

FLUID-STRUCTURE INTERACTION WITH A STAGED ALGORITHM

Mario A. Storti, Norberto N. Nigro, Rodrigo R. Paz, Lisandro D. Dalcín, Gustavo A. Ríos Rodríguez and Ezequiel López

Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), INTEC(CONICET-UNL), Güemes 3450, (S3000GLN), Santa Fe, Argentina, mstorti@intec.unl.edu.ar,

<http://www.cimec.org.ar/mstorti>

Keywords: Compressible flow, fluid-structure interaction

Abstract. A common approach to solving fluid-structure interaction problems is to solve each subproblem in a partitioned procedure where time and space discretization methods could be different. Such a scheme simplifies explicit/implicit integration and it is in favor of the use of different codes specialized on each sub-area. In this work a staggered fluid-structure coupling algorithm is considered. For each time step a “stage-loop” is performed. In the first stage a high order predictor is used for the structure state, then the fluid and the structure systems are advanced in that order. In subsequent stages of the loop each system uses the previously computed state of the other system until convergence. For weakly coupled problems a stable and efficient algorithm is obtained using one stage and an accurate enough predictor. For strongly coupled problems, stability is enhanced by increasing the number of stages in the loop. If the stage loop is iterated until convergence, a monolithic scheme is recovered. In addition, two items that are specially important in fluid structure problems are discussed, namely invariance of the stabilization terms and dynamic absorbing boundary conditions. Finally, numerical examples are presented.

1 INTRODUCTION

Multidisciplinary and Multiphysics coupled problems represent nowadays a paradigm when studying/analyzing even more complex phenomena that appear in nature and in new technologies. There exists a great number of problems where converge different physical processes (or models), each interacting with others in a strong or weak fashion (e.g., Acoustics/Noise disturbances in flexible structures, Magneto-Hydrodynamics devices, Micro-Electro-Mechanical devices, Thermo-Mechanical problems like continuous casting process, Fluid-Structure interaction like wing flutter problem or flow-induced pipe vibrations, etc.). In the fluid-structure interaction area, the dynamic interaction between an elastic structure and a compressible fluid has been the subject of intensive investigations in the last years (Stein et al., 2003; Gnesin and Rzadkowski, 2005; Piperno and Farhat, 2001; Lefrancois, 2005). This article is concerned with the numerical integration of this type of problems when they are coupled in a loose or strong manner.

For simple structural problems (like hinged rigid rods with one or two vibrational degrees of freedom) it is possible to combine into a single (simple) formulation the fluid and the structural governing equations (Dowell et al., 1995). In those cases, a fully explicit or fully implicit treatment of the coupled fluid-structure equations is attainable. The "monolithic" methods (see, for example, the "direct" and "quasi-direct" coupling approaches described in Tezduyar et al. (2004, 2006a,b)), can be very robust but are not in general very modular, and also difficult to reach good parallel efficiency with. Furthermore, the monolithic coupled formulation would change significantly if different fluid and/or structure models were considered.

An efficient alternative is to solve each subproblem in a partitioned procedure where time and space discretization methods could be different. Such a scheme simplifies explicit/implicit integration and it is in favor of the use of different codes specialized on each sub-area. In this work a staggered fluid-structure coupling algorithm is considered. A detailed description of the 'state of the art' in the computational fluid-structure interaction can be found in Mittal and Tezduyar (1995); Kalro and Tezduyar (2000); Tezduyar and Osawa (2001); Piperno and Farhat (2001); Felippa et al. (2001); Park and Felippa (2000); Dettmer and Peric (2006) and the references therein.

After introducing the staged algorithm, two key points for fluid structure interaction with ALE formulations are discussed: invariance of the ALE formulation (specially the stabilization term) and wall boundary conditions, and dynamic boundary conditions. Finally an example showing the aerodynamics of a falling ellipse is shown. A more detailed example with analysis of temporal precision of the scheme is presented in a companion paper at this same conference (Paz et al., 2006).

2 STRONGLY COUPLED PARTITIONED STAGED ALGORITHM

In this section the temporal algorithm that performs the coupling between the structure and the fluid codes is described, so that if the most outer loop, i.e. the 'stage loop', converges then a 'strongly coupled' algorithm is obtained. The basic staggered algorithm considered in this work proceeds as follows: (i) transfer the motion of the wet boundary of the solid to the fluid problem, (ii) update the position of the fluid boundary and the bulk fluid mesh accordingly, (iii) advance the fluid system and compute new pressures (and the stress field if it is necessary), (iv) convert the new fluid pressure (and stress field) into a structural load, and (v) advance the structural system under the flow loads. Such a staggered procedure, which can be treated as a weakly coupled solution algorithm, can also be equipped with an outer loop in order to assure

the convergence of the interaction process. The algorithm can be stated as follow:

```

1: Initialize variables:
2: for  $n = 0$  to  $n_{\text{step}}$  do {Main time step loop}
3:    $t^n = n\Delta t$ ,
4:   {FLUID CODE: previous to advance time step}
5:   receive ('struct-step-ok',  $n - 1$ ) message from STRUCTURE
6:   receive  $u^n$  from STRUCTURE
7:    $\mathbf{X}^n = \text{CMD}(\mathbf{u}^n)$  {run CMD code}
8:    $\mathbf{u}^{(n+1)P} = \mathbf{u}^{(n+1,0)} = \text{predictor}(\mathbf{u}^n, \mathbf{u}^{n-1})$  { compute predictor }
9:   send ('fluid-step-ok',  $n$ ) message to STRUCTURE
10:  {STRUCTURE CODE: previous to advance time step}
11:  receive ('fluid-step-ok',  $n$ ) message from FLUID
12:  receive  $\mathbf{w}^n$  {fluid state}
13:  send ('struct-stage-ok', -1) message to FLUID
14:  for  $i = 0$  to  $n_{\text{stage}}$  do {stage loop}
15:    {FLUID CODE: previous to the stage iteration}
16:    receive ('struct-stage-ok',  $i - 1$ ) message from STRUCTURE
17:    receive  $\mathbf{u}^{(n+1,i)}$  from STRUCTURE
18:     $\mathbf{X}^{n+1,i+1} = \text{CMD}(\mathbf{u}^{n+1,i})$ 
19:    {Compute skin normals and velocities}
20:    for  $k = 0$  to  $n_{\text{nw}}^t$  do {Fluid Newton loop}
21:       $\mathbf{w}^{n+1,i+1} = \text{CFD}(\mathbf{w}^n, \mathbf{X}^{n+1,i+1}, \mathbf{X}^n)$ 
22:    end for
23:    send  $\mathbf{w}^{n+1,i+1}$  to STRUCTURE
24:    {FLUID CODE: after each stage iteration}
25:    send ('fluid-stage-ok',  $i$ )
26:    {STRUCTURE CODE: previous to the stage iteration}
27:    receive ('fluid-stage-ok',  $i$ )
28:    receive  $\mathbf{w}^{n+1,i+1}$ 
29:    compute structural loads ( $\mathbf{w}^n, \mathbf{w}^{n+1,i+1}$ )
30:    for  $k = 0$  to  $n_{\text{nw}}^t$  do {Struct Newton loop}
31:       $\mathbf{u}^{n+1,i+1} = \text{CSD}(\mathbf{u}^n, \mathbf{w}^n, \mathbf{w}^{n+1,i+1})$ 
32:    end for
33:    send  $\mathbf{u}^{n+1,i+1}$  to FLUID
34:    {STRUCTURE CODE: after each stage iteration}
35:    send ('struct-stage-ok',  $i$ )
36:  end for
37:  {FLUID CODE: after each time step}
38:  receive ('struct-stage-ok',  $n_{\text{stage}} - 1$ )
39:  send  $\mathbf{u}^n$  to FLUID
40:  send ('fluid-step-ok',  $n$ ) to FLUID
41:  {STRUCTURE CODE: after each time step}
42:  receive ('fluid-step-ok',  $n$ )
43:  send  $\mathbf{w}^n$  to STRUCTURE
44:  send ('struct-step-ok',  $n$ )
45: end for

```

2.1 Notes on the Fluid-Structure Interaction (FSI) Algorithm

- \mathbf{w}^n is the fluid state (ρ, \mathbf{v}, p) at time t^n , \mathbf{u}^n is the structure state (displacements) at time t^n , $\dot{\mathbf{u}}^n$ are the structure velocities at time t^n , \mathbf{X}^n is the fluid mesh node position at time t^n , n_{step} is the number of time steps in the simulation, n_{nwt} is the number of Newton loops in the nonlinear problem, n_{stage} is the number of stages in the coupling scheme.
- Three codes (**FLUID**, **MESH-MOVE**, and **STRUCTURE**) are running simultaneously. For simplicity, the basic algorithm can be thought as if there were no ‘concurrency’ between the codes, i.e. at a given time only one of them is running. This can be controlled using ‘semaphores’ but here this is done using MPI ‘synchronization messages’. Each synchronization message is of the form

send/receive (**<tag>**, **<index>**) message from **<CODE>**

for instance,

receive (‘struct-step-ok’, $n - 1$) message from **STRUCTURE**

This is an algorithmic form of representing the synchronization messages used under the MPI library. It is assumed that the send/receives between programs are well balanced. Failing in receiving a given message throws an exception.

- $\mathbf{u}^{n+1} = \text{CSD}(\mathbf{u}^n, \mathbf{w}^n, \mathbf{w}^{n+1})$, is the operator inside the *Computational Structure Dynamics* code **STRUCTURE** that advances the structure state \mathbf{u} from step n to $n + 1$ using the fluid forces.
- $\mathbf{w}^{n+1} = \text{CFD}(\mathbf{w}^n, \mathbf{X}^{n+1}, \mathbf{X}^n)$ is the operator inside the *Computational Fluid Dynamics* code **FLUID** that advances the fluid state using the fluid mesh displacements $\mathbf{X}^n, \mathbf{X}^{n+1}$.
- $\mathbf{X} = \text{CMD}(\mathbf{u})$ is the operator inside the *Computational Mesh Dynamics* code **MESH-MOVE** that computes fluid mesh node displacements \mathbf{X} from the structure skin displacements \mathbf{u} . For some simple examples a very simple CMD model based on ‘spines’ is used. In general a third independent code **MESH-MOVE** is used for relocating the mesh, using an elastic or pseudo-elastic model (López et al., 2006). This mesh relocation algorithm is similar to those used in Tezduyar et al. (1992); Johnson and Tezduyar (1994).
- The most external loop is over time steps. Internal to it there is the ‘stage loop’. ‘Weak coupling’ is achieved if only one stage is performed (i.e. $n_{\text{stage}} = 1$). In each stage the fluid is first advanced using the previously computed structure state u^n and the current estimate value $u^{n+1,i}$. In this way, a new estimate for the fluid state $w^{n+1,i+1}$ is computed. Next the structure is updated using the forces of the fluid from states w^n and $w^{n+1,i+1}$. At the first stage, the state $u^{n+1,0}$ is predicted using a second or higher order approximation (see equation (1)). Inside the *stage loop* there are Newton loops for each code to solve the non-linearities.
- The general form of the predictor for the structure state was taken from Piperno and Farhat (2001) and can be written as

$$\mathbf{u}^{(n+1)P} = \alpha_0 \Delta t \dot{\mathbf{u}}^n + \alpha_1 \Delta t (\dot{\mathbf{u}}^n - \dot{\mathbf{u}}^{n-1}) \quad (1)$$

It is at least first order accurate. For $\alpha_1 = 0$ and $\alpha_0 = 1$ it is second order accurate in time and for $\alpha_0 = 1, \alpha_1 = 1/2$ it is fourth order accurate. Note that, if the trapezoidal rule with $\alpha = 0.5$ is used for both the structure and the fluid and the predictor is chosen with at least second order precision, then the whole algorithm is second order, even if only one stage is performed regardless the number of stages used.

2.2 Multi-physics code paradigm

Beyond the physical and engineering importance, this problem is interesting from the computational point of view as a paradigm of multiphysics code implementation that reuses preexistent fluid and elastic solvers. The partitioned algorithm is implemented in the **PETSc-FEM** code (www.cimec.org.ar/petscfem) which is a parallel multi-physics finite element program based on the *Message Passing Interface (MPI)* and the *Portable Extensible Toolkit for Scientific Computations (PETSc)*. Two instances of the PETSc-FEM code simulate each subproblem and communicate interface pressure and displacements via Standard C *FIFO* files or ‘pipes’. The key point in the implementation of this partitioned scheme is the data exchange and synchronization between both parallel processes. These tasks are made in a small external C++ routine. More details are available in previous works (Battaglia et al., 2006, 2005).

3 FRAME INVARIANCE OF ALE FORMULATION

A key point in fluid-structure interaction problems is the use of the “*Arbitrary Lagrangian Eulerian formulation*” (ALE), which allows the use of moving meshes. As the ALE convective terms affect the advective terms, some modifications are needed to the standard stabilization terms in order to get the correct amount of stabilization. Also boundary conditions at walls (slip or non-slip) and absorbing boundary conditions must be modified when ALE is used.

Since all these features are well known in a non-ALE frame, one can simply compute the appropriate objects (stabilization operators, absorbing boundary condition projections) in the non-ALE frame and transform them to the ALE frame. If this is properly done, then a formulation that is “*invariant*” to ALE transformations is obtained. In this article a ALE formulation that is invariant under Galilean transformations, which are the simplest of all the possible ALE transformations, is presented. The key points are the computation of the stabilization terms and wall boundary conditions. A different approach to Galilean invariance has been proposed by Scovazzi (2005a,b).

3.1 Transformations laws for Jacobians

Consider the governing equations for compressible flow of an ideal gas in a Galilean frame $S = (\mathbf{x}, t)$ in the form of an advective-diffusive system

$$\frac{\partial \mathbf{U}_c}{\partial t} + \frac{\partial \mathcal{F}_{c,x}}{\partial x} = 0 \quad (2)$$

where

$$\begin{aligned} \mathbf{U}_c &= [\rho, \rho u, \rho e]^T, && \text{(conservative variables)} \\ \mathcal{F}_{c,x} &= [\rho u, \rho u^2 + p, \rho h]^T, && \text{(convective flux along } x \text{ direction)} \\ e &= p[\rho/(\gamma - 1)]^{-1} + \frac{1}{2}\rho u^2, && \text{(specific total energy)} \\ h &= e + p/\rho, && \text{(specific total enthalpy)} \\ \mathbf{U} &= [\rho, \mathbf{u}, p]^T, && \text{(primitive variables)} \end{aligned} \quad (3)$$

ρ is the density, u velocity, p pressure and $\gamma = C_p/C_v$ is the “*adiabatic index*” of the gas.

Now consider the corresponding system of equations in a second Galilean frame $S' = (\mathbf{x}', t)$ moving with a constant velocity \mathbf{v} with respect to S , i.e. $\mathbf{x}' = \mathbf{x} - \mathbf{v}t$. The fluid states \mathbf{U}' , \mathbf{U}'_c in a given point (\mathbf{x}', t) in the S' frame, are related to the states in the original S frame through

the relations

$$\begin{aligned} \mathbf{U}' &= [\rho', u', p']^T = [\rho, u - v, p]^T, && \text{(for primitive variables),} \\ \mathbf{U}_c' &= [\rho', \rho' u', \rho' e']^T = [\rho, \rho(\mathbf{u} - \mathbf{v}), \rho e - u'v + 1/2 v^2]^T, && \text{(for conservative variables),} \end{aligned} \quad (4)$$

so that the Jacobian of the transformations are

$$\begin{aligned} \frac{\partial \mathbf{U}'}{\partial \mathbf{U}} &= \mathbf{I}, \\ \frac{\partial \mathbf{U}_c'}{\partial \mathbf{U}_c} &= \begin{bmatrix} 1 & 0 & 0 \\ -\mathbf{v} & \mathbf{I} & 0 \\ 1/2 v^2 & -\mathbf{v}^T & 1 \end{bmatrix} = \mathbf{T}(-\mathbf{v}) \end{aligned} \quad (5)$$

It can be easily verified that they are a “group” of transformations, i.e.

$$\begin{aligned} \mathbf{T}(\mathbf{v} + \mathbf{w}) &= \mathbf{T}(\mathbf{v}) \mathbf{T}(\mathbf{w}), \\ \mathbf{T}(-\mathbf{v}) &= \mathbf{T}(\mathbf{v})^{-1} \end{aligned} \quad (6)$$

Now, if $\mathbf{U}(\mathbf{x}, t)$ is a solution of (2,3) then $\mathbf{U}'(\mathbf{x}', t)$ obtained with the transformations (4) must be a solution of the governing equations in frame S' . In order to simplify the algebra, consider the quasi-linear form of the governing equations

$$\mathbf{C} \frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_x \frac{\partial \mathbf{U}}{\partial x} = 0 \quad (7)$$

where

$$\begin{aligned} \mathbf{C} &= \frac{\partial \mathbf{U}_c}{\partial \mathbf{U}}, && \text{(enthalpy Jacobian),} \\ \mathbf{A}_x &= \frac{\partial \mathcal{F}_{c,x}}{\partial \mathbf{U}}. && \text{(advective Jacobian).} \end{aligned} \quad (8)$$

The temporal derivative transforms in the following way,

$$\begin{aligned} \left. \frac{\partial \mathbf{U}}{\partial t} \right|_x &= \left. \frac{\partial \mathbf{U}}{\partial t} \right|_{x'} + \mathbf{v} \frac{\partial \mathbf{U}}{\partial x}, \\ &= \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \left(\left. \frac{\partial \mathbf{U}'}{\partial t} \right|_{x'} + \mathbf{v} \frac{\partial \mathbf{U}'}{\partial x'} \right) = \left. \frac{\partial \mathbf{U}'}{\partial t} \right|_{x'} + \mathbf{v} \frac{\partial \mathbf{U}'}{\partial x'}. \end{aligned} \quad (9)$$

Replacing in (7) the following expression is obtained,

$$\mathbf{C} \left(\left. \frac{\partial \mathbf{U}'}{\partial t} \right|_{x'} + \mathbf{v} \frac{\partial \mathbf{U}'}{\partial x'} \right) + \mathbf{A}_x \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \frac{\partial \mathbf{U}'}{\partial x} = 0, \quad (10)$$

and multiplying by $(\partial \mathbf{U}_c' / \partial \mathbf{U}_c)$ at the left,

$$\begin{aligned} \left(\frac{\partial \mathbf{U}_c'}{\partial \mathbf{U}_c} \frac{\partial \mathbf{U}_c}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right) \left(\left. \frac{\partial \mathbf{U}'}{\partial t} \right|_{x'} + \mathbf{v} \frac{\partial \mathbf{U}'}{\partial x'} \right) + \left(\frac{\partial \mathbf{U}_c'}{\partial \mathbf{U}_c} \mathbf{A}_x \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right) \frac{\partial \mathbf{U}'}{\partial x} &= 0, \\ \mathbf{C}' \frac{\partial \mathbf{U}'}{\partial t} + \mathbf{A}' \frac{\partial \mathbf{U}'}{\partial x} &= 0, \end{aligned} \quad (11)$$

from which the following transformation laws for the Jacobians are deduced,

$$\begin{aligned}
 \mathbf{C}' &= \frac{\partial \mathbf{U}_c'}{\partial \mathbf{U}_c} \mathbf{C} \frac{\partial \mathbf{U}}{\partial \mathbf{U}'}, \\
 \mathbf{C} &= \frac{\partial \mathbf{U}_c}{\partial \mathbf{U}_c'} \mathbf{C}' \frac{\partial \mathbf{U}'}{\partial \mathbf{U}}, \\
 \mathbf{A}'_x &= v_x \mathbf{C}' + \frac{\partial \mathbf{U}_c'}{\partial \mathbf{U}_c} \mathbf{A}_x \frac{\partial \mathbf{U}}{\partial \mathbf{U}'}, \\
 \mathbf{A}_x &= v_x \mathbf{C} + \frac{\partial \mathbf{U}_c}{\partial \mathbf{U}_c'} \mathbf{A}'_x \frac{\partial \mathbf{U}'}{\partial \mathbf{U}}.
 \end{aligned} \tag{12}$$

It can be shown that source and diffusive terms also transform in the appropriate way. We say that \mathbf{C} and \mathbf{A} transform as $\mathbf{U}_c \times \mathbf{U}$, i.e. as \mathbf{U}_c (conservative variables) at left and as \mathbf{U} (primitive variables) at right.

3.2 Definition of ALE invariance

Of course, the discrete equations are not invariant under an arbitrary Galilean transformation, mainly because the importance of the advective terms are relative to the frame of reference. For instance, a fluid which is at rest in frame S does not need stabilization, whereas in a frame S' with relative velocity \mathbf{v} it may have a high Péclet number and then it will need stabilization. However, when using ALE formulations with moving domains, stabilization is based on the velocity of the *fluid relative to the mesh*. With this additional degree of freedom introduced with moving meshes a physical problem can be posed in different Galilean frames and in such a way that the velocity of the *fluid relative to the mesh is the same*. Then the question can be posed of whether discrete stabilized equations give the same solution (after appropriate transformation laws) in these equivalent situations. If the scheme is not invariant then great chances exist that the scheme adds more diffusion in one frame than in other, and then to be unstable or too diffusive. If the discrete formulation pass the test we say that it is “*ALE invariant*”. In this article only invariance under Galilean (i.e. constant velocity) transformations between the systems is considered. As ALE can be applied to more general mesh movements involving, for instance, rotations and accelerations, the same question can be posed for those cases as well.

3.3 Test case. Sudden stop of gas container

Consider the case of a 1D flow in a container, i.e. between two walls at $x = 0, L$. For $t < 0$ the container and the fluid are moving with constant velocity u_0 and at $t = 0$ the container is suddenly stopped. It is assumed that the fluid is in a homogeneous state for $t < 0$, so that the governing equations, along with the initial and boundary conditions are

$$\begin{aligned}
 \frac{\partial \mathbf{U}_c}{\partial t} + \frac{\partial \mathcal{F}_{c,x}}{\partial x} &= \frac{\partial \mathcal{F}_{d,x}}{\partial x}; \quad 0 \leq x \leq L, \\
 u(x = 0, t) &= u(x = L, t) = 0, \\
 \mathbf{U}(x, t = 0) &= [\rho, u, p]_0^T,
 \end{aligned} \tag{13}$$

where $\mathcal{F}_{d,j}$ are the “*diffusive fluxes*” and $u_0 > 0$. Due to the presence of the walls at $t = 0$ the fluid starts to compress at the $x = L$ wall starting eventually a shock (depending on the Mach number $\text{Ma}_0 = u_0/c_0$, $c_0 = \sqrt{\gamma p_0/\rho_0}$). At the same time an expansion fan is formed at the $x = 0$ wall. In this frame S fixed to the container the domain $\Omega = [0, L]$ is fixed and no ALE

terms are needed. But the same problem can be described in a frame S' moving with velocity $v = u_0$, so that the fluid in this frame is initially at rest. The walls move with velocity $-u_0$ with respect to this frame, so that the governing equations are

$$\begin{aligned} \frac{\partial \mathbf{U}'_c}{\partial t} + \frac{\partial \mathcal{F}'_{c,x}}{\partial x'} &= \frac{\partial \mathcal{F}'_{d,x}}{\partial x}; \quad 0 \leq x' + u_0 t \leq L, \\ u(x' = -u_0 t, t) &= u(x = L - u_0 t, t) = -u_0, \\ \mathbf{U}(x, t = 0) &= [\rho, 0, p]_0^T. \end{aligned} \quad (14)$$

Of course, in the continuum both systems of equations are completely equivalent, this can be verified by writing down the equations for one frame and transform to the other using the transformation rules described above. But the numerical stabilization terms can break the invariance of the discrete equations.

3.4 Transformation of the stabilization term

The key point in obtaining an ALE invariant discrete system is to make the stabilization numerical diffusion operator invariant. Recall that the “*Streamline Upwind Petrov Galerkin*” (SUPG, see Hughes and Tezduyar (1984)) stabilization term is obtained by weighting the governing equations with a weight function W that is the sum of the interpolation function and a perturbation function P defined (in the scalar case) as

$$\begin{aligned} W &= N + P, \\ P &\propto \nabla N \cdot \mathbf{a} \end{aligned} \quad (15)$$

where \mathbf{a} is the scalar version of \mathbf{A} which is, for scalar problems, a vector of scalar matrices. Similarly, c will denote the scalar version of \mathbf{C} . The \mathbf{a} factor in the definition of P introduces the bias of the weight function and guarantees that the formulation tends to its *Galerkin* (“*centered*”) form when the advective term is negligible. As P should be nondimensional it can be shown that for scalar problems P should be of the form

$$P = (\tau/c) \nabla N \cdot \mathbf{a}, \quad (16)$$

where τ has dimensions of time (the so called “*intrinsic time*”). Typically

$$\tau = \frac{hc}{|a|} \quad (17)$$

where h is the mesh size. In the S frame, where the domain is fixed, the stabilized equations are

$$\mathbf{C} \frac{\partial \mathbf{U}_j}{\partial t} + \mathbf{A} \frac{\mathbf{U}_{j+1} - \mathbf{U}_j}{2h} = (\mathbf{K}_{\text{num}} + \mathbf{K}) \frac{\mathbf{U}_{j+1} - 2\mathbf{U}_{j-1} + \mathbf{U}_{j-1}}{h^2} \quad (18)$$

where, in the scalar case

$$\mathbf{K}_{\text{num}} = (\tau/c) a^2. \quad (19)$$

The key point of this section is how to extend (17,19) to systems as in the case of the gas-dynamics equations with an ALE formulation. It will be shown that if P is chosen in the form

$$\begin{aligned} P &= \nabla N \cdot \tilde{\mathbf{A}} \tau \mathbf{C}^{-1}, \\ \tilde{\mathbf{A}} &= \mathbf{A} - \mathbf{v}_{\text{mesh}} \mathbf{C}, \end{aligned} \quad (20)$$

then the discrete scheme is ALE invariant, provided that τ transforms as $\mathbf{U} \times \mathbf{U}$, i.e.

$$\tau' = \frac{\partial \mathbf{U}'}{\partial \mathbf{U}} \tau \frac{\partial \mathbf{U}}{\partial \mathbf{U}'}. \quad (21)$$

In (20) $\tilde{\mathbf{A}}$ is the ALE corrected advective Jacobian.

After some algebra it can be shown that the numerical diffusion operator produced by a stabilization term like (20) is

$$\mathbf{K}_{\text{num}} = \tilde{\mathbf{A}} \tau \mathbf{C}^{-1} \tilde{\mathbf{A}}. \quad (22)$$

Now, let us transform the discrete equations (18) to the S' frame. Note that, as in the S frame the mesh is fixed, the equation has not ALE term (i.e. $v_{\text{mesh}} = 0$). To transform the equation, multiply the equations at left by $(\partial \mathbf{U}'_c / \partial \mathbf{U}_c)$ and by $(\partial \mathbf{U} / \partial \mathbf{U}')$ at right, resulting in

$$\begin{aligned} \left(\frac{\partial \mathbf{U}'_c}{\partial \mathbf{U}_c} \mathbf{C} \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right) \left(\frac{\partial \mathbf{U}'}{\partial \mathbf{U}} \frac{\partial \mathbf{U}_j}{\partial t} \right) + \left(\frac{\partial \mathbf{U}'_c}{\partial \mathbf{U}_c} \mathbf{A} \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right) \left(\frac{\partial \mathbf{U}'}{\partial \mathbf{U}} \frac{\mathbf{U}_{j+1} - \mathbf{U}_{j-1}}{2h} \right) = \\ \left[\frac{\partial \mathbf{U}'_c}{\partial \mathbf{U}_c} (\mathbf{K}_{\text{num}} + \mathbf{K}) \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right] \left(\frac{\partial \mathbf{U}'}{\partial \mathbf{U}} \frac{\mathbf{U}_{j+1} - 2\mathbf{U}_j + \mathbf{U}_{j-1}}{h^2} \right) \end{aligned} \quad (23)$$

and using the transformation law for the Jacobians (12),

$$\mathbf{C}' \frac{\partial \mathbf{U}'_j}{\partial t} + (\mathbf{A}' - v \mathbf{C}') \frac{\mathbf{U}'_{j+1} - \mathbf{U}'_{j-1}}{2h} = (\mathbf{K}'_{\text{num}} + \mathbf{K}') \frac{\mathbf{U}'_{j+1} - 2\mathbf{U}'_j + \mathbf{U}'_{j-1}}{h^2} \quad (24)$$

The term added to the advective Jacobian stands for the ALE formulation (since $v_{\text{mesh}} = v$ in frame S'). The expression for \mathbf{K}'_{num} is

$$\begin{aligned} \mathbf{K}'_{\text{num}} &= \frac{\partial \mathbf{U}'_c}{\partial \mathbf{U}_c} \mathbf{A} \tau \mathbf{C} \mathbf{A} \frac{\partial \mathbf{U}}{\partial \mathbf{U}'}, \\ &= \left(\frac{\partial \mathbf{U}'_c}{\partial \mathbf{U}_c} \mathbf{A} \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right) \left(\frac{\partial \mathbf{U}'}{\partial \mathbf{U}} \tau \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right) \left(\frac{\partial \mathbf{U}'}{\partial \mathbf{U}} \mathbf{C}^{-1} \frac{\partial \mathbf{U}_c}{\partial \mathbf{U}'_c} \right) \left(\frac{\partial \mathbf{U}'_c}{\partial \mathbf{U}_c} \mathbf{A} \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right), \\ &= \tilde{\mathbf{A}}' \tau' \mathbf{C}'^{-1} \tilde{\mathbf{A}}', \end{aligned} \quad (25)$$

showing the invariance of the numerical diffusion term. As \mathbf{C} , $\tilde{\mathbf{A}}$ and \mathbf{K} all transform as $\mathbf{U}_c \times \mathbf{U}$, the combinations $\mathbf{C}^{-1} \tilde{\mathbf{A}}$ and $\mathbf{C}^{-1} \mathbf{K}$ transform as $\mathbf{U} \times \mathbf{U}$, so equation (21) is verified if it is computed as a matrix function of $\mathbf{C}^{-1} \tilde{\mathbf{A}}$ and $\mathbf{C}^{-1} \mathbf{K}$. For instance, typical extensions of (17) to systems of equations in the inviscid case are

$$\begin{aligned} \tau &= \frac{h}{\max |\lambda_j|} \mathbf{I}, \quad \lambda_j = \text{eig}(\mathbf{C}^{-1} \tilde{\mathbf{A}}), \quad (\text{based on maximum eigenvalue, in magnitude}) \\ \tau &= h |\mathbf{C}^{-1} \tilde{\mathbf{A}}|^{-1}. \quad (|\cdot| \text{ in matrix sense}) \end{aligned} \quad (26)$$

3.5 Invariant form of wall boundary conditions

Special care must be taken with boundary conditions at walls in order to keep invariance. If the 1D discrete equations for the suddenly stopped container in the S frame are considered, a system of discrete equations of the following form is obtained,

$$\begin{aligned} u_0 &= 0; \\ \mathbf{F}_0(\mathbf{U}_0^{n+1}, \mathbf{U}_j^n, \mathbf{U}_{j+1}^n) &= [0, F_{\text{wall},x=0}, 0]^T, \\ \mathbf{F}_j(\mathbf{U}_j^{n+1}, \mathbf{U}_{j-1}^n, \mathbf{U}_j^n, \mathbf{U}_{j+1}^n) &= 0, \quad (\text{for } j = 1, \dots, N-1), \\ \mathbf{F}_N(\mathbf{U}_N^{n+1}, \mathbf{U}_{N-1}^n, \mathbf{U}_N^n) &= [0, F_{\text{wall},x=L}, 0]^T, \\ u_N &= 0; \end{aligned} \quad (27)$$

where F_j represents the discrete equations for node j that arise from the variational Galerkin and stabilization terms. The $F_{\text{wall},x=0,L}$ terms are the reaction forces exerted by the container walls on the fluid in order to keep the conditions of impenetrability at the walls $u_{0,L} = 0$. They act as “Lagrange multipliers” for the verification of the boundary conditions $u_0 = u_L = 0$.

The equations in the S' frame can be found by multiplying these equations row by row at left by $(\partial U_c' / \partial U_c)$. After some algebra the following equations are obtained

$$\begin{aligned}
 u_0 &= v; \\
 \mathbf{F}'_0(\mathbf{U}_0^{n+1}, \mathbf{U}_j^n, \mathbf{U}_{j+1}^n) &= [0, F_{\text{wall},x=0}, -vF_{\text{wall},x=0}]^T, \\
 \mathbf{F}_j(\mathbf{U}_j^{n+1}, \mathbf{U}_{j-1}^n, \mathbf{U}_j^n, \mathbf{U}_{j+1}^n) &= 0, \quad (\text{for } j = 1, \dots, N - 1), \\
 \mathbf{F}_N(\mathbf{U}_N^{n+1}, \mathbf{U}_{N-1}^n, \mathbf{U}_N^n) &= [0, F_{\text{wall},x=L}, -vF_{\text{wall},x=L}]^T, \\
 u_N &= v.
 \end{aligned} \tag{28}$$

Note that the new terms $vF_{\text{wall},x=0,L}$ are the work done by the container walls on the fluid.

3.6 Numerical example

Consider the 1D problem described in §3.3 for the case of inviscid flow with $\gamma = 1.4$ and $\text{Ma}_0 = 0.5$. The values are made nondimensional by selecting L , ρ_0 and c_0 as reference values for length, density and velocity, so that the nondimensional quantities are $\rho'_0 = 1$, $p'_0 = 1/\gamma$, $u'_0 = 0.5$, and in the following the prime indicating nondimensional quantities is dropped.

The problem was simulated in both frames, with the results being equivalent to machine precision. In the following the energy balance in each frame is discussed in detail.

In figure 1 the pressure in a $x - t$ axis in the frame fixed to the container walls is shown. A shock wave starts at the $x = 1$ wall and propagates backwards with a velocity of approximately 1. At the same time an expansion fan starts at the $x = 0$ wall and propagates forward. The shock wave reflects several times at the walls, with a background state that is not homogeneous due to the expansion fan. The intensity of the shock wave decays and eventually the system reaches a new homogeneous thermodynamic state with null velocity. In figure 2 the pressure profiles as a function of x at several stages can be seen. In figure 3 the energy balance in this frame is shown. As the container is at rest, the walls don't do work on the fluid and then the total energy is conserved.

$$\begin{aligned}
 E_{\text{kin}}(t) + E_{\text{int}}(t) &= E_{\text{kin}}(0) + E_{\text{int}}(0), \quad (S \text{ frame}) \\
 E_{\text{kin}}(t) &= \int_0^L (\frac{1}{2}\rho u^2)_{(x,t)} dx. \\
 E_{\text{int}}(t) &= \int_0^L \frac{1}{\gamma - 1} p(x, t) dx.
 \end{aligned} \tag{29}$$

In the process the shock wave converts the initial kinetic energy in internal energy. The kinetic and the internal energy and the sum of the two are shown in the figure. This last one is constant up to a 1.5% of the mean total energy.

In frame S' the fluid is initially at rest, and at $t = 0$ the container starts moving to the left with velocity $-u_0$. As the container is moving, it does some work on the fluid

$$\dot{W} = (F_{\text{wall},x=L} - F_{\text{wall},x=0})u_0, \tag{30}$$

so that the energy balance is

$$E_{\text{kin}}(t) + E_{\text{int}}(t) = E_{\text{kin}}(0) + E_{\text{int}}(0) + \int_0^t \dot{W}(t') dt'. \tag{31}$$

The energy balance is shown in figure 4. The internal energy has the same increment as in frame S , but now the kinetic energy also has a positive increment. The total energy of the fluid then increases, which is balanced with the work done by the container on the fluid. The error in the energy balance is the same as before, since the results are completely equivalent (up to machine precision).

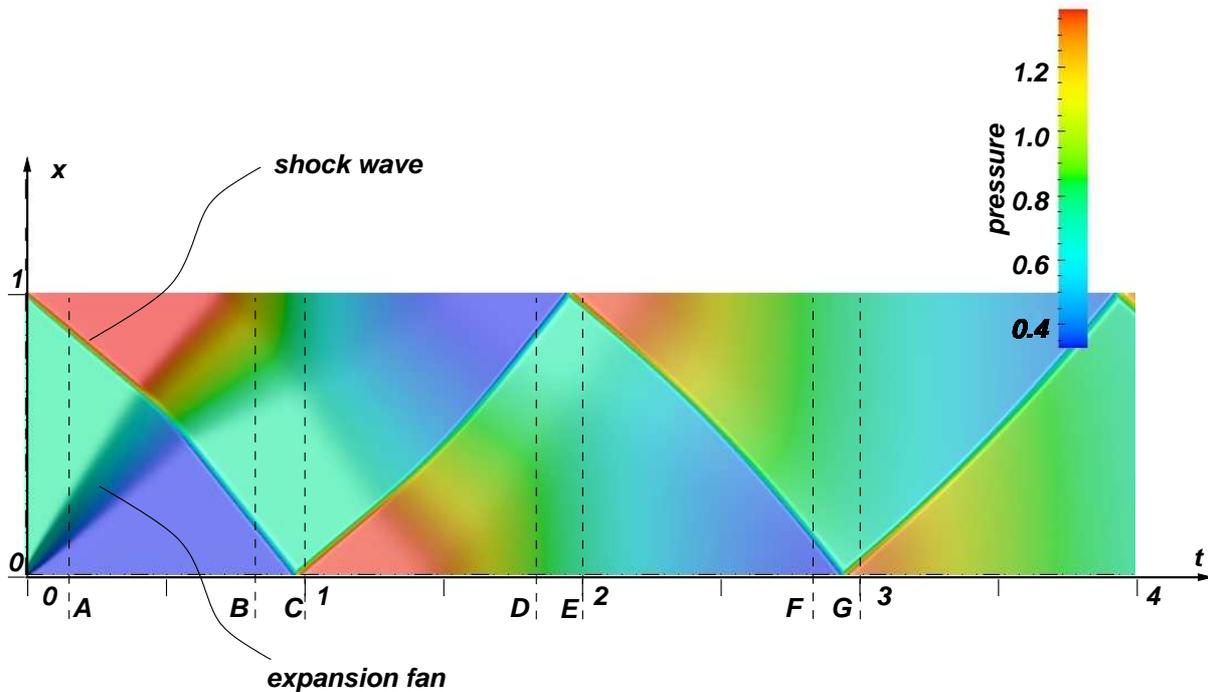


Figure 1: Suddenly stopped container test case. Color map of pressure in x, t plane.

4 DYNAMICALLY VARYING BOUNDARY CONDITIONS

When dealing with fluid-structure interaction problems for bodies immersed in an otherwise infinite fluid it is common to have artificial boundaries with a number of incoming/outgoing characteristics varying spatially and temporally. Absorbing boundary conditions that can handle this situations are discussed in this section.

4.1 Varying boundary conditions in external aerodynamics

During flow computation the number of incoming characteristics n_+ may change. This can occur due to the flow changing regime (i.e. from subsonic to supersonic) or due to the flow changing sense (flow reversal). A typical case is the external flow around an aerodynamic body. Consider first a steady subsonic flow. The flow is normally subsonic at the whole infinite boundary, even if some supersonic pockets can develop at transonic speeds. Then the only two possible regimes are subsonic inlet ($n_+ = n_d + 1$, n_d is the spatial dimension) and subsonic outlet ($n_+ = 1$). Whether the boundary is inlet or outlet can be determined by looking at the projection of the unperturbed flow velocity \mathbf{u}_∞ with the local normal $\hat{\mathbf{n}}$. For the steady supersonic case the situation is very different. A bow shock develops in front of the body and forms a subsonic region which propagates downstream. Far downstream the envelope of the subsonic region approaches a cone with an aperture angle equal to the Mach angle for the non-perturbed flow. At the boundary there is now a supersonic inlet region, and on the outlet region

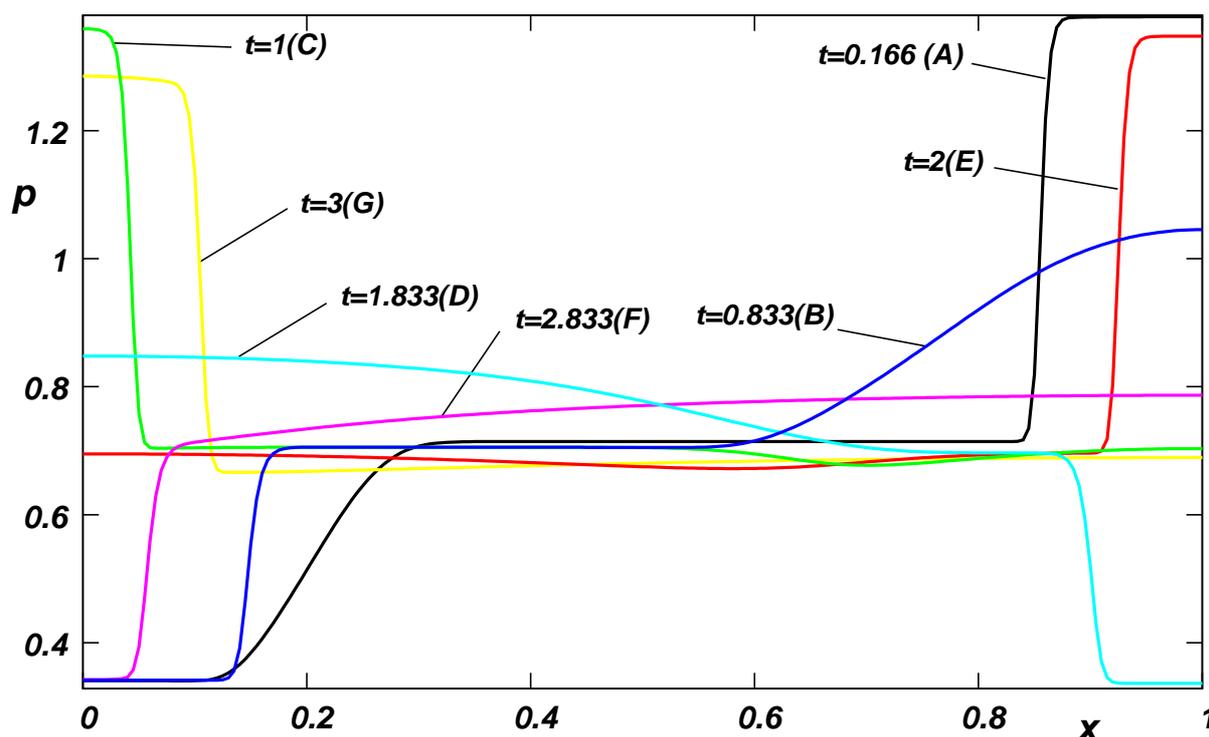


Figure 2: Suddenly stopped container test case. Pressure at several instants.

there are both subsonic and supersonic parts. The point where the flow at outlet changes from subsonic to supersonic may be estimated from the Mach angle, but it may very inaccurate if the boundary is close to the body. Having a boundary condition that can automatically adapt to the whole possibilities can be of great help in such a case. Now, consider the unsteady case, for instance a body accelerating slowly from subsonic to supersonic speeds. The inlet part will change at some point from subsonic to supersonic. At outlet, some parts will change also from subsonic to supersonic, and the point of separation between both will be changing position, following approximately the instantaneous Mach angle. A complete methodology for developing absorbing boundary conditions that automatically switch between the different regimes has been presented in a previous work (Storti et al., 2005).

5 AERODYNAMICS OF FALLING OBJECTS

For instance, one interesting case is the aerodynamics of a falling body (Mittal and Tezduyar, 1994; Field et al., 1997; Belmonte et al., 1998; Huang, 2000, 2001, 2002). Consider, for simplicity, a two dimensional case of an homogeneous ellipse in free fall. As the body accelerates, the pitching moments tend to increase the angle of attack until it stalls (A), and then the body starts to fall towards its other end accelerating while its main axis aligns with gravity (B). As the body accelerates the pitching moment grows until it eventually stalls again (c), and so on... This kind of falling mechanism is typical of slender bodies with relatively small moment of inertia like a sheet of paper and is called “flutter”. However, depending of several parameters, but mainly depending of the moment of inertia of the body, if it has a large angular moment at (B) then it may happen that it rolls on itself, keeping always the same sense of rotation. This kind of falling mechanism is called tumbling and is characteristic of less slender and more massive objects. For massive objects (like a ballistic projectile, for instance) tumbling may convert a large amount of potential energy in the form of rotation, causing the object to rotate at very

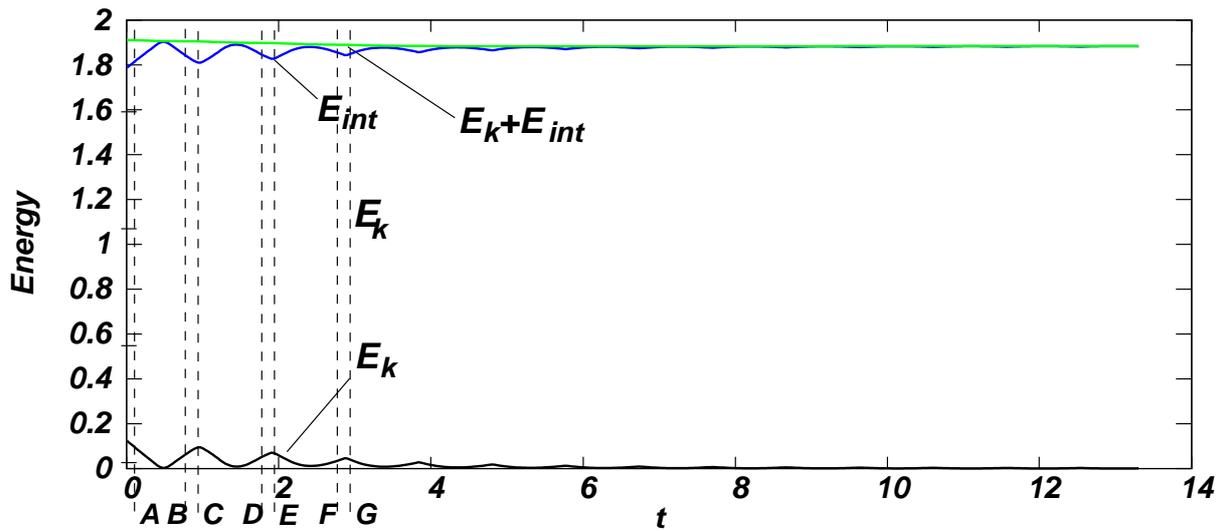


Figure 3: Suddenly stopped container test case. Energy balance at frame S . (Fixed with respect to the container)

large speeds.

As the body falls it accelerates and can reach supersonic speeds. This depends on the density of the body relative to the surrounding atmosphere and its dimensions and shape. As the weight of the body goes with $\propto L^3$, being L the characteristic length while the drag force goes with $\propto L^2$, larger bodies tend to reach larger limit speeds and eventually reach supersonic regime.

5.1 Numerical example. Ellipse falling at supersonic speed.

As an example consider the fall of an ellipse with the following physical data

- $a = 1, b = 0.6$ (major and minor semi-axes, eccentricity $e = \sqrt{1 - b^2/a^2} = 0.8$),
- $m = 1$, (mass),
- $w = 2.5$, (weight of body),
- $r = 1$, (Radius of inertia),
- c.m. = $(-0.15, 0.0)$, (center of mass),
- $\rho_a = 1$, (atmosphere density),
- $p = 1$, (atmosphere pressure),
- $\gamma = 1.4$, (gas adiabatic index $\gamma = C_p/C_v$),
- $R_{ext} = 10$, (Radius of the fictitious boundary),
- $\mathbf{u}_{ini} = [0, 0, 1.39, 0, 1.3, 0]$, (ellipse initial position and velocity $[x, y, \alpha, u, v, \dot{\alpha}]$),

A coarse estimation of the limit speed v can be obtained balancing the vertical forces on the body, i.e. the drag on the body (F_{aero}), the weight and the hydrostatic flotation

$$F_{aero} + W + F_{float} = C_D \rho_a v^2 A - \rho_s g V + \rho_a g V \tag{32}$$

where $V = \pi ab$ is the volume of the body (the area in 2D) and $A = 2b$ the area of the section facing the fluid (length in 2D). $C_D = 0.2$ is an estimation for the drag coefficient of the body and $\rho_s = m/V, \rho_a$ the densities of solid and atmosphere respectively. For the data above this estimation gives a limit speed of $v = 2.8$ approximately. As the speed of sound of the atmosphere is $c = \sqrt{\gamma p/\rho_a} = 1.18$, so that it is expected that the body will reach supersonic speeds. Of course, if the body does reach supersonic speed, then the drag coefficient will be higher and probably the average speed will be lower than that one estimated above.

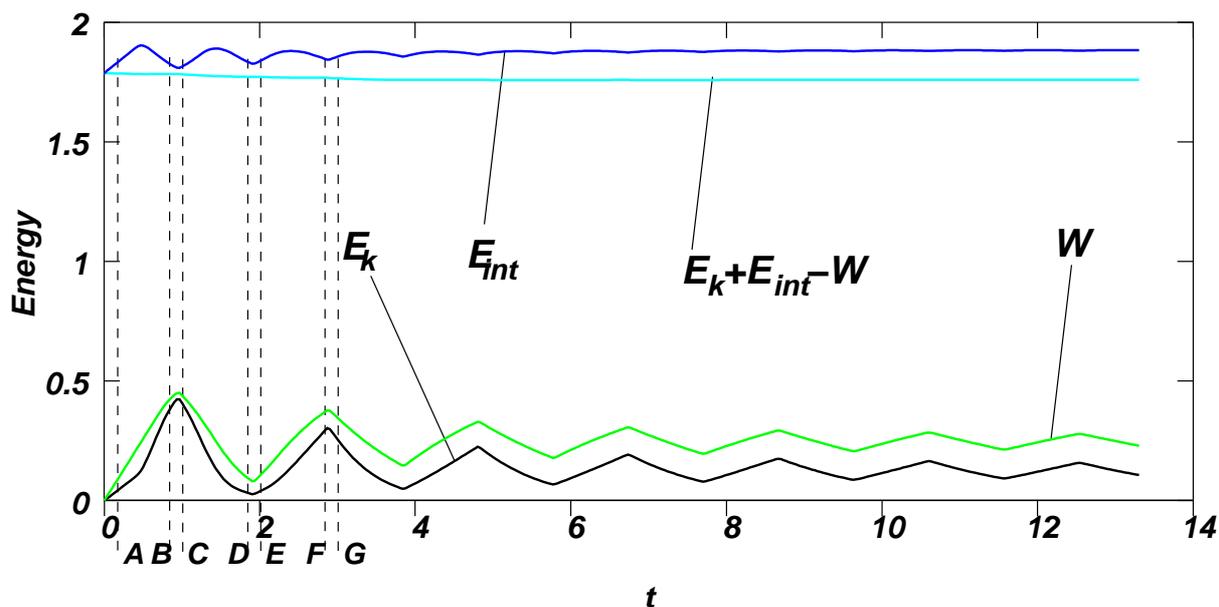


Figure 4: Suddenly stopped container test case. Energy balance at frame S' . (Moving with respect to the container)

The initial conditions are the ellipse starting at velocity $(0, -1.39)$, and an angle of its major axis of 80° with respect with the vertical, the fluid is initially at rest. The computed trajectory until $t = 50$ time units is shown in figure 6. The computed trajectory is shown in a reference system falling at velocity $v = (-0.5, 0.50)$ (this is done in order to reduce the horizontal and vertical span of the plot). In figures 7 color maps of Mach number at four instants are shown in the non inertial frame fixed to the body. The instants are marked as A, B, C and identified in the trajectory. Note that as the ellipse rotates, each part of the boundary experiments all kind of regimes and the absorbing boundary condition cope with all of them. Note also that the artificial boundary is located very near to the body, the radius of the external circle is 3.25 times the major semi-axis of the ellipse (in the case simulated with the minor external radius, i.e. $R_{\text{ext}} = 5$).

In Figure 8 the velocities of the ellipse are shown in order to evaluate the absorption of dynamic boundary conditions when waves reach boundaries as the ellipse falls and tumble/flutter when the fictitious boundary (exterior circle) is located at $R_{\text{ext}} = 5\text{m}$ and $R_{\text{ext}} = 10\text{m}$ and the size of finite elements remain constant.

In this example the structure code is a simple integrator of the 2D rigid body equations. A numerical example with full integration of the continuum equations for the structure is presented in a companion paper at this same conference (Paz et al., 2006).

6 CONCLUSIONS

A partitioned scheme for fluid-structure interaction was presented. As in other partitioned schemes each code computes its new state using predicted values for the fields of the other code, but an outer loop (the stage loop) iterates in order to verify that the thole scheme converges to a monolithic one. For one only one stage the scheme is equivalent to most partitioned schemes. In addition, the stabilization operators and wall boundary conditions are modified in order to have an ALE invariant spatial discretization scheme.

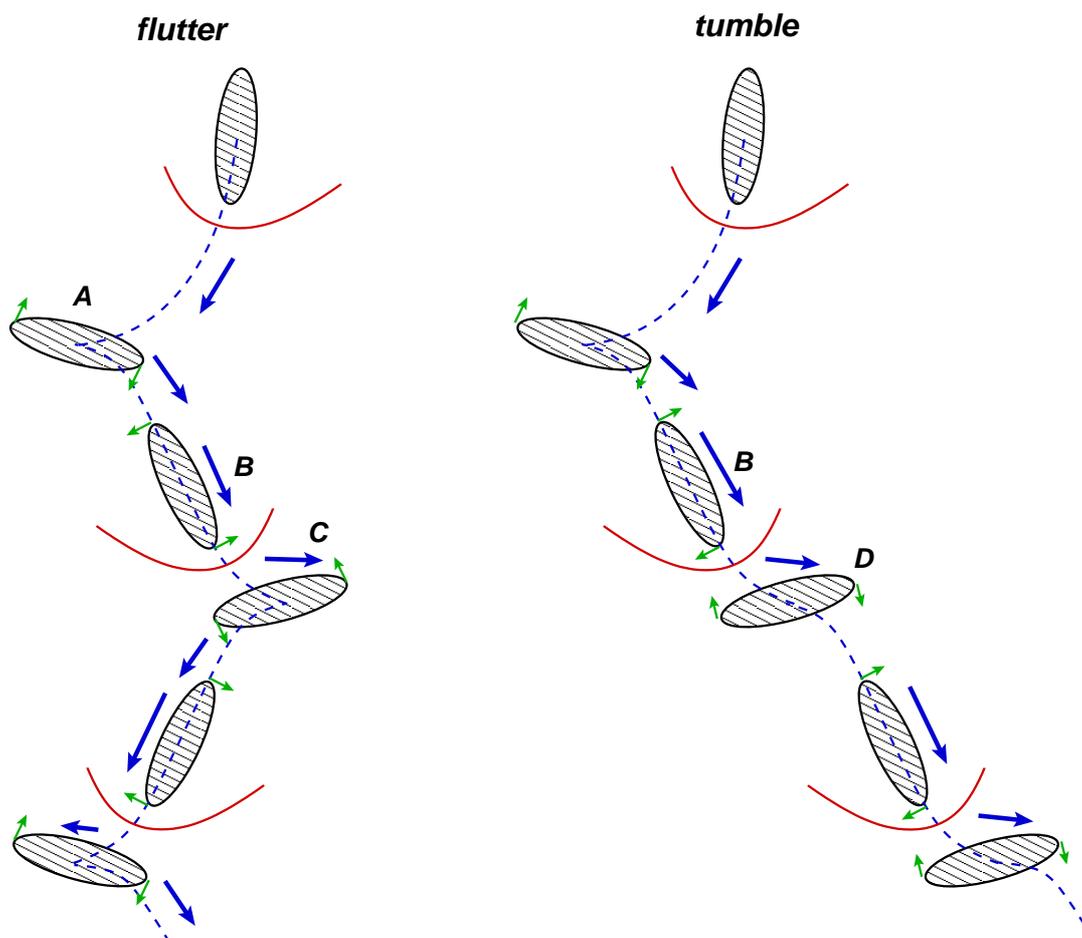


Figure 5: Falling ellipse

6.1 Acknowledgment

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, grants PIP 02552/00, PIP 5271/05), Universidad Nacional del Litoral (UNL, Argentina, grant CAI+D 2005-10-64) and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT 12-14573/2003, PME 209/2003). Authors made extensive use of freely distributed software as GNU/Linux OS, MPI, PETSc, GCC compilers, Octave, Open-DX, Perl, among many others. Also, many ideas from these packages have been inspirational to them.

REFERENCES

- Battaglia L., D'Elía J., Storti M., and Nigro N. Free-surface flows in a multi-physics programming paradigm. In L. A., editor, *Mecánica Computacional*, volume 24, pages 105–116. AMCA, 2005.
- Battaglia L., D'Elía J., Storti M., and Nigro N. Numerical simulation of transient free surface flows. *ASME J Appl Mech*, 2006. To appear.
- Belmonte A., Eisenberg H., and Moses E. From flutter to tumble: Inertial drag and froude similarity in falling paper. *Phys. Rev. Lett.*, 81(2):345–348, 1998. doi:10.1103/PhysRevLett.81.345.
- Dettmer W. and Peric D. A computational framework for fluid-rigid body interaction: Finite

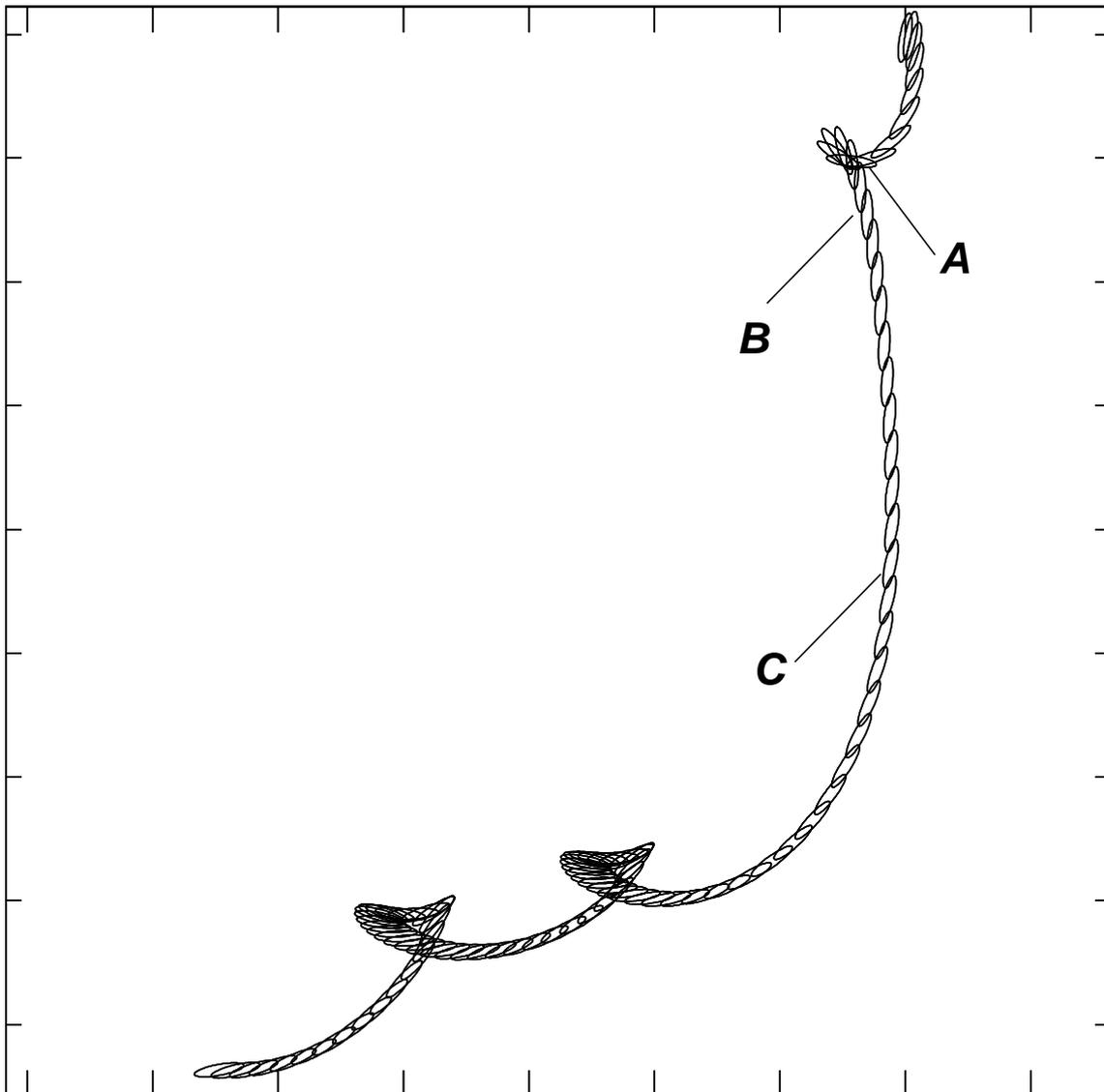


Figure 6: Computed trajectory of falling ellipse

element formulation and applications. *Computer Methods in Applied Mechanics and Engineering*, 195:1633–1666, 2006.

Dowell E., Crawley E., Curtiss H., Peters D., Scanlan R., and Sisto F. *A Modern Course in Aeroelasticity*. Kluwer Academic Publishers, Dordrecht, 1995.

Felippa C.A., Park K.C., and Farhat C. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3247–3270, 2001.

Field S., Klaus M., Moore M., , and Nori F. Chaotic dynamics of falling disks. *Nature*, 388(6639):252–254, 1997.

Gnesin V. and Rzadkowski R. A coupled fluid structure analysis for 3-d inviscid flutter of IV standard configuration. *Jornal of Sound and Vibration.*, 49:349–369, 2005.

Huang J. Trajectory of a moving curveball in viscid flow. In *Proceedings of the Third International Conference: Dynamical Systems and Differential Equations*, pages 191–198. 2000.

Huang J. *Moving Boundaries VI*, chapter Moving Coordinates Methods and Applications to the Oscillations of a Falling Slender Body, pages 73–82. WIT Press, 2001.

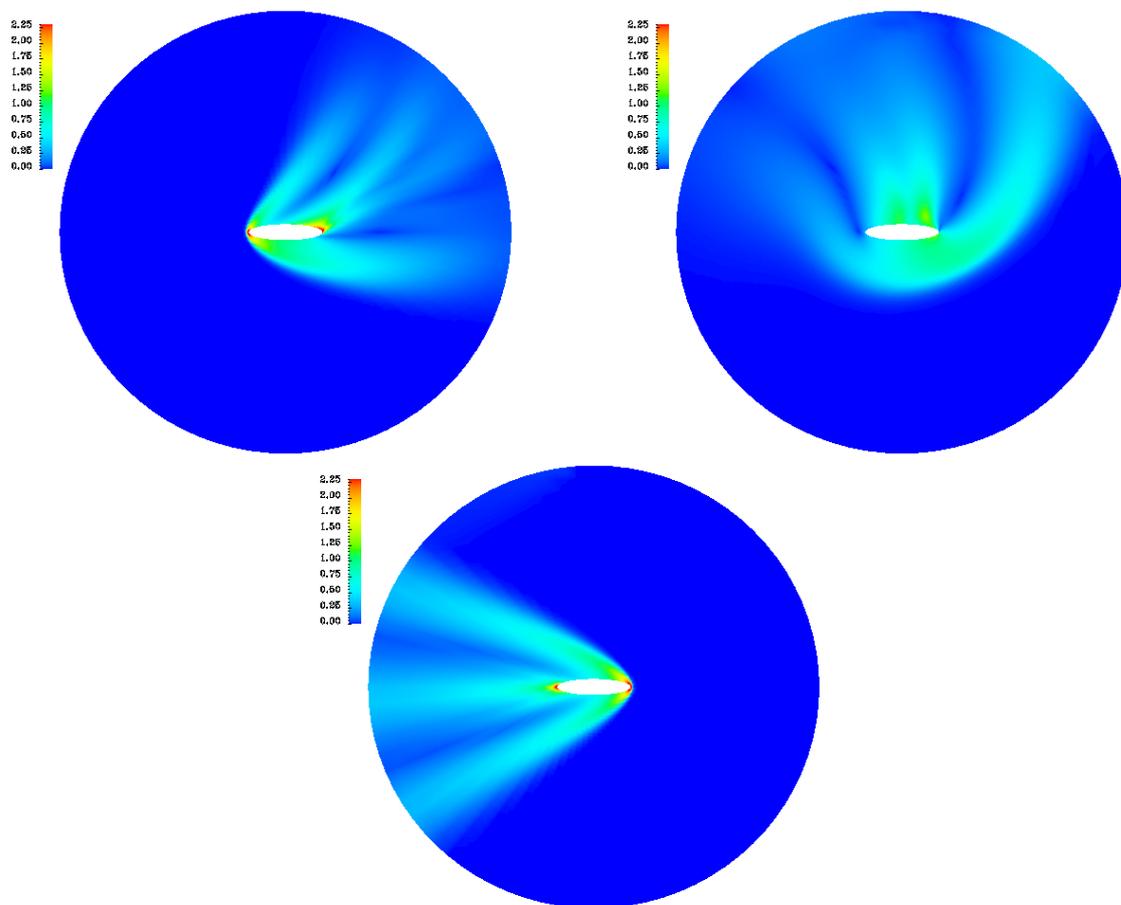


Figure 7: Ellipse falling at supersonic speeds. Color maps of Mach number. Top left: station A ($t = 2.75$), top right: station B ($t = 5$), bottom: station C ($t = 11.25$). Stations in the trajectory refer to figure 6. Results are shown in a non-inertial frame attached to the ellipse.

- Huang J. *Advances in Fluid Mechanics IV*, chapter Aerodynamics of a Moving Curveball in Newtonian Flow, pages 597–608. WIT Press, 2002.
- Hughes T. and Tezduyar T. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45:217–284, 1984.
- Johnson A. and Tezduyar T. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994.
- Kalro V. and Tezduyar T. A parallel 3d computational method for fluid-structure interactions in parachute systems. *Computer Methods in Applied Mechanics and Engineering*, 190:321–332, 2000.
- Lefrancois E. Numerical validation of a stability model for a flexible over-expanded rocket nozzle. *International Journal for Numerical Methods in Fluids.*, 49:349–369, 2005.
- López E., Nigro N.M., Storti M.A., and Toth J. A minimal element distortion strategy for computational mesh dynamics. *International Journal for Numerical Methods in Engineering.*, 2006. To appear.
- Mittal S. and Tezduyar T. Massively parallel finite element computation of incompressible flows involving fluid-body interactions. *Computer Methods in Applied Mechanics and Engineering*, 112:253–282, 1994.

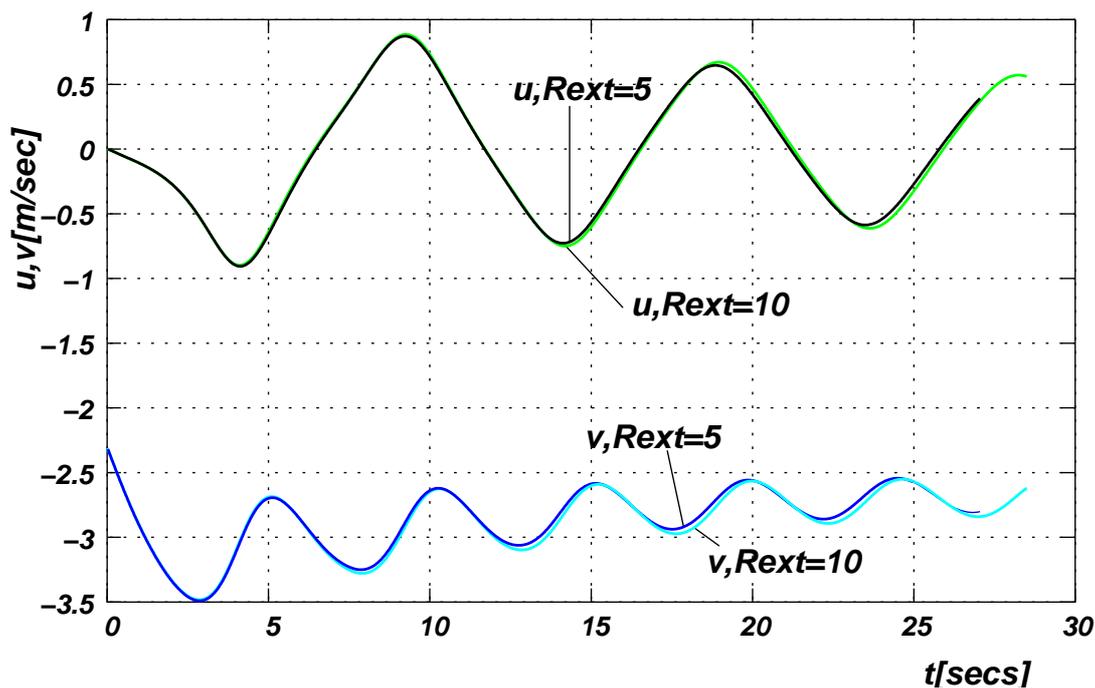


Figure 8: Ellipse velocities for different external radius

- Mittal S. and Tezduyar T. Parallel finite element simulation of 3d incompressible flows-fluid-structure interactions. *International Journal for Numerical Methods in Fluids*, 21:933–953, 1995.
- Park K.C. and Felippa C.A. A variational principle for the formulation of partitioned structural systems. *International Journal for Numerical Methods in Engineering*, 47:395–418, 2000.
- Paz R., Dalcín L., Storti M., and Nigro N. Flow-induced vibration of elastic bodies in supersonic regime via fixed point iteration algorithm, 2006. To be presented at XV Congreso sobre Métodos Numéricos y sus Aplicaciones, Santa Fe, Argentina, November 7-10.
- Piperno R. and Farhat C. Partitioned procedures for the transient solution of coupled aeroelastic problems. Part II: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190:3147–3170, 2001.
- Scovazzi G. A discourse on galilean invariance, supg stabilization, and the variational multiscale framework. Technical report, Sandia National Labs., 2005a. SAND2005-7746. http://www.cs.sandia.gov/~gscovaz/publications/SUPG_hydroSAND_REPORT.pdf.
- Scovazzi G. Stabilized shock hydrodynamics: I. a lagrangian method. Technical report, Sandia National Labs., 2005b. SAND2005-7563J. http://www.cs.sandia.gov/~gscovaz/publications/SUPG_hydroSAND_REPORT.pdf.
- Stein K., Tezduyar T., and Benney R. Computational methods for modeling parachute systems. *Computing in Science & Engineering*, 5:39–46, 2003.
- Storti M., Nigro N., Paz R., and Dalcín L. Dynamic boundary conditions in fluid mechanics. In A. Larregey, editor, *Mecánica Computacional Vol. XXIV*, pages 2573–2596. 2005.
- Tezduyar T., Behr M., Mittal S., and Johnson A. Computation of unsteady incompressible flows with the stabilized finite element methods—space-time formulations, iterative strategies and massively parallel implementations. In N. ASME, editor, *New Methods in Transient Analysis*, volume 246, pages 7–24. 1992.

- Tezduyar T. and Osawa Y. Fluid-structure interactions of a parachute crossing the far wake of an aircraft. *Computer Methods in Applied Mechanics and Engineering*, 191:717–726, 2001.
- Tezduyar T., Sathe S., Keedy R., and Stein K. Space-time techniques for finite element computation of flows with moving boundaries and interfaces. In *Proceedings of the III International Congress on Numerical Methods in Engineering and Applied Sciences. Monterrey, Mexico, CD-ROM*. 2004.
- Tezduyar T., Sathe S., Keedy R., and Stein K. Space-time finite element techniques for computation of fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 195:2002–2027, 2006a.
- Tezduyar T., Sathe S., and Stein K. Solution techniques for the fully-discretized equations in computation of fluid-structure interactions with the space–time formulations. *Computer Methods in Applied Mechanics and Engineering*, 195:5743–5753, 2006b.