AIAA JOURNAL
Vol. 35, No. 8, August 1997

Space-Marching Method on Unstructured Grid for Supersonic Flows with Embedded Subsonic Regions

Kazuhiro Nakahashi* and Eiji Saitoh†
Tohoku University, Sendai 980-77, Japan

A space-marching algorithm on an unstructured grid to solve supersonic flows that may contain embedded subsonic regions is proposed. The method employs a domain-marching algorithm in which a bandlike computational subdomain is marched in the hyperbolic direction starting from the upstream boundary of the flowfield. A masking procedure is employed where the outside of the selected domain is masked during the flux computations to minimize the overall arithmetic operations. The unmasked domain is integrated in time to get a locally converged solution. If embedded subsonic regions are present, the unmasked region is enlarged to cover the entire subsonic regions surrounded by supersonic points. The method is applied to calculations of a fully supersonic flow in a scramjet and a supersonic flow with embedded subsonic regions around a blunt-nose body. It is demonstrated that the computational work can be reduced by as much as a factor of 10 as compared to conventional, time-marching unstructured grid methods.

I. Introduction

An efficient approach for computing supersonic flows is to employ space-marching techniques. They have been used extensively to compute steady, supersonic, inviscid and viscous flows (cf. Refs. 1–3). However, the conventional space-marching methods have basically two difficulties when they are applied to realistic high-speed flight vehicle configurations.

First is the difficulty of generating a grid for complex configurations because the conventional space-marching methods assume the use of a structured grid. Moreover, the marching direction is usually aligned to one of the coordinate lines of the grid, so that the grid must have a flow through, H-grid topology. This situation limits the utilization of the hyperbolic property of supersonic flows for complex three-dimensional configurations.

The second difficulty of the space-marching methods is the treatment of subsonic pockets, which often appear for flows of realistic high-speed-flight vehicle configurations even though they fly at supersonic speeds. For such predominantly supersonic flows, a combination of a time-marching code for subsonic parts and a space-marching code for supersonic regions of the flowfield has been used. However, this approach uses two different codes so that it substantially decreases the advantage in the efficiency of space-marching methods. Chakravathy and Szema[4] proposed a unified method on structured grids to treat supersonic flows with subsonic pockets. The method is based on the unsteady Euler equations and used a Gauss–Seidel relaxation method in the marching direction to realize a unified approach for space- and time-marching methods.

The aim of this paper is to develop a new space-marching method that can solve supersonic flows with subsonic pockets about complex and realistic flight vehicle configurations. To remove the aforementioned difficulties of conventional space-marching techniques, the new space-marching method is developed on fully unstructured grids. McGrory et al.[5] also developed a space-marching method on an unstructured grid. However, they retained a structure of the grid in the marching direction so that the method was not a fully unstructured one.

Unstructured grids allow easy adaptations to complex configurations. Moreover, with a suitable algorithm, a space-marching method on an unstructured grid can be developed whose marching direction is independent to the grid. To allow the existence of subsonic regions, the new approach proposed is based on the Euler equations that retain the time-dependent terms, and a computational subdomain instead of the computational front is marched in the hyperbolic direction. This domain marching is a simple and versatile approach for realizing a space-marching method on an arbitrary unstructured grid.

The remainder of the paper is organized as follows. Section II describes the governing equations and the basic solution algorithm on unstructured grids. A basic strategy of the space-marching procedure is at first outlined in Sec. III. Then a masking method for computations of a selective region and a subsonic pocket treatment are summarized. Section IV shows two computed results and discusses the efficiency improvements gained by the present marching method. Finally, some conclusions are drawn and further extensions are considered in Sec. V.

II. Basic Solution Algorithm

Governing Equations

The present method uses the Euler equations that retain the unsteady form. The two-dimensional, compressible Euler equations can be expressed in an integral form as

$$\frac{\partial}{\partial t}\int_{\Omega} \boldsymbol{Q}\,\mathrm{d}S + \oint_{\partial\Omega} \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{n})\,\mathrm{d}l = 0 \tag{1}$$

where $\boldsymbol{Q} = [\rho, \rho u, \rho v, e]^T$ is the vector of conserved variables, ρ is the density, u and v are the velocity components in the x and y directions, and e is the total energy. The vector $\boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{n})$ represents the inviscid flux vector of mass, the x and y momentums, and the energy quantity out of the control volume Ω, through the boundary $\partial\Omega$ when $\boldsymbol{n}$ is the outward pointing normal of $\partial\Omega$. This system of equations is closed by the perfect gas equation of state

$$p = (\gamma - 1)\left[e - \tfrac{1}{2}\rho(u^2 + v^2)\right] \tag{2}$$

where p is the pressure and γ is the ratio of the specific heats.

Time-Dependent Solution Algorithm

The variables $\boldsymbol{Q}$ are stored at the vertices of a triangular mesh. The control volume Ω_j at mesh j employed here is a median dual that is constructed by lines connecting center of triangular cell and midpoint of each edge (Fig. 1). The control volume boundary is denoted by $\partial\Omega_j$ and its normal vectors $\boldsymbol{n}$.

With an integral cell average

$$\bar{\boldsymbol{Q}}_j = \frac{1}{S_j}\int_{\Omega_j} \boldsymbol{Q}\,\mathrm{d}S$$

Received Jan. 10, 1996; presented as Paper 96-0418 at the AIAA 34th Aerospace Sciences Meeting, Reno, NV, Jan. 15–18, 1997; revision received May 2, 1997; accepted for publication May 2, 1997.

*Professor, Department of Aeronautics and Space Engineering. Associate Fellow AIAA.
†Graduate Student, Department of Aeronautics and Space Engineering.

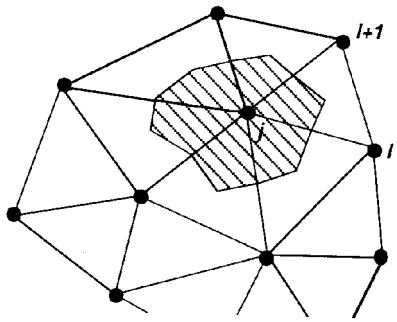


Fig. 1 Control volume.

where

$$S_j = \int_{\Omega_j} \mathrm{d}S$$

is an area of the control volume Ω_j, the first term of Eq. (1) becomes a time derivative of the integral cell average $\bar{\boldsymbol{Q}}_j$. Then, writing the second term of Eq. (1) in algebraic form, we can get

$$\frac{\mathrm{d}\bar{\boldsymbol{Q}}}{\mathrm{d}t} = -\frac{1}{S_j}\sum \Delta l_{ji}\, \boldsymbol{h}(\boldsymbol{Q}^+_{ji}, \boldsymbol{Q}^-_{ji}, \boldsymbol{n}_{ji}) \tag{3}$$

where Δl_{ji} is a segment line of the control volume boundary associated with edge connecting points j and i. This segment line Δl_{ji}, as well as its unit normal $\boldsymbol{n}_{ji}$, can be computed by summing up the segment vectors of two segments of triangular cells having this edge. The term $\boldsymbol{h}$ is a numerical flux vector normal to the control volume boundary, and $\boldsymbol{Q}^{\pm}_{ji}$ is values on both sides of the control volume boundary.

The basic solution scheme on the unstructured grid employed here is based on the scheme proposed by Barth and Jespersen.[6] If we evaluate $\boldsymbol{Q}^{\pm}_{ji}$ in $\boldsymbol{h}$ at both endpoints of the edge, we get the first-order scheme. Using the Roe's numerical flux functions[7] for $\boldsymbol{h}$, we can write the first-order evaluation for a control volume $\boldsymbol{Q}^{\pm}_{ji}$ as

$$\oint_{\partial\Omega_j} \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{n})\,\mathrm{d}S \approx \sum \Delta S_{ji}\left\{\frac{1}{2}[\boldsymbol{F}(\bar{\boldsymbol{Q}}_i, \boldsymbol{n}_{ji}) + \boldsymbol{F}(\bar{\boldsymbol{Q}}_j, \boldsymbol{n}_{ji})] - \frac{1}{2}\left|\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}\right|_{ji}(\bar{\boldsymbol{Q}}_i - \bar{\boldsymbol{Q}}_j)\right\} \tag{4}$$

where $\sum_j$ is treated for a control volume, but this is computed using an edge by edge loop in the coding.

The second-order spatial accuracy can be realized by assuming a distribution of $\boldsymbol{Q}$ using the cell average $\bar{\boldsymbol{Q}}$ in the control volume. For this reconstruction, gradients of physical values are evaluated at each node point. Then, they are used for extrapolations to get $\boldsymbol{Q}^{\pm}$ at the interface. If we write the gradient of the primitive variables at node j as $\nabla \boldsymbol{q}_j$, then at the control volume interface whose position vector is $\boldsymbol{r}_m$ on an edge ji, $\boldsymbol{q}^{\pm}_{ji}$ are given as

$$\boldsymbol{q}^-_{ji} = \bar{\boldsymbol{q}}_j + \Phi_j \nabla \boldsymbol{q}_j \cdot (\boldsymbol{r}_m - \boldsymbol{r}_j) \tag{5a}$$

$$\boldsymbol{q}^+_{ji} = \bar{\boldsymbol{q}}_i + \Phi_i \nabla \boldsymbol{q}_i \cdot (\boldsymbol{r}_m - \boldsymbol{r}_i) \tag{5b}$$

The gradient $\nabla \boldsymbol{q}_j$ at nodal point j is evaluated using a Green–Gauss formula

$$\nabla \boldsymbol{q}_j = \frac{1}{S_j}\sum \frac{1}{2}(\boldsymbol{q}_j + \boldsymbol{q}_i)\Delta l_{ji}\boldsymbol{n}_{ji} \tag{6}$$

In Eq. (5), Φ is a limiter[8] to make the scheme monotonic.

Equation (3) is integrated in time using a two-step, explicit time-integration scheme with a local-time stepping.

III. Marching Algorithm

Basic Strategy

In contrast to conventional space-marching techniques where a computational front is marched downward, the present method marches a selected domain to which a time-marching computation is applied. There are two reasons to use the domain marching. First, the time derivatives must be retained to allow the existence of subsonic regions in the flowfield. The second reason is to remove a stability limit of the marching step size. Most of the conventional space-marching methods employ an implicit integration in the marching direction for removing the stability limit. Here, for an unstructured grid, however, it is difficult to construct such an implicit integration algorithm because of the fully unstructured connections between node points.

Instead, a time-integrational algorithm is applied to the domain. Although this requires time iterations to get a converged solution of the region, the number of iterations is much smaller than that required for entire flow computations. Therefore, the domain marching is a versatile choice for realizing a space-marching method on an arbitrary unstructured grid.

The schematic explanation of the marching algorithm is shown in Fig. 2, where the selected domain in the flowfield is called as an active domain. It is a laterally expanded, bandlike computational region bounded by two parallel lines that are approximately normal to the freestream direction. At each marching step, a time-dependent computation is applied to the active domain. For supersonic flow problems, the active domain is marched downward starting from the upstream boundary, and one pass marching solves the entire flowfield. If the width of the active domain is enlarged to cover the entire flowfield, the method reduces to a conventional time-dependent flow solver.

During the time-marching computations in the active domain, the residual is monitored only in the residual-monitor subregion that is usually upstream one-fifth of the active domain. If the residual in this subdomain decreases below a threshold, this subdomain is expelled from the active domain to the upstream computed field and a new subdomain is included from the downstream uncomputed field to

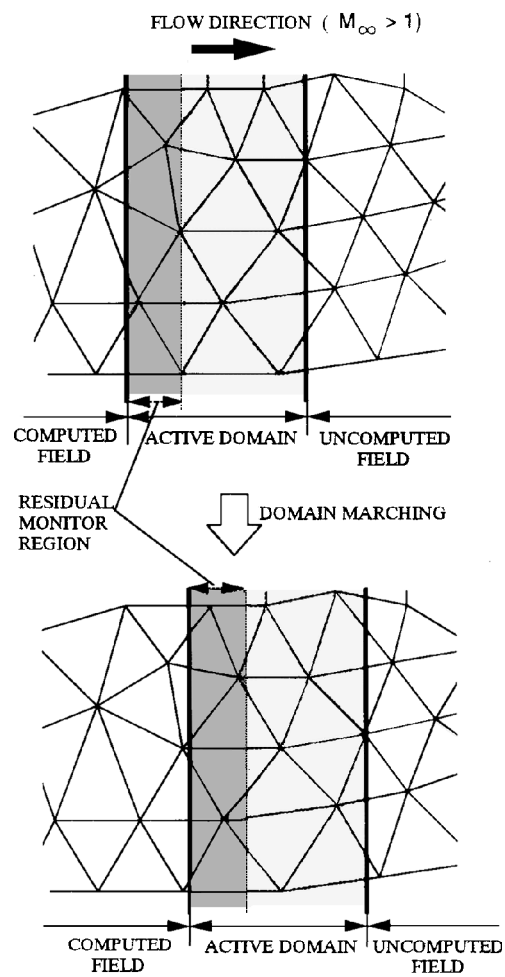


Fig. 2 Marching of active domain.

the active domain, as shown in Fig. 2. Therefore, every spatially marching step has an overlapped region in the active domain. This overlapping is required as a buffer for removing the limitation of the marching step size for various flow conditions on unstructured grids.

The width of the active domain is determined basically by input data, but the choice of the width does not affect the computational efficiency very much. Starting from the upstream boundary, each active domain is determined by, e.g., $x_1 < x < x_1 + \Delta s_m$, if the main flow direction is in the x direction and Δs_m is a width of the active domain. The residual monitor subregion is determined by $x_1 < x < x_1 + a\Delta s_m$, and $a = 0.1$–0.2.

Because of the unstructured grid, the marching direction can be determined arbitrarily. Although it can be determined step by step using a locally averaged flow angle, here it is basically determined by the upstream uniform flow direction for simplicity. For an extension of the method to complex three-dimensional flow problems, it is important to keep the simplicity of the algorithm.

Masking

For a selected computation in the entire flowfield, all node points and edges have flags by arrays of MASKN(in) and MASKE(ie), respectively. These arrays have values depending on locations and connections of nodes (Fig. 3), where MASKN 0 is the node in the downstream, uncomputed field, MASKN 1 the node located in the uncomputed region but with connections to nodes in the active domain, MASKN 2 the node in the active domain, MASKN 3 the node of MASKN 2 but with connections to nodes of MASKN 4, MASKN 4 the node in the residual-monitor subregion of the active domain, MASKN 5 the node in the upstream computed field having connections with nodes in the active domain, and MASKN 6 the node in the upstream computed field.

Each edge has a flag of MASKE(ie). If both end nodes of an edge have the same values of MASKN of 0 or 6, MASKE(ie) of the edge is assigned to 0, otherwise edges have MASKE(ie) = 1.

The flux computations of Eq. (4) are performed on the edges of MASKE(ie) = 1 and other edges are skipped. Those computed fluxes are summed to get new values of $\boldsymbol{Q}$ at nodes of MASKN(in) = 2–4. For the second-order scheme, gradients of flow variables are computed for nodes of MASKN(in) = 2–5.

Subsonic Pockets

An important advantage of the active domain marching is that the domain can include subsonic regions in it. If subsonic regions appear during the time-marching computations, the boundaries of residual-monitor subregion are shifted so as to cover the entire subsonic regions by supersonic points.

There are three situations for the boundary shift. The first is shown in Fig. 4a, where Mach numbers of any points of MASKN = 4, which have connections to points of MASKN = 5, become less than 1.001. For this case, the upstream boundary of the residual-monitor subregion, which is the upstream boundary of the active domain as well, is shifted upward (Fig. 4a).

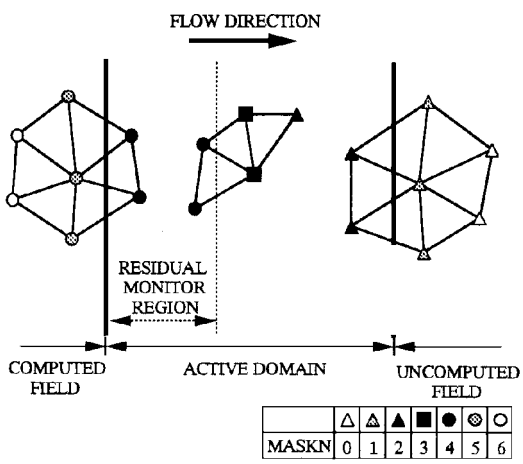


Fig. 3 Masking of nodes.

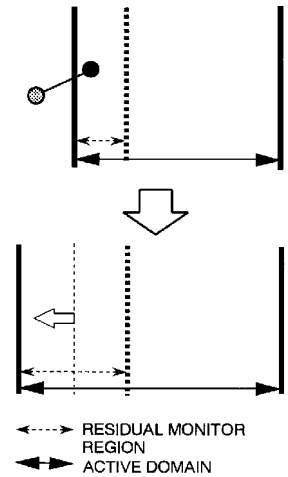


Fig. 4a Upstream boundary shift of residual-monitor subregion.

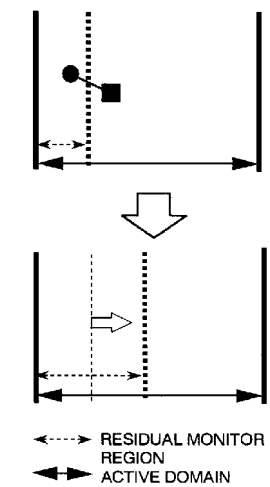


Fig. 4b Downstream boundary shift of residual-monitor subregion.

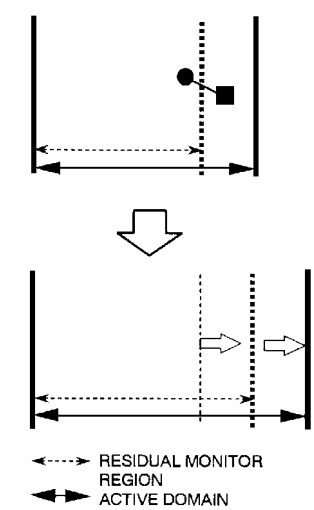


Fig. 4c Downstream boundary shift of active domain.

Figure 4b shows a situation where Mach numbers of any points of MASKN = 4 that have connections to points of MASKN = 3 become less than 1.001. For this case, the downstream boundary of the residual-monitor subregion is shifted downward. If the boundary reaches the downstream boundary of the active domain, the active domain boundary is also shifted downward the same amount so as to keep a buffer region downstream of the residual-monitor subregion (Fig. 4c).

These boundary shifts are repeated until all points of MASKN = 4 with connections outside of the subregion become supersonic points. If the time-dependent computations of the residual-monitor subregion converge, the whole subregion is expelled to the computed region. This means that the downstream boundary of the residual-monitor subregion becomes the upstream boundary of the new active domain. The downstream boundary is determined by the prescribed marching step width Δs_m.

The computation is terminated if the upstream boundary of the active domain reaches the downstream boundary of the flowfield.

IV. Applications

The method just described has been tested on two cases, which were chosen so as to show the flexibility and efficiency of the present marching method. One is a pure supersonic flow through a duct, and the other is an external flow that has embedded subsonic regions in it.

Supersonic Duct Flow

The method is applied to a pure supersonic flow inside of a duct modeling a half-plane of a scramjet intake whose unstructured grid is shown in Fig. 5. The computed Mach contours by the second-order method at selected marching steps are shown in Fig. 6 for an inflow Mach number of 3. The computed result was validated by comparing with the result obtained by the conventional time-marching method.

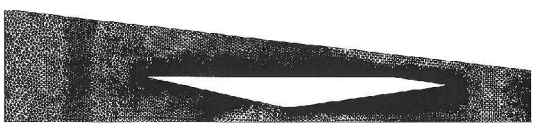
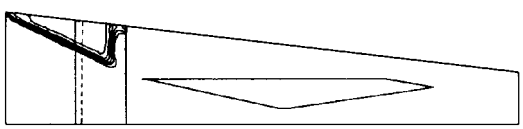
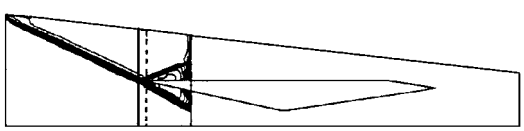


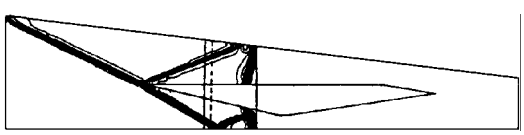
Fig. 5 Grid inside a supersonic duct.



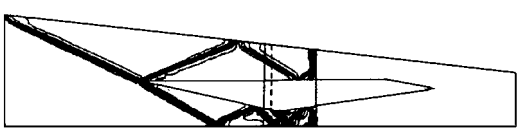
a) 10 steps



b) 20 steps

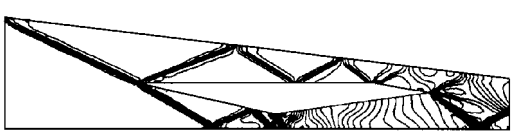


c) 30 steps



d) 40 steps

e) 60 steps



f) 80 steps

Fig. 6 Computed Mach contours, M_{in} = 3.

Table 1 First-order CPU times, supersonic duct flow

No. of total marching steps	CPU times, s	CPU time ratios
1	109.4	1.00
51	17.2	0.157
81	16.3	0.149

Table 2 Second-order CPU times, supersonic duct flow

No. of total marching steps	CPU times, s	CPU time ratios
1	672.8	1.00
51	63.3	0.094
81	73.6	0.109

The marching direction is parallel to the duct centerline, and the active domain width for this case is one-tenth of the duct length. The width of the residual monitor subregion is one-eighth of the active domain so that in total 80 marching-steps were required to compute the entire flowfield. In Fig. 6, boundaries of active domains are also shown, and the dotted lines are the downstream boundaries of the residual-monitor subregions. The dense contour lines appearing at the downstream boundary of the active domain are due to the difference of the nodal values between the computed and uncomputed regions.

Tables 1 and 2 show the efficiency improvements by the method for this case. The computations were performed using a single Cray C-90 processor. The first columns in the tables are for computing the entire flowfield at once, namely, using a conventional time-dependent method. The second columns are cases in which the active domain is one-tenth the width of the duct length and the residual-monitor subregion is one-fifth the width of the active domain. The third is a case (Fig. 6) with a narrower monitor subregion one-eighth of the active domain. Computed results of these three cases with different marching steps did not show any noticeable difference in the contour lines.

As shown in the tables, the computational work is reduced by as much as a factor of 10 as compared to conventional, time-marching methods. It is also shown that the efficiency improvement is not sensitive to the selection of the widths of the active domain and residual-monitor subregion. The second-order method shows better improvements as compared to the first-order scheme. This means that the present method becomes more effective for problems that require larger arithmetic operations per point.

Double Ellipse

The second result is for a supersonic flow around a blunt-nose body defined by two ellipses.[9] This flowfield has embedded subsonic regions near the nose and the intersecting point of two ellipses.

The grid is shown in Fig. 7, and the computed Mach contours at selected marching steps are shown in Fig. 8 for a freestream Mach number 8.15 and 30-deg angle of attack. The marching direction of this case has an inclination of 30 deg, which is equal to the freestream direction. The active domain width is a half of the downstream body height. The width of the residual-monitor subregion is one-fifth of the active domain width for this case. Dotted lines in the Mach contours are sonic lines. As shown in the final results of Fig. 8f, the subsonic region near the nose region is relatively large due to the two-dimensional computation. The computed results with different marching steps as well as the standard non-space-marching result did not show any noticeable difference in the contour lines.

Figure 8a is the beginning, when the residual-monitor subregion has subsonic points in it. The downstream boundary of the residual-monitor subregion was then shifted downward (Fig. 8b) so as to

cover the subsonic points by supersonic points. The active domain boundary was also sifted to keep a buffer region. Figure 8c is the next marching step, where the previous downstream boundary of the residual-monitor subregion became the new upstream boundary of the active domain.

Efficiency improvements obtained by the present method are shown in Tables 3 and 4. The widths of the residual monitor subregion are two-fifths of the active domain for a case of 27 marching steps, one-fifth for 52 steps, and one-sixth for 64 steps. The improvement in the second-order scheme for this double ellipse flow is a factor of 3 smaller than the preceding pure supersonic case. This is mainly due to the relatively large subsonic region at the nose, where a large part of the total CPU time was consumed. Another reason of this efficiency reduction comes from the rectangle computational field whose vertical length is larger than the streamwise length. If the computational region is set up to cover only for near and downstream of the bow shock wave, the efficiency will be improved some more.

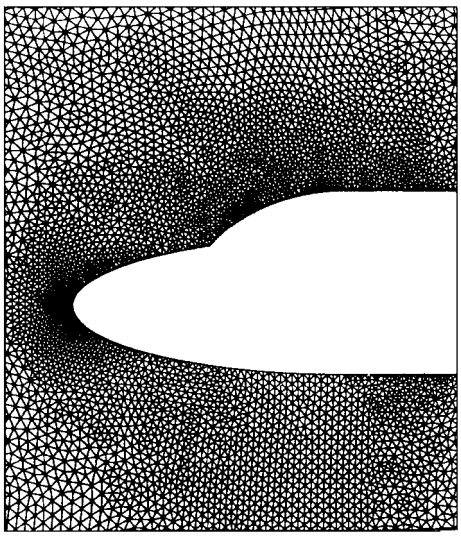


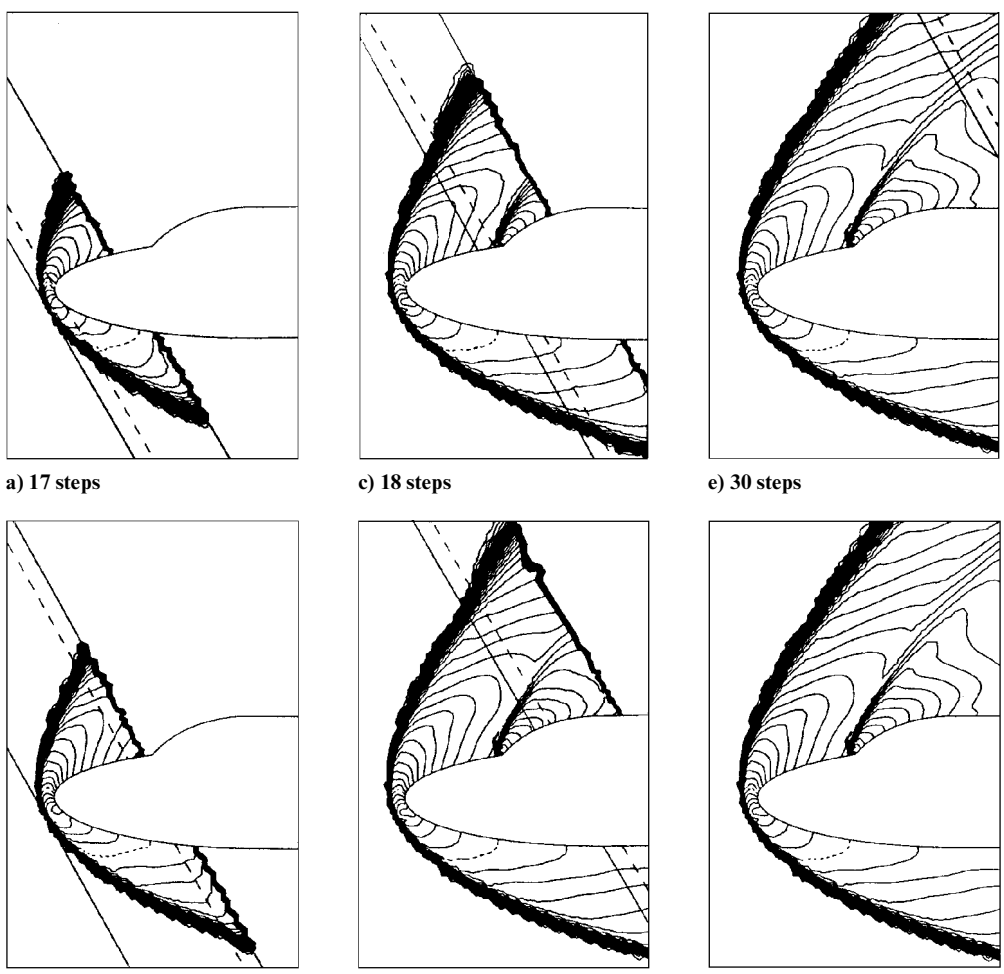
Fig. 7 Grid of a double ellipse.

Table 3 First-order CPU times, double ellipse

No. of total marching steps	CPU times, s	CPU time ratios
1	74.2	1.00
27	35.4	0.477
52	34.0	0.458
64	35.2	0.474

Table 4 Second-order CPU times, double ellipse

No. of total marching steps	CPU times, s	CPU time ratios
1	976.4	1.00
27	304.0	0.311
52	306.6	0.314
64	336.0	0.343



a) 17 steps **c) 18 steps** **e) 30 steps**

b) 17 steps **d) 22 steps** **f) 52 steps**

Fig. 8 Computed Mach contours, $M_\infty = 8.15$ and $\alpha = 30$ deg.

V. Concluding Remarks

An efficient and flexible space-marching algorithm using unstructured grids has been developed to solve supersonic flows that may contain embedded subsonic regions. Utilizing the unstructured connections of grid points, the method has several important advantages over the conventional methods. First, the scheme can treat a supersonic flow having embedded subsonic regions. Moreover, the flexibility of unstructured grids allows easy adaptations to complex configurations. The marching algorithm is simple but the improvement in the computational efficiency is as much as a factor of 10 as compared to conventional, time-marching unstructured grid methods.

An extension of the method to three dimensions is straightforward. The method also will be extendable to high Reynolds number viscous flows by proper treatment of the subsonic region in the boundary layer. The method is expected to be more effective for flows that require large arithmetic operations per point, such as hypersonic reacting flows.

Acknowledgment

The computations were carried out on the Cray C-90 of the Institute of Fluid Science at Tohoku University, Sendai, Japan.

References

[1]Kutler, P., Reinhardt, W. A., and Warming, R. F., "Multishocked, Three-Dimensional Supersonic Flowfields with Real Gas Effects," *AIAA Journal*, Vol. 11, No. 5, 1973, pp. 657–664.

[2]Schiff, L. B., and Steger, J. L., "Numerical Simulation of Steady Supersonic Viscous Flow," AIAA Paper 79-0130, Jan. 1979.

[3]Lawrence, S. L., Chaussee, D. S., and Tannehill, J. C., "Development of a Three-Dimensional Upwind Parabolized Navier–Stokes Code," *AIAA Journal*, Vol. 28, No. 6, 1991, pp. 971, 972.

[4]Chakravathy, S. R., and Szema, K. Y., "Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets," *Journal of Aircraft*, Vol. 24, No. 2, 1987, pp. 73–83.

[5]McGrory, W. D., Walters, R. W., and Lohner, R., "Three-Dimensional Space-Marching Algorithm on Unstructured Grids," *AIAA Journal*, Vol. 29, No. 11, 1991, pp. 1844–1849.

[6]Barth, T. J., and Jespersen, D. C., "The Design and Applications of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, Jan. 1989.

[7]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[8]Barth, T. J., "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier–Stokes Equations," *Special Course on Unstructured Grid Methods for Advection Dominated Flows*, AGARD Rept. 787, May 1992, pp. 6–61.

[9]Desideri, J.-A., Glowinski, R., and Periaux, J., *Hypersonic Flows for Reentry Problems*, Vol. 2, Springer–Verlag, Berlin, 1991, pp. 17, 18.

D. S. McRae
Associate Editor