

# Acceleration and Stabilization of Algebraic Multigrid Solver Applied to Incompressible Flow Problems

Aleksandar Jemcov\*

*Ansys/Fluent Inc., Lebanon, NH, 03766, USA*

Joseph P. Maruszewski<sup>†</sup>

*Ansys/Fluent Inc., Lebanon, NH, 03766, USA*

Hrvoje Jasak<sup>‡</sup>

*Wikki Ltd., London, W8 7PU, UK*

Acceleration and stabilization of the Algebraic Multigrid solver (AMG) through  $n$ -th order Recursive Projection Method (RPM( $n$ )) is described. It is shown that significant acceleration can be obtained if RPM( $n$ ) is applied to AMG during the inner iteration loop in a typical implicit incompressible CFD codes. In addition to accelerating the solution, RPM( $n$ ) provides increased stability to the AMG Solver extending it beyond its normal range of applicability in terms of matrix conditioning and M-matrix properties. RPM( $n$ ) algorithm allows the use of agglomerative AMG solver with simple smoothers to be effectively applied to matrices that are not an M-matrix. Theoretical foundations of RPM( $n$ )-AMG algorithm are presented with some practical aspects of the algorithm implementation. Numerical experiments that involve pressure correction matrices of various sizes that appear in segregated pressure based algorithms together with coupled momentum and pressure matrices stemming from coupled pressure based algorithms are used to illustrate the effectiveness of the method.

## Nomenclature

$u_i$	Velocity, m/s
$p$	Static pressure, Pa
$p'$	Pressure correction, Pa
$\mu$	Dynamic viscosity, kg/(ms)
$\rho$	Mass density, kg/m <sup>3</sup>
$\Gamma$	Boundary of computational domain
$\Omega$	Computational domain
$x_i$	position, m
<b>A</b>	Linear system matrix
<b>u</b>	Vector of unknowns
<b>b</b>	Right hand side vector
<b>P</b>	Projector on unstable space
<b>Q</b>	Projector on stable space
<b>Z</b>	Basis of unstable space
<b>R</b>	Iteration matrix
<b>M</b>	Preconditioning matrix
<b>N</b>	Defect matrix
<b>S</b>	Smoother matrix
<b>L</b>	Lower triangular matrix
<b>U</b>	Upper triangular matrix
<b>D</b>	Diagonal matrix

---

\*Lead Development Engineer, Development Department, 10 Cavendish Court

<sup>†</sup>Principal Development Engineer, Development Department, 10 Cavendish Court

<sup>‡</sup>Director, 33 Palmerston House, 60 Kensington Place, AIAA Member

<b>p</b>	Unstable vector
<b>q</b>	Stable vector
<b>T</b>	Upper triangular matrix in $QR$ -decomposition
<b>B</b>	Rectangular matrix of $q$ -differences
<b>C</b>	AMG restriction matrix
<b>r</b>	residual vector
$s_r$	spectral radius
$\lambda$	eigenvalue
$\mathcal{P}$	Unstable space
$\mathcal{Q}$	Stable space
$\mathcal{R}$	Real axis
$e_i$	$i$ -th unit vector
$n_i$	$i$ -th component of the face normal
$a$	matrix coefficient

*Subscript*

$i, j$	Tensor indexes
$l$	off-diagonal index
$p$	diagonal index

*Superscript*

$u$	momentum coefficient
$p$	pressure correction coefficient
$\nu$	Iteration number
$b$	boundary value
$0$	initial value
$h$	AMG level
$T$	Transpose

*Overlines*

$\hat{\phantom{x}}$	Orthogonal matrix
$\tilde{\phantom{x}}$	Block matrix

## I. Introduction

Today's CFD problems are characterized by increasing physical and numerical complexities requiring implementation of algorithms that lead to efficient solvers. This statement is particularly true in application of CFD methodology to the solution of various industrial problems ranging from internal and external aerodynamics where geometric complexity of plays important role, to chemically reacting multiphase flows in which physical complexity is significant. These complexities often represent the limiting factors for the efficient solution of the numerical problem and sometimes the limitations are so severe that CFD solver cannot handle them. This is recognized as stalled convergence or even divergence of the solution algorithm.

In particular, pressure based solvers are frequently used to solve low speed problems characterized by geometric and physical complexity. An important step in the the iterative procedure employed by SIMPLE-like algorithms<sup>1,2</sup> is the solution of a large systems of linear equations that are obtained by discretizing the momentum and continuity equations. Efficient iterative solvers exist provided that the resulting matrix has M-matrix properties.<sup>3,4</sup> In other words, if the application of discretization to the momentum and continuity equation results in a diagonally dominant matrix in which the diagonal is positive and off-diagonal is strictly non-positive, stationary iterative methods such as Jacobi and Gauss-Seidel sweeps produce a convergent sequence of solutions. It is well known that stationary iterative methods act efficiently on high frequency error component resulting in fast removal of this error component from the solution. After the high frequency error is removed, the rate of convergence deteriorates significantly<sup>5,6</sup> resulting in long computation times. This problem becomes very pronounced with increasing grid size and many iterations are required to reduce residuals to an acceptable level.

A remedy for this problem was given by the Multigrid strategy of sequencing progressively coarser grids in which the error from the finer grids are projected, thus transforming the low frequency error content of the fine grid to a high frequency error content on the coarse grid.<sup>4,5,7</sup> In contrast to geometrically based multigrid,

in Algebraic Multigrid no coarse grids are required. Instead, the sequence of progressively coarser matrices and corresponding restriction and projection operators are created by the process of agglomeration.<sup>5,6</sup> Once the hierarchy of coarse matrices is created, basic iterative methods are used at each level to smooth out the error and compute a correction that are interpolated to the fine grid. This particular strategy significantly improves the convergence properties of basic iterative solvers by allowing them to efficiently remove high frequency error content from the solution vector. Moreover, it can be shown that the AMG method applied to linear system obtained by discretizing strongly elliptic operators reduce the error by one order of magnitude per multigrid cycle.<sup>6</sup>

However, the underlying assumption in AMG is that the matrix has M-matrix properties in order to benefit from the acceleration strategy. Any departure from this will either lead to increased number of iterations, to stalled residual or even divergence. In fact, theoretical properties of AMG acceleration are rarely observed in practice since the underlying matrix rarely represent the discrete version of a strongly elliptic operator. Quality of the computational mesh plays an important role in determining the convergence rates of AMG solvers. Nowadays use of unstructured meshes is very common, and the loss of orthogonality in the computational mesh sometimes leads the discretization methods to produce matrices that do not satisfy the M-matrix property. Also, application of cell-centered finite volume schemes to physically complex problems sometimes produces matrices that violate M-matrix property leading to loss of convergence AMG solver.

In order to recover AMG convergence rates, a modification to the AMG solver is needed. One approach to allowing the AMG solver to converge on matrices that weakly violate the M-matrix property is to have an intelligent agglomeration process that efficiently identifies the smooth error and is capable of creating the agglomeration stencils that follow the smooth error.<sup>6</sup> However, the drawback of this approach is its lack of universality because heuristic criteria must be used to identify smooth error and the performance of AMG will be directly linked to the relevance of the heuristic criteria to the underlying characteristics of the problem. Hence, efficient agglomeration in one particular problem can also be inefficient in some other problem that does not have the same error characteristics. It is preferable to use a general version of agglomeration method<sup>8</sup> and enhance the AMG strategy to handle matrices that violate the M-matrix property.

The solution to this problem can be found in enhancing the AMG solver with the Recursive Projection Method (RPM)<sup>11</sup> that stabilizes general fixed point algorithms. The basic idea of the RPM method is to identify the unstable space of the fixed point function represented by the iterative matrix of AMG solver<sup>5</sup> and to project the solution onto this unstable space. To achieve this, the basis of the unstable space must be identified and corresponding projectors can be formed. Once the projectors are known, the AMG fixed point function can be decomposed into a stable and unstable parts allowing the appropriate algorithms to be used on each parts. In this way, the AMG strategy can remain stable even if the underlying matrix does not have the M-matrix property because this particular error mode can be removed from the linear system and handled by special methods. At the end of each AMG cycle the solution can be reconstructed resulting in convergent solution. Furthermore, it is possible to enhance the convergence properties of the RPM algorithm by constructing the  $n$ -th order RPM algorithm.<sup>12</sup>

## II. Equation Discretization

Incompressible flow problems are described by the Navier-Stokes equations:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_i u_j) = -\frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j} \left( \mu \frac{\partial u_i}{\partial x_j} \right) \quad \text{in } \Omega, \quad (2)$$

with the corresponding boundary conditions

$$\begin{aligned} u(x_i, t) &= u_b(t), \\ p(x_i, t) &= p_b(t) \quad \text{on } \Gamma \end{aligned} \quad (3)$$

and corresponding initial conditions

$$u(x_i, 0) = u_0(x_i), \quad (4)$$

$$p(x_i, 0) = p_0(x_i) \quad \text{in } \Omega.$$

Without any loss of generality, consider a steady state flow and integrate Eq. (1) and Eq. (2) over a typical finite volume  $\Omega_k$  bounded by the boundary  $\Gamma_k$ :

$$\oint_{\Gamma_k} u_j n_j d\Gamma = 0 \quad (5)$$

$$\oint_{\Gamma_k} \rho u_i u_j n_j d\Gamma = - \oint_{\Gamma_k} p e_j n_j d\Gamma + \oint_{\Gamma_k} \mu \frac{\partial u_i}{\partial x_j} n_j d\Gamma. \quad (6)$$

Using the standard collocated arrangement for primitive variables and cell centered finite volume scheme<sup>9</sup> the following systems of linear equations can be obtained:

$$a_p^u u_p + \sum_l a_l^u u_l = b_p^u, \quad (7)$$

$$a_p^p p'_p + \sum_l a_l^p p'_l = b_p^p. \quad (8)$$

This particular form of the discrete continuity and momentum equations correspond to so-called segregated pressure based method where Picard linearization is used. It can be written in the following matrix form:

$$\mathbf{A}_p \mathbf{p}' = \mathbf{b}_p, \quad (9)$$

$$\mathbf{A}_u \mathbf{u} = \mathbf{b}_u. \quad (10)$$

Similarly, coupled pressure based system can be obtained by combining the pressure matrix from Eq. (9) and the momentum matrix from Eq. (10) into one linear system:

$$\tilde{\mathbf{A}} \tilde{\mathbf{u}} = \tilde{\mathbf{b}} \quad (11)$$

In this case, matrix  $\tilde{\mathbf{A}}$  solution vector  $\tilde{\mathbf{u}}$  and right-hand-side  $\tilde{\mathbf{b}}$  have a block structure with the following coefficients in the  $k$ -th row and  $l$ -th column

$$\tilde{\mathbf{a}}(k, l) = \begin{bmatrix} a^{u_1 u_1} & a^{u_1 u_2} & a^{u_1 u_3} & a^{u_1 p} \\ a^{u_2 u_1} & a^{u_2 u_2} & a^{u_2 u_3} & a^{u_2 p} \\ a^{u_3 u_1} & a^{u_3 u_2} & a^{u_3 u_3} & a^{u_3 p} \\ a^{p u_1} & a^{p u_2} & a^{p u_3} & a^{p p} \end{bmatrix}_{(k, l)} \quad (12)$$

$$\tilde{\mathbf{u}}(k) = \begin{bmatrix} u^1 \\ u^2 \\ u^3 \\ p' \end{bmatrix}_{(k)} \quad (13)$$

$$\tilde{\mathbf{b}}(k) = \begin{bmatrix} b^{u_1} \\ b^{u_2} \\ b^{u_3} \\ b^{p'} \end{bmatrix}_{(k)} \quad (14)$$

Given the fact that the original system of equations Eq. (1), Eq. (2) was nonlinear and since Picard iterations were used to linearize the system, some fixed point algorithm is needed to guide the nonlinear system to convergence. In pressure based methods, the SIMPLE family of algorithms<sup>1,9</sup> is a popular and efficient choice for a fixed point algorithm. At every outer nonlinear SIMPLE step, a solution to linear system of equations given by Eq. (9) and Eq. (10) or Eq. (11) is required. Once the solution fields are obtained, corrections to the pressure and velocity fields can be performed.<sup>1,9</sup>

An important step in SIMPLE-like algorithms is the solution of linear systems of equations and overall performance of the nonlinear algorithm will depend on the efficiency and accuracy of the solution of the linear systems of equations Eq. (9), Eq. (10) or Eq. (11). Traditionally, linear systems of equations are usually solved with an AMG solver that utilizes simple smoothers such as Gauss-Seidel or ILU(0) stationary methods.<sup>8,9</sup> The major reason for this selection of the linear system solution methodology is in its simplicity, memory and computational efficiency, and universality. In contrast to Krylov space methods,<sup>3</sup> AMG is fast and stationary in nature due to the underlying smoother properties, whereas Krylov space methods are nonlinear and require a good preconditioner in order to speed up the convergence.<sup>9</sup>

The difficulty with the linear systems in incompressible flow solvers based on the pressure correction method is mostly associated with the inversion of pressure correction matrix. The pressure correction equation is formed to create a pressure field that enforces divergence-free condition on velocity, it is important to have a good solution to the linear system in order to maintain conservation. The condition number of the pressure matrix can be high and efficient and robust iterative solvers are required to solve it. It is also important to have an efficient and reliable linear solver applied to discrete momentum equations but since pressure correction matrix governs the mass conservation of the scheme, the cost of pressure solution dominates. In the rest of this paper the pressure matrix is considered more closely although all of the methods presented for are also applicable to linear systems stemming from the discretization of momentum equation.

### III. Algebraic Multigrid Solver

Given the matrix problem in the form

$$\mathbf{A}\mathbf{u} = \mathbf{b} \quad (15)$$

and assuming matrix  $\mathbf{A}$  has the M-matrix property, convergent splitting<sup>10</sup> can be introduced:

$$\mathbf{A} = \mathbf{M} - \mathbf{N} \quad (16)$$

This is used to create a fixed-point algorithm characterized by the fixed-point function  $F$

$$\mathbf{x}^{(\nu+1)} = F(\mathbf{x}^{(\nu)}) \quad (17)$$

The fixed-point function  $F$  is given by

$$F(\mathbf{x}^{(\nu)}) = (\mathbf{M}^{-1}\mathbf{N})\mathbf{x}^{(\nu)} + \mathbf{M}^{-1}\mathbf{b} \quad (18)$$

A particular choice of the splitting in Eq. (16) can lead to different stationary methods such as Gauss-Seidel, Jacobi, SOR, etc.<sup>4,5,10</sup> If upper  $\mathbf{U}$ , lower  $\mathbf{L}$  and diagonal  $\mathbf{D}$  of matrix  $\mathbf{A}$  are defined, it is possible to represent some of the stationary methods in matrix form. The Jacobi iteration matrix is defined as

$$\mathbf{R} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}, \quad (19)$$

whereas the symmetric Gauss-Seidel iteration matrix is given by

$$\mathbf{R} = \mathbf{I} - \mathbf{L}^{-1}\mathbf{D}\mathbf{U}^{-1}\mathbf{A}. \quad (20)$$

It is a well known fact that the performance of fixed point methods is dependent on the spectral radius of the Jacobian matrix of the fixed-point function Eq. (18)

$$\mathbf{R} = \frac{\partial F(\mathbf{x}^{(\nu)})}{\partial \mathbf{x}^{(\nu)}} = \mathbf{M}^{-1}\mathbf{N} = \mathbf{I} - \mathbf{M}^{-1}\mathbf{A}. \quad (21)$$

The Jacobian matrix  $\mathbf{R}$  in the context of fixed-point methods used for an iterative solution of a linear system of equations is usually called *the iteration matrix*.

If error  $\mathbf{e}^{(\nu)}$  of the current iteration  $\nu$  is defined as the difference between the current iterate and true solution

$$\mathbf{e}^{(\nu)} = \mathbf{u}^{(\nu)} - \mathbf{x}^* \quad (22)$$

the iteration matrix  $\mathbf{R}$  governs the error propagation

$$\mathbf{e}^{(\nu+1)} = \mathbf{R}\mathbf{e}^{(\nu)} \quad (23)$$

Moreover, it can be shown that the following recursive property of the iteration matrix holds:

$$\mathbf{e}^{(m)} = \mathbf{R}^m \mathbf{e}^{(0)} \quad (24)$$

Based on this recursive property, the error is forced to zero

$$\lim_{m \rightarrow \infty} \mathbf{R}^m = 0 \quad (25)$$

if and only if the spectral radius of the iteration matrix is less than 1

$$s_r(\mathbf{R}) < 1 \quad (26)$$

where the spectral radius  $s_r$  is defined as

$$s_r(\mathbf{R}) = \max |\lambda(\mathbf{R})| \quad (27)$$

Furthermore, if the asymptotic convergence rate observed in reducing the error to  $10^{-d}$  is introduced as

$$c \geq -\frac{d}{\log_{10}(s_r(\mathbf{R}))}, \quad (28)$$

it can be shown that for the iteration matrices whose spectral radius approaches 1, the asymptotic convergence rate decreases sharply.

In practice, spectral content of the iteration matrix is very broad and with increasing size of the computational grid, some eigenvalues will be approaching the edge of the unit circle. Thus, with increasing problem size, linear system of equations becomes stiffer, requiring many iterations of stationary methods to achieve acceptable reduction of residual. A remedy to this problem was found in AMG, where a hierarchy of progressively coarser matrices are constructed in order to speed up the convergence.

In AMG, instead of solving the coarse system

$$\mathbf{A}^{2h} \mathbf{u}^{2h} = \mathbf{b}^{2h} \quad (29)$$

error correction solution is performed<sup>5,6</sup>

$$\mathbf{A}^{2h} \mathbf{e}^{2h} = -\mathbf{r}^{2h} \quad (30)$$

Once the error correction is obtained, it is possible to correct the fine level solution by adding the correction to the solution. In order to do correction and construct the coarse matrix, restriction  $\mathbf{C}$  and prolongation operators need to be obtained. The restriction operator defines the hierarchy of coarse matrices and prolongation operator performs the interpolation of the error from the coarse onto fine level. Since Galerkin projection is used in defining the coarse levels, prolongation operator will be a transpose of the restriction operator,  $\mathbf{C}^T$ . Coarse matrices are obtained by performing the operation of orthogonal projection

$$\mathbf{A}^{2h} = \mathbf{C} \mathbf{A}^h \mathbf{C}^T. \quad (31)$$

The construction of the restriction operator is performed through the process of agglomeration in which every row of the matrix  $\mathbf{A}$  is visited and off-diagonal coefficients that have a largest value are selected for agglomeration. The criterion for the selection of the largest off-diagonal coefficients is heuristic and usually based on some form of the following expression:

$$-a_{max}(k, l) > \theta \max_{k \neq l} (-a(k, l)), \quad 0 \leq \theta \leq 1. \quad (32)$$

This criterion decides which  $l$ -th equation strongly influences the  $k$ -th equation and therefore is a candidate for agglomeration. Depending on how the selection criterion is performed, two AMG agglomeration methods can be defined. In the first one, sets of equations are divided into coarse and fine sets, and interpolation coefficients are computed based on the fact that not all equations have a coarse representation.<sup>5,6</sup> This approach is called Selective AMG (SAMG). In agglomerative AMG (AAMG) every fine equation has a coarse representation and in this case interpolation coefficients are all equal. The underlying assumption is that coarse equations equally influence all neighboring equations leading to homogeneous error correction. AAMG<sup>8</sup> performs well when smooth error is homogeneously distributed in the equation neighborhood.

This is rarely the case in practical computations and results in increased number of iterations required to reduce residual compared to SAMG. However, increase number of iteration counts is offset by simplicity of implementation and low memory requirements compared to SAMG.

In order to illustrate the acceleration performance of AAMG, consider the discretization of two-dimensional scalar Laplace equation.

$$\Delta u = f. \tag{33}$$

The choice of Laplace equation is driven by desire to use a linear system that closely represent the pressure correction problem. In fact, the Laplace equation is used in many projection algorithms<sup>2</sup> as a pressure equation. If central differences are used on a structured grid, the matrix structure of the resulting linear system results in banded matrix pattern with main diagonal and super-diagonal and sub-diagonal bands. Second order finite differences discretization produces a diagonally dominant matrix which results in convergent splitting, Eq. (16). Convergence history for the grid size of  $25 \times 25$  is given in Fig. (1) and corresponding iteration matrix spectrum is given in Fig. (2).

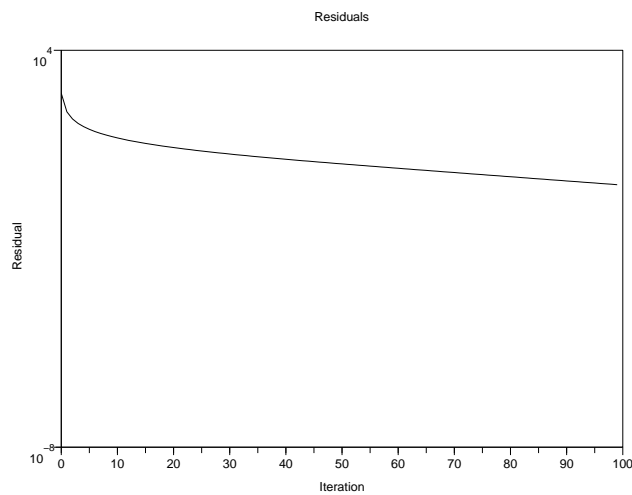


Figure 1. Gauss-Seidel convergence history for Laplace  $25 \times 25$  matrix.

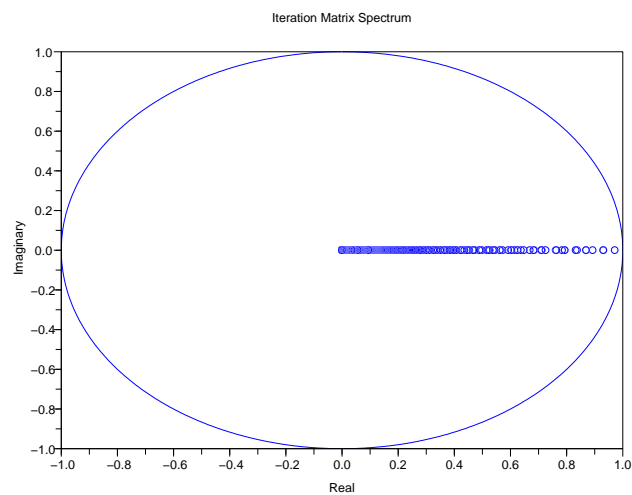


Figure 2. Gauss-Seidel iteration Laplace  $25 \times 25$  matrix spectrum.

The rate of convergence is slow even for this small problem due to unfavorable eigenvalues which approach

the edge of unit circle.

Although in practice multiple levels in AMG solver are usually created, two-level agglomerative AMG is used here to illustrate the AMG speed up. Iteration matrix of two-level AMG solver is given by

$$\begin{aligned}
 \mathbf{R} &= \mathbf{I} - \mathbf{M}^{-1}\mathbf{A} \\
 \mathbf{M}^{-1} &= \mathbf{S} + \mathbf{K} - \mathbf{S}\mathbf{A}\mathbf{K} \\
 \mathbf{S} &= \mathbf{L}^{-1}\mathbf{D}\mathbf{U}^{-1} \\
 \mathbf{K} &= \mathbf{C}^T (\mathbf{C}\mathbf{A}^h\mathbf{C}^T) \mathbf{C}
 \end{aligned}
 \tag{34}$$

The effect of AMG on iteration matrix becomes apparent when we examine Fig. (3) and Fig. (4). Two-level

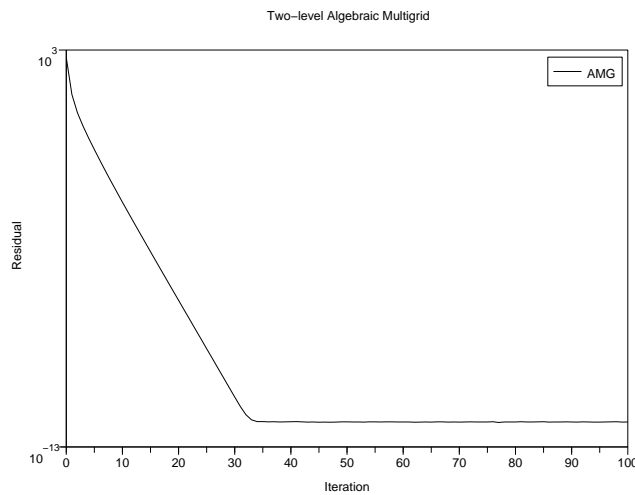


Figure 3. Two-level AMG convergence history for Laplace  $25 \times 25$  matrix.

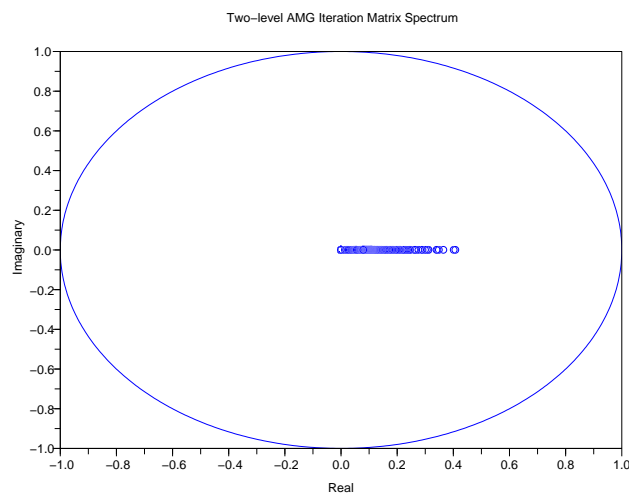
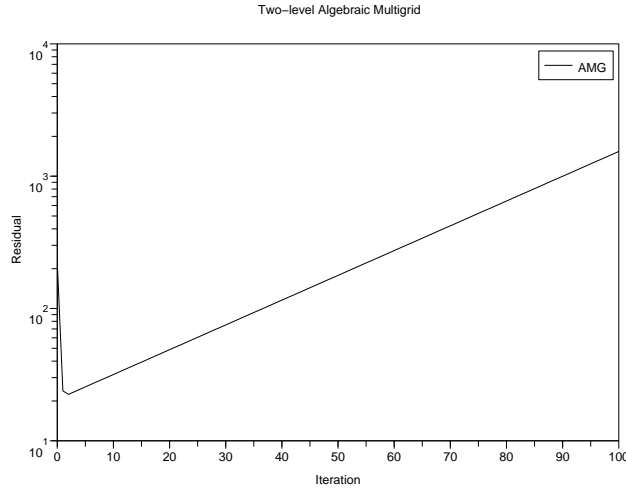


Figure 4. Two-level AMG iteration matrix Laplace  $25 \times 25$  spectrum.

AMG acts a preconditioner on symmetric Gauss-Seidel iterations and results in removal of eigenvalues that are close to the edge of unit circle, Fig. (4). The preconditioning properties of AMG solver can be seen in Eq. (34) where the iteration matrix of AMG solver directly modifies the preconditioning matrix  $\mathbf{M}$ . Further acceleration is possible if  $n$ -level AMG solver is used.

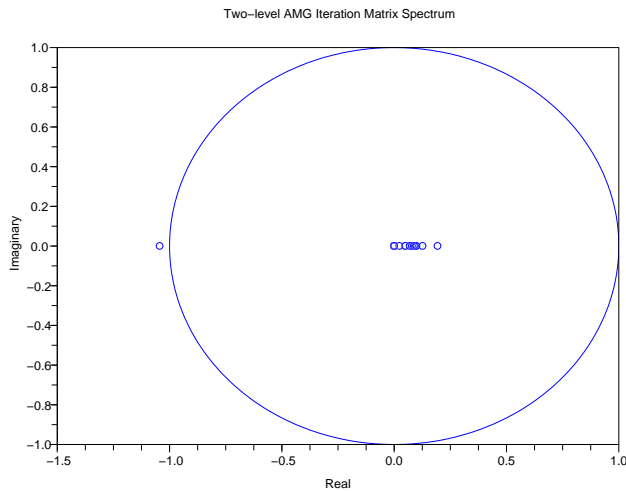
## IV. Recursive Projection Method

The underlying assumption in the previous discussion was that the discretization matrix  $\mathbf{A}$  had M-matrix properties. This guarantees the spectral radius of the iteration matrix is less than 1, resulting in a convergent fixed-point method Eq. (17). However, if any of the eigenvalues of the iteration matrix  $\mathbf{R}$  lie outside the unit circle in the complex plane, fixed-point iterations Eq. (17) diverge. If we consider the Laplace equation discretized with central finite differences where a pair of coefficients are artificially changed to violate the M-matrix properties, the AMG solver will diverge. In particular, for coarse discretization corresponding to the structured grid of the size  $5 \times 5$  and if coefficients  $a(5, 10) = a(10, 5) = -1.0$  are changed to  $a(5, 10) = a(10, 5) = -6.5$ , diagonal dominance of the matrix  $A$  is lost and M-matrix properties are violated. Application of the two level AMG solver will result in divergence as it was shown on Fig. (5).



**Figure 5. Two-level AMG convergence history for non-M-matrix.**

The loss of convergence is due to one eigenvalue lying outside the unit circle as shown in Fig. (6). If the error mode associated with this eigenvalue and corresponding eigenvector can be removed from the iteration matrix, standard AMG iterations will converge.



**Figure 6. Spectrum of Two-level AMG iteration matrix for non-M-matrix.**

This is the idea of recursive projection method (RPM).<sup>11</sup> RPM seeks to identify the unstable eigen-space

and corresponding unstable modes by constructing orthogonal projectors  $\mathbf{Q}$  and  $\mathbf{P}$  associated with stable and unstable eigen-spaces, respectively. If the approximate orthonormal basis of the unstable space  $\mathbf{Z}$  can be identified, then  $\mathcal{R}^n$  space to which solution belongs can be decomposed into a direct sum of a stable space  $\mathcal{Q}$  and unstable space  $\mathcal{P}$

$$\mathcal{R}^n = \mathcal{P} \oplus \mathcal{Q}. \quad (35)$$

Projectors  $\mathcal{P}$  and  $\mathcal{Q}$  are obtained from the eigen-basis  $\mathbf{Z}$  as follows:

$$\mathbf{P} = \mathbf{Z}\mathbf{Z}^T, \quad \mathbf{Q} = \mathbf{I} - \mathbf{Z}\mathbf{Z}^T. \quad (36)$$

Given the projectors  $\mathbf{P}$  and  $\mathbf{Q}$ , the solution to current iterate  $\mathbf{u}^{(\nu)}$  can be decomposed into a stable  $\mathbf{q}$  and unstable part  $\mathbf{p}$ :

$$\mathbf{u}^{(\nu)} = \mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}. \quad (37)$$

Similarly, fixed-point function, Eq. (18) can be also projected onto stable and unstable spaces:

$$\mathbf{p}^{(\nu+1)} = \mathbf{P} \left( \mathbf{R}\mathbf{u}^{(\nu)} + \mathbf{M}^{-1}\mathbf{b} \right) = \mathbf{P}F(\mathbf{p} + \mathbf{q}) \quad (38)$$

$$\mathbf{q}^{(\nu+1)} = \mathbf{Q} \left( \mathbf{R}\mathbf{u}^{(\nu)} + \mathbf{M}^{-1}\mathbf{b} \right) = \mathbf{Q}F(\mathbf{p} + \mathbf{q}) \quad (39)$$

Classical RPM<sup>11</sup> is obtained when one Newton step is applied to the implicit equation for the  $p$ -part given by Eq. (38)

$$\mathbf{p}^{(\nu+1)} = \mathbf{p}^{(\nu)} + \mathbf{Z} \left( \mathbf{I} - \mathbf{Z}^T \mathbf{R} \mathbf{Z} \right)^{-1} \mathbf{Z}^T \left( F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) - \mathbf{p}^{(\nu)} \right). \quad (40)$$

Similarly, the  $q$ -part given by Eq. (39) is given by the solution of the following equation:

$$\mathbf{q}^{(\nu+1)} = F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) - \mathbf{Z} \left( \mathbf{Z}^T F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) \right). \quad (41)$$

It is worth mentioning that in order to evaluate Eq. (41) and Eq. (40) one needs to determine the basis  $\mathbf{Z}$ . The rest of the function is known and it is already available in the form AMG cycle. This means that the  $q$ -part reuses the AMG solver and the additional code is related to the  $p$ -part. The effort to evaluate  $p$ -part is minimal since all that is required is the inversion of the matrix  $\mathbf{H} = \mathbf{I} - \mathbf{Z}^T \mathbf{R} \mathbf{Z}$ . This matrix has the linear size corresponding to the dimension of the unstable space and this is usually small typically of 2 to 6. In some cases this can grow up to 30 if there are many unstable modes. In most cases it is limited to dimension 2 to 6.

Typical RPM algorithm would take the following form:

- initialize  $q^0$  and  $p^0$
- Do until convergence
  - Solve  $\mathbf{q}^{(\nu+1)} = F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) - \mathbf{Z} \left( \mathbf{Z}^T F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) \right)$
  - Solve  $\mathbf{p}^{(\nu+1)} = \mathbf{p}^{(\nu)} + \mathbf{Z} \left( \mathbf{I} - \mathbf{Z}^T \mathbf{R} \mathbf{Z} \right)^{-1} \mathbf{Z}^T \left( F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) - \mathbf{p}^{(\nu)} \right)$
- $\mathbf{u}^{(final)} = \mathbf{p}^{(final)} + \mathbf{q}^{(final)}$

From this pseudo-algorithm it can be seen that RPM can be efficiently implemented around the existing AMG solver with minimal changes to the underlying AMG. Most of RPM logic is outside of the AMG code and it is only required to execute AMG code as a function call. This particular property of the RPM algorithm being implementation non-intrusive is of great value in software environments where very little is known about the actual implementation of fixed-point function. Therefore, it is also possible to treat legacy codes in a similar manner although some means of evaluating the Jacobian matrix Eq. (34) is required. In the case of linear solver this task is relatively easy due to linearity of the fixed-point function. In the case of general fixed-point methods, directional derivatives must be evaluated.<sup>11</sup>

A generalization<sup>12</sup> of the RPM algorithm to  $n$ -th order RPM( $n$ ) algorithm is obtained if inner iterations on the  $q$ -part are performed. In that case the RPM( $n$ ) algorithm takes the following form:

$$\mathbf{p}^{(\nu+1)} = \mathbf{p}^{(\nu)} + \mathbf{Z} \left( \mathbf{I} - \mathbf{Z}^T \mathbf{R} \mathbf{Z} \right)^{-1} \mathbf{Z}^T \left( F(\mathbf{p}^{(\nu)} + \mathbf{q}_{n-1}^{(\nu)}) - \mathbf{p}^{(\nu)} \right) \quad (42)$$

$$\mathbf{q}_j^{(\nu+1)} = F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) - \mathbf{Z} \left( \mathbf{Z}^T F(\mathbf{p}^{(\nu)} + \mathbf{q}^{(\nu)}) \right), \quad j = 1, \dots, n \quad (43)$$

With this definition in mind, RPM algorithm described by Eq. (40) and Eq. (41) would correspond to RPM(0) algorithm.

#### IV.A. Determining the Unstable Basis

In order to perform the projections in Eq. (37), Eq. (38) and Eq. (39), basis  $\mathbf{Z}$  must be available. Furthermore, the cost of obtaining the basis of unstable space must be acceptable from the performance and computer memory point of view. The key to inexpensive determination of the unstable eigen-space is the recursive property of the iteration matrix  $\mathbf{R}$  defined in Eq. (24). If the difference between the current stable part of solution and the previous one is denoted by  $\Delta\mathbf{q}^{(\nu)}$ :

$$\Delta\mathbf{q}^{(\nu)} = \mathbf{q}^{(\nu+1)} - \mathbf{q}^{(\nu)}, \quad (44)$$

the recursive property of the iteration matrix is given by:

$$\Delta\mathbf{q}^{(\nu+1)} = \mathbf{R}^\nu \Delta\mathbf{q}^{(0)}. \quad (45)$$

In other words, the Krylov space  $\mathcal{K}$  is spanned by  $\mathbf{R}^\nu \Delta\mathbf{q}^{(0)}$ :

$$\mathcal{K} = \text{span}(\Delta\mathbf{q}^{(0)}, \mathbf{R}^1 \Delta\mathbf{q}^{(0)}, \mathbf{R}^2 \Delta\mathbf{q}^{(0)}, \dots, \mathbf{R}^\nu \Delta\mathbf{q}^{(0)}) \quad (46)$$

It is possible to extract the dominant eigen-space of from the Krylov space by computing the orthogonal basis from this space using a modified Gram-Schmidt procedure.<sup>13</sup> In practice, for RPM(0) algorithm only last two differences are used and the basis is recursively enlarged. In the case of RPM( $n$ ) algorithms,  $n - 1$   $q$ -difference vectors can be used. Modified Gram-Schmidt orthogonalization operates on the rectangular matrix:

$$\mathbf{B} = [\Delta\mathbf{q}^{(n)}, \Delta\mathbf{q}^{(n-1)}, \Delta\mathbf{q}^{(n-2)}, \dots] \quad (47)$$

that has the number of columns corresponding to the number of  $q$ -differences. The result of modified gram-Schmidt orthogonalization is the  $QR$ -factorization of the original rectangular matrix

$$\mathbf{B} = \hat{\mathbf{B}}\mathbf{T} \quad (48)$$

Here, matrix  $\mathbf{T}$  is upper triangular and matrix  $\hat{\mathbf{B}}$  is orthogonal. Diagonal entries of matrix  $\mathbf{T}$  are be used to estimate the dimension of the unstable space by examining their value. In particular, if  $T(1,1) \gg T(2,2)$ ,  $T(1,1) \gg T(3,3), \dots$  then dominant eigen-space is one-dimensional and the first column of the orthogonal matrix  $\hat{\mathbf{B}}\mathbf{T}$  is added to the basis  $\mathbf{Z}$ . Similarly, if this criterion is not satisfied, the process can be repeated with the next diagonal entry  $T(k,k)$  and the basis can be recursively enlarged. In practice, only last two  $q$ -differences are used because the basis is recursively enlarged, minimizing the storage requirements.

In practical implementations, basis of the unstable space should be enlarged only when criteria that signify divergence or stalled convergence are met. The natural quantity on which the criteria can be based is the residual defined as:

$$\mathbf{r}^{(\nu)} = \mathbf{b} - \mathbf{A}\mathbf{u}^{(\nu)} \quad (49)$$

This quantity is usually monitored in the  $l_2$  norm and any significant increase usually denotes state of divergence. Similarly, stalled divergence can be defined relative to the number of iterations used to reduce the value of the residual to a predetermined level. If either of these two criteria are met, the RPM( $n$ ) algorithm is activated in accordance to the algorithm described.

In order to illustrate the enlargement of the unstable space and RPM algorithm, consider the same non-M-matrix that was shown to diverge when the AMG solver was applied to it, Fig. (5). It was observed that a single eigenvalue lies outside of unit circle in the complex plane, Fig. (6), and therefore the dimension of the unstable space is 1. Convergence history and spectral content of the corresponding iteration matrix of RPM(0)-AMG method is shown in Fig. (7) and Fig. (8), respectively. As expected, residual increase of the AMG solver around iteration number 10 triggered RPM(0) algorithm which has detected the outlying eigenvalue and constructed the basis of the unstable space of dimension 1. This in turn stabilized the spectrum of the corresponding iteration matrix  $\mathbf{R}$ , allowing AMG to converge efficiently on  $q$ -part of the algorithm.

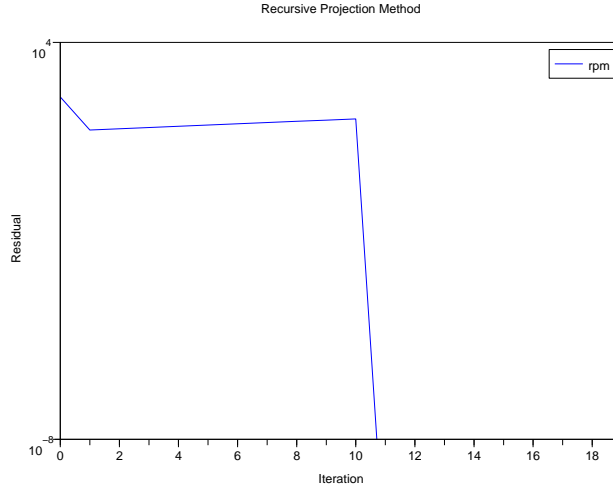


Figure 7. Two-level RPM(0)-AMG convergence history for non-M-matrix.

## V. Results

Performance of four solvers is examined on matrices representative of large CFD problems. Consider a LES simulation of an incompressible Newtonian fluid flow over a forward-facing step geometry at  $Re = 10\,000$  using a cell-centered Finite Volume solver. Fig. (9) shows a representative snapshot of instantaneous LES solution with an enstrophy. In order to capture the near-wall turbulence interaction, the mesh is aggressively graded towards the wall, with the ratio of largest to smallest cell volume of over 5 000. Bulk of simulation time is spent solving the pressure equation, where the matrix represents a discretized Laplace operator. Sufficient convergence of the pressure equation is required both to ensure local mass conservation and accuracy in time.

Eigen-spectrum of the pressure matrix is affected by the cell aspect ratio, making the system expensive to solve. Furthermore, the flow field contains a superposition of numerous resolved vortices of varying size and involved in turbulent interaction where the divergence-free condition plays an important role. In order to capture turbulent interaction with minimal distortion, the pressure equation needs to be solved to a tight tolerance.

Fig. (10) shows the performance of four solvers in terms of iteration count. The AMG solver uses the agglomerative coarsening strategy<sup>8</sup> in a V-cycle with 2 post-smoothing steps and coarsening ratio of 2. Coarsening is stopped when the coarse level matrix size drops below  $20 \times 20$ , at which point the system is solved to machine tolerance. One can readily see that the number of AMG cycles required to reach  $10^{-6}$  approaches 100, which implies high computational cost. The AMG solver when it is required to converge to this level of accuracy was not very good and marginally stable mode is observed once the residual has reached  $10^{-3}$  level. Furthermore, even without this mode, it is clear that the residual reduction rate is slowing down requiring many iterations to reach the level  $10^{-10}$ . In contrast, RPM(0)-AMG identifies slow convergence and builds an unstable space of dimension 18, converging to the required tolerance in 67 iterations. The convergence history of RPM(0)-AMG is non-linear and until all marginally stable modes are removed from the iteration matrix, very little progress is made. Once all of the undesirable modes are identified, solution appears to be stabilized and residual reduction rate is accelerated. Similarly, RPM(2)-AMG identifies unstable space once the residual has reached level  $10^{-3}$  and the dimension of unstable space is 6. However, convergence history appear to be smoother compared to RPM(0)-AMG once the unstable space is identified. Fundamental advantage of RPM(2)-AMG over RPM(0)-AMG is in smaller unstable space resulting in smaller number of operations and memory requirements. This is of great importance in large problems where memory constraints may be important. Finally, RPM(3)-AMG is clearly showing the smooth residual history and with 2 vectors in the unstable space. It is clearly the solver of choice in this case. Large dimension of unstable space for RPM(0)-AMG is related to the inaccurate representation of invariant space which in turn is triggered by heuristic criteria based on residual reduction rate. Contrary

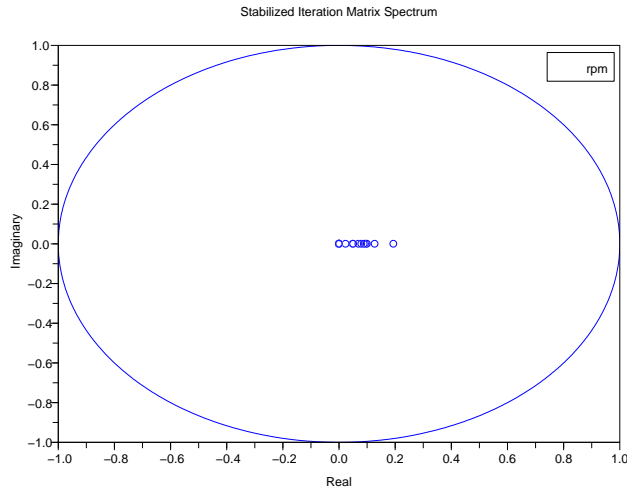


Figure 8. Spectrum of Two-level RPM(0)-AMG iteration matrix for non-M-matrix.

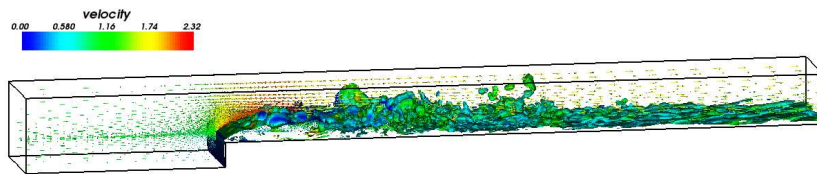


Figure 9. Enstrophy colored by turbulent kinetic energy with superimposed velocity vector field in the central plane.

to this, RPM(2)-AMG and RPM(3)-AMG allow the AMG solver to work more on stable part and delay recursive addition of new vectors to the unstable basis  $\mathbf{Z}$ .

## VI. Conclusion

Theoretical background of RPM( $n$ ) algorithm and its application to the stabilization of algebraic multi-grid method was presented. AMG acceleration and RPM( $n$ ) was initially demonstrated on a small problem that involves the discretization of Laplace equation where the properties of AMG preconditioning and RPM( $n$ ) could be stated in terms of iteration matrix spectrum. Acceleration properties of RPM( $n$ )-AMG are demonstrated on a segregated pressure matrix, where AMG performance was compared to RPM(0)-AMG, RPM(2)-AMG, and RPM(3)-AMG. Benefits of RPM( $n$ ) algorithm include increased convergence rates and removal of marginally stable modes that result in residual oscillations. Additional results of numerical experiments will be made available in the final form of the paper.

## References

- <sup>1</sup>Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
- <sup>2</sup>Chorin, A.J., *A Numerical Method for Solving Incompressible Viscous Flow Problems*, Journal of Computational Physics, vol. 2(1):12-26, 1967.
- <sup>3</sup>Saad, Y., *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2004.
- <sup>4</sup>Hackbush, W., *Multigrid Methods and Applications*, Springer, Berlin, 1985.
- <sup>5</sup>Briggs, W., Henson, V.E., McCormick, S., *A Multigrid Tutorial*, SIAM, Philadelphia, 2000.
- <sup>6</sup>Trottenberg, U., Oosterlee, C.W., Schuller, A. *Multigrid*, SIAM, Philadelphia, 2004.
- <sup>7</sup>Brandt, A., *Multi-Level Adaptive Solutions to Boundary-Value Problems*, Mathematics of Computation, vol. 31:333-390, 1977.

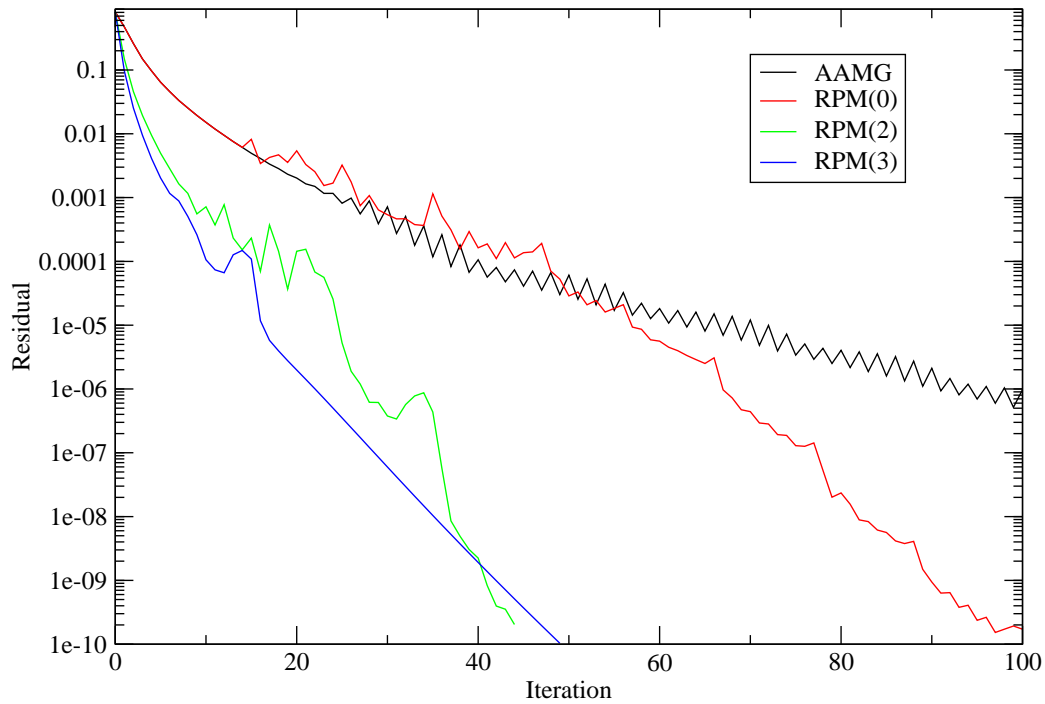


Figure 10. AMG and RPM convergence history for the pressure matrix in LES over a forward facing step.

<sup>8</sup>Hutchinson, B.R., Raithby, G.D., *A Multigrid Method Based on the Additive Correction Strategy*, Numerical Heat Transfer, 9:511-537, 1986.

<sup>9</sup>Ferziger, J.H, Perić, M., *Computational Methods for Fluid Dynamics*, Springer, Berlin, 1996.

<sup>10</sup>Axelsson, O., *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1996.

<sup>11</sup>Shroff, G.M., Keller, H.B., *Stabilization of Unstable Procedures: The Recursive Projection Method*, SIAM Journal on Numerical Analysis, vol. 30(4):1099-1120, 1993.

<sup>12</sup>Görtz, S., Möller, J., *Recursive Projection Method for Efficient Unsteady Flow Simulations*, European Congress on Computational Methods in Applied Sciences and Engineering, Neittaanmäki, P., Rossi, T., Majava, K., Pironneau, O., Jyväskylä, 25-28 July, 2004.

<sup>13</sup>Golub, H.G., Van Loan, C.F., *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.