Journal of Computational Physics **157,** 2–43 (2000)
doi:10.1006/jcph.1999.6303, available online at http://www.idealibrary.com on IDEAL®

Grid Overlapping for Implicit Parallel Computation of Compressible Flows

Zi-Niu Wu*, † and Hui Zou†

**Department of Engineering Mechanics, Tsinghua University, Beijing 100084, People's Republic of China, and †National Laboratory for CFD, Beijing University of Aeronautics and Astronautics, Beijing 100083, People's Republic of China*
E-mail: ziniuwu@tsinghua.edu.cn

Received October 8, 1998; revised March 23, 1999

The suitability of applying the overlapping grid method to parallel computation of steady and unsteady compressible inviscid flows with three-point block-tridiagonal implicit schemes is addressed in this paper. An easily usable interface treatment is constructed and analyzed for both steady and unsteady problems. The performance of the method, such as convergence rate and time accuracy, can be controlled through the overlapping width. The method needs no iteration at each time step or modification of the Thomas algorithm for the solution of the implicit parts. In both steady and unsteady cases a very good absolute parallel efficiency is demonstrated for bidimensional subsonic and transonic flow computations. © 2000 Academic Press

Key Words: overlapping interface condition; stability; convergence; accuracy; parallel computing.

1. INTRODUCTION

Parallel computation is now very important in computational fluid dynamics (CFD). The most natural way to achieve parallelization in CFD problems is by domain decomposition [10, 17]. This is relatively easy for explicit schemes since explicit schemes are defined pointwisely and are inherently parallel. But for implicit schemes, for which the discrete equations are spatially coupled, the situation is obviously more complicated. Only tridiagonal implicit schemes (which involve three points in each space direction) inverted by the Thomas algorithm are considered in this paper. In high dimensions the implicit system will be split to one-dimensional systems by approximate factorization.

The main difficulty of parallel computation for implicit schemes is how to invert the implicit system in a parallel way. Traditionally, one uses the parallel tridiagonal solver [20], in which the original Thomas algorithm for inverting a tridiagonal system is modified. In this paper we use overlapping multiblock grids with time-lagging interface conditions to realize parallel computation. The reason to use an overlap at the interface is different here for steady and unsteady problems.

For steady state problems, a time-lagging interface condition is obviously very simple but normally reduces the convergence speed to a steady state. According to a study based on the well-known normal mode analysis [6, 15], for dissipative schemes the method converges, but very slowly with a small overlapping width. For a particular scheme, a quantitative analysis in [21] demonstrated a very surprising result: *when the overlapping width, in terms of the number of grid points in the overlap, is equal to the CFL* (*Courant–Friedrich–Lewy*) *number, the overlapping method converges as rapidly as the corresponding single domain treatment*. But the quantitative analysis presented in [21] was based on a particular scheme and a particular interface treatment. We do not know whether the previous conclusion remains true for other schemes and for other kinds of interface conditions.

For unsteady problems, a time-lagging treatment will necessarily reduce the time accuracy near the interface. According to [11], for a discrete problem, the wave travels at finite speed. Since the CFL number is based on the maximum wave speed (eigenvalue), it is natural that a local perturbation (due to time-lagging) of the scheme will travel at a distance, in terms of the number of mesh points, no larger than the CFL number at each time step. From this remark it is possible to choose an overlapping width proportional to the CFL number and correct the error (due to time-lagging) confined in the overlap.

The main purpose of this paper is to:

(1) reexamine the results of Ref. [21] for schemes with various dissipation properties (strongly dissipative, moderately dissipative, and nondissipative) and for various interface conditions;

(2) construct a new and easily parallelizable interface treatment for the unsteady problem;

(3) apply the interface treatment to parallel computation.

The present method for both steady and unsteady problems needs no iteration at each time step or modification of the Thomas algorithm for the solution of the implicit parts. In the present paper we are more interested in the numerical efficiency than the parallel aspects. For references where the parallel aspects are emphasized, see for instance [17, 18]. The overlapping is just for the purpose of parallelization and does not involve any interpolation in space. Once interpolation in space is used as in the case of treating complex geometry by arbitrary grid overlapping, there exist problems such as conservation, stability, accuracy, etc. See for instance [1, 3, 7, 16, 19, 22, 23].

The present study will be limited to three-point schemes and to hyperbolic problems. For schemes with more than three points in space and for Navier–Stokes equations, we are obtaining similar results which need further study.

This paper is organized as follows. In Section 2, the one-dimensional interface problem is described using only two subdomains. The easily parallelizable interface conditions for both steady and unsteady computations are presented. In Section 3, we analyze the interface treatments. First we present a stability analysis of various interface treatments. For steady state problems, we will also study the convergence to a steady state for schemes with various dissipation properties and for various interface conditions. For unsteady problems we will study the time accuracy of the proposed interface treatments, both analytically and numerically. Section 4 is devoted to numerical experiments in two-dimensional compressible flow computations. First we perform sequential computation in order to demonstrate the convergence for steady state problems and accuracy for an unsteady problem, with comparison to single domain computations. Then we present PVM-based parallel computation to demonstrate that the proposed approach has a good absolute parallel efficiency. The main conclusions are summerized in Section 5. There are also two appendixes provided at the end of this paper. Appendix A concerns a short remark about conservation which is not the main concern of this paper. Appendix B is a discussion of the usefulness of the present method due to a practical point of view.

2. PRESENTATION OF THE NUMERICAL METHODS ON OVERLAPPING GRIDS

We only present the method in one dimension since its extension to two dimensions is straightforward.

2.1. *Interface Difference Approximations*

Consider the following system of hyperbolic conservation laws,

$$w_t + h(w)_x = 0, \qquad t \in \mathbf{R}^+, -1 < x < 1 \tag{1}$$

with initial data,

$$w(x, t = 0) = w_0(x), \qquad x \in \mathbf{R} \tag{2}$$

and suitable boundary conditions at $x = \pm 1$. By hyperbolic assumption, the Jacobian matrix $C(w) = \frac{dh(w)}{dw}$ has real eigenvalues $\lambda^{(i)}(w)$ and is diagonalizable.

When only two subdomains are considered, the computational domain is split as $\mathcal{D}_u = \{x : x < \frac{1}{2}l_o\}$, $\mathcal{D}_v = \{x : -\frac{1}{2}l_o < x\}$ with an overlapping length l_o. The boundaries $x = -\frac{1}{2}l_o$ and $x = \frac{1}{2}l_o$ of the overlap are called interfaces. A uniform mesh size of δx is assumed in each subdomain, so that the cell centers in the left and right subdomains are respectively given by $x_j^{(u)} = \frac{1}{2}l_o + (j - 0.5)\delta x$, $x_j^{(v)} = -\frac{1}{2}l_o + (j + 0.5)\delta x$. The overlap $(-\frac{1}{2}l_o, \frac{1}{2}l_o)$ contains L_o grid points for both subdomains. We will call L_o the overlapping width (in terms of the number of grid points). The case of more subdomains can be similarly described. For convenience, the analysis is essentially based on two subdomains. But in the application more subdomains are considered.

The numerical solutions are denoted by $u_j^n = w(x_j^{(u)}, n\delta t)(j \leq 0)$ in $\mathcal{D}_u$ and $v_j^n = w(x_j^{(v)}, n\delta t)(j \geq 0)$ in $\mathcal{D}_v$, where δt is the time step. In each subdomain, the system (1) is approximated by a difference scheme in conservation form,

$$L\,\Delta u_j^{n+1} = -\sigma\left(f_{j+1/2}^n - f_{j-1/2}^n\right), \qquad j \leq -1 \tag{3}$$

$$L\,\Delta v_j^{n+1} = -\sigma\left(g_{j+1/2}^n - g_{j-1/2}^n\right), \qquad j \geq 1. \tag{4}$$

Here $\Delta u_j^{n+1} = u_j^{n+1} - u_j^n$, $\Delta v_j^{n+1} = v_j^{n+1} - v_j^n$ denote the time increments; $f_{j+1/2}, g_{j+1/2}$ are numerical fluxes consistent with the exact flux function $h(w)$; σ is the ratio between δt and δx; and L is a three-point implicit operator. For parallel computation, we always use the same scheme in each subdomain.

2.2. *Interface Conditions for Steady State Problems*

In order to independently solve the difference equations in each subdomain as required by parallel computation, we need to use a time lagging of the interface values.

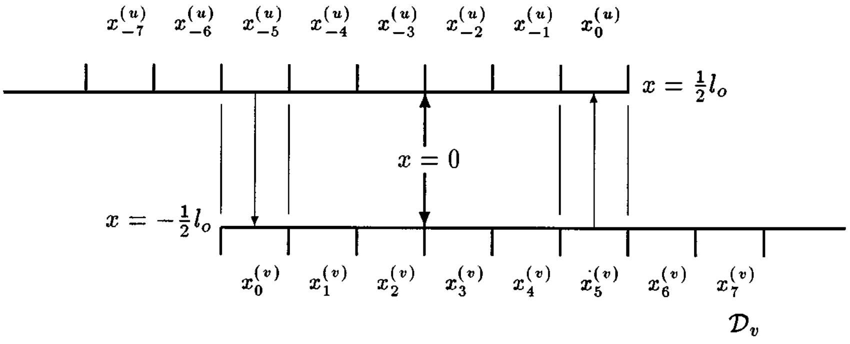


FIG. 1. Interface condition for steady problems.

2.2.1. Time-lagging interface conditions. An implicit scheme involves an explicit stage (depending on values at n and lower) and implicit stage (depending on values at $n+1$). One necessarily lags in time the interface values at $n+1$. But for the explicit stage, one can use the value at n (no time lagging) or the value at $n-1$ (time lagging). This leads to several combinations (see Fig. 1).

Now let us present these possibilities more precisely. Assume everything is known at time level n. Then we can compute the explicit stages (right-hand sides) of (3)–(4). Let us just concentrate on (3). To invert the operator (matrix) on the left-hand side (implicit stage) of (3), we need some condition on Δu^{n+1} at the left-hand boundary which we get as we would for a single domain case, and some condition on Δu^{n+1} at the right-hand boundary (interface), which we could either take as $\Delta u_0^{n+1} = \Delta v_{L_o-1}^n$ (I1) or as $\Delta u_0^{n+1} = 0$ (I2). Solving gives us u^{n+1} (and v^{n+1}) at interior points. Finally define u_0^{n+1} via either $u_0^{n+1} = v_{L_o-1}^n$ (E1) or $u_0^{n+1} = v_{L_o-1}^{n+1}$ (E2). For convenience, we still denote u_0^n as the interface value used in the explicit stage, and Δu_0^{n+1} or u_0^{n+1} as the value used in the implicit stage. Since each of the explicit and implicit stages involves two possibilities, we have in total 4 possibilities, which we summarize as

$$\begin{array}{llll}
\text{(combinations)} & \text{explicit stage,} & \text{implicit stage} & \\
(\text{E1} \times \text{I1}) & u_0^n = v_{L_o-1}^{n-1}, & \Delta u_0^{n+1} = \Delta v_{L_o-1}^n & (5) \\
(\text{E2} \times \text{I2}) & u_0^n = v_{L_o-1}^{n}, & \Delta u_0^{n+1} = 0 & (6) \\
(\text{E1} \times \text{I2}) & u_0^n = v_{L_o-1}^{n-1}, & \Delta u_0^{n+1} = 0 & (7) \\
(\text{E2} \times \text{I1}) & u_0^n = v_{L_o-1}^{n}, & \Delta u_0^{n+1} = \Delta v_{L_o-1}^n & (8)
\end{array}$$

which can also be written as

$$\begin{array}{llll}
\text{(combinations)} & \text{explicit stage,} & \text{implicit stage} & \\
(\text{E1} \times \text{I1}) & u_0^n = v_{L_o-1}^{n-1}, & u_0^{n+1} = v_{L_o-1}^n & (9) \\
(\text{E2} \times \text{I2}) & u_0^n = v_{L_o-1}^{n}, & u_0^{n+1} = v_{L_o-1}^n & (10) \\
(\text{E1} \times \text{I2}) & u_0^n = v_{L_o-1}^{n-1}, & u_0^{n+1} = v_{L_o-1}^{n-1} & (11) \\
(\text{E2} \times \text{I1}) & u_0^n = v_{L_o-1}^{n}, & u_0^{n+1} = 2v_{L_o-1}^n - v_{L_o-1}^{n-1}. & (12)
\end{array}$$

The interface condition defined by (5) or (9) has the particular feature that the interface value lags in time equally at each level of the scheme and will be called the *totally time-lagging interface condition*.

In the interface condition defined by (6) or (10), the time lagging occurs only at the implicit level. We will call this the *partially time-lagging condition*.

The interface condition defined by (7) or (11) uses a value at $n-1$ to define the value at $n+1$ and will be called the *over-time-lagging condition*.

The interface condition defined by (8) or (12) has a first-order accuracy (locally second order). According to Gustafsson [9], the boundary (or interface) treatment can be one order less accurate than the interior difference equation, without dropping the overall order of accuracy. Thus the overall time accuracy for a second order interior treatment will be still second order for this interface condition. As a result, we will call it the *time-accurate time-lagging condition*.

The interface conditions for v should be similarly defined as

$$\begin{array}{llll}
\text{(combinations)} & \text{explicit stage,} & \text{implicit stage} & \\
(\text{E1} \times \text{I1}) & v_0^n = u_{-L_o+1}^{n-1}, & \Delta v_0^{n+1} = \Delta u_{-L_o+1}^n & (13) \\
(\text{E2} \times \text{I2}) & v_0^n = u_{-L_o+1}^{n}, & \Delta v_0^{n+1} = 0 & (14) \\
(\text{E1} \times \text{I2}) & v_0^n = u_{-L_o+1}^{n-1}, & \Delta v_0^{n+1} = 0 & (15) \\
(\text{E2} \times \text{I1}) & v_0^n = u_{-L_o+1}^{n}, & \Delta v_0^{n+1} = \Delta u_{-L_o+1}^n. & (16)
\end{array}$$

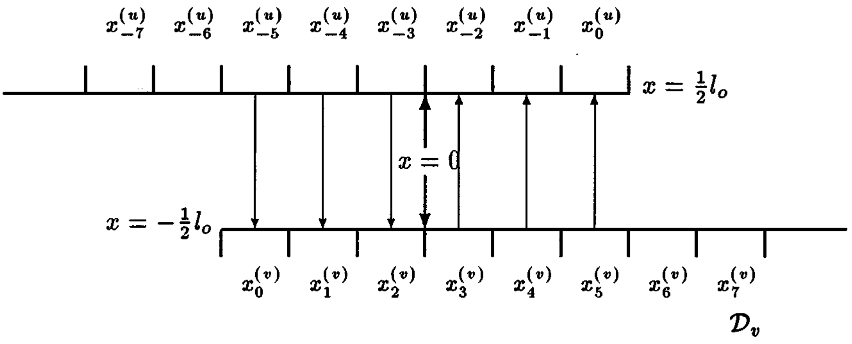
2.2.2. Standard and nonstandard interface conditions. The interface conditions derived above are not all standard. In the standard definition, the interface value at time $n+1$ is obtained from the value at time n augmented by the time increment. Precisely, after defining Δu_0^{n+1} for the solution of the implicit stage, u_0^{n+1} (which is to be used in the next time step) should be defined as

$$u_0^{n+1} = u_0^n + \Delta u_0^{n+1}.$$

There is no difficulty to check that the totally time-lagging condition is standard but the partially, over-, and time-accurate interface conditions are nonstandard. Only the standard interface treatment can be directly analyzed through the stability theory. However, nonstandard treatments, which are as simple as the standard treatment, are often used by engineers and deserve a study here.

2.3. *Interface Treatment for Unsteady Problems*

For unsteady problems, the time accuracy of the interface treatment is very important. But a time-lagging interface treatment, which is not accurate in time, is convenient for an independent solution of the implicit difference schemes, as required by parallel computation. The main idea here is to use a time-lagging treatment; the error caused by this time-lagging is corrected, at each time step, through an additional interpolation which we call projection. The method proposed here is based on the fact that the main part of the numerical wave should travel at a distance (in terms of the number of mesh points) no larger than the CFL number at each time step. We therefore use a time-lagging interface condition and an overlapping width $L_o = 2(\text{CFL}+1)$. At each time step the error produced at the interfaces by time-lagging will spread over CFL mesh points is each subdomain. Consider for instance



$x^{(u)}_{-7}$ $x^{(u)}_{-6}$ $x^{(u)}_{-5}$ $x^{(u)}_{-4}$ $x^{(u)}_{-3}$ $x^{(u)}_{-2}$ $x^{(u)}_{-1}$ $x^{(u)}_{0}$
$x = \frac{1}{2}l_o$
$x = 0$
$x = -\frac{1}{2}l_o$
$x^{(v)}_{0}$ $x^{(v)}_{1}$ $x^{(v)}_{2}$ $x^{(v)}_{3}$ $x^{(v)}_{4}$ $x^{(v)}_{5}$ $x^{(v)}_{6}$ $x^{(v)}_{7}$
$\mathcal{D}_v$

FIG. 2. Interface condition for unsteady problems.

CFL $= 2$ (see Fig. 2). After inverting the implicit system with a time-lagging interface condition, the solution is polluted at $j = -1, -2$ in the left subdomain and at $j = 1, 2$ in the right subdomain. But the solutions at $j \le -3$ (left subdomain) and at $j \ge 3$ (right subdomain) are not (much) polluted. We then project the unpolluted solutions at $j = 3, 4$ (right subdomain) to the polluted points $j = -2, -1$ (left subdomain), and the unpolluted solutions at $j = -4, -3$ (left subdomain) to the polluted points $j = 1, 2$ (right subdomain). The interface treatment by overlapping/projection can now be summarized as follows:

(1) At each time level n, compute the explicit and implicit stages of the scheme in each subdomain by using one of the following time-lagging interface conditions

(combinations)	explicit stage,	implicit stage	
(E1 × I1)	$u_0^n = v_{L_o-1}^{n-1}$,	$\Delta u_0^{n+1} = \Delta v_{L_o-1}^n$	(17)
(E2 × I2)	$u_0^n = v_{L_o-1}^{n}$,	$\Delta u_0^{n+1} = 0$	(18)
(E1 × I2)	$u_0^n = v_{L_o-1}^{n-1}$,	$\Delta u_0^{n+1} = 0$	(19)
(E2 × I1)	$u_0^n = v_{L_o-1}^{n}$,	$\Delta u_0^{n+1} = \Delta v_{L_o-1}^n$	(20)

for u_0 and

(combinations)	explicit stage,	implicit stage	
(E1 × I1)	$v_0^n = u_{-L_o+1}^{n-1}$,	$\Delta v_0^{n+1} = \Delta u_{-L_o+1}^n$	(21)
(E2 × I2)	$v_0^n = u_{-L_o+1}^{n}$,	$\Delta v_0^{n+1} = 0$	(22)
(E1 × I2)	$v_0^n = u_{-L_o+1}^{n-1}$,	$\Delta v_0^{n+1} = 0$	(23)
(E2 × I1)	$v_0^n = u_{-L_o+1}^{n}$,	$\Delta v_0^{n+1} = \Delta u_{-L_o+1}^n$	(24)

for v_0.

(2) Correct the errors by projecting the unpolluted solutions to the polluted points in the overlap $p = L_o/2$

$$\begin{cases} u_0^{n+1} = v_{L_o-1}^{n+1} & v_0^{n+1} = u_{-L_o+1}^{n+1} \\ u_{-1}^{n+1} = v_{L_o-2}^{n+1} & v_1^{n+1} = u_{-L_o+2}^{n+1} \\ \vdots & \vdots \\ u_{-p+1}^{n+1} = v_{L_o-p}^{n+1} & v_{p-1}^{n+1} = u_{-L_o+p}^{n+1}. \end{cases} \tag{25}$$

(3) Go to (1) for the next time step.

The first condition in (25) seems unuseful since it is quickly modified by (17)–(20) in the next time step. However, it should be kept if at the present time step one needs to output the results. Similarly as for the steady case,

- the interface condition defined by (17), (21), and (25) will be called the *totally time-lagging overlapping/projection condition*;
- the interface condition defined by (18), (22), and (25) will be called the *partially time-lagging overlapping/projection condition*;
- the interface condition defined by (19), (23), and (25) will be called the *over-time-lagging overlapping/projection condition*;
- the interface condition defined by (20), (24), and (25) will be called the *time-accurate time-lagging overlapping/projection condition*.

3. ANALYSIS OF THE INTERFACE TREATMENTS

The stability, the convergence speed for steady state problems, the time accuracy for unsteady problems, and the conservation are factors to be taken into account for the interface treatment. Conservation is not the major concern of this paper, so we are satisfied with giving a simple remark in Appendix A.

All the analyses will be based on the linear equation $u_t + au_x = 0$ with $a > 0$. In the present paper we do not consider schemes with more than three points in space. Thus the (linear) difference equation (3) can be written as

$$a_{-1}\Delta u_{j-1}^{n+1} + a_0\Delta u_j^{n+1} + a_1\Delta u_{j+1}^{n+1} = b_{-1}u_{j-1}^n + b_0u_j^n + b_1u_{j+1}^n, \tag{26}$$

where the coefficients a_{-1}, a_0, a_1, b_{-1}, b_0, b_1, are scheme-dependent. The difference scheme (4) is supposed to have the same form as (3).

We will consider three typical schemes,

$$\begin{aligned}&\Delta u_j^n + \frac{1}{2}\beta\sigma^2a^2\left(\Delta u_{j+1}^n - 2\Delta u_j^n + \Delta u_{j-1}^n\right)\\&\quad = -\frac{1}{2}\sigma a\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{1}{2}\sigma^2a^2\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right), \qquad \text{Lerat scheme}\end{aligned} \tag{27}$$

$$\Delta u_j^n + \beta\sigma a\left(\Delta u_j^n - \Delta u_{j-1}^n\right) = -\sigma a\left(u_j^n - u_{j-1}^n\right), \qquad \text{implicit upwind scheme} \tag{28}$$

$$\begin{aligned}&\Delta u_j^n + \frac{1}{2}\sigma a\left(\Delta u_{j+1}^n - \Delta u_{j-1}^n\right) - \frac{1}{2}\sigma\mu\left(\Delta u_{j+1}^n - 2\Delta u_j^n + \Delta u_{j-1}^n\right)\\&\quad = -\frac{1}{2}\sigma a\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{1}{2}\sigma\mu\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right), \qquad \text{backward Euler scheme.}\end{aligned} \tag{29}$$

For convenience, we will denote $\lambda = \sigma a$. The Lerat scheme [12] is a centered dissipative scheme with a variable dissipation controlled by β with $\beta \leq -\frac{1}{2}$. For $\beta = -\frac{1}{2}$ the Lerat scheme is slightly dissipative and for $-1 \leq \beta \leq -\frac{1}{2}$ its dissipation is proportional to $|\beta|$.

The upwind scheme is inherently very dissipative.

The backward Euler scheme is not dissipative. In order to have some dissipation, we have added a numerical viscosity whose coefficient is $\mu \geq 0$. Thus the three schemes cover almost all the typical cases of three-point schemes.

3.1. *Stability Analysis*

The stability in the sense of Gustafsson, Kreiss, and Sundström (GKS) [6] of the totally time-lagging interface treatment was already analyzed in [14] where the following result was proved:

PROPOSITION 3.1. *Consider the totally time-lagging interface condition. When the scheme is dissipative*, *the interface tretment is always stable. When the scheme is not dissipative*, *then the interface treatment is GKS-stable for odd L_o and unstable for even L_o.*

The case of the nonstandard conditions on overlapping grids has not yet been studied. In order to apply the GKS theory to each nonstandard interface treatment, we must define an equivalent interface condition which is standard and which yields the same solution as the original treatment.

Now we transform the nonstandard interface treatment to equivalent standard ones. Let us concentrate on u_0. This can be simply obtained by combining the interface condition at $j = 0$ and the difference scheme (3) at $j = -1$, yielding a new relation which we call the equivalent interface condition.

Now rewrite the linear difference equation at $j = -1$ as

$$\left(a_{-1}\Delta u_{-2}^{n+1} + a_0\Delta u_{-1}^{n+1}\right) - \left(b_{-1}u_{-2}^n + b_0u_{-1}^n\right) = b_1u_0^n - a_1\Delta u_0^{n+1}. \tag{30}$$

Only the right-hand side of (30) is (directly) related to the interface condition. Introduce the interface conditions (6)–(8) into the right-hand side of (30) and subtract the resulting equation from (30). We obtain the equivalent interface conditions

$$(\text{E2} \times \text{I2})\ b_1u_0^n - a_1\Delta u_0^{n+1} = b_1v_{L_o-1}^n \tag{31}$$

$$(\text{E1} \times \text{I2})\ b_1u_0^n - a_1\Delta u_0^{n+1} = b_1v_{L_o-1}^{n-1} \tag{32}$$

$$(\text{E2} \times \text{I1})\ b_1u_0^n - a_1\Delta u_0^{n+1} = (b_1 - a_1)v_{L_o-1}^n + a_1v_{L_o-1}^{n-1} \tag{33}$$

which are used in the analysis.

Similarly, if we rewrite the difference equation at $j = 1$ for v as

$$\left(a_0\Delta v_1^{n+1} + a_1\Delta v_2^{n+1}\right) - \left(b_0v_1^n + b_1v_2^n\right) = b_{-1}v_0^n - a_{-1}\Delta v_0^{n+1} \tag{34}$$

then the corresponding equivalent interface conditions are

$$(\text{E2} \times \text{I2})\ b_{-1}v_0^n - a_{-1}\Delta v_0^{n+1} = b_{-1}u_{-L_o+1}^n \tag{35}$$

$$(\text{E1} \times \text{I2})\ b_{-1}v_0^n - a_{-1}\Delta v_0^{n+1} = b_{-1}u_{-L_o+1}^{n-1} \tag{36}$$

$$(\text{E2} \times \text{I1})\ b_{-1}v_0^n - a_{-1}\Delta v_0^{n+1} = (b_{-1} - a_{-1})u_{-L_o+1}^n + a_{-1}u_{-L_o+1}^{n-1}. \tag{37}$$

Let us briefly present the GKS-stability analysis suitable for interface problems with three-point schemes. It consists in considering the behaviour of normal mode solutions defined by

$$u_j^n = z^n\kappa_u^j\hat{u}_0,\ j \le 0; \qquad v_j^n = z^n\kappa_v^j\hat{v}_0,\ j \ge 0, \tag{38}$$

where $z \in \mathbf{C}$ and $|z| \geq 1$, and κ_u, κ_v are roots of the following characteristic equation related to the internal difference scheme (26):

$$(z-1)\left(a_{-1}\kappa_u^{-1} + a_0 + a_1\kappa_u^1\right) = b_{-1}\kappa_u^{-1} + b_0 + b_1\kappa_u^1. \tag{39}$$

For $|z| \geq 1$, the characteristic equation (39) has two roots κ_1 and κ_2 with $|\kappa_1| < 1$ and $|\kappa_2| > 1$. Only the root κ_2, which ensures the solution to be bounded when $j \to -\infty$, is used in the normal mode solution for u. The characteristic equation for v is the same as (39) and has therefore the same roots. Only the root κ_1, which ensures the solution to be bounded when $j \to \infty$, is used in the normal mode solution for v. Now the normal mode solutions (38) are more explicitly written as

$$u_j^n = z^n \kappa_2^j \hat{u}_0,\ j \leq 0; \qquad v_j^n = z^n \kappa_1^j \hat{v}_0,\ j \geq 0. \tag{40}$$

Inserting the normal mode solution (40) into the interface condition leads to

$$M(z, \kappa_1, \kappa_2)(\hat{u}_0, \hat{v}_0)^t = 0,$$

where $M(z, \kappa_1, \kappa_2)$ is a 2×2 matrix. The overlapping interface problem is called GKS-stable if and only if $\det M \neq 0$ for all $|z| \geq 1$. Equivalently, if the assumption $\det M(z) = 0$ leads to $\max(|z|) < 1$, then the problem is GKS-stable.

For convenience, let us define r, α_u, and α_v by

$$r = \frac{\kappa_1}{\kappa_2}, \qquad \alpha_u = \frac{a_1}{b_1}, \qquad \alpha_v = \frac{a_{-1}}{b_{-1}}.$$

For dissipative schemes, we have $|r| < 1$ when $|z| \geq 1$. See [14].

The determinants of the matrix M for various time-lagging (TL) interface conditions are found to be

$$\text{totally TL } \det M = z^2 - r^{L_o - 1} \tag{41}$$

$$\text{partially TL } \det M = [1 - \alpha_u(z-1)][1 - \alpha_v(z-1)] - r^{L_o - 1} \tag{42}$$

$$\text{over TL } \det M = [1 - \alpha_u(z-1)][1 - \alpha_v(z-1)]z^2 - r^{L_o - 1} \tag{43}$$

$$\text{time-accurate TL } \det M = \frac{1 - \alpha_u(z-1)}{(1-\alpha_u) + \alpha_u z^{-1}} \frac{1 - \alpha_v(z-1)}{(1-\alpha_v) + \alpha_v z^{-1}} - r^{L_o - 1}. \tag{44}$$

With the exception of the totally time-lagging condition, the other conditions have determinants depending on the coefficients of the scheme. Thus the stability conclusion will be scheme-dependent.

PROPOSITION 3.2. *For the Leart scheme* (27) *with* $\beta = -1$,

(a) *the interface problem with the totally time-lagging condition is unconditionally GKS-stable*;

(b) *the interface problem with the partially time-lagging condition is GKS-stable for* $\lambda > 1$ *and GKS-unstable for* $\lambda \leq 1$;

(c) *the interface problem with the over time-lagging condition is GKS-stable for* $\lambda > 1$ *and GKS-unstable for* $\lambda \leq 1$;

(d) *the interface problem with the time-accurate time-lagging condition is GKS-unstable if* λ *is very large and if* L_o *is finite.*

Proof. For $\beta = -1$, we have

$$\alpha_u = \frac{\lambda}{1-\lambda}, \qquad \alpha_v = -\frac{\lambda}{1+\lambda}.$$

Besides, the Lerat scheme is dissipative so that $|r| < 1$ for all $|z| \geq 1$.

(a) Since the Lerat scheme is dissipative, stability follows according to Proposition 3.1.

(b) Introducing the expressions for α_u and α_v into (42) and assuming $\det M = 0$, we find

$$\left[1 - \frac{\lambda}{1-\lambda}(z-1)\right]\left[1 + \frac{\lambda}{1+\lambda}(z-1)\right] = r^{L_o - 1}$$

from which we obtain the eigenvalues

$$z = \pm\sqrt{1 - (1 - r^{L_o - 1})\left(1 - \frac{1}{\lambda^2}\right)}.$$

Obviously, if $\lambda > 1$, then $|z| < 1$ since $|r| < 1$. As a result, the interface problem is GKS-stable for $\lambda > 1$. For $\lambda = 1$, we have $z = \pm 1$ so that the problem is GKS-unstable. A numerical procedure shows $|z| > 1$ when $\lambda < 1$. For example, if $\lambda = 0.5$, then $z = \pm 2$.

(c) Introducing the expressions for α_u and α_v into (43) and assuming $\det M = 0$, we find

$$z^2\left(z^2 - \frac{1}{\lambda^2}\right) = \left(1 - \frac{1}{\lambda^2}\right) r^{L_o - 1}.$$

The end of the proof can be done similarly as for case (b).

(d) Introducing the expression for α_u and α_v into (44) and assuming $\det M = 0$, we find that for $\lambda \to \infty$ and for finite values of L_o,

$$\frac{z^2}{(2 - 1/z)^2} \to 1.$$

One of the roots of the above equation is $-1 - \sqrt{2}$ so that the interface problem is unstable for very large λ. ■

Remark. (a) It is surprising that the partially and over time-lagging condition are GKS-stable for $\lambda > 1$ and unstable for small λ. One would prefer to use this condition since for implicit schemes we use $\lambda > 1$. However, if we solve a system such as the Euler equations in gas dynamics, different eigenvalues of the Jacobian matrix correspond to different values of λ. Even if the largest λ is greater than 1, the smallest one would be still less than 1 so that the problem would be unstable.

(b) Numerical experiments show that the interface treatment with the time-accurate time-lagging condition is stable with small λ or large L_o.

PROPOSITION 3.3. *For the Lerat scheme* (27) *with* $\beta = -\frac{1}{2}$,

(a) *the interface problem with the totally time-lagging condition is unconditionally GKS-stable*;

(b) *the interface problem with the partially time-lagging condition is GKS-unstable for large* λ.

Proof. (a) The Lerat scheme is dissipative for $\beta = -\frac{1}{2}$ so that GKS-stability follows according to Proposition 3.1.

(b) For $\beta = -\frac{1}{2}$, we have

$$\alpha_u = \frac{1}{2}\frac{\lambda}{1-\lambda}, \qquad \alpha_v = -\frac{1}{2}\frac{\lambda}{1+\lambda}.$$

Besides, the Lerat scheme is dissipative so that $|r| < 1$ for all $|z| \geq 1$. Introducing the expressions for α_u and α_v into (42) and assuming $\det M = 0$, we find

$$\left[1 - \frac{1}{2}\frac{\lambda}{1-\lambda}(z-1)\right]\left[1 + \frac{1}{2}\frac{\lambda}{1+\lambda}(z-1)\right] = r^{L_o - 1}$$

from which we obtain the eigenvalues

$$z = 1 \pm 2\sqrt{1 - (1 - r^{L_o-1})\left(1 - \frac{1}{\lambda^2}\right)}.$$

For $\lambda \to \infty$, $z = 1 \pm 2\sqrt{r^{L_o-1}}$ so that $\max(|z|) > 1$. As a result, the problem is GKS-unstable for large λ. ■

PROPOSITION 3.4. *For the upwind scheme* (28), *the interface problem is GKS-stable for all the interface conditions* (*totally time-lagging*, *partially time-lagging*, *over-time-lagging*, *and time-accurate time-lagging*).

Proof. The upwind scheme involves only two space points in the scalar case and does not need the condition for u_0. The condition for v_0 can be viewed as a pure Dirichlet condition so that stability follows from solvability. ■

The stability analysis can be done more in detail. However, the above analysis is sufficient enough to yield the following conclusion:

Conclusion 1. Only the totally time-lagging condition ensures GKS-stability (scheme-independent provided that the scheme be dissipative). The others are only conditionally stable or have an unstable stability range.

3.2. *Linear Convergence Study for the Steady State Problem*

A preliminary convergence study was already conducted in [21], where the quantitative analysis was limited to three-point dissipative schemes and to the totally time-lagging interface condition. Here we want to consider other kinds of schemes and the nonstandard time-lagging interface conditions.

3.2.1. Method for analysis. Like [8, 21], the convergence rate can be studied through normal mode analysis or eigenvalue analysis. The former is suitable for qualitative analysis and the latter for quantitative analysis. Both are relatively easy for the standard interface condition. But for the nonstandard interface treatment, we should use the equivalent standard ones. Here we are only interested in quantitative analysis.

Obviously, the equivalent interface condition involves two time levels for the partially time-lagging condition (Eqs. (31), (35)), but three levels in time for the over-time-lagging condition (Eqs. (32), (36)) and time-accurate time-lagging condition (Eqs. (33), (37)). For the three-level case, caution must be paid for the eigenvalue analysis.

Let us begin with the two-level case. In the multidomain case, the eigenvalue problem is obtained by introducing $u_j^n = z^n \phi_j^{(u)}$ and $v_j^n = z^n \phi_j^{(v)}$ ($z \in \mathbf{C}$) into the difference equations and the equivalent standard interface condition, yielding $zM_1 Y = M_2 Y$ where M_1 and M_2 are two real matrices, and Y is a column vector of components $\phi_j^{(u)}$, $j = -1, -2, \ldots, -N_u$ and $\phi_j^{(v)}$, $j = 1, 2, \ldots, N_v$. There are in total $N_u + N_v$ eigenvalues z_σ. The eigenvalue problem for the single domain case can be defined similarly. Let $P(z) = zM_1 - M_2$. Let $U = \{z : \det P(z) = 0\}$. If U lies strictly inside the unit circle, then convergence is guaranteed, and the convergence rate is given by $\rho = \max\{|z|, z \in U\}$. For convenience, we use ρ_o to denote ρ for the overlapping treatment, and ρ_s for the corresponding single domain treatment.

The three-level case can be similarly done with minor adaptation. It consists in introducing an intermediate variable as done in the usual Cauchy-stability analysis for three-level schemes. Just for illustration, consider (32). Now introduce an additional variable $w^n = v_{L_o-1}^{n-1}$ to the condition (32) so that the latter reduces to

$$b_1^{(u)} u_0^n - a_1^{(u)} \Delta u_0^{n+1} = b_{-1}^{(v)} w^n. \tag{45}$$

The definition for the additional variable can be rewritten as

$$w^{n+1} = v_{L_o-1}^n. \tag{46}$$

The complete system, now involving only two time levels and defined by the interior difference equations, interface conditions rewritten as (45), and the additional relation (46), can be used to do convergence study in a similar way as above.

To reach a prescribed degree of convergence, the number of time iterations n_c and the total CPU time t_c can be estimated by

$$n_c = \frac{C'}{\ln \rho}, \qquad t_c = \frac{C''}{\ln \rho} N_o,$$

where C' and C'' are constants depending only on the required convergence degree, and N_o is the total number of mesh points. The number of additional points due to overlapping is factored into the calculation of t_c.

Similarly as in [21], we can use the overlapping efficiency

$$\epsilon_o = \frac{t_c(\text{overlapping}) - t_c(\text{single domain})}{t_c(\text{single domain})}$$

to measure the performance of the overlapping treatment. The overlapping efficiency measures the change of CPU time by overlapping with respect to the corresponding single domain treatment. In comparison with the single domain treatment, the overlapping case consumes $-100\epsilon_o\%$ more time when $\epsilon_o < 0$ while it consumes $100\epsilon_o\%$ less time when $\epsilon_o > 0$.

3.2.2. Influence of the dissipation of the interior schemes. In all the cases we use CFL = 6. The conclusions with other CFL numbers are similar. Here we only consider the totally time-lagging conditions. The results to be displayed are based on the eigenvalue analysis.

TABLE I

Influence of the Dissipation on the Convergence Speed

	$\beta=-\frac{1}{2}$		$\beta=-1$	
L_o	ρ_o	ϵ_o	ρ_o	ϵ_o
2	0.904	0.062	0.84	−1.152
4	0.909	−0.097	0.686	−0.034
6	0.904	−0.085	0.625	0.13
8	0.904	−0.132	0.628	0.0825
10	0.904	−0.19	0.631	0.0329
12	0.902	−0.207	0.634	−0.018
14	0.903	−0.223	0.637	−0.072
16	0.901	−0.222	0.641	−0.126
	$\rho_s=0.905$		$\rho_s=0.69$	

Note. Lerat scheme with CFL = 6.

Table I gives the results for $\beta=-\frac{1}{2}$ and $\beta=-1$. The case of $\beta=-1$ was already studied in [21]. In this case there is an optimal overlapping width which turns out to be equal to the CFL number. When $\beta=-\frac{1}{2}$, for which the scheme is only slightly dissipative, the overlapping efficiency is a decreasing function of the overlapping width so that the optimal overlapping width is $L_o=2$.

Table II shows the results for the implicit upwind scheme. We remark that for upwind schemes the overlapping efficiency is a decreasing function of the overlapping width. As a result, the optimal overlapping width is $L_o=2$ (the shortest overlapping width).

Table III displays the results for the Backward Euler scheme with and without artificial dissipation (controlled by μ). We remark that when $\mu=0$ for which the scheme is nondissipative, ρ_o is larger than 1 for all (even) L_o. When $\mu=0.1$ for which the scheme is slightly dissipative and makes the problem slightly elliptic, the problem converges, the optimal overlapping width is $L_o=2$.

3.2.3. Convergence study for nonstandard treatments. Let us begin with the Lerat scheme.

TABLE II

Convergence Speed for the Implicit Upwind Scheme with CFL = 6

	$\beta=0.5$		$\beta=1$	
L_o	ρ_o	ϵ_o	ρ_o	ϵ_o
2	0.636	−0.064	0.217	0.005
4	0.639	−0.126	0.216	−0.042
6	0.643	−0.200	0.227	−0.128
8	0.63	−0.199	0.225	−0.172
10	0.63	−0.195	0.222	−0.213
12	0.63	−0.266	0.23	−0.29
14	0.63	−0.288	0.25	−0.41
16	0.63	−0.375	0.25	−0.48
	$\rho_s=0.618$		$\rho_s=0.219$	

TABLE III

Convergence Speed for the Backwind Euler Scheme with CFL = 6

	$\mu = 0$		$\mu = 0.1$	
L_o	ρ_o	ϵ_o	ρ_o	ϵ_o
2	1.15	$-\infty$	0.943	0.011
4	1.067	$-\infty$	0.943	−0.043
6	1.14	$-\infty$	0.944	−0.1
8	1.09	$-\infty$	0.946	−0.205
10	1.087	$-\infty$	0.946	−0.238
12	1.085	$-\infty$	0.947	−0.335
14	1.065	$-\infty$	0.945	−0.330
16	1.10	$-\infty$	0.946	−0.41
	$\rho_s = 1$		$\rho_s = 0.944$	

First consider the partially time-lagging condition. The results for three different β are displayed in Table IV. When $\beta = -\frac{1}{2}$, ρ_o is larger than 1 for all L_o. As a result, the interface treatment does not allow for convergence (in fact unstable). When $\beta = -0.6$, ρ_o is larger than 1 for small overlapping width and is below 1 for $L_o \geq 6$. Thus there is a cut-off overlapping width allowing for convergence. When $\beta = -1$, the interface problem is always convergent.

Now consider the over-time-lagging condition. The results for three different β are displayed in Table V. We remark that the results are very similar to those with the partially time-lagging condition. The difference is minor. For example, when $\beta = -0.6$, the cut-off overlapping length for the over time-lagging condition is 2 while in the case of the partially time-lagging condition the cut-off overlapping length is $L_o = 4$.

Finally consider the time-accurate time-lagging condition. The results for three different β are displayed in Table VI. The interface treatment is nonconvergent not only for $\beta = -0.5$ and $\beta = -0.6$, but also for $\beta = -1$ when L_o is small. Thus the convergence property of the time-accurate time-lagging condition is very poor.

Now we consider the nonstandard interface conditions for the upwind scheme. The results for $\beta = 0.75$ and $\beta = 1$ are respectively displayed in Tables VII and VIII. For $\beta = 0.75$, the nonstandard treatments behave similarly and they have a convergence speed lower than the totally time-lagging treatment. In any case the overlapping grid method has a convergence speed lower than the single domain case. For $\beta = 1$, the nonstandard treatments have a

TABLE IV

Convergence Speed for the Partially Time-Lagging Condition

L_o	$\rho_o(\beta = -\frac{1}{2})$	$\rho_o(\beta = -0.6)$	$\rho_o(\beta = -1.0)$
2	2.08	1.62	0.8466
4	1.335	1.0073	0.70
6	1.33	0.944	0.626
8	1.33	0.944	0.629
10	1.33	0.944	0.632
	$\rho_s = 0.905$	$\rho_s = 0.66$	$\rho_s = 0.69$

Note. Lerat scheme with CFL = 6.

TABLE V

Convergence Speed for the Over Time-Lagging Condition

L_o	$\rho_o(\beta=-\frac{1}{2})$	$\rho_o(\beta=-0.6)$	$\rho_o(\beta=-1.0)$
2	1.57	1.322	0.920
4	1.33	0.996	0.844
6	1.33	0.944	0.781
8	1.33	0.944	0.751
10	1.33	0.944	0.732
	$\rho_s=0.905$	$\rho_s=0.66$	$\rho_s=0.69$

Note. Lerat scheme with CFL = 6.

TABLE VI

Convergence Speed for the Time-Accurate Time-Lagging Condition

L_o	$\rho_o(\beta=-\frac{1}{2})$	$\rho_o(\beta=-0.6)$	$\rho_o(\beta=-1.0)$
2	3.047	2.649	1.872
4	1.896	1.786	1.408
6	1.471	1.182	0.990
8	1.33	1.080	0.801
10	1.33	1.003	0.780
	$\rho_s=0.905$	$\rho_s=0.66$	$\rho_s=0.69$

Note. Lerat scheme with CFL = 6.

TABLE VII

Convergence Speed for Various Interface Treatments

L_o	Totally TL	Partially TL	Over TL	Time-accurate TL
2	0.165	0.334	0.334	0.334
4	0.167	0.334	0.334	0.334
6	0.163	0.334	0.334	0.334
8	0.165	0.334	0.334	0.334
10	0.162	0.334	0.334	0.334
		$\rho_s=0.137$		

Note. Implicit upwind scheme with CFL = 6 and $\beta=0.75$.

TABLE VIII

Convergence Speed for Various Interface Treatments

L_o	Totally TL	Partially TL	Over TL	Time-accurate TL
2	0.217	0.214	0.226	0.226
4	0.216	0.219	0.206	0.228
6	0.227	0.226	0.216	0.231
8	0.225	0.221	0.217	0.238
10	0.222	0.231	0.220	0.246
		$\rho_s=0.219$		

Note. Implicit upwind scheme with CFL = 6 and $\beta=1$.

similar convergence speed as the totally time-lagging treatment. The convergence speed of the overlapping treatment is almost the same as the single domain case.

3.2.4. Summary of conclusions. In [21], we have obtained the following surprising conclusion for the Lerat scheme:

Conclusion 2. For the Lerat scheme with $\beta = -1$, the overlapping treatment with the totally time-lagging condition has an optimal overlapping width which is close to the CFL number. Besides, the overlapping efficiency is positive near the optimal overlapping width.

We are interested in whether such a result has some generality. The above analysis leads to the following new conclusions.

Conclusion 3. For (two-point) upwind schemes, the overlapping treatment with the totally time-lagging condition has its optimal overlapping width equal to $L_o = 2$. As a result, the best convergence rate corresponds to the shortest overlapping width. Besides, the convergence speed is generally lower than the single domain treatment.

Conclusion 4. When the difference schemes are nondissipative, the overlapping scheme is nonconvergent or unstable.

Conclusion 5. When the interface treatment is defined by the nonstandard conditions, the overlapping scheme has a limited stability domain and the convergence speed is generally lower than the standard treatment.

Thus *the dissipation of the difference schemes, a correct time-lagging, and a correct choice of the overlapping width are very important for a good convergence rate*.

3.3. *Accuracy of the Unsteady Problem*

3.3.1. Accuracy analysis. First consider the standard interface treatment (totally time-lagging condition). The key point of the overlapping/projection interface treatment is based on the finite speed of a numerical wave. Consider a time accurate scheme for the transport equation $u_t + au_x = 0$. If the scheme has a pth order accuracy, then the numerical solution $\bar{u}$ satisfies the modified (or equivalent) equation

$$\bar{u}_t + a\bar{u}_x = C\delta t^p + o[\delta t^p], \tag{47}$$

where C is finite and depends on the specific scheme.

At the time step n, let the numerical solution be given by $\bar{u}^n(x)$. Then the numerical solution at $n+1$ becomes

$$\bar{u}^{n+1} = \bar{u}^n(x - a\delta t) + R \tag{48}$$

with $R = \int_x^{x-a\delta t}(C\delta t^p + o[\delta t^p])d\tau = O[\delta t^{p+1}]$.

Equation (48) clearly shows that a numerical wave, within the error of the order of the truncation error of the scheme, travels at a distance equal to $a\delta t$ at each time step. Since the mesh size is δx, this distance is equal to $a\delta t/\delta x$ in terms of the number of mesh points (ignoring the discussion when $a\delta t/\delta x$ is not an integer). By definition, $a\delta t/\delta x = \text{CFL}$. Thus we have obtained the following conclusion:

Conclusion 6. A numerical wave, within the error of the order of the truncation error of the scheme, travels at a distance equal to CFL in terms of the number of mesh points. As a result, the overlapping/projection method using the time-lagging condition maintains the order of accuracy of the interior scheme.

Now consider the nonstandard interface conditions. We only consider the partially and time-accurate time-lagging conditions. The over-time-lagging condition is obviously less interesting for unsteady problems.

Let us consider scheme (4) for the transport equation $u_t + au_x = 0$ and begin with the partially time-lagging condition. Starting from the instant n, let e_j be the difference of the time accurate solution of the problem using for example a time accurate interface condition (or a single domain treatment) and the numerical solution with the partially time-lagging condition. Obviously, the explicit stages of both treatments are the same since the value at n is not lagged in the partially time-lagging treatment. As a result, e_j satisfies the equation[1]

$$Le_j = 0,\ j \geq 1; \qquad e_0 = \Delta u_{-L_o+1},$$

where $e_0 = \Delta u_{-L_o+1}$ is the error at the interface caused by time lagging.

Consider for instance the Lerat scheme, for which the equation for e_j is given by

$$e_j + \frac{1}{2}\beta\sigma^2 a^2 (e_{j+1} - 2e_j + e_{j-1}) = 0.$$

The general solution of the above equation is

$$e_j = c_1\kappa_1^j + c_2\kappa_2^j,$$

where κ_1 and κ_2 are the roots of the characteristic equation

$$\kappa + \frac{1}{2}\beta\sigma^2 a^2(\kappa^2 - 2\kappa + 1) = 0.$$

The root with an absolute value larger than 1 should be excluded since at $j \to \infty$ the solution remains bounded. The acceptable solution is found to be

$$e_j = \Delta u_{-L_o+1}\left(1 - \frac{1}{\beta\sigma^2 a^2} - \sqrt{\frac{1}{\beta^2\sigma^4 a^4} - \frac{2}{\beta\sigma^2 a^2}}\right)^j$$

which reduces to

$$e_j = \Delta u_{-L_o+1}\left(1 - \frac{\sqrt{2}}{\sigma a}\right)^j$$

for $\beta = -1$ and for sufficiently large σa (=CFL). At $j = \sigma a$ (=CFL),

$$e_j = \Delta u_{-L_o+1}\left(1 - \frac{\sqrt{2}}{\sigma a}\right)^{\sigma a} \to e^{-\sqrt{2}}\Delta u_{-L_o+1} \approx 0.24\Delta u_{-L_o+1} = 0.24[\delta t] \qquad (49)$$

[1] In the case of the totally time-lagging condition, where the interface value at n also lags in time, the explicit stage is also affected so that $Le_j \neq 0$ at $j = 1$. In this case, it is substantially more complicated to estimate e_j than to perform an error analysis using the modified equation.

for sufficiently large CFL. As a result, the error was reduced to 24% of the original one at a distance $j = \text{CFL}$. Besides, $|e_j|$ is a decreasing function of j.

Similarly, for the upwind scheme with $\beta = 1$,

$$e_j = \Delta u_{-L_o+1}\left(\frac{(1/2)\sigma a}{1+(1/2)\sigma a}\right)^j.$$

At $j = \sigma a$ (=CFL),

$$e_j = \Delta u_{-L_o+1}\left(\frac{(1/2)\sigma a}{1+(1/2)\sigma a}\right)^{\sigma a} \to e^{-2}\Delta u_{-L_o+1} \approx 0.14\Delta u_{-L_o+1} = 0.14[\delta t] \quad (50)$$

for sufficiently large CFL. As a result, the error was reduced to 14% of the original one at a distance $j =$ CFL. The absolute error $|e_j|$ is a decreasing function of j.

Now consider the time-accurate time-lagging condition, for which e_j satisfies the equation

$$Le_j = 0,\ j \geq 1; \qquad e_0 = \Delta u^{n+1}_{-L_o+1} - \Delta u^{n}_{-L_o+1} = O[\delta t^2],$$

where $e_0 = \Delta u^{n+1}_{-L_o+1} - \Delta u^{n}_{-L_o+1}$ is the error caused by time lagging.

Similarly as above, we obtain

$$e_j \approx 0.24\left(\Delta u^{n+1}_{-L_o+1} - \Delta u^{n}_{-L_o+1}\right) = 0.24O[\delta t^2] \quad (51)$$

for the Lerat scheme with $\beta = -1$ and

$$e_j \approx 0.14\left(\Delta u^{n+1}_{-L_o+1} - \Delta u^{n}_{-L_o+1}\right) = 0.14O[\delta t^2] \quad (52)$$

for the upwind scheme with $\beta = -1$.

Conclusion 7. For the partially time-lagging condition, the time lagging induces a first order perturbation at a distance $j = \text{CFL}$ away from the interface and for the time-accurate time-lagging condition, this perturbation is second order. The errors at $j < \text{CFL}$ caused by time-lagging are larger than the error at $j = \text{CFL}$ (see (49), (50), (51), and (52)) and will be eliminated by the projection step. The errors at $j > \text{CFL}$ due to time-lagging will be smaller than those at $j = \text{CFL}$.

3.3.2. Numerical test with time refinement. The above accuracy analysis does not give information for the overall order of accuracy. In order to check whether the overlapping/projection interface treatment maintains the overall order of accuracy of the interior difference scheme, we use the Lerat scheme (which is second order accurate) to solve the inviscid Burgers equation

$$w_t + \left(\frac{1}{2}w^2\right)_x = 0, \qquad 0 < x < 1$$

with the initial data

$$w(x,0) = x.$$

The exact solution is

$$w(x,t) = \frac{x}{1+t}.$$

In each computation, we fix the CFL number. Time refinement is done in order to measure the order of accuracy. The l_2-error e is defined by

$$e = \sqrt{\frac{\sum_j |e_j|^2}{N}},$$

where e_j is the difference between the numerical solution and the exact solution at j, and N is the total number of mesh points.

Three cases are considered: single domain computation (for which the error is denoted e_{SD}), bidomain computation, and 4-domain computations. In the case of multidomain computations, we test all the four possibilities:

(1) totally time-lagging overlapping/projection condition for which the error is denoted e_{TTL};

(2) partially time-lagging overlapping/projection condition for which the error is denoted e_{PTL};

(3) over-time-lagging overlapping/projection condition for which the error is denoted e_{OTL};

(4) time-accurate time-lagging overlapping/projection condition for which the error is denoted e_{TATL}.

Table IX displays the results for bidomain computation with CFL = 6. We remark that similarly as in the single domain case, the multidomain computations maintain the overall second order accuracy independently of the choice of the interface conditions. The over-time-lagging condition produces an error obviously larger than the other conditions.

Table X displays the results for bidomain computation with CFL = 10. The results are very similar to the case of CFL = 6. The numerical errors are slightly larger than the case with CFL = 6. The multidomain computation yields results very close to the single domain computation except when the over-time-lagging condition is used.

Tables XI and XII display the results for 4-domain computation with CFL = 6 and CFL = 10. Obviously, the multidomain computation with the time-accurate time-lagging condition produces a result very close to the single domain case. When the totally and partially time-lagging conditions are used, the numerical errors are obviously larger than the

TABLE IX

l_2-Error at $t = 3$ of Bidomain Computations Compared to Single Domain Computation with CFL = 6

δt	e_{SD}	e_{TTL}	e_{PTL}	e_{OTL}	e_{TATL}
0.1	1.506E-04	1.308E-04	1.335E-04	1.321E-04	1.357E-04
0.075	9.519E-05	9.702E-05	9.841E-05	1.078E-04	9.168E-05
0.05	5.052E-05	4.999E-05	5.088E-05	5.390E-05	4.796E-05
0.025	1.247E-05	1.273E-05	1.291E-05	1.460E-05	1.216E-05
0.0125	4.322E-06	4.867E-06	4.936E-06	6.133E-06	4.268E-06

TABLE X
l_2-Error at t = 3 of Bidomain Computations Compared to Single Domain Computation with CFL = 10

δt	e_{SD}	e_{TTL}	e_{PTL}	e_{OTL}	e_{TATL}
0.1	1.609E-04	1.908E-04	2.093E-04	3.298E-04	1.396E-04
0.075	1.136E-04	9.403E-05	9.768E-05	1.119E-04	8.683E-05
0.05	5.830E-05	6.221E-05	6.296E-05	7.146E-05	5.811E-05
0.025	1.386E-05	1.460E-05	1.493E-05	1.775E-05	1.326E-05
0.0125	4.356E-06	4.892E-06	4.910E-06	6.448E-06	4.267E-06

single domain result, but the results are still acceptable. The over-time-lagging condition yields the largest error.

Conclusion 8. The time-accurate time-lagging condition is as accurate as the single domain treatment in any case. The over-time-lagging condition produces the largest error in any case.

4. NUMERICAL EXPERIMENTS FOR THE COMPRESSIBLE EULER EQUATIONS

4.1. *Physical Problem and Numerical Methods*

4.1.1. Physical problem. Present calculations are based on the two-dimensional Euler equations

$$w_t + f(w)_x + g(w)_y = 0 \tag{53}$$

with

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \qquad f(w) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho E + p)u \end{pmatrix}, \qquad g(w) = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ (\rho E + p)v \end{pmatrix}$$

and assuming a perfect gas law,

$$p = (\gamma - 1)\rho\left[E - \frac{1}{2}(u^2 + v^2)\right],$$

TABLE XI
l_2-Error at t = 3 of 4-Domain Computations Compared to Single Domain Computation with CFL = 6

δt	e_{SD}	e_{TTL}	e_{PTL}	e_{OTL}	e_{TATL}
0.1	1.506E-04	1.247E-04	1.342E-04	1.496E-04	1.190E-04
0.075	9.519E-05	9.421E-05	9.973E-05	1.207E-04	8.199E-05
0.05	5.052E-05	5.125E-05	5.415E-05	6.610E-05	4.433E-05
0.025	1.247E-05	1.531E-05	1.620E-05	2.396E-05	1.164E-05
0.0125	4.322E-06	7.183E-06	7.476E-06	1.221E-05	4.189E-06

TABLE XII
l_2-Error at $t = 3$ of 4-Domain Computations Compared to Single Domain Computation with CFL = 10

δt	e_{SD}	e_{TTL}	e_{PTL}	e_{OTL}	e_{TATL}
0.075	1.136E-04	9.493E-05	1.034E-04	1.286E-04	7.893E-05
0.05	5.830E-05	6.428E-05	6.795E-05	8.837E-05	5.185E-05
0.025	1.386E-05	1.800E-05	1.930E-05	2.911E-05	1.244E-05
0.0125	4.356E-06	7.585E-06	7.957E-06	1.339E-05	4.143E-06

where ρ, p, u, v, and E denote respectively the density, the pressure, the velocity Cartesian components, and the total energy; $\gamma = 1.4$ is the specific heat ratio.

We always consider a bidimensional symmetric flow around a fixed (steady) or oscillating (unsteady) NACA0012 airfoil.

The steady flow is either transonic with a free-stream Mach number $M_\infty = 0.85$, or subsonic with a free-stream Mach number $M_\infty = 0.536$.

The unsteady flow we consider is produced by a horizontal oscillation of the NACA0012 airfoil [13]. The free-stream Mach number is maintained to be constant $M_\infty = 0.536$. The airfoil oscillates horizontally with a velocity given by

$$u_a = -M_0 a_\infty \sin(kt),$$

where $M_0 = 0.327$, a_∞ is the sound speed at infinity, and $k = 0.185$ is the reduced frequency.

4.1.2. Difference approximations. System (53) is approximated by two different implicit schemes.

The first is the implicit centered scheme of Lerat [12] of second-order accuracy,

$$\begin{aligned}
\Delta\tilde{w}_{i+1/2,j} &= -\Delta t(\delta_1 f/\Delta x + \mu_1\mu_2\delta_2 g/\Delta y)^n_{i+1/2,j} \\
\tilde{f}_{i+1/2,j} &= \left[(\mu_1 f)^n + \frac{1}{2}(\mu_1 A)^n \Delta\tilde{w}\right]_{i+1/2,j} \\
\Delta\tilde{\tilde{w}}_{i,j+1/2} &= -\Delta t(\mu_1\mu_2\delta_1 f/\Delta x + \delta_2 g/\Delta y)^n_{i,j+1/2} \\
\tilde{\tilde{g}}_{i,j+1/2} &= \left[(\mu_2 g)^n + \frac{1}{2}(\mu_2 B)^n \Delta\tilde{\tilde{w}}\right]_{i,j+1/2} \\
\Delta w^{expl}_{i,j} &= -\Delta t(\delta_1 \tilde{f}/\Delta x + \delta_2\tilde{\tilde{g}}/\Delta y)_{i,j} \\
\Delta w^*_{i,j} - \frac{1}{2}(\Delta t/\Delta x)^2\delta_1\left[(\mu_1 A^n)^2\delta_1(\Delta w^*)\right]_{i,j} &= \Delta w^{expl}_{i,j} \\
\Delta w_{i,j} - \frac{1}{2}(\Delta t/\Delta y)^2\delta_2\left[(\mu_2 B^n)^2\delta_2(\Delta w)\right]_{i,j} &= \Delta w^*_{i,j} \\
w^{n+1}_{i,j} &= w^n_{i,j} + \Delta w_{i,j},
\end{aligned}$$

where $A = df(w)/dw$ and $B = dg(w)/dw$ are the Jacobian matrices, and δ_s, μ_s for $s = 1, 2$ are spatial operators such that for $\phi_{i,j}$ defined at the mesh point $x = i\Delta x$ and $y = j\Delta y$,

$$(\delta_1\phi)_{i,j} = \phi_{i+1/2,j} - \phi_{i-1/2,j}, \qquad (\delta_2\phi)_{i,j} = \phi_{i,j+1/2} - \phi_{i,j-1/2}$$
$$(\mu_1\phi)_{i,j} = \frac{1}{2}(\phi_{i+1/2,j} + \phi_{i-1/2,j}), \qquad (\mu_2\phi)_{i,j} = \frac{1}{2}(\phi_{i,j+1/2} + \phi_{i,j-1/2}).$$

This scheme is always linearly stable in L_2 and dissipative except when A or B is singular. It involves 3×3 points at level n and 5 points at level $n+1$ and leads to the solution of block-tridiagonal linear systems. Due to its own dissipative properties, this scheme is always used without artificial viscosity as in [12].

The second one is the implicit version of Roe's upwind scheme,

$$\begin{aligned}
\tilde{f}_{i+1/2,j} &= \left[\mu_1 f + \frac{1}{2}\left|A^{(R)}\right|\delta_1 w\right]^n_{i+1/2,j} \\
\tilde{g}_{i,j+1/2} &= \left[\mu_2 g + \frac{1}{2}\left|B^{(R)}\right|\delta_2 w\right]^n_{i,j+1/2} \\
\Delta w^{expl}_{i,j} &= -\Delta t(\delta_1 \tilde{f}/\Delta x + \delta_2 \tilde{g}/\Delta y)_{i,j}
\end{aligned}$$

$$\Delta w^*_{i,j} + \frac{1}{2}(\Delta t/\Delta x)\left\{\delta_1\left[\left|A^{(R)}\right|^n \delta_1(\Delta w^*)\right]_{i,j} + \delta_1\left[A^n \mu_1(\Delta w^*)\right]_{i,j}\right\} = \Delta w^{expl}_{i,j}$$

$$\Delta w_{i,j} + \frac{1}{2}(\Delta t/\Delta y)\left\{\delta_2\left[\left|B^{(R)}\right|^n \delta_2(\Delta w)\right]_{i,j} + \delta_2\left[B^n \mu_2(\Delta w)\right]_{i,j}\right\} = \Delta w^*_{i,j}$$

$$w^{n+1}_{i,j} = w^n_{i,j} + \Delta w_{i,j},$$

where $A^{(R)}$ and $B^{(R)}$ are the well-known Roe averages of the Jacobians $A = df(w)/dw$ and $B = dg(w)/dw$.

The Lerat scheme will be used for both steady and unsteady computations. The Roe scheme will be used only for steady state computations.

4.1.3. Implementation. Both schemes have been implemented on a structured mesh by using a finite-volume formulation and with the movement of the grid taken into account conservatively.

On a rigid wall, the slip condition is applied and the pressure is computed from a linear combination of the discrete form of the x- and y-momentum equations to obtain a conservative approximation of the normal momentum equation. On an external subsonic inflow boundary, we prescribe the free-stream direction, the entropy, and the enthalpy. On an external subsonic outflow boundary we prescribe the pressure.

Domain spliting is done automatically. A structured grid can be split into $n_x \times n_y$ subdomains with an overlapping of L_o mesh points normal to each interface. Since an approximate factorization is used, the computation of the implicit part in each direction is similar to a one-dimensional problem. As a result, the interface conditions for both steady and unsteady problems can be straightforwardly realized as in the one-dimensional case.

A coarse grid and a fine grid will be considered. In the single domain case, the coarse grid is a 124×25 C-mesh, and the fine grid is a 247×49 C-mesh based on the coarse grid. In multidomain computations, the computational domain is split into 2, 4, or 8 subdomains. For the present mesh where the grid number in the second direction (which is equal to 25 or 49) is small, we have only used horizontal splitting. A splitting in both directions will certainly increase the communication time in comparison with a splitting in one direction. But this negative aspect will become less significant when the mesh size is large as in the case of real applications.

For steady state computations, we use the totally time-lagging interface conditions. For the unsteady problem, we have used both of the totally time-lagging and time-accurate

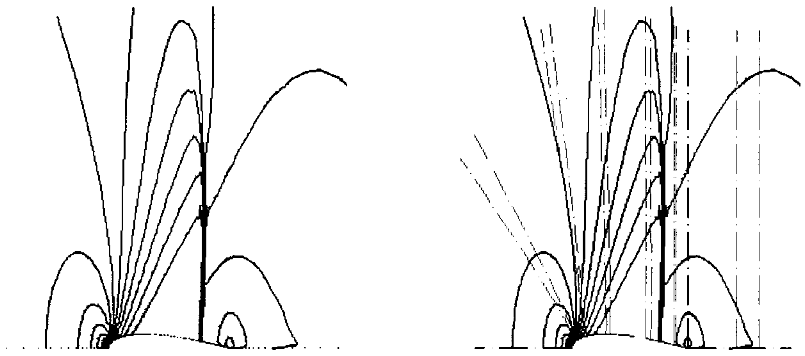


FIG. 3. Mach contours for the steady state problem. $M_\infty = 0.85$, CFL = 5. Left, single domain. Right, 8 domains with $L_o = 4$. Lerat scheme on the fine grid.

time-lagging overlapping/projection interface treatments with $L_o = 2(\mathrm{CFL} + 1)$. Both lead to solutions very close to the single domain computation. As a result, we will only display the results obtained with the totally time-lagging condition compared with the single domain results.

4.2. *Sequential Computations*

The sequential computation is for the purpose of testing the convergence speed and time accuracy of the overlapping grid method.

4.2.1. Steady problem. Here we consider the transonic flow case with $M_\infty = 0.85$ computed with the Lerat scheme and the Roe scheme. Only the results obtained with the fine grid will be displayed.

The Mach contours computed with the Lerat scheme and CFL = 5 are displayed in Fig. 3. Here the 8-domain results are compared to the single domain computation. In the multidomain case, the interfaces are shown through dashed lines. Each pair of closed lines defines the boundaries of an overlap. For the Lerat scheme with CFL = 5, the convergence curves (the time evolution of the root-mean-square residual R_2 for the discrete density equation) of the overlapping computation compared with the single domain one are displayed in Figs. 4–6. When there are only two subdomains, the multidomain treatment converges as rapidly as the single domain one even with a small overlapping width. But for a large number of subdomains, the convergence speed of the overlapping treatment (with a time-lagging interface condition) is slightly smaller than the single domain one if the overlapping width is small. When the overlapping width reaches the CFL number, both treatments almost have the same convergence speed.

The Mach contours computed with the Lerat scheme and CFL = 10 are displayed in Fig. 7. Here the 8-domain results are compared to the single domain computation. In the multidomain case, the interfaces are shown through dashed lines as before. For the Lerat scheme with CFL = 10, the convergence curves of the overlapping computation compared with the single domain one are displayed in Figs. 8–10. When there are only two subdomains, the multidomain treatment converges as rapidly as the single domain one even with a small overlapping width. But for a large number of subdomains, the convergence speed of the overlapping treatment (with a time-lagging interface condition) is significantly smaller than

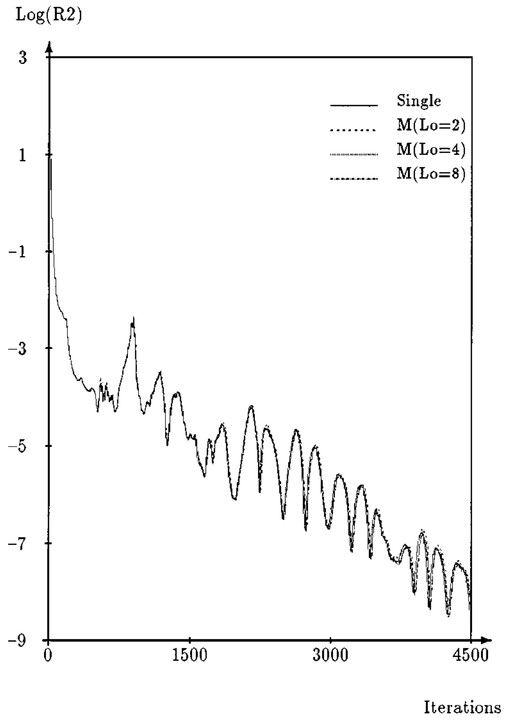


FIG. 4. Dependence of the convergence histories on the overlapping width for the steady state problem with 2 subdomains. $M_\infty = 0.85$, CFL = 5. Lerat scheme on the fine grid.

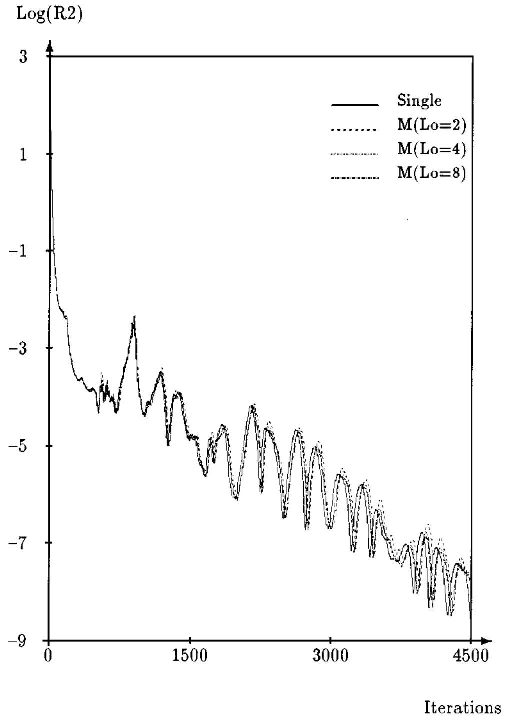


FIG. 5. Dependence of the convergence histories on the overlapping width for the steady state problem with 4 subdomains. $M_\infty = 0.85$, CFL = 5. Lerat scheme on the fine grid.

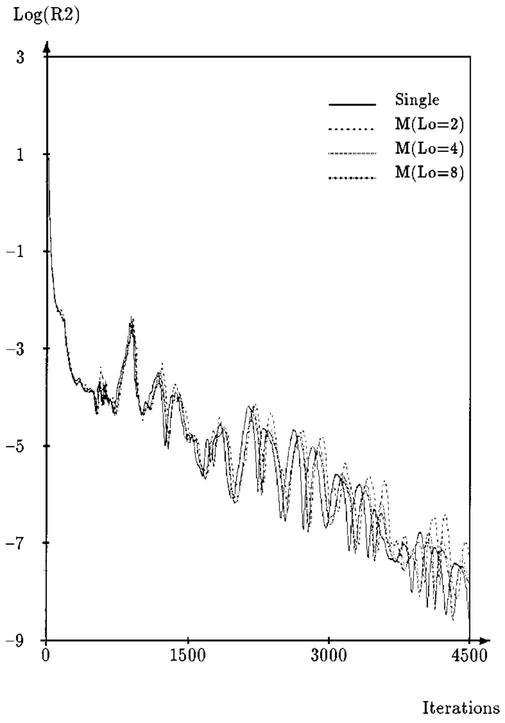


FIG. 6. Dependence of the convergence histories on the overlapping width for the steady state problem with 8 subdomains. $M_\infty = 0.85$, $\mathrm{CFL} = 5$. Lerat scheme on the fine grid.

the single domain one if the overlapping width is small. Both treatments almost have the same convergence speed for $L_o \geq \mathrm{CFL}$.

The Mach contours computed with the Roe scheme and $\mathrm{CFL} = 20$ are displayed in Fig. 11. Here the 8-domain results are compared to the single domain computation. In the multidomain case, the interfaces are shown through dashed lines. The convergence curves of the overlapping computation compared with the single domain one are displayed in Figs. 12–14 for $\mathrm{CFL} = 20$. The multidomain treatment converges as rapidly as the single domain one with a small overlapping width. The convergence speed becomes poor when the overlapping width becomes very large ($L_o \geq 20$).

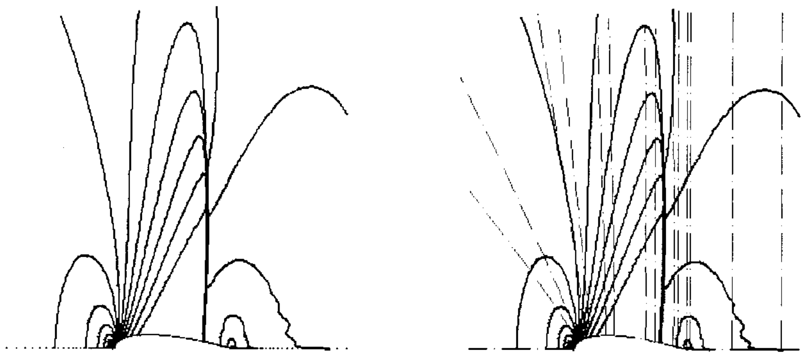


FIG. 7. Mach contours for the steady state problem. $M_\infty = 0.85$, $\mathrm{CFL} = 10$. Left, single domain. Right, 8 domains with $L_o = 10$. Lerat scheme on the fine grid.

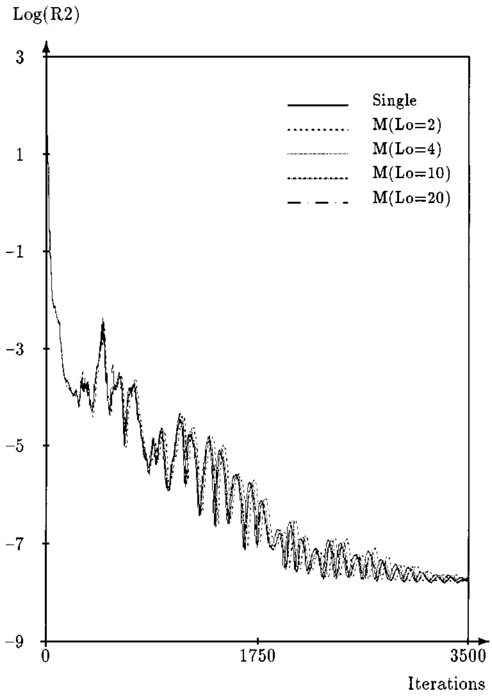


FIG. 8. Dependence of the convergence histories on the overlapping width for the steady state problem with 2 subdomains. $M_\infty = 0.85$, CFL = 10. Lerat scheme on the fine grid.

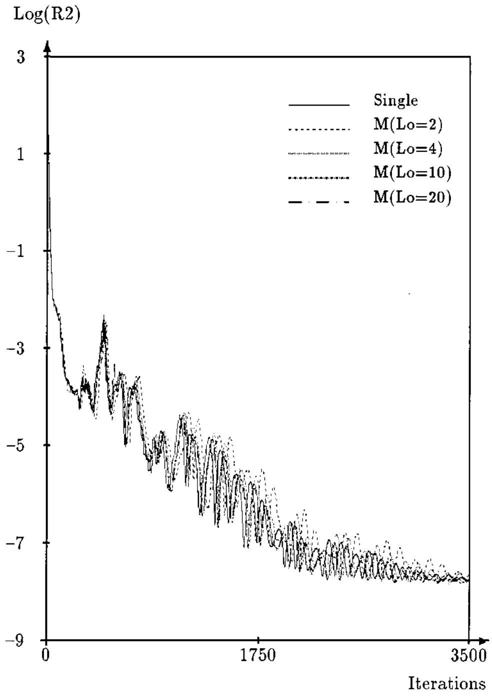


FIG. 9. Dependence of the convergence histories on the overlapping width for the steady state problem with 4 subdomains. $M_\infty = 0.85$, CFL = 10. Lerat scheme on the fine grid.

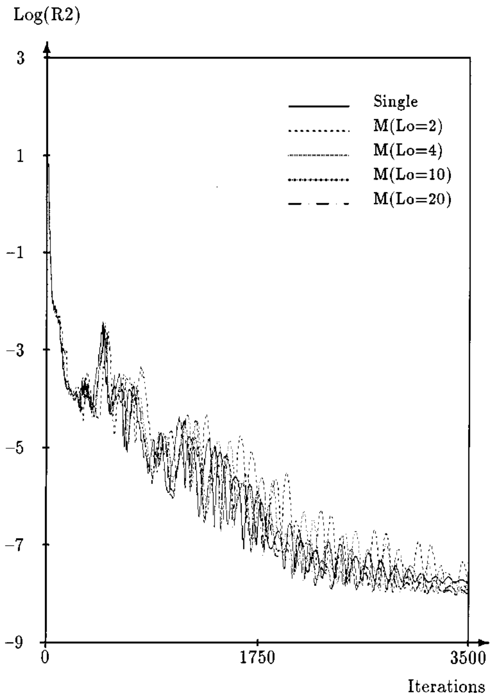


FIG. 10. Dependence of the convergence histories on the overlapping width for the steady state problem with 8 subdomains. $M_\infty = 0.85$, CFL = 10. Lerat scheme on the fine grid.

4.2.2. Unsteady problem. For the unsteady case, we use CFL = 5. The overlapping width is $L_o = 2(\mathrm{CFL} + 1) = 12$. The pressure coefficient C_p[2] distributions around the chord are displayed in Figs. 15, 16 for single domain, bidomain, 4-domain, and 8-domain computations. We remark that the overlapping/projection interface treatments give almost the same solutions as the single domain one. This can be further made clear through the Mach contours shown in Figs. 17, 18. As a result, the very simple overlapping/projection interface treatment gives a good time accuracy.

4.3. *Parallel Computation*

The parallel computation is done on a network of workstations using PVM (parallel virtual machines, see [17] for details and for references). Here we will measure the parallel performance by using the parallel efficiency, here defined as

$$E_p = \frac{cpu(1)}{n\,cpu(n)},$$

where $cpu(k)$ is the CPU time (including the communication time) computed with k processors. The parallel performance can also be measured by the speedup $cpu(1)/cpu(n)$. It is also possible to use the wall clock time T_{wc} to measure parallel efficiency:

$$\bar{E}_p = \frac{T_{wc}(1)}{nT_{wc}(n)}.$$

[2] Defined as $C_p = (p - p_\infty)/(1/2)\rho_\infty V_\infty^2$ where ρ_∞, p_∞, and V_∞ denote, respectively, the density, the pressure, and the velocity at infinity.

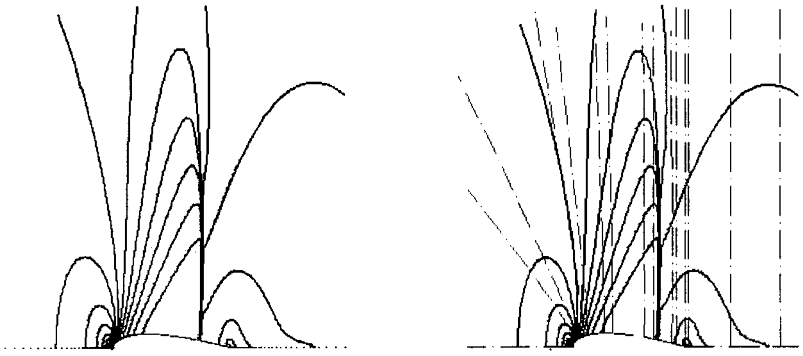


FIG. 11. Mach contours for the steady state problem. $M_\infty = 0.85$, CFL = 20. Left, single domain. Right, 8 domains with $L_o = 10$. Roe scheme on the fine grid.

Normally some of the processors of the parallel machine are occupied by different tasks (different users may occupy a part of the processors), thus the wall clock time necessarily contains an extra part due to synchronization. As a result, we must have the relation

$$E_p > \bar{E}_p \tag{54}$$

and this relation will be confirmed by our numerical experiments. Normally it is the CPU time (not the wall clock time) that is charged in the cost. The use of CPU times (including computation and communication) to measure parallel efficiency is more appropriate here to

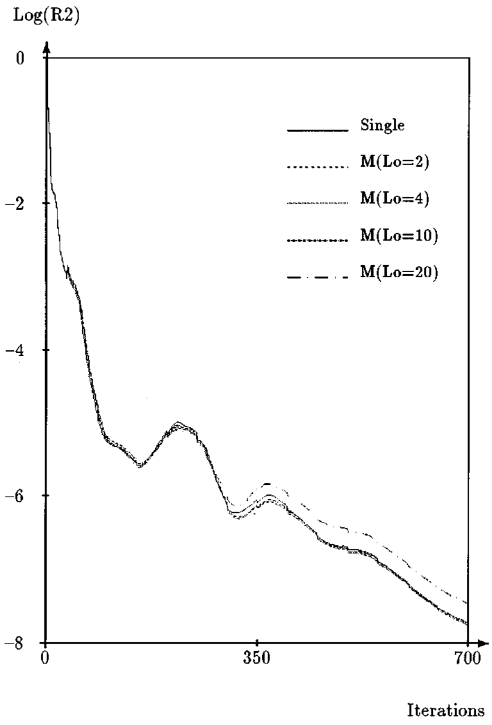


FIG. 12. Dependence of the convergence histories on the overlapping width for the steady state problem with 2 subdomains. $M_\infty = 0.85$, CFL = 20. Roe scheme on the fine grid.

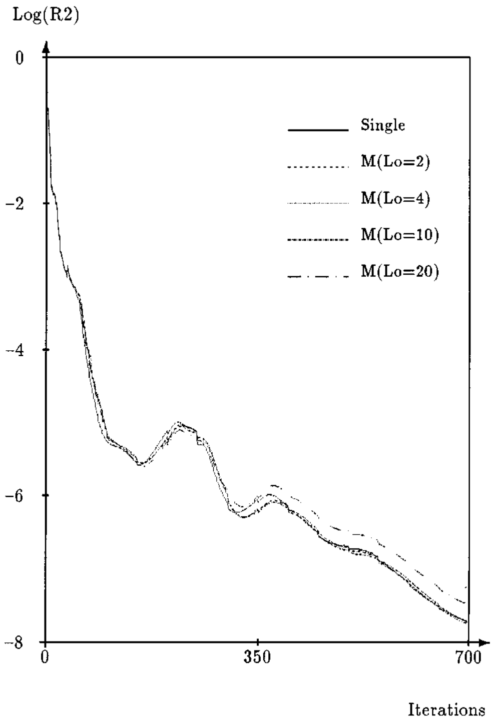


FIG. 13. Dependence of the convergence histories on the overlapping width for the steady state problem with 4 subdomains. $M_\infty = 0.85$, CFL $= 20$. Roe scheme on the fine grid.

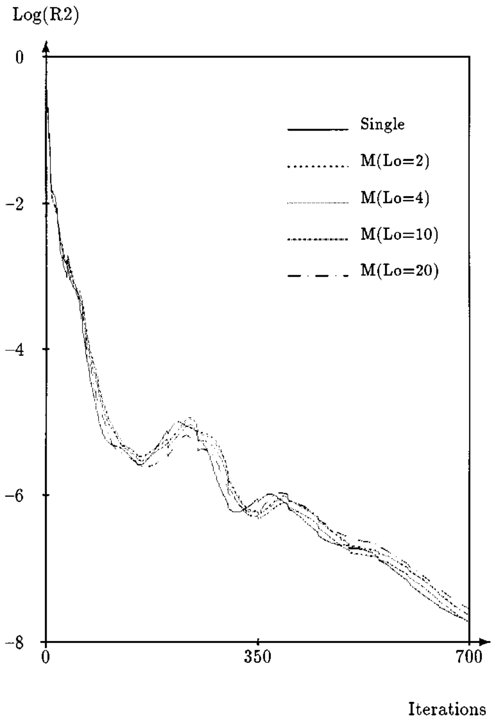


FIG. 14. Dependence of the convergence histories on the overlapping width for the steady state problem with 8 subdomains. $M_\infty = 0.85$, CFL $= 20$. Roe scheme on the fine grid.

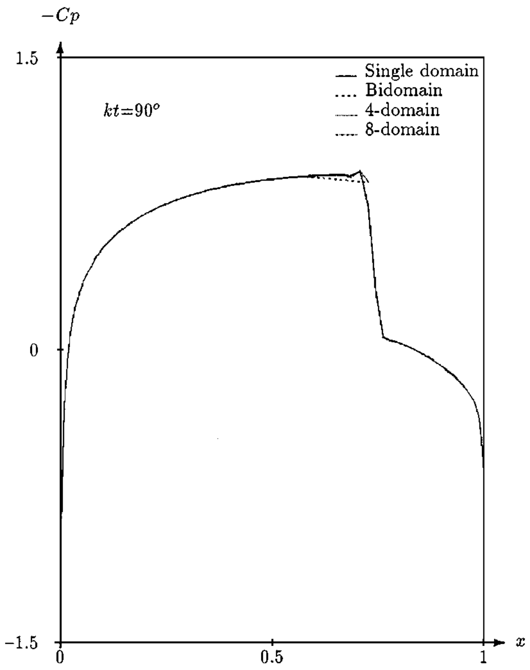


FIG. 15. Comparison of the single domain treatment, 2-domain, 4-domain, and 8-domain treatments. Pressure coefficient distribution on the wall.

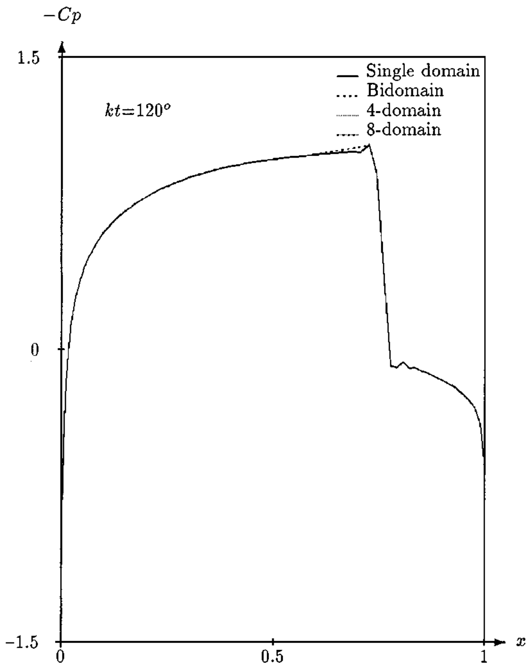


FIG. 16. Comparison of the single domain treatment, 2-domain, 4-domain, and 8-domain treatments. Pressure coefficient distribution on the wall.

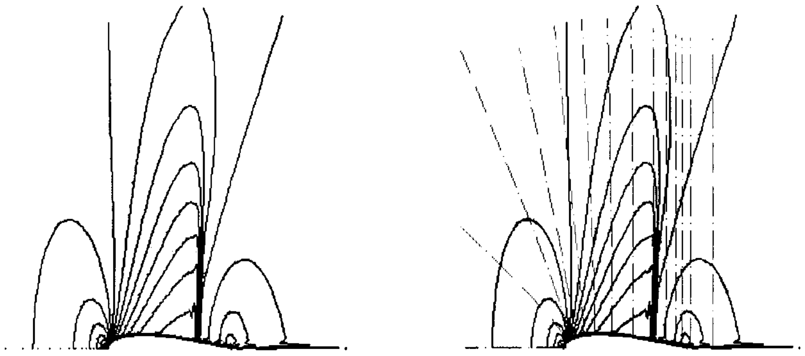


FIG. 17. Comparison of the single domain treatment (left) and 8-domain treatment (right). Mach contours at $kt = 90^\circ$.

study the numerical efficiency of the algorithm. But we will display results based on both CPU times and wall clock times.

In each case we give the parallel efficiency and the CPU time or wall clock time required to reach a prescribed convergence (the residual drops 4 orders) for steady state computations or to reach the prescribed instants for the unsteady problem.

4.3.1. Steady problem. Both the Lerat scheme and the implicit Roe scheme will be used to do steady state computations here. Let us begin with the Lerat scheme.

First we consider the subsonic case ($M_\infty = 0.536$) with CFL $= 5$. The results are displayed in Tables XIII–XV. From Table XIII, which displays the parallel efficiency based on the CPU time and on the coarse grid, we see that when the number of subdomains is small, the absolute parallel efficiency is very high ($E_p > 0.93$). This is even so with a small overlapping width. When the number of subdomains is equal to 8, the optimal overlapping width is approximately equal to CFL, for which the parallel efficiency is the highest ($E_p = 0.97$). Table XIV gives the results based on the wall clock time. The parallel efficiencies based on the wall clock time are slightly smaller than those based on the CPU time, as can be expected from (54). In any case, the parallel efficiency is a decreasing function of the number of subdomains. Table XV displays the results obtained with the fine grid. Comparing Table XV with Table XIV, when the number of subdomains is 8 and when $L_o \leq 4$,

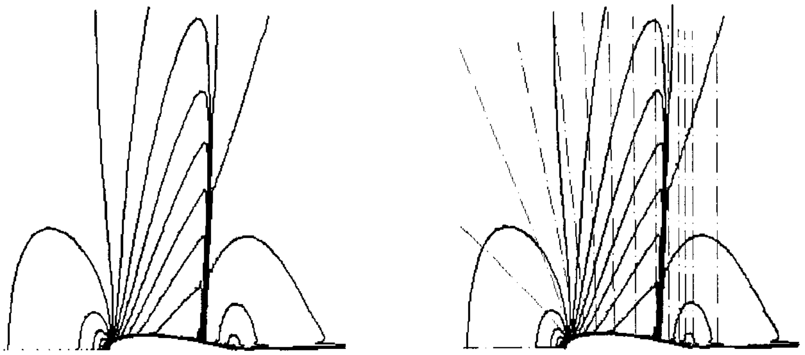


FIG. 18. Comparison of the single domain treatment (left) and 8-domain treatment (right). Mach contours at $kt = 120^\circ$.

TABLE XIII

Parallel Efficiency E_p (Based on the CPU Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	*cpu*	E_p	*cpu*	E_p	*cpu*	E_p
2	726	1.074	343	1.135	209	0.931
4	805	0.968	380	1.024	200	0.972
8	833	0.935	405	0.961	214	0.910

For single domain $cpu = 1558$

Note. Steady problem with $M_\infty = 0.536$. Computed with the coarse grid, Lerat scheme, and CFL = 5.

TABLE XIV

Parallel Efficiency $\bar{E}_p$ (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	777	1.027	401	0.995	274	0.727
4	867	0.920	439	0.908	271	0.735
8	884	0.902	496	0.804	342	0.582

For single domain $T_{wc} = 1595$

Note. Steady problem with $M_\infty = 0.536$. Computed with the coarse grid, Lerat scheme, and CFL = 5.

TABLE XV

Parallel Efficiency $\bar{E}_p$ (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	2137	1.195	1120	1.140	698	0.915
4	2504	1.020	1151	1.110	715	0.893
6	2566	0.995	1213	1.053	736	0.868
8	2587	0.991	1253	1.019	764	0.835
10	2705	0.944	1337	0.955	786	0.812
12	2736	0.934	1401	0.912	801	0.797
16	2763	0.924	1464	0.872	866	0.737

For single domain $T_{wc} = 5108$

Note. Steady problem with $M_\infty = 0.536$. Computed with the fine grid, Lerat scheme, and CFL = 5.

TABLE XVI

Parallel Efficiency E_p (Based on the CPU Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	*cpu*	E_p	*cpu*	E_p	*cpu*	E_p
2	2013	0.982	1067	0.926	751	0.658
4	2030	0.974	1066	0.927	599	0.825
8	2084	0.948	1053	0.939	617	0.800

For single domain $cpu = 3954$.

Note. Steady problem with $M_\infty = 0.85$. Computed with the coarse grid, Lerat scheme, and CFL = 5.

TABLE XVII
Parallel Efficiency E_p (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	2129	0.949	1259	0.803	951	0.531
4	2156	0.937	1239	0.816	780	0.647
8	2198	0.919	1261	0.801	859	0.588
For single domain $T_{wc} = 4041$.						

Note. Steady problem with $M_\infty = 0.85$. Computed with the coarse grid, Lerat scheme, and CFL = 5.

the parallel efficiency is near 0.90 for the fine grid while it is below 0.75 for the coarse grid.

Now we consider the transonic case ($M_\infty = 0.85$) with CFL = 5. The parallel efficiencies (based on the CPU time and the wall clock time on the coarse grid) are displayed in Tables XVI and XVII. The results are very similar to the case of subsonic flows. When the number of subdomains is 8, the optimal overlapping width is $L_o = 4$ (close to the CFL number), for which the parallel efficiency is the highest ($E_p = 0.825$).

Now we consider the transonic case ($M_\infty = 0.85$) with a higher CFL number. The parallel efficiencies for CFL = 10 (based on the CPU time on the coarse grid, wall clock time on the coarse grid, and wall clock time on the fine grid) are displayed in Tables XVIII–XX. The results are still very similar to the case of subsonic flows. When the fine grid is used and when the number of subdomains is 8, the optimal overlapping width is $L_o = 8$ (close to CFL), for which the parallel efficiency is the highest ($E_p = 0.707$). In this case the overlapping efficiencies are smaller than those with CFL = 5, especially when the number of subdomains is large. The reason is that with CFL = 10, the optimal overlapping width is large with respect to the number of mesh points in each subdomain.

In the above computations we do not use a CFL number higher than 10. This is because the Lerat scheme is unstable for the two-dimensional Euler equations when CFL is very large (see [12]). In order to do computations with higher CFL numbers, we have also used the implicit Roe scheme.

The parallel efficiencies for CFL = 20 and $M_\infty = 0.535$ (based on the CPU time and the wall clock time) are displayed in Tables XXI and XXII. Unlike the Lerat scheme, here the

TABLE XVIII
Parallel Efficiency E_p (Based on the CPU Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	*cpu*	E_p	*cpu*	E_p	*cpu*	E_p
2	1062	1.023	589	0.924	356	0.763
4	1089	0.998	605	0.897	369	0.735
10	1099	0.988	614	0.884	375	0.723
20	1255	0.866	736	0.738	477	0.568
For single domain, $cpu = 2173$.						

Note. Steady problem with $M_\infty = 0.85$. Computed with the coarse grid, Lerat scheme, and CFL = 10.

TABLE XIX
Parallel Efficiency E_p (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	1108	0.986	642	0.851	415	0.658
4	1143	0.956	678	0.806	445	0.614
10	1141	0.957	714	0.766	494	0.553
20	1302	0.839	909	0.601	748	0.365
For single domain $T_{wc} = 2189$.						

Note. Steady problem with $M_\infty = 0.85$. Computed with the coarse grid, Lerat scheme, and CFL = 10.

TABLE XX
Parallel Efficiency E_p (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	7120	0.853	3987	0.762	2330	0.652
4	6412	0.947	3860	0.787	2271	0.669
6	6374	0.953	4041	0.752	2427	0.626
8	6378	0.952	3529	0.861	2149	0.707
10	6286	0.966	3574	0.850	2224	0.683
12	6430	0.945	3711	0.818	2309	0.658
14	6458	0.941	3823	0.795	2421	0.627
20	6538	0.929	3932	0.773	2849	0.533
For single domain $T_{wc} = 12150$						

Note. Steady problem with $M_\infty = 0.85$. Computed with the fine grid, Lerat scheme, and CFL = 10.

TABLE XXI
Parallel Efficiency E_p (Based on the CPU Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	*cpu*	E_p	*cpu*	E_p	*cpu*	E_p
2	2365	0.732	1190	0.728	609	0.711
4	2386	0.726	1209	0.716	627	0.690
10	2436	0.711	1274	0.680	688	0.629
20	2513	0.689	1378	0.629	786	0.551
30	2595	0.668	1473	0.588	867	0.499
For single domain CPU = 3466						

Note. Steady problem with $M_\infty = 0.535$. Computed with the fine grid, Roe scheme, and CFL = 20.

TABLE XXII

Parallel Efficiency E_p (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	2391	0.730	1228	0.711	654	0.667
4	2414	0.723	1259	0.693	684	0.648
10	2462	0.709	1373	0.636	788	0.554
20	2537	0.688	1541	0.566	964	0.454
30	2641	0.661	1710	0.511	1144	0.333
For single domain $T_{wc} = 3494$						

Note. Steady problem with $M_\infty = 0.535$. Computed with the fine grid, Roe scheme, and CFL = 20.

optimal overlapping width is always equal to $L_o = 2$. The overlapping efficiencies for large overlapping width are unacceptable.

The parallel efficiencies for CFL = 20 and $M_\infty = 0.85$ (based on the CPU time and the wall clock time) are displayed in Tables XXIII and XXIV. Similarly as above, here the optimal overlapping width is always equal to $L_o = 2$. The overlapping efficiencies are slightly larger than those obtained with the subsonic flow case.

The performance of the overlapping treatment with the implicit Roe scheme seems to be independent of the CFL number. To see that, we display in Tables XXV and XXVI the parallel efficiencies E_p and $\bar{E}_p$ obtained with CFL = 10 for $M_\infty = 0.85$. They are very close to the parallel efficiencies obtained with CFL = 20 (see Tables XXIII and XXIV).

4.3.2. Unsteady problem. For the overlapping/projection interface treatment, the communication time required to project the unpolluted values to the polluted points is proportional to the overlapping width and thus the CFL number. Thus the CFL number should be small with respect to the mesh size in each subdomain.

Only the Lerat scheme is used for unsteady flow computations. Three CFL numbers are used: CFL = 2, CFL = 5, and CFL = 8. The corresponding overlapping widths are given by $L_o = 6$, 12, 18, respectively. The parallel efficiencies based on the CPU time and on the coarse grid are displayed in Tables XXVII, XXVIII, and XXIX. When CFL is relatively

TABLE XXIII

Parallel Efficiency E_p (Based on the CPU Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	*cpu*	E_p	*cpu*	E_p	*cpu*	E_p
2	4021	1.001	2029	0.994	1071	0.942
4	4072	0.991	2079	0.970	1110	0.908
10	4156	0.971	2181	0.925	1189	0.848
20	4682	0.861	2584	0.781	1475	0.683
30	4881	0.826	2783	0.725	1623	0.621
For single domain CPU = 8067						

Note. Steady problem with $M_\infty = 0.85$. Computed with the fine grid, Roe scheme, and CFL = 20.

TABLE XXIV

Parallel Efficiency E_p (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	4074	0.998	2093	0.972	1151	0.884
4	4120	0.987	2164	0.939	1209	0.841
10	4200	0.968	2349	0.866	1361	0.747
20	4726	0.860	2889	0.704	1809	0.562
30	4968	0.819	3230	0.630	2139	0.475
For single domain $T_{wc} = 8133$						

Note. Steady problem with $M_\infty = 0.85$. Computed with the fine grid, Roe scheme, and CFL $= 20$.

TABLE XXV

Parallel Efficiency E_p (Based on the CPU Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	*cpu*	E_p	*cpu*	E_p	*cpu*	E_p
2	6110	0.994	3099	0.980	1590	0.955
4	6164	0.985	3154	0.963	1642	0.924
10	6305	0.963	3329	0.912	1811	0.838
20	6750	0.900	3739	0.812	2162	0.702
For single domain CPU $= 12145$						

Note. Steady problem with $M_\infty = 0.85$. Computed with the fine grid, Roe scheme, and CFL $= 10$.

TABLE XXVI

Parallel Efficiency E_p (Based on the Wall Clock Time) vs Overlapping Width L_o

	2 domains		4 domains		8 domains	
L_o	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$	T_{wc}	$\bar{E}_p$
2	6177	0.991	3195	0.958	1706	0.897
4	6235	0.982	3283	0.932	1788	0.856
10	6372	0.961	3585	0.853	2073	0.738
20	6812	0.899	4173	0.733	2650	0.577
For single domain $T_{wc} = 12243$						

Note. Steady problem with $M_\infty = 0.85$. Computed with the fine grid, Roe scheme, and CFL $= 10$.

TABLE XXVII

Parallel Efficiency E_p (Based on the CPU Time) vs the Number of Subdomains

	90°	120°	150°	
	cpu	*cpu*	*cpu*	E_p
2 domains	2061	2748	3435	0.981
4 domains	1060	1414	1768	0.953
8 domains	578	771.8	964	0.873
Single domain	4036	5390	6726	

Note. Unsteady problem with $L_o = 2(\text{CFL} + 1) = 6$. Lerat scheme on the coarse grid.

TABLE XXVIII
Parallel Efficiency E_p (Based on the CPU Time) vs the Number of Subdomains

	90°	120°	150°	
	cpu	*cpu*	*cpu*	E_p
2 domains	864	1151	1440	0.942
4 domains	471	628	786	0.862
8 domains	279	372	465	0.728
1 domain	1627	2168	2711	

Note. Unsteady problem with $L_o = 2(\text{CFL} + 1) = 12$. Lerat scheme on the coarse grid.

small with respect to the number of mesh points in each subdomain, the (absolute) parallel efficiency is sufficiently high since the interface treatment is very simple.

Table XXX displays the parallel efficiencies based on the wall clock time and on the coarse grid, computed with $L_o = 2(\text{CFL} + 1) = 12$. As in the steady case, the parallel efficiencies based on the wall clock time are slightly smaller than those based on the CPU time.

Table XXXI displays the parallel efficiencies based on the wall clock time and on the fine grid, computed with $L_o = 2(\text{CFL} + 1) = 12$. As in the steady case, the parallel efficiencies based on the fine grid are larger than those based on the coarse grid.

4.4. *Conclusions of the Numerical Results*

The numerical experiments based on the two dimensional Euler equations yield the following conclusions:

(1) For the Lerat scheme and the steady state problem, when the number of subdomains is small, the optimal overlapping width is close to $L_o = 2$. When the number of subdomains is equal to 8, we indeed obtain an optimal overlapping width close to CFL, for which the parallel efficiency takes its largest value. This result confirms the linear study which is based on a scalar 1D equation. The role of approximate factorization, exterior boundary treatments, nonuniform mesh sizes, etc., are however not taken into account in the linear analysis. As in the linear study where we sometimes obtain the same convergence speed as

TABLE XXIX
Parallel Efficiency E_p (Based on the CPU Time) vs the Number of Subdomains

	90°	120°	150°	
	cpu	*cpu*	*cpu*	E_p
2 domains	564	753	941	0.893
4 domains	324	432	540	0.776
8 domains	204	271	340	0.617
Single domain	1006	1342	1681	

Note. Unsteady problem with $L_o = 2(\text{CFL} + 1) = 18$. Lerat scheme on the coarse grid.

TABLE XXX
Parallel Efficiency E_p (Based on the Wall Clock Time) vs the Number of Subdomains

	90°	120°	150°	
	T_{wc}	T_{wc}	T_{wc}	$\bar{E}_p$
2 domains	856	1141	1426	0.943
4 domains	519	689	861	0.781
8 domains	304	406	507	0.663
1 domain	1614	2153	2691	

Note. Unsteady problem with $L_o = 2(\mathrm{CFL} + 1) = 12$. Lerat scheme on the coarse grid.

the single domain case (overlapping efficiency close to 0), the parallel computation using the Lerat scheme is sometimes capable of recovering the same convergence speed as the single domain case, i.e., the parallel efficiency is sometimes close to 1.

(2) For the implicit upwind scheme and the steady state problem, the optimal overlapping width is always equal to $L_o = 2$, which confirms exactly the linear study. The parallel efficiency is independent of the CFL number.

(3) Since the Thomas algorithm for inverting the implicit tridiagonal system is unchanged with respect to a code for sequential computation, the parallel efficiency E_p or $\bar{E}_p$ we present here is the absolute one. The parallel efficiencies we obtained lie in the range of (0.5, 1.0) for both steady and unsteady problems. This result is acceptable since we have used the definition of absolute parallel efficiency, a very simple interface treatment, and a very small grid.

(4) For the Lerat scheme and the steady state problem, the parallel efficiency is a decreasing function of the number of subdomains. This is not due to the inefficiency of the interface treatment, but essentially due to the decreasing of the number of mesh points in each subdomain. We note that the total number of mesh points, when the additional points due to overlapping are not taken into account, remains fixed independently of the number of subdomains. The total number of mesh points in each subdomain is inversely proportional to the number of subdomains. As a result, the communication time between processors

TABLE XXXI
Parallel Efficiency E_p (Based on the Wall Clock Time) vs the Number of Subdomains

	90°	120°	150°	
	T_{wc}	T_{wc}	T_{wc}	$\bar{E}_p$
2 domains	6246	8328	10410	0.977
4 domains	3585	4780	5976	0.851
6 domains	2624	3499	4374	0.775
8 domains	2132	2842	3553	0.714
12 domains	1627	2170	2712	0.625
Single domain	12199	16265	20331	

Note. Unsteady problem with $L_o = 2(\mathrm{CFL} + 1) = 12$. Lerat scheme on the fine grid.

becomes relatively important when the number of subdomains increases. For example, when there are 8 subdomains, there are only $124/8 = 15$ mesh points in the direction normal to the interface. The cost due to overlapping is therefore large in comparison with the total CPU time. The situation can obviously be imporved once a finer grid is used, as can be seen from Table XV (based on the fine grid) and from Table XIV (based on the coarse grid). Such a test, with the total number of meshes maintained fixed and small, is indeed the worst case for parallel compuation. In real applications, the size of the mesh is either very large or proportional to the number of processors. The parallel efficiency for real applications should be greater than those given by the present calculations. Here the parallel computation is just for testing the numerical efficiency of the easily workable interface treatments.

5. CONCLUSIONS

In this paper we have considered parallel computations for three-point implicit schemes by grid overlapping. We have derived, analyzed, and validated:

(1) some time-lagging interface treatments suitable for parallel computations of steady state problems;
(2) an overlapping/projection interface treatment suitable for parallel computations of unsteady problems.

These interface treatments are very simple to use. They need no iterations at each time step, or modification of the implicit solver (tridiagonal systems inverted by approximation factorization and the Thomas algorithm). Both treatments lie in the correct choice of an overlapping width. Though they are apparently very simple, the following key points should be kept in mind:

(1) The interior difference equations must be sufficiently dissipative in order to have a good convergence speed for the steady state problems. Of course, one rarely uses a nondissipative scheme for steady state computations.
(2) Among the various possible time-lagging treatments, only the totally time-lagging condition behaves always very well with regard to stability, convergence. For unsteady problems, the time-accurate time-lagging condition produces a result as accurate as the single domain computation. But the totally time-lagging condition also yields acceptable results in the nontrivial two dimensional cases.
(3) The choice of the overlapping width is scheme-dependent. For the Lerat scheme, the optimal one is close to the CFL number. For the upwind scheme, the optimal overlapping width is equal to 2.
(4) Though the present numerical experiments demonstrate a good absolute parallel efficiency for both steady and unsteady problems, this parallel efficiency is expected to be better in the case of real applications where the number of mesh points is very large in comparison with the CFL number.

APPENDIX A: REMARK ON CONSERVATION

It was believed that conservation at grid interfaces was important for a nonlinear system of conservation laws with discontinuous solutions (shock waves). When spatial interpolation is involved, Berger's flux interpolation [1] leads to conservative solutions; see also [4, 16] for subsequent studies. Conservation is important only when a shock wave approaches a grid interface. A recent study [23] based on shock/interface interaction shows that if the interior difference equations are sufficiently dissipative or the shock speed is sufficiently large, then a nonconservative treatment also works. Most of the schemes currently used for shocked flow computations have enough of an amount of dissipation for the computation to yield a correct shock position. For the present problem interpolation in time (in fact the time-lagging treatment is an extrapolation in time) is involved, and the result of [23] remains valid; that is, for a difference scheme whose interior dissipation near the shock is not too much smaller than that of the standard first-order Roe scheme, then a shock wave will successfully transmit the overlapping interface. This is the case for first-order upwind schemes, MUSCL schemes with limiters, TVD schemes, NND [24] schemes, schemes equipped with nonlinear filters or other kinds of artificial dissipation. Besides, the overlapping/projection interface treatment for unsteady problems has an additional projection which is similar to the penetrator constructed in [23]. This penetrator forces shock/interface transmission in all cases.

APPENDIX B: PRACTICAL ASPECTS OF THE PRESENT OVERLAPPING METHOD

B.1. *Simplicity of the present method.* The present method presents the following advantages:

(1) The present method can be quite easily incorporated into an existing sequential code in order to do parallel computations. It does not require any modification of the algorithm for inverting the implicit system. The interface conditions are as simple as (in fact simpler than) an ordinary boundary condition.

(2) The parallel tridiagonal solver [20] requires a thourough modification of the original Thomas algorithm for inverting the tridiagonal system. It requires 50% more CPU time than the original Thomas algorithm. As a result, the absolute parallel efficiency $E_p < \frac{1}{1+0.50} \approx 0.67$.

(3) For elliptical problems, there exist the well-known Schwartz and Schur Complement methods [2]. These methods are often based on the differential equations. Precisely, the domain decomposition is done on the differential equations. The resulting problem is then discretized by a finite element method or other methods. This is different than the present approach where the domain decomposition is done on the difference equations. The original Schwartz algorithm needs some iterations at each time step. The Schur complement method needs to solve a subsystem inherient to the accurate interface treatment.

B.2. *For steady state computations, one does not need to absolutely use the optimal overlapping width.* The apparent shortcoming is that when using high CFL numbers, one needs to add a high overlapping width which obviously leads to high storage and more CPU time. In fact, one does not need to absolutely take the optimal overlapping width.

(1) For schemes behaving similarly as the Lerat scheme, though optimal convergence occurs (theoretically) for $L_o = \text{CFL}$, one can still take an overlapping width as short as he likes if his main concern is storage. Besides, numerical experiments show that very good convergence also occurs at short overlapping width.

(2) For upwind schemes, the best convergence occurs at $L_o = 2$ so that both storage and CPU time are the minimal. The only restriction is that we should use the standard (totally time-lagging) interface condition.

B.3. *For unsteady problems, the CFL number should not be too large by the accuracy consideration.* Unsteady problems can be classified into two classes: fast unsteady flow and slowly unsteady flow. For fast unsteady flow, one can simply use Range–Kutta type explicit schemes. The implicit method is mainly useful for slowly unsteady flow. The implicitation is used to increase the physical time step in order to reduce the computational time. But if we take $L_o = 2(\mathrm{CFL} + 1)$, then it appears that the overall performance with regard to CPU time and storage appears poor. However, if we keep in mind that the method is designed for parallel computation, then this disadvantage is largely compensated by its advantage because of the following reasons.

(1) Unlike the steady state problems where we seek large CFL numbers to increase the convergence speed, in the unsteady case we still require the CFL number to be not too large in order to have a good time accuracy. This is even so for single domain treatments. In computations of unsteady flow using implicit schemes, one typically uses a CFL number of the order of 10 even for single domain (processor) computations.

(2) A parallel computation for real applications is not done on a small grid. It is generally used for the purpose of handling large size problems in a reasonable time. The number of mesh points in each subdomain and per direction, denoted N_p for convenience, should be large enough since each processor is fast enough to handle such a subdomain. If we keep the ratio $\mathrm{CFL}/N_p \ll 1$, as should be so in real applications, the additional time and storage wasted in the overlap is still negligible. As a result, if N_p is high (as is the case for the parallel computation purpose), we may still use high CFL numbers. If N_p is small, one can simply use an explicit method which is stable for a CFL of order 1.

ACKNOWLEDGMENTS

This work was supported by Chinese National Science Foundations and by Foundations of Chinese Ministry of Education for Young Teachers.

REFERENCES

1. M. Berger, On conservation at grid interfaces, *SIAM J. Numer. Anal.* **24**, 967 (1987).
2. T. F. Chan and D. Goovearts, *Schwarz-Schur, Overlapping versus Non-overlapping Domain Decomposition*, Technical Reports, CAM 88-21, UCLA, 1988.
3. G. Chesshire and D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, *J. Comput. Phys.* **90**, 1 (1990).
4. G. Chesshire and D. Henshaw, A scheme for conservative interpolation on overlapping grids, *SIAM J. Sci. Comput.* **15**, 819 (1994).
5. G. De-Spiegeleer, A. Lerat, and Z. N. Wu, Implicit multidomain computation of compressible flows with a large number of subdomains, *Comput. Fluid Dynam. J.* **7**, 245 (1998).
6. B. Gustafsson, H.-O. Kreiss, and A. Sundström, Stability theory of difference approximations for initial boundary value problems, II, *Math. Comput.* **26**, 649 (1972).
7. B. Gustafsson, The Euler and Navier–Stokes equations: Wellposedness, stability and composite grids, in *Computational Fluid Dynamics* (von Karman Institute for Fluid Dynamics, February 18–22, 1991); Lecture Series 1991-01.
8. B. Gustafsson, The choice of numerical boundary conditions for hyperbolic systems, *J. Comput. Phys.* **48**, 270 (1982).
9. B. Gustafsson, The convergence rate for difference approximations to general mixed initial boundary value problem, *SIAM J. Numer. Anal.* **18**, 179 (1981).

10. D. E. Keyes, *Domain Decomposition: A Bridge between Nature and Parallel Computers*, ICASE Report No. 92-44, 1992.
11. H.-O. Kreiss and J. Lorenz, *Initial-Boundary Value Problems and the Navier–Stokes Equations* (Academic Press, San Diego, 1989).
12. A. Lerat, Multidimensional centered schemes of the Lax–Wendroff type, in *CFD Review* (Wiley, New York, 1995), p. 124.
13. A. Lerat and J. Sidès, Numerical simulation of unsteady transonic flows using the Euler equations in integeral form, *Israel J. Tech.* **17**, 302 (1979).
14. A. Lerat and Z. N. Wu, Stable conservative multidomain treatments for implicit Euler solvers, *J. Comput. Phys.* **123**, 45 (1996).
15. S. Osher, Stability of difference equations of dissipative type for mixed initial-boundary value problems, I, *Math. Comput.* **23**, 235 (1969).
16. E. Pärt-Enander and B. Sjögreen, Conservative and nonconservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems, *Comput. & Fluids* **23**, 551 (1994).
17. D. Roose and R. V. Driessche, Parallel computers and parallel algorithms for CFD: An introduction, in *AGARD-R-807* (1995), p. 1.
18. M. L. Sawley and J. K. Tegnér, A comparison of parallel programming models for multiblock flow computations, *J. Comput. Phys.* **122**, 280 (1995).
19. M. Thuné, *Stability of Difference Approximations of Hyperbolic Systems on Substructured Domains*, Department of Scientific Computing, Report No. 106, Uppsala University, December 1986.
20. H. Wang, A parallel solver for tridiagonal solvers, *ACM Trans. Math. Software* **7**, 170 (1981).
21. Z. N. Wu, Convergence study of an implict multidomain method for compressible flow computations, *Comput. & Fluids* **25**, 181 (1996).
22. Z. N. Wu, Uniqueness of steady state solutions for difference equations on overlapping grids, *SIAM J. Numer. Anal.* **33**, 1336 (1996).
23. Z. N. Wu, Steady and unsteady shock waves on overlapping grids, *SIAM J. Sci. Comput.* **20**, 1851 (1999).
24. H. X. Zhang and F. G. Zhuang, NND schemes and their applications to numerical simulations of two- and three-dimensional flows, *Adv. Appl. Mech.* **29**, 193 (1992).