

Capítulo 2

Conocimientos previos necesarios

Para la realización del proyecto ha sido necesario el aprendizaje de varios recursos informáticos. En primer lugar, Linux, y su entorno de trabajo, utilizando la distribución Debian y KDE como entorno gráfico. Se programó en Octave, como alternativa al programa Matlab de Windows para la estructura principal del programa, para desde ahí, hacer una llamada al programa de Fortran que resuelve el problema elástico. Para la construcción de dicho programa se utilizó como base un algoritmo que utiliza el método de elementos de contorno para resolver el problema.

Este programa, llamado SERBA, estaba escrito en Fortran-77 por lo que también fue necesario conocer la programación en Fortran y lo más importante, el uso de sus librerías más avanzadas relacionadas con la paralelización para ser utilizadas en el clúster beowulf del que disponemos, la más importante, la librería de paso de mensajes MPI.

Se presenta a continuación una breve reseña sobre algunos de los temas anteriormente mencionados.

2.1.- Método de Elementos de Contorno

2.1.1.- La necesidad de usar métodos numéricos

El número de problemas para los cuales se puede encontrar una solución analítica es muy limitado. Apenas unos pocos ejemplos pueden ser resueltos, como el de la viga simple, un tubo a presión constante o una placa. Un ejemplo sería el de la viga flexionada representado en la figura 1:

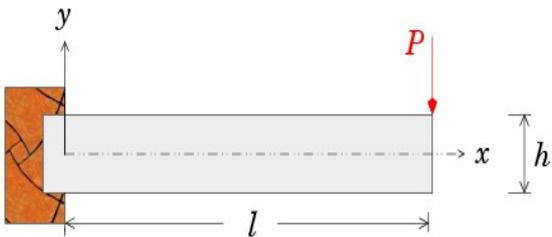


Figura 2.1

$$u_x = -Ax^2y + A\frac{2+\nu}{3}y^3 - 2A(1+\nu)h^2y + Al^2y$$

$$u_y = Avxy^2 + \frac{1}{3}Ax^3 - Al^2x - \frac{2}{3}Al^3$$

$$A = \frac{P}{2EI_z}$$

En la mayoría de los casos, alguna característica del problema presenta un alto grado de dificultad: la geometría de la pieza, la forma en que se aplican las cargas, etc. Existen algunas alternativas para solventar este tipo de problemas:

- Hacer suficientes suposiciones y/o simplificaciones de tal forma que se pueda encontrar una solución analítica pero se pierde precisión.
- Utilizar técnicas experimentales aunque no siempre es posible.
- Utilizar algún método numérico, como FEM (método de elementos finitos), BEM(método de elementos de contorno) o FDM(método de diferencias finitas).

2.1.2.- Elección de BEM como método numérico

En nuestro caso elegimos el BEM, porque entre otras ventajas sólo precisa de la discretización del contorno para resolver el problema. Esta característica lo hace especialmente apto para resolver problemas de optimización de forma como las que se plantea en este trabajo debido fundamentalmente al ahorro de tiempo de cálculo.

2.1.3.- Reseña histórica de BEM

Kellogg (1953) fue uno de los pioneros en la aplicación de las técnicas de integración para la solución de problemas tipo Laplace no obstante, su aplicación no fue muy popular y fueron utilizadas principalmente en la resolución de problemas de la mecánica del fluido y en problemas generales de potencial, por lo que tales técnicas eran conocidas como métodos “fuente” los cuales son métodos “indirectos” de análisis. En este campo destacan los aportes hechos por Jaswon (1963), Symm (1963), Massonet (1966) y Hess (1967) Se podría considerar que Cruse y Rizzo (1968) desarrollaron la primera versión directa del método, aplicada a problemas bidimensionales y basada en la solución fundamental de Lord Kelvin para un medio elástico, isótropo, homogéneo e infinito.

En los años setenta, otros investigadores contribuyeron al desarrollo del método de los elementos de contorno como Rizzo (1977), Shippy (1979) y Nakaguma (1979), quienes consideraron la influencia del peso propio, fuerzas centrífugas, etc. dentro de la formulación integral del método directo. Brebbia (1978) trabajó con la relación existente entre los diferentes métodos de aproximación, para relacionarlos con las ecuaciones integrales en el contorno y fue él, junto a Dominguez, quien utilizó por primera vez el nombre de “Elementos de Contorno” en 1977.

Actualmente existen diversos paquetes comerciales que utilizan el Método de Elementos de Contorno para la resolución de todo tipo de problemas de elasticidad, transferencia de calor, mecánica de fluidos, mecánica de la fractura, geomecánica, etc.

2.1.4.- Formulación del método para problemas de potencial

Aunque en nuestro caso analizaremos problemas de elasticidad, por simplicidad el proyecto comenzó con el estudio del método para problemas de potencial, cuya resolución se expone a continuación.

Considerando la ecuación de Laplace como la que gobierna los problemas de potencial (por ejemplo el problema de transferencia de calor) en un dominio V en 2D (Figura 2.2)

$$\nabla^2 \phi(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in V; \quad (1)$$

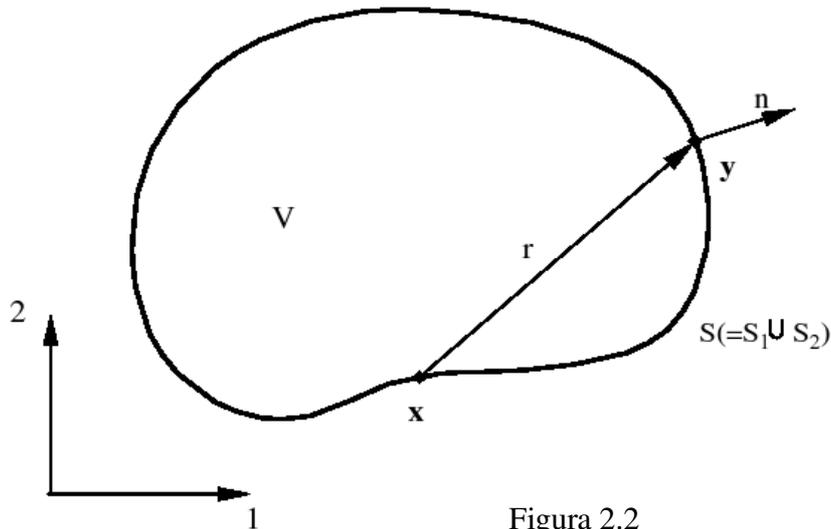


Figura 2.2

sometido a las siguientes condiciones de contorno:

$$\phi(\mathbf{x}) = \bar{\phi}(\mathbf{x}), \quad \forall \mathbf{x} \in S_1; \quad (2)$$

$$q(\mathbf{x}) \equiv \frac{\partial \phi}{\partial n}(\mathbf{x}) = \bar{q}(\mathbf{x}), \quad \forall \mathbf{x} \in S_2; \quad (3)$$

donde $\phi(\mathbf{x})$ es el campo potencial en el dominio V , $S = S_1 \cup S_2$, \mathbf{n} es la normal exterior, y los valores con barra son los dados por la frontera.

La solución del problema de valor de contorno descrito en las ecuaciones (1), (2) y (3) puede representarse por la siguiente integral para un punto \mathbf{x} ubicado dentro del dominio V :

$$\phi(\mathbf{x}) = \int_S [G(\mathbf{x}, \mathbf{y})q(\mathbf{y}) - F(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})]dS(\mathbf{y}), \quad \forall \mathbf{x} \in V; \quad (4)$$

donde $\mathbf{G}(\mathbf{x}, \mathbf{y})$ es la función de Green que para problemas 2D esta dada por

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi} \ln\left(\frac{1}{r}\right), \quad (5)$$

y que provee la solución de potencial $\phi(\mathbf{x})$ en el punto \mathbf{y} (punto del campo) debida a una fuente puntual unitaria en \mathbf{x} (punto de colocación). r está dada por la distancia entre el punto de colocación \mathbf{x} y el punto de campo \mathbf{y} (figura 2.2).

La derivada normal de $\mathbf{G}(\mathbf{x}, \mathbf{y})$ resulta:

$$F(\mathbf{x}, \mathbf{y}) = \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial n(\mathbf{y})} = \frac{1}{2\pi r} \frac{\partial r}{\partial n} \quad (6)$$

Haciendo $\mathbf{x} \rightarrow \mathbf{S}$ (superficie del dominio) obtenemos la siguiente solución integral en el contorno

$$C(\mathbf{x})\phi(\mathbf{x}) = \int_S [G(\mathbf{x}, \mathbf{y})q(\mathbf{y}) - F(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})] dS(\mathbf{y}), \quad \forall \mathbf{x} \in S; \quad (7)$$

en la cual el segundo término a la derecha es una integral del tipo los valores principales de Cauchy (CPV) y el coeficiente

$$C(\mathbf{x}) = - \int_S F(\mathbf{x}, \mathbf{y}) dS(\mathbf{y}), \quad (8)$$

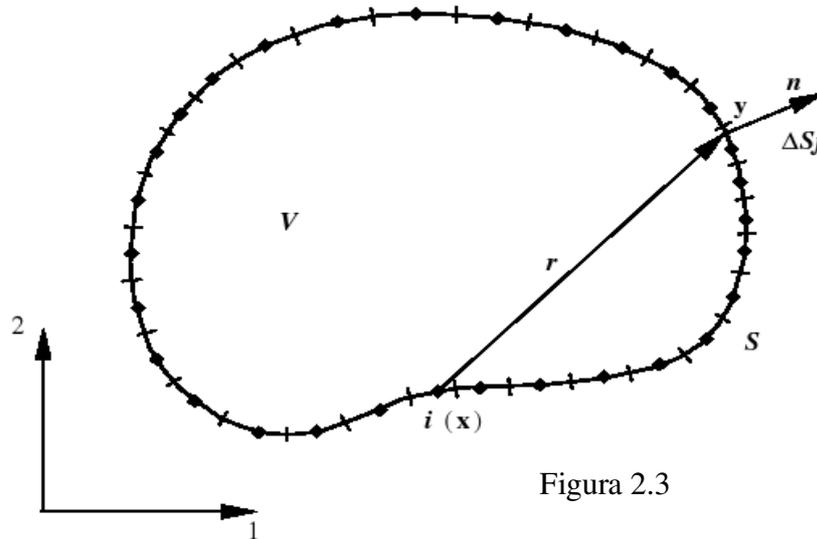
el cual es también una integral CPV. Si el contorno S es suave en el punto de colocación de \mathbf{x} , tenemos simplemente $C(\mathbf{x}) = 1/2$. Sustituyendo (8) en (7) podemos obtenerla siguiente forma de BIE para problemas de potencial:

$$\int_S F(\mathbf{x}, \mathbf{y}) [\phi(\mathbf{y}) - \phi(\mathbf{x})] dS(\mathbf{y}) = \int_S G(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) dS(\mathbf{y}), \quad \forall \mathbf{x} \in S; \quad (9)$$

Se puede utilizar cualquiera de las ecuaciones para BIE, tanto la (7) como la (9) dependiendo de lo que sea más conveniente según el esquema de discretización

utilizado. Como ejemplo de esquema de discretización, usaremos elementos de contorno constantes, es decir, dividiremos la frontera S en N segmentos (elementos) y poniendo un nodo en cada elemento (ver figura 2.3). Con esto obtenemos la siguiente ecuación BIE (7) discretizada para cada nodo i

$$\frac{1}{2}\phi_i = \sum_{j=1}^N [g_{ij}q_j - f_{ij}\phi_j], \quad \text{for } i = 1, 2, \dots, N \quad (10)$$



donde ϕ_j y q_j ($j=1,2,\dots,N$) son los valores nodales de ϕ y q del elemento ΔS_j , respectivamente y los coeficientes vienen dados por

$$g_{ij} = \int_{\Delta S_j} G(\mathbf{x}, \mathbf{y}) dS(\mathbf{y}), \quad f_{ij} = \int_{\Delta S_j} F(\mathbf{x}, \mathbf{y}) dS(\mathbf{y}), \quad (11)$$

con $i, j = 1, 2, \dots, N$ y con el punto de colocación \mathbf{x} puesto en el nodo i . En forma matricial, la ecuación 10 se escribe como

$$\frac{1}{2} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{Bmatrix} + \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & f_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N1} & f_{N2} & \cdots & f_{NN} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{Bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N1} & g_{N2} & \cdots & g_{NN} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{Bmatrix}. \quad (12)$$

En el planteamiento convencional de BEM, el sistema estándar de ecuaciones lineales esta formado como sigue aplicando las condiciones de contorno (ecuaciones (2) y (3) a cada nodo y conmutando las columnas de las dos matrices de (12).

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{Bmatrix}, \quad \text{o} \quad \mathbf{A}\boldsymbol{\lambda} = \mathbf{b}$$

(13)

donde \mathbf{A} es la matriz de coeficientes, $\boldsymbol{\lambda}$ el vector incógnita y \mathbf{b} el vector conocido. De la solución de (13) se obtiene la solución del problema, es decir, los valores de $\boldsymbol{\varphi}$ desconocidos en \mathbf{S}_2 y los de \mathbf{q} en \mathbf{S}_1 .

2.1.5.- *Formulación del método para problemas elásticos*

La formulación para el caso de problemas elásticos planos es análoga a la de potencial pero con diferentes condiciones de contorno y soluciones fundamentales. la diferencia fundamental entre los dos casos será el hecho de que por cada nodo habrá un vector solución, con los desplazamientos en x e y, o la fuerza aplicada en dicho punto. En esta sección se comenta de forma resumida dicha formulación.

La ecuación que gobierna los problemas de elasticidad bien podría representarse por la de Navier, que en su forma estándar tiene la siguiente estructura:

$$\mathbf{G}u_{i,jj}(\mathbf{x}) + (\lambda + \mathbf{G})u_{j,ij}(\mathbf{x}) + X_i(\mathbf{x}) = 0, \quad \mathbf{x} \in \mathbf{S} \quad (14)$$

Las condiciones de contorno pueden ser en desplazamientos o tensiones:

$$\mathbf{u}_i(\mathbf{y}) = \bar{\mathbf{u}}_i(\mathbf{y}), \quad \mathbf{y} \in \mathbf{S}_u \quad (15)$$

$$\mathbf{t}_i(\mathbf{y}) = \bar{\mathbf{t}}_i(\mathbf{y}), \quad \mathbf{y} \in \mathbf{S}_t \quad (16)$$

donde \mathbf{u} es un vector $\mathbf{u}(\mathbf{x},\mathbf{y})$ que representa los desplazamientos en los dos ejes y \mathbf{t} las tensiones en el contorno.

La solución del problema de valor de contorno descrito en las ecuaciones (14), (15) y (16) puede representarse por la siguiente integral para un punto \mathbf{x} ubicado dentro del dominio \mathbf{V} :

$$\mathbf{C}_{ji}(\mathbf{x})\mathbf{u}_i(\mathbf{x})\int\mathbf{T}_{ji}(\mathbf{x},\mathbf{y})\mathbf{u}_i(\mathbf{y})\mathbf{d}s(\mathbf{y}) = \int\mathbf{U}_{ji}(\mathbf{x},\mathbf{y})\mathbf{t}_i(\mathbf{y})\mathbf{d}s(\mathbf{y}), \quad \mathbf{x},\mathbf{y}\in\mathbf{V}$$

Operando de manera análoga a la empleada para el problema de potencial llegamos al mismo sistema de ecuaciones de (13) pero con la salvedad de que el tamaño de la matriz es el doble, ya que los desplazamientos y tensiones tienen dos coordenadas, una por cada eje. El vector incógnita $\boldsymbol{\lambda}$ estará formado por los desplazamientos y tensiones desconocidas según el caso. Puesto que ambas magnitudes son vectoriales de tamaño dos, dicho vector tendrá en este caso una dimensión de $2N$, y la matriz será de $2N \times 2N$.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{Bmatrix}, \quad \text{o} \quad \mathbf{A}\boldsymbol{\lambda} = \mathbf{b}$$

2.2.- Cluster Beowulf

El término de **cluster** se aplica a los conjuntos o conglomerados de computadoras construidos utilizando componentes de hardware comunes y, la mayor parte de las veces, software libre. Ellos juegan hoy en día un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

La tecnología de clusters ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores Web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con clusters surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, cluster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio. De un cluster se espera que presente combinaciones de los siguientes servicios:

1. Alto rendimiento (High Performance)
2. Alta disponibilidad (High Availability)
3. Equilibrio de carga (Load Balancing)
4. Escalabilidad (Scalability)

Los ordenadores del cluster pueden tener, todos, la misma configuración de hardware y sistema operativo (cluster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (cluster semi-homogéneo), o tener diferente hardware y sistema operativo (cluster heterogéneo), lo que hace más fácil y económica su construcción.

Para que un cluster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del cluster, el cual se encargue de interactuar con el usuario y los procesos que

corren en él para optimizar el funcionamiento.

Un **Beowulf** es una clase de computador masivamente paralelo de altas prestaciones principalmente construido a base de un cluster de componentes hardware estándar. Un Beowulf ejecuta un sistema operativo de libre distribución como Linux o FreeBSD, y se interconecta mediante una red privada de gran velocidad. Generalmente se compone de un grupo de PCs o estaciones de trabajo dedicados a ejecutar tareas que precisan una alta capacidad de cálculo. Los nodos en el cluster de computadoras no se hayan en los puestos de trabajo de los usuarios, sino que están totalmente dedicados a las tareas asignadas al cluster. Generalmente, el cluster se haya conectado al mundo exterior por un solo nodo.

El software puede ejecutarse más rápido en un Beowulf si se dedica algún tiempo a reestructurar los programas (paralelizarlos). En general es necesario partirlos en tareas paralelas que se comunican usando alguna librería como MPI o PVM, o sockets o SysV IPC.



Figura 2.4: Cluster tipo Beowulf

2.3.- Librerías de trabajo en Fortran MPI

La Interfaz de Paso de Mensajes (conocido ampliamente como MPI, siglas en inglés de *Message Passing Interface*) es un protocolo de comunicación entre computadoras. Es el estándar para la comunicación entre los nodos que ejecutan un programa en un sistema de memoria distribuida. Las implementaciones en MPI consisten en un conjunto de librerías de rutinas que pueden ser utilizadas en programas escritos en los lenguajes de programación C, C++, Fortran y Ada. La ventaja de MPI sobre otras librerías de paso de mensajes, es que los programas que utilizan la librería son portables (dado que MPI ha sido implementado para casi toda arquitectura de memoria distribuida), y rápidos, (porque cada implementación de la librería ha sido optimizada para el hardware en la cual se ejecuta).

En un ambiente de comunicación con memoria distribuida en la cual las rutinas de paso de mensajes de nivel bajo, los beneficios de la estandarización son muy notorios. La principal ventaja al establecer un estándar para el paso de mensajes es la portabilidad y el ser fácil de utilizar.

MPI es un sistema complejo, el cual comprende 129 funciones, de las cuales la mayoría tiene muchos parámetros y variantes.

Metas del MPI

- Diseñar una interfaz de programación aplicable (no necesariamente para compiladores o sistemas que implementan una librería).
- Permitir una comunicación eficiente. Evitando el copiar de memoria a memoria y permitiendo (donde sea posible) la sobreposición de computación y comunicación, además de aligerar la comunicación con el procesador.
- Permitir implementaciones que puedan ser utilizadas en un ambiente heterogéneo.
- Permitir enlaces convenientes en C y Fortran 77 para el interfaz.
- Asumir un interfaz de comunicación seguro. El usuario no debe lidiar con fallas de comunicación. Tales fallas son controladas por el subsistema de comunicación interior.
- Definir un interfaz que no sea muy diferente a los actuales, tales como

PVM, NX, Express, p4, etc., y proveer de extensiones para permitir mayor flexibilidad.

- Definir un interfaz que pueda ser implementado en diferentes plataformas, sin cambios significativos en el software y las funciones internas de comunicación.
- La semántica del interfaz debe ser independiente del lenguaje.
- La interfaz debe ser diseñada para producir tareas seguras.

La librería para el lenguaje de programación Fortran se llama “mpif.h”.

7.2 Esquema de trabajo del programa *SERBA* modificado

