CHAPTER 35

Kinematic Domain Decomposition for Boundary-Motion-Induced Flow Simulations

Oktay Baysal*
Guan-Wei Yen†

ABSTRACT

A method is developed to solve the unsteady Navier-Stokes equations on a composite grid, which consists of subdomain grids moving with respect to each other. These subdomains are structured grids with different topologies. This method eliminates assuming the moving components to be instantaneously stationary, where deciding on the particular frozen instants is difficult and affects the solution adversely. Moreover, this method captures the boundary-motion-induced flow component. The method is demonstrated through a transonic flow past an airfoil, which experiences a combined motion of pitching and plunging. An O-grid around the airfoil is overlapped on a fine Cartesian grid, which is zonally embedded in a coarse Cartesian grid. The coarse grid is stationary but the other two grids are plunging. Only the O-grid is also sinusoidally pitching. The results are compared successfully with the experimental data.

NOMENCLATURE

C	:	chord length	q_f	:	primitive variables in a finer grid
C_l	:	lift coefficient	q_p	:	primitive variables of pseudo-fine grid points
C_p	:	pressure coefficient			
d	:	grid step size	t	:	time
DDT	:	domain decomposition technique	u_1, u_2, u_3	:	velocity components
DOF	:	degree-of-freedom	Re	:	Reynolds number
G_1, G_2, G_3	:	subdomain grids	s	:	distance between a finer and a coarser grid centers
HDD	:	hybrid domain decomposition			
i	:	index in ξ^3 direction	x_1, x_2, x_3	:	Cartesian coordinates
k	:	reduced frequency	α	:	angle of attack
KDD	:	kinematic domain decomposition	Δ, Δ^2	:	upwind and central differences
M_∞	:	freestream Mach number	ξ^1, ξ^2, ξ^3	:	generalized curvilinear coordinates
q_c	:	primitive variables in a coarser grid	$\uparrow\downarrow$	:	up and down motions

*Old Dominion University, Mechanical Engineering and Mechanics Department, Norfolk, VA 23529. Associate Professor, Member SIAM.
†Old Dominion University, Mechanical Engineering and Mechanics Department, Norfolk, VA 23529. Graduate Research Assistant.

INTRODUCTION

A domain decomposition technique (DDT) subdivides the flow domain into simpler subdomains which accept easily constructed grids. Basically, there are four DDT's: block-structured grids, overlapped grids, zonal grids, and degenerate zonal grids. Block structured grids are the easiest to apply, but the least flexible with geometries since the grid lines normal to the intergrid boundaries must be contiguous. Zonal grids require common boundaries, which must be planar for the conservation of fluxes across the interface. Degenerate zonal grids are a special form of zonal grids, which require only a predetermined set of grid lines normal to the intergrid boundary to be contiguous, such as, every n-th line. Overlapped grids are the most flexible with virtually no restrictions on geometries. However, it is rather difficult to construct a flux conserving scheme for the intergrid communications.

A class of unsteady flow problems involve configurations with one or more of the components in relative motion with respect to the other components. It is difficult, if not often impossible, to generate a single-domain, structured grid for such a problem. The present method is the extension of the Hybrid Domain Decomposition reported in Ref. 1–4, to the moving boundary problems. Since at least one of the subdomain grids is moving, this method is called Kinematic Domain Decomposition (KDD) herein. Simulating such moving-boundary flowfields is also possible through static animations [e.g. Refs. 2, 3, 5]. KDD, however, eliminates the need to assume the moving components to be instantaneously static. Deciding on the particular frozen instants is often difficult and different choices may produce non-unique results. KDD has previously been successfully applied to solve the flow past a sinusoidally pitching airfoil and the flow past a cosine plunging airfoil [Ref. 6].

Possible applications of KDD include the separation of a store from a cavity, rotor stator interaction in turbomachinery, propellers on a complete aircraft, tube-launched projectiles, staging process for the multistage rockets, separation of the fuel tanks from the space shuttle.

GOVERNING EQUATIONS

The thin-layer approximation to the Reynolds-averaged, unsteady, compressible, Navier-Stokes equations are written in conservation form and in the time-dependent curvilinear coordinates,

$$\xi^m = \xi^m(x_1, x_2, x_3, t)\ ,\ m = 1, 2, 3 \tag{1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial E_1}{\partial \xi^1} + \frac{\partial E_2}{\partial \xi^2} + \frac{\partial (E_3 - E_{3v})}{\partial \xi^3} = 0 \tag{2}$$

where Q is the vector of conserved flow properties. The inviscid fluxes are denoted by E_1, E_2, E_3, and the thin layer viscous flux is denoted by E_{3v} [Refs. 1, 6]. The fluid is assumed to be a perfect gas. The Sutherland formula is used to determine the molecular viscosity. Reynolds stresses are modeled with a modified Baldwin-Lomax turbulence model [Ref. 7].

Equation 2 is solved using the implicit, finite-volume, upwind algorithm described in Refs. 1, 2, 3 and 8. Roe flux-difference splitting is used to construct the upwind differences for the convective and pressure terms. The diffusion terms are centrally differenced. Spatial approximate factorization and Euler backward integration, after linearization in time, result in the solution through 5x5 block-tridiagonal matrix inversions in three directions. The delta form of Eq. 2 obtained in this manner is given below,

$$\begin{aligned} &\left[\frac{I}{J\Delta t} + \Delta_{\xi^1}(\partial_Q E_1)\right] * \left[\frac{I}{J\Delta t} + \Delta_{\xi^2}(\partial_Q E_2)\right] \\ &* \left[\frac{I}{J\Delta t} + \Delta_{\xi^3}(\partial_Q E_3) - \Delta^2_{\xi^3}(\partial_Q E_{v3})\right] * \Delta Q = -\mathrm{Res}(Q^n) \end{aligned} \tag{3}$$

where

$$Res = \left[\Delta_{\xi m}\hat{E}_m - \Delta^2_{\xi 3}\hat{E}_{v3}\right]\ \ (sum\ over\ m) \tag{4}$$

The residual, $\mathrm{Res}(Q^n)$, is the discretized representation of the spatial derivative terms in Eq. 2 evaluated at the known time level (n). The accuracy of this scheme is second-order spatially and first-order temporally.

Kinematic Domain Decomposition

In addition to the more popular domain decomposition methods (their descriptions are given in Refs. 1–4, 9, 10), the degenerate zonal method is included in KDD for its simplicity and suitability in refining the grid where necessary. It is required that only every n-th line normal to the grid interface is contiguous. If n=2, then the line which is not contiguous bisects the distance between the neighboring two contiguous lines. This concept is a natural extension from the generation of coarse-fine grids necessary in the multigrid convergence acceleration methods [Refs. 1, 8]. A schematic of the degenerate zonal grid scheme, when all the grids are static, is shown in Fig. 1 for two dimensions. The dashed lines define a finer grid embedded completely within a coarser grid depicted by the solid lines. The crosses are the cell-center locations of the finer embedded grid. Each of the grids are a rectangularly-ordered set of points; in the sketch, a portion of the flowfield is covered both by the embedded grid and a portion of the coarser grid. The grids are coupled together during the solution process. The cell-center variables on a coarser grid cell which underlies a finer embedded grid cell are replaced with a volume-weighted restriction of variables from the four (two dimensional) or eight (three dimensional) finer grid cells, similar to the restriction operators used in a global multigrid scheme.

For the embedded finer grid, the computation boundaries occur either at a physical boundary or along an interior line of a coarser grid. Along such a boundary, two additional lines of data corresponding to an analytical continuation of the finer grid cell centers are constructed from interpolation (or prolongation) of the coarser grid primitive variables (Fig. 1). Since for most of the applications subdomain grids are nonuniform, the interpolation formula should allow variable step sizes (d) in all directions. Although a linear interpolation formula is most commonly used for the prolongation, an inverse averaging formula is more effective when more than two points are used for the interpolation. First, the flow properties at coarse grid cell centers are interpolated to "pseudo-fine" cell centers in each ξ^3—constant plane of a three-dimensional grid by the following inverse averaging on a four-point stencil (Fig. 2.a),

$$q_p(i) = \left[\frac{q_c(A)}{\sqrt{s_1^2+s_2^2}} + \frac{q_c(B)}{\sqrt{s_1^2+(d_2-s_2)^2}} + \frac{q_c(C)}{\sqrt{(d_1-s_1)^2+(d_2-s_2)^2}} + \frac{q_c(D)}{\sqrt{(d_1-s_1)^2+s_2^2}}\right] \Bigg/ \left[\frac{1}{\sqrt{s_1^2+s_2^2}} + \frac{1}{\sqrt{s_1^2+(d_2-s_2)^2}} + \frac{1}{\sqrt{(d_1-s_1)^2+(d_2-s_2)^2}} + \frac{1}{\sqrt{(d_1-s_1)^2+s_2^2}}\right] \tag{5}$$

Then, the flow properties of a fine grid cell center is obtained by interpolating the values of three contiguous pseudo-fine cell centers in the ξ^3 direction (Fig. 2.b),

$$q_f = \left[\frac{q_p(1)}{d_3-s_3} + \frac{q_p(2)}{s_3} + \frac{q_p(3)}{d_4+s_3}\right] \Bigg/ \left[\frac{1}{d_3-s_3} + \frac{1}{s_3} + \frac{1}{d_4+s_3}\right] \tag{6}$$

In general, none of the (d) or (s) values are equal to each other. A two-dimensional problem requires only Eq. 5 and the pseudo-fine cell centers become the actual cell centers of the fine grid.

During the restriction procedure, the primitive variables of a coarse grid are replaced from the nearest eight (three-dimensional) finer grid cells using the following inverse averaging formula based

on the instantaneous relative distances (Fig. 3),

$$q_c = \left[\frac{q_f(1)}{\sqrt{s_1^2+s_2^2+s_3^2}} + \frac{q_f(2)}{\sqrt{s_1^2+(d_2-s_2)^2+s_3^2}} + \frac{q_f(3)}{\sqrt{s_1^2+s_2^2+(d_3-s_3)^2}} \right.$$
$$+ \frac{q_f(4)}{\sqrt{s_1^2+(d_2-s_2)^2+(d_3-s_3)^2}} + \frac{q_f(5)}{\sqrt{(d_1-s_1)^2+s_2^2+s_3^2}} + \frac{q(6)}{\sqrt{(d_1-s_1)^2+s_2^2+(d_3-s_3)^2}}$$
$$\left. + \frac{q_f(7)}{\sqrt{(d_1-s_1)^2+(d_2-s_2)^2+s_3^2}} + \frac{q_f(8)}{\sqrt{(d_1-s_1)^2+(d_2-s_2)^2+(d_3-s_3)^2}} \right]$$
$$\Big/ \left[\frac{1}{\sqrt{s_1^2+s_2^2+s_3^2}} + \frac{1}{\sqrt{s_1^2+(d_2-s_2)^2+s_3^2}} + \frac{1}{\sqrt{s_1^2+s_2^2+(d_3-s_3)^2}} \right.$$
$$+ \frac{1}{\sqrt{s_1^2+(d_2-s_2)^2+(d_3-s_3)^2}} + \frac{1}{\sqrt{(d_1-s_1)^2+s_2^2+s_3^2}} + \frac{1}{\sqrt{(d_1-s_1)^2+s_2^2+(d_3-s_3)^2}}$$
$$\left. + \frac{1}{\sqrt{(d_1-s_1)^2+(d_2-s_2)^2+s_3^2}} + \frac{1}{\sqrt{(d_1-s_1)^2+(d_2-s_2)^2+(d_3-s_3)^2}} \right] \tag{7}$$

To decide on the type of hybrid domain decomposition for a problem with static boundaries [Refs. 1–4], two major concerns are the geometry of the configuration and a priori knowledge of the flowfield. In the case of dynamic objects, an additional concern is the type of motion of each nonstationary component of the configuration. Using KDD, one can resolve a complex motion into its simple kinematic components and assign a subdomain grid for each component of the motion. For example, a two-component configuration with each component moving in six degrees-of-freedom (DOF), can have a composite of twelve subdomain grids, each of which moving in a single DOF. It should be noted that such a superposition can be done for the kinematics of the problem but not for the nonlinear dynamics.

There are two major differences between the present KDD and HDD [Refs. 2–4]. First, each subdomain grid can be moving. This is accounted for by the time-dependent curvilinear coordinate transformations (Eq. 1), whereby the grid velocities are also determined. The nonlinear equations (Eq. 2) are solved for the nonlinear rigid-body dynamics and the flowfield around it. Secondly, the intergrid information transfer is time dependent. That is, all the searches and the book keeping for the interpolations or other modes of transfers are renewed and updated as functions of time. The accuracy and the efficiency of this process are the important issues in developing KDD.

Since the governing equations (Eq. 2) are written in the space-fixed frame of reference, all the primitive variables are absolute everywhere and at anytime regardless of the subdomain grid in which they are computed. Therefore, they can be transferred from one subdomain grid to the other regardless of the relative motion of these grids. This, however, is not true for the dynamic quantities, such as forces and moments. If the primitive variables in a given subdomain were computed in the local and relative frame of reference of that subdomain (Lagrangian approach), then they would be transferred across the subdomain boundary after subtracting the relative velocity vector between the subdomains. Hence, in the degenerate zonal method, restricting the residual values (Eq. 4) from the finer grid to the coarser grid can only be done for static grids. Restrictions and prolongations in the kinematic zonal method are performed for the absolute primitive variables only (Eqs. 5–7).

When degenerate zonal grids are in relative motion, even every n-th grid line normal to the interface may no longer be contiguous. If the grid motion is modified at every time step in order to ensure this contiguity, a fully conservative transfer of fluxes can be maintained as in the static case. However the grid motion is no longer the desired continuous motion, but a motion described by a

step-function. An alternative method is developed in this study. This method does not restrict the grid displacement to the integer multiples of one spatial step, i.e it allows the displacement (s) to be any fraction of one spatial step. Therefore, the weight factors of the prolongation (Eqs. 5, 6) and the restriction (Eq. 7) keep changing with the motion. Equations of 5, 6, and 7 are used at every time step and they account for the nonuniform grids with non-contiguous lines normal to subdomain interfaces. Depending on the aerodynamic problem, it may be possible to freeze these weight factors for a few time steps in order to save some computer time.

The demonstrative case presented in this paper is for an airfoil engaged in a prescribed motion of pitching and plunging, which is described in the next section. The KDD grid generated for this case is shown in Fig. 4. A stretched O-grid (117×29) is wrapped around the airfoil and overlapped on a fine Cartesian grid (41×41). Then the fine grid is embedded in a coarse Cartesian grid (49×49). The total number of grid points is 7,475. For convenience, the coarse and the fine Cartesian grids and the O-grid are denoted by G_1, G_2, G_3, respectively. The absolute frame of coordinates are fixed on G_1 which is also space fixed. G_2 and G_3 are free to move relative to each other and G_1. Grid G_1 is space-fixed in order to accommodate a conceivably stationary component of a configuration. Also, G_1 is relatively coarser to cover the farfield with fewer cells. Topologies of G_2 and G_3 are nonsimilar. Grid G_2 is needed for two reasons : first, it is a relatively finer grid with cell sizes comparable to those of G_3 as needed by the overlapping algorithm (Ref. 1), hence a fine grid is used only where needed; second, G_2 experiences only the plunge motion and the information transfer between the degenerate-zonally embedded G_2 and G_1 is much easier than the overlapping algorithm.

RESULTS

The kinematic domain decomposition (KDD) is demonstrated through a transonic flow past an airfoil, which experiences a combined motion of pitching and plunging. The computational case is the simulation of a transonic flow past a NACA-0012 airfoil, which experiences a combined motion of sinusoidal pitching and constant-rate plunging. The freestream flow conditions are $M_\infty = 0.6$ and Re=4.8×10^6. This pitching motion is described by the following time function for the angle of attack,

$$\alpha(t) = \alpha_o + \alpha_1 \sin(M_\infty kt) \tag{8}$$

where the amplitudes are α_o=0°, α_1=2.44°. The reduced frequency is k=0.162 radians. The rate of plunge, M_p, is determined as the vertical component of the freestream velocity approaching at an angle of attack (Fig. 5),

$$M_p = M_\infty \sin(\alpha_2) \tag{9}$$

where α_2=4.86°. Therefore, the airfoil is plunged down 3.286 chord lengths during one cycle of the sinusoidal pitching. There is no available experimental data for a flow past an object in such a complex motion. However, the experimental data reported in Ref. 11 can be used for comparison with a small modification. This experimental data is for the sinusoidally pitching motion (Eq. 8) with α_o=4.86°. The effect of constant-rate plunge in the computational case is compensated by the initial amplitude of the angle of attack (4.86°) in the experiment. Therefore, the flowfield of the computational case differs from the flowfield of the experiment by only the plunge-motion-induced flow component.

The initial solution is obtained for the steady flow at α=0° using local time steps and the multigrid convergence acceleration. Then the unsteady equations are solved time accurately for the sinusoidal pitching motion only (Eq. 8) until the limit cycle is reached. Finally, the solution is obtained time accurately for the flow past the airfoil in combined motion of pitching (Eq. 8) and plunging (Eq. 9), where the airfoil plunges from x_3=0 to x_3=6.56C. It also experiences two cycles of pitching in the meantime. One cycle of the motion corresponds to 64.64 time units. Using a time step of 0.01, one cycle is completed in 6,464 time steps. The computer time needed for one cycle is 24.7 minutes on a Cray-2.

The accuracy of KDD in simulating a flow past an oscillating and plunging airfoil is studied by comparing the result with the experimental data. The instantaneous Mach number contours at α=2.11°↑ and x_3=-0.544 C are shown in Fig. 6. The instantaneous effect of the plunge-down at M_p=0.0508 on the flow is somewhat similar to a steady flow at 4.86° angle of attack. The flow expands around the leading edge from the stagnation point to form a supersonic pocket, which is terminated by a shock. The wake starts with a small separation region on the upper surface close to the trailing edge and relatively lower Mach numbers are visible in this region.

Instantaneous surface pressure coefficients at angles of attack 5.95°↑, 6.97°↑, 6.57°↓ and 5.11°↓ are shown in Fig. 7. The comparison is favorable with the experimental data [Ref. 11]. The maximum and the minimum C_p values of all the instants are captured. Small discrepancies are observed in the shock region. Variation of the lift coefficient with the angle of attack is presented in Fig. 8. This curve can be followed in a counterclockwise sense where the pitch-up and the pitch-down are represented by the lower and the upper portions of the curve, respectively. The C_l values for $\alpha\uparrow$ and $\alpha\downarrow$ are not equal to each other. This is due to the boundary-motion-induced flow, which can only be captured by dynamic-body calculations. The computed results agree well with the experimental data. Some discrepancy, however, is detected at higher angles of attack. This may be due to the plunge-motion-induced flow which are not represented in the experiment. Also, contributing to the discrepancy are the numerical errors involved in the temporal and spatial representations of the fluxes across the subdomain grid interfaces.

CONCLUSIONS

A method is developed to simulate computationally an unsteady flow past an object engaged in a complex motion. The method employs a composite of subdomain grids, where each subdomain grid can engage in a different type of motion. Hence, it is called Kinematic Domain Decomposition (KDD) herein. The composite grid can be a hybrid of multiblock, zonal, and overlapped grids. The flow equations are solved using the time dependent generalized coordinates in the absolute frame of reference. The absolute values of the primitive variables are transferred across the grid interfaces.

KDD is demonstrated through a transonic, viscous flow simulation. The results compare favorably with the experimental data. Numerical errors can be further reduced by; (1) a judicious hybridization of KDD, whereby the weaknesses of each method are optimally avoided, (2) decreasing the time steps, (3) refining the grids at the subdomain interfaces, and (4) more frequent updating of the intergrid information.

Simulating the flowfield with the objects in motion, as opposed to a series of solutions where the objects are frozen instantaneously (static), gives more accurate results, captures the boundary-motion-induced flow component, and it is more efficient.

The strength of the KDD method can further be utilized by applying it to a flow past a complex multicomponent configuration with one or more of its components in relative motion with respect to each other.

The current improvements of this KDD include the second-order time integration, extensions for six-degree-of-freedom motion, removal of prescribing the motion, and finally applying to the unsteady interference flows involving the relative motion.

ACKNOWLEDGMENT

This work is supported by the grant NAG-1–1150 from NASA Langley Research Center. The technical monitor is David S. Miller.

REFERENCES

1. Baysal, O., Fouladi, K., Lessard, V. R., "A Multigrid and Upwind Viscous Flow Solver on 3–D Overlapped and Embedded Grids," AIAA Paper No. 89–0464, January 1989. Also, AIAA Journal, Vol. 29, No. 4, April, 1991.
2. Baysal, O., Fouladi, K., Leung, R. W., Sheftic, J. S., "Viscous Analysis of Interference Flows Past Cylinder-Fin-Sting Assemblies Near Cavities," AIAA Paper No. 90–3095 CP, Proceedings of 8th Applied Aerodynamics Conference, August 1990, pp. 884–892. Also, to appear in Journal of Aircraft, 1991.
3. Fouladi, K., Baysal, O., "Viscous Simulation Method for Unsteady Flows Past a Configuration with Nonsimilar Multicomponents," Recent Advances and Applications in CFD (Ed. : O. Baysal), ASME-FED-Vol. 103, Winter Annual Meeting, November 1990, pp.181–190.
4. Fouladi, K., Baysal, O., "Hybrid Domain Decomposition for Configurations with Multiple and Nonsimilar Components," Proceedings of 5th Conference on Domain Decomposition Methods for Partial Differential Equations, SIAM, May 6–8, 1991, Norfolk, VA.
5. Baysal, O., "Navier-Stokes Analysis of Muzzle-Blast-Type Waves," AIAA Journal, Vol. 24, No. 5, May 1986, pp. 800–806.
6. Baysal, O., Yen, G. W., "Kinematic Domain Decomposition to Simulate Flows Past Moving Objects," AIAA Paper No. 91–0725, January 1991.
7. Baysal, O., Yen, G. W., "Implicit and Explicit Computations of Flows Past Cavities With and Without Yaw," AIAA Paper No. 90–0049, January, 1990.
8. Krist, S. L., Thomas, J. L., Sellers, W. L., Kjelgaard, S. O., "An Embedded Grid Formulation Applied to a Delta Wing," AIAA Paper No. 90–0429, January 1990.
9. Steger, J. L., Dougherty, F. C., Benek, J. A., "A Chimera Scheme," Advances in Grid Generation, Ed. : K. Ghia, ASME-FED-Vol. 5, 1983, pp. 59–69.
10. Thomas, J. L., Walters, R. W., Reu, T., Ghaffari, F., Weston, R. P., Luckring, J. M., "A Patched-Grid Algorithm for Complex Configurations Directed Towards the F/A-18 Aircraft," AIAA Paper No. 89–0121, January 1989.
11. Landon, R., "NACA-0012 Oscillatory and Transient Pitching," Compendium of Unsteady Aerodynamic Measurements, AGARD Report No. 702, August 1982, pp. 3.3–3.25.

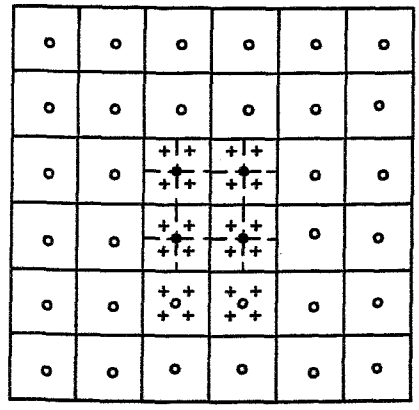

Fig. 1 : A typical stencil for coarse-fine grid communication in degenerate zonal grids.

● } coarse grid cell center
o }

\+ fine grid cell center

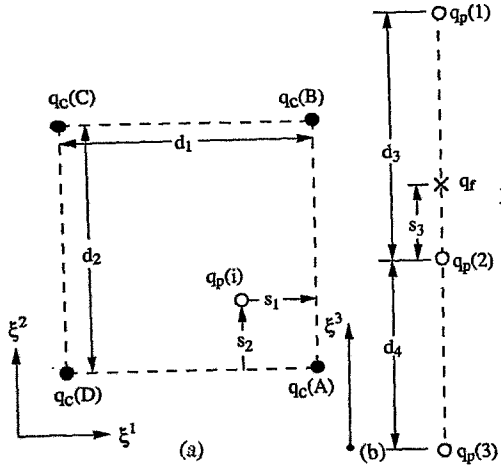

Fig. 2 : A typical stencil for kinematic degenerate zonal grids during communication from the coarse grid to the fine grid boundary (prolongation).
(a) two dimensional planes
● coarse grid cell center (A, B, C, D)
o pseudo-fine cell center
(b) the third dimension
x fine grid cell center
o pseudo-fine cell center (q(1), q(2), q(3))

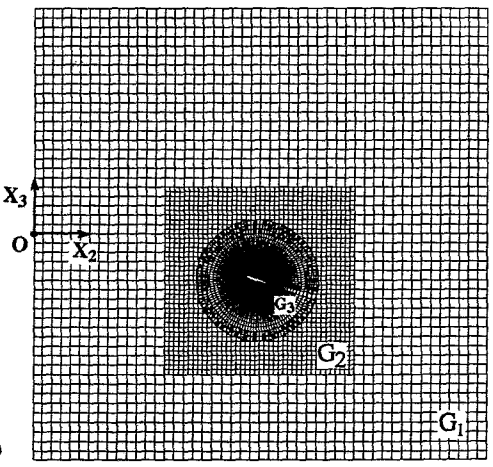

Fig. 4 : Kinematic Domain Decomposition grid for an airfoil moving with two degrees-of-freedom.
G_1 : Global Cartesian grid
G_2 : Fine Cartesian grid zonally embedded in G_1 for translational motion.
G_3 : O-grid overlapped on G_2 for rotational motion.

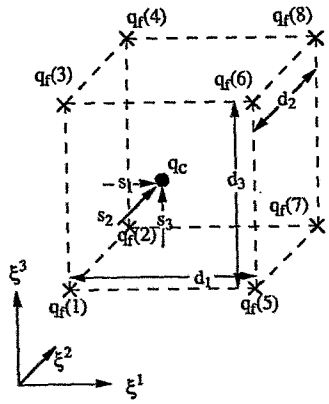

Fig. 3 : A typical stencil for kinematic degenerate zonal grids during communication from the fine grid to the coarse grid (restriction).
● coarse grid cell center
x fine grid cell center

Fig. 5 : Velocity diagram for an airfoil plunging down at a constant rate given by Eq. 9.

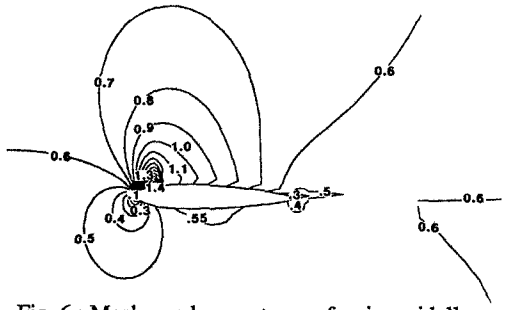

Fig. 6 : Mach number contours of a sinusoidally pitching and constant-rate plunging NACA-0012 airfoil at α=2.11 °and M_p=0.0508 computed on KDD grid.

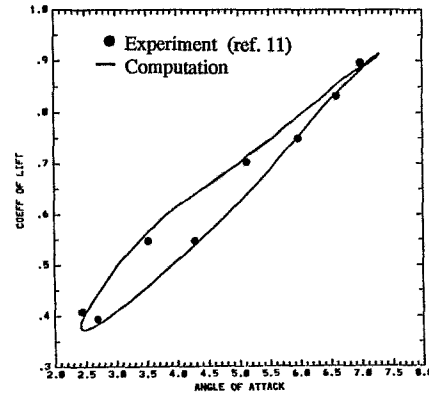

Fig. 8 : Lift coefficients versus the angle-of-attack for one cycle of a sinusoidally pitching NACA-0012 airfoil.

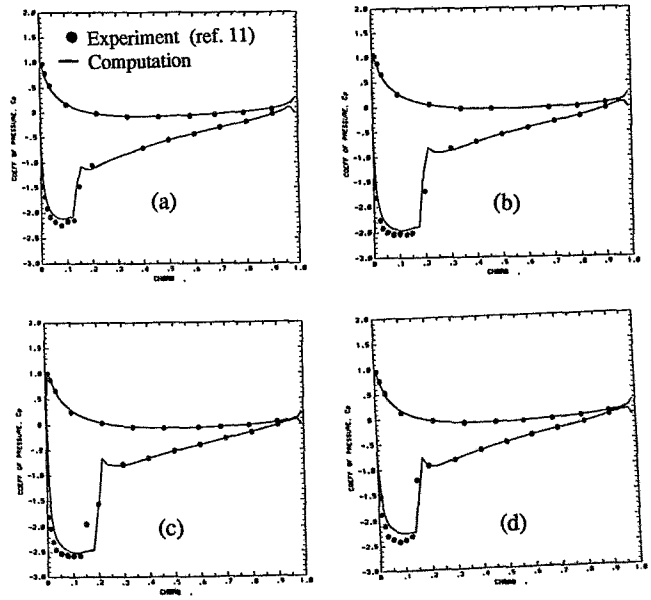

Fig. 7 : Surface pressure coefficients at various angles-of-attack for a sinusoidally pitching NACA-0012 airfoil.
(a) α=5.95 °↑, (b) α=6.97 °↑, (c) α=6.57 °↓,
(d) α=5.11 °↓