NASA Cooperative Agreement NNX07AT97A

Final Report

Feasibility Study of a Unified CFD-CSD Computational Formulation

Principal Investigator: Dr. Bassam Mussaffar
bassam.musaffar-1@nasa.gov

Dr. Venkateswaran Sankaran
vsankaran@merlin.arc.nasa.gov

Dr. Jayanarayanan Sitaraman
jsitaram@uwyo.edu

Dr. Charbel Farhat
cfarhat@stanford.edu

30 June 2010

Abstract

This work concerns the formulation of a unified coupled solution procedure for fluids-structures problems. Although the eventual goal is to apply the methodology to rotorcraft applications, the present study is focused on fundamental algorithmic issues and testing. Specifically, we present the formulation of the structural dynamics using a common finite-volume infrastructure. The same flux procedures and time-iterative framework is used for both the fluids and the structural equations. Moreover, a set of pseudo-structural equations are used to describe the motion of the fluid dynamic mesh to conform to the deforming solid surface. We study the numerical properties of the structural dynamics systems including the eigenvalues or wave speeds and use this information to formulate appropriate computational solution strategies. The unified algorithm adopts a three-solver partitioning---one each for the fluids, mesh and structures---that are embedded within a dual-time-based sub-iterative framework. At each physical time-level, we iterate between the three solvers to insure that the interface conditions are properly represented at the correct time instance. Each solver module is implemented in Fortran-90 and wrapped within a Python environment. Python scripts control the execution sequence of the solver modules and the transfer of interface data between the modules. Systematic testing of the structures, fluids and mesh modules are carried out to verify that each component is operating correctly. Following this, detailed testing of the coupled fluid-structure system is presented for representative one-dimensional cases. In all cases, grid resolution studies and order of accuracy verification are carried out. Comparisons of the coupled method with the conventional serial staggered (CSS) and generalized serial staggered (GSS) schemes are also presented.

Table of Contents

1.0 Introduction

1.1 Background

For many aerospace applications, successful computational design and analysis involves robust, accurate and efficient coupling between computational fluid dynamics (CFD) of the external aerodynamics and computational structural dynamics (CSD) of the airframe. This is particularly the case for rotorcraft applications, wherein the coupling between fluids and structures is critical to the successful prediction of aircraft performance, structural fidelity, stability, maneuverability, safety and noise [2]. Previous CFD-CSD studies in rotorcraft applications have involved coupling dissimilar techniques for the fluids and structures (usually with legacy codes), and has been largely limited to steady flight and benign maneuvers [3-6]. Moreover, structural solutions in rotorcraft problems are typically obtained using beam models and high-fidelity structural solutions have not been used in rotorcraft simulations [7-9]. Given these challenges, the present research is targeted towards investigating a fundamentally different approach for fluid-structure coupling---one based on a unified and high-fidelity computational paradigm for fluids, structures and mesh motions.

A unified coupled solution framework promises many fundamental advantages. Specifically, it formulates the fluid and structural equations as a coupled system from the outset and ensures proper conservation of fluxes at the fluid-structure interface, uniform temporal and spatial discretization accuracy, and strong coupling of the governing equations for robustness and solution efficiency. Moreover, it facilitates the continued evolution of fundamental CFD and CSD methodologies by making enhancements automatically available to both equation systems. Above all, the inherent robustness of the unified approach would broaden the range of fluid-structures problems that can be successfully tackled.

The proposed unified computational platform comprises of:

• Common discretization framework based on finite volume procedures for the constituent partial differential equations representing fluid flow, structural deflections and mesh motions,

• Coupled time-integration, linearization and solution strategies,

• Modular-CFD infrastructure for the solution of different sets of partial differential equations, and

• Python-based framework and domain connectivity tools to provide the software infrastructure for time-advancement of the CFD solutions

The research focus is on fundamental algorithmic issues of conservation, robustness, efficiency and accuracy for moving and deforming body problems, including appropriate preconditioning for stiff time-scales and fully implicit solutions for robustness. The resulting computational platform is tested and validated using a variety of 1D test problems, and the formulation is also extended to the multi-dimensional system and preliminary two-dimensional results are presented.

1.2 Research Objectives and Scope

The overall objective of the proposed research is to advance the state of the art in CFD-CSD coupling algorithms for aerodynamics with eventual application to rotorcraft problems. The current study consists of a feasibility study geared towards developing the fundamental algorithmic and code infrastructures and testing them for representative test problems.

The specific objectives of this work are:

1. Fundamental CFD/CSD algorithm development: The theoretical basis of CFD-CSD coupling schemes is established by expressing the structural and fluid dynamics and mesh systems in a common finite volume framework. Importantly, the structural equations are used to describe the motion and deformation of the elastic solid material as well as the motion and deformation of a fictitious material that represents the fluid-dynamic mesh. Thus, the complete system of equations involves the solution of the fluid dynamics and pseudo-structural dynamics in the aerodynamics zone and the physical structural equations in the solid-dynamics zone. The equations in the two zones communicate through appropriate interface conditions.

Analysis of the eigenvalues and integral finite-volume form of the structural dynamics equations are presented to under-score the commonality with the fluid-dynamics equations. This is given in both 1D and 2D to systematically address multi-dimensional formulation issues. The full equation system is then formulated as a single unified system and the algorithms to discretize and solve the equations are presented. Specific issues related to satisfying the discrete Geometric Conservation Law (GCL) [24,25], the formulation of appropriate preconditioning scalings for iterative solution [10-13] and interfacial equations are also discussed. Finally, the coupled approach is also contrasted with classical and generalized staggered solution schemes [14, 15, 18-21].

2. Development of a unified computational code: The unified algorithm is implemented within a common computational platform that is based upon finite-volume integration of the fluid dynamics, structural dynamics and mesh motion equations. A modular code structure is used to automatically allow for the solution of varying numbers of partial differential equations for the fluids and structural domains. The different systems share the same discretization and time-integration procedures, which simplifies the code development process and avoids unnecessary duplication of common elements such as flux discretization and implicit solution.

3. Development of python-based framework infrastructure: The fluids, structures and mesh motion codes are combined together using a python-based framework. The Python scripts are used to carry out all data communication between the individual modules, and, moreover, they carry out the necessary time-integration process of the fluids and structural zones. This is done in such a manner as to ensure that the equations are solved in a fully-coupled fashion at the non-linear level.

4. Computational verification and validation: The developed platform is systematically tested for a variety of problems. First, simple verification tests are used to check the individual fluids and structural modules. Secondly, the framework is applied to a one-dimensional shock-tube fitted with an elastic material on one-end. The coupled solutions are used to assess temporal accuracy, robustness, GCL issues and compare against segregated solution techniques. Future work will address extensions of these studies to rotorcraft problems [16,17].

2.0 Structural Dynamics Equations

Fundamental development of any coupled system of equations requires that the system be treated within a single unified framework. Such an approach ensures that all potential interactions and coupling between the constituent equations are automatically reflected in the computational procedures. It is important to note that this is true even if the coupled systems are eventually solved in a partitioned manner in the final solution algorithm (for instance, due to parallel domain decomposition). Accordingly, we start by expressing the governing equations in a common vector form and by describing a unified finite-volume discretization of these equations. We pay special attention to the structural dynamics equation because the finite-volume form of these equations is relatively novel and their numerical characteristics are not well-established in the literature. We then discuss the details of the unified algorithm including the fully implicit time-integration, linearization and solution schemes, as well as issues related to the Geometric Conservation Law, preconditioning and fluid-structural interface conditions.

2.1 Differential Form of the Structural Dynamics Equations

We start with the structural system written in differential form:

$$\frac{\partial Q_s}{\partial t} + \frac{\partial E_{s,i}}{\partial x_i} = \frac{\partial V_{s,i}}{\partial x_i} + H_s \quad \text{.....(2.1)}$$

where:

$$Q_s = \begin{pmatrix} \rho^s u_j^s \\ d_j^s \end{pmatrix} \qquad E_{s,i} = \begin{pmatrix} \rho^s u_i^s u_j^s \\ u_i^s d_j^s \end{pmatrix} \qquad V_{s,i} = \begin{pmatrix} \tau_{ij}^s \\ 0 \end{pmatrix} \qquad H_s = \begin{pmatrix} f_j^s + b_j^s + c u_j^s \\ u_j^s + d_j^s \nabla \cdot \vec{V}^s \end{pmatrix}$$

.....(2.2)

and:

$$\tau_{ij}^s = \frac{E^s}{2(1+\nu^s)}\left(\frac{\partial d_i^s}{\partial x_j} + \frac{\partial d_j^s}{\partial x_i}\right) + \delta_{ij}\frac{E^s \nu^s}{(1+\nu^s)(1-2\nu^s)}\nabla \cdot \vec{d^s} \quad \text{.....(2.3)}$$

Specifically, the 2D structural equations become:

$$\frac{\partial \rho^s u^s}{\partial t} + \frac{\partial \rho^s {u^s}^2}{\partial x} + \frac{\partial \rho^s u^s v^s}{\partial y} = \frac{\partial}{\partial x}\left[\left(E' + \frac{E'\nu}{1-2\nu}\right)\frac{\partial d_x^s}{\partial x} + \frac{E'\nu}{1-2\nu}\frac{\partial d_y^s}{\partial y}\right] + \frac{\partial}{\partial y}\left[\frac{E'}{2}\frac{\partial d_x^s}{\partial y} + \frac{E'}{2}\frac{\partial d_y^s}{\partial x}\right]$$

...(2.4)

$$\frac{\partial \rho^s v^s}{\partial t} + \frac{\partial \rho^s u^s v^s}{\partial x} + \frac{\partial \rho^s {v^s}^2}{\partial y} = \frac{\partial}{\partial x}\left[\frac{E'}{2}\frac{\partial d_x^s}{\partial y} + \frac{E'}{2}\frac{\partial d_y^s}{\partial x}\right] + \frac{\partial}{\partial y}\left[\frac{E'\nu}{1-2\nu}\frac{\partial d_x^s}{\partial x} + \left(E' + \frac{E'\nu}{1-2\nu}\right)\frac{\partial d_y^s}{\partial y}\right]$$

...(2.5)

where we have defined $E' = E^s/(1+\nu)$ for convenience.

The remaining equations which close the above system are simply given as:

$$\frac{Dd}{Dt} = u \quad \text{.....(2.6)}$$

$$\frac{Dd_y^s}{Dt} = v^s \quad \text{.....(2.7)}$$

Note the use of the substantial derivative. In other words, the velocity of a particle (in the solid medium) is given by the rate of change of the displacement following the particle. This formulation will be clarified once below when we examine the equations in their more familiar scalar transport forms. Or:

$$\frac{\partial d_x^s}{\partial t} + u^s \frac{\partial d_x^s}{\partial x} + v^s \frac{\partial d_x^s}{\partial y} = u^s \quad \text{.....(2.8)}$$

$$\frac{\partial d_y^s}{\partial t} + u^s \frac{\partial d_y^s}{\partial x} + v^s \frac{\partial d_y^s}{\partial y} = v^s \quad \text{.....(2.9)}$$

Alternately, we can write in the following conservative form:

$$\frac{\partial d_x^s}{\partial t} + \frac{\partial u^s d_x^s}{\partial x} + \frac{\partial v^s d_x^s}{\partial y} = u_s + d_x^s \nabla \cdot \vec{V}^s \quad \text{.....(2.10)}$$

$$\frac{\partial d_y^s}{\partial t} + \frac{\partial u^s d_y^s}{\partial x} + \frac{\partial v^s d_y^s}{\partial y} = v^s + d_y^s \nabla \cdot \vec{V}^s \quad \text{.....(2.11)}$$

which is the form given in Eqn. (2.1).

Generalized Coordinates

We now transform from the fixed Cartesian coordinate frame to a generalized space-time coordinate system. The situation may be visualized as follows (shown in 1D for representation simplicity). The figure on the left shows a one-dimensional deforming solid material. We note that the displacement of the points results in a skewing of the cartesian grid. This may now be transformed to a generalized coordinate system, which may be so defined that it retains its rigid shape as shown. In other words, the transformed grid is defined as a mapping of the deformed material onto a constant length.

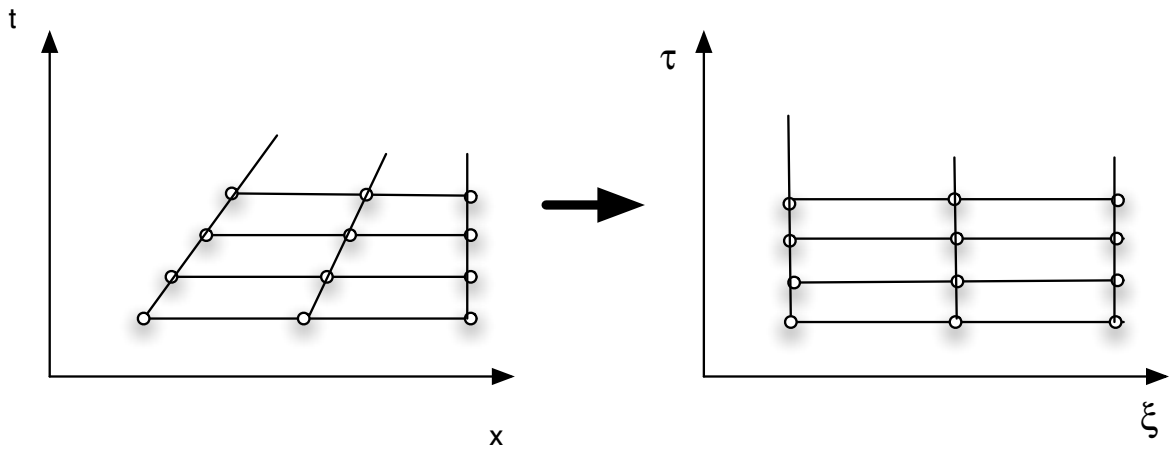

Figure 2.1: Schematic of physical space-time mesh and generalized space-time mesh.

In two dimensions, this transformation may be written as:

$$\xi = \xi(x, y, t) \qquad \eta = \eta(x, y, t) \qquad \tau = t \quad \text{.....(2.12)}$$

Note that the τ is the transformed time-coordinate.

We then have:

$$d\xi = \xi_x dx + \xi_y dy + \xi_t dt \quad \text{.....(2.13)}$$

$$d\eta = \eta_x dx + \eta_y dy + \eta_t dt \quad \text{.....(2.14)}$$

$$d\tau = \tau_x dx + \tau_y dy + \tau_t dt \quad \text{.....(2.15)}$$

Note: $\tau_x = \tau_y = 0$.

Taking partial derivatives wrt to ξ, η and τ and since $\tau_x = \tau_y = 0$, we get:

$$\begin{pmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & \tau_t \end{pmatrix} \times \begin{pmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & t_\tau \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{.....(2.16)}$$

We can define the determinant of transformation Jacobian (first term on the LHS):

$$J = \tau_t(\xi_x \eta_y - \xi_y \eta_x) = \xi_x \eta_y - \xi_y \eta_x \quad \text{.....(2.17)}$$

Solving the above system, we get:

$$x_\xi = \eta_y / J \qquad y_\xi = -\eta_x / J \qquad x_\eta = -\xi_y / J \qquad y_\eta = \xi_x / J \quad \text{.....(2.18)}$$

$$x_\tau = (\xi_y \eta_t - \eta_y \xi_t)/J \qquad y_\tau = -(\xi_x \eta_t - \eta_x \xi_t)/J \quad \text{.....(2.19)}$$

The last two terms can be recognized to be the mesh (or in our case the structural) velocities:

$$x_\tau|_{\xi,\eta} = u^s \qquad y_\tau|_{\xi,\eta} = v^s \quad \text{.....(2.20)}$$

We can likewise write the above system in inverse form as:

$$\begin{pmatrix} x_\xi & y_\xi & 0 \\ x_\eta & y_\eta & 0 \\ x_\tau & y_\tau & t_\tau \end{pmatrix} \times \begin{pmatrix} \xi_x & \eta_x & 0 \\ \xi_y & \eta_y & 0 \\ \xi_t & \eta_t & \tau_t \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{.....(2.21)}$$

The determinant of the inverse transformation is:

$$J' = x_\xi y_\eta - x_\eta y_\xi \quad \text{.....(2.22)}$$

Substituting for $x_\xi,\ y_\xi,\ x_\eta$ and y_η, we get:

$$J' = x_\xi y_\eta - x_\eta y_\xi = \Big(\eta_y \xi_x - (-\xi_y)(-\eta_x)\Big)/J^2 = J/J^2 = 1/J \quad \text{.....(2.23)}$$

We further recognize that J' equivalently describes the cell-volume.

Finally, we can solve the above system to determine the inverse variables. The only useful ones in addition to those already described in Eqns. (2.18) and (2.19) are:

$$\xi_t = J(y_\tau x_\eta - x_\tau y_\eta) \qquad \eta_t = J(x_\tau y_\xi - y_\tau x_\xi) \quad \text{.....(2.24)}$$

With that algebra, we are now ready to transform the equations of motion to the generalized coordinate system:

$$\frac{\partial}{\partial t} = \xi_t \frac{\partial}{\partial \xi} + \eta_t \frac{\partial}{\partial \eta} + \frac{\partial}{\partial \tau} \quad \text{.....(2.25)}$$

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} \quad \text{.....(2.26)}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} \quad \text{.....(2.27)}$$

We next apply this transformation to the structural system.

Transformed Equations

The 2D structural equations of motion from Eqn. (2.1):

$$\frac{\partial Q^s}{\partial t} + \frac{\partial E^s}{\partial x} + \frac{\partial F^s}{\partial y} = \frac{\partial V_x^s}{\partial x} + \frac{\partial V_y^s}{\partial y} + H^s \quad \text{.....(2.1)}$$

In transformed coordinates:

$$\frac{\partial Q^s}{\partial \tau} + \xi_t \frac{\partial Q^s}{\partial \xi} + \eta_t \frac{\partial Q^s}{\partial \eta} + \xi_x \frac{\partial E^s}{\partial \xi} + \eta_x \frac{\partial E^s}{\partial \eta} + \xi_y \frac{\partial F^s}{\partial \xi} + \eta_y \frac{\partial F^s}{\partial \eta} = \xi_x \frac{\partial V_x^s}{\partial \xi} + \eta_x \frac{\partial V_x^s}{\partial \eta} + \xi_y \frac{\partial V_y^s}{\partial \xi} + \eta_y \frac{\partial V_y^s}{\partial \eta} + H^s \quad \text{.....(2.28)}$$

It is convenient to divide through by J and then put the terms in conservative form:

$$\frac{\partial Q^s/J}{\partial \tau} + \frac{\partial}{\partial \xi}\left(Q^s\xi_t + E^s\xi_x + F^s\xi_y\right)/J + \frac{\partial}{\partial \eta}\left(Q^s\eta_t + E^s\eta_x + F^s\eta_y\right)/J - \frac{\partial}{\partial \xi}\left(V_x^s\xi_x + V_y^s\xi_y\right)/J - \frac{\partial}{\partial \eta}\left(V_x^s\eta_x - V_y^s\eta_y\right)/J - H/J = Q^s\frac{\partial(1/J)}{\partial \tau} + Q^s\frac{\partial \xi_t/J}{\partial \xi} + E^s\frac{\partial \xi_x/J}{\partial \xi} + F^s\frac{\partial \xi_y/J}{\partial \xi} + Q^s\frac{\partial \eta_t/J}{\partial \eta} + E^s\frac{\partial \eta_x/J}{\partial \eta} + F^s\frac{\partial \eta_y/J}{\partial \eta} - V_x^s\frac{\partial \xi_x/J}{\partial \xi} - V_y^s\frac{\partial \xi_y/J}{\partial \xi} - V_x^s\frac{\partial \eta_x/J}{\partial \eta} + V_y^s\frac{\partial \eta_y/J}{\partial \eta} \quad \text{.....(2.29)}$$

Let us just consider the terms on the RHS:

$$Q^s\frac{\partial(1/J)}{\partial \tau} + Q^s\frac{\partial \xi_t/J}{\partial \xi} + E^s\frac{\partial \xi_x/J}{\partial \xi} + F^s\frac{\partial \xi_y/J}{\partial \xi} + Q^s\frac{\partial \eta_t/J}{\partial \eta} + E^s\frac{\partial \eta_x/J}{\partial \eta} + F^s\frac{\partial \eta_y/J}{\partial \eta} - V_x^s\frac{\partial \xi_x/J}{\partial \xi} - V_y^s\frac{\partial \xi_y/J}{\partial \xi} - V_x^s\frac{\partial \eta_x/J}{\partial \eta} - V_y^s\frac{\partial \eta_y/J}{\partial \eta} = Q^s\left(\frac{\partial(1/J)}{\partial \tau} + \frac{\partial \xi_t/J}{\partial \xi} + \frac{\partial \eta_t/J}{\partial \eta}\right) + E^s\left(\frac{\partial \xi_x/J}{\partial \xi} + \frac{\partial \eta_x/J}{\partial \eta}\right) + F^s\left(\frac{\partial \xi_y/J}{\partial \xi} + \frac{\partial \eta_y/J}{\partial \eta}\right) - V_x^s\left(\frac{\partial \xi_x/J}{\partial \xi} + \frac{\partial \eta_x/J}{\partial \eta}\right) - V_y^s\left(\frac{\partial \xi_y/J}{\partial \xi} + \frac{\partial \eta_y/J}{\partial \eta}\right) \quad \text{.....(2.30)}$$

Now, look at each of the parenthetical terms:

$$\left(\frac{\partial(1/J)}{\partial \tau} + \frac{\partial \xi_t/J}{\partial \xi} + \frac{\partial \eta_t/J}{\partial \eta}\right) = \Big(x_{\xi\tau}y_\eta + x_\xi y_{\eta\tau} - x_{\eta\tau}y_\xi - x_\eta y_{\xi\tau} + y_{\tau\xi}x_\eta + y_\tau x_{\eta\xi} - x_{\tau\xi}y_\eta - x_\tau y_{\eta\xi} + x_{\tau\eta}y_\xi + x_\tau y_{\xi\eta} - y_{\tau\eta}x_\xi - y_\tau x_{\xi\eta}\Big) = 0 \quad \text{.....(2.31)}$$

Likewise:

$$\left(\frac{\partial \xi_x/J}{\partial \xi} + \frac{\partial \eta_x/J}{\partial \eta}\right) = y_{\eta\xi} - y_{\xi\eta} = 0 \quad \text{.....(2.32)}$$

$$\left(\frac{\partial \xi_y/J}{\partial \xi} + \frac{\partial \eta_y/J}{\partial \eta}\right) = -x_{\eta\xi} + x_{\xi\eta} = 0 \quad \text{.....(2.33)}$$

Thus, the entire RHS in Eqn. (2.30) is zero, leading to the following governing equation set in transformed coordinates:

$$\frac{\partial Q^s/J}{\partial \tau} + \frac{\partial}{\partial \xi}\left(Q^s\xi_t + E^s\xi_x + F^s\xi_y\right)/J + \frac{\partial}{\partial \eta}\left(Q^s\eta_t + E^s\eta_x + F^s\eta_y\right)/J - \frac{\partial}{\partial \xi}\left(V_x^s\xi_x + V_y^s\xi_y\right)/J - \frac{\partial}{\partial \eta}\left(V_x^s\eta_x + V_y^s\eta_y\right)/J - H/J = 0 \quad \text{.....(2.34)}$$

or:

$$\frac{\partial \hat{Q}^s}{\partial \tau} + \frac{\partial \hat{E}^s}{\partial \xi} + \frac{\partial \hat{F}^s}{\partial \eta} - \frac{\partial \hat{V}_\xi^s}{\partial \xi} - \frac{\partial \hat{V}_\eta^s}{\partial \eta} - \hat{H} = 0 \quad \text{.....(2.35)}$$

Let us next examine the flux vectors $\hat{E}^s$ and $\hat{F}^s$:

$$\hat{E}^s = \left(Q^s\xi_t + E^s\xi_x + F^s\xi_y\right)/J = Q^s(y_\tau x_\eta - x_\tau y_\eta) + E^s y_\eta - F^s x_\eta \text{.....(2.36)}$$

Noting that $x_\tau|_{\xi,\eta} = u^s,\ y_\tau|_{\xi,\eta} = v^s$, we have

$$\hat{E}^s = Q^s(v^s x_\eta - u^s y_\eta) + E^s y_\eta - F^s x_\eta \text{.....(2.37)}$$

Further, noting that $E^s = u^s Q^s$ and $F^s = v^s Q^s$, we get: $\hat{E}^s = 0$. Likewise, we can surmise that $\hat{F}^s = 0$ as well.

Thus, Eqn. (2.35) becomes:

$$\frac{\partial \hat{Q}^s}{\partial \tau} - \frac{\partial \hat{V}_\xi^s}{\partial \xi} - \frac{\partial \hat{V}_\eta^s}{\partial \eta} - \hat{H} = 0 \quad \text{.....(2.38)}$$

The above equation system contains the structural momentum equations and the closure equations for the structural (or mesh) velocities. Note that the latter equations could have been transformed separately,

but we will simply take Eqn. (2.38) and substitute in Eqn. (2.1) and see what the final form of these equations look like.

$$\frac{\partial d_j^s/J}{\partial \tau} = (u_j^s + d_j^s \nabla \cdot \vec{V}^s)/J \quad \text{.....(2.39)}$$

This can be simplified further (learning from our 1D experience):

$$\frac{\partial d_j^s/J}{\partial \tau} = \frac{u_j^s}{J} + \frac{d_j^s}{J}\left(\xi_x \frac{\partial u^s}{\partial \xi} + \eta_x \frac{\partial u^s}{\partial \eta}\right) + \frac{d_j^s}{J}\left(\xi_y \frac{\partial v^s}{\partial \xi} + \eta_y \frac{\partial v^s}{\partial \eta}\right) \quad \text{.....(2.40)}$$

Multiplying by J:

$$\frac{\partial d_j^s}{\partial \tau} - \frac{d_j^s}{J}\frac{\partial J}{\partial \tau} = u_j^s + d_j^s\left(\xi_x \frac{\partial u^s}{\partial \xi} + \eta_x \frac{\partial u^s}{\partial \eta}\right) + d_j^s\left(\xi_y \frac{\partial v^s}{\partial \xi} + \eta_y \frac{\partial v^s}{\partial \eta}\right) \quad \text{.....(2.41)}$$

Now, $u^s = x_\tau$ and $v^s = y_\tau$ and $(1/J)dJ = -Jd(1/J)$

$$\frac{\partial d_j^s}{\partial \tau} = u_j^s + \frac{d_j^s}{J}\frac{\partial J}{\partial \tau} + d_j^s\left(\xi_x x_{\tau\xi} + \eta_x x_{\tau\eta}\right) + d_j^s\left(\xi_y y_{\tau\xi} + \eta_y y_{\tau\eta}\right)$$

$$= u_j^s - d_j^s J(x_\xi y_{\eta\tau} + x_{\xi\tau} y_\eta - x_\eta y_{\xi\tau} - x_{\eta\tau} y_\xi) + d_j^s\left(\xi_x x_{\tau\xi} + \eta_x x_{\tau\eta}\right) + d_j^s\left(\xi_y y_{\tau\xi} + \eta_y y_{\tau\eta}\right) \quad \text{.....(2.42)}$$

Substituting for $\xi_x,\ \xi_y,\ \eta_x,\ \eta_y$ in terms of $x_\eta,\ y_\eta,\ x_\xi,\ y_\xi$, we get:

$$\frac{\partial d_j^s}{\partial \tau} = u_j^s - d_j^s J(x_\xi y_{\eta\tau} + x_{\xi\tau} y_\eta - x_\eta y_{\xi\tau} - x_{\eta\tau} y_\xi) + J d_j^s\left(y_\eta x_{\tau\xi} - y_\xi x_{\tau\eta} - x_\eta y_{\tau\xi} + x_\xi y_{\tau\eta}\right) \quad \text{.....(2.43)}$$

or:

$$\frac{\partial d_j^s}{\partial \tau} = u_j^s \quad \text{.....(2.44)}$$

This means that the definition of the structural velocity follows simply from the definition of the temporal derivative of the displacement.

Equation (2.38) then becomes:

$$\frac{\partial \hat{Q}^s}{\partial \tau} - \frac{\partial \hat{V}_\xi^s}{\partial \xi} - \frac{\partial \hat{V}_\eta^s}{\partial \eta} - \hat{H} = 0 \quad \text{.....(2.45)}$$

where:

$$\hat{Q} = \frac{1}{J}\begin{pmatrix} \rho^s u_j^s \\ d_j^s J \end{pmatrix} \qquad \hat{H} = \frac{1}{J}\begin{pmatrix} -cu_j^s + f_j^s + b_j^s \\ u_j^s J \end{pmatrix} \quad \text{.....(2.46)}$$

and:

$$\hat{V}_\xi^s = \left(V_x^s \xi_x + V_y^s \xi_y\right)/J$$

$$\hat{V}_\eta^s = \left(V_x^s \eta_x + V_y^s \eta_y\right)/J \quad \text{.....(2.47)}$$

where V_x^s and V_y^s are as defined earlier.

2.2 Integral Form of the Structural Equations

We can equivalently express the structural system in integral form. Once again, we show only the physical time-derivative. We start by expressing the integral form of the equations for a moving control volume in Arbitrary Eulerian Lagrangian (ALE) form:

$$\int_V \frac{\partial Q^s}{\partial t} dV + \int_S E_i^s dS = \int_S V_i^s dS + \int_V H^s dV \quad \text{.....(2.48)}$$

Using the differential identity:

$$\frac{\partial}{\partial t}\int_V Q^s dV = \int_V \frac{\partial Q^s}{\partial t} dV + \int_S Q^s u_i^s dS \quad \text{.....(2.49)}$$

One can interpret the above identity in physical terms, i.e., the rate of change of the integrated value of Q within the control volume is the sum of the integral of the rate of change of Q within the CV and the net rate of flux of Q into the control volume.

Substituting:

$$\frac{\partial}{\partial t}\int_V Q^s dV + \int_S (E_i^s - Q^s u_i^s) dS = \int_S V_i^s dS + \int_V H^s dV \quad \text{.....(2.50)}$$

This is equivalent to the classic ALE formulation.

We can then write:

$$\frac{\partial}{\partial t}\int_V Q^s dV + \int_S E_i^{s\prime} dS_i = \int_S V_i^s dS_i + \int_V H^s dV \quad \text{.....(2.51)}$$

where:

$$Q_s = \begin{pmatrix} \rho^s u_j^s \\ d_j^s/V \end{pmatrix} \qquad E'_{s,i} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad V_{s,i} = \begin{pmatrix} \tau_{ij}^s \\ 0 \end{pmatrix} \qquad H_s = \begin{pmatrix} f_j^s + b_j^s + cu_j^s \\ u_s^j/V \end{pmatrix}$$

.....(2.52)

Here, u_j^s is the structural velocity and d_j^s is the structural displacement. The first equation is self explanatory. The second equation is written to match the correct definition of the structural velocity in terms of the displacement. To cast this in a proper control volume formulation, we have expressed the terms on a per unit volume basis. Note also that this form is consistent with the generalized coordinate version of the differential form given in Eqn. (2.46).

As before, the structural stresses are given by Hooke's Law for an isotropic homogeneous material:

$$\tau_{ij}^s = \frac{E^s}{2(1+\nu^s)}\left(\frac{\partial d_i^s}{\partial x_j} + \frac{\partial d_j^s}{\partial x_i}\right) + \delta_{ij}\frac{E^s\nu^s}{(1+\nu^s)(1-2\nu^s)}\nabla\cdot\vec{d}^s$$

.....(2.53)

And, E is the modulus of elasticity and ν is Poisson's ratio.

2.3 Numerical Characteristics of the 1D Structural System

The numerical character of the fluid dynamics equations are well-known: for instance, the inviscid equations represent a hyperbolic system governed by the particle and acoustic speeds that act as agents for the propagation of error through the system. In comparison, the characteristics of the structural equations are less well understood. In this section, we perform some fundamental analytical analysis to determine the structural characteristics. Simple stability analysis is also performed to understand the numerical implications as well.

In 1D, the differential form of the structural dynamics equations can be written as the following coupled set:

$$\frac{\partial \rho_s u_s}{\partial t} = \frac{\partial}{\partial x}E'\frac{\partial d_s}{\partial x} - cu_s + \mathcal{F}(t) + b$$

.....(2.54)

$$\frac{\partial d_s}{\partial t} = u_s$$

.....(2.55)

where c is the damping constant, $\mathcal{F}$ is the external forcing function and b is a body force term.

$$E' = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$$

.....(2.56)

and E is the Young's modulus and ν is Poisson's ratio.

We can equivalently express the above equations in a more compact and general vector form:

$$\Gamma \frac{\partial Q_p}{\partial t} = \frac{\partial}{\partial x} R \frac{\partial Q_p}{\partial x} + H \quad \text{.....(2.57)}$$

where

$$\Gamma = \left(\begin{array}{c|c} \rho_s & 0 \\ 0 & 1 \end{array} \right) \qquad Q_p = \begin{pmatrix} u_s \\ d_s \end{pmatrix} \qquad R = \left(\begin{array}{c|c} 0 & E' \\ 0 & 0 \end{array} \right) \qquad H = \begin{pmatrix} -cu_s + \mathcal{F}(t) + b \\ u_s \end{pmatrix}$$

.....(2.58)

Note that the above form of the equations is similar to the fluid dynamic conservation laws, except that there is no explicit convective flux. This does not necessarily mean that the numerical properties of these equations are identical to the CFD case. Indeed, in the following, we perform some fundamental analytical studies to determine the specific physical and numerical behavior of the equations.

Physical Characteristics: Dispersion Analysis

The physical behavior of the above equation system can be determined by carrying out a dispersion or wave analysis. To do this, we explicitly consider the linear constant coefficient form of the equations:

$$\bar{\Gamma} \frac{\partial Q_p}{\partial t} = \bar{R} \frac{\partial^2 Q_p}{\partial x^2} + \bar{D} Q_p \quad \text{.....(2.59)}$$

where

$$D = \begin{pmatrix} -c & 0 \\ 1 & 0 \end{pmatrix} \quad \text{.....(2.60)}$$

We look for solutions of the form:

$$Q_p = \hat{Q}_p e^{i\omega t} e^{-ikx} \quad \text{.....(2.61)}$$

Substituting the solution into the linearized equations, we get:

$$\left[\Gamma(i\omega) - R(-ik)^2 - D \right] \hat{Q}_p e^{i\omega t} e^{-ikx} = 0 \quad \text{.....(2.62)}$$

For the above linear system to have a non-trivial solution, the LHS matrix operator must have zero determinant:

$$det\left[\Gamma(i\omega) + R(k)^2 - D \right] = 0 \quad \text{.....(2.63)}$$

or:

$$\left| \begin{array}{cc} \rho_s(i\omega) + c & E'k^2 \\ -1 & i\omega \end{array} \right| = 0 \quad \text{.....(2.64)}$$

or:

$$-\rho_s \omega^2 + ic\omega + E' k^2 = 0 \quad \text{.....(2.65)}$$

or

$$\frac{\omega}{k}^2 - \frac{ic}{\rho_s}\frac{\omega}{k} - \frac{E'}{\rho_s} = 0 \quad \text{.....(2.66)}$$

or:

$$\frac{\omega}{k} = \frac{1}{2}\left(i\frac{c}{\rho_s} \pm \sqrt{\left(\frac{c}{\rho_s}\right)^2 + 4\frac{E'}{\rho_s}} \right) \quad \text{.....(2.67)}$$

Note that in the absence of the damping term, i.e., when $c = 0$, we have:

$$\frac{\omega}{k} = \pm\sqrt{\frac{E'}{\rho_s}} \quad \text{.....(2.68)}$$

Interestingly, in the absence of damping, the structural equations are purely hyperbolic (real wave speeds). These speeds are typically referred to as "sound-speeds" in the material. For representative values of E and ρ_s, the sound-speed turns out be of the order of 5000 m/s.

In the presence of the damping term, the wave speeds become complex with a positive imaginary component, which corresponds to damping in time (of course). Moreover, it is noteworthy that the damping coefficient modifies the real part of the wave speed as well, i.e., it increases the speed of sound.

As a next step, we determine the stability characteristics of numerical discretization schemes for solving the 1D structural equations. This is necessary because the system of equations is very different form what we are used to in the fluid dynamics world.

Euler-Explicit/Central Difference Scheme:

For the stability analysis, we employ the linearized constant-coefficient equations given in Eqn. (2.59). We start with a central difference, Euler-explicit scheme:

$$\Gamma\frac{Q_i^{n+1} - Q_i^n}{\Delta t} = R\frac{Q_{i-1}^n - 2Q_i^n + Q_{i+1}^n}{\Delta x^2} + DQ_i^n \quad \text{.....(2.69)}$$

Using Fourier Series, we can write:

$$Q_i = \sum \hat{Q}_n(t) e^{ik_n x} \quad \text{.....(2.70)}$$

Now, for a linear problem, each mode will independently satisfy the equations of motion and so we can consider each mode independently, i.e.,

$$Q_i = Q_n(t)e^{ik_n x} \quad \text{.....(2.71)}$$

$$Q_{i+1} = \hat{Q}_n(t)e^{ik_n(x+\Delta x)} = Q_i e^{ik\Delta x} \quad \text{.....(2.72)}$$

$$Q_{i-1} = \hat{Q}_n(t)e^{ik_n(x-\Delta x)} = Q_i e^{-ik\Delta x} \quad \text{.....(2.73)}$$

Substituting into Eqn. (2.69), we get:

$$\Gamma\frac{Q_i^{n+1} - Q_i^n}{\Delta t} = -\frac{2(1-C_x)}{\Delta x^2}RQ_i^n + DQ_i^n \quad \text{.....(2.74)}$$

where $C_x = \cos k_n \Delta x$.

Noting the "error" in the solution will also satisfy the same governing equations, we can equivalently identify the "Q" in the above equation with the error. We can then define an amplification matrix as follows:

$$Q_i^{n+1} = GQ_i^n \quad \text{.....(2.75)}$$

and the magnitude of the eigenvalues of G determine whether the errors in the solution decay or grow. Since the physical equations indicate constant amplitude waves (in the absence of the damping constant) and/or decay (in the presence of the damping term), we will interpret any growth of the errors as arising from numerical instability.

Substituting Eqn. (2.75) into Eqn. (2.74), we get:

$$\Gamma\frac{G-I}{\Delta t} = -\frac{2(1-C_x)}{\Delta x^2}R + D \quad \text{.....(2.76)}$$

or:

$$\frac{\Gamma}{\Delta t}G = \frac{\Gamma}{\Delta t} - \frac{2(1-C_x)}{\Delta x^2}R + D \quad \text{.....(2.77)}$$

which is of the form:

$$K_1\ G = K_2 \quad \text{.....(2.78)}$$

with:

$$K_1 = \frac{\Gamma}{\Delta t} \qquad K_2 = \frac{\Gamma}{\Delta t} - \frac{2(1-C_x)}{\Delta x^2}R + D \quad \text{.....(2.79)}$$

Then:

$$G = K_1^{-1}K_2 = I - \frac{2(1-C_x)\Delta t}{\Delta x^2}\Gamma^{-1}R + \Delta t\Gamma^{-1}D \quad \text{.....(2.80)}$$

Substituting for the terms, we get:

$$G = \begin{bmatrix} 1 - \frac{c\Delta t}{\rho_s} & -\frac{E'}{\rho_s}\frac{2(1-C_x)\Delta t}{\Delta x^2} \\ \Delta t & 1 \end{bmatrix} \quad \text{.....(2.81)}$$

For simplicity, let us look at the case without damping. The eigenvalues of G are:

$$(1-\lambda)^2 + \frac{E'}{\rho_s}\frac{2(1-C_x)\Delta t^2}{\Delta x^2} = 0 \quad \text{.....(2.82)}$$

or

$$\lambda = 1 \pm i\sqrt{\frac{E'}{\rho_s}\frac{2(1-C_x)\Delta t^2}{\Delta x^2}} \quad \text{.....(3.83)}$$

or

$$\lambda = 1 \pm i\frac{a_s \Delta t}{\Delta x}\sqrt{2(1-C_x)} \quad \text{.....(2.84)}$$

where a_s is the speed of sound and $a_s \Delta t/\Delta x = \sigma$, the CFL number.

Then:

$$\lambda = 1 \pm i\sigma\sqrt{2(1-C_x)} \quad \text{.....(2.85)}$$

or:

$$\lambda\lambda^* = 1 + 2\sigma^2(1-C_x) \quad \text{.....(2.86)}$$

which is always greater than unity. This means that the scheme is ***unconditionally unstable*** for all wave modes (i.e., all values of C_x from -1 to 1).

One can further examine the system with the damping term, although this is a futile exercise for the explicit scheme since it is already unconditionally unstable. At any rate, it is typically easier to determine the amplification factors computationally by substituting reasonable values for the solid properties and numerically evaluating the eigenvalues.

Euler-Implicit/Central Difference Scheme:

We next examine the central differenced, Euler-implicit scheme:

$$\Gamma\frac{Q_i^{n+1} - Q_i^n}{\Delta t} = R\frac{Q_{i-1}^{n+1} - 2Q_i^{n+1} + Q_{i+1}^{n+1}}{\Delta x^2} + DQ_i^{n+1} \quad \text{.....(2.87)}$$

The details of the Fourier analysis are the same. The stability equation is again given by:

$$K_1\ G = K_2 \text{.....(2.88)}$$

where:

$$K_1 = \frac{\Gamma}{\Delta t} + \frac{2(1-C_x)}{\Delta x^2}R - D \qquad K_2 = \frac{\Gamma}{\Delta t} \text{.....(2.89)}$$

It is again possible to evaluate the amplification matrix in closed form. For simplicity, it is easier to determine the inverse of the amplification matrix and then require that the eigenvalues of the inverse matrix be greater than unity for stability.

$$G^{-1} = K_2^{-1}K_1 = I + \frac{2(1-C_x)\Delta t}{\Delta x^2}\Gamma^{-1}R - \Delta t\Gamma^{-1}D \text{.....(2.90)}$$

or:

$$G^{-1} = \begin{bmatrix} 1 + \frac{c\Delta t}{\rho_s} & \frac{E'}{\rho_s}\frac{2(1-C_x)\Delta t}{\Delta x^2} \\ -\Delta t & 1 \end{bmatrix} \text{.....(2.91)}$$

Again, dropping the damping term, the eigenvalues of G^{-1} are:

$$\lambda = 1 \pm i\sigma\sqrt{2(1-C_x)} \text{.....(2.92)}$$

or:

$$\lambda\lambda^* = 1 + 2\sigma^2(1-C_x) \text{.....(2.93)}$$

which is always greater than unity. This means that the scheme is ***unconditionally stable*** for all wave modes (i.e., note that for values of C_x from -1 to 1, the factor $(1-C_x) > 0$).

It is also interesting to see if the central differenced implicit scheme provides any damping at high wavenumbers. This is an indicator of the presence of inherent artificial dissipation in the scheme. To do this, we look at the π wave mode or when $C_x = -1$:

$$\lambda\lambda^*(\pi) = 1 + 4\sigma^2 \text{.....(2.94)}$$

which indicates strong damping of the high wave numbers. Thus, the central differenced scheme possesses inherent damping properties and no additional damping is required for computational purposes (although they may be needed for physical reasons).

2.4 Numerical Characteristics of the 2D Structural System

The 2D structural equations in differential form are:

$$\frac{\partial \rho^s u^s}{\partial t} = \frac{\partial}{\partial x}\left[\left(E' + \frac{E'\nu}{1-2\nu}\right)\frac{\partial d_x^s}{\partial x} + \frac{E'\nu}{1-2\nu}\frac{\partial d_y^s}{\partial y}\right] + \frac{\partial}{\partial y}\left[\frac{E'}{2}\frac{\partial d_x^s}{\partial y} + \frac{E'}{2}\frac{\partial d_y^s}{\partial x}\right] \quad \text{.....(2.95)}$$

$$\frac{\partial \rho^s v^s}{\partial t} = \frac{\partial}{\partial x}\left[\frac{E'}{2}\frac{\partial d_x^s}{\partial y} + \frac{E'}{2}\frac{\partial d_y^s}{\partial x}\right] + \frac{\partial}{\partial y}\left[\frac{E'\nu}{1-2\nu}\frac{\partial d_x^s}{\partial x} + \left(E' + \frac{E'\nu}{1-2\nu}\right)\frac{\partial d_y^s}{\partial y}\right] \quad \text{.....(2.96)}$$

$$\frac{\partial d_x^s}{\partial t} = u^s \quad \text{.....(2.97)}$$

$$\frac{\partial d_y^s}{\partial t} = v^s \quad \text{.....(2.98)}$$

where we have defined $E' = E^s/(1+\nu)$ for convenience.

Stress Form of Equations of Motion

Let us examine the stress form of the structural system in two-dimensions:

$$\frac{\partial \rho^s u^s}{\partial t} - \frac{\partial}{\partial x}\sigma_{xx}^s - \frac{\partial}{\partial y}\sigma_{yx}^s = 0 \quad \text{.....(2.99)}$$

$$\frac{\partial \rho^s v^s}{\partial t} - \frac{\partial}{\partial x}\sigma_{xy}^s - \frac{\partial}{\partial y}\sigma_{yy}^s = 0 \quad \text{.....(2.100)}$$

$$\frac{\partial \sigma_{xx}^s}{\partial t} - \left[\frac{E}{1+\nu} + \frac{E\nu}{(1+\nu)(1-2\nu)}\right]\frac{\partial u^s}{\partial x} - \frac{E\nu}{(1+\nu)(1-2\nu)}\frac{\partial v^s}{\partial y} = 0 \quad \text{.....(2.101)}$$

$$\frac{\partial \sigma_{xy}^s}{\partial t} - \frac{E}{2(1+\nu)}\left[\frac{\partial u^s}{\partial y} + \frac{\partial v^s}{\partial x}\right] = 0 \quad \text{.....(2.102)}$$

$$\frac{\partial \sigma_{yy}^s}{\partial t} - \frac{E\nu}{(1+\nu)(1-2\nu)}\frac{\partial u^s}{\partial x} - \left[\frac{E}{1+\nu} + \frac{E\nu}{(1+\nu)(1-2\nu)}\right]\frac{\partial v^s}{\partial y} = 0 \quad \text{.....(2.103)}$$

We note that, again, the source and sink terms have been dropped for convenience.

We can equivalently write this in the following coupled vector form:

$$\frac{\partial Q_q}{\partial t} + \tilde{A}_q \frac{\partial Q_q}{\partial x} + \tilde{B}_q \frac{\partial Q_q}{\partial y} = 0 \quad \text{.....(2.104)}$$

where:

$$Q_q = \begin{pmatrix} u^s \\ v^s \\ \sigma^s_{xx} \\ \sigma^s_{xy} \\ \sigma^s_{yy} \end{pmatrix} \qquad \tilde{A}_q = \begin{pmatrix} 0 & 0 & -\frac{1}{\rho_s} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\rho_s} & 0 \\ -E'\frac{1-\nu}{1-2\nu} & 0 & 0 & 0 & 0 \\ 0 & -\frac{E'}{2} & 0 & 0 & 0 \\ -E'\frac{\nu}{1-2\nu} & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \tilde{B}_q = \begin{pmatrix} 0 & 0 & 0 & -\frac{1}{\rho_s} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\rho_s} \\ 0 & -E'\frac{\nu}{1-2\nu} & 0 & 0 & 0 \\ -\frac{E'}{2} & 0 & 0 & 0 & 0 \\ 0 & -E'\frac{1-\nu}{1-2\nu} & 0 & 0 & 0 \end{pmatrix}$$

.....(2.105)

where $E' = E/(1+\nu)$.

We can confirm that the eigenvalues of the above system are:

$$\lambda(\tilde{A}_q) = 0 \ , \ \pm\sqrt{\frac{E}{2(1+\nu)\rho^s}}, \ \pm\sqrt{\frac{E(1-\nu)}{(1+\nu)(1-2\nu)\rho^s}} \quad \text{.....(2.106)}$$

or, alternately,

$$\lambda(\tilde{A}_q) = 0 \ , \ \pm\sqrt{\frac{E'}{2\rho^s}}, \ \pm\sqrt{\frac{E'(1-\nu)}{(1-2\nu)\rho^s}} \quad \text{.....(2.107)}$$

These represent the two kinds of "seismic" waves: the so-called ***p waves*** and ***s waves***, with the p waves traveling 2 times faster than the n waves.

Now, for the general 2D case, when the cell faces are not aligned with the coordinate directions, we need to determine the eigenvalues of the general Jacobian matrix of the following form:

$$\hat{A}_q = \tilde{A}k_1 + \tilde{B}k_2 = \begin{pmatrix} 0 & 0 & -\frac{k_1}{\rho_s} & -\frac{k_2}{\rho_s} & 0 \\ 0 & 0 & 0 & -\frac{k_1}{\rho_s} & -\frac{k_2}{\rho_s} \\ -E'k_1\frac{1-\nu}{1-2\nu} & -E'k_2\frac{\nu}{1-2\nu} & 0 & 0 & 0 \\ -\frac{E'k_2}{2} & -\frac{E'k_1}{2} & 0 & 0 & 0 \\ -E'k_1\frac{\nu}{1-2\nu} & -E'k_2\frac{1-\nu}{1-2\nu} & 0 & 0 & 0 \end{pmatrix} \quad \text{.....(2.108)}$$

The eigenvalues of the above matrix can be determined to be:

$$\lambda(\hat{A}_q) = 0 \ , \ \pm\sqrt{\frac{E'}{2\rho^s}(k_1^2+k_2^2)}, \ \pm\sqrt{\frac{E'(1-\nu)}{(1-2\nu)\rho^s}(k_1^2+k_2^2)} \quad \text{.....(2.109)}$$

It is interesting to note that structural waves are like acoustic waves, they travel omni-directionally (unlike particle waves, which travel in the direction of particle motion).

2.5 Artificial Diffusion Schemes

To gain further insight into the structural system, we examine the stress form of the equations in 1D:

$$\frac{\partial \rho_s u_s}{\partial t} - \frac{\partial}{\partial x}\sigma_s = 0 \quad \text{.....(2.110)}$$

$$\frac{1}{E}\frac{\partial \sigma_s}{\partial t} - \frac{\partial u_s}{\partial x} = 0 \quad \text{.....(2.111)}$$

where σ_s is the solid stress and is related to the strain through the Young's modulus: $\sigma_s = E\epsilon_s$, and:

$$\epsilon_s = \frac{\partial d_s}{\partial x} \quad \text{.....(2.112)}$$

We note that Eqns. (2.110) and (2.111) follow directly from Eqns. (2.54) and (2.55), except that the source and sink terms have been dropped for convenience.

We can equivalently write this in the following coupled vector form:

$$\frac{\partial Q_q}{\partial t} + \tilde{A}_q \frac{\partial Q_q}{\partial x} = 0 \quad \text{.....(2.113)}$$

where:

$$Q_q = \begin{pmatrix} u_s \\ \sigma_s \end{pmatrix} \qquad \tilde{A}_q = \begin{pmatrix} 0 & -\frac{1}{\rho_s} \\ -E & 0 \end{pmatrix} \quad \text{.....(2.114)}$$

We can readily confirm that the eigenvalues of the above system are:

$$\lambda(\tilde{A}_q) = \pm\sqrt{\frac{E}{\rho_s}} \quad \text{.....(2.115)}$$

Of course, this indicates that the above first-order system is indeed hyperbolic.

The matrix of left eigenvectors:

$$M^{-1} = \begin{bmatrix} \frac{1}{\rho_s} & \frac{1}{\rho_s} \\ -\sqrt{\frac{E}{\rho_s}} & \sqrt{\frac{E}{\rho_s}} \end{bmatrix} \quad \text{.....(2.116)}$$

and its inverse:

$$M = \begin{bmatrix} \frac{\rho_s}{2} & \frac{1}{2\sqrt{E/\rho_s}} \\ \frac{\rho_s}{2} & \frac{1}{2\sqrt{E/\rho_s}} \end{bmatrix} \quad \text{.....(2.117)}$$

Central-differencing of Eqn. (2.113) will contain no damping terms since all the truncation errors related to a first-order spatial derivative are purely dispersive. Thus, it is necessary to add appropriate artificial dissipation terms. A conventional approach is to express the numerical discretization using an upwind formulation. The first-order upwind scheme can be written as:

$$\frac{\partial Q_q}{\partial t} + \tilde{A}_q \frac{\partial Q_q}{\partial x} = \frac{\Delta x}{2} \frac{\partial}{\partial x} |A_q| \frac{\partial Q_q}{\partial x} \quad \text{.....(2.118)}$$

where

$$|A| = M|\Lambda|M^{-1} \quad \text{.....(2.119)}$$

and:

$$|\Lambda| = \begin{bmatrix} \sqrt{\frac{E}{\rho_s}} & 0 \\ 0 & \sqrt{\frac{E}{\rho_s}} \end{bmatrix} \quad \text{.....(2.120)}$$

Since $|\Lambda|$ is of the form $|\lambda^{\pm}|I$, we note that: $|A| = |\Lambda|$.

In other words:

$$\frac{\partial Q_q}{\partial t} + \tilde{A}_q \frac{\partial Q_q}{\partial x} = \frac{\Delta x}{2} \frac{\partial}{\partial x} |\Lambda| \frac{\partial Q_q}{\partial x} \quad \text{.....(2.121)}$$

In scalar form, the equations of motion with the additional artificial dissipation terms:

$$\frac{\partial \rho_s u_s}{\partial t} - \frac{\partial}{\partial x}\sigma_s = \frac{\Delta x}{2} \frac{\partial}{\partial x} \rho_s \sqrt{\frac{E}{\rho_s}} \frac{\partial u_s}{\partial x} \quad \text{.....(2.122)}$$

$$\frac{\partial \sigma_s}{\partial t} - E\frac{\partial u_s}{\partial x} = \frac{\Delta x}{2} \frac{\partial}{\partial x} E \sqrt{\frac{E}{\rho_s}} \frac{\partial \sigma_s}{\partial x} \quad \text{.....(2.123)}$$

Note that it is implied that the spatial derivatives are central-differenced and the combined form yields the first-order upwind form for the coupled hyperbolic system.

1D Formulation with Added Dissipation Terms

Equations (2.122) and (2.123) can now be transformed back to the $Q_p = (u_s,\ d_s)^T$ variable system.

$$\frac{\partial \rho_s u_s}{\partial t} = \frac{\partial}{\partial x} E \frac{\partial d_s}{\partial x} + \frac{\Delta x}{2} \frac{\partial}{\partial x} \rho_s \sqrt{\frac{E}{\rho_s}} \frac{\partial u_s}{\partial x} \quad \text{.....(2.124)}$$

$$\frac{\partial d_s}{\partial t} = u_s + \frac{\Delta x}{2} \frac{\partial}{\partial x} \sqrt{\frac{E}{\rho_s}} \frac{\partial d_s}{\partial x} \quad \text{.....(2.125)}$$

Or,

$$\Gamma_p \frac{\partial Q_p}{\partial t} = \frac{\partial}{\partial x} R \frac{\partial Q_p}{\partial x} + H + \frac{\Delta x}{2} \frac{\partial}{\partial x} \Gamma_p |\Lambda| \frac{\partial Q_p}{\partial x} \quad \text{.....(2.126)}$$

where

$$|\Lambda| = \begin{bmatrix} \sqrt{\frac{E}{\rho_s}} & 0 \\ 0 & \sqrt{\frac{E}{\rho_s}} \end{bmatrix} \quad \text{.....(2.127)}$$

The last term in Eqn. (2.126) is the equivalent artificial dissipation which formally reduces the system to a first-order upwind scheme. The main advantage of such a formulation would lie in the preservation of monotonicity in the presence of strong gradients in the structural solution. We reiterate that the additional terms are not required for smoothing of odd-even decoupling; they are potentially needed only for monotonicity preservation or for maintaining diagonal dominance for linear solution purposes.

2D Artificial Dissipation Formulation

As noted in the one-dimensional development, one may derive artificial dissipation formulations for the above hyperbolic system of equations. When you use the second-order form of the structural equations, the addition of dissipation is not necessary to maintain solution smoothness because the second-order difference operators of the second-derivative terms naturally contain artificial dissipation terms. In contrast, the central-differencing of the first-order operators do not contain artificial dissipation naturally and such terms need to be explicitly or implicitly added. Upwind discretizations are an example of the latter.

In our work, we are employing the second-order form of the equations and, therefore, additional artificial dissipation terms are not required. However, as explained in the one-dimensional case, the governing equations are not diagonally dominant, which stymies the application of certain iterative solution techniques such as Gauss-Seidel. Therefore, we introduce second-order dissipation terms on the LHS operator alone. Such a formulation is considered later. Here, we merely present a suggested form of the artificial dissipation operator to help introduce diagonal dominance and maintain stability.

A simple form of artificial dissipation model comes from the spectral radius-based formulation of Jameson and Pulliam:

$$E'_{s,i+1/2} = \frac{E_i + E_{i+1}}{2} - \frac{\Delta x}{2}|\sigma(\hat{A})|(Q_{i+1} - Q_i) \quad \text{.....(2.128)}$$

where the above expression refers to the interfacial definition of the flux term in Eqn. (2.48) and $\boldsymbol{\sigma}(\hat{A})$ is the spectral radius of the generalized system Jacobian. This is of the following form, but will be updated with the correct maximum eigenvalue:

$$\sigma(\hat{A}) = \sqrt{\frac{2E(1-\nu)}{(1+\nu)(1-2\nu)\rho^s}(k_1^2 + k_2^2)} \quad \text{.....(2.129)}$$

Note that the above formulation is used only on the LHS of the implicit operator in order to provide diagonal dominance for the iterative linear solver. The flux scheme is not used on the RHS and has no influence on the accuracy of the solution itself. It is also worth pointing out that the physical inviscid flux vector is a null vector for the structural equations; however, the dissipation part of the flux-vector is non-zero, which is of course necessary to provide the diagonal dominance. Finally, we note that other more sophisticated formulations of the matrix dissipation term are possible, but are probably unnecessary given the somewhat limited goals of achieving diagonal dominance.

3.0 Unified Solution Methodology

Fundamental development of any coupled system of equations requires that the system be treated within a single unified framework. Such an approach ensures that all potential interactions and coupling between the constituent equations are automatically reflected in the computational procedures. It is important to note that this is true even if the coupled systems are eventually solved in a partitioned manner in the final solution algorithm (for instance, due to parallel domain decomposition). Accordingly, we start by expressing the governing equations in a common vector form and by describing a unified finite-volume discretization of these equations. We pay special attention to the structural dynamics equation because the finite-volume form of these equations is relatively novel and their numerical characteristics are not well-established in the literature. We then discuss the details of the unified algorithm including the fully implicit time-integration, linearization and solution schemes, as well as issues related to the Geometric Conservation Law, preconditioning and fluid-structural interface conditions.

3.1 Equations of Motion

Fundamental development of any coupled system of equations requires that the system be treated within a single unified framework. Such an approach ensures that all potential interactions and coupling between the constituent equations are automatically reflected in the computational procedures, even if the coupled systems are eventually solved in a partitioned manner in the final solution algorithm (for instance, due to parallel domain decomposition).

Accordingly, we express the governing equations in a common vector form and in a unified finite-volume framework:

$$\frac{\partial}{\partial t}\int_V \tilde{Q}dV + \int_S \tilde{E}_i' dS_i = \int_S \tilde{V}_i dS_i + \int_V \tilde{H}dV \quad \text{.....(3.1)}$$

where $\tilde{E}_i' = \tilde{E}_i - u_i^f \tilde{Q}$ represents the inviscid flux, $\tilde{V}_i$ is the viscous flux, $\tilde{H}$ is the source term and u_i^f is the mesh velocity. Note that the "tildes" connote the entire pde system:

$$\tilde{Q} = \begin{pmatrix} \rho \\ \rho u_j \\ e \\ \rho^f u_j^f \\ d_j^f/V \\ \rho^s u_j^s \\ d_j^s/V \end{pmatrix} \qquad \tilde{E}_i' = \begin{pmatrix} \rho(u_i - u_i^f) \\ \rho u_j(u_i - u_i^f) + p\delta_{ij} \\ e(u_i - u_i^f) + pu_i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \tilde{V}_i = \begin{pmatrix} 0 \\ \tau_{ij}' \\ \kappa\partial T/\partial x_i + u_j\tau_{ij} \\ \tau_{ij}^f \\ 0 \\ \tau_{ij}^s \\ 0 \end{pmatrix}$$

$$\tilde{H} = \begin{pmatrix} 0 \\ 0 \\ \Phi \\ f_j^f + b_j^f + cu_j^f \\ u_j^f/V \\ f_j^s + b_j^s + cu_j^s \\ u_j^s/V \end{pmatrix}$$

$$\tau_{ij}' = \mu\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) + \delta_{ij}\lambda\nabla\cdot\vec{u}$$

$$\tau_{ij}^s = \frac{E^s}{2(1+\nu^s)}\Big(\frac{\partial d_i^s}{\partial x_j} + \frac{\partial d_j^s}{\partial x_i}\Big) + \delta_{ij}\frac{E^s\nu}{(1+\nu^s)(1-2\nu^s)}\nabla\cdot\vec{d^s}$$

$$\tau_{ij}^f = \frac{E^f}{2(1+\nu^f)}\Big(\frac{\partial d_i^f}{\partial x_j} + \frac{\partial d_j^f}{\partial x_i}\Big) + \delta_{ij}\frac{E^f\nu}{(1+\nu^f)(1-2\nu^f)}\nabla\cdot\vec{d^f}$$

.....(3.2)

In the above system, the first three equations are the fluid dynamic equations in the ALE form, the next two equations represent the mesh equations represented by a pseudo-structure and the last two equations are the actual solid mechanics equations. Note that the second equation in each structural system is simply a compatibility relation that defines the mesh velocity as the time-derivative of the displacement. Note that these terms are written in a per cell-volume basis to be consistent with the finite-volume integral formulation.

The above system is not actually solved at all points in the computational domain; rather, the fluids and mesh motion equations are solved in the fluids domain and the structural equations are solved in the solid region. For purposes of this demarcation, we equivalently write the above coupled system as:

$$\tilde{Q} = \begin{pmatrix} Q \\ Q_f \\ Q_s \end{pmatrix} \quad \text{.....(3.3)}$$

where $Q = (\rho, \rho u_j, e)^T$ represent the fluids equations, $Q_f = (\rho^f u_j^f, d_j^f)^T$ represents the mesh system and $Q_s = (\rho^s u_j^s, d_j^s)^T$ represents the structural system on the other. Note that we have partitioned the equations differently than we did in the past; particularly, the mesh equations are expressed as

a separate partition of equations; the reason for this difference will be apparent later. We note that the first two systems are active in the fluid "zone", while the third system is active only in the structural "zone". The two physical zones are coupled to each other at the interface and the unified framework considered here would treat this coupling in a naturally conservative and implicit fashion.

We note that it useful sometimes to think of the three sets of equations as individual conservation laws, i.e.

$$\frac{\partial}{\partial t}\int_V Q dV + \int_S E_i' dS_i = \int_S V_i dS_i + \int_V H dV$$

$$\frac{\partial}{\partial t}\int_V Q_f dV + \int_S E_{f,i}' dS_i = \int_S V_{f,i} dS_i + \int_V H_f dV$$

$$\frac{\partial}{\partial t}\int_V Q_s dV + \int_S E_{s,i}' dS_i = \int_S V_{s,i} dS_i + \int_V H_s dV \quad \text{.....(3.4)}$$

We further note that the iterative solution of the above system of equations would potentially converge to exactly the same solution as the tightly-coupled form in Eqn. (3.2) provided the boundary force and displacement information is exchanged during the solution process.

3.2 Time-Derivative Preconditioning

The solution of the equations in (3.4) would converge to exactly the same solution as the coupled form in Eqn. (3.1). For definiteness, we will refer to such an uncoupled formulation as the ***non-linear-level partitioned approach***. Alternately, it is possible to retain the fully coupled system until the linear solver stage and then iteratively solve the constituent linear equation counter-parts. This scheme is developed later and will be referred to as the ***linear-level-partitioned approach***.

A further remark that can be made with respect to the partitioned equation system in Eqn. (3.4) is that the three sets of equations can be solved iteratively in sequential fashion. In earlier formulations, we partitioned the fluid-structural system into a fluid-mesh system and a structural system. However, we note that the fluid dynamics equations depend upon the mesh equations through the mesh velocity terms in the inviscid fluxes, but the mesh equations do not explicitly depend upon the fluid equations. This fact suggests that it would be more economical to consider the three-way partitioning given in Eqn. (3.4) or its linear counterpart that will be discussed later. Specifically, one may consider the following sequence of solutions at each physical time-step: first, the structural solution (Q_s), using the latest fluid dynamic forces. This is then followed by the mesh solution using the structural displacement (Q_f) as the boundary condition. Finally, we can perform the fluid-dynamic solution (Q) using the mesh velocity (and displacement) from the structural solution. Then, the cycle is repeated until convergence is obtained for that particular time-step. In fact, the actual solution scheme will follow this solution pattern.

Time-Scaling of the Equations

Before we proceed further, we will look further at the physical scaling of the above equations. One of the issues that arises with the equation set in (3.1) or, equivalently, in (3.4) is that the same physical time-scale is being used in all of the equations. This is, of course, exactly what we want for the physical equa-

tions, but the fluid-mesh equations are not physical and to limit the mesh motion to the physical time-scales of motion is an unnecessary constraint. Specifically, it may be preferable to allow the fluid-mesh to relax completely to a pseudo-steady state at every time-step in response to the actual motion of the structure. This would potentially insure better behaved fluid grids during the course of the computations.

It is easy to manipulate the equations to introduce a time-scaling of the mesh equations. For instance:

$$\frac{\partial}{\partial t}\int_V Q dV + \int_S E_i' dS_i = \int_S V_i dS_i + \int_V H dV$$

$$\frac{1}{\Upsilon_f}\frac{\partial}{\partial t}\int_V Q_f dV + \int_S E_{f,i}' dS_i = \int_S V_{f,i} dS_i + \int_V H_f dV$$

$$\frac{\partial}{\partial t}\int_V Q_s dV + \int_S E_{s,i}' dS_i = \int_S V_{s,i} dS_i + \int_V H_s dV \quad \text{.....(3.5)}$$

Note that the time-derivative of the second equation is scaled by Υ_f. When this is a large number, the equivalent time-step used for the fluid-mesh equation is much greater than the physical fluids and structural equations. In other words, by appropriately defining this parameter, one can essentially use an infinite time-step for the pseudo-structural mesh equations, or equivalently, solve these equations to their "steady-state". We should also note that this is the same as dropping the physical time-derivative in the mesh equations altogether.

The coupled system in Eqn. (3.1) can also be written to express this "physical" time-scaling in the following manner:

$$\tilde{\Gamma}\frac{\partial}{\partial t}\int_V \tilde{Q} dV + \int_S \tilde{E}_i' dS_i = \int_S \tilde{V}_i dS_i + \int_V \tilde{H} dV \quad \text{....(3.6)}$$

where the time-scaling matrix Gamma has the following form:

$$\tilde{\Gamma} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\Upsilon^f} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\Upsilon^f} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Once again the term Υ_f has the same definition as given earlier and its inclusion allows the pseudo-structural equations to advance to a steady-state for each physical time-step of the fluids and structural equations.

So, Eqn. (3.5) or equivalently Eqn. (3.6) will be the system of equations we will employ in our analysis. In the next section, we consider time-accurate or unsteady flows and formulate the sub-iterative solution process using a dual-time scheme. Following this, we discuss the interface treatment for the above sets

of equations and the discretization procedure to ensure that discrete-GCL is preserved. In the final section, we develop the linear-level partition method mentioned earlier.

Dual-Time Iterative Formulation

The above system of equations has many potential sources of stiffness: between the particle speeds and acoustic speeds, viscous time scales, inherent unsteady fluid time scales, imposed external frequencies (such as from the rotor rotation) and the structural time scales. In order to counter these stiffnesses and to construct a well-conditioned computational procedure, it is necessary to employ a preconditioned pseudo-time-stepping scheme. The preconditioning scaling introduces artificial time-scales which preserves proper conditioning of the system characteristics under a wide range of physical conditions and scales, thereby providing a means of controlling both accuracy and efficiency of the computational scheme.

To emphasize that the preconditioning scaling is introduced only in the iterative level, we express the fully coupled equations in the so-called dual-time version:

$$\tilde{\Gamma}_p \frac{\partial}{\partial \tau} \int_V \tilde{Q}_p dV + \tilde{\Gamma} \frac{\partial}{\partial t} \int_V \tilde{Q} dV + \int_S \tilde{E}'_i dS_i = \int_S \tilde{V}_i dS_i + \int_V \tilde{H} dV \quad \text{.....(3.7)}$$

where the preconditioning matrix, Γ_p, is defined with respect to a vector of primitive variables, Q_p, for simplicity:

$$\tilde{\Gamma}_p = \begin{pmatrix} \rho'_p & 0 & \rho_T & 0 & 0 & 0 & 0 \\ u_j \rho'_p & \rho \delta_{ij} & u_j \rho_T & 0 & 0 & 0 & 0 \\ h_0 \rho'_p - (1 - \rho h_p) & \rho u_i & h_o \rho_T + \rho h_T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\rho^f}{\varepsilon^f} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\varepsilon^f} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\rho^s}{\varepsilon^s} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\varepsilon^s} \end{pmatrix} \qquad \tilde{Q}_p = \begin{pmatrix} p \\ u_j \\ T \\ u_j^f \\ d_j^f \\ u_j^s \\ d_j^s \end{pmatrix} \quad \text{.....(3.8)}$$

where $\rho_p,\ \rho_T,\ h_p,\ h_T$ represent partial derivatives of the thermodynamic properties and

Here, ρ'_p represents the traditional pseudo-time scaling of the fluids equations for low Mach number and different physical time scales:

$$\rho'_p = \frac{1}{V_p^2} - \frac{\rho_T (1 - \rho h_p)}{\rho h_T} = \frac{1}{\min\left[\max\left(V^2, V^2 Str^2, V^2/Re^2\right), c^2\right]} - \frac{\rho_T (1 - \rho h_p)}{\rho h_T} \quad \text{.....}$$

(3.9)

where $V^2 = u_k u_k$. In other words, the scaling is controlled by the local Mach number, Reynolds number and Strouhal number (for unsteady problems).

The other artificial scalings ε^s and ε^f represent appropriate pseudo-time scalings for the structural and mesh motion equations. The definitions of these terms depend on the controlling physics of these equations and are given as:

The above derivations clarify the correct choice of the fluid-structure pseudo-time scalings in the choice of the preconditioning matrix. Given that the fluid dynamics eigenvalues are given by λ_f, the two structural scalings can be written as:

$$\varepsilon_f = \frac{\lambda_f}{\sqrt{E'_f/\rho_f}} \qquad \varepsilon_s = \frac{\lambda_f}{\sqrt{E'_s/\rho_s}} \quad \text{.....(3.9)}$$

The pseudo-time step is selected according to the fluid eigenvalue. Thus:

$$\Delta\tau = \frac{CFL\,\Delta x_j}{\lambda_f} \quad \text{.....(3.10)}$$

We note that the above scalings ensure that the structural equations operate at the optimal time-step choice based on the relevant characteristic speeds.

The preconditioned dual-time system can also be written in the partitioned form given in Eqn. 3.5:

$$\Gamma_p \frac{\partial}{\partial\tau}\int_V Q_p dV + \frac{\partial}{\partial t}\int_V Q dV + \int_S E'_i dS_i = \int_S V_i dS_i + \int_V H dV$$

$$\frac{1}{\epsilon_f}\frac{\partial}{\partial\tau}\int_V Q_f dV + \frac{1}{\Upsilon_f}\frac{\partial}{\partial t}\int_V Q_f dV + \int_S E'_{f,i} dS_i = \int_S V_{f,i} dS_i + \int_V H_f dV$$

$$\frac{1}{\epsilon_s}\frac{\partial}{\partial\tau}\int_V Q_s dV + \frac{\partial}{\partial t}\int_V Q_s dV + \int_S E'_{s,i} dS_i = \int_S V_{s,i} dS_i + \int_V H_s dV \quad \text{.....(3.11)}$$

where Γ_p refers to the top 3 x 3 section of the fully coupled matrix in Eqn. (3.8) and the remaining terms have their usual meanings.

We use the coupled and partitioned forms of the equations interchangeably because, while the two forms are identical, each has specific advantages for certain situations. For example in the following section, we will discuss the formulation of the discrete scheme using the partitioned form because the inviscid flux terms and GCL issues are present only in the fluid dynamics equations.

3.3 Fluid-Structure Interface Treatment

We next consider the interface implementation for fluid-structures problems, which is *critical because it is at the interface that all the coupling between the fluids and structural equations occurs.* The overall guide-

lines for the interface formulation include: (1) clear exposition of the boundary equations, (2) water-tight conservation, (3) unified formulation for both node- and cell-centered schemes, (4) ability to do both explicit and implicit interface solution, and, in addition, (5) exact flux-transfer of momentum fluxes (i.e., forces and displacements between the fluid and structure).

For both cell-centered and node-centered formulations, we first locate a set of interface points, which stores the unknown at the interface. The list of unknowns at these interface points:

Fluids: $\rho,\ u_j,\ T,\ u_j^f,\ d_j^f$

Structures: $u_j^s,\ d_j^s$

where the subscripts "s" and "f" refer to the structures and fluid structural mesh equations and I have used primitive variables for simplicity, although any complete set of dependent variables can be used.

The above represents a total of seven unknowns in 1D, 12 unknowns in 2D or 17 unknowns in 3D, which means that we require seven (1D), 12 (2D) or 17 (3D) equations to specify them. To obtain the necessary interface equations or conditions, we consider the conservation laws applied to an infinitesimal control volume around the interface.

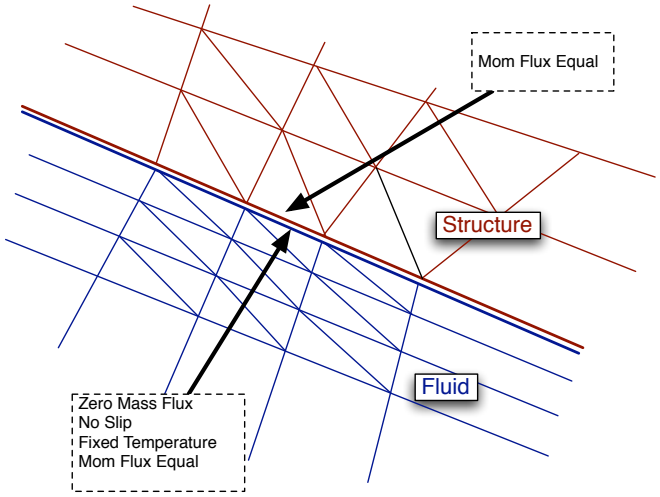

Figure 1: Schematic of fluids and structures meshes showing the details of the fluid-structure interface conditions.

1. Mass Conservation:

$$u_n = u_n^f \text{.....(3.12)}$$

2. Momentum Conservation:

$$p + \tau'_{nn} = \tau^s_{nn}$$

$$\tau'_{nt} = \tau^s_{nt} \text{.....(3.13)}$$

3. Adiabatic wall:

$$\frac{\partial T}{\partial n} = 0 \quad \text{.....(3.14)}$$

4. Compatibility Conditions:

$$d_j^s = d_j^f$$

$$u_j^s = u_j^f$$

$$u_j^f = \frac{\partial d_j^f}{\partial t}$$

$$u_j^s = \frac{\partial d_j^s}{\partial t} \quad \text{.....(3.15)}$$

The above represents seven, 12 or 17 equations, which can in turn be written in compact form as follows:

$$\Omega(Q_{in}) = 0 \quad \text{.....(3.16)}$$

where Ω is a vector of length seven (or 12 or 17) and contains the above conditions and Q_{in} are the interface Q variables. The interface Q variables are used to define the interface fluxes on the fluid and structural sides and are therefore coupled with the conservation laws on each side.

Cell-Centered Formulation

For simplicity, we tackle the cell-centered case first and limit ourselves to the 1D situation for clarity:

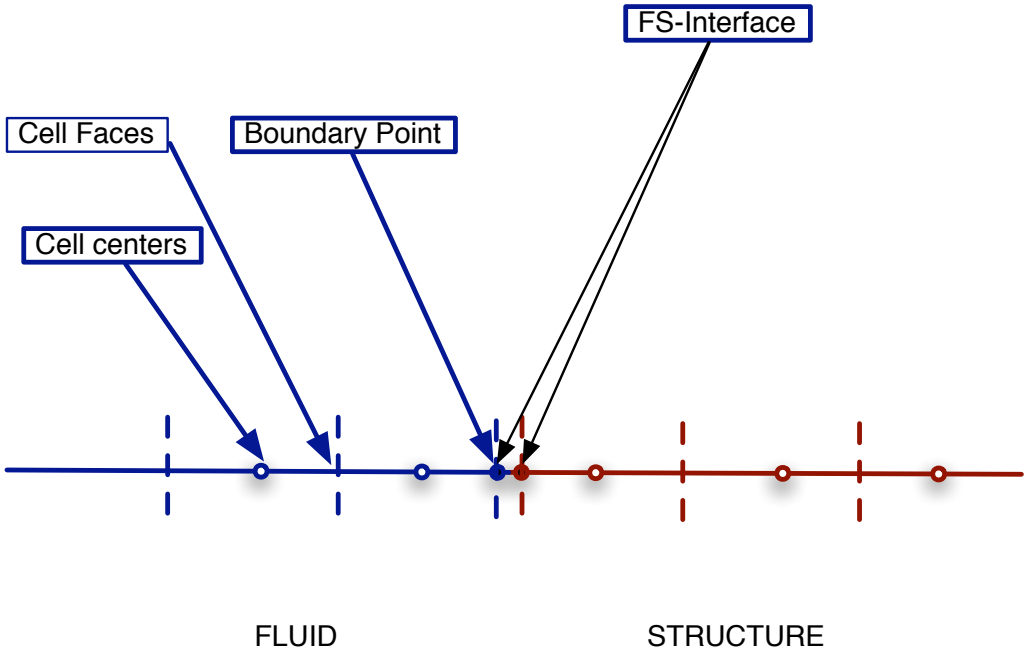

Figure 3.2. Schematic showing cell-centered grid structure and interface locations where the boundary (i.e., interface) variables are stored.

We position a point on the interface on both the fluids-side and the structures-side. *These points are naturally coincident, but need to be defined on both the fluids and structural sides because they are used to store the respective unknowns for the two systems.* Moreover, these terms are defined so as to satisfy the appropriate interface conditions and are subsequently utilized for calculating the boundary (or interface) fluxes for the fluids and structures calculations.

Node-Centered Formulation

We now have two "half-control-volumes": one on the fluid side and one on the structures side. The "cell-centers" of these control volumes lie on the interface. In addition, we have two interface points, one for the fluids-side and one for the structures-side, which are utilized to store the interface values of the dependent variables and to calculate the interface fluxes. Note that, in the node-centered case, the grid locations of the cell-averaged values for the boundary half-cells are coincident with the grid locations for the interface vales. Nevertheless, *separate storage locations are maintained in order to properly enforce the boundary conditions and, at the same time, ensure water-tight flux conservation in the fluid and structural cells.*

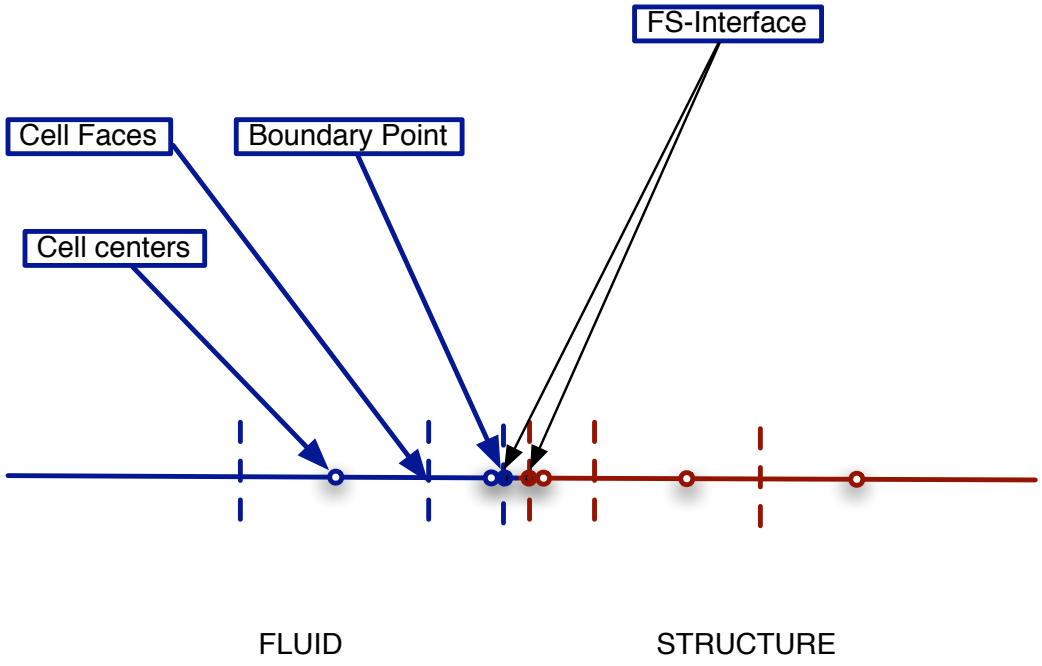

Figure 3.3. Schematic showing node-centered grid structure and interface locations where the boundary (i.e., interface) variables are stored.

3.4 Discrete Formulation and Geometric Conservation Law

The important element of the discretization of Eqn. (3.7) or (3.11) is the inviscid flux formulation. We note that the inviscid terms are non-zero only in the fluid-dynamic equations and, therefore, we consider only this subset for the purposes of the present discussion. Common flux formulations of hyperbolic systems involve the addition of artificial dissipation terms similar to the discussions given previously. The flux formulation can take a number of different forms. A scalar dissipation model would yield:

$$\vec{\mathcal{E}}'_{k+1/2} = \frac{\vec{\mathcal{E}}'_L + \vec{\mathcal{E}}'_R}{2} - \frac{\Delta x}{2}\Gamma_p|\sigma(\Gamma_p^{-1}A'_p)|(Q_{p,R} - Q_{p,L}) \quad \text{.....(3.17)}$$

where A'_p is the Jacobian of the entire inviscid flux E', i.e., $A'_p = \partial E'/\partial Q_p$.

On the other hand, a matrix dissipation model would take the form:

$$\vec{\mathcal{E}}'_{k+1/2} = \frac{\vec{\mathcal{E}}'_L + \vec{\mathcal{E}}'_R}{2} - \frac{\Delta x}{2}\Gamma_p|\Gamma_p^{-1}A'_p|(Q_{p,R} - Q_{p,L}) \quad \text{.....(3.18)}$$

We further note that the definition of the inviscid flux $\mathcal{E}'$ contains the interface grid velocity term, which can be derived from the above interfacial flux expressions. We next consider the GCL aspects of the above formulation(s).

The discrete geometric conservation law states that a constant flow, Q = C, satisfies the discrete equations exactly. Now, for the coupled fluid dynamic and mesh motion equations, it is easy to see that this is readily satisfied because the grid velocity and grid displacements are constant as well. This may be why it is commonly understood that GCL is automatically insured for coupled fluid and fluid-mesh solution schemes.

However, we can go a step further and require that GCL be satisfied for uniform fluid dynamics and non-uniform mesh motion. In other words, we consider the case when $\vec{u}_f \neq C$. Essentially, this introduces a new constraint on how certain cell-geometry related terms must be computed. We further note that we can disregard the mesh motion or structural equations from further consideration since these equations are not impacted by the requirement that the fluid dynamics is constant.

By substituting into the equations of motion for constant flow, we get:

$$Q\frac{\partial}{\partial t}\int_V dV - Q\int_S \vec{u}_f \cdot \hat{n} dS = 0 \quad \text{.....(3.19)}$$

which, if we are using II-order BDF, reduces to:

$$\frac{3}{2}V^{n+1} - 2V^n + \frac{1}{2}V^{n-1} = \Delta t\left(u^{n+1}_{f,k+1/2} - u^{n+1}_{f,k-1/2}\right) \quad \text{.....(3.20)}$$

where the right hand side terms are the grid velocities at the faces of the cell, which are obtained from the standard inviscid flux formulation for the system and, in general, includes the artificial dissipation terms.

GCL: Method I

In general, there are two methods for implementing GCL. The obvious approach for the unified coupled formulation is to use Eqn. (3.20) to update the cell volume at the new time-level. In other words, solution of the coupled equation system at each time-step yields the values of the grid displacements, d_f, at the cell centers at the new time-level. Knowing the grid displacements, the cell-center grid velocities can be determined and the interfacial velocities can in turn be determined from the flux formulation given in Eqn. (3.17) and then used in Eqn. (3.20) to determine the new volume V^{n+1}. This procedure to determine the new cell volume will automatically insure that GCL is satisfied.

It is important to note that the above flux procedure (for a constant flow) results in the following simple averaging formula for the interfacial mesh velocities:

$$u^{n+1}_{f,k+1/2} = \frac{1}{2}\left(u_{f,k} + u_{f,k+1}\right) \quad \text{.....(3.21)}$$

Thus, the update equation becomes:

$$\frac{3}{2}V^{n+1} = 2V^n - \frac{1}{2}V^{n-1} + \frac{\Delta t}{2}\left[\left(u_{f,k} + u_{f,k+1}\right)^{n+1} - \left(u_{f,k} + u_{f,k-1}\right)^{n+1}\right] \quad \text{.....(3.22)}$$

which will clearly satisfy the GCL condition.

For further elucidation, we also note that Eqn. (24) is equivalently:

$$\frac{3}{2}V^{n+1} = 2V^n - \frac{1}{2}V^{n-1} + \frac{\Delta t}{2}\left[\left(\frac{\partial d}{\partial t}\Big|_{f,k} + \frac{\partial d}{\partial t}\Big|_{f,k+1}\right)^{n+1} - \left(\frac{\partial d}{\partial t}\Big|_{f,k} + \frac{\partial d}{\partial t}\Big|_{f,k-1}\right)^{n+1}\right] \quad .....(3.23)$$

or,

$$\frac{3}{2}V^{n+1} = 2V^n - \frac{1}{2}V^{n-1} + \Delta t\left[\frac{\partial d}{\partial t}\Big|_{f,k+1/2} - \frac{\partial d}{\partial t}\Big|_{f,k-1/2}\right] \quad .....(3.24)$$

or,

$$\frac{3}{2}V^{n+1} - 2V^n + \frac{1}{2}V^{n-1} = \left[\left(\frac{3}{2}d^{n+1} - 2d^n + \frac{1}{2}d^{n-1}\right)\Big|_{f,k+1/2} - \left(\frac{3}{2}d^{n+1} - 2d^n + \frac{1}{2}d^{n-1}\right)\Big|_{f,k-1/2}\right] \quad .....(3.25)$$

or, we can write:

$$\Delta V = d_{f,k+1/2} - d_{f,k-1/2} \quad .....(3.26)$$

which simply states that the cell volume change is given by the difference in the displacements of the right- and left-cell boundaries, which is a physically accurate description of the cell volume. Thus, the GCL condition leads to a natural and precise definition of the cell volume.

GCL Method II

An alternate approach to GCL can be equivalently defined by not requiring that the interfacial mesh velocity be specified by the overall flux formulation. We note that this is, in principle, counter to the unified solution strategy for the fluids, mesh and structural equations. However, conventional fluid-structure treatments do not utilize a fully coupled solution procedure and the mesh may simply be updated by algebraic means. It is therefore instructive to see what the traditional GCL constraint leads to.

The procedure starts with the specification of the cell volume. Again, we are concerned mainly with the change in the volume due to displacement of the cell faces. Thus:

$$\Delta V = V - V_0 = d_{f,k+1/2} - d_{f,k-1/2} \quad ....(3.27)$$

where V_0 is the cell volume for the original undeformed fluid mesh.

The interfacial mesh displacements can be simply written as the average across the face:

$$d_{f,k+1/2} = \frac{1}{2}(d_{f,k} + d_{f,k+1}) \quad \text{.....(3.28)}$$

Using the latest available values of the interfacial mesh displacements, the cell volume can be calculated.

The interfacial mesh velocities are then computed using the following relationship:

$$u^{n+1}_{f,k+1/2} = \frac{\partial d_f}{\partial t}\Big|_{k+1/2} = \frac{1}{\Delta t}\left(\frac{3}{2}d^{n+1} - 2d^n + \frac{1}{2}d^{n-1}\right)\Big|_{k+1/2} \quad \text{.....(3.29)}$$

It is straightforward to show that the cell volume from Eqn. (3.27) and the mesh velocity obtained from Eqn. (3.29) together ensure that GCL is satisfied (Eqn. (3.20)). In fact, the substitution of Eqn. (3.27) to the LHS of Eqn. (3.21) and the substitution of Eqn. (3.29) to the RHS of Eqn. (3.21) lead to cancellation of all the terms, thereby proving that GCL is preserved identically.

In conclusion, we note that the two methods are closely related, but distinct. In the first, the mesh velocity solution at the nodes is averaged at the faces and the interfacial mesh displacements are calculated to be consistent with the interfacial mesh velocities. In the second, it is the mesh displacement at the nodes that are averaged at the cell faces and the interfacial displacements are used to define the interfacial mesh velocities. In both cases, discrete GCL is insured because the change in cell volume is calculated according the volumes swept by each cell face.

3.5 Coupled Solution Procedure

Implicit coupled solutions are necessary to provide unconditional stability of the computational procedure. However, fully implicit solutions are difficult to obtain at all levels of the solution procedure for a variety of reasons, which will be discussed later. In order to carefully analyze implicit solutions and to formulate a methodical approach for solving large systems of coupled equations, we clearly delineate the numerical solution procedure into three stages: physical time-stepping, non-linear iterations and linear solver iterations. In each case, we introduce distinct time-stepping procedures to facilitate the iterative process as well as to provide a natural means of under-relaxation for ensuring robustness.

The simplest scheme is the so-called non-linear partitioned scheme which is written directly by invoking Eqn. (3.11) and solving the equations in sequential fashion. Because of the dependencies, it is best to carry out the structural solution first, followed by the mesh solution and, finally, the fluid-dynamic solution. The entire sequence is repeated within the non-linear sub-iteration process until the equations are converged for the particular physical time-step. Once converged, the equations are advanced to the next physical time-step and the iterative procedure is repeated.

The linear-partitioned scheme, on the other hand, starts with the coupled vector system in Eqn. (3.7) in discrete form:

$$\tilde{\Gamma}_p \frac{\partial}{\partial \tau}\int_V \tilde{Q}_p dV + \tilde{\Gamma}\frac{\partial}{\partial t}\int_V \tilde{Q} dV + \int_S \tilde{E}'_i dS_i = \int_S \tilde{V}_i dS_i + \int_V \tilde{H} dV$$

$$\tilde{\Gamma}_p \frac{\partial \tilde{Q}_p V}{\partial \tau} = -\tilde{\Gamma}\Big[\frac{3\tilde{Q}^{n+1} - 4\tilde{Q}^n + \tilde{Q}^{n-1}}{\Delta t} V^{n+1} + \sum_{faces} (\tilde{E}_i' S_i)^{n+1} - \sum_{faces} (\tilde{V}_i S_i)^{n+1} - (\tilde{H}V)^{n+1}\Big] = -\tilde{R}^{n+1}$$

.....(3.30)

where $\tilde{R}$ represents the residual of the physical unsteady system at each time-step. Note that all the flux terms are evaluated at the new time level, signifying that the physical system is being treated in a fully implicit fashion. Consequently, there are no formal restrictions on the stability of the physical time-marching process, other than the restrictions imposed by the accurate representation of the physical transients.

We next linearize the equations and, recognizing that an iterative solution is necessary to eliminate the linearization errors, we introduce the pseudo-time-stepping scheme to orchestrate the iterations:

$$\Big[\frac{\tilde{\Gamma}_p V}{\Delta \tau_l} + \Big(\frac{\partial \tilde{R}}{\partial \tilde{Q}_p}\Big)\Big](\tilde{Q}_p^{l+1} - \tilde{Q}_p^l) = -\tilde{R}^l$$

.....(3.31)

where l is the counter for the non-linear iterations and the subscript l signifies the corresponding pseudo-time-stepping scheme.

Each non-linear iteration involves the solution of a large multi-dimensional matrix operator, which in turn requires an iterative solution. For generality, we cast this "inner" iteration procedure as an independent pseudo-time-stepping scheme:

$$\Big[\frac{\tilde{\Gamma}_p}{\Delta \tau_k} + \frac{\tilde{\Gamma}_p}{\Delta \tau_l} + \Big(\frac{\partial \tilde{R}}{\partial \tilde{Q}_p}'\Big)\Big](\tilde{Q}_p^{k+1} - \tilde{Q}_p^k) = -\left[\Big[\frac{\tilde{\Gamma}_p V}{\Delta \tau_l} + \Big(\frac{\partial \tilde{R}}{\partial \tilde{Q}_p}\Big)\Big](\tilde{Q}_p^k - \tilde{Q}_p^l) - \tilde{R}^l\right]$$

.....(3.32)

where k is the counter for the linear iterations and the subscript k signifies the correspond pseudo time-step. The prime on the Jacobian on the LHS indicates that some approximate Jacobian (such as the first-order version) may be employed to suit the properties of the linear solver.

The final step concerns the solution of the coupled linear system, which is rewritten below in terms of the fluid and structural sub-systems:

$$\left[\frac{\tilde{\Gamma}_p}{\Delta \tau_k} + \begin{pmatrix} A' & B' & C' \\ D' & E' & F' \\ G' & H' & K' \end{pmatrix}\right] \Delta_k \begin{pmatrix} Q \\ Q_f \\ Q_s \end{pmatrix} = -\left[\begin{pmatrix} A & B & C \\ D & E & F \\ G & H & K \end{pmatrix} \Delta_l \begin{pmatrix} Q \\ Q_f \\ Q_s \end{pmatrix} - \tilde{R}^l\right]$$

.....(3.33)

Solution of the above system requires the solution of a full matrix system involving seven equations (in 1D) and 17 equations (in 3D). Fortunately, because of the iterative nature of the linear solution procedure and the inevitable need for domain decomposition in parallel computing environments, we can segregate the linear solution matrix in the following manner:

$$\left[\frac{\tilde{\Gamma}_p}{\Delta \tau_k} + \begin{pmatrix} A' & 0 & 0 \\ 0 & E' & 0 \\ 0 & 0 & K' \end{pmatrix}\right] \Delta_k \begin{pmatrix} Q \\ Q_f \\ Q_s \end{pmatrix} = -\left[\begin{pmatrix} A & B & C \\ D & E & F \\ G & H & K \end{pmatrix} \Delta_l \begin{pmatrix} Q \\ Q_f \\ Q_s \end{pmatrix} - \tilde{R}^l\right]$$

.....(3.34)

This means that the fluid, mesh and structural linear solutions can be carried out independent of each other (in parallel, if so desired). The coupling occurs through the RHS residual of the linear-sub-iterations and, importantly, full coupling is retained at the non-linear and physical time-stepping levels. This property has the important advantage of insuring that any time-step limitations ensuing from the solver partitioning only influences the selection of $\Delta\tau_k$ and not $\Delta\tau_l$ or the physical time-step, Δt.

3.6 Comparison to Staggered Solution Schemes

In this section, we compare the coupled solution schemes with segregated solution schemes. Representative classes of the segregated schemes have also been implemented in this study in order to carry out comparisons between them and the proposed coupled approach.

The following block-diagram, Fig 3.4, represents the coupled method. Starting with the fluid, structures and mesh solutions at time-level n, we iterate, either linearly or non-linearly, between the three sets of partitioned equation systems until convergence is attained. At this point, we advance the solution to the new time level, n+1.

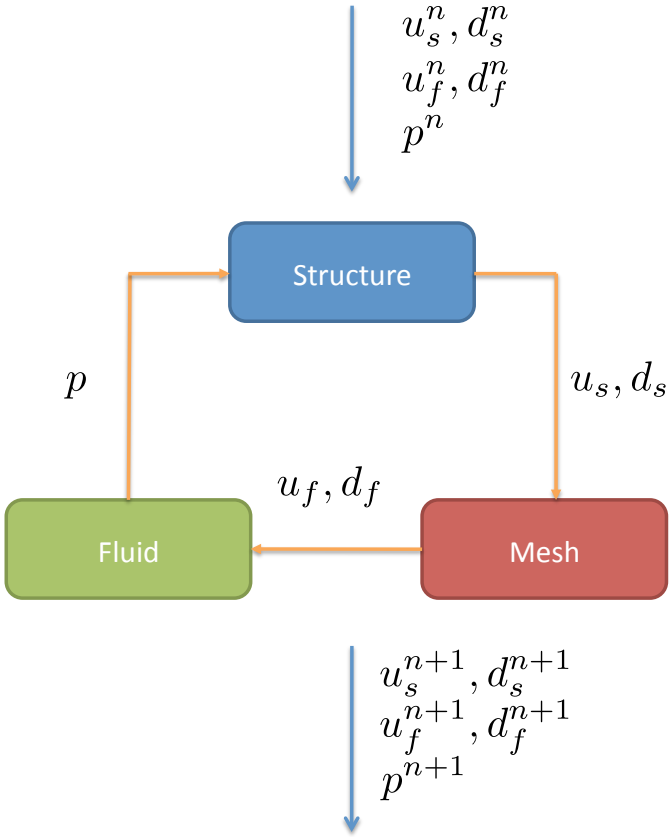

Figure 3.4 Block diagram showing coupled solution approach.

In the next block diagram shown in Fig. 3.5, the conventional serial staggered (CSS) scheme is shown. In this case, the three sets of equations are solved serially without any iterations between them. Thus, starting with the fluid dynamic solution at time-level n, we first solve the structures equations, followed by the fluid mesh equations, before finally solving the fluid dynamic equations to get the solution at time-level

n+1. The lack of an iterative procedure implies that the final solution will not be the same as in the coupled scheme, although for small enough time-steps, the solution will eventually converge to the correct solution. More importantly, because the interface solutions are "lagged", there will likely be stability issues that would need to be controlled by selecting small enough time-step sizes.

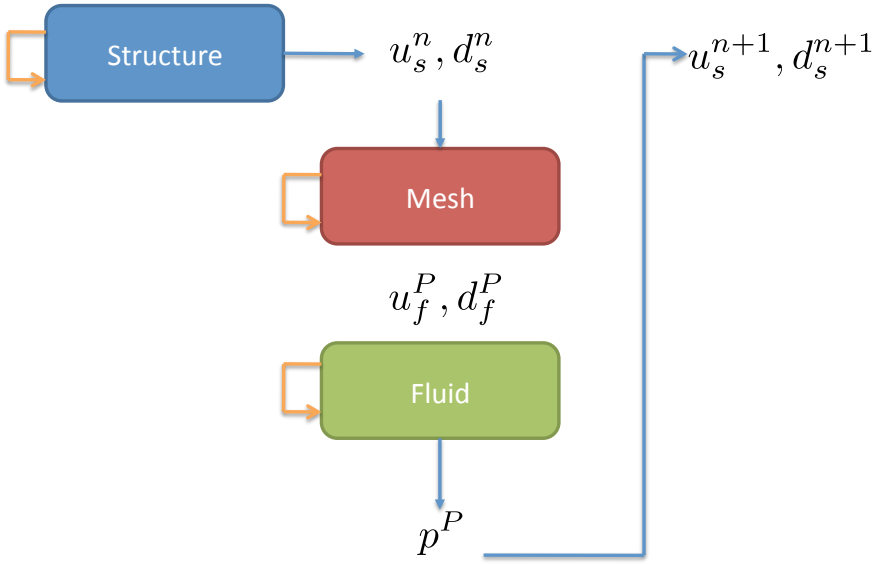

Figure 3.5 Block diagram showing conventional serial staggered scheme (CSS).

In the final block diagram, Fig. 3.6, we show the Generalized Serial Staggered (GSS) scheme. In this approach, structures and fluid/mesh solutions are separated by a half-time-step. In other words, the structures equations are solved at time levels, n and n+1, while the fluid and mesh equations are solved at time-levels n+1/2 and n+3/2 and so on. The overall scheme proceeds as follows. Initially, the structures equations are solved for time-level n providing surface deformations to the fluids and mesh equations. The fluid and mesh equations are then integrated to the n+1/2 time-level. Using the fluid dynamic forces from the n+1/2 time-level, we then obtain the structural solution for time-level n+1 and the cycle then repeats.

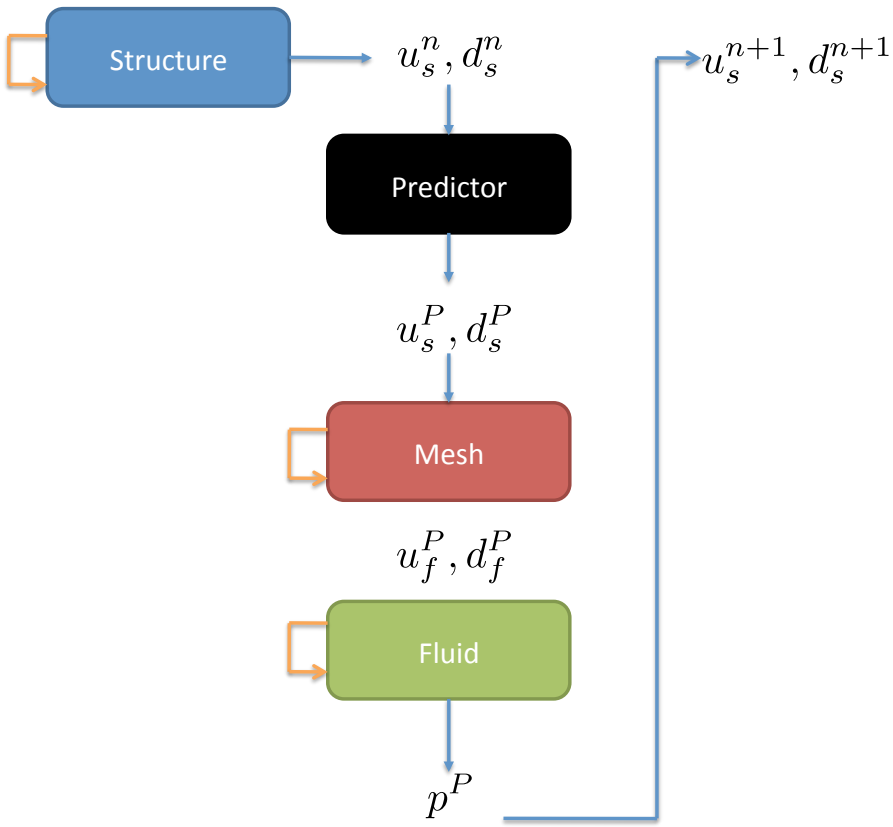

Figure 3.6 Block diagram showing the generalized serial staggered scheme (GSS).

Note that the GSS scheme is similar to the CSS scheme with the main difference being in the staggered-time approach. This latter distinction ensures second-order accuracy without requiring an sub-iterations.

4.0 Computational Infrastructure

4.1 Python Infrastructure

In the previous section, the three-way partitioned solution system was described. The computational domain is partitioned into two physical zones: the fluid and the structural zones. Further, each physical zone can typically be further partitioned, i.e., the fluids zone is partitioned into the fluid solver and the pseudo-structural solver for the mesh equations.

As discussed in the previous section, the physical partitioning and domain decomposition can be carried out either at the non-linear level or at the linear solution stage, with appropriate data exchange taking place at the end of each linear or k-level iteration. Full implicitness at the interface is obtained through the use of sub-iterations similar to traditional domain decomposition methods in standard CFD codes.

We have developed a modular CFD approach to provide a general implementation framework that is suitable for the fluid dynamics and structural equation systems. This framework is composed of a control layer and three functional layers of code (see Fig. 4.1). The control layer is simply the interface layer that provides the subroutine calls and data structures needed for each solver partition. Underlying the control layer is the first functional layer, the time-integration layer, which is responsible for the solution of the equations in each domain and the transfer of fluxes and grid motions at the fluid-structure interface. The second layer comprises of the spatial finite-volume discrete formulation for arbitrary numbers of partial differential equations and the solution of general linear systems. Finally, the third layer consists of flux definitions for each system of equations. The codes themselves are written in Fortran-90 with dynamic memory allocation and use pointers to avoid data replication between the layers or between the different solver partitions.

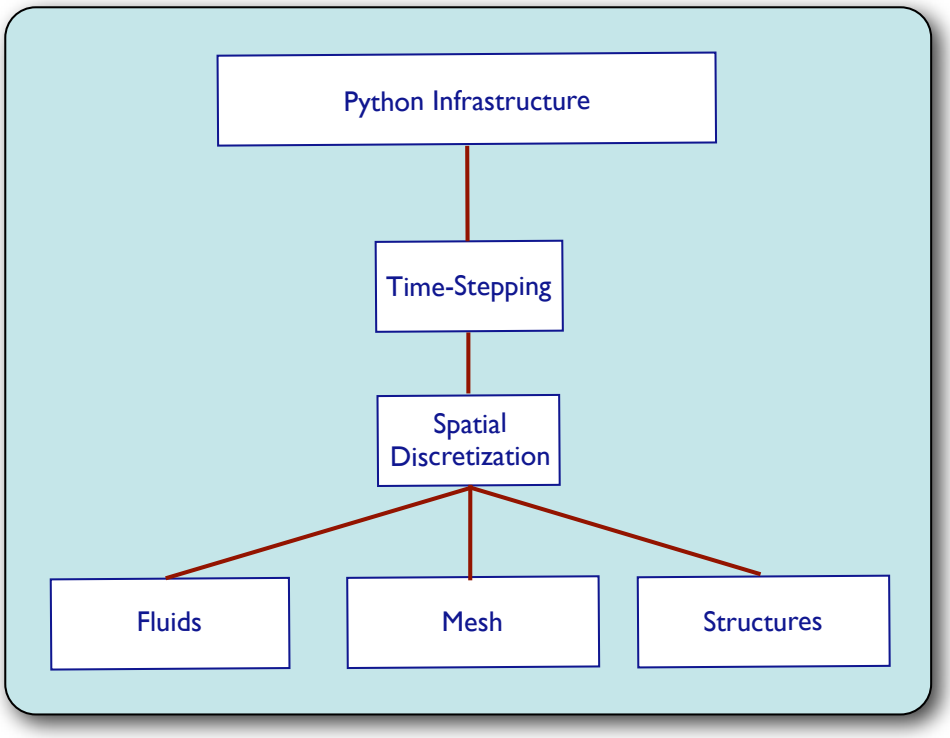

Figure 4.1 Schematic showing modular code structure used in the unified fluid-structure code development.

The solvers are coupled together using a light-weight Python-based infrastructure. The Python scripts control the data transfers between the solvers and the execution sequence of the algorithm. A section of the main Python script is shown in Fig. 4.2. The temporal solution of the fluid and structural systems involves three distinct instances of the solver, one for the fluids solution, one for the mesh solution and one for the structural solution. This solution process is orchestrated by the python code, which is tailored to the current application and, therefore, does not introduce any additional memory or CPU overheads. Further, the solution procedure also requires the periodic transfer of fluxes and mesh motions and/or linearized Jacobian matrices between the fluid and structural components of the problem and this data transfer is also achieved through the Python framework. Finally, the modular Python framework is also generalizeable for parallelization, although these aspects have not yet been developed.

```
for i in range(1,nsteps+1):

      for j in range(1,niter+1):
            #
            # run inner iterations
            #

            mesh1d.runsubstep(i,j)
            fluid1d.runsubstep(i,j)
            struct1d.runsubstep(i,j)

            #
            # get data to be exchanged
            #

            structData=struct1d.getboundarydata(structDataSize)
            meshData=mesh1d.getmeshdata(meshDataSize1,meshDataSize2)
            fluidData=fluid1d.getboundarydata(fluidDataSize)

            #
            # set data to be exchanged
            #

            struct1d.setboundarydata(fluidData)
            mesh1d.setboundarydata(structData)
            fluid1d.setmeshdata(meshData)

      fluid1d.update(i)
      struct1d.update(i)
      mesh1d.update(i)
```

It can be further seen that the three solvers: fluids, mesh and structures, share the same code for the time-stepping and spatial discretization layers and differ only in the system layer. In fact this is the major advantage of utilizing the modular-CFD infrastructure and there is little or no code replication of all the standard time integration and finite volume operations. Moreover, the code infrastructure is also be easy to maintain, upgrade and extend. For instance, changes in the fundamental flux formulation naturally extend to both fluid and structural systems. Further, the infrastructure allows ease of extension from one dimensions to two dimensions, again with the advantage of utilizing much of the same code and code structure, which greatly minimizes the time for development as well as debugging effort.

5.0 Results

Results of the individual components for fluids and structure solution as well as the fully coupled fluid-structures-mesh solution are presented in this section. The results shown are for the one-dimensional instance and concern a 1D beam or flow in a shock-tube. For the coupled fluid-structure case, we consider flow in a shock-tube with an elastic beam forming one end-wall of the shock-tube. High pressure waves in the shock-tube cause the beam to compress and the shock-tube itself to expand. In addition to fundamental performance of the methods, we carry out several parametric studies, focusing on discrete-GCL effects, grid resolution, temporal accuracy, boundary condition robustness as well as a comparison of performance of the coupled method with the conventional and generalized segregated solution procedures.

5.1 Structures Code Performance

The stand-alone 1D structures code has been verified for both steady and unsteady operation on a flexible beam. Two different boundary condition options are of interest. For the physical structure, we are typically interested in the case where force is applied on one end of the beam. In the steady-case instance, the beam compresses in response to the force. On the other hand, for the non-physical structural solution that is used for fluid-mesh motion, the boundary condition of interest is fixed displacement, i.e., the displacement of one fluid boundary in response to a moving or deforming solid surface. Results using both boundary conditions are given here.

Force Boundary Condition

Figure 5.1 shows the scenario for a forced boundary beam. The beam is fixed at the right-end, while at the left-end, a force of a specified magnitude is applied. In a coupled scenario, this forced boundary would be exposed to the aerodynamic field and the force would be derived from the fluid pressure.

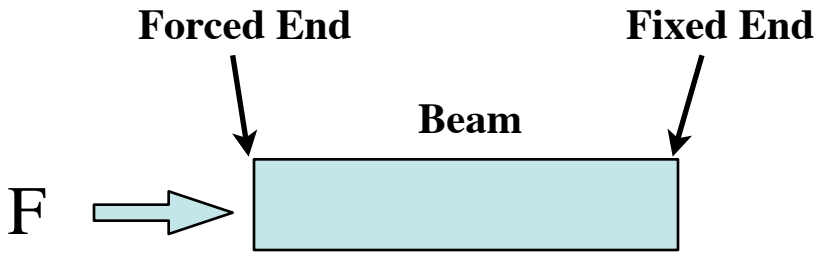

Figure 5.1 Schematic showing a 1D beam with boundary forcing on the left-end.

Convergence for a range of forcing magnitudes is shown in Fig. 5.2. For these steady cases, the plot shows the L2 norm of the residual a a function of the iteration number. Three difference forcings are shown ranging from a non-dimensionalized magnitude of 1.e-4 to 1.e-2 and 1.0. Convergence rates are approximately the same for all three cases, although the higher forces which lead to larger displacements seem to "saturate" at a higher residual level. This is probably due to less number of accurate digits that are preserved by the finite-precision of the computer for the higher displacement results. These results are for a CFL = 100 and show that the convergence rates (i.e., the slopes of the semi-log lines) are very nearly the same for the different forcings. In all cases, machine zero convergence is attained in about 100 iterations.

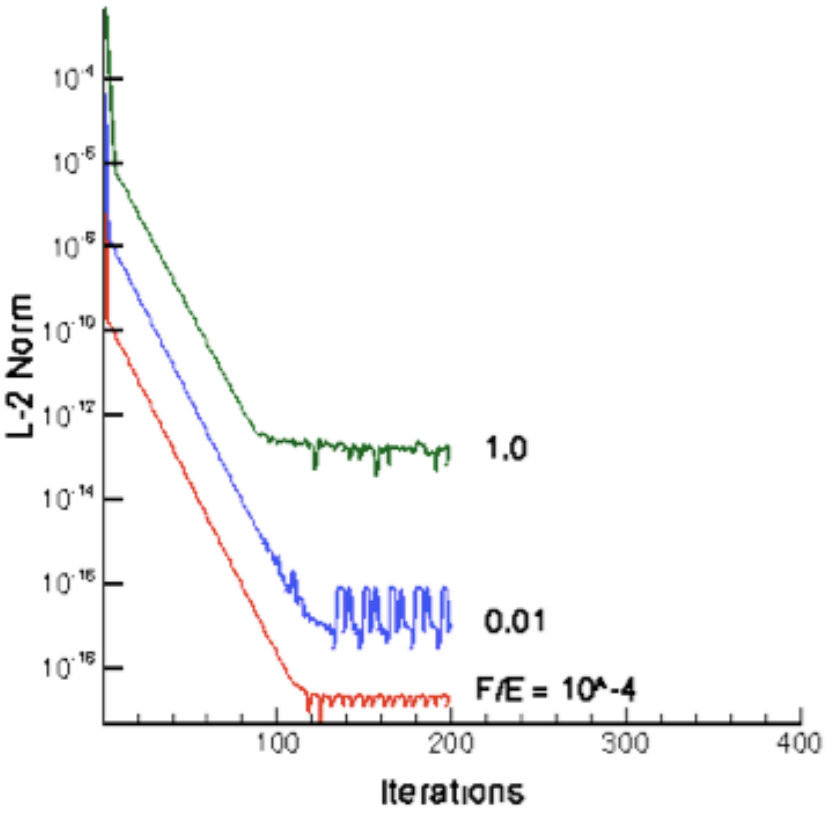

Figure 5.2 Convergence rates for different forcings of a 1D flexible beam using the stand-alone structures code.

Corresponding results for the beam displacement along the axial length of the beam are shown in Fig. 5.3. These results show that for the most strongly forced case, F/E = 1, the displacement at the left-end of the beam is about 1 m which, for a total beam length of 10 m, represents 10% of the total beam length. These results thus demonstrate that the fully implicit structural solver is robust and efficient even for heavily loaded beams.

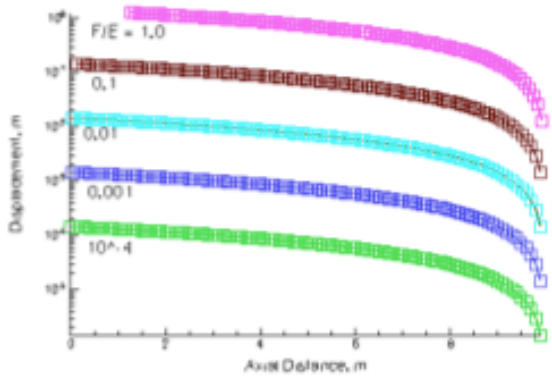

Figure 5.3 Beam displacements for different forcings of a 1D flexible beam using the stand-alone structures code.

Displacement Boundary Condition

As mentioned earlier, the force boundary condition is useful for physical structures, but for the mesh motion problem, the relevant boundary condition would be a specified displacement. Specifically, the displacement would arise from the motion or deformation of a physical surface and the fluid-dynamic mesh would need to deform to adjust to the new location of the boundary. Figure 5.4 shows the schematic of the fixed displacement boundary condition scenario.

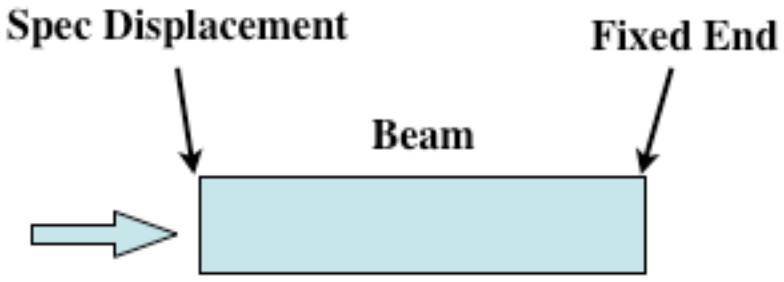

Fig. 5.4 Schematic of the fixed displacement boundary condition specification at teh left-end of the beam.

Convergence of the displacement boundary case for a CFL=100 (Fig. 5.5a) shows similar convergence rates to the previous forced boundary case. Note that the boundary displacement used in this case was 1 m for a 10-m long beam and so the case involved considerable beam deformation. Results are also shown for a lower CFL number of 10 (Fig. 5.5b) and the convergence is observed to be somewhat slower, which is to be expected. Interestingly, further decreases in the CFL number resulted in non-convergence, which seems counter to expectation.

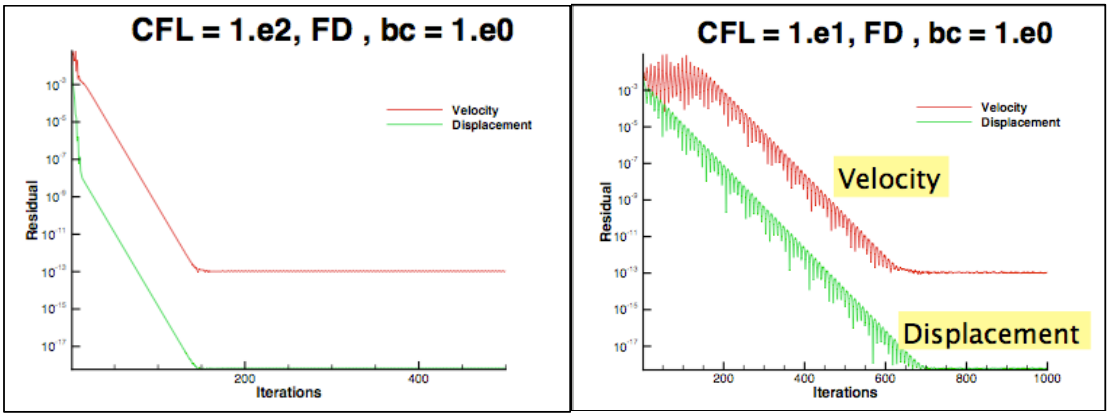

Fig. 5.4 Convergence results for fixed-displacement boundary condition.

The reason for the loss of stability for lower CFL numbers becomes evident when the interim solutions are carefully examined. For large boundary displacements, the internal grid points of the beam need to respond quickly to the compression (or expansion) of the beam. For small CFL numbers, the internal displacements tend to be very slow and may therefore lead to non-physical beam/mesh topologies.

To avoid this problem, it is therefore necessary to introduce a boundary condition relaxation which insures that the boundary displacement is no more than a fraction of the first cell-size:

$$d_s^{k+1} = d_s^k + \min\left[(d_{bc} - d_s^k),\ k(\Delta x_{ib})\right] \quad \text{.....(5.1)}$$

or,

$$\Delta d_s = \min\left[1, \left|\frac{k(\Delta x_{ib})}{d_{spec} - d_s^k}\right|\right](d_s^k - d_{bc}) \quad \text{.....(5.2)}$$

or,

$$\Delta d_s = -\omega(d_s^k - d_{bc}) \quad \text{.....(5.3)}$$

The last equation ensures that the boundary displacement is under-relaxed by an amount that limits the amount of displacement that can be applied to a fraction of the size of the first cell, Δx_{ib}.
The value of the fraction "K" is typically selected to be 0.1.

Convergence results with the boundary relaxation are shown in Fig. 5.5. The results for CL=10 are comparable to the earlier result when boundary relaxation was not used. The results for CFL=1 are now observed to be convergent, whereas this case proved unstable when the relaxation scheme was not invoked. We observe that the convergence result is quite slow, but this is expected given the relatively small value of the CFL number.

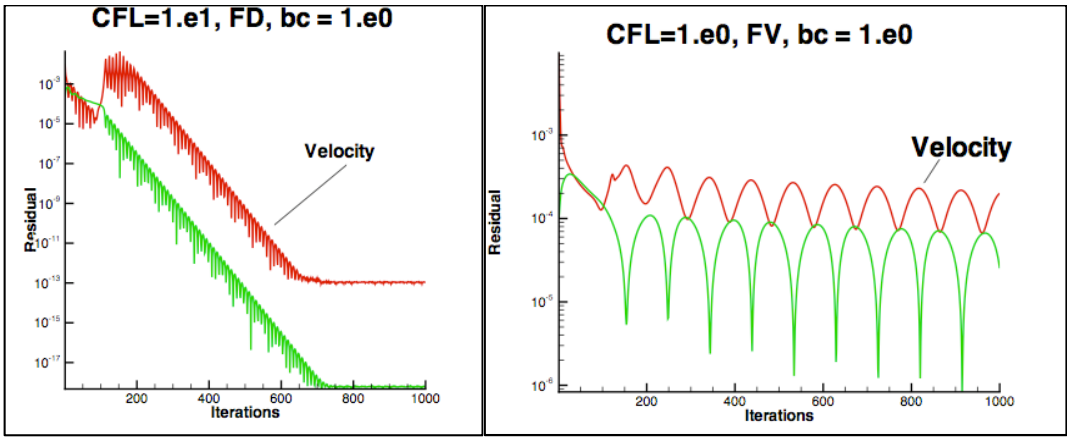

Fig. 5.5 Convergence results for fixed-displacement boundary condition using boundary under-relaxation.

5.2 Fluid Dynamics Code Performance

Shock-Tube Problem

The stand-alone fluid dynamics code has also been verified for steady and unsteady test cases for a variety of boundary conditions, including inflow, outflow, wall, etc. Here, we show representative unsteady results for a 1D shock-tube with the diaphragm at the center of the duct and walls on both ends (see Fig. 5.6).

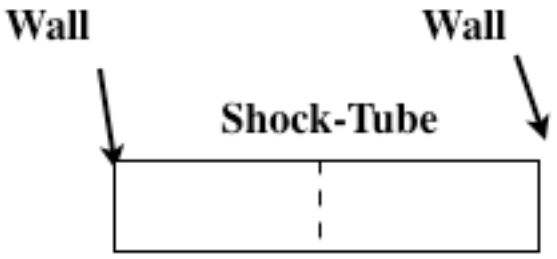

Fig. 5.6 Schematic of the shock-tube set-up used for fluid-dynamics code testing.

Figure 5.7 shows sample convergence results showing the residual drop in the sub-iterations (or pseudo-time-steps) for several consecutive physical time-steps. In this case, 10 sub-iterations were employed for each physical time-step and we note that about 5 orders of residual drop are obtained, indicating good performance of the dual-time algorithm.

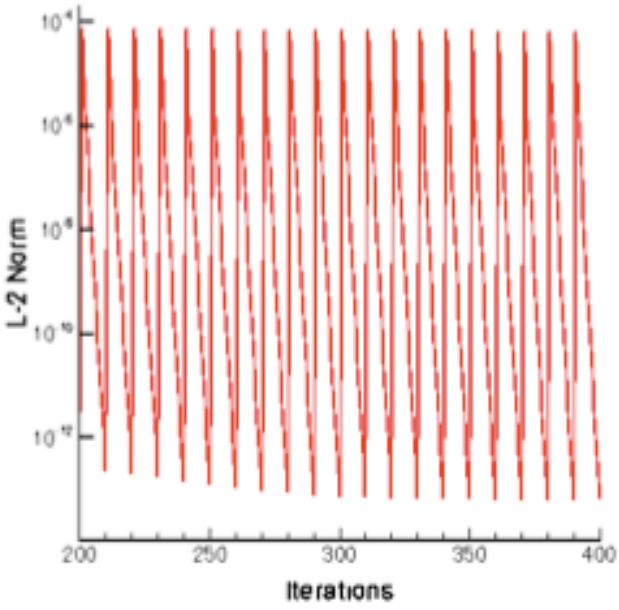

Figure 5.7 Convergence rates for the sub-iterations of the unsteady fluid dynamics code for a 1D shock-tube problem.

Figure 5.8 shows pressure variation along the shock-tube for several time-steps. We can observe the rarefaction wave moving leftward into the high pressure gas, while a shock and contact discontinuity proceed rightward into the low pressure gas. eventually, when the shock wave hits the right-wall of the shock tube, it is reflected and then travels leftward back towards the center of the shock tube. These results verify that the fluid dynamics code provides the right results for the shock tube computation.

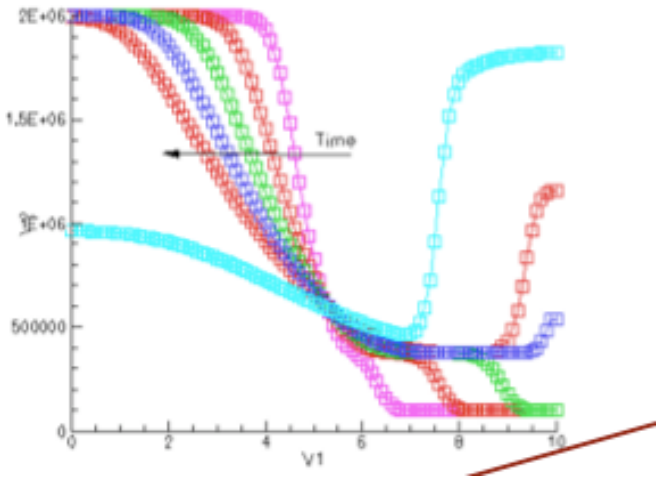

Figure 5.8 Pressure solutions at different time-steps of the unsteady fluid dynamics code for the 1D shock-tube problem.

Grid resolution studies and comparison versus the exact solution for a 1D shock-tube is provided in Fig. 5.9. Both pressure and velocity contours are shown at an instant in time prior to the shock-wave reaching the end-wall of the shock-tube. It is clear that the results agree very well with the exact solution and that increasing the number of grid points brings the agreement even closer. Note that the physical time-step size was also lowered for finer grids. The pseudo-CFL was maintained at 100 for all the calculations.

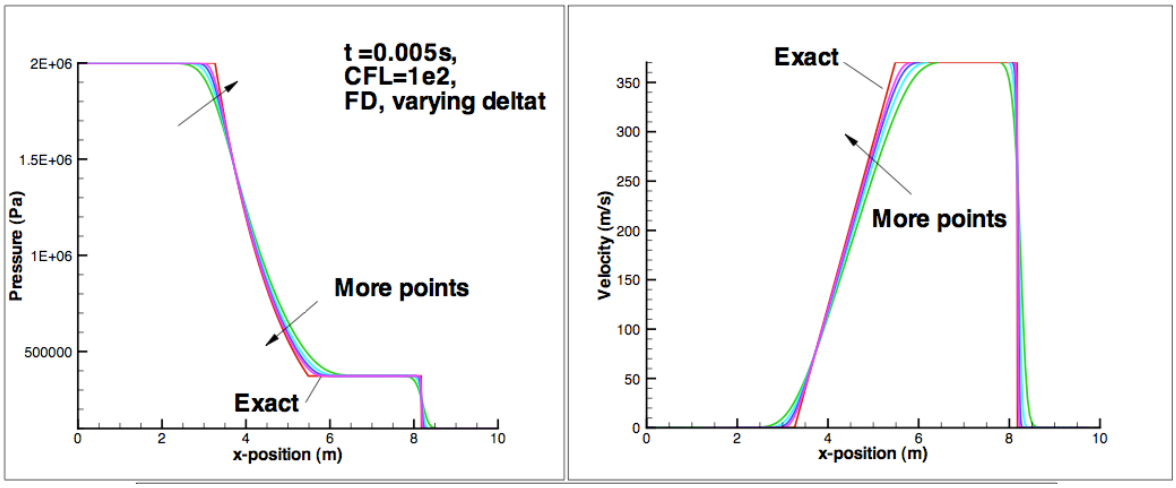

Figure 5.9 Pressure and velocity solutions at a given instant in time compared with the exact solution for a 1D shock-tube.

Additional tests of ducted flows with different boundary condition specifications have also been carried out to verify that the code performs well for different cases and conditions. These are not presented here because they do not add anything new to the discussions.

5.3 Fluid-Mesh Code Performance

Fluid-Mesh Convergence

Verification of the performance of the fluid and mesh motion code are provided next. Only steady results are shown here for a flow through a straight duct with an initial perturbation of the solution. Figure 5.10a shows the convergence rates for different CFLs with the initial perturbation applied only to the fluid solution. As a result, the mesh motion solution is static and there is no variation in the grid locations. In contrast, in Fig. 5.10b, both the fluid and mesh solutions are perturbed and, therefore, all equations vary throughout the flowfield and need to converge to the correct solution. The two convergence plots demonstrate that the fluids-mesh convergence rates are comparable to the fluid-only convergence rates for low CFL numbers, but slightly slower for the highest CFL number shown. Nevertheless, these results confirm that the fluid-mesh motion code is performing adequately and is robust and efficient.

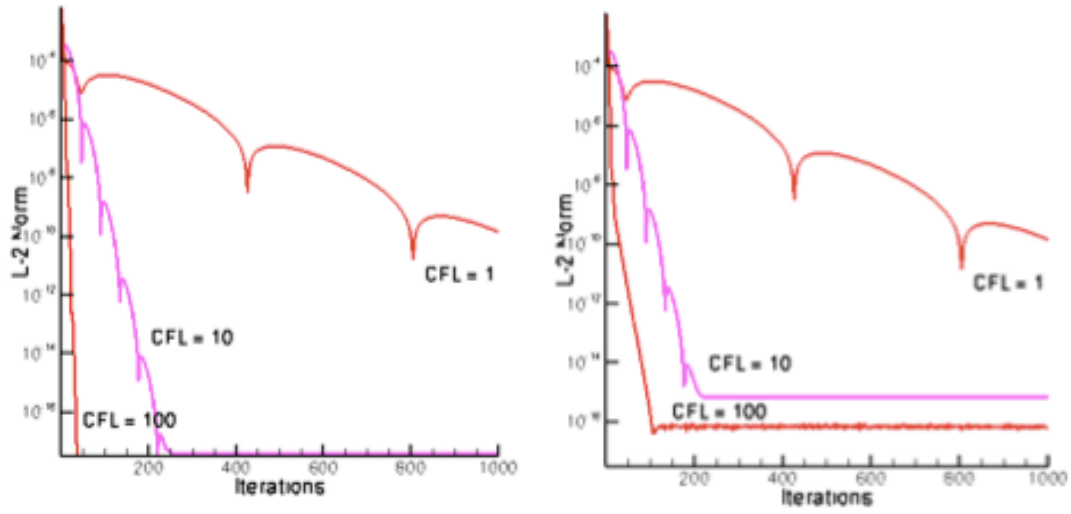

Figure 5.10 Convergence rates for steady fluid dynamics (left) and fluid dynamics + mesh (right) calculation in 1D duct.

As a second test of the fluid and mesh solutions, we consider the shock-tube problem again but impose a sinusoidal grid motion in the problem:

$$d_f(x,t) = \tfrac{1}{2}\sin(\omega_t t)\sin(\omega_x d_f(x,t)) \quad \text{.....(5.4)}$$

where the frequencies were chosen so that there would be no cell-penetration during the simulation, $\omega_t = \pi/2,\ \omega_x = \pi/5$.

Verification of Temporal Accuracy

The grid motion should not affect the shock-tube solution if the methodology is implemented correctly. Importantly, this test case also allows us to verify the two GCL schemes outlined in the previous section. Figures 5.11a and 5.11b show the error in the shock-tube solution (obtained by comparing the exact solution) as a function of the time-step size for the two GCL schemes. It is clear from the slope of the error that both GCL schemes demonstrate second-order temporal accuracy. Formal proof of the second order temporal accuracy of Method II is given in the Appendix.

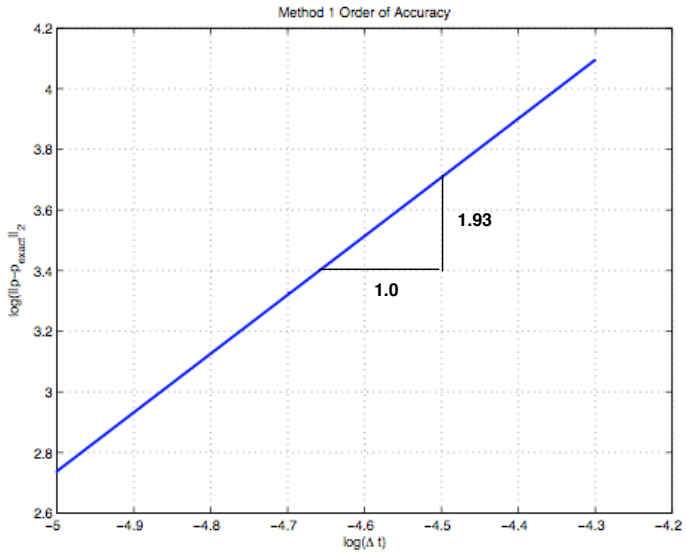

Figure 5.11a Temporal order of accuracy for shock-tube problem with moving fluid mesh. GCL Method I is used in these studies.

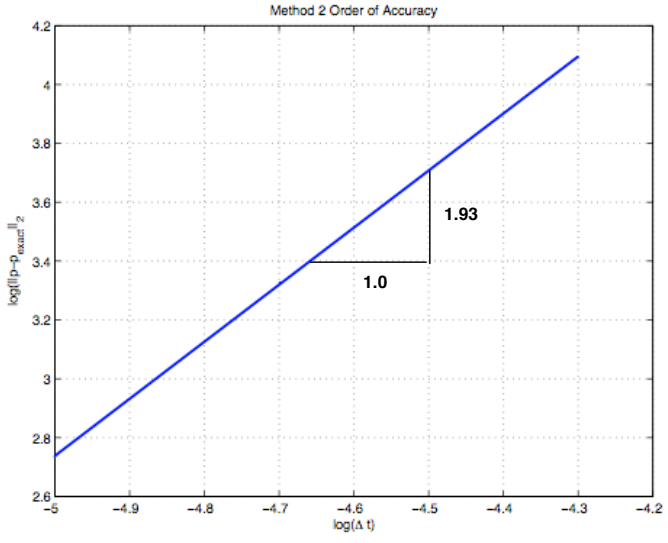

Figure 5.11b Temporal order of accuracy for shock-tube problem with moving fluid mesh. GCL method II is used in these studies.

While both schemes show second-order accuracy, we note that Method II is easier to implement for the partitioned systems used here. Specifically, Method I uses the flux formulation to define the interfacial mesh velocities and then the GCl constraint is used to define the cell volume. This means that the cell volume will not be precisely the same as the geometric volume of the cell. On the other hand, in Method II, the cell volume is calculated using the facial displacements and the interfacial velocities are given by applying consistent temporal discretizations to the interfacial displacements. The latter is more commonly applied in finite volume methods and is the recommended choice here as well.

5.4 Coupled CFD-CSD Code Performance

We next turn our attention to the fully coupled fluid-structure code. Here, we consider a 1D shock-tube with a flexible end-wall (see Fig. 5.12). In addition to the fluid dynamics solution in the tube, we also solve for the structural deformation of the end-wall, which is taken to be a flexible beam.

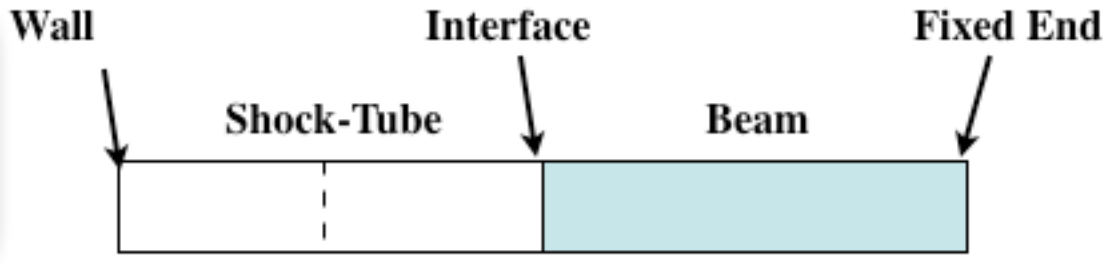

Fig. 5.12 Schematic of the shock-tube set-up with a flexible beam at the left-end. This set-up is used for the coupled fluids-structures testing.

Coupled Solution Verification

In Fig 5.13, the plot on the right shows pressure solutions in the shock-tube, which are qualitatively similar to the results shown earlier. In this case, however, when the shock hits the right-wall, the pressure rise causes the beam structure to compress as the wave traverses the beam. These results are shown on the right side. It can be observed that as the pressure rises on the right wall, the beam displacement increases. At the highest point shown, the pressure is about 10 atm, which is twenty times compared to the initial pressure (on the right side) of 1 atm. The corresponding beam displacement is observed to be about 25 mm. Importantly, the sub-iteration convergence (not shown) is virtually identical to the fluids-only case given earlier, which confirms the overall robustness and efficiency of the unified solution scheme.

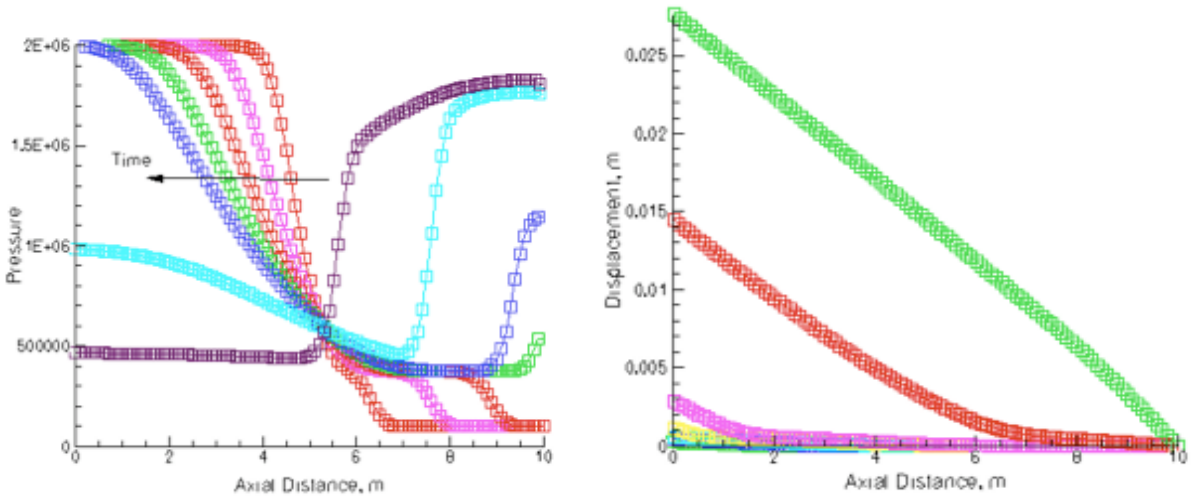

Figure 5.13 1D Shock tube calculation using coupled fluid-structures code with a flexible wall on one end.

To further verify the performance of the coupled fluid-structures code, we next perform a grid resolution study. Since there is no exact solution to this problem, we cannot formally verify order of accuracy, but we can check if the solution is grid converged. Figure 5.14 shows the results in the shock-tube with flexible end at a given instant in time using three different meshes and physical time-step sizes. It is clear that the pressure field on the fluids-side is approaching a consistent converged solution as a function of the grid resolution. On the structures side, the solution convergence is even close with the medium and fine results giving solutions that are nearly the same.

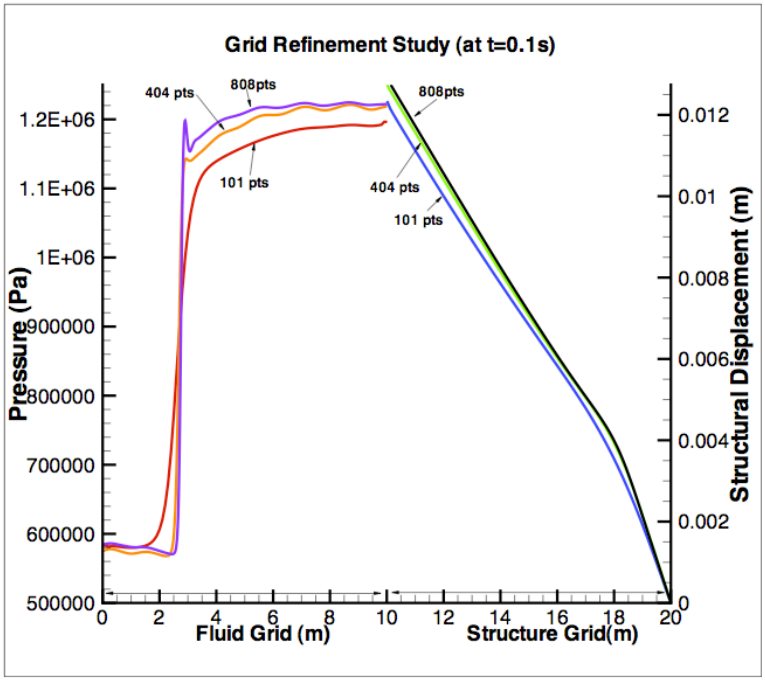

Figure 5.14 Grid resolution studies for 1D Shock tube calculation using coupled fluid-structures code with a flexible wall on one end.

Comparison of Coupled and Segregated Schemes

As a final set of comparisons, we consider the shock-tube case with the flexible end-wall and apply the conventional serial staggered (CSS) and generalized serial staggered (GSS) schemes in addition to the coupled scheme given earlier. This will allow us to compare the accuracy and performance of the different approaches to the solution of the fluid-mesh-structures problem. Figure 5.15 shows the fluid dynamic part of the solution for all three schemes and the results are observed to be indistinguishable from each other. This indicates that the three schemes are consistent in their formulation and converge to the same solution (at least to within plottable accuracy).

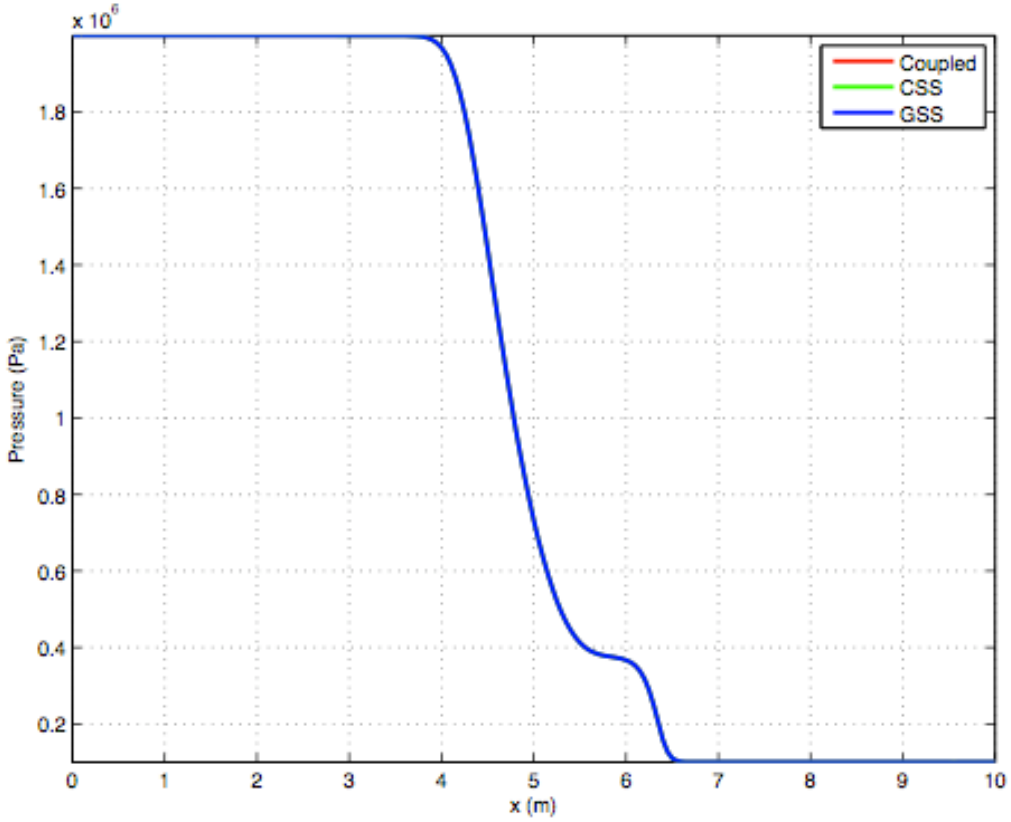

Figure 5.15 Comparison of Coupled, CSS and GSS solutions for the 1D Shock tube calculation with a flexible wall on one end.

Figures 5.16 show temporal resolution results for all three schemes: coupled, CSS and GSS. Since there is no exact solution for these cases, a very fine grid solution was used in lieu of an exact solution in order to calculate the error. All plots show similar error behavior and the schemes are observed to be approximately second-order accurate in time. The slightly lower than second-order slope is probably attributable to the approximate nature of the reference "exact" solution.

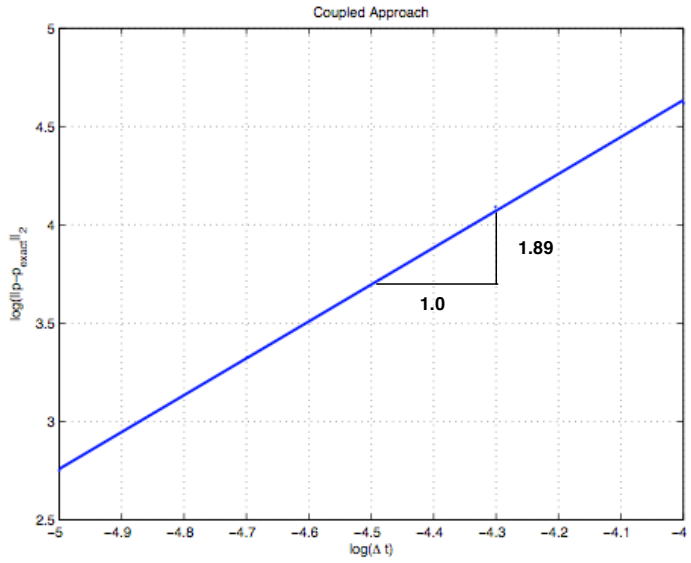

Figure 5.16a Temporal order of accuracy of the coupled scheme for the 1D Shock tube calculation with a flexible wall on one end.

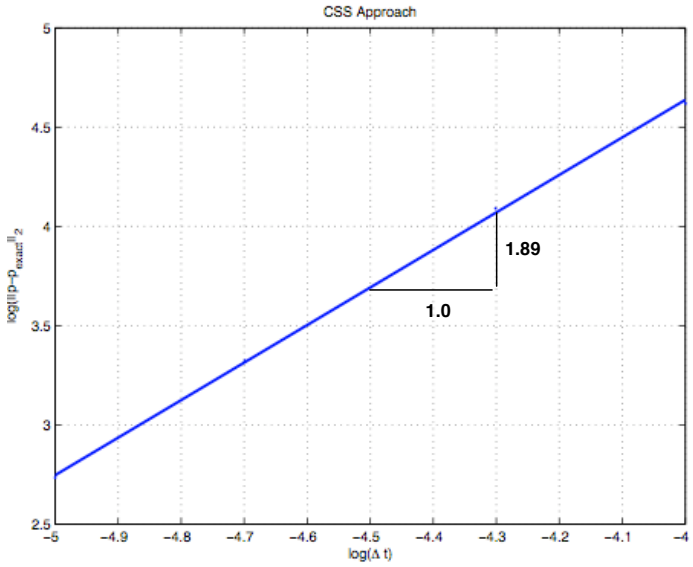

Figure 5.16b Temporal order of accuracy of the CSS scheme for the 1D Shock tube calculation with a flexible wall on one end.

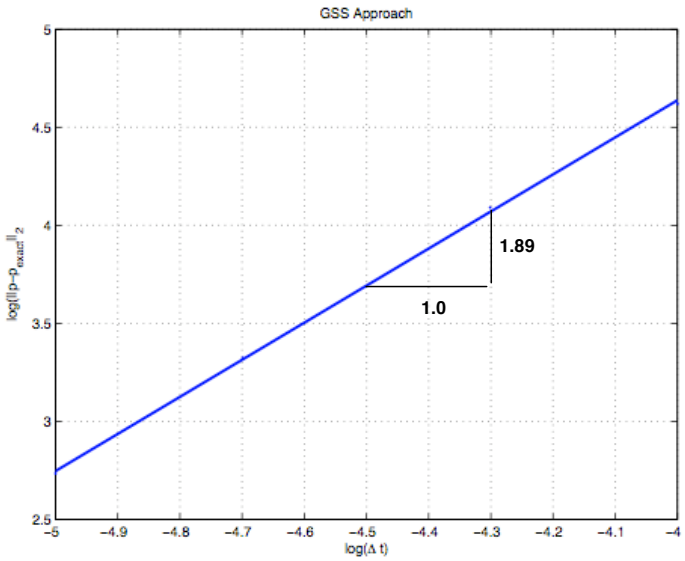

Figure 5.16c Temporal order of accuracy of the GSS scheme for the 1D Shock tube calculation with a flexible wall on one end.

Comparisons of the robustness of the three cases were also carried out by systematically running for increasingly larger physical time-step sizes. Interestingly, all schemes performed equally well in the robustness study as well. All schemes converged for time-steps less than or equal to 1.e-4s and diverged when larger time-steps were used. Based on these studies, it appears that the schemes share similar performance and accuracy for the one-dimensional test problems studied here.

6.0 Summary and Conclusions

The present work has concerned the development of a unified solution approach for coupled fluid-structure problems. The approach is based on using a common finite-volume framework for the fluid dynamics, mesh motion and structural dynamics equations. The finite-volume framework facilitates the implementation of fully conservative fluid-structure interface conditions and the coupled solution framework enables implicit non-linear solutions of the combined equations. In all three sets of equations are solved: the fluid-dynamics equations for the fluid flow, the structural dynamics equations in the flexible solid medium and a second set of structural equations that governs the motion of the fluid-dynamic mesh in response to the deforming solid surface. The three sets of equations are cast within a partitioned solution framework which are then solved sequentially using sub-iterations at each physical time-step. The algorithm is implemented within a modular finite-volume infrastructure that uses the same time-integration and discretization sub-routines for both the fluids and structures equations. A set of Python scripts are used to wrap the interface subroutines in each solver and control the sequence of execution of the solvers.

The main conclusions of the present study may be separated into three areas: numerical analysis of the structural dynamics equations, formulation of the unified algorithm, and numerical testing results. We present these in order in the following.

6.1 Numerical Analysis of the Structural Dynamics Equations

The structural dynamics equations are not conventionally written in integral form. To elucidate matters, the present research work considers the formulation of the structural system in both differential and integral forms. The equations themselves consist of the conservation of momentum in the solid and a set of compatibility relations that express the structural velocities as the temporal derivative of the solid-phase displacements. Fundamental dispersion analyses are used to determine the numerical characteristics of the structural equations. It is shown that they represent a set of hyperbolic equations and the eigenvalues correspond to the so-called p-waves and s-waves, also referred to as seismic waves. This means that the structural system shares many of the numerical properties of the fluid dynamics system, which greatly simplifies the construction of appropriate numerical algorithms.

Stability analysis of the structural equations show that the system is unconditionally unstable when explicit algorithms are used, but they are unconditionally **stable** for implicit schemes. Moreover, it is interesting that central differencing of the basic second-order conservation laws leads to adequate high-frequency damping, which is different from the classical result for first-order hyperbolic systems (such as the fluids equations). Consequently, there is no need to add additional artificial dissipation terms to the formulation. Nevertheless, we argue that the structural system is not naturally diagonally dominant, which is a limitation when it comes to the use of implicit relaxation algorithms such as the Gauss-Seidel method. Therefore, we have developed a spectral-radius-based artificial dissipation methodology that can be utilized to provide diagonal dominance. In practice, these additional terms will be necessary only in the discretization of the implicit operator and are not required for the calculation of the explicit residual. Numerical studies validate these findings.

6.2 Formulation of the Unified Algorithm

The unified algorithmic formulation developed in this work has several important elements:

1. Solver partitioning: It is not necessary to solve all the equations as one large coupled matrix system since all the equations do not appear simultaneously in all sections of the flow. Thus, the fluid dynamics and mesh equations are used only in the fluid zone, while the structural dynamics equations are used only in the solid medium. It therefore makes sense to partition the solution methodology according to zones. Moreover, the fluids and mesh equations are not directly coupled together: the mesh equations depend upon the structural dynamics and the fluid equations depend upon the interfacial mesh velocities. Thus, it is convenient to partition the equation system into three solver modules: fluids, mesh and structures. Because of the data dependencies, the best order in which to solve the equations is: structures first, followed by the mesh equations and finally the fluids equations. The three solvers are cast within a sub-iterative framework so that interfacial data are exchanged within the sub-iterative process, thereby ensuring that all parts of the equation system are treated implicitly at the physical time-level.

2. Time-scaling: Using the same physical time-step for all the solver modules means that the pseudo-structural equations representing the fluid-dynamics-mesh will also be solved time-accurately. However, since the mesh equations are not physical, time-accuracy is not relevant. Moreover, the slow propagation of waves through the mesh would likely result in poorer quality intermediate meshes. The best approach, therefore, is to use an infinitely large time-step for the mesh equations so that at each physical time-step, the mesh equations are solved to convergence. This is achieved by introducing a time-scaling parameter to the physical time-derivative in the mesh equation that artificially scales the physical time-step to be very large. Note that this scaling is done to the physical time-derivatives of the pseudo-structural mesh equations only.

3. Preconditioning: This refers to time-scaling of the pseudo-time-derivatives that is employed within the dual-time-framework to perform sub-iterations at each physical time-step. For steady-state computations, this formulation reduces to a single-time formulation that contains only the pseudo-time-derivatives (the physical time-derivatives are dropped). Preconditioning is used for the pseudo-time scaling to ensure that all the equations are advanced by the appropriate CFL number in the iteration process. The eigenvalues of the fluid system are the particle speeds and the acoustic speeds, while the eigenvalues of the structural equations are the seismic speeds. These speeds can be of very different magnitudes, hence leading to an additional source of stiffness. For the fully coupled system of equations, we have formulated a pseudo-time-scaling procedure that eliminates this stiffness from the pseudo-time system. For the partitioned-solver formulation that is used in this work, this scaling is not explicitly needed since it is straightforward to use the proper definition of the CFL number in each of the solvers, i.e., the CFL definition must be based on the eigenvalues of the relevant system eigenvalues in each solver. This is the approach followed here.

4. Geometric Conservation Law or GCL: Proper formulation of the cell volumes and interfacial velocities is needed to ensure that the discrete system satisfies the GCL property. This means that the numerical solution is able to exactly preserve a constant flow situation (in the presence of moving meshes). Two different dGCL-preserving schemes have been formulated. the first method starts with the fully coupled fluid-mesh-structural system and formulates the interfacial velocities as an inherent part of the flux formulation. The implication of doing this is that the cell volume must be computed in such a manner that the GCL property is maintained. The second method starts with the systems written as individual solver partitions. In that case, the interfacial velocity is not an explicit part of the flux formulations. Then, the cell volume may be computed using the geometry of the cell, i.e., by adding or subtracting the volumes swept by each cell face. The interfacial velocities are then computed as the time derivative of the facial displacements. To satisfy GCL, it is important that the same time discretization be used for the interfacial velocity calculation as is used for the physical time-derivatives in this equation. Both method shave been evaluated in this study, but since our method employs solver-partitioning, the second method is the more natural one to use. We note that this choice is consistent with the typical implementation used within the finite-volume community for moving mesh problems.

5. Interface Formulation: The coupling between fluids and structures and the coupling between structures and mesh are both controlled by the formulation of the interface conditions between the fluids and

structures physical zones. We have carefully laid out all the conservation laws and compatibility relations that must be satisfied at the interface. Importantly, the fluid dynamics transfers the forces at the surface to the solid equations. And, the structural dynamics transfers the deformation of the surface to the fluid mesh equations. In this initial work, we have assumed that the fluids and structural meshes are conforming, i.e., they have one-to-one point matching. As a result, the face areas of the two fluids and structural grid systems are precisely the same. We further note that the interface conditions are solved implicitly as boundary conditions within each solver partition and full implicitness at the physical time-level is ensured by the sub-iteration process.

6.3 Numerical Testing Results

A variety of important findings resulted from the numerical testing studies pursued in this work. These are summarized below:

1. Mesh System Boundary Condition Relaxation: The pseudo-structural equations that are used to represent mesh motion employ the displacement of the solid boundary surface as the boundary condition. We found, however, that if the boundary displacement is large, the fluid dynamic mesh may become non-physical (i.e., have cells with negative volumes). Our results show that this can happen particularly when the displacement is large and the physical time-step size (used in the mesh equations) is small. In order to counter this effect, we have formulated an under-relaxation procedure for the boundary displacement. This ensures that the specified boundary displacement is limited by the physical size of the first cell, thereby avoiding any non-physical cell volumes during the iterative process.

2. Three-solver and two-solver partitioning: In our work, we initially formulated the coupled system with a two solver partitioning system that was based strictly on the two physical zones: i.e., fluids and structures. The fluids partition contained the combined solver for the fluids and mesh equations, while the structures solver contained only the physical structural dynamics solver. The issue with such a formulation was that the mesh equations are constrained to use the same physical time-step as the fluid dynamics equations. As noted earlier, this is unnecessary since the mesh equations are not physical and therefore work best with an infinitely large time-step size. A second observation in this regard is the fact that the mesh equations are not directly dependent on the fluids equations (of course, the fluids equations depend upon the mesh equations through the definition of the interfacial velocity). Because of this one-way coupling, it is preferable to partition the coupled system into three parts: i.e., the fluids solver, the structures solver and the fluid-mesh solver. Our studies show that the three-solver approach works as well as the two-solver approach in terms of performance and accuracy for the same time-step specifications. The important advantage with the three-solver approach is that it has the flexibility of using an "infinite" physical time-step for the mesh equations. Our studies have further confirmed that this choice does not impact robustness of the overall numerical solution.

3. Two different GCL methods: Our studies indicate that both GCL procedures outlined earlier are provably second-order accurate in time. However, method II is ideally suited to the three-solver partitioning approach used in this study and is therefore recommended for use. We note again that this choice is consistent with current practice in finite-volume moving-mesh codes.

4. Coupled and segregated schemes: In addition to the coupled approach proposed here, we also formulated and compared the performance of two popular segregated approaches: the so-called Conventional Serial Staggered (CSS) scheme and the Generalized Serial Staggered Scheme (GSS). We point out that our modular approach enabled the testing of these alternate algorithms simply by reconfiguring only the main execution loop and no changes were required in the solver modules themselves. Interestingly, our 1D tests shows that all three schemes were provably second-order accurate and equally robust

in terms of stability as a function of the time-step size. While these results are a bit surprising (we would expect the CSS scheme to be less accurate and potentially less robust), we note that the current results are one-dimensional and the extension to multi-dimensions may reveal more significant differences between the approaches.

6.4 Future Work

The current work has established a coupled fluid-mesh-structures framework for multi-dimensional problems, but the current practical studies have relied on the one-dimensional formulation. Initial implementation of the methods for two-dimensional problems have been carried out, but the methods have not been fully tested in this mode as yet. Future work will focus on the extension of these studies in two-dimensions and (eventually) to more practical three-dimensional problems as well. Specific two-dimensional problems include airfoil flows, while three-dimensional studies will focus on rotorcraft problems.

The present work also considered grid-point-matching between the fluid dynamics and structural dynamics. This clearly simplifies the specification of fluxes between the two systems at the fluid-structure interface. Future work will extend the formulation to allow arbitrary meshes in the fluids and structural domains. In both cases, it is nevertheless important that the mesh systems utilize the same consistent definition of the surface. The extension to arbitrary meshes will be handled by a domain connectivity formulation which will carry out appropriate area-weighted mappings of the fluid-dynamic cell faces into the structural cell faces on the surface. Such a component does not presently exist but would be an important component for future development.

Extension of the current methodology for parallel processing is further area of future work. The general approach would be to introduce a mesh and domain-decomposition layer that handles the communication between different grid partitions for each solver. The solver modules themselves that have been developed as part of this work will remain unchanged and will be invoked by the meshing layer to perform the necessary solution on a mesh-block at a time. In this way, the current infrastructure can be used as the basis for more comprehensive future development without having to rewrite all aspects of the code.

References and Citations

1. Porter, L., ``NASA's New Aeronautics Research Program,'' 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2007.

2. Bousman, W. G., ``Putting the Aero into Aeroelasticity,'' Paper, Eigth ARO Workshop on Aeroelasticity of Rotorcraft Systems, Penn State University, University Park, PA, 1999.

3. Datta, A., Nixon, M. and Chopra, I., "Review of Rotor Loads Prediction With the Emergence of Rotorcraft CFD," 31st European Rotorcraft Forum, Florence, Italy, September 2005.

4. Potsdam, M., Yeo, H., and Johnson, W., "Rotor Air-loads Prediction Using Loose Aerodynamic/ Structural Coupling," Journal of Aircraft, Vol. 43, No. 3, May-June 2006.

5. Pomin, H. and Wagner, S. “Aeroelastic Analysis of Helicopter Rotor Blades on Deformable Chimera Grids,” Journal of Aircraft, Vol. 41, No. 3, May-June 2004.

6. Sitaraman, J., Datta., A., Baeder, I. and Chopra, I., “Coupled CFD/CSD Prediction of Rotor Aerodynamic and Structural Dynamic Loads For Three Critical Flight Conditions,” Presented at the 31st European Rotorcraft Forum, Firenze, Italy, Sep 13-15, 2005.

7. Saberi, H., Khoshlahjeh, M., Ormiston, R. and Rutkowski, M., "Overview of RCAS and Application to Advanced Rotorcraft Problems," Proceedings of the American Helicopter Society 4th Decennial Specialists' Conference on Aeromechanics, San Francisco, California, January 21-23, 2004.

8. Johnson, W. "Technology Drivers in the Development of CAMRAD II," Proceedings of American Helicopter Society Aeromechanics Specialists Conference, San Francisco, CA, January 19-21 1994.

9. Bauchau O.A. "DYMORE: A Finite Element Based Tool for the Analysis of Nonlinear Flexible Multibody Systems", Techical Note, School of Aerospace Engineering, Georgia Institute of Technology Atlanta, GA, February 2, 2007.

10. Sankaran, V. and Merkle, C., ``Efficiency and Accuracy Issues in Navier-Stokes Computations,'' AIAA 2000-2251, AIAA Fluids 2000 Conference, Denver, CO, June, 2000.

11. Sankaran, V. and Merkle, C.,``Analysis of Preconditioning Methods for the Euler and Navier-Stokes Equations,'' Von Karman Institute Lecture Series on Computational Fluid Dynamics, March, 1999.

12. Sankaran, V. and Merkle, C., ``Artificial Dissipation Control for Unsteady Computations,'' AIAA 2003-3695, AIAA Computational Fluid Dynamics Conference, Orlando, FL, 2003.

13. D. Li, G. Xia, V. Sankaran and C.L. Merkle, ``Computational Framework for Complex Fluids Applications,'' 3rd International Conference on Computational Fluid Dynamics, Toronto, Canada, July 12-16, 2004.

14. Farhat, C., van Der Zee, K. and Geuzaine, P., ``Provably Second-Order Time-Accurate Loosely-Coupled Solution Algorithms for Transient Nonlinear Computational Aeroelasticity,'' Comput. Methods in Applied Mech. Engrg., 195, 2006, pp. 1973-2001.

15. Piperno, S. and Farhat, C., ``Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems---Part II: Energy Transfer Analysis and Three-Dimensional Applications,'' Comput. Methods in Applied Mech. Engrg., 190, 2001, pp. 3147-3170.

16. Sitaraman, J. and Baeder, J., "Evaluation of the Wake Prediction Methodologies Used in CFD Based Rotor Air-load Computation," AIAA Paper 2006-3472, 24th AIAA Applied Aerodynamics Conference, San Francisco, CA, June 5-8, 2006.

17. Sitaraman, J., Datta, A., Baeder, J. D., and Chopra, I., “Fundamental Understanding and Prediction of Rotor Vibratory Loads in High-Speed Forward Flight,” 29th European Rotorcraft Forum, Friedrichshafen, Germany, September 2003.

18. Wheel, M. A., ``A Geometrically Versatile Finite Volume Formulation for Plane Elastostatic Stree Analysis," J. of Strain Analysis, Vol. 31, No. 2, 1996.

19. Xia, G.H., Zhao, Y., Yeo, J.H. and Lv, X., ``A 3D Implicit unstructured-grid finite volume method for structural dynamics," Computational Mechanics, Springer, 2006.

20. Slone, A., Pericelous, K., Bailey, C. and Cross, M., ``Dynamic Fluid Structure Interaction using Finite Volume Unstructured Mesh Procedures," Computers and Structures, Vol. 80, 2002, pp. 371-390.

21. Guruswamy, G. P., ``Development and Applications of a Large Scale Fluids/Structures Simulation Process on Clusters,'' Computers and Fluids, Vol. 36, 2007, pp. 530-539.

22. Liou, M.-S., ``A Sequel to AUSM, Part II: AUSM+up for all Speeds," Journal of Computational Physics, 214, 2006, pp. 137-170.

23. Jameson, A., ``Analysis and Design of Numerical Schemes for Gas Dynamics Part 1: Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence,'' RIACS Technical Report 94.15, International Journal of Computational Fluid Dynamics, Vol. 4, 1995, pp. 171-218.

24. Alonso, J. J. and Jameson, A., ``Fully Implicit Time-Marching Aeroelastic Solutions," AIAA-94-0056, 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV, 1994.

25. Yang, Z. and Mavriplis, D. J., ``Unstructured Dynamic Meshes with Higher-Order Time-Integration Schemes for the Unsteady Navier-Stokes Equations," AIAA 2005-1222, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2005.

26. A. Slone, K. Pericleous, C. Bailey, M. Cross, “Dynamic fluid-structure interaction using finite volume unstructured mesh procedures” Computers and Structures 80 (2002) 371-390.

27. R. Jaiman, X. Jiao, X. Geubelle, E. Loth, “Assesment of conservative load transfer for fluid-solid interface with non-matching meshes” Int. Journal Numerical Methods in Engineering 64 (2005) 2014-2038.

28. M. Price, K. Koffi, S. Raghunathan, “Static aeroelastic effect of local skin deformations on a NACA0012 airfoil” AIAA 2002-1210.

Appendix

Proof of temporal order of accuracy of the discrete GCL approach considered here. See attached.

1 Appendix A: Proof in One Dimension

The following section contains the proof that Scheme 2 satisfies its DGCL and possess second-order accuracy for prescribed motion.

1.0.1 Introduction

Let us define a few operators which will be used in the derivation. In the definitions, $d_\zeta = \zeta \Delta x,\ t^\tau = \tau \Delta t$ where ζ, τ are dummy variables for the spatial and temporal indices respectively. The operators are defined for $\forall \tau \geq 0,\ \forall \zeta \in [0, N]$ where N is the total number of grid points. We will use the notation that for some quantity f which is a function of space and time

$$(f(x,t))|_{x=d_a}^{t=t^b} = f(d_a, t^b) \tag{1}$$

We can extend the above definition to functions of functions, say, for example if

$$B(f)_\zeta^\tau = f(d_\zeta, t^\tau) + f(d_{\zeta+1}, t^{\tau+1}) \tag{2}$$

Then,

$$(B(f))|_{x=d_a}^{t=t^b} = B(f)_a^b = f(d_a, t^b) + f(d_{a+1}, t^{b+1}) \tag{3}$$

Now, we are ready to introduce the operators which are defined below

$$\delta_x(\cdot)_\zeta^\tau = \frac{(\cdot)|_{x=d_{\zeta+1}}^{t=t^\tau} + (\cdot)|_{x=d_\zeta}^{t=t^\tau}}{2} - \frac{(\cdot)|_{x=d_\zeta}^{t=t^\tau} + (\cdot)_{x=d_{\zeta-1}}^{t=t^\tau}}{2} \tag{4}$$

$$\delta_t(\cdot)_\zeta^\tau = \frac{3}{2}(\cdot)|_{x=d_\zeta}^{t=t^\tau} - 2(\cdot)|_{x=d_\zeta}^{t=t^{\tau-1}} + \frac{1}{2}(\cdot)|_{x=d_\zeta}^{t=t^{\tau-2}} \tag{5}$$

$$\delta_t^1(\cdot)_\zeta^\tau = (\cdot)|_{x=d_\zeta}^{t=t^\tau} - (\cdot)|_{x=d_\zeta}^{t=t^{\tau-1}} \tag{6}$$

We can use the above operators to define another operator shown below

$$\begin{aligned}
\delta_{x,t}(\cdot)_\zeta^\tau &= \delta_x(\delta_t(\cdot))_\zeta^\tau \\
&= \frac{\delta_t(\cdot)|_{x=d_{\zeta+1}}^{t=t^\tau} + \delta_t(\cdot)|_{x=d_\zeta}^{t=t^\tau}}{2} - \frac{\delta_t(\cdot)|_{x=d_\zeta}^{t=t^\tau} + \delta_t(\cdot)|_{x=d_{\zeta-1}}^{t=t^\tau}}{2} \\
&= \frac{\delta_t(\cdot)_{\zeta+1}^\tau + \delta_t(\cdot)_\zeta^\tau}{2} - \frac{\delta_t(\cdot)_\zeta^\tau + \delta_t(\cdot)_{\zeta-1}^\tau}{2}
\end{aligned} \tag{7}$$

Similarly,

$$\begin{aligned}
\delta_{t,x}(\cdot)_\zeta^\tau &= \delta_t(\delta_x(\cdot))_\zeta^\tau \\
&= \frac{3}{2}\delta_x(\cdot)|_{x=d_\zeta}^{t=t^\tau} - 2\delta_x(\cdot)|_{x=d_\zeta}^{t=t^{\tau-1}} + \frac{1}{2}\delta_x(\cdot)|_{x=d_\zeta}^{t=t^{\tau-2}} \\
&= \frac{3}{2}\delta_x(\cdot)_\zeta^\tau - 2\delta_x(\cdot)_\zeta^{\tau-1} + \frac{1}{2}\delta_x(\cdot)_\zeta^{\tau-2}
\end{aligned} \tag{8}$$

Because δ_x and δ_t are linear, we have that

$$\delta_{t,x}(\cdot)_\zeta^\tau = \delta_{x,t}(\cdot)_\zeta^\tau \tag{9}$$

In this method, the mesh velocity is defined as

$$u_i^{n+1} = \frac{\delta_t(d)_i^{n+1}}{\Delta t} \tag{10}$$

and the volume $V^\star$ of a cell is defined as follows

$$(V_i^\star)^{n+1} = \frac{2}{3}\left(\delta_x(u)_i^{n+1}\Delta t + 2(V_i^\star)^n - \frac{1}{2}(V_i^\star)^{n-1}\right) \tag{11}$$

which can be written more concisely using the above defined operators as

$$\delta_t(V^\star)_i^{n+1} = \delta_x(u)_i^{n+1}\Delta t \tag{12}$$

Putting this back into (12), we get

$$\delta_t(V^\star)_i^{n+1} = \delta_x\left(\frac{\delta_t(d)}{\Delta t}\right)_i^{n+1}\Delta t = \delta_{x,t}(d)_i^{n+1} \tag{13}$$

In the case of motion induced by a displacement of the structure that is part of the unknowns of the coupled fluid-structure interaction problem, we have

$$u_i^{P,n+1} = \frac{\delta_t(d^P)_i^{n+1}}{\Delta t} \tag{14}$$

where d^P are the predicted positions and u^P is the predicted mesh velocity. Here the volume $(V^{\star,P})$ would be defined as

$$\delta_t(V^{\star,P})_i^{n+1} = \delta_x(u^P)_i^{n+1}\Delta t = \delta_{x,t}(d^P)_i^{n+1} \tag{15}$$

Now, if we created a C^0-interpolant of the predicted displacements $(d^P(t))$ and plugged it into the above equation, we would obtain

$$\delta_t(\overline{V^{\star,P}})_i^{n+1} = \delta_x(\overline{u^P})_i^{n+1}\Delta t = \delta_{x,t}(d^P(t))_i^{n+1} \tag{16}$$

We can also have the exact volume which is shown below

$$V_i(t^{n+1}) = \delta_x(d(t))_i^{n+1} \tag{17}$$

We also have the predicted volume V^P and the interpolated predicted volume $V^P(t)$, the exact form of which depends on the form of the predictor. In order to simplify notation, we shall drop the spatial index i though it should be understood that we are talking about a given cell.

1.0.2 DGCL

This method uses the implicit 3-point BDF with a volume modification and can be written as

$$\boxed{\delta_t(V^{\star,P}Q)^{n+1} + \Delta t\tilde{E}(Q^{n+1}, u^{P,n+1}) = \Delta t\tilde{E}_v(Q^{n+1}, d^{P,n+1})} \tag{18}$$

where the quantity $V^{\star,P}$ is calculated from the following equation

$$\delta_t(V^{\star,P})^{n+1} = \delta_x(u^P)^{n+1}\Delta t \tag{19}$$

If we plug in a constant solution $Q = Q^\dagger$, we get

$$\delta_t(V^{\star,P})^{n+1}Q^\dagger + \Delta t\tilde{E}(Q^\dagger, u^{P,n+1}) = \Delta t\tilde{E}_v(Q^\dagger, d^{P,n+1}) \tag{20}$$

We can simplify the flux terms as follows

$$\tilde{E_v}(Q^\dagger, d^{P,n+1}) = 0 \tag{21}$$
$$\tilde{E}(Q^\dagger, u^{P,n+1}) = -Q^\dagger\tilde{F}(u^{P,n+1}) \tag{22}$$

where

$$\tilde{F}(u^{P,n+1}) = \delta_x(u^P)^{n+1} \tag{23}$$

Therefore, we get the DGCL for this scheme which is

$$\boxed{\delta_t(V^{\star,P})^{n+1} - \Delta t\delta_x(u^P)^{n+1} = 0} \tag{24}$$

Now we need to see if the scheme satisfies its DGCL. We can plug in (19) into the LHS of the above equation to get

$$\Delta t\delta_x(u^P)^{n+1} - \Delta t\delta_x(u^P)^{n+1} = 0 \tag{25}$$

Thus, this method satisfies its DGCL.

1.0.3 Accuracy of the scheme

Before we assess the accuracy of the scheme, let us assume that

$$\exists\, q \in \mathbb{R}, q \geq 1/\forall t \in \mathbb{R}, \quad d^P(t) - d(t) = O(\Delta t^q) \tag{26}$$

Now, when we plug the exact solution into the scheme we obtain the truncation error which is shown below

$$\Psi = \delta_t(\overline{V^{\star,P}}Q(t))^{n+1} + \Delta t\tilde{E}(Q(t^{n+1}), \overline{u^{P,n+1}}) - \Delta t\tilde{E}_v(Q(t^{n+1}), d^P(t^{n+1})) \tag{27}$$

If we add and subtract $\delta_t(V^P(t)Q(t))^{n+1}$ then we obtain

$$\begin{aligned}\Psi = \delta_t(\overline{V^{\star,P}}Q(t))^{n+1} - \delta_t(V^P(t)Q(t))^{n+1} + \Delta t\tilde{E}(Q(t^{n+1}), \overline{u^{P,n+1}}) - \\ \Delta t\tilde{E}_v(Q(t^{n+1}), d^P(t^{n+1})) + \delta_t(V^P(t)Q(t))^{n+1}\end{aligned} \tag{28}$$

Now, we will simplify the last term in the above equation. We can expand the interpolated predicted volume about the exact volume to obtain

$$(V^P(t)Q(t)) = (V(t)Q(t)) + O(||d^P(t) - d(t)||) \tag{29}$$

Therefore, by linearity of δ_t, we have

$$\delta_t(V^P(t)Q(t))^{n+1} = \delta_t(V(t)Q(t))^{n+1} + \delta_t(O(||d^P(t) - d(t)||))^{n+1} \tag{30}$$

From a Taylor series expansion of the quantity $(V_i^P(t)Q_i(t))$, we obtain

$$\delta_t(V(t)Q(t))^{n+1} = \Delta t\frac{d}{dt}(V(t)Q(t))(t^{n+1}) + O(\Delta t^3) \tag{31}$$

Also, we have that

$$\begin{aligned}\delta_t(O(||d^P(t)-d(t)||))^{n+1} &= \frac{3}{2}O(||d^P(t)-d(t)||)(t^{n+1}) - 2O(||d^P(t)-d(t)||)(t^n)+ \\ &\frac{1}{2}O(||d^P(t)-d(t)||)(t^{n-1}) \\ &= \frac{3}{2}C_1(t^{n+1},\Delta t(t^{n+1}))||d^P(t^{n+1})-d(t^{n+1})|| \\ &- 2C_1(t^n,\Delta t(t^n))||d^P(t^n)-d(t^n)|| + \frac{1}{2}C_1(t^{n-1},\Delta t(t^{n-1}))||d^P(t^{n-1}) \ -d(t^{n-1})|| \\ &= \sum_{k=-1}^{1} O(||d^P(t^{n+k})-d(t^{n+k})||) \end{aligned} \tag{32}$$

Therefore, we have that

$$\delta_t(V^P(t)Q(t))^{n+1} = \Delta t\frac{d}{dt}(V(t)Q(t))(t^{n+1}) + \Delta t\sum_{k=-1}^{1} O(||d^P(t^{n+k})-d(t^{n+k})||) + O(\Delta t^3) \tag{33}$$

Now, we know that from the ALE form of the equations

$$\Delta t\frac{d}{dt}(V(t)Q(t))(t^{n+1}) = -\Delta t\tilde{E}(Q(t^{n+1}),u(t^{n+1})) + \Delta t\tilde{E_v}(Q(t^{n+1}),d(t^{n+1})) \tag{34}$$

Using this, we get that

$$\begin{aligned}\delta_t(V^P(t)Q(t))^{n+1} = &-\Delta t\tilde{E}(Q(t^{n+1}),u(t^{n+1})) + \Delta t\tilde{E_v}(Q(t^{n+1}),d(t^{n+1}))+ \\ &\Delta t\sum_{k=-1}^{1} O(||d^P(t^{n+k})-d(t^{n+k})||) + O(\Delta t^3)\end{aligned}$$

Plugging this back into the truncation error, we get

$$\begin{aligned}\Psi = &\delta_t(\overline{V^{\star,P}}Q(t))^{n+1} - \delta_t(V^P(t)Q(t))^{n+1} + \Delta t\tilde{E}(Q(t^{n+1}),\overline{u^{P,n+1}}) - \Delta t\tilde{E_v}(Q(t^{n+1}),d^P(t^{n+1})) \\ &- \Delta t\tilde{E}(Q(t^{n+1}),u(t^{n+1})) + \Delta t\tilde{E_v}(Q(t^{n+1}),d(t^{n+1})) + \Delta t\sum_{k=-1}^{1} O(||d^P(t^{n+k})-d(t^{n+k})||) + O(\Delta t^3)\end{aligned}$$

We will now try and simplify the flux terms in the above equation. If $\tilde{E}$ is sufficiently smooth, we can expand it as

$$\begin{aligned}\tilde{E}(Q(t^{n+1}),\overline{u^{P,n+1}}) = &\tilde{E}(Q(t^{n+1}),u(t^{n+1})) + \nabla_u\tilde{E}(Q(t^{n+1}),u(t^{n+1}))(\overline{u^{P,n+1}}-u(t^{n+1})) \\ &+ O(||\overline{u^{P,n+1}}-u(t^{n+1})||^2)\end{aligned} \tag{35}$$

We know that

$$\overline{u^{P,n+1}} = \frac{\delta_t(d^P(t))^{n+1}}{\Delta t} \tag{36}$$

$$u(t^{n+1}) = \frac{\delta_t(d(t))^{n+1}}{\Delta t} + O(\Delta t^2) \tag{37}$$

$$\implies \overline{u^{P,n+1}} - u(t^{n+1}) = \frac{\delta_t(d^P(t)-d(t))^{n+1}}{\Delta t} + O(\Delta t^2) \tag{38}$$

Now, we have that

$$\begin{aligned}\frac{\delta_t(d^P(t)-d(t))^{n+1}}{\Delta t} &= \frac{1}{\Delta t}\left[\frac{3}{2}(d^P(t^{n+1})-d(t^{n+1}))-2(d^P(t^n)-d(t^n))+\frac{1}{2}(d^P(t^{n-1})-d(t^{n-1}))\right]\\ &= \frac{1}{\Delta t}\Bigg[\frac{3}{2}(C_2(t^{n+1},\Delta t(t^{n+1}))\Delta t^q(t^{n+1}))-2(C_2(t^n,\Delta t(t^n))\Delta t^q(t^n))\\ &+\frac{1}{2}(C_2(t^{n-1},\Delta t(t^{n-1}))\Delta t^q(t^{n-1}))\Bigg]\\ &= \frac{1}{\Delta t}\left[O(||d^P(t^{n+1})-d(t^{n+1})||)+O(||d^P(t^n)-d(t^n)||)+O(||d^P(t^{n-1})-d(t^{n-1})||)\right]\\ &= \frac{1}{\Delta t}\sum_{k=-1}^{1}O(||d^P(t^{n+k})-d(t^{n+k})||) \qquad (39)\end{aligned}$$

So we get

$$\overline{u^{P,n+1}}-u(t^{n+1})=\Delta t^{-1}\sum_{k=-1}^{1}O(||d^P(t^{n+k})-d(t^{n+k})||)+O(\Delta t^2) \qquad (40)$$

Therefore, we have that

$$\tilde{E}(Q(t^{n+1}),\overline{u^{P,n+1}})=\tilde{E}(Q(t^{n+1}),u(t^{n+1}))+\Delta t^{-1}\sum_{k=-1}^{1}O(||d^P(t^{n+k})-d(t^{n+k})||)+O(\Delta t^2)$$

$$\implies \Delta t\tilde{E}(Q(t^{n+1}),\overline{u^{P,n+1}})=\Delta t\tilde{E}(Q(t^{n+1}),u(t^{n+1}))+\sum_{k=-1}^{1}O(||d^P(t^{n+k})-d(t^{n+k})||)+O(\Delta t^3)$$

Similarly, if $\tilde{E}_v$ is sufficiently smooth, we can expand it as

$$\begin{aligned}\tilde{E}_v(Q(t^{n+1}),d^P(t^{n+1})) &= \tilde{E}_v(Q(t^{n+1}),d(t^{n+1}))+O(||d^P(t^{n+1})-d(t^{n+1})||)\\ \implies \Delta t\tilde{E}_v(Q(t^{n+1}),d^P(t^{n+1})) &= \Delta t\tilde{E}_v(Q(t^{n+1}),d(t^{n+1}))+\Delta tO(||d^P(t^{n+1})-d(t^{n+1})||)\end{aligned} \qquad (41)$$

Putting all of this back into the truncation error, we obtain

$$\Psi=\delta_t(\overline{V^{\star,P}}Q(t))^{n+1}-\delta_t(V^P(t)Q(t))^{n+1}+(1+\Delta t)\sum_{k=-1}^{1}O(||d^P(t^{n+k})-d(t^{n+k})||)+O(\Delta t^3) \qquad (42)$$

Now, we need to simplify the first term. We know that

$$\delta_t(\overline{V^{\star,P}})^{n+1}=\delta_{t,x}(d^P(t))^{n+1} \qquad (43)$$

$$V(t^{n+1})=\delta_x(d(t))^{n+1} \qquad (44)$$

Now, we know that (we have omitted the temporal superscript from δ_x to indicate that the following equations hold at any time)

$$\delta_x(d(t)-d^P(t))=O(||d^P(t)-d(t)||) \qquad (45)$$

$$V^P(t)=V(t)+O(||d^P(t)-d(t)||) \qquad (46)$$

Therefore, we have that

$$\delta_x(d(t)) = \delta_x(d^P(t) + d(t) - d^P(t)) = \delta_x(d^P(t)) + \delta_x(d(t) - d^P(t)) = V(t) \tag{47}$$

$$\implies \delta_x(d^P(t)) = V(t) - \delta_x(d(t) - d^P(t)) = V^P(t) + O(||d^P(t) - d(t)||) \tag{48}$$

If we now apply the $\delta_t(\ \cdot\ )^{n+1}$ operator on both sides, we obtain

$$\delta_{t,x}(\ d^P(t))^{n+1} = \delta_t(V^P(t))^{n+1} + \delta_t(O(||d^P(t) - d(t)||)\ )^{n+1} \tag{49}$$

This gives us a difference equation of the form

$$\delta_t(\overline{V^{\star,P}} - V^P(t) + O(||d^P(t) - d(t)||))^{n+1} = 0 \tag{50}$$

The solution to the above difference equation is

$$\overline{V^{\star,P,n}} - V^P(t^n) + O(||d^P(t^n) - d(t^n)||) = \frac{C_1}{3^n} + C_2 \tag{51}$$

where we need two initial conditions to solve for the constants. We can assume that at time t^0,

$$\overline{V^{\star,P,0}} - V^P(t^0) = 0 \implies \frac{C_1}{3^0} + C_2 = O(||d^P(t^0) - d(t^0)||) \tag{52}$$

At the first time step, if we used a first-order temporal discretization, we would obtain

$$\delta_t^1(\overline{V^{\star,P}} - V^P + O(||d^P(t^n) - d(t^n)||)^1 = 0 \tag{53}$$

$$\implies \overline{V^{\star,P,0}} - V^P(t^0) + O(||d^P(t^0) - d(t^0)||) = \overline{V^{\star,P,1}} - V^P(t^1) + O(||d^P(t^1) - d(t^1)||) \tag{54}$$

$$\implies \frac{C_1}{3^1} + C_2 = O(||d^P(t^0) - d(t^0)||) \tag{55}$$

Therefore, we get that

$$C_1 = 0,\ C_2 = O(||d^P(t^0) - d(t^0)||) \tag{56}$$

$$\implies \overline{V^{\star,P,n}} = V^P(t^n) + O(||d^P(t^0) - d(t^0)||) + O(||d^P(t^n) - d(t^n)||) \tag{57}$$

We can simplify the above equation using the assumption about the prediction error (26) or we can assume that $d^P(t^0) = d(t^0)$ to get

$$\overline{V^{\star,P,n}} = V^P(t^n) + O(||d^P(t^n) - d(t^n)||) \tag{58}$$

Thus, we see that the modified volume differs from the interpolated predicted volume volume by a term which depends on the form of the predictor. If the predictor gives the exact mesh positions, then the modified volume is equal to the interpolated true volume. Now, we can use the above information to evaluate the truncation error term shown below

$$\delta_t(\overline{V^{\star,P}}Q(t))^{n+1} - \delta_t(V^P(t)Q(t))^{n+1} = \delta_t((\overline{V^{\star,P}} - V^P(t))Q(t))^{n+1} \tag{59}$$

$$= \delta_t((O(||d^P(t) - d(t)||)Q(t))^{n+1} = \delta_t(O(||d^P(t) - d(t)||))^{n+1} \tag{60}$$

From (32), we have that

$$\delta_t(O(||d^P(t) - d(t)||))^{n+1} = \sum_{k=-1}^{1} O(||d^P(t^{n+k}) - d(t^{n+k})||) \tag{61}$$

Thus plugging this into the truncation error, we obtain

$$\Psi = (1 + \Delta t) \sum_{k=-1}^{1} O(||d^P(t^{n+k}) - d(t^{n+k})||) + O(\Delta t^3) \tag{62}$$

Thus, the truncation error depends on the temporal order of accuracy of the predictor in the case of non-prescribed motion. In the case of prescribed motion, we obtain a second-order accurate scheme.

If we assume that the prediction error, i.e., $d^P(t) - d(t)$, is of polynomial form, we can show that the above error reduces to

$$\Psi = \Delta t \sum_{k=-1}^{1} O(||d^P(t^{n+k}) - d(t^{n+k})||) + O(\Delta t^3) \tag{63}$$