A New Convergence Acceleration Technique for Scramjet Flowfields[1]

Bernard Parent and In-Seuck Jeung
Dept. of Aerospace Engineering, Seoul National University
Seoul 151-744, Korea
bernard@snu.ac.kr, enjis@snu.ac.kr

Keywords: Convergence acceleration, Turbulent hypersonic flows, FANS equations, scramjet flowfields.

Abstract

This paper outlines a new convergence acceleration designed to solve scramjet flowfields with zones of recirculation. Named the "marching-window", the algorithm consists of performing pseudo-time iterations on a minimal width subdomain composed of a sequence of cross-stream planes of nodes. The upstream boundary of the subdomain is positioned such that all nodes upstream exhibit a residual smaller than the user-specified convergence threshold. The advancement of the downstream boundary follows the advancement of the upstream boundary, except in zones of significant streamwise ellipticity where a streamwise ellipticity sensor ensures its continuous progress. Compared to the standard pseudo-time marching approach, the marching-window is here seen to decrease the work required for convergence by up to 24 times for supersonic flows with little streamwise ellipticity and by up to 8 times for supersonic flows with large streamwise separated regions. The memory requirements are observed to be reduced sixfold by not allocating memory to the nodes not included in the computational subdomain. The marching-window satisfies the same convergence criterion as the standard pseudo-time stepping methods, hence resulting in the same converged solution within the tolerance of the user-specified convergence threshold. The extension of the marching-window to the weakly-ionized Navier-Stokes equations is also discussed.

Introduction

There is little doubt that the most efficient way to solve supersonic or hypersonic flow with no streamwise ellipticity is through a space-marching method, as numerous extremely efficient marching methods developed over the years can attest (see for example Refs. 1–5). The Navier-Stokes equations at supersonic speeds do, however, exhibit some ellipticity in the marching direction through the streamwise viscous terms and the subsonic layer of the boundary layer, and it is necessary for a space-marching method to ignore these mechanisms by solving a reduced set of the original equations of motion, such as the parabolized Navier-Stokes equations (PNS). The PNS are defined here as the equation set obtained from the Navier-Stokes equations by neglecting all viscous terms in the streamwise direction and by modifying the streamwise momentum equation to prevent any pressure disturbance to travel upstream, using characteristics splitting or pressure splitting as suggested by Vigneron *et al.*[1] The applicability of the space-marching methods is limited to flows with negligible streamwise ellipticity, hence preventing their deployment to many practical flowfields.

The need to tackle streamwise ellipticity prompted the development of the "global iteration" space-marching methods in which a sweep is performed several times on the entire computational domain to permit the upstream propagation of information (see Ref. 6 for a detailed review). Such are characterized, compared to the pseudo-time marching schemes, by a smaller memory requirement due to the storage of temporary variables in one marching plane only and by enhanced wave propagation mechanism in the streamwise direction. The reduced Navier-Stokes (RNS) equations, derived from the Navier-Stokes equations by ignoring all streamwise diffusion terms but not altering the momentum convection terms, are usually solved in this manner leading to fast convergence of subsonic / supersonic streamwise unseparated flows[7–9] and even of viscous/inviscid interactions creating streamwise separation.[10,11] In a similar vein, Bardina[12] shows that significant reduction in work is achievable by the use of global marching sweeps to solve the full Navier-Stokes equations for high-speed flows. However, if not limited to some predetermined zones of the computational domain, the global iteration approach loses some performance when solving large reverse flow regions, as the number of sweeps can become excessive due to its dependence on the size of the separation bubble. Further, some computing might be inefficiently allocated to the nodes downstream of the separation bubble, *prior* to its convergence. These deficiencies can be remedied by using a space-marching scheme solving the PNS equations until an elliptic/reverse flow region is encountered, then switching to a global iteration RNS method for the length of the elliptic region, iterating until convergence is reached, and pursuing with the marching PNS scheme (see Miller *et al.*[6] for instance). However, such a strategy forces the solution of the PNS equations in certain regions of the flowfield for which the PNS assumption might induce appreciable errors.

The accuracy of the final solution is hence strongly dependent on the ability of the method at predicting correctly which regions of the flowfield can be accurately predicted with the PNS equations, and which regions require the use of the RNS equations.

Recently, a novel approach at solving inviscid supersonic flow with embedded subsonic regions has been proposed.[13] The method, named "active-domain", consists of performing pseudo-time iterations on a small band-like computational domain that advances in the streamwise direction every time the residual of the active-domain near the upstream boundary falls below a user-defined threshold. Using sensors based on the streamwise component of the Mach number, the active-domain boundaries automatically surround any locally subsonic region, on which sufficient iterations are performed to reach steady-state. When the residual inside the subsonic region decreases below the user-defined threshold, the active-domain advances past the subsonic region further downstream. By marching in the streamwise direction, the active-domain results in a decrease in work of up to 10 times compared to standard pseudo-time marching methods for several inviscid problems. However, the ability of the active-domain at solving accurately a streamwise elliptic region is limited by the accuracy of the sensor responsible for the upstream movement of the upstream boundary of the active-domain. Extension of the active-domain method to viscous flow is hampered by the difficulty of formulating a streamwise ellipticity sensor that captures all significant upstream propagating waves while restricting the size of the active-domain to a minimum. Success has been reported in solving viscous flow without streamwise separation by maintaining the active-domain width equal to the height of the boundary layer.[14] However, to the authors' knowledge, the active-domain method has not yet been extended to streamwise separated flows.

This paper proposes an alternate form of the active-domain method, named the "marching-window" algorithm, that is applicable to streamwise separated flows. Similarly to the active-domain, the marching-window performs localized pseudo-time stepping on a subdomain composed of a sequence of cross-stream planes of nodes. The width of the marching-window decreases to only a few planes in regions of quasi-hyperbolic flow and increases to the size of the streamwise-elliptic region when encountered. However, in contrast to the active-domain algorithm, the marching-window is strictly a convergence acceleration technique as it guarantees the residual of all nodes to be below the user-defined threshold when convergence is attained, and hence does not modify the final solution as obtained with the standard pseudo-time stepping schemes. This is accomplished by keeping the residual upstream of the marching window subdomain updated at all times, and by positioning the upstream boundary such that the residual of all nodes upstream is below the user-defined threshold. This results in an algorithm that captures all upstream propagating waves affecting the residual significantly. The upstream propagating waves can originate from (but are not necessarily limited to) large subsonic pockets, streamwise separation, streamwise viscous fluxes, or the flux limiters in the streamwise convection flux derivative, for instance. Further, to enhance the performance of the algorithm, a sensor based on the Vigneron splitting[1] is developed to advance the downstream boundary when significant streamwise ellipticity is detected.

Several numerical experiments are presented including the inviscid solution of a supersonic inlet with a blunt leading edge and a turbulent flowfield including shock boundary layer interactions with considerable streamwise flow separation solved with the Favre-averaged Navier-Stokes (FANS) equations closed by the $k\omega$ turbulence model of Wilcox.[15] A comparison between the marching-window cycle and the standard pseudo-time marching cycle is made on the basis of CPU time, effective iterations, and storage.

Governing Equations

The Favre-averaged Navier Stokes equations closed by the $k\omega$ turbulence model of Wilcox[15] are here expressed in generalized coordinates in strong conservation form[16,17] as $\partial Q/\partial\tau = -R$ with the residual

$$R = \sum_{i=1}^{d}\left[\frac{\partial F_i}{\partial X_i} - \sum_{j=1}^{d}\frac{\partial}{\partial X_i}\left(K_{ij}\frac{\partial G}{\partial X_j}\right)\right] - S\,, \quad (1)$$

of which a minimization is sought. In the latter, X_i stands for the curvilinear coordinate in the ith dimension, while τ refers to pseudotime. Due to the nonlinearity of the equations, the fictitious unsteady term $\partial Q/\partial\tau$ is necessary to obtain the right physical root by marching in pseudotime. The conservative variable vector Q, convective flux vector F_i, diffusion matrix K_{ij}, and diffusion term G are the same as those outlined in Ref. 18. The total energy E and effective pressure P^* include molecular and turbulent properties, $E = e + k + \frac{1}{2}q^2$ and $P^* = P + \frac{2}{3}\rho k$, with e being the internal energy, q the flow speed corresponding to $q = \left(\sum_{i=1}^{d} v_i^2\right)^{1/2}$, v_i the velocity component in the Cartesian x_i direction, P the static pressure, and k the turbulence kinetic energy. The pressure P is found through the ideal gas law from the temperature and the density. The source terms include the baseline source terms of the Wilcox 1988 $k\omega$ model[15] as well as some additional source terms needed to account for turbulence compressibility effects.[19]

Discretization and Pseudotime Integration

The fluid flow equations are discretized using second order accurate finite-difference central stencils except for the convection derivative, which is discretized using the Yee-Roe[20,21] flux-limited method. The entropy

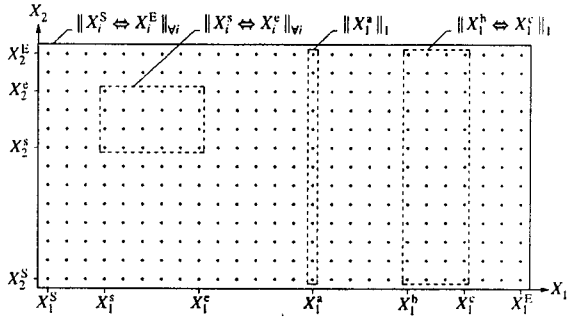

Fig. 1: Example of computational domain and subdomain notation in two dimensions; the computational domain limits are denoted by the superscripts E and S.

correction term recommended in Ref. 20 is used for some of the test cases shown herein. For more details on the eigenstructure of the fluid flow equations, details regarding the discretization stencils, the definition of $\widehat{X}_i$, and the block-implicit approximate factorization algorithm used to advance the solution in pseudo-time, the reader is invited to consult Ref. 18. The local pseudotime step is defined such that its value lies in-between the value indicated by the minimum and the maximum CFL conditions:

$$\Delta\tau = \mathrm{CFL}\max_{i=1}^{d}\left(\frac{1}{|V_i|+a\widehat{X}_i}\right)^{\sigma}\min_{i=1}^{d}\left(\frac{1}{|V_i|+a\widehat{X}_i}\right)^{1-\sigma}, \tag{2}$$

where a σ varying between 0 and 1 induces a time step of a magnitude situated respectively between a minimum and a maximum CFL based time step. Unless otherwise indicated, σ is set to 0.5. Convergence is attained when

$$\xi \leq \xi_{\mathrm{verge}}\ \forall \text{ inner nodes}, \tag{3}$$

with ξ a convergence criterion based on the maximum between the discretized residuals of the continuity and energy equations:

$$\xi \equiv J\max\left(\frac{\sum_{k=1}^{\mathrm{ns}}|R_\Delta^{\rho c_k}|}{\rho}, \frac{|R_\Delta^{\rho E}|}{\rho E}\right). \tag{4}$$

Domain Decomposition Algorithms

While domain decomposition is generally used for parallel computing purposes or used to enable the implementation of different discretization/integration methods in different subdomains, it is utilized here as a means to accelerate the convergence of quasi-hyperbolic systems. We define as quasi-hyperbolic a system of equations 1) which is elliptic, 2) where some of the terms, but not all, can be regrouped to form a hyperbolic set of equations, and 3) whose solution is very close to the solution of the hyperbolic set of terms. For instance, the steady-state Navier-Stokes equations in the hypersonic regime away from the surfaces would exhibit a weak influence of the diffusion terms (responsible for the ellipticity of the system) on the solution compared to the convection terms (the hyperbolic set) and would hence be classified as quasi-hyperbolic. Similarly, a quasi-parabolic system is defined as a system of equations 1) which is elliptic, 2) where some of the terms, but not all, can be regrouped to form a set of parabolic equations, and 3) whose solution is very close to the solution of the parabolic set of terms. The Favre averaged Navier-Stokes equations closed by the $k\omega$ model solved at steady-state over a turbulent flat plate would be termed quasi-parabolic, as the streamwise diffusion terms and the upstream component of the convection terms play a negligible role compared to the other terms.

The acceleration techniques presented in this paper are aimed at reducing the work needed to solve quasi-hyperbolic or quasi-parabolic systems through the use of domain decomposition. Nonetheless, the effectiveness of the methods is not limited to entirely quasi-hyperbolic/parabolic systems and extends to systems where some regions are quasi-hyperbolic/parabolic and others strictly elliptic. It is emphasized that domain decomposition is used here solely as a convergence acceleration technique, and *does not* modify the discretized residual, the time stepping schemes, and the convergence criterion.

Identifying the limits of the computational domain by X_i^S and X_i^E with $i \in [1,...,d]$ and the limits of a subdomain by X_i^s and X_i^e, with $i \in [1,...,d]$ the region spanned by the subdomain is referred to by the notation $\|X_i^s \Leftrightarrow X_i^e\|_{\forall i}$, as shown in Fig. 1. For a subdomain with limits different from the computational domain limits in only one dimension the notation $\|X_n^s \Leftrightarrow X_n^e\|_n$ is employed, where it is implied that the limits in the dimensions other than the nth do not differ from those of the computational domain (see Fig. 1). Also, $\|X_n^s\|_n$ is a shortcut that stands for the subdomain $\|X_n^s \Leftrightarrow X_n^s\|_n$. A property that is used in conjunction with the domain decomposition algorithms is the number of nodes of dependence of the discretized residual, b_r, which is defined as the maximum number of nodes on which the discretized residual depends on each side of the center node. For example, the minmod TVD discretization stencil (which is the longest of all stencils contained in the residual) would give $b_r = 2$ but should a first order Roe scheme be employed instead, then b_r would be set to one. Similarly, the number of nodes of dependence of the boundary nodes, b_b is defined as the maximum number of nodes any boundary node depends on along one direction. For the discretization stencils under consideration, b_b is set to 2, since the properties at the boundary nodes are extrapolated from at most 2 inner nodes using a blend of zeroth and first order extrapolation polynomials.

When the nodes comprised in the subdomain $\|X_i^s \Leftrightarrow X_i^e\|_{\forall i}$ are updated in pseudo-time, then it follows

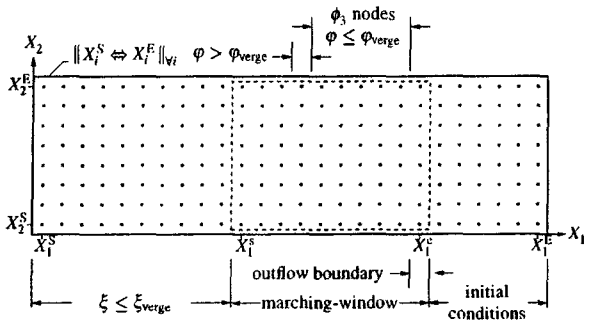

Fig. 2: Schematic of the marching-window cycle with the upstream and downstream boundaries in equilibrium surrounding an embedded streamwise elliptic region which is bounded upstream by the condition $\xi \le \xi_{verge}$ and downstream by the condition $\varphi \le \varphi_{verge}$; a dynamic outflow boundary condition is forced on all inner nodes in $\|X_1^e\|_1$.

from the definition of b_b that the boundary nodes situated inside $\|X_i^s - b_b \Leftrightarrow X_i^e + b_b\|_{\forall i}$ must be updated. The residual, which depends on both inner and boundary nodes must then be updated between $\|X_i^s - b_b - b_r \Leftrightarrow X_i^e + b_b + b_r\|_{\forall i}$. In many cases where there are no boundary nodes situated in the region $\|X_i^s - b_b \Leftrightarrow X_i^e + b_b\|_{\forall i}$, it is sufficient to update the residual in $\|X_i^s - b_r \Leftrightarrow X_i^e + b_r\|_{\forall i}$. For all methods presented in this paper, however, this shortcut is not implemented.

Standard Cycle

The "standard cycle" here implies the usual way of updating the solution in pseudo-time, by first finding the residual for all nodes and then updating the solution. The algorithm can be written in the following steps:

1. update the boundary nodes in the domain $\|X_i^S \Leftrightarrow X_i^E\|_{\forall i}$,
2. update the residual in the domain $\|X_i^S \Leftrightarrow X_i^E\|_{\forall i}$,
3. update Q (by pseudo-time stepping) in the domain $\|X_i^S \Leftrightarrow X_i^E\|_{\forall i}$, and,
4. convergence is attained when $\xi \le \xi_{verge}$ in the domain $\|X_1^S \Leftrightarrow X_1^E\|_{\forall i}$.

Marching-Window Cycle

An alternate form of the active-domain cycle that permits the solution of viscous streamwise separated flows and satisfies the convergence criterion of Eq. (3) is here presented. Named the marching-window, the algorithm differs from the active-domain on three points, namely 1) a dynamic outflow boundary is forced at the downstream boundary of the marching-window (see Fig. 2), 2) the ellipticity sensor responsible for a shift downstream of the downstream boundary of the marching-window is based on a Vigneron splitting of the streamwise pressure derivative instead of the streamwise component of the Mach number, and 3) the upstream boundary of the marching-window is positioned such that $\xi \le \xi_{verge}$ for all nodes upstream, instead of being a function of a residual monitor region and a streamwise ellipticity sensor based on the streamwise component of the Mach number. At the first iteration, the upstream boundary of the marching-window is set to the upstream boundary of the computational domain with the downstream boundary of the marching-window separated from the upstream boundary by b_b nodes. Denoting the upstream boundary of the marching-window by X_1^s and the downstream boundary by X_1^e, the marching-window cycle can be written as:

1. update Q (by pseudo-time stepping) in the subdomain $\|X_1^s \Leftrightarrow X_1^e\|_1$,
2. update the boundary nodes in the subdomain $\|X_1^s - b_b \Leftrightarrow X_1^e\|_1$,
3. update the residual (hence, ξ) in the subdomain $\|X_1^s - b_b - b_r \Leftrightarrow X_1^e\|_1$,
4. redefine the marching-window boundaries:
   (a) find the maximum value for X_1^s such that $\xi \le \xi_{verge}$ for all nodes in the subdomain $\|X_1^S \Leftrightarrow X_1^s - 1\|_1$,
   (b) every ϕ_2 iterations, if $\varphi > \varphi_{verge}$ for any node in the subdomain $\|X_1^e - \phi_3 \Leftrightarrow X_1^e\|_1$ or if $X_1^s > X_1^e - \phi_3$ then 1) increment X_1^e by one, 2) update the boundary nodes in the subdomain $\|X_1^e - 1 - b_b \Leftrightarrow X_1^e\|_1$ and 3) update the residual in the subdomain $\|X_1^e - 1 - b_b - b_r \Leftrightarrow X_1^e - 1\|_1$, and,
5. convergence is attained when $\xi \le \xi_{verge}$ for all nodes in the subdomain $\|X_1^s \Leftrightarrow X_1^e\|_1$ and when $X_1^e = X_1^E$.

The marching-window cycle is not self-starting and it must be ensured that the residual is updated for all nodes part of the computational window before the first iteration.

The ability of the marching-window at satisfying the convergence criterion of Eq. (3) lies in Steps 1-3, where Q is updated in pseudo-time in Step 1 *before* determining the residual in Step 3. Once Q is updated in Step 1 on the subdomain $\|X_1^s \Leftrightarrow X_1^e\|_1$, since a boundary node depends on at most b_b neighbors, it is sufficient to update the boundary nodes in the subdomain $\|X_1^s - b_b \Leftrightarrow X_1^e\|_1$ to guarantee that all boundary nodes upstream of X_1^e are up to date after Step 2. Once the boundary nodes have been updated in the subdomain $\|X_1^s - b_b \Leftrightarrow X_1^e\|_1$, since the discretized residual depends on at most b_r neighbors, it is sufficient to update the residual in the subdomain $\|X_1^s - b_b - b_r \Leftrightarrow X_1^e\|_1$ to guarantee that the residual of all nodes upstream of X_1^e are up to date after Step 3. Since ξ is a function of the residual, ξ upstream of X_1^e is up to date, and the upstream boundary of the marching-window X_1^s can be positioned correctly in Step 4a by

ensuring for all nodes upstream of X_1^e that $\xi \leq \xi_{verge}$. This serves two purposes: 1) the convergence criterion of Eq. (3) is satisfied if convergence is attained in Step 5, and 2) the upstream boundary of the marching-window moves upstream for any upstream propagating wave that affects the residual significantly and raises ξ above the user-defined convergence threshold ξ_{verge}. Contrarily to the active-domain, the upstream propagating wave is not limited to locally subsonic flow but includes all *significant* streamwise elliptic phenomena, such as streamwise separated flow, streamwise viscous derivatives, or flux limiters in the streamwise convection derivative, for instance.

Step 4b advances the marching-window downstream boundary when the width of the window is smaller than a user-specified constant ϕ_3, or when the streamwise ellipticity sensor φ is greater than the user-specified constant φ_{verge} for any node part of the subdomain $\| X_1^e - \phi_3 \Leftrightarrow X_1^e \|_1$. The streamwise ellipticity sensor φ is here defined as:[18]

$$\varphi \equiv \frac{\hat{X}_1}{\rho a} \max\left(0, \frac{1-M_1^2}{1+P_{\rho E} M_1^2}\right) \left|\frac{\partial P^*}{\partial X_1}\right| . \qquad (5)$$

It is important to note that the ellipticity sensor φ makes two important assumptions: 1) the streamwise ellipticity originating from the streamwise viscous derivative terms and the flux limiter part of the streamwise convection derivative is assumed negligible, and 2) at the point where φ is evaluated, the solution is assumed to be converged to steady-state. The first assumption is remedied by introducing a minimum width of the marching-window, ϕ_3, which is typically given a value ranging from 9 to 15. The second assumption can lead to some performance degradation of the marching-window when the flow near the downstream boundary is far from convergence. For this reason, the user-adjustable parameter ϕ_2 is introduced in Step 4a, with the consequence of evaluating φ every ϕ_2 iterations only. Therefore, a high value given to ϕ_2 helps in ensuring a more converged solution near the downstream boundary, and reduces the error in the ellipticity sensor φ due to temporarily non steady-state flow.

In Step 4b, after the downstream boundary of the marching-window is advanced by one station, the update of the boundary nodes in the subdomain $\| X_1^e - 1 - b_b \Leftrightarrow X_1^e \|_1$ and of the residual in the subdomain $\| X_1^e - 1 - b_b - b_r \Leftrightarrow X_1^e - 1 \|_1$ is necessary to ensure that the residual is properly updated in the marching-window, which is necessary for Step 1 to be performed correctly at the following iteration.

While the user-definable constants ϕ_2, ϕ_3, and φ_{verge} affect the performance of the marching-window cycle as a convergence acceleration technique, they do *not* affect the accuracy of the solution when convergence is attained due to the convergence criterion of Eq. (3) being satisfied.

Marching-Window / Multizone Cycle

The performance of the marching-window algorithm can be enhanced by introducing multizone decomposition inside the marching-window. Before each iteration, the marching-window subdomain $\| X_1^s \Leftrightarrow X_1^e \|_1$ is decomposed into several zones of length no more than ϕ_1 nodes in each dimension. Then, Steps 1-3 of the marching-window cycle (see Section) are replaced by Steps 1-3 of the multizone cycle. Details about the latter can be found in Ref. 18.

Test Cases

Two steady state supersonic flowfields are solved using the different types of cycles mentioned in the last section, and the performance of each is assessed on the basis of 1) the number of effective iterations, 2) CPU time, and 3) maximum storage required. To enable a fair comparison between the different cycle strategies, the number of effective iterations is defined as

$$\text{effective iterations} \equiv \frac{\text{number of times an inner node is updated}}{\text{total number of inner nodes}} , \qquad (6)$$

which is a good measure of the cycle performance as long as most of the computing effort is spent on the pseudo-time stepping instead of the residual, due to the overlap of the residual determination when a multizone decomposition is used. The implicit scheme used herein spends three quarters of its computing effort on the time stepping side, therefore reducing the residual overlap overhead work and justifying the use of Eq. (6) as a performance parameter. In spite of being accurately measured, the number of CPU seconds is not regarded as a more meaningful performance parameter due to the unavoidable bias that might occur in the programming of the cycles and the high dependence of the work on the architecture of the computer. Certain enhancements to the multizone cycle, such as unifying adjacent zones, could be implemented which would result in a non-negligible decrease in work, while the use of a vector computer (of CRAY type) would advantage the longer loops present in the standard cycle. Therefore, both the number of effective iterations and CPU time are monitored for the two test cases.

Inviscid Supersonic Inlet with a Blunt Leading Edge

A first comparison between the different cycles is performed for a steady-state inviscid flow over a 1 m long supersonic inlet. Air enters the channel at a Mach number of 5, a pressure of 4 kPa, and a temperature of 240 K. The grid size is set to 512 × 256 nodes. The user-defined parameters of interest are set to (when ap-

Table 1: Effective iteration count, work, and storage comparison at a CFL number of unity, for the blunt leading edge inviscid supersonic inlet case using a 512×256 node grid.

cycle	iter.	work	stor.
marching-window / multizone	44.9	1.0	1.0
marching-window	86.8	1.8	1.0
standard cycle	1391.0	23.6	17.6

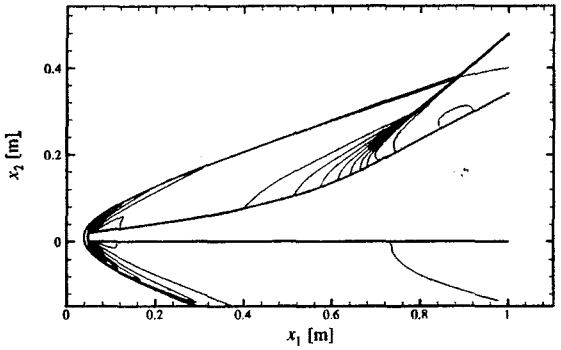

Fig. 3: Pressure contours for the blunt leading edge inviscid supersonic inlet case obtained using a 512×256 grid; the inflow conditions correspond to $M = 5$, $P = 4$ kPa and $T = 240$ K; no difference is noticeable between the pressure contours obtained with the different cycles.

plicable)

$$\sigma = 0.5, \quad \xi_{\text{verge}} = 100\ \tfrac{1}{s}, \quad \varphi_{\text{verge}} = 5000\ \tfrac{1}{s},$$
$$\phi_1 = 20, \quad \phi_2 = 3, \quad \text{and} \quad \phi_3 = 9,$$

where the value of 0.5 given to σ translates into a geometric average between the minimum CFL condition based pseudo-time step and the maximum CFL condition based pseudo-time step. The convergence threshold ξ_{verge} is low enough that a decrease in ξ_{verge} would not result in any noticeable difference of the pressure contours in Fig. 3. It is noted that the use of the entropy correction by Yee *et al.*[20] with $\tilde{\zeta} = 0.2$ is here used to avoid a carbuncle phenomenon near the blunt leading edge.

Table 1 shows the CPU time and effective iterations needed to reach convergence for the marching-window, marching-window/multizone, active-domain, multizone, and standard cycles. The marching-window cycle decreases the iteration count by allowing a computational window to travel in space following the propagation of the waves. This results in a decrease in effective iterations, compared to the standard cycle of 16 times. Furthermore, the use of multizone decomposition inside the marching-window focuses the pseudo-time stepping effort to the regions requiring more iterations to reach convergence, such as the region of subsonic flow upstream of the inlet blunt leading edge, hence resulting in only 45 effective iterations to reach convergence and an overall reduction in effective iterations of 31 times compared to the standard cycle.

Table 2: Effective iteration count, work, and storage comparison for the concatenated channels test case; the mesh size is of 256×128 nodes and the CFL number is varied between 0.1 and 10.

cycle	iter.	work	stor.
marching-window / multizone	219	1.0	1.0
marching-window	444	1.7	1.0
multizone	1215	5.9	6.2
standard cycle	2342	8.0	6.2

Table 3: Sensitivity of the effective iteration count and work to the user-defined constants for the concatenated channels test case; the marching-window/multizone cycle is used with a mesh size of 256×128 nodes, and a CFL range $0.1 \leq \text{CFL} \leq 10$.

ϕ_1	ϕ_2	ϕ_3	φ_{verge}	work	iter.
20	3	9	5×10^3	1.00	1.0
10	3	9	5×10^3	1.20	1.12
40	3	9	5×10^3	1.00	1.02
20	15	9	5×10^3	1.15	1.12
20	1	9	5×10^3	0.96	0.98
20	3	5	5×10^3	1.46	1.41
20	3	18	5×10^3	1.19	1.24
20	3	36	5×10^3	1.89	1.99
20	3	9	5×10^2	1.33	1.40
20	3	9	5×10^4	1.57	1.54

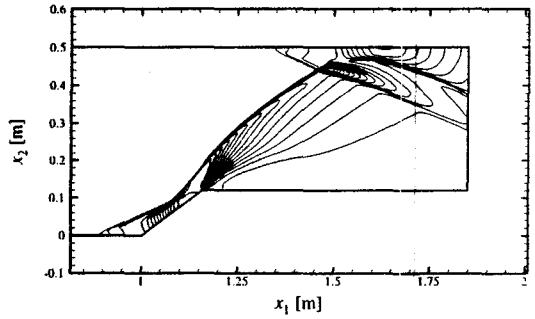

Fig. 4: Effective pressure contours of the concatenated channels case obtained using a 512×256 mesh; air enters the first channel at a Mach number of 5, a pressure of 1000 Pa and a temperature of 450 K.

Shock / Boundary Layer Interaction

The ability of the marching-window algorithm to solve shock / boundary layer interactions at hypersonic flow conditions is now tested. The geometry involves the concatenation of a 1.0×0.5 m channel to a 0.69×0.38 m channel through a 37° compression ramp. Air enters the first channel at uniform conditions of $M = 5$, $P = 1000$ Pa, and $T = 450$ K. Fixed temperature ($T_{\text{wall}} = 450$ K) wall boundary conditions are applied on bottom and top boundaries, with a grid clustered at

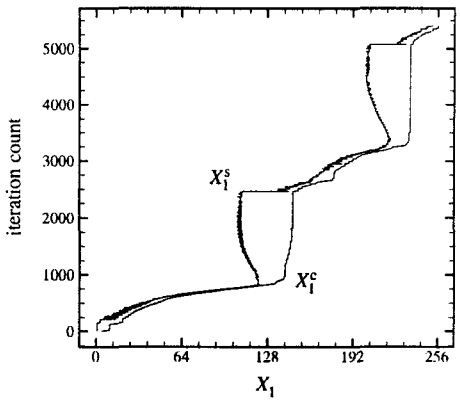

Fig. 5: Location of the marching-window upstream and downstream boundaries for the concatenated channels case using the marching-window/multizone cycle, with a variable CFL number, *ie* $0.1 \leq \text{CFL} \leq 10$; notice the high amount of work spent on the recirculation zones in the vicinity of $X_1 \sim 128$ and $X_1 \sim 218$, while very few steps are needed to converge the quasi-hyperbolic/parabolic regions.

both walls. As for the backward facing step case, a geometric averaged local time step is utilized to enhance wave propagation through high aspect ratio cells, while the other user-adjustable parameters are set to

$$\xi_{\text{verge}} = 100\ \tfrac{1}{s},\quad \varphi_{\text{verge}} = 5000\ \tfrac{1}{s},$$
$$\phi_1 = 20,\quad \phi_2 = 3,\quad \text{and}\quad \phi_3 = 9.$$

From the effective pressure contours of Fig. 4, two recirculation regions are visible: one at the start of the shock formed by the 37° wedge, and one at the point where the shock impinges on the top wall boundary layer. Both recirculation zones are of appreciable size due to the very low Reynolds number of the flow which helps generate thick incoming boundary layers. The major obstacle in converging this flowfield efficiently comes from the high difference in time scales between the convection dominated flow in the middle of the channels and the viscous dominated recirculation zones. Time accurate simulations of a similar problem indicate that the amount of time required for the separated flow regions to reach steady state is typically one order of magnitude more than the time needed for the shock structure to establish itself. Consequently, one would prefer high pseudo-time steps to be used in the recirculation zones for fast convergence, but unfortunately the step size is limited by nonlinear stability restrictions which are of importance especially near the non-converged shock waves. For these reasons, it is not surprising that as many as 2342 iterations are needed for the standard cycle to reach convergence, as Table 2 attests. Similarly to the backward facing step, the CFL number is linked to ξ_{max} such that at $\xi_{\text{max}} = 10^4$, the CFL number is 10 and at $\xi_{\text{max}} = 10^6$, the CFL number is 0.1.

The marching-window / multizone cycle performs particularly well as the work is focused on the reverse flow regions, while the rest of the domain is quasi-hyperbolic/parabolic and needs only a small amount of work to reach convergence (see Fig. 5). The use of the marching-window coupled with a multizone strategy makes possible a decrease in effective iterations of 8 times compared to the standard cycle as shown in Table 2.

As in the previous test case, a convergence criterion of $\xi_{\text{verge}} = 100\ \frac{1}{s}$ is found necessary to obtain reasonable accuracy, and no discernible difference is observed between the contours of properties obtained with the different cycles. Even if both the marching-window and the standard cycle guarantee the convergence criterion of Eq. (3) to be satisfied once convergence is attained, the governing equations have multiple roots due to their non-linearity, and a different flow solution could be obtained by the different cycle strategies. For all test cases presented here, however, it is verified that the same root is obtained independently of the acceleration technique.

The sensitivity of the user-adjustable parameters for the marching-window cycle is assessed for this test case in Table 3. It is seen that the performance of the marching-window is not affected considerably by a change of the average zone length ϕ_1 or by a change in ϕ_2, the latter being the number of iterations before a reading of the streamwise ellipticity sensor ξ is taken. For ϕ_1 varied from 10 to 40, the number of effective iterations is observed to change by only 12%, and for ϕ_2 varied from 1 to 15, the number of effective iterations increases by 14%. On the other hand, the parameters ϕ_3 and φ_{verge} are seen to affect the performance of the algorithm more significantly. Raising ϕ_3 from 9 to 36 increases twofold the number of effective iterations, and increasing φ_{verge} tenfold results in an increase of 54% in the effective iterations count. The high sensitivity of the effective iterations on either ϕ_3 or φ_{verge} is due to the high dependence of the width of the marching-window on these parameters. When the marching-window encloses too tightly a zone of streamwise ellipticity, the solution needs to be converged locally several times, hence increasing the work. When the marching-window overestimates the size of a streamwise elliptic region, the high number of iterations needed locally to converge a streamwise-elliptic region is spent on a larger portion of the computational domain, hence resulting in decreased performance.

Extension to Weakly-Ionized Flow

In a paper soon to be published,[22] the marching window is extended to the reacting weakly-ionized Favre-averaged Navier-Stokes equations (WIFANS), consisting of the time-averaged Navier-Stokes equations closed by the $k\omega$ turbulence model coupled to the electric field potential equation. The addition of the electric

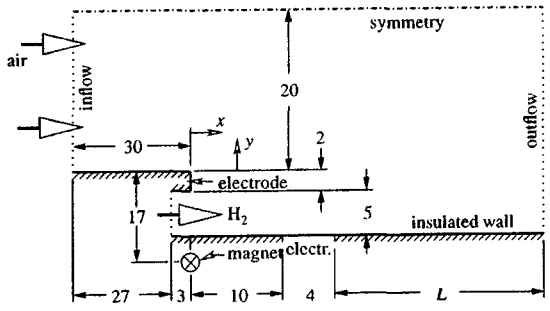

Fig. 6: Problem setup for the second test case. All dimensions in cm.

field potential equation prohibits the straightforward deployment of the marching window due to the system of equations not exhibiting some quasihyperbolicity in the streamwise direction. Despite the lack of quasihyperbolicity of the WIFANS equations, it is observed that a significant reduction in work can be achieved while solving supersonic viscous flows if (i) the use of the marching window is limited to all equations except the electric field potential equation, and (ii) the sensors at the upstream and downstream boundaries of the marching window are modified to include the influence of the electric field potential on the residual of the fluid flow equations. Upon completion of one streamwise sweep, the so-modified marching window cycle satisfies the same convergence criterion as the standard pseudotime stepping cycle, hence resulting in the same converged solution within the tolerance of a user-specified convergence threshold.

Fuel Injection Through a Backward-Facing Step

To assess the performance of the algorithm upon solving the weakly-ionized FANS equations, a reacting hydrogen/air parallel mixing problem is chosen as a test-case. The hydrogen is injected through a backward-facing step as depicted in Fig. 6. The Jachimoswky chemical solver is used with 9 species solved, namely H_2, O_2, H, O, OH, H_2O, HO_2, H_2O_2, N_2. The air inflow conditions are such that the nitrogen and oxygen mass fractions correspond to 0.765 and 0.235 respectively, the Mach number is 4, the static pressure is 4 kPa, and the temperature is 700 K. The fuel inflow Mach number is set to 2.6, the pressure to 4 kPa, and the temperature to 700 K. On the vertical side of the backward-facing step separating the hydrogen from the air at the point of injection, the electric field potential ψ on the side wall is set to 0 while being set to 55 V on the 4-cm wide electrode located on the bottom wall 10 cm downstream of the start of the mixing region. At all other boundary nodes, insulated conditions are in effect. A magnet of strength $B_d = 20$ Teslas is located in the channel near the bottom wall, and positioned such that $\vec{A} = (0, -0.17\text{m}, 0)$ and $\vec{B} = (0, -0.17\text{m}, 0.05\text{m})$ with the definition of B_d, $\vec{A}$, and $\vec{B}$ outlined in Ref. 22. It is noted that this arrangement results in the magnitude of the magnetic field not exceeding 2.5 Teslas throughout the computational domain. At all wall boundaries, the temperature is extrapolated such that the temperature gradient normal to the wall vanishes.

The mesh is constructed of 148 gridlines in the cross-stream direction and of either 338 gridlines along the streamwise direction when $L = 56$ cm or of 974 gridlines along the streamwise direction when $L = 456$ cm. The mesh spacing at the top and bottom boundaries is set to 3×10^{-5} m and exponentially becomes greater near the center of the channel. The gridlines in the streamwise direction are clustered at the interface between the electrodes and the insulated walls, with a spacing equal to 3×10^{-5} m. The electrical conductivity is set to 30 mho/m, while the Hall parameter for the electrons is set to 0.1. The user-defined constants related to the marching window are set to

$$\phi_1 = 20, \quad \phi_2 = 6, \quad \phi_3 = 18, \quad \varphi_{\text{verge}} = 7 \times 10^4 \, \tfrac{1}{s}, \quad \text{and} \quad \xi_{\text{verge}} = 10^3 \, \tfrac{1}{s}.$$

In addition to the latter, two parameters need to be specified when solving weakly-ionized flow.[22] More specifically,

$$\phi_4 = 3 \quad \text{and} \quad \xi_{\psi\text{verge}} = 5 \times 10^3 \frac{1}{s}.$$

A value of 0.2 is assigned to σ for both the marching window and the standard cycles. Nonlinear stability restrictions prevent the use of a CFL number higher than 5.0 for either the marching window or the standard cycle for most of the convergence history. Furthermore, a CFL number higher than 1.0 was observed to induce oscillations in the fluid flow convergence parameter ξ, hence preventing convergence. Therefore, we here compare the standard cycle to the marching window cycle at a CFL number of 1.0 and a Von Neumann number alternated between 70, 490, and 3430, with the latter being used to solve in pseudo-time the electric field potential equation.

The impact of the electromagnetic effects on the fluid flow properties is shown in Fig. 7 through a comparison of the y-velocity contours with an additional case obtained with the same geometry and inflow conditions, but without the presence of a potential difference at the electrodes and without an externally-applied magnetic field. It can be seen that the Lorentz force affects significantly the velocity near the electrode at the bottom wall situated between $x = 10$ cm and $x = 14$ cm, while having a less pronounced influence further downstream. A study of the convergence history reveals that the electromagnetic source terms entail a zone of streamwise ellipticity located between $0 \leq x \leq 45$ cm. This is significantly greater than the zone of streamwise ellipticity associated with the recirculation region located downstream of the step, which spans the range $0 \leq x \leq 8$ cm. Due to the zone of

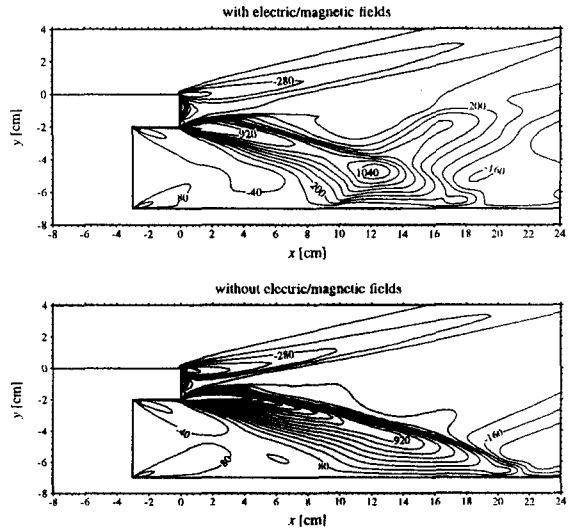

Fig. 7: Effect of the presence of the electric/magnetic fields on the y-velocity contours in m/s for the second test case consisting of fuel injection through a backward facing step.

Table 4: Work (in normalized CPU seconds) for the weakly-ionized testcase consisting of fuel injection through a backward facing step with $L = 56$ cm and a grid size of 338×148 nodes.

cycle	$\varphi_{\psi,\ \text{verge}}$, $\frac{1}{s}$	work
marching window / multizone	1×10^3	1.02
marching window / multizone	1×10^4	1.00
marching window / multizone	3×10^4	1.01
marching window / multizone	1×10^5	1.31
standard	–	6.04

streamwise ellipticity induced electromagnetically being larger than the one induced by flow recirculation, this test case permits the adequate assessment of the performance of the streamwise ellipticity sensors related to electromagnetic effects.

The effect of the electromagnetic streamwise ellipticity sensor $\varphi_{\psi,\ \text{verge}}$ (as defined in Ref. 22) on the performance of the marching window is assessed in Table 4 for a distance between the electrode on the bottom wall and the domain exit, L, fixed to 56 cm. Interestingly, varying $\varphi_{\psi,\ \text{verge}}$ between 10^3 1/s and 10^5 1/s does not affect greatly the work which vary by no more than 31%. The use of the marching window is seen to decrease the work required for convergence by more than 6 times.

The results shown in Table 4 are obtained by setting L to 56 cm, so that the streamwise elliptic region ends just upstream of the domain exit. While this is useful to assess the performance of the marching window for a flowfield where a large part of the computational domain has some streamwise ellipticity, it does not permit to assess adequately the loss in performance related to an overestimation of the length of the

Table 5: Work (in normalized CPU seconds) for the second testcase consisting of fuel injection through a backward facing step with $L = 456$ cm and a grid size of 974×148 nodes.

cycle	$\varphi_{\psi\ \text{verge}}$, $\frac{1}{s}$	work
marching window / multizone	3×10^3	1.01
marching window / multizone	3×10^4	1.00
marching window / multizone	3×10^5	1.35
standard	–	24.06

streamwise ellipticity region by giving a too-low value to $\varphi_{\psi,\ \text{verge}}$. For this reason, the computational domain is here extended by 4 meters in the streamwise direction, with the domain exit situated well downstream of the streamwise ellipticity region. The effect of $\varphi_{\psi,\ \text{verge}}$ is assessed in Table 5 for $L = 456$ cm. What is particularly surprising is the small effect of $\varphi_{\psi,\ \text{verge}}$ on the work when being decreased tenfold from 3×10^4 1/s to 3×10^3 1/s. A study of the convergence history reveals that the tenfold reduction in $\varphi_{\psi,\ \text{verge}}$ results in an over-prediction of the streamwise ellipticity region of 120 gridlines or approximately 37 cm. Compared to the size of the domain, this is a rather small over-prediction, which explains partly the small addition in work. Using a $\varphi_{\psi,\ \text{verge}}$ value of 3×10^4 1/s, a twenty-fourfold decrease in the work is obtained through the use of the marching window compared to the standard cycle.

Conclusions

A novel acceleration technique is presented which is aimed at accelerating the convergence of the Favre-averaged Navier-Stokes equations in the supersonic regime for flowfields with large streamwise separated flow regions. Similarly to the active-domain method,[13] the marching-window iterates in pseudo-time a band-like computational domain of minimal width which adjusts to the size of the streamwise elliptic regions when encountered. However, contrarily to the active-domain method, it is shown that the marching-window guarantees the residual on all nodes to be below a user-defined threshold when convergence is reached, and hence results in the same converged solution (within the tolerance of the convergence criterion) as the one obtained by standard pseudo-time marching methods. A multizone decomposition is implemented inside the marching-window to restrict the computing to the zones where the residual is above the user-defined convergence threshold. This is shown to further decrease the work needed for convergence by close to 2 times for the problems shown herein.

The use of the marching-window with multizone decomposition on a shock boundary layer interaction flowfield (where two large streamwise separated regions are present) reveals a six-fold decrease in storage and a eight-fold decrease in work compared to the

standard cycle. When extended to the weakly-ionized FANS equations, the algorithm performs equally well for a problem involving fuel injection through a backward-facing step where the fluid properties are subject to considerable electromagnetic effects. In the latter case, the use of the marching window results in up to a twenty-four-fold decrease in work compared to the standard cycle.

The proposed algorithm is also shown to work well at a low CFL number in regions of quasi-hyperbolicity and is recommended for stiff problems with high nonlinear stability restrictions on the time step size. The reduction in computational work through the use of the marching-window is made possible by focusing the high number of iterations needed to converge the streamwise separated regions to the region in question. The amount of storage needed is also significantly reduced if no memory is allocated to the nodes outside of the marching-window subdomain.

The performance of the algorithm is seen not to be too sensitive to the user-defined ellipticity threshold constants and the window minimal width. However, it is unclear at this stage by how much these parameters would need to be altered for very different flow properties and physical domain size.

Acknowledgements

This work has been supported by the BK21 program and by the Natural Sciences and Engineering Research Council of Canada.

References

1) Vigneron, Y. C., Rakich, J. V., and Tannehill, J. C., "Calculations of Supersonic Viscous Flow Over Delta Wings with Sharp Subsonic Leading Edges," 1978, AIAA Paper 78-1137.

2) Lawrence, S. L. and Tannehill, J. C., "An Upwind Algorithm for the Parabolized Navier-Stokes Equations," 1986, AIAA Paper 86-1117, presented at the AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics and Lasers Conference.

3) Tannehill, J., Ievalts, J. O., and Lawrence, S., "An Upwind Parabolized Navier-Stokes Code for Real Gas Flows," Jan. 1988, AIAA Paper 89-0713, presented at the 26th Aerospace Sciences Meeting.

4) D'Ambrosio, D. and Marsilio, R., "A Numerical Method for Solving the Parabolized Three-Dimensional Navier-Stokes Equations," *Computers & Fluids*, Vol. 26, No. 6, 1997, pp. 587–611.

5) Miller, J. H., Tannehill, J. C., Lawrence, S. L., and Edwards, T. A., "Parabolized Navier-Stokes Code for Hypersonic Flows in Thermo-Chemical Equilibrium or Nonequilibrium," *Computers & Fluids*, Vol. 27, No. 2, 1998, pp. 199–215.

6) Miller, J. H., Tannehill, J. C., and Lawrence, S. L., "Parabolized Navier-Stokes Algorithm for Solving Supersonic Flows with Upstream Influences." *AIAA Journal*, Vol. 38, No. 10, 2000, pp. 1837–1845.

7) Power, G. D. and Barber, T. J., "Analysis of Complex Hypersonic Flows with Strong Viscous/Inviscid Interaction," *AIAA Journal*, Vol. 26, No. 7, 1988, pp. 832–840.

8) Chang, C.-L. and Merkle, C. L., "The Relation between Flux Vector Splitting and Parabolized Schemes," *Journal of Computational Physics*, Vol. 80, 1989, pp. 344–361.

9) Yamalcev, N. K. and Ballmann, J., "Iterative Space-Marching Method for Compressible Sub, Trans, and Supersonic Flows," *AIAA Journal*, Vol. 38, No. 2, 2000, pp. 225–233.

10) Rubin, S. G. and Reddy, D. R., "Analysis of Global Pressure Relaxation for Flows with Strong Interaction and Separation," *Computers & Fluids*, Vol. 11, No. 4, 1983, pp. 281–306.

11) Barnett, M. and Davis, R. T., "Calculation of Supersonic Flows with Strong Viscous-Inviscid Interaction," *AIAA Journal*, Vol. 24, No. 12, 1986, pp. 1949–1955.

12) Bardina, J. E., "Three-Dimensional Navier-Stokes Method with Two-Equation Turbulence Models for Efficient Numerical Simulation of Hypersonic Flows," 1994, AIAA Paper 94-2950, presented at the 30th Joint Propulsion Conference and Exhibit, Indianapolis, IN.

13) Nakahashi, K. and Saitoh, E., "Space-Marching Method on Unstructured Grid for Supersonic Flows with Embedded Subsonic Regions," *AIAA Journal*, Vol. 35, No. 8, 1997, pp. 1280–1285.

14) Morino, H. and Nakahashi, K., "Space-Marching Method on Unstructured Hybrid Grid for Supersonic Viscous Flows," 1999, AIAA Paper 99-0661.

15) Wilcox, D. C., "Reassessment of the Scale Determining Equation for Advanced Turbulence Models," *AIAA Journal*, Vol. 26, No. 11, 1988, pp. 1299–1310.

16) Vinokur, M., "Conservative Equations of Gas-Dynamics in Curvilinear Coordinate Systems," *Journal of Computational Physics*, Vol. 14, 1974, pp. 105–125.

17) Viviand, H., "Conservative Forms of Gas Dynamics Equations," *La Recherche Aérospatiale*, , No. 1, Jan. 1974, pp. 65–68.

18) Parent, B. and Sislian, J. P., "The Use of Domain Decomposition in Accelerating the Convergence of Quasihyperbolic Systems," *Journal of Computational Physics*, Vol. 179, No. 1, 2002, pp. 140–169.

19) Wilcox, D. C., "Dilatation-Dissipation Corrections for Advanced Turbulence Models," *AIAA Journal*, Vol. 30, No. 11, 1992, pp. 2639–2646.

20) Yee, H. C., Klopfer, G. H., and Montagné, J.-L., "High-Resolution Shock-Capturing Schemes for Inviscid and Viscous Hypersonic Flows," *Journal of Computational Physics*, Vol. 88, 1990, pp. 31–61.

21) Laney, C. B., *Computational Gasdynamics*, Cambridge University Press, Cambridge, New-York, 1998.

22) Parent, B. and Jeung, I.-S., "Extension of the Marching Window Algorithm to the Weakly-Ionized Navier-Stokes Equations," *Journal of Computational Physics*, 2004, recommended for publication, Dec. 2003; can be downloaded online at `ftp://147.46.241.213/wifanswindow.pdf`.