Automatic high order absorption layers for advective-diffusive systems of equations

by Mario Storti, Laura Battaglia, and Rodrigo Paz

Centro Internacional de Métodos Computacionales en Ingeniería - CIMEC

INTEC, (CONICET-UNL), Santa Fe, Argentina

<mario.storti at gmail.com>

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

Motivation for Absorbing Boundary Conditions

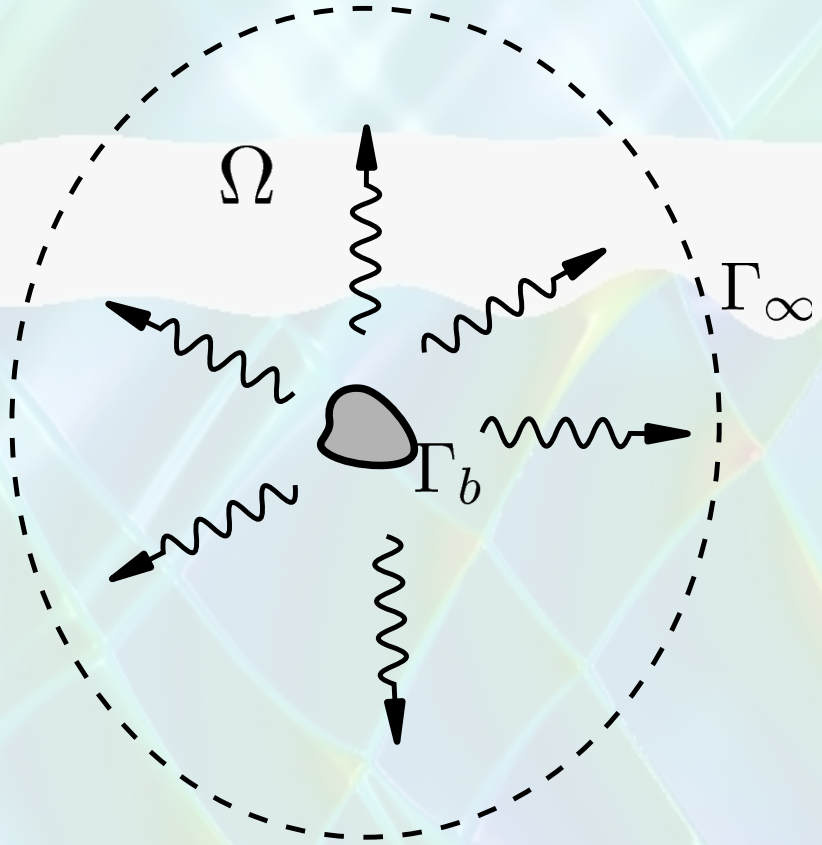
In wave-like propagation problems, not including ABC may lead to

non-convergent solutions.

$$u_{tt} = c^2 \Delta u, \text{ in } \Omega$$

$$u = \bar{u}(x,t), \text{ at } \Gamma_b, \quad u = 0, \text{ at } \Gamma_\infty v$$

- u **doesn't converge to the correct solution irradiating energy from the source, even if** $\Gamma_\infty \to \infty$**. A *standing wave* is always found.**
- u **is unbounded if** $\bar{u}$ **emits in an eigenfrequency which is a resonance mode of the closed cavity.**

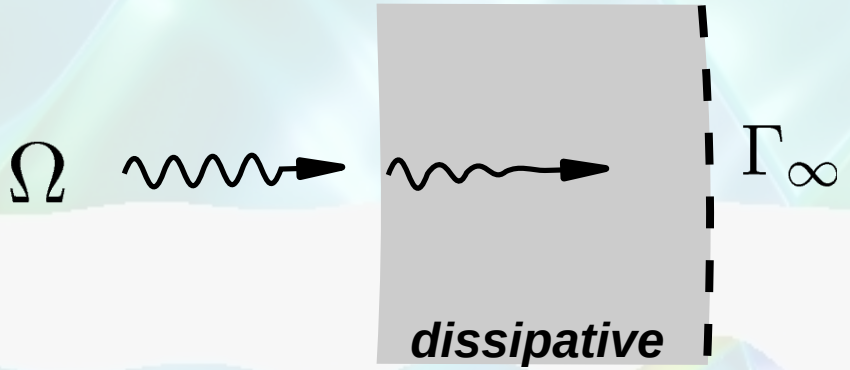


Absorbing boundary conditions must be added to the outer boundary in order to let energy be extracted from the domain.

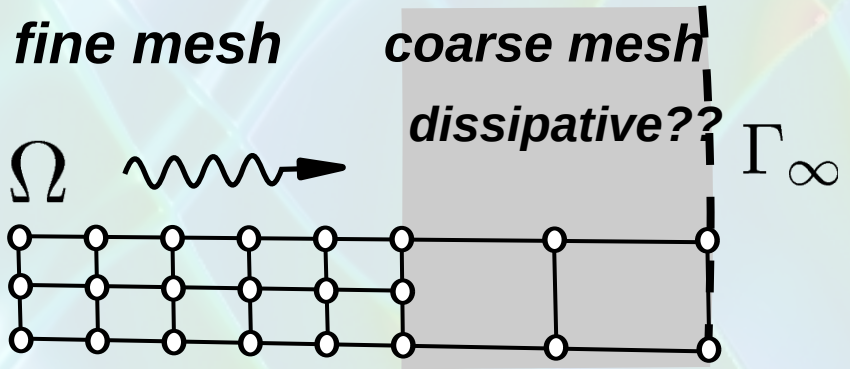
((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

Motivation for Absorbing Boundary Conditions (cont.)

If a dissipative region is large enough so that many wavelengths are included in the region may act as an absorbing layer, i.e. as an absorbing boundary condition.



However, it is a common misconception to assume that a coarse mesh adds dissipation and consequently may improve absorption. For instance for the wave equation a coarse mesh may lead to evanescent solutions and then to act as a fully reflecting boundary.



((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

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.

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

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

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

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

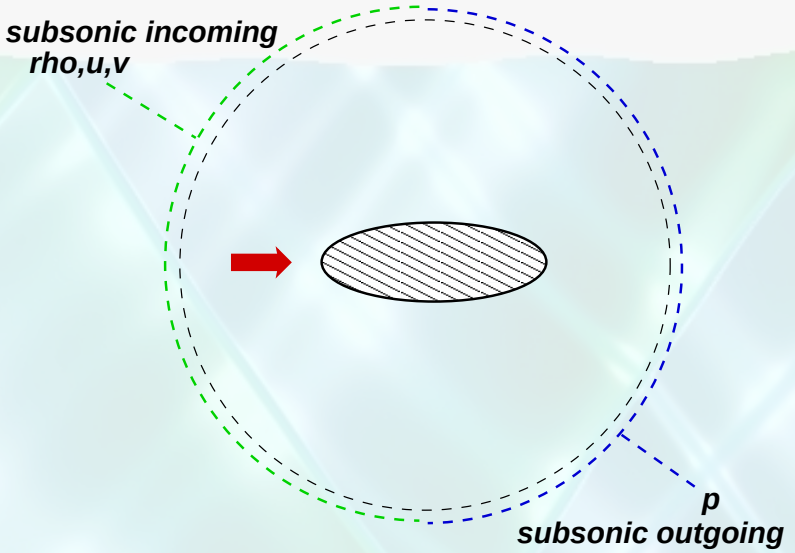
Adding extra Dirichlet conditions leads to spurious shocks, and lack of enough Dirichlet conditions leads to instability.

Boundary conditions for advective diffusive systems (cont.)

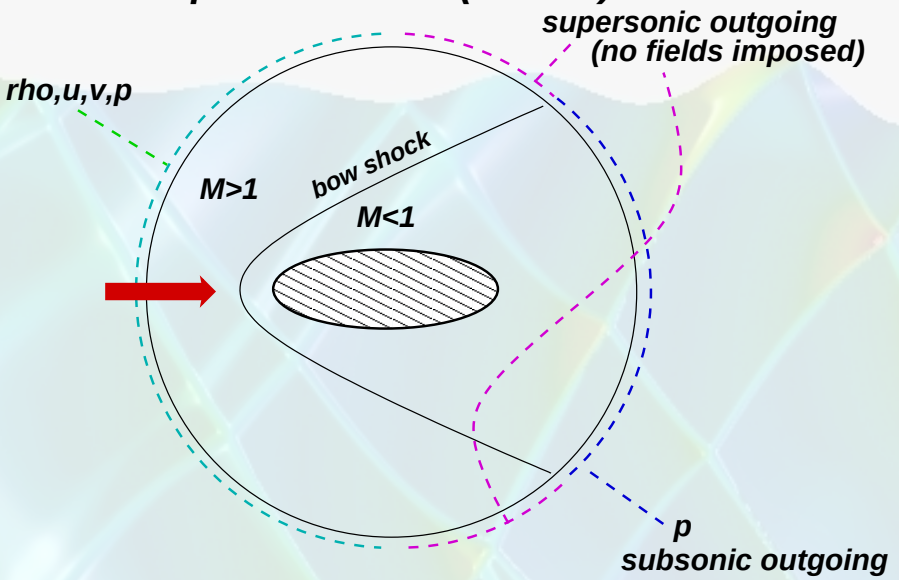
For simple scalar advection problems the Jacobian is the transport velocity. The rule is then to check the projection of velocity onto the exterior normal.

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.

subsonic flow (Minf<1)



supersonic flow (Minf>1)



((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

Absorbing boundary conditions

However, this kind of conditions are, generally, reflective. Consider a pure advective system of equations in 1D, i.e., $\mathcal{F}_{d,j} \equiv 0$

$$\frac{\partial \mathcal{H}(\mathbf{U})}{\partial t} + \frac{\partial \mathcal{F}_{c,x}(\mathbf{U})}{\partial x} = 0, \text{ in } [0, L]. \tag{1}$$

If the system is *"linear"*, i.e., $\mathcal{F}_{c,x}(\mathbf{U}) = \mathbf{A}\mathbf{U}$, $\mathcal{H}(\mathbf{U}) = \mathbf{C}\mathbf{U}$ ($\mathbf{A}$ and $\mathbf{C}$ do not depend on $\mathbf{U}$), a first order linear system is obtained

$$\mathbf{C}\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} = 0. \tag{2}$$

The system is *"hyperbolic"* if $\mathbf{C}$ is invertible, $\mathbf{C}^{-1}\mathbf{A}$ is diagonalizable with real eigenvalues. If this is the case, it is possible to make the following eigenvalue decomposition for $\mathbf{C}^{-1}\mathbf{A}$

$$\mathbf{C}^{-1}\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}, \tag{3}$$

where $\mathbf{S}$ is real and invertible and $\mathbf{\Lambda}$ is real and diagonal. If new variables are

defined $\mathbf{V} = \mathbf{S}^{-1}\mathbf{U}$, then equation (2) becomes

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{\Lambda}\frac{\partial \mathbf{V}}{\partial x} = 0. \tag{4}$$

Now, each equation is a linear scalar advection equation

$$\frac{\partial v_k}{\partial t} + \lambda_k \frac{\partial v_k}{\partial x} = 0, \quad \textbf{(no summation over } k\textbf{)}. \tag{5}$$

v_k **are the *"characteristic components"* and** λ_k **are the *"characteristic velocities"* of propagation.**

Linear 1D absorbing boundary conditions

Assuming $\lambda_k \neq 0$, the absorbing boundary conditions are, depending on the sign of λ_k,

$$\begin{aligned} &\textbf{if } \lambda_k > 0\textbf{: } v_k(0) = \bar{v}_{k0}; \quad &&\textbf{no boundary condition at } x = L \\ &\textbf{if } \lambda_k < 0\textbf{: } v_k(L) = \bar{v}_{kL}; \quad &&\textbf{no boundary condition at } x = 0 \end{aligned} \tag{6}$$

This can be put in compact form as

$$\begin{aligned} \mathbf{\Pi}_V^+(\mathbf{V} - \bar{\mathbf{V}}_0) = 0; \qquad &\textbf{at } x = 0 \\ \mathbf{\Pi}_V^-(\mathbf{V} - \bar{\mathbf{V}}_L) = 0; \qquad &\textbf{at } x = L \end{aligned} \tag{7}$$

Linear 1D absorbing boundary conditions (cont.)

$\mathbf{\Pi}_V^{\pm}$ **are the *projection matrices onto the right/left-going characteristic modes* in the** $\mathbf{V}$ **basis,**

$$\mathbf{\Pi}^{+}_{V,jk} = \begin{cases} 1; & \text{if } j = k \text{ and } \lambda_k > 0 \\ 0; & \text{otherwise}, \end{cases} \tag{8}$$

$$\mathbf{\Pi}^{+} + \mathbf{\Pi}^{-} = \mathbf{I}.$$

It can be easily shown that they are effectively *projection matrices*, i.e., $\mathbf{\Pi}^{\pm}\mathbf{\Pi}^{\pm} = \mathbf{\Pi}^{\pm}$ **and** $\mathbf{\Pi}^{+}\mathbf{\Pi}^{-} = 0$**. Coming back to the boundary condition at** $x = L$ **in the** $\mathbf{U}$ **basis, it can be written**

$$\mathbf{\Pi}_V^{-}\,\mathbf{S}^{-1}(\mathbf{U} - \bar{\mathbf{U}}_L) = 0 \tag{9}$$

or, multiplying by $\mathbf{S}$ **at the left**

$$\mathbf{\Pi}_U^{\pm}\,(\mathbf{U} - \bar{\mathbf{U}}_{0,L}) = 0, \quad \text{at } x = 0, L, \tag{10}$$

where

$$\mathbf{\Pi}_U^{\pm} = \mathbf{S}\,\mathbf{\Pi}_V^{\pm}\,\mathbf{S}^{-1}, \tag{11}$$

((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

Linear 1D absorbing boundary conditions (cont.)

$$\begin{aligned} &\mathbf{\Pi}_U^{\pm}\left(\mathbf{U}-\bar{\mathbf{U}}_{0,L}\right)=0, \quad \text{at } x=0,L, \\ &\mathbf{\Pi}_U^{\pm}=\mathbf{S}\,\mathbf{\Pi}_V^{\pm}\,\mathbf{S}^{-1}, \end{aligned} \tag{12}$$

These conditions are completely absorbing for 1D linear advection system of equations (2).

The rank of $\mathbf{\Pi}^{+}$ is equal to the number n_{+} of positive eigenvalues, i.e., the number of right-going waves. Recall that the right-going waves are incoming at the $x=0$ boundary and outgoing at the $x=L$ boundary. Conversely, the rank of $\mathbf{\Pi}^{-}$ is equal to the number n_{-} of negative eigenvalues, i.e., the number of left-going waves (incoming at $x=L$ and outgoing at the $x=0$ boundary).

ABC for nonlinear problems

First order absorbing boundary conditions may be constructed by imposing exactly the components along the incoming characteristics.

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

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

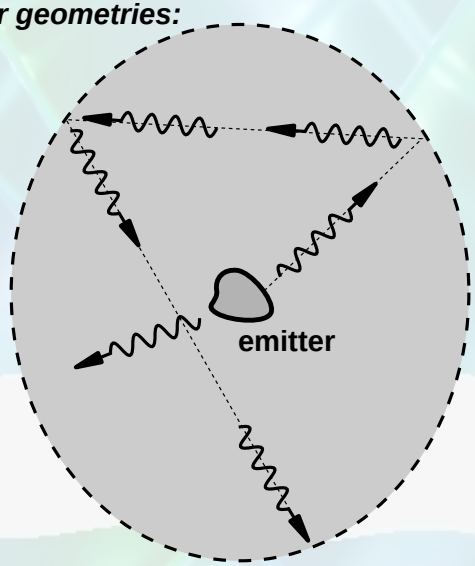
What for non-linear problems? A posisble strategy is to linearize the state around the last state or a predictor state.

- **Storti, M.; Nigro, N.; Paz, R.R.; Dalcin, L. "Dynamic boundary conditions in Computational Fluid Dynamics"** ***Computer Methods in Applied Mechanics and Engineering*** **197(13-16), pp. 1219-1232 (2008)**
- **Paz, R.R.; Storti, M.; Garelli, L. "Local Absorbent Boundary Condition for Nonlinear Hyperbolic Problems with Unknown Riemann Invariants"** ***Computers and Fluids*** **40, pp. 52-67 (2011)**

(launch video cylabso-xoffset015-Rext10)

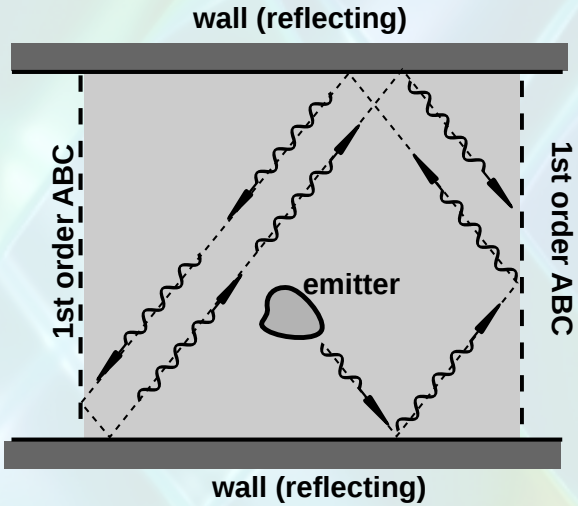
Higher order absorbing boundary conditions

irregular geometries:



In 2D or 3D a first order ABC will be partially reflecting for non-normal incidence. This can lead to large errors, speciall for very regular geometries like a channel.

regular geometries:



((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

Higher order absorbing boundary conditions (cont.)

There are several approaches for higher order aborbing layers, among them a popular approach is the **Perfectly Matched Layer** (PML, Makefile. J-P Berenger, J. Comput. Phys. 114, 185 (1994)). However,

- It is **addhoc** for each physical problem.
- It is formulated only on **rectangular domains**.
- It's **not very robust**.
- It requires **additional variables** to be defined in the program in the absorbing layer.

The first is the most concerning us. We want an absorbing numerical device with the following characteristics

- To be **automatically computable** from the flux function (and perhaps the Jacobian of the fluxes).
- To be adjustable **higher order** (more probably first and second order).
- To be **robust**.

- **If auxiliary variables are needed, they must be easily computable in the actual context.**

Advective/diffusive systems implemented

- **Navier-Stokes compressible flow (),**
- **Shallow water equations (),**
- **Stratified shallow water equations (),**
- **1D shallow water equations in channels of arbitrary section (),**
- **Scalar advection/diffusion (),**
- **Scalar wave equation ()**
- **Maxwell equations ()**

( = implemented in PETSc-FEM)

A second order absorbing layer

If, for the *first order* aborbing boundary condition we condense the Lagrange multiplier version, we get a penalized version of the form

$$\mathbf{C}\frac{\mathbf{U}_0^{n+1}-\mathbf{U}_0^n}{\Delta t}+\mathbf{A}\,\frac{\mathbf{U}_1^{n+1}-\mathbf{U}_0^{n+1}}{h}+\frac{1}{\epsilon}\mathbf{C}\mathbf{\Pi}_U^+\mathbf{U}_0^{n+1}=0. \qquad (13)$$

Assuming that we want to add an absorbing layer in $0\leq x\leq L$, then the equation would be

$$\mathbf{C}\frac{\partial\mathbf{U}}{\partial t}+\mathbf{H}\mathbf{U}+\mathbf{A}\frac{\partial\mathbf{U}}{\partial x}+\mathbf{B}\frac{\partial\mathbf{U}}{\partial y}=0, \qquad (14)$$

where $\mathbf{H}$ is the **matrix of absorbing coefficients** to be yet defined. We now transform Fourier in t and y with associated variables $\mathrm{i}\omega$ and $\mathrm{i}k_y$, i.e. we assume

$$\mathbf{U}(x,y,t)=\hat{\mathbf{U}}(x)\exp\{\mathrm{i}(k_y y-\omega t)\}. \qquad (15)$$

Then, we get (for simplicity we drop the hat symbol ($\hat{\mathbf{U}} \rightarrow \mathbf{U}$),

$$\begin{aligned} &\mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + (-\mathrm{i}\omega\mathbf{C} + \mathbf{H} + \mathrm{i}k_y\mathbf{B})\mathbf{U} = 0, \\ \Longrightarrow \quad &\frac{\partial \mathbf{U}}{\partial x} + (\mathbf{A}^{-1}\mathbf{H} - \mathrm{i}\omega\mathbf{M}(z))\mathbf{U} = 0, \end{aligned} \tag{16}$$

where

$$\begin{aligned} z &= k_y/\omega, \\ \mathbf{M}(z) &= \mathbf{A}^{-1}(\mathbf{C} - z\mathbf{B}). \end{aligned} \tag{17}$$

In order to not have reflections $\mathbf{A}^{-1}\mathbf{H}$ must **have the same eigenvectors** that $\mathbf{M}(z)$. One possibility is then to diagonalize $\mathbf{M}(z)$

$$\mathbf{M}(z) = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}, \tag{18}$$

and then to force $\mathbf{A}^{-1}\mathbf{H}$ to be **diagonal in the same basis**

$$\mathbf{A}^{-1}\mathbf{H} = \mathbf{Q}\mathbf{\Lambda}_H\mathbf{Q}^{-1}. \tag{19}$$

In order to have an absorbing layer we must have

$$\mathrm{sign}(\lambda(\mathbf{A}^{-1}\mathbf{H})_j) = \mathrm{sign}(\lambda_j), \tag{20}$$

Transforming Fourier back to t and y we obtain the desired absorbing layer. However, $\mathbf{Q}$ depends on z and so will $\mathbf{H}$ so it would be non-local. In order to have a local operator we perform an expansion of it in powers of z. For instance if we can approximate it to

$$\mathbf{H}(z) \approx \mathbf{H}_0 + z\delta\mathbf{H}, \tag{21}$$

then the absorbing term would be

$$\mathbf{H}_0\mathbf{U} + z\delta\mathbf{H}\mathbf{U} = \mathbf{H}_0\mathbf{U} + (\mathrm{i}k_y/\mathrm{i}\omega)\delta\mathbf{H}\mathbf{U}. \tag{22}$$

Transforming Fourier back to (t, y) we get

$$\begin{aligned} \mathbf{C}\frac{\partial \mathbf{U}}{\partial t} + \mathbf{H}(\{\mathbf{U}\}) + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} &= 0, \\ \mathbf{H}(\{\mathbf{U}\}) = \mathbf{H}_0\mathbf{U} + \delta\mathbf{H}\int_{t=0}^{t} \frac{\partial \mathbf{U}}{\partial y}\,\mathrm{d}t. & \end{aligned} \tag{23}$$

The simplest choice for (29) is $\lambda(\mathbf{A}^{-1}\mathbf{H})_j = K\lambda_j$, and we get $\mathbf{H}_0 = K\mathbf{C}$, $\mathbf{H}_1 = -K\mathbf{B}$, so that the absorbing layer is

$$\mathbf{C}\frac{\partial \mathbf{U}}{\partial t} + K\left\{\mathbf{C}\mathbf{U} - \mathbf{B}\int_{t=0}^{t}\frac{\partial \mathbf{U}}{\partial y}\,\mathrm{d}t\right\} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} = 0, \qquad (24)$$

Note that in the first order case we get a local operator.

Numerical evaluation of the absorbing operator

- It involves an **integral over time**, that involves storing an auxiliary variable say $\mathbf{W}$ and then updating with

$$\mathbf{W} = \int_{t=0}^{t} \frac{\partial \mathbf{U}}{\partial y} \, \mathrm{d}t,$$
$$\frac{(3\mathbf{W}^{n+1} - 4\mathbf{W}^{n} + \mathbf{W}^{n-1})_{jk}}{2\Delta t} = \frac{(\mathbf{U}_{j,k+1} - \mathbf{U}_{j,k-1})^{n+1}}{2\Delta y}. \tag{25}$$

- The derivative with respect to y is computed by standard finite difference approximations ($j(k)$ indices is along $x(y)$ axis).
- The operator can be easily evaluated in the context of an **unstructured grid solver** (FEM for instance).
- $K[=]\mathrm{T}^{-1}$ but if a velocity scale $\bar{v}$ is available then we get a **length scale** $\bar{L}_{\mathrm{abso}} = \bar{v}/K$

Examples. The scalar wave equation

The standard representation of the scalar wave equation is

$$\phi_{tt} = c^2 \Delta \phi \tag{26}$$

In 2D we can put it in the form of an advective-diffusive as

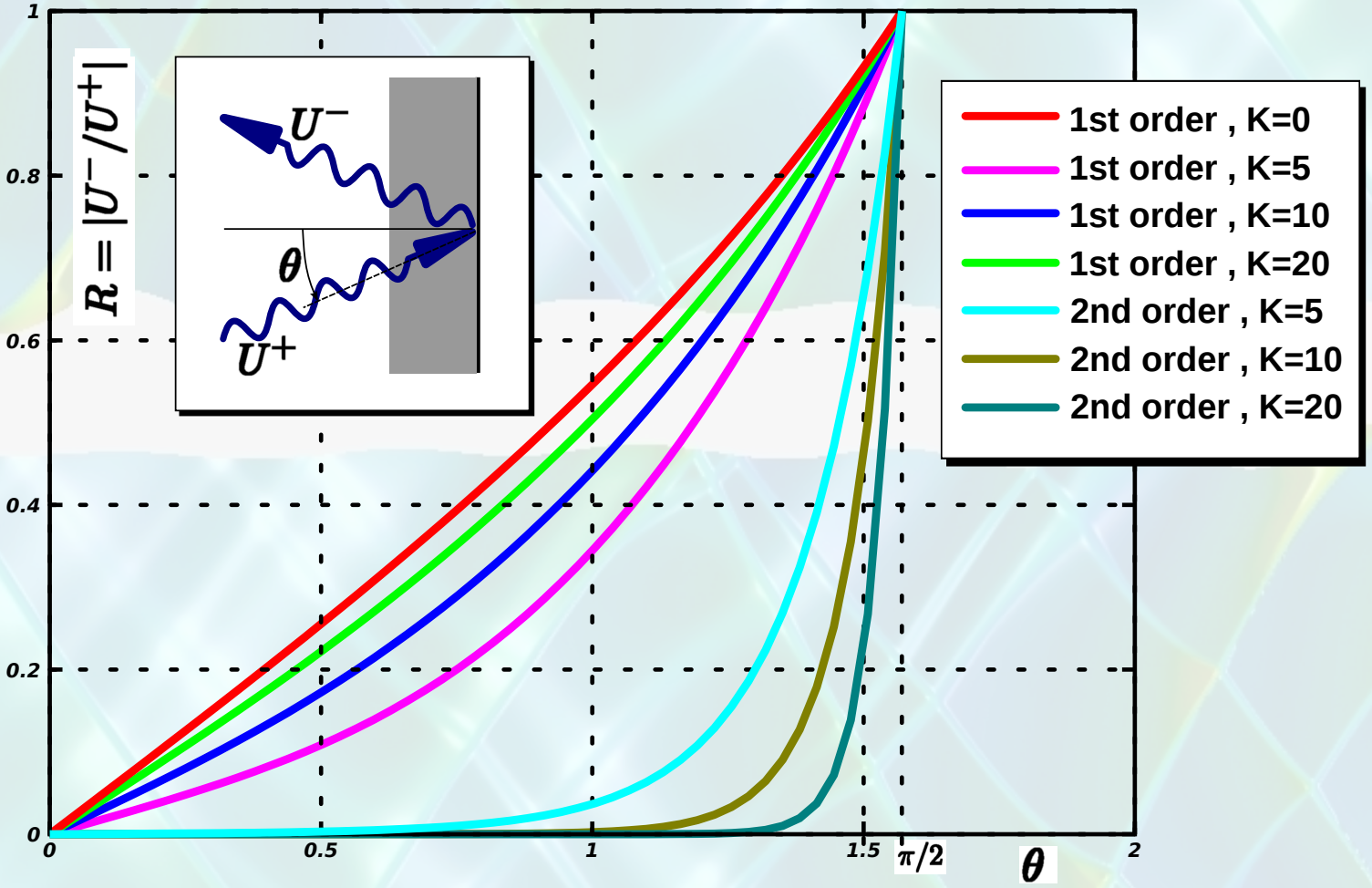
$$\begin{aligned} u_t + c(u_x + v_y) &= 0, \\ v_t + c(-v_x + u_y) &= 0. \end{aligned} \tag{27}$$

and the corresponding vectorial form is

$$\mathbf{U} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad A_x = c \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad A_y = c \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{28}$$

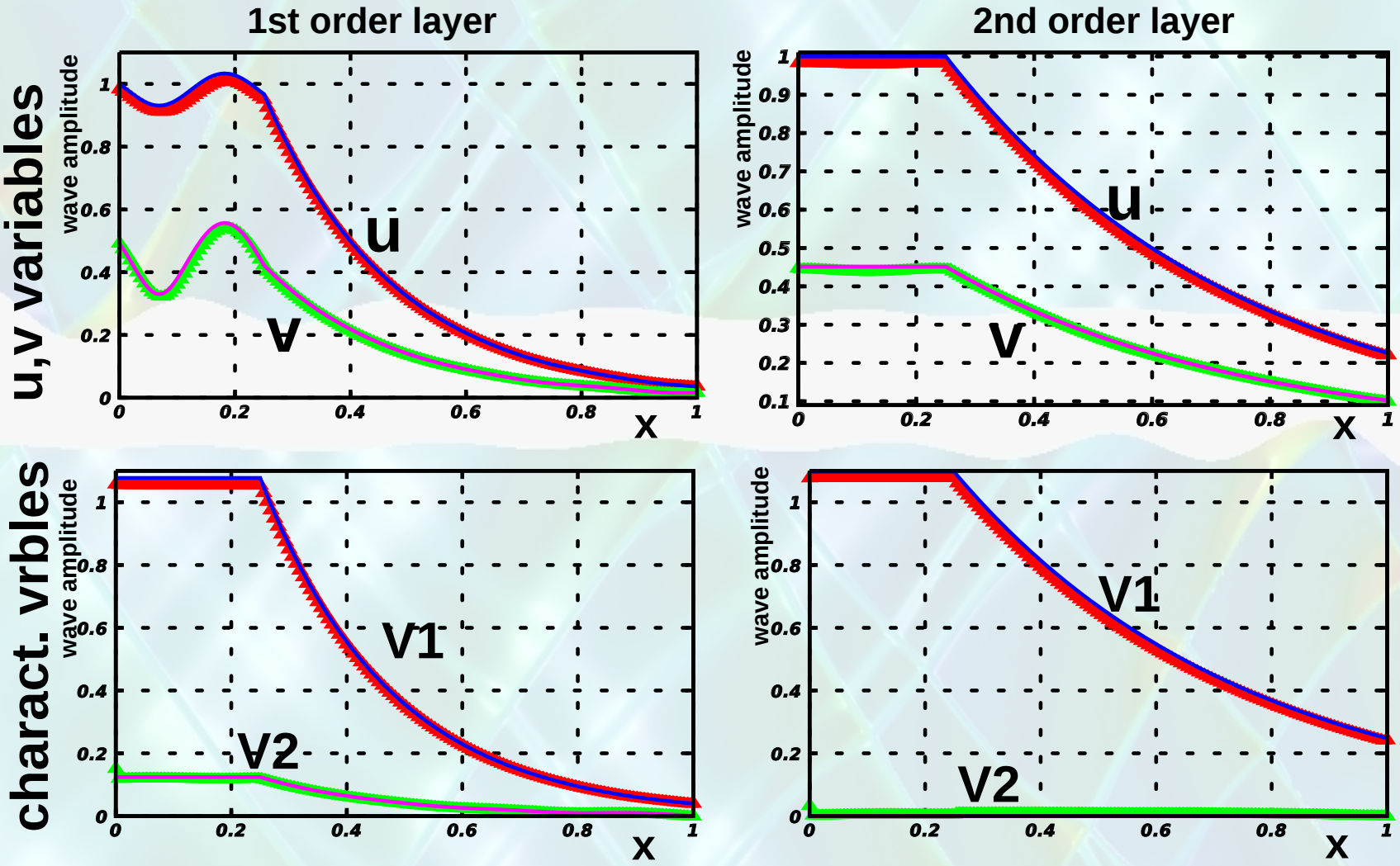
u and v satisfy the wave equation, and the dispersion relation is $\omega = ck$ (the same as for the wave equation).

Reflection coefficients



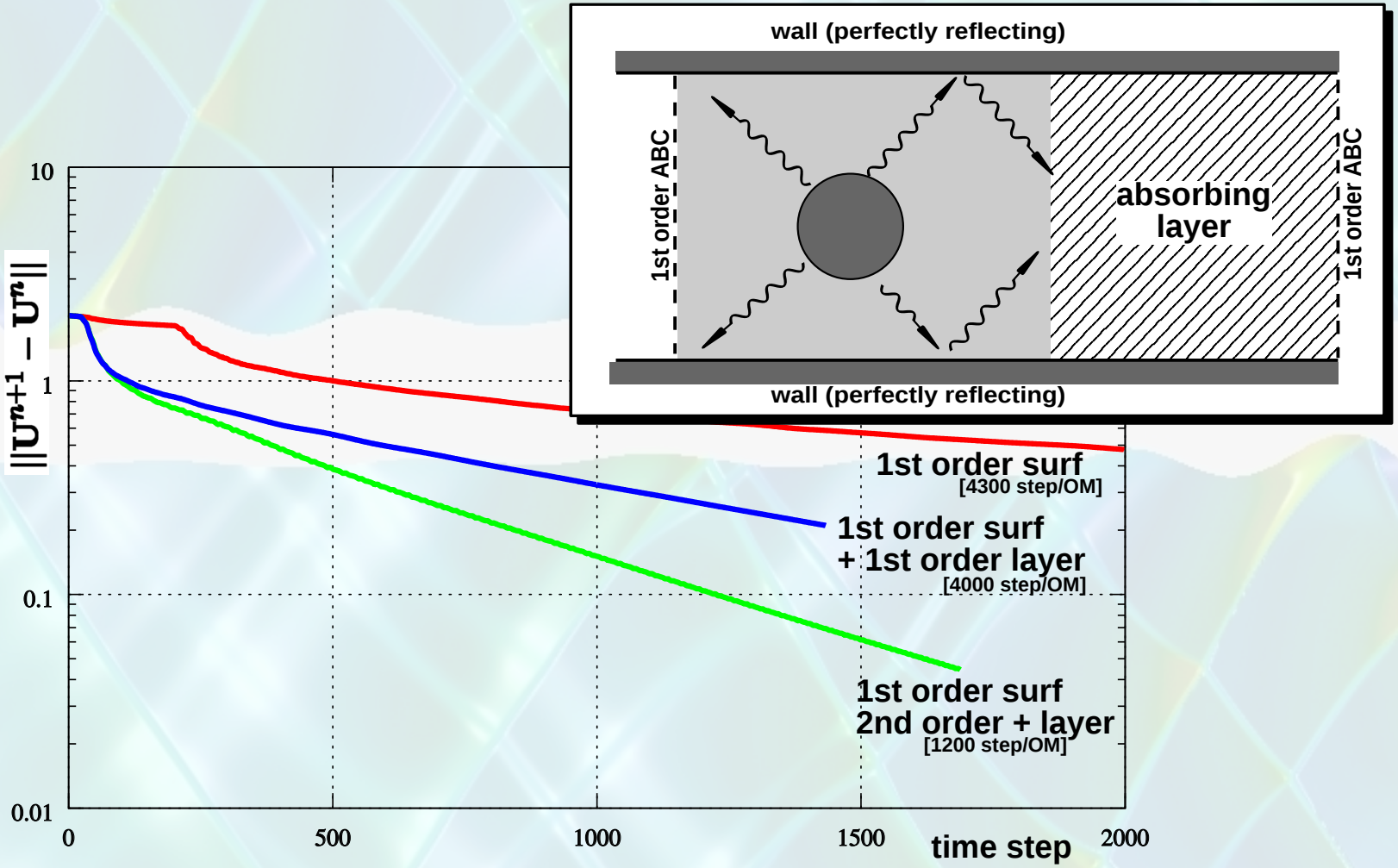
((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

Reflection coefficients (cont.)



((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

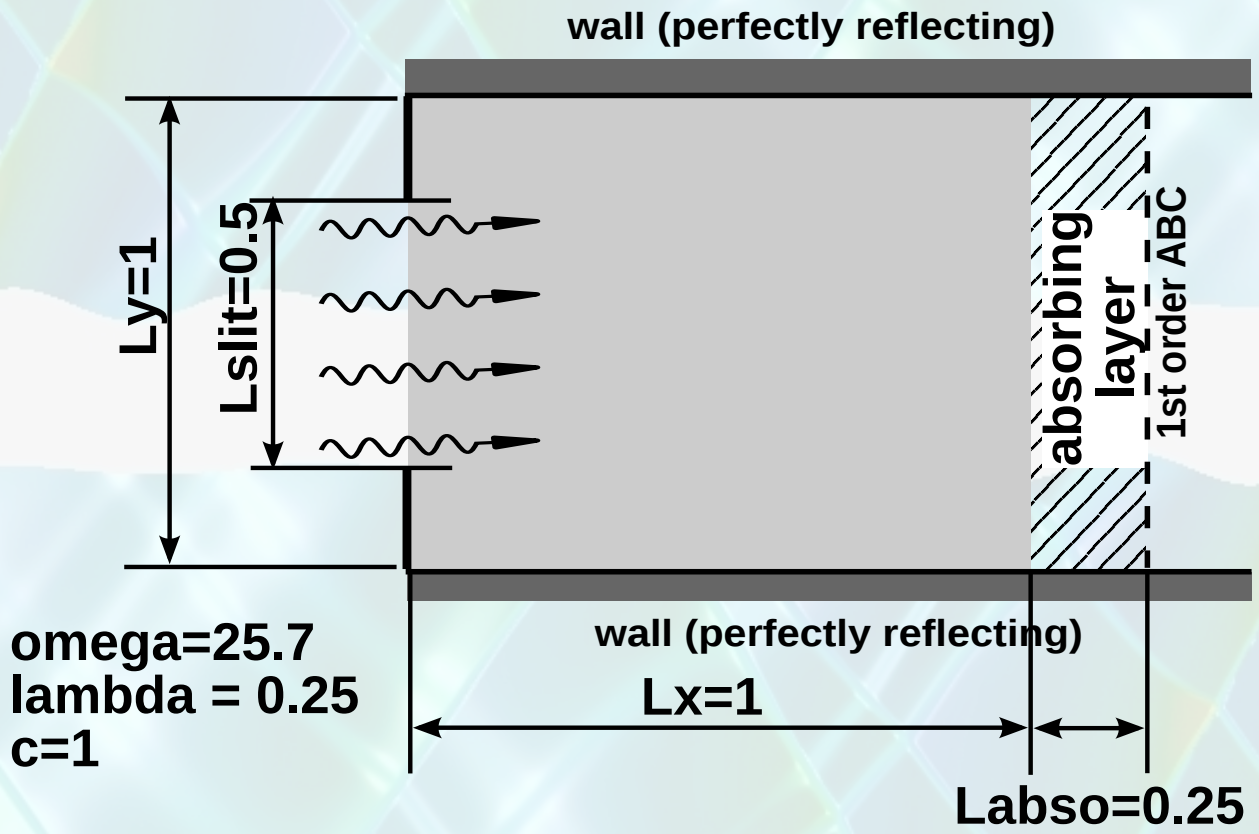
Decaying perturbation example



(launch video fsabso2d-pertini)

((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

Diffraction by a slit



(launch video fsabso2d-slit-best)

Other choices for the absorbing matrix coefficient

Recall that the requirement on $\mathbf{H}$ is

$$\text{sign}(\lambda(\mathbf{A}^{-1}\mathbf{H})_j) = \text{sign}(\lambda_j), \tag{29}$$

some possible choices are

$$\begin{aligned}
\lambda(\mathbf{A}^{-1}\mathbf{H})_j) = \lambda_j \quad &\Longrightarrow \quad \mathbf{H}_0 = \mathbf{C}, \\
\lambda(\mathbf{A}^{-1}\mathbf{H})_j) = \text{sign}(\lambda_j) \quad &\Longrightarrow \quad \mathbf{H}_0 = \mathbf{A}|\mathbf{A}^{-1}\mathbf{C}|, \\
\lambda(\mathbf{A}^{-1}\mathbf{H})_j) = 1/\lambda_j \quad &\Longrightarrow \quad \mathbf{H}_0 = \mathbf{A}|\mathbf{C}^{-1}\mathbf{A}|,
\end{aligned} \tag{30}$$

The expansion $\mathbf{H}(z) \approx \mathbf{H}_0 + z\delta\mathbf{H}$ can be donde numerically if it is not possible to do analytically, i.e. compute $\mathbf{H}(z)$ for several z and fit with polynomials for each entry in $\mathbf{H}$.

CIMEC

Conclusions

- An aborbing high order layer has been presented.
- It can be automatically computed programatically from the flux function and its Jacobians  .
- It requires an auxiliary variable that is basically the integral over time of the lateral derivatives. (It's non-local in time)  .
- Several examples for the wave equation have been shown with the simplest choice $\mathbf{H} = \mathbf{C} - z\mathbf{B}$.
- Other choices for $\mathbf{H}$ may be computed by polynomial fitting on z.
- Computational cost is negligible (but results on the absorbing layer must be discarded) ( / )

((version texstuff-1.0.36-102-gdb116be Thu Nov 3 09:14:55 2011 -0300) (date Thu Nov 3 09:15:35 2011 -0300))

Acknowledgment

This work has received financial support from Universidad Nacional del Litoral (UNL, Argentina, grants CAI+D 2005-10-64) and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT PME 209/2003, PICT-1141/2007, PICT-1506/2006).

We made extensive use of *Free Software* (`http://www.gnu.org`) as GNU/Linux OS, MPI, PETSc, GCC/G++ compilers, Octave, VTK, Python, Git, among many others. In addition, many ideas from these packages have been inspiring to us.