Multi-Domain Multi-Model Formulation for Compressible Flows: Conservative Interface Coupling and Parallel Implicit Solvers for 3D Unstructured Meshes

Marius Paraschivoiu* Xiao-Chuan Cai† Marcus Sarkis‡ David P. Young§

David E. Keyes¶

Abstract

In this paper we present a multi-domain multi-model formulation for three-dimensional compressible flows. Using multi-domains we can map the computation efficiently onto multi-processor parallel computers and using multi-models we can reduce the arithmetic cost. The goal is to minimize the overall time and memory required to simulate the flow by using locally selected, more computational efficient physical models without sacrificing the global fidelity of the simulation. We introduce a finite volume based conservative interpolation for the coupling of the full potential equation and the Euler equations, and the resulting nonlinear systems are solved by a Defect Correction method. To demonstrate the feasibility of this method, we present computational results for subsonic and transonic flows around wings, and also a comparison with results obtained using the TRANAIR package of Boeing.

*Dept. of Mech. and Ind. Eng., Univ. of Toronto, Toronto, Canada, M5S 3G8. *marius@mie.utoronto.ca*

†Dept. of Comp. Sci., Univ. of Colorado, Boulder, CO 80309. *cai@cs.colorado.edu*

‡Math. Sci. Dept., Worcester Poly. Inst., Worcester, MA 01609. *msarkis@wpi.edu*

§AIAA member. The Boeing Company, Seattle, WA 98124. *dpy6629@cfdd51.cfd.ca.boeing.com*

¶AIAA member. Dept. of Comp. Sci., Old Dominion Univ., Norfork, VA 23529 and ICASE, NASA Langley Research Center, Hampton, VA 23681. *keyes@icase.edu*

1 Introduction

Numerical simulations of fluid flow have sufficiently matured to be considered accurate for engineering design and analysis. However, for large scale simulations, the computation time remains unacceptably large for the software to be used as an interactive tool even on the latest supercomputers. While parallel computing may reduce computation time nearly proportionally to computational resources, new algorithms should be constructed to perform faster on new and existing resources.

Compressible fluid flow simulations needed for aerodynamic applications can be modeled with different degrees of complexity. The simplest model is the full potential equation which assumes inviscid, irrotational and isentropic flows. This model utilizes a single second-order nonlinear differential equation that is inexpensive with respect to the execution time and the memory requirement. Validity of the full potential equation is, however, restricted. The isentropic assumption of the potential flow model leads to inaccurate physics for transonic flows with strong shocks. The next level of approximation is the Euler equations which describe the complete behavior of inviscid compressible flows. The Euler model utilizes a coupled system of five nonlinear differential equations of first order. The five field variables lead to a fivefold increase in the memory needed over the full potential, for the same mesh density. Finally, the Navier-Stokes equations include the viscous effects needed for accurate modeling of the boundary layer. These equations are not only more time consuming to solve but also require an associated mesh that is stretched and very fine in viscous regions. Nevertheless, for complex flows with separation of the boundary layer, the Navier-Stokes

equations are mandatory to provide an accurate simulation. Furthermore, for high Reynolds number flows, turbulence appears and needs to be modeled.

The approach proposed herein involves splitting the computational domain into different fluid flow regions and using the full potential model instead of the Euler or Navier-Stokes equations in regions where this approximation is valid. When considering transonic flows over a wing, three regions can be identified: the boundary layer, the region around the shock and the farfield. A multi–model formulation can be used to combine the strengths of each model described above. Indeed, a multi-model formulation will take advantage of the quick computational time associated with solving the full potential equation while capturing all the important features of the flow such as boundary layers and shocks using the Navier-Stokes equations and/or the Euler equations, respectively. It can be shown numerically that solving the full potential equation in regions of irrotational flow is not only more efficient but also improves the accuracy.

This research benefits from the extensive experience of numerical methods and software developed over the years to solve the full potential equation, the Euler equations and the Navier–Stokes equations separately. Numerical techniques for the solution of the full potential equation and the Euler/Navier-Stokes equations were developed respectively in the past 30 years [3, 5, 11, 13, 15, 18, 24]. These methods have matured and can simulate compressible flows around entire aircraft. For example, the full potential equation was solved for a Boeing 747-200 transport configuration with wing, body, struts, and nacelles [24]. For the Euler model, calculations over a complete aircraft were performed as early as 1986 [15]. However, challenges remain for viscous flows. Accurate simulations at high Reynolds number over such complex geometries require enormous computational resources. Approximate solutions, i.e., with less than adequate number of mesh points, have been performed. A Navier-Stokes prediction for the McDonnell Douglass F–18 wing and fuselage is presented in [8]. A discussion of the drastic difference in computational cost related to the choice of models can be found in [14].

Only recently has there been interest in coupling these solvers to reduce the computational cost, to reduce the memory requirement, and to improve the accuracy of the solution. Certainly, boundary layer coupling or thin layer Navier-Stokes coupling have been widely used but such approaches do not quite include all the physics required for complicated flows [23].

For a mathematical description of heterogeneous domain decomposition methods, in particular for coupling of compressible flows, we refer the reader to [19]. A characteristic analysis, using the quasi-linear vector form of the full potential equation, concludes that, contrary to the Euler system, the characteristic curves along which vorticity and entropy propagate do not exist. Recall that the full potential equation assumes irrotational flow and constant entropy. This suggests that incoming information for the Euler domain must be provided.

For three-dimensional flows, it is shown in [1, 21] that the computational cost can be reduced by a factor of two for a Navier-Stokes/full potential coupling. This method was enhanced with the addition of a Lagrangian wake to include the wake effect in the full potential region in order to study its unsteady behavior. The formulation in these studies is based on a structured grid discretization in which the full potential equation is solved using a finite difference method and the Navier-Stokes equations are solved with either a finite difference or a finite volume discretization. The savings are justified by the fact that two-thirds to one-half of the computational elements are outside the Navier-Stokes region. In general, the cost of the full potential solver can be considered negligible compared to the Navier-Stokes solver. The solution strategy is based on solving the different equations alternatively in each region, similar to a subdomain iterative method.

Our formulation differs from [1, 21] in providing a general finite volume approach and therefore ensures that the mass will also remain conserved at the discrete level. In addition, an unstructured discretization of the computational domain provides more flexibility to mesh complex geometries and for adaptive control of the numerical error. Lastly, a parallel version is implemented to obtain reasonable execution time.

In [6], initial steps of this research were taken. We presented the derivation and implementation of the interface condition between the full potential equation and the Euler equations. Due to the natural parallelization, an explicit approach was first considered not only to validate the spatial discretization but also as a precursor to the implicit implementation to be presented herein. In this paper we report on the development of an implicit approach for solving the coupled models. We also only investigate the coupling between the full potential equation and the Euler equations with the intention of gaining the required basic understanding before extending this method to include the Navier-Stokes model. Indeed, to account for viscous effect, the Navier Stokes equations need to be included in our multi-model formulation.

The purpose of this paper is to describe the implicit two-model formulation. Section 2 describes the implicit Euler solver and full potential solver with more empha-

sis on the latter. In Section 3, we briefly introduce the coupled solver and describe the implicit implementation for the interface condition. Subsequently, we present the algorithmic framework in Section 4. To demonstrate the feasibility of our approach we simulate subsonic and transonic flows over a NACA0012 airfoil at zero angle of attack and around a AGARD wing 445.6. The numerical results are summerized in Section 5. We conclude the report with remarks and extensions given in Section 6.

2 Simulation of compressible flows

Our interest lies in the numerical simulation of three-dimensional compressible inviscid flows. We assume that there are no external forces or heat transfer. As described above, these flows can be modeled with the Euler equations or with the full potential equation for the particular case when the assumptions of irrotationality and isentropy are satisfied.

2.1 Governing equations

Let $\Omega \subset \Re^3$ be the computational flow domain and Γ its boundary. The conservative form of the Euler equations is given by

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0. \tag{1}$$

Here U contains the conservative variables, i.e., $U = (\rho, \rho u, \rho v, \rho w, \rho E)^T$. The explicit definitions of $F()$ can be found on page 87 of [12]. When the flow is irrotational, there exists a potential variable Φ satisfying the full potential equation

$$\frac{\partial \rho(\Phi)}{\partial t} + \nabla \cdot G(\Phi) = 0, \tag{2}$$

where $G(\Phi) = \rho \nabla \Phi$ and

$$\nabla \Phi = (u, v, w)^T. \tag{3}$$

In the rest of the paper, we shall refer to U as the Euler variable, which is a vector, and Φ as the full potential variable, which is a scalar.

By appealing the isentropic flow assumption we can write the density ρ as a nonlinear function of the potential, such as

$$\rho(\Phi) = \rho_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - \frac{||\nabla \Phi||_2^2}{q_\infty^2})\right)^{1/(\gamma-1)}. \tag{4}$$

Two different types of boundaries must be considered: the farfield boundary and the solid wall boundary. On the solid wall boundary, Γ_w, the normal velocity $\mathbf{v}_n$ is zero, since no mass crosses the boundary. In the farfield, Γ_∞, we impose a uniform free-stream state defined by the following parameters: the density, ρ_∞, the velocity vector, $\mathbf{v}_\infty$, the pressure, p_∞, and the Mach number M_∞. These conditions are given by $\mathbf{v}_n = 0, \quad \text{on} \quad \Gamma_w$ and $\rho_\infty = 1$, $\mathbf{v}_\infty = (\cos(\alpha) \times \cos(\theta), \sin(\theta), \sin(\alpha) \times \cos(\theta))$, $p_\infty = \frac{1}{\gamma M_\infty^2}$ on Γ_∞. Here α and θ are the angles of the flow direction (angle of attack and yaw angle, respectively).

2.2 Euler solver

To solve the Euler equations, we take advantage of an existing code based on an unstructured finite volume discretization [10]. The computational domain is divided into tetrahedra to provide maximum flexibility for tessellating complex geometries. Euler variables are located at the vertices of the elements. This code uses a second-order flux discretization based on the MUSCL (Monotonic Upwind Scheme for Conservative Laws) scheme [16]. The spatial discretization of the boundary conditions is obtained using a non-reflecting version of the flux–splitting scheme [9].

For clarity, we now briefly describe the implicit approach used for solving (1) with boundary conditions given above. For more details we refer the reader to [4, 17]. To ease notations, the subscript h identifies the spatial discretized functions, and Ψ_h denotes the second-order discretization of the convective fluxes. A fully discretized scheme, which is of first order in time and of second-order in space, can be written as

$$\frac{U_h^{n+1} - U_h^n}{\Delta t_h^n} + \Psi_h(U_h^{n+1}) = 0, \tag{5}$$

where n is a running time step index. The local time step size Δt_h^n is defined for each control volume τ_i^c (with characteristic size $\|\tau_i^c\|$) by

$$\Delta t_h^n = \|\tau_i^c\| \frac{\text{CFL}}{C_{\tau_i^c} + \|\mathbf{U}_{\tau_i^c}\|_2}, \tag{6}$$

where CFL is a preselected positive number, $C_{\tau_i^c}$ is the sound speed and $\mathbf{U}_{\tau_i^c}$ is the velocity vector. Suppose we know an initial guess for $U^{n+1,0}$, say U^n. To obtain U^{n+1} we solve (5) using the so–called Defect Correction (DeC) method [2] and form the following linear system of equations

$$\begin{aligned} &\left(\frac{\|\tau_i^c\|}{\Delta t_h^n} + \frac{\partial \Psi_h^{(1st)}(U_h^{n+1,0})}{\partial U}\right)\left(U_h^{n+1,s+1} - U_h^{n+1,s}\right) \\ &= -\left((U_h^{n+1,s} - U_h^n)\frac{\|\tau_i^c\|}{\Delta t_h^n} + \Psi_h^{(2nd)}(U_h^{n+1,s})\right). \end{aligned} \tag{7}$$

Clearly, this approach reduces to Newton's method in the limit of large time steps. We remark that the accuracy of the numerical solution is determined by the time discretization scheme and how the term $\Psi_h^{(2nd)}(U_h^{n+1,s})$ on the right-hand side of (7) is discretized in space. However, the *cost* is mostly determined by how the left–hand side of (7) is constructed and solved. The advantage of this technique is that we can solve a first order problem but still obtain a second-order spatial accuracy by using a first order Jacobian with a second-order residual. To conclude, each of the s DeC iterations of the $(n+1)$st time step is solved using a diagonally preconditioned GMRES approach.

2.3 Full potential solver

A finite volume full potential solver has been developed to interface with our existing Euler solver [6]. In the resulting composite discretization, the same control volume is used and only the flux calculations are different. We briefly review the spatial discretization of the mass flux required in this scheme and then discuss an implicit solution scheme.

2.3.1 Spatial discretization

The integral form of the full potential equation for the control volume τ_i^c is simply

$$\int_{\tau_i^c} \nabla \cdot G(\Phi) \mathrm{d}A = 0. \tag{8}$$

Note that the union of τ_i^c covers the whole domain Ω, i.e., $\overline{\Omega} = \bigcup \overline{\tau_i^c}$. By analogy to the discretization of the Euler equations, the discretization here is accomplished by dividing the domain into tetrahedral elements, $\tau_{i,j}$. The potential variable is stored at the vertices. This choice is illustrated in Figure 1 for two space dimensions. The space of the potential solution is taken to be piecewise linear continuous functions determined by the vertex values Φ_i.

For the control volume τ_i^c associated with the dual mesh, we can write the discrete form of (8) as

$$\begin{aligned} \int_{\tau_i^c} \nabla(\rho \nabla \Phi) dA &= \int_{\partial \tau_i^c} \rho \nabla \Phi \cdot \mathbf{n}\, dS \\ &= \sum\nolimits_{\tau_{i,j}} \rho_{i,j} (\nabla \Phi)_{i,j} \cdot \mathbf{S}_{i,j}^c, \end{aligned} \tag{9}$$

where $\tau_{i,j}$ is the "triangulation" associated with the control volume τ_i^c and $\mathbf{S}_{i,j}^c = \int_{\partial \tau_i^c \cap \tau_{i,j}} \mathbf{n} dA$. Here $\mathbf{n}$ is the unit outward normal vector of the face associated with

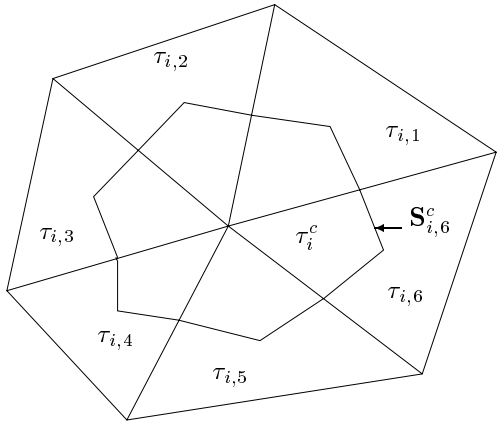

Figure 1: Two space dimensions representation of the control volume.

the control volume τ_i^c in the element $\tau_{i,j}$. Note that $\rho_{i,j}$, the discrete density, is a function of $(\nabla \Phi)_{i,j}$ which is a constant for each element $\tau_{i,j}$.

2.3.2 Implicit approach

To solve (8) we introduce a pseudo–time step. Hence, we rewrite (8) as

$$\frac{d}{dt} \int_{\tau_i^c} \Phi \mathrm{d}A + \int_{\tau_i^c} \nabla \cdot G(\Phi) \mathrm{d}A = 0. \tag{10}$$

Using the DeC method [2, 17], the resulting semi-discrete form of (10) is

$$\begin{aligned} &\left(\frac{\|\tau_i^c\|}{\Delta t_h^n} + \frac{\partial (\Upsilon_h^{(1st)}(\Phi^{n+1,s}))}{\partial \Phi} \right) \left(\Phi_h^{n+1,s+1} - \Phi_h^{n+1,s} \right) \\ &= - \left((\Phi_h^{n+1,s} - \Phi_h^n) \frac{\|\tau_i^c\|}{\Delta t_h^n} + \Upsilon_h^{(2nd)}(\Phi^{n+1,s}) \right), \end{aligned} \tag{11}$$

where Υ_h is the discrete mass flux. In this approach we require the computation of the Jacobian matrix $\partial \Upsilon_i(\Phi_i^{n+1,s})/\partial \Phi$. A finite difference approximation of this Jacobian is introduced. For each pair of indices i, j we define

$$\frac{\partial (\Upsilon_i(\Phi))}{\partial \Phi_j} = \frac{\Upsilon_i(\Phi + \delta \Phi_j) - \Upsilon_i(\Phi)}{\delta \Phi_j}. \tag{12}$$

This first order accurate approach is chosen because a first order Jacobian is sufficient in the DeC method. The resulting matrix is sparse, with contributions from the neighbors of node i only. Therefore, we do not consider this matrix to be computationally very expensive. In fact, if no upwinding is needed the calculation of the

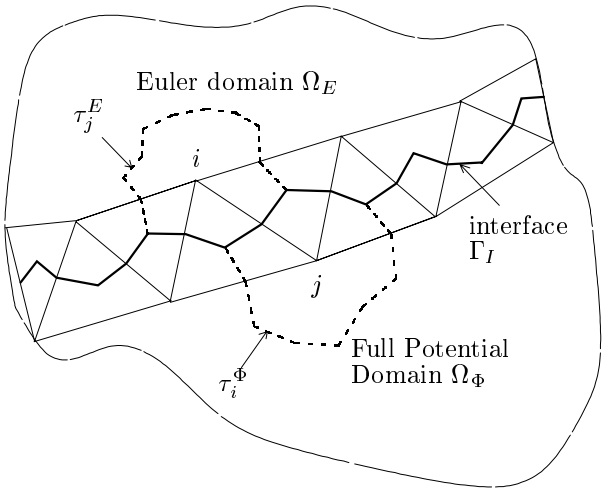

Figure 2: Non–overlapping interface of the Euler and full potential domains.

Jacobian is only about five times more expensive than calculating the mass flux for the meshes we use.

The time step size Δt_h^n is determined from (6) where $C_{\tau_i^c}$ and $\mathbf{U}_{\tau_i^c}$ are calculated from Φ. This time step is adopted to have the same diagonal scaling in the Jacobian matrix associated with the coupled solver. For transonic flows, upwinding is required; therefore, the density is modified to add artificial compressibility [6]. The full potential spatial discretization of the boundary condition is also described in [6].

3 Coupled solver

The computational domain, Ω, is split into two subdomains, Ω_E and Ω_Φ, wherein the Euler equations and the full potential equation are solved, respectively. We denote by Γ_I the interface between Ω_E and Ω_Φ, as shown in Figure 2.

The formulation presented for the full potential is similar to the unsteady Euler formulation for finite volumes. In fact, we can define W as the *simulation variable*, which represents either U or Φ. A general formulation can thus be constructed. In a future version of our software implementation, W will be a pointer and its true value and size will be determined while the flow is calculated. We assume that W is the solution of the equation

$$\frac{\partial W}{\partial t} + \nabla \cdot P(W) = 0, \tag{13}$$

where the flux function P is called the *model function* that equals to either F or G. The decision to choose a specific model will be made for each subdomain.

To simplify the notation, we now introduce

$$g_{n+1,s} = -\left((W_h^{n+1,s} - W_h^n) \frac{\|\tau_i^c\|}{\Delta t_h^n} + \chi_h(W^{n+1,s}) \right), \tag{14}$$

which is the right-hand side of (7) or (11) where χ_h is either Ψ_h or Υ_h. The combined linear DeC system (13) becomes

$$J_n(W_h^{n+1,s+1} - W_h^{n+1,s}) = g_{n+1,s}. \tag{15}$$

Let's examine some characteristics of the matrix J_n. This matrix is sparse with the following non-zero entries:
$5 \times 5 \times$ number of Euler vertices;
$5 \times 5 \times 2 \times$ number of Euler edges;
$1 \times$ number of full potential vertices;
$2 \times$ number of full potential edges;
$5 \times 2 \times$ number of interface edges.
This matrix has two contributions: a diagonal term and the differencing of the fluxes with respect to the full potential and Euler variables which includes the differencing of the full potential mass flux with respect to the Euler variables and the differencing of the Euler fluxes with respect to the potential variable.

3.1 Interface boundary conditions

We reported in [6] two different domain partitioning approaches: overlapping and non–overlapping partitioning. Here we review only the non–overlapping partition and we describe the construction of the Jacobian for our two–model formulation.

In the non–overlapping approach, control volumes are flagged either for solving the full potential equation or for solving the Euler equations. The location of the interface, therefore, lies between the control volumes as in the two dimensional illustration in Figure 2. We identify by $\mathbf{S}_{i,j}^I$ the intersection of the control volumes τ_i^E and τ_j^Φ located on each side of the interface, i.e., in the Euler and the full potential subdomains, respectively. For simplicity, we describe the first order flux calculation. The approximation of the interface flux uses a classical Roe approximate Riemann solver [20] for both control volumes. Let us call $\Psi_{i,j}$ the interface flux which can be calculated from the vortex values of node i and j as

shown below

$$\begin{aligned} \Psi_{i,j} &= \|S^I_{i,j}\| \ \mathcal{F}_{Roe}(U_i, U_j, \mathbf{n}_{i,j}) \\ \mathcal{F}_{Roe}(U_i, U_j, \mathbf{n}_{i,j}) &= \tfrac{1}{2}\left[F(U_i) + F(U_j)\right] \cdot \mathbf{n}_{i,j} + \\ & \quad \tfrac{1}{2}\left[\overline{A}(U_i, U_j, \mathbf{n}_{i,j})(U_i - U_j)\right], \end{aligned} \tag{16}$$

where $\mathbf{n}_{i,j}$ is the outward unit normal of $\mathbf{S}^I_{i,j}$ and $\overline{A}(U_i, U_j, \mathbf{n}_{i,j})$ is the Roe matrix. Note that node j is not in the Euler domain and therefore we need to convert, at this node, the potential Φ to the Euler variable U. For this conversion, we introduce an operator Q related to the transfer of the potential variable to the Euler variable defined as

$$U = Q(\Phi). \tag{17}$$

The calculation of the operator Q is based on the values of the potential, in fact, $\nabla\Phi$ in the potential domain and the volume average of the primitive variables in the Euler domain. The resulting Roe flux is written as

$$\begin{aligned} &\mathcal{F}_{Roe}(U_i, Q(\Phi_j), \mathbf{n}_{i,j}) = \\ &\tfrac{1}{2}\left[F(U_i) + F(Q(\Phi_j))\right] \cdot \mathbf{n}_{i,j} \\ &+\tfrac{1}{2}\left[\overline{A}(U_i, Q(\Phi_j), \mathbf{n}_{i,j})(U_i - Q(\Phi_j))\right]. \end{aligned} \tag{18}$$

Finally, to obtain the flux conservation on τ^E_i the contribution $\Psi_{i,j}$ is added to the other intercell fluxes. In this brief presentation of the interface condition we have not covered the second-order accurate discretization for the Euler model. To obtain a second-order accuracy of the Euler fluxes, the classical MUSCL method by van Leer [16] is used for the flux calculation.

For the flux balance on τ^Φ_j, the mass flux across the interface, Γ_I, is calculated from the average velocity in each of the elements that intersect the interface. To be more precise this mass flux is calculated by using the average velocity in that element. The interface solution can be written as

$$\sum_{\tau_I} \rho V \cdot \mathbf{n}_I S^I_{i,j} + \sum_{\tau_{i,j}} \rho_{i,j} (\nabla\Phi)_{i,j} \cdot \mathbf{n}_\Phi S^\Phi_{i,j} = 0, \tag{19}$$

where ρV is the average momentum in τ_I which is an element that intersects the interface Γ_I.

The principal new ingredient in the implicit scheme (15) is the calculation of the Jacobian. A discrete approximation of the Jacobian is calculated as in (12). However, the multi–model formulation contains other terms, including the interface terms. For a full potential and Euler coupled solver we write

$$J = \frac{D}{\Delta t} + \frac{\partial\Psi(U)}{\partial U} + \frac{\partial\Psi(U)}{\partial\Phi} + \frac{\partial\Upsilon(\Phi)}{\partial U} + \frac{\partial\Upsilon(\Phi)}{\partial\Phi}, \tag{20}$$

where D is the diagonal matrix of cell volumes and Δt is the time step. We have discussed in Section 2.3.2 how to approximate $\partial\Upsilon(\Phi)/\partial\Phi$. In addition, the term $\partial\Psi(U)/\partial U$ is obtained from a classical approximation [12]. Note that $\Psi(U)$ and $\Upsilon(\Phi)$ exist only in their respective domains and that the resulting Jacobian matrix is of the form:

$$\begin{bmatrix} \frac{\partial\Psi(U)}{\partial U} & \frac{\partial\Psi(U)}{\partial\Phi} \\ \frac{\partial\Upsilon(\Phi)}{\partial U} & \frac{\partial\Upsilon(\Phi)}{\partial\Phi} \end{bmatrix}. \tag{21}$$

The calculation of $\partial\Psi(U)/\partial\Phi$ is somewhat less standard. As before, we use a finite–difference approximation to calculate this term in the Jacobian. Our approximation becomes

$$\frac{\partial\Psi_i(U)}{\partial\Phi_j} \approx \frac{\Psi_i(U + \delta\Phi_j) - \Psi_i(U)}{\delta\Phi_j}. \tag{22}$$

Here again, we transfer $\delta\Phi$ to δU using the above described procedure. We calculate the Euler fluxes, Ψ, as in (18). Similarly, the following approximation is used

$$\frac{\partial(\Upsilon_j(\Phi))}{\partial U_i} \approx \frac{\Upsilon_j(\Phi + \delta U_i) - \Upsilon_j(\Phi)}{\delta U_i}. \tag{23}$$

The calculation of $\Upsilon_{i,j}$ is performed by modifying the value of node i with δU_i. Subsequently, the resulting mass flux is added to Υ_j, associated with τ^Φ_j.

We have omited on purpose the description of the upwinding for the full potential solver because it is not needed in our approach as we restrict the full potential domain to subsonic regions. Recall that in the Euler domain, the upwinding is intrinsic to the Roe's scheme.

4 Algorithmic framework

A DeC–Krylov approach is used to obtain a steady state solution of our coupled full potential and Euler equations. We perform a given number of DeC iterations at the $(n+1)$st pseudo-time step. In each iteration we solve the linear problem using a diagonally preconditioned GMRES method such that

$$\|B(J_n(W^{n+1,s+1}_h - W^{n+1,s}_h) - g_{n+1,s})\|_2 \leq \epsilon \|g_{n+1,s}\|_2, \tag{24}$$

where $\epsilon > 0$ is the linear tolerance and B is the diagonal preconditioner for J_n. The DeC iterations are used to decrease the nonlinear residual. Less than the prescribed number of iterations is used if the nonlinear tolerance $\mathcal{T} > 0$ is satisfied, i.e.,

$$\|g_{n+1,s}\|_2 \leq \mathcal{T}\|g_n\|_2. \tag{25}$$

Furthermore, the solution is advanced using a pseudo–time step. The final stopping criterion is

$$\frac{\|g_{n+1,s}\|_2}{\|g_1\|_2} \leq 10^{-5}. \tag{26}$$

The advantage of using this approach is two-fold; first, the calculation of the Jacobian matrix is only performed at the first DeC iteration bringing additional saving and second, we have noticed that the convergence rate is improved for the full potential solver and the coupled solver compared with doing just one DeC iteration, i.e., if we double the number of DeC iterations per time step we observe more than a two-fold reduction in the total number of iterations. We are presently investigating this behavior.

In addition to the above strategy, we use an initial solution for the entire computational domain. This global solution is obtained from two full potential iterations. This allows us to use a higher CFL number which leads to a faster convergence. In general, the CFL number is increased by a constant number at each iteration from an initial value.

5 Results

5.1 First model problem: Flow past a NACA0012 airfoil

In this section we present heterogeneous full potential and Euler solutions in three space dimensions. We test our scheme for a two-dimensional flow over a NACA0012 airfoil in a three-dimensional computational domain. Only half of the geometry is required for this symmetric flow. The computational domain is such that Ω is a rectangle domain of size $0.1 \times 1.5 \times 1.5$, where an upper surface of a NACA0012 is located on the bottom face as in Figure 3. The boundary conditions of this problem are as follows: on Γ_1, Γ_2, and Γ_3 we impose farfield conditions; on Γ_4, Γ_6, Γ_7, and Γ_8 we impose the nonpenetration condition for symmetry and on Γ_5 we impose the solid wall condition. Note that the implementation of the nonpenetration and the solid wall conditions are identical. For the farfield boundary Γ_1, Γ_2, and Γ_3, the full potential discretization of this condition is different for Γ_2 where we impose Φ_∞ as a Dirichlet condition, and for Γ_1 and Γ_3, where we only specify a flux on the boundary. The computational mesh which contains 16200 nodes is presented in Figure 4. For parallel processing, this mesh is partitioned into eight or twelve subdomains, respectively, for the transonic model problem and the subsonic model problem. Each of the subdomains have roughly the same number of mesh points even though ultimately the node distribution per processor should take into account the type of solvers used, since cost reduction in the full potential regions is one of the main motivations of the work.

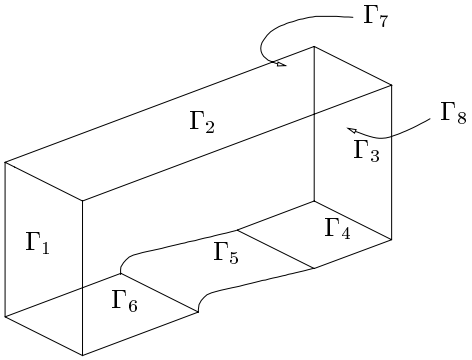

Figure 3: Computational domain for the NACA0012 geometry.

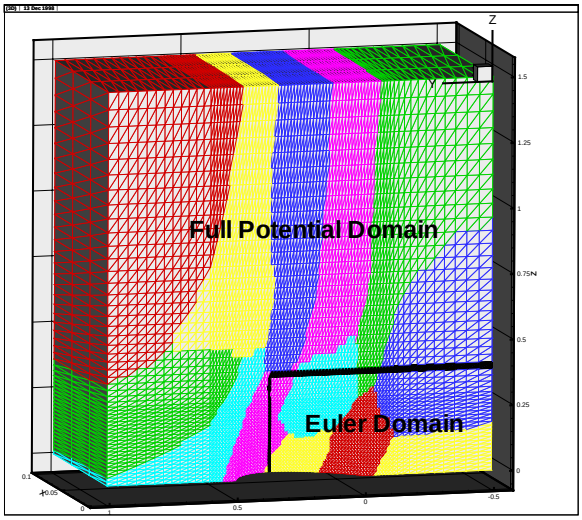

Figure 4: Computational mesh and partition for the NACA0012 airfoil; subsonic flow.

5.1.1 Subsonic flow

We investigate a subsonic flow around the NACA0012 wing at $M_\infty = 0.5$ mainly to analyze the leading edge solution sensitivity to low Mach numbers. The domain partitioning of this mesh into the Euler domain and the full potential domain is shown in Figure 4: the lower right domain is the Euler domain and the rest is the full potential domain which contain 4576 and 11624 nodes, respectively. In Figure 5, the Mach number contours, values of 0.33 to 0.60 with an increment of 0.01, show reasonable agreement at the interface. In Figure 6, the pressure distribution on the airfoil associated with this solution is compared with results obtained for the computer program TRANAIR used for aerodynamic analysis and design at The Boeing Company [24]. We note the good agreement between our full potential solution and the full potential solution from TRANAIR. Also note the numerical entropy generated by the Euler solver, in the vicinity of the stagnation point, which appears as a peak in the pressure distribution at the leading edge. This feature is not present when using the coupled model. Indeed, the full potential solver is more accurate in regions when the Mach number is small.

Figure 7 compares the nonlinear iterations of the various solvers: the full potential model, the Euler model and the coupled model. The CFL number is 100 at the first iteration and is increased by 50 on each iteration thereafter. Note that the full potential solver is very fast. The solution is obtained in three iterations with a residual less than 10^{-8}. However, when this solver is coupled with the Euler solver, the number of iterations increased to 167. This is far more than the 58 pseudo-time iterations needed in the Euler model. In this paper we do not show the total CPU time comparison, because, at present we have not yet implemented a load balancing procedure which is mandatory for such a comparison.

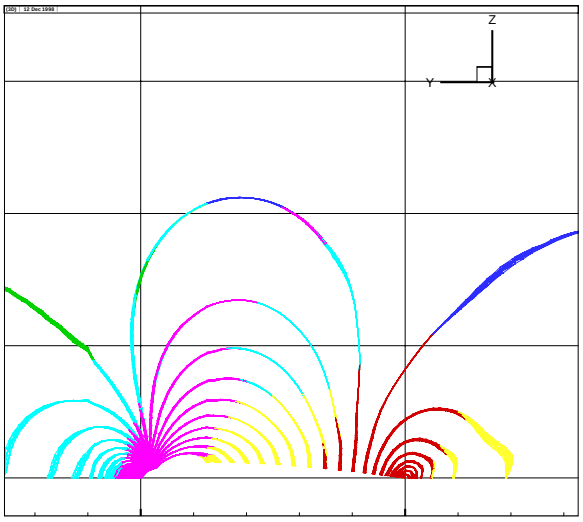

Figure 5: Mach number contours for the NACA0012 airfoil at $M_\infty = 0.5$.

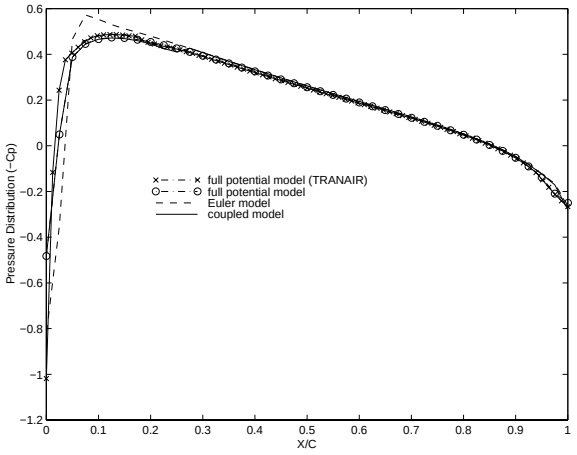

Figure 6: Pressure coefficient distribution on a NACA0012 airfoil at $M_\infty = 0.5$.

5.1.2 Transonic flow

To illustrate the multi–model formulation for transonic problems, we investigate a flow around the NACA0012 wing at $M_\infty = 0.8$. The domain is partitioned along a constant z plane located at $Z = 0.35$. The Euler domain (bottom) and the full potential domain (top) contain the same number of nodes. This simulation is performed on 8 processors. In Figure 8 and Figure 9, we show the Mach number contours, values of 0.37 to 1.24 with an increment of 0.03, for the Euler solution and the coupled solution, respectively. Note the smooth transition of the isocontours between the Euler and full potential domains in the latter. We also report that the pressure coefficient

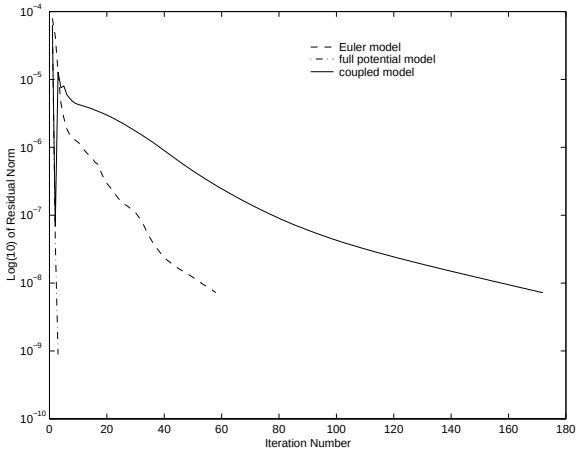

Figure 7: Convergence iteration history for subsonic flow around the NACA0012.

distribution over the airfoil is almost unchanged between the Euler and the coupled solution, Figure 10. The full potential solution obtained from TRANAIR is also presented in this figure. As expected the full potential shock is further back and stronger.

The Euler solution is obtained using a CFL history evolution different from the coupled solution. The CFL number is increased by 30 instead of 50 otherwise the convergence is not possible. Similarly to the subsonic case, the performance of the coupled nonlinear solver, in terms of number of iterations presented in Figure 11, is less desirable. The factors leading to this slow convergence are presently under investigation.

5.2 Second model problem: Flow over a AGARD wing 445.6

We now simulate a three-dimensional flow over a AGARD wing 445.6 at $M_\infty = 0.84$ with an angle of attack of 3.06^o. Details of this wing can be found in [22]. The numerical discretization is performed on an unstructured mesh containing 22014 nodes; the surface mesh is presented in Figure 12. This mesh is decomposed in two domains, i.e., full potential and Euler domains, as shown in Figure 12.

The Mach number isocontours (from 0.60 to 1.95 with an increments of 0.05) on the surface of the wing and on the symmetry plane is presented in Figure 14 for the coupled solution and in Figure 13 for the Euler solution. We report that the solutions in both cases are almost identical. The iteration history, Figure 15, indicates the slow convergence of the coupled solver. The procedure,

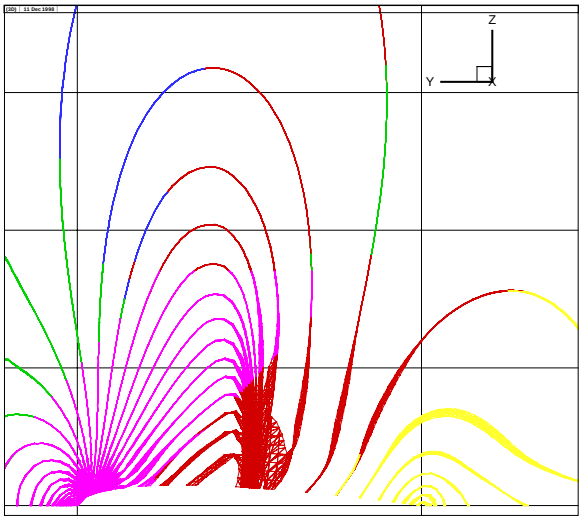

Figure 8: Mach number contours for the NACA0012 airfoil at $M_\infty = 0.8$; Euler solution.

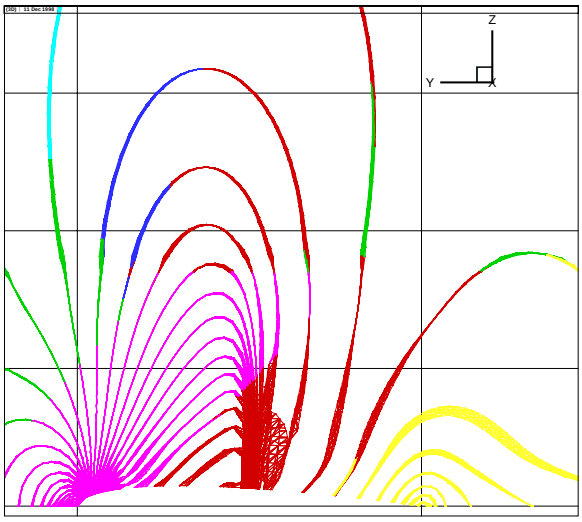

Figure 9: Mach number contours for the NACA0012 airfoil at $M_\infty = 0.8$; coupled solution.

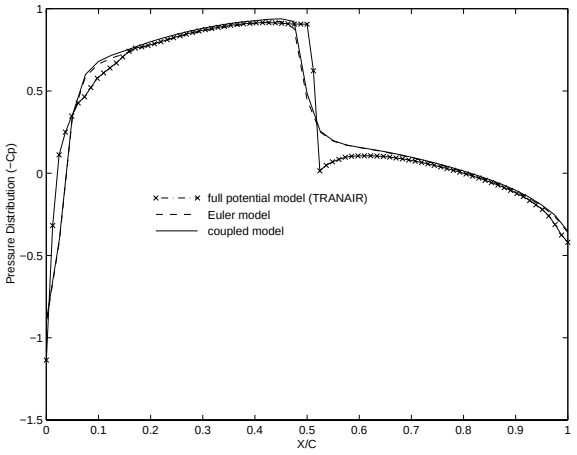

Figure 10: Pressure coefficient distribution on a NACA0012 airfoil at $M_\infty = 0.8$.

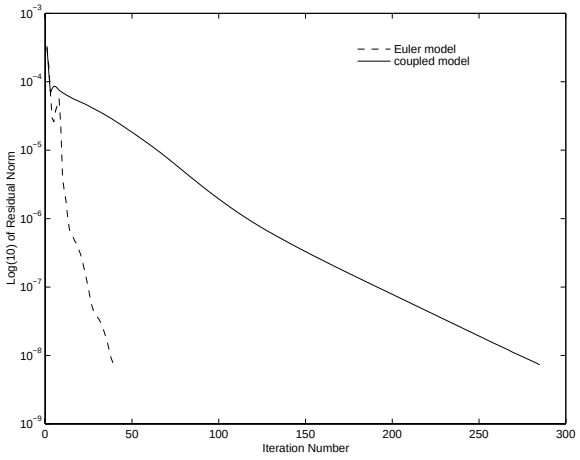

Figure 11: Comparison of the convergence in terms of number of iterations for Euler and the coupled solution.

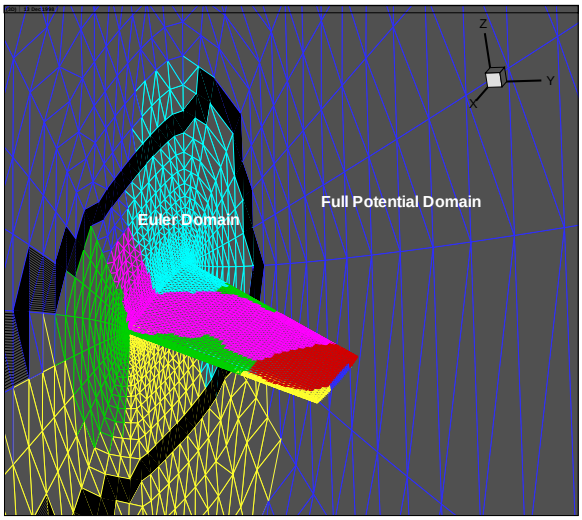

Figure 12: Surface mesh and domain partitioning for the AGARD wing 445.6.

in both cases, uses five DeC iterations and a CFL number which is increased by ten at each pseudo-time step. The first two iterations only solve the full potential equation before activating each of the solvers.

6 Remarks and conclusions

Before concluding, we comment on our analytical and numerical experience of the implicit coupling of the full potential equation and the Euler equations.

First, we report that the solution is smooth across the interface which indicates that our conservative coupling approach is well suited for coupling of the full potential and the Euler equations.

Second, physical solutions for potential flow over a wing are obtained by imposing the Kutta condition; that the flow leaves the trailing edge smoothly. For a full potential solver such a condition is enforced by adding a jump in the potential equal to the circulation. For non–lifting airfoils, such as in the first model problem, we do not need to enforce the Kutta condition. For complete lifting wings, such as the flow over the AGARD wing 445.6 in the second model problem , it is fortunate that the Euler solution intrinsically respects this condition. For such lifting wings we define the Euler domain to cover the wing and the wake region. This partition avoids any special treatment in the full potential domain because the full potential region does not cross the trailing edge vortex sheet.

Finally, we discuss the convergence. In general, the

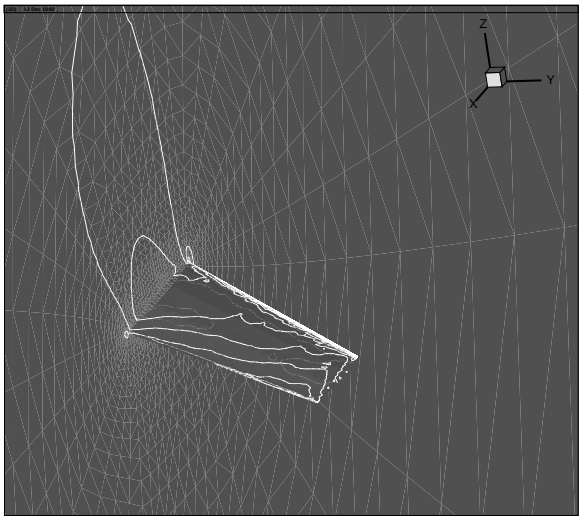

Figure 13: Mach number isocontours on the surface of the AGARD wing 445.6; Euler solution.

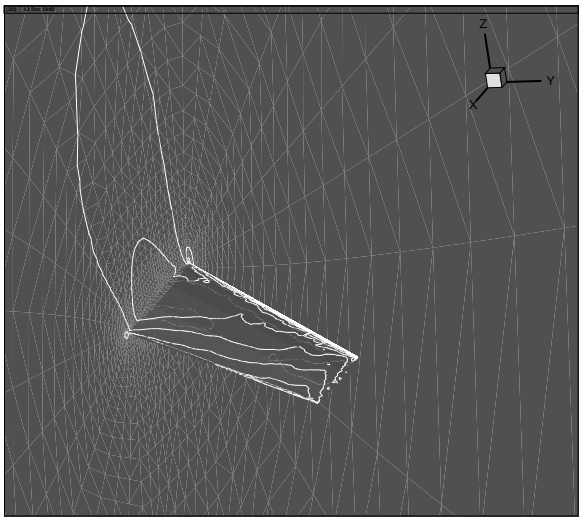

Figure 14: Mach number isocontours on the surface of the AGARD wing 445.6; the coupled solution.

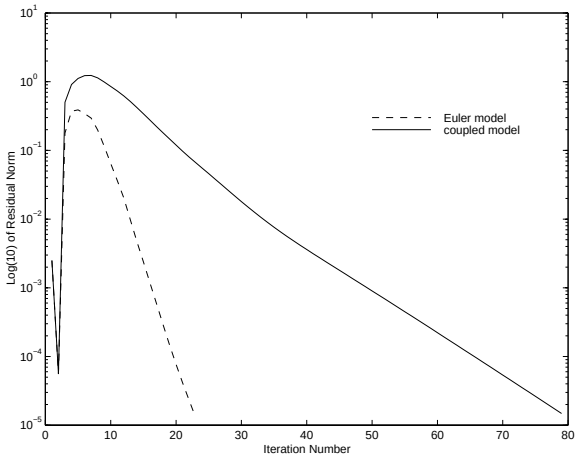

Figure 15: Iteration history for the AGARD wing 445.6.

Jacobian matrix J is ill–conditioned and Krylov methods need a large number of iterations. It is clear that the introduction of a preconditioner is mandatory. A Krylov–Schwarz method, such as a restricted additive Schwarz [7] preconditioned GMRES, is being studied in part because of good data locality which is an advantage for parallel computing.

Other more practical issues are related to automate the multi-model formulation leading to additional reduction in computation time and memory. We plan to develop a procedure to automatically position the interface based on the existing field variables (i.e., dynamic zonal configuration). In addition, load balancing for parallel computations is needed in the proposed software development to take full advantage of parallel computing, in particular when dynamic zonal configuration procedures are used. Recall that different partial differential equations are solved in different regions but each equation does not require the same number of operations. One palliative is to breakdown each region into subregions equal to the number of processors. By using this decomposition each processor will contain one subregion of each type and therefore executes the same number of operations. A novel multi–level model approach to increase the convergence rate is under investigation. Such an approach uses different solvers not only in different areas of the computational domain but also at different computational steps to obtain a faster solution which is later improved upon by a more accurate model.

In conclusion, the goal of this research is to develop a general-purpose software to calculate numerical solutions of compressible three-dimensional flows with a sig-

nificant reduction in computation time and memory. A multi-model formulation based on unstructured grids is chosen due to their geometric flexibility. To be competitive with implicit approaches that run on multiple processors, the proposed software must also include the most advanced solution methods and parallel computing techniques. Numerous extensions can be incorporated in this software to address more types of applications. For example, additional models can be coupled into the iterative framework such as structural models for aeroelastic applications.

We have showed herein the extension of the explicit multi–model formulation to an implicit approach. At this point we note that both the linear and the nonlinear convergence are slow. Implementation of a restricted additive Schwarz [7] should improve the linear convergence. Further research is needed to analyze and improve the nonlinear convergence.

Acknowledgments. The work was supported in part by the NSF grants ASC-9457534, ECS-9527169, and ECS-97255 04.

References

[1] M. E. Berkman, L. N. Sankar, C. R. Berezin, and M. S. Torok. Navier–Stokes/full potential/free–wake method for rotor flows. J. of Aircraft, 34:635–640, 1997.

[2] K. Bohmer, P. Hemker, and H. Stetter. The defect correction approach. Comput. Suppl., 5:1–32, 1984.

[3] M. Bristeau, R. Glowinski, J. Periaux, P. Perrier, O. Pironneau, and C. Poirier. On the numerical solution of nonlinear problems in fluid dynamics by least squares and finite element methods (ii), application to transonic flow simulations. Comput. Meths. Appl. Mech. Engrg., 51:363–394, 1985.

[4] X.-C. Cai, C. Farhat, and M. Sarkis. A minimum overlap restricted additive Schwarz preconditioner and applications in 3D flow simulations. In C. Farhat J. Mandel and X.-C. Cai, editors, The Tenth International Conference on Domain Decomposition Methods for Partial Differential Equations. AMS, 1998.

[5] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young. Parallel Newton–Krylov–Schwarz algorithms for the transonic full potential equation. SIAM J. Sci. Comput., 19:246–265, 1998.

[6] X.-C. Cai, M. Paraschivoiu, and M. Sarkis. An explicit multi–model compressible flow formulation based on the full potential equation and the Euler equations. In Proc. of the Eleventh Intl. Conference on Domain Decomposition Methods in Scientific and Engineering Computing, 1999. To appear.

[7] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comput., 1999. To appear.

[8] R. Cummings, Y. Rizk, L. Schiff, and N. Chaderjian. Navier–Stokes predictions for the F–18 wing and fuselage at large incidence. J. of Aircraft, 29:565–574, 1992.

[9] C. Farhat and S. Lanteri. Simulation of compressible viscous flows on a variety of MPPs: Computational algorithms for unstructured dynamic meshes and performance results. Comput. Meths. Appl. Mech. Engrg., 119:35–60, 1994.

[10] C. Farhat, M. Lesoinne, P. S. Chen, and S. Lantéri. Parallel heterogeneous algorithms for the solution of the three–dimensional transient coupled aeroelastic problems. AIAA Paper 95-1290, 1995.

[11] M. Hafez, J. South, and E. Murman. Artificial compressibility method for numerical solution of the transonic full potential equation. AIAA J., 17:838–844, 1979.

[12] C. Hirsch. Numerical Computation of Internal and External Flows. John Wiley & Sons, New York, 1990.

[13] T. L. Holst, J. W. Slooff, H. Yoshihara, and W. F. Ballhaus. Applied computational transonic aerodynamics. Technical Report AG-266, AGARD, 1982.

[14] A. Jameson. Essential elements of computational algorithms for aerodynamics. Technical Report 97-68, ICASE, 1997.

[15] A. Jameson, T. Baker, and N. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103, 1986.

[16] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second order sequel to Godunov's method. J. Comput. Phys., 32:361–370, 1979.

[17] R. Martin and H. Guillard. A second order defect correction scheme for unsteady problems. Computers & Fluids, 25(1):9–27, 1996.

[18] R. B. Pelz and A. Jameson. Transonic flow calculations using triangular finite elements. AIAA Paper 83–1922, 1983.

[19] A. Quarteroni and A. Valli. Domain Decomposition Methods for Partial Differential Equations. Oxford University Press, Oxford, 1999.

[20] P. Roe. Approximate Reimann solver, parameter vectors, and difference schemes. J. Comput. Phys., 43:357–372, 1981.

[21] L. N. Sankar, B. K. Bharadvaj, and F. Tsung. Three-dimensional Navier–Stokes/full potential coupled analysis for transonic viscous flow. AIAA J., 31:1857–1862, 1993.

[22] E. C. Yates. AGARD standard aeroelastic configuration for dynamic response, candidate configuration I. Wing 445.6. Technical Report TM-100492, NASA, 1987.

[23] D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, and S. S. Samant. Global convergence of inexact Newton methods for transonic flow. Int. J. Numer. Meths. Fluids, 11:1075–1095, 1990.

[24] D. P. Young, R. G. Mervin, M. B. Bieterman, F. T. Johnson, S. S. Samant, and J. E. Bussoletti. A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics. J. Comput. Phys., 92:1–66, 1991.

A Transfer Operators

To carry out our multi–model formulation, we do not require a transfer operator from potential to Euler $R : U \longrightarrow \Phi$. Only the mass flux, given by $G(R(U))$, is required to update the potential in each control volume.

The full potential to Euler transfer operator is defined as the map

$$Q : \Phi \longrightarrow U. \tag{27}$$

Recall that U has five components. To obtain its first component, we appeal to

$$\rho(\Phi) = \rho_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - \frac{||\nabla \Phi||_2^2}{q_\infty^2})\right)^{1/(\gamma-1)}. \tag{28}$$

The next three components can be computed with relation (3). The last component ρE is computed via

$$\rho E = \rho(e + \frac{u^2 + v^2 + w^2}{2}) = \frac{p}{\gamma - 1} + \rho \frac{u^2 + v^2 + w^2}{2},$$

where the pressure satisfies the isentropic equation of state

$$p = p_\infty \left(\frac{\rho}{\rho_\infty}\right)^\gamma.$$