The page features a decorative design with three blue circles of varying sizes, each composed of concentric rings of different shades of blue. These circles are arranged in a descending staircase pattern from the top right towards the bottom right. Two thin, light blue lines intersect at the top left and extend diagonally across the page, one towards the top-right circle and the other towards the bottom-right circle.

**DESARROLLO DE UN SOFTWARE EN
ELEMENTOS FINITOS COMPATIBLE CON
GID PARA RESOLUCIÓN DE
PROBLEMAS ELÁSTICO LINEALES.**

Verónica Gallego Otalvaro
Coordinador Santiago Correa
Grupo de Bioingeniería
EAFIT
10/12/2009

Contenido

INTRODUCCIÓN	3
MARCO TEÓRICO	4
1. Elemento triangular lineal bidimensional	6
2. Elemento triangular cuadrático bidimensional	12
3. Elemento rectangular lineal bidimensional	19
4. Elemento rectangular cuadrático	24
5. Elementos tridimensionales	30
6. Tetraedro lineal.....	31
7. Hexaedro lineal.....	36
8. Integración numérica.....	40
9. Ensamble de la matriz de rigidez (K).....	42
METODOLOGÍA.....	50
TRABAJO FUTURO.....	50
CONCLUSIONES	51
REFERENCIAS	52

Introducción

El método de los elementos finitos es un método numérico utilizado frecuentemente para obtener la solución aproximada de un sistema de ecuaciones diferenciales parciales, permitiendo encontrar soluciones apropiadas para diversos casos de la ingeniería, uno de los tópicos en el que se utiliza más comúnmente es en los problemas físicos que poseen geometrías muy complicadas lo que hace necesario el uso de un computador para llevar a cabo la implementación del método.

El método es altamente utilizado ya que permite obtener una solución numérica aproximada sobre el comportamiento de una estructura bien sea un sólido, un gas o un fluido, sobre el que están definidas ciertas ecuaciones diferenciales en forma débil o integral que se encargan de definir el comportamiento físico del problema dividiéndolo en un número elevado de subdominios no intersectantes entre sí, esto se denomina elementos finitos.

Dentro de cada elemento se encuentran una serie de puntos que permiten el movimiento de cada elemento, estos puntos son denominados nodos y son adyacentes entre ellos si pertenecen al mismo elemento, también un nodo sobre la frontera puede pertenecer a varios elementos, el conjunto de nodos y las relaciones que existen entre ellos constituyen una malla, esta malla es de gran importancia ya que a partir de esta se realiza la discretización del dominio en elementos finitos, y a su vez permite realizar los cálculos sobre cada uno de los nodos. La construcción de la malla generalmente se realiza a partir de programas especializados, pero en este libro se mostrará una forma de realizar la implementación en un lenguaje de programación como lo es Matlab, para esto se hace necesario entender las relaciones de adyacencia o conectividad que poseen las variables que no conocemos en cada uno de los nodos que además nos permiten saber los grados de libertad que posee la pieza a estudiar. El conjunto de relaciones entre el valor de una determinada variable y los nodos se puede expresar como un sistema de ecuaciones donde el número de ecuaciones es proporcional al número de nodos, este sistema también puede ser escrito como una matriz llamada matriz de rigidez global.

Generalmente el método de elementos finitos se programa computacionalmente para calcular los desplazamientos, las deformaciones y tensiones cuando se trata de un problema de sólidos deformables o más generalmente un problema de mecánica de medios continuos, además el método es fácilmente adaptable a problemas de transmisión de calor, de mecánica de fluidos para calcular campos de velocidades y presiones o de campo electromagnético. Dada la dificultad de encontrar la solución a estos problemas de manera analítica, con frecuencia en la práctica los métodos numéricos y, en particular, los elementos finitos, se convierten en una herramienta alternativa para encontrar estas soluciones, ya que la convergencia de esta metodología puede decirse es buena.

Marco teórico

Para resolver un problema definido mediante ecuaciones diferenciales por medio de un algoritmo de elementos hay que tener en cuenta las siguientes etapas.

1. El problema debe reformularse en forma variacional.
2. El dominio de variables independientes debe dividirse mediante una partición en subdominios. Asociada a la partición anterior se construye un espacio vectorial de dimensión finita, llamado espacio de elementos finitos. Siendo la solución numérica aproximada obtenida por elementos finitos una combinación lineal en dicho espacio vectorial.
3. Se obtiene la proyección del problema variacional original sobre el espacio de elementos finitos obtenido de la partición. Esto da lugar a un sistema con un número de ecuaciones finito, aunque en general con un número elevado de ecuaciones incógnitas. El número de incógnitas será igual a la dimensión del espacio vectorial de elementos finitos obtenido y, en general, cuanto mayor sea dicha dimensión mejor será la aproximación numérica.
4. Cálculo numérico de la solución del sistema de ecuaciones.

Los pasos anteriores permiten construir un problema de cálculo diferencial en un problema de álgebra lineal. Dicho problema en general se plantea sobre un espacio vectorial de dimensión no-finita, pero que puede resolverse aproximadamente encontrando una proyección sobre un subespacio de dimensión finita, y por tanto con un número finito de ecuaciones. La discretización en elementos finitos ayuda a construir un algoritmo de proyección sencillo, logrando además que la solución por el método de elementos finitos sea generalmente exacta en un conjunto finito de puntos.

Formulación débil: La formulación débil de una ecuación diferencial permite convertir un problema de cálculo diferencial formulado en términos de ecuaciones diferenciales en términos de un problema de álgebra lineal planteado sobre un espacio de Banach, generalmente de dimensión no finita, pero que puede ser aproximado por un sistema finito de ecuaciones algebraicas.

Dada una ecuación diferencial lineal de la forma:

$$\xi(u) = f$$

Donde la solución es una cierta función definida sobre un dominio $\Omega \subset \mathbb{R}^n$, y se han especificado un conjunto de condiciones de contorno adecuadas, puede suponerse que la función buscada es un elemento de un espacio de funciones o espacio de Banach V .

$$A(u) = f, \quad \begin{cases} u \in V & f \in V \\ A: V \rightarrow V' & A \in \xi(V, V') \end{cases}$$

Donde V' es el espacio dual de V , la forma variacional débil se obtiene buscando la única solución $u \in V$ tal que:

$$a(u, v) = \langle f, v \rangle, \quad \forall v \in V \quad \text{donde} \quad \begin{cases} a(u, v) = \langle Au, v \rangle \\ \langle f, v \rangle = \int_{\Omega} f v dx \end{cases}$$

Cuando el operador lineal es un operador elíptico, el problema se puede plantear como un problema de minimización sobre el espacio de Banach.

Discretización del dominio: Dado un dominio $\Omega \subset \mathbb{R}^n$ con una frontera continua en el sentido de Lipschitz una partición en n "elementos finitos", es una colección de n subdominios $\{\Omega^{(e)}\}_{e=1}^n$ que satisface:

1. $\Omega \subset \mathbb{R}^n$
2. Cada $\Omega^{(e)}$ es un conjunto compacto con una frontera Lipschitz-continua.
3. $\text{int}(\Omega^{(i)}) \cap \text{int}(\Omega^{(j)}) = 0, \quad i \neq j$

Usualmente por conveniencia práctica y sencillez de análisis, todos los "elementos finitos" tienen la misma "forma", es decir, existe un dominio de referencia $\hat{\Omega} \subset \mathbb{R}^n$ y una colección de funciones biyectivas de la siguiente forma:

$$\left\{ F^{(e)} \Big|_{F^{(e)}} : \hat{\Omega} \rightarrow \Omega^{(e)} \right\}$$

En los análisis 2D el dominio de referencia $\hat{\Omega}$ se suele tomar como un triángulo equilátero o un cuadrado, mientras que en los análisis 3D, el dominio de referencia típicamente es un tetraedro o un hexaedro. Además sobre cada elemento se considerarán algunos puntos especiales, llamados nodos y que generalmente incluirán los vértices del elemento finito y se requerirá la condición adicional de que dos elementos adyacentes tengan los compartan los nodos sobre el subconjunto $\Omega^{(i)} \cap \Omega^{(j)}$, es decir:

$$x \in \Omega^{(i)} \cap \Omega^{(j)} \wedge (x \in \text{nod}(\Omega^{(i)})) \rightarrow (x \in \text{nod}(\Omega^{(j)}))$$

Una vez definida la partición en elementos finitos, se define sobre cada elemento un espacio funcional de dimensión finita, usualmente formado por polinomios de grado. Este espacio funcional servirá para aproximar localmente la solución del problema variacional. El problema variacional en su forma débil se plantea sobre un espacio de dimensión no-finita, y por tanto la función buscada será una función de dicho espacio. El problema en esa forma exacta es computacionalmente inabordable, así que en la práctica se considerará un subespacio de dimensión finita V^h del espacio vectorial original V . Y en lugar de la solución exacta presentada anteriormente se automatizará la proyección de la solución original sobre dicho el subespacio vectorial de dimensión finita, es decir, se resolverá numéricamente el siguiente problema:

$$a(u^h, v^h) = \langle f, v^h \rangle, \quad \forall v^h \in V^h$$

Donde

$u^h = \Pi_e(u) \in V^h$, es la solución aproximada.

$\Pi_e: V \rightarrow V^h \quad V^h \subset V$ Es el proyector ortogonal del espacio original sobre el subespacio vectorial asociado a la discretización.

Si la discretización es suficientemente fina y el espacio funcional finito sobre cada elemento está bien escogido, la solución numérica obtenida aproximará razonablemente bien la solución original. Eso implicará en general considerar un número muy elevado de elementos finitos y por tanto un subespacio de proyección de dimensión elevada. El error entre la solución exacta y la solución aproximada puede acotarse gracias al lema de Céa, que en esencia afirma que la solución exacta y la solución aproximada satisfacen:

$$\|u - u^h\|_V \leq \inf_{v^h \in V^h} \|u - v^h\|_V$$

Es decir, el error dependerá ante todo de lo bien que el subespacio vectorial asociado a la discretización en elementos finitos V^h aproxime el espacio vectorial original V .

1. Elemento triangular lineal bidimensional

1.1 Funciones de forma

Los elementos finitos pueden ser clasificados por grupos de acuerdo al orden polinomial que posean, lineal, cuadrático, cúbico o de mayor orden.

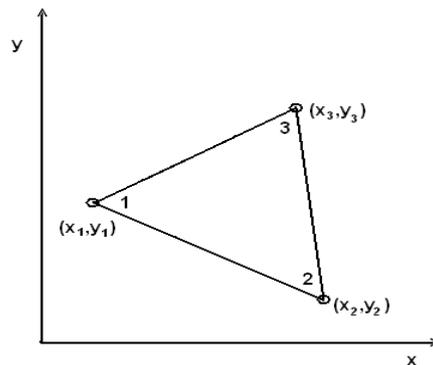


Fig.1: *elemento triangular lineal*

La función de forma de un elemento triangular lineal bidimensional puede ser escrita a partir de un polinomio que posee elementos lineales y un término constante como se muestra en la ecuación (1.1)

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{1.1}$$

El polinomio es lineal en x , en y , y además contiene unos coeficientes que hacen referencia a cada uno de los nodos.

El polinomio de interpolación del elemento triangular lineal se define por la relación (1.1), donde ϕ se utiliza para representar cualquier variable desconocida. En cada nodo.

$$\begin{aligned}\phi &= \phi_1 & \text{para } x = x_1 & \quad y = y_1 \\ \phi &= \phi_2 & \text{para } x = x_2 & \quad y = y_2 \\ \phi &= \phi_3 & \text{para } x = x_3 & \quad y = y_3\end{aligned}\tag{1.3}$$

Así,

$$\begin{aligned}\phi &= \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 \\ \phi &= \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 \\ \phi &= \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3\end{aligned}\tag{1.4}$$

Ahora para encontrar los valores de α_1 , α_2 y α_3 , tenemos:

$$\begin{aligned}\alpha_1 &= \frac{1}{2A} [(x_2 y_3 - x_3 y_2) \phi_1 + (x_3 y_1 - x_1 y_3) \phi_2 + (x_1 y_2 - x_2 y_1) \phi_3] \\ \alpha_2 &= \frac{1}{2A} [(y_2 - y_3) \phi_1 + (y_3 - y_1) \phi_2 + (y_1 - y_2) \phi_3] \\ \alpha_3 &= \frac{1}{2A} [(x_3 - x_2) \phi_1 + (x_1 - x_3) \phi_2 + (x_2 - x_1) \phi_3]\end{aligned}\tag{1.5}$$

Donde el área, A , está definida por:

$$2A = (x_1 y_2 - x_2 y_1) + (x_3 y_1 - x_1 y_3) + (x_2 y_3 - x_3 y_2)\tag{1.6}$$

El área del elemento está definida en términos de las coordenadas nodales. Obteniendo las funciones de forma, donde se realiza una aproximación de ϕ con una función que depende de las tres funciones nodales, como se presenta a continuación.

$$\phi = N_1 \phi_1 + N_2 \phi_2 + N_3 \phi_3\tag{1.7}$$

Las funciones de forma quedan definidas como

$$N_1(x, y) = \frac{1}{2A} [(x_2 y_3 - x_3 y_2) + (y_2 - y_3)x + (x_3 - x_2)y]\tag{1.8.1}$$

$$N_2(x, y) = \frac{1}{2A} [(x_3 y_1 - x_1 y_3) + (y_3 - y_1)x + (x_1 - x_3)y]\tag{1.8.2}$$

$$N_3(x, y) = \frac{1}{2A} [(x_1 y_2 - x_2 y_1) + (y_1 - y_2)x + (x_2 - x_1)y]\tag{1.8.3}$$

Estas ecuaciones se ajustan a la relación general expresada por la ecuación (1.1). Como ejemplo, el valor de N_1 en el nodo 1 se puede obtener mediante la sustitución de $x = x_1$, $y = y_1$ en la ecuación (1.8.1) así obtenemos:

$$N_1 = \frac{1}{2A} [x_2 y_3 - x_3 y_2 + y_2 x_1 - y_3 x_1 + x_3 y_1 - x_2 y_1] \quad (1.9)$$

N_1 es igual a cero en los nodos 2, 3, y todos los puntos de una línea que pasan a través de estos nodos. El valor entre paréntesis en la ecuación (1.9) también es igual a $2A$, por lo tanto tenemos

$$\begin{aligned} N_1 &= \frac{2A}{2A} = 1 && \text{con } x = x_1, \quad y = y_1 \\ N_1 &= 0 && \text{con } x = x_2, \quad y = y_2 \quad \text{y } x = x_3, \quad y = y_3 \end{aligned} \quad (1.10)$$

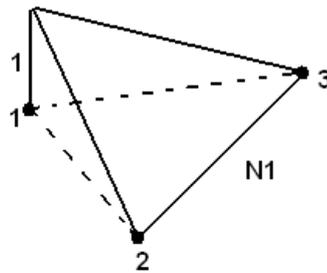


Fig.2: Representación grafica de la ecuación (1.10) cuando $N_1 = 1$

El gradiente de la variable ϕ está dado por la siguiente expresión

$$\begin{aligned} \frac{\partial \phi}{\partial x} &= \frac{\partial N_1}{\partial x} \phi_1 + \frac{\partial N_2}{\partial x} \phi_2 + \frac{\partial N_3}{\partial x} \phi_3 \\ \frac{\partial \phi}{\partial y} &= \frac{\partial N_1}{\partial y} \phi_1 + \frac{\partial N_2}{\partial y} \phi_2 + \frac{\partial N_3}{\partial y} \phi_3 \end{aligned} \quad (1.11)$$

El valor de $\partial N_1 / \partial x$, se puede determinar fácilmente de la ecuación (1.8.1)

$$\frac{\partial N_1}{\partial x} = \frac{y_2 + y_3}{2A} \quad (1.12)$$

El resto de las derivadas de las funciones de forma se obtienen de una manera sencilla y el valor de la derivada se convierte en

$$\frac{\partial \phi}{\partial x} = \frac{1}{2A} [(y_2 - y_3)\phi_1 + (y_3 - y_1)\phi_2 + (y_1 - y_2)\phi_3] \quad (1.13)$$

Sin embargo es algo complicado encontrar los α_i por medio de este sistema de ecuaciones por lo cual se hace necesario escribir las funciones de forma en términos de las coordenadas de área.

1.2 Coordenadas de área

A fin de simplificar el proceso de solución utilizando elementos triangulares, se construye un sistema de coordenadas bidimensionales que permite facilitar la evaluación de las integrales que se presentaran posteriormente

Se construyen las coordenadas de área definidas a partir de las variables, L_1 , L_2 y L_3

$$\begin{aligned} L_1 &= \frac{A_1}{A} \\ L_2 &= \frac{A_2}{A} \\ L_3 &= \frac{A_3}{A} \end{aligned} \tag{1.17}$$

Cuando A_1 , A_2 y A_3 son áreas parcialmente definidas por la unión de un punto P en el triángulo con los nodos en las esquinas (Fig.3), de manera tal que A_i es el área situada frente al nodo i .

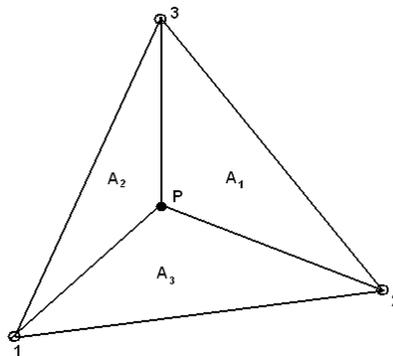


Fig.3: Sistema coordenado de área

Las líneas L_i serán paralelas al lado del triángulo opuesto al nodo i . Teniendo en cuenta que $L_1 + L_2 + L_3 = 1$, en cualquier punto dentro del triángulo. La Fig.4 muestra la dirección cada vez mayor para cada coordenada. Es evidente que cada L_i varía entre 0 y el largo del lado opuesto el nodo i .

Para un triángulo lineal, podemos definir las funciones de forma como:

$$\begin{aligned} N_1 &= L_1 \\ N_2 &= L_2 \\ N_3 &= L_3 \end{aligned} \quad (1.18)$$

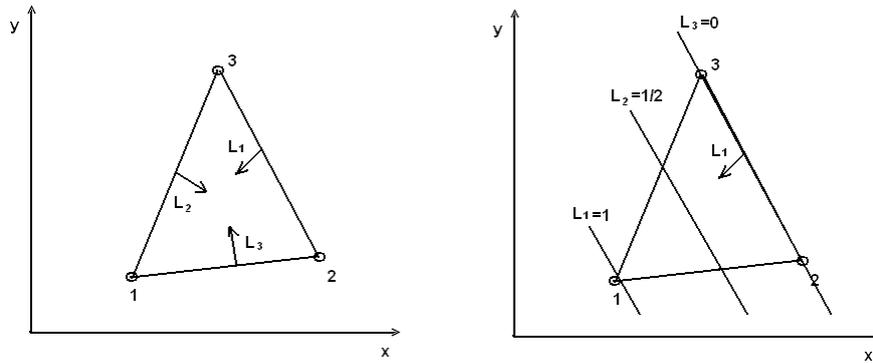


Fig.4: Coordenadas de área en un triángulo lineal

Por lo tanto, las coordenadas (x, y) están relacionadas con las coordenadas (L_1, L_2, L_3) a través de las siguientes expresiones

$$\begin{aligned} x &= L_1x_1 + L_2x_2 + L_3x_3 \\ y &= L_1y_1 + L_2y_2 + L_3y_3 \\ 1 &= L_1 + L_2 + L_3 \end{aligned} \quad (1.19)$$

Los cálculos de las derivadas de las funciones de forma requieren algunas modificaciones ya que las coordenadas no son independientes, es decir, $L_1 + L_2 + L_3 = 1$. Lo que se hace habitualmente es asumir L_1 y L_2 son independientes. Así, utilizando la regla de la cadena tenemos

$$\begin{aligned} \frac{\partial N_i}{\partial L_1} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial L_1} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial L_1} \\ \frac{\partial N_i}{\partial L_2} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial L_2} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial L_2} \end{aligned} \quad (1.21)$$

La ecuación (1.21) puede ser reescrita como

$$\begin{bmatrix} \frac{\partial N_i}{\partial L_1} \\ \frac{\partial N_i}{\partial L_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial L_1} & \frac{\partial y}{\partial L_1} \\ \frac{\partial x}{\partial L_2} & \frac{\partial y}{\partial L_2} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = J \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad (1.22)$$

La matriz de 2 x 2, se llama la matriz Jacobiana (**J**). Además se puede ver que L_3 es una función de ϕ

$$\frac{\partial \phi}{\partial L_1} = \frac{\partial \phi}{\partial L_1} \frac{\partial L_1}{\partial L_1} + \frac{\partial \phi}{\partial L_2} \frac{\partial L_2}{\partial L_1} + \frac{\partial \phi}{\partial L_3} \frac{\partial L_3}{\partial L_1} \quad (1.23)$$

Sin embargo, L_1 y L_2 se supone que son independientes, y $L_3 = 1 - L_1 - L_2$ por lo tanto, se tiene que:

$$\frac{L_3}{L_1} = -1 \quad (1.24)$$

En consecuencia, las derivadas de ϕ respecto a las coordenadas naturales son las siguientes:

$$\begin{aligned} \frac{\partial \phi}{\partial L_1} &\equiv \frac{\partial \phi}{\partial L_1} - \frac{\partial \phi}{\partial L_3} \\ \frac{\partial \phi}{\partial L_2} &\equiv \frac{\partial \phi}{\partial L_2} - \frac{\partial \phi}{\partial L_2} \end{aligned} \quad (1.25)$$

A continuación se presenta el código en Matlab par la construcción de la matriz B y el área A de un elemento triangular de 3 nodos.

```

%-----
% Calculo Del Área
function A = calcular_A(coordenate_node1,coordenate_node2,coordenate_node3)

    matrix=[ 1 coordenate_node1;...
            1 coordenate_node2;...
            1 coordenate_node3];

    A = 0.5*det(matrix);
end

%-----
% Calculo de B
function B = calcular_B(A,coordenate_node1,coordenate_node2,coordenate_node3)

b1 = coordenate_node2(2)-coordenate_node3(2);
b2 = coordenate_node3(2)-coordenate_node1(2);
b3 = coordenate_node1(2)-coordenate_node2(2);
c1 = coordenate_node3(1)-coordenate_node2(1);
c2 = coordenate_node1(1)-coordenate_node3(1);
c3 = coordenate_node2(1)-coordenate_node1(1);

B = (1/(2*A))* [b1 0 b2 0 b3 0 ;...
                0 c1 0 c2 0 c3;...
                c1 b1 c2 b2 c3 b3];
end
%-----

```

2. Elemento triangular cuadrático bidimensional

2.1 Funciones de forma

La función de forma de un elemento triangular cuadrático bidimensional puede ser escrita a partir de un polinomio que posee elementos lineales, de orden superior y un término constante. Sin importar que la forma sea igual a un elemento lineal bidimensional, la interpolación polinomial de un elemento cuadrático es:

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 \quad (2.1)$$

Los seis coeficientes indican que el elemento debe tener seis nodos.

Los elementos cúbicos o de orden superior también utilizan polinomios con términos lineales y de orden superior, sin embargo, los límites del elemento deben ser paralelos a los ejes de coordenadas.

Se puede realizar una aproximación de segundo grado sobre el elemento triangular. El elemento triangular bidimensional tiene seis nodos, como se muestra en la Fig. 5.

Los α se determinan de la misma manera que para el elemento lineal. Definiendo ϕ por cada nodo.

$$\begin{aligned} \phi_1 &= \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 + \alpha_4 x_1^2 + \alpha_5 x_1 y_1 + \alpha_6 y_1^2 \\ \phi_2 &= \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 + \alpha_4 x_2^2 + \alpha_5 x_2 y_2 + \alpha_6 y_2^2 \\ &\vdots \\ \phi_6 &= \alpha_1 + \alpha_2 x_6 + \alpha_3 y_6 + \alpha_4 x_6^2 + \alpha_5 x_6 y_6 + \alpha_6 y_6^2 \end{aligned} \quad (2.2)$$

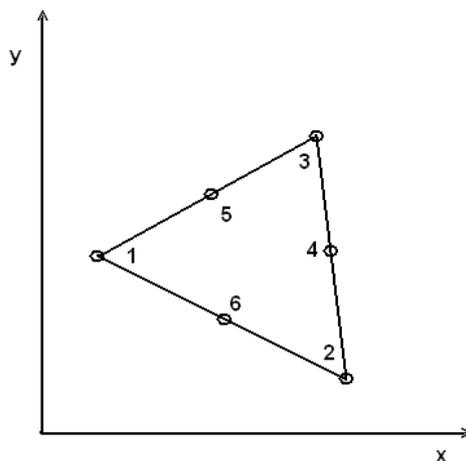


Fig.5: Elemento triangular cuadrático

Como se menciono anteriormente esta metodología es un poco complicada para encontrar los α por lo cual se escriben las ecuaciones de forma con respecto a las coordenadas de área de la misma manera que se realizo para los triángulos.

Para el triángulo de segundo orden que se muestra en la Fig.5, las funciones de forma, se escriben como funciones de L_1 , L_2 , y L_3 :

$$\begin{aligned}
 N_1 &= L_1(2L_1 - 1) \\
 N_2 &= L_2(2L_2 - 1) \\
 N_3 &= L_3(2L_3 - 1) \\
 N_4 &= 4L_2L_3 \\
 N_5 &= 4L_1L_3 \\
 N_6 &= 4L_1L_2
 \end{aligned} \tag{2.3}$$

Como el objetivo es encontrar la matriz de rigidez para este elemento es necesario construir la matriz B y encontrar el jacobiano a partir de las funciones de forma, además en este caso no es tan fácil resolver las integrales que se presentan para encontrar la matriz K (matriz de rigidez), como en el caso del triangulo lineal por lo que se hace necesario realizar una integración numérica.

Ejemplo 1: con el fin de ilustrar el uso de las coordenadas de área, se encontrara el valor de $\partial N_5 / \partial y$ y $\partial N_5 / \partial x$ en el punto (1,1) en el triángulo que se muestra en figura 6

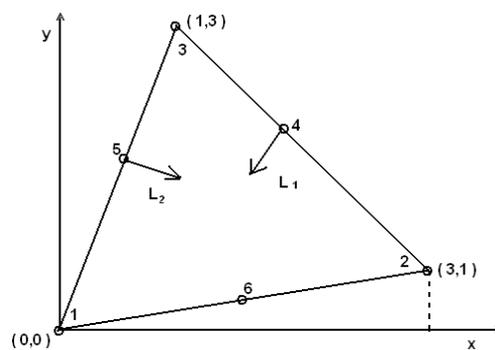


Fig.6: Ejemplo para calcular el valor de $\partial N_5 / \partial y$ y $\partial N_5 / \partial x$

Los valores de x y y están definidos por la siguiente expresión.

$$\begin{aligned}
 x &= L_1x_1 + L_2x_2 + L_3x_3 \\
 y &= L_1y_1 + L_2y_2 + L_3y_3
 \end{aligned} \tag{2.4}$$

Sustituyendo por los valores nodales obtenemos.

$$\begin{aligned}
 x &= 3L_2 + L_3 \\
 y &= L_2 + 3L_3
 \end{aligned} \tag{2.5}$$

La derivada de la ecuación (1.25) utilizando la matriz Jacobiana, se convierten en:

$$\begin{aligned}\frac{\partial x}{\partial L_1} &\equiv \frac{\partial x}{\partial L_1} - \frac{\partial x}{\partial L_3} = -1 \\ \frac{\partial x}{\partial L_2} &\equiv \frac{\partial x}{\partial L_2} - \frac{\partial x}{\partial L_2} = 2 \\ \frac{\partial y}{\partial L_1} &\equiv \frac{\partial y}{\partial L_1} - \frac{\partial y}{\partial L_3} = -3 \\ \frac{\partial y}{\partial L_2} &\equiv \frac{\partial y}{\partial L_2} - \frac{\partial y}{\partial L_2} = -2\end{aligned}\quad (2.6)$$

El Jacobiano de la ecuación (1.22) es

$$J = \begin{bmatrix} -1 & -3 \\ 2 & -2 \end{bmatrix} \quad (2.7)$$

Como queremos encontrar la $\partial N_5 / \partial y$ y $\partial N_5 / \partial x$ un términos del nodo 5, reordenamos la ecuación (1.22) en la siguiente forma

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial L_3} \\ \frac{\partial N_i}{\partial L_1} \end{bmatrix} \quad (2.8)$$

Ahora determinemos $\partial N_5 / \partial y$ y $\partial N_5 / \partial x$ para el nodo 5,

$$\begin{aligned}\frac{\partial N_5}{\partial L_1} &= \frac{\partial N_5}{\partial L_1} - \frac{\partial N_5}{\partial L_3} = 4L_3 - 4L_1 \\ \frac{\partial N_5}{\partial L_2} &= \frac{\partial N_5}{\partial L_2} - \frac{\partial N_5}{\partial L_2} = -4L_1\end{aligned}\quad (2.9)$$

Sustituyendo en la ecuación (2.9)

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} 4L_3 - 4L_1 \\ -4L_1 \end{bmatrix} \quad (2.10)$$

Donde la inversa del Jacobiano esta dada por

$$J^{-1} = \frac{1}{8} \begin{bmatrix} -2 & 3 \\ -2 & -1 \end{bmatrix} \quad (2.11)$$

Reemplazando el Jacobiano en la ecuación (2.11)

$$\begin{aligned}
\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} &= \frac{1}{8} \begin{bmatrix} -2 & 3 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} 4L_3 - 4L_1 \\ -4L_1 \end{bmatrix} \\
&= \frac{1}{8} \begin{bmatrix} -8L_3 - 4L_1 \\ -8L_3 + 12L_1 \end{bmatrix} \\
&= \begin{bmatrix} -L_3 - \frac{1}{2}L_1 \\ -L_3 + \frac{3}{2}L_1 \end{bmatrix}
\end{aligned} \tag{2.12}$$

Estas dos ecuaciones son aplicables en todo el elemento. Para encontrar los valores reales de las dos derivadas, determinamos L_1 , L_2 y L_3 en el punto (1,1) mediante la ecuación (4.32) y el hecho de que $L_1 + L_2 + L_3 = 1$

$$\begin{aligned}
1 &= 0L_1 + 3L_2 + L_3 \\
1 &= 0L_1 + L_2 + 3L_3 \\
1 &= L_1 + L_2 + L_3
\end{aligned} \tag{2.13}$$

Resolviendo el sistema de ecuaciones tenemos

$$\begin{aligned}
L_1 &= \frac{1}{2} \\
L_2 &= \frac{1}{4} \\
L_3 &= \frac{1}{4}
\end{aligned} \tag{2.14}$$

Los valores finales de las dos derivadas son

$$\begin{aligned}
\frac{\partial N_5}{\partial x} &= -L_3 - \frac{1}{2}L_1 = -\frac{1}{2} \\
\frac{\partial N_5}{\partial y} &= -L_3 + \frac{3}{2}L_1 = \frac{1}{2}
\end{aligned} \tag{2.15}$$

A continuación se presenta el código en Matlab par la construcción de la matriz B de un elemento triangular de 6 nodos.

```

%-----
%Calcular B
%Esta función entrega la matriz B y su determinante y lo que requiere como
%entradas son los puntos x, y y los puntos de integración que se explicaran
%mas adelante.
function [B detJ1] = calcular_B(x,y,L,p)

% Calculo de las derivadas de x e y respecto a los ejes coordenados
dxdl1 = x(1)-x(3);
dxdl2 = x(2)-x(3);

```

```

dyd11 = y(1)-x(3);
dyd12 = y(2)-x(3);

% Calculo de las derivadas de las funciones de forma respecto a las
% coordenadas parametrizadas
dN1d11 = 4*L(p,1)-1;
dN1d12 = 0;
dN2d11 = 0;
dN2d12 = 4*L(p,2)-1;
dN3d11 = 0;
dN3d12 = 0;
dN4d11 = 0;
dN4d12 = 4*L(p,3);
dN5d11 = 4*L(p,3);
dN5d12 = 0;
dN6d11 = 4*L(p,2);
dN6d12 = 4*L(p,1);

% Almacenamiento del jacobiano
J1 = [dxd11 dyd11; ...
      dxd12 dyd12];

detJ1 = det(J1);

% Almacenamiento de la matriz de derivadas respecto a coordenadas
% parametrizadas
derN = [dN1d11 dN2d11 dN3d11 dN4d11 dN5d11 dN6d11; ...
        dN1d12 dN2d12 dN3d12 dN4d12 dN5d12 dN6d12];

% Calculo de la matriz de derivadas de las funciones de forma respecto a
% los ejes coordenados
dN = (inv(J1)*derN)';

% Calculo de la matriz B
B = [dN(1,1) 0 dN(2,1) 0 dN(3,1) 0 dN(4,1) 0 dN(5,1) 0 dN(6,1) 0; ...
     0 dN(1,2) 0 dN(2,2) 0 dN(3,2) 0 dN(4,2) 0 dN(5,2) 0 dN(6,2); ...
     dN(1,2) dN(1,1) dN(2,2) dN(2,1) dN(3,2) dN(3,1) dN(4,2) dN(4,1) dN(5,2)
     dN(5,1) dN(6,2) dN(6,1)];
end
%-----

```

2.2 Integración numérica

La ventaja de utilizar coordenadas de área está en la capacidad de evaluar las integrales, como en el caso de una dimensión. Para el elemento bidimensional con dos coordenadas (L_1, L_2) se tiene

$$\int_0^L L_1^a L_2^b dx = \frac{a!b!}{(1+a+b+c)!} 2A \quad (2.16)$$

La integral de longitud L se define entre dos nodos a, b . esta relación también será válida en la definición de las integrales que son sólo una función de la longitud a lo largo de un borde de un elemento. Para los dos elementos tridimensionales con L_1, L_2 , y L_3 se tiene:

$$\int_0^L L_1^a L_2^b L_3^c dA = \frac{a!b!}{(2+a+b+c)!} 2A \quad (2.17)$$

Cuando A denota el área. Por ejemplo, en un triángulo lineal, el área de integración de $N_1 N_2$ esta dada por:

$$\int_0^L N_1 N_2 dA = \int_0^L L_1^1 L_2^1 L_3^0 dA \frac{1!!0!}{(2+1+1+0)!} 2A = \frac{A}{12} \quad (2.18)$$

Las integrales de área que se plantean son las siguientes

$$\int_0^1 \int_0^{1-L_1} f(L_1, L_2, L_3) |J| dL_1 dL_2 \quad (2.19)$$

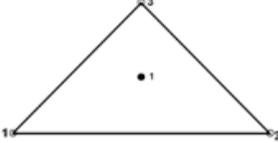
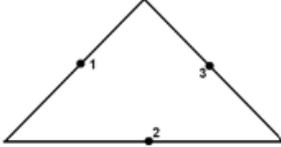
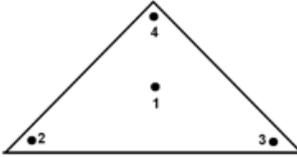
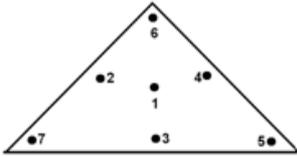
Cuando las funciones de orden superior de interpolación (cuadrática, cúbica, entre otras.) están escritas en función de las coordenadas de área, las matriz de las integrales debe ser evaluada numéricamente, esto se debe a la inversa de la matriz Jacobiana, que es una función racional propia de las coordenadas de área, y aparece en la ecuación (1.22), cuando la derivada de las funciones de forma están involucradas. Las formulas utilizadas en la ecuación (1.26) son válidas para las integrales simples. Sin embargo, para integrales complejas es mejor dejar que el computador haga la integración a fin de evitar los errores.

En el método de elementos finitos, se supone como:

$$\int_0^1 \int_0^{1-L_1} f(L_1, L_2, L_3) |J| dL_1 dL_2 = \frac{1}{2} \sum_{m=1}^M w_m g((L_1)_m, (L_2)_m, (L_3)_m) \quad (2.20)$$

Donde $g(\cdot)$ Incluye $|J|$, el factor $\frac{1}{2}$ corresponde al área en el sistema de coordenadas local, $(L_1)_m$ denota los puntos específicos en el triangulo, y w_m son los pesos asociados. El orden de la fórmula de cuadratura debe ser del menor número entero mayor que la suma de las influencias de las coordenadas L_1 , L_2 , y L_3 . Por ejemplo, para integrar el producto $L_1 L_2 L_3$, la suma de los exponentes es tres, y se utiliza un esquema de integración de cuarto grado.

Tabla1: Muestra los pesos y los valores de las coordenadas para las fórmulas de integración numérica de diversos órdenes.

Puntos de Gauss	m	L1	L2	L3	Peso (w)	Orden
	1	1/3	1/3	1/3	1	2
	1	1/2	0	1/2	1/3	3
	2	1/2	1/2	0	1/3	
	3	0	1/2	1/2	1/3	
	1	1/3	1/4	1/5	-27/48	4
	2	11/15	2/15	2/16	25/48	
	3	2/15	2/16	11/15	25/48	
	4	2/15	11/15	2/15	25/48	
	1	1/3	1/4	1/5	0.225	6
	2	β	α	β		
	3	β	β	α		
	4	α	β	β	0.13239415	
	5	λ	δ	λ		
	6	λ	λ	δ		
	7	δ	λ	λ	0.12593918	
	α	0.05971587				
	β	0.47014206				
	λ	0.10128651				
	δ	0.79742699				

A continuación se presenta el código en Matlab par la construcción de la integración numérica de un triangulo de 6 nodos.

```

%-----
%%Calcular k de del elemento
material_element = elements(element,end); % Numero de material
thickness_element = 1;
%Valores de L para la integración numerca
L =[1/2 0 1/2; 1/2 1/2 0; 0 1/2 1/2];
w = 1/3; %peso de Gauss
sum = zeros(12);
B_aux = zeros(3,12);
for p=1:3
    [B detJ1] = calcular_B(x,y,L,p);
    f = detJ1*B'*D{material_element}*B ;
    sum = sum + f;
    B_aux = B_aux + B;
end
k_element = 0.5*w*sum*thickness_element;
%-----

```

3. Elemento rectangular lineal bidimensional

3.1 Funciones de forma

En su forma más simple, el cuadrilátero se convierte en un elemento rectangular con un sistema de coordenadas (en el espacio físico). Utilizando un sistema de coordenadas locales, (naturales), tenemos un cuadrilátero generalizado.

La función de interpolación bilineal de un rectángulo de cuatro nodos esta dada como

$$\phi = \alpha_1 + \alpha_2x + \alpha_3y + \alpha_4xy \tag{3.1}$$

El término xy fue elegido en lugar de x^2 o y^2 dado que xy no cambia el sistema y supone una variación lineal de ϕ a lo largo de x o y . Para x constante, el elemento es lineal en y , y viceversa, por lo tanto el nombre del elemento es bilineal.

$$\begin{aligned} \frac{\partial \phi}{\partial x} &= \alpha_2 + \alpha_4y \\ \frac{\partial \phi}{\partial y} &= \alpha_3 + \alpha_4x \end{aligned} \tag{3.2}$$

Fig.7 muestra el elemento de configuración rectangular y nodal. Si las distancias al origen de coordenadas, que se encuentra en el punto medio del elemento, se denotan por a y b , los valores de nodo se dan como

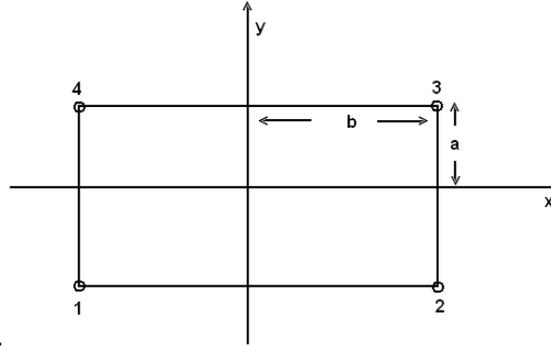


Fig.7: Rectángulo de cuatro nodos

$$\begin{aligned}
 \phi &= \phi_1 & \text{para } x = -b & \quad y = -a \\
 \phi &= \phi_2 & \text{para } x = b & \quad y = -a \\
 \phi &= \phi_3 & \text{para } x = b & \quad y = a \\
 \phi &= \phi_4 & \text{para } x = -b & \quad y = a
 \end{aligned} \tag{3.3}$$

Sustituyendo estos valores en la ecuación (3.1) tenemos

$$\begin{aligned}
 \alpha_1 &= \frac{\phi_1 + \phi_2 + \phi_3 + \phi_4}{4} \\
 \alpha_2 &= \frac{\phi_1 - \phi_2 - \phi_3 + \phi_4}{4a} \\
 \alpha_3 &= \frac{\phi_1 + \phi_2 - \phi_3 - \phi_4}{4b} \\
 \alpha_4 &= \frac{\phi_1 - \phi_2 + \phi_3 - \phi_4}{4ab}
 \end{aligned} \tag{3.4}$$

Si ϕ se aproxima en términos de los valores nodales y las funciones de forma como se menciona anteriormente, se obtienen

$$\phi = N_1\phi_1 + N_2\phi_2 + N_3\phi_3 + N_4\phi_4 \tag{3.5}$$

Las funciones de forma son las siguientes

$$\begin{aligned}
 N_1 &= \frac{1}{4ab}(b-x)(a-y) \\
 N_2 &= \frac{1}{4ab}(b+x)(a-y) \\
 N_3 &= \frac{1}{4ab}(b+x)(a+y) \\
 N_4 &= \frac{1}{4ab}(b-x)(a+y)
 \end{aligned} \tag{3.6}$$

Para poder modelar estos elementos sin importar su ubicación y características se procede a hacer una transformación que se presenta a continuación

3.2 Sistema de coordenadas naturales

En la mayoría de las situaciones, el dominio físico a ser modelado no consiste en los costados rectos, fronteras ortogonales. Una región limitada por curvas puede ser desacreditada con elementos cuadriláteros con lados curvos para obtener una solución más precisa. La transformación de la recta de lados curvos se logra expresar las coordenadas x , y , en términos de coordenadas curvilíneas:

$$\begin{aligned}x &= x(r, s) \\ y &= y(r, s)\end{aligned}\tag{3.6}$$

La elección de r y s depende de la geometría del elemento. El sistema de coordenadas r , s se conoce normalmente como sistema de coordenadas naturales, cuando el sistema de coordenadas se encuentran dentro del rango $-1 \leq r \leq 1$, $-1 \leq s \leq 1$. Para el caso rectangular, podemos encontrar el conjunto de dimensiones coordenadas r , s tales que:

$$\begin{aligned}r &= \frac{x}{b} \\ s &= \frac{y}{a}\end{aligned}\tag{3.7}$$

Donde r y s efectivamente sustituirán la x y y que son las coordenadas globales por coordenadas locales para un elemento individual, se había referido con anterioridad a estos coeficientes específicos como índices de longitud (Fig. 7) por lo tanto, la utilización de nuestra definición de la ecuación (3.7), se determinan las funciones de forma bilineal como:

$$\begin{aligned}N_1 &= \frac{1}{4}(1-r)(1-s) \\ N_2 &= \frac{1}{4}(1+r)(1-s) \\ N_3 &= \frac{1}{4}(1+r)(1+s) \\ N_4 &= \frac{1}{4}(1-r)(1+s)\end{aligned}\tag{3.8}$$

El sistema de coordenadas local siempre varía entre -1 y 1. Tomamos ventaja de este hecho cuando se formulan los términos derivados. Fig.8 muestra el sistema de coordenadas r , s esta centrado en el elemento. Hay que tener en cuenta que no es importante para este nuevo sistema de coordenadas que sean ortogonales también puede ser curvilínea. Sin embargo, las derivadas con respecto a x y y no se pueden obtener directamente y requieren de una transformación adicional. Vamos a tratar este punto más adelante, por ahora, vamos a suponer un sistema simple de coordenadas cartesianas.

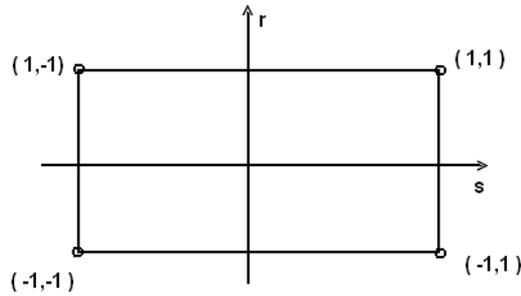


Fig.8: Sistema de coordenadas naturales

La evaluación de las derivadas $\partial N_i/\partial x$, $\partial N_i/\partial y$ es similar a las derivadas de los elementos triangulares mencionada anteriormente. La matriz Jacobiana puede ser usada para encontrar las derivadas a través del inverso del Jacobiano con respecto a la x , y y determinar la transformación.

Ejemplo 2: los valores de $\partial N_1/\partial x$ y $\partial N_1/\partial y$ en $x = 2$ y $y = 2$ para el elemento que se muestra en la Fig.8. La transformación de x y y se expresan de la siguiente forma

$$\begin{aligned} x &= \frac{1}{2}(3s + 5) \\ y &= r + 2 \end{aligned} \quad (3.9)$$

Al evaluar la matriz Jacobiana obtenemos:

$$J = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \quad (3.10)$$

La inversa del Jacobiano esta dada por:

$$J^{-1} = \frac{1}{3} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \quad (3.11)$$

Las derivadas $\partial N_1/\partial x$ y $\partial N_1/\partial y$ se puede calcular una vez que la derivada $\partial N_1/\partial s$ y $\partial N_1/\partial r$ sean conocidas. Sabemos que

$$N_1 = \frac{(1-s)(1-r)}{4} \quad (3.12)$$

Las derivadas están dadas por

$$\begin{aligned} \frac{\partial N_1}{\partial s} &= \frac{(1-r)}{4} \\ \frac{\partial N_1}{\partial r} &= \frac{(1-s)}{4} \end{aligned} \quad (3.13)$$

Despejando de la ecuación (3.9) y reemplazando $x = 2$ y $y = 2$ tenemos

$$\begin{aligned} s &= \frac{1}{3}(2x-5) = -\frac{1}{3} \\ r &= y-2 = 0 \end{aligned} \quad (3.14)$$

Así

$$\begin{aligned} \frac{\partial N_1}{\partial s} &= \frac{(1-0)}{4} = -\frac{1}{4} \\ \frac{\partial N_1}{\partial r} &= \frac{\left(1-\frac{1}{3}\right)}{4} = -\frac{1}{3} \end{aligned} \quad (3.15)$$

Reemplazando obtenemos

$$\begin{bmatrix} \frac{\partial N_1}{\partial x} \\ \frac{\partial N_1}{\partial y} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} -\frac{1}{4} \\ -\frac{1}{3} \end{bmatrix} = \begin{bmatrix} -\frac{1}{6} \\ -\frac{1}{3} \end{bmatrix} \quad (3.16)$$

A continuación se presenta el código en Matlab para la construcción de la matriz B de un elemento cuadrilátero de 4 nodos

```

%-----
function [B detJ1] = calcular_B(r,s,x,y)

% Calculo de las derivadas de x e y respecto a los ejes coordenados
dxdr = 1/4*(1+s)*x(1)-1/4*(1+s)*x(2)-1/4*(1-s)*x(3)+1/4*(1-s)*x(4);
dxds = 1/4*(1+r)*x(1)+1/4*(1-r)*x(2)-1/4*(1-r)*x(3)-1/4*(1+r)*x(4);
dydr = 1/4*(1+s)*y(1)-1/4*(1+s)*y(2)-1/4*(1-s)*y(3)+1/4*(1-s)*y(4);
dyds = 1/4*(1+r)*y(1)+1/4*(1-r)*y(2)-1/4*(1-r)*y(3)-1/4*(1+r)*y(4);

% Calculo de las derivadas de las funciones de forma respecto a las
% coordenadas parametrizadas
dN1dr = 1/4 + 1/4*s;
dN1ds = 1/4 + 1/4*r;
dN2dr = -1/4 - 1/4*s;
dN2ds = 1/4 - 1/4*r;
dN3dr = -1/4 + 1/4*s;
dN3ds = -1/4 + 1/4*r;
dN4dr = 1/4 - 1/4*s;
dN4ds = -1/4 - 1/4*r;

% Almacenamiento del jacobiano
J1 = [dxdr dydr; ...
      dxds dyds];

detJ1 = det(J1);

% Almacenamiento de la matriz de derivadas respecto a coordenadas
% parametrizadas
derN = [dN1dr dN2dr dN3dr dN4dr; ...
        dN1ds dN2ds dN3ds dN4ds];

```

```

% Calculo de la matriz de derivadas de las funciones de forma respecto a
% los ejes coordenados
dN = (inv(J1)*derN)';

% Calculo de la matriz B
B = [dN(1,1)    0    dN(2,1)    0    dN(3,1)    0    dN(4,1)    0; ...
     0    dN(1,2)    0    dN(2,2)    0    dN(3,2)    0    dN(4,2); ...
     dN(1,2) dN(1,1) dN(2,2) dN(2,1) dN(3,2) dN(3,1) dN(4,2) dN(4,1)];

end
%-----

```

4. Elemento rectangular cuadrático bidimensional

4.1 Funciones de forma

Se puede desarrollar un modelo dos dimensional, de orden superior, para un elemento cuadrilátero, tal como lo hicimos para el elemento triangular. Hay dos formas en que esto se puede hacer que conducen a dos diferentes elementos de segundo grado: uno con ocho nodos, uno en cada esquina y otra en el punto medio de cada lado, y el otro cuenta con nueve nodos, siendo el noveno en el centro de la elemento este último es sobre el que vamos a realizar la construcción ya que este se utilizan predominantemente en la industria y se ha comprobado que es suficiente para la mayoría de los efectos de cálculo.

El polinomio de interpolación para el elemento de ocho nodos, se muestra en la Fig.9 Consta de ocho términos.

La variable desconocida ϕ se expresa como

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy + \alpha_5 x^2 + \alpha_6 y^2 + \alpha_7 x^2 y + \alpha_8 xy^2 + \quad (4.1)$$

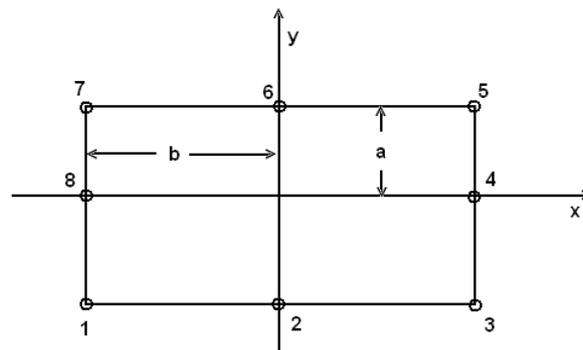


Fig.9: Elemento cuadrilátero de ocho nodos

Si los valores del nodo se definen como en la Fig. 9.

$$\begin{aligned}
\phi &= \phi_1 & \text{para } x = -b \quad y = -a \\
\phi &= \phi_2 & \text{para } x = 0 \quad y = -a \\
\phi &= \phi_3 & \text{para } x = b \quad y = -a \\
\phi &= \phi_4 & \text{para } x = b \quad y = 0 \\
\phi &= \phi_5 & \text{para } x = b \quad y = a \\
\phi &= \phi_6 & \text{para } x = 0 \quad y = a \\
\phi &= \phi_7 & \text{para } x = -b \quad y = a \\
\phi &= \phi_8 & \text{para } x = -b \quad y = 0
\end{aligned} \tag{4.2}$$

Sustituyendo los valores de la ecuación (2.8) a (2.7) los valores de las siguientes ocho ecuaciones (expresado en forma de matriz):

$$\begin{bmatrix} 1 & -b & -a & ab & b^2 & a^2 & -ab^2 & -a^2b \\ 1 & 0 & -a & 0 & 0 & a^2 & 0 & 0 \\ 1 & b & -a & -ab & b^2 & a^2 & -ab^2 & -a^2b \\ 1 & b & 0 & 0 & b^2 & 0 & 0 & 0 \\ 1 & b & a & ab & b^2 & a^2 & -ab^2 & -a^2b \\ 1 & 0 & a & 0 & 0 & a^2 & 0 & 0 \\ 1 & -b & a & -ab & b^2 & a^2 & -ab^2 & -a^2b \\ 1 & -b & 0 & 0 & b^2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \end{bmatrix} \tag{4.3}$$

Desde que el α_i 's con $i=1,\dots,8$, se puede obtener.

La aproximación de ϕ en función de los valores nodales y de cinco funciones de forma

$$\phi = N_1\phi_1 + N_2\phi_2 + N_3\phi_3 + N_4\phi_4 + N_5\phi_5 + N_6\phi_6 + N_7\phi_7 + N_8\phi_8 \tag{4.4}$$

Las funciones de forma son las siguientes

$$\begin{aligned}
N_1 &= \frac{1}{4(ab)^2} (b-x)(a-y)(ab+ca+yb) \\
N_2 &= \frac{1}{4ab^2} (b^2+x^2)(a-y) \\
N_3 &= \frac{1}{4(ab)^2} (b+x)(a-y)(ax+by-ab) \\
N_4 &= \frac{1}{4a^2b} (a^2-y^2)(b+x) \\
N_5 &= \frac{1}{4(ab)^2} (b+x)(a+y)(ax+by-ab) \\
N_6 &= \frac{1}{4ab^2} (b^2-x^2)(a+y) \\
N_7 &= \frac{1}{4(ab)^2} (b-x)(a+y)(ab+ax-by) \\
N_8 &= \frac{1}{4a^2b} (a^2-y^2)(b-x)
\end{aligned} \tag{4.5}$$

A continuación se presenta el código en Matlab para la construcción de la matriz B de un elemento cuadrilátero de 8 nodos

```

%-----
function [B detJ1] = calcular_B(r,s,x,y)

% Calculo de las derivadas de x e y respecto a los ejes
% coordenados
dxdr = 1/4*((1+s)*x(1)-(1+s)*x(2)-(1-s)*x(3)+(1-s)*x(4)-2*(1+s)*(2*r)*x(5)
-2*(1-s^2)*x(6)-2*(1-s)*(2*r)*x(7)+2*(1-s^2)*x(8));
dxds = 1/4*((1+r)*x(1)+(1-r)*x(2)-(1-r)*x(3)-(1+r)*x(4)+2*(1-r^2)*x(5)-2*
(1-r)*(2*s)*x(6)-2*(1-r^2)*x(7)-2*(1+r)*(2*s)*x(8));
dydr = 1/4*((1+s)*y(1)-(1+s)*y(2)-(1-s)*y(3)+(1-s)*y(4)-2*(1+s)*(2*r)*x(5)
-2*(1-s^2)*x(6)-2*(1-s)*(2*r)*x(7)+2*(1-s^2)*x(8));
dyds = 1/4*((1+r)*y(1)+(1-r)*y(2)-(1-r)*y(3)-(1+r)*y(4)+2*(1-r^2)*x(5)-2*
(1-r)*(2*s)*x(6)-2*(1-r^2)*x(7)-2*(1+r)*(2*s)*x(8));

% Calculo de las derivadas de las funciones de forma respecto a las
% coordenadas parametrizadas
dN1dr = 1/4*(1+s);
dN1ds = 1/4*(1+r);
dN2dr = -1/4*(1+s);
dN2ds = 1/4*(1-r);
dN3dr = -1/4*(1-s);
dN3ds = -1/4*(1-r);
dN4dr = 1/4*(1-s);
dN4ds = -1/4*(1+r);
dN5dr = -1/2*(1+s)*(2*r);
dN5ds = 1/2*(1-r^2);
dN6dr = -1/2*(1-s^2);
dN6ds = -1/2*(1-r)*(2*s);
dN7dr = -1/2*(1-s)*(2*r);
dN7ds = -1/2*(1-r^2);
dN8dr = 1/2*(1-s^2);
dN8ds = -1/2*(1+r)*(2*s);

% Almacenamiento del jacobiano
J1 = [dxdr dydr; ...
      dxds dyds];

detJ1 = det(J1);

% Almacenamiento de la matriz de derivadas respecto a coordenadas
parametrizadas
derN = [dN1dr dN2dr dN3dr dN4dr dN5dr dN6dr dN7dr dN8dr ; ...
        dN1ds dN2ds dN3ds dN4ds dN5ds dN6ds dN7ds dN8ds ];

% Calculo de la matriz de derivadas de las funciones de forma respecto a
% los ejes coordenados
dN = (inv(J1)*derN)';

% Calculo de la matriz B
B=[dN(1,1) 0 dN(2,1) 0 dN(3,1) 0 dN(4,1) 0 dN(5,1) 0 dN(6,1) 0 dN(7,1) 0
   dN(8,1) 0; 0 dN(1,2) 0 dN(2,2) 0 dN(3,2) 0 dN(4,2) 0 dN(5,2) 0 dN(6,2)
   0 dN(7,2) 0 dN(8,2);dN(1,2) dN(1,1) dN(2,2) dN(2,1) dN(3,2) dN(3,1)
   dN(4,2) dN(4,1) dN(5,2) dN(5,1) dN(6,2) dN(6,1) dN(7,2) dN(7,1) dN(8,2)
   dN(8,1)];
%-----

```

4.2 Integración numérica por medio de la cuadratura de Gauss

La formulación de las integrales dependen de r y s los límites de integración simple. Esto es especialmente ventajoso cuando se trata de los límites de curvas o límites que no son paralelos a los ejes de coordenadas. Las integrales se definen como

$$\int_{-a-b}^a \int_{-1}^b F(x,y) dx dy = \int_{-1}^1 \int_{-1}^1 f(r,s) |J| dr ds \quad (4.6)$$

La matriz se convierte así en

$$\begin{aligned} K &= \int_{-1}^1 \int_{-1}^1 K \left[\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right] dx dy \\ &= \int_{-1}^1 \int_{-1}^1 K \left[\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right] |J| dr ds \end{aligned} \quad (4.7)$$

Donde las derivadas $\partial N_i / \partial x$ y $\partial N_i / \partial y$ se pueden obtener de la ecuación 3.10 en función de r , s , $\partial N_i / \partial r$ y $\partial N_i / \partial s$. Todas las integrales de área se convierten el producto $N_i N_j$ como se muestra a

$$\int \int_{\Omega} N_i N_j dx dy = \int_{-1}^1 \int_{-1}^1 N_i N_j |J| dr ds \quad (4.8)$$

Integración del termino derecho esta dado por la ecuación 4.7, ya no implica polinomios simples debido a la condición $1/|J|$ de la expresión de las derivadas con respecto a r y s , puede llegar a ser molesto. La práctica más común es utilizar el procedimiento de integración numérica se conoce como la cuadratura de Gauss. El procedimiento es fácil de programar y tiene soluciones muy aproximadas para las integrales

Una cuadratura de Gauss aproxima una integral definida por medio de una suma ponderada de un conjunto finito de puntos. Por ejemplo, en una dimensión tenemos

$$\int_{-1}^1 f(r) dr \cong \sum_{i=1}^N W_i f(r_i) \quad (4.9)$$

Donde N es el número de puntos de integración, W_i son los pesos de Gauss, y los R_i son los puntos de Gauss. Con los puntos de Gauss, podemos integrar exactamente un polinomio de grado $2N-1$. Así pues, si $f(r)$ es un polinomio cúbico $f(r) = a + br + cr^2 + dr^3$, entonces la elección $N=2$, los valores de R_i se definen como

$$\begin{aligned} r_1 &= -1/\sqrt{3} \\ r_2 &= 1/\sqrt{3} \end{aligned} \quad (4.10)$$

Las ponderaciones serian $W_1 = W_2 = 1$. Para las integrales dobles en la ecuación 4.8, tenemos.

$$\int_{-1}^1 \int_{-1}^1 f(r,s) dr ds \cong \sum_{i=1}^N \sum_{j=1}^N W_i W_j f(r_i, s_j) \quad (4.11)$$

Los puntos de Gauss están definidos por los siguientes valores

$$\begin{aligned} r_1 &= -1/\sqrt{3} \\ r_2 &= 1/\sqrt{3} \\ s_1 &= -1/\sqrt{3} \\ s_2 &= 1/\sqrt{3} \end{aligned} \quad (4.12)$$

Para cuatro puntos de Gauss se utilizan los cuatro elementos del cuadrilátero como se muestra en la Fig. 10 y el tabla2

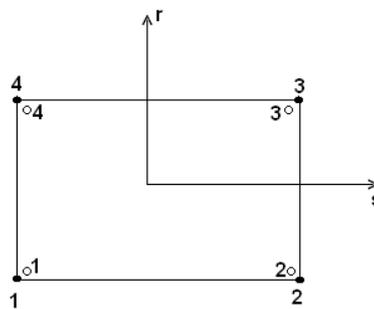


Fig.10: Cuadrilátero con cuatro nodos y cuatro puntos de Gauss

Tabla2: Integración con cuatro puntos de Gauss

Puntos de Gauss	r_i	s_i
1	-0,577	-0,577
2	0,577	-0,577
3	0,577	0,577
4	-0,577	0,577

Si el elemento cuadrático es de ocho nodos o de nueve se utilizan la integral que aparece en la ecuación 4.7 se quedando definida como :

$$\int_{-1}^1 \int_{-1}^1 f(r,s) dr ds \cong \sum_{i=1}^3 \sum_{j=1}^3 W_i W_j f(r_i, s_j) \quad (4.13)$$

Los nueve puntos de integración son presentados en la Fig11 y los pesos se muestran en la tabla 3

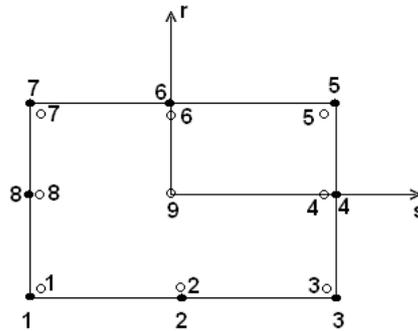


Fig.11: Integración numérica con nueve puntos de Gauss

Tabla3: Pesos de Gauss.

Puntos de Gauss	ri	si	wi	wj
1	-0,775	-0,775	0,556	0,556
2	0,000	-0,775	0,889	0,556
3	0,775	-0,775	0,556	0,556
4	0,775	0,000	0,556	0,889
5	0,775	0,775	0,556	0,556
6	0,000	0,775	0,889	0,556
7	-0,775	0,775	0,556	0,556
8	-0,775	0,000	0,556	0,889
9	0,000	0,000	0,889	0,889

La solución obtenida con cuadriláteros bilineales implica una matriz 4x4, mientras que un cuadrilátero cuadrático implica una matriz de 8x8. Además, los elementos bilineales requieren una evaluación en 4 puntos de Gauss, y no involucran ecuaciones cuadráticas, por lo que es nueve veces menos costoso en términos de cálculo para construir la matriz de rigidez de elemento por elemento. Esto puede ser una consideración importante al escoger el tipo de elemento que se utilizará, ya que significa una diferencia de casi un orden de magnitud en el tiempo computacional.

A continuación se presenta el código en Matlab para la construcción de la integración numérica para un cuadrilátero de 4 nodos

```

%-----
pg = [sqrt(1/3); -sqrt(1/3)]; %pesos de gauss

%%Calcular k de del elemento
material_element = elements(element,end);
thickness_element = 1;
%matris k de un elemento
aux = 0;
for ri = 1:2
    r = pg(ri);

```

```

    for si = 1:2
        s = pg(si);
        %%Calcular B del elemento
        [B detJ1] = calcular_B(r,s,x,y);
        k_element = k_element +
B'*D{material_element}*B*detJ1*thickness_element;
        aux = aux + B;
    end
end
%-----

```

A continuación se presenta el código en Matlab para la construcción de la integración numérica para un cuadrilátero de 8 nodos

```

%-----
pg = [0.77459669241483; 0 ; -0.77459669241483];

%%Calcular k de del elemento
material_element = elements(element,end);
thickness_element = 1;
k_element = zeros(16,16);
%matris k de un elemento
aux = 0;
for ri = 1:3
    r = pg(ri);
    for si = 1:3
        s = pg(si);
        %%Calcular B del elemento
        [B detJ1] = calcular_B(r,s,x,y);
        k_element = k_element + B'*D{material_element}*B*detJ1*thickness_element;
        aux = aux + B;
    end
end
end
%-----

```

5. Elementos tridimensionales

Los elementos más comunes tridimensionales provienen de variaciones de los elementos bidimensionales, presentados anteriormente, es decir, el tetraedro es un elemento lineal limitado por los lados rectos; pero el hexaedro que es de orden superior puede tener superficies curvas. Esto mismo ocurre con los elementos tetraédricos y los cilindros.

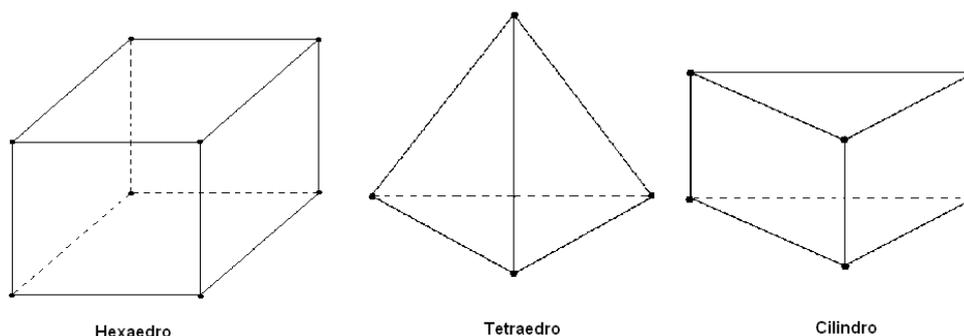


Fig.12: Algunos elementos tridimensionales

6. Tetraedro lineal

Un tetraedro lineal esta definido por cuatro nodos como se muestra en la Fig.13, y su función de interpolación esta dada por:

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \quad (6.1)$$

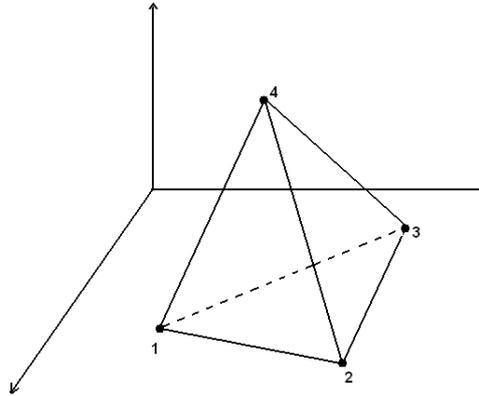


Fig.13: *Elemento Tetraedro*

Para los cuatro nodos situados en los vértices del tetraedro, se puede escribir el siguiente sistema de ecuaciones

$$\begin{aligned} \phi_1 &= \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 + \alpha_4 z_1 \\ \phi_2 &= \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 + \alpha_4 z_2 \\ \phi_3 &= \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 + \alpha_4 z_3 \\ \phi_4 &= \alpha_1 + \alpha_2 x_4 + \alpha_3 y_4 + \alpha_4 z_4 \end{aligned} \quad (6.2)$$

de forma matricial tenemos

$$\phi = C \alpha \quad (6.3)$$

Que también se puede escribir como

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \quad (6.4)$$

Tomando la inversa de C , el vector de la columna α se obtiene como

$$\phi = [1 \quad x \quad y \quad z] \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = [1 \quad x \quad y \quad z] \cdot C^{-1} \bar{\phi} \quad (6.5)$$

La cantidad $[1 \ x \ y \ z] \cdot C^{-1}$ es en realidad la forma de las funciones matriciales, es decir,

$$\phi = N \bar{\phi} \quad (6.6)$$

Donde

$$N = [1 \ x \ y \ z] \cdot C^{-1} \quad (6.7)$$

Un sistema de coordenadas de volumen para el tetraedro se puede establecer de la misma manera como el sistema coordenado de áreas para un triangulo de dos dimensiones, en el hexaedro de tres dimensiones, requiere cuatro relaciones de distancia, cada una normal a un lado: L_1 , L_2 , L_3 y L_4 . El volumen V_1 se muestra en la Fig.14.

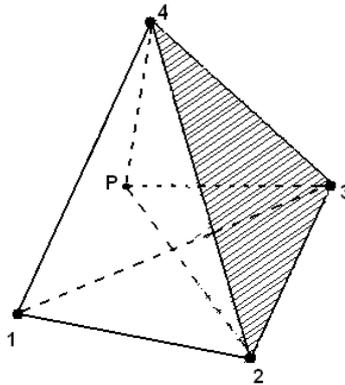


Fig.14: Sistema coordenado de volumen

Las cuatro coordenadas están relacionadas de la siguiente forma

$$L_1 + L_2 + L_3 + L_4 = 1 \quad (6.8)$$

La forma lineal de la función queda definida por

$$\begin{aligned} N_1 &= L_1 \\ N_2 &= L_2 \\ N_3 &= L_3 \\ N_4 &= L_4 \end{aligned} \quad (6.9)$$

Las integrales de volumen pueden ser evaluadas fácilmente de la siguiente relación

$$\int_V L_1^a L_2^b L_3^c L_4^d dV = \frac{a! b! c! d!}{(3 + a + b + c + d)!} 6V \quad (6.10)$$

Del mismo modo que se utilizo para los elementos triangulares.

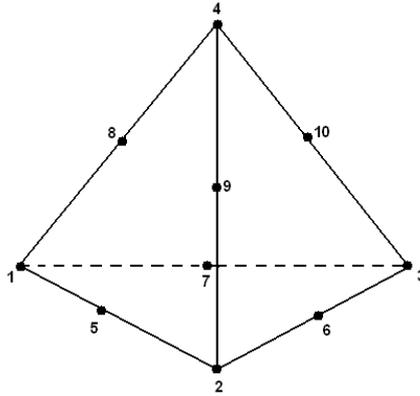


Fig.15: *Tetraedro cuadrático*

La funciones de forma de los elementos de orden superior son similares a las del tetraedro lineal, Por ejemplo, la función de forma en el nodo 1 para el tetraedro de segundo grado, que se muestra en la Fig.15 queda definida por

$$N_1 = (2L_1 - 1)L_1 \quad (6.11)$$

Cuando $L_1 = 0$ pasa a través de los nodos 2, 3, 4, 6, 9 y 10. Para $L_1 = 1/2$, el plano pasa a través de los nodos 5, 7 y 8. En el nodo 5,

$$N_5 = 4L_1L_2 \quad (6.12)$$

Cuando $L_2 = 0$ pasa a través de los nodos 1, 3, 4, 7, 8 y 10. La función de las relaciones de forma adicional puede ser fácilmente evaluada por referencia a la ecuación 4.26. Que da las funciones de forma para el elemento triangular de dos dimensiones de segundo grado.

Ejemplo 3: Determinar las funciones de forma para el elemento de tetraedro lineal que se muestra en la Fig. 16 de la ecuación 6.4.

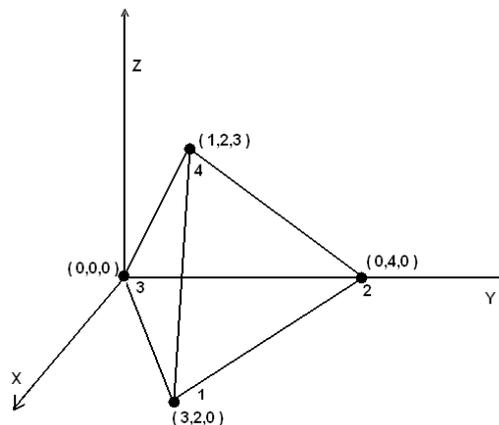


Fig.16: *Tetraedro lineal*

En primer lugar evaluaremos C :

$$C = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 2 & 0 \\ 1 & 0 & 4 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 2 & 5 \end{bmatrix} \quad (6.13)$$

Se hace necesario encontrar la inversa C^{-1} a fin de obtener las funciones de forma como se define por la ecuación 6.7. La inversa se obtiene de la siguiente manera

$$C = \frac{adjC}{|c|} = \frac{\begin{bmatrix} 0 & 0 & 60 & 0 \\ 20 & -10 & -10 & 0 \\ 0 & 15 & -15 & 0 \\ -4 & -4 & -4 & 12 \end{bmatrix}}{60} \quad (6.14)$$

Las funciones de forma son dadas por el siguiente producto

$$N = [1 \quad x \quad y \quad z] \cdot C^{-1} = \frac{[1 \quad x \quad y \quad z]}{60} \begin{bmatrix} 0 & 0 & 60 & 0 \\ 20 & -10 & -10 & 0 \\ 0 & 15 & -15 & 0 \\ -4 & -4 & -4 & 12 \end{bmatrix}$$

$$= \frac{1}{60} [(20x - 4z)(-10x + 15y - 4z)(60 - 10x - 15y - 4z)(12z)] \quad (6.15)$$

Así, quedan definidas las funciones de forma para cada nodo

$$N_1 = \frac{1}{15}(5x - z)$$

$$N_2 = \frac{1}{60}(-10x + 15y - 4z)$$

$$N_3 = \frac{1}{60}(60 - 10x - 15y - 4z)$$

$$N_4 = \frac{z}{5} \quad (6.16)$$

A continuación se presenta el código en Matlab para la construcción la matriz B y el volumen V para un tetra de 4 nodos

```

%-----
%calcular_V
function V = calcular_V(coordenate_node1,coordenate_node2,coordenate_node3,
    coordenate_node4)

m = [coordenate_node1-coordenate_node2,coordenate_node2- coordenate_node3,
    coordenate_node3-coordenate_node4];

V = (1/6)*abs(det(m));

```

end

```
%-----  
function B = calcular_B(coordenate_nodel,coordenate_node2,coordenate_node3,  
    coordenate_node4)  
  
% Calculo de las derivadas de de x e y respecto a los ejes coordenados  
x(1) = coordenate_nodel(1);  
x(2) = coordenate_node2(1);  
x(3) = coordenate_node3(1);  
x(4) = coordenate_node4(1);  
y(1) = coordenate_nodel(2);  
y(2) = coordenate_node2(2);  
y(3) = coordenate_node3(2);  
y(4) = coordenate_node4(2);  
z(1) = coordenate_nodel(3);  
z(2) = coordenate_node2(3);  
z(3) = coordenate_node3(3);  
z(4) = coordenate_node4(3);  
dxdr = -1*x(1)+x(2);  
dxds = -1*x(1)+x(3);  
dxdt = -1*x(1)+x(4);  
dydr = -1*y(1)+y(2);  
dyds = -1*y(1)+y(3);  
dydt = -1*y(1)+y(4);  
dzdr = -1*z(1)+z(2);  
dzds = -1*z(1)+z(3);  
dzdt = -1*z(1)+z(4);  
  
% Calculo de las derivadas de las funciones de forma respecto a las  
% coordenadas parametrizadas  
dN1dr = -1;  
dN1ds = -1;  
dN1dt = -1;  
dN2dr = 1;  
dN2ds = 0;  
dN2dt = 0;  
dN3dr = 0;  
dN3ds = 1;  
dN3dt = 0;  
dN4dr = 0;  
dN4ds = 0;  
dN4dt = 1;  
  
% Almacenamiento del jacobiano  
J1 = [dxdr dydr dzdr; dxds dyds dzds; dxdt dydt dzdt];  
% Almacenamiento de la matriz de derivadas respecto a coordenadas  
% parametrizadas  
derN = [dN1dr dN2dr dN3dr dN4dr;  
        dN1ds dN2ds dN3ds dN4ds;  
        dN1dt dN2dt dN3dt dN4dt];  
  
% Calculo de la matriz de derivadas de las funciones de forma respecto a  
% los ejes coordenados  
dN = inv(J1)*derN;  
  
B = [dN(1,1) 0 0 dN(1,2) 0 0 dN(1,3) 0 0 dN(1,4) 0 0; 0 dN(2,1) 0 0 dN(2,2)  
    0 0 dN(2,3) 0 0 dN(2,4) 0; 0 0 dN(3,1) 0 0 dN(3,2) 0 0 dN(3,3) 0 0  
    dN(3,4); dN(2,1) dN(1,1) 0 dN(2,2) dN(1,2) 0 dN(2,3) dN(1,3) 0 dN(2,4)  
    dN(1,4) 0; 0 dN(3,1) dN(2,1) 0 dN(3,2) dN(2,2) 0 dN(3,3) dN(2,3) 0  
    dN(3,4) dN(2,4); dN(3,1) 0 dN(1,1) dN(3,2) 0 dN(1,2) dN(3,3) 0 dN(1,3)  
    dN(3,4) 0 dN(1,4)];  
  
end  
%-----
```

7. Hexaedro lineal

El tetraedro es un elemento simple, que se extiende directamente desde el elemento triangular. Sin embargo, el tetraedro es difícil de visualizar en una compleja maraña tridimensional y puede requerir un esfuerzo considerable para definir el elemento de conectividad. Una solución más precisa con menos esfuerzo, pero más puntos de integración, se puede lograr con el elemento de hexaedro. El elemento más simple de esta familia es un hexaedro de ocho nodos, que se muestra en la Fig. 17

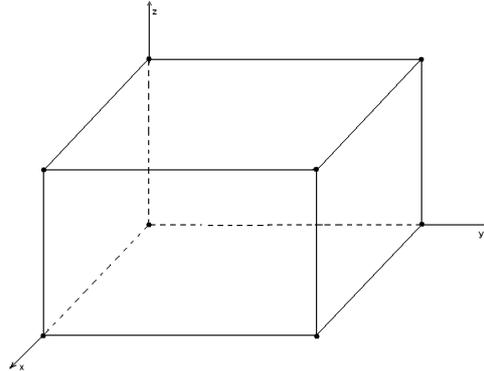


Fig.17: Hexaedro de ocho nodos

La función de interpolación esta dada por

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z + \alpha_5 xy + \alpha_6 xz + \alpha_7 yz + \alpha_8 xyz \quad (7.1)$$

La ecuación 7.1 se puede expresar en forma de matriz como se presenta a continuación

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_8 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 & x_1 y_1 & x_1 z_1 & y_1 z_1 & x_1 y_1 z_1 \\ 1 & x_2 & y_2 & z_2 & x_2 y_2 & x_2 z_2 & y_2 z_2 & x_2 y_2 z_2 \\ \vdots & \vdots \\ 1 & x_8 & y_8 & z_8 & x_8 y_8 & x_8 z_8 & y_8 z_8 & x_8 y_8 z_8 \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_8 \end{bmatrix} \quad (7.2)$$

Dado que $\alpha = C^{-1}\phi$, podemos encontrar los cuatro valores de la expresión

$$\phi = [1 \quad x \quad y \quad z \quad xy \quad xz \quad yz \quad xyz] \cdot C^{-1} \bar{\phi} \quad (7.3)$$

A partir de qué forma las funciones se obtienen como se mostró anteriormente

$$N = [1 \quad x \quad y \quad z \quad xy \quad xz \quad yz \quad xyz] \cdot C^{-1} \quad (7.3)$$

Como en los elementos cuadriláteros bidimensionales, un sistema de coordenadas naturales también puede ser diseñado para el hexaedro, como se muestra en la Fig. 18. Las funciones de forma quedan definidas así.

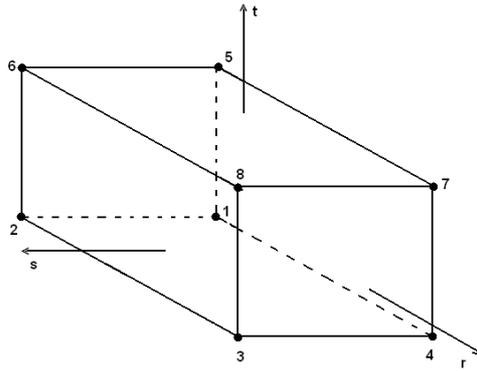


Fig.18: Sistema de coordenadas naturales

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} (1-r)(1-s)(1-t) \\ (1+r)(1-s)(1-t) \\ (1+r)(1+s)(1-t) \\ (1-r)(1+s)(1-t) \\ (1-r)(1-s)(1+t) \\ (1+r)(1-s)(1+t) \\ (1+r)(1+s)(1+t) \\ (1-r)(1+s)(1+t) \end{bmatrix} \quad (7.4)$$

La ecuación 7,4 se puede escribir en forma más compacta como

$$N_i = \frac{1}{8}(1 - rr_i)(1 - ss_i)(1 - tt_i) \quad \text{con } i = 1, 2, \dots, 8 \quad (7.5)$$

Cuando r_i , s_i y $t_i = \pm 1$, dependiendo de la ubicación nodal.

Ejemplo 4: Determinar la expresión N_1 con coordenadas r, s, t . De la ecuación 7.5.

$$N_1 = \frac{1}{8}(1 - rr_1)(1 - ss_1)(1 - tt_1) \quad (7.6)$$

Podemos ver de la a la Fig.15 que

$$r_1 = -1 \quad s_1 = -1 \quad t_1 = -1$$

Así,

$$N_1 = \frac{1}{8}(1 - r)(1 - s)(1 - t) \quad (7.7)$$

La extensión de los 8 nodos de elemento de segundo grado a los 20 nodos hexaedro cuadrática puede hacerse fácilmente mediante la adición de un nodo en la mitad de cada borde del elemento. Esto puede verse en la Fig.19

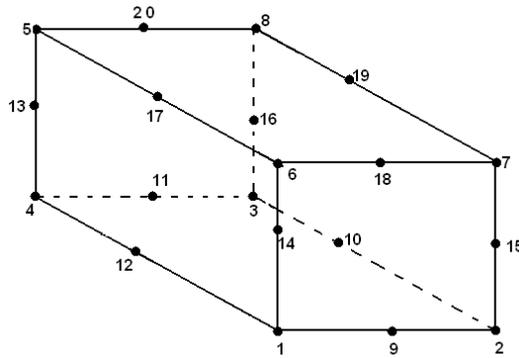


Fig.19: *Hexaedro cuadrático*

Las funciones de forma para los nodos de esquina se obtienen de la siguiente relación

$$N_i = \frac{1}{8}(1 - rr_i)(1 - ss_i)(1 - tt_i)(rr_i + ss_i + tt_i - 2) \quad \text{con } i = 1,2,\dots,8 \quad (7.8)$$

Para los nodos intermedios se tienen estas relaciones,

$$N_i = \frac{1}{4}(1 - r^2)(1 - ss_i)(1 - tt_i) \quad \text{con } i = 10,12,18,20 \quad (7.9)$$

$$N_i = \frac{1}{8}(1 - rr_i)(1 - s^2)(1 - tt_i) \quad \text{con } i = 9,11,17,19 \quad (7.10)$$

$$N_i = \frac{1}{8}(1 - rr_i)(1 - ss_i)(1 - t^2) \quad \text{con } i = 13,14,15,16 \quad (7.11)$$

Para un sistema de coordenadas locales orientado como se muestra en la Fig.15. El elemento hexaedro de 9 nodos se extiende a uno de 27 nodos con un nodo en el centro de cada una de las seis caras y uno en el centro del elemento.

Observe que cuando una de las coordenadas natural es ± 1 , los tres elementos de dimensiones se reducen a sus dos dimensiones homólogas. Asimismo, cuando dos de las coordenadas de la iguales ± 1 , la forma de la función se reduce a la función de una forma tridimensional, permitiendo tres elementos de dimensiones que se unió a los elementos de línea. La continuidad entre los elementos de existir más de una cara de elemento (área) de un elemento tridimensional, de los valores nodales describir la variación de la variable (ϕ) desconocida de forma idéntica en la cara elemento común a los elementos adyacentes.

A continuación se presenta el código en Matlab para la construcción la matriz B para un hexaedro de 8 nodos

```

%-----
function [B detJ1] = calcular_B(r,s,t,x,y)

% Calculo de las derivadas de x e y respecto a los ejes coordenados
dxdr = (1/2)*(x(1)-x(4));
dxds = 0;
dxdt = 0;
dydr = 0;

```

```

dyds = (1/2)*(y(2)-y(1));
dydt = 0;
dzdr = 0;
dzds = 0;
dzdt = (1/2)*(z(5)-z(1));

% Calculo de las derivadas de las funciones de forma respecto a las
% coordenadas parametrizadas
dN1dr = (-1/8)*(1-s)*(1-t);
dN1ds = (-1/8)*(1-r)*(1-t);
dN1dt = (-1/8)*(1-r)*(1-s);
dN2dr = (1/8)*(1-s)*(1-t);
dN2ds = (-1/8)*(1+r)*(1-t);
dN2dt = (-1/8)*(1+r)*(1-s);
dN3dr = (1/8)*(1+s)*(1-t);
dN3ds = (1/8)*(1+r)*(1-t);
dN3dt = (-1/8)*(1+r)*(1+s);
dN4dr = (-1/8)*(1+s)*(1-t);
dN4ds = (1/8)*(1-r)*(1-t);
dN4dt = (-1/8)*(1-r)*(1+s);
dN5dr = (-1/8)*(1-s)*(1+t);
dN5ds = (-1/8)*(1-r)*(1+t);
dN5dt = (1/8)*(1-r)*(1-s);
dN6dr = (1/8)*(1-s)*(1+t);
dN6ds = (-1/8)*(1+r)*(1+t);
dN6dt = (1/8)*(1+r)*(1-s);
dN7dr = (1/8)*(1+s)*(1+t);
dN7ds = (1/8)*(1+r)*(1+t);
dN7dt = (1/8)*(1+r)*(1+s);
dN8dr = (-1/8)*(1+s)*(1+t);
dN8ds = (1/8)*(1-r)*(1+t);
dN8dt = (1/8)*(1-r)*(1+s);

% Almacenamiento del jacobiano
J1 = [dxdr dydr dzdr; ...
      dxds dyds dzds;...
      dxdt dydt dzdt];

detJ1 = det(J1);

% Almacenamiento de la matriz de derivadas respecto a coordenadas
% parametrizadas
derN = [dN1dr dN2dr dN3dr dN4dr dN5dr dN6dr dN7dr dN8dr; ...
        dN1ds dN2ds dN3ds dN4ds dN5ds dN6ds dN7ds dN8ds;...
        dN1dt dN2dt dN3dt dN4dt dN5dt dN6dt dN7dt dN8dt];

% Calculo de la matriz de derivadas de las funciones de forma respecto a
% los ejes coordenados
dN = (inv(J1)*derN)';

% Calculo de la matriz B
B = [dN(1,1) 0 0 dN(2,1) 0 0 dN(3,1) 0 0 dN(4,1) 0 0 dN(5,1) 0 0 dN(6,1) 0
     0 dN(7,1) 0 0 dN(8,1) 0 0; 0 dN(1,2) 0 0 dN(2,2) 0 0 dN(3,2) 0 0
     dN(4,2) 0 0 dN(5,2) 0 0 dN(6,2) 0 0 dN(7,2) 0 0 dN(8,2) 0; 0 0
     dN(1,3) 0 0 dN(2,3) 0 0 dN(3,3) 0 0 dN(4,2) 0 0 dN(5,3) 0 0 dN(6,3) 0
     0 dN(7,3) 0 0 dN(8,3); dN(1,2) dN(1,1) 0 dN(2,2) dN(2,1) 0 dN(3,2)
     dN(3,1) 0 dN(4,2) dN(4,1) 0 dN(5,2) dN(5,1) 0 dN(6,2) dN(6,1) 0 dN(7,2)
     dN(7,1) 0 dN(8,2) dN(8,1) 0; dN(1,3) 0 dN(1,1) dN(2,3) 0 dN(2,1) dN(3,3)
     0 dN(3,1) dN(4,3) 0 dN(4,1) dN(5,3) 0 dN(5,1) dN(6,3) 0 dN(6,1) dN(7,3)
     0 dN(7,1) dN(8,3) 0dN(8,1); 0dN(1,3) dN(1,2) 0dN(2,3) dN(2,2) 0 dN(3,3)
     dN(3,2) 0 dN(4,3) dN(4,2) 0 dN(5,3) dN(5,2) 0 dN(6,3) dN(6,2) 0 dN(7,3)
     dN(7,2) 0 dN(8,3) dN(8,2)];

end
%-----

```

8. Integración numérica

En este punto, la dificultad en la evaluación de las integrales de funciones de forma relacionadas con más de unos pocos nodos que debería ser obvio. Utilizando las coordenadas de volumen, los elementos tetraédricos se pueden evaluar exactamente usando la ecuación. 6,10, sin embargo, la cantidad de álgebra asociados con la multiplicación del producto interno se hace muy laboriosa y no rápida. Las integrales de área que se producen en dos elementos triangulares son dimensiones de la forma.

$$I = \int_0^1 \int_0^{1-L_2} f(L_1, L_2, L_3) |J| dL_1 dL_2$$

Cuando las funciones de interpolación se expresan en términos de coordenadas de área. Dado que el jacobiano esta en función de las coordenadas de área, una forma explícita de la inversa del jacobiano es casi imposible. Si bien algunas integrales pueden ser evaluadas usando la ecuación 2.17, que es las dos dimensiones de la forma de la ecuación 7.14, la probabilidad de error es grande ya que el número de términos aumenta dramáticamente.

$$\int_0^1 \int_0^{1-L_2} f(L_1, L_2, L_3) |J| dL_1 dL_2 = \frac{1}{2} \sum_{i=1}^n w_i g[(L_1)_i, (L_2)_i, (L_3)_i]$$

Si el orden de la integración (n) se determina mediante la suma de la potencia de las coordenadas L_1, L_2 y L_3

De manera similar, los elementos del tetraedro son evaluados como

$$\begin{aligned} I &= \int_0^1 \int_0^{1-L_1} \int_0^{1-L_2} f(L_1, L_2, L_3, L_4) |J| dL_1 dL_2 dL_3 \\ &= \frac{1}{6} \sum_{i=1}^n w_i g[(L_1)_i, (L_2)_i, (L_3)_i, (L_4)_i] \end{aligned} \quad (8.1)$$

La integración numérica de los puntos nodales del tetraedros se muestran en la tabla4 es que queremos integrar el producto $L_1 L_2 L_4$, la suma de los exponentes es 3, y requiere una integración de cuarto orden ($n = 4$ puntos).

Tabla4: Puntos nodales para un tetraedro.

m	L1	L2	L3	L4	Pesos	Orden
1	1/4	1/4	1/4	1/4	1	2
1	0,5854	0,1382	0,1382	0,1382	1/4	3
2	0,1382	0,5854	0,1382	0,1382	1/4	
3	0,1382	0,1382	0,5854	0,1382	1/4	
4	0,1382	0,1382	0,1382	0,5854	1/4	
1	1/4	1/4	1/4	1/4	-16/20	4
2	1/3	1/6	1/6	1/6	9/20	
3	1/6	1/3	1/6	1/6	9/20	
4	1/6	1/6	1/3	1/6	9/20	
5	1/6	1/6	1/6	1/3	9/20	

El procedimiento para la evaluación de las integrales del elemento para hexaedros es casi idéntico al procedimiento utilizado para dos cuadriláteros tridimensionales. La matriz jacobiana se convierte ahora en una matriz de 3x3, a diferencia de la anterior que era de 2x2 para el cuadrilátero. La forma general de la integración esta dada por

$$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(r, s, t) |J| dr ds dt = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n w_i w_j w_k f(r_i, s_j, t_k) |J(r_i, s_j, t_k)| \quad (8.2)$$

w_i, w_j, w_k son los factores de peso asociada con los puntos de Gauss respectivos. Obsérvese que la ecuación. 8.2 se obtiene combinando una fórmula tridimensional establecida originalmente para el único elemento tridimensional. Los factores de ponderación y el punto de integración de las coordenadas se muestran en la tabla5.

Tabla5: Factores de ponderación y puntos de integración.

Puntos de integracion	i(j)(k)	ri(sj)(tk)	wi(wj)(wk)
1	1	0	2
2	3	0,5774	1
	2	-0,5774	1
3	1	0,0000	8/9
	2	0,7746	5/9
	3	-0,7746	5/9
4	1	0,8611	0,3479
	2	-0,8611	0,3479
	3	0,3400	0,6521
	4	-0,3400	0,6521

A continuación se presenta el código en Matlab para la construcción de la integración numérica de un hexaedro de 8 nodos.

```

%-----
pg = [sqrt(1/3); -sqrt(1/3)]; %pesos de gauss

%%Calcular k de del elemento
material_element = elements(element,4);
thickness_element = 1;
k = zeros(24,24); %matris k de un elemento
for ri = 1:2
    r = pg(ri);
    for si = 1:2
        s = pg(si);
        for ti = 1:2
            t = pg(ti);
            %%Calcular B del elemento
            [B detJ1] = calcular_B(r,s,t,x,y);
            k = k + B'*D{material_element}*B*detJ1*thickness_element;
            B_T{element} = B_T{element} + B; % Matrices B de cada elemento
        end
    end
end
end
%-----

```

9. Ensamble de la matriz de rigidez (K)

Para construir la matriz de rigidez (K) es necesario construir la k de cada elemento ya que al ensamblar cada una de estas matrices obtenemos la matriz de rigidez global.

La matriz k para cada elemento esta dada por:

$$k = \int_{\Omega} B_i' D B_i d\Omega \quad (9.1)$$

Donde B es una matriz que depende de cada tipo de elemento ya que esta definida por las derivadas parciales de las funciones de forma con respecto a x y y (las funciones de forma cambian para cada tipo de elemento como se presento anteriormente), la matriz B queda definida por:

$$B_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \quad (9.2)$$

Y D es la matriz de constantes elásticas definida a partir del modulo de Young (E) y el coeficiente de Poisson (ν).

Matriz D para elementos planos:

$$D = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 0.5(1-\nu) \end{bmatrix} \quad (9.3)$$

Matriz D para elementos de volumen:

$$D = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \quad (9.4)$$

Para explicar el ensamblaje de cada k por elemento utilizaremos el siguiente ejemplo

Ejemplo 5: A partir de la Fig.20 construir la matriz de rigidez global K

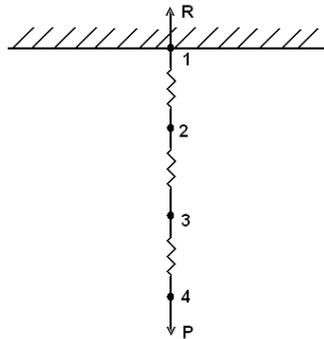


Fig.20: Ejemplo 5 donde R indica una restricción y P un peso

Para el primer elemento que tiene los nodos 1, 2 construimos la matriz k multiplicamos por el vector de desplazamientos y obtenemos el vector de restricciones como se presenta en la ecuación (9.6).

$$k_1 = \frac{EA}{l_1} \quad (9.5)$$

$$\begin{bmatrix} k_{1(1,1)} & k_{1(1,2)} \\ k_{1(2,1)} & k_{1(2,2)} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} \quad (9.6)$$

Para el segundo elemento compuesto por los nodos 2 y 3 tenemos:

$$k_2 = \frac{EA}{l_2} \quad (9.7)$$

$$\begin{bmatrix} k_{2(1,1)} & k_{2(1,2)} \\ k_{2(2,1)} & k_{2(2,2)} \end{bmatrix} \cdot \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (9.8)$$

Para el tercer elemento compuesto por los nodos 3 y 4 tenemos:

$$k_3 = \frac{EA}{l_3} \quad (9.9)$$

$$\begin{bmatrix} k_{3(1,1)} & k_{3(1,2)} \\ k_{3(2,1)} & k_{3(2,2)} \end{bmatrix} \cdot \begin{bmatrix} u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0 \\ P \end{bmatrix} \quad (9.10)$$

Después de tener cada una de las k por cada elemento comenzamos a ensamblar la matriz de rigidez global para esto construimos una matriz cuadrada del numero total de desplazamientos, para este caso es una matriz de 4x4; la forma como se ensambla es colocando la matriz k de tal forma que al multiplicar por el vector de desplazamientos obtengamos el mismo sistema de ecuaciones, como se muestra a continuación

$$\begin{bmatrix} k_{1(1,1)} & k_{1(1,2)} & 0 & 0 \\ k_{1(2,1)} & k_{1(2,2)} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} R \\ 0 \\ 0 \\ P \end{bmatrix} \quad (9.11)$$

Esto se realiza para cada una de las matrices k , teniendo en cuenta que si hay algo en la posición que le corresponde se suma a lo que había, como se muestra en la ecuación (9.12)

$$\begin{bmatrix} k_{1(1,1)} & k_{1(1,2)} & 0 & 0 \\ k_{1(2,1)} & k_{1(2,2)} + k_{2(1,1)} & k_{2(1,2)} & 0 \\ 0 & k_{2(2,1)} & k_{2(2,2)} + k_{3(1,1)} & k_{3(1,2)} \\ 0 & 0 & k_{3(2,1)} & k_{3(2,2)} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} R \\ 0 \\ 0 \\ P \end{bmatrix} \quad (9.12)$$

Otra forma de construir la matriz de rigidez global de manera más simple para trabajar con muchos elementos será explicada a partir del ejemplo 6

Ejemplo 6: A partir de la Fig.21 construir la matriz de rigidez global K y encontrar la solución para $KU = F$, donde U es el vector de deformaciones y F es el vector de fuerzas.

Lo primero que vamos a realizar es una tabla que esta compuesta por el nodo, la posición del nodo y las restricciones en x , y que este posee.

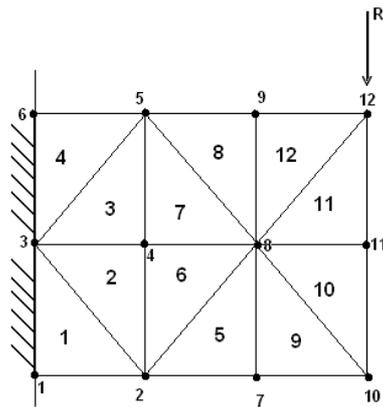


Fig.21: Viga con 12 elementos triangulares

Tabla6: Presenta la posición y las restricciones que tiene cada nodo.

# de nodos	Posición		Restricciones	
1	0	0	1	1
2	1	0	0	0
3	0	1	1	1
4	1	1	0	0
5	1	2	0	0
6	0	2	1	1
7	2	0	0	0
8	2	1	0	0
9	2	2	0	0
10	3	0	0	0
11	3	1	0	0
12	3	2	0	0

Después de esto se construye una tabla que nos permita saber que nodos pertenecen a un mismo elemento, además de esto construimos una tabla que nos permita identificar donde están las fuerzas a las que es sometida la pieza.

Tabla7: Nodos por elemento.

Elemento	nodo i	nodo j	noda k
1	1	2	3
2	2	4	3
3	3	4	5
4	3	5	6
5	2	7	8
6	2	8	4
7	4	8	5
8	8	9	5
9	7	10	8
10	10	11	8
11	8	11	12
12	8	12	9

Tabla8: Fuerzas por elemento

# de nodos	Fuerza x	Fuerza y
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	10000

Para determinar la magnitud de la matriz de rigidez global es necesario construir una tabla a partir de las restricciones que se mostraron en la Tabla6. Lo que se hace es cambiar por los unos de las restricciones por ceros y los ceros de las restricciones cambiarlos por el número de ceros que tenga en la tabla de las restricciones hasta esa posición, como se muestra a continuación:

Tabla9: Matriz ID

Restricciones		Matriz ID	
1	1	0	0
0	0	1	2
1	1	0	0
0	0	3	4
0	0	5	6
1	1	0	0
0	0	7	8
0	0	9	10
0	0	11	12
0	0	13	14
0	0	15	16
0	0	17	18

El numero de ceros de la tabla de restricciones es 18 son los grados de libertad y por esto la matriz de rigidez global es de 18x18.

Construimos ahora un vector LM que consiste en tomar la información perteneciente al elemento de la matriz ID, si tomamos el primer elemento que tiene los nodos 1, 2 y 3 entonces tomamos los valores de la tabla9 en las filas 1,2 y 3, entonces el vector LM queda definido como:

$$LM = [0 \ 0 \ 1 \ 2 \ 0 \ 0] \quad (9.13)$$

Después de esto se multiplica cada casilla por el vector LM y las coordenadas de los productos que den diferentes de cero son las coordenadas del valor de la matriz k del elemento que tiene que ser ensamblado en las posiciones del vector LM en la matriz de rigidez global, para el caso del primer elemento al multiplicar la casilla tres del

vector LM por la casilla tres LM da diferente de cero, lo que indica que el valor en la posición (3,3) de la matriz k del elemento es el valor que corresponde a la posición (1,1) en la matriz de rigidez global.

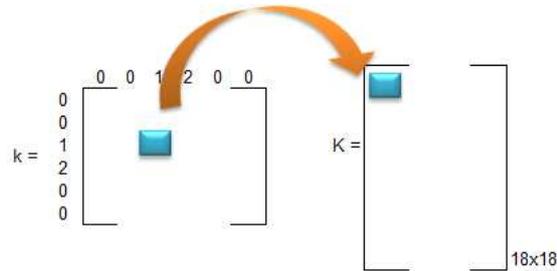


Fig.22: Ensamble de de la matriz global K

Cuando dos valores tienen que ocupar la misma posición en la matriz de rigidez global K , lo que se hace es sumar estos elementos.

A continuación se presenta el código en Matlab para el ensamble de la matriz de la matriz de rigidez global K (para un elemento hexaedro)

NOTA: para los otros elementos solo es necesario cambiar el número de nodos.

```

%-----
% Calculo de ID y de grados de libertad libres
function [ID n_f dof] = calcular_ID(constraints)
    [n_elements n_dof] = size(constraints);
    ID = zeros(n_elements,n_dof);
    n_f dof = 0;
    for i = 1:n_elements
        for j = 1:n_dof
            if not(constraints(i,j))
                n_f dof = n_f dof + 1;
                ID(i,j) = n_f dof;
            end
        end
    end
end
end
end

%%-----

%%Ensamblaje de K
LM_element = [ ID(node1,:) ID(node2,:) ID(node3,:) ID(node4,:) ID(node5,:)
               ID(node6,:) ID(node7,:) ID(node8,:) ];
assemble_K(k_element,LM_element);

%%-----

%%Nested functions:

%Ensamblaje de K
function assemble_K(k,LM)
    %Guarda en m el número de columnas de la matriz k
    m = length(k);
    %Guarda el número de elementos del vector K_VAL (que es igual que K_COL y
    K_ROW)
    LENGTH_K = length(K_VAL);
    for i=1:m

```

```

for j=1:m
    if (LM(i)*LM(j))>0 % Si el nodo no está restringido
        CONTADOR = CONTADOR+1;
        %Este bloque duplica la longitud de los vectores cuando
        ya se han llenado los espacios que estaban predefinidos
        if CONTADOR>LENGTH_K
            % Duplica la longitud del vector K_ROW llenando lo
            nuevo con ceros
            K_ROW(2*CONTADOR)= 0;
            % Duplica la longitud del vector K_COL llenando lo
            nuevo con ceros
            K_COL(2*CONTADOR)= 0;
            % Duplica la longitud del vector K_VAL llenando lo
            nuevo con ceros
            K_VAL(2*CONTADOR)= 0;
            LENGTH_K = 2*CONTADOR;
        End
        %Fila de la K global donde ira el valor k(i,j)
        K_ROW(CONTADOR) = LM(i);
        %Columna de la K global donde ira el valor k(i,j)
        K_COL(CONTADOR) = LM(j);
        % Valor en la K global
        K_VAL(CONTADOR) = k(i,j);
    end
end
end
end
end

%% -----
%% Creacion de K dispersa y solucion del sistema K*U=F
K = sparse(K_ROW(1:CONTADOR),K_COL(1:CONTADOR),K_VAL(1:CONTADOR)

%%Liberar Memoria
clear K_ROW K_COL K_VAL CONTADOR

%%Resolver sistema
U = K\applied_forces; % esto es una funcion de matlab
R = K*U - applied_forces;
F = R + applied_forces;
disp(F)
%-----

```

Para encontrar los esfuerzos principales es necesario resolver la siguiente ecuación:

$$[\sigma] = [D] \cdot [B] \cdot [u] \quad (9.14)$$

A continuación se presenta el código en Matlab para el encontrar los esfuerzos principales

```

%-----
%% Calculo de esfuerzos principales

%Desplazamientos por nodo
%Se guarda el desplazamiento del nodo respectivo encontrado en U.
displacements_by_node = zeros(n_nodes,n_dims);
for node=1:n_nodes
    for dim=1:n_dims
        if ID(node,dim)
            displacements_by_node(node,dim)=U(ID(node,dim));
        end
    end
end

```

```

end
end

%-----

S = zeros(6,n_nodes);
node_count = zeros(n_nodes,1);

%%Tensiones por elemento
s = zeros(6,1);
for element = 1:n_elements
    % Se toman los números de los nodos que están presentes en el elemento
    nodes = elementsTrian(element,1:Tr);
    % Se toman los desplazamientos de los nodos del elemento
    a = displacements_by_node(nodes,:);
    % Se hace un vector vertical con los desplazamientos de los nodos
    a = a(:);
    s(1:3,1) = D{elementsTrian(element,end)} * B{element} * a;
    %matriz combined stressed Los eigenvalores se devuelven de menor a mayor
    sp = eig([[s(1) s(3)];[s(3) s(2)]]);
    vm = sqrt(0.5*((s(1))^2+(s(3))^2+(s(3)-s(1))^2));
    s(4:6,1) = [[sp(2) sp(1)]]; vm ];

    %suavizado promedio de los esfuerzos por nodo
    node_count(nodes) = node_count(nodes) + 1;
    for n1 = 1 : Tr
        S(:,nodes(n1)) = (S(:,nodes(n1))*(node_count(nodes(n1))-1)
            + s)/node_count(nodes(n1));
    end
end
end
%-----

```

Metodología

En la primera etapa del proyecto se construyo una función en GID que permite por medio de ficheros almacenar la información de la maya generada por GID, en el primer fichero se encontrara la cantidad de nodos que posee la maya, cuantos elementos son triangulares, cuantos cuadriláteros, cuantos tetraedros y por ultimo el tipo de material; en el segundo fichero se encontrara las restricciones y los momentos para cada nodo en x, y, z; en los ficheros 3, 4 y 5 se presenta el numero de elemento, números de nodos de este elemento y los nodos i, j, k, l, m, n, o, p, según sea el caso; en el fichero sexto se encuentra las coordenadas de los nodos; en el séptimo numero de materiales y por ultimo en el fichero octavo se encuentra las fuerzas y los momentos en x, y ,z.

Después de esto se construyo una plataforma en Matlab que permite leer y almacenar la información que se encuentra en cada uno de los ficheros.

Se procedió entonces a realizar la implementación en Matlab de los siguientes elementos

- Elemento triangular de 3 nodos
- Elemento triangular de 6 nodos
- Elemento cuadrático de 4 nodos
- Elemento cuadrático de 9 nodos
- Elemento tetraedro de 4 nodos
- Elemento hexaédrico de 8 nodos

A partir de esto se construyo una función en Matlab que permite ensamblar la matriz de rigidez global de una manera mas eficiente (utilizando matrices dispersas), para así poder encontrar la solución a nuestro problema ($KU = F$).

Trabajo futuro

Después de determinar numéricamente cuales son las tenciones y deformaciones a la que la pieza es sometida se requiere realizar una interfaz entre Matlab y GID que permita realizar la apreciación de los resultados de forma visual.

Después que el software este refinado realizarle algunas pruebas y lograr con esto optimizar el tiempo de computo y o la memoria requerida en su ejecución.

Conclusiones

- Se exploró los sistemas de almacenamiento de matrices dispersas optimizados en Matlab.
- Se estudiaron varios algoritmos FEA y se programaron los elementos planos y de volumen.
- Se implementó una función que lee la información y determina con esta que tipo de elementos requiere utilizar.
- Se implementó un software inicial que permite calcular problemas hasta de 75000 grados de libertad.
- Se probó que los resultados obtenidos coinciden con la solución analítica de un problema sencillo.
- Se debe mejorar el manejo de memoria del programa.
- Se logró unificar una estructura en el programa que permite realizar mejoras y correcciones de una forma fácil, para las personas que intervienen en el proyecto.
- Se desarrolló una aplicación real y se resolvió con el software.

Referencias

Pepper, Darrell y Heinrich, Juan. (1992), *The Finite Element Method Basic Concepts and Applications*. United States of America: Hemisphere Publishing Corporation.

Bathe, Klaus Jürgen. (1982), *Finite Element Procedures*. New Jersey: Prentice Hall.

Zienkiewicz, O. C. y Leroy, Robert. (2000), *The Finite Element Method: Solid mechanics*, Barcelona, España: Printed and bound by MPG Books Ltd.

Fornóns, José-María. (1982), *El método de los elementos finitos en la ingeniería de estructuras*. España. Talleres gráficos.

P. G. Ciarlet (1978): *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam.

P. G. Ciarlet (1991): *Basic error estimates for elliptic problems*, en Handbook of Numerical Analysis (Vol II) J.L. Lions y P. G. Ciarlet (ed.), North-Holland, Amsterdam.

Paginas web

- <http://gid.cimne.upc.es/index.html> (manual)
- www.mathworks.com