



Universidad Nacional Autónoma de México Instituto de Geofísica

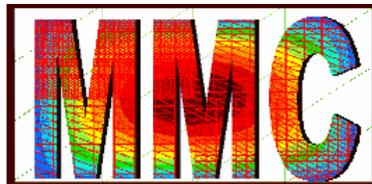
“Aplicación del Cómputo Paralelo a la Modelación de Sistemas Continuos en Ciencias e Ingeniería”

Presentan:

Antonio Carrillo Ledesma

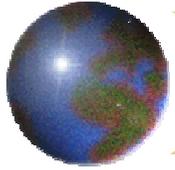
Ismael Herrera Revilla

*4 to Congreso Internacional y 2do Nacional de Métodos
Numéricos en Ingeniería y Ciencias Aplicadas
Morelia, Michoacán, México*



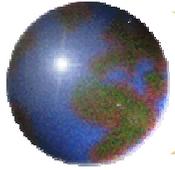
Modelación Matemática y Computacional

Enero de 2007

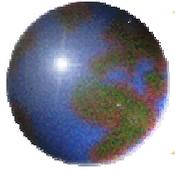


Contenido

- ⊕ Motivación
- ⊕ Objetivos
- ⊕ Ejemplo a Desarrollar
- ⊕ Método de Elemento Finito Secuencial
- ⊕ Cómputo en Paralelo
- ⊕ Método de Subestructuración Paralelo Precondicionado
- ⊕ Análisis de Rendimiento
- ⊕ Conclusiones



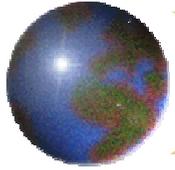
Motivación



Motivación

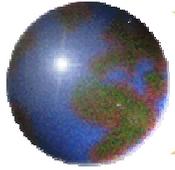
Uno de los grandes retos del área de cómputo científico es poder analizar a priori una serie de consideraciones dictadas por factores externos a la solución del problema de interés, estas consideraciones influirán de manera decisiva en la implementación computacional de la solución numérica, algunas de estas son:

- ✦ Número de Procesadores Disponibles.
- ✦ Tamaño y Tipo de Partición del Dominio.
- ✦ Tiempo de Ejecución Predeterminado.

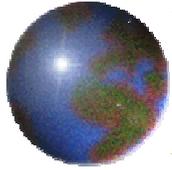


Motivación ...

- Los esquemas de discretización tradicionales tipo diferencias finitas y elemento finito generan sistemas algebraicos de ecuaciones lineales y estos pueden ser de gran tamaño.
- Estos algoritmos son secuenciales y su implementación se hace en equipos de cómputo secuencial, teniendo como limitante del problema a tratar el número de nodos n de la discretización ya que el sistema algebraico asociado es de orden n^2 .
- En la actualidad es una práctica común tomar estos algoritmos secuenciales y mediante directivas de compilación tratar de paralelizarlo. Esta solución no es óptima ya que desperdicia recursos computacionales.

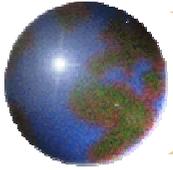


Objetivos



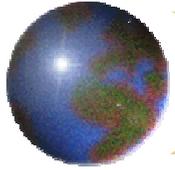
Objetivos Generales

- ✦ Mostrar las bases de una metodología que se utiliza para aplicar el cómputo en paralelo a la modelación matemática y computacional de sistemas continuos de forma flexible, escalable y eficiente.
- ✦ Mostrar los alcances y limitaciones de la metodología usando como herramientas de evaluación a los métodos de elemento finito secuencial, método de subestructuración secuencial y método de subestructuración paralelo.
- ✦ Mostrar los diversos esquemas de optimización aplicables a la metodología.

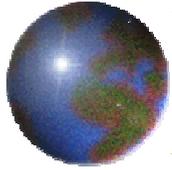


Objetivos Particulares

- ✦ Mostrar cómo aplicar la metodología para manejar problemas de gran tamaño (descomposición de malla fina).
- ✦ Mostrar cómo descomponer un dominio en un conjunto de subdominios que den una partición en la que el tiempo de cálculo sea mínimo para una configuración de hardware dada.
- ✦ Mostrar cuales son las posibles optimizaciones aplicables a una configuración de hardware dada.



Ejemplo a Desarrollar

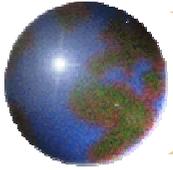


Ejemplo a Desarrollar

Para ejemplificar las ideas básicas de esta metodología, se ha tomado la ecuación de Poisson:

$$\begin{aligned} -\nabla^2 u &= f_{\Omega} \text{ en } \Omega \\ u &= g_{\partial\Omega} \text{ en } \partial\Omega. \end{aligned}$$

Este es un ejemplo muy sencillo pero gobierna muchos sistemas de la ingeniería y la ciencia, entre ellos el flujo de agua subterránea a través de un acuífero isotrópico, homogéneo bajo condiciones de equilibrio. También es muy usada en la conducción de calor en un sólido bajo condiciones de equilibrio.

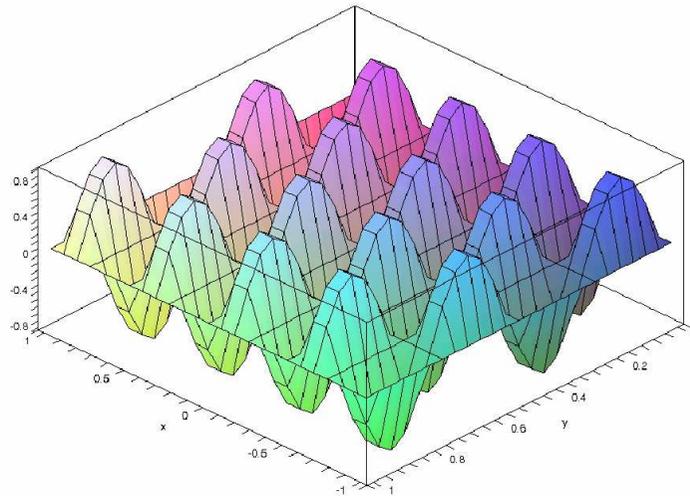


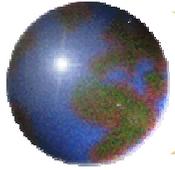
Ejemplo a Desarrollar ...

En particular tomemos $\Omega = [-1, 1] \times [0, 1]$, con $n \in \mathbb{N}$,
 $f_{\Omega} = 2n^2\pi^2 \sin(n\pi x) * \sin(n\pi y)$ y $g_{\partial\Omega} = 0$.

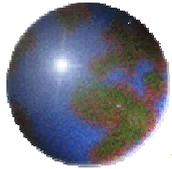
Cuya solución es $u(x, y) = \sin(n\pi x) * \sin(n\pi y)$.

Si por ejemplo tomamos $n = 4$, la solución se ve como:



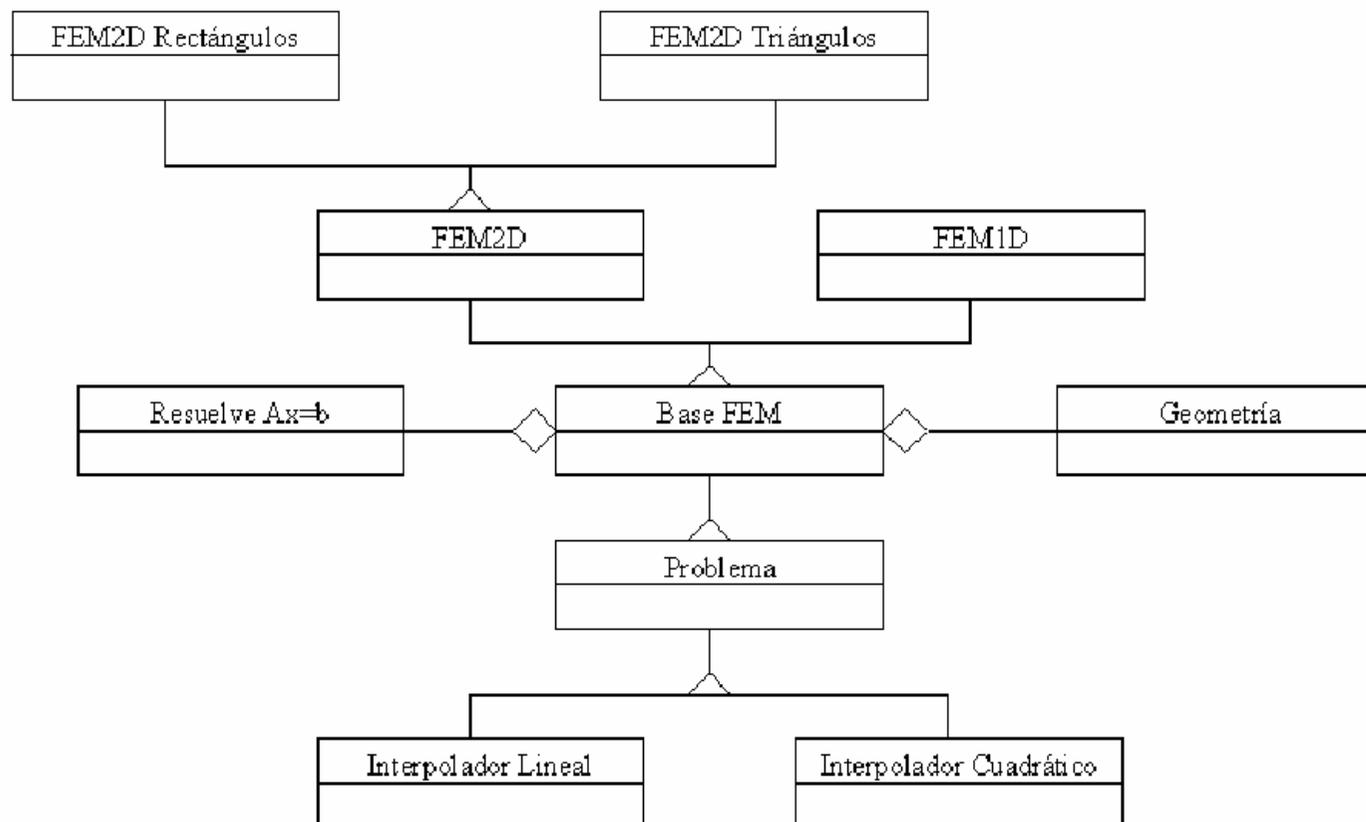


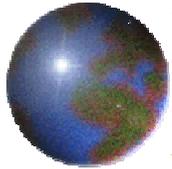
Método de Elemento Finito Secuencial



Método de Elemento Finito ...

Jerarquía de clases para el método de elemento finito:



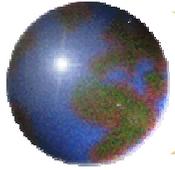


Método de Elemento Finito ...

Si descomponemos el dominio Ω en una partición rectangular de 81x81 nodos, genera 6400 elementos rectangulares con 6561 nodos en total, donde 6241 son desconocidos. El sistema algebraico asociado es de 6241x6241, resolviendo por diversos métodos numéricos:

Método Iterativo	Iteraciones	Tiempo Total
Jacobi	14115	9511 seg.
Gauss-Seidel	7359	4962 seg.
Gradiente Conjugado	91	150 seg.

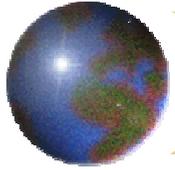
Precondicionador	Iteraciones	Tiempo Total
Jacobi	89	150 seg.
SSOR	85	149 seg.
Factorización Incompleta	79	149 seg.



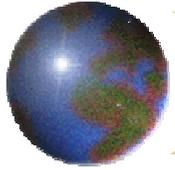
Método de Elemento Finito ...

Limitaciones del método:

- ⊕ Al aumentar el número de nodos n , el sistema algebraico crece del orden n^2 .
- ⊕ El número de condicionamiento de la matriz del método de FEM es del orden de $\frac{1}{h^2}$, donde h es el diámetro de la malla.
- ⊕ Un solo equipo de cómputo puede no soportar al sistema algebraico cuando este proviene de una descomposición fina.
- ⊕ Si se paraleliza, es necesario usar memoria compartida para soportar el sistema algebraico.
- ⊕ Es un algoritmo secuencial, por ello no es eficiente su paralelización.



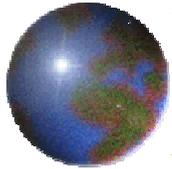
Cómputo Paralelo



Cómputo Paralelo

Entenderemos por una arquitectura en paralelo a un conjunto de procesadores interconectados capaces de cooperar en la solución de un problema.

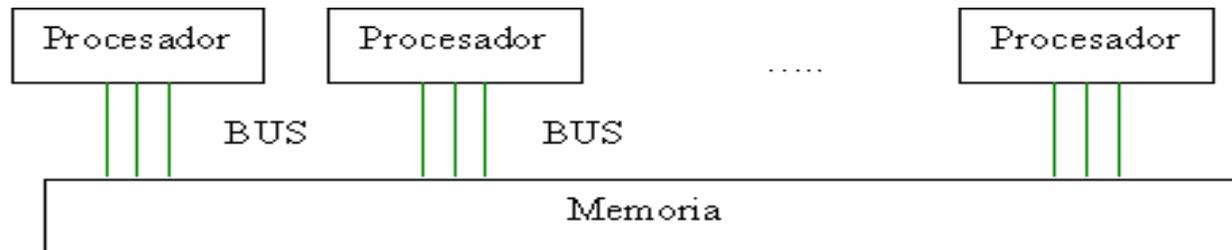
- ✦ La computación en paralelo es una técnica que nos permite distribuir una gran carga computacional entre muchos procesadores.
- ✦ Y es bien sabido que una de las mayores dificultades del procesamiento en paralelo es la coordinación de las actividades de los diferentes procesadores y el intercambio de información entre los mismos.



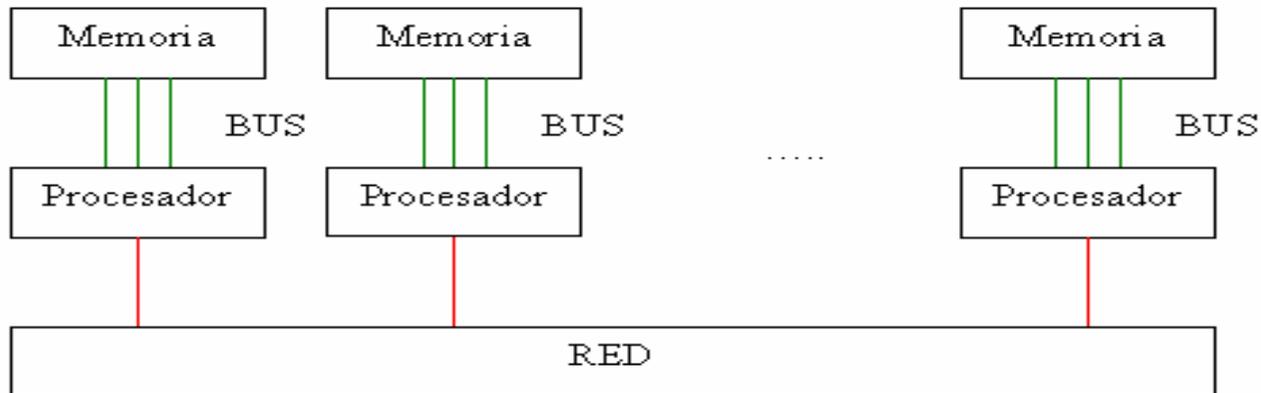
Cómputo Paralelo ...

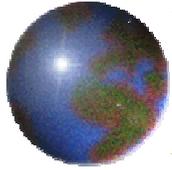
Arquitecturas Paralelas

● Memoria Compartida (multiprocesadores)



● Memoria Distribuida (multicomputadoras)



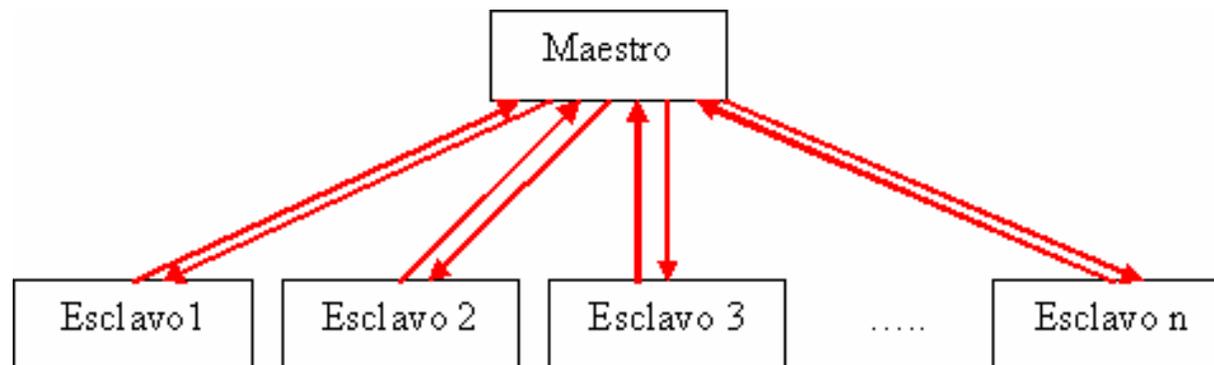


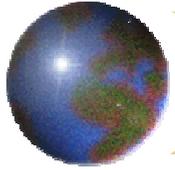
Cómputo en Paralelo ...

Ventajas del uso de Clusters de PC's

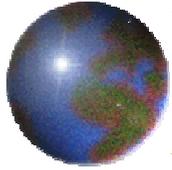
- ✚ La construcción y puesta en marcha de un cluster es barata.
- ✚ Remplazar componentes defectuosos y escalar el cluster es sencillo.

Cluster (Bajo Esquema Maestro-Esclavo)





Método de Subestructuración Paralelo Precondicionado



Método de Subestructuración ...

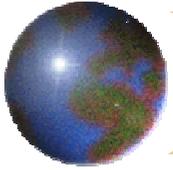
El dominio Ω ahora es dividido en $E = n \times m$ subdominios $\Omega_i, i = 1, 2, 3, \dots, E$ tales que

$$\Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j \quad \text{y} \quad \bar{\Omega} = \bigcup_{i=1}^E \bar{\Omega}_i,$$

y al conjunto $\Sigma = \bigcup_{i=1}^E \Sigma_i, \quad \text{si} \quad \Sigma_i = \partial\Omega_i \setminus \partial\Omega$

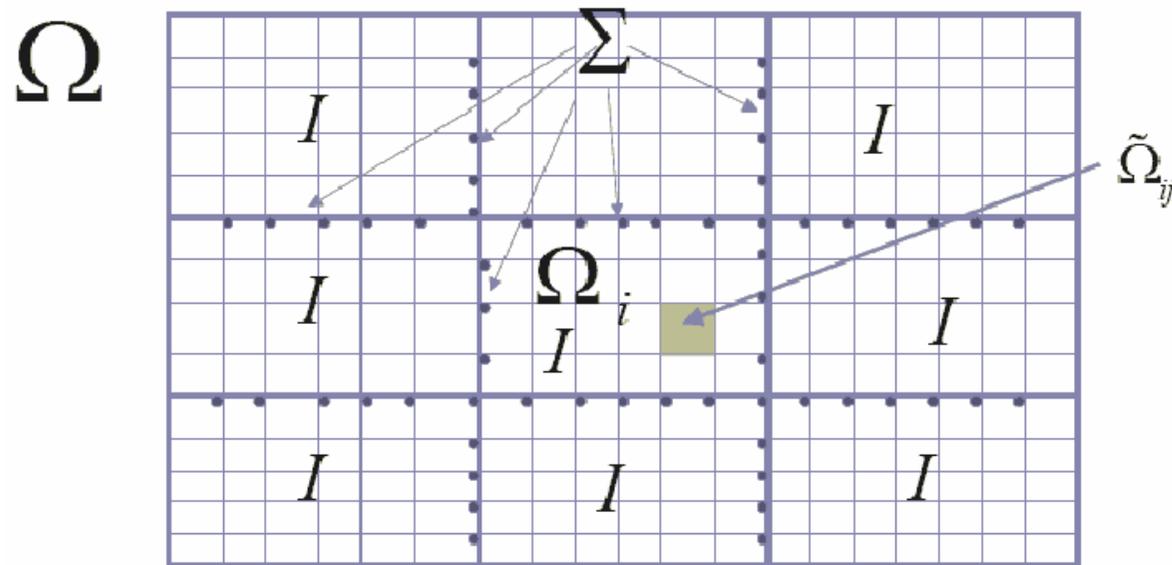
lo llamaremos la frontera interior de los subdominios.

Adicionalmente cada Ω_i es descompuesto en $p \times q$ elementos, obtenemos así la descomposición fina del dominio.

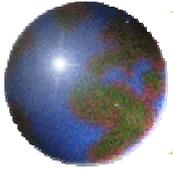


Método de Subestructuración ...

Un ejemplo de una descomposición del dominio se muestra en la siguiente figura:



Descomposición 3x3 y 6x5



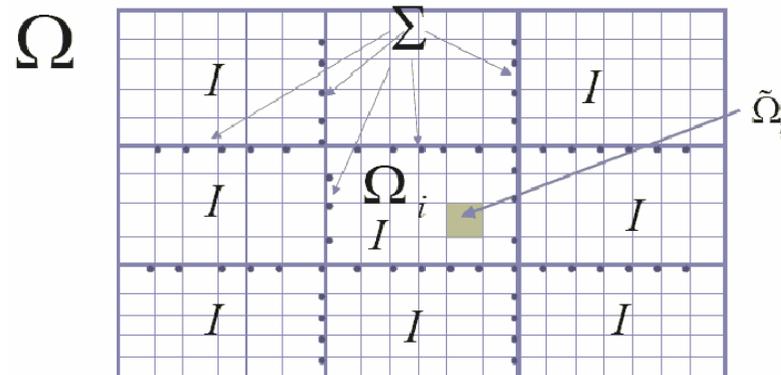
Método de Subestructuración ...

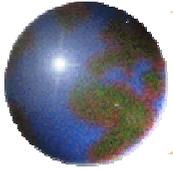
Definiendo ahora para cada subdominio Ω_δ , con $\delta = 1, \dots, E$
las matrices

$$\underline{\underline{A}}_\delta^{II} \equiv \left[\langle w_\delta^i, w_\delta^j \rangle \right] \quad \underline{\underline{A}}_\delta^{\Sigma\Sigma} \equiv [\langle w_\Sigma^\alpha, w_\Sigma^\alpha \rangle]$$

y

$$\underline{\underline{A}}_\delta^{I\Sigma} \equiv [\langle w_I^i, w_\Sigma^\alpha \rangle], \quad \underline{\underline{A}}_\delta^{\Sigma I} \equiv [\langle w_\Sigma^\alpha, w_I^i \rangle]$$





Método de Subestructuración ...

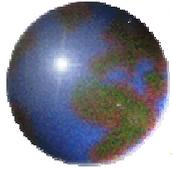
Pero como nosotros tenemos definidas de forma local en cada Ω_i a las matrices $\underline{\underline{A}}_i^{\Sigma\Sigma}$, $\underline{\underline{A}}_i^{\Sigma I}$, $\underline{\underline{A}}_i^{I\Sigma}$ y $\underline{\underline{A}}_i^{II}$, entonces el complemento

de Schur local queda definido por $\underline{\underline{S}}_i = \underline{\underline{A}}_i^{\Sigma\Sigma} - \underline{\underline{A}}_i^{\Sigma I} \left(\underline{\underline{A}}_i^{II} \right)^{-1} \underline{\underline{A}}_i^{I\Sigma}$.

Generándose el sistema virtual cuyas incógnitas son los nodos de la frontera interior

$$\left[\sum_{i=1}^E \underline{\underline{S}}_i \right] \underline{\underline{u}}_{\Sigma} = \left[\sum_{i=1}^E \underline{\underline{b}}_i \right]$$

donde $\underline{\underline{b}}_i = \underline{\underline{A}}_i^{\Sigma I} \left(\underline{\underline{A}}_i^{II} \right)^{-1} \underline{\underline{b}}_{I_i}$.



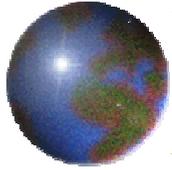
Método de Subestructuración ...

Usando el método de gradiente conjugado preconditionado, resolvemos los nodos de la frontera interior \underline{u}_Σ .

Para ello, es necesario hacer por cada iteración del método una multiplicación de un vector de la dimensión de \underline{u}_Σ con todos los \underline{S}_i , esto se logra pasando sólo las entradas correspondientes al subdominio y así poderlo multiplicar por \underline{S}_i .

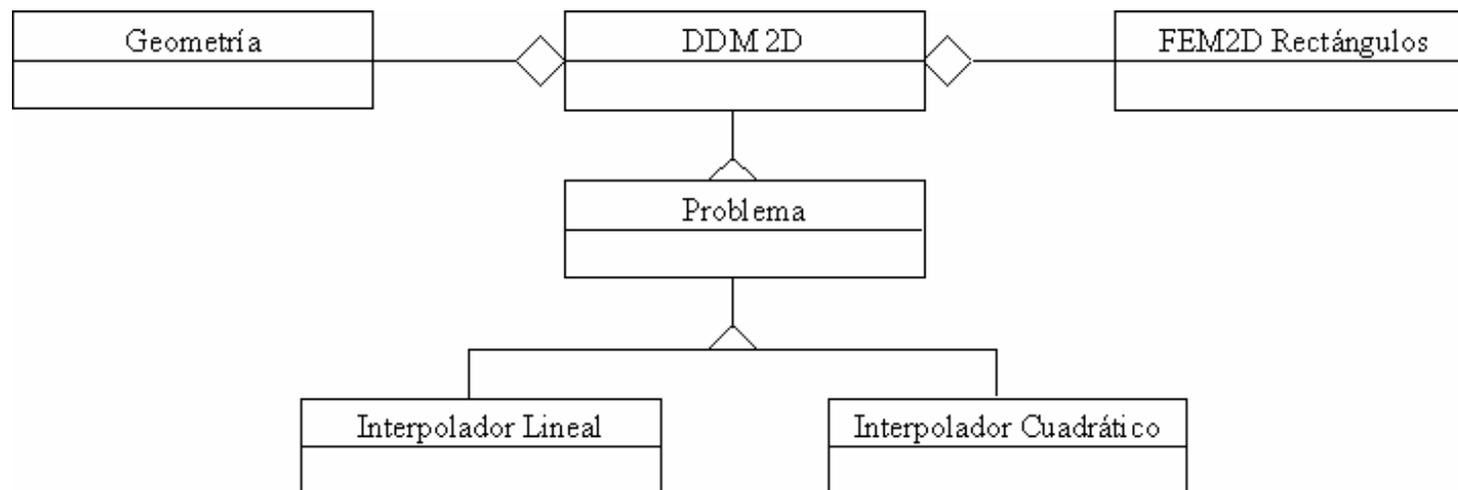
El número de condicionamiento de la matriz del método de subestructuración es a lo más del orden de $\frac{1}{h}$.

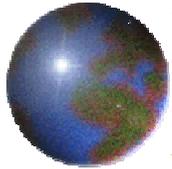
Una vez conocido \underline{u}_Σ se calcula $\underline{u}_{I_i} = \left(\underline{A}_i^{II}\right)^{-1} \left(\underline{b}_{I_i} - \underline{A}_i^{\Sigma I} \underline{u}_{\Sigma_i}\right)$ resolviendo con esto los nodos interiores.



Método de Subestructuración ...

Jerarquía de clases para el método de subestructuración paralelo
precondicionado:





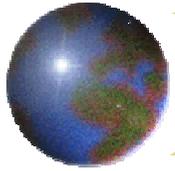
Método de Subestructuración ...

Para resolver el mismo dominio Ω del ejemplo anterior (FEM cuyo tiempo de ejecución es de 149 seg), podemos construir algunas de las siguientes descomposiciones en un procesador:

Descomposición	Nodos Interiores	Subdominios	Elementos Subdominio	Total Nodos Subdominio	Nodos Desconocidos Subdominio
2x2 y 40x40	6084	4	1600	1681	1521
4x4 y 20x20	5776	16	400	441	361
8x8 y 10x10	5184	64	100	121	81
16x16 y 5x5	4096	256	25	36	16

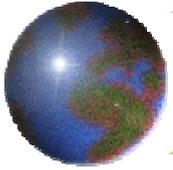
Obteniendo los siguientes resultados:

Partición	Nodos Frontera Interior	Iteraciones	Tiempo Total
2x2 y 40x40	157	66	1040 seg.
4x4 y 20x20	465	92	68 seg.
8x8 y 10x10	1057	128	23 seg.
16x16 y 5x5	2145	169	87 seg.

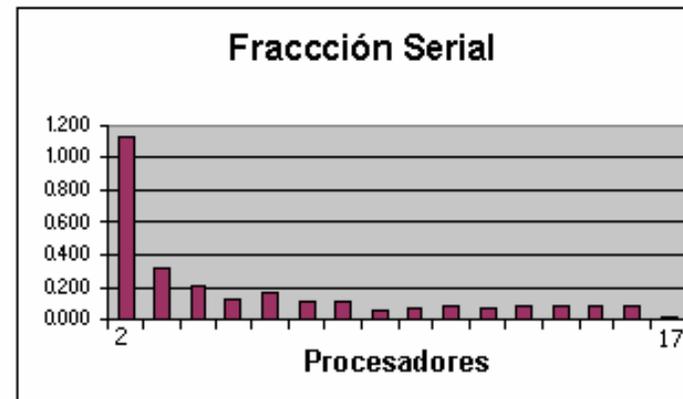
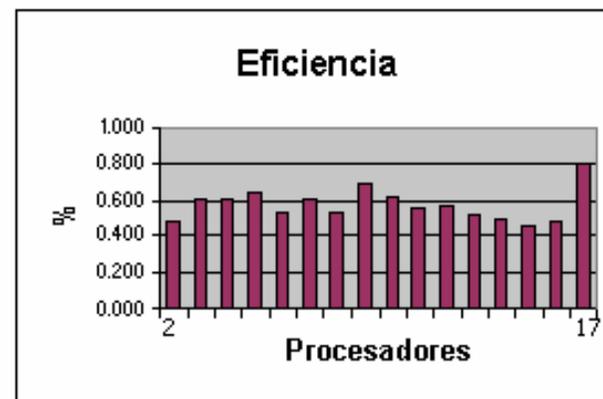
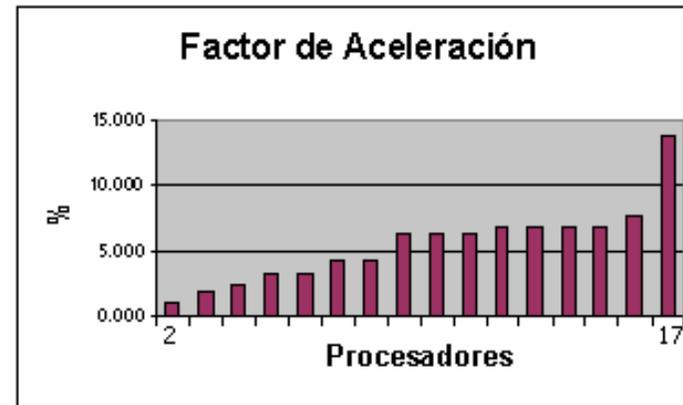
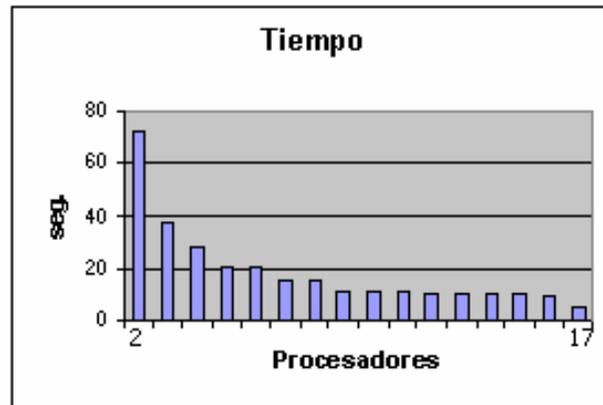


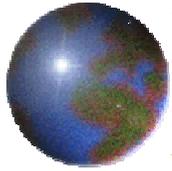
Método de Subestructuración ...

Procesadores	Tiempo	Factor de Aceleración	Eficiencia	Fracción Serial
1	68 seg.			
2	72 seg.	0.94444	0.47222	1.11764
3	37 seg.	1.83783	0.61261	0.31617
4	28 seg.	2.42857	0.60714	0.21568
5	21 seg.	3.23809	0.64761	0.13602
6	21 seg.	3.23809	0.53968	0.17058
7	16 seg.	4.25000	0.60714	0.10784
8	16 seg.	4.25000	0.53125	0.12605
9	11 seg.	6.18181	0.68686	0.05698
10	11 seg.	6.18181	0.61818	0.06862
11	11 seg.	6.18181	0.56198	0.07794
12	10 seg.	6.80000	0.56666	0.06951
13	10 seg.	6.80000	0.52307	0.07598
14	10 seg.	6.80000	0.48571	0.08144
15	10 seg.	6.80000	0.45333	0.08613
16	9 seg.	7.55555	0.47222	0.07450
17	5 seg.	13.60000	0.80000	0.01562

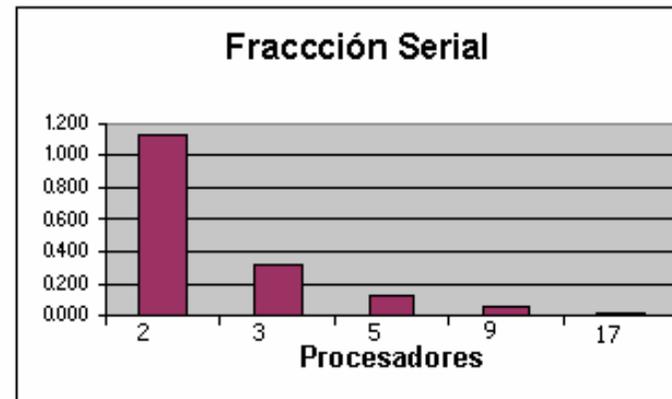
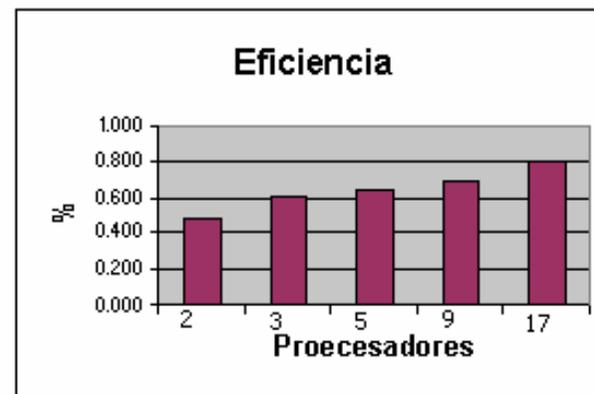
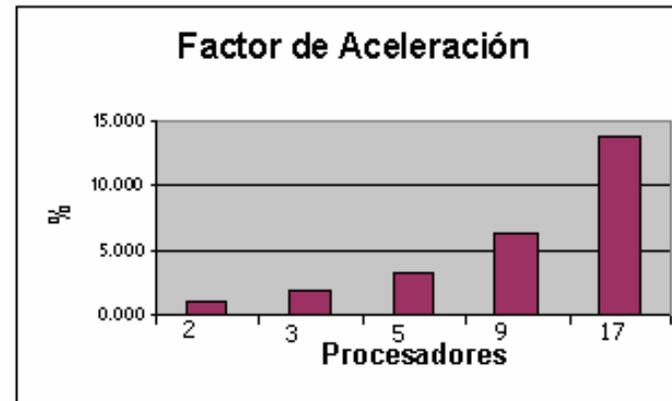
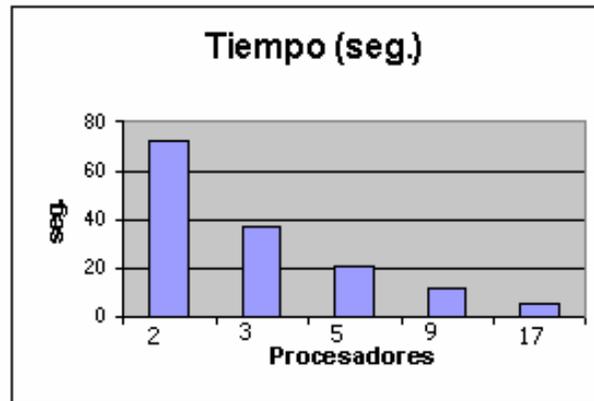


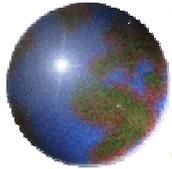
Método de Subestructuración ...





Método de Subestructuración ...





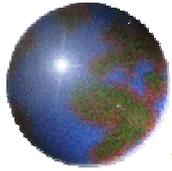
Método de Subestructuración ...

Método de Subestructuración Paralelo Precondicionado

Procesadores	Tiempo	Factor de Aceleración	Eficiencia	Fracción Serial
1	68 seg.			
2	72 seg.	0.94444	0.47222	1.11764
3	37 seg.	1.83783	0.61261	0.31617
5	21 seg.	3.23809	0.64761	0.13602
9	11 seg.	6.18181	0.68686	0.05698
17	5 seg.	13.60000	0.80000	0.01562

Método de Subestructuración Paralelo

Procesadores	Tiempo	Factor de Aceleración	Eficiencia	Fracción Serial
1	68 seg.			
2	73 seg.	0.93150	0.46575	1.14705
3	38 seg.	1.78947	0.59649	0.33823
5	22 seg.	3.09090	0.61818	0.15441
9	12 seg.	5.66666	0.62962	0.07352
17	6 seg.	11.33333	0.66666	0.03125



Método de Subestructuración ...

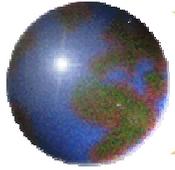
Descomposición óptima para una configuración de hardware paralelo dada. Deseamos resolver el mismo dominio Ω descompuesto en 81×81 nodos pero ahora usando 3,4,5 y 6 procesadores

✦ Cluster homogéneo

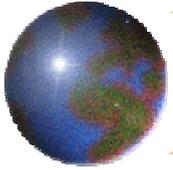
Partición	Tiempo en 3 Procesadores	Tiempo en 4 Procesadores	Tiempo en 5 Procesadores	Tiempo en 6 Procesadores
2×2 y 40×40	533 seg.	533 seg.	267 seg.	—
4×4 y 20×20	41 seg.	31 seg.	24 seg.	24 seg.
8×8 y 10×10	40 seg.	42 seg.	41 seg.	42 seg.
16×16 y 5×5	184 seg.	188 seg.	172 seg.	150 seg.

✦ Cluster heterogéneo

Partición	Tiempo en 3 Procesadores	Tiempo en 4 Procesadores	Tiempo en 5 Procesadores	Tiempo en 6 Procesadores
2×2 y 40×40	558 seg.	533 seg.	255 seg.	—
4×4 y 20×20	36 seg.	25 seg.	21 seg.	17 seg.
8×8 y 10×10	24 seg.	24 seg.	24 seg.	24 seg.
16×16 y 5×5	125 seg.	133 seg.	140 seg.	144 seg.

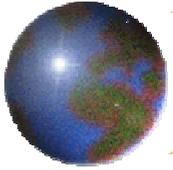


Análisis de Rendimiento



Análisis de Rendimiento

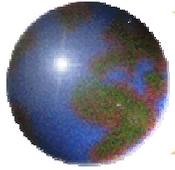
- ✦ Afectación del rendimiento al aumentar el número de subdominios en la descomposición.
- ✦ Consideraciones para aumentar el rendimiento.



Análisis de Rendimiento ...

Afectación del rendimiento al aumentar el número de subdominios en la descomposición es debido a que:

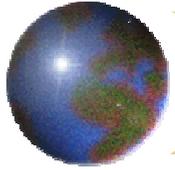
- ✦ En cada subdominio se deberán generar $\underline{\underline{A}}_i^{II}$, $\underline{\underline{A}}_i^{\Sigma\Sigma}$, $\underline{\underline{A}}_i^{\Sigma I}$, $\underline{\underline{A}}_i^{I\Sigma}$ y calcular $\left(\underline{\underline{A}}_i^{II}\right)^{-1}$.
- ✦ Si el número de subdominios es pequeño, el tamaño de las matrices será grande. El costo computacional de su generación será alto.
- ✦ Si por el contrario el número de subdominios es grande, el tamaño de las matrices es pequeño, pero el esquema maestro-esclavo se degradará inexorablemente al aumentar el número de subdominios por el incremento de trabajo en el nodo maestro.



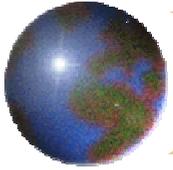
Análisis de Rendimiento ...

Consideraciones para aumentar el rendimiento

- ⊕ Balance de cargas de trabajo homogéneo.
- ⊕ Usar el mejor preconditionador a priori para ese problema.
- ⊕ Usar bibliotecas que optimizan las operaciones en el manejo de los elementos de las matrices (densas y/o bandadas).
- ⊕ Usar al momento de compilar los códigos, directivas de optimización.
- ⊕ Implementar otras estrategias de paralelización
 - ⊞ Al generar las matrices.
 - ⊞ Al realizar los cálculos requeridos entre las matrices y el vector.
 - ⊞ A nivel del compilador.

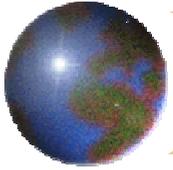


Conclusiones



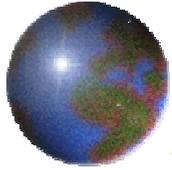
Conclusiones

- ✦ Conjuntando los métodos de descomposición de dominio, la programación orientada a objetos y esquemas de paralelización que usan el paso de mensajes, es posible construir aplicaciones que coadyuven a la solución de problemas en dos o más dimensiones concomitantes en ciencia e ingeniería, los cuales pueden ser de tamaño considerable.
- ✦ Las aplicaciones desarrolladas bajo este paradigma serán eficientes, flexibles y escalables; a la vez que son abiertas a nuevas tecnologías y desarrollos computacionales y al ser implantados en clusters, permiten una codificación ordenada y robusta, dando con ello una alta eficiencia en la adaptación del código a nuevos requerimientos, como en la ejecución del mismo.



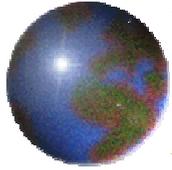
Conclusiones ...

- De forma tal que esta metodología permite tener a disposición de quien lo requiera, una gama de herramientas flexibles y escalables para coadyuvar de forma eficiente y adaptable a la solución de problemas en medios continuos de forma sistemática.



Trabajo Futuro

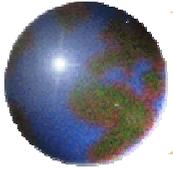
- ✦ Trabajar en otros métodos de descomposición de dominio como Trefftz-Herrera, FETI, Galerkin Discontinuo, Multigrid, entre otros. Permitiendo tener un grupo de herramientas que pueden ser usadas en múltiples problemas escogiendo la que ofrezca mayores ventajas computacionales para un problema dado.
- ✦ Construcción de preconditionadores a priori para los diversos métodos. Con la visión de que la construcción de estos sean óptimos.
- ✦ Trabajar en problemas elípticos, parabólicos e hiperbólicos, tanto lineales y no lineales. Permitiendo así, atacar una gran gama de problemas en medios continuos.

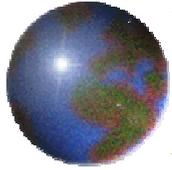


Gracias...

Nota: Los programas desarrollados en el presente trabajo pueden descargarse de la página Web:

<http://www.mmc.igeofcu.unam.mx/acl/femp>





Método de Subestructuración Secuencial

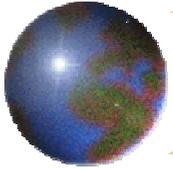
El dominio Ω ahora es dividido en $E = n \times m$ subdominios Ω_i , $i = 1, 2, 3, \dots, E$ tales que

$$\Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j \quad \text{y} \quad \bar{\Omega} = \bigcup_{i=1}^E \bar{\Omega}_i,$$

y al conjunto $\Sigma = \bigcup_{i=1}^E \Sigma_i$, si $\Sigma_i = \partial\Omega_i \setminus \partial\Omega$

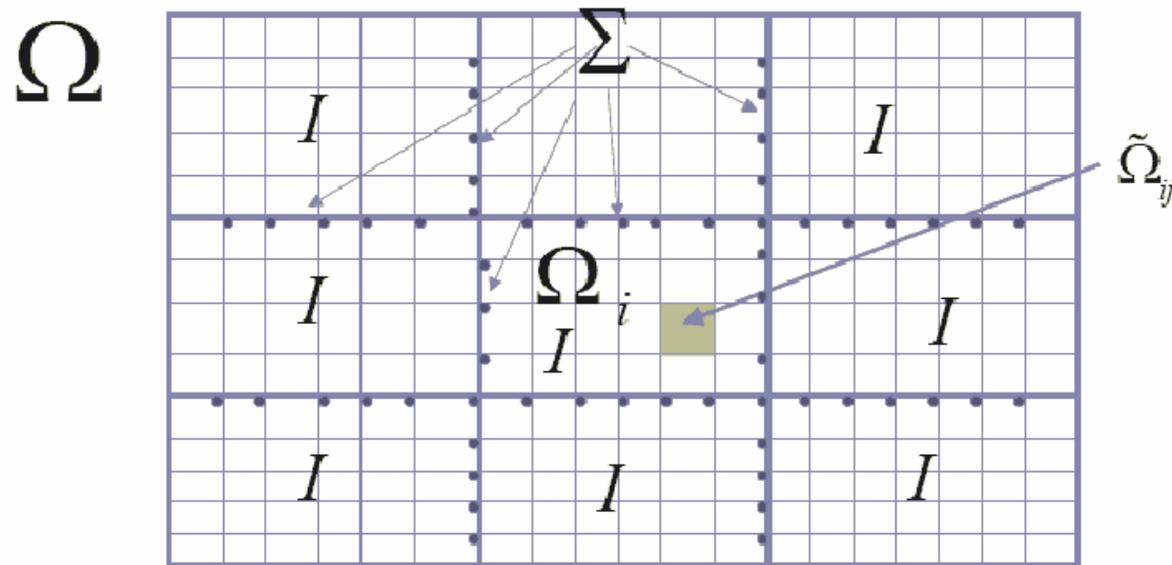
lo llamaremos la frontera interior de los subdominios.

Adicionalmente cada Ω_i es descompuesto en $p \times q$ elementos, obtenemos así la descomposición fina del dominio.

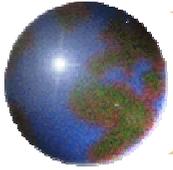


Método de Subestructuración ...

Un ejemplo de una descomposición del dominio se muestra en la siguiente figura:



Descomposición 3x3 y 6x5



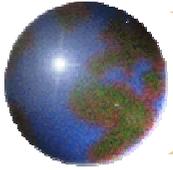
Método de Subestructuración ...

Si elegimos al conjunto $\mathbb{P}^h[k]$ como el espacio de funciones lineales ϕ_i , definidas por pedazos en cada Ω_i , de grado menor o igual a k , entonces el espacio a trabajar es

$$V^h = \text{Generado} \{ \phi_i \in \mathbb{P}^h[k] \mid \phi_i(x) = 0 \text{ en } \partial\Omega \} .$$

La solución aproximada de $\int_{\Omega} \nabla v \cdot \nabla u dx dy = \int_{\Omega} f_{\Omega} v dx dy$

queda en términos de $\int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j dx dy = \int_{\Omega} f_{\Omega} \phi_j dx dy$.



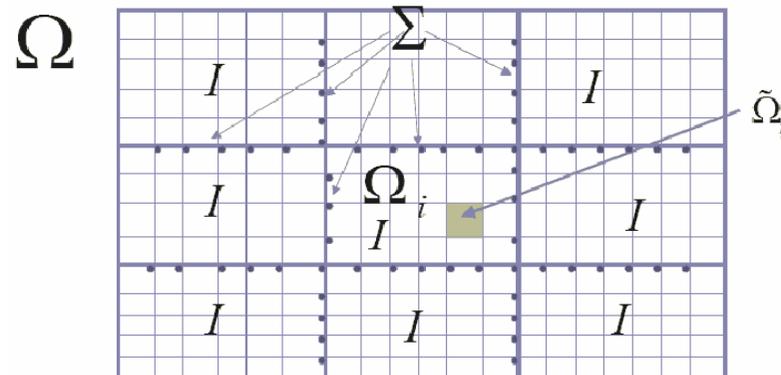
Método de Subestructuración ...

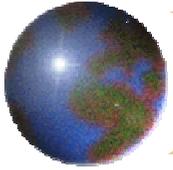
Definiendo ahora para cada subdominio Ω_δ , con $\delta = 1, \dots, E$
las matrices

$$\underline{\underline{A}}_\delta^{II} \equiv [\langle w_\delta^i, w_\delta^j \rangle] \quad \underline{\underline{A}}_\delta^{\Sigma\Sigma} \equiv [\langle w_\Sigma^\alpha, w_\Sigma^\alpha \rangle]$$

y

$$\underline{\underline{A}}_\delta^{I\Sigma} \equiv [\langle w_I^i, w_\Sigma^\alpha \rangle], \quad \underline{\underline{A}}_\delta^{\Sigma I} \equiv [\langle w_\Sigma^\alpha, w_I^i \rangle]$$



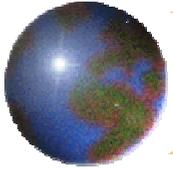


Método de Subestructuración ...

Podríamos definir (no se construyen estas matrices globales) las matrices $\underline{\underline{A}}^{II}$, $\underline{\underline{A}}^{\Sigma I}$, $\underline{\underline{A}}^{I\Sigma}$ y $\underline{\underline{A}}^{\Sigma\Sigma}$ donde

$$\underline{\underline{A}}^{II} \equiv \begin{bmatrix} \underline{\underline{A}}_1^{II} & & & \\ & \underline{\underline{A}}_2^{II} & & \\ & & \ddots & \\ & & & \underline{\underline{A}}_E^{II} \end{bmatrix} \quad \underline{\underline{A}}^{I\Sigma} \equiv \begin{bmatrix} \underline{\underline{A}}_1^{I\Sigma} \\ \underline{\underline{A}}_2^{I\Sigma} \\ \vdots \\ \underline{\underline{A}}_E^{I\Sigma} \end{bmatrix}$$

$$\underline{\underline{A}}^{\Sigma I} \equiv \begin{bmatrix} \underline{\underline{A}}_1^{\Sigma I} & \underline{\underline{A}}_2^{\Sigma I} & \dots & \underline{\underline{A}}_E^{\Sigma I} \end{bmatrix} \quad \underline{\underline{A}}^{\Sigma\Sigma} \equiv \begin{bmatrix} E \\ \sum_{i=1} \underline{\underline{A}}_i^{\Sigma\Sigma} \end{bmatrix}$$



Método de Subestructuración ...

Definiendo a $\underline{u} = (\underline{u}_I, \underline{u}_\Sigma)$ como $u_I = (u_1, \dots, u_{N_I})$ y $u_\Sigma = (u_1, \dots, u_{N_\Sigma})$. entonces podemos generar el sistema:

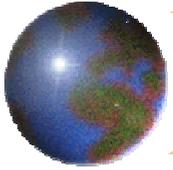
$$\begin{aligned}\underline{\underline{A}}^{II} \underline{u}_I + \underline{\underline{A}}^{I\Sigma} \underline{u}_\Sigma &= \underline{b}_I \\ \underline{\underline{A}}^{\Sigma I} \underline{u}_I + \underline{\underline{A}}^{\Sigma\Sigma} \underline{u}_\Sigma &= \underline{b}_\Sigma\end{aligned}$$

el cual es equivalente al sistema algebraico generado por el método de elemento finito.

Despejando \underline{u}_I del sistema anterior, tenemos el sistema lineal

$$\left(\underline{\underline{A}}^{\Sigma\Sigma} - \underline{\underline{A}}^{\Sigma I} (\underline{\underline{A}}^{II})^{-1} \underline{\underline{A}}^{I\Sigma} \right) \underline{u}_\Sigma = \underline{b}_\Sigma - \underline{\underline{A}}^{\Sigma I} (\underline{\underline{A}}^{II})^{-1} \underline{b}_I$$

a $\underline{\underline{S}} = \underline{\underline{A}}^{\Sigma\Sigma} - \underline{\underline{A}}^{\Sigma I} (\underline{\underline{A}}^{II})^{-1} \underline{\underline{A}}^{I\Sigma}$ comúnmente se le llama el complemento de Schur.



Método de Subestructuración ...

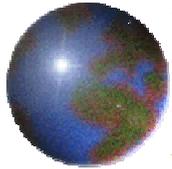
Pero como nosotros tenemos definidas de forma local en cada Ω_i a las matrices $\underline{\underline{A}}_i^{\Sigma\Sigma}$, $\underline{\underline{A}}_i^{\Sigma I}$, $\underline{\underline{A}}_i^{I\Sigma}$ y $\underline{\underline{A}}_i^{II}$, entonces el complemento

de Schur local queda definido por $\underline{\underline{S}}_i = \underline{\underline{A}}_i^{\Sigma\Sigma} - \underline{\underline{A}}_i^{\Sigma I} \left(\underline{\underline{A}}_i^{II} \right)^{-1} \underline{\underline{A}}_i^{I\Sigma}$.

Generándose el sistema virtual cuyas incógnitas son los nodos de la frontera interior

$$\left[\sum_{i=1}^E \underline{\underline{S}}_i \right] \underline{\underline{u}}_{\Sigma} = \left[\sum_{i=1}^E \underline{\underline{b}}_i \right]$$

donde $\underline{\underline{b}}_i = \underline{\underline{A}}_i^{\Sigma I} \left(\underline{\underline{A}}_i^{II} \right)^{-1} \underline{\underline{b}}_{I_i}$.



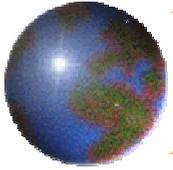
Método de Subestructuración ...

Usando el método de gradiente conjugado preconditionado, resolvemos los nodos de la frontera interior \underline{u}_Σ .

Para ello, es necesario hacer por cada iteración del método una multiplicación de un vector de la dimensión de \underline{u}_Σ con todos los \underline{S}_i , esto se logra pasando sólo las entradas correspondientes al subdominio y así poderlo multiplicar por \underline{S}_i .

El número de condicionamiento de la matriz del método de subestructuración es del orden de $\frac{1}{h}$.

Una vez conocido \underline{u}_Σ se calcula $\underline{u}_{I_i} = \left(\underline{A}_i^{II}\right)^{-1} \left(\underline{b}_{I_i} - \underline{A}_i^{\Sigma I} \underline{u}_{\Sigma_i}\right)$ resolviendo con esto los nodos interiores.



Cómputo en Paralelo ...

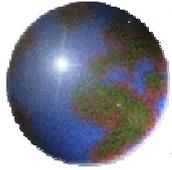
Métricas de desempeño

● Factor de aceleración $s = \frac{T(1)}{T(p)}$

En el caso ideal debería crecer de forma lineal al aumento del número de procesadores.

● Eficiencia $e = \frac{T(1)}{pT(p)} = \frac{s}{p}$

En el caso ideal debería de ser cercana a la unidad cuando el hardware es usado eficientemente.



Cómputo en Paralelo ...

✦ Fracción serial $f = \frac{\frac{1}{s} - \frac{1}{p}}{1 - \frac{1}{p}}$

En el caso ideal debería decrecer a cero, un incremento en su valor es aviso de granularidad fina con la correspondiente sobrecarga en los procesos de comunicación.