

New Features of MIGAL solver

Michel Ferry

MFRDC
6 rue de la Perche
44700 Orvault France
<http://www.mfrdc.com>

Email : michel.ferry@mfrdc.com

September 2002

Abstract

Since its first release in 2000 MIGAL for PHOENICS has been improved for robustness. In particular, a new GMRES capability, new internal self-adjustments and new default parameters have been introduced to facilitate and extend users' control. To illustrate the new paradigm of this strongly coupled multi-grid algebraic solver and the large benefits that one can gain in using it, this paper firstly addresses the basics of the method. The velocity-pressure coupling, the multi-grid procedure, the ILU smoother as well as the momentum preconditioning technique and the GMRES accelerator are investigated. A second part is dedicated to the solver performance and several laminar, turbulent, buoyancy-driven and industrial flows are discussed regarding the global non-linear convergence efficiency of each case.

Introduction

When computational fluid dynamics first engaged the attention of engineers, during the 1960s, only algorithms with small demands on computing power could survive. Computer memory was then extremely rare and expensive. In this context, use of the Patankar-Spalding SIMPLE algorithm and its descendants (SIMPLER, SIMPLEC, SIMPLEST, PISO...) was the best strategy to adopt. But it is now well-known that the drawback of these methods, based on the segregation of momentum and continuity equations, is a slowing-down of convergence when the number of cells increases. It is also experienced that improving the efficiency of the sole pressure solver brings few benefits for the overall procedure and that the weakness of the methods lies in the velocity-pressure coupling [8]. The first attempt to solve simultaneously the momentum and continuity equations was the so-called 'SIVA' algorithm [3] (1972). Latter different way have been explored that either kept the continuity equation in its primitive form [15] or used an approximate Poisson equation [1][4][5][13]. But, requiring more memory to save the coefficients and solve the linear systems, these methods had to wait the end of the 1990s for their first industrial applications [4][11]. In the following, we will develop the basics of the MIGAL solver i.e. the way it treats the velocity-pressure coupling, its multi-grid procedure, its preconditioning techniques, and some new features implemented since its first attachment to PHOENICS [4].

Velocity-Pressure coupling

When applied to the hydrodynamic variables, MIGAL is a whole-field linear solver that updates simultaneously the velocity and pressure fields in the entire domain once per sweep. To understand the implications of such a technique on the global iteration process, let's first review the SIMPLEST algorithm as implemented in PHOENICS.

SIMPLEST

The acronym SIMPLEST stands for Semi-Implicit Method for Pressure-Linked Equations Shortened and is certainly the most efficient derivative of the SIMPLE method introduced by Patankar and Spalding [10]. Considering the momentum equations, and regardless of the way they are discretised, SIMPLEST solves them in their defect correction form:

$$a_p^\phi \delta\phi_p = \sum a_{nb}^\phi \delta\phi_{nb} + r^\phi \quad (1)$$

where $\delta\phi$ represents the velocity components corrections and r the residuals of the momentum equations. In that way, since the corrections tend to zero as convergence is approached, the round-off errors are limited and the coefficients a_p and a_{nb} may be only approximate. This defect correction approach is a key of the SIMPLEST success since it allows numerical

designers to adapt (1) to the iterative algorithm and to the nature of the problem. For example SIMPLEST differs from SIMPLE in that only diffusive contribution are kept in the updating coefficients a_p and a_{nb} .

Concerning the pressure variable, no direct equation is available and SIMPLEST, like most of the segregating approaches, builds an equation for pressure from the continuity constraint.

$$\frac{\partial}{\partial x}(\rho\delta u) + \frac{\partial}{\partial y}(\rho\delta v) = r^p \quad (2)$$

For this, the defect correction form of the continuity equation (2) is also retained and the coupling between the pressure gradients and the velocity components is approximated by:

$$\begin{cases} \delta u = d^u \frac{\partial}{\partial x}(\delta p) \\ \delta v = d^v \frac{\partial}{\partial y}(\delta p) \end{cases} \quad (3)$$

where the coefficients d^u and d^v are the so-called DVELDP. Then substituting (3) into (2) eliminates the velocity corrections and leads to a Poisson-like pressure correction equation whose generic form is also:

$$a_p^p \delta p_p = \sum a_{nb}^p \delta p_{nb} + r^p \quad (4)$$

Finally the SIMPLEST algorithm consists in solving sequentially the momentum equations (1) and the pressure equation (4) as follows:

1. **Start:** initialize velocity and pressure fields
2. **Segregated solution:**
 - Compute first momentum coefficients and solve (1) for δu
 - Compute second momentum coefficients and solve (1) for δv
 - Compute pressure coefficients and solve (4) for δp
3. **Update:** update pressure and velocity fields from correction solutions
4. **Iterate:** if not converged go to next sweep step 2.

Figure 1: SIMPLEST segregated algorithm

MIGAL

MIGAL differs from SIMPLEST in that the velocity and pressure corrections are computed simultaneously for each control volume. During a sweep, the momentum equations are discretised but not solved and their coefficients are accumulated until the continuity equation is discretised. Then only, MIGAL is called and solves the linear system made of all the equations of all cells.

1. **Start:** initialize velocity and pressure fields
2. **Coupled solution:**
 - Compute momentum and continuity coefficients
 - Solve (5) for $(\delta u, \delta v, \delta p)$
3. **Update:** update pressure and velocity fields from correction solutions
4. **Iterate:** if not converged go to next sweep step 2.

Figure 2: MIGAL coupled algorithm

The algebraic coupled system solved in step 2 may be organized in different ways depending on the ordering of unknowns. In order to use a block extension of the existing solvers, the momentum and continuity equations are lexicographically gathered to fit the usual generic form (5).

$$A_P \phi_P = \sum A_{nb} \phi_{nb} + b \quad (5)$$

where, unlike (1) and (4), the coefficients A , the variables ϕ and sources terms b are not single numbers but small matrixes and vectors. For example for a two-dimensional co-locative formulation:

$$A_{nb} = \begin{pmatrix} x_{nb}^u & x_{nb}^v & x_{nb}^p \\ y_{nb}^u & y_{nb}^v & y_{nb}^p \\ c_{nb}^u & c_{nb}^v & c_{nb}^p \end{pmatrix} \quad \phi_{nb} = \begin{pmatrix} \delta u_{nb} \\ \delta v_{nb} \\ \delta p_{nb} \end{pmatrix} \quad b_{nb} = \begin{pmatrix} r_{nb}^u \\ r_{nb}^v \\ r_{nb}^p \end{pmatrix} \quad (6)$$

and where the X , Y and C notations stand for x-momentum, y-momentum and continuity and the subscript nb stands also for the central coefficient. The coefficients of the first lines are straightforwardly derived from linearization of the momentum equations while the pressure coefficients of the last line are obtained from the continuity equation following a technique similar to (3). Beside simplicity, the advantage of this formulation is that the right-hand-side of

equation (5) represents the strict residuals provided by PHOENICS for the momentum and continuity equation. In that way, the precision of the solution is not affected by MIGAL when convergence is reached. It remains the sole responsibility of PHOENICS.

Multi-grid procedure

The set of linear equations resulting of the discretisation of the continuity and the transport equations can be expressed as:

$$A x = b \quad (7)$$

where A represents the matrix of coefficients, x the variables array and b the right hand side vector of source terms.

There are several well-established iterative schemes to solve this set of linear equations. These include Jacobi, Gauss-Seidel, incomplete LU factorization, etc. But, each of them has a rate of convergence depending on the condition number of the matrix A . Hence, as the number of cells will increase, and because of the elliptic nature of the diffusion operator contained in the equation, the condition number of the matrix will increase and the rate of convergence will deteriorate. Further, a characteristic of all these iterative schemes is that the initial rate of convergence is rapid for the first iterations, and deteriorates as the iterations progress. It can be shown that the cause of this slow convergence is primarily the sluggish rate of convergence of the low frequency errors that are present in the solution. As the grid refined, these low frequency errors dominate the overall rate of convergence.

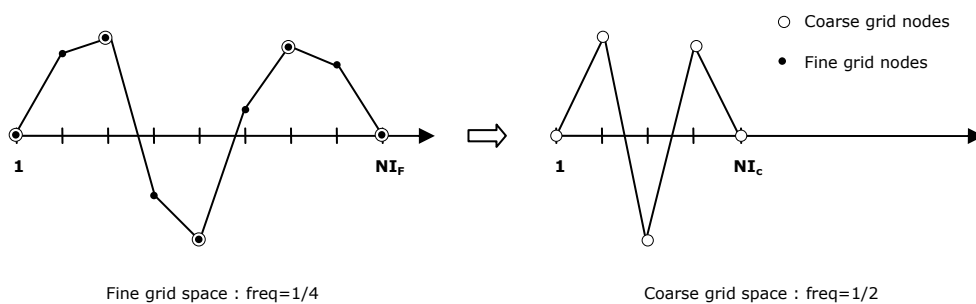


Figure 3: Fine to coarse grid error frequency mutation.

Hence, the concept of the multi-grid technique implemented by MIGAL is as follows. Given the fact that the low frequencies converge slowly, it is possible to accelerate their rate of convergence by making them behave as high frequencies on coarser grids. For this, the basic

principle consists in working on a subset of the fine grid points (e.g. keeping only odd or even nodes) to reduce the number of points and automatically raise the error signal frequency in the transformed grid space (see *Figure 3*).

MIGAL proceeds as follows. Consider that we initiate a solution on a given fine grid. A few iterations are performed on this fine grid to obtain a new estimation x_F . For these iterations, the convergence is usually fast.

$$x_F = \tilde{A}^{-1}b \quad (8)$$

The notation (\sim) means that \tilde{A}^{-1} is not the inverse of matrix A but only an approximate (e.g. few relaxations of an iterative solver). Subsequently, the convergence begins to worsen so that the calculations are switches to a coarser grid with the aim of improving the fine grid estimation x_F at lower cost. For that, the residuals and the corresponding defect correction operator are formed on the fine grid and interpolated ("restricted") to the next coarse grid by:

$$R A \delta x_F = R (b - Ax_F) \quad (9)$$

where the restriction operator R (e.g. pure injection) is a $(NI_c \times NI_F)$ matrix that shorten the dimension of the right hand side from NI_F to NI_c .

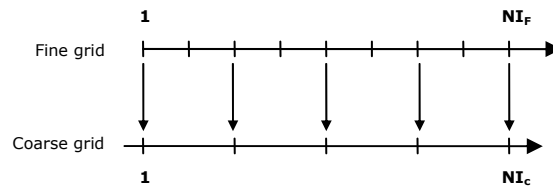


Figure 4: Pure injection restriction.

At this point a second operator is introduced to shorten the left hand side of the operator. This is done by changing the fine grid correction variable δx_F for a coarse grid variable named δx_c so that the former can be interpolated ("prolongated") from the latter by:

$$\delta x_F = P \delta x_c \quad (10)$$

where the prolongation operator P (e.g. linear interpolation) is a $(NI_F \times NI_c)$ matrix that reduces the coarse grid operator size to $(NI_c \times NI_c)$.

$$(R A P) \delta x_c = R (b - A x_F) \tag{11}$$

Once the coarse grid operator (RAP) is formed, a few iterations are performed to obtain δx_c with the required accuracy and the fine grid correction is retrieved using (10).

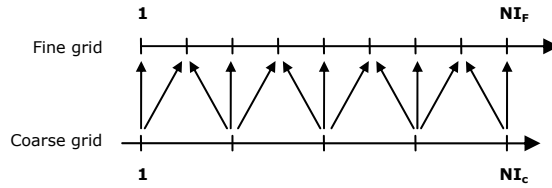


Figure 5: Linear prolongation.

Of course, since the error spectrum contains a wide range of frequencies, it is necessary to consider a number of coarse grids and successively build coarse operators of coarse operators. On the coarsest grid, which must be a small grid, a direct solver can be used or, like with MIGAL, the necessary number of iterations can be performed. Finally the manner in which the grids are visited can vary.

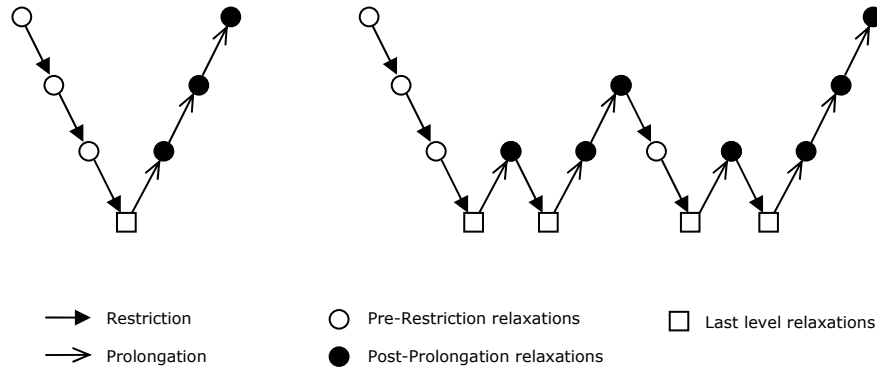


Figure 6: V and W multi-grid cycles

In the simplest case, called V-cycle, each grid is visited in turn on the downward and upward legs of a V-cycle. On each grid a number of iterations are performed and the next grid is visited. With the W-cycle, each grid is visited in turn on the downward leg of the cycle, then the correction is prolonged by one level, the error is again restricted to the lower level and

the correction prolonged to the next upper level (see Figure 6). The iterations can be done during both the restriction part of the cycle (downward limb) and during the prolongation part (upward limb).

In spite of a higher cost per cycle, and because of its robustness and efficiency with coupled sets of equations, MIGAL retains the W cycle as default. However, users can switch to V cycle by setting the ICYCLE parameter to 0 with the hope of saving CPU time in case of easy flow.

The default numbers of relaxations per grid are set to NBPRER=1 for pre-restriction and NBRELAX=3 for post-prolongation and last-level. Those parameters, especially NBRELAX, may have to be increased by users for large grids, encumbered domains or crucial scalar equations to ensure a sufficient smoothness and residual reduction during each sweep of the non-linear process.

Finally the deepness of the cycles i.e. the number of grid levels that MIGAL performs is automatically computed. Nevertheless, users can force it by setting the parameter NBGRID. In particular NBGRID=1 turn off the multi-grid feature and MIGAL becomes a simple ILU(0) solver.

ILU(0) Smoother

As evoked above, the result of the finite volume discretisation process is a system of algebraic equations whose form is:

$$A_p \phi_p = \sum A_{nb} \phi_{nb} + b \quad (12)$$

The coefficients of this generic form are either scalars for single equations or small matrices for coupled equations. In both cases the sum of equations (12) over all the control volumes produces such large and sparse systems (13) that only iterative methods are of interest for solving them.

$$A x = b \quad (13)$$

The basic operation of iterative methods usually consists in updating an estimate x^{n-1} from the residual of the system (13) in three steps:

$$r = b - A x^{n-1} \quad (14)$$

$$\delta x = \tilde{M}^{-1} r \quad (15)$$

$$x^n = x^{n-1} + \omega \delta x \quad (16)$$

Because the residual r and therefore the correction δx become null when convergence is reached, the matrix M of the second step is just an iteration matrix, as good approximation of A as possible, but making (15) easier and cheaper than solving (13). For example, with the partitioning:

$$A = C - E - F \quad (17)$$

in which C is the diagonal of A , $-E$ its strict lower part, and $-F$ its strict upper part, the Jacobi method corresponds to $M=C$, the forward Gauss-Seidel method to $M=C-E$ and the backward Gauss-Seidel method to $M=C-F$.

For solving (15) MIGAL implements the Incomplete LU decomposition ILU(0) in which the iteration matrix is given by:

$$M = (D - E)D^{-1}(D - F) \quad (18)$$

where D is not the diagonal of A like with the SSOR($\omega=1$) method but a diagonal matrix recursively defined from A 's coefficients accordingly to some formula such as:

$$D_{(i,j)} = A_{P(i,j)} - A_{W(i,j)}D_{(i-1,j)}^{-1}A_{E(i-1,j)} - A_{S(i,j)}D_{(i,j-1)}^{-1}A_{N(i,j-1)} \quad (19)$$

This method was named ILU(0) in respect of the fact that, because of (19) M becomes an approximated LU decomposition of A which doesn't fill-in its zero pattern. The former property ensures efficiency when solving (15) while the latter allows M to keep the structure of A and limits the requirement for additional storage to the inverse of the diagonal D .

$$\begin{cases} L = D - E \\ U = I - F \end{cases} \quad (20)$$

After the factorisation stage, the solution of (15) is obtained in two substitution steps firstly computing $\delta x = L^{-1}r$ and secondly $\delta x = U^{-1}\delta x$. These steps, accordingly to (20), follow forward and backward recursive formulas such as:

$$\begin{cases} \text{step1: } \delta x_{(i,j)} = D_{(i,j)}^{-1} \left(r_{(i,j)} - A_{W(i,j)} \delta x_{(i-1,j)} - A_{S(i,j)} \delta x_{(i,j-1)} \right) \\ \text{step2: } \delta x_{(i,j)} = \delta x_{(i,j)} - D_{(i,j)}^{-1} \left(A_{E(i,j)} \delta x_{(i+1,j)} - A_{N(i,j)} \delta x_{(i,j+1)} \right) \end{cases} \quad (21)$$

The relaxation factor ω in equation (16) corresponds to the parameter named RELAX in the MIGAL parameters list. Its default value is 0.95.

Momentum preconditioning

To improve the robustness of the ILU(0) coupled solver at high Reynolds numbers, MIGAL implements a preconditioner that increases the diagonal dominance of the linearised set of equations in the regions where the continuity equation is temporarily unsatisfied. To understand this preconditioner, let's consider the continuity (22) and the momentum (23) equations written as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j) = 0 \quad (22)$$

$$\frac{\partial}{\partial t} (\rho \phi) + \frac{\partial}{\partial x_j} (\rho u_j \phi) = \frac{\partial}{\partial x_j} \left(\Gamma \frac{\partial \phi}{\partial x_j} \right) + S \quad (23)$$

After discretisation, equation (23) takes the generic form (24) where a_{nb} and a_p represent respectively the coefficients of the neighbour and central points.

$$a_p \phi_p = \sum a_{nb} \phi_{nb} + b \quad (24)$$

As an iterative method, the ILU method starts from a guessed solution and performs successive updates until a converged solution is obtained. Then, depending on the initial guess and the structure of the coefficient matrix, this iterative process can fail to produce a solution i.e. may diverge. Historically, Scarborough [13] has shown that a sufficient condition for the Gauss-Seidel method to converge is that:

$$\frac{\sum_{nb} |a_{nb}|}{|a_p|} \begin{cases} \leq 1 & \text{for all nodes} \\ < 1 & \text{for at least one node} \end{cases} \quad (25)$$

Then, although this not a necessary requirement for the ILU method, experience proved that it is desirable for equation (24) to satisfy the diagonal dominance criterion (25). The common practice for this consists in ensuring the first part of the criterion by forcing the equality (26) for the coefficients of the transport equations and by letting the second part being provided by the boundary conditions and the time derivative:

$$a_p = \sum_{nb} a_{nb} \quad (26)$$

Unfortunately, the finite volume integration doesn't naturally satisfy the relation (26) but rather the relation (27) in which appears the term $div(\rho U)$ due to the convection terms.

$$a_p = \sum_{nb} a_{nb} + div(\rho U) \quad (27)$$

In order to remove this potentially dangerous term, the common practice (see e.g. [9]) consists in subtracting ϕ_p time the continuity equation (22) to the transport equations (23) after discretisation. But this manipulation has a drawback when, during the iteration process, all the neighbour coefficients a_{nb} become locally null. Then the coefficient matrix becomes badly conditioned and the convergence owes its salvation to the sole presence of the time derivative. Such a temporary situation is typically encountered in the stagnation zones of low viscosity flows where the fluxes are upwinded and $div(\rho U)$ is positive (see *Figure 7*).

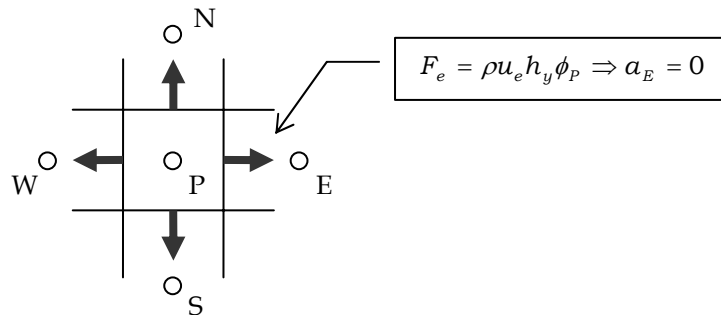


Figure 7: Typical velocity field producing singular coefficients

Since an advantage of MIGAL is to allow larger time steps when solving the velocity-pressure coupling, the time step restriction due to relation (26) has been removed by preconditioning the momentum equations of the coupled operator as follows:

$$(A + D)\delta x = r \quad (28)$$

Where A is the coupled operator, r the residuals, δx the velocity and pressure corrections and D a diagonal matrix whose non-zero terms correspond to the momentum lines and are such as:

$$D_{jj} = COEFF * \max[0, \text{div}(\rho U)] \quad (29)$$

The coefficient $COEFF$ has been introduced to provide users with the possibility of switching off the preconditioner when the initial non-linear convergence is perturbed. Its default value is 1.

GMRES acceleration

The emphasis of basic multi-grid procedures is to improve the performance of the classical iterative solvers (smoothers) by adapting the operator to their intrinsic capabilities. This technique usually encounters some limits when the coefficients are highly anisotropic and when the mesh aspect ratios are quite large. Nevertheless, due to some internal adaptations of the restriction and prolongation operators, the ILU(0) smoother of MIGAL has demonstrated to be robust enough in many applications. However, difficulties sometimes arise from some blockages and boundary conditions that couple the high and low frequencies error components during the restriction/prolongation procedure and finally deteriorate the overall multi-grid performance. To fight this wavelength coupling problem MIGAL may also be used as a GMRES preconditioner.

The GMRES method [12] is a projection based method that consists in finding the optimal solution of the system that belongs to the m -th Krylov subspace K_m .

$$K_m = \text{span} \left\{ r_0, AM^{-1}r_0, [AM^{-1}]^2 r_0, \dots, [AM^{-1}]^{m-1} r_0 \right\} \quad (30)$$

where r_0 is the initial residual of and where M^{-1} is a right preconditioning matrix whose function is to lower the condition number of the algebraic set of equations.

$$AM^{-1}u = b, \quad u = Mx \quad (31)$$

For a given dimension m , the MIGAL-GMRES algorithm involves an Arnoldi loop that constructs an orthogonal basis of the right-preconditioned Krylov subspace by a modified Gram-Schmidt process, in which the new vector to be orthogonalized is obtained from the previous vector of the process.

1. **Start:** Choose x_0 and a dimension m of the Krylov subspaces and initialize a $(m+1) \times m$ matrix H to zero
2. **Arnoldi process:**
 - Compute $r_0 = b - Ax_0$, $d = \|r_0\|$ and $v_1 = r_0/d$
 - For $j=1, \dots, m$
 - Compute $z_j = M^{-1} v_j$
 - Compute $w = A z_j$
 - For $i=1, \dots, j$

$$H_{i,j} = (w, v_i)$$

$$w = w - H_{i,j} v_i$$
 - Compute $h_{j+1,j} = \|w\|$ and $v_{j+1} = w/H_{j+1,j}$
 - Define $Z_m = [z_1, \dots, z_m]$
3. **From the approximate solution:** Compute $x_m = x_0 + Z_m y_m$ where $y_m = \operatorname{argmin}_y \|d e_1 - Hy\|$ and $e_1 = [1, 0, \dots, 0]^T$
4. **Restart:** if satisfied stop, else set $x_0 = x_m$ and goto 2.

Figure 8: MIGAL-GMRES algorithm

The preconditioning matrix is only involved in the $z = M^{-1}v$ products and does not need to be explicitly formulated. Instead, given that $M=A$ is the best preconditioning choice (see e.g. (31)), some multi-grid cycles are advantageously used to compute an approximated value of z as being the solution of $Az=v$.

The drawback is that, the matrix M now changes for each vector z depending on the convergence of the multi-grid cycles, and that it is necessary to store the orthogonal basis Z_m to retrieve the solution x_m . Therefore, since reaching an expected level of accuracy may involve large sub-spaces, the method may become impractical because of large memory and computational requirements. For this reason MIGAL use a restarted GMRES procedure (step 4 Figure 8) which limits the Krylov-subspace basis to a given size and iterates the initial estimation x_0 .

By default MIGAL doesn't implement its GMRES acceleration capability because of this additional memory requirement but its use is recommended for difficult cases. So, to switch on GMRES, users have to specify a non-zero size, IGMRES, for the Krylov subspace basis. IGMRES values of 3 or 5 are usual. Then MIGAL will precondition the Krylov subspace by one multi-grid cycle. To increase this number of preconditioning cycles users may specify the parameter IPRECO in the MIGAL parameters list.

Finally, for particularly difficult problems (rarely observed for velocity-pressure coupling), it is possible, on coarse grid levels, to replace the ILU(0) smoother by a GMRES ILU(0) preconditioned smoother. For this users have to stipulate the size of the desired Krylov subspace by setting the parameter IGMS to any non-zero value. The coarse grid smoother then becomes a GMRES solver right-preconditioned by NBPRER or NBRELAX ILU(0) relaxations depending of the limb of the multi-grid cycle actually performed. For flexibility, the IGMS parameter may be set independently of IGMRES, i.e. that MIGAL can implement different combinations of the multi-grid and GMRES algorithms.

<i>SOLVER</i>	<i>NBGRID</i>	<i>IGMRES</i>	<i>IPRECO</i>	<i>IGMS</i>
ILU(0)	1	0	-	0
GMRES not-preconditioned	1	n	0	0
GMRES ILU(0) preconditioned	1	n	n	0
Multi-grid[ILU(0)]*	n	0	-	0
GMRES multi-grid[ILU(0)] preconditioned	n	n	n	0
Multi-grid[GMRES-ILU(0) preconditioned]	n	0	-	n
GMRES multi-grid[GMRES-ILU(0) preconditioned] preconditioned	n	n	n	n

Figure 9: Multi-grid/GMRES possible combinations. ()=default*

Performance

Though MIGAL is only a linear solver, this work is not focussed on its performance as a linear equation solver. Because of the non-linearities and the explicit couplings in the equations there is no point trying to achieve a very precise solution on any one sweep. The point of this paper is rather to demonstrate the performance of MIGAL in accelerating the global convergence on concrete test cases. Therefore, all the curves presented below refer to the non-linear convergence process instead of the convergence of a single linearised system of equations. For the industrial test cases (supermarket, racing car), the outer loop was driven to a reasonable, though not fully converged, level of convergence: about 3 orders of residuals reduction. When compared, the convergence curves are gathered in one, the residuals are normalised by a common initial value and only the mean curve is plotted.

Lid-driven cavity

This first problem illustrated *Figure 20* is the classical lid-driven cavity laminar flow at a Reynolds number $Re=10,000$ for the two-dimensional case and $Re=1,000$ for the three-dimensional case. This test is the most suitable case to focus on the coupling between the continuity and the

momentum equations since boundary conditions and physical property are constant and uniform. *Figure 10* shows how the non-linear convergence is modified by the strong coupling procedure of MIGAL and how a large speed-up can be expected on a reasonable two-dimensional grid definition (194x194). The same trend is observed for the three-dimensional case which also demonstrates how the traditional segregating techniques performance fails when the number of grid cell increases and how MIGAL's feature becomes more and more desirable. In both cases, no false time step has been used to reach the converged solution.

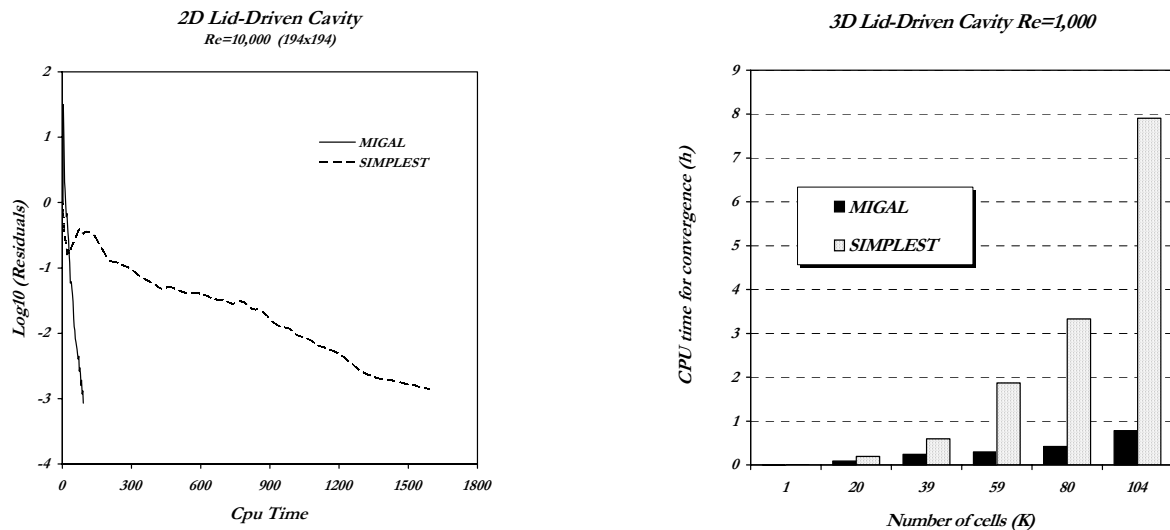


Figure 10: Performance comparison between MIGAL and SIMPLEST

Low Reynolds Flows

This case illustrated *Figure 22* concerns the flow past a flat plate at low Reynolds number. It has been chosen to show the benefit of the MIGAL multi-grid feature on the global non-linear convergence. In particular, one may observe on *Figure 12* the large speed-up produced by involving more grid levels. Given the moderate grid size 40x140 and its anisotropic distribution, the automatic setting of MIGAL used a semi-coarsening strategy and produced 8 grid levels. This case also illustrates the great predilection of MIGAL for low-Reynolds numbers with speed-up over 50 as observed on *Figure 19* for larger grids. Indeed the multi-grid acceleration effect of MIGAL on the global convergence is due to the coupling technique. As shown by *Figure 11*, using MIGAL separately as a solver for the momentum and continuity equations, does not improve the SIMPLEST performance but even deteriorates it because of the higher cost inherent to the multi-grid procedure. This proves, at least for this case, that the over all procedure needs to be strongly coupled to benefit from multi-grid acceleration and that there is no point to work too much on segregated equations. In this case, using the multi-grid solver for separate variables U_1 , V_1 and P_1 increases the CPU time per sweep from 0.14 to 1.83 but does not sufficiently increase the convergence rate per sweep. Alternatively, for coupled variables, when MIGAL increases the CPU time per sweep up to 2.41 it also considerably reduces the required number of sweeps. Only 17 sweeps were performed to reach the converged solution vs. 5,700 for SIMPLEST and 2,400 for SIMPLEST(multi-grid).

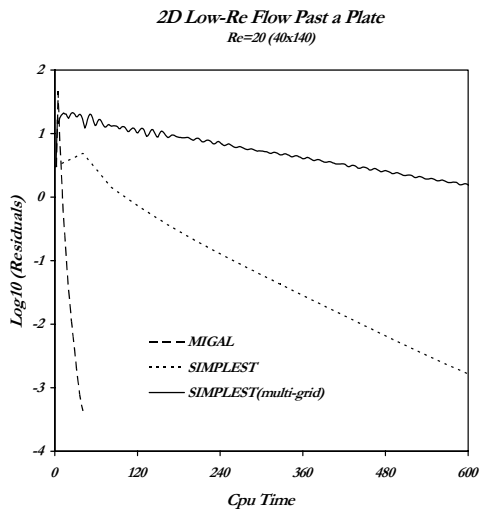


Figure 11: Segregated vs. Coupled

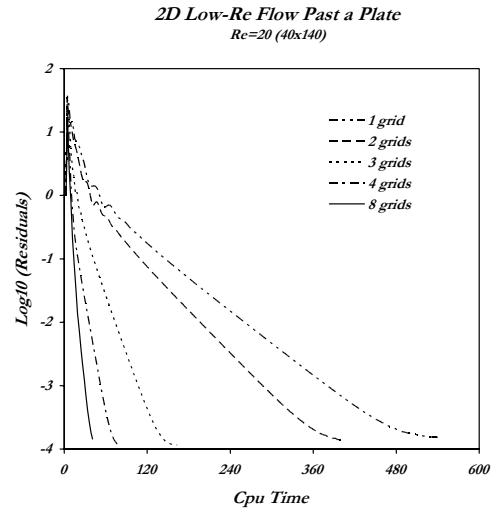


Figure 12: Varying multi-grid depth

Buoyancy-driven flows

The third case is also an academic problem illustrated *Figure 23*. It is the classical free convection flow in a square cavity formed by two vertical walls, one of which is heated and the other cooled. The top and the bottom of the cavity are bounded by walls at which there is friction but no heat transfer. The Rayleigh and Prantl numbers of this problem are respectively $Ra=100,000$ and $Pr=0.1$. Though MIGAL does not directly coupled the temperature equation to the momentum and continuity equations, *Figure 13* shows a large speed-up in this particular case of Bousinesq approximation with boundary conditions not strongly affected by the flow. This figure plots the mean normalized residual (by initial values) history for the velocity, pressure and temperature equations gathered in one curve.

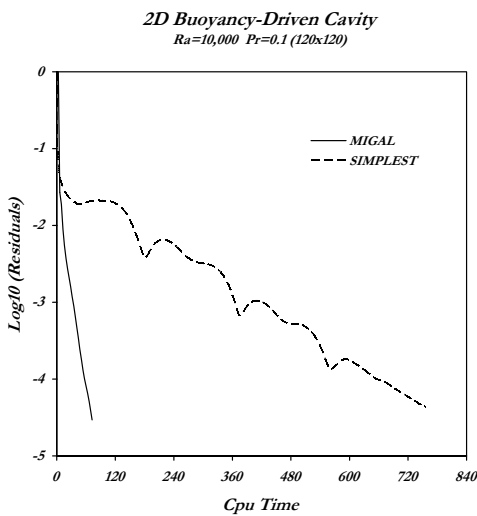


Figure 13: 2D Buoyancy driven cavity

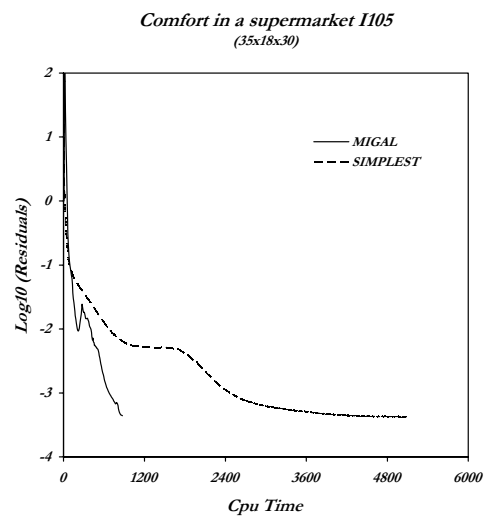


Figure 14: 3D Supermarket flow

Because of the segregation of the temperature equation, the solution has been obtained by using a false time step on momentum equations. Nevertheless, this constraint was negligible compare to the benefit of a globally better continuity fulfilment during iterations and *Figure 19* shows that this speed-up greatly increases with the number of cells. *Figure 14* shows that seed-ups are also observed on industrial buoyancy driven flows.

Turbulent flows

The next cases are illustrated *Figure 23* and *Figure 24*. They concern respectively the turbulent flows of a flow-driven cavity at Reynolds $Re=80,000$ and of the classical backward facing step at Reynolds $Re=50,000$. Both are computed using the $k-\varepsilon$ model. Turbulent flows are usually less suited to MIGAL because of the explicit coupling between the momentum equations and the effective viscosity. For this reason, the backward facing step doesn't benefit of any significant speed-up *Figure 15*. Nevertheless, it is not always the case and *Figure 16* shows a clear advantage for the coupled solution. The curves of these figures are the mean normalized residual (by initial values) history for the velocity, pressure, k and ε equations gathered in one curve.

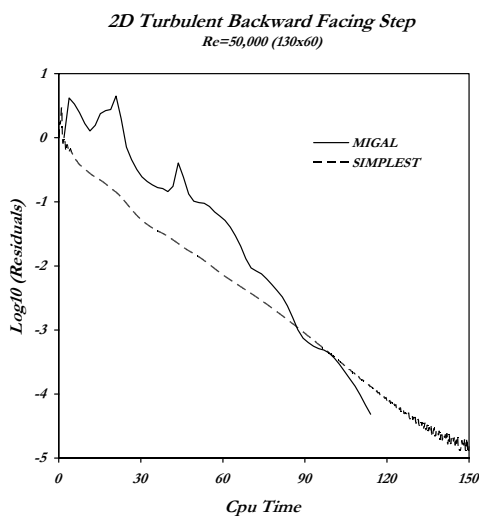


Figure 15: Backward Facing Step

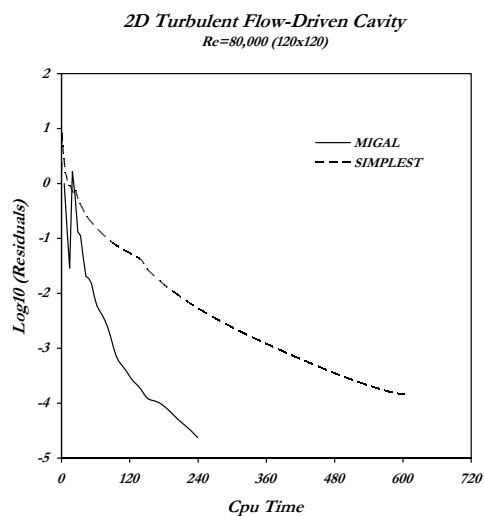


Figure 16: Flow-Driven Cavity

The backward facing step case has been run with no false time step for the momentum equations and a linear relaxation for the turbulent variables $LINRLX=0.7$. Because the critical point for a good convergence of the $k-\varepsilon$ model stands in smoothness of the production term G the post-prolongation relaxations has been rise to 10. MIGAL has not been used for the turbulent variables. Usually the number of sweeps to reach a converged solution is larger for turbulent flows (about 80) than for laminar flows (about 20). But this number (60 for both cases) remains small compare to the segregated approach where thousands are needed (1400 for backward facing step, 4300 for flow-

driven cavity). For the flow-driven cavity case, the same false time step than used with SIMPLEST has been kept for the coupled solution and MIGAL has been applied to the hydrodynamic coupling as well as for the turbulent variables. Other speed-up examples of turbulent flows can be observed for the industrial cases *Figure 14* and *Figure 18*.

GMRES acceleration

To illustrate the benefit of using GMRES with MIGAL, the PHOENICS VR tutorial 4 *Figure 25* has been run with an 80x40 grid. This case is a two-dimensional internal flow with blockages that implements the LVEL turbulent model. The Reynolds number based on the width of the domain is $Re=153,000$. Because of the two arrays of blockages, twelve recirculation zones are present and the multi-grid technique suffers from linking together different types of flows. *Figure 17* shows that after a similar reduction rate for the two first orders the convergence is clearly improved by GMRES. The size of the Krylov-subspace was fixed at 5 and the number of post-prolongation relaxations at 10. Others cases implementing GMRES are the industrial cases of the supermarket and of the racing car *Figure 14* and *Figure 18*.

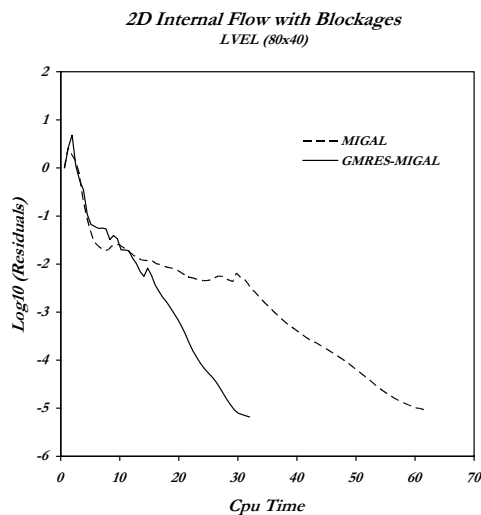


Figure 17: GMRES acceleration

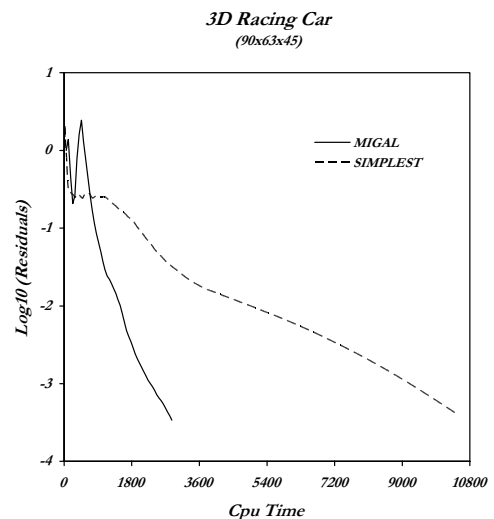


Figure 18: Racing car flow's convergence

Industrial flows

MIGAL is also currently used on industrial flows for its robustness and convergence acceleration. The very optimal parameters need sometimes experience to be found but once settled they are not very sensitive to grid refinement and save hours of computation. To illustrate MIGAL's capabilities on industrial flows, two additional cases are considered.

The first one is the PHOENICS-FLAIR library case I105 *Figure 26*. It is a three-dimensional model of a supermarket in which customer's comfort is affected both by sun when shining on the side of a

big glass window and by two large freezers. Air is constantly exchanged through 8 air-conditioning vents and 6 extracting openings. Turbulence is modelled by the $k-\varepsilon$ model and the grid set-up is moderate $35 \times 18 \times 35$. This case is difficult since a strong residual reduction is necessary to establish the flow equilibrium. In particular one can see, for the segregated approach on *Figure 14*, a plateau after two order of reduction followed by a long degradation of the rate of convergence. Alternatively, MIGAL keeps its performance and the needed level of convergence is reached in 120 sweeps vs. 4000 for SIMPLEST. The final speed-up stands about 6.

The second case illustrated *Figure 27* concerns the flow around a racing car model. The characteristic Reynolds number is $Re=400,000$ and $k-\varepsilon$ turbulence model activated. In spite of the body symmetry, the entire domain is meshed by a relatively coarse $90 \times 63 \times 45$ grid. This case is difficult in reason of the complex geometry and of some instability in the mid-plane of the wake. Nevertheless, the coupled procedure was found to drive the solution to convergence in 50 sweeps vs. 800 for SIMPLEST and to produce a speed-up of 3.5.

Finally, a sum up of some of the MIGAL speed-ups observed during the validation process [4] are plotted *Figure 19*. In particular, one can note, that the multi-grid feature is more effective with two-dimensional cases since, for a given number of cells, the number of points in each direction is larger. Nevertheless, because of longer runs, the three-dimensional speed-ups are much more impressive in term of time saving.

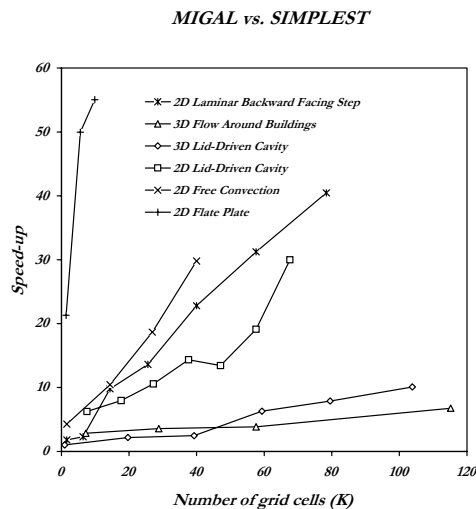


Figure 19: Speed-up examples

Conclusions

During the last two years, attention has been focussed on the robustness of MIGAL for PHOENICS and some new features like a GMRES option or internal tunings have been introduced. Nowadays, MIGAL produces a large degree of improvement of the convergence rate in many academic and

industrial flows. These speed-ups are robust in the sense that optimal parameters need only few adjustments when the grid density increases. Moreover, due to a poor behaviour of the traditional segregated approach, they also increase simultaneously with the number of grid cell.

The main drawback of the method is a serious storage penalty due partly to the coefficients of the fully coupled system and partly to the ILU factorisation. The additional memory requirement is approximately of $7.E-4$ Mb per cell for three-dimensional cases and $4.E-4$ Mb per cell for two-dimensional cases. Fortunately, if this drawback was unaffordable until the 1990's, it is naturally vanishing with progress of the computer technology. The available memory constantly increases and the size of the problems that MIGAL can achieve consequently enlarges, producing stronger and stronger speed-ups. For example, considering the three-dimensional lid-driven cavity problem, 8,000 cells filled the 8Mb of PC's memory when using MIGAL in 1990 and the solution took 1h30 to converge. Nowadays, 512Mb of memory allows running MIGAL for 640,000 cells and take the same time (1h30) because of processor improvements. Simultaneously, for the same grid sizes of the same problem the traditional segregate approach took 2h in 1990 and takes more than 90h nowadays. It is therefore clear that the future belongs to new algorithms such as MIGAL.

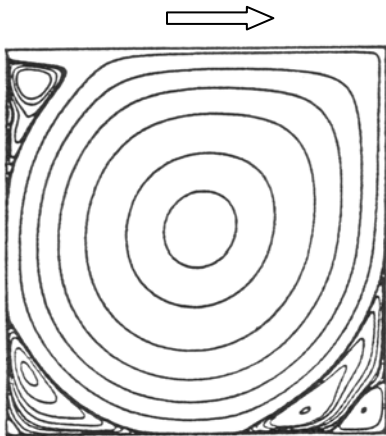


Figure 20: Lid-Driven Cavity

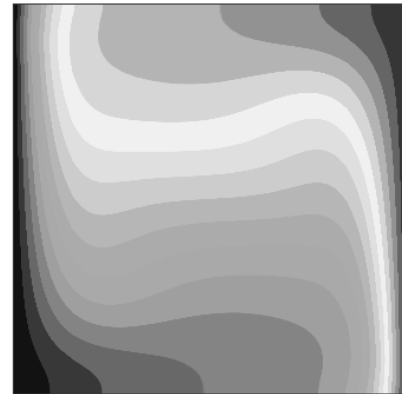


Figure 21: Laminar free convection cavity

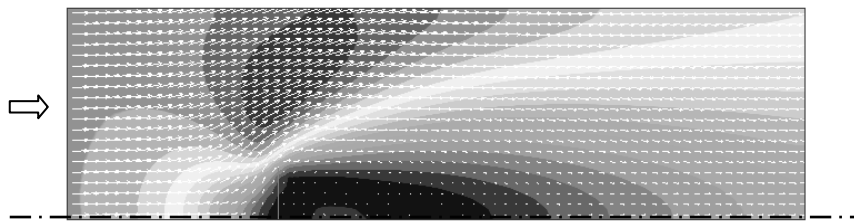


Figure 22: Flow past a flat plate at low-Reynolds number $Re=40$.

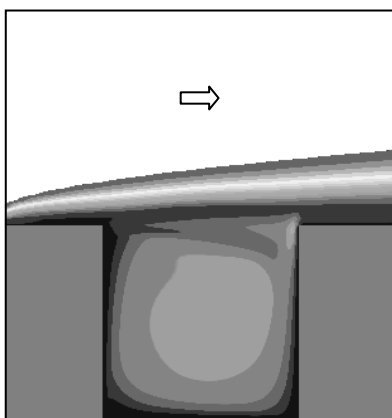


Figure 23: Flow-driven cavity

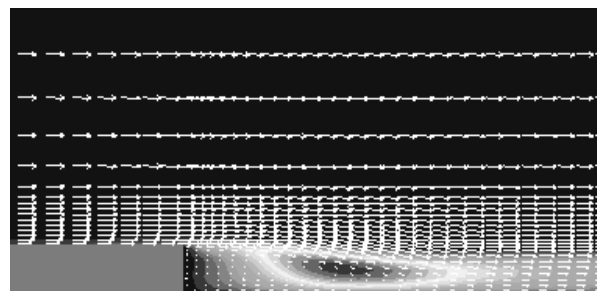


Figure 24: Backward facing step

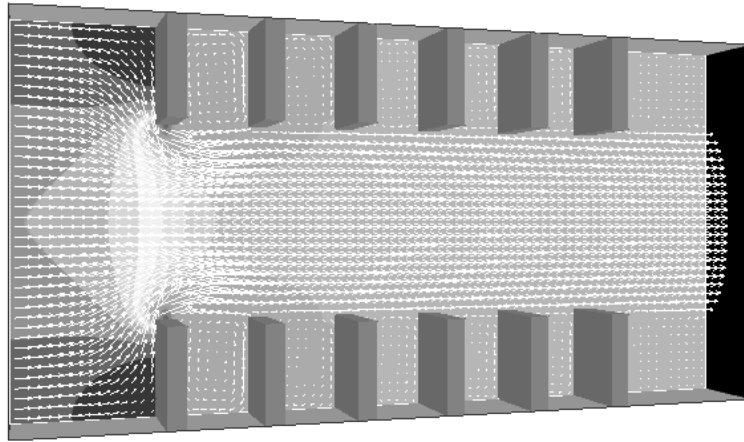


Figure 25: Two-dimensional internal flow with arrays of blockages.

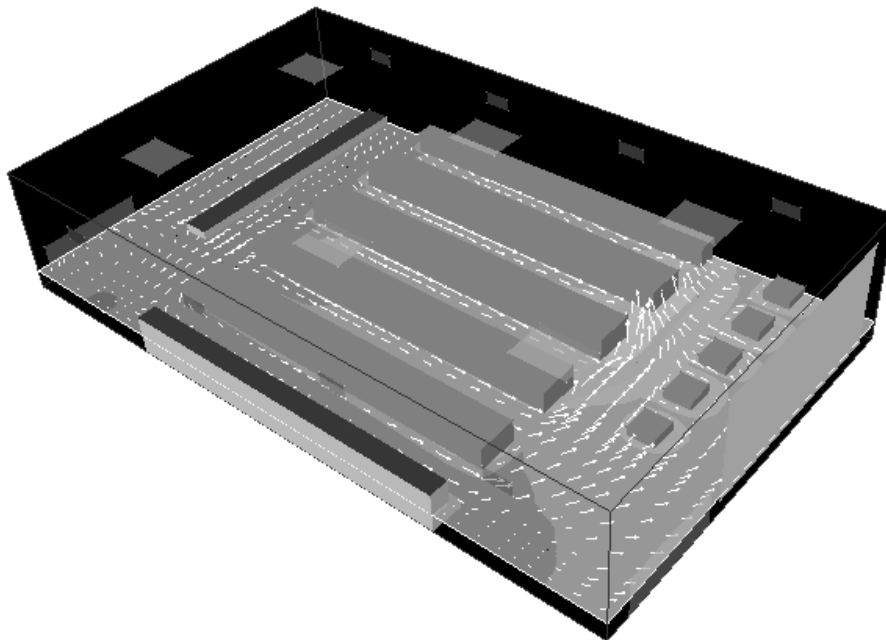


Figure 26: Supermarket air velocity and ambient temperature.

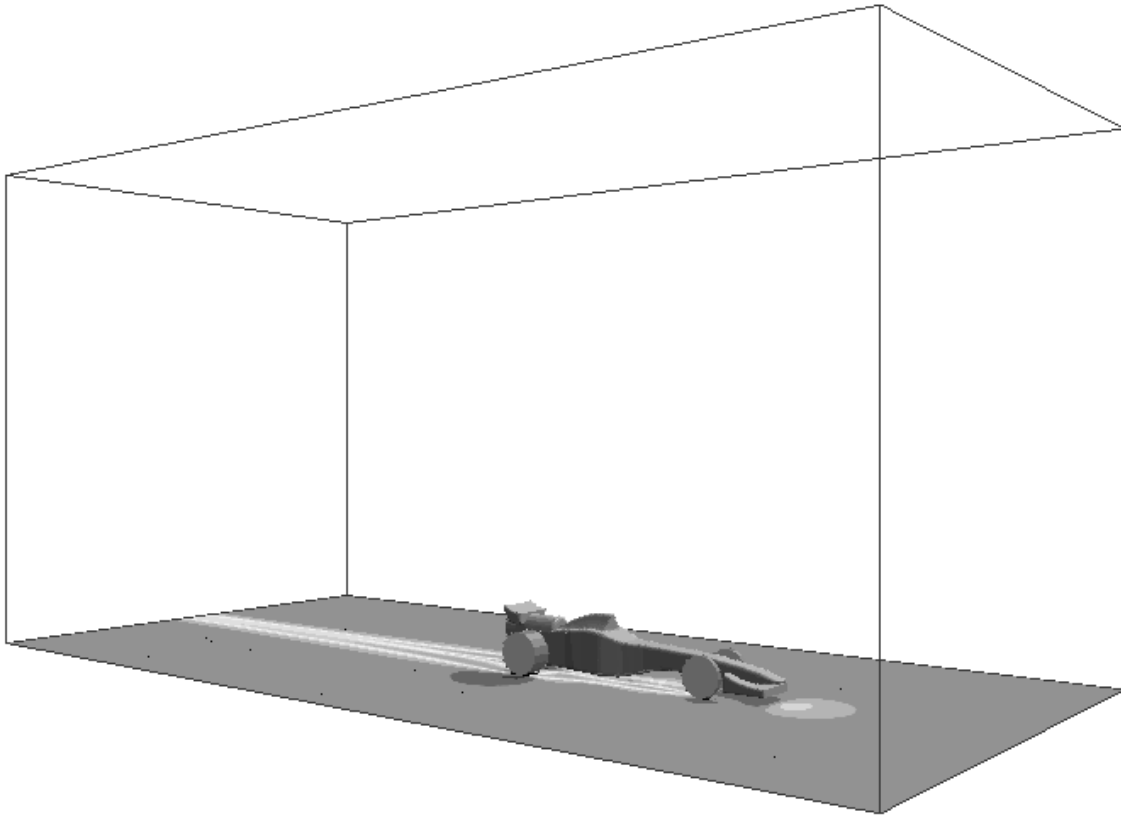


Figure 27: Flow past a racing car model.

References

- [1] Ch. Arakawa, A.O. Demuren, W. Rodi, B. Schönung, "Application of Multigrid Methods for the Coupled and Decoupled Solution of the Incompressible Navier-Stokes Equations", Notes on Num. Fluids Mech. GAMM-Conference, Vol. 20 pp 1-8, 1987.
- [2] W.L. Briggs, "A Multigrid Tutorial", SIAM, Philadelphia 1987.
- [3] L.S. Carretto, R.M. Curr and D.B. Spalding, "A Calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows", Int. J. Heat Mass Transfer, 15, pp 1878-1806, 1972.
- [4] M. Ferry, "Introducing MIGAL for PHOENICS 3.3", International PHOENICS users Conference, Luxembourg, May 2000.
- [5] M. Ferry and J. Piquet, "A new fully coupled method for the solution of Navier-Stokes equations", Proc. 12th. Int. Conf. Num. Methods Fluid Dynamics, Oxford, Lecture Notes in Physics, Vol. 371, pp.186-187, 1991.
- [6] M. Ferry and J. Piquet, "Evaluation of a new fully coupled method for the solution of Navier-Stokes equations" Proc. 3rd. European Multigrid Conference, Bonn. GMD Report, Trottenberg, Ed. 1991.
- [7] B.R. Hutchinson, P.F Galpin, and G.D. Raithby, "Application of additive correction Multigrid to the Coupled Fluid Flow Equations", Numerical Heat Transfer, Vol. 13, pp. 133-147, 1988.
- [8] K.C. Karki and H.C. Mongia, "Evaluation of a coupled solution approach for fluid flow calculation in body fitted coordinates", Int. J. Num. Meth. Fluids, 9, 325-340, 1990.
- [9] S.V. Patankar, "Numerical Heat Transfer and Fluid Flow", Hemisphere, Washington D.C., 1980.
- [10] S.V. Patankar and D.B. Spalding, "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-dimensional Parabolic Flows", Int. J. Heta Mass Transfer, Vol. 15, p. 1787.
- [11] M. Raw, "Robustness of Coupled Algebraic Multigrid for the Navier-Stokes Equations", AIAA 96-0297, 1996.
- [12] Y. Saad and M.H. Schultz, "GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems", SIAM J. Sci. Stat. Comput., 7(3):856-869, 1986.
- [13] G.E. Schneider and M. Zedan, "A Coupled modified strongly implicit procedure for the numerical solution of coupled continuum problem", AIAA 84-1743, 1984.
- [14] J.B. Scarborough, "Numerical Mathematical Analysis", 4th Ed, Johns Hopkins Press, Baltimore, 1958.
- [15] S.P. Vanka, "Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables", J. of Computational Physics, Vol. 65, pp 138-158, 1986.