

# Métodos de Descomposición de Dominio en Mecánica de Fluidos Computacional

Rodrigo R. PAZ  
Director: Mario A. STORTI

::Centro Internacional de Métodos Computacionales en Ingeniería::  
CONICET-INTEC-UNL  
Argentina  
::2006::

Defensa de Tesis Doctoral en Ingeniería, Mención Mecánica  
Computacional



## Temario

- 1 **Parte I – Métodos de Descomposición de Dominio**
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 **Parte II – Aplicaciones y Usos**
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 **Parte III – Conclusiones Finales**
  - Trabajos Publicados
  - Agradecimientos

# Temario

- 1 Parte I – Métodos de Descomposición de Dominio
  - **Introducción - Motivos - Objetivos**
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## Introducción

- **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'.
- Es una vieja idea que data del siglo XIX, probablemente propuesto en 1870 por Hermann Amandus Schwarz (1843 - 1921), discípulo de Karl Weierstrass y que se pone en práctica en la industria aeronáutica de manera rudimentaria y casi artesanal (al igual que los métodos Multigrilla).
- 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.

## Introducción

- **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'.
- Es una vieja idea que data del siglo XIX, probablemente propuesto en 1870 por Hermann Amandus Schwarz (1843 - 1921), discípulo de Karl Weierstrass y que se pone en práctica en la industria aeronáutica de manera rudimentaria y casi artesanal (al igual que los métodos Multigrilla).
- 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.

## Introducción

- **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'.
- Es una vieja idea que data del siglo XIX, probablemente propuesto en 1870 por Hermann Amandus Schwarz (1843 - 1921), discípulo de Karl Weierstrass y que se pone en práctica en la industria aeronáutica de manera rudimentaria y casi artesanal (al igual que los métodos Multigrilla).
- 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.

## Motivos y Necesidades (cont.)

- En el CIMEC se tenía un 'cluster' tipo 'Beowulf' de 20 PC's para hacer simulaciones en gran escala en **Mecánica de Fluidos** y Problemas Multifísica en general. También se estaba proyectando la construcción de un cluster más grande (approx. 100-120 nodos).
- Al mismo tiempo se comenzaba a desarrollar un código de Elementos Finitos en Paralelo para resolver problemas Mutlifísica (código PETSc-FEM:  
<http://www.cimec.org.ar/petscfem> - Storti, Nigro, Paz, Dalcín).

## Motivos y Necesidades (cont.)

- En el CIMEC se tenía un 'cluster' tipo 'Beowulf' de 20 PC's para hacer simulaciones en gran escala en Mecánica de Fluidos y Problemas Multifísica en general. También se estaba proyectando la construcción de un cluster más grande (approx. 100-120 nodos).
- Al mismo tiempo se comenzaba a desarrollar un código de Elementos Finitos en Paralelo para resolver problemas Mutlifísica (código PETSc-FEM:  
<http://www.cimec.org.ar/petscfem> - Storti, Nigro, Paz, Dalcín).



## 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.

## 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]**, **Papadarakakis 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 ( $n_{\text{dof}}$  **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**.

## 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 ( $n_{\text{dof}}$  **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**.

## 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 ( $n_{\text{dof}}$  **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.

## 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.

## 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 **Parte I – Métodos de Descomposición de Dominio**
  - Introducción - Motivos - Objetivos
  - **Solución de Sistemas de Ecuaciones Lineales**
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## 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

- 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

- 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 (GMRs), 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

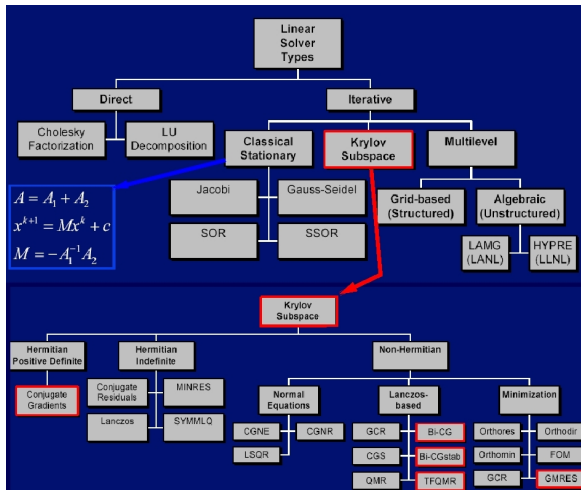
- Existen un gran número de métodos para resolver sistemas de ecuaciones lineales dependiendo principalmente del tipo de matrices entre manos.
- Básicamente hay **tres (o cinco)** grandes grupos:
  - 1 Métodos Directos: basados en la factorización LU y de Cholesky y todas sus variantes.
  - 2 Métodos 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

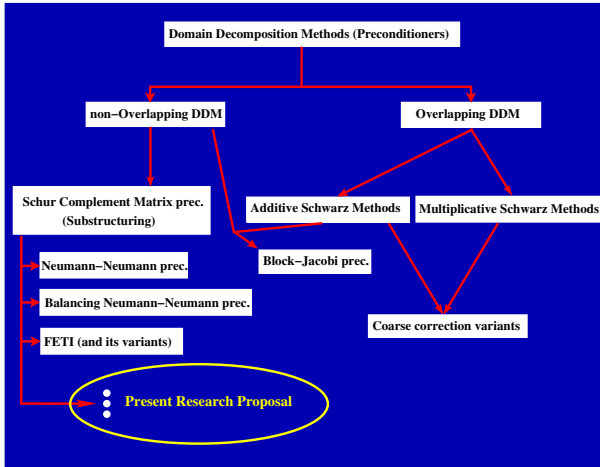
- Existen un gran número de métodos para resolver sistemas de ecuaciones lineales dependiendo principalmente del tipo de matrices entre manos.
- Básicamente hay **tres (o cinco)** grandes grupos:
  - 1 Métodos Directos: basados en la factorización LU y de Cholesky y todas sus variantes.
  - 2 Métodos 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 (GMRs), 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



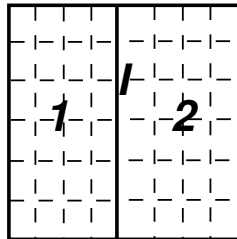


# Temario

- 1 **Parte I – Métodos de Descomposición de Dominio**
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - **Método de Descomposición de Dominio (DDM)**
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## Descomposición de Dominio 'No Solapado' (non-Overlapping DDM)

$$\begin{bmatrix} \mathbf{A}_{11} & 0 & \mathbf{A}_{1l} \\ 0 & \mathbf{A}_{22} & \mathbf{A}_{2l} \\ \mathbf{A}_{l1} & \mathbf{A}_{l2} & \mathbf{A}_{ll} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_l \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_l \end{bmatrix}$$



- La solución de  $\mathbf{Ax} = \mathbf{b}$  es equivalente a la solución de

$$(\mathbf{A}_{ll} - \mathbf{A}_{l1}\mathbf{A}_{11}^{-1}\mathbf{A}_{1l} - \mathbf{A}_{l2}\mathbf{A}_{22}^{-1}\mathbf{A}_{2l}) \mathbf{x}_l = (\mathbf{b}_l - \mathbf{A}_{l1}\mathbf{A}_{11}^{-1}\mathbf{b}_1 - \mathbf{A}_{l2}\mathbf{A}_{22}^{-1}\mathbf{b}_2)$$

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

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

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

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  $\mathbf{w} = \mathbf{A}_{/L} \mathbf{x}$
- 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

- 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

- 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 😊.



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

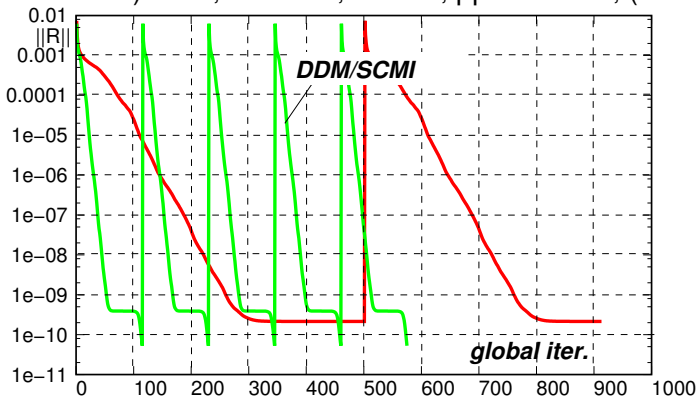
- Debido a que la cantidad de iteraciones resulta menor y que el espacio de iteración es significativamente menor, los requerimientos de memoria RAM para almacenar los vectores que generan el espacio de Krylov resulta menor 😊. Pero esto está compensado por la memoria necesaria para la factorización de 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 😊.

## 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 **Parte I – Métodos de Descomposición de Dominio**
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - **Precondicionamiento para el sistema  $Ax = b$ .**
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## 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 precondicionadores para este sistema.

### Definition

- Se propone entonces resolver un problema equivalente:

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

- La solución es la misma que la del sistema  $Sx_I = \tilde{b}_I$ , 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**: Astrakhandtsev [1971], Nicolaidis [1875], Brandt [1977], Hackbusch [1977], Wesseling [1978]

## 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**: Astrakhansev [1971], Nicolaidis [1875], Brandt [1977], Hackbusch [1977], Wesseling [1978]

## 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**: Astrakhandtsev [1971], Nicolaidis [1875], Brandt [1977], Hackbusch [1977], Wesseling [1978]

## 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**: Astrakhansev [1971], Nicolaidis [1875], Brandt [1977], Hackbusch [1977], Wesseling [1978]

## 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**: Astrakhandtsev [1971], Nicolaidis [1875], Brandt [1977], Hackbusch [1977], Wesseling [1978]

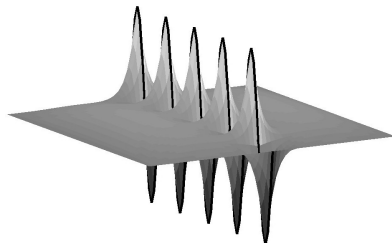
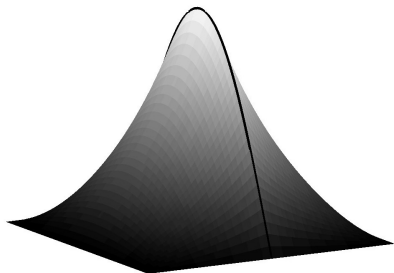
## Precondicionamiento del problema de Interfaz. Operador de Steklov

- Es claro que conocer el espectro de **autovalores y autofunciones** del operador de Steklov (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  $\mathcal{S}$**  son

$$\psi_n(y) = \sin(k_n y)$$
$$\omega_n = k_n [\coth(k_n L_1) + \coth(k_n L_2)]$$

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

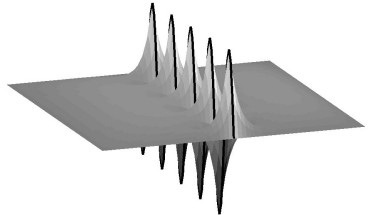
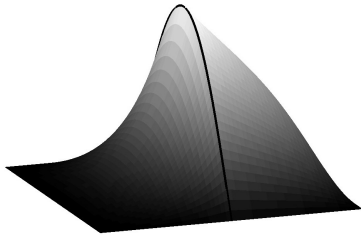
## 1<sup>ra</sup> y 10<sup>ma</sup> autofunción



- 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<sup>ra</sup> y 10<sup>ma</sup> autofunción

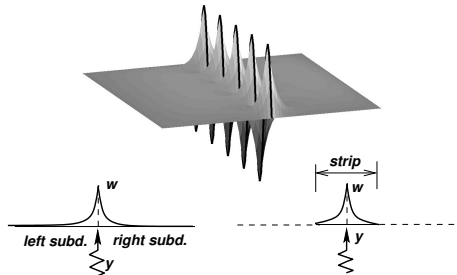
- En presencia de operadores con términos advectivos las autofunciones dejan de ser simétricas. La separación de los flujos en la interfaz es desviado aguas abajo del subdominio.





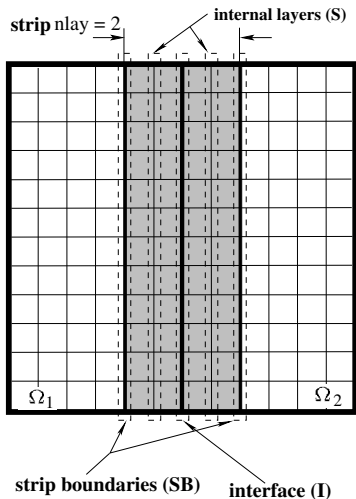
## Prec. de Banda Alrededor de la Interface (**Interface Strip Prec.**)

- Para los modos de alta frecuencia (autovalores altos) las autofunciones del operador de Steklov 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.)

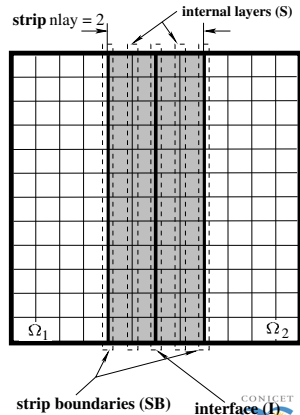
$$\begin{bmatrix} A_{II} & A_{IS} & A_{I,SB} \\ A_{SI} & A_{SS} & A_{S,SB} \\ A_{SB,I} & A_{SB,S} & A_{SB,SB} \end{bmatrix} \begin{bmatrix} v_I \\ v_S \\ v_{SB} \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \\ 0 \end{bmatrix},$$

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 en una malla de  $50 \times 50$  elementos.

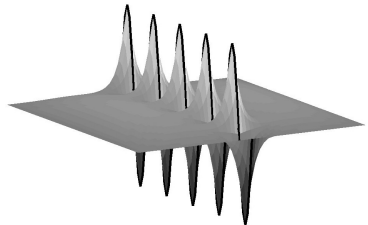
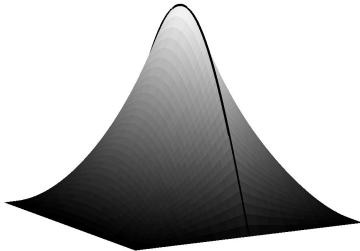
## Resultados teóricos para advección lineal escalar, malla: 100x100, Pe=50

$u$	$\text{cond}(S)$	$\text{cond}(\mathcal{P}_{\text{NN}}^{-1}S)$	$\text{cond}(\mathcal{P}_{\text{ISP}}^{-1}S)$
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 en una malla de  $100 \times 100$  elementos.

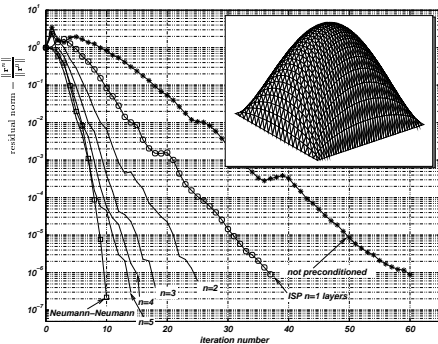
## Flux splitting - Precondicionador de Neumann-Neumann

- 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 son simétricas.)

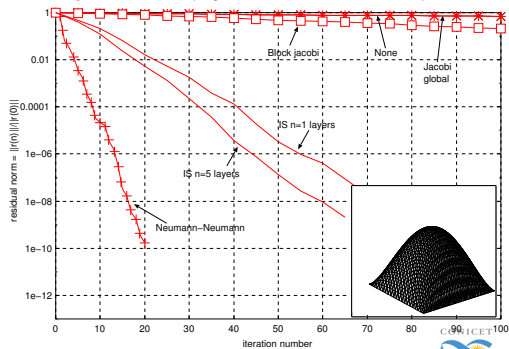


# Resultados numéricos para Poisson

test secuencial 120x120 el

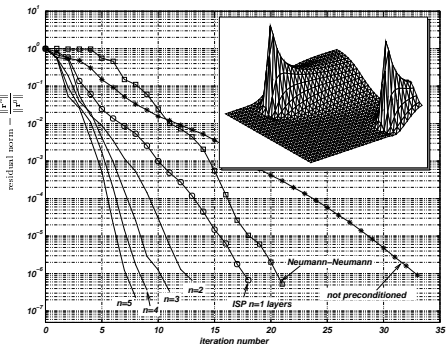


test paralelo (np=4, 500x500 el)

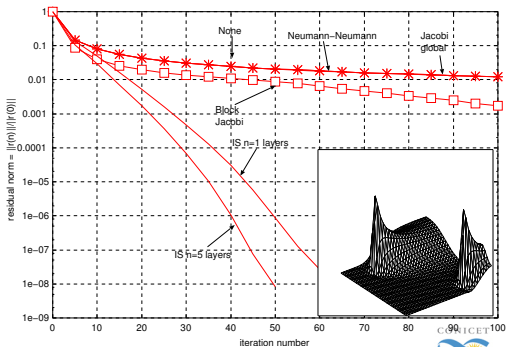


# Resultados numéricos para advección escalar: $Pe=25$

test secuencial 120x120 el

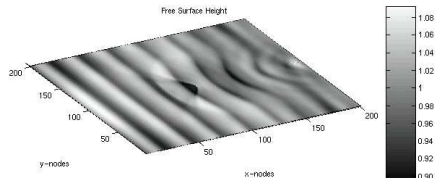
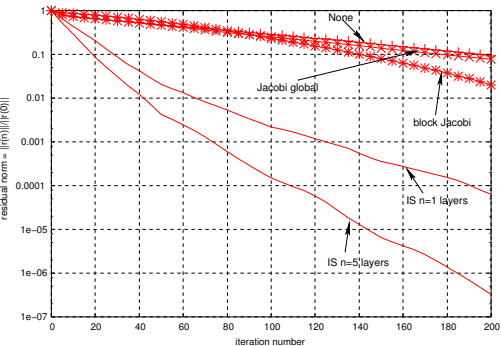


test paralelo (np=4, 500x500 el)



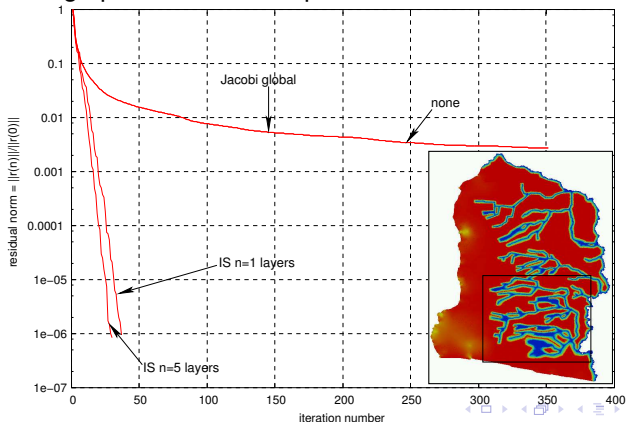


## Desempeño en Shallow-water 2D (malla $500 \times 500$ elem.)



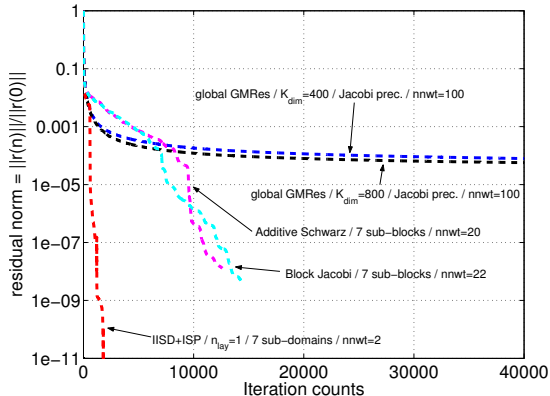
## Test: Interacción Río-Acuífero

Shallow-Water 1D/Groundwater interaction over several basins in Sta Fe. 32900Km<sup>2</sup>, 1.65M triangles, 9 subdomains (one per processor).  
Annual average periodical raindrop.



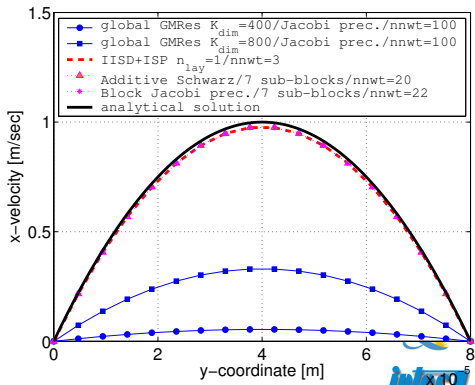
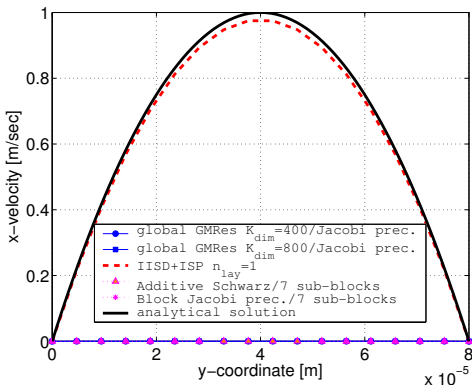
## Test: Flujo de Stokes en un canal horizontal largo

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



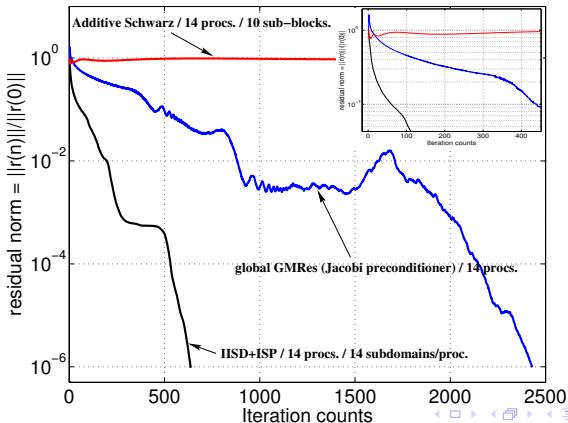
## Flujo de Stokes en un canal horizontal: Velocidad

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).

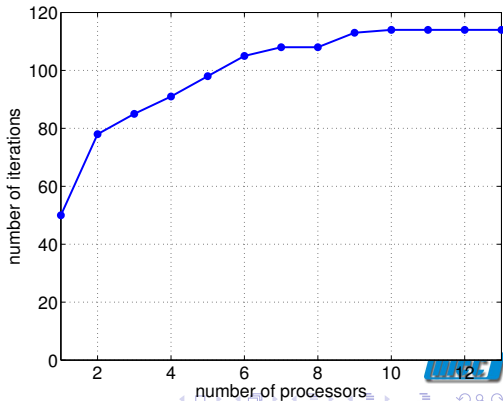
- Solución del Paso de Poisson.  $Re = 1000$ .  
( $\Delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G}(\mathbf{P}^{n+1} - \gamma \mathbf{P}^n) = \mathbf{D} \hat{\mathbf{U}}^{n+1}$ ) en una malla de  $400 \times 400$  en 14 procs. (resultados: Residuos  $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$ ).



## Escalabilidad Práctica de ISP

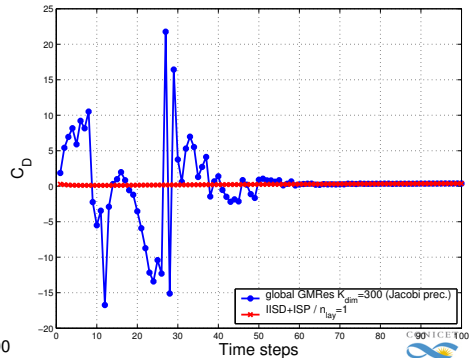
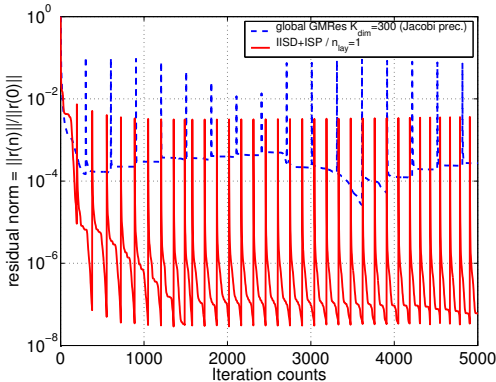
- **Escalabilidad:** El número de iteraciones para alcanzar una tolerancia dada permanece constante a medida que el tamaño del problema global y el número de procesadores aumenta en la misma proporción (tamaño de problema fijo en cada procesador). ‘Escalabilidad de Gustafson’.

- 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.



# Temario

- 1 Parte I – Métodos de Descomposición de Dominio
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - **Conclusiones**
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos



## 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 Parte I – Métodos de Descomposición de Dominio
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusions
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## 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  $\lambda_k$ ,

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  $\Pi_V^\pm$  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.

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

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

- This assumes linearization of the equations around a state  $\mathbf{U}_{\text{ref}}$ . 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).

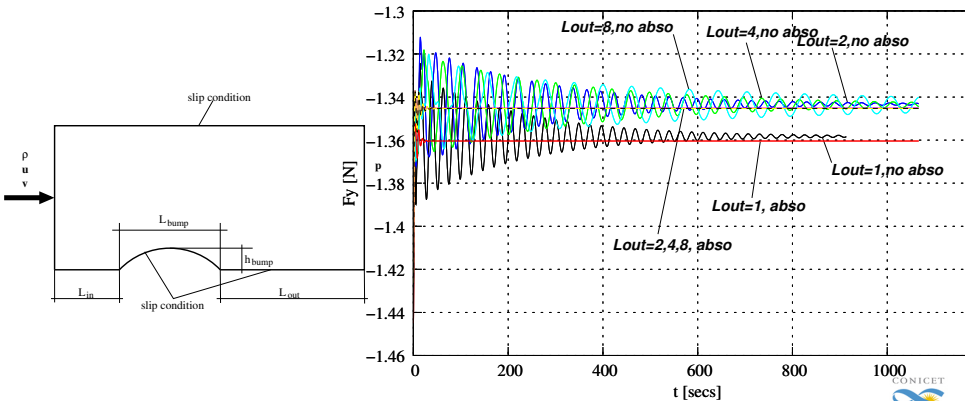
**Solution: Use last state as reference state, ULSAR.**

$\mathbf{U}_{\text{ref}} = \mathbf{U}^n$ ,  $n =$  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. $M_\infty = 0.5$



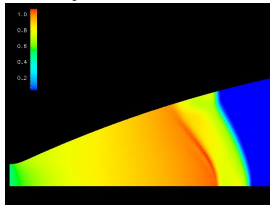


## Dynamic boundary conditions

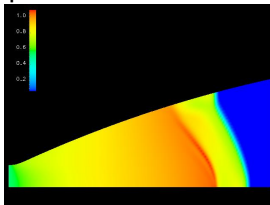
- Finally, there is a last twist in the use of absorbing boundary conditions.
- As the flow is computed it may happen that the number of characteristics **to impose** changes in time, i.e., it involves the change of the structure of the Jacobian matrix
- Two typical examples:

## 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  $6 \times 10^5$  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.

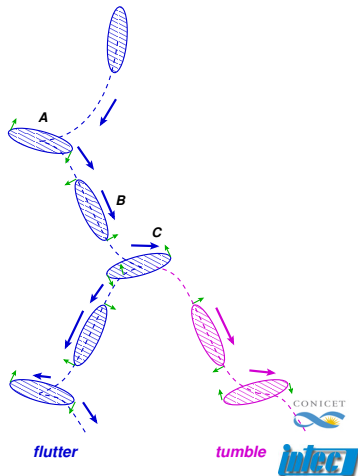
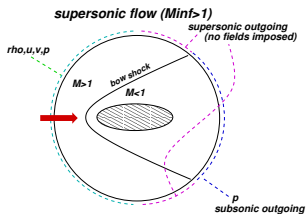
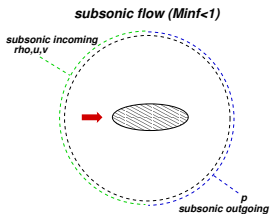


- We impose density, pressure and tangential velocity at inlet (assuming subsonic inlet), slip condition at the nozzle wall. The problem is with the outlet boundary. Initially the flow is subsonic (fluid at rest) there, and switches to supersonic. The rule states to impose 1 condition, as a subsonic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

$$\mathbf{C} \frac{\mathbf{U}_0^{n+1} - \mathbf{U}_0^n}{\Delta t} + \mathbf{A} \frac{\mathbf{U}_1^{n+1} - \mathbf{U}_0^n}{h} = 0;$$
$$\mathbf{C} \frac{\mathbf{U}_k^{n+1} - \mathbf{U}_k^n}{\Delta t} + \mathbf{A} \frac{\mathbf{U}_{k+1}^{n+1} - \mathbf{U}_{k-1}^n}{2h} = 0, \quad k \geq 1$$

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

## Lagrange Multipliers in 'characteristics' basis

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

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

$\Pi^\pm$  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 ( $\lambda_j \geq 0$ )

$$v_{j0}^{n+1} - v_{j0}^n = 0$$

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

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

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

$$v_{j,1m} = 0$$

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

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

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

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

## Absorbing boundary conditions and ALE

When using *Arbitrary Lagrangian-Eulerian* formulations:

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

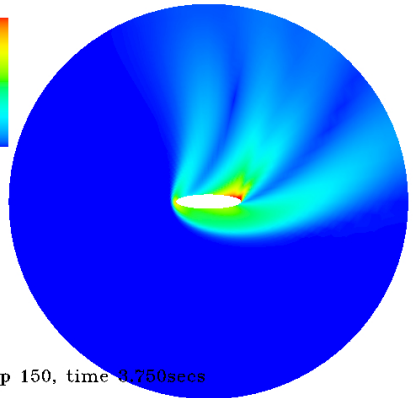
$$A_{\text{ALE},j} = \frac{\partial \mathcal{F}_{\text{adv},j}(\mathbf{U})}{\partial \mathbf{U}} - v_{\text{mesh},j} \mathbf{I}, \quad \text{ALE advective Jacobian}$$

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

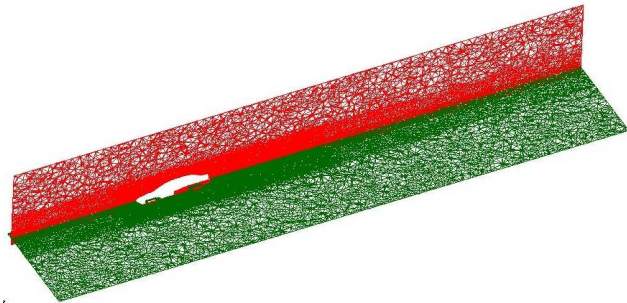
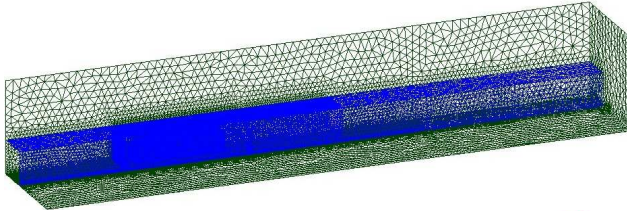
## Supersonic falling of an Ellipse

- $a = 1$ ,  $b = 0.6$  (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  $[x, y, \alpha, u, v, \dot{\alpha}]$ ),

2.25  
2.00  
1.75  
1.50  
1.25  
1.00  
0.75  
0.50  
0.25  
0.00



## Future Work: Absorbent B.C. for Incompressible Flow

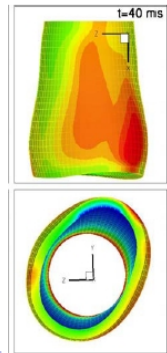


# Temario

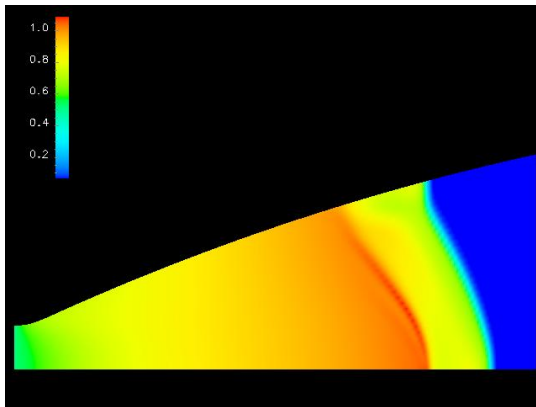
- 1 Parte I – Métodos de Descomposición de Dominio
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - **Aplicación en Problemas de Interacción Fluido-Estructura**
  - Conclusions
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## 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:



## Supersonic gas exhaust in the nozzle of the Vulcain rocket engine



## Staged (strongly coupled) partitioned FSI algorithm

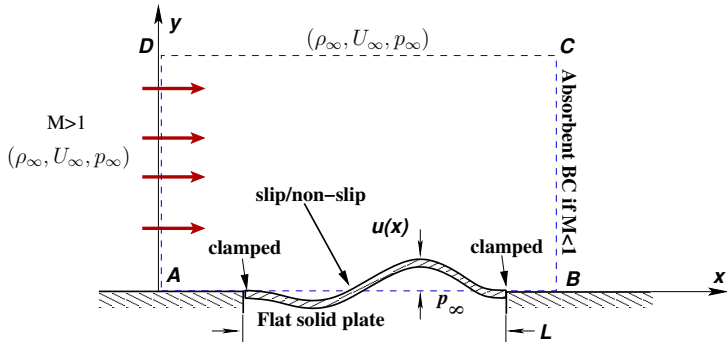
### FSI Algorithm

States:  $\mathbf{u}$  structure,  $\mathbf{w}$  fluid,  $\mathbf{X}$  mesh.

- 1: Initialize variables:
- 2: **for**  $n = 0$  to  $n_{\text{step}}$  **do** {Main time step loop}
- 3:  $t^n = n\Delta t$ ,
- 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: **for**  $i = 0$  to  $n_{\text{stage}}$  **do** {stage loop}
- 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:  $\mathbf{u}^{n+1,i+1} = \text{CSD}(\mathbf{u}^n, \mathbf{w}^n, \mathbf{w}^{n+1,i+1})$  {Structure solver CSD}
- 11: **end for**
- 12: **end for**



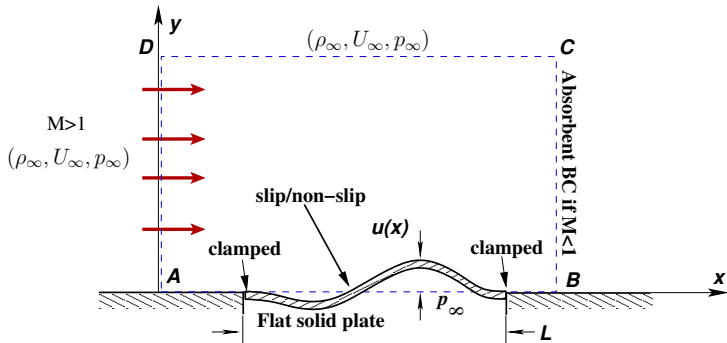
## Test: Flutter of a flat solid plate aligned with the supersonic gas flow



- **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

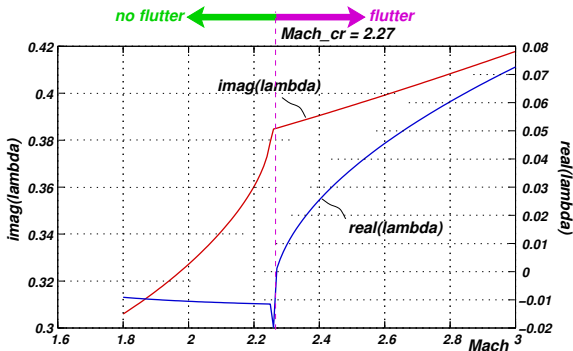
$$N_T = \left( \frac{T_{fl}}{T_{str}} \right)^2 = \frac{L/U_\infty}{\sqrt{mL^4/D}} = \frac{D}{mL^2 U_\infty^2} \quad \text{and}$$

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

the space of parameters is full covered.

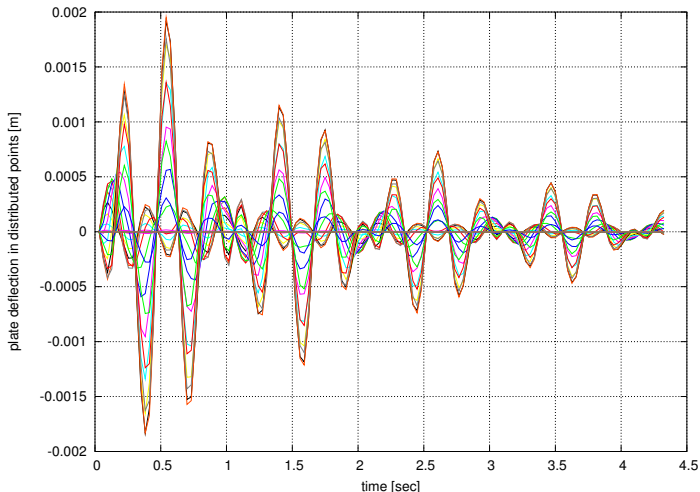
## Results for the Houbolt's model

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

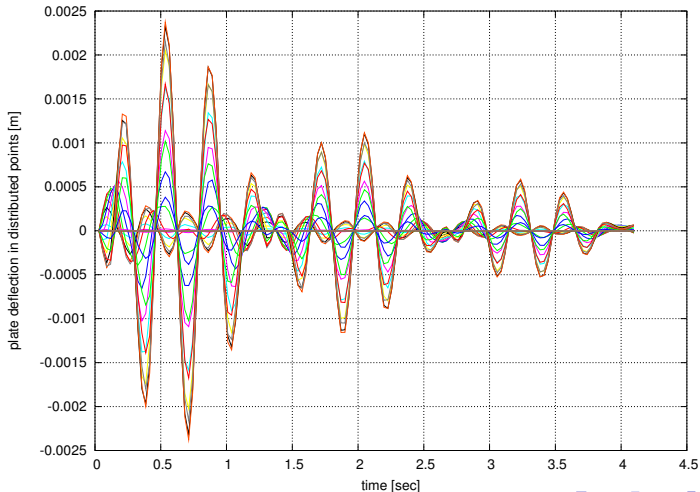


Flutter mode appears for  $M_\infty \geq 2.27$  (or  $N_T = 4.3438 \times 10^{-5}$  and  $N_M = 0.055$ ).

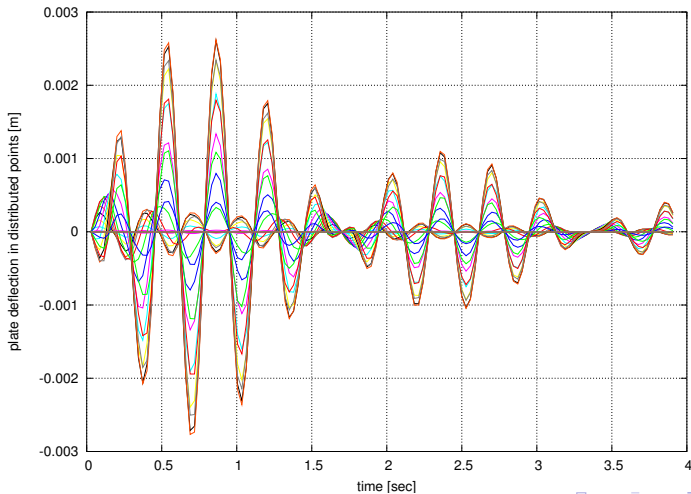
## Staged FSI-FEM Results for flutter: Structure response at $M=1.8$



## Structure response at $M=2.0$

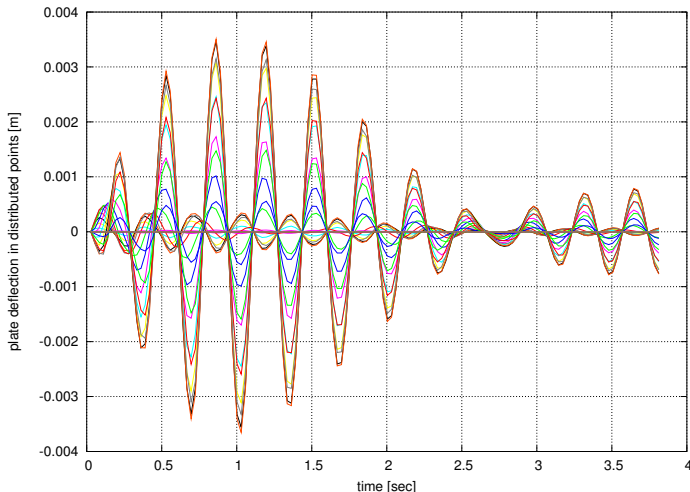


## Structure response at $M=2.1$

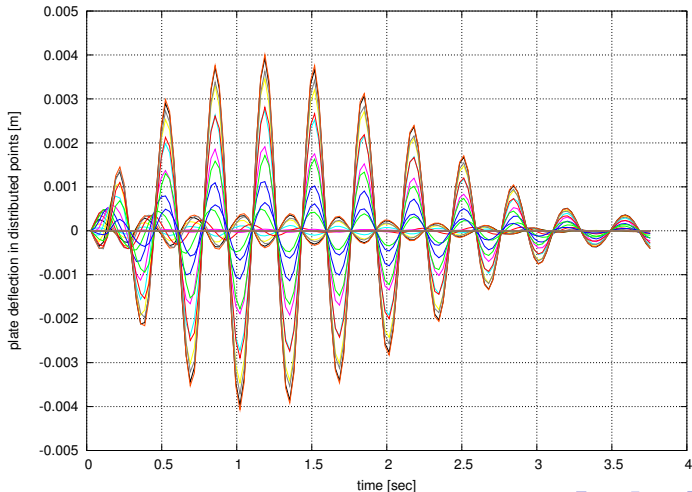




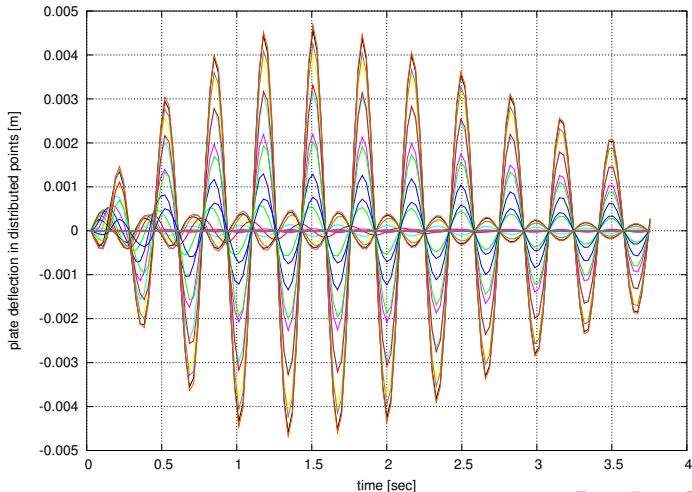
## Structure response at $M=2.2$



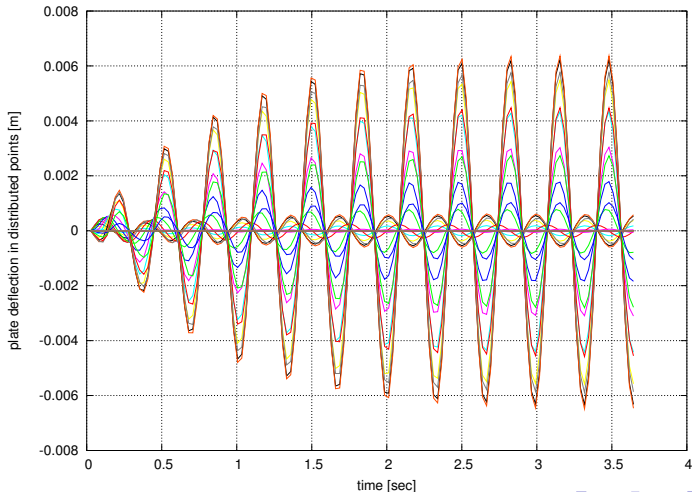
## Structure response at $M=2.225$



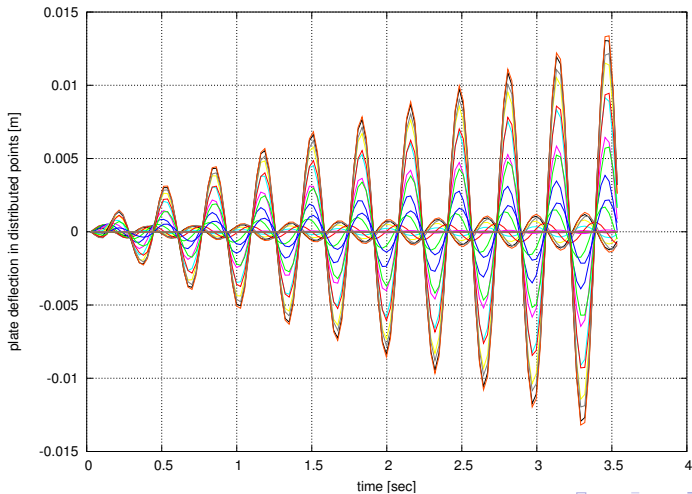
## Structure response at $M=2.25$



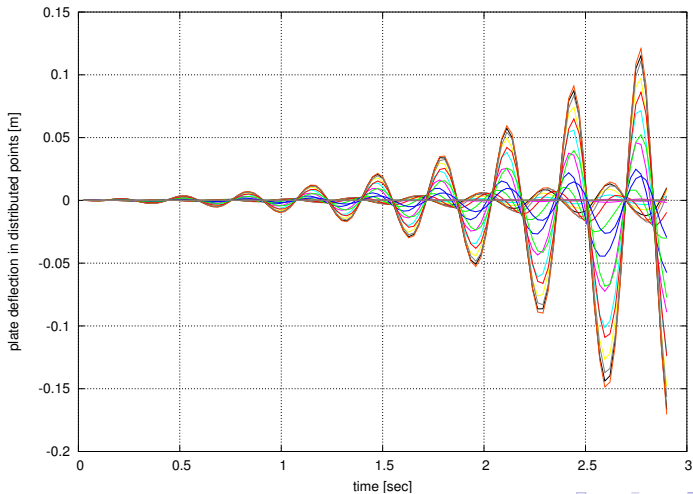
## Structure response at $M=2.275$



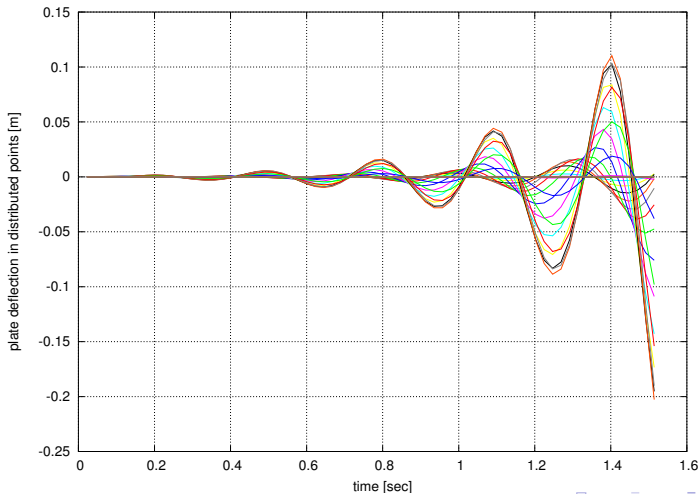
## Structure response at $M=2.3$



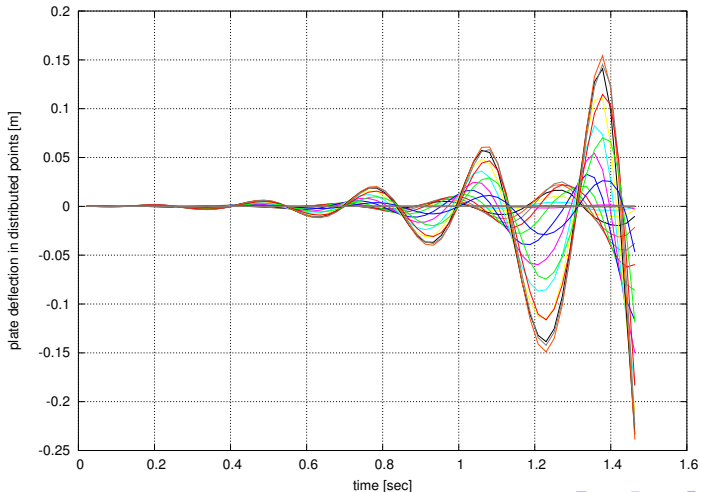
## Structure response at $M=2.5$



## Structure response at $M=3.0$

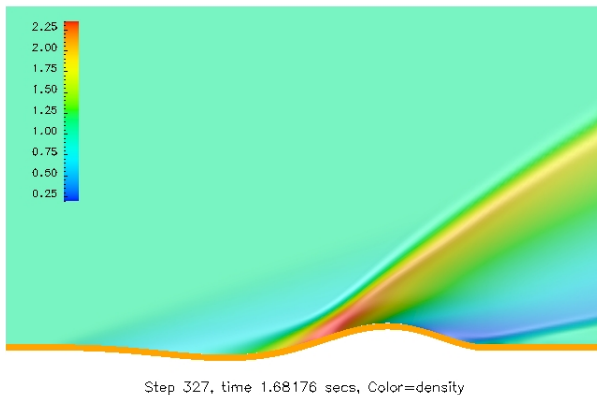


## Structure response at $M=3.2$





## Flat plate in Flutter



- Being

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

sweep in  $N_T$ ,  $N_M$  and  $M_\infty$  estimated the **flutter region** as

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

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

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

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

$$t = 0.06$$

$$\nu = 0.33$$

$$m = 0.002$$

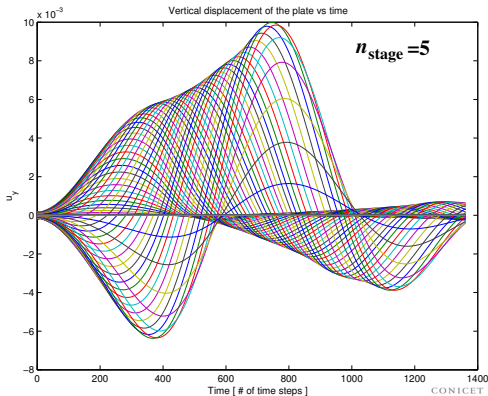
$$E = 39.6$$

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

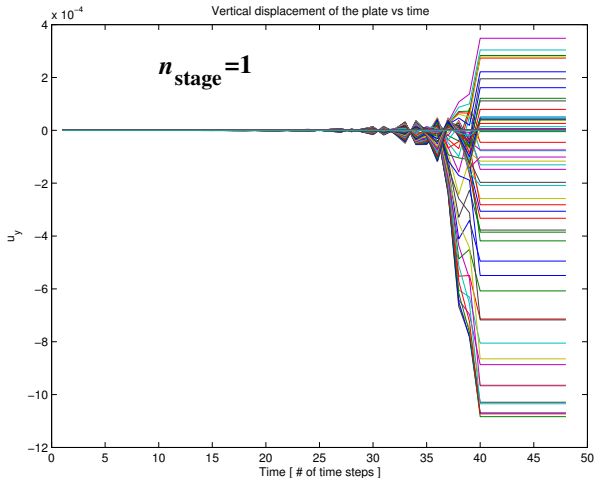
$$N_T = \frac{D}{mL^2 U_{\infty}^2} = 0.025$$

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

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



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



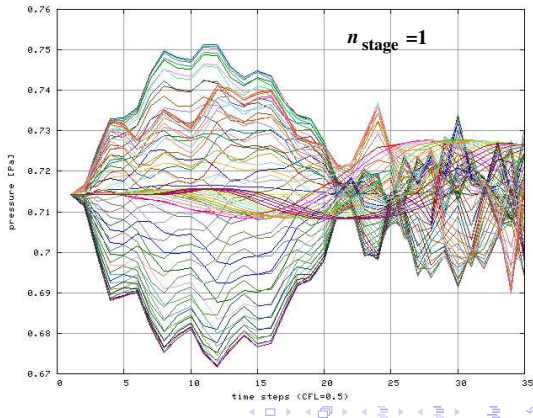
## Stability of the staged algorithm outside the flutter region, i.e. $N_M/(N_T M_\infty^2) \ll 200$

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

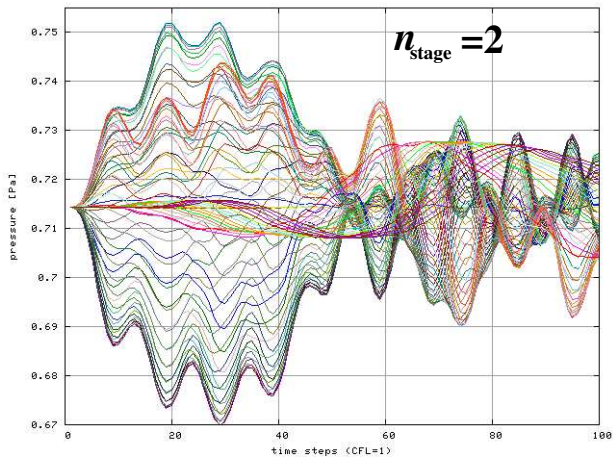
with  $m = 0.00135$

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

(i.e., outside the  
flutter region)



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



# Temario

- 1 Parte I – Métodos de Descomposición de Dominio
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 **Parte II – Aplicaciones y Usos**
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - **Conclusions**
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## 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.



## 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 Parte I – Métodos de Descomposición de Dominio
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - Agradecimientos

## 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*. Submitted. 2005.
- [8] Mario A. Storti, Norberto M. Nigro, Rodrigo R. Paz and Lisandro Dalcín.  
Dynamics boundary conditions in CFD. *Journal of Computational Physics*. Submitted. 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*, submitted. 2006.

# Temario

- 1 Parte I – Métodos de Descomposición de Dominio
  - Introducción - Motivos - Objetivos
  - Solución de Sistemas de Ecuaciones Lineales
  - Método de Descomposición de Dominio (DDM)
  - Precondicionamiento para el sistema  $Ax = b$ .
  - Conclusiones
- 2 Parte II – Aplicaciones y Usos
  - Condiciones de Borde Dinámicas en CFD
  - Aplicación en Problemas de Interacción Fluido-Estructura
  - Conclusiones
- 3 Parte III – Conclusiones Finales
  - Trabajos Publicados
  - **Agradecimientos**

## Agradecimientos

A Mario, Beto y Lisandro.

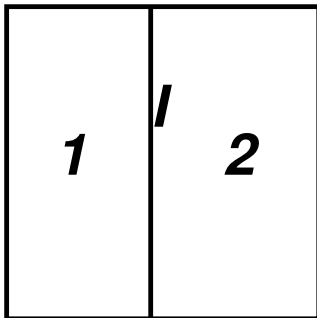
A Eliana, Guadalupe, Mis Padres, Mi Hermana, Mi Abuelo Agustín, Adrian, Juan, Sergio, Carlos, Facundo, Ruperto, Gerardo; a las Familias Aguirre y Coria; a Ricardo y Rosa, Pablo, Ángel; al CIMEC-INTEC-CONICET-UNL; a Vitoriano Ruas, Grigory Panasenko, Victorio, Néstor; a Cleve Ashcraft y John Hallquist; Jorge, Laura, Gustavo, Martín. A Gustavo Torresán y a mis compañeros de facultad.

::GRACIAS::



## Iteración Global

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

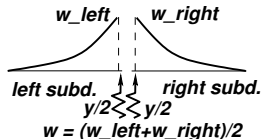
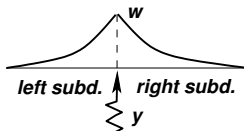


- En iteración global (e.g., algoritmo CG) hay que calcular los productos matriz-vector, i.e., calcular los términos diagonales  $\mathbf{A}_{11}\mathbf{x}_1$  y  $\mathbf{A}_{22}\mathbf{x}_2$  en cada procesador y,
- Comunicar las contribuciones no-diagonales que correspondan.



## Precondicionador Neumann-Neumann para IISD

- Cuando resolvemos el problema reducido de la interface  $\mathbf{S}x_I = y_I$ , el preconditionador resuelve en forma iterativa (aproximada)  $\mathbf{S}w = y$  para  $w$  en términos de  $y$ .
- Para el problema de Laplace  $P_{NN}^{-1}S$  es equivalente a aplicar “un flujo concentrado de calor” ( $\delta$  de Dirac) y en la interface  $y$  resolver para el campo de temperaturas. La traza en la interface es el valor de  $w$ .
- El preconditionador de Neumann-Neumann separa (divide) el flujo de calor en la mitad para cada lado de la interface (si dicha interface es compartida por dos subdominios).



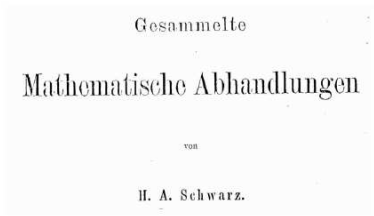
## Splitting de los flujos - Prec. Neumann-Neumann

- El prec. de Neumann-Neumann funciona bien cuando los operadores y los subdominios son simétricos. Las autofunciones del operador de Steklov son simétricas.
- En presencia de términos advectivos (presentes en la mayoría de las aplicaciones CFD), la los subdominios que se encuentran aguas abajo deberían tener mayor peso en la separación de los flujos a ambos lados de la interface.
- El preconditionador tiene la siguiente forma:

$$P_{NN}^{-1} = (I - \underbrace{P_0}_{\text{'coarse'}}) \left( \sum_{i=1}^{N_{\text{subdo}}} \underbrace{R_i^T D_i S_i^\dagger D_i R_i}_{\text{operador local}} \right) (I - P_0^T)$$

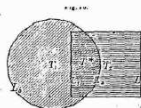
## Descomposición de Dominio ‘Solapados’ (Overlapping DDM).

- Método Alternante de Schwarz (Additive Schwarz Method).

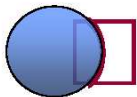


Ueber einen Grenzübergang durch alternirendes  
Verfahren.

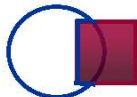
Zürich, im August 1870.



What Schwarz proposed...



Solve PDE in circle  
with BC taken from  
interior of square



Solve PDE in square  
with BC taken from  
interior of circle

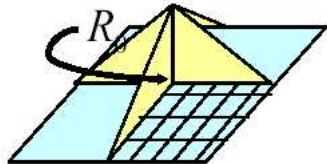
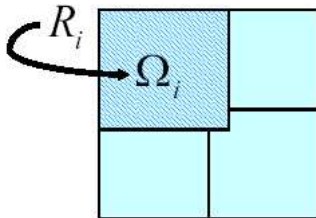
And iterate!

## Método Alternante de Schwarz

- Expresión matemática del preconditionador:

$$P^{-1} = \underbrace{R_0^T A_0^{-1} R_0}_{\text{operador 'global' o 'coarse'}} + \sum_{i=1}^{N_{\text{subdo}}} \underbrace{R_i^T A^{-1} R_i}_{\text{operador local o distribuido}}$$

- $R_i$  es el operador de restricción y  $R_i^T$  es el operador de extensión.



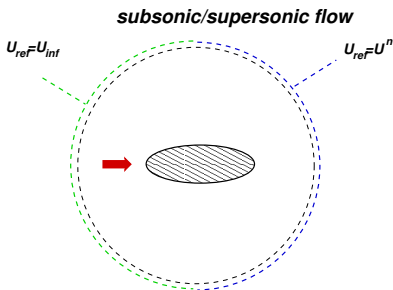
## A.B.C's based on Riemann Invariants

- 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:

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

- 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).

- **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.



## Using penalization

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

$$\begin{aligned} -\epsilon\lambda + \Pi^+ (\mathbf{U}_0 - \mathbf{U}_{\text{ref}}) + \Pi^- \lambda &= 0, \\ \mathbf{C} \frac{\mathbf{U}_0^{n+1} - \mathbf{U}_0^n}{\Delta t} + \mathbf{A} \frac{\mathbf{U}_1^{n+1} - \mathbf{U}_0^n}{h} + \mathbf{C}\Pi^+ \lambda &= 0; \end{aligned}$$

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

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

## Global basis for Houbolt's model

- Using a global basis for displacements

$$u(x) = \sum_{k=1}^N a_k \psi_k(x),$$
$$\psi_k(x) = \frac{4x(L-x)}{L^2} \sin(k\pi x/L).$$

- The basis functions satisfy the essential boundary conditions for plate equation  $u = (\partial u / \partial x) = 0$  at  $x = 0, L$ .



- Replacing  $u(x)$  in the Houbolt's approximation and using Galerkin method

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

where

$$M_{jk} = \int_0^L m \psi_j(x) \psi_k(x) dx,$$

$$K_{jk} = \int_0^L D \psi_j''(x) \psi_k''(x) dx,$$

$$H_{x,jk} = \int_0^L C_x \psi_j(x) \psi_k'(x) dx,$$

$$H_{t,jk} = \int_0^L C_t \psi_j(x) \psi_k(x) dx.$$

## Flutter Region Predictions

- If  $(\partial u / \partial t)$  is neglected (i.e., characteristic times of struct. are much lower than those of the fluid,  $N_T \ll 1$ ),

$$\det(\bar{\lambda}^2 \bar{\mathbf{M}} + \mathbf{K} + \mathbf{H}_x) = 0$$

$$\bar{\lambda} = \sqrt{m} \lambda,$$

$$\bar{\mathbf{M}} = \frac{1}{\sqrt{m}} \mathbf{M},$$

the coefficients in  $\bar{\mathbf{M}}$ ,  $\mathbf{K}$ ,  $\mathbf{H}_x$  do not depend on  $m$ , neither do the eigenvalues of equation, then the  $\lambda$  eigenvalues are of the form

$$\lambda_j = \frac{\bar{\lambda}_j}{\sqrt{m}},$$

with  $\bar{\lambda}$  not depending on  $m$ . This means that the sign of the real part of the  $\lambda$  is independent of  $m$ .