

Analysis Tools for CFD Multigrid Solvers

Raymond E. Mineck* and James L. Thomas†
NASA Langley Research Center, Hampton, VA 23681-2199

Boris Diskin‡
National Institute of Aerospace, Hampton, VA 23666

Analysis tools are needed to guide the development and evaluate the performance of multigrid solvers for the fluid flow equations. Classical analysis tools, such as local mode analysis, often fail to accurately predict performance. Two-grid analysis tools, herein referred to as Idealized Coarse Grid and Idealized Relaxation iterations, have been developed and evaluated within a pilot multigrid solver. These new tools are applicable to general systems of equations and/or discretizations and point to problem areas within an existing multigrid solver. Idealized Relaxation and Idealized Coarse Grid are applied in developing textbook-efficient multigrid solvers for incompressible stagnation flow problems.

Nomenclature

a, c	=	arbitrary constants
f_u, f_v, f_p, f_ϕ	=	forcing functions
n	=	normal
p	=	pressure
q	=	solution vector
r_u, r_v, r_p	=	residuals in u, v, and p equations
R	=	residual
u, v	=	Cartesian velocity components
x, y	=	coordinates in physical domain
z	=	variable in complex coordinates
ϕ	=	unknown in Poisson equation
θ	=	leading edge angle (see figure 10)
ξ, η	=	coordinates in computational domain
μ	=	viscosity
ζ	=	variable in transformed complex coordinates

I. Introduction

FULLY elliptic equations can be solved using multigrid methods at a cost comparable to a few (<10) residual evaluations. This performance is termed Textbook Multigrid Efficiency (TME)¹⁻³; the descriptor refers to the efficiencies that are now widely available in textbooks for various model problems.⁴⁻⁷ A long-standing goal of computational fluid dynamics (CFD) is to obtain efficiencies in solving the Navier-Stokes equations that are comparable to those obtained in solving fully elliptic problems. The Navier-Stokes equations are a coupled set of non-linear equations of mixed type. Each obstacle on the way to TME for the Navier-Stokes equations needs to be identified, analyzed, and overcome. The process starts with simple problems and proceeds to increasingly more complex problems. Analysis methods are needed to understand and repair the cause of each deficiency. Classical analysis methods generally have simplifying assumptions (periodicity, constant coefficients, etc.) that limit their applicability in complex problems.

A pilot multigrid code has been developed to explore attainment of TME for fluid flow equations. The underlying framework is based on a full multigrid (FMG) algorithm using the full approximation scheme (FAS)

* Aerospace Engineer, Computational Modeling and Simulation Branch (CMSB)

† Head, CMSB, AIAA Fellow.

‡ Staff scientist, 144 Research Drive, Hampton, VA 23666

multigrid to converge each succeeding grid level.⁸ The FMG- k employs k FAS cycles on the current grid before proceeding to the next finer grid. The infrastructure required to implement the algorithm is dependent on the type of discretization scheme (unstructured, cell-centered, etc.); the current capability is limited to a finite difference discretization, with provisions for multiple blocks connecting point-wise across an interface. Within this discretization type, it is rather straightforward to implement different equation sets. The focus to date has been on a Pressure-Poisson formulation of the incompressible Navier-Stokes (INS) equations⁹⁻¹³; other model equations (elliptic, convection, convection-diffusion) have been implemented to investigate issues associated with separate component operators of these equations. Different boundary conditions are implemented in a modular fashion through a local mapping from global to local coordinates. Several relaxation techniques, including point and line Gauss-Seidel applied to one or more equations, are incorporated along with a direct solve of the linearized equations on coarse grids. The relaxation process is divided into two stages, a global relaxation and a local/boundary relaxation. The global relaxation is motivated by the observation that in many cases an efficient treatment of the interior discretization can be found. The local relaxations are intended to account for coupling between the equations that occur in special regions, such as near discontinuities and boundaries. For the particular problems investigated here, these regions typically are near block boundaries.

Previous studies with the Pressure-Poisson formulation showed fast convergence for inviscid flows including stagnation points,^{10,11,13} although the convergence was noticeably slower than that expected for TME solvers. The understanding of the differences is difficult because the flow solver does not generally provide indication of the cause of the degraded performance and the problems do not show up when the limitations of classical local mode analysis¹⁴⁻¹⁷ are satisfied. Two new analysis tools, termed Idealized Coarse Grid (ICG) and Idealized Relaxation (IR), were developed to analyze convergence of multigrid solvers. These tools are counterparts to those in classical analysis methods but are applicable to a larger class of problems. These tools have been incorporated into the pilot multigrid code and used to identify and overcome convergence difficulties. This paper defines the capabilities of the pilot multigrid code, including the new analysis tools, and demonstrates their use in application to stagnation flow.

II. Pilot Full Multigrid Flow Solver

The pilot full multigrid (FMG) flow solver is coded in FORTRAN 90 to take advantage of the recursive subroutine feature, dynamic memory allocation, and structured programming constructs. The design follows closely the implementation of Thor Gjesdal for a one-dimensional multigrid code.¹⁸ The program framework allows for different equation sets and discretizations. Infrastructure exists for the full three-dimensional equations; so far, however, the implementations are restricted to two dimensions. Currently, only a node-centered, finite-difference scheme for structured grids has been implemented. Multi-block computational grids are supported with three block-to-block interface types (periodic, C^0 , and C^∞). All three interface types require the grid points to match at the interface. The periodic and C^∞ interface types also require the derivative of the grid metrics to be continuous at the interface. Three equation sets are available: a scalar elliptic equation, scalar convection equation, and the Pressure-Poisson formulation of the incompressible Navier-Stokes equations. All the equation sets are implemented with non-zero forcing functions to enable the use of user-specified (i.e., manufactured) solutions to assist in the verification and validation process. Restriction to the next coarser mesh uses direct injection for the variables and full weighting for the residuals. Prolongation of the coarse grid correction from the coarse mesh to the fine mesh uses bilinear interpolation; prolongation of the coarse grid solution uses bicubic interpolation.

A. Equation sets

The three equation sets are described along with the available boundary conditions. The scalar elliptic equation is the Poisson equation defined as follows:

$$\Delta\phi \equiv \phi_{xx} + \phi_{yy} = f_\phi. \quad (1)$$

Edge conditions defined for the elliptic equation are: Dirichlet, Neumann, and the three block-to-block interface types.

The scalar nonlinear convection-diffusion equation is defined as follows:

$$Q(u) \equiv uu_x + vv_y - \mu\Delta u = f_u \quad (2)$$

Edge conditions defined for the convection-diffusion equations are: inflow, outflow, and periodic block-to-block interfaces. Linearized and constant coefficient versions of the convection-diffusion equation are also available. These capabilities are implemented as debugging tools.

The Pressure-Poisson formulation of the incompressible Navier-Stokes equations is defined as follows:

$$\begin{aligned} uu_x + vv_y - \mu\Delta u + p_x &= f_u \\ uv_x + vv_y - \mu\Delta v + p_y &= f_v \\ u_x^2 + 2u_yv_x + v_y^2 + \Delta p &= f_p \end{aligned} \quad (3)$$

These equations are referred herein as the u-momentum, v-momentum, and pressure equations. Edge conditions defined for the incompressible Navier-Stokes equations are: Dirichlet, inflow, outflow, solid surface, and the three block-to-block interface types. The viscous terms in the convection-diffusion operators are included in the implementation but are neglected for simplicity in this presentation.

B. Discretizations

Details regarding the discretization for each equation set are described below. Second-order, finite difference expressions in generalized coordinates are used whenever possible.

For the scalar elliptic equation, second-order central differences are used in the interior, corresponding to the usual 5-point Laplace operator on a rectangular grid. Boundary points may have edge conditions of Dirichlet, Neumann, or block-to-block interface type. For Dirichlet boundary conditions, ϕ is set to the user-specified value ($\phi = \phi_{\text{bdry}}$). For Neumann boundary conditions, there are two options. In the first option, referred to as a strong form, the discretized normal derivative is set to the user-specified value ($\phi_n = \phi_{n,\text{bdry}}$). The discretized derivative uses a three-point, one-sided, second-order accurate difference approximation. In the second option, referred to as a weak form, the Poisson equation at the boundary is discretized by a finite-volume scheme with the flux along the boundary evaluated using the user-specified normal derivative; the fluxes on the other faces are determined by central differences or by one-sided differences for non-orthogonal grids. On a uniform grid, this weak form is identical to solving the elliptic equation at the boundary with the virtual (ghost) point outside the domain eliminated through central differencing of the Neumann boundary condition.

Block-to-block edge interior and endpoints are treated differently. For interior points on periodic and C^∞ block-to-block interfaces, the discretization is the same as that used in the interior of the grid. For interior points on C^0 block-to-block interfaces, the Poisson equation is discretized by combining the fluxes across the faces of a control volume that straddles the grid point on the interface. The flux on each face is determined using central differences. For endpoints, there are two possibilities: all endpoints at the node in question are block-to-block interfaces or at least one endpoint is not. If all endpoints are block-to-block interfaces, the discretization is the same as that used in the interior of the block, otherwise, the discretization is determined by the different physical boundary conditions at the endpoint using a hierarchy of dominance for each boundary type.

For the scalar convection equation, three-point, upwind, second-order difference approximations are used for the derivatives at all interior points except those adjacent to an inflow boundary. Since there is only one upstream node for a grid point adjacent to an inflow boundary, either a second-order central difference or a first-order, upwind difference is used. Boundary points may have edge conditions of inflow, outflow, or periodic interface type. At inflow boundaries, the velocity is given as the user-specified value. At outflow boundaries or periodic interfaces, there are no boundary conditions required and the interior discretization can be applied.

For the Pressure-Poisson formulation of the incompressible Navier-Stokes equations, the interior discretization consists of three parts: (1) convective terms in the momentum equations differenced as for the scalar convection

equation; (2) the Laplacian in the pressure equation differenced as for the scalar elliptic equation; and (3) the remaining derivatives differenced with second-order, central differences. The boundary conditions for the Pressure-Poisson formulation are similar to those used for the primitive-equation formulation (momentum and continuity equations) with the additional condition that continuity is enforced at inflow. The continuity equation is not enforced explicitly in the interior, but the residual of its discrete central-difference approximation is monitored to ensure it converges to zero as the mesh is refined. Four types of boundary conditions are implemented: Dirichlet, inflow, outflow, and solid surface. A number of different implementations have been examined; typical examples are described below. At Dirichlet boundaries, all values are set to the user-prescribed values. At inflow boundaries, u and v are specified; continuity is enforced weakly through the Neumann condition for the normal pressure derivative evaluated from the momentum equations. In the weak form of this Neumann condition, the velocity derivative terms in the pressure equation are discretized as one-sided differences. At outflow boundaries, p is specified and the standard u-momentum and v-momentum equations are discretized with one-sided, second-order differences for the pressure gradient terms normal to the boundary. At an inviscid solid-surface boundary, flow tangency is imposed (zero normal velocity). The tangential and normal momentum equations are solved by centering the discretization at a location shifted half a mesh size into the domain from the body surface. Further details of these boundary conditions are available elsewhere.¹⁹

C. Relaxation

Relaxation consists of at most three phases: optional local relaxation over a part of the domain (typically near a boundary), then a global relaxation over the entire domain, and finally a second optional local relaxation. The pilot code contains many options to configure the relaxation phases. Relaxations are configured by specifying the groups of equations/variables to be solved during various phases. The appropriate matrices during each phase are assembled from equations of the full linearization; this approach to assembly is not optimal from a performance standpoint but does allow rapid assessment of different relaxation schemes. Within a given phase, the equations are solved for corrections $\delta(\delta q)$ with contributions from previous phases updated through δq .

$$\frac{\partial \tilde{R}}{\partial q}(\delta(\delta q)) = (f - R(q)) - \frac{\partial R}{\partial q}(\delta q) \quad (4)$$

where $(f-R(q))$ denotes the residual vector evaluated at the solution vector q and $\frac{\partial R}{\partial q}$ is the Jacobian. Normally, $\frac{\partial \tilde{R}}{\partial q} = \frac{\partial R}{\partial q}$ is the full linearization of the differential operator. However, when marching the momentum equations into the stagnation region, a modification to the full linearization has been used; this modification (referred to as a diagonally dominant form) is more stable for marching toward stagnation.¹⁹ This modification is used for the parabola computations that follow. At the start of relaxation, δq is initialized to zero; at the end of relaxation, q is updated ($q^{new} = q + \delta q$), and a new nonlinear residual is calculated. Optionally, nonlinear residuals and q can be updated after each relaxation phase. Four direct solvers are available: a sparse equation solver, a full-bandwidth Gaussian elimination, and block tridiagonal and block pentadiagonal solvers.

Several global relaxation schemes are available, including point and line Gauss Seidel relaxation. Each scheme allows for the relaxation of one or more equations/unknowns at each point (or line of points). For point Gauss Seidel relaxation, all eight possible sequences through a two-dimensional mesh are available. Point relaxation can also be done in a multi-color order in which several relaxation passes are performed, each pass relaxes only the points of one color. For instance, in a two-color scheme, commonly referred to as red/black or checkerboard, the first pass would relax the unknowns at every (red) node with an odd sum of grid indices; the second pass would relax the unknowns at every (black) node with an even sum of grid indices. For line Gauss Seidel relaxation, the unknowns on a line of grid points (a sequential group of points with a common grid index) are relaxed in a sequence on the structured mesh. All four possible sequences for a two-dimensional mesh are available, including multiple colored relaxations. The combination of a line and a block relaxation can account for a stronger coupling among the equations near the boundary than in the interior. This proved necessary for robustness in viscous simulations but was not important for the applications here.

Local or boundary relaxations are needed near boundaries when the boundary condition equations are different from the interior equations. Combinations of different discrete equations may easily result in a local coupling of the equations that can be solved only by simultaneous solution. Local relaxations are also useful in avoiding inaccurate residual transfers in the vicinity of the boundary during restriction. They may be implemented as multiple relaxation

passes throughout the local region or as the direct solution of the equations. Relaxation passes may be either point or line Gauss Seidel. Direct solution is used for all the local relaxations for the cases presented herein. The width of the local relaxation region typically encompasses all grid points associated with the discrete boundary condition equations as well as the grid points for the interior discretization that share data with the boundary condition equations.

D. Unified Implementation of Boundary Conditions

To unify the evaluation of the equations to be solved at a block boundary or edge, the data along and near that edge (grid coordinates, metrics, and unknowns) are rotated to a temporary computational coordinate system with the ξ coordinate aligned with the boundary tangential direction, the η coordinate oriented into the interior, and the boundary point mapped to the point at $\xi=1$ and $\eta=1$.²⁰ Both the periodic and the C^∞ block-to-block interface edge conditions map additional grid points, metrics, and unknowns from the connecting periodic or C^∞ edge to the points below the $\eta=1$ line. This provides a repeating edge condition for the periodic edge or a continuous mesh across blocks with a C^∞ edge. All residual evaluations and matrix coefficient assemblies use the same routine; common routines, coded once, perform the conversion to and from this temporary coordinate system.

For each equation set, companion modules are coded for each interior equation and each edge condition: one for the residual evaluation and the other for the implicit matrix assembly. Except for a C^0 block-to-block interface point or the corner point (end of a boundary), all information is available locally. For a C^0 block-to-block interface point, these operations are divided into two steps, one for each side of the interface. On each side the interface, the contributions are accumulated from the three interior faces of a control volume. The contribution from the edge along the interface is not calculated since its contribution is cancelled by the contribution from the abutting interface. The contributions from each side of the interface are summed to obtain the total. For a corner point, each boundary evaluation provides the contribution on the boundary face and the face normal to the boundary. The boundary contributions are summed to obtain the total.

III. Analysis Tools

The development of a multigrid flow solver involves the implementation of various functional capabilities. Debugging and verification of each capability as it is integrated can greatly simplify the debugging and verification of the final product. This pilot code incorporates all components required for an FMG code: residual evaluations, restriction to a coarser grid, relaxation or direct solution on a grid, and prolongation of the coarse grid correction to a finer grid. Tools are available to analyze the linearizations that are used in the direct solution and relaxation schemes as well as residual and discretization error reductions with grid refinement, on the entire grid or its subsets. Verification (or benchmark) problems have been prepared to support testing specific features of the code. The efficiency of various relaxation schemes can be checked by comparison with ICG iterations; ICG iterations are two-grid iterations, in which the coarse-grid correction is replaced with an idealized imitation that assuredly reduces smooth errors. Similarly, the performance of the coarse-grid correction can be checked by IR iterations that are two-grid iterations, in which parts of the relaxation are replaced with explicit error-averaging procedures. These built-in tools are supplemented by the typical stand-alone tools used to analyze the performance of a solver, such as local mode (Fourier) analysis.

A. Check of Solution Process

A library of verification problems has been included in the code to test various features. Separate libraries are available for each equation set. Each library is organized as a series of problems, each building on the verified features from the previous problems. Each verification problem is designed to test a specific feature of FMG development and/or implementation. Typical tests check the physical boundary conditions, multigrid process, block partitions, block-to-block interfaces, etc. Results from each test problem are saved in separate directories to serve as a reference for future tests. When changes are made in the code, each test in the library is repeated and the results compared to those in the reference directories.

A linearization check has been incorporated into the code to check for consistency of the linearizations used in the matrix assembly. The residual is evaluated for each equation at each point on the grid. Sequencing through the mesh, the value of each unknown is perturbed by a small amount and the residuals at all points are reevaluated. The Jacobian entries are computed numerically by divided differences and are compared to analytical values. Warnings are given when there is not a one-to-one correspondence in the occurrence of non-zero Jacobian entries. The maximum and L_2 -norm differences between the analytical and calculated Jacobians are listed to provide simple

indicators of consistency. An additional check for coarse grids is that the residuals converge quadratically when applying Newton's method.

B. Error evaluations

Three types of errors are considered: residual errors, discretization errors (differences between the exact differential and exact discrete solutions), and algebraic errors (differences between the approximate and exact discrete solutions). As mentioned previously, each equation set is implemented with a non-zero forcing function so that arbitrary manufactured solutions can be used to check the discretization error reduction with grid refinement. Provisions are also made to obtain an exact solution to the discrete equations for each FMG level; thus the algebraic error reduction can be checked. The exact discrete solution can be supplied from an analytical function or from files storing the fully converged solutions obtained on each FMG level.

Two discretization checks have been incorporated into the code: one utilizing the residuals and the other utilizing the discretization errors. In the FMG process, the residuals are calculated using the current solution. The code provides an option to sequence through each finest grid in the FMG process, initializing the solution on each grid from the exact solution to the differential equations. These norms (L_∞ , L_1 , and L_2) of the residual are calculated for each grid. The domain for the residual calculations can be the entire grid, the interior, or selected block edges. The ratio of the residual norm on one grid level to the residual norm on the next finer grid level is used to check that the implementation of the discretizations has the intended order reduction of the residuals with grid refinement. For a second order discretization with uniform halving of the mesh size, the factor should approach 4 as the grid is refined. The reduction factor can be different for those discrete equations where interior equations and boundary conditions are combined, as in the weak boundary condition implementation discussed earlier for the elliptic equation. Using the option to restrict the domain to the interior of the grid, the order of the residual reduction for the interior discretization can be checked. Similarly, restricting the domain to one edge allows the residual reduction on that edge to be checked. In the FMG process, the three norms of the discretization error are calculated from the converged solutions. The reduction factors for the discretization error between each succeeding pair of grids in the FMG process check consistency with the intended order of accuracy. In order to check boundary condition implementations, the equations along a boundary can be solved with the exact values in the interior; the discretization errors should then converge according to the intended order property of the discretization.

At the end of each FMG cycle, the residual error, the discretization error (if an exact solution to the differential equations is available), and algebraic error (if an exact solution to the discrete equations is available) are written to separate disk files for subsequent analysis. The L_2 -norms of the residuals before and during multigrid cycles on each block and on the entire domain are used to evaluate the spectral radius of the convergence. When the solution has converged on the current grid or when the specified number of cycles has been completed, the three norms of the residual and the discretization error are written to separate disk files along with the reduction factors from the previous grid.

C. Multigrid Solver Analysis

The efficiency of a multigrid cycle can be tested through analyzing two-grid cycles. A two-grid cycle includes four parts: relaxing the fine-grid equations, restriction of the residuals and current solution to the coarse grid, solution of the coarse-grid equations, and prolongation of the corrections to the fine grid. The current restriction and prolongation operators are considered suitable for efficient multigrid. Two analysis tools have been developed to diagnose problems with relaxation and coarse-grid correction. The analysis tools are applied to an available, imperfect solver dealing with a practical problem. The purpose of the analysis is to isolate, identify, and improve parts of the solver responsible for the less than optimal performance. Detection of inefficient parts of the solver is achieved by comparing the performance of the actual two-grid cycle with idealized (IR and ICG) iterations.

In IR iterations, the actual relaxation scheme is replaced with full-weighted averaging of algebraic errors; the coarse-grid correction scheme is the same as in the actual two-grid cycle. The smoothing rate of this idealized relaxation is 0.5, matching the rate of the point Gauss-Seidel relaxation for the Laplace equation. The idealized relaxation scheme does not depend on the discrete equations solved in the actual cycle. It guarantees reduction of the high-frequency components of the algebraic error function. If IR iterations converge much faster than the actual cycle, then the actual relaxation scheme is not a good smoother.

In ICG iterations, the actual relaxation scheme is used and the coarse grid correction is replaced with the following idealized imitation: first the full-weighting averages of the fine-grid algebraic errors are computed at the coarse-grid nodes; then the computed averages are prolonged to the fine grid by bilinear interpolation and subtracted from the initial algebraic errors. The idealized coarse-grid correction is also independent of the discretizations used on the fine and coarse grids. This procedure efficiently reduces smooth algebraic errors. If convergence of the ICG

iterations is much better than convergence of the actual cycle, then the actual coarse-grid correction is not sufficient to reduce smooth error components.

Idealized parts of IR and ICG iterations directly address the fine-grid algebraic errors. An exact discrete solution is needed to evaluate the algebraic errors. An exact solution can be obtained by selecting a manufactured solution to the discrete equations or by using a solver to provide a converged solution on the fine grid of the two-grid scheme.

IV. Applications

The new analysis tools have been applied to the multigrid solution schemes for two problems: flow against an infinite flat plate (plane stagnation) and flow about a parabolic arc (forward portion of an airfoil). The equation set chosen is the Pressure-Poisson formulation of the incompressible Navier-Stokes equations, Eq. (3). The equations are nonlinear with highly variable coefficients near the stagnation point so that conventional analysis techniques are of limited use. (See details of the limitations using local mode analysis in reference 19). The inviscid form of the equations is used in these applications to highlight the difficulties encountered in the full set of Navier-Stokes equations at high Reynolds number.¹³

A. Plane stagnation

Multigrid convergence is often degraded in the region near a stagnation point. Plane stagnation flow was selected for investigation because the plane-flow solution characterizes the behavior near the stagnation point in general geometries. The model problem involves flow directed towards an infinite flat plate at $x = 0$. Analysis of this problem is supported by the existence of an exact solution to the plane-stagnation partial differential equations, shown below:

$$\begin{aligned} u &= -x \\ v &= y \\ p &= -\frac{1}{2}(x^2 + y^2) + c \end{aligned} \tag{5}$$

where the constant c corresponds to the total pressure. Note that a second-order accurate discretization has no discretization errors for this plane-stagnation solution. We typically use second-order, one-sided, upwind discretizations for the convection terms in momentum equations, the second-order five-point discretization for Laplacian in the pressure equation, and second-order central differences for the pressure gradient in the momentum equations and the velocity derivatives in the pressure equation. Two stagnation flow regions, the regular stagnation region and the deep stagnation region, are shown in Figure 1.

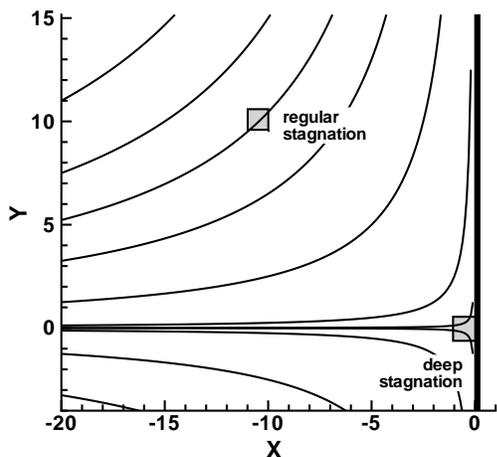


Figure 1. Plane stagnation flow streamlines and location of studied regions.

The deep stagnation region is a unit square with the center of the right face at the origin (stagnation point). The computational domain for the deep stagnation flow region is shown in Figure 2. The regular stagnation region is a unit square shifted far away from stagnation; the center of the right face is located at $(-10, 10)$.

The regular stagnation flow region is governed by nonlinear equations with relatively small variations across the domain. The flow for this case is approximately directed from the lower left corner to the upper right corner of the domain. Inflow conditions (u , v , and p_n specified) are enforced at the left and bottom boundaries and outflow conditions (p specified) are enforced on the top and right boundaries. The relaxation scheme utilizes a vertical-line, Gauss-Seidel relaxation of the pressure equation from left to right, downstream marching of the u - and v -momentum equations, and local relaxation of all three coupled equations on 3-nodes-wide bands along the inflow boundaries.

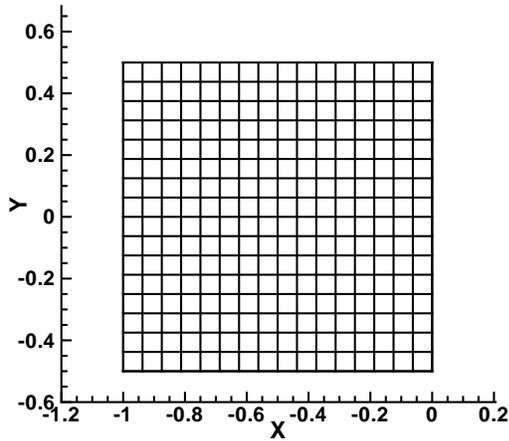


Figure 2. Computational mesh used for the deep stagnation flow region. Every 4th node shown.

IR and ICG analyses were applied to a two-grid cycle at 65x65 mesh. Four iterations were run: two-grid (actual) cycle, ICG iterations, IR(p) iterations where the explicit averaging of algebraic errors in pressure replaced the global relaxation of the pressure equation (followed by the downstream marching of the momentum equations and local relaxation), and IR(uvp) iterations where the entire relaxation is replaced with explicit error averaging for all three variables. The initial error distribution was random in all variables. Convergence rates of the pressure equation residual are shown in Fig. 4. The convergence rates of the two-grid actual cycle and the ICG iterations are nearly indistinguishable, indicating that the relaxation scheme provides adequate smoothing. Convergence rates of IR(p) iterations are comparable to the rates of the actual cycle. The convergence of the actual cycle is better than IR(p) convergence because the actual cycle employs a line Gauss-Seidel smoothing that is somewhat more efficient than the idealized relaxation. The convergence rates of IR(uvp) iterations are significantly worse indicating that coarse-grid correction cannot reduce some smooth error components; in actual relaxation these components are reduced in downstream marching of the momentum equations. Subsequent analyses, not shown here, indicated that on finer grids, the convergence rates of the IR(uvp) increases, approaching the limit of 0.75 per cycle. This limit is well-known in the multigrid theory. This limit bounds convergence of standard multigrid solvers for second-order discretizations of convection-dominated flow equations; the standard multigrid solvers rely on coarse grid correction for all smooth components. The convergence rates of the actual cycle, the ICG, and the IR(p) iterations are grid independent.

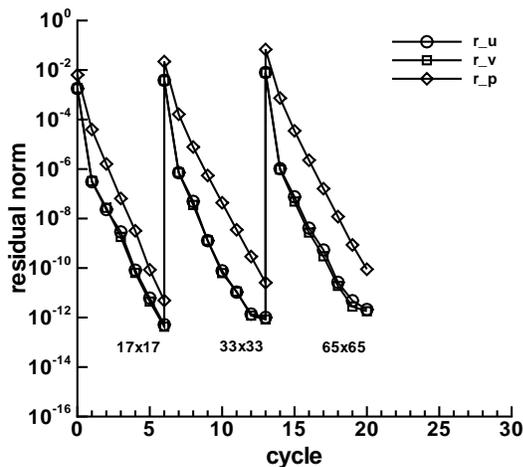


Figure 3. Residual convergence of the FMG solver for regular stagnation flow.

The equations were solved for the regular stagnation region with the pilot code using a FV(1,2) cycle with a 5x5 coarsest mesh for each FMG level. The convergence of the L_2 -norm of the residuals on a series of finer grids is shown in Fig. 3. As expected, the residuals for the momentum equations are significantly less than those for the pressure equation because they are reduced by marching after the pressure relaxation; the residual convergence of the system of equations is always dominated by the convergence of the pressure equation. The results show that the convergence of the pressure equation is about 0.08 per cycle and the asymptotic rate is grid independent. As a point of reference, the local mode analysis for the five-point Laplace operator predicts a convergence rate of 0.085 (0.44^3) per cycle for a two-grid cycle with three line-relaxation sweeps on the fine grid. The convergence rate for the residuals of the pressure equation is in very good agreement with this value.

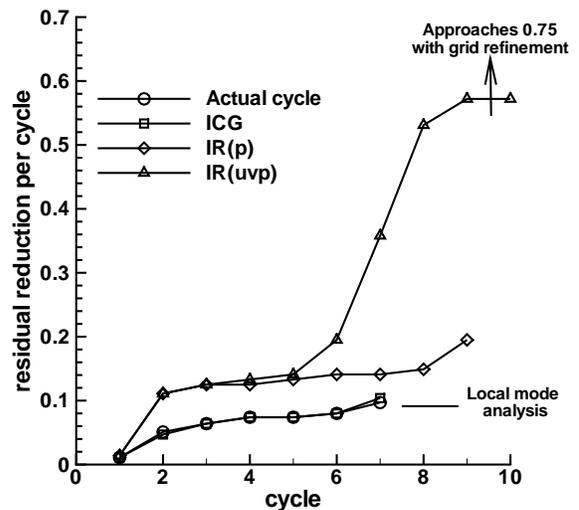


Figure 4. Two-grid analysis (fine-grid 65x65) for the regular stagnation region for the pressure equation.

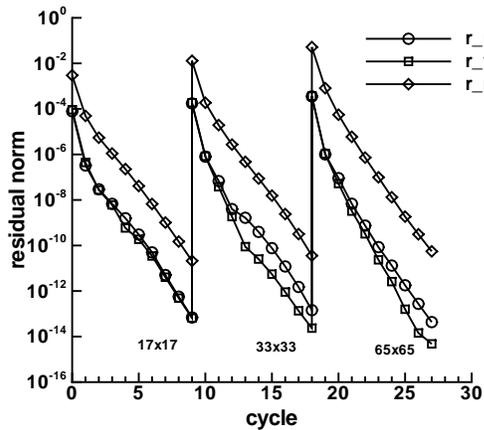


Figure 5. Residual convergence of the FMG solver for the deep stagnation region.

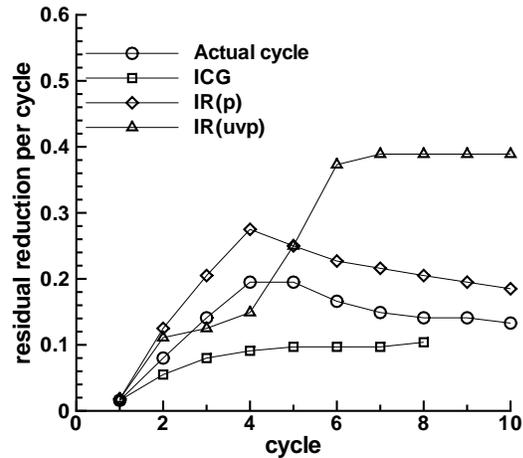


Figure 6. Two-grid analysis (fine grid 65x65) for the deep stagnation region.

The deep stagnation region is governed by equations with nonlinear coefficients that vary substantially across the domain. Inflow conditions (u , v , and p_x specified) are enforced at the left boundary, outflow conditions (p specified) are enforced on the top and bottom boundaries, and tangency conditions ($u=0$) are specified on the right boundary. A uniform mesh with 65×65 nodes was selected for the finest grid in the FMG process. The relaxation scheme is similar to the relaxation scheme for the regular stagnation flow with the local relaxations applied to the inflow (left) and tangency (right) boundaries. The equations were solved for the deep stagnation region with a FV(1,2) cycle that proceeded down to a 5×5 coarsest mesh for each FMG level. The convergence of the L_2 -norm of the residuals on a series of FMG levels is shown in Figure 5. The results show that the convergence of the pressure equation is about 0.14 per cycle and the asymptotic rate is grid independent.

The results of IR and ICG analyses are shown in Figure 6. The iterations were performed on a 65×65 fine mesh. The same four cases described above were run, again starting with a random distribution of errors in all variables. The convergence rate for the actual multigrid solver degrades to a maximum value of about 0.21 per cycle with an asymptotic value of about 0.14 per cycle. The asymptotic convergence rate is only slightly slower than that of the ICG iterations, indicating that the relaxation scheme provides adequate smoothing. The convergence rates for the IR(uvp) iterations are again much slower than for other iterations because the marching element of the actual solver is neglected. The convergence rate for the IR(p) iterations degrades to a maximum value of about 0.29 per cycle before improving to an asymptotic rate 0.18 per cycle. IR and ICG analyses performed on finer grids (results are not shown here) confirms that the convergence rates of all iterations are grid independent with asymptotic rates for IR(uvp) iterations approaching 0.75 and the actual cycle providing error reduction of 0.1 to 0.2 per iteration.

B. Parabolic arc stagnation

The plane-stagnation flow represents a flow in a simple geometry; a uniform Cartesian grid was used. To demonstrate applications of the IR and ICG analyses on more realistic, body-fitted grids, the stagnation flow over a parabolic arc was selected. A parabolic arc was chosen to represent the forward portion of an airfoil. A uniform mesh in transformed complex coordinates, $\zeta = \xi + i \eta$, is transformed into a parabolic mesh in complex coordinates, $z = x + i y$, by the conformal mapping $\zeta = z^2$. The physical grid is determined from a square grid in the transformed complex plane, extending from $\eta = \eta_0$ to $\eta = \eta_1$ and centered about $\xi = 0$. The solid surface (parabolic arc) corresponds to the line $\eta_0 = \text{constant}$. Like the flat plate stagnation flow case, an exact solution to the differential equations exists.

The exact solution, given in terms of (ξ, η) is:

$$\begin{aligned} u &= 1 - \frac{\eta_0 \eta}{\xi^2 + \eta^2} \\ v &= \frac{\eta_0 \xi}{\xi^2 + \eta^2} \\ p &= p_\infty + \frac{1}{2}(1 - u^2 - v^2) \end{aligned} \quad (5)$$

where: $x = \xi^2 + \eta^2$ and $y = 2\xi\eta$.

The solid surface, $\eta_0 = 1/\sqrt{2}$, is the parabola with a leading edge radius of 0.5 centered at the origin. A body-fitted mesh with 129×129 nodes and $\eta_1 = 15\eta_0$ was selected for the finest grid in the FMG process. The computational domain for the parabolic arc region is shown in Fig. 7. Inflow conditions (u , v , and p_n specified) are enforced at the left boundary, outflow conditions (p specified) are enforced on the upper and lower boundaries, and tangency conditions ($\vec{v} \cdot \vec{n} = 0$) are specified on the solid surface.

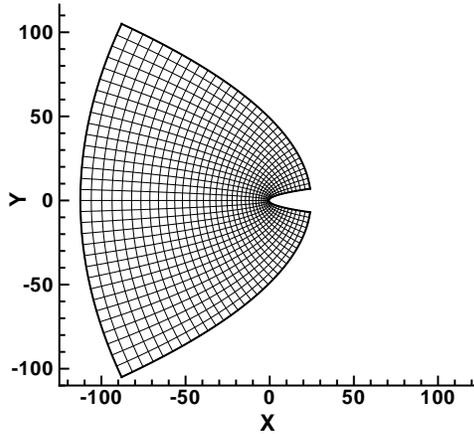


Figure 7. Computational mesh for the parabolic arc region. Every 4th node shown.

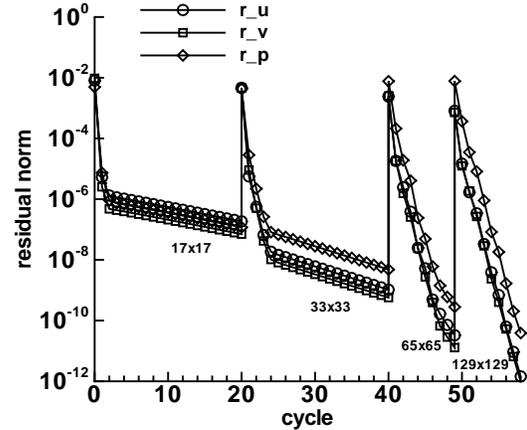


Figure 8. Residual convergence rate of the FMG solver for the parabolic arc region.

The relaxation scheme consists of a global relaxation pass with point Gauss-Seidel relaxation of the pressure equation (the relaxation proceeds from the upper outflow boundary to the lower outflow boundary, starting at the inflow edge and ending at the solid surface), marching using the diagonally-dominant modifications of the u - and v -momentum equations (collective line Gauss-Seidel relaxation proceeds from the inflow boundary to the solid surface), and a local relaxation for the three equations on a 3-nodes-wide band along the inflow boundary and a 4-nodes-wide band along the tangency boundary. The equations were solved with the pilot FMG code using a FV(2,1) cycle that proceeded down to a 9×9 coarsest mesh for each FMG level. The convergence of the L_2 -norm of the residuals on a series of FMG grids is shown in Figure 8. The initial convergence on the two coarser grids is good but degrades rapidly to about 0.88 per cycle after the initial high frequency errors are smoothed. On the two finer grids, convergence rates are about 0.12 per cycle. The IR and ICG analyses are applied at a mesh with poor convergence (17×17) and at a mesh with good convergence (129×129).

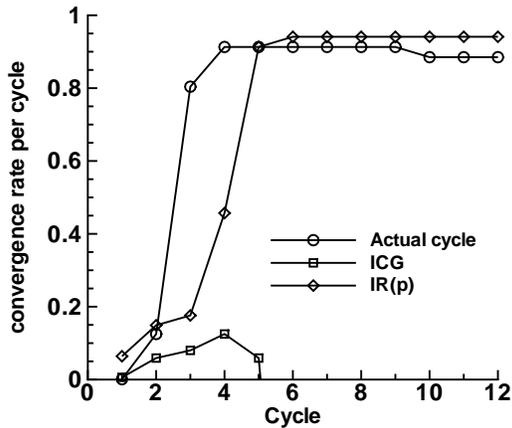


Figure 9. Two-grid analysis (fine grid 17x17) for the parabolic arc for the pressure equation.

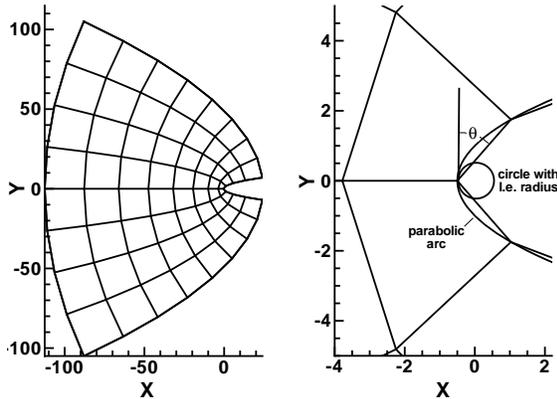


Figure 10. Details of the coarsest grid (9x9) used for the parabolic arc.

The IR and ICG analyses (actual cycle, ICG iterations, and IR(p) iterations) were performed on the 17x17 grid and the results are shown in figure 9. The asymptotic convergence rate of the actual two-grid cycle for the pressure equation residuals is about 0.88 per cycle, the same case as the 17x17 fine mesh shown in figure 8. The asymptotic convergence rate for the IR(p) is 0.94 per cycle, slightly poorer than the actual cycle rate. However, the ICG convergence rate is typically less than 0.10 per cycle, solving the equations in 5 iterations. The slow convergence of IR(p) iterations and fast convergence of ICG iterations indicate that the corrections prolonged from the 9x9 grid are poor, but the relaxation scheme provides adequate smoothing. Examination of the 9x9 mesh, shown in Figure 10, suggests the cause of the poor coarse-grid correction. Whereas the 9x9 mesh appears to represent reasonably the whole computational domain, there is insufficient resolution of the leading edge region. To obtain reasonable discretization accuracy in the vicinity of the leading edge, one would expect to have at least a few nodes defining the leading 90° arc sector. With the studied 9x9 grid, the solid surface effectively introduces a very sharp leading edge, with an angle, $\theta=41^\circ$, defining θ as the angle between the tangent to parabola at the leading edge and the effective grid line; we will refer to this angle as leading edge angle. As the grid is refined, leading edge angle

becomes smaller and the leading edge bluntness is better resolved resulting in a more accurate discrete approximation. The accuracy of the discrete solution on the 9x9 mesh is so poor that the coarse-grid correction provided to the finer grids is irrelevant. One possible solution to this problem is to use stretching to cluster additional points near the leading edge.

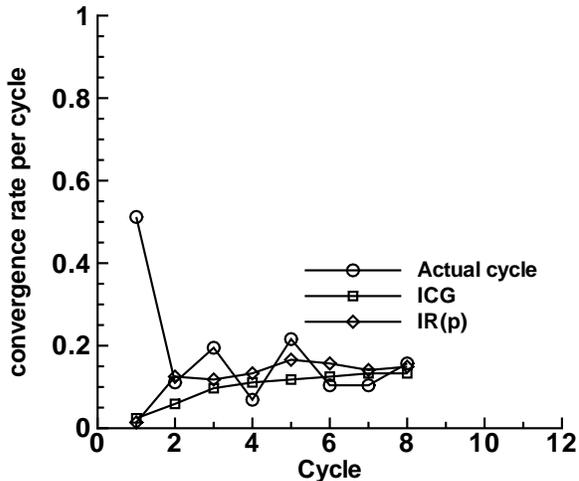


Figure 11. Two-grid analysis (fine grid 129x129) for the parabolic arc for the pressure equation.

The IR and ICG analyses (actual cycle, ICG iterations, and IR(p) iterations) were also performed on the 129x129 grid and the results are presented in figure 11. The actual two-grid cycle shows good convergence rates, with an average of about 0.15 per cycle. The convergence rates for the Idealized Relaxation and Idealized Coarse Grid are similar to those for the actual two-grid cycle. Thus, on this fine mesh (129x129), both the relaxation scheme and the coarse grid correction are adequate.

The IR and ICG analyses (actual cycle, ICG iterations, and IR(p) iterations) were also performed on the 129x129 grid and the results are presented in figure 11. The actual two-grid cycle shows good convergence rates, with an average of about 0.15 per cycle. The convergence rates for the Idealized Relaxation and Idealized Coarse Grid are similar to those for the actual two-grid cycle. Thus, on this fine mesh (129x129), both the relaxation scheme and the coarse grid correction are adequate.

C. Efficiency Evaluation

Stagnation flow against parabolic arcs was selected to assess the efficiency of the developed multigrid solvers. Domains of three sizes are considered: a large domain ($\eta_1=15\eta_0$), a reduced domain ($\eta_1=8\eta_0$), and a leading-edge domain ($\eta_1=1.25\eta_0$). Sketches of these domains are shown in Figure 12. FMG-1 and FMG-2 solvers with three levels were tested on each domain

The FMG-1 and the FMG-2 solvers were evaluated on the large domain covered by a 129×129 parabolic mesh. According to the analysis results, the coarsest mesh was chosen as 33×33 with the leading edge angle $\theta \approx 12^\circ$, a

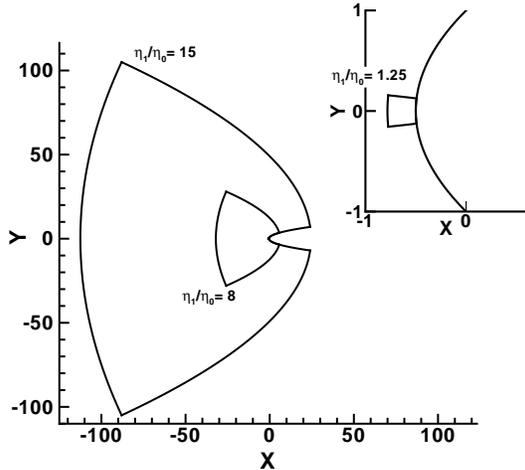


Figure 12. Computational domains for stagnation flows against parabolic arcs.

reasonable representation of the blunt leading edge. The L_2 -norm of the discretization error, the L_2 -norm of the algebraic error, and their ratio are listed in Table 1. The FMG-1 solver was able to provide solutions with the algebraic errors below the level of the discretization errors. The algebraic errors of FMG-2 solutions are about 10% of the discretization errors. Results (not presented herein) for the four-level FMG-1 cycle (with the coarsest grid 17×17) showed some degradation in performance with algebraic errors slightly above discretization errors. This reinforces the conclusion that convergence deteriorates when too coarse (incapable of providing a reasonable approximation accuracy) grids are involved.

The FMG-1 and the FMG-2 solvers were evaluated on the reduced domain with a 65×65 mesh. The size of the reduced domain and the finest mesh were chosen to provide the same leading edge resolution (identical leading edge angle θ) as in the experiments on the large domain. Results, presented in Table 2, are very

similar to the results obtained for the large domain, documented in Table 1. Algebraic errors were reduced below discretization errors by the FMG-1 solver. The FMG-2 solver was able to reduce the algebraic errors to the 10% level of the discretization errors.

Using a finer 129×129 mesh on the reduced domain while maintaining three multigrid levels improves the leading edge resolution on each mesh. The leading edge angles ranged from about 2° to 6° . Results from the FMG-1 and FMG-2 solvers on the reduced domain with the refined 129×129 mesh are presented in Table 3; they are similar to the results obtained on 65×65 mesh. This similarity suggests that the coarsest meshes used on the large domain (33×33) and on the reduced domain (17×17) provide adequate resolution for fast convergence.

Finally, the FMG-1 and the FMG-2 solvers were evaluated on the leading edge domain with a 33×33 mesh. The resulting mesh spacing is finer than those used on the large and reduced domains, with the leading edge angles all less than 1° . Results, presented in Table 4, demonstrate excellent performance with the FMG-1 cycle reducing the algebraic error to about 10% of the discretization error and the FMG-2 cycle reducing the algebraic error by another order of magnitude. The total computational cost of the FMG-1 solver is less than cost of ten residual evaluations; thus, TME has been attained in all tests.

Table 1 . L2-norms of the errors for the parabolic arc on the large domain ($\eta_1/\eta_0=15$), finest mesh 129x129, $\theta=3.1^\circ$.

Variable	Disc. Error	FMG-1		FMG-2	
		Algeb. Error	Algeb. Error/ Disc. Error	Algeb. Error	Algeb. Error/ Disc. Error
u	0.00083	0.00060	0.7	0.00011	0.1
v	0.00106	0.00067	0.6	0.00012	0.1
p	0.00069	0.00059	0.9	0.00009	0.1

Table 2. L2-norms of the errors for the parabolic arc on the reduced domain ($\eta_1/\eta_0=8$), finest mesh 65x65, $\theta=3.1^\circ$.

Variable	Disc. Error	FMG-1		FMG-2	
		Algeb. Error	Algeb. Error/ Disc. Error	Algeb. Error	Algeb. Error/ Disc. Error
u	0.00133	0.00045	0.3	0.00010	0.1
v	0.00182	0.00047	0.3	0.00010	0.1
p	0.00110	0.00043	0.4	0.00009	0.1

Table 3. L2-norms of the errors for the parabolic arc on the reduced domain, ($\eta_1/\eta_0=8$), finest mesh 129x129, $\theta=1.6^\circ$.

Variable	Disc. Error	FMG-1		FMG-2	
		Algeb. Error	Algeb. Error/ Disc. Error	Algeb. Error	Algeb. Error/ Disc. Error
u	0.00034	0.00016	0.5	0.00005	0.1
v	0.00047	0.00015	0.3	0.00005	0.1
p	0.00028	0.00014	0.5	0.00004	0.1

Table 4. L2-norms of the errors for the parabolic arc on the leading edge domain, ($\eta_1/\eta_0=1.25$), finest mesh 33x33, $\theta=0.1^\circ$.

Variable	Disc. Error	FMG-1		FMG-2	
		Algeb. Error	Algeb. Error/ Disc. Error	Algeb. Error	Algeb. Error/ Disc. Error
u	0.0000322	0.0000022	0.1	0.0000001	0.003
v	0.0000185	0.0000024	0.1	0.0000001	0.004
p	0.0000027	0.0000004	0.1	0.0000000	0.005

V. Conclusion

General analysis tools are needed to evaluate the performance of solvers of the fluid flow equations. Novel analysis tools, Idealized Coarse Grid (ICG) and Idealized Relaxation (IR), have been developed and built into a pilot multigrid solver. These tools were applied to two stagnation flow problems. The new analysis tools were able to analyze convergence in deep stagnation regions that were inaccessible through classical analysis. They also successfully identified coarse-grid correction as the reason for poor convergence of multigrid solvers for parabolic stagnation flows on very coarse meshes. The factor limiting the coarse-grid correction and degrading performance proved to be an underresolution of the leading edge. The analysis tools were instrumental in development of TME solvers for stagnation flow problems.

References

- ¹Brandt, A., "Recent developments in multigrid efficiency in computational fluid dynamics," Appendix C in *Multigrid*, Trottenberg, U, Oosterlee, C. W., and Schuler, A., eds. Academic Press, London, 2000, pp. 573-589.
- ²Brandt, A., Diskin, B., and Thomas, J. L., "Textbook multigrid efficiency for computational fluid dynamics simulations," AIAA Paper 2001-2570, *15th AIAA CFD Conference*, Anaheim, CA, June 2001.

- ³Thomas, J. L., Diskin, B., and Brandt, A., "Textbook multigrid efficiency for fluid simulations." *Ann. Rev. Fluid Mech.*, 35 (2003), pp. 317-340. An extended version has been published as ICASE Report 2002-16.
- ⁴Brandt, A., "Multigrid techniques: 1984 guide with applications to fluid dynamics," in *Lecture Notes for the Computational Fluid Dynamics*, Lecture Series at the Von-Karman Institute for Fluid Dynamics, The Weizmann Institute of Science, Rehovot, Israel, 1984. ISBN-3-88457-081-1, GMD-Studien Nr. 85, Available from GMD-AIW, Postfach 1316, D-53731, St. Augustin 1, Germany. Also available from Secretary, Department of Mathematics, University of Colorado at Denver, CO 80204-5300.
- ⁵Wesseling, P., *An introduction to multigrid methods*, Pure and Applied Mathematics, John Wiley & Sons, Chichester, 1992.
- ⁶Trottenburg, U., Oosterlee, C. W. and Schuler, A., *Multigrid*, Academic Press, London, 2000.
- ⁷Briggs, W. L., McCormick, S. F., and Henson, V. E., *Multigrid Tutorial*, 2nd edition, SIAM, USA, 2000.
- ⁸Brandt, A., "Multi-level adaptive solutions to boundary-value problems.", *Math. Comp.* 31, 1977, pp. 333-390.
- ⁹Roberts, T. W., Sidilkover, D., and Swanson, R. C., "Textbook multigrid efficiency for the steady Euler equations," AIAA Paper 97-1949, *13th AIAA CFD meeting*, Snowmass Village, CO, June-July 1997.
- ¹⁰Roberts, T. W., and Swanson, R. C., "Extending ideally converging multigrid methods to airfoil flows," AIAA Paper 99-3337, *14th AIAA CFD meeting*, Norfolk, VA, June-July 1999.
- ¹¹Roberts, T.R., Swanson, R. C., and Sidilkover, D., "An algorithm for ideal multigrid convergence for the steady Euler equations," *Computers and Fluids*, 28 (1999), pp. 427-442.
- ¹²Sidilkover, D., and Asher, U., "A multigrid solver for the steady-state Navier-Stokes equations using the pressure-Poisson formulation," *Matematica Aplicada e Computacional*, 14 (1995), pp. 21-35.
- ¹³Swanson, R. C., "Toward Optimal Multigrid Efficiency for the Navier-Stokes Equations." AIAA-2001-2574, 15th AIAA CFD Conference, Anaheim, CA. June 2001.
- ¹⁴Brandt, A., "Rigorous local mode analysis of multigrid," in *Proc. 4th Copper Mountain Conf. on Multigrid Methods*, Copper Mountain, CO, April 1989.
- ¹⁵Brandt, A., "Rigorous quantitative analysis of multigrid, I: Constant coefficients two-level cycle with L_2 -norm," *SIAM J. Num. Anal.*, 31 (1994), pp. 1695-1730.
- ¹⁶D'sid'eri, J. A., and Hemker, P. W., "Convergence analysis of the defect-correction iteration for hyperbolic problems," *SIAM J. Sci. Comp.*, 16 (1995), pp. 88-118.
- ¹⁷Diskin, B., and Thomas, J. L., "Half-space analysis of the defect-correction method for Fromm discretization of convection," *SIAM J. Sci. Comp.*, 22 (2000), pp. 633-655.
- ¹⁸Thor Gjesdal: "Programming Multigrid in Fortran 90." Technical Report CMR-93-A30007, Christian Michelsen Research AS, Bergen, Norway. 2003. (Contributed November 19, 1993 to MGNET <http://www.mgnet.org>).
- ¹⁹Diskin, B., Thomas, J. L., and Mineck, R. E., "Textbook multigrid efficiency for leading edge stagnation." NASA TM-2004-213037, 2004.
- ²⁰White, J. A. and Morrison, J. H., "A Pseudo-Temporal Multi-Grid Relaxation Scheme for Solving the Parabolized Navier-Stokes Equations," AIAA Paper No. 99-3360, June 1999.