

*Universidad de Matanzas “Camilo Cienfuegos”.*  
*Facultad de Ingenierías Química y Mecánica.*



**MODELACIÓN MATEMÁTICA DEL ENDURECIMIENTO POR  
DEFORMACIÓN PLÁSTICA SUPERFICIAL DEL PROCESO  
DE BRUÑIDO POR RODILLO SIMPLE.**

Tesis Presentada en Opción al Título Académico de Máster en  
Ingeniería Asistida por Computadora.

**Autor: Lic. Yoslandy Lazo Robaina.**

**Tutor: Dr.C. Eleno Alfonso Brindis.**

Matanzas, 2010.

## **RESUMEN.**

Este trabajo realiza el cálculo de las deformaciones que se producen en el proceso de endurecimiento por deformación plástica superficial, en la operación de bruñido por rodillo simple. La formulación matemática se desarrolló a partir de las expresiones y leyes que rigen la mecánica del medio continuo y la geometría de los ensayos para la modelación se discretizó por elementos finitos isoparamétricos. En la tesis la modelación matemática e implementación de los códigos de MATLAB se realizaron mediante un algoritmo de solución elaborado por el autor. Los resultados obtenidos para las deformaciones están en correspondencia con la interpretación y análisis del fenómeno físico, validando la aplicación del modelo matemático empleado y el algoritmo propuesto.

## **ABSTRACT.**

The aim of this work, consists on calculate the deformations that occur in the process of case strain hardening, in the operation of simple roller burnishing. The mathematical formulation was developed by means of governing continuum mechanics expressions and laws. In this sense, tests geometry for modeling is discretized by isoparametric finite elements. On the other hand, mathematical modeling and codes implementation were performed using a solution algorithm developed by the author. Deformations results are in concordance with the interpretation and analysis of physical phenomenon. This fact, allow us, the validation and application of the mathematical model used and the proposed algorithm.

## TABLA DE CONTENIDO.

RESUMEN.....	i
ABSTRACT. ....	ii
TABLA DE CONTENIDO. ....	iii
INTRODUCCIÓN.....	1
CAPÍTULO 1. Fundamentación Teórica. ....	6
1.1 Introducción a la Teoría del Medio Continuo.....	6
1.2 Definición del medio continuo. ....	7
1.3 Tensores y Mecánica del Medio Continuo.....	7
1.4 Notación indicial .....	9
1.4.1 Convenio de sumación de Einstein. ....	10
1.4.2 Índice mudo e índice libre. ....	11
1.5 Cinemática de Deformación y Movimiento.....	11
1.5.1 Partículas. Configuraciones. Deformación y Movimiento. ....	12
1.5.2 Descripción material. ....	13
1.5.3 Descripción espacial.....	15
1.5.4 Campo de desplazamientos. ....	17
1.6 Tensor Gradiente Deformación. ....	18
1.6.1 Tensor de deformación infinitesimal. ....	20
1.6.2 Interpretación física del tensor deformación infinitesimal.....	21
1.7 Discretización del continuo.....	22
1.7.1 Generación de mallas. ....	24
1.7.2 Formulación isoparamétrica. ....	25

1.7.3 Concepto isoparamétrico. ....	27
1.7.4 Elementos isoparamétricos en la discretización del continuo. ....	29
1.8 Integración Numérica. ....	31
1.8.1 Funciones de Interpolación de LAGRANGE. ....	31
1.8.1.1 Polinomio de interpolación lineal de Lagrange.....	32
1.8.1.2 Polinomio de interpolación de Lagrange de grado n.....	33
1.8.2 Aplicación de los polinomios de interpolación de Lagrange en los elementos finitos.....	34
1.9 Conclusiones parciales del capítulo. ....	35
CAPÍTULO 2. Descripción del Fenómeno Físico. ....	36
2.1 Descripción de la esencia del fenómeno físico. ....	36
2.2 Teoría de la deformación.....	37
2.3 Mecanismos por deformación. ....	38
2.3.1 Endurecimiento por deformación.....	39
2.3.2 Endurecimiento por deformación en frío. ....	39
2.4 Deformación plástica superficial.....	40
2.5 Bruñido por rodillo simple.....	41
2.6 Descripción del experimento a modelar.....	42
2.7 Algoritmo propuesto para la modelación matemática. ....	44
2.8 Conclusiones parciales del capítulo. ....	45
CAPÍTULO 3. Modelación Matemática. ....	46
3.1 Entrada de datos.....	46
3.2 Toma de decisiones. ....	47

3.3	Definición de la geometría y discretización en elementos.....	49
3.3.1	Modelación Geométrica. ....	49
3.3.2	Discretización en elementos. (Generación del mallado).....	50
3.4	Definición de la funciones de interpolación.....	52
3.5	Obtención de la Matriz Jacobiana.....	54
3.6	Cálculo del Tensor Deformación. ....	56
3.7	Interpretación de las deformaciones por cada elemento finito.....	57
3.8	Metodología de pasos para el cálculo de las deformaciones .....	57
3.9	Modelación matemática del bruñido por rodillo simple.....	59
3.9.1	Entrada de los datos generales para la modelación. ....	59
3.9.2	Generación de la matriz de puntos (nodos) en coordenadas cilíndricas para las configuraciones de referencia y deformada .....	59
3.9.3	Generación de la matriz de elementos para las configuraciones de referencia y deformada .....	60
3.9.4	Representación del mallado hexaédrico para las configuraciones de referencia y deformada .....	61
3.9.5	Funciones de interpolación .....	62
3.9.6	Transformación de las coordenadas cilíndricas a coordenadas rectangulares (globales) de las configuraciones de referencia y deformada.....	62
3.9.7	Transformación de las coordenadas rectangulares (globales) a coordenadas naturales (locales) de las configuraciones de referencia y deformada.....	65

3.9.8	Generación de la matriz de derivadas de las funciones de interpolación y evaluación de esta para el cálculo de la matriz Jacobiana de cada elemento finito.....	66
3.9.9	Obtención de la matriz Jacobiana para cada elemento finito de la geometría discretizada de las configuraciones de referencia y deformada.....	68
3.9.10	Cálculo del Tensor de Deformación.....	69
3.10	Interpretación de los resultados obtenidos en el tensor deformación .....	70
3.11	Conclusiones parciales del capítulo .....	71
	Conclusiones.....	72
	Recomendaciones.....	73
	Bibliografía.....	74
	Anexos .....	87

## **INTRODUCCIÓN.**

El rápido progreso de la Tecnología ha motivado la aparición de nuevos materiales de tipos estructurales y complejos que el ingeniero ha debido estudiar para hacerlos seguros y económicos. Es decir, se ha hecho necesario buscar métodos racionales para determinar su resistencia y deformación. La disciplina que tradicionalmente se ha ocupado de estos temas es la Resistencia de Materiales, que gracias a las simplificaciones que se introducen debido a las peculiaridades geométricas del sólido estudiado, proporcionan al investigador procedimientos para resolver los problemas con suficiente aproximación. En otros casos, sin embargo, la complejidad del problema es superior, las simplificaciones ocurridas en la resistencia de materiales ya no son válidas, y es necesario utilizar la teoría más general de la Mecánica del Medio Continuo. Con esta ciencia se puede estudiar el comportamiento de cualquier medio sometido a cualquier sistema de cargas exteriores [López, J. J.; 1999].

Las ventajas de este enfoque han sido apreciadas por ingenieros y científicos por 200 años y han servido para modelar exitosamente la respuesta de materiales de ingeniería. Las ecuaciones constitutivas del material se establecen en general a nivel diferencial, y se relacionan las componentes de esfuerzo-deformación.

El éxito de la Mecánica del Medio Continuo se ha visto potenciado en las últimas décadas por la aparición de los métodos numéricos, que han

generado algoritmos y procedimientos que conducen a la solución aproximada de problemas mediante un número finito de pasos que pueden ejecutarse de manera lógica.

La aplicación de estos métodos numéricos basados en la mecánica del medio continuo consiste en discretizar la geometría del cuerpo a analizar desechando las imperfecciones de la estructura atómica de los materiales a pequeña escala. Dentro de los métodos de discretización más empleados en la solución de problemas ingenieriles en los últimos años se encuentra el de dividir la geometría en elementos finitos isoparamétricos (elementos curvos) [Ma y Liu, 2006; Long et al. 2009], los cuales permiten obtener una solución numérica aproximada sobre un cuerpo, estructura o dominio (medio continuo) dividiéndolo en un número elevado de subdominios no intersectantes entre sí denominados «*elementos finitos*».

La discretización en elementos finitos ayuda a construir un algoritmo de proyección sencillo, logrando además que la solución sea generalmente exacta en un conjunto finito de puntos.

El análisis numérico en sentido general proporciona todo el andamiaje necesario para llevar a cabo todos los procedimientos matemáticos susceptibles de expresarse algorítmicamente, basándose en algoritmos que permiten la simulación o cálculo de un proceso o fenómeno representado de forma más sencilla, que admite analizar sus características.

En ingeniería de materiales, los procesos de endurecimiento por deformación desempeñan un importante papel, en ellos se aprovecha las características estructurales (atómicas) de los materiales (fundamentalmente metales) mejorando las propiedades físico-mecánicas de estos, aumentando la fiabilidad y longevidad en la asignación de servicios de los mecanismos y piezas elaborados.

La modelación matemática de estos procesos permite predecir el comportamiento y límite de resistencia a la rotura de los materiales empleados en los diseños de maquinas y mecanismos de la industria.

La mayoría de los mecanismos de endurecimiento por deformación se emplean en la industria desde hace algunas décadas, sin embargo los métodos tradicionales de cálculo basados en criterios simplificados de la resistencia de materiales no son aplicables en la búsqueda de resultados apropiados en el análisis de situaciones complicadas, como son las deformaciones infinitesimales que se producen en los procesos de endurecimiento por deformación plástica.

En las últimas dos décadas a partir del desarrollo de los métodos numéricos y en especial el método de elementos finitos se han realizado trabajos [Eshwara, et al. 2006a; Eshwara, et al. 2006b] que simulan el proceso de deformación plástica superficial por rodillo en software profesionales, sin embargo estos trabajos no ofrecen un algoritmo de solución detallado, además de que estos software tienen limitaciones

para resolver problemas de esta complejidad y resulta complicado poder establecer las restricciones que se ajusten al fenómeno físico.

Ante esta situación, se define como **problema científico**: ¿Cómo diseñar un algoritmo para calcular las deformaciones infinitesimales que se producen en el proceso de endurecimiento por deformación plástica superficial (rodillado)?

Para darle solución al problema planteado se ha formulado la siguiente **hipótesis**: Las deformaciones que ocurren en el proceso de endurecimiento por deformación plástica superficial (rodillado) es posible calcularlas empleando la discretización de la geometría en elementos finitos isoparamétricos y las expresiones y leyes que rigen la mecánica del medio continuo.

De ahí que se plantee para este trabajo el siguiente **objetivo**: Diseñar un algoritmo de solución para calcular las deformaciones infinitesimales que se producen en el proceso de endurecimiento por deformación plástica superficial (rodillado) cumpliendo con las leyes que rigen la mecánica del medio continuo.

Se trazan como tareas necesarias para el cumplimiento de este objetivo, las siguientes:

- Revisión de la bibliografía para la fundamentación teórica de la investigación y desarrollo de la modelación matemática.

- Implementar códigos para la modelación matemática del endurecimiento por deformación plástica superficial (rodillado) en el software profesional MATLAB.
- Realizar el cálculo de las deformaciones infinitesimales que ocurren en el endurecimiento por deformación plástica superficial (rodillado) mediante el algoritmo y códigos (.m) realizados para la modelación matemática.
- Interpretación y análisis de los resultados obtenidos de la modelación matemática del endurecimiento por deformación plástica superficial (rodillado).

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.**

El objetivo de este capítulo es realizar una revisión detallada y crítica de los antecedentes y bases teóricas de la mecánica del medio continuo publicadas en la literatura científica sobre su aplicación a fenómenos físicos, de forma tal que sirva de marco teórico-referencial al resto de la investigación.

### **1.1 Introducción a la Teoría del Medio Continuo.**

La materia esta formada por moléculas, que a su vez consisten de átomos y partículas subatómicas. Por tanto, la materia no es continua. Sin embargo, son muchos los aspectos que a diario se experimentan con respecto al comportamiento de materiales, tales como la flexión de una estructura bajo cargas [Aminanda, et al., 2009 y Hu, et al., 2009], la proporción de descargue de agua en un tubo bajo una presión por gravedad [Vashisht, 2008], o la fuerza de rozamiento por el movimiento de un cuerpo en el aire, que puede ser descrito y predicho con teorías que no asumen la estructura molecular de los materiales. La teoría que apunta a describir la relación entre grandes fenómenos, desechando la estructura atómica del material a pequeña escala, es conocida como: *teoría del continuo*. La teoría del continuo considera la materia como divisible infinitamente [Ingber, 2008; Belevich, 2008]. Así que, dentro de esta teoría, se acepta la idea de un volumen infinitesimal de material, referido como una partícula en el continuo, y en cada vecindario de una

partícula esta tendrá siempre partículas vecinas. Por tanto, un medio continuo se concibe como una porción de materia formada por un conjunto infinito de partículas (que forman parte, de un sólido, de un fluido o de un gas) que va a ser estudiado macroscópicamente.

### **1.2 Definición del medio continuo.**

Una definición del medio continuo es desarrollada por [Oliver y Agelet, 2002], la cual plantea:

“Se entiende por *Medio Continuo* un conjunto infinito de partículas (que forman parte, de un sólido, de un fluido o de un gas) que va a ser estudiado macroscópicamente, es decir, sin considerar las posibles discontinuidades existentes en el nivel microscópico (nivel atómico o molecular). En consecuencia, se admite que no hay discontinuidades entre las partículas y que la descripción matemática de este medio y de sus propiedades se puede realizar mediante funciones continuas”.

### **1.3 Tensores y Mecánica del Medio Continuo.**

Algunos autores plantean que la mecánica está compuesta por la mecánica de partículas y la del continuo, donde estas han jugado un rol significativo en el desarrollo de la ingeniería moderna. [Bai, et al., 2008]. La mecánica del continuo trata con cantidades físicas que son independientes de cualquier sistema de coordenadas particular que pueda ser usado para describirlas. Al mismo tiempo, estas cantidades físicas son muy a menudo y convenientemente especificadas para

referirlas a un sistema de coordenadas apropiado. Matemáticamente, tales cantidades físicas son representadas por *tensores* [Mase, et al, 2009].

Como entidad matemática, un tensor tiene una existencia independiente de cualquier sistema de coordenadas. Mayormente son especificados en un sistema coordenadas particulares para ciertos grupos de cantidades, conocidas como sus *componentes*. Especificando las componentes de un tensor en un sistema de coordenadas, determinas las componentes en otro sistema cualquiera [Mase, et al, 2009].

El concepto de tensor fue originalmente presentado como un operador en álgebra multi-linear y ahora es una herramienta utilizada en diversos campos [Ni y Wang, 2007], tales como, tratamiento de señales y de imágenes, análisis de datos, y mecánica del continuo no lineal [Ni, et al., 2007].

Autores como [Qi, 2007] clasifican a los tensores según su orden como:

- **Escalar** (*Tensor de orden 0*): Cantidad que tiene magnitud pero no dirección (ejemplo: densidad, temperatura, presión)[Chen, et al., 2008]. Los escalares pueden ser funciones del espacio y del tiempo y no necesariamente han de ser constantes.
- **Vector** (*Tensor de orden 1*): Cantidad que tiene magnitud y dirección (ejemplo: velocidad, fuerza, momento, campo eléctrico)[Kapuscik y Lanczewski, 2009; Lee, et al., 2007].

- **Tensor de segundo orden** (*Tensor de orden 2*): Cantidad que tiene magnitud y dos direcciones (ejemplo: tensión, deformación)[ Tua, et al., 2008; Nguyen y Caron, 2009].

Nótese que todas estas propiedades o magnitudes físicas dependen de cada punto del medio continuo y del instante en que las estemos midiendo (*intervalo de tiempo*), estas se conocen con el nombre genérico de campos. Cada una de estas magnitudes es función  $f(X;t)$ , donde  $X \in R^3$  y toma valores en una cierta región del espacio físico tridimensional  $R$  y durante un intervalo de tiempo  $t \in (t_1; t_2)$  arbitrario.

Como las distintas magnitudes escalares, vectoriales, tensoriales, que en general son funciones de punto y del tiempo, con las que se realizan las operaciones normales de cálculo y se derivan respecto a las variables espaciales y del tiempo, se adopta para simplificar las expresiones matemáticas la siguiente notación.

#### 1.4 Notación indicial.

La notación indicial emplea coordenadas  $x_1$ ,  $x_2$ , y  $x_3$  para denotar las clásicas coordenadas  $x$ ,  $y$ ,  $z$ , respectivamente [Drake y Dogancay, 2008]. Los componentes de un vector  $v$  suponen  $v_1$ ,  $v_2$ ,  $v_3$  (en tres dimensiones), en lugar de las convencionales  $u$ ,  $v$ ,  $w$ . Por lo que elementos matriciales son concernidos, en notación indicial, tal como  $A_{23}$  para identificar el elemento en la segunda fila y tercera columna, que se emplea hace algún tiempo. La ventaja de la notación indicial, en conjunción con el

convenio de sumación, es que se logran escribir expresiones matemáticas muy grandes de forma compacta [Nair, 2009].

Considerando un sistema de ecuaciones  $M$ , en  $N$  desconocidas:

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + \dots + A_{1N}x_N &= c_1, \\ A_{21}x_1 + A_{22}x_2 + \dots + A_{2N}x_N &= c_2, \\ \dots\dots\dots &=, \dots, \\ A_{M1}x_1 + A_{M2}x_2 + \dots + A_{MN}x_N &= c_M, \end{aligned} \tag{1.1}$$

donde este sistema de ecuaciones también puede ser escrito:

$$\sum_{j=1}^N A_{ij} x_j = c_i \quad (i = 1, 2, \dots, M; j = 1, 2, \dots, N), \tag{1.2}$$

De acuerdo con el convenio de sumación de Einstein [Chapelle y Paris, 2008; Mao, et al., 2008 ] la expresión se puede simplificar más y escribir así:

$$A_{ij} x_j = c_i \quad (i = 1, 2, \dots, M; j = 1, 2, \dots, N), \tag{1.3}$$

### 1.4.1 Convenio de sumación de Einstein.

La presencia en un monomio de un índice repetido indica una suma de términos [ Hsieh y Katul, 2009; Dumbser y Zanotti, 2009], iguales al monomio dado y en número igual al valor superior atribuido a dicho índice,

$$\sum_{i=1}^{i=n} m^i y_i = m^1 y_1 + m^2 y_2 + m^n y_n \tag{1.4}$$

y puede escribirse

$$m^i y_i \quad (i = 1, 2, \dots, n) \tag{1.5}$$

sin necesidad de escribirse el símbolo de suma, pero debe indicarse los límites del índice.

### 1.4.2 Índice mudo e índice libre.

El índice sobre el cual se aplica la suma, se denomina índice mudo [Livermore y Ierley, 2009], y por consiguiente no importa que letra se utiliza para indicarlo.

$$\frac{\partial A}{\partial x^i} \frac{\partial x^i}{\partial y^k} = \frac{\partial A}{\partial x^m} \frac{\partial x^m}{\partial y^k} = \frac{\partial A}{\partial x^j} \frac{\partial x^j}{\partial y^k} \quad (1.6)$$

el superíndice  $i, m, j$ , es el índice mudo, o sea el índice sobre el cual se realiza la suma y resulta ser el índice que no aparece en el resultado.

El índice  $k$  que aparece en los términos escritos es el que se denomina índice libre [Jiang, et al., 2009] e indica cada una de las formas de la expresión dada, mediante el valor atribuido a  $k$ . También recibe el nombre de índice de identificación.

### 1.5 Cinemática de Deformación y Movimiento.

La descripción cinemática de las deformaciones y movimientos de los cuerpos continuos, es un punto de vista que se adopta en la Mecánica de Medios Continuos para estudiar estos movimientos en su mayor generalidad posible, sin las restricciones características de teorías como la elasticidad clásica. Los conceptos de la cinemática pertenecen fundamentalmente a la geometría, y más concretamente a la geometría diferencial. La metodología empleada en mecánica del medio continuo describe de forma cuantitativa no sólo los movimientos de los sólidos deformables sino también sus deformaciones locales. Esta generalidad permite, cuando se estudia las leyes de balance y los modelos

constitutivos, plantear completamente las ecuaciones que rigen el movimiento de una gran variedad de cuerpos deformables.

### 1.5.1 Partículas. Configuraciones. Deformación y Movimiento.

En mecánica del continuo, se comienza a describir el modelo usando representaciones de cuerpos. Un cuerpo material  $B$  es definido con un grupo de elementos  $X$ , llamados partículas o punto material, donde pueden ser situadas con una correspondencia uno-a-uno con los puntos de una región regular del espacio físico. Se considera un cuerpo, para un tiempo inicial  $t_0$ , la configuración inicial  $K_0$ . El cuerpo subsecuentemente esta bajo cargas externas y son asumidas para hallar un crecimiento volumétrico [Ambrosi, et al., 2006 ].

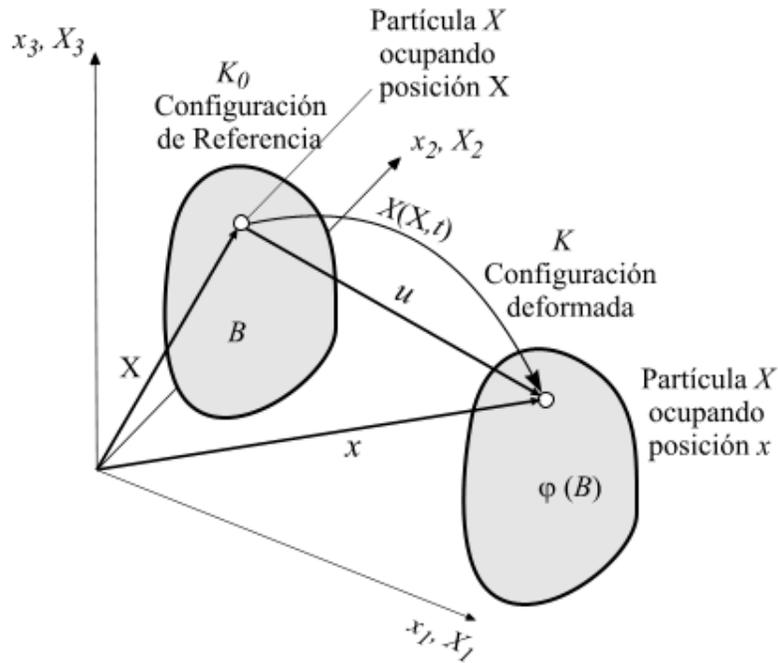
La especificación de la posición de todas las partículas de  $K_0$  con respecto a un mismo origen para algunos instantes de tiempo es dada para definir la configuración del cuerpo para ese instante [Mase, et al, 2009]. Matemáticamente, esto es expresado:

$$x = K(X) \tag{1.7}$$

en la función del vector  $K$  se designa la posición relativa de cada partícula  $X$  del cuerpo para un mismo origen. Se asume que esta expresión es singularmente invertible y diferenciable tantos tiempos como se requiera; en general, dos o tres tiempos son suficientes. La inversa de esta expresión es escrita:

$$X = K^{-1}(x) \tag{1.8}$$

y e identifica la partícula  $X$  ubicada en la posición  $x$ .



**Figura 1.1.** Configuraciones de referencia y deformada de un cuerpo.

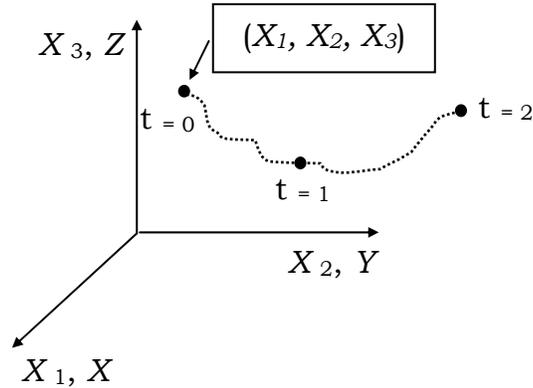
### 1.5.2 Descripción material.

En la descripción material<sup>1</sup> [Xu y Belytschko, 2008] se describe cierta propiedad ( $f$ ) mediante una función  $\bar{f}(\bullet, t): R^3 \times R^+ \rightarrow R^+$  donde el argumento( $\bullet$ ) en  $f(\bullet, t)$  son las coordenadas materiales [Khoei, et al., 2008; Oliver y Agelet, 2002]. Es decir:

$$f = \bar{f}(X, t) = \bar{f}(X_1, X_2, X_3, t) \quad (1.9)$$

<sup>1</sup> La literatura sobre el tema suele referirse también a la descripción material como *descripción Lagrangiana*.

Los tres argumentos  $X \equiv (X_1, X_2, X_3)$  de la ecuación (1.9) están siguiendo a una partícula determinada (ver **Figura 1.2**), de ahí proviene la denominación de descripción material.



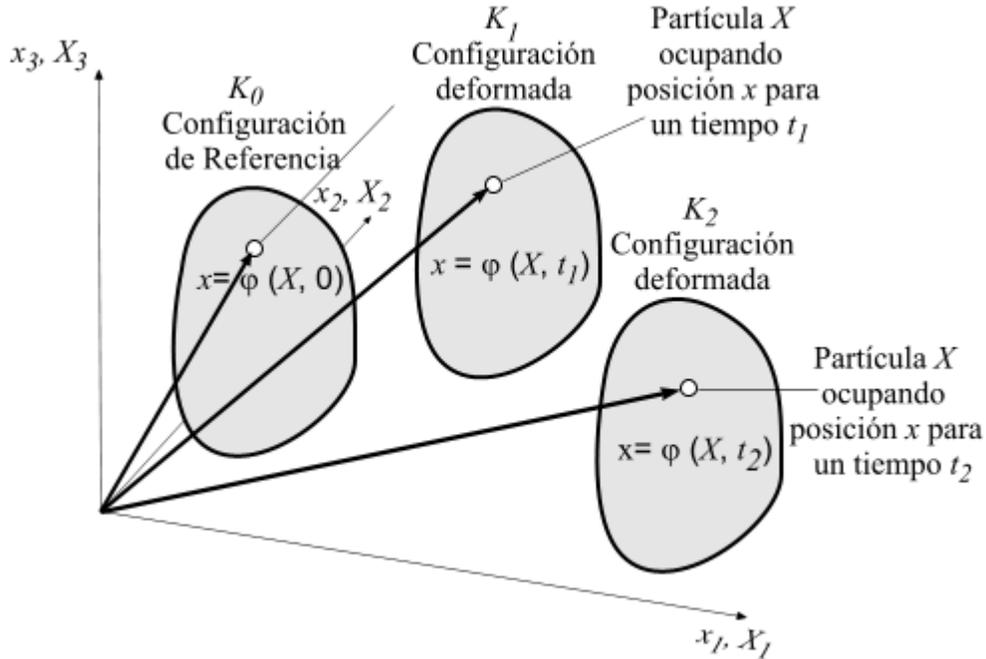
**Figura 1.2.** Descripción material

Nótese que en la descripción material, el movimiento del cuerpo es referido a la configuración de referencia  $K_0$ , que es la configuración no deformada. Así en la descripción *Lagrangiana*, las coordenadas actuales ( $X \in K$ ) son expresadas en términos de las coordenadas de referencia ( $X \in K_0$ ), y la variación de una variable típica  $\phi$  sobre el cuerpo es descrita con respecto a las coordenadas materiales  $X$  y un tiempo  $t$  [Reddy, 2008] como:

$$\phi = \phi(X, t). \quad (1.10)$$

dado un valor fijo de ( $X \in K_0$ ),  $\phi(X, t)$ , los valores de  $\phi$  para un tiempo  $t$  relacionados con el punto fijo material cuya posición en la configuración de referencia es  $X$ , se muestra en la (**Figura 1.3**). De esta manera un

cambio en tiempo  $t$  implica que la misma partícula material  $X$ , ocupando posición  $X$  en  $K_0$ , tiene un diferente valor  $\phi$ . Así son enfocadas las partículas materiales en el continuo [Reddy, 2008].



**Figura 1.3** Configuración de referencia y configuraciones deformadas para dos tiempos diferentes en coordenadas materiales.

### 1.5.3 Descripción espacial.

En la descripción espacial<sup>2</sup> [Nguyen y Abousleiman, 2010] la atención se centra en un punto del espacio. Se describe la propiedad como una función  $f(\bullet, t): R^3 \times R^+ \rightarrow R^+$  del punto del espacio y del tiempo [Oliver y Agelet, 2002]:

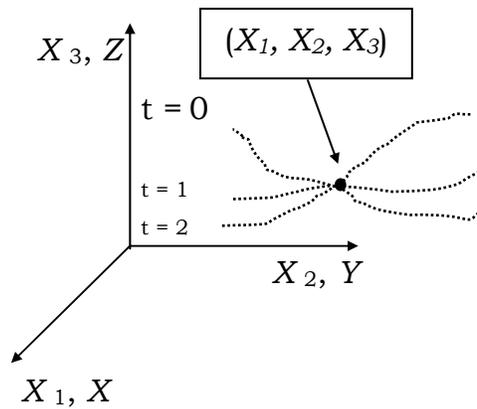
$$f = f(x, t) = f(x_1, x_2, x_3, t) \quad (1.11)$$

<sup>2</sup> La literatura sobre el tema suele referirse también a la descripción material como *descripción Euleriana*.

de tal forma que al asignar un cierto valor al argumento  $x$  en  $f = f(x,t)$  se obtiene la evolución de distintas partículas que van pasando por dicho punto del espacio a lo largo del tiempo (*ver Figura 1.4*). Por otro lado, al fijar el argumento tiempo en la ecuación se obtiene una distribución instantánea de la propiedad en el espacio. Es evidente que las ecuaciones del movimiento directas e inversas [Frey, et al., 2008] permiten pasar de una descripción a otra de la forma:

$$f(x,t) = f(x(X,t),t) = \bar{f}(X,t) \quad (1.12)$$

$$\bar{f}(X,t) = \bar{f}(X(x,t),t) = f(x,t) \quad (1.13)$$



**Figura 1.4** Descripción espacial.

En la descripción espacial, el movimiento es referido a la configuración actual  $K$  ocupada por el cuerpo  $B$ , y  $\phi$  es descrita con respecto a la posición actual ( $x \in K$ ) en el espacio, actualmente ocupado por partículas  $X$

$$\phi = \phi(x,t), \quad X = X(x,t) \quad (1.14)$$

Las coordenadas  $x$  son los términos de las coordenadas espaciales. Para un valor fijo ( $x \in K$ ),  $\phi(x, t)$ , dado los valores de  $\phi$  relacionados con un punto fijo  $x$  en el espacio, que puede ser el valor de  $\phi$  relacionado con diferentes puntos material para diferentes tiempos. De esta manera un cambio de tiempo  $t$  implica que un valor diferente  $\phi$  es observado para la misma localización espacial ( $x \in K$ ), ahora ocupada probablemente por una partícula material  $X$  diferente [Reddy, 2008].

#### **1.5.4 Campo de desplazamientos.**

Como puede observarse desde la (**Figura 1.1**), la típica partícula del cuerpo  $B$  sufre un *desplazamiento* [Civalek, et al., 2009; Areias y Matous, 2009]:

$$u = x - X \tag{1.15}$$

en la transición desde de la configuración de referencia a la configuración actual. Debido a esta relación que influye en todas las partículas, es frecuentemente usada para analizar la deformación o movimiento en términos del campo de desplazamiento de un cuerpo. El vector desplazamiento  $u$  puede ser escrito en notación indicial por la siguiente expresión:

$$u = u_i \hat{e}_i = u_A \hat{I}_A \tag{1.16}$$

Adicionalmente, considerando las descripciones material y espacial esta expresión puede ser interpretada en la forma material [Soares, et al., 2010]:

$$u(X, t) = x(X, t) - X \quad (1.17)$$

o en la forma espacial [Areias y Matous, 2008]:

$$u(x, t) = x - X(x, t) \quad (1.18)$$

En la primera de este par de ecuaciones se describe el desplazamiento que ocurre para la partícula de inicio  $X$ , y en la segunda ecuación el desplazamiento que esa partícula ha sufrido  $x$ .

### 1.6 Tensor Gradiente Deformación.

En la formulación clásica de mecánica del continuo una partícula del cuerpo ocupa la posición  $X$  en el movimiento del estado de referencia para la posición  $x(X)$  en la configuración actual. La deformación de la partícula es definida por el *Tensor Gradiente de Deformación* [Volokh, 2008 ].

El gradiente deformación es una cuestión clave en análisis de deformación, aquí  $K$  es relativo a la configuración de referencia  $K_0$ , denotada por  $F_K$ , que dada la relación de una línea material  $dX$  antes de la deformación para la línea  $dx$  (compuestas del mismo material  $dX$ ) después de la deformación (*ver Figura 1.5*). Este se define como<sup>3</sup> [Aghaei, et al., 2009; Kumar y Kumar, 2010]:

$$dx = F \cdot dX = dX \cdot F^T \quad (1.19)$$

$$F = \left( \frac{\partial}{\partial X} \right)^T = \left( \frac{\partial x}{\partial X} \right)^T \equiv (\nabla_0 x)^T \quad (1.20)$$

---

<sup>3</sup> Con el interés de abreviar, el subíndice  $K$  en  $F$  se omite.

y  $\nabla_0$  es el operador gradiente con respecto a  $X$ . Por definición,  $F$  es un tensor de segundo orden [Khoei, et al., 2009]. La relación inversa esta dada por:

$$dX = F^{-1} \cdot dx = dx \cdot F^{-T}, \quad \text{donde} \quad F^{-T} = \frac{\partial X}{\partial x} \equiv \nabla X, \quad (1.21)$$

y  $\nabla$  es el operador gradiente con respecto a  $x$ . En notación indicial las ecuaciones (1.19) y (1.21) pueden ser escritas como:

$$\begin{aligned} F &= F_{iJ} \hat{e}_i \hat{E}_J, & F_{iJ} &= \frac{\partial x_i}{\partial X_J}, \\ F^{-1} &= F_{Ji}^{-1} \hat{E}_J \hat{e}_i, & F_{Ji}^{-1} &= \frac{\partial X_J}{\partial x_i}. \end{aligned} \quad (1.22)$$

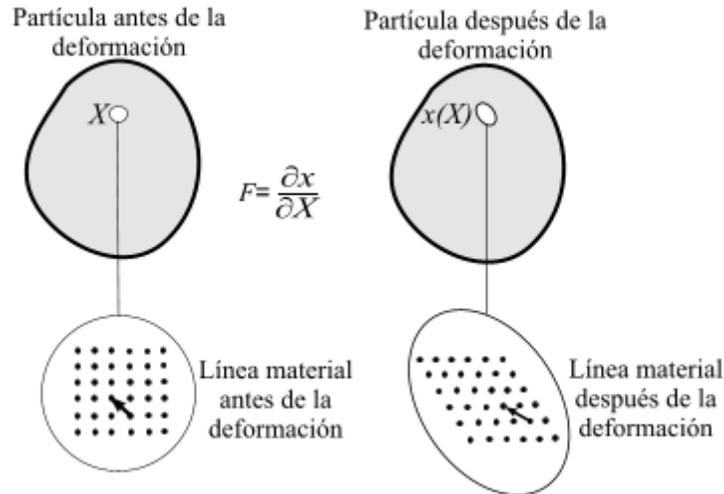
donde explícitamente tenemos:

$$[F] = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix}, \quad [F]^{-1} = \begin{bmatrix} \frac{\partial X_1}{\partial x_1} & \frac{\partial X_1}{\partial x_2} & \frac{\partial X_1}{\partial x_3} \\ \frac{\partial X_2}{\partial x_1} & \frac{\partial X_2}{\partial x_2} & \frac{\partial X_2}{\partial x_3} \\ \frac{\partial X_3}{\partial x_1} & \frac{\partial X_3}{\partial x_2} & \frac{\partial X_3}{\partial x_3} \end{bmatrix}.$$

En la ecuación (1.21) y (1.22), los índices en letras minúsculas se refieren a las coordenadas cartesianas actuales (espacial), mientras que los índices con letras mayúsculas se refieren a las coordenadas cartesianas de referencia (material) [Reddy, 2008]. El determinante de  $F$  es llamado la *Jacobiana* del movimiento, y es denotado por,  $J = \det F$ . La ecuación  $F \cdot dX = 0$  para  $dX \neq 0$  implica que una línea material en la configuración de referencia es reducida a cero por la deformación. Como esto no es físicamente real, podemos decir que  $F \cdot dX \neq 0$  para  $dX \neq 0$ .

Por tanto  $F$  es un tensor no singular;  $J \neq 0$ . Aquí  $F$  tiene una inversa  $F^{-1}$ . El gradiente de deformación puede ser expresado en función del vector desplazamientos como:

$$F = (\nabla_0 x)^T = (\nabla_0 u + I)^T \quad \text{o} \quad F^{-1} = (\nabla X)^T = (I - \nabla u)^T. \quad (1.23)$$



**Figura 1.5.** Representación de una partícula antes y después de la deformación.

### 1.6.1 Tensor de deformación infinitesimal.

En el caso de deformaciones infinitesimales [Mello, et al., 2009], ninguna distinción es realizada entre las coordenadas materiales  $X$  y las coordenadas espaciales  $x$ . El tensor de deformación infinitesimal es denotado por  $\varepsilon$ , y está dado por [Lion, et al., 2008]:

$$\varepsilon = \frac{1}{2} [\nabla u + (\nabla u)^T] \quad (1.24)$$

Las componentes cartesianas rectangulares del tensor de deformación infinitesimal están dados por [Holden, 2008]:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \quad (1.25)$$

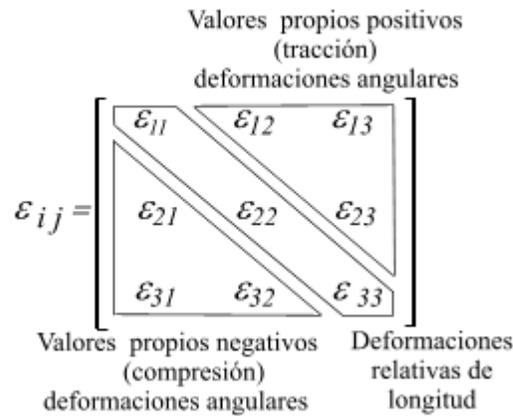
o en forma expandida

$$\begin{aligned} \varepsilon_{11} &= \frac{\partial u_1}{\partial X_1}; & \varepsilon_{13} &= \frac{1}{2} \left( \frac{\partial u_1}{\partial X_3} + \frac{\partial u_3}{\partial X_1} \right); \\ \varepsilon_{22} &= \frac{\partial u_2}{\partial X_2}; & \varepsilon_{12} &= \frac{1}{2} \left( \frac{\partial u_1}{\partial X_2} + \frac{\partial u_2}{\partial X_1} \right); \\ \varepsilon_{33} &= \frac{\partial u_3}{\partial X_3}; & \varepsilon_{23} &= \frac{1}{2} \left( \frac{\partial u_2}{\partial X_3} + \frac{\partial u_3}{\partial X_2} \right); \end{aligned}$$

Las componentes del tensor deformación  $\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}$  son las deformaciones infinitesimales normales y  $\varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23}$  son las deformaciones tangenciales. Las deformaciones tangenciales  $\gamma_{12} = 2\varepsilon_{12}, \gamma_{13} = 2\varepsilon_{13}, \gamma_{23} = 2\varepsilon_{23}$  son llamadas las deformaciones tangenciales ingenieriles [Closet, et al., 2009].

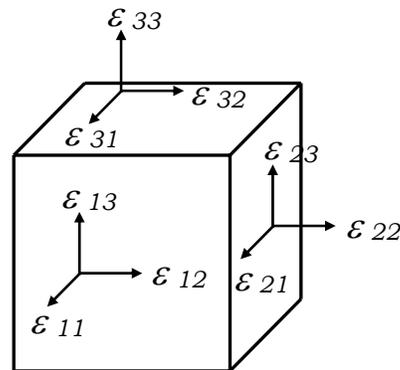
### 1.6.2 Interpretación física del tensor deformación infinitesimal.

La interpretación física y características matemáticas de los campos de tensores son altamente específicas a su aplicación, es importante que las técnicas de representación sean conducidas lo más cercana a su aplicación especial. Estos campos de tensores son caracterizados por la propiedad de tener valores propios positivos y negativos (*ver Figura 1.6*). La muestra de los valores propios indica regiones de tracción y compresión, y está por lo tanto son de interés especial [Hotz, et al., 2006].



**Figura 1.6** Representación de las componentes del Tensor Deformación.

Para evaluar las deformaciones en un punto material cualquiera, este se representa por un hexaedro de caras ortogonales (ver **Figura 1.7**) en los que cada plano representa cada una de las componentes del tensor deformación para ese punto.



**Figura 1.7.** Componentes del tensor deformación en las caras de un hexaedro.

### 1.7 Discretización del continuo.

Las limitaciones de la mente humana impiden abordar globalmente el comportamiento de los sistemas complejos que existen en la naturaleza. Por este motivo, el estudio de cualquier campo se acostumbra a dividir

los problemas en componentes más simples, que se puedan comprender básicamente y que permita posteriormente reconstruir el comportamiento del sistema global [Zhang, et al., 2008]. Este es el modo habitual en que se construyen numerosos modelos teóricos: a partir de un número finito de componentes bien definidos. Esta división sucesiva, llevada al límite, conduce al concepto matemático de *infinitesimal* [Sergeyev, 2007] como constituyente elemental de un sistema continuo.

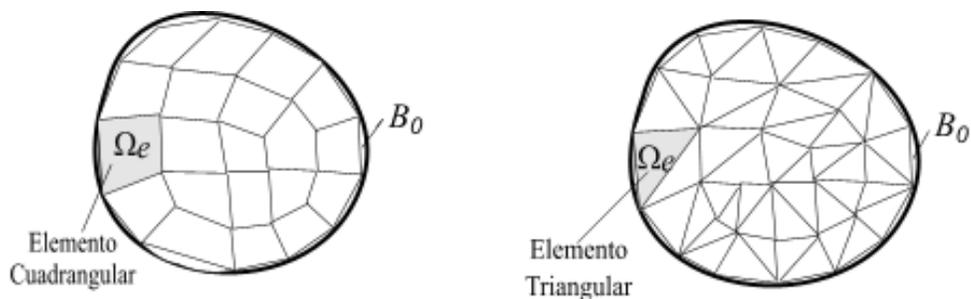
Sin embargo, ni la mente humana ni las computadoras, por muy potentes que sean, pueden tratar el número infinito de elementos infinitesimales que constituyen un sistema continuo, salvo en problemas simples que permitan una manipulación matemática exacta [Budd y Williams, 2009]. Para sortear estos obstáculos se han desarrollado varios *métodos de discretización*. Todos ellos involucran aproximaciones, pero, afortunadamente, la solución aproximada puede acercarse todo lo que se desee a la solución exacta, sin más que aumentar el número de variables discretas definidas [Gunawan y Papalambros, 2007].

La forma más sencilla de discretizar un problema, consiste en reducir el dominio a un número finito de puntos situados en los nodos de una malla [Cordero, et al., 2004]. La generación de mallas frecuentemente forma una parte crucial de los procedimientos de solución numérica en muchos problemas ingenieriles, desde la simulaciones de flujo hasta el análisis estructural [Du y Wang, 2006].

### 1.7.1 Generación de mallas.

La generación de mallas es definida como el proceso de discretización a un dominio físico dentro de pequeños sub-dominios (elementos), en orden para realizar soluciones numéricas a ecuaciones integrales o en derivadas parciales [Braun y Awruch, 2008; Bunin, 2008]. Aplicando el estándar de procedimiento de los elementos finitos, el dominio material  $B_0$  es dividido en  $n_{el}$  elementos (ver **Figura 1.9**), que son representados por el dominio correspondiendo con elementos material [Scherer, et al., 2008]. Los algoritmos de mallados generalmente definen las formas y distribuciones de los elementos [Geuzaine y Remacle, 2009].

La generación de mallas es usualmente considerada como el paso de pre-proceso de las técnicas de computación numérica [Lili, et al., 2008; Okafor, et al., 2009]. Las mallas usadas en los algoritmos de solución numérica satisfacen diversas condiciones dependiendo del problema en análisis [Veyhl, et al, 2010].

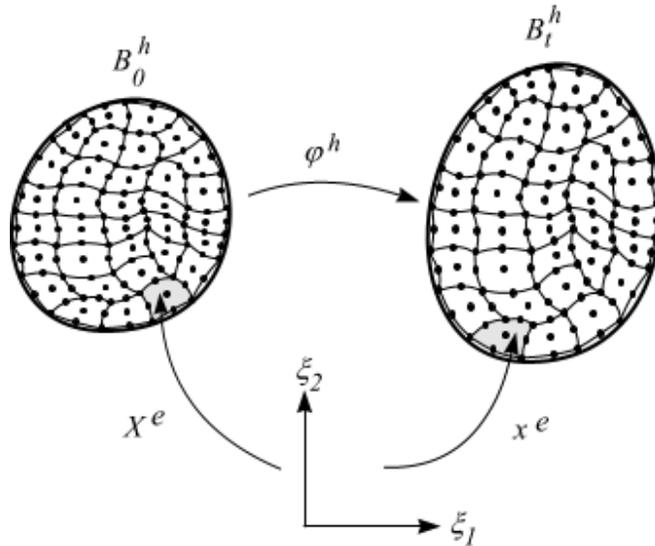


**Figura 1.9.** Representación de la discretización de un sólido en elementos cuadrangulares y triangulares.

### 1.7.2 Formulación isoparamétrica.

En el orden para resolver problemas numéricamente variacionales, la deformación espacial de los cuerpos es aproximada por elementos finitos. El diseño de la aproximación de los elementos finitos  $\varphi^h : B_0^h \rightarrow B_t^h$  de la deformación espacial es basado en el concepto de isoparamétrico de [Wriggers, 2001].

La formulación isoparamétrica mayormente es usada para problemas en que las aproximaciones son continuas  $C^0$ . En una formulación isoparamétrica un elemento se define en términos de un grupo de coordenadas naturales [Zienkiewicz y Taylor, 2005].



**Figura 1.10.** Discretización isoparamétrica del continuo.

El dominio material  $B_0$  y el dominio espacial  $B_t$  son aproximados por elementos finitos y  $B_t^h = \bigcup_{e=1}^{n_{el}} B_t^e$  (ver **Figura 1.10**). El mapeo de la

deformación espacial de cada elemento material para su complemento espacial [Wriggers, 2006; Scherer, et al., 2007; Scherer, et al., 2008].

$$\varphi^e : B_0^e \rightarrow B_t^e, \quad \varphi^e = x^e \circ X^{e,-1} \quad (1.26)$$

y esto es basado en:

$$X^e : B_\xi^e \rightarrow B_0^e, \quad X^e(\xi) = \sum_{i=1}^{n_{en}} N_i(\xi) X_i, \quad (1.27)$$

$$x^e : B_\xi^e \rightarrow B_t^e, \quad x^e(\xi) = \sum_{i=1}^{n_{en}} N_i(\xi) x_i, \quad (1.28)$$

donde  $B_\xi^e$  denota el llamado dominio en el  $\xi$ - espacio,  $i = 1, \dots, n_{en}$  es el numero de nodos locales y  $N_i$  son las funciones de interpolación asociadas con los nodos de los elementos,  $X_i$  denota las posición del punto nodo material y  $x_i$  son la posición de los puntos de los nodos espaciales. El ensamble de todas las deformaciones  $\varphi^e(X)$  produce la aproximación global del mapeo de la deformación espacial  $\varphi^h$  que puede ser escrita en la forma:

$$\varphi^h(X) = \sum_{i=1}^{n_n} N_i^g(X) x_i \quad (1.29)$$

donde  $i = 1, \dots, n_n$  denota el numero de nodos global,  $N_i^g$  son las funciones de interpolación  $C^0$ -continuas globales asociadas con los nodos de la malla y  $X$  es el vector posición de un punto material arbitrario dentro del dominio material discreto. En forma compacta el vector columna se escribe:

$$X = \left[ [x_1], \dots, [x_{n_{nx}}] \right]^T \quad (1.30)$$

que contiene todas las posiciones variables de los puntos nodales espaciales.

### **1.7.3 Concepto isoparamétrico.**

El término isoparamétrico surge al utilizar las mismas expresiones aritméticas para interpolar la geometría y el campo de desplazamientos, donde se expresa la geometría de un elemento finito isoparamétrico a partir de las coordenadas reales cartesianas, para posteriormente realizar un mapeo entre el espacio real y un espacio ideal unitario (coordenadas locales) donde todo está normalizado [Zienkiewicz, et al., 2010].

Los mallados estructurados pueden ser generados eficientemente con el mapeo (*ver Figura 1.11*) donde  $\xi_I$  son las coordenadas de cualquier punto arbitrario del plano de referencia en el elemento en el nodo  $I$  [Ma y Liu, 2006]. Para el uso del mapeo, los nodos entre cada pequeña región son posicionados automáticamente y referenciados para un sistema de coordenadas cartesianas global  $(X_1, \dots, X_n)$ , y finalmente los elementos son ensamblados desde sus nodos. Estos refinados elementos de bordes curvos o distorsionados son especialmente útiles para ser adaptados a estructuras continuas con perímetros no rectos [Konyukhov y Schweizerhof, 2008; Moslemi y Khoei, 2009]. El uso de transformación isoparamétrica ha demostrado ser computacionalmente eficiente, y

razonablemente sencillo para implementarlo en la generación de mallas estructuradas [Fahs, 2010].

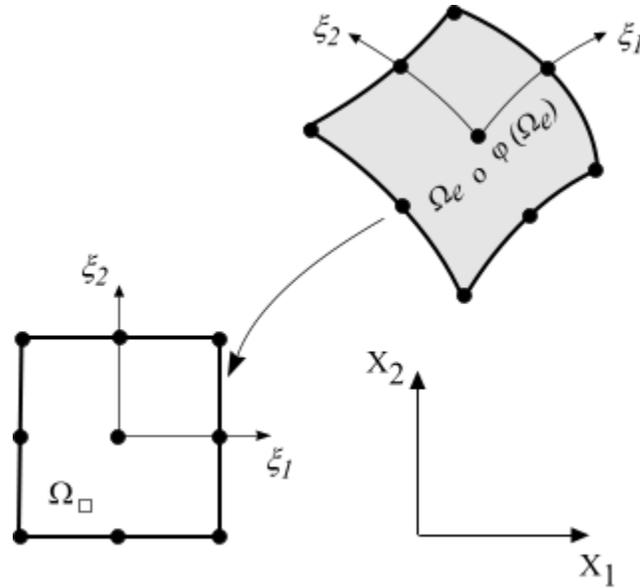
En el caso de la descripción del campo de desplazamiento, la discretización del continuo requiere que el campo de variables sea aproximado por elementos finitos isoparamétricos, de esta manera se puede expresar el campo de desplazamiento como:

$$u_{exact}(X) \approx u^h(X) = \sum_{i=1}^N N_i(X) u_i \quad (1.31)$$

donde  $X$  es la posición del vector en  $\Omega_e$ ,  $N_i(X)$  son las funciones de interpolación que son definidas en  $\Omega_e$ , y  $u_i$  denota las variables nodales desconocidas. Como en la formulación isoparamétrica la geometría y el campo de variables son aproximadas por las mismas funciones de interpolación los desplazamientos quedan expresados [Wriggers, 2006; Zienkiewicz y Taylor, 2005]:

$$u_e^h = \sum_{i=1}^n N_i(\xi) u_i \quad (1.32)$$

Las funciones de interpolación en la ecuación (1.32) son definidas en base a la configuración de referencia  $\Omega_\square$  (ver **Figura 1.11**).



**Figura 1.11** Mapeo Isoparamétrico.

#### 1.7.4 Elementos isoparamétricos en la discretización del continuo.

La (Figura 1.12) describe las dos posibilidades para describir la deformación en mecánica del continuo usando el concepto de isoparamétrico. Esto puede ser visto fácilmente que la (Figura 1.12) es la versión discreta de la (Figura 1.1), donde adicionalmente es introducida la configuración de referencia  $\Omega_{\square}$ . La relación cinemática dentro de un elemento esta dada por[Wriggers, 2006].:

$$F_e = j_e J_e^{-1} \quad (1.33)$$

que muestra que el gradiente de deformación es únicamente definido por el mapeo isoparamétrico de  $\Omega_{\square}$  sobre  $\Omega_e$  en la configuración inicial, o sobre  $\varphi(\Omega_e)$  en la configuración actual. En estas ecuaciones los gradientes  $j_e$  y  $J_e$  son definidos como:

$$j_e = \text{Grad}_\xi x = \frac{\partial x}{\partial \xi} = \sum_{I=1}^n N_{I,\xi}(\xi) x_I \otimes E_\xi, \quad (1.34)$$

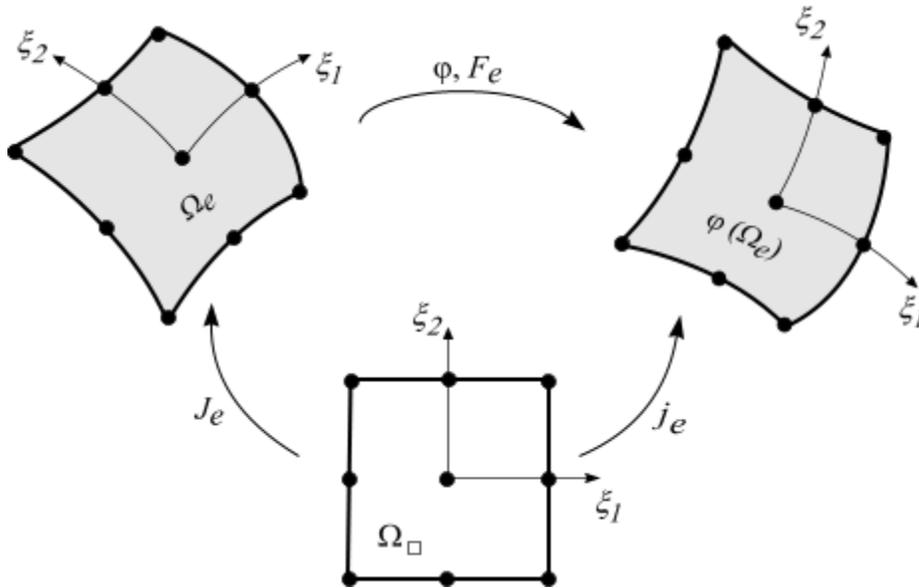
$$J_e = \text{Grad}_\xi X = \frac{\partial X}{\partial \xi} = \sum_{I=1}^n N_{I,\xi}(\xi) X_I \otimes E_\xi. \quad (1.35)$$

donde las derivadas  $N_{I,\xi}$  son cantidades escalares y moviendo las bases del vector obtenemos  $E_\xi$ :

$$j_e = \sum_{I=1}^n x_I \otimes N_{I,\xi}(\xi) E_\xi = \sum_{I=1}^n x_I \otimes \nabla_\xi N_I, \quad (1.36)$$

$$J_e = \sum_{I=1}^n X_I \otimes N_{I,\xi}(\xi) E_\xi = \sum_{I=1}^n X_I \otimes \nabla_\xi N_I. \quad (1.37)$$

donde  $\nabla_\xi N_I$  es el gradiente de la función escalar  $N_I$  con respecto a las coordenadas  $\xi$ .



**Figura 1.12.** Descripción isoparamétrica de las deformaciones.

## 1.8 Integración Numérica.

Ingenieros y científicos comúnmente asumen que la relación entre variables en un problema físico puede ser aproximadamente representada [Constantine, et al., 2009], dado los datos del problema. El objetivo final podría ser determinar los valores para los puntos intermedios, para aproximar la integral o derivada de las funciones fundamentales, o para simplemente dar una prueba o representación continua de las variables del problema [Doublas y Burden, 2002].

Interpolación se refiere a determinar una función que represente exactamente un grupo de datos. El tipo más elemental de interpolación consiste en ajustar un polinomio a un grupo de puntos de datos. Los polinomios tienen derivadas e integrales que no son mas que los mismos polinomios, y estos son una opción natural para aproximar derivadas e integrales [Doublas y Burden, 2002].

### 1.8.1 Funciones de Interpolación de LAGRANGE.

La interpolación de Lagrange es un simple método para hallar un único polinomio de grado  $n$  que pase exactamente a través de  $n + 1$  puntos [Xu y Cen, 2009]. Los polinomios de interpolación de *Lagrange* han sido el estándar para el análisis de elementos finitos, casi desde el principio que se creó el método. Ellos son la opción inexorable hoy en día, la preferida y la que prevalece en los códigos comerciales y en los artículos científicos [Benson, et al., 2009]. Trabajos como [Hou, et al., 2008, Peng et al.,

2009, Ha y Chang 2010] corroboran la afirmación de aplicabilidad de estos en la actualidad.

### 1.8.1.1 Polinomio de interpolación lineal de Lagrange.

Determinar un polinomio de primer grado que pase por medio de los distintos puntos  $(x_0, y_0)$  y  $(x_1, y_1)$  es igual que aproximar una función  $f$  para  $f(x_0) = y_0$  y  $f(x_1) = y_1$  por medio de un polinomio de interpolación de primer grado, concerniendo que, los valores de  $f$  para los puntos dados son [Occorsio, 2009]:

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \quad y \quad L_1(x) = \frac{x - x_0}{x_1 - x_0} \quad (1.38)$$

y esta definición implica que:

$$L_0(x_0) = \frac{x_0 - x_1}{x_0 - x_1} = 1, \quad L_0(x_1) = \frac{x_1 - x_1}{x_0 - x_1} = 0, \quad (1.39)$$

$$L_1(x_0) = 0 \quad y \quad L_1(x_1) = 1$$

por lo que se define:

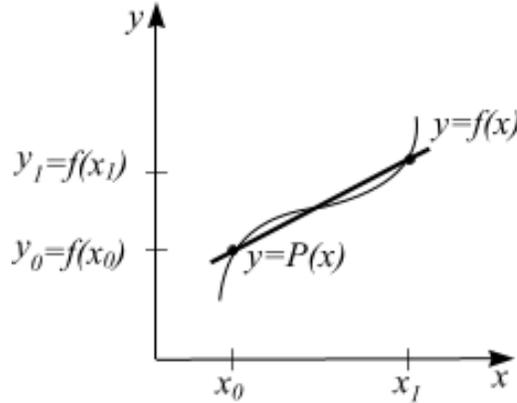
$$P(x) = L_0(x) f(x_0) + L_1(x) f(x_1) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \quad (1.40)$$

resultando:

$$P(x_0) = 1 \cdot f(x_0) + 0 \cdot f(x_1) = f(x_0) = y_0 \quad (1.41)$$

$$P(x_1) = 0 \cdot f(x_0) + 1 \cdot f(x_1) = f(x_1) = y_1 \quad (1.42)$$

Por lo que,  $P$  es la única función lineal que pasa por los puntos  $(x_0, y_0)$  y  $(x_1, y_1)$ . (ver **Figura 1.13**).



**Figura 1.13.** Polinomio de interpolación lineal de Lagrange.

### 1.8.1.2 Polinomio de interpolación de Lagrange de grado $n$ .

Para generalizar el concepto de interpolación lineal a polinomios de mayor orden, considerando la construcción de un polinomio de mayor grado  $n$  que pase por los puntos  $n + 1$

$$(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_n, f(x_n)). \quad (1.43)$$

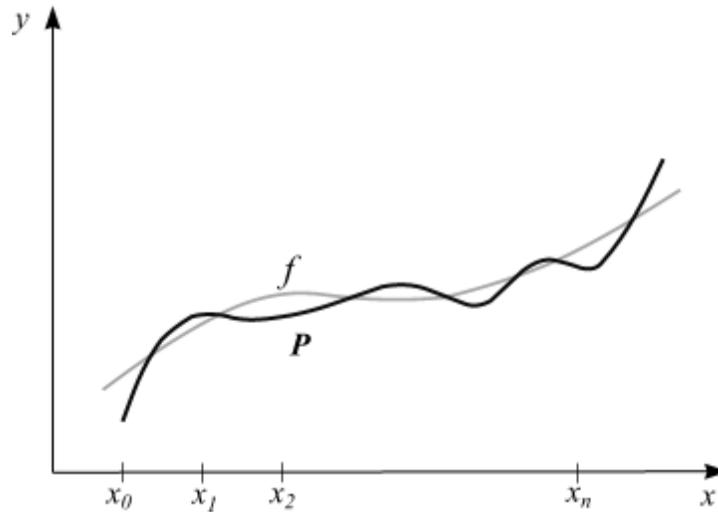
en este caso, se construye para cada  $k = 0, 1, \dots, n$ , un polinomio de grado  $n$ , denotado por  $L_{n,k}(x)$ , con la propiedad que  $L_{n,k}(x_i) = 0$  cuando  $i \neq k$  y  $L_{n,k}(x_k) = 1$ .

Por lo que  $L_{n,k}(x_i) = 0$  para cada  $i \neq k$ , el numerador de  $L_{n,k}(x)$ , contiene los términos:

$$(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n) \quad (1.44)$$

y para  $L_{n,k}(x_k) = 1$ , el denominador de  $L_{n,k}(x_i)$  este término debe ser evaluado para  $x = x_k$ , por lo que

$$L_{n,k}(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)} \quad (1.45)$$



**Figura 1.14.** Polinomio de interpolación de Lagrange de grado  $n$ .

### 1.8.2 Aplicación de los polinomios de interpolación de Lagrange en los elementos finitos.

Los elementos finitos emplean funciones de forma para la interpolación y estas son aplicadas en las mallas estructuradas [Paluszny, et al., 2007]. Una de las principales ventajas de las mallas es que ellas proporcionan una construcción explícita de los polinomios bases de Lagrange como productos de factores lineales. [Jaklic, et al., 2008].

En el orden para trabajar con aproximaciones polinomiales con el patrón de elementos finitos, los polinomios de Lagrange (es decir, interpolación nodal estándar) pueden ser considerados. De hecho, ellos pueden ser definidos en los elementos curvados  $\Omega_e$  en el dominio de referencia o equivalente del elemento real en el dominio físico [Sevilla, et al., 2008].

En general los elementos finitos residen en la elección de las funciones de forma locales y estas siguen el estándar de aplicación de los métodos

de interpolación donde toma valores no nulos el punto en análisis y se anulan sus valores para todos los demás puntos. Cada función de forma queda definida por una expresión analítica (generalmente polinómica) por el punto donde toma valores no nulos, es decir que para cada nodo  $i$  se define una función  $N_i$  y esta se elige de forma que sea igual a la unidad en el nodo  $i$  y nula en el resto de nodos.

### **1.9 Conclusiones parciales del capítulo.**

Como resultado del análisis crítico de la bibliografía se ha arribado a las siguientes conclusiones:

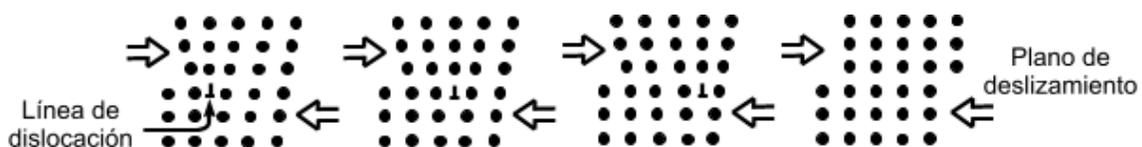
1. Los fenómenos donde las dimensiones son muy superiores a la estructura atómica del material, pueden ser modelados matemáticamente por la *Teoría del Continuo*.
2. La mecánica del medio continuo desarrolla la formulación matemática apropiada para calcular las deformaciones que se producen en fenómenos físicos complejos.
3. Se demuestra que la discretización en elementos finitos isoparamétricos (mallado) logran una buena aproximación de la geometría real para la modelación matemática.
4. Se demuestra que los polinomios de *Lagrange* logran una aproximación considerable para emplearlos en las funciones de interpolación en el cálculo por elementos finitos.

## CAPÍTULO 2. DESCRIPCIÓN DEL FENÓMENO FÍSICO.

En este capítulo se hace la descripción de la esencia del fenómeno físico (endurecimiento por deformación) y se propone el algoritmo de solución para la modelación matemática cumpliendo con las leyes que rigen la Mecánica del Medio Continuo.

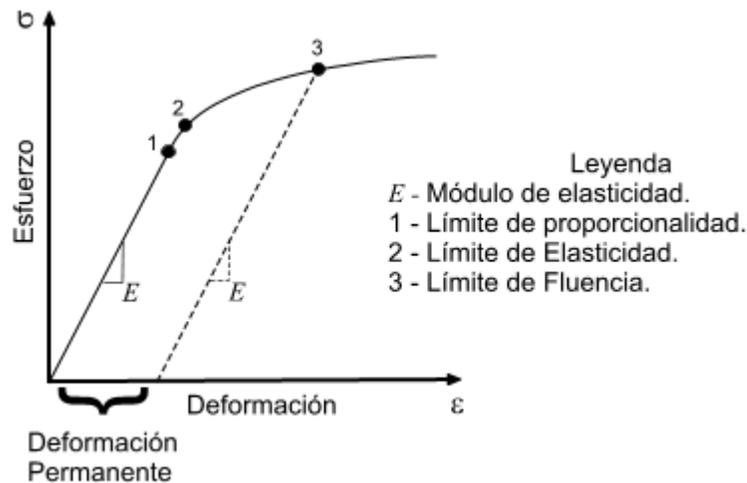
### 2.1 Descripción de la esencia del fenómeno físico.

Como ha sido planteado en el capítulo anterior la materia no es continua, y básicamente es conocido que el arreglo entre los átomos de los materiales cristalinos, no es perfecto. Estos defectos en ingeniería han sido de singular importancia, pues a partir de ellos se han generado procesos tecnológicos que mejoran las propiedades mecánicas de los materiales. Microscópicamente, en la escala de la red cristalina de los metales, la plasticidad es una consecuencia de la existencia de ciertas imperfecciones en la red llamadas *dislocaciones*. Las dislocaciones son responsables de la ductilidad de los metales. La línea de las dislocaciones se desliza sobre los planos cristalinos de la estructura cuando se aplica un esfuerzo y esto produce la deformación plástica (*ver Figura 2.1*).



**Figura 2.1.** Movimiento de las dislocaciones.

La plasticidad es la propiedad mecánica de un material, biológico o de otro tipo, de deformarse permanentemente e irreversiblemente cuando se encuentra sometido a esfuerzos por encima de su rango elástico, es decir, por encima de su límite elástico. En la (Figura 2.2) se muestra la curva esfuerzo-deformación.



**Figura 2.2.** Curva de esfuerzo-deformación.

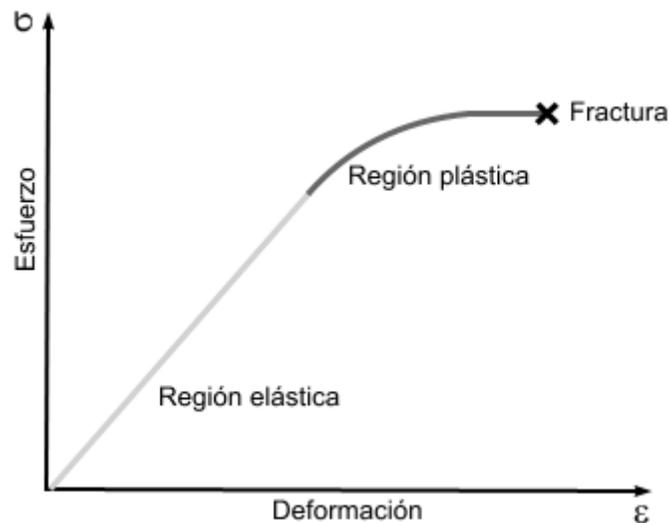
En los procesos de deformación de metales se aprovechan las propiedades de flujo plástico del material a medida que es deformado.

## 2.2 Teoría de la deformación.

En la teoría de la deformación el tensor de tensiones (de orden  $n$  en  $m$  dimensiones) esta en función del tensor deformación. Aunque esta descripción es aproximada cuando una pequeña parte del material es sujeta a cargas de incremento (tales como cargas de deformación), esta teoría no puede ser calculada irreversiblemente.

Los materiales dúctiles pueden sostener grandes deformaciones plásticas sin fractura. Sin embargo, incluso metales dúctiles fracturan cuando la deformación convenientemente es bastante grande -este es el resultado del endurecimiento por deformación de un material, que causa su quebramiento-.

En la (**Figura 2.3**) una idealizada curva uniaxial de esfuerzo-deformación muestra los regimenes de deformación elástica y plástica para la teoría de la plasticidad.



**Figura 2.3.** Curva uniaxial de esfuerzo-deformación.

### 2.3 Mecanismos por deformación.

En los procesos ingenieriles los mecanismos por deformación, se definen como la secuencia de eventos que producen la deformación en un material. Estos se refieren únicamente a la deformación plástica, ya que la deformación elástica es el resultado del alargamiento de enlaces

atómicos, mientras que la deformación plástica implica el reacomodo de enlaces.

### **2.3.1 Endurecimiento por deformación.**

El trabajo de endurecimiento, también conocido como endurecimiento por deformación, es el endurecimiento de un metal por deformación plástica; este se produce con el incremento en el esfuerzo necesario para continuar la deformación de un material

### **2.3.2 Endurecimiento por deformación en frío.**

En los metales se distinguen dos tipos de deformación: de baja temperatura o en frío y de alta temperatura o en caliente. La diferencia es que en la deformación de alta temperatura predominan procesos térmicos-activados, como son la difusión y el flujo viscoso, que transcurren a través de tiempo; mientras que a baja temperatura la deformación es producida únicamente por los esfuerzos y no depende del tiempo.

El trabajo de endurecimiento puede ser deseable o no dependiendo del contexto. El endurecimiento ocurre en los procesos de la metalurgia que intencionalmente inducen a la deformación plástica para imponer un cambio de forma. Estos procesos son los conocidos como trabajo en frío o procesos de conformación en frío. Ellos son caracterizados por dar forma a la pieza a una temperatura por debajo de su temperatura de recristalización, por lo general a la temperatura ambiente.

## **2.4 Deformación plástica superficial.**

Dentro de los mecanismos por deformación los procesos de deformación plástica superficial (DPS) son de especial interés, estos constituyen un método de endurecimiento superficial<sup>4</sup> de las piezas para incrementar sus cualidades físico-mecánicas y de acabado aprovechando las características de plasticidad de los metales. En la última década estos procesos por (DPS) han sido empleados con frecuencia, producto que además de mejorar las propiedades mecánicas de los materiales, evitan el empleo de esquemas tecnológicos tradicionales como el rectificado y los tratamientos térmicos, tales como el temple superficial, que son altamente consumidores de energía y contaminadores del medio ambiente.

Estos procesos de tratamiento por (DPS) se clasifican según el elemento deformante empleado y entre los más conocidos se encuentran:

1. Bruñido por rodillo. (múltiples y simple) (*Roller burnishing*)
2. Bruñido por bola. (*Ball burnishing*)
3. Pulido con herramienta de diamante. (*Diamond polishing*)
4. Bruñido de baja plasticidad. (*Low plasticity burnishing*)
5. Bruñido vibratorio o vibrorodilado. (*Vibratory burnishing*).
6. Lanzamiento de granallas o chorreado de municiones. (*Shot peening*)

---

<sup>4</sup> **Nota:** Solo se deforma la capa superficial, esta es conocida en ingeniería como profundidad de acritud (aproximadamente 0.1 mm para los aceros).

7. Impacto de vibración ultrasónica. (*Ultrasonic vibration impact*)
8. Impacto por láser. (*Laser shock*)
9. Combinados.

### **2.5 Bruñido por rodillo simple<sup>5</sup>.**

El bruñido, que emplea el rodillo como elemento deformante, es un proceso de elaboración en frío en la superficie de una pieza previamente maquinada. La pequeña deformación plástica superficial originada por esta operación consiste en el desplazamiento del material de los “picos, protuberancias o crestas” a los “valles o depresiones” de las microirregularidades superficiales, este flujo ocurre bajo una fuerza controlada del rodillo que excede el punto de fluencia del material de la superficie de la pieza no endurecida creándose una capa de metal consolidada que provoca el aumento de las propiedades funcionales en la superficie.

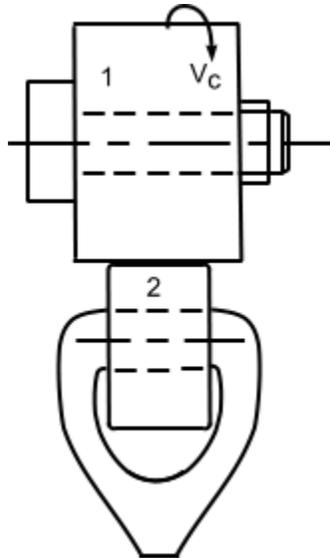
Este método de elaboración resulta sencillo, de fácil aplicación, es posible emplearlo en una gran variedad de piezas y de metales, y le incrementa a las piezas resistencia al desgaste, a la fatiga y a la corrosión, además de que mejora sustancialmente el acabado superficial y la precisión de las piezas.

---

<sup>5</sup> **Nota:** De los procesos de deformación plástica superficial (DPS) solo se explica el proceso de deformación de bruñido por rodillo para posteriormente realizar el cálculo de las deformaciones a través de la mecánica del medio continuo.

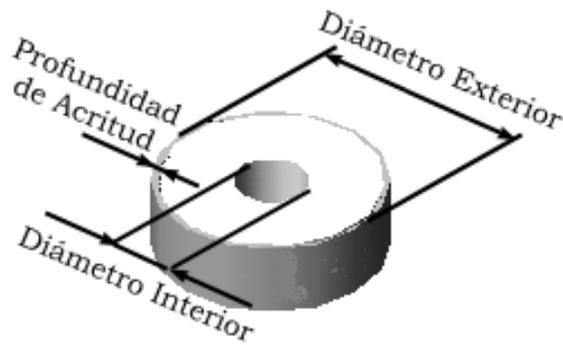
## 2.6 Descripción del experimento a modelar.

Las probetas (cuerpos de revolución) fueron sometidas a deformación plástica superficial mediante la operación de bruñido por rodillo simple [ver **Figura 2.4**].



**Figura 2.4** Proceso de bruñido por rodillo simple. (1 Probeta, 2 Rodillo).

La maquina herramienta empleada para el proceso tecnológico fue el torno universal de modelo C11MT de fabricación búlgara. Las mediciones de los diámetros y longitud [ver **Figura 2.5**] de las probetas antes y después de la deformación se realizaron con el pie de rey digital MITUTOYO de fabricación japonés.



**Figura 2.5** Dimensiones de las probetas.

En la siguiente (**Tabla 6 2.1**) se muestran las dimensiones de las mismas:

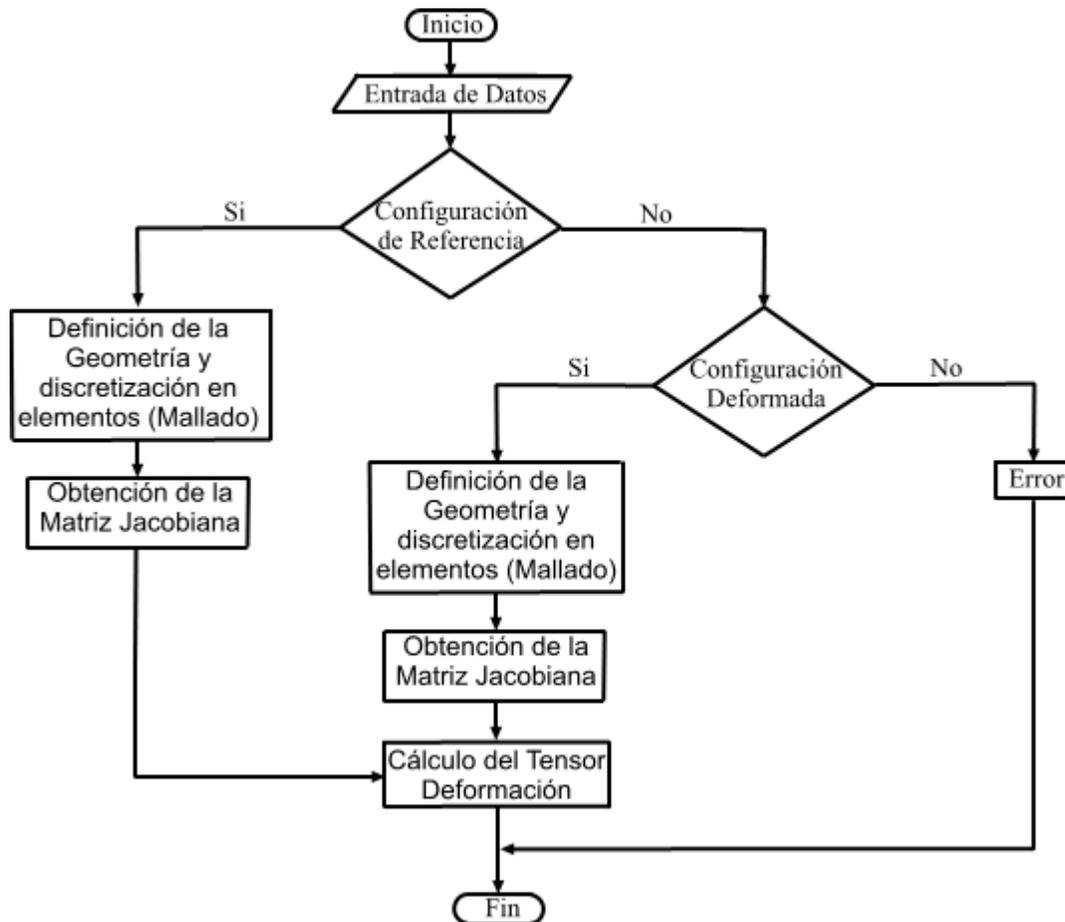
No.	Dimensiones de la probeta antes de la deformación.		Dimensiones de la probeta después de la deformación.	
	Diámetro.		Diámetro.	
	Int.	Ext.	Int.	Ext.
1	20 mm	35 mm	20 mm	34.94 mm
2	20 mm	35 mm	20 mm	34.96 mm
3	20 mm	35 mm	20 mm	34.94 mm
4	20 mm	35 mm	20 mm	34.95 mm
Media Total	20 mm	35 mm	20 mm	34.948 mm

**Tabla 2.1** Dimensiones de las probetas antes y después de la deformación por rodillo simple.

<sup>6</sup> **Nota:** Para la modelación matemática del proceso de bruñido por rodillo simple se toma como diámetro menor el perteneciente a la profundidad de acritud (es decir, se le sustrae 0.1mm al diámetro exterior de la configuración deformada)

### 2.7 Algoritmo propuesto para la modelación matemática.

En la modelación matemática de un fenómeno físico la secuencia o metodología de pasos para el desarrollo de la misma es de vital importancia. En la **(Figura 2.6)** se muestra el algoritmo de solución general realizado por el autor de este trabajo para el cálculo de las deformaciones que se producen en los procesos de endurecimiento por deformación en piezas con configuración geométrica de cuerpos de revolución mediante la operación de bruñido por rodillo simple.



**Figura 2.6.** Algoritmo de solución.

La implementación de este algoritmo de solución se realiza mediante un paquete de funciones de MATLAB (archivos *.m*), elaboradas por el autor de este trabajo. Los códigos implementados en cada una de las funciones forman parte del software **MECACONTINUO** en su *versión 1.0* registrado en el “Centro Nacional de Derecho de Autor” (*CENDA*). Dicho software permite realizar cálculos de mecánica del medio continuo basados en la matemática numérica para realizar la modelación de cuerpos sólidos y es compatible con las versiones 5.1 y posteriores, de MATLAB. (*ver Anexos 2 “códigos de MATLAB para la modelación matemática”* ).

### **2.8 Conclusiones parciales del capítulo.**

Según lo abordado en este capítulo se arriba a las siguientes conclusiones:

1. Se ha realizado la descripción del fenómeno físico de endurecimiento por deformación para su posterior modelación matemática.
2. Se ha detallado las ventajas (tecnológicas, económicas y ambientales) y el empleo actual del endurecimiento por deformación plástica superficial, básicamente para el proceso de bruñido por rodillo simple (rodillado).
3. Se formula el algoritmo de solución para la modelación matemática del endurecimiento por deformación plástica superficial a través de la teoría y expresiones que rigen la mecánica del medio continuo.

## CAPÍTULO 3. MODELACIÓN MATEMÁTICA.

En este capítulo se hace la descripción y aplicación del algoritmo desarrollado por el autor para determinar las deformaciones que se producen en el proceso de endurecimiento por deformación, (básicamente para la operación de bruñido por rodillo simple).

### 3.1 Entrada de datos.

La entrada de datos es de vital importancia para la solución, a continuación se listan los datos generales para la modelación matemática del endurecimiento por deformación (bruñido por rodillo simple).

- Radio mínimo del sólido a modelar. Debe ser introducido en mm. Se asigna la variable **Rmin.**
- Radio Máximo de la configuración inicial. Debe ser introducido en mm. Se asigna la variable **Rmax.**
- Radio Máximo de la configuración deformada Debe ser introducido en mm. Se asigna la variable **rmax.**
- Cantidad de pasos en dirección radial. Se asigna la variable **Rn.**
- Ángulo mínimo del sólido a modelar. Debe ser introducido el valor del ángulo en grados. Se asigna la variable **Amax.**
- Ángulo máximo del sólido a modelar. Debe ser introducido el valor del ángulo en grados Se asigna la variable **Amin.**

- Cantidad de pasos en dirección angular. Se asigna la variable **An**.
- Altura mínima del sólido a modelar. Debe ser introducido en mm.  
Se asigna la variable **Zmin**.
- Altura máxima del sólido a modelar. Debe ser introducido en mm.  
Se asigna la variable **Zmax**.
- Cantidad de pasos en dirección vertical. Se asigna la variable **Zn**.
- Las variables lógicas en las tres direcciones del modelo tridimensional del sólido. A las mismas se les asignan las variables  $\xi, \eta, \zeta$ , llamadas estas: **Xi, Eta, Dzeta**, respectivamente.

### 3.2 Toma de decisiones.

El análisis de problemas de mecánica del medio continuo implica para la solución, fraccionar al menos en dos tiempos el fenómeno físico que se desea modelar. El algoritmo propuesto en el capítulo anterior presenta una estructura ramificada originada por la inclusión de dos elementos condicionales que corresponden a la configuración de referencia y la configuración deformada respectivamente.

En el algoritmo la toma de decisiones se produce después de la entrada de datos generales (es decir datos pertenecientes a las configuraciones de referencia y deformada) para la solución del problema. Cada toma de decisiones produce dos alternativas posibles, las cuales originan las dos posibles ramas a seguir en la secuencia siguiente:

1. Si se cumple la condición para la configuración de referencia se procede a discretizar la geometría (cuerpos de revolución) para las probetas sin deformar generando un mallado con elementos hexaédricos.
2. Posteriormente a partir de los puntos que conforman la malla y las funciones de interpolación de cada nodo se calcula la matriz *Jacobiana* para cada elemento finito de la geometría discretizada de la configuración de referencia.
3. En caso que no se cumpla la condición para la configuración de referencia se accede a la toma de decisiones para la configuración deformada.
4. Si se cumple la condición para la configuración deformada se procede a discretizar la geometría (cuerpos de revolución) para las probetas después de la deformación generando un mallado con elementos hexaédricos.
5. Posteriormente a partir de los puntos que conforman la malla y las funciones de interpolación de cada nodo se calcula la matriz *Jacobiana* para cada elemento finito de la geometría discretizada de la configuración deformada.
6. Cálculo del Tensor Deformación a partir de las matrices *Jacobiana* obtenidas para cada elemento finito que conforma la geometría discretizada en las configuraciones de referencia y deformada.

7. En caso de no cumplirse ninguna de las dos condiciones (configuración de referencia y configuración deformada) se genera automáticamente un *Error*.

### 3.3 Definición de la geometría y discretización en elementos.

La definición de la geometría del objeto físico a modelar es elaborada a partir del siguiente concepto:

*“Un sistema de coordenadas es una correspondencia uno a uno entre vectores en el espacio  $n$ -dimensional Euclidiano  $E^n$  y un grupo de números reales  $n$   $(x^1, x^2, \dots, x^n)$ ”.*

Estos números son llamados coordenadas de vectores correspondientes y son escritas:

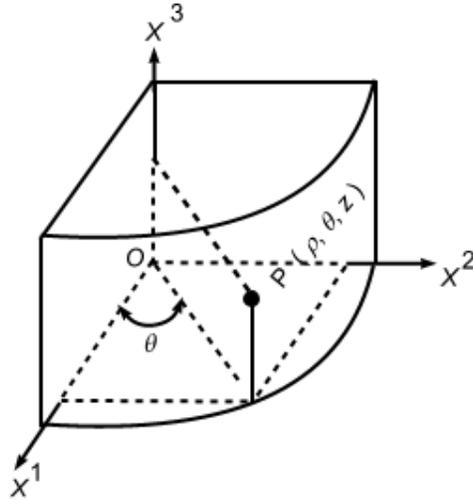
$$x^i = x^i(R) \quad \Leftrightarrow \quad R = R(x^1, x^2, \dots, x^n) \quad (3.1)$$

donde  $R \in E^n$  y  $x^i \in \mathbb{R}$  ( $i = x^1, x^2, \dots, x^n$ ). Por lo que las funciones  $x^i = x^i(R)$  y  $R = R(x^1, x^2, \dots, x^n)$  son suficientemente derivables.

#### 3.3.1 Modelación Geométrica.

La geometría del cuerpo en revolución es generada inicialmente en coordenadas cilíndricas en  $E^3$  definidas por la posición de sus puntos del espacio mediante un ángulo, una distancia con respecto a un eje y una altura en la dirección del eje (*ver Figura 3.1*).

Donde  $\rho$  es el radio vector de la proyección del punto sobre el plano  $X^1X^2$ , el ángulo  $\theta$  que forma dicha proyección  $OX^1$  y su altura  $X^3$  del punto.



**Figura 3.1.** Coordenadas cilíndricas.

La transformación de las coordenadas cilíndricas en  $E^3$  a coordenadas rectangulares en  $E^3$  es realizada a través de las siguientes ecuaciones:

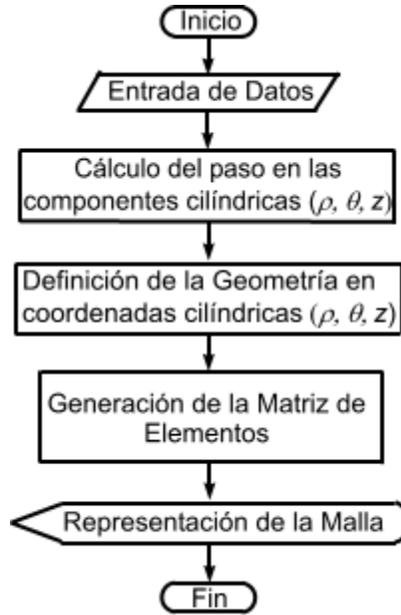
$$X^1 = \rho \cos \theta \quad X^2 = \rho \operatorname{sen} \theta \quad X^3 = z \quad (3.2)$$

por lo que podemos formular las coordenadas que conforman el cuerpo de un cilindro mediante la siguiente expresión:

$$R = R(\rho, \theta, z) = \rho \cos \theta e_1 + \rho \operatorname{sen} \theta e_2 + z e_3 \quad (3.3)$$

### 3.3.2 Discretización en elementos. (Generación del mallado).

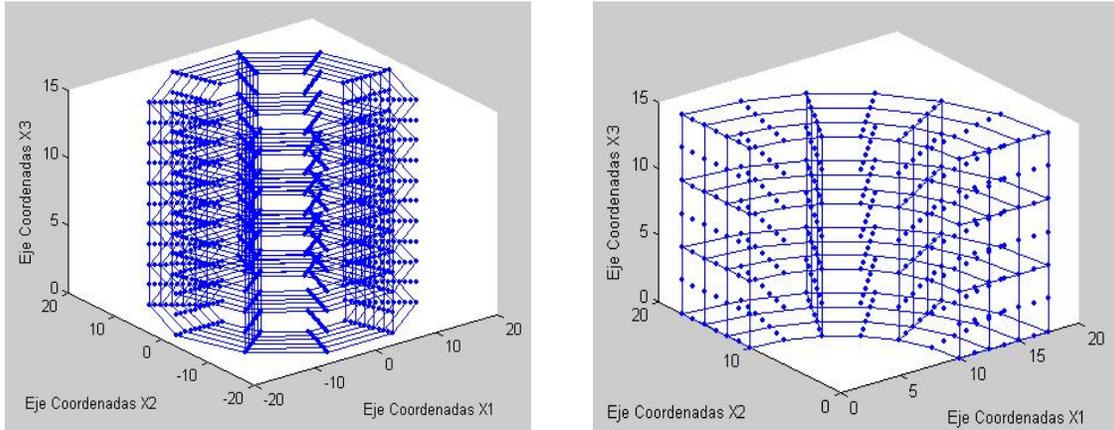
La secuencia de pasos para la discretización de la geometría en elementos y su posterior representación en una malla se muestra en el algoritmo de la **(Figura 3.2)**.



**Figura 3.2.** Algoritmo de generación de la Matriz de Elementos y representación del Mallado.

Iniciado el programa y entrado los datos generales se realiza el cálculo del paso en dirección radial, el cálculo del paso en dirección angular y el cálculo del paso en dirección vertical (altura). Seguidamente se realiza la definición de cada punto de la geometría en coordenadas cilíndricas. Obtenidas las tres componentes cilíndricas  $(\rho, \theta, z)$  de cada punto que conforman la geometría discretizada del sólido, se genera la *Matriz de Elementos*. En esta matriz se almacenan los valores de cada punto en sus tres componentes pertenecientes al elemento que este corresponda. El número del elemento que este representa en la geometría global discretizada coincidirá con el número de fila de la *Matriz de Elementos*.

La generación de la malla de elementos hexaédricos que discretiza la geometría (cuerpo revolución) es realizada desde la definición de todos los puntos y la matriz de elementos.



Geometría total discretizada

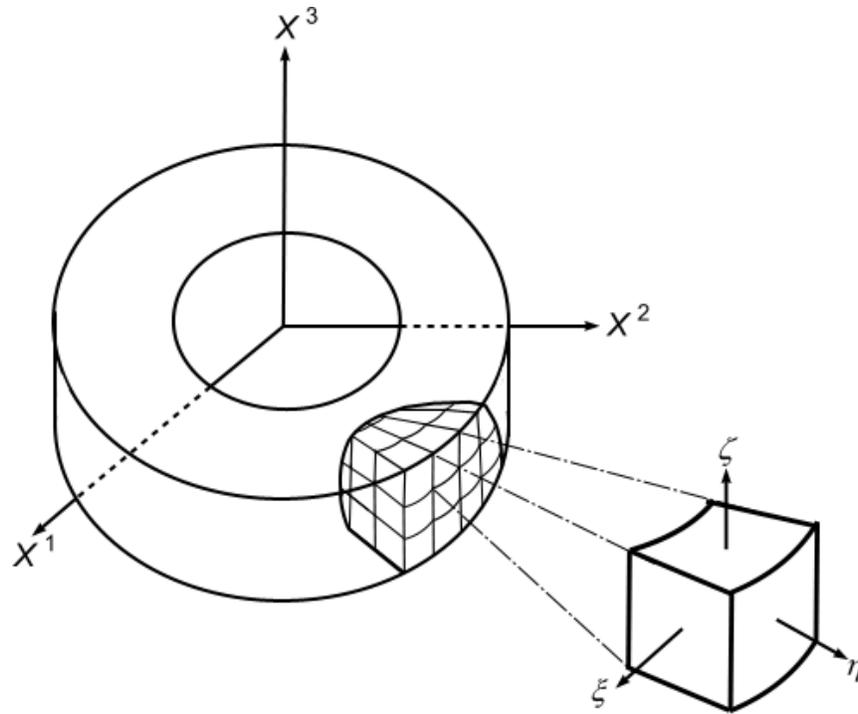
Cuarto de la geometría discretizada

**Figura 3.3.** Representación del mallado hexaédrico.

En la (**Figura 3.3**) se muestra la representación de la probeta (cuerpo de revolución) discretizada en elementos hexaédricos cuadráticos para la geometría total y un cuarto de la misma respectivamente.

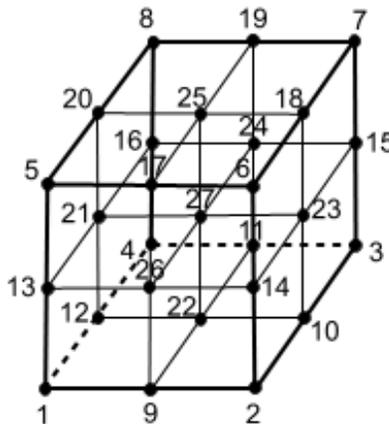
### 3.4 Definición de las funciones de interpolación.

Las funciones de interpolación del elemento hexaédrico se obtuvieron a través de la combinación del polinomio de *Lagranje* según las tres direcciones ( $\xi, \eta, \zeta$ ) que representan las coordenadas naturales (locales) de cada elemento hexaédrico en que es discretizada la probeta como indica la (**Figura 3.4**).



**Figura 3.4.** Discretización de la probeta en coordenadas naturales (locales).

por lo que la función de forma del nodo 1 del elemento hexaédrico de 27 nodos que se muestra en la (**Figura 3.5**), es obtenida por el producto de



**Figura 3.5.** Enumeración de los nodos en el elemento hexaédrico.

los tres polinomio de *Lagrange* según las tres direcciones, si tomamos como valores el intervalo [-1 1] en las tres direcciones, tenemos que:

$$N_1^{(1)}(\xi) = \frac{(\xi - \xi_9)(\xi - \xi_2)}{(\xi_1 - \xi_9)(\xi_1 - \xi_2)} = \frac{(\xi - 0)(\xi - 1)}{(-1 - 0)(-1 - 1)} = \frac{\xi(\xi - 1)}{2} = \frac{1}{2}(\xi^2 - \xi) \quad (3.4)$$

$$N_1^{(2)}(\eta) = \frac{(\eta - \eta_{12})(\eta - \eta_4)}{(\eta_1 - \eta_{12})(\eta_1 - \eta_4)} = \frac{(\eta - 0)(\eta - 1)}{(-1 - 0)(-1 - 1)} = \frac{\eta(\eta - 1)}{2} = \frac{1}{2}(\eta^2 - \eta) \quad (3.5)$$

$$N_1^{(3)}(\zeta) = \frac{(\zeta - \zeta_{13})(\zeta - \zeta_5)}{(\zeta_1 - \zeta_{13})(\zeta_1 - \zeta_5)} = \frac{(\zeta - 0)(\zeta - 1)}{(-1 - 0)(-1 - 1)} = \frac{\zeta(\zeta - 1)}{2} = \frac{1}{2}(\zeta^2 - \zeta) \quad (3.6)$$

obteniendo como función de forma para el nodo 1

$$N_1(\xi, \eta, \zeta) = \frac{1}{8}(\xi^2 - \xi)(\eta^2 - \eta)(\zeta^2 - \zeta) \quad (3.7)$$

análogamente se obtuvieron las funciones de forma para los demás nodos (*ver Anexo I*).

### 3.5 Obtención de la Matriz Jacobiana.

Como las funciones de forma son definidas en términos de coordenadas naturales,  $(\xi, \eta, \zeta)$  para obtener las derivadas con respecto a  $(X^1, X^2, X^3)$  para el cálculo de las deformaciones, es empleada para la diferenciación parcial la regla de la cadena:

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial X^1} \frac{\partial X^1}{\partial \xi} + \frac{\partial N_i}{\partial X^2} \frac{\partial X^2}{\partial \xi} + \frac{\partial N_i}{\partial X^3} \frac{\partial X^3}{\partial \xi} \quad (3.8)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial X^1} \frac{\partial X^1}{\partial \eta} + \frac{\partial N_i}{\partial X^2} \frac{\partial X^2}{\partial \eta} + \frac{\partial N_i}{\partial X^3} \frac{\partial X^3}{\partial \eta} \quad (3.9)$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{\partial N_i}{\partial X^1} \frac{\partial X^1}{\partial \zeta} + \frac{\partial N_i}{\partial X^2} \frac{\partial X^2}{\partial \zeta} + \frac{\partial N_i}{\partial X^3} \frac{\partial X^3}{\partial \zeta} \quad (3.10)$$

que puede ser expresada en forma matricial

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} = J \begin{Bmatrix} \frac{\partial N_i}{\partial X^1} \\ \frac{\partial N_i}{\partial X^2} \\ \frac{\partial N_i}{\partial X^3} \end{Bmatrix} \quad (3.11)$$

donde  $J$  es la matriz *Jacobiana* definida por

$$J = \begin{bmatrix} \frac{\partial X^1}{\partial \xi} & \frac{\partial X^2}{\partial \xi} & \frac{\partial X^3}{\partial \xi} \\ \frac{\partial X^1}{\partial \eta} & \frac{\partial X^2}{\partial \eta} & \frac{\partial X^3}{\partial \eta} \\ \frac{\partial X^1}{\partial \zeta} & \frac{\partial X^2}{\partial \zeta} & \frac{\partial X^3}{\partial \zeta} \end{bmatrix} \quad (3.12)$$

Las coordenadas  $(X^1, X^2, X^3)$  son interpoladas por la función de forma desde las coordenadas nodales. Aquí es sustituida la interpolación de las coordenadas

$$J = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \dots & \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \dots & \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \dots & \frac{\partial N_i}{\partial \zeta} \end{bmatrix} \begin{bmatrix} X_1^1 & X_1^2 & X_1^3 \\ X_2^1 & X_2^2 & X_2^3 \\ \vdots & \vdots & \vdots \\ X_i^1 & X_i^2 & X_i^3 \end{bmatrix} \quad \text{donde } (i = 1, \dots, 27) \quad (3.13)$$

donde queda expresada:

$$J = \begin{bmatrix} \sum_{i=1}^i X_i^1 \partial N_i / \partial \xi & \sum_{i=1}^i X_i^2 \partial N_i / \partial \xi & \sum_{i=1}^i X_i^3 \partial N_i / \partial \xi \\ \sum_{i=1}^i X_i^1 \partial N_i / \partial \eta & \sum_{i=1}^i X_i^2 \partial N_i / \partial \eta & \sum_{i=1}^i X_i^3 \partial N_i / \partial \eta \\ \sum_{i=1}^i X_i^1 \partial N_i / \partial \zeta & \sum_{i=1}^i X_i^2 \partial N_i / \partial \zeta & \sum_{i=1}^i X_i^3 \partial N_i / \partial \zeta \end{bmatrix} \quad (3.14)$$

de forma más compacta queda expresada como:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial X^1} \\ \frac{\partial N_i}{\partial X^2} \\ \frac{\partial N_i}{\partial X^3} \end{Bmatrix} = J^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} \quad (3.15)$$

### 3.6 Cálculo del Tensor Deformación.

El cálculo del tensor deformación en las configuraciones de referencia y deformada es realizado a partir de la expresión (*Matriz Jacobiana*) que describe el movimiento de las partículas para cada configuración cumpliendo con las leyes que rigen la mecánica del medio continuo:

$$[F] = \frac{\partial x_i}{\partial X_j} \quad \text{donde } (i = 1, \dots, 3) \text{ y } (j = 1, \dots, 3) \quad (3.16)$$

$$[F]^{-1} = \frac{\partial X_i}{\partial x_j} \quad \text{donde } (i = 1, \dots, 3) \text{ y } (j = 1, \dots, 3) \quad (3.17)$$

estas expresiones describen las deformaciones ocurridas en las configuraciones de referencia y deformada respectivamente.

Según la discretización empleada en los elementos finitos y sustituyendo la expresión (3.16) en (3.17) tenemos que el tensor deformación puede ser calculado mediante la expresión:

$$F_e = j_e J_e^{-1} \quad (3.18)$$

donde  $J_e$  es la matriz *Jacobiana* para cada elemento de la configuración de referencia y  $j_e$  es la matriz *Jacobiana* de cada elemento para la configuración deformada.

### 3.7 Interpretación de las deformaciones por cada elemento finito.

El tensor de deformación obtenido por el cálculo representa la deformación aproximada de cada elemento analizado. El tensor de deformación en su conjunto está compuesto por  $m$ -filas,  $n$ -columnas y  $k$ -dimensiones. Las filas y columnas representan las deformaciones ocurridas en cada punto o partícula del medio continuo en las coordenadas  $X_i$ , (para  $i = 1, 2, 3$ ) y las dimensiones representan cada elemento finito (partícula) en que está discretizada la geometría. Cada elemento representa las deformaciones de cada partícula del continuo en las direcciones de las coordenadas  $X_i$  (para  $i=1,2,3$ ).

### 3.8 Metodología de pasos para el cálculo de las deformaciones.

En los siguientes epígrafes se muestran los resultados de la modelación aplicando las ecuaciones que rigen la mecánica del medio continuo para el cálculo de las deformaciones en el proceso endurecimiento por deformación plástica superficial (bruñido con rodillo simple).

La secuencia de pasos para el cálculo es la siguiente:

- Entrada de datos generales.
- a) Si se cumple la condicional para la configuración de referencia.
  - Generación en coordenadas cilíndricas de los puntos (nodos) que conforman la geometría a discretizar (cuerpo de revolución).
  - Generación de la matriz elementos según los puntos (nodos) que conforman cada uno de los elementos.

- Representación de la malla.
  - Cálculo de las funciones de interpolación de cada punto (nodo).
  - Transformación de coordenadas cilíndricas de cada elemento a coordenadas rectangulares (globales).
  - Transformación por elementos de cada punto (nodo) de coordenadas globales a coordenadas naturales (locales).
  - Generación de la matriz de derivadas de las funciones de interpolación y evaluación de estas para el cálculo de la matriz *Jacobiana*.
  - Cálculo de la matriz *Jacobiana* para la configuración de referencia.
- b) Si no se cumple la condicional para la configuración de referencia y se cumple para la configuración deformada se procede a la misma secuencia de pasos que en la configuración de referencia.
- c) Seguidamente a partir de las matriz *Jacobiana* para las configuraciones de referencia y deformada se calcula el *Tensor Deformación* que describe en sus componentes las deformaciones ocurridas.
- d) En caso de no cumplirse ninguna de las dos condicionales se genera automáticamente un error.

**Nota:** Como la secuencia de pasos es la misma para las dos configuraciones hasta obtener la matriz *Jacobiana*, se muestran los resultados en cada paso para cada una de ellas, no siendo esta la secuencia en que calcula el Software, sino la presentada anteriormente.

### 3.9 Modelación matemática del bruñido por rodillo simple.

#### 3.9.1 Entrada de los datos generales para la modelación.

Diámetro (mm)	Long. (mm)	Ángulo (°)	Var. Lógicas	Paso
<b>Rmin</b> = 17.424	<b>Zmin</b> = 0.	<b>Amax</b> = 90 °	<b>Xi</b>	<b>An</b> =15
<b>Rmax</b> = 17.5	<b>Zmax</b> = 15	<b>Amin</b> = 0 °	<b>Eta</b>	<b>Zn</b> = 15
<b>rmax</b> = 17.474			<b>Dzeta</b>	<b>Rn</b> = 1

#### 3.9.2 Generación de la matriz de puntos (nodos) en coordenadas cilíndricas para las configuraciones de referencia y deformada.

La generación de los puntos que discretizan la geometría del cuerpo de revolución para formar parte de la malla se realiza en coordenadas cilíndricas. En las siguientes (**Tabla 3.1 y Tabla 3.2**) se muestran los valores de las componentes  $(\rho, \theta, z)$  de cada punto (nodo) (desde 1,...,2883) para la configuración de referencia y deformada respectivamente:

No. Nodo	Puntos en coordenadas cilíndricas.		
	$\rho$	$\theta$	$z$
1	17.424	0	0
2	17.424	0	0.5
⋮	⋮	⋮	⋮
2883	1.0716e-015	17.5	15

**Tabla 3.1.** Componentes de los puntos en coordenadas cilíndricas (configuración de referencia).

No. Nodo	Puntos en coordenadas cilíndricas.		
	$\rho$	$\theta$	$z$
1	17.424	0	0
2	17.424	0	0.5
⋮	⋮	⋮	⋮
2883	1.07e-015	17.474	15

**Tabla 3.2.** Componentes de los puntos en coordenadas cilíndricas (configuración deformada).

### 3.9.3 Generación de la matriz de elementos para las configuraciones de referencia y deformada.

La matriz de elementos se presenta en la (**Tabla 3.3** y **Tabla 3.4**), muestra el número de nodos que conforman cada elemento en el cuarto de la probeta que es discretizada para las configuraciones de referencia y deformada. El número total de elementos para el cálculo de las deformaciones ocurridas en el cuarto de la probeta analizado es de 225 para ambas configuraciones.

No. Elem.	Puntos (Nodos) que conforma el elemento									
	1	1	2	3	32	33	34	...	1985	1986
2	3	4	5	34	35	36	...	1987	1988	1989
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
225	897	898	899	928	929	930	...	2881	2882	2883

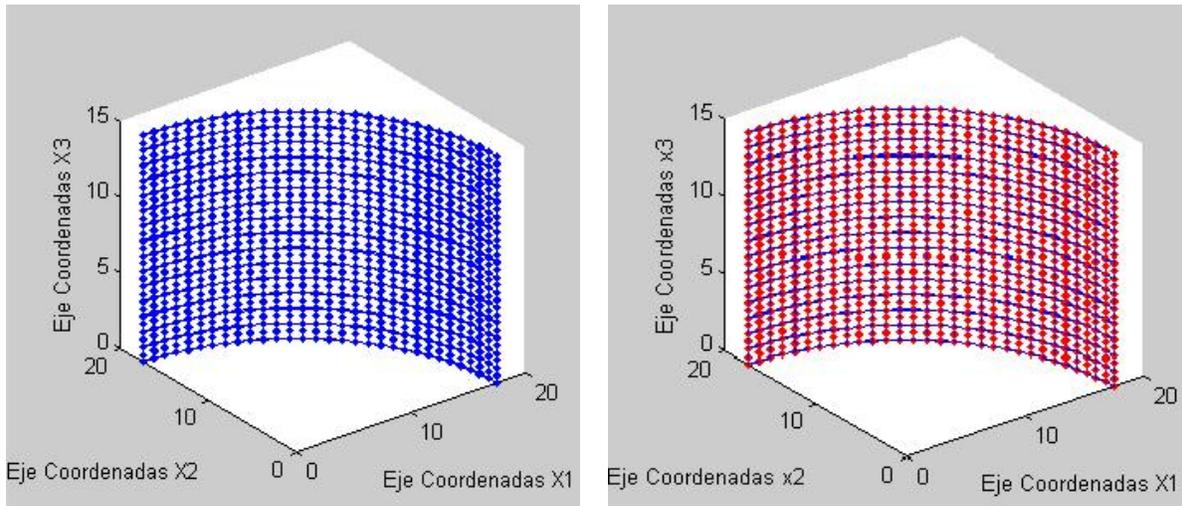
**Tabla 3.3.** Número de puntos (nodos) de los elementos de la configuración de referencia.

No. Elem.	Puntos (Nodos) que conforma el elemento									
1	1	2	3	32	33	34	...	1985	1986	1987
2	3	4	5	34	35	36	...	1987	1988	1989
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
225	897	898	899	928	929	930	...	2881	2882	2883

**Tabla 3.4** Número de puntos (nodos) de los elementos de la configuración deformada

**3.9.4 Representación del mallado hexaédrico para las configuraciones de referencia y deformada.**

En la (**Figura 3.6**) se representa el mallado a partir de los puntos y elementos hexaédricos de la geometría discretizada (cuarto de probeta) para las configuraciones de referencia y deformada.



Configuración de Referencia.

Configuración Deformada.

**Figura 3.6.** Representación del mallado.

### 3.9.5 Funciones de interpolación.

El cálculo de las funciones de interpolación se realiza para las dos configuraciones (referencia y deformada) en el siguiente orden:

- Cálculo de las funciones de interpolación en dirección del eje de coordenadas naturales  $\xi$ .
- Cálculo de las funciones de interpolación en dirección del eje de coordenadas naturales  $\eta$ .
- Cálculo de las funciones de interpolación en dirección del eje de coordenadas naturales  $\zeta$ .
- Obtención de las funciones de interpolación para cada punto (nodo) multiplicando las funciones calculadas en las direcciones de los ejes de coordenadas naturales  $(\xi, \eta, \zeta)$ .

### 3.9.6 Transformación de las coordenadas cilíndricas a coordenadas rectangulares (globales) de las configuraciones de referencia y deformada.

Las coordenadas rectangulares (globales) obtenidas por la transformación desde las coordenadas cilíndricas son almacenadas en un tensor para cada configuración (referencia y deformada), donde cada dimensión de este representa las componentes de todos los puntos (nodos) que conforman cada elemento, coincidiendo el número de la dimensión con el número de elemento que representa este para la geometría discretizada.





**3.9.7 Transformación de las coordenadas rectangulares (globales) a coordenadas naturales (locales) de las configuraciones de referencia y deformada.**

Las coordenadas naturales (locales) para ambas configuraciones (referencia y deformada) son obtenidas desde las coordenadas rectangulares (globales) en dependencia de la configuración en análisis, estas son almacenadas en un tensor donde cada dimensión representa las componentes naturales (locales) de todos los puntos (nodos) que conforman cada elemento, coincidiendo el número de la dimensión con el número de elemento que representa para la geometría discretizada. El origen de coordenadas para cada elemento es el punto central (nodo 27).

No. Nodo	$\xi$	$\eta$	$\zeta$
1	-0.06187898643630	-0.91190170157705	-0.500000000000000
2	-0.06187898643630	-0.91190170157705	0
	$\vdots$	$\vdots$	$\vdots$
27	0.01401685820504	0.91587923425152	0.500000000000000

**Tabla 3.7** Coordenadas naturales (locales) para la configuración de referencia. (Dimensión 1 del tensor).

**Nota:** Solo se ofrecen los valores del primer elemento debido a que todos los nodos de cada elemento para cada configuración (referencia y deformada) tienen igual valor en sus tres componentes  $(\xi, \eta, \zeta)$  producto a que su origen de coordenadas es el nodo central interior.

No. Nodo	$\xi$	$\eta$	$\zeta$
1	-0.04887898643630	-0.91190170157705	-0.500000000000000
2	-0.04887898643630	-0.91190170157705	0
	$\vdots$	$\vdots$	$\vdots$
27	0.00105249030143	0.91451849938920	0.500000000000000

**Tabla 3.8** Coordenadas naturales (locales) para la configuración deformada. (Dimensión 1 del tensor).

**3.9.8 Generación de la matriz de derivadas de las funciones de interpolación y evaluación de esta para el cálculo de la matriz Jacobiana de cada elemento finito.**

La derivadas de la funciones de interpolación en coordenadas naturales son evaluadas y almacenadas en un tensor de dimensión igual al número de elementos que conforman la geometría discretizada. Cada dimensión de este tensor coincide con el número de elemento finito de la geometría discretizada (cuarto del cuerpo de revolución). En la (**Tabla 3.9** y **Tabla 3.10**) se muestran los valores obtenidos (solo para la primera dimensión del tensor donde son almacenados) de la evaluación de las derivadas de las funciones de interpolación para cada punto (nodo) para la configuración de referencia y deformada.

No. Nodo	$\frac{\partial N_1}{\partial \xi}$	$\frac{\partial N_2}{\partial \xi}$	...	$\frac{\partial N_{27}}{\partial \xi}$
1	-0.10938929056595	0	...	0
	$\frac{\partial N_1}{\partial \eta}$	$\frac{\partial N_2}{\partial \eta}$	...	$\frac{\partial N_{27}}{\partial \eta}$
2	0.87715173816210	0	...	0
	$\frac{\partial N_1}{\partial \zeta}$	$\frac{\partial N_2}{\partial \zeta}$	...	$\frac{\partial N_{27}}{\partial \zeta}$
27	-1.59916736315110	1.27933389052088	...	0

**Tabla 3.9** Derivadas de las funciones de interpolación en la configuración de referencia (dimensión uno del tensor).

No.	$\frac{\partial N_1}{\partial \xi}$	$\frac{\partial N_2}{\partial \xi}$	...	$\frac{\partial N_{27}}{\partial \xi}$
1	-0.25226480283761	0	...	0
	$\frac{\partial N_1}{\partial \eta}$	$\frac{\partial N_2}{\partial \eta}$	...	$\frac{\partial N_{27}}{\partial \eta}$
2	-2.01907080414398	0	...	0
	$\frac{\partial N_1}{\partial \zeta}$	$\frac{\partial N_2}{\partial \zeta}$	...	$\frac{\partial N_{27}}{\partial \zeta}$
27	-3.68104170966326	2.9448333677306	...	-0.00000000000000

**Tabla 3.10** Derivadas de las funciones de interpolación en la configuración de Deformada (dimensión 1 del tensor).

**3.9.9 Obtención de la matriz Jacobiana para cada elemento finito de la geometría discretizada de las configuraciones de referencia y deformada.**

La matriz *Jacobiana* de cada elemento es almacenada en un tensor con el número de dimensiones igual al número total de elementos de la geometría discretizada. Cada dimensión de este tensor coincide con el número de elemento que este representa en la geometría global donde se expresan los valores obtenidos para cada uno de ellos.

26.98767145583406	0.00000000000953	0
0.94507120270282	27.11154571321094	0
-0.00000000186265	-0.00000000000000	26.97534853668185

**Tabla 3.11.1** Matriz *Jacobiana* de la configuración de referencia (Dimensión 1 del Tensor).

26.98767145909369	0.00000000000953	0.00000000008731
0.94507120933849	27.11154571312363	-0.00000000101863
0.00000001117587	0.00000000000000	26.97534853732213

**Tabla 3.11.2** Matriz *Jacobiana* de la configuración de referencia (Dimensión 2 del Tensor).

⋮	⋮	⋮
5.69275429840807	26.27103673666716	-0.00000017136335
-26.79691486117361	8.67507207393646	2.35232591629028
0.00000005587935	0.00000042468309	26.76404156535864

**Tabla 3.11.3** Matriz *Jacobiana* de la configuración de referencia (Dimensión 225 del Tensor).

26.98767145545753	-0.00000000001552	0
0.94507118035108	27.11154571315274	0
0.00000002421439	0.00000000000000	26.97534853669559

**Tabla 3.12.1** Matriz *Jacobiana* de la configuración deformada  
(Dimensión 1 del Tensor).

26.98767145103375	-0.00000000001552	0.00000000005821
0.94507125765085	27.11154571591760	0.00000000413274
0.00000000745058	0.00000000000000	26.97534853588068

**Tabla 3.12.1** Matriz *Jacobiana* de la configuración deformada  
(Dimensión 2 del Tensor).

⋮

⋮

⋮

5.69275419404697	26.27103707194328	-0.00000044703484
-26.79691499602862	8.67507195472717	2.35407567024231
-0.00000000745058	0.00000113248825	26.76404099166393

**Tabla 3.12.1** Matriz *Jacobiana* de la configuración deformada  
(Dimensión 225 del Tensor).

### 3.9.10 Cálculo del Tensor de Deformación.

El *Tensor Deformación* es calculado a partir de la matriz *Jacobiana* obtenida para cada elemento en cada una de las configuraciones (referencia y deformada). El resultado obtenido por el *Tensor Deformación* muestra las deformaciones ocurridas en cada partícula del medio continuo, representas cada una de estas por los elementos finitos en que se discretizó la geometría (cuarto del cuerpo de revolución).

0.99999999998608	-0.00000000000092	0
-0.00000000082815	0.9999999999785	0
0.0000000096626	-0.0000000000000	1.00000000000051

**Tabla 3.13.1** Tensor Deformación (Dimensión 1 del Tensor).

0.99999999970138	-0.00000000000092	-0.00000000000108
0.00000000178656	1.00000000010305	0.00000000019097
-0.00000000013804	0.0000000000000	0.9999999994656

**Tabla 3.13.2** Tensor Deformación (Dimensión 2 del Tensor).

⋮

⋮

⋮

1.00000001072388	0.00000000617271	-0.00000001084260
-0.00000000579411	1.00000000380172	0.00006537671095
0.00000002444703	0.00000000755687	0.99999997790054

**Tabla 3.13.3** Tensor Deformación (Dimensión 225 del Tensor).

### 3.10 Interpretación de los resultados obtenidos en el tensor deformación.

Como puede observarse en los resultados obtenidos en el tensor deformación (**Tabla 3.13.1**, **Tabla 3.13.2** y **Tabla 3.13.3**) por la aplicación del modelo matemático desarrollado están en correspondencia con el análisis del fenómeno físico y los resultados esperados del mismo. Con ello se valida que la aplicación del algoritmo y formulación matemática realizada en este trabajo ofrece resultados con una

aproximación aceptable para fenómenos de endurecimiento por deformación plástica.

### **3.11 Conclusiones parciales del capítulo.**

La descripción y aplicación del algoritmo de solución para la modelación matemática del endurecimiento por deformación en este capítulo se arriba a las siguientes conclusiones:

1. Se ha realizado la descripción del algoritmo de solución (paso a paso) confeccionado por el autor para la modelación matemática del proceso de endurecimiento por deformación plástica superficial (rodillado).
2. Se ha realizado la modelación según la formulación matemática (teoría del medio continuo) propuesta por el autor para el cálculo de las deformaciones producidas en el proceso de bruñido por rodillo simple (rodillado).
3. Los resultados obtenidos en el cálculo de las deformaciones por la modelación matemática están en correspondencia con la interpretación del fenómeno físico.

## CONCLUSIONES

Como resultado del trabajo de investigación desarrollado se arriba a las siguientes conclusiones:

1. Con el algoritmo diseñado para la modelación matemática del proceso de endurecimiento por deformación plástica superficial (rodillado) a partir de las leyes que rigen la mecánica del medio continuo, se establece la metodología de cálculo para las deformaciones superficiales ocurridas.
2. Con la implementación del algoritmo diseñado para el cálculo de las deformaciones en el proceso de endurecimiento por deformación plástica superficial (rodillado), se obtienen valores aproximados de las deformaciones ocurridas.
3. Los valores obtenidos por la modelación matemática en las deformaciones ocurridas en el proceso de endurecimiento por deformación plástica superficial (rodillado), están en correspondencia con la interpretación del fenómeno físico.

## RECOMENDACIONES

Según los resultados obtenidos en este trabajo se recomienda:

1. Elaborar una herramienta de discretización de cuerpos para geometrías de configuraciones diversas, que se apropien al fenómeno físico de endurecimiento por deformación.
2. Continuar desarrollando el algoritmo de solución propuesto mediante la modelación matemática para el cálculo de las deformaciones ocurridas en los procesos de endurecimiento por deformación plástica hasta obtener el valor aproximado de las tensiones que se producen en los mismos.

## BIBLIOGRAFÍA

1. Aghaei, A.; Abdolhosseini, M., J.; Kazemi, M., T.; Khoei A., R.; 2009. “Stability and size-dependency of Cauchy–Born hypothesis in three-dimensional applications”. *International Journal of Solids and Structures*, 46 (9), pp. 1925-1936.
2. Ambrosi, D.; Guillou, A.; Di Martino. E., S.; 2006. “Stress-modulated remodeling of a non-homogeneous body”. *5<sup>th</sup> World Congress of Biomechanics*, Munich (Germany), 29<sup>th</sup> / July – 4<sup>th</sup> / August / 2006. Elsevier: Journal of Biomechanics.
3. Aminanda, Y.; Castanié, B.; Barrau, J.-J.; Thevenet, P., 2009. “Experimental and numerical study of compression after impact of sandwich structures with metallic skins”. *Composites Science and Technology*, 69 (1), pp.50-59.
4. Areias, P. y Matous, K.; 2009. “Finite element formulation for modeling nonlinear viscoelastic elastomers”. *Computer Methods in Applied Mechanics and Engineering*, 197 (51-52) pp. 470-4717.
5. Areias, P.; Matous, K.; 2008. “Stabilized four-node tetrahedron with nonlocal pressure for modeling hyperelastic materials”. *International Journal For Numerical Methods In Engineering*, 76 (8), pp.1185–1201.
6. Bai, Y., L.; Wang, H., Y.; Xia, M., F.; Ke, F., J.; 2008. “ Trans-scale mechanics: looking for the missing links between continuum and

- micro/nanoscale reality". *Acta Mechanica Sinica*, 24 (2), pp.111–126.
7. Belivich, M; 2008. "Non-relativistic abstract continuum mechanics and its possible physical interpretations". *Journal Physics A: Mathematical Theoretical*, 41 (4), (19 pp).
  8. Benson, D. J.; Bazilevs, Y.; De Luycker, E.; Hsu, M.-C.; Scott, M.; Hughes, T.J.R.; Belytschko, T. "A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM". *International Journal for Numerical Methods in Engineering*, 2009. John Wiley & Sons, Ltd.
  9. Braun, A., L.; Awruch, A., M.; 2008. "Geometrically non-linear analysis in elastodynamics using the eight-node Finite Element with one-point quadrature and the generalized- $\square$  method". *Latin American Journal of Solids and Structures*, 5 (1), pp. 15-45.
  10. Budd, C., J.; Williams, J., F.; 2009. "Moving mesh generation using the Parabolic Monge-Ampere equation". *SIAM Journal on Scientific Computing*, 31 (5), pp. 3438-3465.
  11. Bunin, G; 2008. "A Continuum Theory for Unstructured Mesh Generation in Two Dimensions". *Computer Aided Geometric Design*, 25 (1), pp. 14-40.
  12. Chapelle, D.; Paris, I.; 2008. "Detailed reliability assessment of triangular MITC elements for thin shells". *Computers and Structures*, 86 (23-24), pp. 2192-2202.

13. Chen, H., Y.; Yang, T.; Luo, X., D.; Ma, H., R.; 2008. “The impedance-matched reduced acoustic cloaking with realizable mass and its layered design”. *Chinese Physics Letters*, 25 (10), pp. 3696-3699.
14. Civalek, O.; Demir, C.; Akgöz, B.; 2009. “Static analysis of single walled carbon nanotubes (swcnt) based on eringen’s nonlocal elasticity theory”. *International Journal of Engineering and Applied Sciences*, 1 (2), pp. 47-56.
15. Clotet, J.; Magret, M., D.; Peña M.; 2009. “Differentiable Structure of the Set of Coaxial Stress-Strain Tensors”. *Mathematical Methods In The Applied Sciences*, 32 (14), pp. 1753-1767.
16. Constantine, P., G.; Doostan, A.; Iaccarino, G.; 2009. “A hybrid collocation/Galerkin scheme for convective heat transfer problems with stochastic boundary conditions”. *International Journal for Numerical Methods in Engineering*, 80 (6-7), pp. 868-880.
17. Cordero, A.; Hueso, J. L.; Torregrosa, J. R., 2004. *Cálculo numérico: Teoría y problemas*. Editorial. Universidad. Politécnica. Valencia. ISBN 849705539X.
18. Douglas J.; Burden, R.,L. 2002. *Numerical Methods*. 3rd. Ed. Hardcover. ISBN-10: 0534407617.
19. Drake, P., S.; Dogancay, K.; ISWPC 2008. “Some Applications of Tensor Algebra to Estimation Theory”. *Wireless Pervasive Computing*, 3<sup>rd</sup> International Symposium, pp. 106-110.

20. Du, Q.; Wang, D., 2006. “Recent progress in robust and quality Delaunay mesh generation”. *Journal of Computational and Applied Mathematics*, 195, (1-2), pp 8-23.
21. Dumbser, M.; Zanotti, O.; 2009. “Very High Order PNPM Schemes on Unstructured Meshes for the Resistive Relativistic MHD Equations”. *Journal of Computational Physics*, 228 (18), pp. 6991-7006.
22. Eshwara K.; Nahavandi, S.; Mohammed, M. H.; Aditya, V.N.; 2006a. “Prediction of Residual Stresses in Roller Burnished Components- A Finite Element Approach”. *International Journal of Applied Engineering*, 1 (2) pp. 153-163.
23. Eshwara, K.; Murali, R.; Janardhan, G. R.; Raju, A. V. S.; Nahavandi, S.; 2006b. “Experimental Investigation and Finite Element Analysis for the Study of Residual Stresses in Roller Burnished Components” *International Journal of Applied Engineering*, 1 (3) pp. 427-436.
24. Fahs, H.; 2010. “Discontinuous Galerkin Method for Time-Domain Electromagnetics on Curvilinear Domains”. *Applied Mathematical Sciences*, 4 (19), pp. 943 – 958.
25. Frey, S.; Martins, M., L.; Saldanha, R., M.; 2008. “Petrov-Galerkin approximation for advective-diffusive heat transfer in saturated porous media”. *Latin American Applied Research*, 38 (2), pp. 169-178.

26. Geuzaine, C. Remacle, J., F.; 2009. "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities". *International Journal for Numerical Methods in Engineering*, 79 (11), pp. 1309-1331.
27. Gunawan, S.; Papalambros, P., Y.; 2007. "Reliability Optimization With Mixed Continuous-Discrete Random Variables and Parameters". *Journal of Mechanical Design*, 129 (2), pp. 158-165.
28. Ha, S., Chang, F. 2010. "Optimizing a spectral element for modeling PZT-induced Lamb wave propagation in thin plates". *Smart Materials and Structures*. 2010 IOP Publishing Ltd Printed in the UK.
29. Hammoud, M.; Duhamel, D.; Sab, K., 2010. "Static and dynamic studies for coupling discrete and continuum media; Application to a simple railway track model". *International Journal of Solids and Structures*,. 47 (2), pp. 276-290.
30. Holden, M.; 2008. "A review of geometric transformations for nonrigid body registration". *Medical Imaging, IEEE Transactions on*, 27 (1), pp. 111-128.
31. Hotz, I.; Feng, L.; Hagen, H.; Hamann, B.; Joy, K.; 2006. "Tensor Field Visualization Using a Metric Interpretation". En: Weickert, J.; Hagen, H. (eds), *Visualization and Processing of Tensor Fields*. Verlag Berlin Heidelberg: Springer. Cap. 16.

32. Hou, Y.; Tian, J.; Wu, Y.; Liang, J.; He X. "A new numerical method for BLT forward problem based on high-order finite elements". *Communications in Numerical Methods in Engineering*, 2008. John Wiley & Sons, Ltd.
33. Hsieh, C., I.; Katul, G.; 2009. "The Lagrangian stochastic model for estimating footprint and water vapor fluxes over inhomogeneous surfaces". *International Journal Biometeorol* 53 (1), pp. 87-100.
34. Hu, H.; Belouettar, S.; 1 Potier-Ferry, M.; Makradi A., 2009. "A novel finite element for global and local buckling analysis of sandwich beams". *Composites Structure*, 90 (3), pp. 270-278.
35. Ingber, M., S.; 2008. "Near-contact interactions between a sphere and a plane". *Journal Physical Review E*. 77 (2), (10 pp).
36. Jaklic, G.; Kozak, J.; Krajnc, M.; Vitrih, V.; Zagar, E.; 2008. "Barycentric coordinates for Lagrange interpolation over lattices on a simplex". *Numerical Algorithms*, 48 (1-3), pp. 93-104.
37. Jiang, L.; HongBo, L.; YuanHao C.; 2009. "Simplification and normalization of indexed differentials involving coordinate transformation". *Science in China Series A: Mathematics*, 52 (10), pp. 2266-2286.
38. Kapuscik, E.; Lanczewski, T.; 2009. "Elementary and particles and fields theory on the velocity tensors". *Physics of Atomic Nuclei*, 72 (5), pp. 809-812.

39. Khoei, A. R.; Anahid, M.; Shahim, K.; DorMohammadi, H.; 2008. "Arbitrary Lagrangian–Eulerian method in plasticity of pressure-sensitive material; Application to powder forming processes". *Computational Mechanics*, 44 (6-7), pp. 401-416.
40. Khoei, A.,R.; Abdolhosseini Qomi, M.,J.; Kazemi, M.,T.; Aghaei, A.;2009. "An Investigation on the Validity of Cauchy-Born Hypothesis Using Sutton-Chen Many-Body Potential". *Computational Materials Science*, 44(3), pp. 999-1006.
41. Konyukhov, A.; Schweizerhof, K.; 2008. "On the solvability of closest point projection procedures in contact analysis: Analysis and solution strategy for surfaces of arbitrary geometry". *Computer Methods in Applied Mechanics and Engineering*, 197 (33-40), pp. 3045-3056.
42. Kumar, K.; Kumar, R.; 2010. "On Inhomogeneous Deformations in ES Materials". *International Journal of Engineering Science*, 48 (4), pp. 405-416.
43. Lee, H.; Kim, Y. -D.; Cichocki, A.; Choi, S.; 2007. "Nonnegative tensor factorization for continuous EEG classification". *International Journal of Neural Systems*, 17 (4), pp. 305-17.
44. Lili, J.; Hyung-Chun, L.; Li, T.; 2008. "Numerical simulations of the steady Navier-Stokes equations using adaptive meshing schemes". *International Journal for Numerical Methods in Fluids*, 56 (6), pp. 703 – 721.

45. Lion, A.; Yagimli, B.; Baroud, G.; Goerke, U.; 2008. "Constitutive modelling of PMMA-based bone cement: a functional model of viscoelasticity and its approximation for time domain investigations". *Archives of Mechanics*, 60 (3), pp. 221-242.
46. Livermore, P., W.; Ierley, G., R.; 2009. "A new identity linking coefficients of a certain class of homogeneous polynomials and Gauss hypergeometric functions". *Advances in Applied Mathematics*, 43 (4), pp. 390-393.
47. Long, C. S.; Loveday, P. W.; Groenwold, A. A.; 2009. "Axisymmetric solid-of-revolution finite elements with rotational degrees of freedom". *Finite Elements in Analysis and Design*, 45 (2), pp 121-131.
48. López, J. J., 1999. *Mecánica de los Medios Continuos*. Cuenca: Ediciones de la Universidad de Castilla-La Mancha (España). ISBN 84-8427-030-0.
49. Ma, S.; Liu, P.; 2006. "Modeling of the Perfectly Matched Layer Absorbing Boundaries and Intrinsic Attenuation in Explicit Finite-Element Methods". *Bulletin of the Seismological Society of America*, 96 (5), pp. 1779-1794
50. Mao, S.; Leclere, M.; Michaelides, E.; 2008. "Passive scalar flux footprint analysis over horizontally inhomogeneous plant canopy using large-eddy simulation". *Atmospheric Environment*, 42 (21), pp. 5446-5458.

51. Mase, G., T.; Smelser, R., E.; Mase, G., E.; 2009. *Continuum Mechanics for Engineers*. 3rd Ed. Hardback (USA). ISBN: 9781420085389.
52. Mello, M.; Hong, S.; Rosakis, A.,J.; 2009. “Extension of the Coherent Gradient Sensor (CGS) to the Combined Measurement of In-Plane and Out-of-Plane Displacement Field Gradients”. *Experimental Mechanics*, 49 (2), pp. 277-289.
53. Moslemi, H.; Khoei, A., R.; 2009. 3D adaptive finite element modeling of non-planar curved crack growth using the weighted superconvergent patch recovery method. *Engineering Fracture Mechanics*, 76 (11), pp. 1703-1728.
54. Nair, S.; 2009, “*Introduction to Continuum Mechanics*”, Cambridge University Press. ISBN-13 978-0-511-50718-2.
55. Nguyen, V., T.; Caron, Y., F.; 2009. “Finite element analysis of free-edge stresses in composite laminates under mechanical and thermal loading”. *Composites Science And Technology*, 69 (1) 40-49.
56. Nguyen, V., X.; Abousleiman, Y., N.; 2010. “Poromechanics Solutions to Plane Strain and Axisymmetric Mandel-Type Problems in Dual-Porosity and Dual-Permeability Medium”. *Journal of Applied Mechanics*, 77 (1), pp. 11002- 11020.
57. Ni, G.; Qi, L.; Wang, F.; Wang Y.; 2007. “ The degree of the E-characteristic polynomial of an even order tensor”. *Journal of Mathematical Analysis And Applications*. 329 (2), pp. 1218–1229.

58. Ni, G.; Wang, Y.; 2007. "On the best rank-1 approximation to higher-order symmetric tensors". *Mathematical And Computer Modelling*, 46 (9-10), pp. 1345–1352.
59. Occorsio, D.; 2009. "Extended Lagrange interpolation in weighted uniform norm". *Applied Mathematics and Computation*, 211 (1), pp.10–22.
60. Okafor, E., N., C.; Okon, P., E.; Okoro, C.C.; 2009. "Magnetic Field Mapping of a Direct Current Electrical Machine Using Finite Element Method". *Journal of Applied Sciences Research*, 5 (11), pp. 1889-1898.
61. Oliver, X.; Agelet, C.; 2002. *Mecánica de medios continuos para ingenieros*, 2 da. Ed. (Barcelona): Ediciones UPC. ISBN: 84-95999-06-4.
62. Paluszny, A.; Matthai, S., K.; Hohmeyer, M.; 2007. "Hybrid finite element–finite volume discretization of complex geologic structures and a new simulation workflow demonstrated on fractured rocks". *Geofluids*, 7 (2), pp. 186 - 208.
63. Peng, H.; Meng, G.; Li, F. "Modeling of wave propagation in plate structures using three-dimensional spectral element method for damage detection". *Journal of Sound and Vibration* Volume 320, Issues 4-5, 6 March 2009, Pages 942-954 Elsevier.
64. Qi, L.; 2007. "Eigenvalues and invariants of tensors". *Journal Mathematical Analysis And Applications* 325 (2) 1363–1377.

65. Reddy, J., N.; 2008. *An Introduction to Continuum Mechanics*, Cambridge (UK): Cambridge University Press. ISBN 978-0-511-48036-2.
66. Scherer, M.; Denzer, R.; Steinmann, P.; 2007. “Energy-based r-adaptivity: a solution strategy and applications to fracture mechanics”. *International Journal of Fracture*,147 (1-4), pp. 117–132.
67. Scherer, M.; Denzer, R.; Steinmann, P.; 2008. “On a solution strategy for energy-based mesh optimization in finite hyperelastostatics”. *Computer Methods Applied Mechanics and Engineering* 197 (6-8), pp. 609–622.
68. Sergeyev, Y.,D., 2007. “Blinking fractals and their quantitative analysis using infinite and infinitesimal numbers”, *Chaos, Solitons & Fractals*, 33, (1), pp. 50–75.
69. Sevilla, R.; Fernández, S.; Huerta, A.; 2008. “NURBS-Enhanced Finite Element Method for Euler equations”. *International Journal for Numerical Methods in Fluids*, 57 (9), pp. 1051 - 1069.
70. Soares, D.; Rodrigues, G.,G.; Gonçalves, K., A.; 2010. “An efficient multi-time-step implicit–explicit method to analyze solid–fluid coupled systems discretized by unconditionally stable time-domain finite element procedures”. *Computers & Structures*, 88 (5-6), pp. 387-394.

71. Tua, X.; Andrade, J., E.; Chena, Q.; 2009. “Return mapping for nonsmooth and multiscale elastoplasticity”. *Computer Methods in Applied Mechanics and Engineering*, 198 (30-32), pp. 2286-2296.
72. Vashisht, A., K.; 2008. “Ingenious techniques for irrigation sustainability in Himalayan and Shiwalik foothill regions”. *Current Science*, 95 (12), pp. 1688-1693.
73. Veyhl, C.; Velova, I., V.; Murch, G., E.; Ochsner, A.; Fiedler, T.; 2010. “On the mesh dependence of non-linear mechanical finite element analysis”. *Finite Elements in Analysis and Design*, 46 (5), pp. 371–378.
74. Volokh, K., Y.; 2008. “ Prediction of arterial failure based on a microstructural bi-layer fiber–matrix model with softening”. *Journal of Biomechanics*, 41(2), pp. 447- 453.
75. Wriggers P; 2001. *Nichtlineare finite-element-methoden*. Springer-Verlag, Berlin Heidelberg. (New York) ISBN: 3-540-67747-X.
76. Wriggers P; 2006. *Computational Contact Mechanics*, 2da. Ed. Springer-Verlag Berlin Heidelberg. (New York) ISBN 3-540-32608-3.
77. Xu, A.; Cen, Z.; 2009. “A remainder formula of numerical differentiation for the generalized Lagrange interpolation”. *Journal of Computational and Applied Mathematics*, 230 (2), pp. 418 - 423.
78. Xu, M.; Belytschko, T.; 2008. “Conservation properties of the bridging domain method for coupled molecular/continuum

- dynamics”. *International Journal For Numerical Methods In Engineering*, 76 (3), pp. 278–294.
79. Zhang, B.; Zhang, J., 2008. “A load reconstruction model for advanced grid-stiffened composite plates”. *Composites Structures*, 82, (4), pp. 600-608.
80. Zienkiewicz, O., C.; Taylor, R., L.; Zhu, J., Z.; Nithiarasu, P.(2010); *El Método de los Elementos Finitos*. 6th Ed. CIMNE. ISBN 978-84-96736-74-0.
81. Zienkiewicz, O.C.; Taylor, R.L. 2005. *The element method for solid and structural mechanics*. 6th Ed. Oxford (UK): Elsevier. ISBN 0-7506-6321-9.

## ANEXO 1 Funciones de interpolación para el hexaedro de 27 nodos.

*Nodos esquina*

$$\begin{aligned}
N_1(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 - \xi)(\eta^2 - \eta)(\zeta^2 - \zeta) & N_2(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 + \xi)(\eta^2 - \eta)(\zeta^2 - \zeta) \\
N_3(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 + \xi)(\eta^2 + \eta)(\zeta^2 - \zeta) & N_4(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 - \xi)(\eta^2 + \eta)(\zeta^2 - \zeta) \\
N_5(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 - \xi)(\eta^2 - \eta)(\zeta^2 + \zeta) & N_6(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 + \xi)(\eta^2 - \eta)(\zeta^2 + \zeta) \\
N_7(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 + \xi)(\eta^2 + \eta)(\zeta^2 + \zeta) & N_8(\xi, \eta, \zeta) &= \frac{1}{8}(\xi^2 - \xi)(\eta^2 + \eta)(\zeta^2 + \zeta)
\end{aligned}$$

*Nodos laterales*

$$\begin{aligned}
N_9(\xi, \eta, \zeta) &= \frac{1}{4}(\eta^2 - \eta)(\zeta^2 - \zeta)(1 - \xi^2) & N_{10}(\xi, \eta, \zeta) &= \frac{1}{4}(\zeta^2 - \zeta)(\xi^2 + \xi)(1 - \eta^2) \\
N_{11}(\xi, \eta, \zeta) &= \frac{1}{4}(\eta^2 + \eta)(\zeta^2 - \zeta)(1 - \xi^2) & N_{12}(\xi, \eta, \zeta) &= \frac{1}{4}(\zeta^2 - \zeta)(\xi^2 - \xi)(1 - \eta^2) \\
N_{13}(\xi, \eta, \zeta) &= \frac{1}{4}(\xi^2 - \xi)(\eta^2 - \eta)(1 - \zeta^2) & N_{14}(\xi, \eta, \zeta) &= \frac{1}{4}(\xi^2 + \xi)(\eta^2 - \eta)(1 - \zeta^2) \\
N_{15}(\xi, \eta, \zeta) &= \frac{1}{4}(\xi^2 + \xi)(\eta^2 + \eta)(1 - \zeta^2) & N_{16}(\xi, \eta, \zeta) &= \frac{1}{4}(\xi^2 - \xi)(\eta^2 + \eta)(1 - \zeta^2) \\
N_{17}(\xi, \eta, \zeta) &= \frac{1}{4}(\eta^2 - \eta)(\zeta^2 + \zeta)(1 - \xi^2) & N_{18}(\xi, \eta, \zeta) &= \frac{1}{4}(\zeta^2 + \zeta)(\xi^2 + \xi)(1 - \eta^2) \\
N_{19}(\xi, \eta, \zeta) &= \frac{1}{4}(\eta^2 + \eta)(\zeta^2 + \zeta)(1 - \xi^2) & N_{20}(\xi, \eta, \zeta) &= \frac{1}{4}(\zeta^2 + \zeta)(\xi^2 - \xi)(1 - \eta^2)
\end{aligned}$$

*Nodos en el centro de las caras*

$$\begin{aligned}
N_{21}(\xi, \eta, \zeta) &= \frac{1}{2}(\xi^2 - \xi)(1 - \eta^2)(1 - \zeta^2) & N_{22}(\xi, \eta, \zeta) &= \frac{1}{2}(\zeta^2 - \zeta)(1 - \xi^2)(1 - \eta^2) \\
N_{23}(\xi, \eta, \zeta) &= \frac{1}{2}(\xi^2 + \xi)(1 - \eta^2)(1 - \zeta^2) & N_{24}(\xi, \eta, \zeta) &= \frac{1}{2}(\eta^2 + \eta)(1 - \xi^2)(1 - \zeta^2) \\
N_{25}(\xi, \eta, \zeta) &= \frac{1}{2}(\zeta^2 + \zeta)(1 - \xi^2)(1 - \eta^2) & N_{26}(\xi, \eta, \zeta) &= \frac{1}{2}(\eta^2 - \eta)(1 - \xi^2)(1 - \zeta^2)
\end{aligned}$$

*Nodo central interior*

$$N_{27}(\xi, \eta, \zeta) = (1 - \xi^2)(1 - \eta^2)(1 - \zeta^2)$$

## ANEXO 2 Códigos de MATLAB para la Modelación Matemática.

### Datos.m

```
% Datos
Rmin = 17.424; % Radio mínimo
Rmax = 17.5; % Radio Máximo de la configuración inicial
Rn = 1; % Cantidad de pasos en dirección radial
Amin = 0; % Ángulo mínimo
Amax = 90; % Ángulo máxima
An = 15; % Cantidad de pasos en direccion angular
Zmin = 0; % Altura mínima
Zmax = 15; % Altura máxima configuración de referencia
Zn = 15; % Cantidad de pasos en dirección vertical
rmax=17.474; % Radio Máximo de la configuración deformada

Rs = (Rmax - Rmin)./Rn; % Cálculo del paso en dirección radial de la
% configuración inicial
As = (Amax - Amin)./An; % Cálculo del paso en dirección angular
Zs = (Zmax - Zmin)./Zn; % Cálculo del paso en dirección vertical
(altura) conf. Ref.
rs = (rmax - Rmin)./Rn; % Cálculo del paso en dirección radial de la
% configuración deformada
syms Xi Eta Dzeta % Variables Lógicas
```

### puntoscoord.m

```
function [P]= puntoscoord(Rmin,Rs,Rmax,Zmin,Zs,Zmax);
% P -- Puntos para la configuración de referencia
% Fi -- ángulo con respecto al eje x1
% R -- radio vector
Datos;
P = []; % Puntos
for R = Rmin:Rs./2:Rmax,
    for Fi = Amin:As./2:Amax,
        for x3 = Zmin:Zs./2:Zmax,
            x1 = R.*cos(Fi.*pi./180);
            x2 = R.*sin(Fi.*pi./180);
            P = [P; x1 x2 x3];
        end
    end
end
```

**NumeroElementos.m**

```

function E=NumeroElementos(P);
% E - Matriz de elementos *configuración de referencia)
Datos;
E = []; % Listado de todos los elementos
for I = 0:Rn-1,
    for J = 0:An-1,
        for K = 0:Zn-1,
            e = []; % coordenadas de cada elemento
            for L = 0:0.5:1,
                for M = 0:0.5:1,
                    for N = 0:0.5:1,
                        R = Rmin + Rs.*(I + L);
                        A = Amin + As.*(J + M);
                        x3 = Zmin + Zs.*(K + N);
                        x1 = R.*cos(A.*pi./180);
                        x2 = R.*sin(A.*pi./180);
                        for Q = 1:size(P,1),
                            if apcomp(P(Q,1), x1) & apcomp(P(Q,2), x2) &
apcomp(P(Q,3), x3),
                                e = [e, Q];
                                break;
                            end
                        end
                    end
                end
            end
        end
    end
    E = [E; e];
end
end
end

```

**plotprobeta.m**

```

function plotprobeta(E,P);
plot3(P(:,1),P(:,2),P(:,3),'b. '),hold on
for I = 1:size(E,1),
    plotq27(E(I,:),P);
end;
xlabel ( 'Eje Coordenadas X1')
ylabel ( 'Eje Coordenadas X2')
zlabel ( 'Eje Coordenadas X3')

```

**funforXi.m**

```

function NX1=funforXi(Xi,E,P);
% Cálculo de los polinomios de Lagrange en dirección
% del eje de coordenadas local Xi (configuración de referencia)
for i=1:size(E,1)
Xi1=P(E(i,1),1);Xi2=P(E(i,2),1);Xi3=P(E(i,3),1);Xi4=P(E(i,4),1);
Xi5=P(E(i,5),1);Xi6=P(E(i,6),1);Xi7=P(E(i,7),1);Xi8=P(E(i,8),1);
Xi9=P(E(i,9),1);Xi10=P(E(i,10),1);Xi11=P(E(i,11),1);Xi12=P(E(i,12),1);
Xi13=P(E(i,13),1);Xi14=P(E(i,14),1);Xi15=P(E(i,15),1);Xi16=P(E(i,16),1);
Xi17=P(E(i,17),1);Xi18=P(E(i,18),1);Xi19=P(E(i,19),1);Xi20=P(E(i,20),1);
Xi21=P(E(i,21),1);Xi22=P(E(i,22),1);Xi23=P(E(i,23),1);Xi24=P(E(i,24),1);
Xi25=P(E(i,25),1);Xi26=P(E(i,26),1);Xi27=P(E(i,27),1);

NX1(1,1,i)=(Xi-Xi10).*(Xi-Xi19)./((Xi1-Xi10).*(Xi1-Xi19));
NX1(2,1,i)=(Xi-Xi11).*(Xi-Xi20)./((Xi2-Xi11).*(Xi2-Xi20));
NX1(3,1,i)=(Xi-Xi12).*(Xi-Xi21)./((Xi3-Xi12).*(Xi3-Xi21));
NX1(4,1,i)=(Xi-Xi13).*(Xi-Xi22)./((Xi4-Xi13).*(Xi5-Xi22));
NX1(5,1,i)=(Xi-Xi14).*(Xi-Xi23)./((Xi5-Xi14).*(Xi5-Xi23));
NX1(6,1,i)=(Xi-Xi15).*(Xi-Xi24)./((Xi6-Xi15).*(Xi6-Xi24));
NX1(7,1,i)=(Xi-Xi16).*(Xi-Xi25)./((Xi7-Xi16).*(Xi7-Xi25));
NX1(8,1,i)=(Xi-Xi17).*(Xi-Xi26)./((Xi8-Xi17).*(Xi8-Xi26));
NX1(9,1,i)=(Xi-Xi18).*(Xi-Xi27)./((Xi9-Xi18).*(Xi9-Xi27));
NX1(10,1,i)=(Xi-Xi1).*(Xi-Xi19)./((Xi10-Xi1).*(Xi10-Xi19));
NX1(11,1,i)=(Xi-Xi2).*(Xi-Xi20)./((Xi11-Xi2).*(Xi11-Xi20));
NX1(12,1,i)=(Xi-Xi3).*(Xi-Xi21)./((Xi12-Xi3).*(Xi12-Xi21));
NX1(13,1,i)=(Xi-Xi4).*(Xi-Xi22)./((Xi13-Xi4).*(Xi13-Xi22));
NX1(14,1,i)=(Xi-Xi5).*(Xi-Xi23)./((Xi14-Xi5).*(Xi14-Xi23));
NX1(15,1,i)=(Xi-Xi6).*(Xi-Xi24)./((Xi15-Xi6).*(Xi15-Xi24));
NX1(16,1,i)=(Xi-Xi7).*(Xi-Xi25)./((Xi16-Xi7).*(Xi16-Xi25));
NX1(17,1,i)=(Xi-Xi8).*(Xi-Xi26)./((Xi17-Xi8).*(Xi17-Xi26));
NX1(18,1,i)=(Xi-Xi9).*(Xi-Xi27)./((Xi18-Xi9).*(Xi18-Xi27));
NX1(19,1,i)=(Xi-Xi10).*(Xi-Xi11)./((Xi19-Xi10).*(Xi19-Xi11));
NX1(20,1,i)=(Xi-Xi2).*(Xi-Xi11)./((Xi20-Xi2).*(Xi20-Xi11));
NX1(21,1,i)=(Xi-Xi3).*(Xi-Xi12)./((Xi21-Xi3).*(Xi21-Xi12));
NX1(22,1,i)=(Xi-Xi4).*(Xi-Xi13)./((Xi22-Xi4).*(Xi22-Xi13));
NX1(23,1,i)=(Xi-Xi5).*(Xi-Xi14)./((Xi23-Xi5).*(Xi23-Xi14));
NX1(24,1,i)=(Xi-Xi6).*(Xi-Xi15)./((Xi24-Xi6).*(Xi24-Xi15));
NX1(25,1,i)=(Xi-Xi7).*(Xi-Xi16)./((Xi25-Xi7).*(Xi25-Xi16));
NX1(26,1,i)=(Xi-Xi8).*(Xi-Xi17)./((Xi26-Xi8).*(Xi26-Xi17));
NX1(27,1,i)=(Xi-Xi9).*(Xi-Xi18)./((Xi27-Xi9).*(Xi27-Xi18));
expand (NX1);
end

```

**funforEta.m**

```

function NX2=funforEta(Eta,E,P);
% Cálculo de los polinomios de Lagrange en dirección
% del eje de coordenadas local Eta (configuración de referencia)
for i=1:size(E,1)
Eta1=P(E(i,1),2);Eta2=P(E(i,2),2);Eta3=P(E(i,3),2);Eta4=P(E(i,4),2);
Eta5=P(E(i,5),2);Eta6=P(E(i,6),2);Eta7=P(E(i,7),2);Eta8=P(E(i,8),2);
Eta9=P(E(i,9),2);Eta10=P(E(i,10),2);Eta11=P(E(i,11),2);Eta12=P(E(i,12),2);
Eta13=P(E(i,13),2);Eta14=P(E(i,14),2);Eta15=P(E(i,15),2);Eta16=P(E(i,16),2);
Eta17=P(E(i,17),2);Eta18=P(E(i,18),2);Eta19=P(E(i,19),2);Eta20=P(E(i,20),2);
Eta21=P(E(i,21),2);Eta22=P(E(i,22),2);Eta23=P(E(i,23),2);Eta24=P(E(i,24),2);
Eta25=P(E(i,25),2);Eta26=P(E(i,26),2);Eta27=P(E(i,27),2);

NX2(1,1,i)=((Eta-Eta4).*(Eta-Eta7))./((Eta1-Eta4).*(Eta1-Eta7));
NX2(2,1,i)=((Eta-Eta5).*(Eta-Eta8))./((Eta2-Eta5).*(Eta2-Eta8));
NX2(3,1,i)=((Eta-Eta6).*(Eta-Eta9))./((Eta3-Eta6).*(Eta3-Eta9));
NX2(4,1,i)=((Eta-Eta1).*(Eta-Eta7))./((Eta4-Eta1).*(Eta4-Eta7));
NX2(5,1,i)=((Eta-Eta8).*(Eta-Eta2))./((Eta5-Eta8).*(Eta5-Eta2));
NX2(6,1,i)=((Eta-Eta3).*(Eta-Eta9))./((Eta6-Eta3).*(Eta6-Eta9));
NX2(7,1,i)=((Eta-Eta4).*(Eta-Eta1))./((Eta7-Eta4).*(Eta7-Eta1));
NX2(8,1,i)=((Eta-Eta2).*(Eta-Eta5))./((Eta8-Eta2).*(Eta8-Eta5));
NX2(9,1,i)=((Eta-Eta3).*(Eta-Eta6))./((Eta9-Eta3).*(Eta9-Eta6));
NX2(10,1,i)=((Eta-Eta13).*(Eta-Eta16))./((Eta10-Eta13).*(Eta10-Eta16));
NX2(11,1,i)=((Eta-Eta14).*(Eta-Eta17))./((Eta11-Eta14).*(Eta11-Eta17));
NX2(12,1,i)=((Eta-Eta15).*(Eta-Eta18))./((Eta12-Eta15).*(Eta12-Eta18));
NX2(13,1,i)=((Eta-Eta10).*(Eta-Eta16))./((Eta13-Eta10).*(Eta13-Eta16));
NX2(14,1,i)=((Eta-Eta11).*(Eta-Eta17))./((Eta14-Eta11).*(Eta14-Eta17));
NX2(15,1,i)=((Eta-Eta12).*(Eta-Eta18))./((Eta15-Eta12).*(Eta15-Eta18));
NX2(16,1,i)=((Eta-Eta10).*(Eta-Eta13))./((Eta16-Eta10).*(Eta16-Eta13));
NX2(17,1,i)=((Eta-Eta11).*(Eta-Eta14))./((Eta17-Eta11).*(Eta17-Eta14));
NX2(18,1,i)=((Eta-Eta12).*(Eta-Eta15))./((Eta18-Eta12).*(Eta18-Eta15));
NX2(19,1,i)=((Eta-Eta22).*(Eta-Eta25))./((Eta19-Eta22).*(Eta19-Eta25));
NX2(20,1,i)=((Eta-Eta23).*(Eta-Eta26))./((Eta20-Eta23).*(Eta20-Eta26));
NX2(21,1,i)=((Eta-Eta24).*(Eta-Eta27))./((Eta21-Eta24).*(Eta21-Eta27));
NX2(22,1,i)=((Eta-Eta19).*(Eta-Eta25))./((Eta22-Eta19).*(Eta22-Eta25));
NX2(23,1,i)=((Eta-Eta20).*(Eta-Eta26))./((Eta23-Eta20).*(Eta23-Eta26));
NX2(24,1,i)=((Eta-Eta21).*(Eta-Eta27))./((Eta24-Eta21).*(Eta24-Eta27));
NX2(25,1,i)=((Eta-Eta19).*(Eta-Eta22))./((Eta25-Eta19).*(Eta25-Eta22));
NX2(26,1,i)=((Eta-Eta20).*(Eta-Eta23))./((Eta26-Eta20).*(Eta26-Eta23));
NX2(27,1,i)=((Eta-Eta21).*(Eta-Eta24))./((Eta27-Eta21).*(Eta27-Eta24));
expand (NX2);
end

```

**funforDzeta.m**

```

function NX3=funforDzeta(Dzeta,E,P);
% Cálculo de los polinomios de Lagrange en dirección
% del eje de coordenadas local Dzeta (Configuración de referencia)
for i=1:size(E,1)
Dzeta1=P(E(i,1),3);Dzeta2=P(E(i,2),3);Dzeta3=P(E(i,3),3);Dzeta4=P(E(i,4),3);
Dzeta5=P(E(i,5),3);Dzeta6=P(E(i,6),3);Dzeta7=P(E(i,7),3);Dzeta8=P(E(i,8),3);
Dzeta9=P(E(i,9),3);Dzeta10=P(E(i,10),3);Dzeta11=P(E(i,11),3);Dzeta12=P(E(i,12),3);
Dzeta13=P(E(i,13),3);Dzeta14=P(E(i,14),3);Dzeta15=P(E(i,15),3);
Dzeta16=P(E(i,16),3);Dzeta17=P(E(i,17),3);Dzeta18=P(E(i,18),3);
Dzeta19=P(E(i,19),3);Dzeta20=P(E(i,20),3);Dzeta21=P(E(i,21),3);
Dzeta22=P(E(i,22),3);Dzeta23=P(E(i,23),3);Dzeta24=P(E(i,24),3);
Dzeta25=P(E(i,25),3);Dzeta26=P(E(i,26),3);Dzeta27=P(E(i,27),3);

NX3(1,1,i)=(Dzeta-Dzeta2).*(Dzeta-Dzeta3)/((Dzeta1-Dzeta2).*(Dzeta1-Dzeta3));
NX3(2,1,i)=(Dzeta-Dzeta1).*(Dzeta-Dzeta3)/((Dzeta2-Dzeta1).*(Dzeta2-Dzeta3));
NX3(3,1,i)=(Dzeta-Dzeta1).*(Dzeta-Dzeta2)/((Dzeta3-Dzeta1).*(Dzeta3-Dzeta2));
NX3(4,1,i)=(Dzeta-Dzeta5).*(Dzeta-Dzeta6)/((Dzeta4-Dzeta5).*(Dzeta4-Dzeta6));
NX3(5,1,i)=(Dzeta-Dzeta4).*(Dzeta-Dzeta6)/((Dzeta5-Dzeta4).*(Dzeta5-Dzeta6));
NX3(6,1,i)=(Dzeta-Dzeta4).*(Dzeta-Dzeta5)/((Dzeta6-Dzeta4).*(Dzeta6-Dzeta5));
NX3(7,1,i)=(Dzeta-Dzeta8).*(Dzeta-Dzeta9)/((Dzeta7-Dzeta5).*(Dzeta7-Dzeta9));
NX3(8,1,i)=(Dzeta-Dzeta7).*(Dzeta-Dzeta9)/((Dzeta8-Dzeta7).*(Dzeta8-Dzeta9));
NX3(9,1,i)=(Dzeta-Dzeta7).*(Dzeta-Dzeta8)/((Dzeta9-Dzeta7).*(Dzeta9-Dzeta8));
NX3(10,1,i)=(Dzeta-Dzeta11).*(Dzeta-Dzeta12)/((Dzeta10-Dzeta11).*(Dzeta10-Dzeta12));
NX3(11,1,i)=(Dzeta-Dzeta10).*(Dzeta-Dzeta12)/((Dzeta11-Dzeta10).*(Dzeta11-Dzeta12));
NX3(12,1,i)=(Dzeta-Dzeta10).*(Dzeta-Dzeta11)/((Dzeta12-Dzeta10).*(Dzeta12-Dzeta11));
NX3(13,1,i)=(Dzeta-Dzeta14).*(Dzeta-Dzeta15)/((Dzeta13-Dzeta14).*(Dzeta13-Dzeta15));
NX3(14,1,i)=(Dzeta-Dzeta13).*(Dzeta-Dzeta15)/((Dzeta14-Dzeta13).*(Dzeta14-Dzeta15));
NX3(15,1,i)=(Dzeta-Dzeta13).*(Dzeta-Dzeta14)/((Dzeta15-Dzeta13).*(Dzeta15-Dzeta14));
NX3(16,1,i)=(Dzeta-Dzeta17).*(Dzeta-Dzeta18)/((Dzeta16-Dzeta17).*(Dzeta16-Dzeta18));
NX3(17,1,i)=(Dzeta-Dzeta16).*(Dzeta-Dzeta18)/((Dzeta17-Dzeta16).*(Dzeta17-Dzeta18));
NX3(18,1,i)=(Dzeta-Dzeta16).*(Dzeta-Dzeta17)/((Dzeta18-Dzeta16).*(Dzeta18-Dzeta17));
NX3(19,1,i)=(Dzeta-Dzeta20).*(Dzeta-Dzeta21)/((Dzeta19-Dzeta20).*(Dzeta19-Dzeta21));
NX3(20,1,i)=(Dzeta-Dzeta19).*(Dzeta-Dzeta21)/((Dzeta20-Dzeta19).*(Dzeta20-Dzeta21));
NX3(21,1,i)=(Dzeta-Dzeta19).*(Dzeta-Dzeta20)/((Dzeta21-Dzeta19).*(Dzeta21-Dzeta20));
NX3(22,1,i)=(Dzeta-Dzeta23).*(Dzeta-Dzeta24)/((Dzeta22-Dzeta23).*(Dzeta22-Dzeta24));
NX3(23,1,i)=(Dzeta-Dzeta22).*(Dzeta-Dzeta24)/((Dzeta23-Dzeta22).*(Dzeta23-Dzeta24));
NX3(24,1,i)=(Dzeta-Dzeta22).*(Dzeta-Dzeta23)/((Dzeta24-Dzeta22).*(Dzeta24-Dzeta23));
NX3(25,1,i)=(Dzeta-Dzeta26).*(Dzeta-Dzeta27)/((Dzeta25-Dzeta26).*(Dzeta25-Dzeta27));
NX3(26,1,i)=(Dzeta-Dzeta25).*(Dzeta-Dzeta27)/((Dzeta26-Dzeta25).*(Dzeta26-Dzeta27));
NX3(27,1,i)=(Dzeta-Dzeta25).*(Dzeta-Dzeta26)/((Dzeta27-Dzeta25).*(Dzeta27-Dzeta26));
expand (NX3);
end

```

**funcform.m**

```

function N=funcform(NX1,NX2,NX3);
% Cálculo de las funciones de interpolación en las direcciones
% Xi,Eta,Dzeta (Configuración de referencia)
N=NX1.*NX2.*NX3;

```

**coordglobal.m**

```

function C= coordglobal(P,E);
% C -- Coordenadas globales por elemento (Configuración de referencia)
for i=1:size(E,1)
C(1,1,i)=P(E(i,1),1);C(2,1,i)=P(E(i,2),1);C(3,1,i)=P(E(i,3),1);
C(4,1,i)=P(E(i,4),1);C(5,1,i)=P(E(i,5),1);C(6,1,i)=P(E(i,6),1);
C(7,1,i)=P(E(i,7),1);C(8,1,i)=P(E(i,8),1);C(9,1,i)=P(E(i,9),1);
C(10,1,i)=P(E(i,10),1);C(11,1,i)=P(E(i,11),1);C(12,1,i)=P(E(i,12),1);
C(13,1,i)=P(E(i,13),1);C(14,1,i)=P(E(i,14),1);C(15,1,i)=P(E(i,15),1);
C(16,1,i)=P(E(i,16),1);C(17,1,i)=P(E(i,17),1);C(18,1,i)=P(E(i,18),1);
C(19,1,i)=P(E(i,19),1);C(20,1,i)=P(E(i,20),1);C(21,1,i)=P(E(i,21),1);
C(22,1,i)=P(E(i,22),1);C(23,1,i)=P(E(i,23),1);C(24,1,i)=P(E(i,24),1);
C(25,1,i)=P(E(i,25),1);C(26,1,i)=P(E(i,26),1);C(27,1,i)=P(E(i,27),1);

C(1,2,i)=P(E(i,1),2);C(2,2,i)=P(E(i,2),2);C(3,2,i)=P(E(i,3),2);
C(4,2,i)=P(E(i,4),2);C(5,2,i)=P(E(i,5),2);C(6,2,i)=P(E(i,6),2);
C(7,2,i)=P(E(i,7),2);C(8,2,i)=P(E(i,8),2);C(9,2,i)=P(E(i,9),2);
C(10,2,i)=P(E(i,10),2);C(11,2,i)=P(E(i,11),2);C(12,2,i)=P(E(i,12),2);
C(13,2,i)=P(E(i,13),2);C(14,2,i)=P(E(i,14),2);C(15,2,i)=P(E(i,15),2);
C(16,2,i)=P(E(i,16),2);C(17,2,i)=P(E(i,17),2);C(18,2,i)=P(E(i,18),2);
C(19,2,i)=P(E(i,19),2);C(20,2,i)=P(E(i,20),2);C(21,2,i)=P(E(i,21),2);
C(22,2,i)=P(E(i,22),2);C(23,2,i)=P(E(i,23),2);C(24,2,i)=P(E(i,24),2);
C(25,2,i)=P(E(i,25),2);C(26,2,i)=P(E(i,26),2);C(27,2,i)=P(E(i,27),2);

C(1,3,i)=P(E(i,1),3);C(2,3,i)=P(E(i,2),3);C(3,3,i)=P(E(i,3),3);
C(4,3,i)=P(E(i,4),3);C(5,3,i)=P(E(i,5),3);C(6,3,i)=P(E(i,6),3);
C(7,3,i)=P(E(i,7),3);C(8,3,i)=P(E(i,8),3);C(9,3,i)=P(E(i,9),3);
C(10,3,i)=P(E(i,10),3);C(11,3,i)=P(E(i,11),3);C(12,3,i)=P(E(i,12),3);
C(13,3,i)=P(E(i,13),3);C(14,3,i)=P(E(i,14),3);C(15,3,i)=P(E(i,15),3);
C(16,3,i)=P(E(i,16),3);C(17,3,i)=P(E(i,17),3);C(18,3,i)=P(E(i,18),3);
C(19,3,i)=P(E(i,19),3);C(20,3,i)=P(E(i,20),3);C(21,3,i)=P(E(i,21),3);
C(22,3,i)=P(E(i,22),3);C(23,3,i)=P(E(i,23),3);C(24,3,i)=P(E(i,24),3);
C(25,3,i)=P(E(i,25),3);C(26,3,i)=P(E(i,26),3);C(27,3,i)=P(E(i,27),3);
end

```

**coordnat.m**

```

function [CN]= coordnat(C);
% CN -- Coordenadas naturales.
% T -- Traslación de coordenadas.
% Rot-- Rotación de los ejes de coordenadas.
for i=1:size(C,3)
    dx=C(14,1,i);dy=C(14,2,i);dz=C(14,3,i);
    x1=C(1,1,i);x2=C(2,1,i);x3=C(3,1,i);x4=C(4,1,i);x5=C(5,1,i);x6=C(6,1,i);
    x7=C(7,1,i);x8=C(8,1,i);x9=C(9,1,i);x10=C(10,1,i);x11=C(11,1,i);
    x12=C(12,1,i);x13=C(13,1,i);x14=C(14,1,i);x15=C(15,1,i);x16=C(16,1,i);
    x17=C(17,1,i);x18=C(18,1,i);x19=C(19,1,i);x20=C(20,1,i);x21=C(21,1,i);
    x22=C(22,1,i);x23=C(23,1,i);x24=C(24,1,i);x25=C(25,1,i);x26=C(26,1,i);
    x27=C(27,1,i);y1=C(1,2,i);y2=C(2,2,i);y3=C(3,2,i);y4=C(4,2,i);
    y5=C(5,2,i);y6=C(6,2,i);y7=C(7,2,i);y8=C(8,2,i);y9=C(9,2,i);
    y10=C(10,2,i);y11=C(11,2,i);y12=C(12,2,i);y13=C(13,2,i);y14=C(14,2,i);
    y15=C(15,2,i);y16=C(16,2,i);y17=C(17,2,i);y18=C(18,2,i);y19=C(19,2,i);
    y20=C(20,2,i);y21=C(21,2,i);y22=C(22,2,i);y23=C(23,2,i);y24=C(24,2,i);
    y25=C(25,2,i);y26=C(26,2,i);y27=C(27,2,i);z1=C(1,3,i);z2=C(2,3,i);
    z3=C(3,3,i);z4=C(4,3,i);z5=C(5,3,i);z6=C(6,3,i);z7=C(7,3,i);z8=C(8,3,i);
    z9=C(9,3,i);z10=C(10,3,i);z11=C(11,3,i);z12=C(12,3,i);z13=C(13,3,i);
    z14=C(14,3,i);z15=C(15,3,i);z16=C(16,3,i);z17=C(17,3,i);z18=C(18,3,i);
    z19=C(19,3,i);z20=C(20,3,i);z21=C(21,3,i);z22=C(22,3,i);z23=C(23,3,i);
    z24=C(24,3,i);z25=C(25,3,i);z26=C(26,3,i);z27=C(27,3,i);
    t(:, :, i)=[1 0 0 -dx; 0 1 0 -dy; 0 0 1 -dz; 0 0 0 1];

    r1= [x1; y1; z1; 1]; r2= [x2; y2; z2; 1]; r3= [x3; y3; z3; 1];
    r4= [x4; y4; z4; 1]; r5= [x5; y5; z5; 1]; r6= [x6; y6; z6; 1];
    r7= [x7; y7; z7; 1]; r8= [x8; y8; z8; 1]; r9= [x9; y9; z9; 1];
    r10= [x10; y10; z10; 1]; r11= [x11; y11; z11; 1]; r12= [x12; y12; z12; 1];
    r13=[x13; y13; z13; 1];r14=[x14; y14; z14; 1];r15=[x15; y15; z15; 1];
    r16=[x16; y16; z16; 1];r17=[x17; y17; z17; 1];r18=[x18; y18; z18; 1];
    r19=[x19; y19; z19; 1];r20=[x20; y20; z20; 1];r21=[x21; y21; z21; 1];
    r22=[x22; y22; z22; 1];r23=[x23; y23; z23; 1];r24=[x24; y24; z24; 1];
    r25=[x25; y25; z25; 1];r26=[x26; y26; z26; 1];r27=[x27; y27; z27; 1];

    T1=[t(:, :, i)*r1]';T2=[t(:, :, i)*r2]';T3=[t(:, :, i)*r3]';T4=[t(:, :, i)*r4]';
    T5=[t(:, :, i)*r5]';T6=[t(:, :, i)*r6]';T7=[t(:, :, i)*r7]';T8=[t(:, :, i)*r8]';
    T9=[t(:, :, i)*r9]';T10=[t(:, :, i)*r10]';T11=[t(:, :, i)*r11]';
    T12=[t(:, :, i)*r12]'; T13=[t(:, :, i)*r13]';T14=[t(:, :, i)*r14]';
    T15=[t(:, :, i)*r15]';T16=[t(:, :, i)*r16]';T17=[t(:, :, i)*r17]';
    T18=[t(:, :, i)*r18]';T19=[t(:, :, i)*r19]';T20=[t(:, :, i)*r20]';
    T21=[t(:, :, i)*r21]';T22=[t(:, :, i)*r22]';T23=[t(:, :, i)*r23]';
    T24=[t(:, :, i)*r24]';T25=[t(:, :, i)*r25]';T26=[t(:, :, i)*r26]';
    T27=[t(:, :, i)*r27]';

    T(:, :, i)=[T1;T2;T3;T4;T5;T6;T7;T8;T9;T10;T11;T12;T13;T14;T15;T16;T17;T18;...
        T19;T20;T21;T22;T23;T24;T25;T26;T27];

    A(:, :, i)=atan(T(23,2,i)./T(23,1,i)); % Cálculo del ángulo rotación
    A(:, :, i)=A(:, :, i)*180./pi; % Transformación de radianes a grados cercios

    rot(:, :, i)=[cos(A(:, :, i).*pi./180),sin(A(:, :, i).*pi./180),0,0;...
        -sin(A(:, :, i).*pi./180) cos(A(:, :, i).*pi./180) 0 0; ...
        0 0 1 0; 0 0 0 1];

```

```

X1=T(1,1,i);X2=T(2,1,i);X3=T(3,1,i);X4=T(4,1,i);X5=T(5,1,i);X6=T(6,1,i);
X7=T(7,1,i);X8=T(8,1,i);X9=T(9,1,i);X10=T(10,1,i);X11=T(11,1,i);X12=T(12,1,i);
X13=T(13,1,i);X14=T(14,1,i);X15=T(15,1,i);X16=T(16,1,i);X17=T(17,1,i);
X18=T(18,1,i);X19=T(19,1,i);X20=T(20,1,i);X21=T(21,1,i);X22=T(22,1,i);
X23=T(23,1,i);X24=T(24,1,i);X25=T(25,1,i);X26=T(26,1,i);X27=T(27,1,i);

Y1=T(1,2,i);Y2=T(2,2,i);Y3=T(3,2,i);Y4=T(4,2,i);Y5=T(5,2,i);Y6=T(6,2,i);
Y7=T(7,2,i);Y8=T(8,2,i);Y9=T(9,2,i);Y10=T(10,2,i);Y11=T(11,2,i);Y12=T(12,2,i);
Y13=T(13,2,i);Y14=T(14,2,i);Y15=T(15,2,i);Y16=T(16,2,i);Y17=T(17,2,i);
Y18=T(18,2,i);Y19=T(19,2,i);Y20=T(20,2,i);Y21=T(21,2,i);Y22=T(22,2,i);
Y23=T(23,2,i);Y24=T(24,2,i);Y25=T(25,2,i);Y26=T(26,2,i);Y27=T(27,2,i);

Z1=T(1,3,i);Z2=T(2,3,i);Z3=T(3,3,i);Z4=T(4,3,i);Z5=T(5,3,i);Z6=T(6,3,i);
Z7=T(7,3,i);Z8=T(8,3,i);Z9=T(9,3,i);Z10=T(10,3,i);Z11=T(11,3,i);Z12=T(12,3,i);
Z13=T(13,3,i);Z14=T(14,3,i);Z15=T(15,3,i);Z16=T(16,3,i);Z17=T(17,3,i);
Z18=T(18,3,i);Z19=T(19,3,i);Z20=T(20,3,i);Z21=T(21,3,i);Z22=T(22,3,i);
Z23=T(23,3,i);Z24=T(24,3,i);Z25=T(25,3,i);Z26=T(26,3,i);Z27=T(27,3,i);

R1= [X1; Y1; Z1; 1];R2= [X2; Y2; Z2; 1];R3= [X3; Y3; Z3; 1];
R4= [X4; Y4; Z4; 1];R5= [X5; Y5; Z5; 1];R6= [X6; Y6; Z6; 1];
R7= [X7; Y7; Z7; 1];R8= [X8; Y8; Z8; 1];R9= [X9; Y9; Z9; 1];
R10= [X10; Y10; Z10; 1];R11= [X11; Y11; Z11; 1];R12= [X12; Y12; Z12; 1];
R13=[X13; Y13; Z13; 1];R14=[X14; Y14; Z14; 1];R15=[X15; Y15; Z15; 1];
R16=[X16; Y16; Z16; 1];R17=[X17; Y17; Z17; 1];R18=[X18; Y18; Z18; 1];
R19=[X19; Y19; Z19; 1];R20=[X20; Y20; Z20; 1];R21=[X21; Y21; Z21; 1];
R22=[X22; Y22; Z22; 1];R23=[X23; Y23; Z23; 1];R24=[X24; Y24; Z24; 1];
R25=[X25; Y25; Z25; 1];R26=[X26; Y26; Z26; 1];R27=[X27; Y27; Z27; 1];

Rot1=[rot(:, :, i)*R1]';Rot2=[rot(:, :, i)*R2]';Rot3=[rot(:, :, i)*R3]';
Rot4=[rot(:, :, i)*R4]';Rot5=[rot(:, :, i)*R5]';Rot6=[rot(:, :, i)*R6]';
Rot7=[rot(:, :, i)*R7]';Rot8=[rot(:, :, i)*R8]';Rot9=[rot(:, :, i)*R9]';
Rot10=[rot(:, :, i)*R10]';Rot11=[rot(:, :, i)*R11]';Rot12=[rot(:, :, i)*R12]';
Rot13=[rot(:, :, i)*R13]';Rot14=[rot(:, :, i)*R14]';Rot15=[rot(:, :, i)*R15]';
Rot16=[rot(:, :, i)*R16]';Rot17=[rot(:, :, i)*R17]';Rot18=[rot(:, :, i)*R18]';
Rot19=[rot(:, :, i)*R19]';Rot20=[rot(:, :, i)*R20]';Rot21=[rot(:, :, i)*R21]';
Rot22=[rot(:, :, i)*R22]';Rot23=[rot(:, :, i)*R23]';Rot24=[rot(:, :, i)*R24]';
Rot25=[rot(:, :, i)*R25]';Rot26=[rot(:, :, i)*R26]';Rot27=[rot(:, :, i)*R27]';

Rot(:, :, i)=[Rot1;Rot2;Rot3;Rot4;Rot5;Rot6;Rot7;Rot8;Rot9;Rot10;Rot11;...
Rot12;Rot13;Rot14;Rot15;Rot16;Rot17;Rot18;Rot19;Rot20;Rot21;Rot22;...
Rot23;Rot24;Rot25;Rot26;Rot27];
CN(:, :, i)=[Rot(:, 1, i) Rot(:, 2, i) Rot(:, 3, i)];
end

```

**vectorXi.m**

```

function VXi= vectorXi(N,E,CN);
% VXi - Vectores tangente en dirección Xi (Configuración de referencia)
Datos;
VXi=zeros(3,size(E,2),size(E,1));
for i=1:size(E,1)
    DXi(1,1,i)=diff(N(1,1,i),Xi);DXi(1,2,i)=diff(N(2,1,i),Xi);
    DXi(1,3,i)=diff(N(3,1,i),Xi);DXi(1,4,i)=diff(N(4,1,i),Xi);
    DXi(1,5,i)=diff(N(5,1,i),Xi);DXi(1,6,i)=diff(N(6,1,i),Xi);
    DXi(1,7,i)=diff(N(7,1,i),Xi);DXi(1,8,i)=diff(N(8,1,i),Xi);
    DXi(1,9,i)=diff(N(9,1,i),Xi);DXi(1,10,i)=diff(N(10,1,i),Xi);
    DXi(1,11,i)=diff(N(11,1,i),Xi);DXi(1,12,i)=diff(N(12,1,i),Xi);
    DXi(1,13,i)=diff(N(13,1,i),Xi);DXi(1,14,i)=diff(N(14,1,i),Xi);
    DXi(1,15,i)=diff(N(15,1,i),Xi);DXi(1,16,i)=diff(N(16,1,i),Xi);
    DXi(1,17,i)=diff(N(17,1,i),Xi);DXi(1,18,i)=diff(N(18,1,i),Xi);
    DXi(1,19,i)=diff(N(19,1,i),Xi);DXi(1,20,i)=diff(N(20,1,i),Xi);
    DXi(1,21,i)=diff(N(21,1,i),Xi);DXi(1,22,i)=diff(N(22,1,i),Xi);
    DXi(1,23,i)=diff(N(23,1,i),Xi);DXi(1,24,i)=diff(N(24,1,i),Xi);
    DXi(1,25,i)=diff(N(25,1,i),Xi);DXi(1,26,i)=diff(N(26,1,i),Xi);
    DXi(1,27,i)=diff(N(27,1,i),Xi);

    VXi(1,1,i)=subs(DXi(1,1,i),{Xi, Eta,Dzeta},{CN(1,1,i),CN(1,2,i),CN(1,3,i)});
    VXi(1,2,i)=subs(DXi(1,2,i),{Xi, Eta,Dzeta},{CN(2,1,i),CN(2,2,i),CN(2,3,i)});
    VXi(1,3,i)=subs(DXi(1,3,i),{Xi, Eta,Dzeta},{CN(3,1,i),CN(3,2,i),CN(3,3,i)});
    VXi(1,4,i)=subs(DXi(1,4,i),{Xi, Eta,Dzeta},{CN(4,1,i),CN(4,2,i),CN(4,3,i)});
    VXi(1,5,i)=subs(DXi(1,5,i),{Xi, Eta,Dzeta},{CN(5,1,i),CN(5,2,i),CN(5,3,i)});
    VXi(1,6,i)=subs(DXi(1,6,i),{Xi, Eta,Dzeta},{CN(6,1,i),CN(6,2,i),CN(6,3,i)});
    VXi(1,7,i)=subs(DXi(1,7,i),{Xi, Eta,Dzeta},{CN(7,1,i),CN(7,2,i),CN(7,3,i)});
    VXi(1,8,i)=subs(DXi(1,8,i),{Xi, Eta,Dzeta},{CN(8,1,i),CN(8,2,i),CN(8,3,i)});
    VXi(1,9,i)=subs(DXi(1,9,i),{Xi, Eta,Dzeta},{CN(9,1,i),CN(9,2,i),CN(9,3,i)});
    VXi(1,10,i)=subs(DXi(1,10,i),{Xi, Eta,Dzeta},{CN(10,1,i),CN(10,2,i),CN(10,3,i)});
    VXi(1,11,i)=subs(DXi(1,11,i),{Xi, Eta,Dzeta},{CN(11,1,i),CN(11,2,i),CN(11,3,i)});
    VXi(1,12,i)=subs(DXi(1,12,i),{Xi, Eta,Dzeta},{CN(12,1,i),CN(12,2,i),CN(12,3,i)});
    VXi(1,13,i)=subs(DXi(1,13,i),{Xi, Eta,Dzeta},{CN(13,1,i),CN(13,2,i),CN(13,3,i)});
    VXi(1,14,i)=subs(DXi(1,14,i),{Xi, Eta,Dzeta},{CN(14,1,i),CN(14,2,i),CN(14,3,i)});
    VXi(1,15,i)=subs(DXi(1,15,i),{Xi, Eta,Dzeta},{CN(15,1,i),CN(15,2,i),CN(15,3,i)});
    VXi(1,16,i)=subs(DXi(1,16,i),{Xi, Eta,Dzeta},{CN(16,1,i),CN(16,2,i),CN(16,3,i)});
    VXi(1,17,i)=subs(DXi(1,17,i),{Xi, Eta,Dzeta},{CN(17,1,i),CN(17,2,i),CN(17,3,i)});
    VXi(1,18,i)=subs(DXi(1,18,i),{Xi, Eta,Dzeta},{CN(18,1,i),CN(18,2,i),CN(18,3,i)});
    VXi(1,19,i)=subs(DXi(1,19,i),{Xi, Eta,Dzeta},{CN(19,1,i),CN(19,2,i),CN(19,3,i)});
    VXi(1,20,i)=subs(DXi(1,20,i),{Xi, Eta,Dzeta},{CN(20,1,i),CN(20,2,i),CN(20,3,i)});
    VXi(1,21,i)=subs(DXi(1,21,i),{Xi, Eta,Dzeta},{CN(21,1,i),CN(21,2,i),CN(21,3,i)});
    VXi(1,22,i)=subs(DXi(1,22,i),{Xi, Eta,Dzeta},{CN(22,1,i),CN(22,2,i),CN(22,3,i)});
    VXi(1,23,i)=subs(DXi(1,23,i),{Xi, Eta,Dzeta},{CN(23,1,i),CN(23,2,i),CN(23,3,i)});
    VXi(1,24,i)=subs(DXi(1,24,i),{Xi, Eta,Dzeta},{CN(24,1,i),CN(24,2,i),CN(24,3,i)});
    VXi(1,25,i)=subs(DXi(1,25,i),{Xi, Eta,Dzeta},{CN(25,1,i),CN(25,2,i),CN(25,3,i)});
    VXi(1,26,i)=subs(DXi(1,26,i),{Xi, Eta,Dzeta},{CN(26,1,i),CN(26,2,i),CN(26,3,i)});
    VXi(1,27,i)=subs(DXi(1,27,i),{Xi, Eta,Dzeta},{CN(27,1,i),CN(27,2,i),CN(27,3,i)});
end

```

**vectorEta.m**

```

function VEta= vectorEta(N,E,CN);
% Veta - vectores tangentes en dirección Eta (Configuración de referencia)
Datos;
VEta=zeros(3,size(E,2),size(E,1));

for i=1:size(E,1)
    DEta(1,1,i)=diff(N(1,1,i),Eta);DEta(1,2,i)=diff(N(2,1,i),Eta);
    DEta(1,3,i)=diff(N(3,1,i),Eta);DEta(1,4,i)=diff(N(4,1,i),Eta);
    DEta(1,5,i)=diff(N(5,1,i),Eta);DEta(1,6,i)=diff(N(6,1,i),Eta);
    DEta(1,7,i)=diff(N(7,1,i),Eta);DEta(1,8,i)=diff(N(8,1,i),Eta);
    DEta(1,9,i)=diff(N(9,1,i),Eta);DEta(1,10,i)=diff(N(10,1,i),Eta);
    DEta(1,11,i)=diff(N(11,1,i),Eta);DEta(1,12,i)=diff(N(12,1,i),Eta);
    DEta(1,13,i)=diff(N(13,1,i),Eta);DEta(1,14,i)=diff(N(14,1,i),Eta);
    DEta(1,15,i)=diff(N(15,1,i),Eta);DEta(1,16,i)=diff(N(16,1,i),Eta);
    DEta(1,17,i)=diff(N(17,1,i),Eta);DEta(1,18,i)=diff(N(18,1,i),Eta);
    DEta(1,19,i)=diff(N(19,1,i),Eta);DEta(1,20,i)=diff(N(20,1,i),Eta);
    DEta(1,21,i)=diff(N(21,1,i),Eta);DEta(1,22,i)=diff(N(22,1,i),Eta);
    DEta(1,23,i)=diff(N(23,1,i),Eta);DEta(1,24,i)=diff(N(24,1,i),Eta);
    DEta(1,25,i)=diff(N(25,1,i),Eta);DEta(1,26,i)=diff(N(26,1,i),Eta);
    DEta(1,27,i)=diff(N(27,1,i),Eta);

    VEta(2,1,i)=subs(DEta(1,1,i),{Xi, Eta, Dzeta},{CN(1,1,i),CN(1,2,i),CN(1,3,i)});
    VEta(2,2,i)=subs(DEta(1,2,i),{Xi, Eta, Dzeta},{CN(2,1,i),CN(2,2,i),CN(2,3,i)});
    VEta(2,3,i)=subs(DEta(1,3,i),{Xi, Eta, Dzeta},{CN(3,1,i),CN(3,2,i),CN(3,3,i)});
    VEta(2,4,i)=subs(DEta(1,4,i),{Xi, Eta, Dzeta},{CN(4,1,i),CN(4,2,i),CN(4,3,i)});
    VEta(2,5,i)=subs(DEta(1,5,i),{Xi, Eta, Dzeta},{CN(5,1,i),CN(5,2,i),CN(5,3,i)});
    VEta(2,6,i)=subs(DEta(1,6,i),{Xi, Eta, Dzeta},{CN(6,1,i),CN(6,2,i),CN(6,3,i)});
    VEta(2,7,i)=subs(DEta(1,7,i),{Xi, Eta, Dzeta},{CN(7,1,i),CN(7,2,i),CN(7,3,i)});
    VEta(2,8,i)=subs(DEta(1,8,i),{Xi, Eta, Dzeta},{CN(8,1,i),CN(8,2,i),CN(8,3,i)});
    VEta(2,9,i)=subs(DEta(1,9,i),{Xi, Eta, Dzeta},{CN(9,1,i),CN(9,2,i),CN(9,3,i)});
    VEta(2,10,i)=subs(DEta(1,10,i),{Xi, Eta, Dzeta},{CN(10,1,i),CN(10,2,i),CN(10,3,i)});
    VEta(2,11,i)=subs(DEta(1,11,i),{Xi, Eta, Dzeta},{CN(11,1,i),CN(11,2,i),CN(11,3,i)});
    VEta(2,12,i)=subs(DEta(1,12,i),{Xi, Eta, Dzeta},{CN(12,1,i),CN(12,2,i),CN(12,3,i)});
    VEta(2,13,i)=subs(DEta(1,13,i),{Xi, Eta, Dzeta},{CN(13,1,i),CN(13,2,i),CN(13,3,i)});
    VEta(2,14,i)=subs(DEta(1,14,i),{Xi, Eta, Dzeta},{CN(14,1,i),CN(14,2,i),CN(14,3,i)});
    VEta(2,15,i)=subs(DEta(1,15,i),{Xi, Eta, Dzeta},{CN(15,1,i),CN(15,2,i),CN(15,3,i)});
    VEta(2,16,i)=subs(DEta(1,16,i),{Xi, Eta, Dzeta},{CN(16,1,i),CN(16,2,i),CN(16,3,i)});
    VEta(2,17,i)=subs(DEta(1,17,i),{Xi, Eta, Dzeta},{CN(17,1,i),CN(17,2,i),CN(17,3,i)});
    VEta(2,18,i)=subs(DEta(1,18,i),{Xi, Eta, Dzeta},{CN(18,1,i),CN(18,2,i),CN(18,3,i)});
    VEta(2,19,i)=subs(DEta(1,19,i),{Xi, Eta, Dzeta},{CN(19,1,i),CN(19,2,i),CN(19,3,i)});
    VEta(2,20,i)=subs(DEta(1,20,i),{Xi, Eta, Dzeta},{CN(20,1,i),CN(20,2,i),CN(20,3,i)});
    VEta(2,21,i)=subs(DEta(1,21,i),{Xi, Eta, Dzeta},{CN(21,1,i),CN(21,2,i),CN(21,3,i)});
    VEta(2,22,i)=subs(DEta(1,22,i),{Xi, Eta, Dzeta},{CN(22,1,i),CN(22,2,i),CN(22,3,i)});
    VEta(2,23,i)=subs(DEta(1,23,i),{Xi, Eta, Dzeta},{CN(23,1,i),CN(23,2,i),CN(23,3,i)});
    VEta(2,24,i)=subs(DEta(1,24,i),{Xi, Eta, Dzeta},{CN(24,1,i),CN(24,2,i),CN(24,3,i)});
    VEta(2,25,i)=subs(DEta(1,25,i),{Xi, Eta, Dzeta},{CN(25,1,i),CN(25,2,i),CN(25,3,i)});
    VEta(2,26,i)=subs(DEta(1,26,i),{Xi, Eta, Dzeta},{CN(26,1,i),CN(26,2,i),CN(26,3,i)});
    VEta(2,27,i)=subs(DEta(1,27,i),{Xi, Eta, Dzeta},{CN(27,1,i),CN(27,2,i),CN(27,3,i)});
end

```

**vetordzeta.m**

```

function VDzeta = vectorDzeta(N,E,CN);
% VDzeta - vectores tangentes en dirección Dzeta (Configuración de referencia)
Datos;
    VDzeta=zeros(3,size(E,2),size(E,1));
for i=1:size(E,1)
    DDzeta(1,1,i)=diff(N(1,1,i),Dzeta);DDzeta(1,2,i)=diff(N(2,1,i),Dzeta);
    DDzeta(1,3,i)=diff(N(3,1,i),Dzeta);DDzeta(1,4,i)=diff(N(4,1,i),Dzeta);
    DDzeta(1,5,i)=diff(N(5,1,i),Dzeta);DDzeta(1,6,i)=diff(N(6,1,i),Dzeta);
    DDzeta(1,7,i)=diff(N(7,1,i),Dzeta);DDzeta(1,8,i)=diff(N(8,1,i),Dzeta);
    DDzeta(1,9,i)=diff(N(9,1,i),Dzeta);DDzeta(1,10,i)=diff(N(10,1,i),Dzeta);
    DDzeta(1,11,i)=diff(N(11,1,i),Dzeta);DDzeta(1,12,i)=diff(N(12,1,i),Dzeta);
    DDzeta(1,13,i)=diff(N(13,1,i),Dzeta);DDzeta(1,14,i)=diff(N(14,1,i),Dzeta);
    DDzeta(1,15,i)=diff(N(15,1,i),Dzeta);DDzeta(1,16,i)=diff(N(16,1,i),Dzeta);
    DDzeta(1,17,i)=diff(N(17,1,i),Dzeta);DDzeta(1,18,i)=diff(N(18,1,i),Dzeta);
    DDzeta(1,19,i)=diff(N(19,1,i),Dzeta);DDzeta(1,20,i)=diff(N(20,1,i),Dzeta);
    DDzeta(1,21,i)=diff(N(21,1,i),Dzeta);DDzeta(1,22,i)=diff(N(22,1,i),Dzeta);
    DDzeta(1,23,i)=diff(N(23,1,i),Dzeta);DDzeta(1,24,i)=diff(N(24,1,i),Dzeta);
    DDzeta(1,25,i)=diff(N(25,1,i),Dzeta);DDzeta(1,26,i)=diff(N(26,1,i),Dzeta);
    DDzeta(1,27,i)=diff(N(27,1,i),Dzeta);

    VDzeta(3,1,i)=subs(DDzeta(1,1,i),{Xi,Eta,Dzeta},{CN(1,1,i),CN(1,2,i),CN(1,3,i)});
    VDzeta(3,2,i)=subs(DDzeta(1,2,i),{Xi,Eta,Dzeta},{CN(2,1,i),CN(2,2,i),CN(2,3,i)});
    VDzeta(3,3,i)=subs(DDzeta(1,3,i),{Xi,Eta,Dzeta},{CN(3,1,i),CN(3,2,i),CN(3,3,i)});
    VDzeta(3,4,i)=subs(DDzeta(1,4,i),{Xi,Eta,Dzeta},{CN(4,1,i),CN(4,2,i),CN(4,3,i)});
    VDzeta(3,5,i)=subs(DDzeta(1,5,i),{Xi,Eta,Dzeta},{CN(5,1,i),CN(5,2,i),CN(5,3,i)});
    VDzeta(3,6,i)=subs(DDzeta(1,6,i),{Xi,Eta,Dzeta},{CN(6,1,i),CN(6,2,i),CN(6,3,i)});
    VDzeta(3,7,i)=subs(DDzeta(1,7,i),{Xi,Eta,Dzeta},{CN(7,1,i),CN(7,2,i),CN(7,3,i)});
    VDzeta(3,8,i)=subs(DDzeta(1,8,i),{Xi,Eta,Dzeta},{CN(8,1,i),CN(8,2,i),CN(8,3,i)});
    VDzeta(3,9,i)=subs(DDzeta(1,9,i),{Xi,Eta,Dzeta},{CN(9,1,i),CN(9,2,i),CN(9,3,i)});
    VDzeta(3,10,i)=subs(DDzeta(1,10,i),{Xi,Eta,Dzeta},{CN(10,1,i),CN(10,2,i),CN(10,3,i)});
    VDzeta(3,11,i)=subs(DDzeta(1,11,i),{Xi,Eta,Dzeta},{CN(11,1,i),CN(11,2,i),CN(11,3,i)});
    VDzeta(3,12,i)=subs(DDzeta(1,12,i),{Xi,Eta,Dzeta},{CN(12,1,i),CN(12,2,i),CN(12,3,i)});
    VDzeta(3,13,i)=subs(DDzeta(1,13,i),{Xi,Eta,Dzeta},{CN(13,1,i),CN(13,2,i),CN(13,3,i)});
    VDzeta(3,14,i)=subs(DDzeta(1,14,i),{Xi,Eta,Dzeta},{CN(14,1,i),CN(14,2,i),CN(14,3,i)});
    VDzeta(3,15,i)=subs(DDzeta(1,15,i),{Xi,Eta,Dzeta},{CN(15,1,i),CN(15,2,i),CN(15,3,i)});
    VDzeta(3,16,i)=subs(DDzeta(1,16,i),{Xi,Eta,Dzeta},{CN(16,1,i),CN(16,2,i),CN(16,3,i)});
    VDzeta(3,17,i)=subs(DDzeta(1,17,i),{Xi,Eta,Dzeta},{CN(17,1,i),CN(17,2,i),CN(17,3,i)});
    VDzeta(3,18,i)=subs(DDzeta(1,18,i),{Xi,Eta,Dzeta},{CN(18,1,i),CN(18,2,i),CN(18,3,i)});
    VDzeta(3,19,i)=subs(DDzeta(1,19,i),{Xi,Eta,Dzeta},{CN(19,1,i),CN(19,2,i),CN(19,3,i)});
    VDzeta(3,20,i)=subs(DDzeta(1,20,i),{Xi,Eta,Dzeta},{CN(20,1,i),CN(20,2,i),CN(20,3,i)});
    VDzeta(3,21,i)=subs(DDzeta(1,21,i),{Xi,Eta,Dzeta},{CN(21,1,i),CN(21,2,i),CN(21,3,i)});
    VDzeta(3,22,i)=subs(DDzeta(1,22,i),{Xi,Eta,Dzeta},{CN(22,1,i),CN(22,2,i),CN(22,3,i)});
    VDzeta(3,23,i)=subs(DDzeta(1,23,i),{Xi,Eta,Dzeta},{CN(23,1,i),CN(23,2,i),CN(23,3,i)});
    VDzeta(3,24,i)=subs(DDzeta(1,24,i),{Xi,Eta,Dzeta},{CN(24,1,i),CN(24,2,i),CN(24,3,i)});
    VDzeta(3,25,i)=subs(DDzeta(1,25,i),{Xi,Eta,Dzeta},{CN(25,1,i),CN(25,2,i),CN(25,3,i)});
    VDzeta(3,26,i)=subs(DDzeta(1,26,i),{Xi,Eta,Dzeta},{CN(26,1,i),CN(26,2,i),CN(26,3,i)});
    VDzeta(3,27,i)=subs(DDzeta(1,27,i),{Xi,Eta,Dzeta},{CN(27,1,i),CN(27,2,i),CN(27,3,i)});
end

```

### matdervfuncd.m

```
function mdf= matdervfuncd(vxi,veta,vdzeta);
% MDF - Matriz de derivadas de las funciones de interpolación
% (Configuración de referencia)
A=vxi(1, :, :);
B=veta(2, :, :);
C=vdzeta(3, :, :);
mdf= [A;B;C];
```

### jacobianref.m

```
function J= jacobianref(MDF,C);
% Cálculo de la matriz jacobiana (configuración de referencia)

for i=1:size(MDF,3)
    J(:, :, i)=MDF(:, :, i)*C(:, :, i);
end
```

### puntoscoordef.m

```
function [p]= puntoscoordef(Rmin,rs,rmax,Zmin,Zs,Zmax,Amx,Amin,As);
% p -- puntos coordenados (configuración deformada)
% Fi -- ángulo con respecto al eje x1
% R -- radio vector
Datos;
p = []; % Puntos
for R = Rmin:rs./2:rmax,
    for Fi = Amin:As./2:Amx,
        for x3 = Zmin:Zs./2:Zmax,
            x1 = R.*cos(Fi.*pi./180);
            x2 = R.*sin(Fi.*pi./180);
            p = [p; x1 x2 x3];
        end
    end
end
```

**NumeroElementdef.m**

```

function Ne=NumeroElementdef(p);
% Ne - Número de elementos (Configuración deformada)
Datos;
Ne = []; % Listado de todos los elementos
for I = 0:Rn-1,
    for J = 0:An-1,
        for K = 0:Zn-1,
            e = []; % coordenadas de cada elemento
            for L = 0:0.5:1,
                for M = 0:0.5:1,
                    for N = 0:0.5:1,
                        R = Rmin + rs.*(I + L);
                        A = Amin + As.*(J + M);
                        x3 = Zmin + Zs.*(K + N);
                        x1 = R.*cos(A.*pi./180);
                        x2 = R.*sin(A.*pi./180);
                        for Q = 1:size(p,1),
                            if apcomp(p(Q,1), x1) & apcomp(p(Q,2), x2) &
apcomp(p(Q,3), x3),
                                e = [e, Q];
                                break;
                            end
                        end
                    end
                end
            end
        end
    end
    Ne = [Ne; e];
end
end
end

```

**plotprobetadef.m**

```

function plotprobetadef(Ne,p);
plot3(p(:,1),p(:,2),p(:,3),'r. '),hold on
for I = 1:size(Ne,1),
    plotq27(Ne(I,:),p);
end;
xlabel ( 'Eje Coordenadas x1')
ylabel ( 'Eje Coordenadas x2')
zlabel ( 'Eje Coordenadas x3')

```

**funcforxi.m**

```

function nxl=funcforxi(Xi,Ne,p);
% Cálculo de los polinomios de Lagrange en dirección
% del eje de coordenadas local Xi (Configuración deformada)
for i=1:size(Ne,1)
Xi1=p(Ne(i,1),1);Xi2=p(Ne(i,2),1);Xi3=p(Ne(i,3),1);Xi4=p(Ne(i,4),1);
Xi5=p(Ne(i,5),1);Xi6=p(Ne(i,6),1);Xi7=p(Ne(i,7),1);Xi8=p(Ne(i,8),1);
Xi9=p(Ne(i,9),1);Xi10=p(Ne(i,10),1);Xi11=p(Ne(i,11),1);Xi12=p(Ne(i,12),1);
Xi13=p(Ne(i,13),1);Xi14=p(Ne(i,14),1);Xi15=p(Ne(i,15),1);Xi16=p(Ne(i,16),1);
Xi17=p(Ne(i,17),1);Xi18=p(Ne(i,18),1);Xi19=p(Ne(i,19),1);Xi20=p(Ne(i,20),1);
Xi21=p(Ne(i,21),1);Xi22=p(Ne(i,22),1);Xi23=p(Ne(i,23),1);Xi24=p(Ne(i,24),1);
Xi25=p(Ne(i,25),1);Xi26=p(Ne(i,26),1);Xi27=p(Ne(i,27),1);

nxl(1,1,i)=(Xi-Xi10).*(Xi-Xi19)./((Xi1-Xi10).*(Xi1-Xi19));
nxl(2,1,i)=(Xi-Xi11).*(Xi-Xi20)./((Xi2-Xi11).*(Xi2-Xi20));
nxl(3,1,i)=(Xi-Xi12).*(Xi-Xi21)./((Xi3-Xi12).*(Xi3-Xi21));
nxl(4,1,i)=(Xi-Xi13).*(Xi-Xi22)./((Xi4-Xi13).*(Xi5-Xi22));
nxl(5,1,i)=(Xi-Xi14).*(Xi-Xi23)./((Xi5-Xi14).*(Xi5-Xi23));
nxl(6,1,i)=(Xi-Xi15).*(Xi-Xi24)./((Xi6-Xi15).*(Xi6-Xi24));
nxl(7,1,i)=(Xi-Xi16).*(Xi-Xi25)./((Xi7-Xi16).*(Xi7-Xi25));
nxl(8,1,i)=(Xi-Xi17).*(Xi-Xi26)./((Xi8-Xi17).*(Xi8-Xi26));
nxl(9,1,i)=(Xi-Xi18).*(Xi-Xi27)./((Xi9-Xi18).*(Xi9-Xi27));
nxl(10,1,i)=(Xi-Xi1).*(Xi-Xi19)./((Xi10-Xi1).*(Xi10-Xi19));
nxl(11,1,i)=(Xi-Xi2).*(Xi-Xi20)./((Xi11-Xi2).*(Xi11-Xi20));
nxl(12,1,i)=(Xi-Xi3).*(Xi-Xi21)./((Xi12-Xi3).*(Xi12-Xi21));
nxl(13,1,i)=(Xi-Xi4).*(Xi-Xi22)./((Xi13-Xi4).*(Xi13-Xi22));
nxl(14,1,i)=(Xi-Xi5).*(Xi-Xi23)./((Xi14-Xi5).*(Xi14-Xi23));
nxl(15,1,i)=(Xi-Xi6).*(Xi-Xi24)./((Xi15-Xi6).*(Xi15-Xi24));
nxl(16,1,i)=(Xi-Xi7).*(Xi-Xi25)./((Xi16-Xi25).*(Xi16-Xi25));
nxl(17,1,i)=(Xi-Xi8).*(Xi-Xi26)./((Xi17-Xi8).*(Xi17-Xi26));
nxl(18,1,i)=(Xi-Xi9).*(Xi-Xi27)./((Xi18-Xi9).*(Xi18-Xi27));
nxl(19,1,i)=(Xi-Xi10).*(Xi-Xi1)./((Xi19-Xi10).*(Xi19-Xi1));
nxl(20,1,i)=(Xi-Xi2).*(Xi-Xi11)./((Xi20-Xi2).*(Xi20-Xi11));
nxl(21,1,i)=(Xi-Xi3).*(Xi-Xi12)./((Xi21-Xi3).*(Xi21-Xi12));
nxl(22,1,i)=(Xi-Xi4).*(Xi-Xi13)./((Xi22-Xi4).*(Xi22-Xi13));
nxl(23,1,i)=(Xi-Xi5).*(Xi-Xi14)./((Xi23-Xi5).*(Xi23-Xi14));
nxl(24,1,i)=(Xi-Xi6).*(Xi-Xi15)./((Xi24-Xi6).*(Xi24-Xi15));
nxl(25,1,i)=(Xi-Xi7).*(Xi-Xi16)./((Xi25-Xi7).*(Xi25-Xi16));
nxl(26,1,i)=(Xi-Xi8).*(Xi-Xi17)./((Xi26-Xi8).*(Xi26-Xi17));
nxl(27,1,i)=(Xi-Xi9).*(Xi-Xi18)./((Xi27-Xi9).*(Xi27-Xi18));
expand (nxl);
end

```

**funcforeta.m**

```

function nx2=funforeta(Eta,Ne,p);
% Cálculo de los polinomios de Lagrange en dirección
% del eje de coordenadas local Eta (Configuración deformada)
for i=1:size(Ne,1)
Eta1=p(Ne(i,1),2);Eta2=p(Ne(i,2),2);Eta3=p(Ne(i,3),2);Eta4=p(Ne(i,4),2);
Eta5=p(Ne(i,5),2);Eta6=p(Ne(i,6),2);Eta7=p(Ne(i,7),2);Eta8=p(Ne(i,8),2);
Eta9=p(Ne(i,9),2);Eta10=p(Ne(i,10),2);Eta11=p(Ne(i,11),2);Eta12=p(Ne(i,12),2);
Eta13=p(Ne(i,13),2);Eta14=p(Ne(i,14),2);Eta15=p(Ne(i,15),2);Eta16=p(Ne(i,16),2);
Eta17=p(Ne(i,17),2);Eta18=p(Ne(i,18),2);Eta19=p(Ne(i,19),2);Eta20=p(Ne(i,20),2);
Eta21=p(Ne(i,21),2);Eta22=p(Ne(i,22),2);Eta23=p(Ne(i,23),2);Eta24=p(Ne(i,24),2);
Eta25=p(Ne(i,25),2);Eta26=p(Ne(i,26),2);Eta27=p(Ne(i,27),2);

nx2(1,1,i)=((Eta-Eta4).*(Eta-Eta7))./((Eta1-Eta4).*(Eta1-Eta7));
nx2(2,1,i)=((Eta-Eta5).*(Eta-Eta8))./((Eta2-Eta5).*(Eta2-Eta8));
nx2(3,1,i)=((Eta-Eta6).*(Eta-Eta9))./((Eta3-Eta6).*(Eta3-Eta9));
nx2(4,1,i)=((Eta-Eta1).*(Eta-Eta7))./((Eta4-Eta1).*(Eta4-Eta7));
nx2(5,1,i)=((Eta-Eta8).*(Eta-Eta2))./((Eta5-Eta8).*(Eta5-Eta2));
nx2(6,1,i)=((Eta-Eta3).*(Eta-Eta9))./((Eta6-Eta3).*(Eta6-Eta9));
nx2(7,1,i)=((Eta-Eta4).*(Eta-Eta1))./((Eta7-Eta4).*(Eta7-Eta1));
nx2(8,1,i)=((Eta-Eta2).*(Eta-Eta5))./((Eta8-Eta2).*(Eta8-Eta5));
nx2(9,1,i)=((Eta-Eta3).*(Eta-Eta6))./((Eta9-Eta3).*(Eta9-Eta6));
nx2(10,1,i)=((Eta-Eta13).*(Eta-Eta16))./((Eta10-Eta13).*(Eta10-Eta16));
nx2(11,1,i)=((Eta-Eta14).*(Eta-Eta17))./((Eta11-Eta14).*(Eta11-Eta17));
nx2(12,1,i)=((Eta-Eta15).*(Eta-Eta18))./((Eta12-Eta15).*(Eta12-Eta18));
nx2(13,1,i)=((Eta-Eta10).*(Eta-Eta16))./((Eta13-Eta10).*(Eta13-Eta16));
nx2(14,1,i)=((Eta-Eta11).*(Eta-Eta17))./((Eta14-Eta11).*(Eta14-Eta17));
nx2(15,1,i)=((Eta-Eta12).*(Eta-Eta18))./((Eta15-Eta12).*(Eta15-Eta18));
nx2(16,1,i)=((Eta-Eta10).*(Eta-Eta13))./((Eta16-Eta10).*(Eta16-Eta13));
nx2(17,1,i)=((Eta-Eta11).*(Eta-Eta14))./((Eta17-Eta11).*(Eta17-Eta14));
nx2(18,1,i)=((Eta-Eta12).*(Eta-Eta15))./((Eta18-Eta12).*(Eta18-Eta15));
nx2(19,1,i)=((Eta-Eta22).*(Eta-Eta25))./((Eta19-Eta22).*(Eta19-Eta25));
nx2(20,1,i)=((Eta-Eta23).*(Eta-Eta26))./((Eta20-Eta23).*(Eta20-Eta26));
nx2(21,1,i)=((Eta-Eta24).*(Eta-Eta27))./((Eta21-Eta24).*(Eta21-Eta27));
nx2(22,1,i)=((Eta-Eta19).*(Eta-Eta25))./((Eta22-Eta19).*(Eta22-Eta25));
nx2(23,1,i)=((Eta-Eta20).*(Eta-Eta26))./((Eta23-Eta20).*(Eta23-Eta26));
nx2(24,1,i)=((Eta-Eta21).*(Eta-Eta27))./((Eta24-Eta21).*(Eta24-Eta27));
nx2(25,1,i)=((Eta-Eta19).*(Eta-Eta22))./((Eta25-Eta19).*(Eta25-Eta22));
nx2(26,1,i)=((Eta-Eta20).*(Eta-Eta23))./((Eta26-Eta20).*(Eta26-Eta23));
nx2(27,1,i)=((Eta-Eta21).*(Eta-Eta24))./((Eta27-Eta21).*(Eta27-Eta24));
expand (nx2);
end

```

**funcfordzeta.m**

```

function nx3=funcfordzeta(Dzeta,Ne,p);
% Cálculo de los polinomios de Lagrange en dirección
% del eje de coordenadas local Dzeta (Configuración deformada)
for i=1:size(Ne,1)
Dzeta1=p(Ne(i,1),3);Dzeta2=p(Ne(i,2),3);Dzeta3=p(Ne(i,3),3);Dzeta4=p(Ne(i,4),3);
Dzeta5=p(Ne(i,5),3);Dzeta6=p(Ne(i,6),3);Dzeta7=p(Ne(i,7),3);Dzeta8=p(Ne(i,8),3);
Dzeta9=p(Ne(i,9),3);Dzeta10=p(Ne(i,10),3);Dzeta11=p(Ne(i,11),3);Dzeta12=p(Ne(i,12),3);
Dzeta13=p(Ne(i,13),3);Dzeta14=p(Ne(i,14),3);Dzeta15=p(Ne(i,15),3);
Dzeta16=p(Ne(i,16),3);Dzeta17=p(Ne(i,17),3);Dzeta18=p(Ne(i,18),3);
Dzeta19=p(Ne(i,19),3);Dzeta20=p(Ne(i,20),3);Dzeta21=p(Ne(i,21),3);
Dzeta22=p(Ne(i,22),3);Dzeta23=p(Ne(i,23),3);Dzeta24=p(Ne(i,24),3);
Dzeta25=p(Ne(i,25),3);Dzeta26=p(Ne(i,26),3);Dzeta27=p(Ne(i,27),3);

nx3(1,1,i)=(Dzeta-Dzeta2).*(Dzeta-Dzeta3)./((Dzeta1-Dzeta2).*(Dzeta1-Dzeta3));
nx3(2,1,i)=(Dzeta-Dzeta1).*(Dzeta-Dzeta3)./((Dzeta2-Dzeta1).*(Dzeta2-Dzeta3));
nx3(3,1,i)=(Dzeta-Dzeta1).*(Dzeta-Dzeta2)./((Dzeta3-Dzeta1).*(Dzeta3-Dzeta2));
nx3(4,1,i)=(Dzeta-Dzeta5).*(Dzeta-Dzeta6)./((Dzeta4-Dzeta5).*(Dzeta4-Dzeta6));
nx3(5,1,i)=(Dzeta-Dzeta4).*(Dzeta-Dzeta6)./((Dzeta5-Dzeta4).*(Dzeta5-Dzeta6));
nx3(6,1,i)=(Dzeta-Dzeta4).*(Dzeta-Dzeta5)./((Dzeta6-Dzeta4).*(Dzeta6-Dzeta5));
nx3(7,1,i)=(Dzeta-Dzeta8).*(Dzeta-Dzeta9)./((Dzeta7-Dzeta8).*(Dzeta7-Dzeta9));
nx3(8,1,i)=(Dzeta-Dzeta7).*(Dzeta-Dzeta9)./((Dzeta8-Dzeta7).*(Dzeta8-Dzeta9));
nx3(9,1,i)=(Dzeta-Dzeta7).*(Dzeta-Dzeta8)./((Dzeta9-Dzeta7).*(Dzeta9-Dzeta8));
nx3(10,1,i)=(Dzeta-Dzeta11).*(Dzeta-Dzeta12)./((Dzeta10-Dzeta11).*(Dzeta10-Dzeta12));
nx3(11,1,i)=(Dzeta-Dzeta10).*(Dzeta-Dzeta12)./((Dzeta11-Dzeta10).*(Dzeta11-Dzeta12));
nx3(12,1,i)=(Dzeta-Dzeta10).*(Dzeta-Dzeta11)./((Dzeta12-Dzeta10).*(Dzeta12-Dzeta11));
nx3(13,1,i)=(Dzeta-Dzeta14).*(Dzeta-Dzeta15)./((Dzeta13-Dzeta14).*(Dzeta13-Dzeta15));
nx3(14,1,i)=(Dzeta-Dzeta13).*(Dzeta-Dzeta15)./((Dzeta14-Dzeta13).*(Dzeta14-Dzeta15));
nx3(15,1,i)=(Dzeta-Dzeta13).*(Dzeta-Dzeta14)./((Dzeta15-Dzeta13).*(Dzeta15-Dzeta14));
nx3(16,1,i)=(Dzeta-Dzeta17).*(Dzeta-Dzeta18)./((Dzeta16-Dzeta17).*(Dzeta16-Dzeta18));
nx3(17,1,i)=(Dzeta-Dzeta16).*(Dzeta-Dzeta18)./((Dzeta17-Dzeta16).*(Dzeta17-Dzeta18));
nx3(18,1,i)=(Dzeta-Dzeta16).*(Dzeta-Dzeta17)./((Dzeta18-Dzeta16).*(Dzeta18-Dzeta17));
nx3(19,1,i)=(Dzeta-Dzeta20).*(Dzeta-Dzeta21)./((Dzeta19-Dzeta20).*(Dzeta19-Dzeta21));
nx3(20,1,i)=(Dzeta-Dzeta19).*(Dzeta-Dzeta21)./((Dzeta20-Dzeta19).*(Dzeta20-Dzeta21));
nx3(21,1,i)=(Dzeta-Dzeta19).*(Dzeta-Dzeta20)./((Dzeta21-Dzeta19).*(Dzeta21-Dzeta20));
nx3(22,1,i)=(Dzeta-Dzeta23).*(Dzeta-Dzeta24)./((Dzeta22-Dzeta23).*(Dzeta22-Dzeta24));
nx3(23,1,i)=(Dzeta-Dzeta22).*(Dzeta-Dzeta24)./((Dzeta23-Dzeta22).*(Dzeta23-Dzeta24));
nx3(24,1,i)=(Dzeta-Dzeta22).*(Dzeta-Dzeta23)./((Dzeta24-Dzeta22).*(Dzeta24-Dzeta23));
nx3(25,1,i)=(Dzeta-Dzeta26).*(Dzeta-Dzeta27)./((Dzeta25-Dzeta26).*(Dzeta25-Dzeta27));
nx3(26,1,i)=(Dzeta-Dzeta25).*(Dzeta-Dzeta27)./((Dzeta26-Dzeta25).*(Dzeta26-Dzeta27));
nx3(27,1,i)=(Dzeta-Dzeta25).*(Dzeta-Dzeta26)./((Dzeta27-Dzeta25).*(Dzeta27-Dzeta26));
expand (nx3);
end

```

**funcformdef.m**

```

function n=funcformdef(nx1,nx2,nx3);
% Cálculo de las funciones de interpolación en las direcciones
% Xi,Eta,Dzeta (Configuración deformada)
n=nx1.*nx2.*nx3;

```

**coordglobaldef.m**

```

function c= coordglobaldef(p,Ne);
% c -- coordenadas globales por elementos (Configuración deformada)
for i=1:size(Ne,1)
c(1,1,i)=p(Ne(i,1),1);c(2,1,i)=p(Ne(i,2),1);c(3,1,i)=p(Ne(i,3),1);
c(4,1,i)=p(Ne(i,4),1);c(5,1,i)=p(Ne(i,5),1);c(6,1,i)=p(Ne(i,6),1);
c(7,1,i)=p(Ne(i,7),1);c(8,1,i)=p(Ne(i,8),1);c(9,1,i)=p(Ne(i,9),1);
c(10,1,i)=p(Ne(i,10),1);c(11,1,i)=p(Ne(i,11),1);c(12,1,i)=p(Ne(i,12),1);
c(13,1,i)=p(Ne(i,13),1);c(14,1,i)=p(Ne(i,14),1);c(15,1,i)=p(Ne(i,15),1);
c(16,1,i)=p(Ne(i,16),1);c(17,1,i)=p(Ne(i,17),1);c(18,1,i)=p(Ne(i,18),1);
c(19,1,i)=p(Ne(i,19),1);c(20,1,i)=p(Ne(i,20),1);c(21,1,i)=p(Ne(i,21),1);
c(22,1,i)=p(Ne(i,22),1);c(23,1,i)=p(Ne(i,23),1);c(24,1,i)=p(Ne(i,24),1);
c(25,1,i)=p(Ne(i,25),1);c(26,1,i)=p(Ne(i,26),1);c(27,1,i)=p(Ne(i,27),1);

c(1,2,i)=p(Ne(i,1),2);c(2,2,i)=p(Ne(i,2),2);c(3,2,i)=p(Ne(i,3),2);
c(4,2,i)=p(Ne(i,4),2);c(5,2,i)=p(Ne(i,5),2);c(6,2,i)=p(Ne(i,6),2);
c(7,2,i)=p(Ne(i,7),2);c(8,2,i)=p(Ne(i,8),2);c(9,2,i)=p(Ne(i,9),2);
c(10,2,i)=p(Ne(i,10),2);c(11,2,i)=p(Ne(i,11),2);c(12,2,i)=p(Ne(i,12),2);
c(13,2,i)=p(Ne(i,13),2);c(14,2,i)=p(Ne(i,14),2);c(15,2,i)=p(Ne(i,15),2);
c(16,2,i)=p(Ne(i,16),2);c(17,2,i)=p(Ne(i,17),2);c(18,2,i)=p(Ne(i,18),2);
c(19,2,i)=p(Ne(i,19),2);c(20,2,i)=p(Ne(i,20),2);c(21,2,i)=p(Ne(i,21),2);
c(22,2,i)=p(Ne(i,22),2);c(23,2,i)=p(Ne(i,23),2);c(24,2,i)=p(Ne(i,24),2);
c(25,2,i)=p(Ne(i,25),2);c(26,2,i)=p(Ne(i,26),2);c(27,2,i)=p(Ne(i,27),2);

c(1,3,i)=p(Ne(i,1),3);c(2,3,i)=p(Ne(i,2),3);c(3,3,i)=p(Ne(i,3),3);
c(4,3,i)=p(Ne(i,4),3);c(5,3,i)=p(Ne(i,5),3);c(6,3,i)=p(Ne(i,6),3);
c(7,3,i)=p(Ne(i,7),3);c(8,3,i)=p(Ne(i,8),3);c(9,3,i)=p(Ne(i,9),3);
c(10,3,i)=p(Ne(i,10),3);c(11,3,i)=p(Ne(i,11),3);c(12,3,i)=p(Ne(i,12),3);
c(13,3,i)=p(Ne(i,13),3);c(14,3,i)=p(Ne(i,14),3);c(15,3,i)=p(Ne(i,15),3);
c(16,3,i)=p(Ne(i,16),3);c(17,3,i)=p(Ne(i,17),3);c(18,3,i)=p(Ne(i,18),3);
c(19,3,i)=p(Ne(i,19),3);c(20,3,i)=p(Ne(i,20),3);c(21,3,i)=p(Ne(i,21),3);
c(22,3,i)=p(Ne(i,22),3);c(23,3,i)=p(Ne(i,23),3);c(24,3,i)=p(Ne(i,24),3);
c(25,3,i)=p(Ne(i,25),3);c(26,3,i)=p(Ne(i,26),3);c(27,3,i)=p(Ne(i,27),3);
end

```

**coordnatdef.m**

```

function [cn]= coordnatdef(c);
% cn -- coordenadas naturales.
% T -- Traslación de coordenadas.
% Rot-- Rotación de los ejes de coordenadas.
for i=1:size(c,3)
    dx=c(14,1,i);dy=c(14,2,i);dz=c(14,3,i);
    x1=c(1,1,i);x2=c(2,1,i);x3=c(3,1,i);x4=c(4,1,i);x5=c(5,1,i);x6=c(6,1,i);
    x7=c(7,1,i);x8=c(8,1,i);x9=c(9,1,i);x10=c(10,1,i);x11=c(11,1,i);
    x12=c(12,1,i);x13=c(13,1,i);x14=c(14,1,i);x15=c(15,1,i);x16=c(16,1,i);
    x17=c(17,1,i);x18=c(18,1,i);x19=c(19,1,i);x20=c(20,1,i);x21=c(21,1,i);
    x22=c(22,1,i);x23=c(23,1,i);x24=c(24,1,i);x25=c(25,1,i);x26=c(26,1,i);
    x27=c(27,1,i);
    y1=c(1,2,i);y2=c(2,2,i);y3=c(3,2,i);y4=c(4,2,i);y5=c(5,2,i);y6=c(6,2,i);
    y7=c(7,2,i);y8=c(8,2,i);y9=c(9,2,i);y10=c(10,2,i);y11=c(11,2,i);y12=c(12,2,i);
    y13=c(13,2,i);y14=c(14,2,i);y15=c(15,2,i);y16=c(16,2,i);y17=c(17,2,i);
    y18=c(18,2,i);y19=c(19,2,i);y20=c(20,2,i);y21=c(21,2,i);y22=c(22,2,i);
    y23=c(23,2,i);y24=c(24,2,i);y25=c(25,2,i);y26=c(26,2,i);y27=c(27,2,i);

    z1=c(1,3,i);z2=c(2,3,i);z3=c(3,3,i);z4=c(4,3,i);z5=c(5,3,i);z6=c(6,3,i);
    z7=c(7,3,i);z8=c(8,3,i);z9=c(9,3,i);z10=c(10,3,i);z11=c(11,3,i);z12=c(12,3,i);
    z13=c(13,3,i);z14=c(14,3,i);z15=c(15,3,i);z16=c(16,3,i);z17=c(17,3,i);
    z18=c(18,3,i);z19=c(19,3,i);z20=c(20,3,i);z21=c(21,3,i);z22=c(22,3,i);
    z23=c(23,3,i);z24=c(24,3,i);z25=c(25,3,i);z26=c(26,3,i);z27=c(27,3,i);

    t(:, :, i)=[1 0 0 -dx; 0 1 0 -dy; 0 0 1 -dz; 0 0 0 1];

    r1= [x1; y1; z1; 1];r2= [x2; y2; z2; 1];r3= [x3; y3; z3; 1];r4= [x4; y4; z4; 1];
    r5= [x5; y5; z5; 1];r6= [x6; y6; z6; 1];r7= [x7; y7; z7; 1];r8= [x8; y8; z8; 1];
    r9= [x9; y9; z9; 1];r10= [x10; y10; z10; 1];r11= [x11; y11; z11; 1];
    r12= [x12; y12; z12; 1]; r13=[x13; y13; z13; 1];r14=[x14; y14; z14; 1];
    r15=[x15; y15; z15; 1];r16=[x16; y16; z16; 1]; r17=[x17; y17; z17; 1];
    r18=[x18; y18; z18; 1];r19=[x19; y19; z19; 1];r20=[x20; y20; z20; 1];
    r21=[x21; y21; z21; 1];r22=[x22; y22; z22; 1];r23=[x23; y23; z23; 1];
    r24=[x24; y24; z24; 1];r25=[x25; y25; z25; 1];r26=[x26; y26; z26; 1];
    r27=[x27; y27; z27; 1];

    T1=[t(:, :, i)*r1]';T2=[t(:, :, i)*r2]';T3=[t(:, :, i)*r3]';T4=[t(:, :, i)*r4]';
    T5=[t(:, :, i)*r5]';T6=[t(:, :, i)*r6]';T7=[t(:, :, i)*r7]';T8=[t(:, :, i)*r8]';
    T9=[t(:, :, i)*r9]';T10=[t(:, :, i)*r10]';T11=[t(:, :, i)*r11]';T12=[t(:, :, i)*r12]';
    T13=[t(:, :, i)*r13]';T14=[t(:, :, i)*r14]';T15=[t(:, :, i)*r15]';T16=[t(:, :, i)*r16]';
    T17=[t(:, :, i)*r17]';T18=[t(:, :, i)*r18]';T19=[t(:, :, i)*r19]';T20=[t(:, :, i)*r20]';
    T21=[t(:, :, i)*r21]';T22=[t(:, :, i)*r22]';T23=[t(:, :, i)*r23]';T24=[t(:, :, i)*r24]';
    T25=[t(:, :, i)*r25]';T26=[t(:, :, i)*r26]';T27=[t(:, :, i)*r27]';

    T(:, :, i)=[T1;T2;T3;T4;T5;T6;T7;T8;T9;T10;T11;T12;T13;T14;T15;T16;T17;T18;T19;...
        T20;T21;T22;T23;T24;T25;T26;T27];

    A(:, :, i)=atan(T(23,2,i)./T(23,1,i)); % cálculo del ángulo rotación
    A(:, :, i)=A(:, :, i)*180./pi; % Transformación de radianes a grados cercios

    rot(:, :, i)=[cos(A(:, :, i).*pi./180),sin(A(:, :, i).*pi./180),0,0;...
        -sin(A(:, :, i).*pi./180) cos(A(:, :, i).*pi./180) 0 0; ...
        0 0 1 0; 0 0 0 1];

```

```

X1=T(1,1,i);X2=T(2,1,i);X3=T(3,1,i);X4=T(4,1,i);X5=T(5,1,i);X6=T(6,1,i);
X7=T(7,1,i);X8=T(8,1,i);X9=T(9,1,i);X10=T(10,1,i);X11=T(11,1,i);X12=T(12,1,i);
X13=T(13,1,i);X14=T(14,1,i);X15=T(15,1,i);X16=T(16,1,i);X17=T(17,1,i);
X18=T(18,1,i);X19=T(19,1,i);X20=T(20,1,i);X21=T(21,1,i);X22=T(22,1,i);
X23=T(23,1,i);X24=T(24,1,i);X25=T(25,1,i);X26=T(26,1,i);X27=T(27,1,i);

Y1=T(1,2,i);Y2=T(2,2,i);Y3=T(3,2,i);Y4=T(4,2,i);Y5=T(5,2,i);Y6=T(6,2,i);
Y7=T(7,2,i);Y8=T(8,2,i);Y9=T(9,2,i);Y10=T(10,2,i);Y11=T(11,2,i);Y12=T(12,2,i);
Y13=T(13,2,i);Y14=T(14,2,i);Y15=T(15,2,i);Y16=T(16,2,i);Y17=T(17,2,i);
Y18=T(18,2,i);Y19=T(19,2,i);Y20=T(20,2,i);Y21=T(21,2,i);Y22=T(22,2,i);
Y23=T(23,2,i);Y24=T(24,2,i); Y25=T(25,2,i);Y26=T(26,2,i);Y27=T(27,2,i);

Z1=T(1,3,i);Z2=T(2,3,i);Z3=T(3,3,i);Z4=T(4,3,i);Z5=T(5,3,i);Z6=T(6,3,i);
Z7=T(7,3,i);Z8=T(8,3,i);Z9=T(9,3,i);Z10=T(10,3,i);Z11=T(11,3,i);Z12=T(12,3,i);
Z13=T(13,3,i);Z14=T(14,3,i);Z15=T(15,3,i);Z16=T(16,3,i);Z17=T(17,3,i);
Z18=T(18,3,i);Z19=T(19,3,i);Z20=T(20,3,i);Z21=T(21,3,i);Z22=T(22,3,i);
Z23=T(23,3,i);Z24=T(24,3,i);Z25=T(25,3,i);Z26=T(26,3,i);Z27=T(27,3,i);

R1= [X1; Y1; Z1; 1]; R2= [X2; Y2; Z2; 1]; R3= [X3; Y3; Z3; 1];
R4= [X4; Y4; Z4; 1]; R5= [X5; Y5; Z5; 1]; R6= [X6; Y6; Z6; 1];
R7= [X7; Y7; Z7; 1]; R8= [X8; Y8; Z8; 1]; R9= [X9; Y9; Z9; 1];
R10= [X10; Y10; Z10; 1]; R11= [X11; Y11; Z11; 1];R12= [X12; Y12; Z12; 1];
R13=[X13; Y13; Z13; 1]; R14=[X14; Y14; Z14; 1]; R15=[X15; Y15; Z15; 1];
R16=[X16; Y16; Z16; 1]; R17=[X17; Y17; Z17; 1]; R18=[X18; Y18; Z18; 1];
R19=[X19; Y19; Z19; 1]; R20=[X20; Y20; Z20; 1]; R21=[X21; Y21; Z21; 1];
R22=[X22; Y22; Z22; 1]; R23=[X23; Y23; Z23; 1]; R24=[X24; Y24; Z24; 1];
R25=[X25; Y25; Z25; 1]; R26=[X26; Y26; Z26; 1]; R27=[X27; Y27; Z27; 1];

Rot1=[rot(:, :, i)*R1]';Rot2=[rot(:, :, i)*R2]';Rot3=[rot(:, :, i)*R3]';
Rot4=[rot(:, :, i)*R4]';Rot5=[rot(:, :, i)*R5]';Rot6=[rot(:, :, i)*R6]';
Rot7=[rot(:, :, i)*R7]';Rot8=[rot(:, :, i)*R8]';Rot9=[rot(:, :, i)*R9]';
Rot10=[rot(:, :, i)*R10]';Rot11=[rot(:, :, i)*R11]';Rot12=[rot(:, :, i)*R12]';
Rot13=[rot(:, :, i)*R13]';Rot14=[rot(:, :, i)*R14]';Rot15=[rot(:, :, i)*R15]';
Rot16=[rot(:, :, i)*R16]';Rot17=[rot(:, :, i)*R17]';Rot18=[rot(:, :, i)*R18]';
Rot19=[rot(:, :, i)*R19]';Rot20=[rot(:, :, i)*R20]';Rot21=[rot(:, :, i)*R21]';
Rot22=[rot(:, :, i)*R22]';Rot23=[rot(:, :, i)*R23]';Rot24=[rot(:, :, i)*R24]';
Rot25=[rot(:, :, i)*R25]';Rot26=[rot(:, :, i)*R26]';Rot27=[rot(:, :, i)*R27]';

Rot(:, :, i)=[Rot1;Rot2;Rot3;Rot4;Rot5;Rot6;Rot7;Rot8;Rot9;Rot10;Rot11;...
Rot12;Rot13;Rot14;Rot15;Rot16;Rot17;Rot18;Rot19;Rot20;Rot21;Rot22;...
Rot23;Rot24;Rot25;Rot26;Rot27];
cn(:, :, i)=[Rot(:, 1, i) Rot(:, 2, i) Rot(:, 3, i)];
end

```

**vectxi.m**

```

function vxi= vectxi(n,Ne,cn);
% VXi - Vectores tangente en dirección Xi (Configuración deformada)
Datos;
vxi=zeros(3,size(Ne,2),size(Ne,1));
for i=1:size(Ne,1)
    DXi(1,1,i)=diff(n(1,1,i),Xi);DXi(1,2,i)=diff(n(2,1,i),Xi);
    DXi(1,3,i)=diff(n(3,1,i),Xi);DXi(1,4,i)=diff(n(4,1,i),Xi);
    DXi(1,5,i)=diff(n(5,1,i),Xi);DXi(1,6,i)=diff(n(6,1,i),Xi);
    DXi(1,7,i)=diff(n(7,1,i),Xi);DXi(1,8,i)=diff(n(8,1,i),Xi);
    DXi(1,9,i)=diff(n(9,1,i),Xi);DXi(1,10,i)=diff(n(10,1,i),Xi);
    DXi(1,11,i)=diff(n(11,1,i),Xi);DXi(1,12,i)=diff(n(12,1,i),Xi);
    DXi(1,13,i)=diff(n(13,1,i),Xi);DXi(1,14,i)=diff(n(14,1,i),Xi);
    DXi(1,15,i)=diff(n(15,1,i),Xi);DXi(1,16,i)=diff(n(16,1,i),Xi);
    DXi(1,17,i)=diff(n(17,1,i),Xi);DXi(1,18,i)=diff(n(18,1,i),Xi);
    DXi(1,19,i)=diff(n(19,1,i),Xi);DXi(1,20,i)=diff(n(20,1,i),Xi);
    DXi(1,21,i)=diff(n(21,1,i),Xi);DXi(1,22,i)=diff(n(22,1,i),Xi);
    DXi(1,23,i)=diff(n(23,1,i),Xi);DXi(1,24,i)=diff(n(24,1,i),Xi);
    DXi(1,25,i)=diff(n(25,1,i),Xi);DXi(1,26,i)=diff(n(26,1,i),Xi);
    DXi(1,27,i)=diff(n(27,1,i),Xi);

    vxi(1,1,i)=subs(DXi(1,1,i),{Xi, Eta, Dzeta},{cn(1,1,i),cn(1,2,i),cn(1,3,i)});
    vxi(1,2,i)=subs(DXi(1,2,i),{Xi, Eta, Dzeta},{cn(2,1,i),cn(2,2,i),cn(2,3,i)});
    vxi(1,3,i)=subs(DXi(1,3,i),{Xi, Eta, Dzeta},{cn(3,1,i),cn(3,2,i),cn(3,3,i)});
    vxi(1,4,i)=subs(DXi(1,4,i),{Xi, Eta, Dzeta},{cn(4,1,i),cn(4,2,i),cn(4,3,i)});
    vxi(1,5,i)=subs(DXi(1,5,i),{Xi, Eta, Dzeta},{cn(5,1,i),cn(5,2,i),cn(5,3,i)});
    vxi(1,6,i)=subs(DXi(1,6,i),{Xi, Eta, Dzeta},{cn(6,1,i),cn(6,2,i),cn(6,3,i)});
    vxi(1,7,i)=subs(DXi(1,7,i),{Xi, Eta, Dzeta},{cn(7,1,i),cn(7,2,i),cn(7,3,i)});
    vxi(1,8,i)=subs(DXi(1,8,i),{Xi, Eta, Dzeta},{cn(8,1,i),cn(8,2,i),cn(8,3,i)});
    vxi(1,9,i)=subs(DXi(1,9,i),{Xi, Eta, Dzeta},{cn(9,1,i),cn(9,2,i),cn(9,3,i)});

    vxi(1,10,i)=subs(DXi(1,10,i),{Xi,Eta,Dzeta},{cn(10,1,i),cn(10,2,i),cn(10,3,i)});
    vxi(1,11,i)=subs(DXi(1,11,i),{Xi,Eta,Dzeta},{cn(11,1,i),cn(11,2,i),cn(11,3,i)});
    vxi(1,12,i)=subs(DXi(1,12,i),{Xi,Eta,Dzeta},{cn(12,1,i),cn(12,2,i),cn(12,3,i)});
    vxi(1,13,i)=subs(DXi(1,13,i),{Xi,Eta,Dzeta},{cn(13,1,i),cn(13,2,i),cn(13,3,i)});
    vxi(1,14,i)=subs(DXi(1,14,i),{Xi,Eta,Dzeta},{cn(14,1,i),cn(14,2,i),cn(14,3,i)});
    vxi(1,15,i)=subs(DXi(1,15,i),{Xi,Eta,Dzeta},{cn(15,1,i),cn(15,2,i),cn(15,3,i)});
    vxi(1,16,i)=subs(DXi(1,16,i),{Xi,Eta,Dzeta},{cn(16,1,i),cn(16,2,i),cn(16,3,i)});
    vxi(1,17,i)=subs(DXi(1,17,i),{Xi,Eta,Dzeta},{cn(17,1,i),cn(17,2,i),cn(17,3,i)});
    vxi(1,18,i)=subs(DXi(1,18,i),{Xi,Eta,Dzeta},{cn(18,1,i),cn(18,2,i),cn(18,3,i)});
    vxi(1,19,i)=subs(DXi(1,19,i),{Xi,Eta,Dzeta},{cn(19,1,i),cn(19,2,i),cn(19,3,i)});
    vxi(1,20,i)=subs(DXi(1,20,i),{Xi,Eta,Dzeta},{cn(20,1,i),cn(20,2,i),cn(20,3,i)});
    vxi(1,21,i)=subs(DXi(1,21,i),{Xi,Eta,Dzeta},{cn(21,1,i),cn(21,2,i),cn(21,3,i)});
    vxi(1,22,i)=subs(DXi(1,22,i),{Xi,Eta,Dzeta},{cn(22,1,i),cn(22,2,i),cn(22,3,i)});
    vxi(1,23,i)=subs(DXi(1,23,i),{Xi,Eta,Dzeta},{cn(23,1,i),cn(23,2,i),cn(23,3,i)});
    vxi(1,24,i)=subs(DXi(1,24,i),{Xi,Eta,Dzeta},{cn(24,1,i),cn(24,2,i),cn(24,3,i)});
    vxi(1,25,i)=subs(DXi(1,25,i),{Xi,Eta,Dzeta},{cn(25,1,i),cn(25,2,i),cn(25,3,i)});
    vxi(1,26,i)=subs(DXi(1,26,i),{Xi,Eta,Dzeta},{cn(26,1,i),cn(26,2,i),cn(26,3,i)});
    vxi(1,27,i)=subs(DXi(1,27,i),{Xi,Eta,Dzeta},{cn(27,1,i),cn(27,2,i),cn(27,3,i)});
end

```

**vecteta.m**

```

function veta= vecteta(n,Ne,cn);
% Veta - vectores tangentes en dirección Eta (Configuración deformada)
Datos;
veta=zeros(3,size(Ne,2),size(Ne,1));
for i=1:size(Ne,1)
    DEta(1,1,i)=diff(n(1,1,i),Eta);DEta(1,2,i)=diff(n(2,1,i),Eta);
    DEta(1,3,i)=diff(n(3,1,i),Eta);DEta(1,4,i)=diff(n(4,1,i),Eta);
    DEta(1,5,i)=diff(n(5,1,i),Eta);DEta(1,6,i)=diff(n(6,1,i),Eta);
    DEta(1,7,i)=diff(n(7,1,i),Eta);DEta(1,8,i)=diff(n(8,1,i),Eta);
    DEta(1,9,i)=diff(n(9,1,i),Eta);DEta(1,10,i)=diff(n(10,1,i),Eta);
    DEta(1,11,i)=diff(n(11,1,i),Eta);DEta(1,12,i)=diff(n(12,1,i),Eta);
    DEta(1,13,i)=diff(n(13,1,i),Eta);DEta(1,14,i)=diff(n(14,1,i),Eta);
    DEta(1,15,i)=diff(n(15,1,i),Eta);DEta(1,16,i)=diff(n(16,1,i),Eta);
    DEta(1,17,i)=diff(n(17,1,i),Eta);DEta(1,18,i)=diff(n(18,1,i),Eta);
    DEta(1,19,i)=diff(n(19,1,i),Eta);DEta(1,20,i)=diff(n(20,1,i),Eta);
    DEta(1,21,i)=diff(n(21,1,i),Eta);DEta(1,22,i)=diff(n(22,1,i),Eta);
    DEta(1,23,i)=diff(n(23,1,i),Eta);DEta(1,24,i)=diff(n(24,1,i),Eta);
    DEta(1,25,i)=diff(n(25,1,i),Eta);DEta(1,26,i)=diff(n(26,1,i),Eta);
    DEta(1,27,i)=diff(n(27,1,i),Eta);

    veta(2,1,i)=subs(DEta(1,1,i),{Xi, Eta, Dzeta},{cn(1,1,i),cn(1,2,i),cn(1,3,i)});
    veta(2,2,i)=subs(DEta(1,2,i),{Xi, Eta, Dzeta},{cn(2,1,i),cn(2,2,i),cn(2,3,i)});
    veta(2,3,i)=subs(DEta(1,3,i),{Xi, Eta, Dzeta},{cn(3,1,i),cn(3,2,i),cn(3,3,i)});
    veta(2,4,i)=subs(DEta(1,4,i),{Xi, Eta, Dzeta},{cn(4,1,i),cn(4,2,i),cn(4,3,i)});
    veta(2,5,i)=subs(DEta(1,5,i),{Xi, Eta, Dzeta},{cn(5,1,i),cn(5,2,i),cn(5,3,i)});
    veta(2,6,i)=subs(DEta(1,6,i),{Xi, Eta, Dzeta},{cn(6,1,i),cn(6,2,i),cn(6,3,i)});
    veta(2,7,i)=subs(DEta(1,7,i),{Xi, Eta, Dzeta},{cn(7,1,i),cn(7,2,i),cn(7,3,i)});
    veta(2,8,i)=subs(DEta(1,8,i),{Xi, Eta, Dzeta},{cn(8,1,i),cn(8,2,i),cn(8,3,i)});
    veta(2,9,i)=subs(DEta(1,9,i),{Xi, Eta, Dzeta},{cn(9,1,i),cn(9,2,i),cn(9,3,i)});
    veta(2,10,i)=subs(DEta(1,10,i),{Xi,Eta,Dzeta},{cn(10,1,i),cn(10,2,i),cn(10,3,i)});
    veta(2,11,i)=subs(DEta(1,11,i),{Xi,Eta,Dzeta},{cn(11,1,i),cn(11,2,i),cn(11,3,i)});
    veta(2,12,i)=subs(DEta(1,12,i),{Xi,Eta,Dzeta},{cn(12,1,i),cn(12,2,i),cn(12,3,i)});
    veta(2,13,i)=subs(DEta(1,13,i),{Xi,Eta,Dzeta},{cn(13,1,i),cn(13,2,i),cn(13,3,i)});
    veta(2,14,i)=subs(DEta(1,14,i),{Xi,Eta,Dzeta},{cn(14,1,i),cn(14,2,i),cn(14,3,i)});
    veta(2,15,i)=subs(DEta(1,15,i),{Xi,Eta,Dzeta},{cn(15,1,i),cn(15,2,i),cn(15,3,i)});
    veta(2,16,i)=subs(DEta(1,16,i),{Xi,Eta,Dzeta},{cn(16,1,i),cn(16,2,i),cn(16,3,i)});
    veta(2,17,i)=subs(DEta(1,17,i),{Xi,Eta,Dzeta},{cn(17,1,i),cn(17,2,i),cn(17,3,i)});
    veta(2,18,i)=subs(DEta(1,18,i),{Xi,Eta,Dzeta},{cn(18,1,i),cn(18,2,i),cn(18,3,i)});
    veta(2,19,i)=subs(DEta(1,19,i),{Xi,Eta,Dzeta},{cn(19,1,i),cn(19,2,i),cn(19,3,i)});
    veta(2,20,i)=subs(DEta(1,20,i),{Xi,Eta,Dzeta},{cn(20,1,i),cn(20,2,i),cn(20,3,i)});
    veta(2,21,i)=subs(DEta(1,21,i),{Xi,Eta,Dzeta},{cn(21,1,i),cn(21,2,i),cn(21,3,i)});
    veta(2,22,i)=subs(DEta(1,22,i),{Xi,Eta,Dzeta},{cn(22,1,i),cn(22,2,i),cn(22,3,i)});
    veta(2,23,i)=subs(DEta(1,23,i),{Xi,Eta,Dzeta},{cn(23,1,i),cn(23,2,i),cn(23,3,i)});
    veta(2,24,i)=subs(DEta(1,24,i),{Xi,Eta,Dzeta},{cn(24,1,i),cn(24,2,i),cn(24,3,i)});
    veta(2,25,i)=subs(DEta(1,25,i),{Xi,Eta,Dzeta},{cn(25,1,i),cn(25,2,i),cn(25,3,i)});
    veta(2,26,i)=subs(DEta(1,26,i),{Xi,Eta,Dzeta},{cn(26,1,i),cn(26,2,i),cn(26,3,i)});
    veta(2,27,i)=subs(DEta(1,27,i),{Xi,Eta,Dzeta},{cn(27,1,i),cn(27,2,i),cn(27,3,i)});
end

```

**vectdzeta.m**

```

function vdzeta= vectdzeta(n,Ne,cn);
% VDzeta - vectores tangentes en dirección Dzeta (Configuración deformada)
Datos;
    vdzeta=zeros(3,size(Ne,2),size(Ne,1));
for i=1:size(Ne,1)
    DDzeta(1,1,i)=diff(n(1,1,i),Dzeta);DDzeta(1,2,i)=diff(n(2,1,i),Dzeta);
    DDzeta(1,3,i)=diff(n(3,1,i),Dzeta);DDzeta(1,4,i)=diff(n(4,1,i),Dzeta);
    DDzeta(1,5,i)=diff(n(5,1,i),Dzeta);DDzeta(1,6,i)=diff(n(6,1,i),Dzeta);
    DDzeta(1,7,i)=diff(n(7,1,i),Dzeta);DDzeta(1,8,i)=diff(n(8,1,i),Dzeta);
    DDzeta(1,9,i)=diff(n(9,1,i),Dzeta);DDzeta(1,10,i)=diff(n(10,1,i),Dzeta);
    DDzeta(1,11,i)=diff(n(11,1,i),Dzeta);DDzeta(1,12,i)=diff(n(12,1,i),Dzeta);
    DDzeta(1,13,i)=diff(n(13,1,i),Dzeta);DDzeta(1,14,i)=diff(n(14,1,i),Dzeta);
    DDzeta(1,15,i)=diff(n(15,1,i),Dzeta);DDzeta(1,16,i)=diff(n(16,1,i),Dzeta);
    DDzeta(1,17,i)=diff(n(17,1,i),Dzeta);DDzeta(1,18,i)=diff(n(18,1,i),Dzeta);
    DDzeta(1,19,i)=diff(n(19,1,i),Dzeta);DDzeta(1,20,i)=diff(n(20,1,i),Dzeta);
    DDzeta(1,21,i)=diff(n(21,1,i),Dzeta);DDzeta(1,22,i)=diff(n(22,1,i),Dzeta);
    DDzeta(1,23,i)=diff(n(23,1,i),Dzeta);DDzeta(1,24,i)=diff(n(24,1,i),Dzeta);
    DDzeta(1,25,i)=diff(n(25,1,i),Dzeta);DDzeta(1,26,i)=diff(n(26,1,i),Dzeta);
    DDzeta(1,27,i)=diff(n(27,1,i),Dzeta);

    vdzeta(3,1,i)=subs(DDzeta(1,1,i),{Xi, Eta, Dzeta},{cn(1,1,i),cn(1,2,i),cn(1,3,i)});
    vdzeta(3,2,i)=subs(DDzeta(1,2,i),{Xi, Eta, Dzeta},{cn(2,1,i),cn(2,2,i),cn(2,3,i)});
    vdzeta(3,3,i)=subs(DDzeta(1,3,i),{Xi, Eta, Dzeta},{cn(3,1,i),cn(3,2,i),cn(3,3,i)});
    vdzeta(3,4,i)=subs(DDzeta(1,4,i),{Xi, Eta, Dzeta},{cn(4,1,i),cn(4,2,i),cn(4,3,i)});
    vdzeta(3,5,i)=subs(DDzeta(1,5,i),{Xi, Eta, Dzeta},{cn(5,1,i),cn(5,2,i),cn(5,3,i)});
    vdzeta(3,6,i)=subs(DDzeta(1,6,i),{Xi, Eta, Dzeta},{cn(6,1,i),cn(6,2,i),cn(6,3,i)});
    vdzeta(3,7,i)=subs(DDzeta(1,7,i),{Xi, Eta, Dzeta},{cn(7,1,i),cn(7,2,i),cn(7,3,i)});
    vdzeta(3,8,i)=subs(DDzeta(1,8,i),{Xi, Eta, Dzeta},{cn(8,1,i),cn(8,2,i),cn(8,3,i)});
    vdzeta(3,9,i)=subs(DDzeta(1,9,i),{Xi, Eta, Dzeta},{cn(9,1,i),cn(9,2,i),cn(9,3,i)});
    vdzeta(3,10,i)=subs(DDzeta(1,10,i),{Xi, Eta, Dzeta},{cn(10,1,i),cn(10,2,i),cn(10,3,i)});
    vdzeta(3,11,i)=subs(DDzeta(1,11,i),{Xi, Eta, Dzeta},{cn(11,1,i),cn(11,2,i),cn(11,3,i)});
    vdzeta(3,12,i)=subs(DDzeta(1,12,i),{Xi, Eta, Dzeta},{cn(12,1,i),cn(12,2,i),cn(12,3,i)});
    vdzeta(3,13,i)=subs(DDzeta(1,13,i),{Xi, Eta, Dzeta},{cn(13,1,i),cn(13,2,i),cn(13,3,i)});
    vdzeta(3,14,i)=subs(DDzeta(1,14,i),{Xi, Eta, Dzeta},{cn(14,1,i),cn(14,2,i),cn(14,3,i)});
    vdzeta(3,15,i)=subs(DDzeta(1,15,i),{Xi, Eta, Dzeta},{cn(15,1,i),cn(15,2,i),cn(15,3,i)});
    vdzeta(3,16,i)=subs(DDzeta(1,16,i),{Xi, Eta, Dzeta},{cn(16,1,i),cn(16,2,i),cn(16,3,i)});
    vdzeta(3,17,i)=subs(DDzeta(1,17,i),{Xi, Eta, Dzeta},{cn(17,1,i),cn(17,2,i),cn(17,3,i)});
    vdzeta(3,18,i)=subs(DDzeta(1,18,i),{Xi, Eta, Dzeta},{cn(18,1,i),cn(18,2,i),cn(18,3,i)});
    vdzeta(3,19,i)=subs(DDzeta(1,19,i),{Xi, Eta, Dzeta},{cn(19,1,i),cn(19,2,i),cn(19,3,i)});
    vdzeta(3,20,i)=subs(DDzeta(1,20,i),{Xi, Eta, Dzeta},{cn(20,1,i),cn(20,2,i),cn(20,3,i)});
    vdzeta(3,21,i)=subs(DDzeta(1,21,i),{Xi, Eta, Dzeta},{cn(21,1,i),cn(21,2,i),cn(21,3,i)});
    vdzeta(3,22,i)=subs(DDzeta(1,22,i),{Xi, Eta, Dzeta},{cn(22,1,i),cn(22,2,i),cn(22,3,i)});
    vdzeta(3,23,i)=subs(DDzeta(1,23,i),{Xi, Eta, Dzeta},{cn(23,1,i),cn(23,2,i),cn(23,3,i)});
    vdzeta(3,24,i)=subs(DDzeta(1,24,i),{Xi, Eta, Dzeta},{cn(24,1,i),cn(24,2,i),cn(24,3,i)});
    vdzeta(3,25,i)=subs(DDzeta(1,25,i),{Xi, Eta, Dzeta},{cn(25,1,i),cn(25,2,i),cn(25,3,i)});
    vdzeta(3,26,i)=subs(DDzeta(1,26,i),{Xi, Eta, Dzeta},{cn(26,1,i),cn(26,2,i),cn(26,3,i)});
    vdzeta(3,27,i)=subs(DDzeta(1,27,i),{Xi, Eta, Dzeta},{cn(27,1,i),cn(27,2,i),cn(27,3,i)});
end

```

### **matdervfuncd.m**

```
function mdf= matdervfuncd(vxi,veta,vdzeta);  
% MDF - Matriz de derivadas de las funciones de forma de la  
% (Configuración deformada)  
A=vxi(1, :, :);  
B=veta(2, :, :);  
C=vdzeta(3, :, :);  
mdf= [A;B;C];
```

### **jacobiandef.m**

```
function j= jacobiandef(mdf,c);  
% Cálculo de la matriz jacobiana (Configuración deformada)  
  
for i=1:size(mdf,3)  
    j(:, :, i)=mdf(:, :, i)*c(:, :, i);  
end
```

### **tensordefor.m**

```
function F=tensordefor(J,j);  
% F - Tensor Deformación  
for i=1:size(J,3)  
    F(:, :, i)=j(:, :, i)*inv(J(:, :, i));  
end
```