

Mecánica de Fluidos Computacional y Cálculo Paralelo en Ingeniería

Dr.Ing. Rodrigo R. PAZ

CONICET

::Centro Internacional de Métodos Computacionales en Ingeniería::

INTEC-UNL

Argentina

::2007::

4^{tas} Jornadas de Actualización en Aeronáutica Civil, Colegio de Ingenieros Especialistas de Córdoba, Octubre de 2007.



Temario

- 1 CIMEC
- 2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones
 - Solución de Sistemas de Ecuaciones Lineales
 - Método de Descomposición de Dominio (DDM)
 - Precondicionamiento para el sistema $Ax = b$.
- 3 Parte II – Aplicaciones y Usos
 - ESA CFD Code Evaluation Tests
 - Condiciones de Borde Dinámicas en CFD
 - Fluid-Structure Application Problems
 - Descripción Aerodinámica de un Automóvil de Competición
- 4 Parte III – Conclusiones Finales
 - Trabajos Publicados

¿Qué es el CIMEC?

- Centro Internacional de Métodos Computacionales en Ingeniería.
- Depende del CONICET y UNL.
- CONICET: 14 Investigadores, 21 Becarios de Doctorado, 3 Personal de Apoyo.
- Dedicación: Investigación y Desarrollo en Métodos Numéricos para la Mecánica del Continuo e Ingeniería, Docencia y Servicios a Terceros.

¿Qué es el CIMEC?

- **Investigación:** Publicaciones de trabajos científicos de alto impacto en revistas internacionales especializadas.
- **Docencia:** La mayoría de los Investigadores, Becarios y Personal de Apoyo tienen cargo docente en UNL.
A menudo los alumnos de grado que finalizan sus estudios optan por una Beca de CONICET para realizar el Doctorado.
- Los **Servicios de Transferencia** en los últimos 5 años se hicieron trabajos para las siguientes empresas:



¿Qué es el CIMEC? (cont.)

- ESA, European Space Agency (ESTEC–Open Engineering),
- Autoridad Regulatoria Nuclear (ARN/ex-CNEA),
- Repsol YPF,
- Siderar,
- Ministerio de Asuntos Hídricos de Santa Fe,
- IMPSA,
- Edival SA,
- Siderca,
- Favot SA,
- Instituto Argentino de Siderurgia,
- Fundación San Cayetano,
- Garro Fabril SA,
- Gesval SA,
- Ruedas RAR.

Un gran número de estos derivan en publicaciones internacionales retroalimentando el ciclo.

Cálculo Paralelo y Resolución de Grandes Sistemas de Ecuaciones.

- Para simular un problema en Mecánica de Fluidos con un código implícito hay que integrar temporalmente (**lazo temporal**) las ecuaciones de gobierno (Navier-Stokes compresibles e incompresibles, Euler, etc), y en cada paso de tiempo resolver un problema **NO LINEAL** (**lazo de Newton**) y en cada iteración no lineal resolver un sistema lineal de varios millones de *gdl* iterativamente (loop **lineal**, e.g. GMRES, Conjugate Gradients).

$$\mathbf{R} = \mathbf{K}(\mathbf{U})[\theta\mathbf{U}^{\tau+1} + (1 - \theta)\mathbf{U}^{\tau}] + \mathbf{B}(\mathbf{U})\frac{\mathbf{U}^{\tau+1} - \mathbf{U}^{\tau}}{\Delta t} - \mathbf{Q}^{\tau+1} = 0, \quad (1)$$

para $\theta \neq 0$ y $\tau = n \implies$ resolver **$Ax = b$**

(2) 



- **Descomposición de Dominio** es una técnica o familia de técnicas para resolver sistemas de ecuaciones lineales que surgen de la discretización de problemas gobernados por ‘Ecuaciones Diferenciales en Derivadas Parciales’.
- Esta basado en el paradigma de ‘**Dividir, combinar y vencer**’. Se propone resolver el problema global a partir de la división del mismo en subproblemas que puedan resolverse en forma **casi** independiente. Esta característica es la que hace al método de Descomposición de Dominio un esquema altamente paralelizable.

Cálculo Paralelo y Resolución de Grandes Sistemas de Ecuaciones.

- Se viene desarrollando un código de Elementos Finitos en Paralelo para resolver problemas Multifísica (código PETSc-FEM: <http://www.cimec.org.ar/petscfem> - Storti, Nigro, Paz, Dalcín).
- En el CIMEC se tiene un 'cluster' tipo 'Beowulf' de 100 nodos para hacer simulaciones en gran escala en Mecánica de Fluidos y Problemas Multifísica en general. También se está proyectando la construcción de un cluster para hacer cálculo masivo (approx. 1000-1200 procesadores).

Motivos - Antecedentes (cont.)

- Se sabía que en problemas de la **Mecánica Estructural** y la **Mecánica de Sólidos** surgían serios inconvenientes en la solución del sistema lineal cuando se usaban mallas muy finas en dominios (geometrías) muy complejas o con discontinuidades en las propiedades y capas límites (**Bramble et al. [1986]**, **Gropp et al. [1987]**, **Gambolati et al. [1988]**, **Bramble et al. [1989]**, **Mandel [1990]**, **Farhat et al [1991]**, **Papadrakakis et al. [1993]**, <http://www.ddm.org>, en CMAME, IJNME, JCP, MC)
- En Mecánica de Fluidos computacional muy poca cuenta se daba a este problema (**Tezduyar et al., EBE-M [1989]**). Recién cerca del año 2000 se comienza a estudiar la performance de los algoritmos DDM y se proponen nuevas formas de precondicionar. Antes de esto prácticamente solo se usaba **GMRes con 'scaling' diagonal y 'restarting'**. **Rachowicz, CMAME, 2000 usa Overlapping Schwarz-like methods.**

Problemas en la solución de grandes sistemas de ecuaciones

- 1 Los esquemas de solución **directa** (o exacta) de sistemas de ecuaciones lineales resultan muy acoplados para su solución en paralelo. La comunicación resulta excesiva cuando crece el número de incógnitas del problema (n_{dof} **grande**).
- 2 La convergencia de los métodos iterativos (tanto estacionarios como no estacionarios) sobre la matriz global (solución aproximada) se ve muy afectada si el sistema lineal está muy mal condicionado, típicamente cuando $h \rightarrow 0$. La cantidad de iteraciones para bajar el residuo depende de la distribución de autovalores y del número de condición espectral de la matriz.
- 3 Los métodos iterativos (Richardson, Jacobi, SOR, Gauß-Seidel, métodos de proyección de Krylov como CG y GMRes) sobre la matriz global resultan no tener buena **escalabilidad**.

Objetivos

- Objetivo principal: Solución eficiente (en cuanto a utilización de recursos computacionales y tiempo de CPU) en máquinas paralelas cuando se usan técnicas de discretización del tipo de Elementos Finitos, Volúmenes Finitos o Diferencias Finitas en problemas de CFD.
- Desarrollo e implementación de métodos y preconditionadores para la solución de sistemas lineales dentro de un código de producción orientado a objetos, que simula distintos problemas en CFD.
- Aplicación y evaluación del desempeño en problemas multifísica y en mecánica de fluidos computacional en general.

Temario

1 CIMEC

2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones

- Solución de Sistemas de Ecuaciones Lineales
- Método de Descomposición de Dominio (DDM)
- Precondicionamiento para el sistema $Ax = b$.

3 Parte II – Aplicaciones y Usos

- ESA CFD Code Evaluation Tests
- Condiciones de Borde Dinámicas en CFD
- Fluid-Structure Application Problems
- Descripción Aerodinámica de un Automóvil de Competición

4 Parte III – Conclusiones Finales

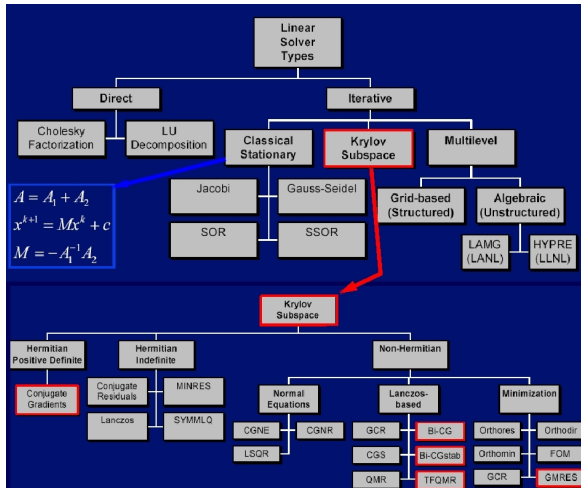
- Trabajos Publicados

Solución de Sistemas de Ecuaciones Lineales

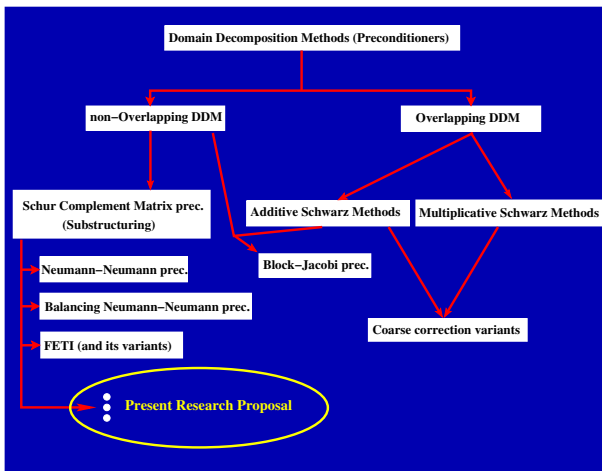
- Existen un gran número de métodos para resolver sistemas de ecuaciones lineales dependiendo principalmente del tipo de matrices entre manos.
- Básicamente hay **tres (o cinco)** grandes grupos:
 - 1 Métodos Directos: basados en la factorización LU y de Cholesky y todas sus variantes.
 - 2 Métodos Iterativos con Matriz de Iteración: Jacobi, SOR, SSOR, Richardson, etc. **“Métodos estacionarios”**.
 - 3 Métodos Iterativos en el Espacio de Krylov: Conjugate Gradients (CG), General Minimized Residuals (GMRes), etc. **“Métodos inestacionarios”**.
 - 4 **Métodos de Descomposición de Dominio: con o sin solapamiento entre subdominios.**
 - 5 **Métodos Multi-niveles (Multilevel): Multigrid y sus variantes.**

Solución de Sistemas de Ecuaciones Lineales

- Las técnicas mas usadas pueden resumirse:



Solución de Sistemas de Ecuaciones Lineales



Temario

1 CIMEC

2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones

- Solución de Sistemas de Ecuaciones Lineales
- **Método de Descomposición de Dominio (DDM)**
- Precondicionamiento para el sistema $Ax = b$.

3 Parte II – Aplicaciones y Usos

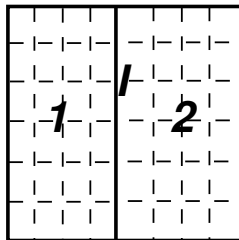
- ESA CFD Code Evaluation Tests
- Condiciones de Borde Dinámicas en CFD
- Fluid-Structure Application Problems
- Descripción Aerodinámica de un Automóvil de Competición

4 Parte III – Conclusiones Finales

- Trabajos Publicados

Descomposición de Dominio ‘No Solapado’ (non-Overlapping DDM)

$$\begin{bmatrix} \mathbf{A}_{11} & 0 & \mathbf{A}_{1/} \\ 0 & \mathbf{A}_{22} & \mathbf{A}_{2/} \\ \mathbf{A}_{/1} & \mathbf{A}_{/2} & \mathbf{A}_{//} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_/ \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_/ \end{bmatrix}$$



- La solución de $Ax = b$ es equivalente a la solución de

$$(\mathbf{A}_{//} - \mathbf{A}_{/1} \mathbf{A}_{11}^{-1} \mathbf{A}_{1/} - \mathbf{A}_{/2} \mathbf{A}_{22}^{-1} \mathbf{A}_{2/}) \mathbf{x}_/ = (\mathbf{b}_/ - \mathbf{A}_{/1} \mathbf{A}_{11}^{-1} \mathbf{b}_1 - \mathbf{A}_{/2} \mathbf{A}_{22}^{-1} \mathbf{b}_2)$$

$$\mathbf{S} \mathbf{x}_/ = \tilde{\mathbf{b}}_/, \quad \mathbf{S} \text{ matriz Complemento de Schur};$$

$$\mathbf{A}_{11} \mathbf{x}_1 = \mathbf{b}_1 - \mathbf{A}_{1/} \mathbf{x}_/;$$

$$\mathbf{A}_{22} \mathbf{x}_2 = \mathbf{b}_2 - \mathbf{A}_{2/} \mathbf{x}_/.$$

Ej.: Algoritmo Gradientes Conjugados Prec (A Hermitiana def. pos.)

Inicialización: $\mathbf{x}, \mathbf{r} = \mathbf{b} - \mathbf{Ax}$

- 1: resolver $\mathbf{Pz} = \mathbf{r}$;
- 2: $\rho = (\mathbf{r}, \mathbf{z}); \rho_0 = \rho$;
- 3: $\mathbf{p} = \mathbf{z}$;
- 4: $k = 1$;
- 5: **while** k to K_{\max} **do**
- 6: **if** $\rho < Tol\rho_0$ **then** {termina iteración}
- 7: {break}
- 8: **end if**
- 9: $\mathbf{a} = \mathbf{Ap}; m = (\mathbf{p}, \mathbf{a}); \alpha = \frac{\rho}{m}$;
- 10: $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}; \mathbf{r} = \mathbf{r} - \alpha\mathbf{a}$;
- 11: resolver $\mathbf{Pz} = \mathbf{r}$;
- 12: $\rho_{old} = \rho; \rho = (\mathbf{r}, \mathbf{z}); \gamma = \frac{\rho}{\rho_{old}}$;
- 13: $\mathbf{p} = \mathbf{z} + \gamma\mathbf{p}$;
- 14: $k = k + 1$;
- 15: **end while**

Método de Descomposición de Dominio (cont.)

- Condensar \mathbf{x}_1 y \mathbf{x}_2 para llegar a una ecuación en los *dof's* de la frontera:

$$\begin{aligned}
 (\mathbf{A}_{//} - \mathbf{A}_{/1} \mathbf{A}_{11}^{-1} \mathbf{A}_{1/} - \mathbf{A}_{/2} \mathbf{A}_{22}^{-1} \mathbf{A}_{2/}) \mathbf{x}_I \\
 = (\mathbf{b}_I - \mathbf{A}_{/1} \mathbf{A}_{11}^{-1} \mathbf{b}_1 - \mathbf{A}_{/2} \mathbf{A}_{22}^{-1} \mathbf{b}_2) \\
 \mathbf{S} \mathbf{x}_I = \tilde{\mathbf{b}}_I
 \end{aligned}$$

- Evaluar la ecuación $\mathbf{y}_I = \mathbf{S} \mathbf{x}_I$ implica:

- 1 Calcular $\mathbf{y} = \mathbf{A}_{//} \mathbf{x}$
- 2 Calcular $\mathbf{w} = \mathbf{A}_{/I} \mathbf{x}$
- 3 Resolver (en cada proc.)
 $\mathbf{A}_{LL} \mathbf{z} = \mathbf{w}$ para \mathbf{z}
- 4 Calcular $\mathbf{v} = \mathbf{A}_{/L} \mathbf{z}$
- 5 Sumar $\mathbf{y} \leftarrow \mathbf{y} + \mathbf{v}$

- Resolver las ecuaciones locales de equilibrio en cada procesador: por separado para \mathbf{x}_j : $\mathbf{A}_{jj} \mathbf{x}_j = -\mathbf{A}_{ji} \mathbf{x}_i$
- Sumar las contribuciones de la interfaz y locales: $\mathbf{y}_I = \mathbf{A}_{//} \mathbf{x}_I + \mathbf{A}_{/1} \mathbf{x}_1 + \mathbf{A}_{/2} \mathbf{x}_2$
- Nos referiremos a este método como **IISD** (“Interface Iterative - SubDomain Direct”).

DDM vs. iteración global - Algunas propiedades características

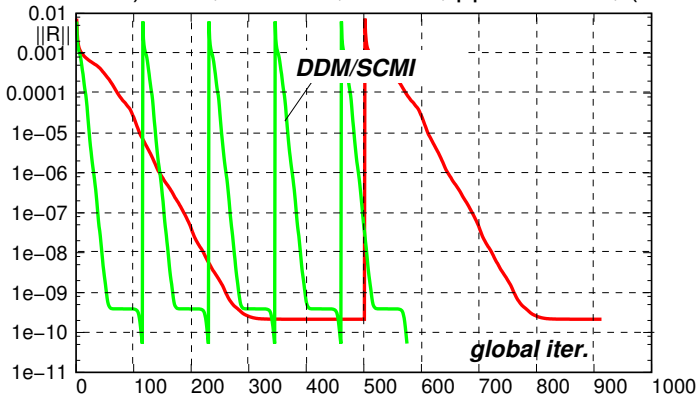
- Iterar sobre la matriz Complemento de Schur (sistema condensado) es equivalente a iterar en un subespacio donde las ecuaciones de los grados de libertad locales están en equilibrio.
- La **tasa de convergencia mejora** 😊 debido a que:
 - el **número de condición espectral** de la matriz Complemento de Schur **es menor**,
 - **la dimensión del espacio donde se itera es menor** 😊 (métodos 'no estacionarios' como CG/GMRes se aceleran a medida que las iteraciones prosiguen). Sin embargo **esto es compensado** por el tiempo consumido por la **factorización de los probl. locales y las sustituciones hacia adelante y hacia atrás** 😞.

DDM vs. iteración global - Algunas propiedades características (cont.)

- Debido a que la cantidad de iteraciones resulta menor y que el espacio de iteración es significativamente menor, los requerimientos de memoria RAM para almacenar los vectores que generan el espacio de Krylov resulta menor 😊. Pero esto está compensado por la memoria necesaria para la factorización de las matrices internas $LU(\mathbf{A}_{jj})$ 😞.
- Un mejor condicionamiento de la matriz compl. de Schur previene el 'breakdown' de GMRes debido a la pérdida de ortogonalidad de los vectores de la base de Krylov 😊.
- Debido a que el tiempo de CPU de GMRes es cuadrático en función del número de iteraciones (paso de ortogonalización) y que iterar sobre la matriz global requiere a menudo mayor número de iteraciones, iterar sobre la matriz Comp. de Schur es comparativamente mejor para alcanzar tolerancias del residuo mas bajas 😊.

Ej. Navier-Stokes cubic cavity. $Re=1000$.

625,000 tetras mesh, $rtol=10^{-4}$, NS monolithic, [Tezduyar et.al (SUPG+PSPG) stab., CMAME, vol. 95, pp. 221-242, (1992)]



Ej. NS cubic cavity $Re=1000$ (cont.)

- Cada iteración de IISD requiere mayor tiempo, pero finalmente en promedio tenemos: para el avance en un paso de tiempo (3 iteraciones en el lazo no lineal)

tiempo de CPU en IISD = 17.7 secs,

tiempo de CPU iteración Global = 63.8 secs.

- Los Residuos son los del problema de la interfaz para IISD, y globales para iteración Global . El vector de residuos para IISD es equivalente a un vector global con residuo nulo en los grados de libertad internos (son entonces comparables).

Temario

- 1 CIMEC
- 2 **Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones**
 - Solución de Sistemas de Ecuaciones Lineales
 - Método de Descomposición de Dominio (DDM)
 - **Precondicionamiento para el sistema $Ax = b$.**
- 3 Parte II – Aplicaciones y Usos
 - ESA CFD Code Evaluation Tests
 - Condiciones de Borde Dinámicas en CFD
 - Fluid-Structure Application Problems
 - Descripción Aerodinámica de un Automóvil de Competición
- 4 Parte III – Conclusiones Finales
 - Trabajos Publicados

Precondicionamiento

- La tasa de convergencia del residuo en los métodos iterativos depende del **espectro** y el **número de condición** de la matriz.
- El mal condicionamiento de la matriz puede provenir de varias fuentes: refinamiento ($h \rightarrow 0$), discontinuidades fuertes en los parámetros que gobiernan al problema, altas relaciones de aspecto en las direcciones características del problema, mala imposición de condiciones de borde que generan capas límites o discontinuidades, imposición de restricciones con multiplicadores de Lagrange, etc.
- Por ej., para el operador de Laplace (o problema de Poisson) en un cuadrado unitario discretizado con diferencias finitas o elementos finitos lineales el número de condición espectral es $\kappa(A) = \|A\| \|A^{-1}\| = O(h^{-2})$. Siendo $\kappa(S) = O(h^{-1})$. **Esto resulta no ser suficiente.**

Precondicionamiento (cont.)

- Para mejorar más aún la convergencia en las iteraciones sobre la matriz complemento de Schur se propone el uso de preconditionadores para este sistema.

Definition

- Se propone entonces resolver un problema equivalente:

$$M^{-1}Sx_l = M^{-1}\tilde{b}_l \quad \text{Precondicionamiento a izquierda}$$

- La solución es la misma que la del sistema $Sx_l = \tilde{b}_l$, pero las propiedades espectrales de la matriz de coeficientes $M^{-1}S$ resultan ser más “favorables” para su solución iterativa.

Algoritmo Gradientes Conjugados Prec (A Hermitiana def. pos.)

Inicialización: $\mathbf{x}, \mathbf{r} = \mathbf{b} - \mathbf{Ax}$

- 1: resolver $\mathbf{Pz} = \mathbf{r}$;
- 2: $\rho = (\mathbf{r}, \mathbf{z}); \rho_0 = \rho$;
- 3: $\mathbf{p} = \mathbf{z}$;
- 4: $k = 1$;
- 5: **while** k to K_{\max} **do**
- 6: **if** $\rho < Tol\rho_0$ **then** {termina iteración}
- 7: {break}
- 8: **end if**
- 9: $\mathbf{a} = \mathbf{Ap}; m = (\mathbf{p}, \mathbf{a}); \alpha = \frac{\rho}{m}$;
- 10: $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}; \mathbf{r} = \mathbf{r} - \alpha\mathbf{a}$;
- 11: resolver $\mathbf{Pz} = \mathbf{r}$;
- 12: $\rho_{old} = \rho; \rho = (\mathbf{r}, \mathbf{z}); \gamma = \frac{\rho}{\rho_{old}}$;
- 13: $\mathbf{p} = \mathbf{z} + \gamma\mathbf{p}$;
- 14: $k = k + 1$;
- 15: **end while**

Precondicionamiento (cont.)

- En general para el sistema global existen varias formas de precondicionamiento:
 - ① **Precondicionamiento Algebraico:** point-Jacobi o escalamiento diagonal, ILU, Block-Jacobi, Additive Schwarz, etc.
 - ② Precondicionamiento Funcional: wire-basket preconditioners (Bramble, Pasciak, Schatz [1986,1989])
 - ③ Precondicionamiento Polinómico: Dubois, Greenbaum y Rodrigue [1979].
 - ④ **Métodos de Descomposición de Dominio:** Neumann-Neumann (Aghoskov [1988]), Balancing Neumann-Neumann (Mandel [1993]), FETI (Farhat [1991]), wire-basket prec., Block-Jacobi, Overlapping Additive Schwarz, ... (**presente tesis**).
 - ⑤ **Métodos Multigrilla:** Astrakhantsev [1971], Nicolaidis [1875], Brandt [1977], Hackbusch [1977], Wesseling [1978]

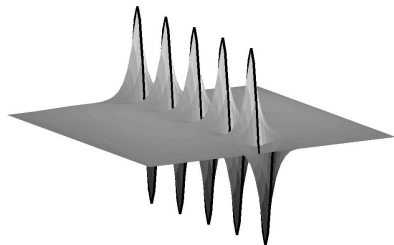
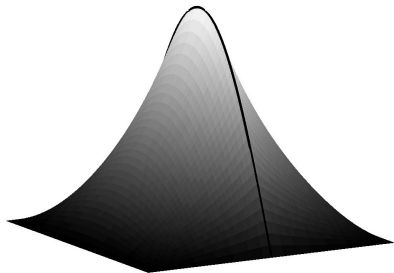
...

Precondicionamiento del problema de Interfaz. Operador de Steklov-Poincaré

- Es claro que conocer el espectro de **autovalores y autofunciones** del operador de Steklov-Poincaré (version continua de la matriz complemento de Schur) es la base para el desarrollo de preconditionadores eficientes para el problema de interfaz.
- Para el problema de Poisson en un dominio cuadrado ($0 \leq (x, y) \leq 1$) con 2 subdominios (long L_1 y L_2) las autofunciones y autovalores del **operador de Steklov-Poincaré \mathcal{S}** son

$$\begin{aligned}\psi_n(y) &= \sin(k_n y) \\ \omega_n &= k_n [\coth(k_n L_1) + \coth(k_n L_2)]\end{aligned}$$

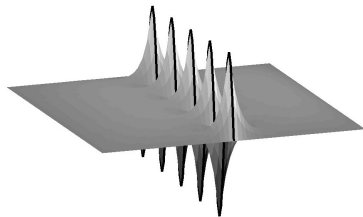
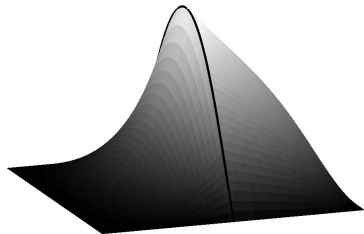
k_n es el número de onda de la n -ésima autofunción.

1^{ra} y 10^{ma} autofunción

- Se puede demostrar que $\text{eig}(S) \propto n$ con n grande y $\kappa(S) \rightarrow \infty$.
- Pero con $\omega_{\max} = 2\pi/h$, entonces $k_{\max} = \omega_{\max}/2 = \pi/h$,
- $\kappa(S) \propto 1/h$.

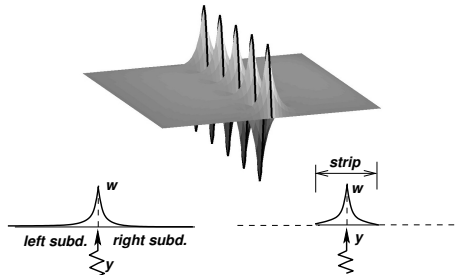
Flux splitting (caso advectivo) 1^{ra} y 10^{ma} autofunción

- En presencia de operadores con términos advectivos las autofunciones dejan de ser simétricas. La separación de los flujos en la interfaz es desviado aguas abajo del subdominio.



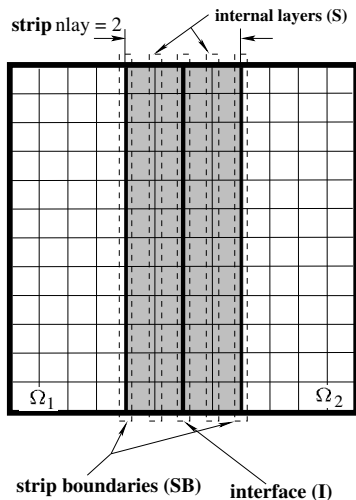
Prec. de Banda Alrededor de la Interface (**Interface Strip Prec.**)

- Para los modos de alta frecuencia (autovalores altos) las autofunciones del operador de Steklov-Poincaré se concentran alrededor de la interface. Las autofunciones decaen según $\exp(-k_n|s|)$.



- El preconditionador de banda alrededor de la interfaz resuelve el problema en una fina banda alrededor de esta.

Interface Strip Preconditioner (cont.)



Considerando el problema de la figura, el precondicionamiento consiste en, dado un vector f_I definido sobre los nodos en I , calcular una solución aproximada v_I dada por

Interface Strip Preconditioner (cont.)

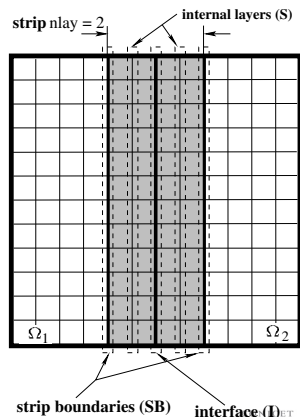
$$\begin{bmatrix} A_{II} & A_{IS} & A_{I,SB} \\ A_{SI} & A_{SS} & A_{S,SB} \\ A_{SB,I} & A_{SB,S} & A_{SB,SB} \end{bmatrix} \begin{bmatrix} v_I \\ v_S \\ v_{SB} \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \\ 0 \end{bmatrix},$$

con “condiciones de borde Dirichlet” sobre $I / v_{SB} = 0$ (ya que las autofunc. decaen según $\exp(-k_n|s|)$). Por lo tanto el sistema se reduce a

$$\begin{bmatrix} A_{II} & A_{IS} \\ A_{SI} & A_{SS} \end{bmatrix} \begin{bmatrix} v_I \\ v_S \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \end{bmatrix},$$

Una vez resuelto, v_I es el valor del preconditionador propuesto aplicado a f_I , i.e.

$$v_I = \mathcal{P}_{ISP}^{-1} f_I$$



Resultados Teóricos para advección lineal escalar, malla: 50x50, Pe=50

u	$\text{cond}(S)$	$\text{cond}(\mathcal{P}_{\text{NN}}^{-1}S)$	$\text{cond}(\mathcal{P}_{\text{ISP}}^{-1}S)$
0	41.00	1.00	4.92
1	40.86	1.02	4.88
10	23.81	3.44	2.92
50	5.62	64.20	1.08

Table: Número de Condición para el operador de Steklov-Poincaré en una malla de 50×50 elementos.

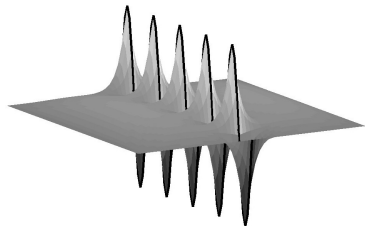
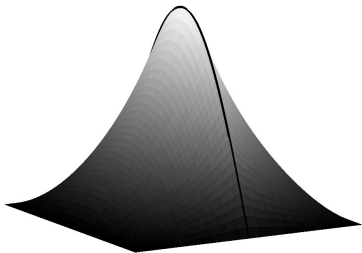
Resultados teóricos para advección lineal escalar, malla: 100x100, $Pe=50$

u	$\text{cond}(S)$	$\text{cond}(\mathcal{P}_{NN}^{-1}S)$	$\text{cond}(\mathcal{P}_{ISP}^{-1}S)$
0	88.50	1.00	4.92
1	81.80	1.02	4.88
10	47.63	3.44	2.92
50	11.23	64.20	1.08

Table: Número de Condición para el operador de Steklov-Poincaré en una malla de 100×100 elementos.

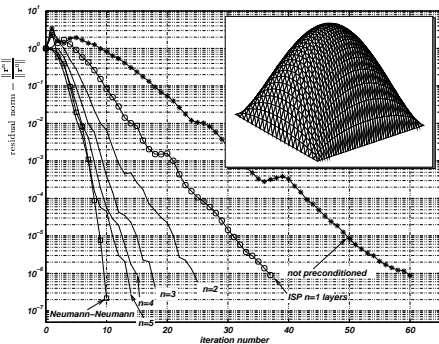
Flux splitting - Precondicionador de Neumann-Neumann

- El prec. de Neumann-Neumann es efectivo cuando el **splitting simétrico** de los flujos resulta ser el correcto: subdominio igual subdominio derecho, operador simétrico. (Las autofunciones del operador de Steklov-Poincaré son simétricas.)

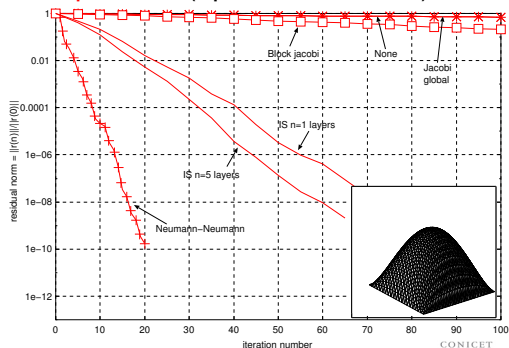


Resultados numéricos para Poisson

test secuencial 120x120 el

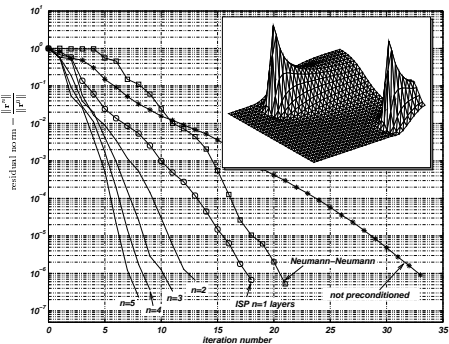


test paralelo (np=4, 500x500 el)

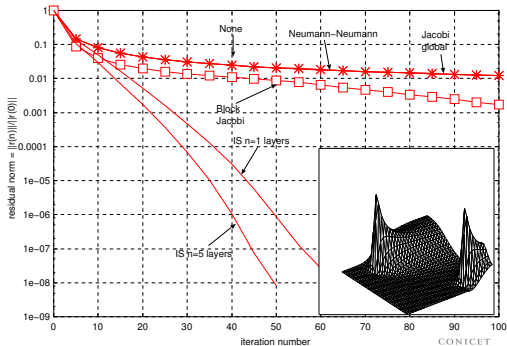


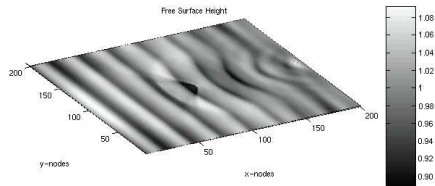
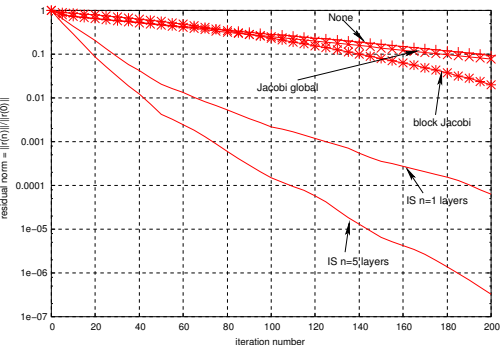
Resultados numéricos para advección escalar: $Pe=25$

test secuencial 120x120 el



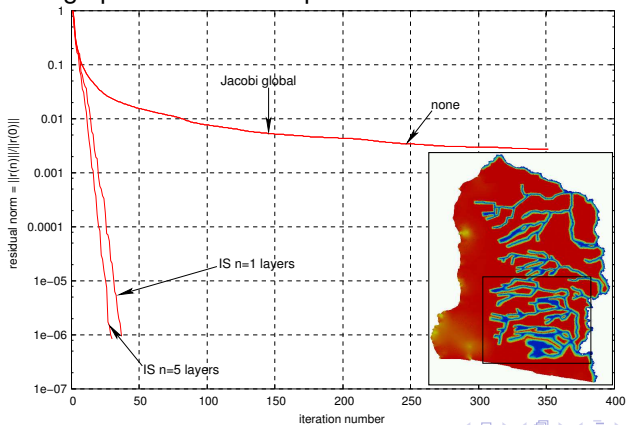
test paralelo (np=4, 500x500 el)



Desempeño en Shallow-water 2D (malla 500×500 elem.)

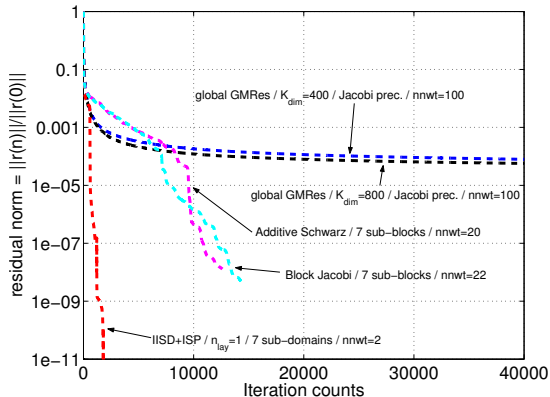
Test: Interacción Río-Acuífero

Shallow-Water 1D/Groundwater interaction over several basins in Sta Fe. 32900Km², 1.65M triangles, 9 subdomains (one per processor).
Annual average periodical raindrop.



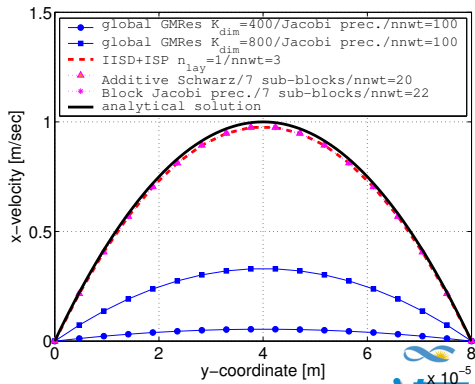
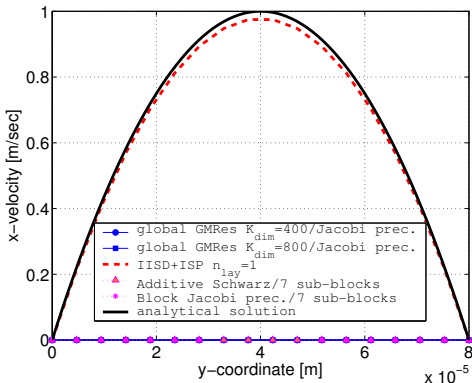
Test: Flujo de Stokes en un canal horizontal largo

Flujo estacionario en un canal con dimensiones características muy disímiles ($H = 8 \cdot 10^{-5} \text{m}$, $L = 9 \cdot 10^{-2} \text{m}$) a muy bajo (pero no nulo) número de Reynolds ($Re = 0.1$). 16 proc. 210000 dof's



Flujo de Stokes en un canal horizontal: Velocidad

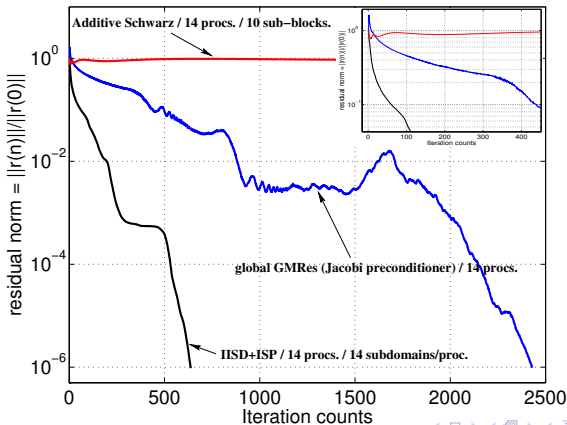
IISD+ISP consume 50-55% menos de tiempo de CPU y memoria por iteración no lineal que los demás DDMs.



Test: Flujo en una cavidad cuadrada (Fractional Step Scheme).

- Solución del Paso de Poisson. $Re = 1000$.

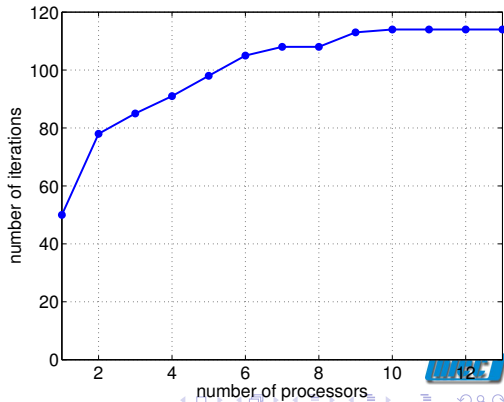
$(\Delta t \mathbf{DM}^{-1} \mathbf{G}(\mathbf{P}^{n+1} - \gamma \mathbf{P}^n) = \mathbf{D}\hat{\mathbf{U}}^{n+1})$ en una malla de 400×400 en 14 procs. (resultados: Residuos $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$).



Escalabilidad Práctica de ISP

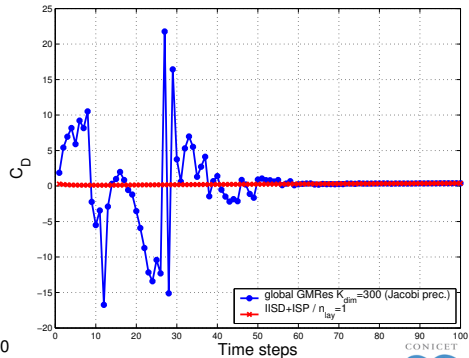
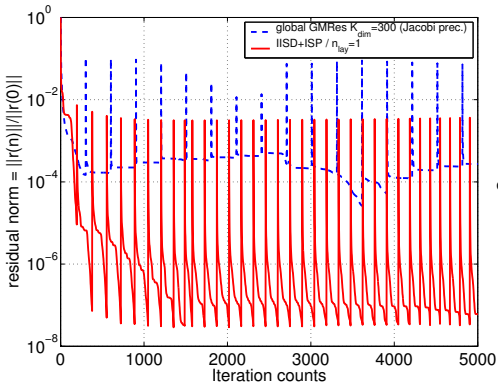
- **Escalabilidad:** El número de iteraciones para alcanzar una tolerancia dada permanece constante a medida que el tamaño del problema global y el número de procesadores aumenta en la misma proporción (tamaño de problema fijo en cada procesador). ‘Escalabilidad de Gustafson’.

- Cavity cuadrada, malla:
 $20n_{\text{proc}} \times 20n_{\text{proc}}$ elementos bilineales.
- Flujo de Stokes Monolítico
 $Re = 0.01$.
- 4 subdominios/proc. y $n_{\text{lay}} = 1$.
- Tolerancia 10^{-8} .



Flujo LES alrededor del cuerpo de AHMED a $Re = 1000$

Malla: 1.5Mtetras. 15 procs.



Conclusiones

- Fueron derivados resultados analíticos para presentar la base matemática y propiedades de convergencia de IISD+ISP.
- El preconditionador propuesto es eficiente para flujos a altos Reynolds donde la contribución de los términos advectivos es predominante mientras es capaz de manejar también eficientemente flujos dominados por términos difusivos. **Especialmente diseñado para CFD** (en contraposición a NN).
- IISD+ISP es una buena alternativa (tiempo de CPU y memoria) para tratar problemas con grandes gradientes de refinamiento (e.g., capas límites, ondas de choque, discontinuidades de contacto, geometrías irregulares)
- El preconditionador ISP es fácil de construir, no requiere ningún cálculo especial (puede ensamblarse a partir del grafo de la matriz de coeficientes).
- Se demostró prácticamente que requiere menos memoria y tiempo de CPU que los clásicos preconditionadores y solvers.
- Permite decidir la cantidad de recursos que se asignaran para la etapa de preconditionamiento (**'variable strip'**).
- **Se debe verificar la escalabilidad de ISP en clusters con mayor número de proc.**
- *"... at this very moment the search is on, every numerical analyst has a favorite preconditioner, and you have a perfect chance to find a better one. - Gil Strang [1986]. ... que sea de propósito general, algebraico, fácil de implementar, etc.*

Temario

- 1 CIMEC
- 2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones
 - Solución de Sistemas de Ecuaciones Lineales
 - Método de Descomposición de Dominio (DDM)
 - Precondicionamiento para el sistema $Ax = b$.
- 3 **Parte II – Aplicaciones y Usos**
 - **ESA CFD Code Evaluation Tests**
 - Condiciones de Borde Dinámicas en CFD
 - Fluid-Structure Application Problems
 - Descripción Aerodinámica de un Automóvil de Competición
- 4 Parte III – Conclusiones Finales
 - Trabajos Publicados

Three tests proposed by **ESA (ESTEC)/OPEN ENGINEERING** to evaluate three CFD codes for the ‘strong *Fluid-Structure Interaction* project’ at hypersonic regime ($M \geq 5$):

- **AeroULG-3D** code from Université de Liège, Belgium,
- **FINE/HEXA-3D** code by Numeca (C. Hirsch), Belgium, and
- **PETSc-FEM** code from CIMEC-INTEC, Argentina.
(<http://www.cimec.org.ar/petscfem>).

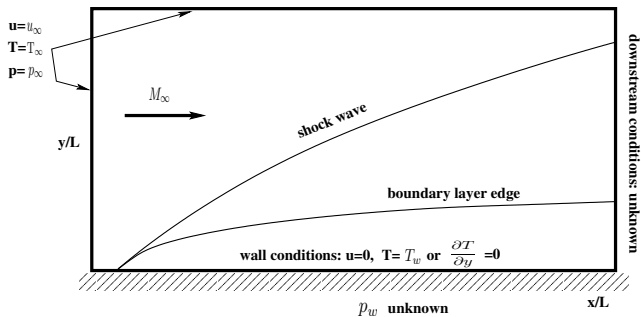
These tests involved different numerical techniques, namely:

- Parallel Solution of **'Inviscid/Viscous Hypersonic Flows'** in **'Beowulf'** clusters of PC's.
- The Galerkin/SUPG formulation with added **'isotropic/anisotropic shock capturing'** operators.
- **'Adaptive mesh refinement'** techniques.
- The use of **'non-reflecting'** boundary conditions on fictitious (and subsonic) walls.
- Finally, a **'Domain Decomposition'** preconditioner for an efficient solution of the linear system.

TEST 1: viscous flow over a flat plate at $M=5$.

$$\begin{aligned}
 M_{\infty} &= 5 \\
 T_{\infty} &= 80\text{K} \\
 \rho_{\infty} &= 10^5\text{Pa} \\
 Re_{\infty} &= 10^4 \\
 R &= 287\text{J/kgK} \\
 T_{wall} &= 288\text{K} \\
 \gamma &= 1.4 \\
 Pr &= 0.72 \\
 \mu &= 2.5 \cdot 10^{-5}\text{ kg/m sec} \\
 \kappa &= 3.47 \cdot 10^{-5}\text{ W/mK.}
 \end{aligned}$$

free stream conditions



Results: Mach number distribution.

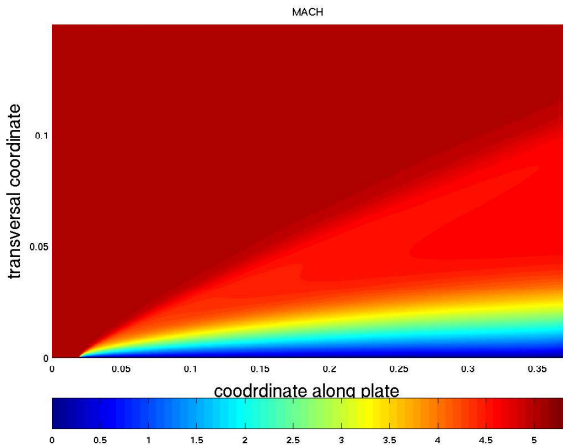
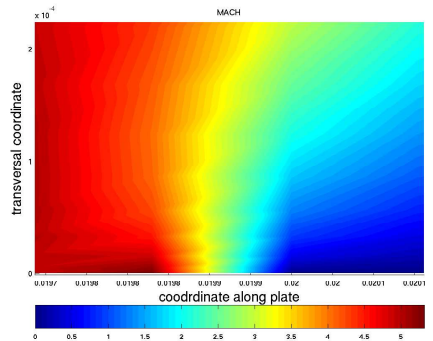
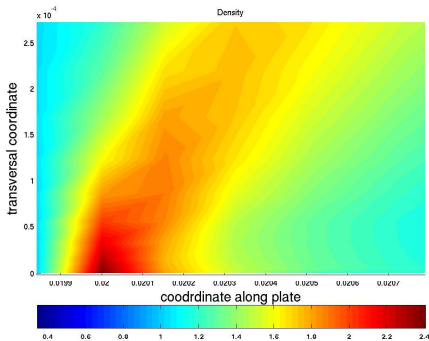


Figure: Mach number distribution.

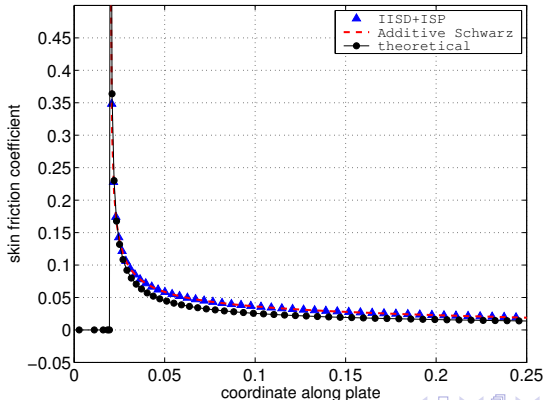
Results: stagnation point



Results: skin friction coefficient.

Definition: $C_f = \tau_{\text{wall}} / (0.5 \rho_{\infty} \|\mathbf{u}_{\infty}\|^2)$.

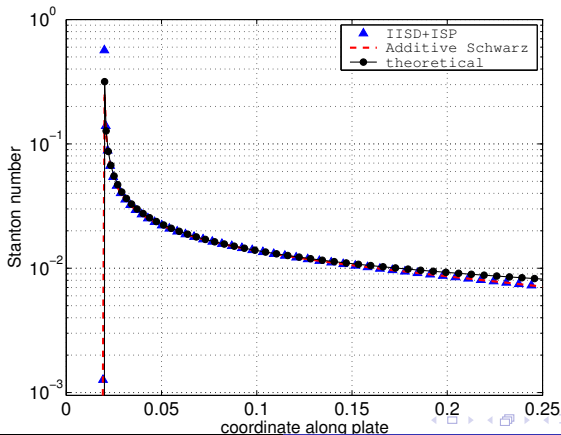
Theoretical: $C_f = 0.664 \left(\frac{T_{\text{wall}}}{T_{\infty}} \right)^{-0.5(1-\omega)} \left(\frac{P_{\text{wall}}}{\rho_{\infty} \text{Re}_x} \right)^{0.5}$, $\omega = 0.75$.



Results: Stanton number.

Definition: $St = \frac{q_{wall}}{\rho_{\infty} U_{\infty} C_p |T_{wall} - T_{\infty}^0|}$.

Theoretical: $St = 0.5 Pr^{-2/3} C_f$.



TEST 2: compression corner viscous flow at M=5

$$\alpha = 15^\circ$$

$$M_\infty = 5$$

$$T_\infty = 80\text{K}$$

$$\rho_\infty = 10^5\text{Pa}$$

$$Re_\infty = 1.510^6$$

$$R = 287\text{J/kgK}$$

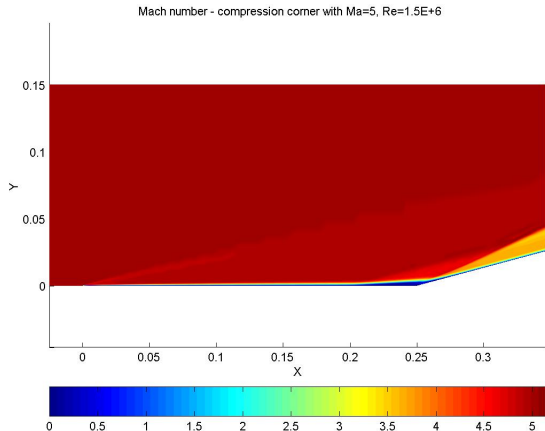
$$T_{wall} = 288\text{K}$$

$$\gamma = 1.4$$

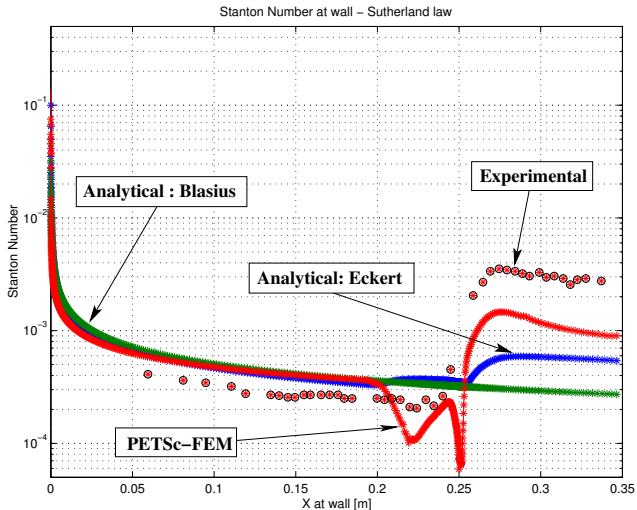
$$Pr = 0.72$$

μ Sutherland form.

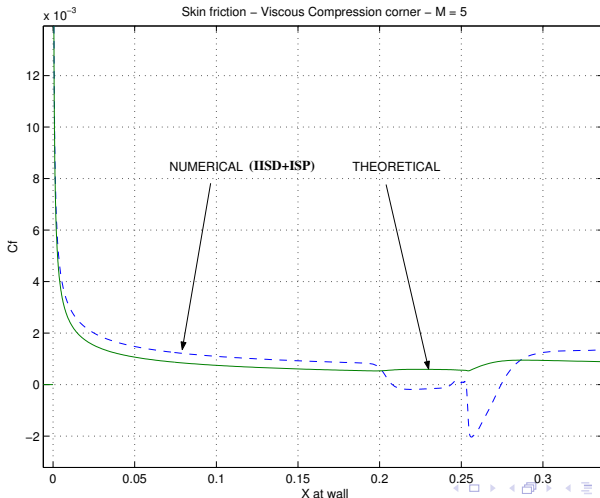
κ Sutherland form.



Results: Stanton number.



Results: Skin friction coefficient.



Inviscid Case: Mach=10. Results.

	p/p_∞	T/T_∞	Mach after shock	shock angle
Present Result	13.482	3.166	5.305	20.37°
Theoretical	13.404	3.194	5.279	19.941°
Relative error	0.6%	0.9%	0.5%	2.1%

Table: comparison between the analytical reference results and the computed results.

Shock wave propagation in a Nozzle.

$$\mathbf{u}_{t<0, \forall x} = 0 \text{ m/s}$$

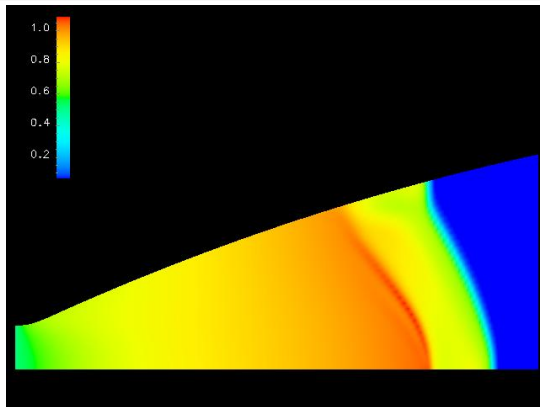
$$p_{t<0, \forall x} = 143 \text{ Pa}$$

$$T_{t<0, \forall x} = 262 \text{ K}$$

$$\rho_{t=0, x=0} = 600000 \text{ Pa}$$

$$T_{t=0, x=0} = 4170 \text{ K}$$

$$\gamma = 1.17$$



Measured wave velocity: 2600-2700m/s

Computed wave velocity: 2620.5m/s.

Temario

- 1 CIMEC
- 2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones
 - Solución de Sistemas de Ecuaciones Lineales
 - Método de Descomposición de Dominio (DDM)
 - Precondicionamiento para el sistema $Ax = b$.
- 3 **Parte II – Aplicaciones y Usos**
 - ESA CFD Code Evaluation Tests
 - **Condiciones de Borde Dinámicas en CFD**
 - Fluid-Structure Application Problems
 - Descripción Aerodinámica de un Automóvil de Competición
- 4 Parte III – Conclusiones Finales
 - Trabajos Publicados

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.**

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

$$\mathbf{A}_{\text{adv},j} = \frac{\partial \mathcal{F}_{\text{adv},j}(\mathbf{U})}{\partial \mathbf{U}} \quad \text{advective Jacobian}$$

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

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

- Adding extra Dirichlet conditions leads to spurious shocks, and lack of a Dirichlet conditions leads to instability (ill-conditioned problem).**

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.

Absorbing boundary conditions for 1D advective case

- Being the characteristic form $\frac{\partial v_k}{\partial t} + \lambda_k \frac{\partial v_k}{\partial x} = 0$, and assuming $\lambda_k \neq 0$, the absorbing boundary conditions are, depending on the sign of λ_k ,

if $\lambda_k > 0$: $v_k(0) = \bar{v}_{k0}$; **no boundary condition** at $x = L$

if $\lambda_k < 0$: $v_k(L) = \bar{v}_{kL}$; **no boundary condition** at $x = 0$

$$\Pi_V^+(\mathbf{V} - \bar{\mathbf{V}}_0) = 0; \quad \text{at } x = 0$$

$$\Pi_V^-(\mathbf{V} - \bar{\mathbf{V}}_L) = 0; \quad \text{at } x = L$$

where Π_V^\pm are the *projection operators onto the right/left-going characteristic modes* in the \mathbf{V} basis,

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

$$\Pi^\pm \Pi^\pm = \Pi^\pm, \quad \Pi^+ \Pi^- = 0, \quad \Pi^+ + \Pi^- = \mathbf{I}.$$

Absorbing Boundary Conditions

- In general, first order absorbing boundary conditions may be constructed by imposing exactly the components along the incoming characteristics.

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

Π^- is the projection operator onto incoming characteristics. It can be obtained from the projected Jacobian.

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

Absorbing Boundary Conditions

The problem stated is: 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., not always known).

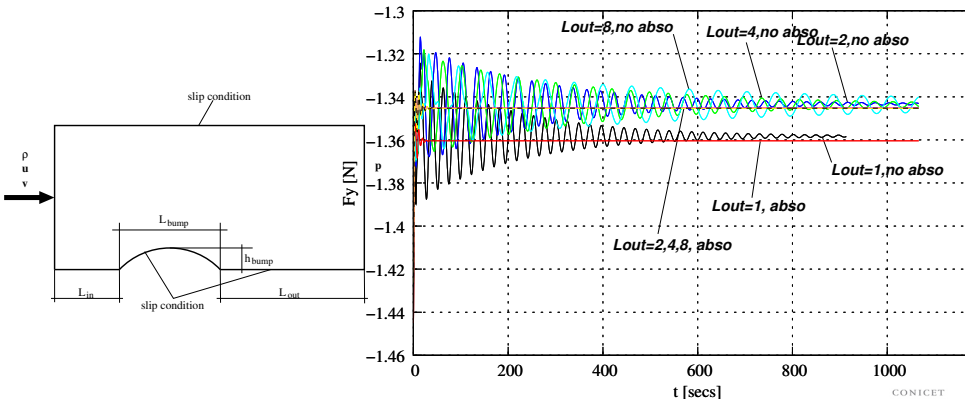
Solution: Use last state as reference state, ULSAR.

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

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

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

Viscous compressible subsonic flow over a parabolic bump. $M_\infty = 0.5$

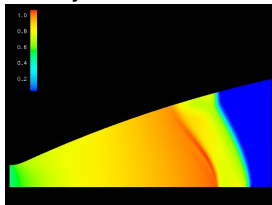


Dynamic boundary conditions

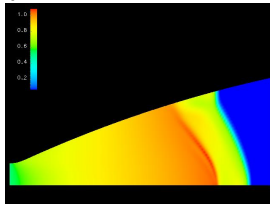
- Finally, there is a last twist in the use of absorbing boundary conditions.
- As the flow is computed it may happen that the number of characteristics **to impose** changes in time, i.e., it involves the change of the structure of the Jacobian matrix
- Two typical examples:

Ignition of a rocket nozzle in a low pressure atmosphere

The fluid is initially at rest (143 Pa, 262 °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 °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 msec) and the final steady flow.

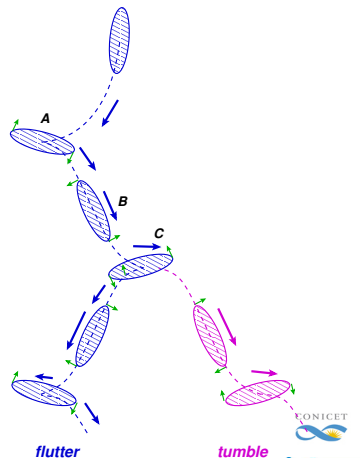
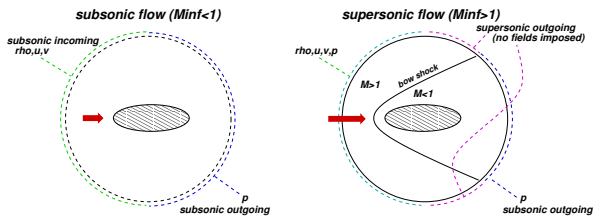


- We impose density, pressure and tangential velocity at inlet (assuming subsonic inlet), slip condition at the nozzle wall. The problem is with the outlet boundary. Initially the flow is subsonic (fluid at rest) there, and switches to supersonic. The rule states to impose 1 condition, as a subsonic 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.



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

Object falling at supersonic speed



Solution

Dynamic Boundary Conditions

- If the number of incoming/outgoing characteristics vary in time this requires to change the profile of the system matrix during time evolution (RI or ULSAR based boundary conditions).
- In order to treat this problem we add dynamic boundary conditions via Lagrange Multipliers (or penalization).
- However this techniques add extra bad conditioning to the system of equations so that special iterative methods are needed.

Example: 1D Linear Advective Case

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

Consider for simplicity a linear system of advective equations discretized with Galerkin-FEM with mass lumping,

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

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

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

Lagrange Multipliers in 'characteristics' basis

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

$$\begin{aligned} \Pi_V^+(\mathbf{V}_{\text{ref}}) (\mathbf{V}_0 - \mathbf{V}_{\text{ref}}) + \Pi_V^-(\mathbf{V}_{\text{ref}}) \mathbf{V}_{1m} &= 0, \\ \frac{\mathbf{V}_0^{n+1} - \mathbf{V}_0^n}{\Delta t} + \Lambda \frac{\mathbf{V}_1^n - \mathbf{V}_0^n}{h} + \Pi_V^+(\mathbf{V}_{\text{ref}}) \mathbf{V}_{1m} &= 0; \\ \frac{\mathbf{V}_k^{n+1} - \mathbf{V}_k^n}{\Delta t} + \Lambda \frac{\mathbf{V}_{k+1}^n - \mathbf{V}_{k-1}^n}{2h} &= 0, \quad k \geq 1. \end{aligned}$$

Π^\pm is the projection operator onto incoming/outgoing waves,
 \mathbf{V}_{1m} are the Lagrange multipliers.

1D advection example in ‘characteristics’ basis - Boundary at $x=0$

In characteristics basis, if j is an incoming wave ($\lambda_j \geq 0$)

$$v_{j0}^{n+1} - v_{j0}^n + \lambda_j \frac{v_{j1}^n - v_{j0}^n}{h} + v_{j,1m} = 0$$

$$\frac{v_{jk}^{n+1} - v_{jk}^n}{\Delta t} + \lambda_j \frac{v_{j,k+1}^n - v_{jk}^n}{2h} = 0,$$

Due to the $v_{j,1m}$ Lagrange multiplier, we can solve for the v_{jk} values from the first and last rows, while the value of the multiplier $v_{j,1m}$ “adjusts” itself in order to satisfy the equations in the second row.

On the other hand, for the outgoing waves ($\lambda_j < 0$), we have

$$\frac{v_{j0}^{n+1} - v_{j0}^n}{\Delta t} + \lambda_j \frac{v_{j1}^n - v_{j0}^n}{h} = 0$$

$$\frac{v_{jk}^{n+1} - v_{jk}^n}{\Delta t} + \lambda_j \frac{v_{j,k+1}^n - v_{jk}^n}{2h} = 0,$$

So that the solution coincides with the unmodified original FEM equation, and the Lagrange multiplier is $v_{j,1m} = 0$.

Coming back to the \mathbf{U} basis ($\mathbf{V} = \mathbf{S}^{-1}\mathbf{U}$), we have

$$\begin{aligned} & \mathbf{n}_{U}^{+}(\mathbf{U}_{\text{ref}}) (\mathbf{U}_0 - \mathbf{U}_{\text{ref}}) + \mathbf{n}_{U}^{-}(\mathbf{U}_{\text{ref}}) \mathbf{U}_{1m} = 0, \\ & \mathbf{C} \frac{\mathbf{U}_0^{n+1} - \mathbf{U}_0^n}{\Delta t} + \mathbf{A} \frac{\mathbf{U}_1^n - \mathbf{U}_0^n}{h} + \mathbf{C} \mathbf{n}_{U}^{+}(\mathbf{U}_{\text{ref}}) \mathbf{U}_{1m} = 0; \\ & \mathbf{C} \frac{\mathbf{U}_k^{n+1} - \mathbf{U}_k^n}{\Delta t} + \mathbf{A} \frac{\mathbf{U}_{k+1}^n - \mathbf{U}_{k-1}^n}{2h} = 0, \quad k \geq 1. \end{aligned}$$

Absorbing boundary conditions and ALE

When using *Arbitrary Lagrangian-Eulerian* formulations:

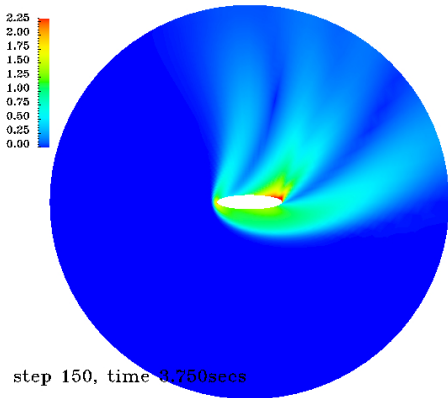
$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathcal{F}_{\text{adv},j}(\mathbf{U})}{\partial x_j} - \mathbf{v}_{\text{mesh}} \mathbf{U} = 0$$

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

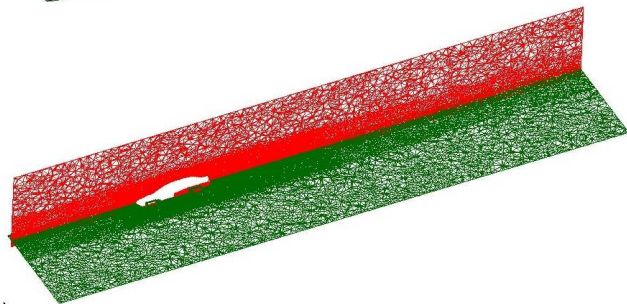
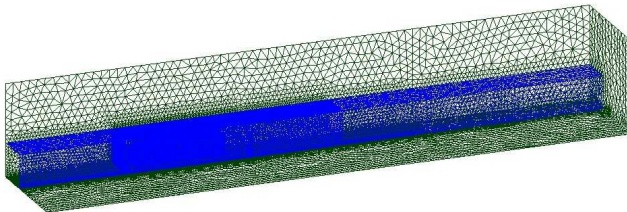
$$\text{Nbr. of incoming characteristics} = \text{sum}(\text{eig}(\mathbf{A} \cdot \hat{\mathbf{n}}) - \mathbf{v}_{\text{mesh}} \cdot \hat{\mathbf{n}} < 0)$$

Supersonic falling of an Ellipse

- $a = 1$, $b = 0.6$ (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 $\gamma = C_p/C_v$),
- $R_{\text{ext}} = 10$, (Radius of the fictitious boundary),
- $\mathbf{u}_{\text{ini}} = [0, 0, 1.39, 0, 1.3, 0]$, (ellipse initial position and velocity $[x, y, \alpha, u, v, \dot{\alpha}]$),



Future Work: Absorbent B.C. for Incompressible Flow

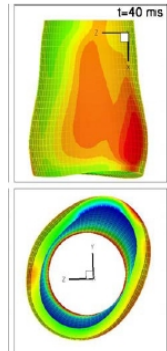


Temario

- 1 CIMEC
- 2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones
 - Solución de Sistemas de Ecuaciones Lineales
 - Método de Descomposición de Dominio (DDM)
 - Precondicionamiento para el sistema $Ax = b$.
- 3 **Parte II – Aplicaciones y Usos**
 - ESA CFD Code Evaluation Tests
 - Condiciones de Borde Dinámicas en CFD
 - **Fluid-Structure Application Problems**
 - Descripción Aerodinámica de un Automóvil de Competición
- 4 Parte III – Conclusiones Finales
 - Trabajos Publicados

Motivation - Introduction

- One of the three tests proposed by “CIMEC CFD Team” to the E.S.A. (ESTEC)/OE to evaluate two CFD codes for its Strong Fluid-Structure Interaction project at supersonic/hypersonic regime:



Monolithic vs. Partitioned Alg.

- For simple structural problems with few vibrational degrees of freedom it is possible to combine the fluid and the structure in a single formulation. Then the full system can be integrated with a explicit or implicit scheme. These “**monolithic**” methods can be very robust but are in general not modular and parallel efficiency is difficult to reach.
- 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.

Basic (weakly coupled) partitioned FSI algorithm

- (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.

Staged (strongly coupled) partitioned FSI algorithm

FSI Algorithm

States: **u** structure, **w** fluid, **X** mesh.

- 1: Initialize variables:
- 2: **for** $n = 0$ to n_{step} **do** {Main time step loop}
- 3: $t^n = n\Delta t$, $n_{\text{stage}} = 0$
- 4: $\mathbf{X}^n = \text{CMD}(\mathbf{u}^n)$ {run CMD code}
- 5: $\mathbf{u}^{(n+1)P} = \mathbf{u}^{(n+1,0)} = \text{predictor}(\mathbf{u}^n, \mathbf{u}^{n-1})$ {compute predictor}
- 6: **while** $\text{tol} \leq \tau_c$ **do** {stage loop (n_{stage})}
- 7: $\mathbf{X}^{n+1,i+1} = \text{CMD}(\mathbf{u}^{n+1,i})$
- 8: $\mathbf{w}^{n+1,i+1} = \text{CFD}(\mathbf{w}^n, \mathbf{X}^{n+1,i+1}, \mathbf{X}^n)$ {Fluid solver CFD}
- 9: compute structural loads ($\mathbf{w}^n, \mathbf{w}^{n+1,i+1}$)
- 10: $\mathbf{u}^{n+1,i+1} = \text{CSD}(\mathbf{u}^n, \mathbf{w}^n, \mathbf{w}^{n+1,i+1})$
- 11: $n_{\text{stage}} = n_{\text{stage}} + 1$; {Structure solver CSD}
- 12: **end while**
- 13: **end for**

Stability, weak vs. strong (staged)

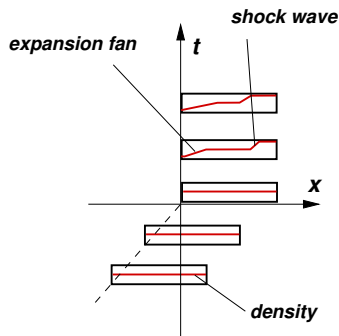
- Stage loop is a **fixed point iteration approach** to the monolithic (strong coupled) integration, so that if the stage loop is iterated and converged the algorithm has the **stability properties of the monolithic one.**
($\Delta t_{\text{crit, staged}} \gg \Delta t_{\text{crit, weak}}$)
- However, time step may be limited by **convergence of the stage loop**, i.e. it may happen that for a given Δt the fixed point stage loop does not converge.
- Computational cost is increased by the number of **stages.**

ALE invariance

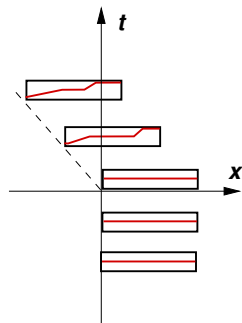
A key point in fluid-structure interaction problems is the use of the “**Arbitrary Lagrangian Eulerian formulation**” (ALE) , which allows the use of moving meshes. As the ALE convective terms affect the advective terms, some modifications are needed to the standard stabilization terms in order to get 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.

- **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"**.

ALE invariance test case. Sudden stop of gas container



**container initially moving
at constant speed u_0
is suddenly stopped**



**container initially at rest
is suddenly put in movement
with constant negative speed $-u_0$**

ALE invariance. SUPG Stabilization term

$$\frac{\partial \mathbf{U}_c}{\partial t} + \frac{\partial \mathcal{F}_{adv,x}}{\partial x} = \frac{\partial \mathcal{F}_{diff,x}}{\partial x}; \quad (\text{gov. eqs. in cons. form})$$

$$\mathbf{C} \frac{\partial \mathbf{U}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{U}}{\partial x} = \mathbf{K} \frac{\partial^2 \mathbf{U}}{\partial x^2}; \quad (\text{gov. eqs. in quasi-linear form}) \quad (3)$$

Sufficient conditions for ALE invariance ($\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{v}_{\text{mesh}} \mathbf{C}$)

$$P = \nabla N \cdot \tilde{\mathbf{A}} \tau \mathbf{C}^{-1}; \quad (\text{SUPG pert. function})$$

$$\tau \text{ transform as } \mathbf{U} \times \mathbf{U}, \quad \left(\text{i.e. } \tau' = \frac{\partial \mathbf{U}'}{\partial \mathbf{U}} \tau \frac{\partial \mathbf{U}}{\partial \mathbf{U}'} \right). \quad (4)$$

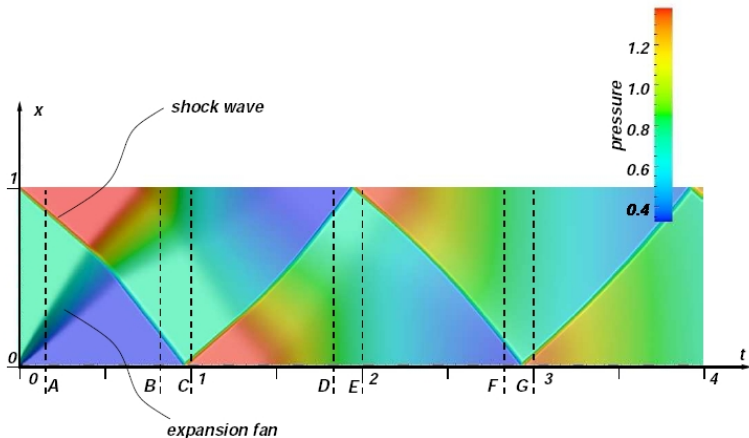
This last is verified if τ is $f(\mathbf{C}^{-1} \tilde{\mathbf{A}})$, for instance (inviscid case):

$$\tau = \frac{h}{\max |\lambda_j|} \mathbf{I}, \quad \lambda_j = \text{eig}(\mathbf{C}^{-1} \tilde{\mathbf{A}}), \quad (\text{max. eigenv.})$$

$$\tau = h |\mathbf{C}^{-1} \tilde{\mathbf{A}}|^{-1}. \quad (|\cdot| \text{ in matrix sense}) \quad (5)$$

ALE invariance. Sudden stop of a gas container

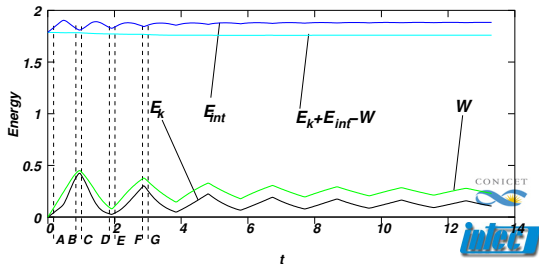
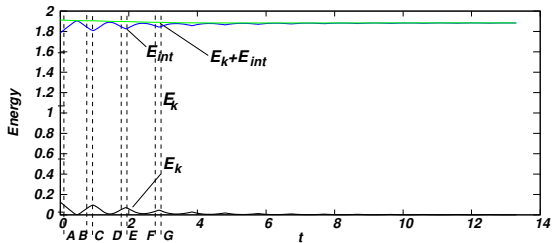
$\gamma = 1.4$, $u_0/c_0 = 0.5$. Results in both reference systems are **equivalent to machine precision.**



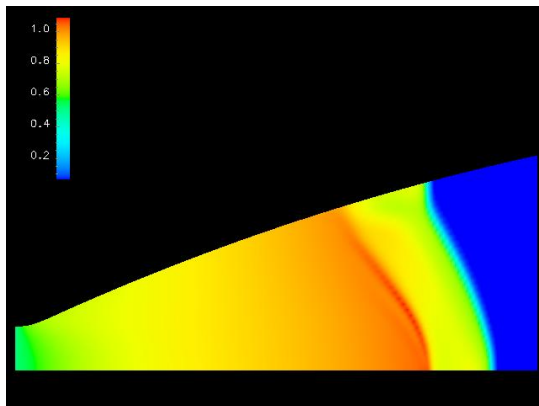
Energy Balance

(Up) Energy balance in reference system fixed w.r.t container

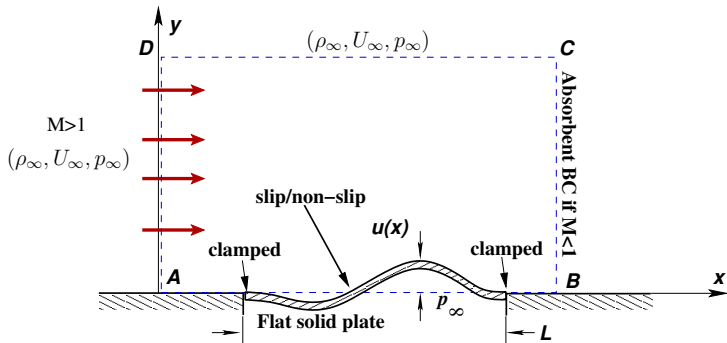
(Down) Energy balance in reference system fixed w.r.t initial gas at rest



Supersonic gas exhaust in the nozzle of the Vulcain rocket engine



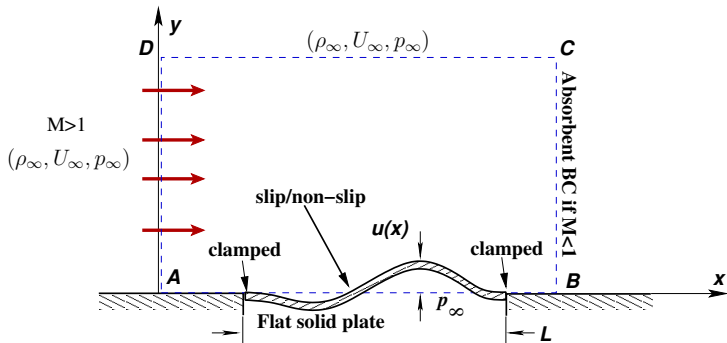
Test: Flutter of a flat solid plate aligned with the supersonic gas flow



- **Viscous/inviscid supersonic flow** (EULER/Compressible Navier-Stokes eqs.).
- **Thin plate theory** for structure:

$$m\ddot{u}(x, t) + D \frac{\partial^4 u(x, t)}{\partial x^4} = -(p - p_\infty) + f(x, t).$$

Test: Flutter of a flat solid plate aligned with the supersonic gas flow



- Coupled through **pressure** and **traction** (viscous case only) on interface boundary.
- Undisturbed flow $(\rho, \mathbf{v}, p)_\infty$ is a solution of the problem for zero initial condition (solid problem).

Theoretical Approximation: Houbolt's model [1958]

- Fluid Problem

$$p - p_\infty = C_x \frac{\partial u}{\partial x} + C_t \frac{\partial u}{\partial t},$$
$$C_x = \frac{\rho_\infty U_\infty^2}{\sqrt{M_\infty^2 - 1}}, \quad C_t = \frac{\rho_\infty U_\infty (M_\infty^2 - 2)}{(M_\infty^2 - 1)^{3/2}}.$$

- Then, for the Plate Problem the deflection becomes

$$m\ddot{u} + D \frac{\partial^4 u}{\partial x^4} = -C_x \frac{\partial u}{\partial x} - C_t \frac{\partial u}{\partial t}.$$

- Replacing $u(x) = \sum_{k=1}^N a_k \psi_k(x)$ in the Houbolt's approximation and using Galerkin method

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{K}\mathbf{a} + \mathbf{H}_x \dot{\mathbf{a}} + \mathbf{H}_t \dot{\mathbf{a}} = 0,$$

- If the ansatz $a(t) = \hat{\mathbf{a}}e^{\lambda t}$ is proposed as a solution for the **Plate Problem**, the following eigenvalue problem is stated

$$\left(\lambda^2 \mathbf{M} + \lambda \mathbf{H}_t + \mathbf{K} + \mathbf{H}_x \right) \hat{\mathbf{a}} = 0.$$

- Using time and mass non-dimensional parameters

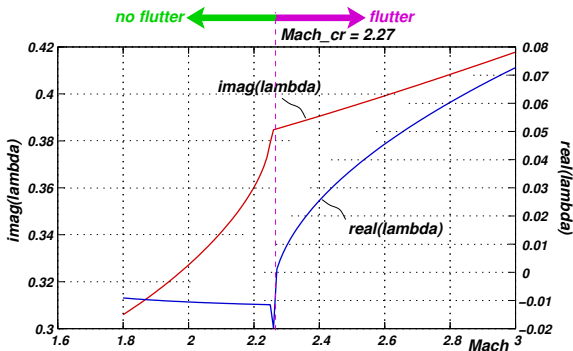
$$N_T = \left(\frac{T_{fl}}{T_{str}} \right)^2 = \frac{L/U_\infty}{\sqrt{mL^4/D}} = \frac{D}{mL^2 U_\infty^2} \quad \text{and}$$

$$N_M = \frac{\rho_\infty L^3}{mL^2} = \frac{\rho_\infty L}{m},$$

the space of parameters is full covered.

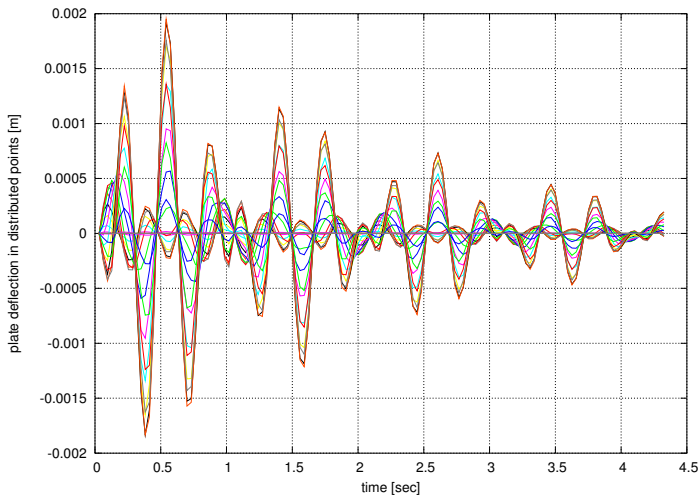
Results for the Houbolt's model

- $N = 20$, $N_x = 5000$, and a sweep in M_∞ while keeping constant ρ_∞ , m , L and D , i.e., $N_M = \text{cte}$ and $N_T \propto M_\infty^{-2}$.

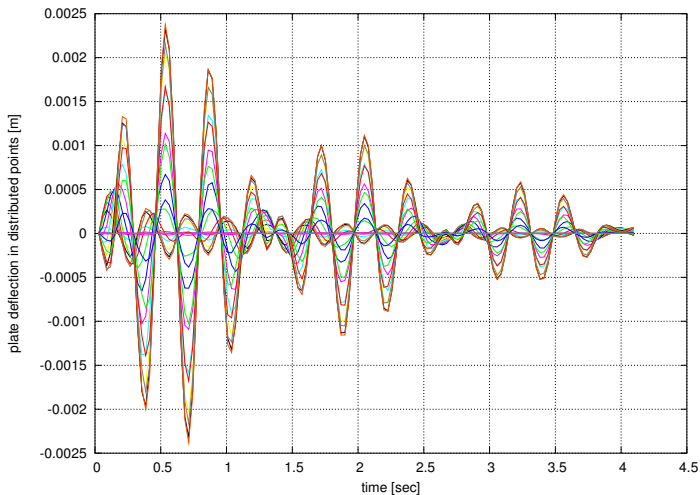


Flutter mode appears for $M_\infty \geq 2.27$ (or $N_T = 4.3438 \times 10^{-5}$ and $N_M = 0.055$).

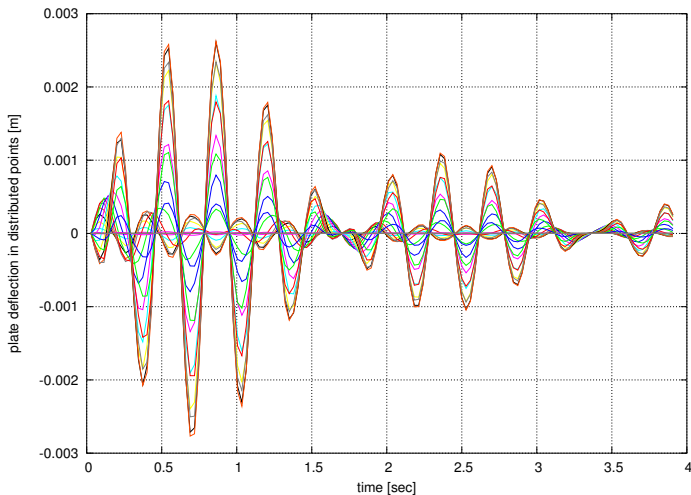
Staged FSI-FEM Results for flutter: Structure response at $M=1.8$



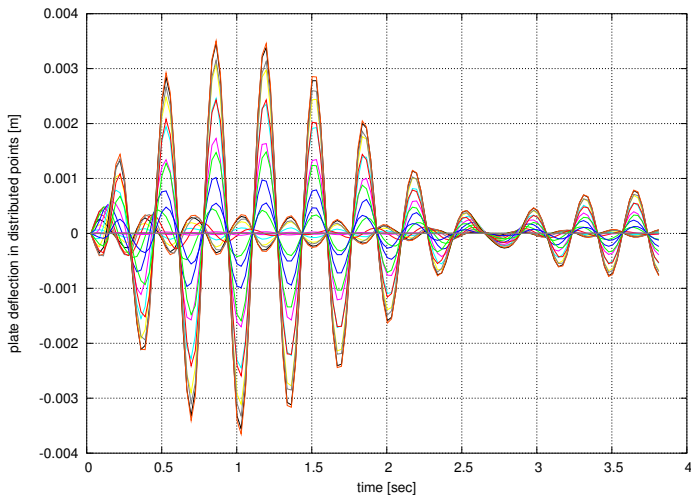
Structure response at $M=2.0$



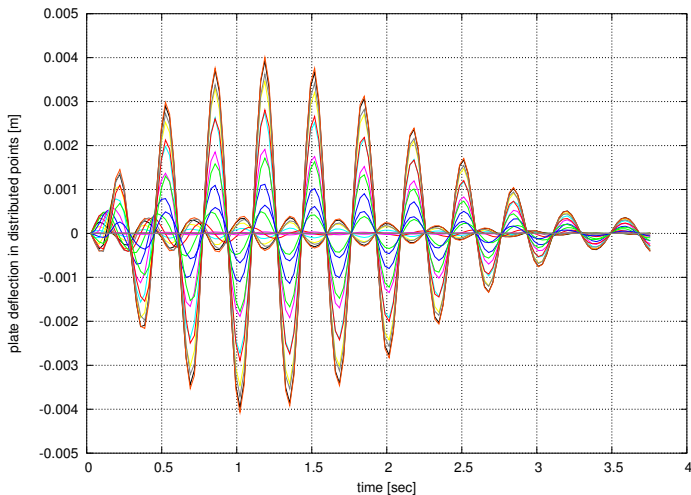
Structure response at $M=2.1$



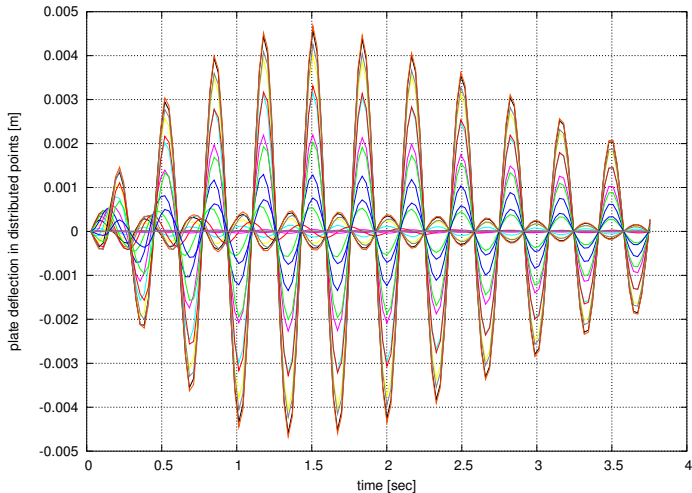
Structure response at $M=2.2$



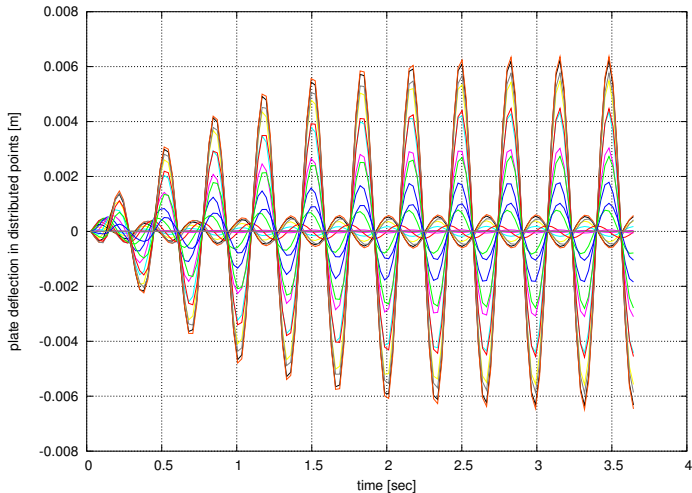
Structure response at $M=2.225$



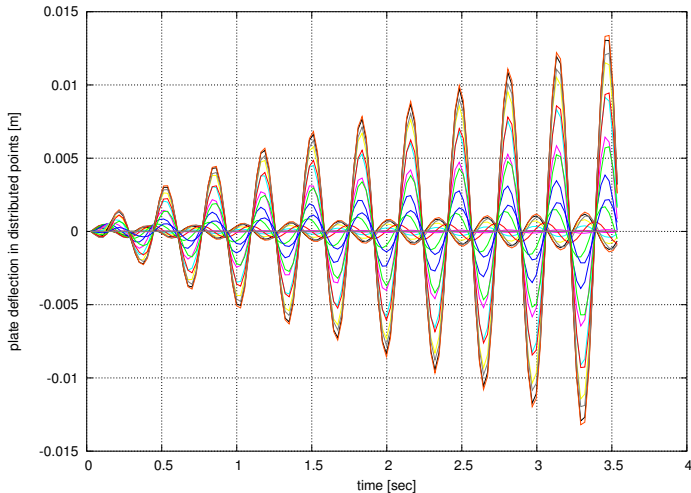
Structure response at $M=2.25$



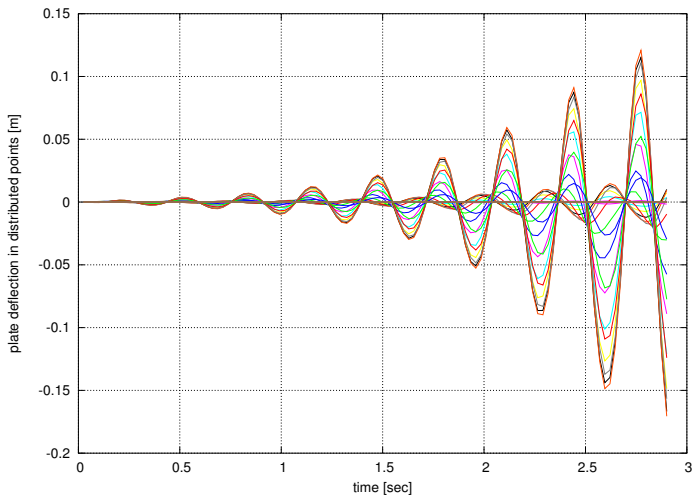
Structure response at $M=2.275$



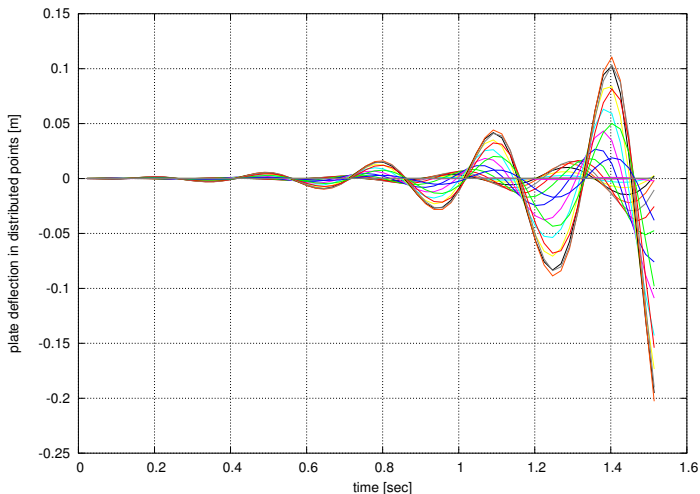
Structure response at $M=2.3$



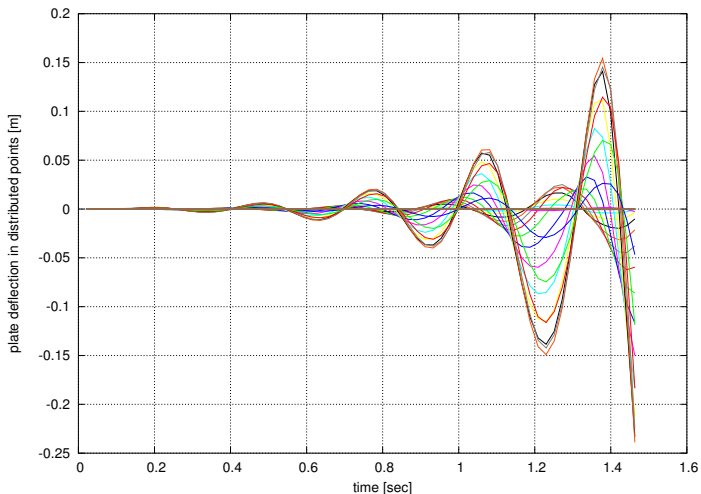
Structure response at $M=2.5$



Structure response at $M=3.0$



Structure response at $M=3.2$



Flat plate in Flutter



Step 327, time 1.68176 secs, Color=density

- Being

$$\frac{N_M}{N_T M_\infty^2} = \frac{\rho_\infty L^3 c_\infty}{D},$$

sweep in N_T , N_M and M_∞ estimated the **flutter region** as

$$\frac{N_M}{N_T M_\infty^2} < 200 \quad \text{no flutter for any Mach number,}$$

$$\frac{N_M}{N_T M_\infty^2} > 300 \quad \text{flutter for the lowest Mach considered (M=1.8)}$$

Stability of the staged algorithm (flutter region, cont.)

$$U_{\infty} = M_{\infty} = 2$$

$$t = 0.06$$

$$\nu = 0.33$$

$$m = 0.002$$

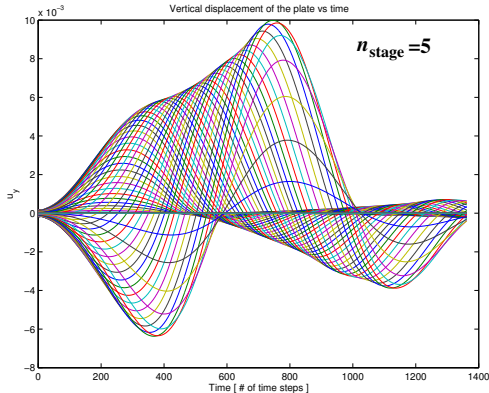
$$E = 39.6$$

$$D = 8.010^{-4}$$

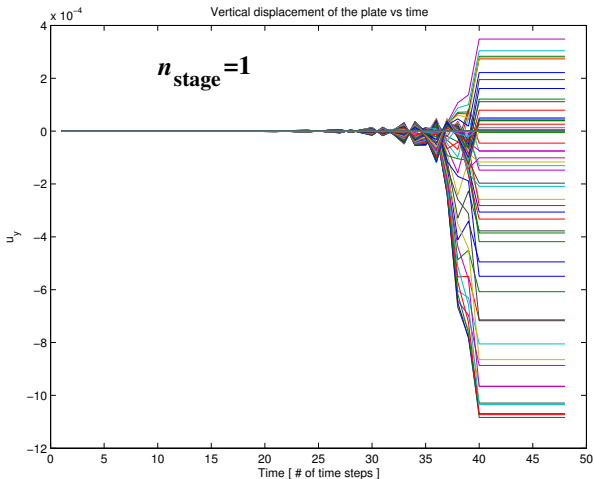
$$N_T = \frac{D}{mL^2 U_{\infty}^2} = 0.025$$

$$N_M = \frac{\rho_{\infty} L}{m} = 1000.0$$

$$\frac{N_M}{N_T M_{\infty}^2} = 10000 > 300 \quad (\text{i.e., inside the flutter region})$$



Stability of the staged algorithm (flutter region, contd.)



Stability of the staged algorithm outside the flutter region, i.e.

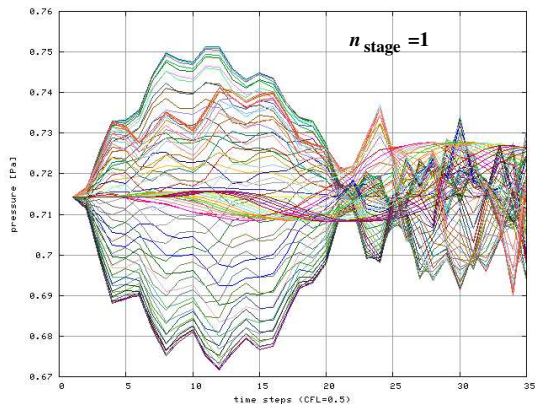
$$N_M / (N_T M_\infty^2) \ll 200$$

- $N_M / (N_T M_\infty^2)$ do not depend on plate density m

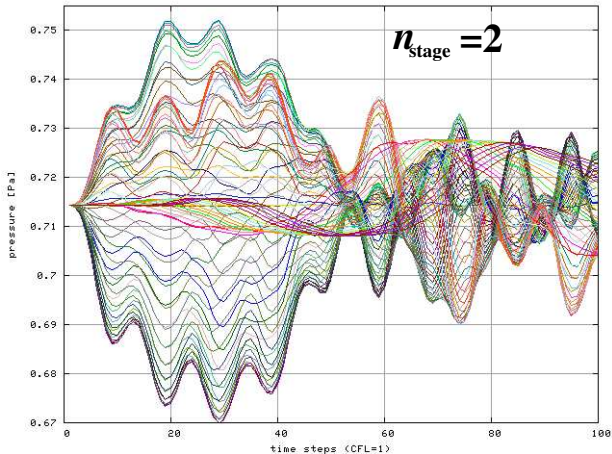
with $m = 0.00135$

$$\frac{N_M}{N_T M_\infty^2} = 12 < 200$$

(i.e., outside the
flutter region)



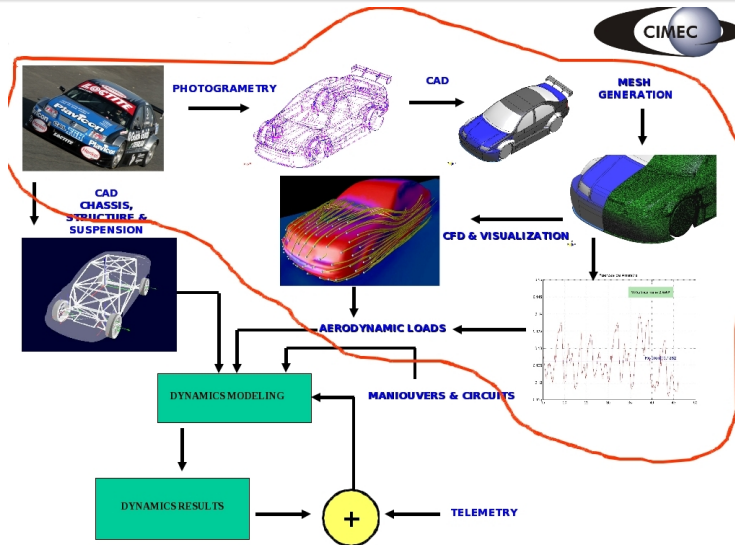
Stability of the staged algorithm **outside** the flutter region (contd.)



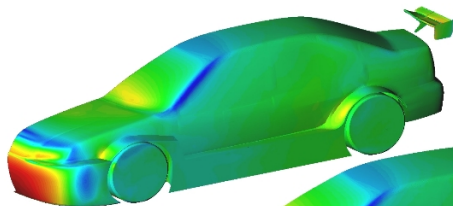
Temario

- 1 CIMEC
- 2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones
 - Solución de Sistemas de Ecuaciones Lineales
 - Método de Descomposición de Dominio (DDM)
 - Precondicionamiento para el sistema $Ax = b$.
- 3 **Parte II – Aplicaciones y Usos**
 - ESA CFD Code Evaluation Tests
 - Condiciones de Borde Dinámicas en CFD
 - Fluid-Structure Application Problems
 - **Descripción Aerodinámica de un Automóvil de Competición**
- 4 Parte III – Conclusiones Finales
 - Trabajos Publicados

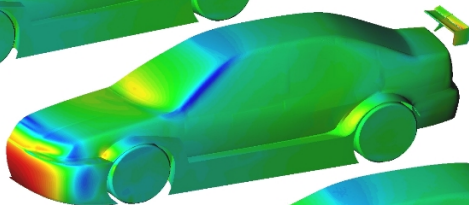
TC2000 Bora – Framework



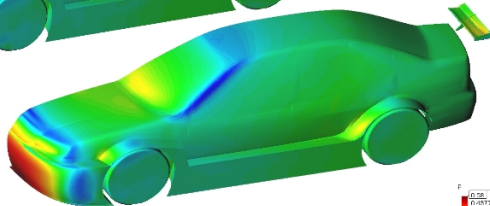
TC2000 Bora – Configuraciones



Bora_wc_large_ep



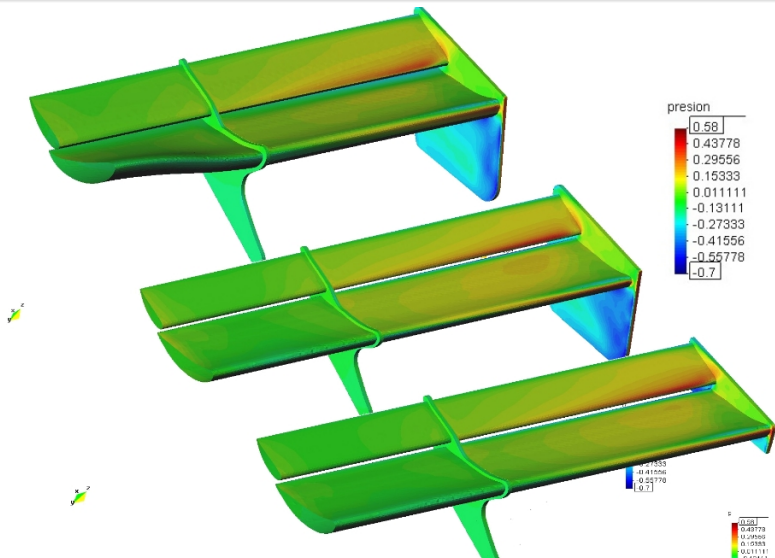
Bora_wr_large_ep



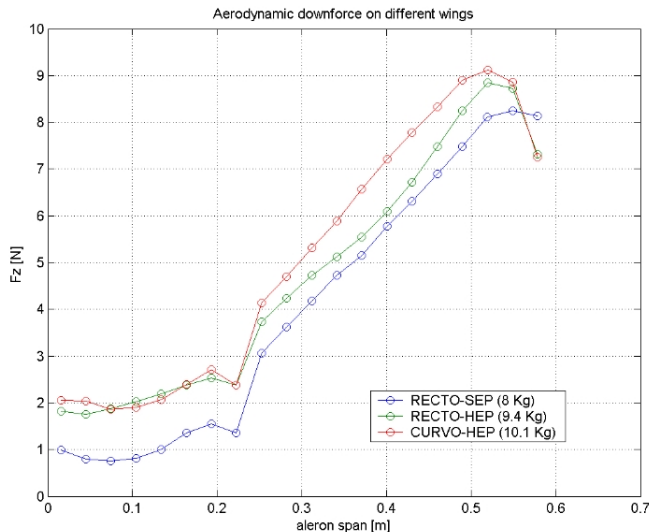
Bora_wr_small_ep



TC2000 Bora – Configuraciones

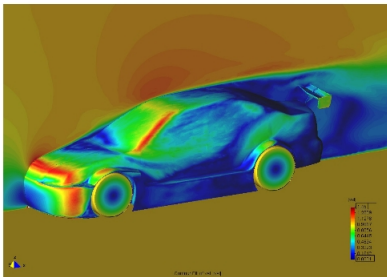


TC2000 Bora – DownForces

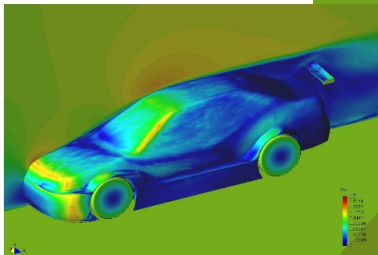
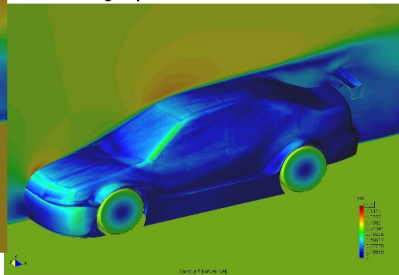


TC2000 Bora – NS-Flow $\Re = 1.7 \cdot 10^7$

movie-1
movie-2

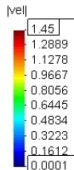


Bora wc large ep



Bora wr large ep

Bora wr small ep



Conclusions

- Extremely useful and versatile “non-reflecting” B.C.
- **Dynamic B.C.** are well suited for complex simulations where the number of incoming characteristic waves may vary during the computation or may not be known a priori.
- **Absorbent** boundary conditions reduce computational cost by allowing to put the artificial exterior boundary closer to the region of interest.
- “Staged” strategy provides a smooth blending between weak coupling and strong coupling.
- Moderately coupled problems that can not be treated with the pure weak coupling approach, can be solved with the staged algorithm using few stages per time step. Staged add extra computational cost. It can be used as preconditioner.

Conclusiones Finales

Las propuestas dentro de DDM y sus aplicaciones a problemas mas complejos, además de los resultados mostrados aquí e incluidos en la versión escrita de la tesis han sido publicados en los siguientes artículos.

Temario

- 1 CIMEC
- 2 Parte I – MPP y Resolución de Grandes Sistemas de Ecuaciones
 - Solución de Sistemas de Ecuaciones Lineales
 - Método de Descomposición de Dominio (DDM)
 - Precondicionamiento para el sistema $Ax = b$.
- 3 Parte II – Aplicaciones y Usos
 - ESA CFD Code Evaluation Tests
 - Condiciones de Borde Dinámicas en CFD
 - Fluid-Structure Application Problems
 - Descripción Aerodinámica de un Automóvil de Competición
- 4 Parte III – Conclusiones Finales
 - Trabajos Publicados

Publicaciones

- [1] Rodrigo R. Paz and Mario A. Storti.
An Interface Strip Preconditioner for Domain Decomposition Methods: Application to Hydrology. *International Journal for Numerical Methods in Engineering*, **62(13)**:1873-1894, 2005.
- [2] Rodrigo R. Paz, Norberto M. Nigro and Mario A. Storti.
On the efficiency and quality of numerical solutions in CFD problems using the Interface Strip Preconditioner for domain decomposition methods. *International Journal for Numerical Methods in Fluids*, **52(1)**:89-118, 2006.
- [3] Mario A. Storti, Lisandro Dalcín, Rodrigo R. Paz, Andrea Yommi, Victorio Sonzogni and Norberto M. Nigro.
A Preconditioner for Schur Complement Matrix. *Advances in engineering Software*, **65(9)**:1108-1115, (2006).
- [4] Mario A. Storti, Lisandro Dalcín, Rodrigo R. Paz, Andrea Yommi, Victorio Sonzogni and Norberto M. Nigro.
Interface Strip Preconditioner for Domain Decomposition Methods. *Journal of Computational Methods in Sciences and Engineering*, (2003). In press.
- [5] Lisandro Dalcín, Rodrigo R. Paz and Mario A. Storti.
MPI for Python. *Journal of Parallel and Distributed Computing*, **37**:754-762, (2005).

Publicaciones

- [6] Mario A. Storti, Ricardo Prado, Norberto M. Nigro, Rodrigo R. Paz, Lisandro D. Dalcín, Jorge D'Elía, Javier Garibaldi.
Computational Methods for Boundary Layer equations (in spanish). *National Academy of Science in Buenos Aires*. In press, (2006).
- [7] Mario A. Storti, Norberto M. Nigro and Rodrigo R. Paz.
Strong coupling strategy for fluid-structure interaction problems in supersonic regime via fixed point iteration. *Journal of Sound and Vibration*. In correction. 2005.
- [8] Mario A. Storti, Norberto M. Nigro, Rodrigo R. Paz and Lisandro Dalcín.
Dynamics boundary conditions in CFD. *Computer Methods in Applied Mechanics and Engineering*. In press. 2006.
- [9] Sergio A. Vera, Mariano Febbo, Carlos G. Méndez and Rodrigo R. Paz.
Vibrations of a plate with an attached two degree of freedom system. *Journal of Sound and Vibration*, **285(1-2)**:457-466, 2004.
- [10] Lisandro Dalcín, Rodrigo R. Paz, Mario A. Storti and Jorge D'Elía.
MPI for Python: Efficiency Improvements and MPI-2 Extensions. *Journal of Parallel and Distributed Computing*, In press. 2006.

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 freely distributed software such as **GNU/Linux OS**, **MPI**, **PETSc**, **Newmat**, **Metis**, **Octave**, the **CGAL geometrical library**, **OpenDX** and many others.