Dynamic boundary conditions in fluid mechanics

by Mario Storti, Norberto Nigro, Lisandro Dalcín and Rodrigo Paz

Centro Internacional de Métodos Numéricos
en Ingeniería - CIMEC
INTEC, (CONICET-UNL), Santa Fe, Argentina
<{mstorti,nnigro,dalcinl, rodrigop}@intec.unl.edu.ar>
http://www.cimec.org.ar/mstorti

(document-version "dynabc2-0.0.5") (document-date "2005/11/13 20:25:50 UTC")

Boundary conditions for advective diffusive systems

Well known theory and practice for advective systems say that at a boundary *the number of Dirichlet conditions should be equal to the number of incoming characteristics*.

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathcal{F}_{c,j}(\mathbf{U})}{\partial x_j} = 0$$

$$A_{c,j} = \frac{\partial \mathcal{F}_{c,j}(\mathbf{U})}{\partial \mathbf{U}}, \qquad \textbf{advective Jacobian}$$

Nbr. of incoming characteristics $= \mathrm{sum}(\mathrm{eig}(\mathbf{A} \cdot \hat{\mathbf{n}}) < 0)$

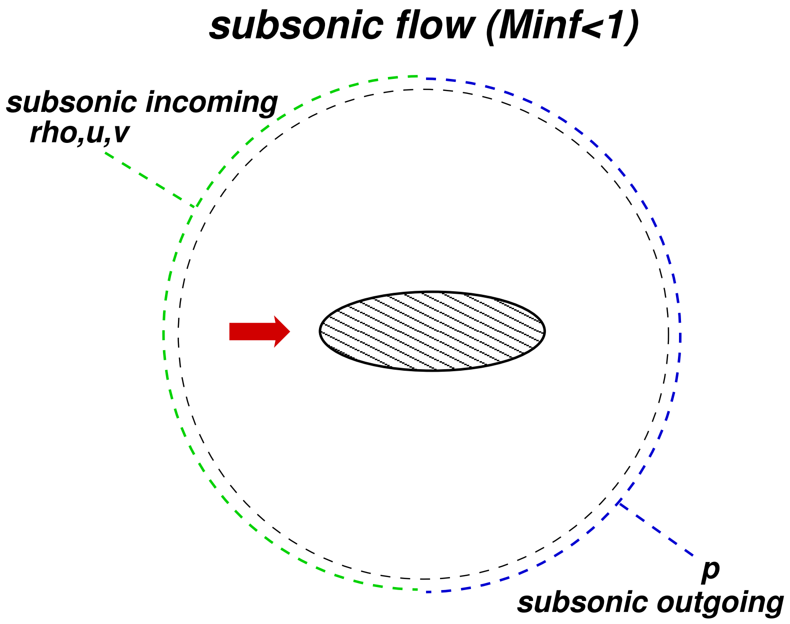
$\hat{\mathrm{n}}$ **is the exterior normal.**

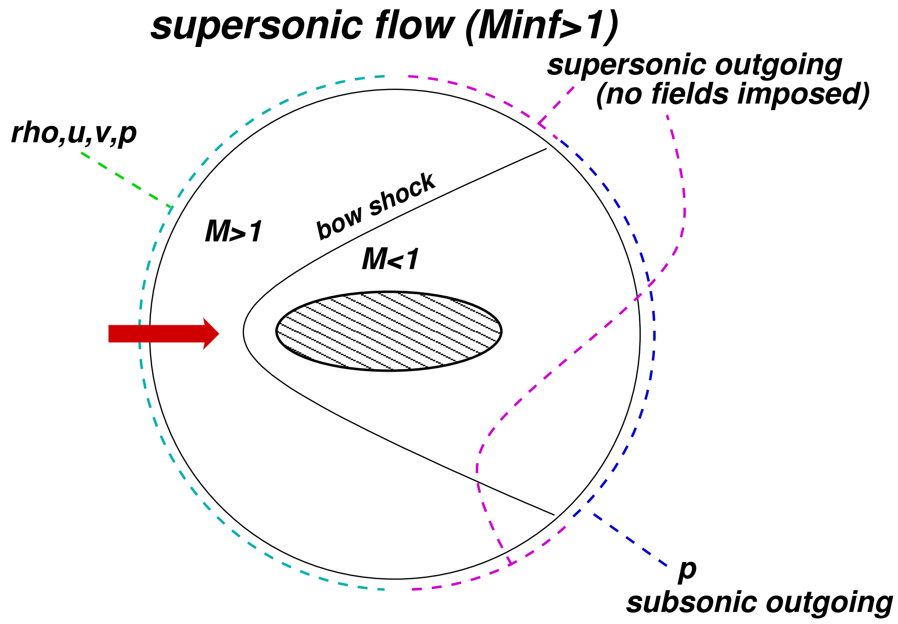
Adding extra Dirichlet conditions leads to spurious shocks, and lack of a Dirichlet conditions leads to instability.

Boundary conditions for advective diffusive systems (cont.)

For simple scalar advection problems the Jacobian is the transport velocity. The rule is then to check the projection of velocity onto the exterior normal.

For more complex flows (i.e. with *non diagonalizable Jacobians*, as gas dynamics or shallow water eqs.) the number of incoming characteristics may be approx. predicted from the flow conditions.





Absorbing boundary conditions

However, this kind of conditions are, generally, *reflective*. First order absorbing boundary conditions may be constructed by imposing exactly the components along the incoming characteristics.

$$\mathbf{\Pi}^{-}(\mathbf{U}_{\mathrm{ref}})\,(\mathbf{U} - \mathbf{U}_{\mathrm{ref}}) = 0.$$

$\mathbf{\Pi}^{-}$ is the projection operator onto incoming characteristics. It can be obtained straightforwardly from the projected Jacobian.

This assumes linearization of the equations around a state $\mathrm{U}_{\mathrm{ref}}$. For linear problems $\mathbf{A}_{c,j}$ do not depend on U, and then neither the projection operator, so that absorbing boundary conditions coefficients are constant.

Absorbing boundary conditions (cont.)

For non-linear problems the Jacobian and projection operator may vary and then the above mentioned b.c.'s are not fully absorbing.

In some cases the concept of characteristic component may be extended to the non-linear case: the *"Riemann invariants"*. Fully absorbing boundary conditions could be written in terms of the invariants:

$$w_j = w_{\mathrm{ref},j}, \quad \text{if } w_j \text{ is an incoming R.I.}$$

- **R.I. are computed analytically. There are no automatic (numerical) techniques to compute them. (They amount to compute an integral in phase space *along a specific path*).**
- **R.I. are known for shallow water, channel flow (for rectangular and triangular channel shape). For gas dynamics the well known R.I. in fact are invariants only under isentropic conditions (i.e. not truly invariant).**

Absorbing boundary conditions (cont.)

Search for an absorbing boundary condition that

- **should be fully absorbent in non-linear conditions, and**
- **can be computed numerically (no need of analytic expressions like R.I.)**

Solution: ***Use last state as reference state, ULSAR*.**

$\mathbf{U}_{\mathrm{ref}} = \mathbf{U}^n$, $n =$ **time step number.**

$$\mathbf{\Pi}^- (\mathbf{U}^n) \, (\mathbf{U}^{n+1} - \mathbf{U}^n) = 0.$$

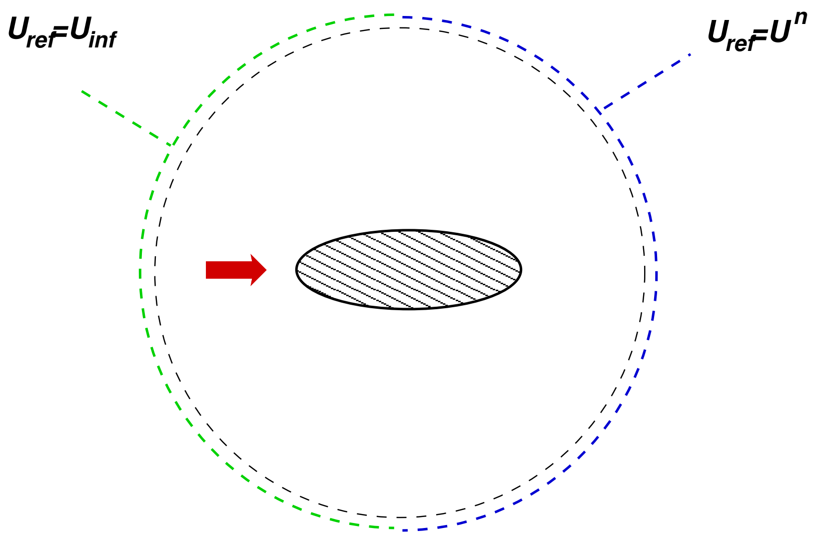
As $\mathbf{U}^{n+1} - \mathbf{U}^n$ is usually small, linearization is valid.

Absorbing boundary conditions (cont.)

Disadvantage: **Flow conditions are only determined from the initial state!! No external information comes from the outside.**

Solution: use a combination of linear/R.I. b.c.'s on incoming boundaries and use fully non-linear a.b.c's with previous state as reference state at the outlet.

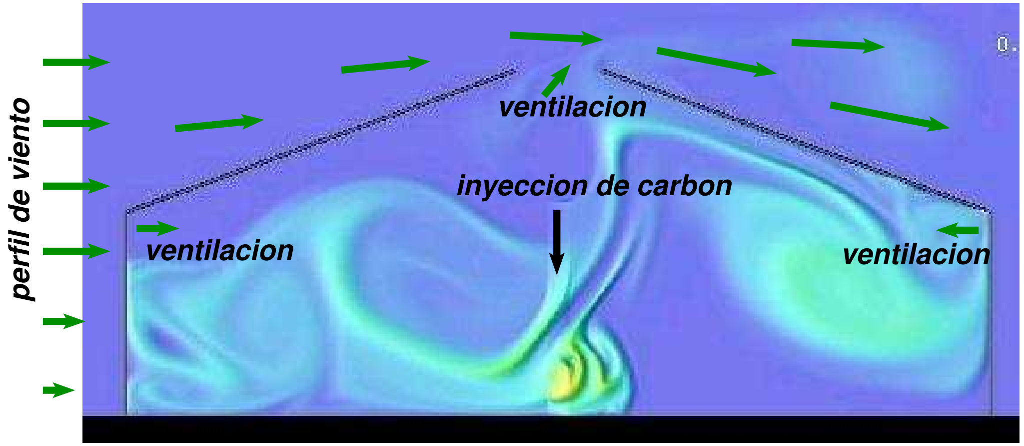
subsonic/supersonic flow



Dynamic boundary conditions

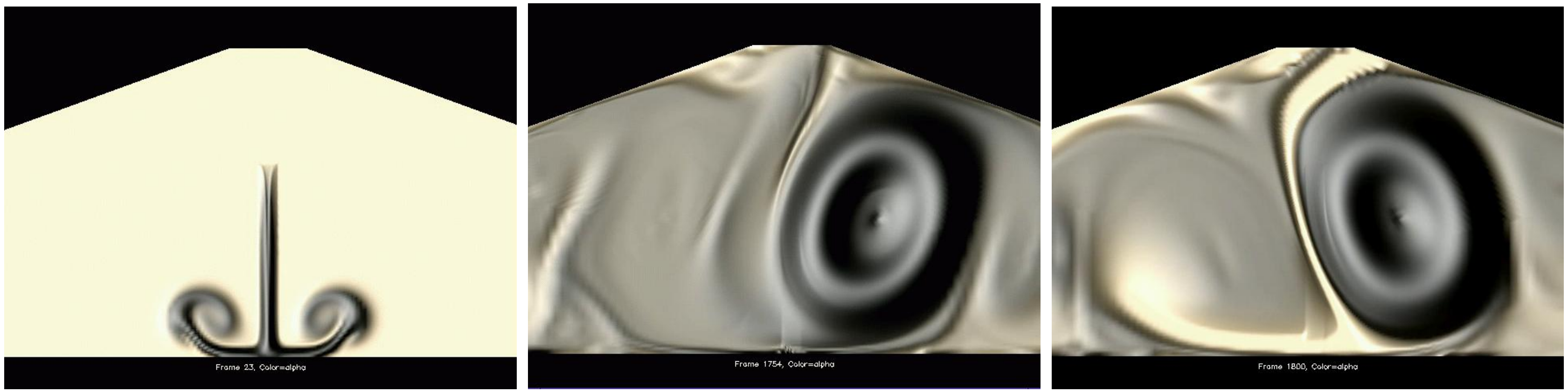
Finally, there is a last twist in the use of absorbing boundary conditions. As the flow is computed it may happen that the number of characteristics changes in time. Two examples follow.

Think at transport of a scalar on top of a velocity field obtained by an incompressible Navier-Stokes solver (not truly that in the example since both systems are coupled). If the exterior flow is not modeled then flow at the top opening may be reverted.



Dynamic boundary conditions (cont.)

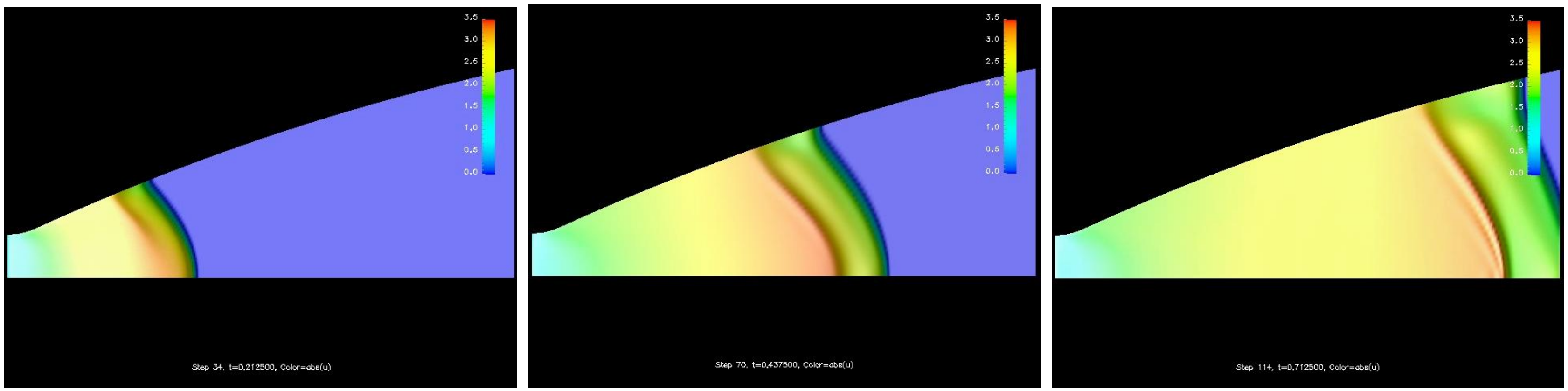
If the interior is modeled only, then it’s natural to leave concentration free at the top opening. However the flow can revert at some portions of the opening producing a large incoming of undetermined values (in practice large negative concentrations are observed). Imposing a value at the opening is stable but would lead to a spurious discontinuity at the outlet.



The ideal would be to switch *dynamically* from one condition to the other during the computation.

Nozzle chamber fill

The case is the ignition of a rocket launcher nozzle in a low pressure atmosphere. The fluid is initially at rest (143 Pa, 262 $^\circ\mathrm{K}$). At time $t = 0$ a membrane at the throat is broken. Behind the membrane there is a reservoir at 6×10^5 Pa, $4170^\circ\mathrm{K}$. A strong shock (intensity $p_1/p_2 >$1000) propagates from the throat to the outlet. The gas is assumed as ideal ($\gamma = 1.17$). In the steady state a supersonic flow with a max. Mach of 4 at the outlet is found. The objective of the simulation is to determine the time needed to fill the chamber ($< 1\mathrm{msec}$) and the final steady flow.



Nozzle chamber fill (cont.)

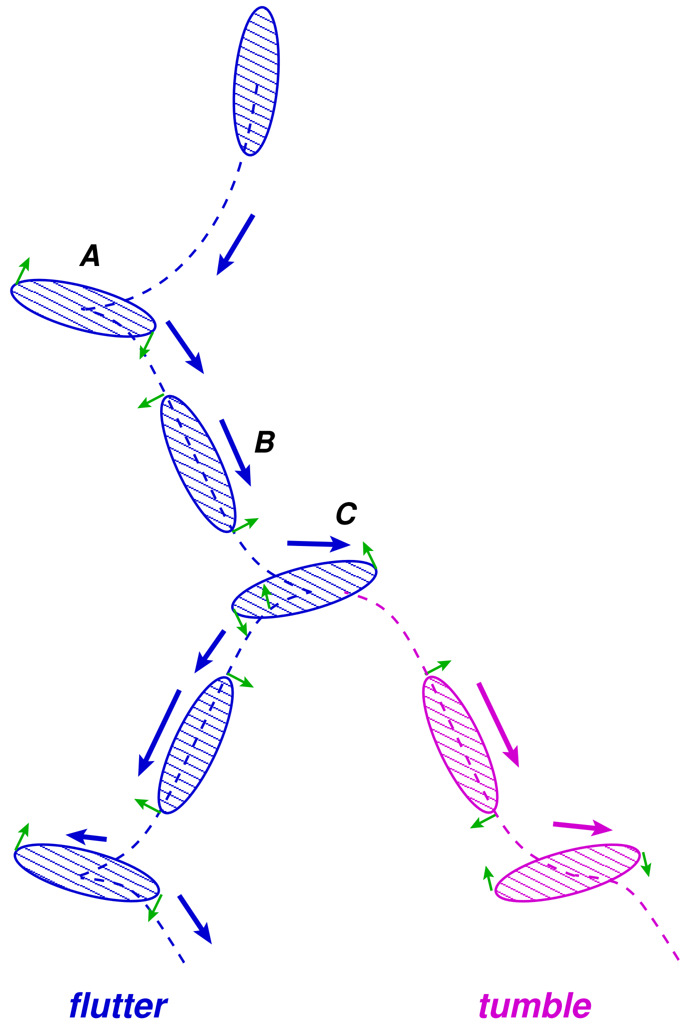
We impose density, pressure and tangential velocity at inlet (assuming subsonic inlet), slip condition at the nozzle wall. The problem is with the outlet boundary. Initially the flow is subsonic (fluid at rest) there, and switches to supersonic. The rule dictaminates to impose 1 condition, as a subsonic outlet (may be pressure, which is known) and no conditions after (supersonic outlet). If pressure is imposed during the wall computation, then a spurious shock is formed at the outlet.

This test case has been contrasted with experimental data obtained at ESTEC/ESA (European Space Research and Technology Centre-European Space Agency, Noordwijk, Holanda). The predicted mean velocity was $2621\ \mathrm{m/s}$ **to be compared with the experimental value of** $2650 \pm 50\ \mathrm{m/sec}$**.**

Again, the ideal would be to switch *dynamically* from one condition to the other *during the computation*.

Object falling at supersonic speed

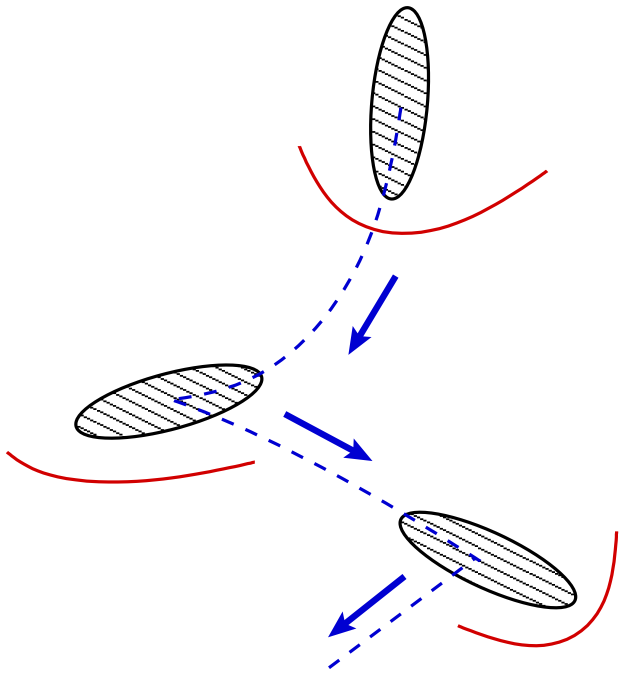
Consider, for simplicity, a two dimensional case of an homogeneous ellipse in free fall. As the body accelerates, the pitching moments tend to increase the angle of attack until it stalls (A), and then the body starts to fall towards its other end and accelerating etc... (“*flutter*”). However, if the body 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.



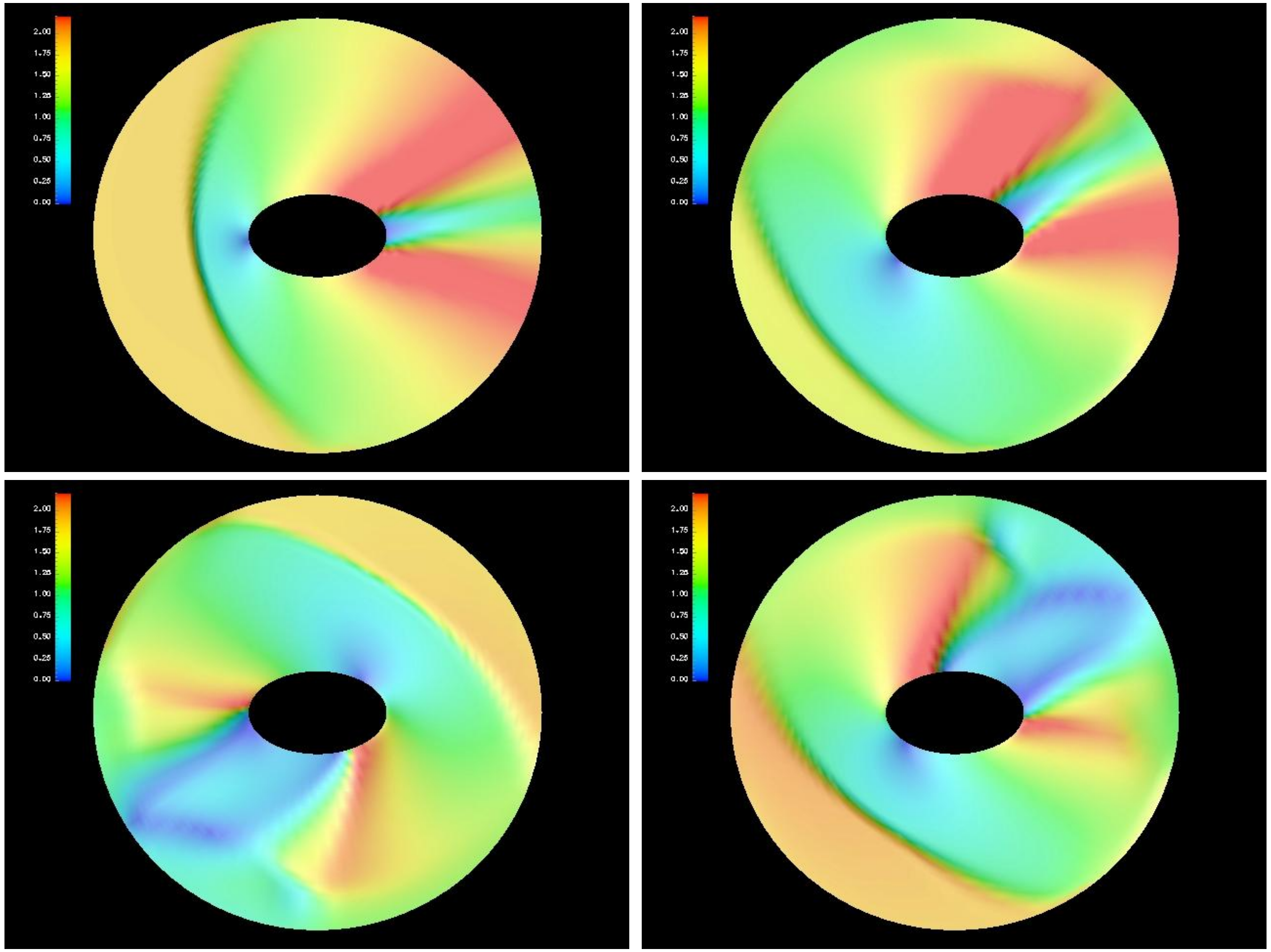
Object falling at supersonic speed (cont.)

Under certain conditions in size and density relation to the surrounding atmosphere it reaches supersonic speeds. In particular as form drag grows like L^2 whereas weight grows like L^3, larger bodies tend to reach larger limit speeds and eventually reach supersonic regime. At supersonic speeds the principal source of drag is the shock wave, we use slip boundary condition at the body in order to simplify the problem.

We also do the computation in a non-inertial system following the body, so that non-inertial terms (Coriolis, centrifugal, etc...) are added. In this frame some portions of the boundary are alternatively in all the conditions (subsonic incoming, subsonic outgoing, supersonic incoming, supersonic outgoing).
Again, the ideal would be to switch **dynamically** ***from one condition to the other during the computation.***



Object falling at supersonic speed (cont.)



Object falling at supersonic speed (cont.)

Whether RI or ULSAR based boundary conditions are used, if the number of incoming/outgoing characteristics vary in time this requires to change the profile of the system matrix during time evolution. In order to do this we add dynamic boundary conditions either via Lagrange multipliers or penalization. However this techniques add extra bad conditioning to the system of equations so that special iterative methods are needed.

Object falling at supersonic speed (cont.)

$$\mathbf{C}\,\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\,\frac{\partial \mathbf{U}}{\partial x} = 0.$$

Consider for simplicity a linear system of advective equations discretized with centered (i.e. no upwind) finite differences

$$\mathbf{C}\,\frac{\mathbf{U}_0^{n+1} - \mathbf{U}_0^n}{\Delta t} + \mathbf{A}\,\frac{\mathbf{U}_1^{n+1} - \mathbf{U}_0^n}{h} = 0;$$

$$\mathbf{C}\,\frac{\mathbf{U}_k^{n+1} - \mathbf{U}_k^n}{\Delta t} + \mathbf{A}\,\frac{\mathbf{U}_{k+1}^{n+1} - \mathbf{U}_{k-1}^n}{2h} = 0, \quad k \geq 1$$

$k \geq 0$ **node index,** $n \geq 0$ **time index,** $h =$ **mesh size,** $\mathbf{C}, \mathbf{A}$ $=$**enthalpy and advective Jacobians.**

Object falling at supersonic speed (cont.)

Using Lagrange multipliers for imposing the boundary conditions leds to the following equations

$$\boldsymbol{\Pi}^{+}\,(\mathbf{U}_0-\mathbf{U}_{\mathrm{ref}})+\boldsymbol{\Pi}^{-}\,\boldsymbol{\lambda}=0,$$

$$\mathbf{C}\,\frac{\mathbf{U}_0^{n+1}-\mathbf{U}_0^{n}}{\Delta t}+\mathbf{A}\,\frac{\mathbf{U}_1^{n+1}-\mathbf{U}_0^{n}}{h}+\mathbf{C}\boldsymbol{\Pi}^{+}\,\boldsymbol{\lambda}=0;$$

$$\mathbf{C}\,\frac{\mathbf{U}_k^{n+1}-\mathbf{U}_k^{n}}{\Delta t}+\mathbf{A}\,\frac{\mathbf{U}_{k+1}^{n+1}-\mathbf{U}_{k-1}^{n}}{2h}=0,\quad k\geq 1.$$

$\boldsymbol{\Pi}^{\pm}$ **is the projection operator onto incoming/outgoing waves,** $\boldsymbol{\lambda}$ **are the Lagrange multipliers.**

Using penalization

Add a small regularization term and then eliminate the Lagrange multipliers.

$$-\epsilon\boldsymbol{\lambda} + \mathbf{\Pi}^{+}\,(\mathbf{U}_0 - \mathbf{U}_{\mathrm{ref}}) + \mathbf{\Pi}^{-}\,\boldsymbol{\lambda} = 0,$$

$$\mathbf{C}\,\frac{\mathbf{U}_0^{n+1} - \mathbf{U}_0^{n}}{\Delta t} + \mathbf{A}\,\frac{\mathbf{U}_1^{n+1} - \mathbf{U}_0^{n}}{h} + \mathbf{C}\mathbf{\Pi}^{+}\,\boldsymbol{\lambda} = 0;$$

Eliminating the Lagrange multipliers $\boldsymbol{\lambda}$ we arrive to a boundary equation

$$\mathbf{C}\,\frac{\mathbf{U}_0^{n+1} - \mathbf{U}_0^{n}}{\Delta t} + \mathbf{A}\,\frac{\mathbf{U}_1^{n+1} - \mathbf{U}_0^{n}}{h} + (1/\epsilon)\mathbf{C}\mathbf{\Pi}^{+}(\mathbf{U}_0 - \mathbf{U}_{\mathrm{ref}}) = 0.$$

Absorbing boundary conditions and ALE

When using *Arbitrary Lagrangian-Eulerian* formulations:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathcal{F}_{c,j}(\mathbf{U})}{\partial x_j} - \mathbf{v}_{\mathrm{mesh}} \mathbf{U} = 0$$

$$A_{\mathrm{ALE},j} = \frac{\partial \mathcal{F}_{c,j}(\mathbf{U})}{\partial \mathbf{U}} - v_{\mathrm{mesh},j} \mathbf{I},$$ **ALE advective Jacobian**

Nbr. of incoming characteristics $= \mathrm{sum}(\mathrm{eig}(\mathbf{A} \cdot \hat{\mathbf{n}}) - \mathbf{v}_{\mathrm{mesh}} \cdot \hat{\mathbf{n}} < 0)$

Acknowledgment

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, grants PIP 0198/98, PIP 02552/00, PIP 5271/05), Universidad Nacional del Litoral (UNL, Argentina, grants CAI+D 2000/43) and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT 6973/99, PID-74/99, PICT Lambda 12-14573/2003, PME 209/2003).

We made extensive use of *Free Software* (`http://www.gnu.org`) as GNU/Linux OS, MPI, PETSc, GCC compilers, Octave, Open-DX among many others.