_{g=g_1}^{g_2} (\Sigma^g \zeta^g - \sum_{g'=g_1}^{g_2} \Sigma_{s0}^{gg'} \zeta^{g'}). \quad (18)$$

The spatial components of the error are ψ_ϵ and ϕ_ϵ ; \bar{R} is the residual. Just as in the original method, it is assumed that the error is separable in space and energy at each iteration: $\epsilon_g^k = \phi_g - \phi_g^k = \phi_\epsilon(\vec{r}) \zeta^g$, where ζ^g is a material-dependent spectral function.

To execute the TTG scheme, a transport sweep is conducted in each group, a residual is calculated, the low-order transport solve described above is performed, an error form of the transport sweep is

conducted, and the scalar flux is updated. All of that can be seen in the following equations:

$$\mathbf{L}_g \psi_g^{k+\frac{1}{2}} = \mathbf{M}(\mathbf{S}_{gg} \phi_g^{k+\frac{1}{2}} + \sum_{g'=g_1}^{g-1} \mathbf{S}_{gg'} \phi_{g'}^{k+\frac{1}{2}} + \sum_{g'=g+1}^{g_2} \mathbf{S}_{gg'} \phi_{g'}^k) + q_{e,g} , \quad (19)$$

$$R_g^{k+\frac{1}{2}} = \mathbf{M} \sum_{g'=g+1}^{g_2} \mathbf{S}_{gg'} (\phi_{g'}^{k+\frac{1}{2}} - \phi_{g'}^k) , \quad l = m = 0 , \quad (20)$$

$$\bar{\mathbf{L}} \psi_\epsilon = \mathbf{M} \bar{\mathbf{S}} \phi_\epsilon + \bar{R} , \quad l = m = 0 , \quad (21)$$

$$\phi_g^{k+1} = \phi_g^{k+\frac{1}{2}} + \phi_\epsilon \zeta^g , \quad l = m = 0 . \quad (22)$$

The operators with over-bars use the collapsed cross sections given in Equations (17) and (18), and $\bar{R} = \sum_{g=g_1}^{g_2} R_g^{k+\frac{1}{2}}$.

The ζ_g term is calculated from the eigenvalue problem obtained through Fourier analysis of the GS method using isotropic scattering, just like the original two-grid method:

$$(\Sigma - \mathbf{S}_L + \mathbf{S}_D)^{-1} \mathbf{S}_U \zeta = \rho \zeta , \quad (23)$$

where Σ is the diagonal, total cross section matrix. As with the original two-grid method, the TTG method is limited to correcting only the isotropic flux moments. This is an adequate limitation as these moments are dominant in thermal groups where upscattering is most prevalent.

The additional cost of the method is like solving one extra group, so for cases with many upscattering groups the cost can be amortized. The acceleration equation can be preconditioned with DSA for additional speed. Finally, a reduced quadrature set can be used when solving Equation (21) to limit the cost. Some test problems have shown TTG to be quite effective in improving the speed of convergence for upscatter problems when compared to unaccelerated GS.