

UNIVERSITAT POLITÈCNICA DE CATALUNYA
ESCOLA TÈCNICA SUPERIOR D'ENGINYERS
DE CAMINS, CANALS I PORTS

**SIMULACIÓN NUMÉRICA DE PROCESOS DE
COMPACTACIÓN DE PULVIMATERIALES.
APLICACIÓN DE TÉCNICAS DE CÁLCULO PARALELO**

TESIS DOCTORAL

PRESENTADA POR:

JUAN CARLOS CANTE TERÁN

DIRIGIDA POR LOS PROFESORES:

DR. JAVIER OLIVER OLIVELLA

Y

DR. SERGIO OLLER MARTÍNEZ

BARCELONA - ABRIL DE 1995.

CAPÍTULO 4

EJEMPLOS DE VALIDACIÓN Y SIMULACIÓN NUMÉRICA

4.1 INTRODUCCIÓN

En este capítulo se estudia el comportamiento del modelo numérico propuesto en el capítulo 3.

En la primera parte se llevan a cabo una serie de ejemplos con los que se pretende validar el modelo numérico, comparándolo con los resultados experimentales obtenidos por diferentes autores. El primer ejemplo consiste en un ensayo de compresión hidrostática y un ensayo triaxial, mediante los cuales se pretende analizar la idoneidad de las reglas de flujo propuestas. El segundo ejemplo consta de dos ensayos, el primero de compresión axial y el segundo de compresión hidrostática, que permiten confirmar el tipo de regla de flujo que debe utilizarse simultáneamente con la función de fluencia empleada aquí. El último ejemplo de esta parte, consiste en la compactación uniaxial de dos muestras cilíndricas de diferentes alturas, de las cuales se tienen experimentalmente, los valores medios de las tensiones en las paredes. Por medio de este ejemplo se pretende validar el estado tensional del modelo numérico.

La segunda parte de ejemplos tiene por objeto mostrar la utilidad que tiene la herramienta numérica en el diseño de piezas complejas. El primer ejemplo de esta serie, consiste en la compactación de tres muestras cilíndricas de diferentes diámetros, que definen diferentes relaciones de esbeltez, mediante el cual se estudia el efecto que tiene la fricción en las propiedades finales del compacto, a medida que la esbeltez de la muestra aumenta. El segundo ejemplo consiste en el diseño del proceso de compactación de una pieza a dos niveles, se presenta un proceso donde los resultados obtenidos son deficientes y otro donde el estado final del compacto es óptimo. El último ejemplo, en el que se pretende diseñar una pieza a tres niveles, muestra como puede usarse la herramienta numérica, para analizar todo el proceso de compactación. El proceso se ha dividido en tres etapas: la primera consiste en la compactación mediante el desplazamiento de cinco punzones en forma simultánea. La segunda etapa consiste en una descarga parcial, que se obtiene después de eliminar los punzones superiores.

En la última etapa se realiza la extracción del compacto de la matriz, mediante el empleo de los punzones inferiores. A lo largo de todo el proceso, se analiza la evolución de la densidad y las tensiones.

4.2 EJEMPLOS DE VALIDACIÓN

4.2.1 Comparación entre la regla de flujo asociada y no asociada

Dorémus y sus colaboradores [27], diseñaron una prensa triaxial (figura 4.1.b), para estudiar el comportamiento de metales en polvo. La prensa se controla numéricamente y permite la combinación de diferentes estados de carga, controlando la tensión y/o la deformación. Mediante esta prensa, se consigue realizar pruebas de compresión isostática y ensayos triaxiales. El límite de la presión de confinamiento alcanza los 400 Mpa (tensión radial), mientras que el máximo de la tensión axial es de 1250 Mpa.

Las pruebas se dividieron en dos grupos, un primer grupo de ensayos de compresión hidrostática y otro de ensayos triaxiales. En todos los ensayos se empleó polvo de hierro de forma irregular y de tamaño entre $10\mu\text{m}$ y $100\mu\text{m}$, y se partió de una densidad inicial de 3.67 gr/cm^3 . Las muestras cilíndricas tenían un diámetro de 20mm y una altura inicial de 24mm (figura 4.1.a). Durante la compresión isostática, la presión sobre la muestra se incrementó a una velocidad de 2.5 Mpa/s hasta que se alcanzó una presión entre 50 Mpa y 400 Mpa, esta presión se mantuvo durante tres minutos. En los ensayos triaxiales, la primera parte, que corresponde a la compresión isostática se llevó a cabo de forma similar al caso anterior, posteriormente a esta carga se aumentó la presión en la dirección axial a razón de 0.1 mm/s , hasta alcanzar una presión máxima de 1250 Mpa. Estos ensayos se realizaron a diferentes presiones de confinamiento.

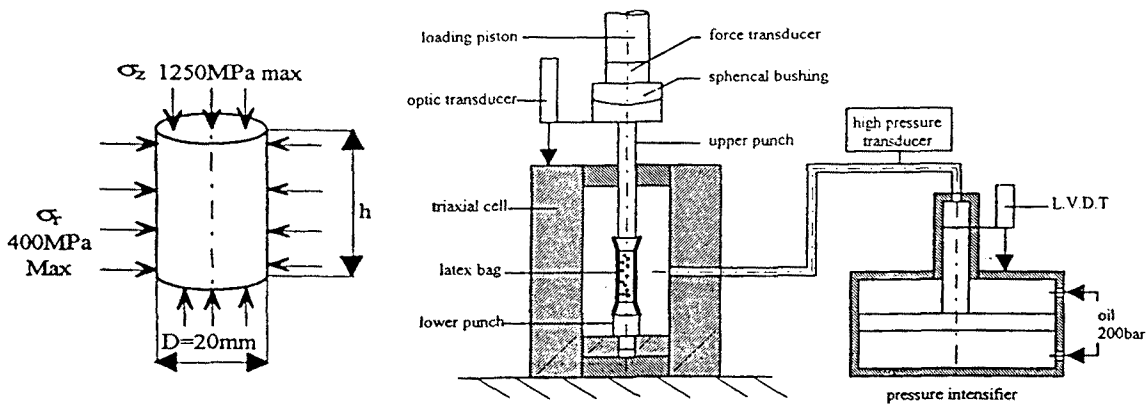


Figura 4.1 (a) Geometría inicial, (b) Esquema de la prensa triaxial [27]

Validación frente a ensayos de compresión isostática

La figura 4.2.a representa la mitad de la muestra cilíndrica que se somete a compresión isostática. La malla consta de 75 elementos (elementos de 4 nodos) y 96 nodos. En las figuras 4.2.b y c se muestran las configuraciones intermedia y final que se obtienen, después de someter a la muestra a una presión hidrostática de 400 Mpa. La aplicación de la carga se lleva a cabo en 15 incrementos, el tamaño de los incrementos varía de forma cuadrática, pequeños al principio y más grandes al final. Se estiman los siguientes valores de los parámetros que dependen del tipo de material:

$$\text{módulo de Young} = 2.0 \times 10^3 \text{ Mpa}$$

$$\text{módulo de Poisson} = 0.37$$

$$\text{densidad relativa inicial} = \eta_0 = \frac{3.67}{7.5} = 0.489$$

$$\text{límite de fluencia} = 90 \text{ Mpa}$$

Los parámetros que se eligen para la superficie de fluencia son:

$$a_1(\eta) = \frac{1 - \eta^2}{2 + \eta^2} \quad a_2(\eta) = \left[\frac{\eta - \eta_0}{1 - \eta_0} \right]^{2.7}$$

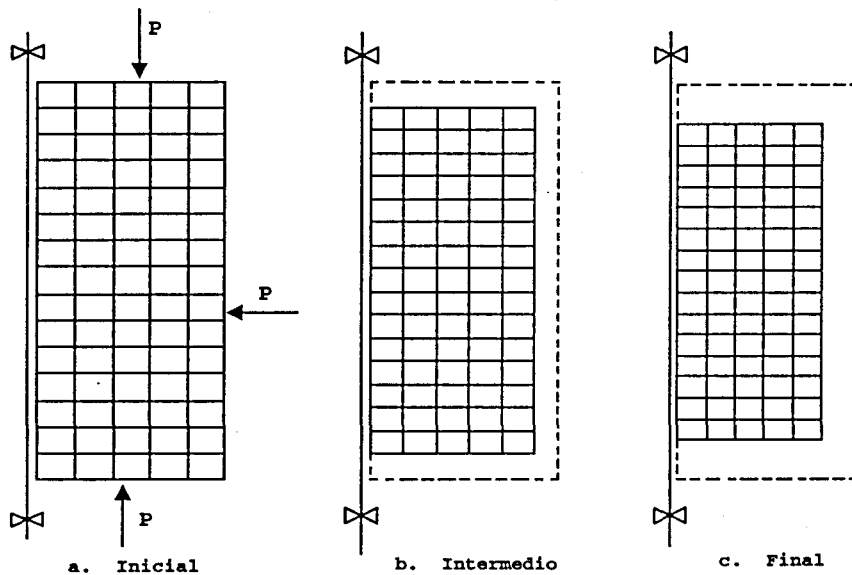


Figura 4.2 Compresión isostática

La figura 4.3 presenta la evolución de la densidad frente a la presión hidrostática. En las figuras 4.3.a y 4.3.b se encuentran los resultados expe-

rimentales obtenidos por Dorémus (y sus colaboradores) [27] y los obtenidos en el presente trabajo.

Los resultados de la figura 4.3.a se obtienen utilizando una regla de flujo *no asociada* mientras que los de la figura 4.3.b se obtienen utilizando una regla de flujo *asociada*. La similitud en los resultados es bastante satisfactoria. Además no se percibe ningún efecto debido al cambio en la regla de flujo.

Validación frente a ensayos triaxiales

La validación de la regla de flujo se lleva a cabo mediante la comparación entre los resultados de los ensayos triaxiales obtenidos por Dorémus (y sus colaboradores) y los obtenidos aquí considerando las dos reglas de flujo. Las figuras 4.4.a, 4.4.b y 4.4.c, representan las configuraciones inicial, intermedia y final, que se obtienen al aumentar la presión en la dirección axial hasta alcanzar cerca de los 800 Mpa. La figuras 4.5.a y 4.5.b muestran la variación de la densidad contra la deformación axial, para diferentes presiones de confinamiento. Se considera aquí que la deformación axial viene dada por la siguiente expresión: $(L_f - L_o)/L_o$, donde L_o corresponde a la longitud de la muestra antes de iniciar la aplicación de la presión axial y L_f a la longitud de la muestra correspondiente al nivel de presión aplicado. En las dos figuras (4.5.a, 4.5.b) se encuentran los resultados experimentales para diferentes presiones de confinamiento, 50, 150 y 400 Mpa. En la figura 4.5.a se presentan los resultados numéricos obtenidos cuando se emplea la regla de flujo *no asociada* y en la figura 4.5.b los resultados numéricos utilizando la regla de flujo *asociada*. Los resultados numéricos obtenidos en el primer caso, se alejan completamente de los resultados experimentales, que muestran como la densidad tiende a la densidad teórica a medida que la presión axial se incrementa. En el segundo caso se observa como los resultados numéricos son bastante más satisfactorios y se consigue simular de forma aceptable el efecto asintótico de la densidad. A continuación se analiza el porqué de estos resultados.

Como se explicó en el capítulo 3, los parámetros de la superficie de fluencia se ajustan de tal forma que al final cuando la densidad relativa η , tiende a uno (es decir cuando la densidad se acerca a la densidad teórica) la superficie de fluencia tiende al cilindro de Von Mises. En el caso de flujo asociado el vector de flujo plástico, por definición, apunta en la dirección normal a la superficie y en las etapas finales del proceso, esta dirección casi coincide con la normal al cilindro de Von Mises, que se caracteriza por ser un modelo de plasticidad incompresible. Esto implica que la deformación plástica volumétrica tiene un valor límite, que a su vez implica un valor límite en la densidad relativa, ya que la deformación total del compacto es esencialmente deformación plástica. En el caso del flujo no asociado el vector de flujo plástico apunta al origen y aunque la superficie de fluencia se acerca al cilindro de Von Mises, el vector de flujo tiene una componente, que nunca se anula en la dirección del eje hidrostático, haciendo

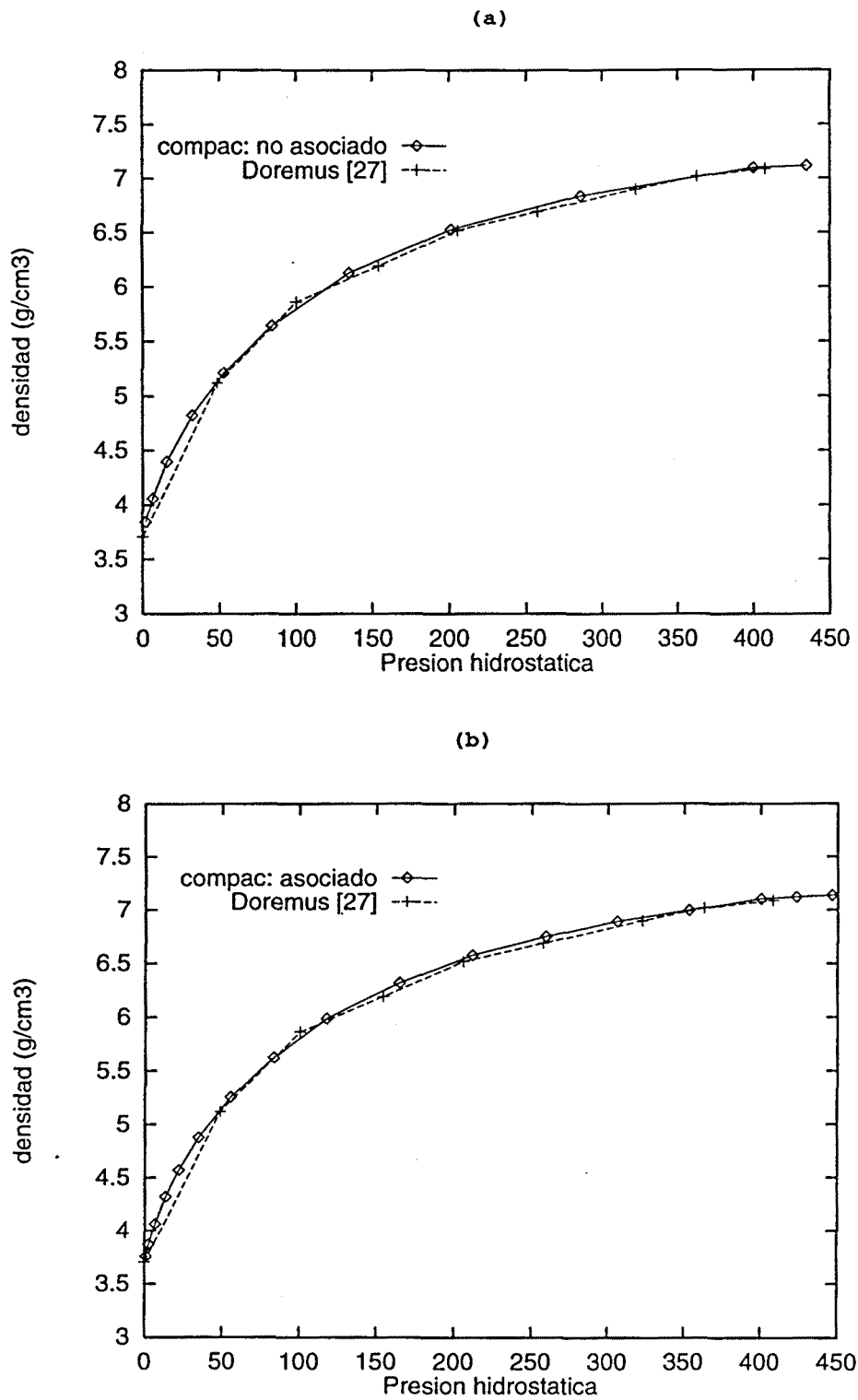


Figura 4.3 Compactación isostática: a) regla de flujo no asociada, b) regla de flujo asociada

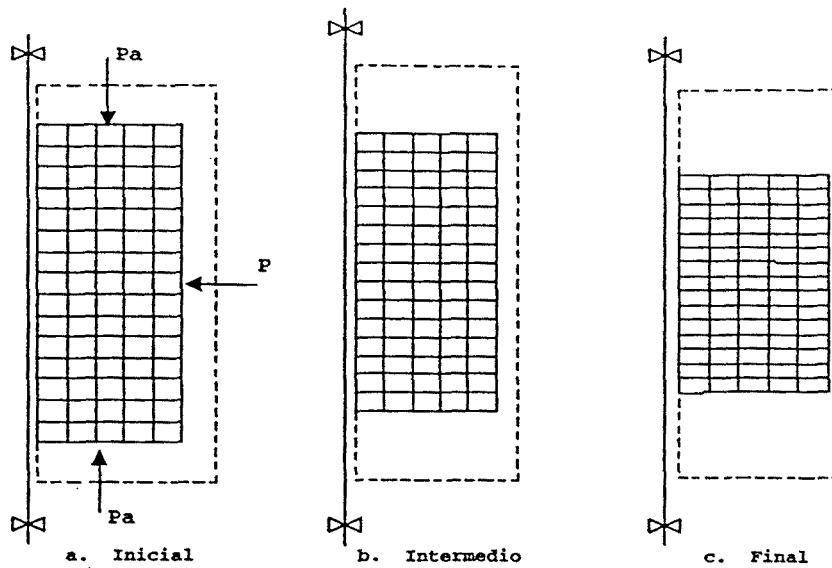


Figura 4.4 Ensayo triaxial: compresión axial e hidrostática: a. estado inicial, b. estado intermedio, c. estado final

que la deformación volumétrica aumente de forma indefinida. Este problema es similar al que algunos investigadores [10][18][14] ponen de manifiesto cuando se emplea una superficie de fluencia definida mediante múltiples superficies y la forma de la superficie que limita la parte de compresión hidrostática permanece invariante. La razón es que, junto con esta superficie se utiliza una regla de flujo asociada, y la componente del vector plástico sobre el eje hidrostático es nuevamente, no nula. Es importante notar que este efecto no se puso en evidencia con los resultados obtenidos en el caso anterior de compresión isostática. Esto es debido a que en este caso, al ser el estado de tensiones hidrostático, las dos direcciones de flujo, tanto la asociada como la no asociada, coinciden.

4.2.2 Validación de la regla de flujo asociada

Ernst y Barnekow [20], diseñaron una prensa experimental, que permite tomar datos de las fuerzas axiales y desplazamientos de todos los elementos en movimiento durante un proceso de compactación uniaxial y de extracción. Las muestras que se analizaron tenían un diámetro de 14.3 mm y una altura máxima de llenado 80 mm. El material empleado en los experimentos fue: 100.29 (Höganäs) con 1% Microwax C (Hoechst). Ernst y Barnekow realizaron una gran variedad de experimentos, de los cuales son de interés aquí un proceso de compactación uniaxial y un proceso de compactación isostático, que permiten validar la regla de flujo asociada en el modelo numérico.

La figura 4.6.a representa la mitad de la muestra cilíndrica de diámetro 14.3 mm y altura 80mm, que en el primer caso, se desea someter la muestra a un

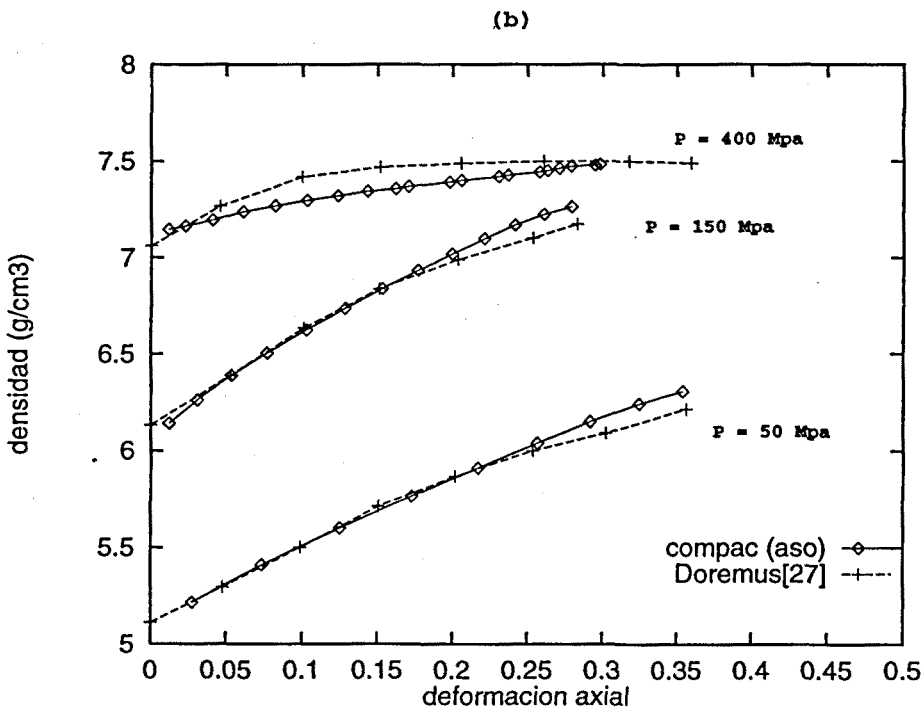
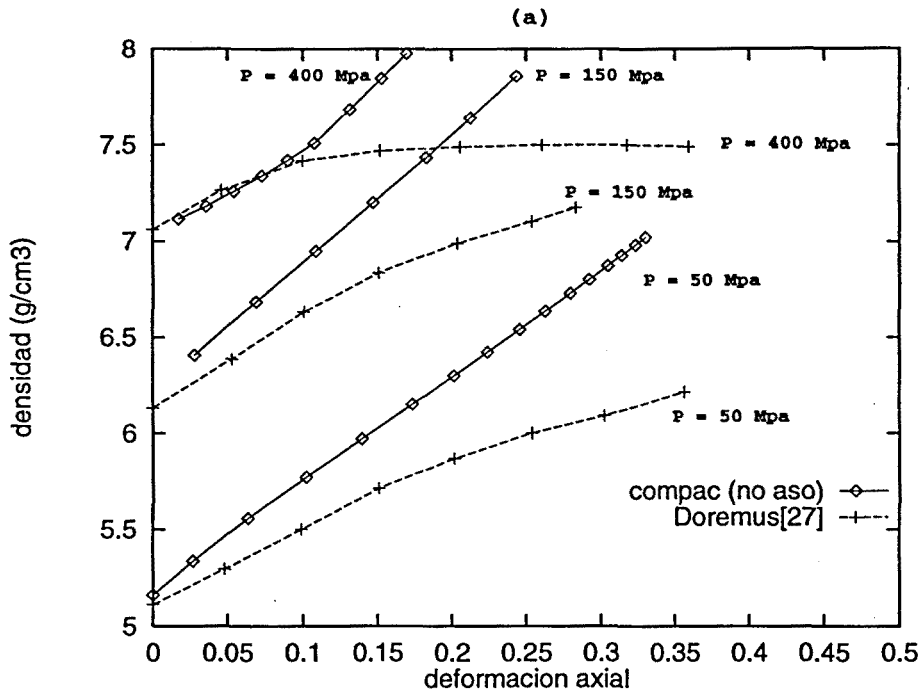


Figura 4.5 Ensayos triaxiales: comparación entre las reglas de flujo a) no asociada y b) asociada

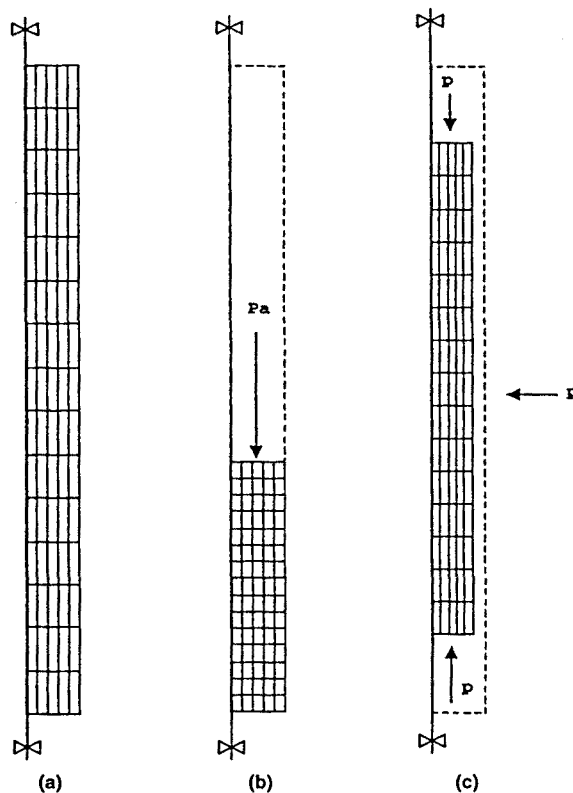


Figura 4.6 Procesos de compactación uniaxial e isostática. a) malla de elementos finitos, b) compactación uniaxial, c) compactación isostática

proceso de compactación uniaxial (figura 4.6.b) y en el segundo a un proceso de compactación isostática (figura 4.6.c). La malla consta de 75 elementos (de 4 nodos) y 96 nodos. Se estiman los siguientes valores de los parámetros que dependen del tipo de material:

$$\text{módulo de Young} = 5.0 \times 10^4 \text{ Mpa}$$

$$\text{módulo de Poisson} = 0.37$$

$$\text{densidad relativa inicial} = \eta_o = \frac{2.94}{7.35} = 0.4$$

$$\text{límite de fluencia} = 170 \text{ Mpa}$$

Los parámetros que se eligen para la superficie de fluencia son:

$$a_1(\eta) = \left[\frac{1 - \eta^2}{2 + \eta^2} \right]^{0.3} \quad a_2(\eta) = \left[\frac{\eta - \eta_o}{1 - \eta_o} \right]^{5.4}$$

En el primer caso (compactación uniaxial) el segmento superior se desplaza una distancia de 48mm (en diez incrementos) que equivale aproximadamente, a la

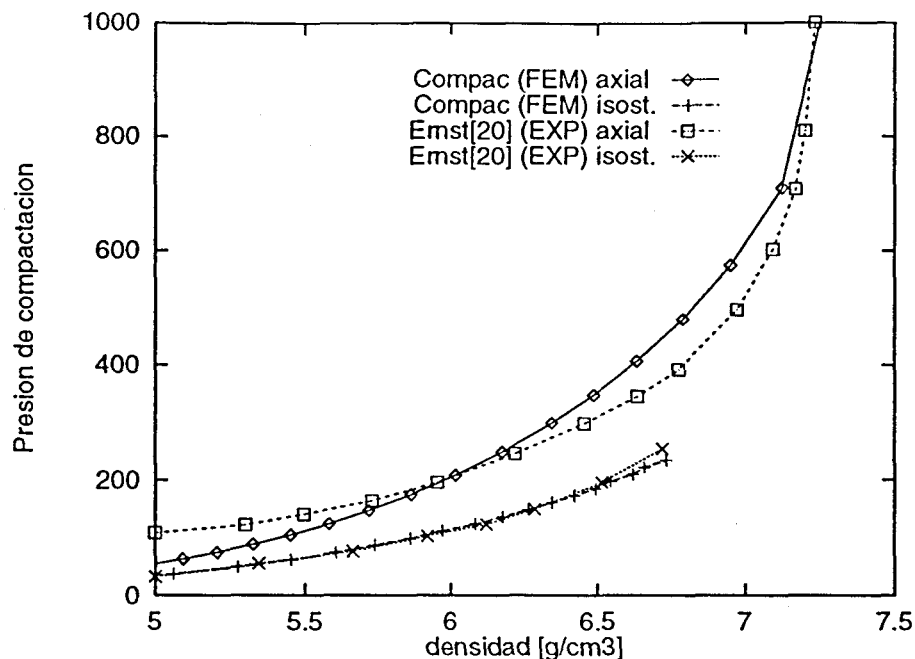


Figura 4.7 Procesos de compactación uniaxial e isostática

aplicación de 900 Mpa en la dirección axial. En el segundo caso (isostático), se somete la muestra a una presión hidrostática de 250 Mpa (en 15 incrementos). En las figuras 4.6.b y 4.6.c, se muestran las configuraciones finales en los dos casos considerados.

En la figura 4.7 se presentan los resultados experimentales obtenidos por Ernst, junto con los resultados numéricos obtenidos en este trabajo con la regla de flujo asociada, para la variación de la presión de compactación contra la densidad. En el caso isostático los resultados numéricos son bastante satisfactorios. En el caso de la compactación uniaxial, los resultados son aceptables aunque pueden mejorarse mediante la elección de otro tipo de curva para el parámetro a_2 , pero ya se puede apreciar como a medida que la presión axial aumenta, la densidad tiende a la densidad teórica del material compactado. Esto confirma que el uso de una regla de *flujo asociada* es lo más apropiado, dada la superficie de fluencia que se emplea aquí, para la simulación del proceso de compactación.

4.2.3 Validación del estado tensional

Gethin y Lewis en [13], realizaron una serie de experimentos usando polvo de hierro con los siguientes componentes: 97% hierro SC100, 2% cobre y 1% zinc. Los parámetros del material (proporcionados por los autores) son:

módulo de Young = 40Mpa

módulo de Poisson = 0.37

ángulo de cohesión = 0°

ángulo de fricción = 30°

El coeficiente de fricción utilizado fue 0.15. Gethin y Lewis realizaron la compactación uniaxial de dos tipos de muestras cilíndricas, ambas de 25mm de diámetro externo y 17mm de diámetro interno. Las alturas de llenado fueron 20mm y 80mm respectivamente. La reducción en la altura fue del 50% y la densidad final fue cerca del 80% de la densidad del metal base. El valor que se estima aquí para la densidad relativa es $\eta_o = 0.41$ y de 12 Mpa para la tensión máxima de fluencia. Los parámetros que se eligen para la superficie de fluencia son:

$$a_1(\eta) = \left[\frac{1 - \eta^2}{2 + \eta^2} \right]^{0.5} \quad a_2(\eta) = \left[\frac{\eta - \eta_o}{1 - \eta_o} \right]^{2.2}$$

El conjunto de figuras que se encuentran en la parte superior de la figura 4.8, representan, a) la malla de elementos finitos, que se ha emplado para realizar los cálculos (la figura representa un corte axial del cilindro, el hueco se encuentra del lado izquierdo), que consta de 250 elementos y 286 nodos. b) la deformada y las densidades en un instante intermedio y c) la deformada y la variación de las densidades al final de la compactación. El efecto de la fricción se puede observar claramente en esta última, donde las densidades son más altas en la parte superior que en la inferior del cilindro. Este efecto será analizado en detalle más adelante. La figura 4.9 presenta los resultados experimentales y numéricos obtenidos por Gethin-Lewis y los resultados obtenidos en este trabajo. Las curvas corresponden a las variaciones de las tensiones en cada una de las caras de la muestra cilíndrica (arriba, abajo, exterior e interior) contra la deformación. Se estimó aquí que esta deformación viene dada por la deformación nominal calculada como: $(L_f - L_o)/L_o$, donde L_o es la longitud inicial de la muestra y L_f es la longitud de la muestra correspondiente al nivel de presión aplicado. Los resultados son bastante aceptables, aunque cabe notar una mayor discrepancia de los resultados en las caras de abajo y la exterior, donde el error relativo máximo (para 50% de la deformación) respecto a los resultados experimentales es del orden del 40% y 10% respectivamente. Esta discrepancia puede ser debida a que el efecto de la fricción es un poco mayor que el estimado.

La compactación uniaxial del cilindro de 80mm de altura se representa en el conjunto de figuras que se encuentran en la parte inferior de la figura 4.8, en

donde se tiene que, la figura de la izquierda es la malla empleada para realizar los cálculos, las dos del centro son: la deformada y las densidades en un instante intermedio y las de la derecha son: la deformada y las densidades al final del proceso de compactación. Se puede apreciar en esta última, que el efecto de la fricción es bastante más pronunciado que en el caso del cilindro de 20mm. La figura 4.10, como en el caso anterior (cilindro de 20mm) presenta las variaciones de las tensiones en las caras del cilindro respecto a la deformación (estimada aquí como deformación nominal). Los resultados numéricos se ajustan bastante bien respecto a los resultados experimentales, excepto en la cara de abajo, donde se alcanza un error relativo del 40%. En la figura 4.11 se presenta la variación de la densidad respecto a la altura, para los dos cilindros estudiados. En ambos casos, los resultados numéricos se ajustan bastante bien respecto a los resultados experimentales. En esta figura se puede apreciar también el gran efecto que tiene la fricción en las paredes, obsérvese como, para el cilindro de 20mm, la densidad en la cara inferior es 5% menor que en la cara superior, pero para el cilindro de 80mm, este valor es 25%.

Con el objetivo de analizar más detalladamente el efecto de la fricción, durante el proceso de compactación uniaxial, se dibujaron las distribuciones de las tensiones a lo largo de las paredes del cilindro de 20mm de altura. Esta distribución puede verse en la figura 4.12, que se repite en la figura 4.13 para permitir apreciar mejor las diferentes magnitudes de las tensiones. En cada figura hay dos curvas, una correspondiente al caso ideal, en que el coeficiente de fricción es cero, y la segunda cuando el coeficiente de fricción es 0.15. Comparando la distribución de tensiones en las caras superior e inferior, se puede observar como debido a la fricción, la presión requerida por el punzón superior es mayor que cuando no hay fricción, y como ésta, no se transmite completamente a la cara inferior, ya que la presión que se obtiene es menor. La distribución a lo largo de las paredes interior y exterior confirma como la presión disminuye al acercarse al fondo del cilindro. Las esquinas superiores del cilindro sufren un efecto de densificación local debido a la fricción, que se confirma con los grandes valores obtenidos para las tensiones y los grandes valores en la densidad (ver figura 4.8). El efecto en las esquinas inferiores es el opuesto, o sea, de baja densificación. Como se vé en la figura 4.12 el estado tensional en esta zona alcanza su mínimo, al igual que la densidad (ver figura 4.8).

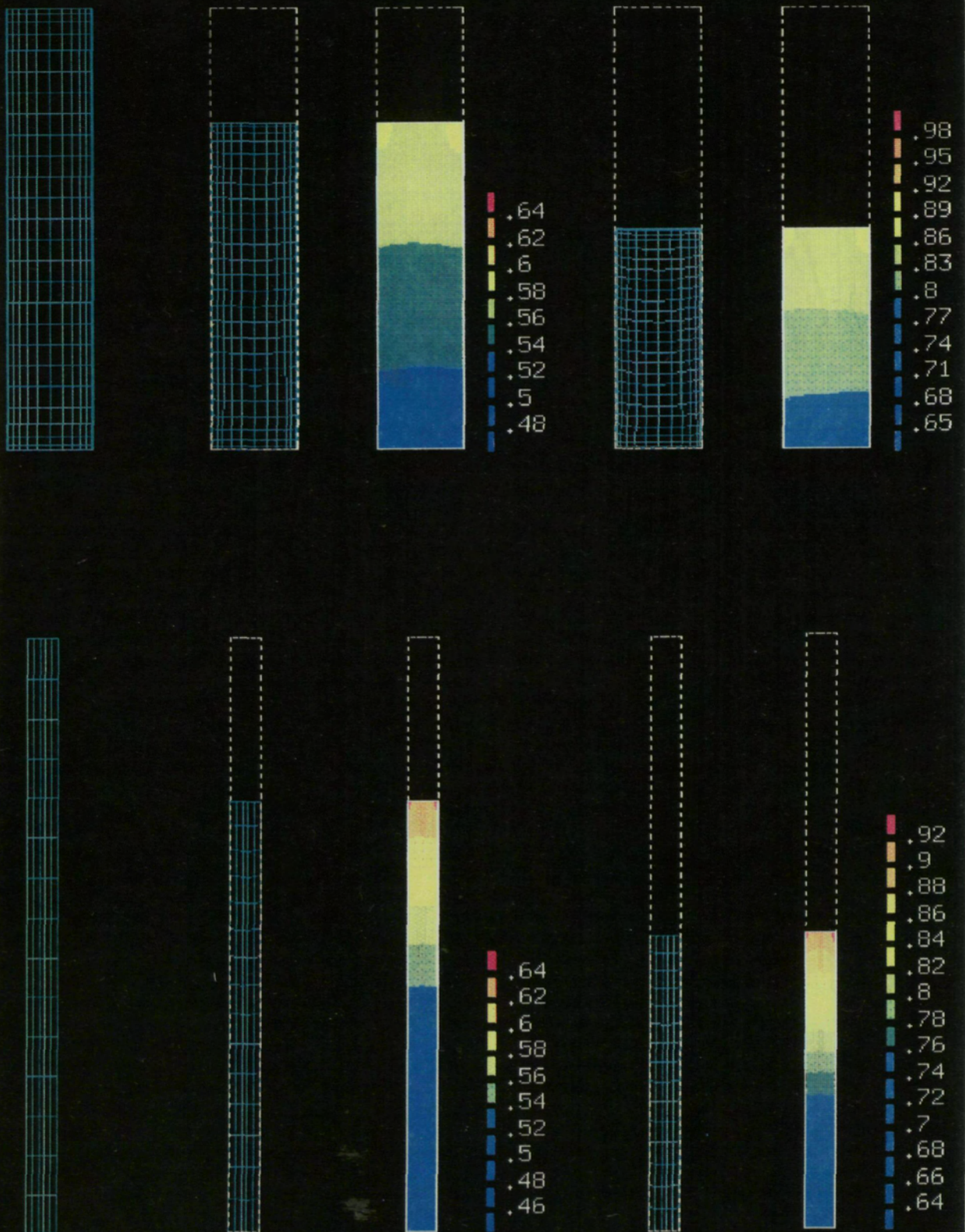


Figura 4.8 Proceso de compactación uniaxial. *Parte superior* (a,b,c,d,e). Compactación de un cilindro de 20mm de altura: deformadas (a,b,d) y distribución de densidades (c,e). *Parte inferior* (f,g,h,i,j). Compactación de un cilindro de 80mm de altura: deformadas (f,g,i) y distribución de densidades (h,j)

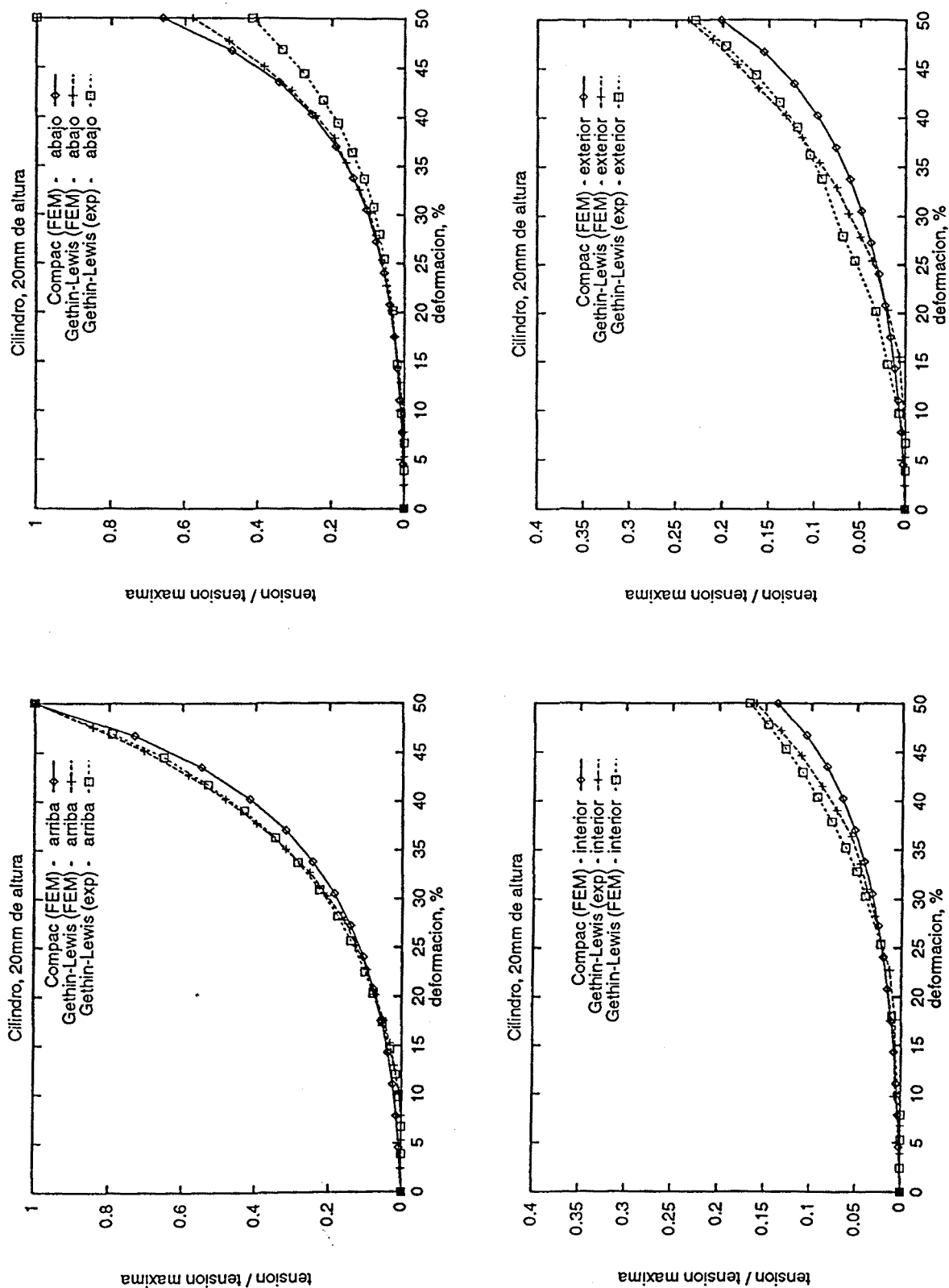


Figura 4.9 Variación de las tensiones en las caras de un cilindro de 20mm de altura. a) cara superior, b) cara inferior, c) cara interior, d) cara exterior

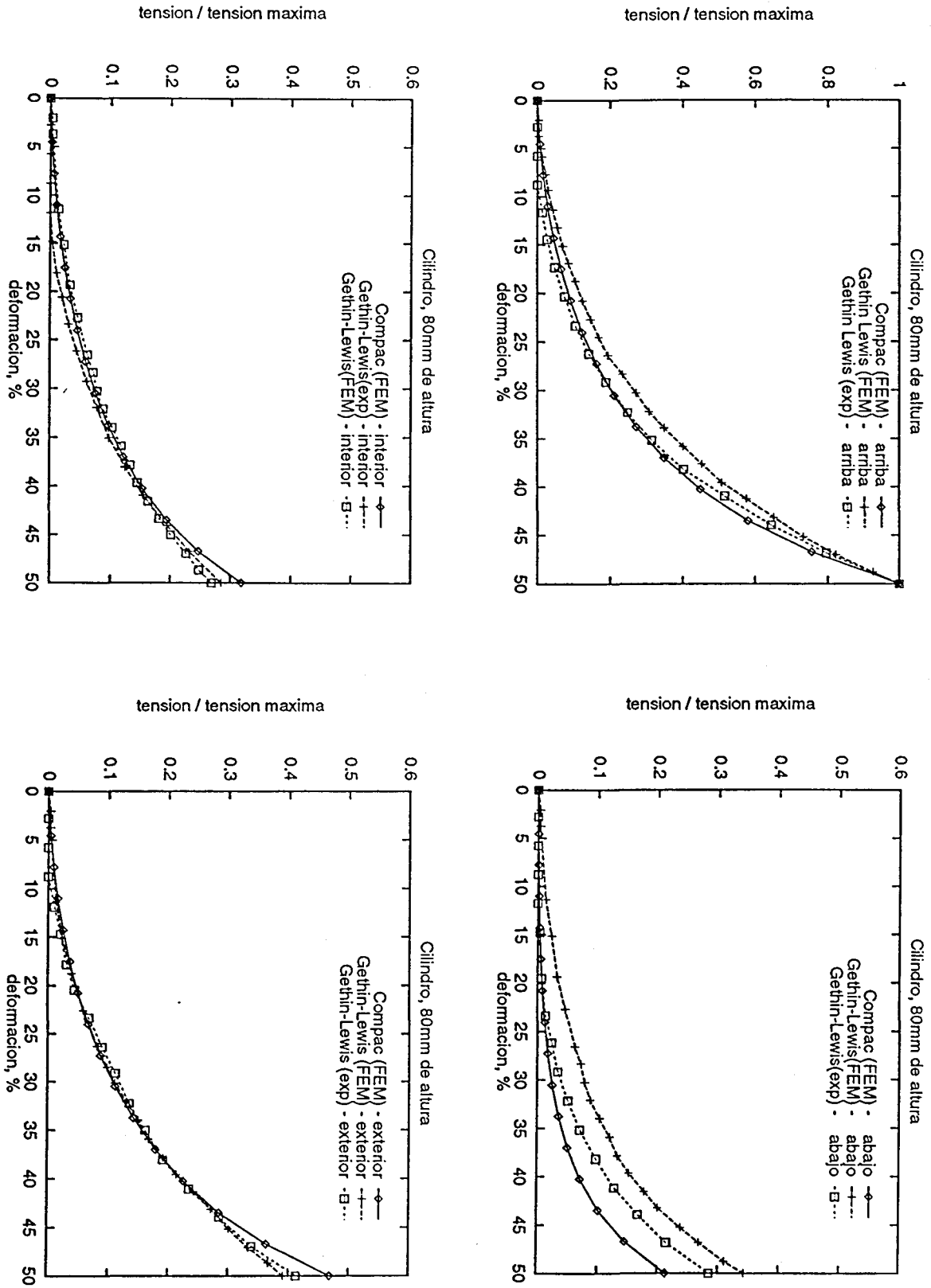


Figura 4.10 Variación de tensiones en las caras de un cilindro de 80mm de altura. a) cara superior, b) cara inferior, c) cara interior, d) cara exterior

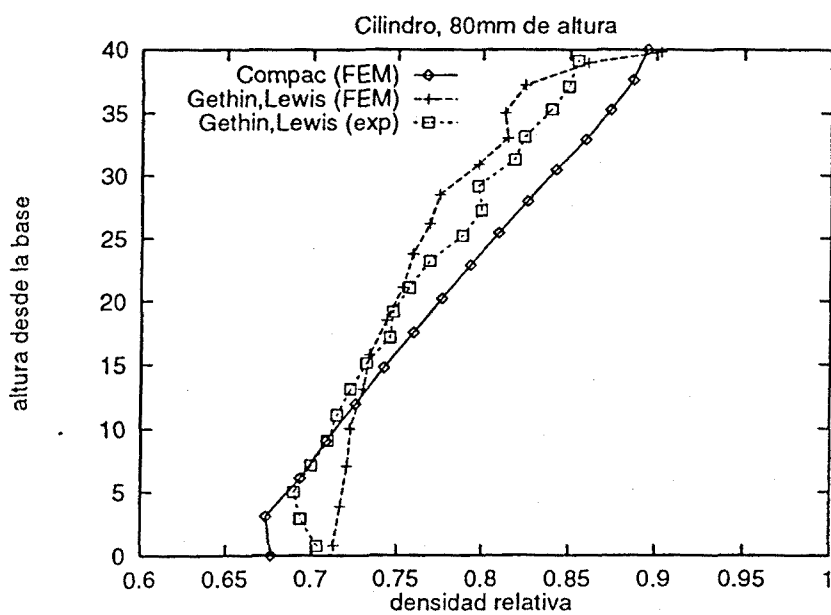
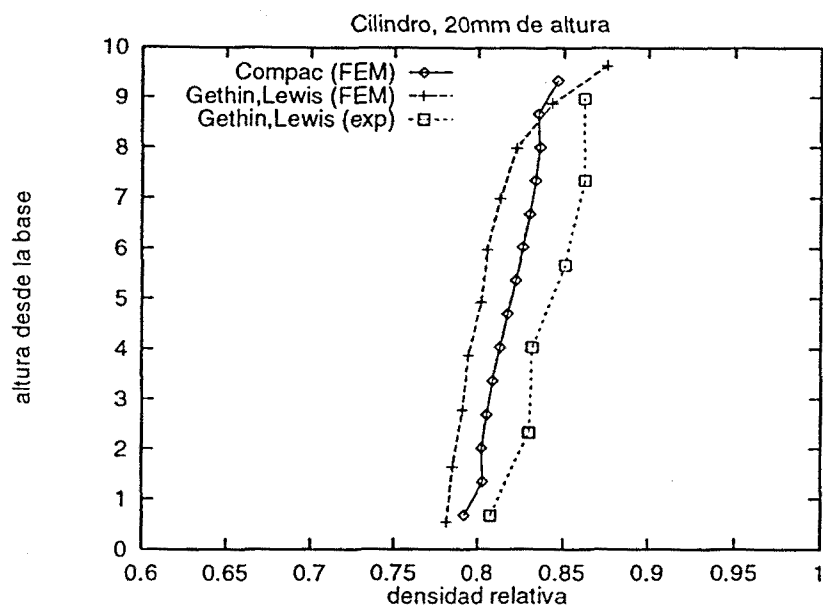


Figura 4.11 Variación de la densidad desde la base a lo largo de un eje medio. a) cilindro de 20mm de altura, b) cilindro de 80mm de altura

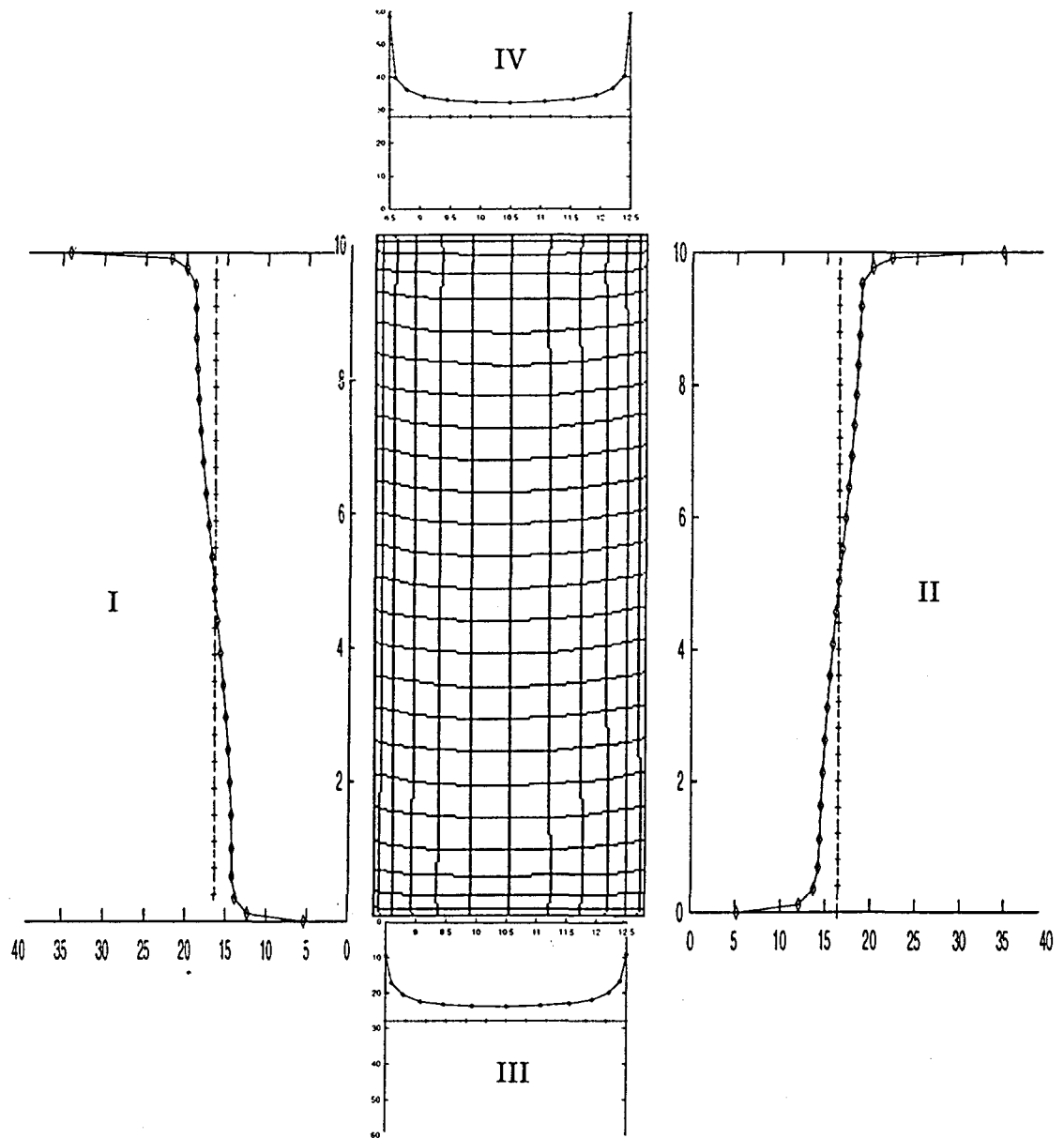


Figura 4.12 Distribución de tensiones en las caras de un cilindro de 20mm de altura. I. interior, II. exterior, III. inferior y IV. superior.

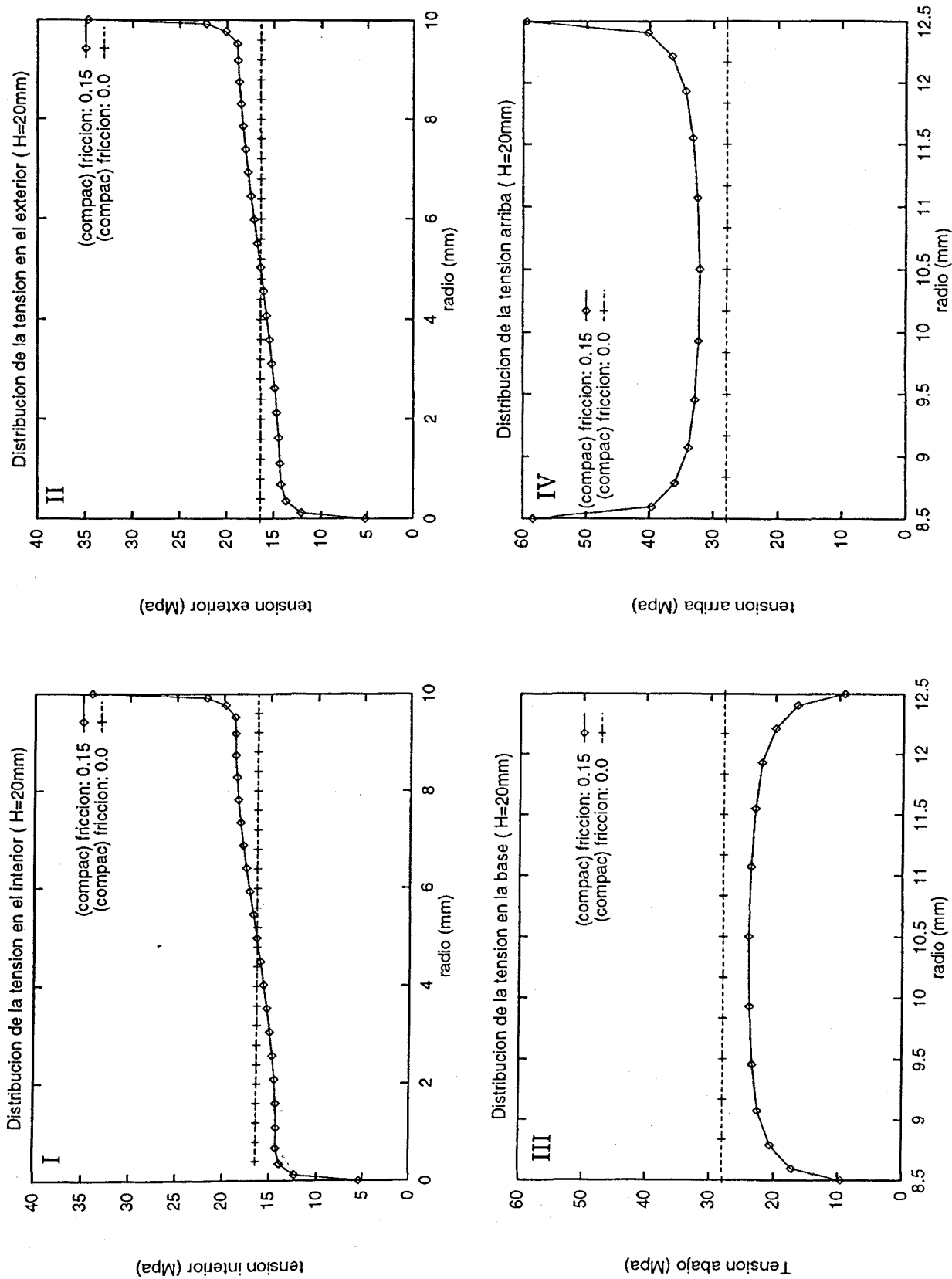


Figura 4.13 Distribución de tensiones en las caras de un cilindro de 20mm de altura. I. interior, II. exterior, III. inferior y IV. superior.

4.3 EJEMPLOS DE SIMULACIÓN

4.3.1 Influencia de la esbeltez: efecto de la fricción

El objetivo de este ejemplo es simular los efectos que tiene la fricción a medida que se modifica la esbeltez de la pieza que se desea fabricar. Se analiza la compactación uniaxial de tres cilindros con diferentes relaciones de esbeltez:

$$\frac{\text{altura inicial}}{\text{diámetro inicial}} = H_o/D_o = 0.7, 2, 6$$

todos los cilindros tienen una altura de 100mm y un hueco interior de 10mm de diámetro. Las mallas empleadas tienen 750, 450 y 300 elementos, respectivamente. Se supusieron los siguientes valores en los parámetros del material:

módulo de Young = 40 Mpa

módulo de Poisson = 0.37

tensión máxima de fluencia = 12 Mpa

y un coeficiente de fricción de 0.15. En todos los casos, se supuso una densidad relativa inicial de 0.4. El proceso de compactación se llevó a cabo desplazando los punzones superior e inferior de forma simultánea hasta que la altura final fue 0.4 veces la altura inicial. En la figura 4.14 se presentan las deformadas y las distribuciones de las densidades relativas obtenidas al final del proceso, para cada uno de los cilindros descritos. La figura 4.15 corresponde al estado de tensiones radial, axial y tangencial para cada uno de los cilindros.

La distribución de densidades muestra como debido a la fricción y a la secuencia de compactación, la densidad es máxima en las esquinas tanto de la cara superior como de la cara inferior y es mínima en los anillos centrales tanto exterior como interior. Se puede ver como el efecto de la fricción es puramente local sobre las paredes (similarmente a lo que ocurre en los fenómenos de capa límite en fluidos), ya que la distribución de las densidades en el interior del cilindro más grueso es bastante homogénea. El efecto de la fricción se acentúa bastante más en el cilindro más esbelto donde la densidad varía en forma de capas horizontales, alcanzando el mínimo en la mitad del cilindro. Estos efectos se pueden confirmar observando la distribución de tensiones (figura 4.15). Dicha distribución es bastante más homogénea en el cilindro menos esbelto. Se puede observar que la tensión tangencial es cero en casi todo el interior, y sólo crece al acercarse a las paredes de los compactos. La distribución de la tensión axial es similar a las densidades, se puede ver claramente una distribución por capas para el cilindro más esbelto, siendo máxima en las zonas superior e inferior del cilindro.

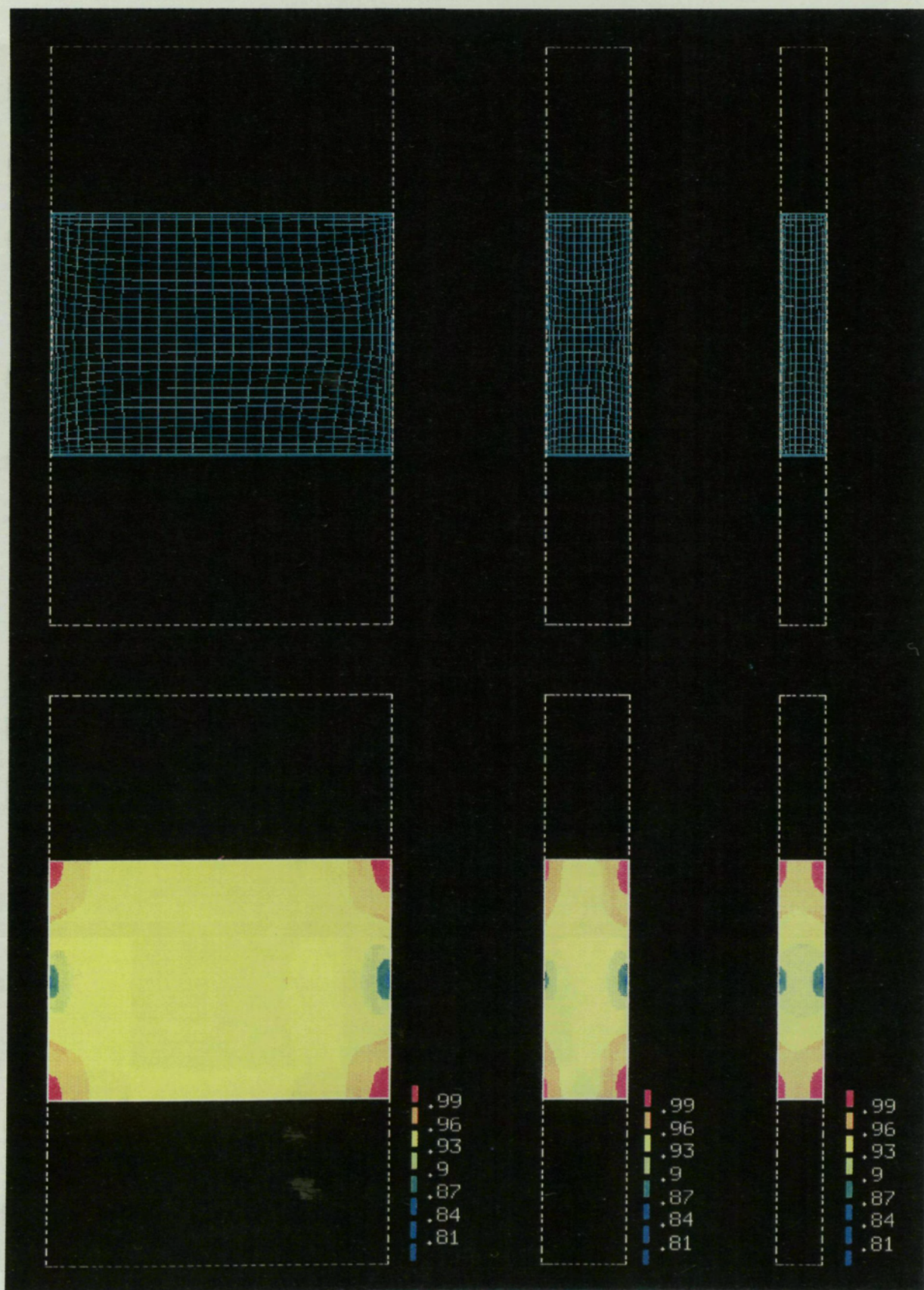


Figura 4.14 Influencia de la esbeltez en la compactación de cilindros.
 $H_o/D_o = 0.7$: a) deformada, d) isóneas de densidad.
 $H_o/D_o = 2.0$: b) deformada, e) isóneas de densidad.
 $H_o/D_o = 6.0$: c) deformada, f) isóneas de densidad.

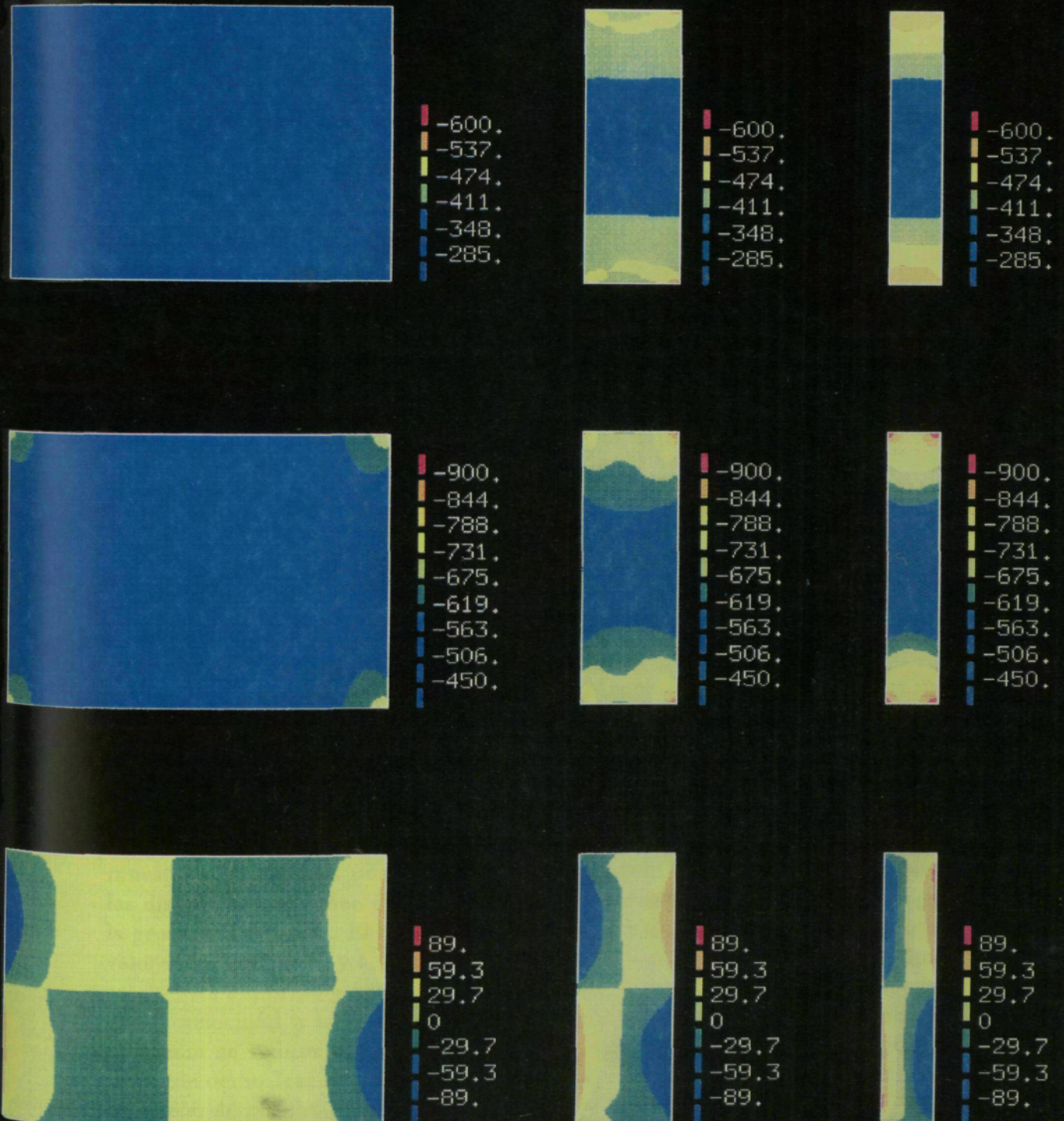


Figura 4.15 Distribución de las tensiones para diferentes relaciones de esbeltez. $H_o/D_o = 0.7$: a) radial, d) axial, g) tangencial. $H_o/D_o = 2.0$: b) radial, e) axial, h) tangencial. $H_o/D_o = 6.0$: c) radial, f) axial, i) tangencial.

4.3.2 Efecto de las diferentes velocidades relativas de los punzones, en el proceso de compactación

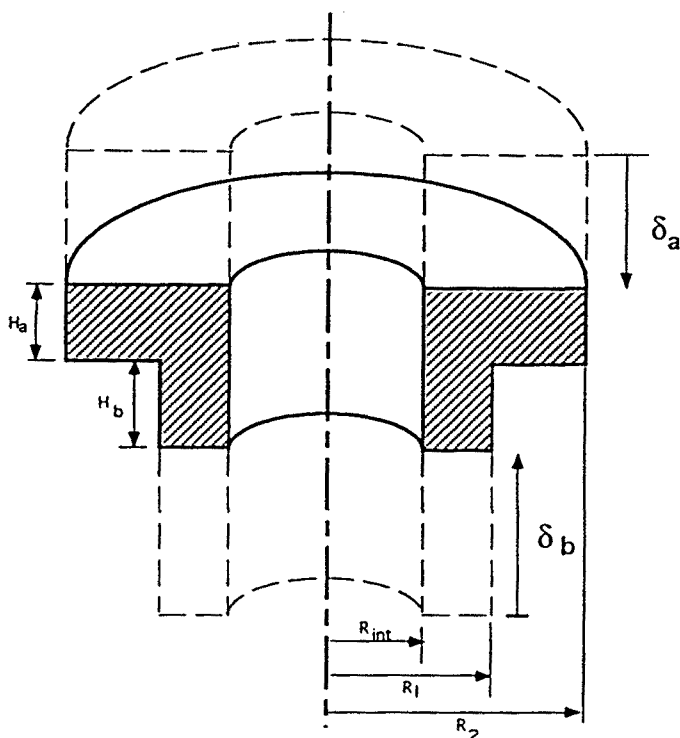


Figura 4.16 Diseño de una pieza a dos niveles. Definición de la geometría y estado inicial del proceso de compactación

Supóngase que se desea diseñar, la pieza axisimétrica de dos niveles, que se muestra en la figura 4.16, que se define mediante los siguientes parámetros: $R_{int} = 5\text{mm}$, $R_1 = 20\text{ mm}$, $R_2 = 35\text{mm}$, $H_a = 32\text{mm}$ y $H_b = 38\text{mm}$. La figura muestra además, una posible configuración inicial, donde δ_a y δ_b representan las distancias que deben desplazarse los punzones superior e inferior para lograr la geometría deseada. El objetivo del proceso de compactación, es encontrar los valores óptimos de δ_a y δ_b , durante todo el proceso, para que la densidad al final, sea lo más uniforme y tan cercana a la teórica, como sea posible. Si se supone que la velocidad a la que se desplazan los punzones es constante, entonces el problema se reduce a encontrar los valores iniciales de δ_a y δ_b . El ejemplo pretende demostrar la utilidad del modelo para ser utilizado en procedimientos de diseño de prueba y error.

A continuación, se llevan a cabo dos de estos posibles *experimentos* o diseños para estimar la efectividad del método. Se suponen los siguientes valores de los

parámetros del material:

$$\text{módulo de Young} = 40 \times 10^4 \text{ Mpa}$$

$$\text{módulo de Poisson} = 0.3$$

$$\text{tensión máxima de fluencia} = 120 \text{ Mpa}$$

se supone además un coeficiente de fricción de 0.15. La figura 4.17 representa las deformaciones y las densidades, que se obtienen a partir de dos estados iniciales diferentes. Las figuras de la parte superior corresponden a suponer que $\delta_a = 19.2\text{mm}$ y $\delta_b = 22.8\text{mm}$. La deformada sólo presenta las distorsiones debidas a la fricción con las paredes, pero no se aprecia ningún flujo de material. Esto se confirma con la distribución de las densidades, que exepctuando las zonas cercanas a las esquinas, es completamente uniforme. La figuras de la parte inferior, se obtienen suponiendo que $\delta_a = 25.2\text{mm}$ y $\delta_b = 10.8\text{mm}$. La densidad inicial es la misma para los dos casos. Se puede verificar fácilmente que el volumen total inicial, es el mismo para los dos ejemplos y como la densidad inicial es la misma entonces se concluye que la cantidad de material en polvo es la misma, sólo que distribuido de forma diferente. La deformada de este segundo diseño, muestra claramente como parte del material que se encuentra en la columna de la derecha, se mueve a la izquierda. La distribución de densidades esta prácticamente dividida en dos, la columna de la derecha, con una densidad casi igual a la teórica y la columna de la izquierda que se encuentra todavía entre el 80% y 90% de la densidad teórica. El movimiento de material se confirma observando la figura 4.18. En ésta se presentan las configuraciones finales, correspondientes a los dos diseños anteriores. Los vectores corresponden a los desplazamientos en la dirección x . Los vectores de color azul, representan desplazamientos hacia la izquierda y los de color rojo, desplazamientos a la derecha. La magnitud de los vectores que corresponden al primer diseño es muy pequeña que confirma el poco flujo en la dirección lateral, pero en el segundo diseño se aprecia claramente el movimiento de material. Las figuras de la parte inferior, representan la tensión de Von Mises al final de la compactación. Nuevamente se obtiene que esta tensión es muy uniforme en el primer diseño pero que varían fuertemente en la zona de la angulosidad, que es la zona crítica en el diseño de esta pieza axisimétrica.

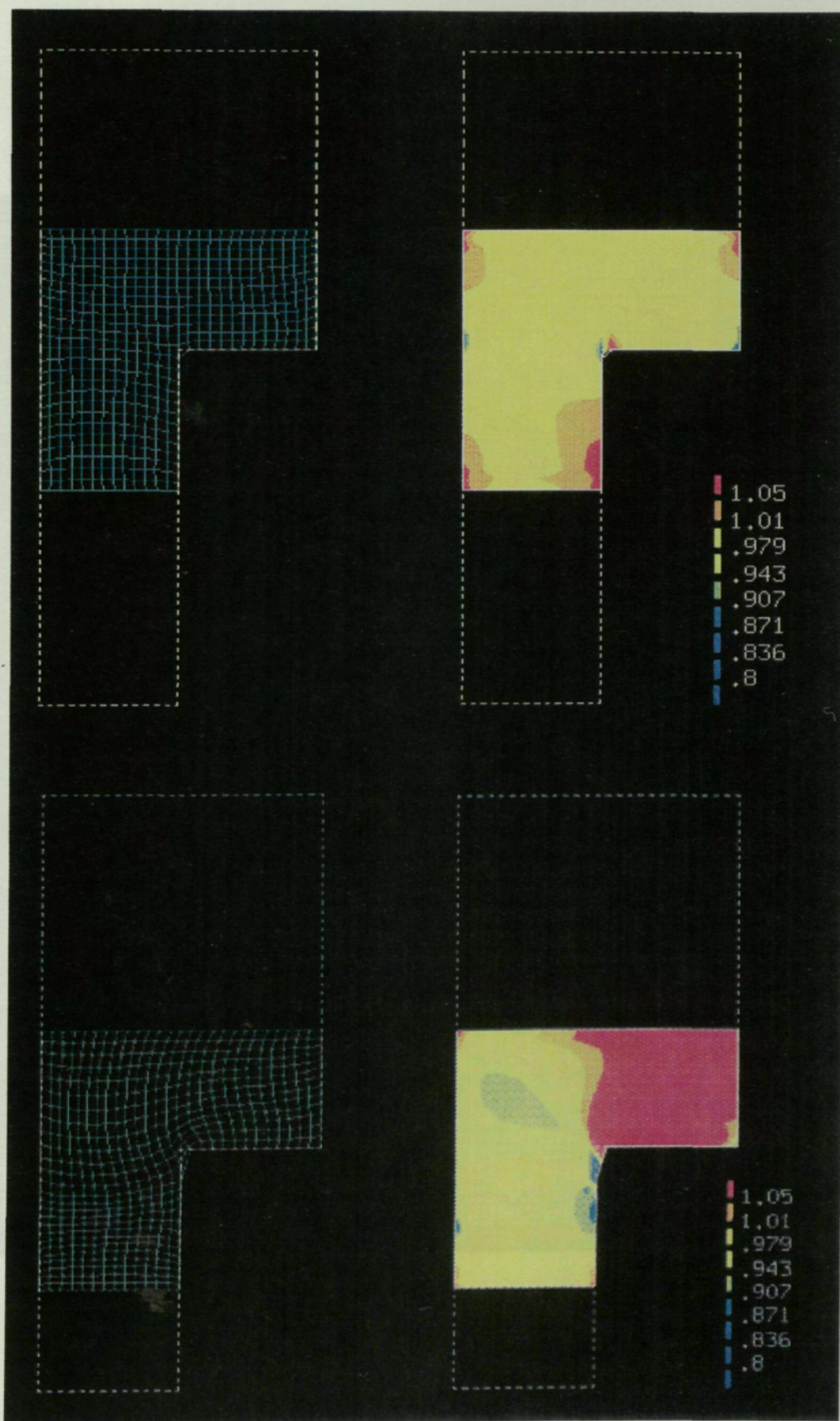


Figura 4.17 Diseño de una pieza a dos niveles. 1) $\delta_a = 19.2\text{mm}$ y $\delta_b = 22.8\text{mm}$: a) deformada, b) Isolíneas de densidad 2) $\delta_a = 25.2\text{mm}$ y $\delta_b = 10.8\text{mm}$: c) deformada, d) Isolíneas de densidad

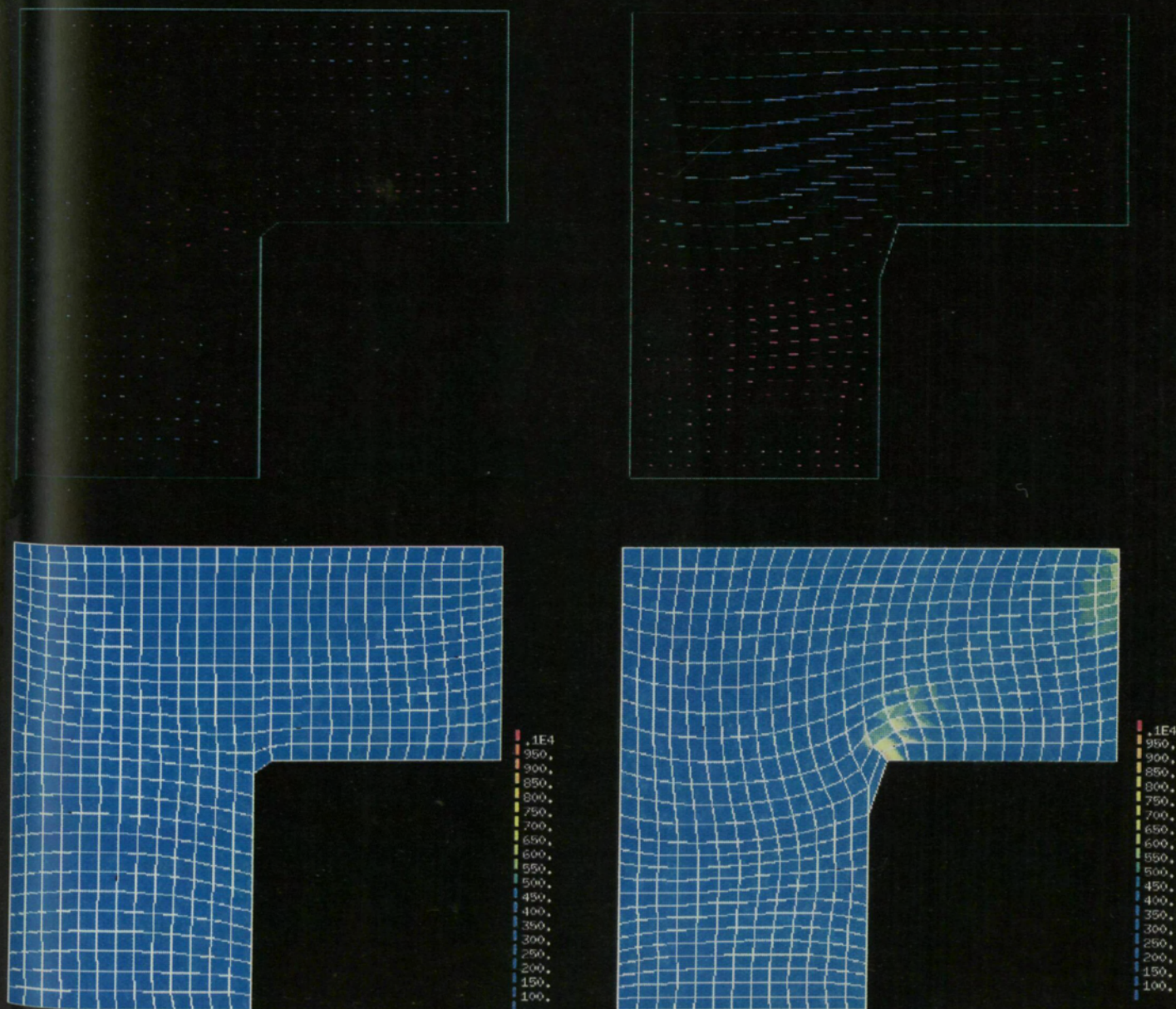


Figura 4.18 Diseño de una pieza a dos niveles. 1) $\delta_a = 19.2\text{mm}$ y $\delta_b = 22.8\text{mm}$: a) Desplazamientos en la dirección x, b) tensión de Von Mises. 2) $\delta_a = 25.2\text{mm}$ y $\delta_b = 10.8\text{mm}$ c) Desplazamientos en la dirección x, d) tensión de Von Mises.

4.3.3 Compactación de una pieza a varios niveles

El objetivo de este ejemplo es presentar un análisis completo del proceso de compactación de una pieza compleja, incluyendo además el proceso de descarga y extracción del molde. La figura 4.19 representa una sección transversal de la pieza que se desea fabricar por el proceso de compactación. La geometría se define mediante los siguientes parámetros (figura 4.19):

$$\begin{aligned} R_1 &= 16.1 \text{ mm} & H_1 &= 40.3 \text{ mm} \\ R_2 &= 22.4 \text{ mm} & H_2 &= 9.4 \text{ mm} \\ R_3 &= 27.7 \text{ mm} & H_3 &= 17.8 \text{ mm} \\ R_4 &= 32.1 \text{ mm} & H_4 &= 60.0 \text{ mm} \\ & & H_5 &= 7.5 \text{ mm} \end{aligned}$$

Se definen además los desplazamientos de los cinco punzones que intervienen en el proceso:

$$\begin{aligned} a &= 8.56 \text{ mm} \\ b &= 12.3 \text{ mm} \\ c &= 12.14 \text{ mm} \\ d &= 8.56 \text{ mm} \\ e &= 18.36 \text{ mm} \end{aligned}$$

Este tipo de montaje de los punzones, se conoce como montaje a tres niveles. Se suponen los siguientes valores de los parámetros del material:

$$\begin{aligned} \text{módulo de Young} &= 1.3 \times 10^4 \text{ Mpa} \\ \text{módulo de Poisson} &= 0.37 \\ \text{tensión máxima de fluencia} &= 100 \text{ Mpa} \end{aligned}$$

Se supone un coeficiente de fricción de 0.1. El proceso se realiza en tres etapas, la primera corresponde a la compactación uniaxial, que se lleva a cabo desplazando los punzones, de forma simultánea, como se muestra en la figura 4.20.b. Los desplazamientos se imponen de forma incremental (30 incrementos), con el siguiente criterio: desplazamientos grandes al principio y pequeños al final. La segunda etapa, de descarga, consiste en liberar al compacto de la presión ejercida por los dos punzones superiores (punzones a y b) y el punzón intermedio inferior (punzón d). La tercera etapa corresponde a la extracción del compacto de la matriz. Esto se consigue desplazando hacia arriba, los punzones laterales inferiores (punzones c y e).

Las figuras 4.21, 4.22 y 4.23 presentan algunos de los resultados obtenidos durante la primera etapa. La figura 4.21 muestra la deformada, al inicio, en dos estados intermedios y al final del proceso. Se puede apreciar como al final del

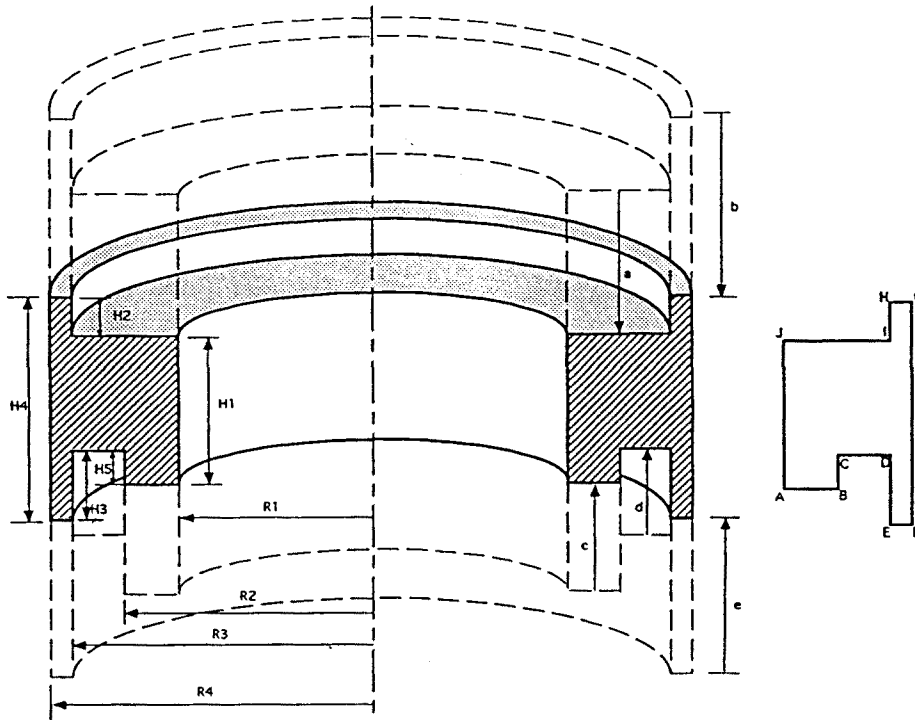


Figura 4.19 Diseño de una pieza a varios niveles: definición de la geometría y del proceso de compactación

proceso, la distorsión que se obtiene (que es un indicativo del flujo de material) es bastante pequeña. Este resultado sugiere que los desplazamientos impuestos a los punzones han sido los adecuados, como se concluirá más adelante.

La figura 4.22 presenta la distribución de las densidades relativas para los estados anteriores. La figura 4.22.d correspondiente a la densidad en el estado final, muestra como en la zona media del compacto la densidad es bastante uniforme, con una densidad relativa de 0.82. Pero nuevamente los grandes gradientes en la densidad aparecen en las esquinas, siendo más acusado el que se localiza en la esquina superior interna (punto I, de la figura 4.19), donde la densidad alcanza el 96% de la densidad teórica. Debido a la esbeltez en los rebordes del compacto, se percibe también el efecto de la densificación por capas. La figura 4.23 presenta el estado tensional que se obtiene al final del proceso. Similarmente a las densidades, el estado de tensiones en la zona media es bastante uniforme. Se obtienen los siguientes niveles en las tensiones: tensión radial: 150 Mpa, tensión axial: 190 Mpa, tensión tangencial: 3 Mpa, tensión circunferencial: 150 Mpa. Obsérvese que el nivel de la tensión tangencial es dos órdenes de magnitud más pequeño que las otras tensiones. La uniformidad generalizada, tanto en las tensiones como en las densidades, permite concluir que el montaje, desplazamientos de los punzones y por tanto el proceso de compactación es el adecuado.

Como se mencionó antes, la etapa de descarga consiste en eliminar la presión ejercida por los punzones superiores y el punzón intermedio inferior. La figura 4.24 presenta la distribución de densidades y la variación de las tensiones en la dirección axial, antes y después de realizar la descarga parcial. Como puede verse no existe ningún cambio apreciable en las densidades, pero si una gran variación en las tensiones, que se confirma con la figura 4.25. Ésta presenta el estado tensional al final de la descarga. Comparando estos resultados con los de la figura 4.23 (antes de la descarga parcial), se puede observar una fuerte disminución en la tensión axial, producto de la eliminación de los punzones superiores. En la zona media del compacto las tensiones radial y circunferencial decrecieron un 80%. La figura 4.20 presenta la expansión (*spring back*) que ocurre, que es del orden del 0.32% de la altura.

La figura 4.26 muestra el compacto durante el proceso de extracción. Esta etapa se lleva a cabo empujando hacia arriba el compacto, por medio de los punzones laterales inferiores (punzones c y e, de la figura 4.19). Se ha supuesto que el compacto deja de tener contacto con la matriz después de que éste pase por la horizontal definida por el punzón superior (punzón b) en su posición inicial. La figura 4.20, muestra la variación de la presión de extracción frente al desplazamiento del compacto. La curva se puede dividir en tres partes: la primera, entre los punto O-A, donde se presenta un cambio casi instantáneo en la presión, alcanzando un valor máximo que después se prolonga hasta el punto B. En esta segunda parte (comprendida entre los puntos A y B), la presión se mantiene prácticamente constante. La porción entre los puntos B y C define la tercera parte, en donde la presión decrece hasta cero a medida que el compacto va perdiendo contacto con la matriz. La figura 4.27, presenta la expansión final y la distribución de las tensiones residuales, que se obtienen al finalizar el proceso de extracción. La figura 4.27.d muestra (ampliando 25 veces) el contorno del compacto al final de la extracción y el contorno del compacto al final de la primera etapa (compactación uniaxial). Se obtiene una expansión axial del 0.32% de la altura y una expansión radial del 0.05% del diámetro. A partir de las tensiones 4.27.b, 4.27.c y 4.27.d, se concluye que las tensiones residuales en la zona media del compacto son prácticamente cero, pero no ocurre lo mismo en las zonas cercanas a las angulosidades, donde se localizan tensiones residuales del orden del 15 al 20% del valor medio de las tensiones al final de la primera etapa.

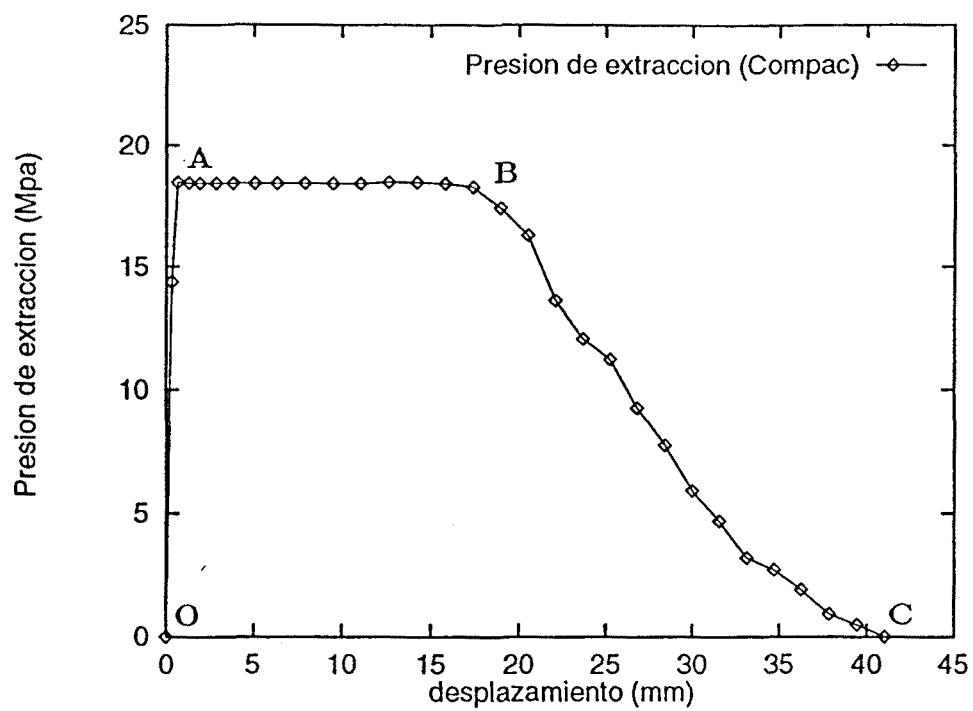


Figura 4.20.a Variación de la presión de extracción frente al desplazamiento del compacto

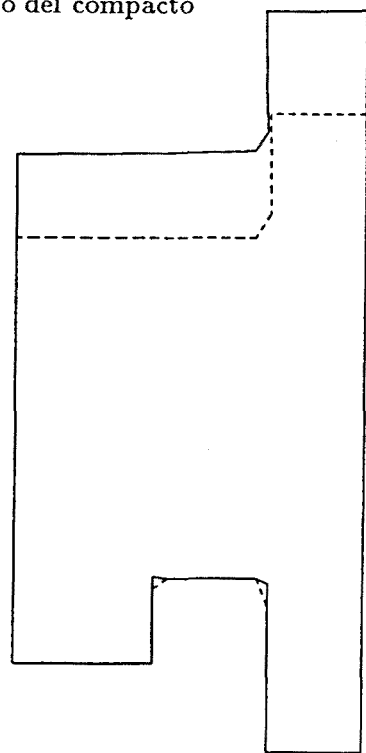


Figura 4.20.b Expansión debida a la descarga parcial. Eliminación de los punzones a, b y d.

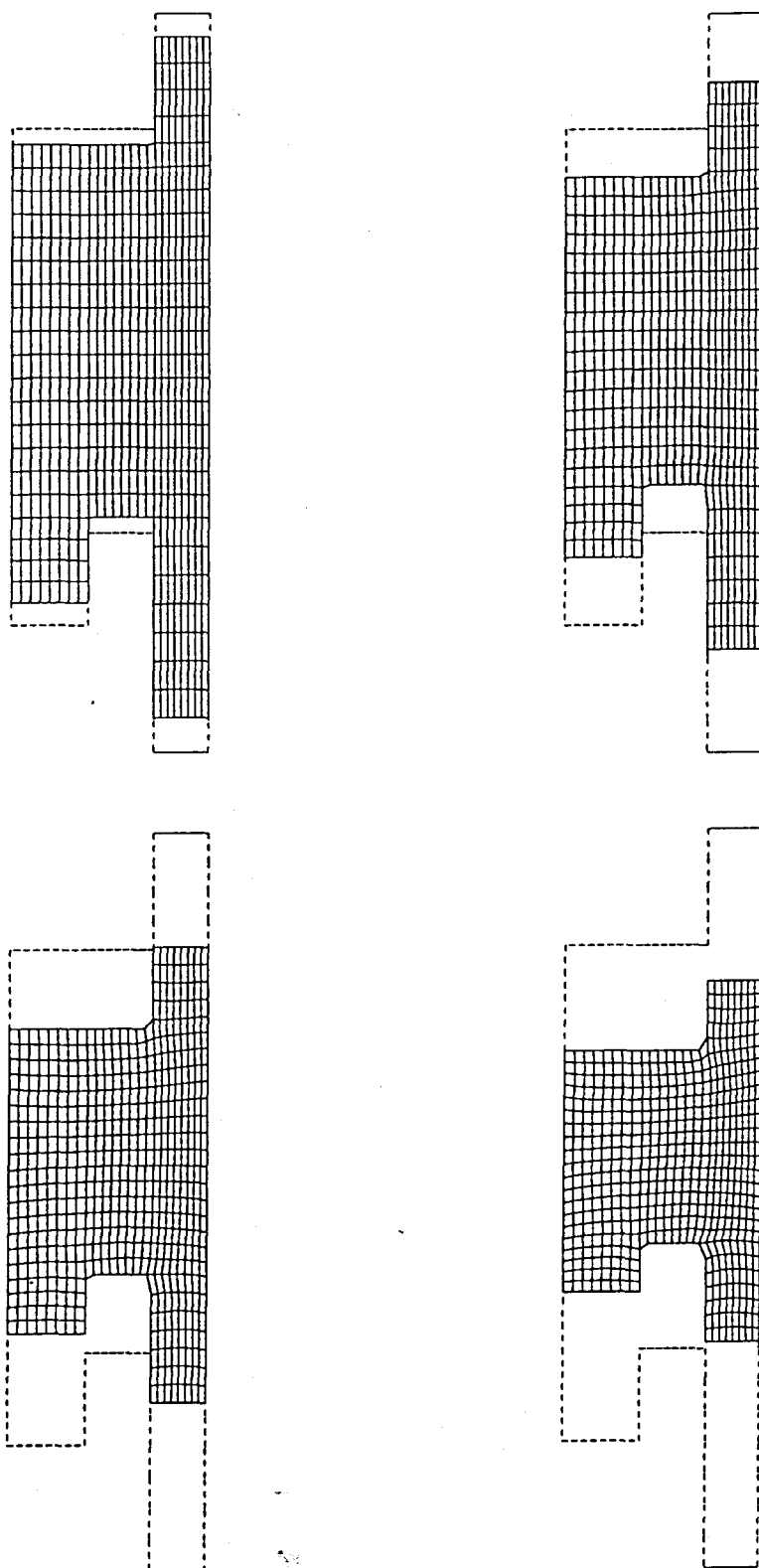


Figura 4.21 Diseño de una pieza a varios niveles. Variación de la deformada durante la compactación uniaxial

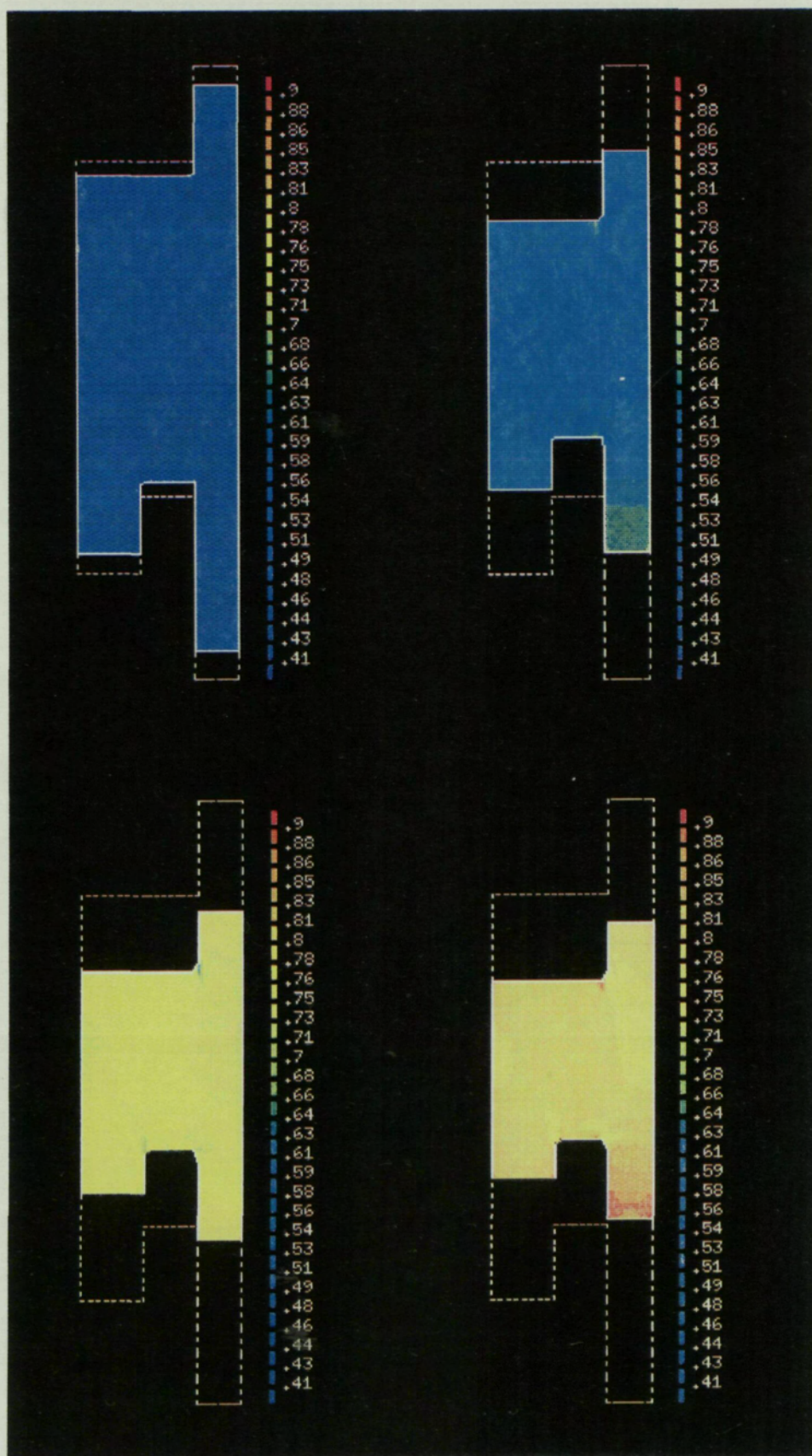


Figura 4.22 Diseño de una pieza a varios niveles. Distribución de la densidad relativa durante la compactación uniaxial

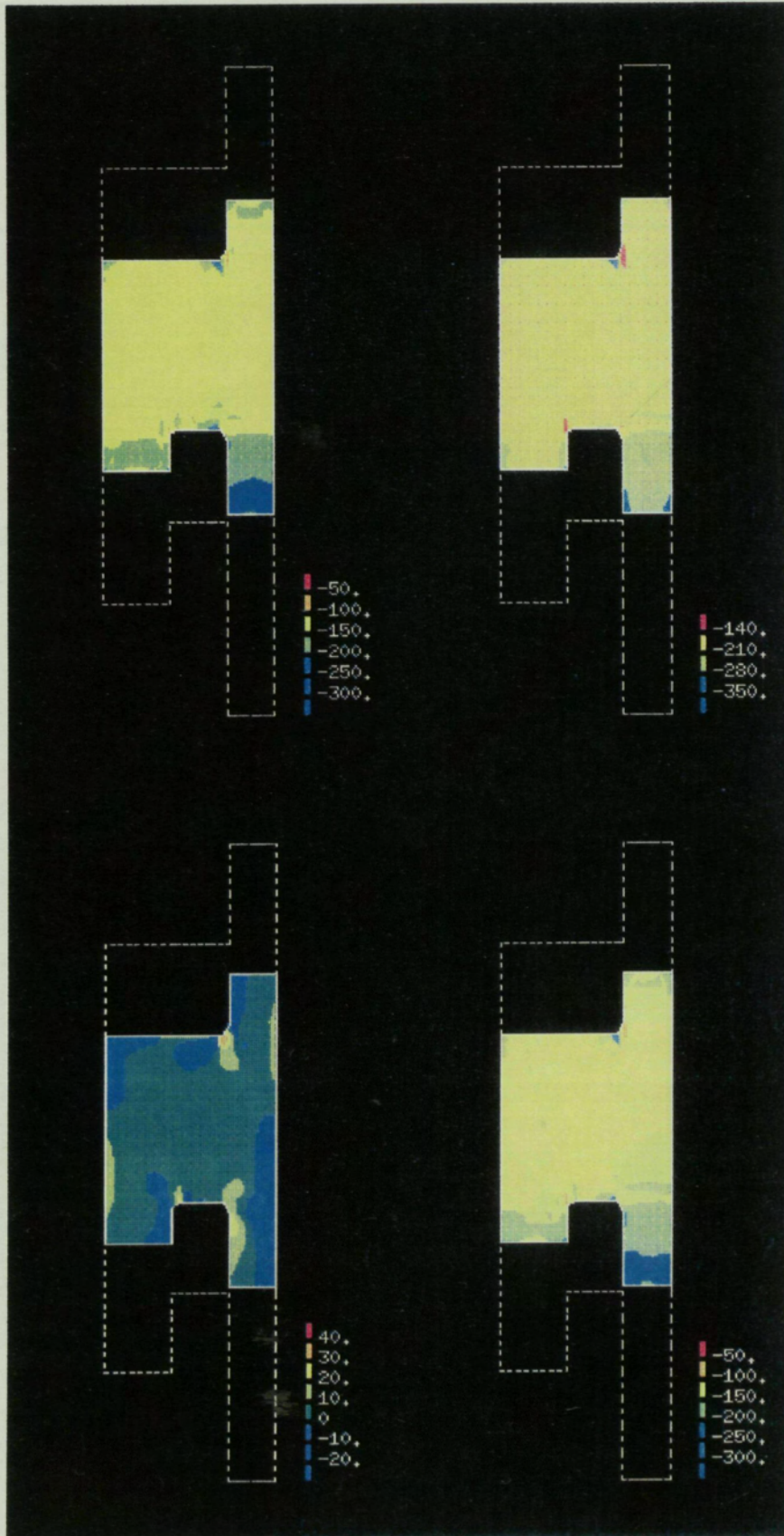


Figura 4.23 Diseño de una pieza a varios niveles. Estado tensional al final del proceso de compactación uniaxial. a) radial, b) axial, c) tangencial, d) circunferencial

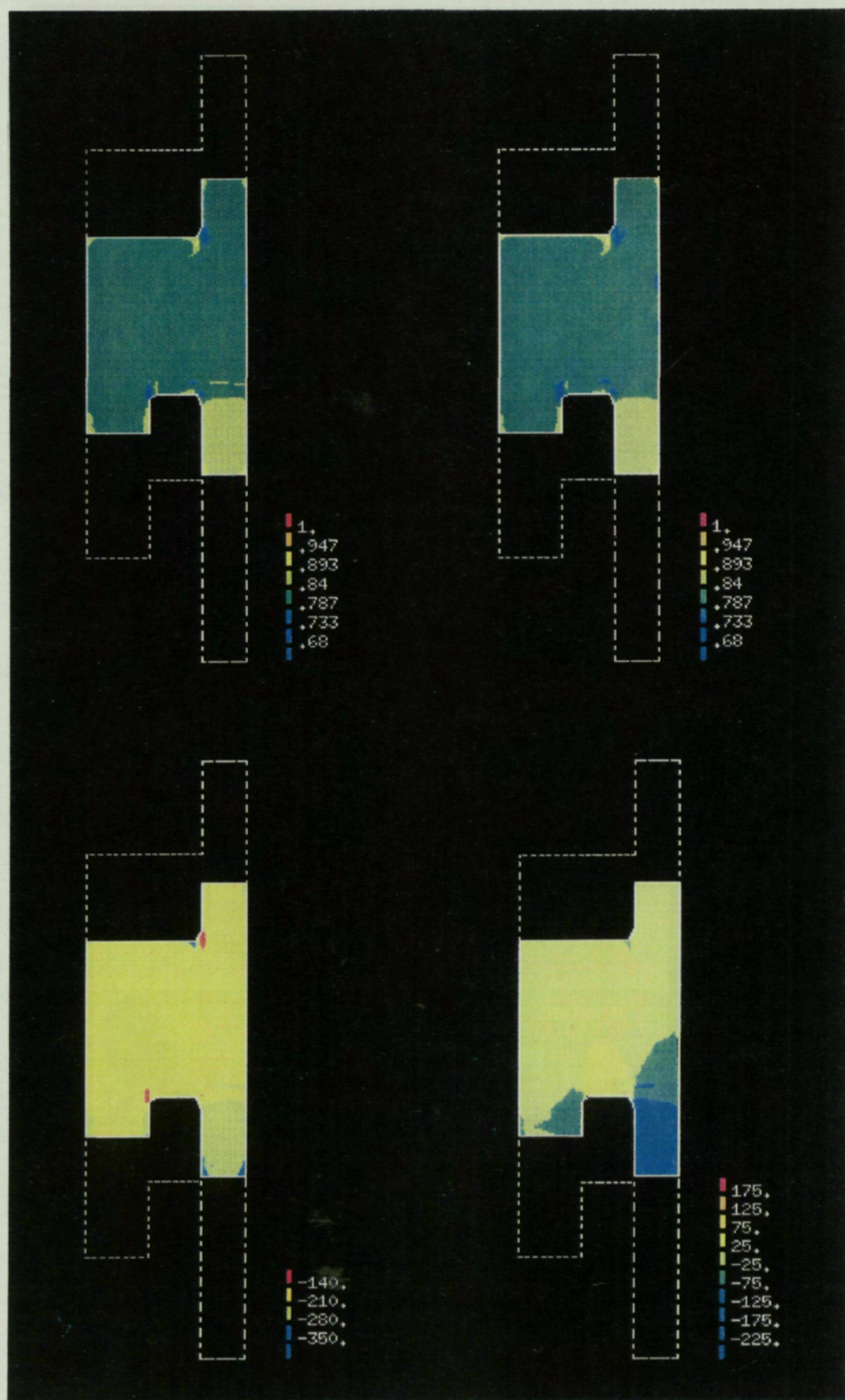


Figura 4.24 Diseño de una pieza a varios niveles. a) Densidad antes de la descarga parcial. b) Densidad después de la descarga parcial. c) Tensión axial antes de la descarga parcial. d) Tensión axial después de la descarga parcial.

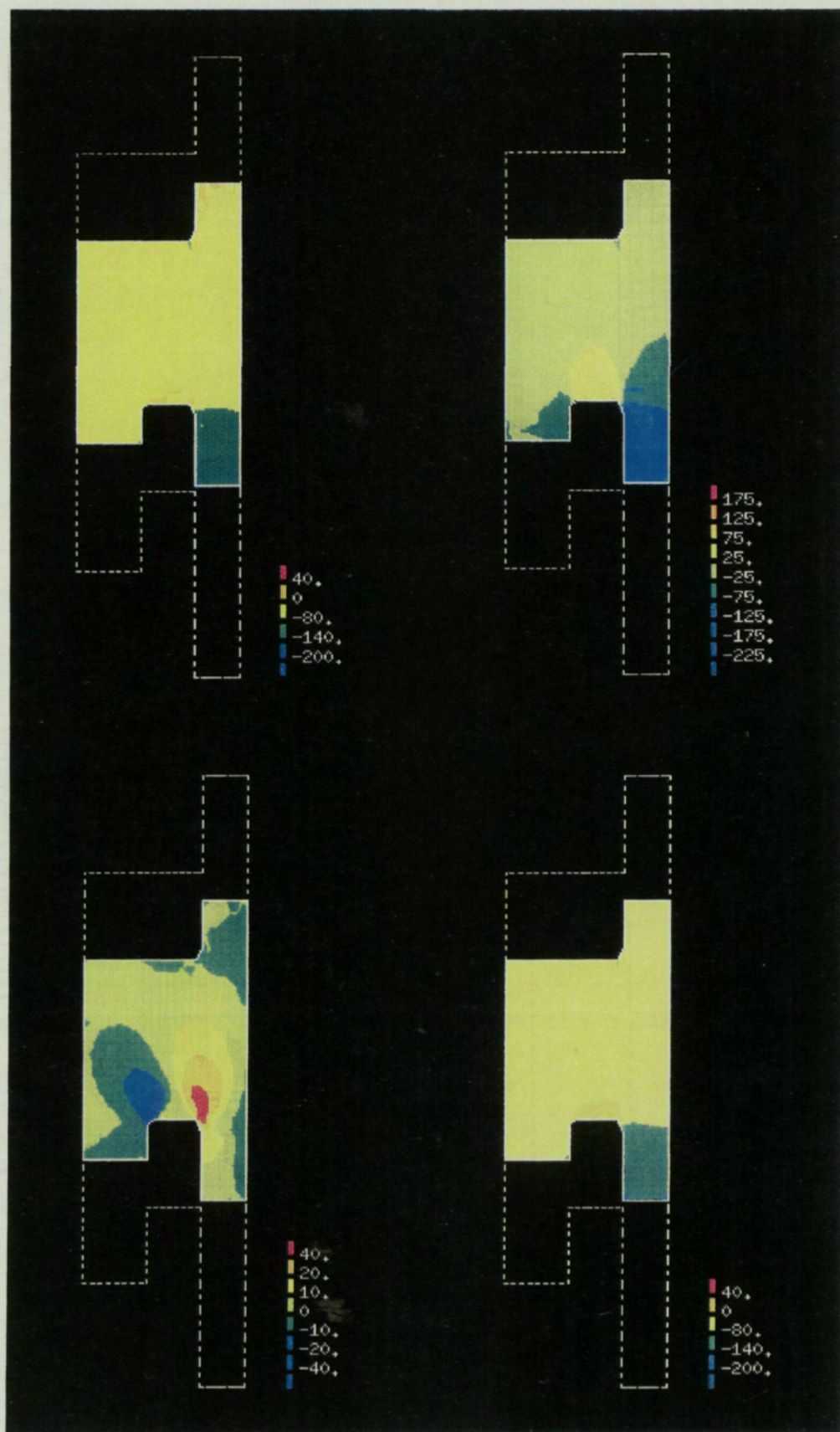


Figura 4.25 Diseño de una pieza a varios niveles. Estado tensional al final de la descarga parcial. a) Radial, b) axial, c) tangencial, d) circunferencial

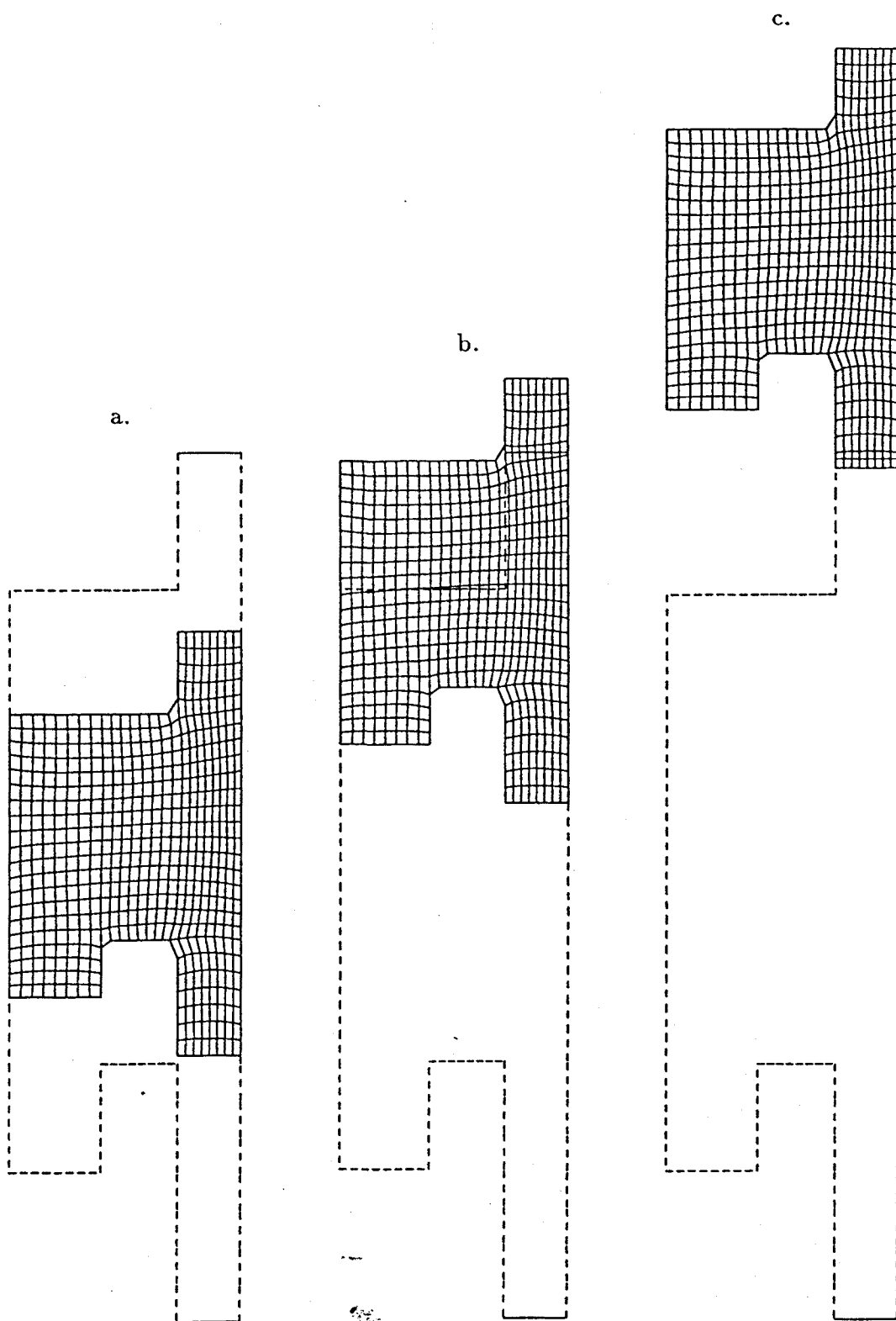


Figura 4.26 Diseño de una pieza a varios niveles. Diferentes estados del compacto, durante la etapa de extracción

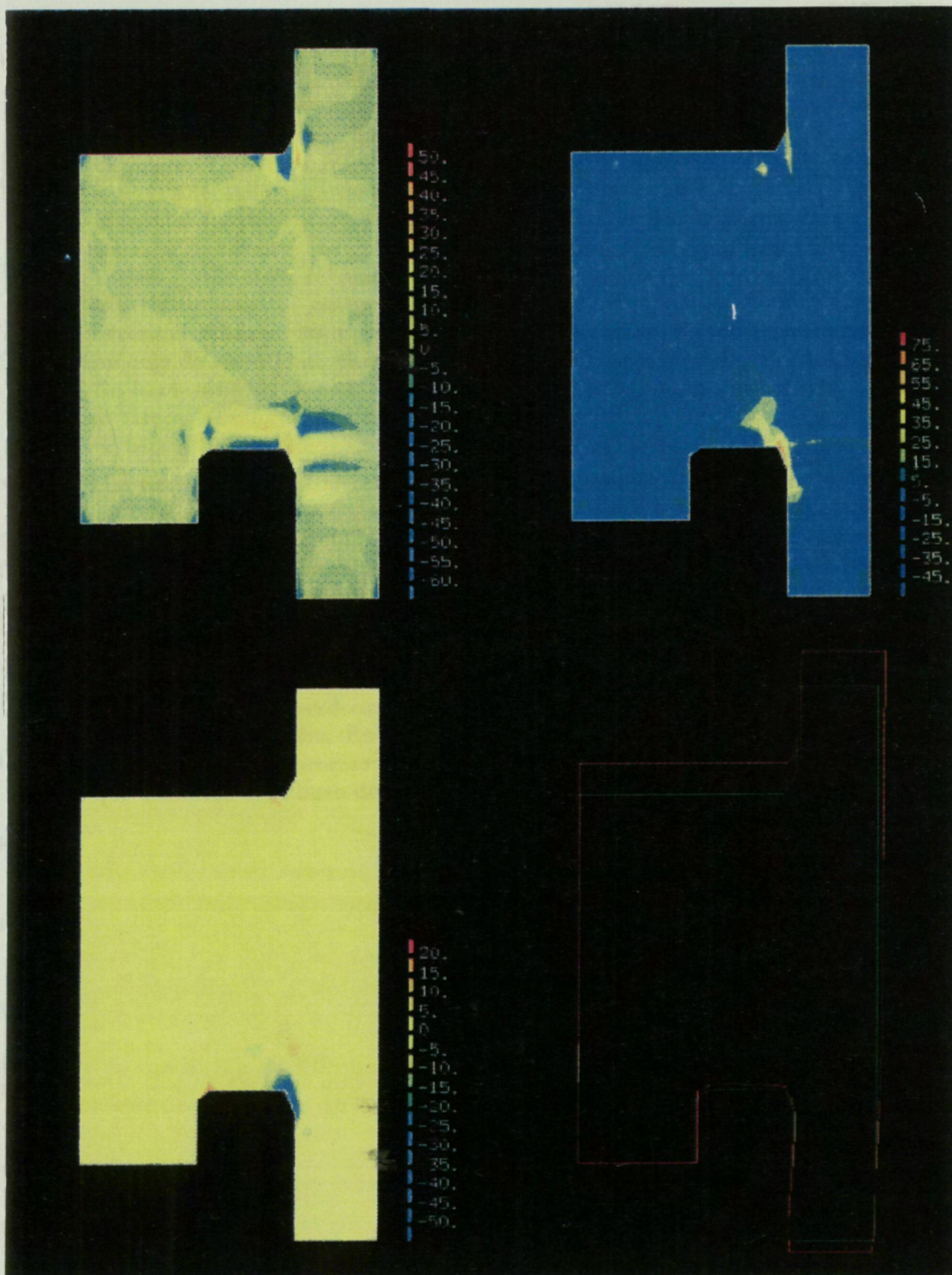


Figura 4.27 Diseño de una pieza a varios niveles. Tensiones residuales:
 a) radial, b) axial, c) tangencial. d) Expansión lateral y axial, al final de la extracción

4.3.4 CONSIDERACIONES FINALES

El análisis de los resultados obtenidos con los ejemplos de validación, permite concluir que el modelo numérico propuesto aquí, es capaz de obtener resultados que muestran buena concordancia con los resultados experimentales disponibles. Dichos resultados indican que una regla de flujo asociada es lo más adecuado para la función de fluencia que se utiliza en este trabajo. El modelo requiere de la calibración de dos parámetros que dependen de la densidad relativa y que gobiernan el crecimiento de la superficie de fluencia. Dichos parámetros se pueden determinar mediante dos grupos de ensayos experimentales: 1) compresión isostática y 2) compactación uniaxial. Se necesitan además, los valores de tres parámetros del material en verde: el módulo de elasticidad, el módulo de Poisson y la tensión máxima de fluencia. En la mayoría de los ejemplos aquí realizados, estos valores fueron estimados, al carecer de información exacta acerca de los mismos.

En el segundo conjunto de ejemplos, se pone de manifiesto, el potencial que tiene dicha herramienta numérica, para la asistencia durante el diseño del proceso de compactación incluyendo la etapa de extracción del compacto. Se puede a partir de ella obtener información acerca de la evolución de la densidad y de las tensiones durante el proceso de compactación. Se puede analizar la modificación de dichos resultados cuando se utiliza una secuencia diferente del movimiento de los punzones, o cuando se considera que el efecto de la fricción es diferente. Se puede también obtener información de lo que sucede durante el proceso de extracción y conocer de esta forma el estado final del compacto, en particular se puede predecir su expansión volumétrica y el nivel y localización de las tensiones residuales. En definitiva, el modelo es capaz de capturar los efectos más relevantes del proceso de compactación y ayudar a definir los parámetros que conducen a un buen diseño del mismo.

CAPÍTULO 5

UTILIZACIÓN DE TÉCNICAS DE CÁLCULO PARALELO EN EL ANÁLISIS POR ELEMENTOS FINITOS

5.1 MOTIVACIÓN PARA EL USO DE ORDENADORES PARALELOS

Los ingenieros, están en constante búsqueda de técnicas que mejoren el diseño de sus productos. Esto motiva el desarrollo de modelos más complejos que permitan representar la física del fenómeno de una manera más precisa y con base en estos resultados mejorar los diseños. La resolución de estos nuevos modelos, requiere en general, ordenadores cada vez más rápidos y eficaces. El diseño de presas, puentes, barcos, aviones supersónicos, edificios y muchos procesos de manufactura, que se analizan por el método de los elementos finitos precisan la resolución de un gran sistema de ecuaciones algebraicas o de ecuaciones en derivadas parciales. Los programas convencionales, desarrollados hasta ahora sobre ordenadores escalares, requieren grandes tiempos de respuesta y grandes capacidades de memoria. La simulación realística de un análisis de dinámica estructural, según reporta Farhat en [81], requiere del orden de 10 horas de CPU en el Cray-2, el análisis aeroelástico de un perfil aerodinámico usando flujo potencial requiere alrededor de 5 horas de CPU, pero para establecer la frontera de vibración transónica, (para un conjunto de parámetros fijos), se requiere la solución de 30 análisis del perfil elástico, lo que implica 6 días de CPU. Si se desean solucionar las ecuaciones de Navier Stokes para este problema, se estima que el incremento del tiempo de CPU es de dos ordenes de magnitud. En el estudio de cálculo de seguridad a los choques, impacto y penetración [85], no es sorprendente requerir 100 horas de CPU en un ordenador escalar. Otra familia de problemas que requieren una gran capacidad de cálculo son los que involucran análisis de fallo. Este fenómeno involucra la localización de severos gradientes en el campo de los desplazamientos. La malla de elementos finitos debe ser extremadamente fina para modelar este tipo de fenómeno. Por ejemplo [89], las bandas de corte son típicamente del orden de 10 a 300 micrones de ancho, el perfil de la deformación en la banda de corte es similar a una onda senoidal y su solución requiere al menos de 5 a 10 elementos a través del ancho de la banda, entonces el tamaño del elemento es aproximadamente un micrón.

Una muestra de $10 \times 10 \text{ cm}^2$ requiere $10^5 \times 10^5$ elementos. Se puede disminuir sustancialmente este número utilizando un método de malla adaptativa en la zona de la banda, pero se necesitan del orden de 10^7 elementos en el caso 2D para obtener una solución adecuada del problema [89]. Se puede anticipar que un análisis no lineal requerirá entre 10^6 y 10^8 elementos y del orden de 10^7 segundos, esto es, aproximadamente 100 días realizando una integración explícita en un ordenador con una velocidad media de 100 mflops. Debido a que la eficiencia de los ordenadores escalares (sólo un procesador) se aproxima a su límite, el uso de ordenadores de procesamiento paralelo (definidos en 5.3) supondría un cambio fundamental en la velocidad de procesamiento y en la capacidad de almacenamiento convirtiéndolo en una de las herramientas de diseño y alta productividad más importantes en el futuro.

Pero la necesidad de una gran velocidad de cálculo y almacenamiento, no se limita sólo al diseño en ingeniería [80]. En muchas áreas científicas, técnicas, de recursos energéticos, médicas, militares, de inteligencia artificial y de investigación básica se requieren también ordenadores muy potentes. Por ejemplo, la modelación predictiva [80] desarrolla modelos multidimensionales de la atmósfera, el medio ambiente terrestre, el espacio exterior, la economía mundial. En esta área se realizan extensos experimentos de simulación por ordenador para predecir los efectos a corto y largo plazo de fenómenos tan complejos como las inundaciones, las sequías, las poluciones ambientales, los cambios en el clima terrestre, la administración de la industria pesquera, la exploración de recursos, la dinámica costera y mareas. Estos ordenadores pueden jugar un papel muy importante en el hallazgo de petróleo y gas y en la administración de su extracción. También en la viabilidad de desarrollo de otros tipos de energía y en la garantía de seguridad en los reactores nucleares. En el área médica se requieren ordenadores rápidos en tomografía asistida (que consiste en la visualización por planos de los órganos internos del cuerpo humano), diseño de órganos artificiales, diagnóstico, estimación de daños cerebrales y estudios de ingeniería genética.

En este capítulo se estudian algunas de las arquitecturas de ordenadores paralelos disponibles en la actualidad. Se analizan y proponen algunas implementaciones sobre estas máquinas. El ejemplo que se usa para llevar a cabo el análisis, es el programa POWCOM desarrollado en esta tesis. En la sección 5.2 se describe paso a paso el algoritmo general del problema, y se analizan las operaciones más importantes: cálculo de fuerzas residuales, cálculo de la matriz tangente y resolución de los sistemas lineales asociados a la linealización por el método de Newton. En la sección 5.3 se lleva a cabo una breve descripción de las diferentes arquitecturas paralelas y su clasificación. El estudio se concentra tanto en ordenadores del tipo SIMD (Simple flujo de Instrucciones Múltiples flujos de Datos) como en ordenadores del tipo MIMD (Multiple flujo de Instrucciones Múltiples flujos de Datos). En la sección 5.4 se analizan los algoritmos, las estructuras de datos, y su implementación sobre las diferentes

arquitecturas. En el caso de los ordenadores del tipo SIMD se analizan los métodos iterativos (exactos) como los métodos de resolución más adecuados para resolver el problema que se trata aquí. Se analiza una estructura de datos mixta, que es una combinación entre las estructuras de datos elemental y nodal. En este caso la matriz de rigidez se calcula a nivel elemental y el resto de las variables vectoriales que intervienen en la resolución, se calculan a nivel nodal. Con los ordenadores del tipo MIMD se utiliza el método del complemento de Schur, asignando a cada procesador un grupo de subdominios que se procesan en paralelo. Se emplea el método GMRES para resolver el problema de la interfaz que se formula mediante el complemento de Schur. Esta técnica de paralelización se define aquí como paralelización de grano grueso. Adicionalmente se estudia una paralelización de grano fino que consiste básicamente en forzar la paralelización mediante la adición de directivas de compilación. En la sección 5.5 se presentan los resultados obtenidos de las diferentes implementaciones. Se analizan y comparan los tiempos de ejecución, para diferentes tamaños del problema global, se analizan también los problemas de comunicación.

5.2 DESCRIPCIÓN DEL ALGORITMO GENERAL

La figura 5.1 resume el algoritmo general que se emplea aquí para la simulación por el método de los elementos finitos, del problema de compactación descrito en el capítulo 3. A continuación se describen más precisamente cada una de sus partes y se destacan las principales operaciones que deben llevarse a cabo para cumplir la tarea asignada a cada una. El diagrama de flujo representa el esquema típico que se utiliza en la solución de un problema no lineal. En este caso, el problema no lineal, se puede escribir como sigue:

Encontrar U desplazamientos nodales tales que:

$$\boxed{\mathbf{F}^{res}(U) = \mathbf{F}^{int}(U) - \mathbf{F}^{ext}(U) = 0} \quad (5.2.1)$$

El vector $\mathbf{F}^{int}(U)$ representa las fuerzas internas y $\mathbf{F}^{ext}(U)$ las fuerzas externas, ambas definidas en A.3.5. La linealización por el método de Newton del problema anterior, se puede escribir como:

Dado U_n ;

$$U_{n+1}^0 = U_n, \quad k = 0$$

Mientras Que ($\|\mathbf{F}^{res}(U_{n+1}^k)\|$) > TOL

$$\Delta U_{n+1}^{k+1} = \left[\frac{\partial \mathbf{F}^{res}}{\partial U_{n+1}}(U_{n+1}^k) \right]^{-1} \cdot (\mathbf{F}^{ext}(U_{n+1}^k) - \mathbf{F}^{int}(U_{n+1}^k)) \quad (5.2.2)$$

$$U_{n+1}^{k+1} = U_{n+1}^k + \Delta U_{n+1}^{k+1}$$

$$k = k + 1$$

FIN(MQ)

El término $\frac{\partial F^{res}}{\partial U}$ corresponde a la matriz tangente del problema global, calculada en el anexo A2. El diagrama de flujo de la figura 5.1, contiene dos lazos iterativos, el de la derecha que controla el número y tamaño de los incrementos de carga (o desplazamiento impuesto) y el lazo de la izquierda que corresponde a la solución por el método de Newton descrita en (5.2.2).

5.2.1 Cálculo de las fuerzas internas

El cálculo del vector de fuerzas internas F^{int} se lleva a cabo realizando la integración numérica de la ecuación (A.3.5.a) primero a nivel elemental y luego se procede a su ensamblaje:

$$F^{int}(U) = \sum_{e=1}^{N_e} F_e^{int}(U) \quad (5.2.3)$$

donde Σ representa el operador de ensamblaje, N_e es el número de elementos y $F_e^{int}(U)$ es el vector de fuerzas internas del elemento e (ver A.3.5.a). La integración numérica requiere la evaluación del integrando en ciertos puntos conocidos como puntos de Gauss [70].

5.2.2 Cálculo de las fuerzas externas

La dificultad en el cálculo del vector F^{ext} definido en (A.3.5) se centra en el cálculo de los términos correspondientes a las fuerzas de fricción y contacto. Dichas fuerzas, deben ser evaluadas en coordenadas locales y luego ensambladas a lo largo de toda la frontera que permanece en contacto con el contorno rígido. En el caso de la fuerza de contacto, el cálculo se reduce a encontrar la distancia mínima entre un punto de la frontera del cuerpo y el contorno para luego multiplicar esta distancia por un factor de penalización (sección 3.4). Pero el cálculo de la fuerza de fricción requiere un trabajo más complejo. Los algoritmos 3.5, 3.6 y 3.7 pueden utilizarse dependiendo del método elegido. Todos tienen en común la distinción entre dos estados: no deslizamiento (*stick*) y deslizamiento (*slip*). Una parte de la frontera en contacto puede estar deslizando y la otra no, esto conduce a que las operaciones requeridas para calcular la fuerza de fricción en cada nodo puedan ser distintas, afectando así la homogeneidad de las operaciones requeridas. La utilización del método del Lagrangiano Aumentado incrementa la complejidad de los cálculos, ya que como puede verse en el cuadro 3.7, la actualización de los multiplicadores de Lagrange, incluidos tanto para el contacto como para la fricción, hace necesario la inclusión de un nuevo lazo, exterior al lazo del problema no lineal que aumenta el costo del problema global (este lazo no ha sido incluido en la figura 5.1).

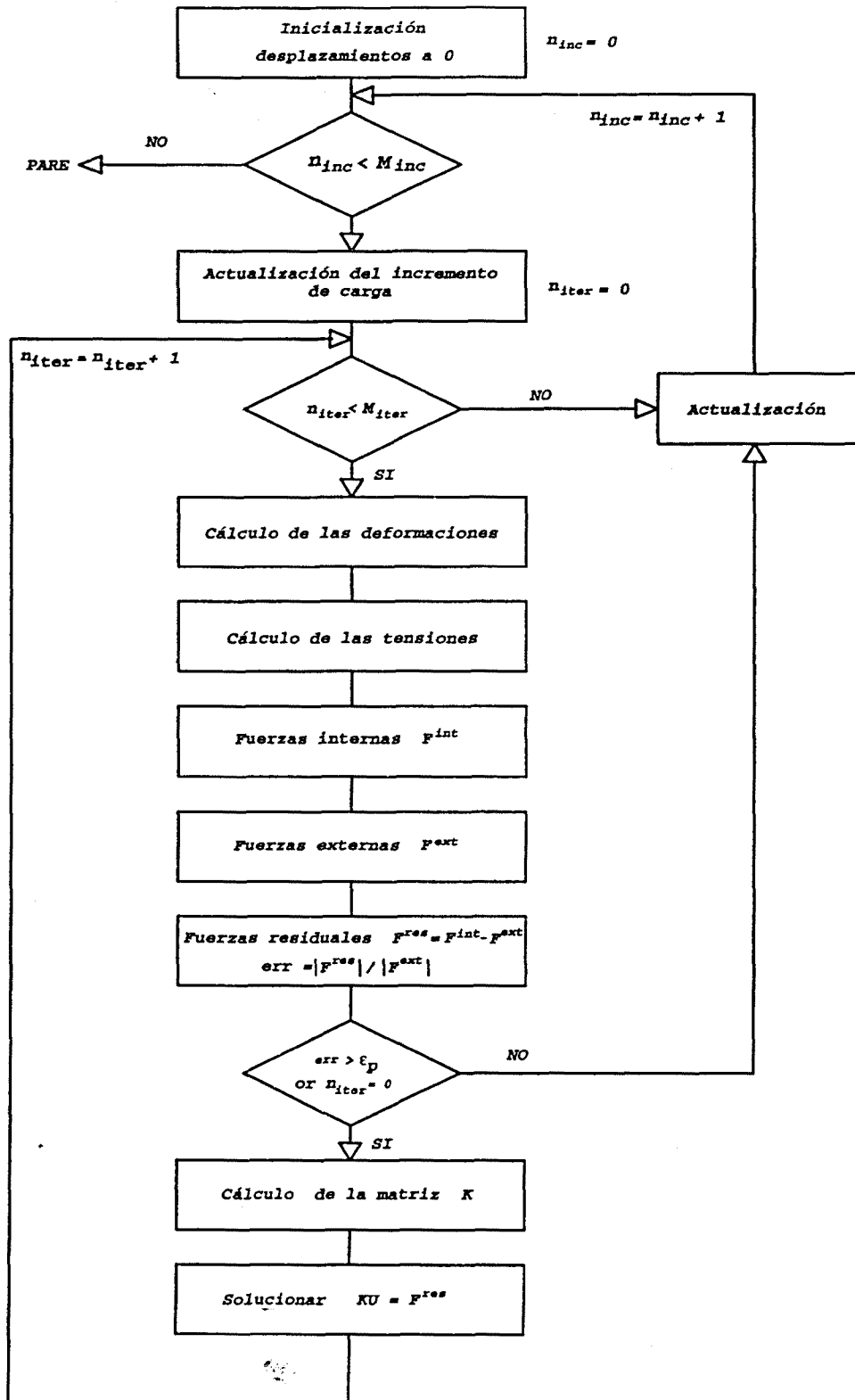


Figura 5.1 Algoritmo del programa no lineal (serial)

5.2.3 Cálculo de la matriz tangente

La matriz tangente del problema global se puede escribir como:

$$\mathbf{K}_T = \frac{\partial \mathbf{F}^{res}}{\partial \mathbf{U}} = \frac{\partial \mathbf{F}^{int}}{\partial \mathbf{U}} - \frac{\partial \mathbf{F}^{ext}}{\partial \mathbf{U}} \quad (5.2.4)$$

La primera componente corresponde a la contribución debida al término de fuerzas internas, dicha expresión se encuentra calculada en el lema A3.1 y es la suma de dos matrices, la primera conocida como componente material y la segunda se conoce como matriz geométrica. El cálculo se lleva a cabo de forma elemental y luego se realiza el ensamblaje de estas matrices elementales:

$$\frac{\partial \mathbf{F}^{int}}{\partial \mathbf{U}} = \mathbf{K}^{int} = \sum_{e=1}^{Ne} \mathbf{K}_e^{int} \quad (5.2.5)$$

La matriz \mathbf{K}_e^{int} representa la matriz elemental que puede descomponerse en:

$$\mathbf{K}_e^{int} = \mathbf{H}_e^{int} + \mathbf{G}_e^{int} \quad (5.2.6)$$

\mathbf{H}_e^{int} corresponde a la componente material y \mathbf{G}_e^{int} a la matriz geométrica. (ver A.3.6). El cálculo de \mathbf{H}_e^{int} requiere esencialmente el cálculo del tensor constitutivo tangente \mathbf{C}^{ep} en la configuración material. Al final de los cuadros 3.2.a y 3.2.b se encuentran las expresiones para dicho tensor, en la configuración espacial. Se debe por tanto utilizar el operador *pull back* (anexo A4) para llevar este tensor a la configuración material.

La contribución a la matriz tangente debida a las fuerzas exteriores, se centra en la contribución debida a las fuerzas de contacto y fricción. Dichas expresiones pueden encontrarse en el anexo A2.

5.2.4 Solución del sistema lineal

La última etapa consiste en resolver el siguiente sistema lineal (en cada iteración):

$$\mathbf{K}_T(\mathbf{U}_{n+1}^k) \cdot \Delta \mathbf{U}_{n+1}^k = -\mathbf{F}^{res}(\mathbf{U}_{n+1}^k) \quad (5.2.7)$$

correspondiente a la linealización del problema (5.2.1). Existen varios métodos para solucionar este sistema (métodos directos y métodos iterativos). La implementación de uno u otro, como se verá más adelante, depende fuertemente del tipo de arquitectura del ordenador sobre el cual se desea realizar dicha implementación.

5.3 DESCRIPCIÓN DE LAS DIFERENTES ARQUITECTURAS PARALELAS

5.3.1 Clasificación de los ordenadores paralelos

Los ordenadores paralelos son aquellos sistemas que favorecen el procesamiento en paralelo [80]. Estos se pueden dividir en tres configuraciones arquitectónicas básicas: (1) ordenadores de segmentación encauzada, (2) ordenadores matriciales y (3) sistemas multiprocesadores. El ordenador de segmentación encauzada, efectúa operaciones solapadas para explotar el paralelismo temporal (Cray-1, Cyber-205). Un procesador matricial emplea múltiples unidades aritmético-lógicas sincronizadas para lograr paralelismo espacial (Illiac-IV, Connection Machine) y un sistema multiprocesador alcanza paralelismo asíncrono gracias a un conjunto de procesadores interactivos que disponen de recursos compartidos, como memorias, bases de datos, etc (IBM 3090, Convex C3480). La utilización eficiente de cualquiera de estas configuraciones arquitectónicas, depende grandemente del algoritmo que se propone para solucionar un determinado problema. El caso que es de interés aquí, es la utilización de estos ordenadores para solucionar un problema de cálculo por el método de los elementos finitos. Como se comentará más adelante, los algoritmos que se utilizan varían grandemente cuando se tiene en cuenta dicha configuración.

Existen otros esquemas de clasificación, que permiten percibir el impacto que tienen sus diferentes arquitecturas sobre el tipo de algoritmos viables para solucionar un problema de Elementos Finitos en dichos ordenadores. Respecto de esta clasificación Farhat [81], destaca las siguientes tres: *granularidad*, *topología* y *control*.

Granularidad: clasifica los ordenadores por el número de sus procesadores y el tamaño de estos. Los puntos extremos se encuentran en ordenadores con pocos procesadores a ordenadores con un gran número de procesadores simples y pequeños. Los primeros llamados ordenadores de grano fino y los segundos de grano grueso. Entre los ordenadores de grano grueso figuran Alliant FX/8 (8 procesadores), IBM 3090-VF (6 procesadores), Cray X-MP(4 procesadores) y Convex C3480 (8 procesadores) entre otros. Como ordenadores del segundo grupo figuran: La Connection Machine(64k procesadores), NCUBE-1024 (1k procesadores). Está claro que la granularidad afecta fuertemente el tipo de algoritmo que puede usarse para solucionar un problema de Elementos Finitos. Es por esto que generalmente se asocian ordenadores de grano grueso con el método de subdominios y ordenadores de grano fino con paralelización a nivel de elementos o de grados de libertad.

Topología: clasifica los ordenadores dependiendo de como sus procesadores están interconectados. Dependiendo de la topología del ordenador, el flujo de datos de un problema específico se realiza en forma óptima o elevará fuerte-

mente el costo de comunicación. La forma como la memoria está organizada afecta la interconexión entre los procesadores. La Connection Machine (CM) es un ordenador de memoria local, donde un procesador sólo puede acceder a su propia memoria y requiere de un sistema de comunicaciones para intercambiar información entre procesadores, cuya topología es un hipercubo (esto se describirá en la sección 5.3.2). Los procesadores de Alliant FX/8 trabajan sobre un sistema de memoria compartida y se conectan a través de un bus de memoria común y pueden acceder al sistema de memoria central. La topología también puede influir fuertemente en la selección del algoritmo para resolver el problema de Elementos Finitos. Este método se caracteriza por su gran localidad, pero cada elemento necesita intercambiar información con sus elementos vecinos. En un ordenador como la CM, implica que elementos vecinos deberían estar en procesadores vecinos para disminuir costo de comunicación. En un ordenador como el Convex C3 o Alliant FX/8 la generación de los subdominios debe también minimizar la comunicación. Otro factor a tener en cuenta es la memoria de cada procesador. Factor que se puede convertir en una limitante del número de elementos asignados a un procesador o una limitante del tamaño del subdominio o subdominios asignados a un procesador.

Control: Esta clasificación fue originalmente propuesta por Michael J. Flynn (1966) y clasifica los ordenadores en cuatro categorías de acuerdo con la multiplicidad de los flujos de instrucciones y datos. El término *flujo* se emplea aquí para denotar una secuencia de elementos (instrucciones o datos) que ejecuta o sobre los que opera un único procesador.

- SISD Simple flujo de instrucciones-simple flujo de datos
- SIMD Simple flujo de instrucciones-múltiples flujos de datos
- MISD Múltiples flujos de instrucciones-simple flujo de datos
- MIMD Múltiples flujos de instrucciones-múltiples flujos de datos

SISD representa a todos los ordenadores en serie que se utilizan actualmente. Las instrucciones se ejecutan secuencialmente. Aunque posea varias unidades funcionales, todas están bajo la supervisión de sólo una unidad de control.

SIMD Esta categoría representa todos los ordenadores con múltiples elementos de proceso (EP) supervisados por la misma unidad de control. Todos los EP reciben la misma instrucción emitida por la unidad de control pero operan sobre diferentes conjuntos de datos procedentes de flujos distintos.

MISD En ésta, existen n unidades procesadoras, cada una recibe distintas instrucciones que operan sobre el mismo flujo de datos y sus derivados. Los resultados de un procesador pasan a ser la entrada del siguiente procesador. Esta estructura ha sido tachada de poco práctica y no se ha construido un ordenador correspondiente a este grupo.

MIMD A este grupo pertenecen la mayoría de los sistemas multiprocesadores y sistemas con múltiples ordenadores. Un ordenador MIMD implica interacciones entre los n procesadores porque todos los flujos de memoria se derivan del mismo espacio de datos compartido por todos los procesadores. Este grupo se puede subdividir a su vez en dos grupos dependiendo si el grado de acoplamiento entre los procesadores es elevado o no.

El Cray X-MP, Alliant FX/8, Convex C3480 son algunos de los ordenadores del tipo MIMD. Los ordenadores del tipo SIMD solucionan el problema de sincronización de sus procesadores automáticamente. Pero éste es un problema nada trivial para los ordenadores MIMD. La CM es un ordenador del tipo SIMD que utiliza una estación de trabajo como front-end.

5.3.2 Ordenadores del tipo SIMD: Connection Machine CM-200

La Connection Machine (CM-200), es el ordenador representativo de este grupo. Consiste de un *front-end* que es un ordenador SISD, una unidad de proceso que consiste de 2K, 4K, 8K, 16K, 32K, 64K procesadores (K representa 1024 unidades), cada uno con memoria local, y un sistema de entrada salida. El *front-end* soporta un sistema operacional estandar y versiones extendidas de los lenguajes de programación estandar que facilitan el desarrollo del código. Los procesadores de la CM, se utilizan siempre y cuando una operación se pueda realizar de forma simultánea sobre un gran conjunto de datos. Estos datos residen en la memoria de la CM durante la ejecución y se opera con ellos, de forma paralela. Claro que una parte pequeña del flujo de datos puede residir en la memoria del *front-end* y se procesa de forma serial de la manera usual. El flujo del control es gobernado enteramente por el *front-end*, esto incluye almacenamiento y ejecución del programa y todas la interacciones con el usuario. Las operaciones paralelas se ejecutan luego de que el *front-end* envía los operandos a los procesadores de la CM. Una estructura de datos paralela (por ejemplo un vector n -dimensional sobre la CM) se distribuye sobre los procesadores de la CM, de tal forma que cada elemento se almacena en la memoria de sólo un procesador. El arreglo A(64000) es una variable paralela con 64000 elementos individuales. El arreglo D(1000,1000) consta de un millón de datos elementales. Cuando el número de elementos es superior al número de procesadores disponibles, el sistema opera en los procesadores virtuales, dando la apariencia de tener bastantes más procesadores. El sistema genera una partición de los procesadores físicos y distribuye la memoria entre estos, permitiendo así que el programador no tenga que forzar su estructura de datos a un determinado tamaño. Por ejemplo el vector A(64000) sobre un sistema de 8K procesadores, involucra la generación de 8 procesadores virtuales por procesador físico.

SISTEMAS DE COMUNICACIÓN

El sistema más general de comunicación en la CM, es el *router*, el cual permite que un procesador se comuniquen con cualquier otro. Se puede pensar en el router como un sistema que permite a todo procesador enviar un mensaje a otro procesador, donde todos los mensajes se envían y reciben en forma simultánea. La configuración más completa de la CM, consta de 4096 nodos, cada uno consta de un chip que contiene 16 procesadores. Cada uno de estos nodos está conectado en forma de hipercubo (En el caso de la CM de 64K procesadores, el hipercubo es de dimensión 12) Cada nodo sirve a sus 16 procesadores y los conecta con el resto mediante esta red. El nodo i está conectado al nodo j si y solamente si $|i - j| = 2^k = (i \oplus j) \vee 2^k$ para algún entero k (esto es, i y j están conectados si difieren sólo en el bit de posición k), en cuyo caso se dice que i y j están conectados a lo largo de la dimensión k .

La CM ofrece otro sistema de comunicación bastante más rápido llamado NEWS (North-East-West-South) que permite a un procesador comunicarse con sus vecinos en una red multidimensional de forma paralela. Este sistema es mucho más eficiente que el sistema de comunicación *router* porque explota tres diferentes métodos de transferencia de datos, dos de los cuales tienen un soporte especial a nivel de hardware. El sistema utiliza un subconjunto de la red física de tal forma que se puede pensar en ésta como una red n -dimensional.

CM FORTRAN

El lenguaje CM fortran, es una extensión del fortran 77, al cual se le han acoplado una serie de instrucciones que permiten el manejo de variables paralelas. Estas extensiones proporcionan una sintaxis y numerosas funciones intrínsecas que permiten manipular arreglos multidimensionales como objetos. La siguiente instrucción en fortran 77, si no se realiza ninguna modificación se ejecuta en el *front-end*:

```
do i =1 , n
  a(i) = a(i) + 1
end do
```

si se escribe como a objetos, se ejecuta en la CM:

```
a = a + 1
```

Se piensa ahora en a como un objeto que representa a un vector. CM-fortran soporta también instrucciones de enmascaramiento como:

```
WHERE(A .NE. 0) C = B/A
```

en este caso los procesadores que almacenan posiciones, donde el vector A es cero, no trabajan. Una presentación más detallada, puede encontrarse en [104].

5.3.3 Ordenadores del tipo MIMD

Estos ordenadores, se diferencian del tipo SIMD, en que los diferentes procesadores pueden ejecutar simultáneamente diferentes grupos de instrucciones. Esto les permite mayor flexibilidad en el tipo de tareas que desarrollan, pero aparece el problema de la sincronización. En las máquinas SIMD, la sincronización se realiza de forma automática, mientras que en las máquinas MIMD, se deben emplear otros mecanismos que aseguren que las tareas se realizan en el orden correcto y sobre el correcto conjunto de datos [94]. Otra variante en estas máquinas el tipo de memoria: local o compartida. En esta última, todos los procesadores pueden acceder a una zona de memoria común. Los procesadores pueden tener además zonas de memoria local para almacenar código o resultados intermedios, pero la zona común almacena los datos y resultados que se usan por otros procesadores. La ventaja se encuentra en la rápida comunicación entre los procesadores, pero la desventaja se encuentra en que varios procesadores deseen escribir en una zona común de forma simultánea. Una forma alternativa son los sistemas de memoria local, en donde cada procesador direcciona su propia memoria e intercambia información con los otros procesadores mediante un sistema de envío de mensaje entre procesadores.

Se introducen a continuación algunas medidas de paralelismo que permiten analizar el grado de paralelismo que se ha alcanzado con determinada implementación. Se define el *speedup* S_p [94], de un algoritmo paralelo como:

$$S_p = \frac{\text{tiempo de ejecución sobre un procesador}}{\text{tiempo de ejecución usando } p \text{ procesadores}}$$

En este caso se compara el tiempo de ejecución del programa que corre en p procesadores con el mismo pero corriendo en sólo un procesador. El máximo valor de S_p es p , que se obtiene cuando se hace una paralelización perfecta. La definición de S_p se puede mejorar comparando el programa que usa p procesadores con respecto al *mejor* programa que corre en sólo un procesador.

Se define eficiencia E_p de un programa paralelo como:

$$E_p = \frac{S_p}{p}$$

El objetivo central en el desarrollo de algoritmos paralelos es mantener el valor de S_p lo más grande que sea posible, pero existen varios factores que causan una degradación de este valor, entre ellos se encuentran [94]: a) la falta de un perfecto balance de carga, b) la comunicación entre procesadores, que debe

mantenerse al mínimo, c) la sincronización entre procesadores, d) tiempos de espera motivados por acceso simultáneo a las mismas zonas de memoria, etc.

La comunicación causa que algunos procesadores se encuentren en estado de espera de datos o resultados de otros procesadores produciéndose así un aumento en el tiempo de ejecución. La sincronización se requiere sobre todo en sistemas de memoria compartida, para evitar posibles errores en los accesos a memoria. El balance de carga se refiere a la manera como se distribuyen las diferentes tareas a los procesadores, esta asignación debe realizarse de tal forma que que todos trabajen lo más uniformemente posible.

5.4 ESTRATEGIAS DE IMPLEMENTACIÓN DE UN PROBLEMA CUASIESTÁTICO DE E.F. EN ORDENADORES PARALELOS

5.4.1 IMPLEMENTACIÓN SOBRE UN ORDENADOR MASIVAMENTE PARALELO (SIMD)

El problema descrito en la sección 5.2 y en general los problemas no lineales cuasiestáticos se pueden formular mediante la ecuación (5.2.1). La aplicación del método de Newton a este tipo de problemas, requiere la solución de un conjunto de sistemas lineales de la forma (5.2.2):

$$K_T(\mathbf{U}) \cdot \Delta \mathbf{U} = -\mathbf{F}^{res}(\mathbf{U}) \quad (5.4.1)$$

donde K_T es la matriz tangente (hessiano) del sistema (5.2.4). Con base en la experiencia que se tiene en la implementación de este tipo de problemas en ordenadores escalares, se puede anticipar que el tiempo de cálculo se concentra en la elaboración de tres tareas básicas:

- Cálculo de fuerzas residuales
- Cálculo de la matriz tangente
- Solución del sistema lineal

En el cálculo de las fuerzas residuales (5.2.1), se incluye el cálculo de \mathbf{F}^{int} y \mathbf{F}^{ext} . En éstas a su vez se incluyen los cálculos de las deformaciones y tensiones. Obsérvese que dependiendo del tipo de algoritmo que se utilice en la integración de la ecuación constitutiva, el costo del cálculo de las tensiones puede ser muy relevante. En el caso que se trata aquí (analizado en el capítulo 3) se consiguen expresiones cerradas de las tensiones, tanto para el caso de flujo asociado como no asociado. Esta característica permite anticipar que el costo computacional de las tensiones respecto a las otras tareas es bastante más bajo.

El empleo del método de Newton, requiere el cálculo de la matriz tangente K_T en cada iteración y si es necesario debe ensamblarse. Dependiendo del problema, el costo computacional de esta actualización puede ser elevado. Con

base en la experiencia con ordenadores escalares, se sabe que el mayor costo computacional se concentra en la solución del sistema lineal. La figura 5.2 corrobora esta afirmación. Esta figura presenta la distribución de tiempos, que se obtuvieron en un ordenador serial (Silicon Graphics Indigo) para llevar a cabo las tres tareas básicas descritas en 5.2.9. En las pruebas, se utilizó el algoritmo descrito en el apartado 5.2.

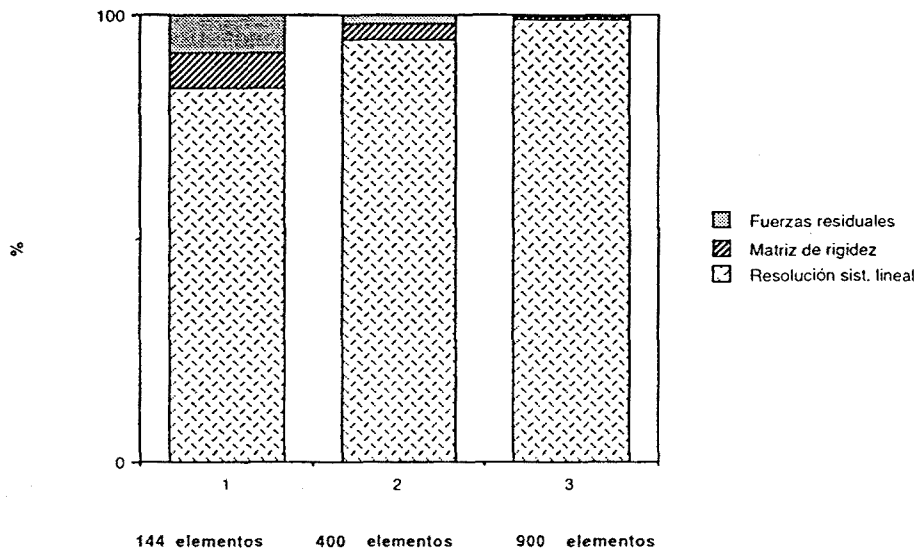


Figura 5.2 Distribución de tiempos en el caso serial, para diferentes números de elementos

5.4.1.1 Análisis de diferentes métodos de resolución del sistema lineal

A partir de la experiencia que se tiene respecto a la distribución de tiempos en un ordenador serial, se propone aquí llevar a cabo un análisis de cual sería el tipo de método más apropiado para solucionar el sistema lineal, en un ordenador masivamente paralelo. Se consideran en el análisis la siguientes tres familias: a) métodos directos, b) métodos iterativos (aproximados) y c) métodos iterativos (exactos).

a) Métodos directos

Existen diferentes métodos para solucionar un sistema lineal en forma directa, pero todos se caracterizan por la necesidad que se tiene de que la matriz y el vector de fuerzas residuales estén ensamblados antes de poder utilizarlos. En un ordenador masivamente paralelo, esto implica una gran cantidad de comunicación global entre procesadores, que atenúa el nivel de paralelismo. En las pruebas, se empleó el método de Gauss-Jordan [91], que se basa en

la eliminación sucesiva de las incógnitas del problema. Se utilizó la rutina proporcionada por la librería CMSSL (CM-FORTRAN) [92].

b) Métodos iterativos (aproximados)

Supóngase que se desea solucionar el siguiente sistema lineal:

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{f} \quad (5.4.2)$$

La idea de este tipo de métodos se basa en encontrar la solución del sistema (5.4.2) mediante la solución en estado estacionario de la siguiente ecuación diferencial de primer orden [92][93]:

$$\begin{aligned} \mathbf{B}\dot{\mathbf{u}}(t) + \mathbf{K} \cdot \mathbf{u}(t) &= \mathbf{f} \\ \mathbf{u}(0) &= \mathbf{u}_0 \end{aligned} \quad (5.4.3)$$

la integración explícita de la ecuación anterior, se puede escribir como:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \mathbf{B}^{-1} \cdot (\mathbf{f} - \mathbf{K} \cdot \mathbf{u}_n) \quad (5.4.4)$$

sujeto a la condición de convergencia:

$$\rho(\mathbf{I} - \Delta t \mathbf{B}^{-1} \cdot \mathbf{K}) < 1 \quad (5.4.5)$$

donde $\rho(\cdot)$ representa el radio espectral de la matriz (\cdot) . Debido a que la solución del sistema (5.4.2) sólo se alcanza como solución del estado estacionario de (5.4.3), estos métodos se conocen también como métodos de relajación viscosa.

Los diferentes métodos iterativos de este tipo, se diferencian por la elección de la matriz preconditionadora \mathbf{B} :

$$\begin{aligned} \text{método de Jacobi} \quad \mathbf{B} &= \text{diag}(\mathbf{K}) \\ \text{método de Gauss-Saidel} \quad \mathbf{B} &= \mathbf{D} + \mathbf{L} \quad \text{donde } \mathbf{L} = \text{triang-inf}(\mathbf{K}) \\ \text{método de sobrerelajación} \quad \mathbf{B}(\omega) &= \frac{\mathbf{D}}{\omega} (\mathbf{I} - \omega \mathbf{A}) \quad \text{donde } \mathbf{A} = \mathbf{D}^{-1} \cdot \mathbf{L} \end{aligned} \quad (5.4.6)$$

el parámetro de relajación ω se elige de tal forma que se mejora la convergencia. Obsérvese que el método de Jacobi a diferencia de los otros métodos sólo requiere el ensamblaje de la diagonal de la matriz \mathbf{K} . El costo computacional del método se puede calcular como:

$$N_{iter} \cdot T \quad (5.4.7)$$

donde T es el tiempo utilizado en calcular la ecuación de recurrencia (5.4.4) y N_{iter} es el número de iteraciones que se necesitan para converger. Como puede verse, éste es un método bastante simple de implementar, con la desventaja, que en general la convergencia es bastante lenta. Esto es debido a que la integración explícita formulada en (5.4.4) es condicionalmente estable y el tamaño del incremento Δt está acotado según la condición (5.4.5).

c) Métodos iterativos (exactos)

Con este nombre se quieren agrupar los métodos como: Gradientes Conjugados (GC) y GMRES, con sus respectivos preconditionadores, que sin incluir problemas de redondeo en las operaciones de punto flotante, alcanzan la solución luego de un número finito de pasos.

Método de los Gradientes Conjugados (GC)

El algoritmo correspondiente a este método es el siguiente [94]:

$$\begin{aligned}
 &\text{Dado } \mathbf{x}^0, \mathbf{r}^0 = \mathbf{F} - \mathbf{K} \cdot \mathbf{x}^0, \tilde{\mathbf{r}}^0 = \mathbf{M}^{-1} \cdot \mathbf{r}^0, \mathbf{p}^0 = \tilde{\mathbf{r}}^0 \\
 &\text{para } k = 0, 1, \dots \\
 &\quad \alpha_k = -(\tilde{\mathbf{r}}^k, \mathbf{r}^k) / (\mathbf{p}^k, \mathbf{K} \cdot \mathbf{p}^k) \\
 &\quad \mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{p}^k \\
 &\quad \mathbf{r}^{k+1} = \mathbf{r}^k + \alpha_k \mathbf{K} \cdot \mathbf{p}^k \\
 &\quad \text{si } \|\mathbf{r}^{k+1}\| \geq \epsilon \text{ entonces} \\
 &\quad \quad \tilde{\mathbf{r}}^{k+1} = \mathbf{M}^{-1} \cdot \mathbf{r}^{k+1} \\
 &\quad \quad \beta_k = (\tilde{\mathbf{r}}^{k+1}, \mathbf{r}^{k+1}) / (\tilde{\mathbf{r}}^k, \mathbf{r}^k) \\
 &\quad \quad \mathbf{p}^{k+1} = \tilde{\mathbf{r}}^{k+1} - \beta_k \mathbf{p}^k \\
 &\quad \text{si no} \\
 &\quad \quad \text{pare} \\
 &\text{fin(para)}
 \end{aligned} \tag{5.4.8}$$

Este método sólo es válido cuando la matriz \mathbf{K} es simétrica definida positiva. \mathbf{M} es la matriz de preconditionamiento del sistema, usualmente se define $\mathbf{M} = \text{diag}(\mathbf{K})$. En cada iteración del algoritmo (5.4.8), se deben calcular cuatro productos escalares, tres operaciones vectoriales (saxpy) de la forma $\alpha \mathbf{x} + \mathbf{y}$, dos divisiones, y un producto *matriz-vector*, de la forma $\mathbf{y} = \mathbf{K} \cdot \mathbf{p}$. Las operaciones vectoriales se llevan a cabo de forma natural y eficiente en un ordenador de estas características. Queda por resolver el problema de calcular eficientemente el producto $\mathbf{K} \cdot \mathbf{p}$. La matriz \mathbf{K} se puede escribir como:

$$\mathbf{K} = \sum_{e=1}^{N_e} \mathbf{K}^e \tag{5.4.9}$$

donde Σ representa el operador de ensamblaje y \mathbf{K}^e las matrices elementales, por lo tanto el producto se puede escribir como:

$$\mathbf{y} = \mathbf{K} \cdot \mathbf{p} = \left(\sum_{e=1}^{N_e} \mathbf{K}^e \right) \cdot \mathbf{p} = \sum_{e=1}^{N_e} \mathbf{K}^e \cdot \mathbf{p}^e \tag{5.4.10}$$

donde \mathbf{p}^e se obtiene como la proyección de \mathbf{p} , sobre los nodos de elemento e . Entonces el producto se resuelve calculando los productos elementales $\mathbf{K}^e \cdot \mathbf{p}^e$ y luego ensamblando los resultados parciales. La implementación eficiente de este producto, depende de la estructura de datos que se elija para almacenar la matriz \mathbf{M} y el vector \mathbf{p} .

Desde el punto de vista de la matriz, lo más natural es definirla a nivel elemental, de esta forma, a cada procesador (en general, procesador virtual) se le asocia un elemento de la malla de elementos finitos. Desde el punto de vista de las operaciones vectoriales, lo más conveniente es asignar a cada procesador, un grado de libertad del vector, así las operaciones se realizan de forma paralela.

La consecuencia de esta elección, es que las operaciones de proyección $\mathbf{p}^e = \mathbf{p}|_e$ y ensamblaje $\mathbf{p} = \sum_{e=1}^{N_e} \mathbf{p}^e$, requieren comunicación entre los procesadores, como se estudiará más adelante.

Método GMRES (Generalized Minimal RESidual)

Éste es un método para solucionar sistemas lineales [95], donde la matriz del sistema puede no ser simétrica. El núcleo del método es la minimización del residuo, usando la descomposición del vector solución sobre un subespacio afín. Este subespacio afín se construye a partir de la evaluación inicial de la solución y una base de los vectores de Krylov. GMRES, al igual que GC, es teóricamente un método directo que finaliza en un número fijo de iteraciones, pero en la práctica se utiliza como un método iterativo permitiendo un gran ahorro en la memoria requerida. Desde el punto de vista de los elementos finitos, GMRES tiene dos ventajas importantes: 1) se puede utilizar para solucionar sistemas no simétricos y 2) no requiere el ensamblaje de la matriz del sistema (similar a GC). El siguiente diagrama representa el algoritmo del método. El término k_0 representa la dimensión del subespacio de Krylov y ϵ el valor de la tolerancia exigida. La memoria requerida en cada iteración crece proporcionalmente con k_0 y el número de multiplicaciones es proporcional a $\frac{1}{2}k_0^2N$ donde N es el número de incógnitas. Si al finalizar la construcción de la solución aproximada, no se ha alcanzado el nivel de precisión requerido, entonces debe procederse a incrementar la dimensión de Krylov y volver a empezar el proceso. Pero esto conduce a un aumento significativo en la memoria requerida, por esta razón, en la práctica el algoritmo se acompaña de un mecanismo de reinicialización, que consiste en volver a empezar el proceso con el último residuo obtenido.

ALGORITMO DEL MÉTODO GMRES

```

x ← 0
TEST ← FALSO
MQ (TEST = FALSO) HAGA
(inicialización del lazo de GMRES)
  u1 ← b - K · x
  e ← (||u1||, 0, 0, ..., 0)t
  u1 ← u1/||u1||
  Para i = 1 hasta k0 haga
    (construir el (i + 1)esimo vector de Krylov)
    ui+1 ← Kui
    para j = 1 hasta i haga
      βi+1,j ← ui+1 · uj
      ui+1 ← ui+1 - βi+1,juj
    fin(para)
    ui+1 ← ui+1/||ui+1||
    construcción de la iesima columna de la matriz de Hessenberg
    y actualización de la factorización QR
    hi ← (βi+1,1, βi+1,2, ..., βi+1,i, ||ui+1||)t
    para j = 1 hasta i - 1 haga
       $\begin{pmatrix} h_{j,i} \\ h_{j+1,i} \end{pmatrix} \leftarrow \begin{pmatrix} \cos\theta_j & \sin\theta_j \\ -\sin\theta_j & \cos\theta_j \end{pmatrix} \begin{pmatrix} h_{j,i} \\ h_{j+1,i} \end{pmatrix}$ 
    fin(para)
    (cosθi, sinθi) ←  $\frac{1}{\sqrt{h_{i,i}^2 + h_{i+1,i}^2}}(h_{i,i}, h_{i+1,i})$ 
    (hi,i, hi+1,i) ← (√(hi,i2 + hi+1,i2), 0)
    (Actualizar: e)
    (ei, ei+1) ← (cosθi ei, -sinθi ei)
    TEST ← (||ei+1|| ≤ ε)
    si TEST entonces
      go to(**)
    fin(si)
  fin(Para)
  (**) calcule x
  solución del sistema triangular: Hy = e
  x ← x + ∑j=1i yjuj
  fin(MQ)

```

Entre las principales operaciones que deben realizarse se destacan: operaciones vectoriales (saxpy), productos escalares y un producto matriz-vector. Adicionalmente se encuentra el costo computacional de una minimización que se lleva a cabo mediante una descomposición QR. Se puede anticipar que en la práctica, cuando la dimensión del espacio de krylov es muy pequeña respecto a la dimensión del problema, el costo de esta minimización es bastante bajo, comparado con el costo computacional del resto del problema. La figura 5.3 presenta la importancia relativa del producto matriz-vector en el GMRES implementado en un ordenador serial (Silicon Graphics). Esta distribución se ha obtenido para diferentes números de elementos. Como se vé, el producto ocupa más del 90% del costo, respecto al resto de las operaciones, e incrementa su importancia con el número de elementos. Por esta razón, el producto matriz-vector se convierte en el punto más importante en el proceso de implementación de este método en un ordenador masivamente paralelo. Esto es debido a que el resto de las operaciones son altamente paralelizables.

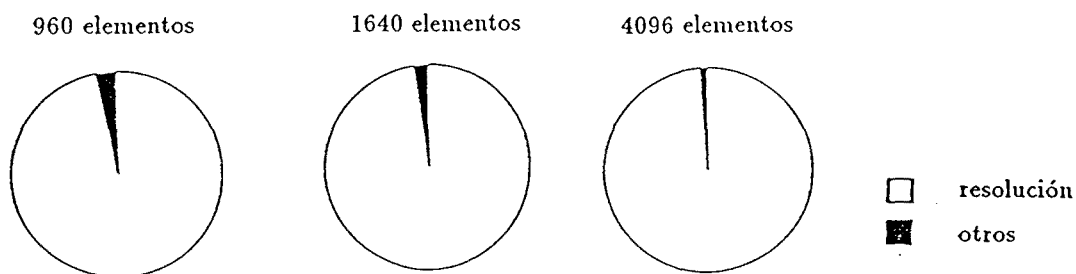


Figura 5.3 Distribución de tiempos en el GMRES

Comparación de los diferentes métodos

En la figura 5.4 se encuentra el tiempo total empleado por los diferentes métodos de resolución para diferentes números de ecuaciones y tres valores del error relativo $\epsilon = 1 \times 10^{-3}$, 1×10^{-5} , 1×10^{-8} . Por supuesto que el error relativo sólo concierne a los métodos iterativos ya que el costo de los métodos directos es el mismo independiente del error relativo. Se utilizó la descomposición LU proporcionada por la librería del sistema. Además se implementó una versión empleando matriz banda. Como se puede ver, los métodos iterativos (exactos, GMRES y G.C.) son los más eficientes y por esta razón, son los métodos que se emplearán en el resto de los cálculos.

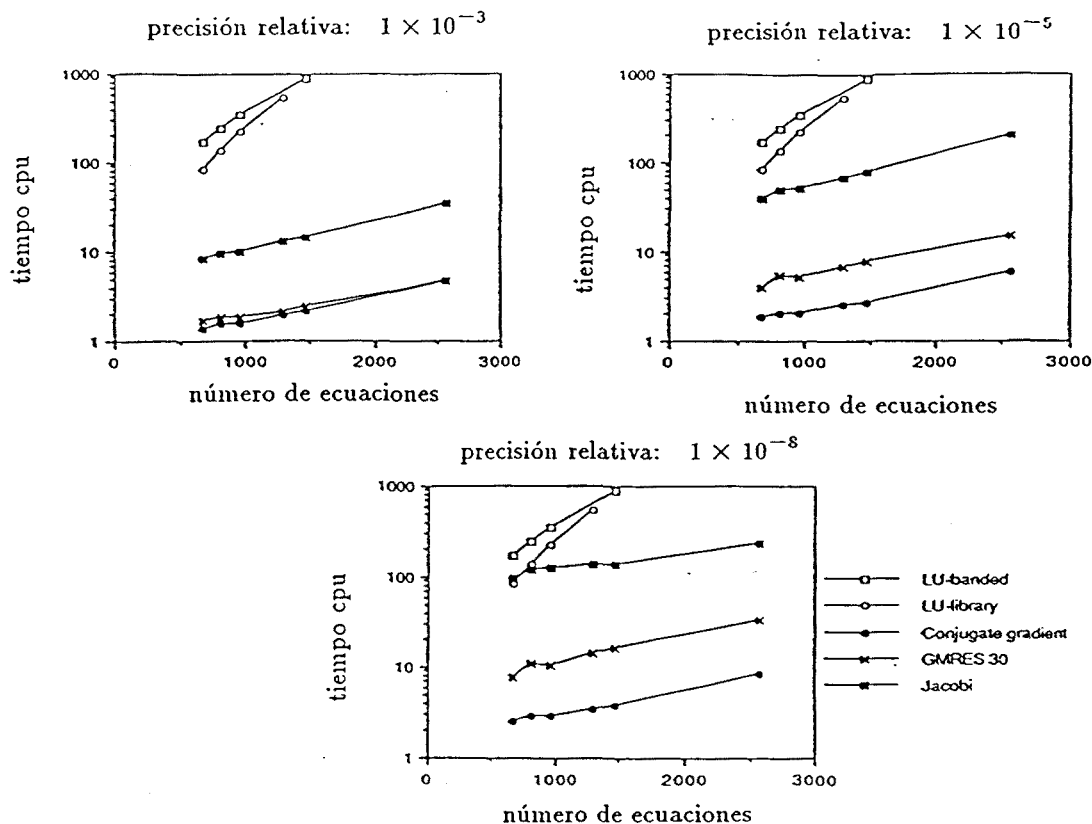


Figura 5.4 Comparación del tiempo de cálculo requerido por los diferentes sistemas de resolución del sistema lineal

5.4.1.2 Estructura de datos sobre un ordenador masivamente paralelo

Uno de los puntos claves en el cálculo masivamente paralelo es la elección de la *estructura de datos*. El éxito de cualquier implementación está basado en la explotación apropiada (con un mínimo de comunicación entre procesadores) de la capacidad del ordenador para realizar cálculos de forma paralela. En el contexto del método de los elementos finitos, muchas de las operaciones que se llevan a cabo a nivel de un elemento se repiten para el resto de los elementos y además, el intercambio de información se realiza entre elementos vecinos. Por esta razón, una estructura de datos a nivel elemental [81][84][85][86] [89], emerge de forma natural. Este tipo de estructura asocia cada elemento de la malla con un procesador físico (o virtual) de la CM. El término procesador virtual se refiere a la capacidad de la CM de dividir cada uno de sus procesadores en el número de procesadores virtuales que sea necesario (potencia de dos) para poder almacenar una variable paralela con una longitud mayor que el número de procesadores físicos. Esta operación es transparente para el usuario, que puede seguir pensando en ellos como procesadores individuales. Todos los datos concernientes a un elemento, se almacenan en la memoria local del procesador

correspondiente, de tal forma que cualquier operación que se realice a nivel del elemento no requiere comunicación.

En la sección 5.4.1.1 se concluyó que los únicos métodos para solucionar los sistemas lineales asociados con el problema no lineal, que son de interés en el contexto de la paralelización masiva son: GC y GMRES. Esto se debe a que en su mayoría involucran operaciones vectoriales y productos matriz-vector, donde la matriz no necesita ensamblarse. La elección de una estructura elemental para la matriz, implica que el vector correspondiente al producto matriz-vector debe estar almacenado de forma similar (elemental) para que el producto se realice de forma eficiente. Este almacenamiento obliga a repetir información, ya que la información concerniente a un nodo (o grado de libertad) debe aparecer tantas veces como elementos que contengan a este nodo (o grado de libertad). El resto de las operaciones tanto del GC como del GMRES, corresponden a operaciones vectoriales y productos escalares, entonces la estructura de datos natural para el resto son: vectores de rango uno, cuya dimensión es el número de grados de libertad, en esta estructura, la información de cada nodo sólo aparece una vez.

Estructura de datos mixta

Con base en el análisis anterior, se sugiere entonces una estructura de datos mixta, que consiste en mezclar las dos estructuras (elemental y nodal) de tal forma que, por ejemplo, la matriz de rigidez se almacena de forma elemental, pero los vectores que intervienen en el resto de las operaciones del método de resolución, se almacenan a nivel nodal. El producto matriz-vector se ilustra en la figura 5.5. Este tipo de estructura implica el uso de dos rutinas de comunicación: una rutina que envía la información de los grados de libertad y la lleva a los elementos (*gather*) y una rutina que pasa la información de los elementos a los grados de libertad, como proceso de ensamblaje (*scatter*).

Otras estructuras de datos

Se pueden considerar adicionalmente, las siguientes estructuras: una estructura completamente elemental y una estructura completamente nodal. La primera estructura, almacena elemento por elemento, todos los vectores que intervienen en la resolución del sistema lineal (figura 5.6). Esto implica que la información que se tiene a nivel de un nodo, aparece más de una vez (aparece en cada procesador asignado a un elemento que contenga este nodo) y que la cantidad de operaciones para llevar a cabo sumas vectoriales, o productos escalares aumenta, pero esta estructura elimina una de las etapas de comunicación, la etapa de envío de información de los vectores de rango uno, a los vectores a nivel elemental.

Lamentablemente, en su gran mayoría, las rutinas disponibles en la librería de la CM, para realizar este tipo de comunicación entre estructuras, restringen

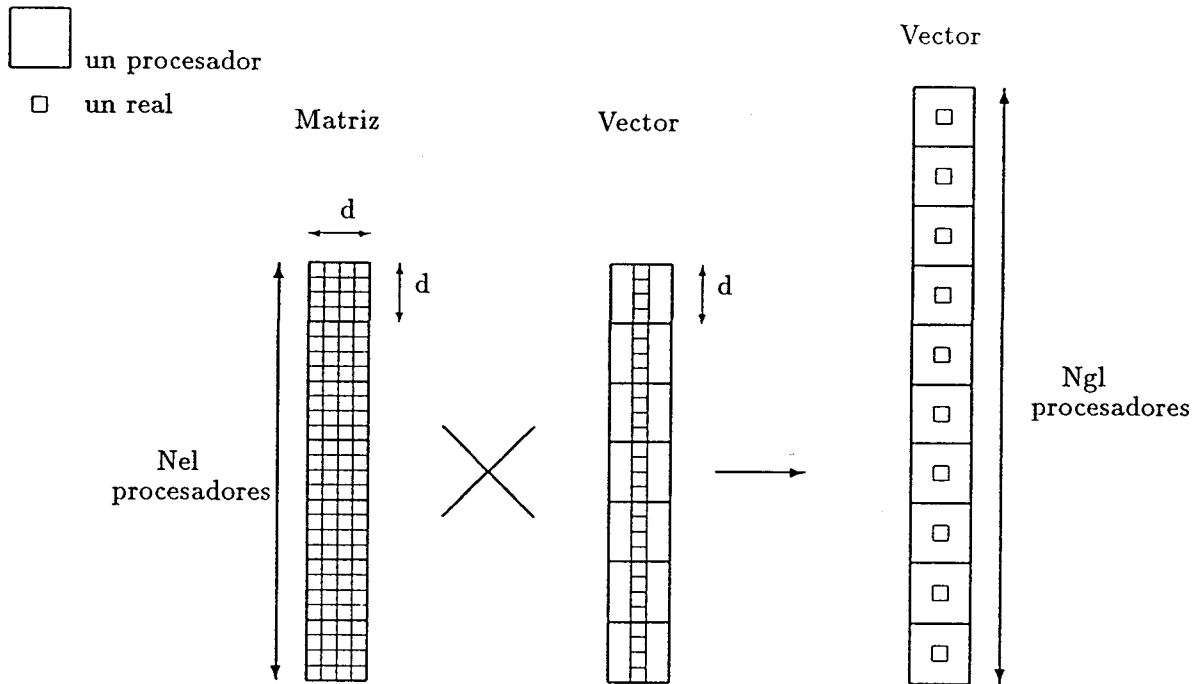


Figura 5.5 Estructura de datos mixta

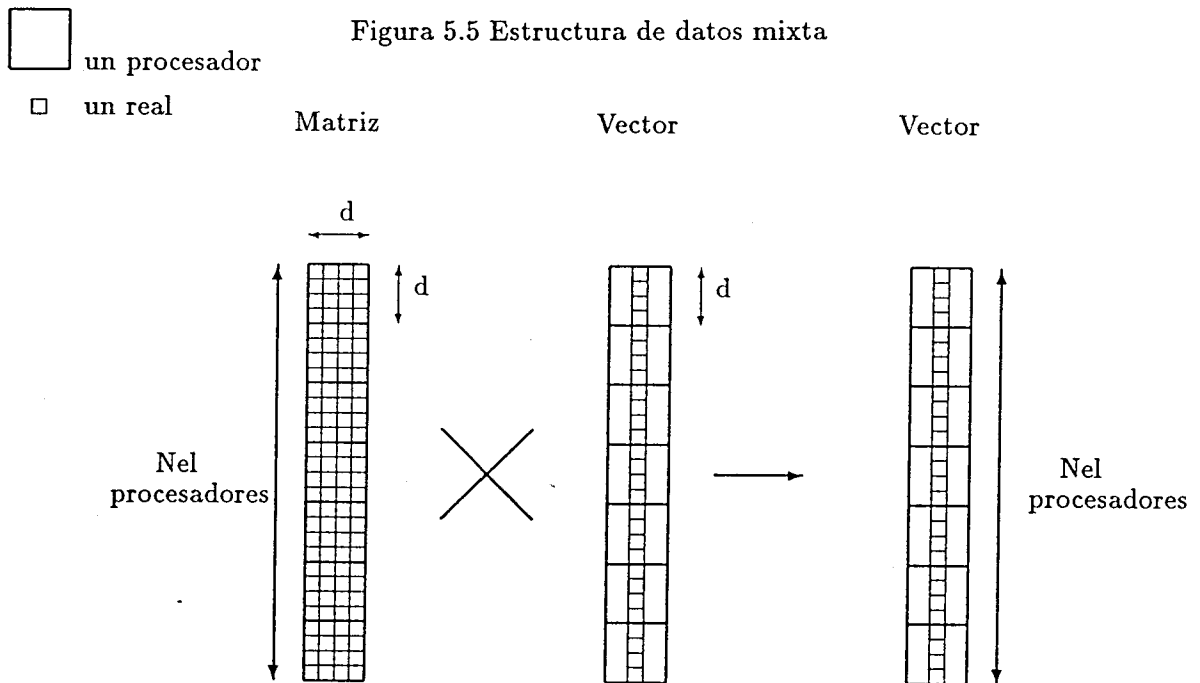


Figura 5.6 Estructura de datos elemental

alguno de los vectores a que sea de rango uno, lo que elimina automáticamente la viabilidad de esta estructura. Una estructura completamente nodal permite realizar las operaciones vectoriales que intervienen en la resolución, de manera muy eficiente, pero con la desventaja que destruye la naturalidad de las operaciones que se llevan a cabo a nivel de un elemento, conduciendo a

complicaciones en su implementación. Este tipo de estructura no se consideró en el presente trabajo. Un análisis de este tipo de estructuras aplicado a un problema elástico lineal puede encontrarse en [96].

Acerca de la comunicación

Las anteriores estructuras de datos obligan a la utilización del sistema de comunicaciones conocido como ROUTER [104], ya que se requiere el envío de mensajes entre procesadores sin un patrón general definido. En el caso particular en que la malla de elementos finitos sea completamente estructurada (figura 5.7), se puede recurrir al sistema de comunicaciones NEWS.

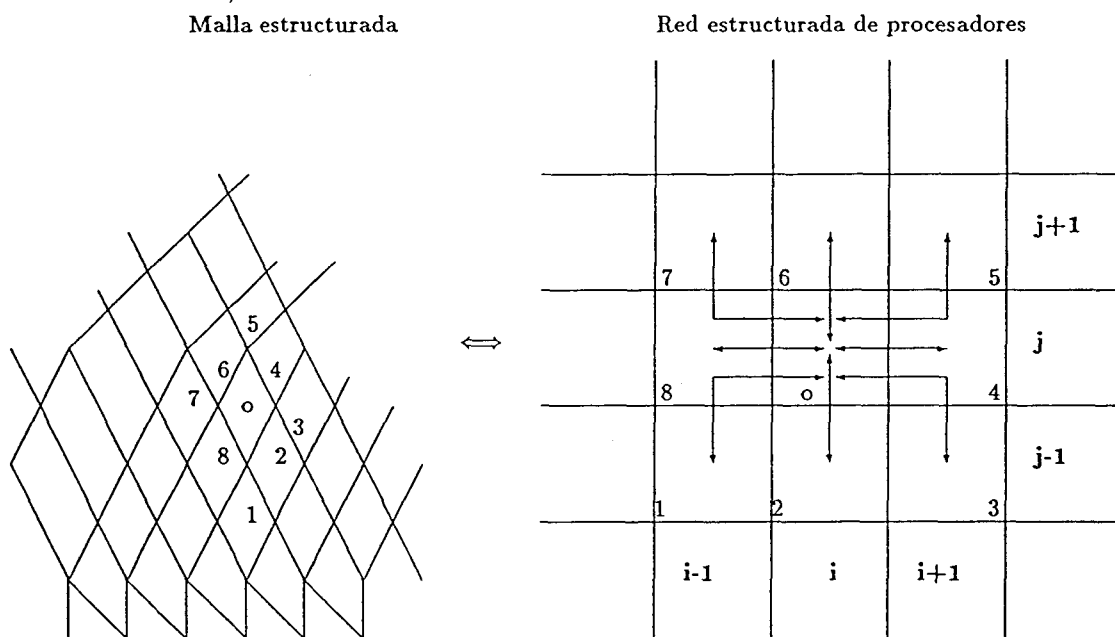


Figura 5.7 malla estructurada

Este sistema, permite la comunicación entre procesadores vecinos de una forma más eficiente. La malla de elementos finitos se proyecta sobre una red bidimensional (o tridimensional según el caso) de procesadores, de tal forma que a elementos vecinos se les asigna procesadores vecinos, esto hace que, junto con la estructura regular de la red, la comunicación entre ellos sea más eficiente. Este tipo de estructura de datos se implementó y comparó en el caso de un problema elástico con una utilizando el sistema de comunicaciones ROUTER. La figura 5.8, presenta los resultados de tiempos de ejecución, que obtuvieron para diferentes números de elementos. El sistema de comunicación NEWS, permite obtener reducciones en los tiempos de ejecución, del orden del 50%, pero bajo la restricción, que la malla de elementos finitos, sea completamente estructurada. Belytschko y Plaskacz [97], extienden el empleo de este sistema de comunicaciones a mallas parcialmente estructuradas. De esta forma, cada

subdominio, se trata como se explicó arriba y los elementos de la interfaz entre subdominios se tratan empleando el ROUTER como sistema de comunicación.

Como se ha dicho, en el caso de mallas no estructuradas debe emplearse el sistema general de comunicación: ROUTER, que permite intercambiar información entre cualesquiera arreglos de procesadores. La librería CMSSL de la CM, ofrece dos conjuntos de rutinas de comunicación que se conocen como: CPS (Communication Primitive Set) y CC (Communication Compiler). Ambas ofrecen rutinas para realizar las operaciones de ensamblaje (*scatter*) y expansión (*gather*). El primer conjunto ofrece las rutinas:

`sparse_util_gather`

`sparse_util_scatter`

y sus correspondientes rutinas de inicialización:

`sparse_util_gather_setup`

`sparse_util_scatter_setup`

Estas rutinas de inicialización calculan una o dos variables internas utilizadas por el resto de las rutinas, con base en la topología de la comunicación entre el arreglo y el vector.

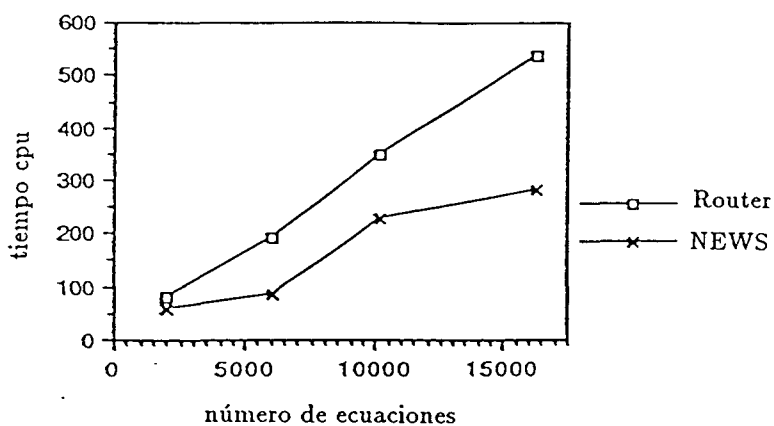


Figura 5.8 Problema elástico. Comparación entre los sistemas ROUTER y NEWS

El segundo grupo (Communication Compiler), es bastante más amplio que el primero y a diferencia de éste, optimiza la manera como se enviará la información entre los distintos procesadores. Dado un patrón de comunicación (la conectividad en este caso) se calcula un camino óptimo llamado traza (*trace*), el cual es una variable entera que contiene un apuntador a una zona de memoria del front-end. Esta zona, que se asigna durante el cálculo de la traza, contiene la descripción del patrón de comunicación optimizado y contiene además apuntadores a los datos almacenados en la CM. Estas zonas se usan mientras la traza no se elimine. Cuando el patrón de comunicación no cambia,

entonces sólo basta calcular la traza una vez y se puede usar indefinidamente. El método elegido, determina como se calculará esta traza. Se dispone de varios métodos:

CMSSL_method_fastgraph
 CMSSL_method_nop
 CMSSL_method_sort_and_scan
 CMSSL_method_automatic

Una completa descripción de los diferentes métodos puede encontrarse en [92]. En [98] se llevó a cabo un estudio comparativo de los diferentes métodos, para varios tamaños de mallas (144, 900, 1980, 8010 elementos). Se midió el tiempo que tomaba calcular la traza y un número determinado de operaciones de expansión y de ensamblaje (*gather y scatter*). De los resultados obtenidos se observó que para un gran número de operaciones de ensamblaje, cuando el tamaño del problema es pequeño o mediano (2000 elementos) es conveniente el uso del método serial combining y para un problema grande (10000 elementos) el método apropiado es **FastGraph**. Para las operaciones de expansión se encontró que el método **nop** es el adecuado para problemas pequeños y **FastGraph** para problemas grandes.

5.4.2 IMPLEMENTACIÓN SOBRE UN ORDENADOR DE BAJA GRANULARIDAD

5.4.2.1 Método del complemento de Schur

El método del complemento de Schur (o de las subestructuras) [99][100][101], que forma parte de los métodos de subdominios, conduce a una forma alternativa de solucionar el sistema lineal:

$$K_T(U_{n+1}^k) \cdot \Delta U_{n+1}^k = -F^{res}(U_{n+1}^k) \quad (5.4.11)$$

correspondiente a la linealización del problema (5.2.1). La idea básica es dividir el dominio de elementos finitos en subdominios, que pueden tratarse de forma independiente, quizás asignando cada subdominio a un procesador (figura 5.9).

Supóngase que la numeración de los grados de libertad es tal que la matriz de rigidez se pueda escribir de la siguiente forma:

$$K = \begin{bmatrix} K_{11} & 0 & 0 & 0 & 0 & K_{1I} \\ 0 & \ddots & 0 & 0 & 0 & \vdots \\ 0 & 0 & K_{jj} & 0 & 0 & K_{jI} \\ 0 & 0 & 0 & \ddots & 0 & \vdots \\ 0 & 0 & 0 & 0 & \ddots & \vdots \\ K_{I1} & \cdots & K_{Ij} & \cdots & \cdots & K_{II} \end{bmatrix} \quad (5.4.12)$$

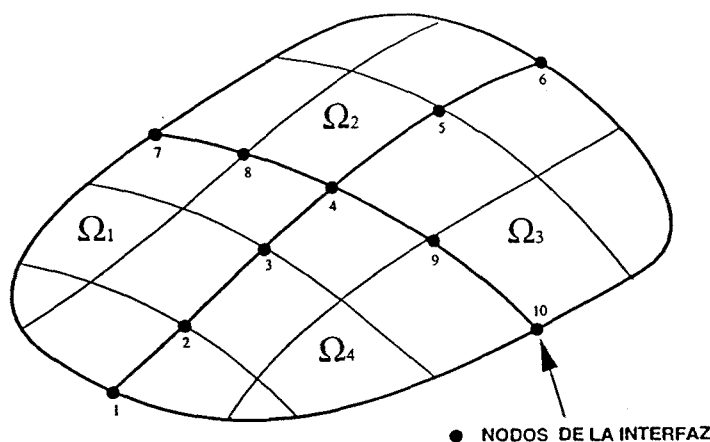


Figura 5.9 Descomposición de una malla de elementos finitos, en subdominios

donde el índice I denota la interfaz entre subdominios. El método consiste en resolver primero el siguiente problema en la interfaz:

$$(K_{II} - \sum_{j=1}^{N_s} K_{Ij} K_{jj}^{-1} K_{jI}) U_I = F_I - \sum_{j=1}^{N_s} K_{Ij} K_{jj}^{-1} F_j \quad (5.4.13)$$

donde U denota el vector de desplazamiento nodal, F las fuerzas exteriores y N_s el número de subdominios. El término $K_{II} - \sum_{j=1}^{N_s} K_{Ij} K_{jj}^{-1} K_{jI}$ se conoce como complemento de Schur [99][100][101]. Esto motiva que aquí se emplee la denominación: método de subestructuras ó método del complemento de Schur indistintamente.

Una vez el problema en la interfaz, ha sido resuelto, los subdominios se pueden tratar de forma independiente, solucionando el siguiente sistema en cada uno de ellos:

$$K_{jj} U_j = F_j - K_{jI} U_I \quad (5.4.14)$$

5.4.2.2 Implementación del método del complemento de Schur

En la práctica, la matriz K no tiene la estructura presentada en (5.4.12). En las implementaciones que se describirán, la matriz de cada subdominio se almacena por aparte, como una matriz banda. Las submatrices pueden calcularse en paralelo. La ecuación (5.4.13) se puede reformular en términos de las submatrices K_s como sigue:

$$[\sum_s (K_{I_s I_s} - K_{I_s s} K_{s s}^{-1} K_{s I_s})] U_I = \sum_s F_{I_s} - \sum_s K_{I_s s} K_{s s}^{-1} F_s \quad (5.4.15)$$

donde \sum representa el operador de ensamblaje y donde se supone que cada submatriz tiene la siguiente forma:

$$K_s = \begin{bmatrix} K_{ss} & K_{sI_s} \\ K_{I_s s} & K_{I_s I_s} \end{bmatrix} \quad (5.4.16)$$

donde I_s es la proyección de la interfaz I sobre el subdominio s . El sistema (5.4.15), puede solucionarse ensamblando o no el complemento, en ambos casos se considera un método iterativo (GC o GMRES) para solucionar este problema en la interfaz, entonces se debe analizar como realizar el producto matriz-vector en estos casos.

Si se ensambla la matriz de la interfaz, estos productos se pueden realizar de forma estandar, si no, entonces los productos se pueden calcular llevando a cabo las siguientes tres etapas:

1) El vector U (almacenado como un arreglo, de tamaño, el número de grados de libertad) restringido a la interfaz U_I se proyecta sobre la interfaz de cada subdominio U_{I_s} .

2) los productos $(K_{I_s I_s} - K_{I_s s} K_{ss}^{-1} K_{s I_s}) U_{I_s}$ se calculan en cada subdominio. Las matrices K_{ss}^{-1} no se calculan explícitamente sino que, el producto se calcula mediante la factorización LU_{ss}

3) los subvectores resultantes se ensamblan en el vector resultante R_I

Nótese que los productos hechos en cada subdominio, se pueden calcular independientemente el uno de los otros y así estos pueden ser hechos en paralelo (figura 5.10, ver además figura 5.9 para la numeración).

Una vez resuelto la interfaz del problema, se procede a resolver N_s sistemas lineales:

$$K_{ss} U_s = F_s - K_{s I_s} U_{I_s} \quad (5.4.17)$$

(posiblemente en paralelo) usando la descomposición LU de cada uno de ellos, esto lleva a la solución completa de (5.4.11). El algoritmo presentado en la figura 5.9.b, presenta las modificaciones necesarias que deben hacerse sobre el programa serial para incluir el método del complemento de Schur. En sombreado se encuentran las operaciones que se realizan en paralelo.

5.4.2.3 Implementación sobre un ordenador MIMD

Se utilizó para la implementación un ordenador CONVEX C3480, con 8 procesadores. Se describen a continuación, algunos aspectos referentes al código empleado.

La manera más simple para tomar ventaja de la arquitectura paralela de un ordenador del tipo MIMD, es contar con la capacidad de generación de código paralelo por parte del compilador. Simplemente activando la opción paralela del compilador, se consigue la generación de código paralelo a nivel de

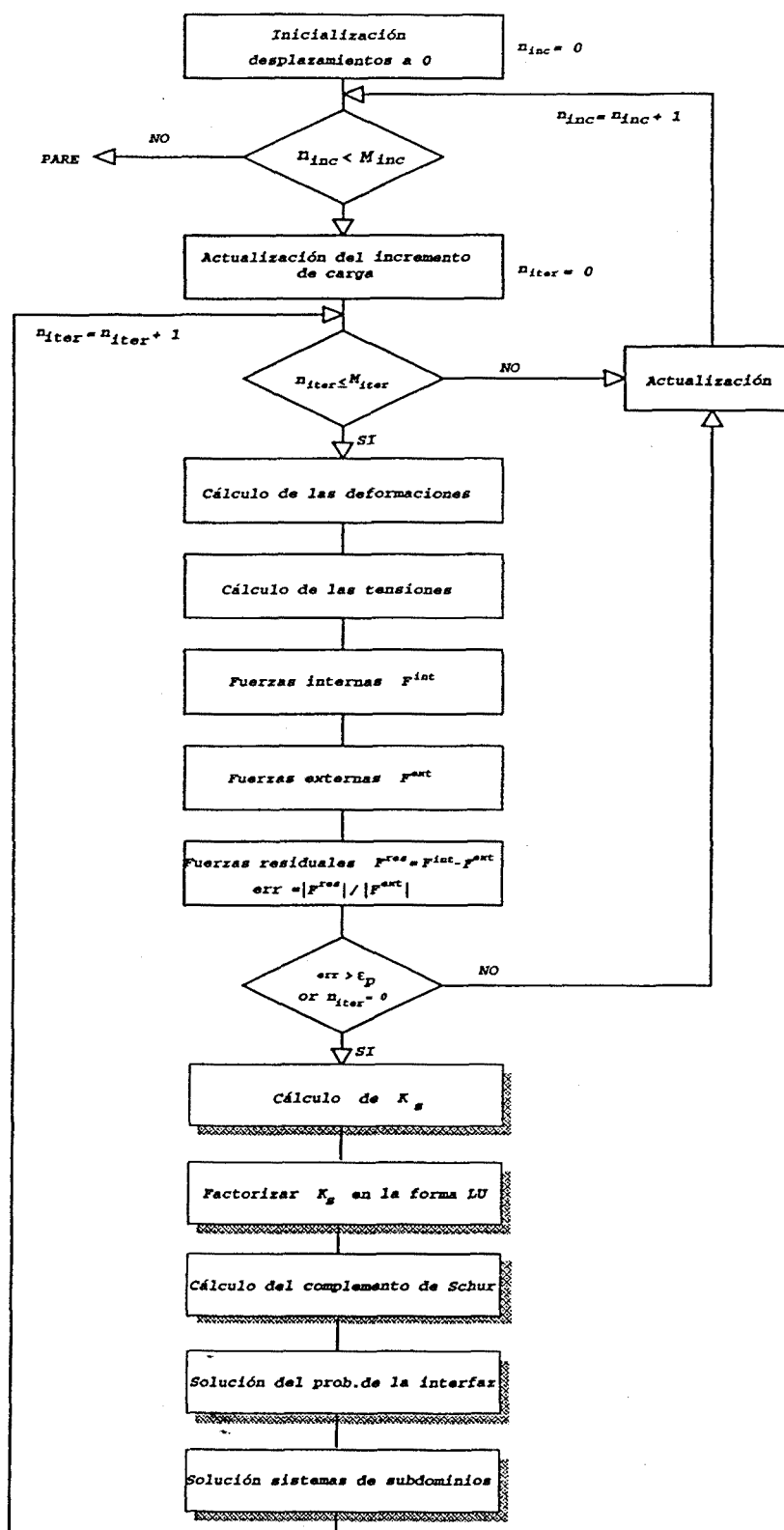


Figura 5.9.b Algoritmo utilizando el método del complemento de Schur (En sombreado se encuentran las operaciones que se realizan en paralelo.)

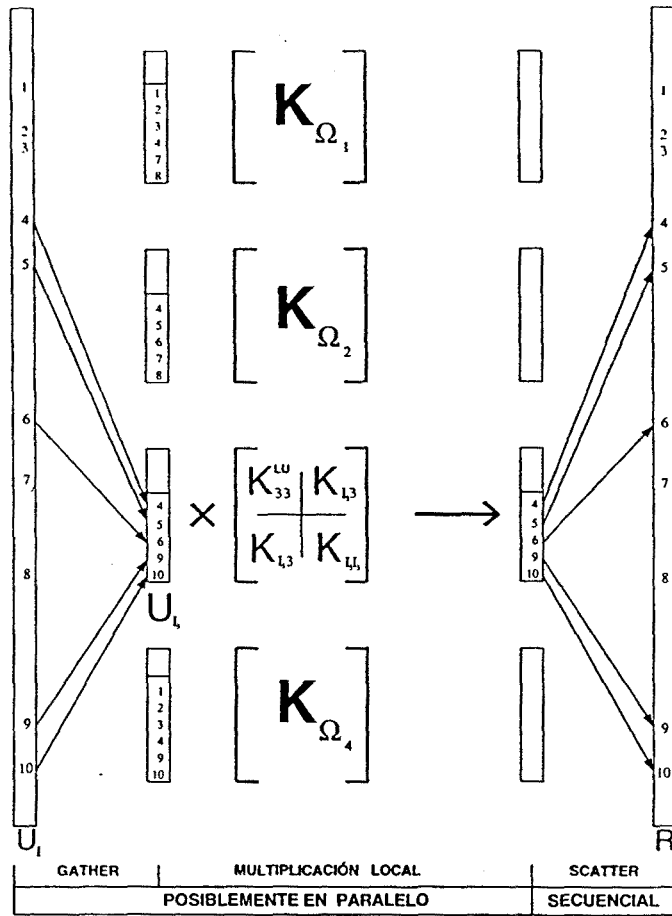


Figura 5.10 Etapas del producto matriz-vector, cuando el complemento de Schur no se ensambla

los lazos internos del programa, sin ninguna modificación del código serial. Sin embargo, como el compilador paraleliza lazos internos (los cuales en su mayoría son pequeños) y no paraleliza lazos conteniendo funciones o llamadas a rutinas, se requiere alguna manipulación para ganar en eficiencia.

La idea es introducir directivas de compilación para forzar la paralelización de lazos de alto nivel [102][103]. Estos lazos deben contener bastante trabajo por hacer, de esta forma el tiempo adicional consumido por el sistema para la manipulación de la información se compensa con la ganancia obtenida de la paralelización. Las funciones o rutinas que se llaman dentro de estos lazos se deben compilar apropiadamente para generar el código de reingreso. Las variables las cuales pueden ser accedidas por varios procesadores al mismo tiempo, deben tratarse con especial atención. Por ejemplo, la siguiente parte

de código (serial) correspondiente al programa de subestructuras, realiza un producto matriz-vector:

```

...
REAL*8    x(Max_Number_Dof),y(Max_Number_Dof)
REAL*8    matrix(Max_Number_Dof_Domain,2*Max_Bandwith-1,Max_Domain)
REAL*8    temp1(Max_Number_Dof_Domain), temp2(Max_Number_Dof_Domain)
DO      d=1,Number_Of_Domain
        ! Gather x in temp1 (degrees of freedom of domain d only)
        CALL    gather(x,temp1,d)
        CALL    matrix_by_vector(matrix,temp1,temp2,d)
        ! Scatter temp2 in y (degrees of freedom of domain d only)
        CALL    scatter(temp2,y,d)
END DO

```

Este código trabaja bien sobre un ordenador serial. Partiendo de la hipótesis de que las rutinas dentro del lazo se compilan apropiadamente, la simple adición de la directiva `C$DIRFORCE_PARALLEL` no generará un código correcto, (el compilador no proporciona ninguna información). En realidad, las variables temporales *temp1* y *temp2* serían accedidas para lectura-escritura, por varios procesadores al mismo tiempo conduciendo a resultados impredecibles. Note que el proceso de ensamblaje debe realizarse secuencialmente, porque si no varios procesadores intentarían escribir en *y* simultáneamente. La solución es:

```

...
REAL*8    x(Max_Number_Dof),y(Max_Number_Dof)
REAL*8    matrix(Max_Number_Dof_Domain,2*Max_Bandwith-1,Max_Domain)
REAL*8    temp1(Max_Number_Dof_Domain,Max_Domain)
REAL*8    temp2(Max_Number_Dof_Domain,Max_Domain)
! The following loop is forced to execute in parallel
C$DIR FORCE_PARALLEL
DO      d=1,Number_Of_Domain
        ! Gather x in temp1(1,d) (degrees of freedom of domain d only)
        CALL    gather(x,temp1(1,d),d)
        CALL    matrix_by_vector(matrix,temp1(1,d),temp2(1,d),d)
END DO
! The following loop executes serially (no directive)

```

```

DO    d=1,Number_Of_Domain
      ! Scatter temp2(1,d) in y (degrees of freedom of domain d only)
      CALL    scatter(temp2(1,d),y,d)
END DO

```

Este pequeño ejemplo demuestra que tipo de manipulaciones deben hacerse para generar el código apropiado. Estas modificaciones, generalmente, afectan partes limitadas del código.

5.4.2.4 Paralelismo de grano fino

El programa estandar (serial), se modificó como fue descrito en la sección anterior. Las modificaciones fueron sencillas y no muy numerosas. Básicamente, se adicionaron directivas de compilación de tal forma que, las deformaciones, las tensiones, las fuerzas internas, la matriz de rigidez y los productos matriz-vector, se calcularan en paralelo. Debido a la forma como estaban almacenadas las variables, no se requirió de ninguna modificación adicional.

Estos cambios, conducen a un tipo de paralelismo que aquí se define como paralelismo de grano fino, ya que los lazos paralelizados de este modo, fueron los lazos sobre el número de elementos (comparado con lazos sobre los subdominios). Cada unidad de proceso, realizó cálculos sobre un número pequeño de elementos. En esta versión, la matriz de rigidez se ensambló, por lo tanto su cálculo no se paralelizó completamente (no se utilizó ninguna técnica de coloreado) pero en este caso se encontró que era bastante eficiente.

5.4.2.5 Método del complemento de Schur: paralelismo de grano grueso

Se implementó el método del complemento de Schur, descrito en 5.4.2.1, y se llevaron a cabo modificaciones similares a las descritas en 5.4.2.3. Se incluyeron directivas de compilación en el cálculo de las deformaciones, las tensiones, fuerzas internas, las matrices de los subdominios (submatrices), su descomposición LU , los productos matriz-vector de tal forma que los problemas locales del subdominio se calcularan o solucionaran en paralelo. Esto conduce a un paralelismo de grano grueso, ya que los lazos que se paralelizan están definidos sobre el número de subdominios y cada unidad de proceso actúa sobre los datos de un subdominio completo.

Nótese que las submatrices se almacenan independientemente, entonces su cálculo es completamente paralelo. Los productos matriz-vector se llevan a cabo como se describió en 5.4.2.2, así cada producto local se calcula en paralelo. Sólo la etapa de ensamblaje se realiza secuencialmente. Los problemas lineales definidos en cada subdominio se solucionan también en paralelo.

5.5 RESULTADOS COMPARATIVOS

En esta sección se presentan los resultados obtenidos empleando las diferentes implementaciones presentadas.

5.5.1 Los diferentes ordenadores y sus respectivas implementaciones

Se realizaron una gran cantidad de pruebas sobre diferentes ordenadores, sobre algunos de ellos se analizaron varios tipos de programas que presentaban diferentes estrategias de implementación. Los ordenadores donde se realizaron las pruebas fueron los siguientes:

- Ordenador escalar Silicon Graphics Indigo, 50MHz, con 48MBytes de RAM y 100MBytes de swap. Este ordenador se empleó para correr la versiones serial descrita en 5.2 y una versión serial (no paralelizada) del método de subestructuras.

- Ordenador Connection Machine CM-200 configurado con 4096 procesadores, con 1MBit de memoria local y una velocidad de reloj de 10MHz. El programa que se analizó sobre este ordenador, fue descrito en 5.4.1.

- Ordenador Connection Machine CM-5 con 128 nodos y configurado con una partición de 64-nodos y dos de 32-nodos. Debido a la alta carga de la máquina y las dificultades para trabajar en un ambiente de poca carga, sólo se hizo uso de la partición de 32-nodos para correr el programa. Este programa coincide con el que se utilizó en la CM-200 (excepto los cambios debidos a la recompilación).

- Ordenador CONVEX C3480 con 8 procesadores vectoriales, 1GByte de RAM compartida y 1GByte de swap. Los programas seriales escritos sobre la Silicon Graphics se corrieron directamente sobre éste para analizar las ventajas de la vectorización. Se probaron también sus versiones paralelas descritas en 5.4.2.4 y 5.4.2.5.

5.5.2 Descripción del ejemplo de prueba

El siguiente ejemplo se resolvió con todas las implementaciones:

Compactación uniaxial de un cilindro de radio $R = 2\text{cm}$ y altura $h = 1\text{cm}$. El dominio se discretizó con elementos cuadrilateros de revolución de cuatro nodos por elemento. El segmento superior se desplazó 0.6cm para realizar la compactación, mientras el resto de segmentos permanecían fijos. El proceso se llevó a cabo en 10 incrementos (figura 5.11).

Las tolerancias empleadas fueron las siguientes: $\epsilon_P = 10^{-3}$ (problema global), $\epsilon_S = 10^{-5}$ resolución del sistema lineal. La dimension de Krylov fue 80 para todos los experimentos.

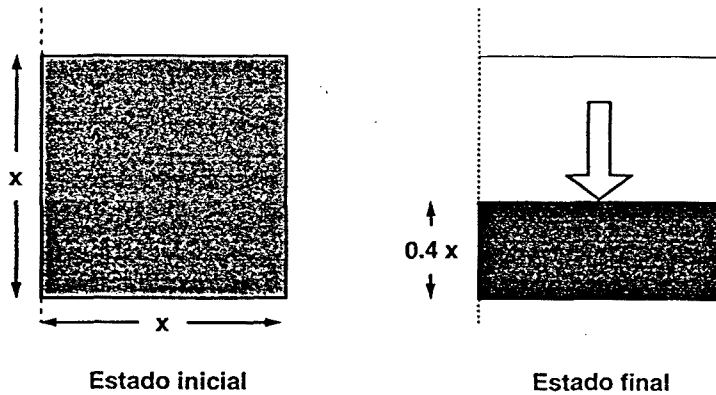


Figura 5.11 Ejemplo de prueba

Se probaron un total de 9 versiones del programa. Se describen a continuación los experimentos numéricos:

- sobre la Silicon Graphics, el dominio se discretizó en mallas de: 10×10 , 20×20 , 30×30 , 40×40 , 50×50 , 60×60 y 65×65 elementos para la versión estandar y hasta 80×80 para la versión con subdominios. Esta última versión, se corrió empleando 8 subdominios para comparar con los resultados obtenidos en el CONVEX C3480.

- sobre la CM-200 y CM-5 el dominio se discretizó desde 10×10 hasta 110×110 elementos

- sobre el CONVEX C3480, puesto que la versión disponible de esta máquina tiene 8 procesadores, se utilizaron 8 subdominios en la versión con subdominios. La mallas que se emplearon fueron: 20×20 , 40×40 , 60×60 elementos sobre 1(serial), 2, 4 y 8 procesadores con la versión con subdominios. La versión estandar se ejecutó para el mismo número de elementos sobre 1(serial) y 8 procesadores.

5.5.3 Resultados sobre un ordenador secuencial: Silicon Graphics Indigo

Como se dijo previamente, se ejecutaron dos versiones sobre el ordenador serial: una implementando el método de subestructuras y otra implementando el método convencional. Los tiempos que se presentan son tiempos CPU dados por el comando *time* de UNIX.

Las mallas de 10×10 , 20×20 , 30×30 , 40×40 , 50×50 , 60×60 , 65×65 elementos se han ejecutado con el programa estandar. Además la versión con subestructuras se ha ejecutado también con 70×70 , 75×75 y 80×80 elementos. El problema límite de la versión estandar de 65×65 elementos es debido a la limitación de memoria, el ancho de banda crece rápidamente con la

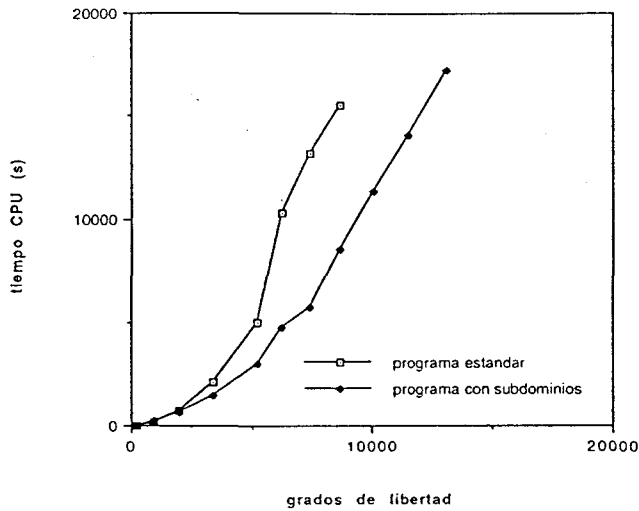


Figura 5.12 Tiempos totales sobre SGI (serial)

discretización. En la versión con subdominios, los anchos de banda son menores que el ancho de banda global y esto permite que se puedan ejecutar problemas mas grandes. Este es un punto interesante de esta implementación. La figura 5.12, presenta los tiempos CPU de ambas versiones. Como se puede ver, ambos programas presentan el mismo comportamiento hasta 2000 grados de libertad, y a partir de aquí la versión con subdominios es claramente más rápida que su contraparte estandar. Esto puede ser debido parcialmente al efecto de la memoria cache (la máquina tiene una memoria cache principal de 8KByte y 1MByte de memoria cache secundaria) el algoritmo de subdominios incrementa la localidad de los datos y también debido a la reducción en el número de instrucciones del algoritmo con subdominios.

5.5.4 Resultados sobre ordenadores SIMD: CM-200 y CM-5

La versión que se desarrolló para la CM-200 in CM-Fortran, se ha ejecutado también sobre la CM-5 con una partición de 32-nodos. Sin embargo las ejecuciones sobre la CM-200 se han hecho en modo uni-usuario, mientras que en la CM-5 se han hecho en modo compartido.

Los programas se han ejecutado con varias rutinas de comunicación y los tiempos que se publican aquí son los mejores obtenidos. Una buena elección de estas rutinas puede conducir a ganancias importantes [98]. En general, se utilizó el método fastgraph para problemas grandes tanto en las rutinas de ensamblaje como en las de expansión.

La figura 5.13, presenta los tiempos de ejecución para la CM-200 y la CM-5. Se presentan los tiempos totales y los tiempos en que la CM estuvo ocupada. Sobre la CM-5, los programas corrieron en condiciones de alta carga. Debido a esto el tiempo total es aproximadamente 30% más alto que el tiempo

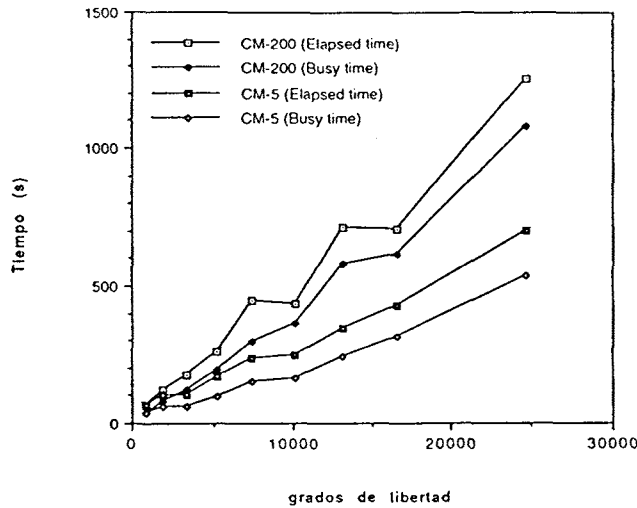


Figura 5.13 Tiempos sobre la CM-200 y CM-5

ocupado por la CM-5. Nótese que en la CM-200, este factor es aproximadamente 15% para problemas grandes, esto debe tenerse en cuenta cuando se leen los resultados de tiempo obtenidos.

5.5.5 Resultados sobre un ordenador de grano grueso: CONVEX C3480

Sobre este ordenador se probaron dos programas diferentes (el programa estandar y la versión con el método del complemento de Schur) cada uno con dos versiones, una vectorizada y otra paralelizada.

En caso de la estrategia de subdominios, las mallas se dividieron en 8 subdominios de igual tamaño y las versiones paralelas de este programa se ejecutaron con 2, 4 y 8 procesadores activos. Se consideraron las siguientes mallas: 20×20 , 40×40 y 60×60 elementos.

Versión estandar

Esta versión se ejecutó serialmente sobre 4 y 8 procesadores. La figura 5.14, presenta los resultados obtenidos para las mallas consideradas.

Como se puede ver el *speedup* que se obtuvo fue bastante alto, entre 3.5 y 4, para 4 procesadores, pero muy pobre, entre 3.7 y 5, para 8 procesadores. Una hipótesis para explicarlo pueden ser los conflictos de memoria resultantes de la arquitectura del C3480. En realidad, los procesadores se agrupan en 4 grupos de procesadores, cada par comparte un bus de comunicación a memoria. Así, los conflictos pueden ocurrir cuando se usan los procesadores de una par. Cuando se usan 4 procesadores, no pueden ocurrir conflictos, ya que sólo se usa uno de cada par. Por el contrario cuando se usan los 8 procesadores, pueden

incrementarse los conflictos en los accesos a memoria, causando unas bajas eficiencias, especialmente para mallas grandes.

Versión con subdominios

Esta versión se ejecutó serialmente y sobre 2, 4, 8 procesadores. Los resultados se presentan en la figura 5.15 para las tres mallas consideradas.

Como se puede ver, la ejecución del programa con 2 procesadores alcanzó un *speedup* cercano a 2, con 4 procesadores, este valor fue de 3.5 a 4 y con 8 procesadores este valor fue cercano a 6 y alrededor de 5 para la malla de 20x20. Este último resultado se debe a que para mallas pequeñas, cada procesador no recibe la suficiente cantidad de trabajo como para justificar realmente la paralelización, llevando a una baja eficiencia.

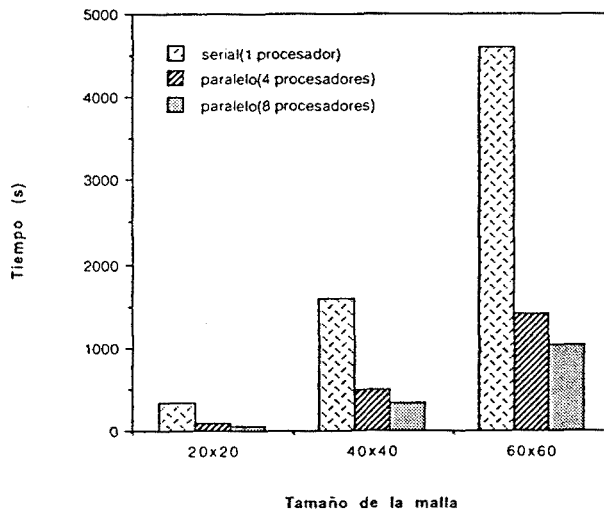


Figura 5.14 Tiempos del programa estandar sobre el C3480

Comparación de tiempos global

Para comparar los diferentes resultados se procede a comparar los mejores tiempos obtenidos en cada máquina. La figura 5.16, presenta los mejores resultados obtenidos para 3 mallas: 20×20 , 40×40 , 60×60 .

Los ordenadores SIMD (CM-200 y CM-5) son en general, los más rápidos para este tipo de problema. Esta arquitectura parece particularmente eficiente para problemas bastante homogéneos, como el que se estudió aquí. El CONVEX C3480 viene después y por último el ordenador serial (Silicon Graphics) como se esperaba.

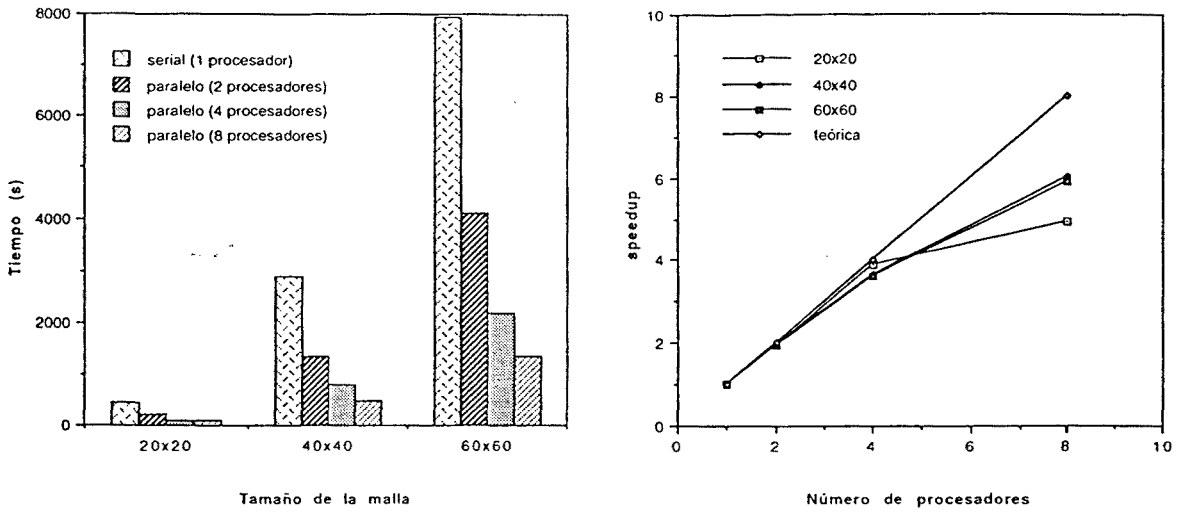


Figura 5.15 Tiempos y speedups del programa con subdominios en el C3480

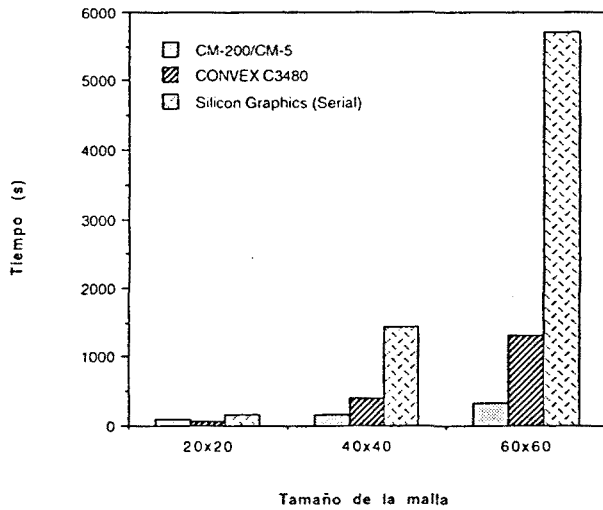


Figura 5.16 Mejores tiempos en cada ordenador

5.6 CONCLUSIONES

En este estudio se han analizado los comportamientos de diferentes tipos de ordenadores aplicados a la computación paralela. Para cada ordenador se han presentado y comentado resultados tales como el *speedup*, la eficiencia, la velocidad, etc. Sin embargo, tales comparaciones, aunque interesantes fallan en el intento de transmitir otros puntos interesantes que se deben tener en cuenta cuando se trabaja con ordenadores paralelos: facilidad de implementación o de uso de la máquina paralela, costo de transcribir una aplicación sobre una máquina particular, etc. Aunque sea muy subjetiva, la tabla 1, trata de dar una idea cualitativa de la experiencia que se adquirió aquí sobre cada uno de estos temas.

	SIMD CM-200/CM-5	MIMD(C3480) grano fino	MIMD(C3480) grano grueso
facilidad	baja	alta	media
eficiencia	alta	media-baja	media-alta
velocidad absoluta	alta	media	media-alta

Tabla 9.1 Evaluación de resultados cualitativos

La facilidad de paralelización, se refiere a la facilidad de portar un código serial ya existente a una arquitectura paralela particular. Respecto a la arquitectura MIMD, la paralelización de grano fino, se realiza a través de las directivas de compilación (pocas modificaciones del código original) mientras que la paralelización de grano grueso se refiere a la implementación del complemento de Schur.

Respecto a la arquitectura SIMD, se concluye que es bastante eficiente para realizar cálculos muy homogéneos, como el problema que se planteó aquí. En este caso fue posible elegir una estructura de datos que se adaptó bastante bien a la arquitectura de la máquina, lo que permitió obtener un código eficiente. Sin embargo la adaptación del código serial a este ordenador, requirió casi de una completa reescritura del mismo.

Sobre la arquitectura MIMD, se han estudiado dos implementaciones:

- la versión de grano fino, la cual básicamente hace uso de la capacidad que tiene el compilador para paralelizar lazos, ésta requiere poco esfuerzo por parte del usuario. El resultado es una velocidad aceptable, pero una eficiencia que es más bien baja.

- la versión de grano grueso requiere más esfuerzo en la adaptación del código, pero los resultados de eficiencia son mejores. Sin embargo los tiempos de ejecución son del mismo orden que los obtenidos con la implementación de grano fino, esto cuestiona el mérito de esta versión particular.

CONCLUSIONES

CONCLUSIONES SOBRE EL TRABAJO DE INVESTIGACIÓN

Las conclusiones que resultan de la realización de esta tesis, se pueden resumir en los siguientes puntos:

1) Se eligió un modelo constitutivo que requiere el conocimiento de las propiedades elásticas y la resistencia máxima (a compresión) del material en verde y de la calibración de dos curvas que dependen de la densidad. Debido a que el número de parámetros que se requiere es reducido, el modelo numérico resultante es fácil de calibrar y por lo tanto atractivo como herramienta de diseño para la industria. Pero por otro lado fenómenos como la resistencia a la tracción, la dilatación, la difusión, efecto de la forma y tamaño de las partículas ,etc, no se tienen en cuenta directamente desde un punto de vista microestructural, sino a través de las propiedades mecánicas del conjunto.

2) El modelo constitutivo se formuló dentro de la teoría de las grandes deformaciones plásticas. Este tipo de formulación permite: a) obtener una respuesta precisa de los grandes movimientos del material dentro del molde, b) simular la recuperación elástica (*spring back*) que ocurre al final de la etapa de extracción.

3) Respecto a la ecuación constitutiva se implementaron dos modelos: un modelo hiperelástico y uno hipoelástico. Al comparar los resultados numéricos de ambos, no se encontró ninguna diferencia. Esto se debe a que las deformaciones elásticas son muy pequeñas durante todo el proceso, sólo al final aparece un crecimiento apreciable de éstas, pero bastante baja comparada con las deformaciones plásticas.

4) El uso de la regla de flujo no asociada genera un crecimiento indefinido de la deformación plástica volumétrica mientras que con una regla de flujo asociada ésta tiende a un valor límite, que se asemeja más al comportamiento durante la compactación de un polvo metálico. De las dos reglas de flujo antes mencionadas se desprende un modelo cerrado, del cual se puede obtener una expresión exacta del tensor constitutivo tangente consistente que permite una convergencia cuadrática del algoritmo global. Desde el punto de vista

computacional el modelo asociado es un poco más costoso ya que el corrector plástico se obtiene como solución de un polinomio de cuarto orden.

5) Para resolver el problema de contacto y fricción entre pieza y molde se estudiaron tres diferentes técnicas numéricas. El método de *penalización* es atractivo tanto por el concepto como por su interpretación física. Se puede implementar preservando la convergencia cuadrática del algoritmo global, pero conduce a serios problemas numéricos debidos al mal condicionamiento de la matriz de rigidez del sistema. La elección de la magnitud de los penalizadores no es una tarea fácil, que se dificulta aún más con el problema de compactación debido a que los términos de la matriz pueden cambiar en uno o dos ordenes de magnitud durante el proceso de compactación. La segunda técnica se formula a partir de la teoría de los *multiplicadores de Lagrange*, se elimina así el problema del mal condicionamiento. El método se implementó sin incrementar el número de incognitas del problema. Las dificultades que se presentan con este método son debidas a las oscilaciones de algunos puntos de la frontera para los cuales es difícil establecer el valor correcto de la fuerza tangencial. La tercera técnica corresponde al método del *Lagrangiano Aumentado* que elimina el problema de las oscilaciones y que permite que el valor de los penalizadores se pueda elegir más fácilmente, ya que un valor grande de éstos no es una condición necesaria para la convergencia. Aquí se ha realizado una implementación de bucles anidados, el bucle exterior controla la actualización de los multiplicadores de Lagrange y el bucle interior corresponde a la solución del problema no lineal. Por esta razón el costo computacional aumenta ya que en general se soluciona un problema no lineal por cada actualización de los multiplicadores de Lagrange. Una correcta elección de los penalizadores mantiene el equilibrio entre el mal condicionamiento de la matriz y el costo computacional. Como conclusión de esta parte se puede decir que el método de penalización genera muchas dificultades en el momento de elegir los penalizadores, la técnica de multiplicadores de Lagrange es bastante sencilla pero pueden presentarse oscilaciones en la respuesta y la técnica del Lagrangiano Aumentado es más robusta pero aumenta el costo computacional y no se elimina completamente el problema de la elección de los penalizadores.

6) Respecto de la paralelización, se analizaron dos tipos de arquitecturas paralelas: SIMD y MIMD (con memoria compartida). Como se muestra en esta tesis, un programa de elementos finitos se adapta de manera natural a una arquitectura del tipo SIMD, ya que se puede asociar cada elemento de la malla de elementos finitos con un procesador de la red (procesador virtual). Debido a la memoria local y los altos costos de comunicación, se debe realizar la asignación de tal forma que se minimice la distancia entre los procesadores que contienen elementos vecinos. En general, un problema de elementos finitos se adapta adecuadamente sobre este tipo de ordenadores en la medida en que es homogéneo. Problemas no homogéneos con diferentes tipos de elementos y materiales disminuyen notablemente la efectividad de estas arquitecturas. La

adaptación de un código serial a este tipo de ordenadores requiere la reescritura de casi todo el programa. Esto conduce a un tipo de paralelismo no portable. En el ejemplo que se estudio en esta tesis se utilizó un esquema implícito, lo que hace necesario un método de resolución del sistema de ecuaciones asociado. En este sentido, debido a la memoria local y el costo de comunicación, los métodos directos son muy caros, entonces debe recurrirse a los métodos iterativos. Esta restricción puede convertirse en un serio inconveniente para la utilización de estos ordenadores con ciertos problemas donde los métodos iterativos tienen problemas de convergencia.

7) En los ordenadores del tipo MIMD (con memoria compartida) se pueden realizar diferentes tipos de implementación. En esta tesis se analizaron dos de estas posibles implementaciones. La primera, llamada paralelización de grano fino, consiste en utilizar directamente las directivas de paralelización ofrecidas por el sistema. En este caso el trabajo consiste en la reordenación de los bucles y la inserción de las directivas de compilación adecuadas. Este tipo de paralelismo, aunque sencillo no consigue explotar completamente el paralelismo del ordenador. La segunda corresponde al método del complemento de Schur (o de las subestructuras). En este caso se requiere la escritura de unas rutinas que resuelvan el problema de la interfaz, con la ventaja de que en cada subdominio se puede utilizar el código ya existente, siendo este un paralelismo más portable y más eficiente que el anterior.

APORTACIONES AL ESTADO DEL CONOCIMIENTO

Como aportaciones realizadas en esta tesis se destacan las siguientes:

a) La formulación de un modelo numérico inmerso en la teoría de las grandes deformaciones plásticas, con capacidad para simular los siguientes fenómenos que se producen durante la fase de compactación:

- el movimiento de las partículas de material de unas zonas a otras en el interior del molde
- la evolución y distribución de la densidad dentro del compacto
- la evolución del estado tensional que se produce durante la compactación y el estado de tensiones residuales que tiene el producto al final del mismo
- el rozamiento del pulvimaterial con las paredes del molde y sus efectos sobre la densificación de las zonas de la frontera
- la fuerza de extracción requerida para expulsar el compacto del molde
- la recuperación elástica (*spring-back*) que se produce después de la etapa de extracción

b) La formulación e implementación del modelo numérico con las siguientes características:

- cálculo del tensor constitutivo tangente consistente para las dos reglas de flujo utilizadas
 - formulación de la matriz tangente del problema global, que junto con el tensor constitutivo tangente consistente permiten una convergencia cuadrática del problema global
 - formulación de un algoritmo que permite simular el problema de contacto y fricción mediante la técnica de los multiplicadores de Lagrange
 - implementación de un algoritmo que permite simular el problema de contacto y fricción mediante la técnica de penalización y el cálculo de la contribución a la matriz tangente global incluyendo los términos de curvatura
 - implementación de un algoritmo que permite simular el problema de contacto y fricción mediante la técnica del Lagrangiano Aumentado
- c) Implementación del modelo numérico aplicando las técnicas del cálculo paralelo.
- definición de la estructura de datos e implementación del programa en un ordenador masivamente paralelo (Connection Machine)
 - implementación del programa en un ordenador de baja granularidad (CONVEX C3480) utilizando el método del complemento de Schur

LÍNEAS DE INVESTIGACIÓN FUTURA

Tomando como punto de partida el modelo numérico desarrollado aquí se pueden proponer una serie de modificaciones y extensiones que pueden generar nuevas líneas de investigación.

Desde el punto de vista del modelo constitutivo, cualquier extensión debe venir acompañada de una campaña de experimentos que permita verificarla. Entre estas extensiones se encuentran los fenómenos de dilatancia, un modelo de fallo por tracción, la adición de variables internas tensoriales que registren los efectos anisótropos como la forma y tamaño del grano y una generalización del modelo constitutivo que permita simular la compactación de materiales cerámicos y cermets (cerámicos y metálicos).

Con respecto al modelo numérico, un primer reto es la extensión del modelo a tres dimensiones, esto permitirá analizar piezas reales más complejas, que en la actualidad presentan serias dificultades para su fabricación. Respecto al modelo de contacto y fricción, se puede estudiar una implementación del tipo mixto que consiste en añadir los multiplicadores de Lagrange a las incógnitas del problema. La matriz tangente del problema global debe calcularse nuevamente, su obtención permite mantener una convergencia cuadrática disminuyendo así el costo computacional. Sin embargo este tipo de esquema aumenta el costo de

la memoria requerida, que en el caso de un problema 3D, se puede convertir en una limitante muy seria.

A partir del modelo numérico desarrollado aquí se pueden predecir las propiedades finales del material en verde, pero no las propiedades del compacto final, para esto, se requiere la adición de un modelo térmico que permita simular la etapa de sinterizado. Con el modelo actual, se puede también simular un proceso de compactación isostática en frío, por lo tanto otra posible generalización es simular un proceso de compactación isostática en caliente.

La extensión del modelo a tres dimensiones, pone a prueba el verdadero potencial de las técnicas de cálculo paralelo que ofrecen un aumento en la velocidad de cálculo y un aumento en la capacidad de memoria. Una línea alternativa en el uso de ordenadores paralelos es la utilización de una red heterogénea de ordenadores por medio de librerías de comunicación como PVM, que permiten distribuir el trabajo en los diferentes ordenadores.

BIBLIOGRAFÍA

- [1] Arunachalam, V.S. y Roman, O.V. (Editores) [1990] *Powder Metallurgy*. Aspect publishing, India.
- [2] Dixon, R.H.T. y Clayton, A. [1971] *Powder Metallurgy for Engineers*, Machinery Publishing Co Ltd. England.
- [3] Ichinose Noboru (Editor) [1987] *Introduction to Fine Ceramics*, John Wiley y Sons Ltd. England.
- [4] Peñafiel Vercher J. *Nuevas Tecnicas de Lubricación y Mezclado de Polvos en Pulvimetalurgia de Aceros*, Tesis de Doctorado, Departamento de Ingeniería Química y Metalurgia
- [5] Popper P. [1976] *Isostatic Pressing*, Heyden. England.
- [6] Sands R.L. y Shakespeare C.R. [1966] *Powder Metallurgy*, Newnes International Monographs on Mat. Science and Tech. England.
- [7] Svarovsky L. [1987] *Powder Testing Guide*, Elsevier Applied Science England.
- [8] Alberto Adrés Domènech *Apuntes de Clase, Depto. de resistencia de Materiales y Estructuras*, Universidad Politécnica de Cataluña, Barcelona
- [9] Hausner, Henry H. [1973] *Handbook of POWDER METALLURGY*, Chemical Publishing Co. Inc. US.
- [10] Bandstra, James y P. Otto, W. L. y Massa, Ted R. [1990] *Finite Element Simulation of Cold Die Compaction Leading to Sintered Part Dimension Control*, Advances in Powder Metallurgy Proc. 1990 Powder Metallurgy Conf. Exhib. (Metal Powder Industries Federation) pp 181-196
- [11] Brekelmans, W.A.M. Janssen, J.D. y Van De Ven, A.A.F. [1991] *An Eulerian Approach for Die Compaction Processes*, International Journal for Numerical Methods in Engineering, vol 31, 509-524
- [12] Ferguson B. Lynn y Krauss Tony M. [1990] *Modeling Ejection of Die Pressed Parts*, Advances in Powder Metallurgy Proc. 1990 Powder Metallurgy Conf. Exhib.(P1) pp 147-164

- [13] Gethin, D.T. Lewis, R.W. Tran, D.V. Ashoka, J.G. *Finite Element Simulation in Metal Powder Die Compaction*, Proceedings of the 1992 Powder Metallurgy World Congress P2 (92-V2) pp 11-27
- [14] Krauss, Tony M. Petrus, G.J. y Ferguson, B.L. [1991] *Die compaction Process Design Part 2: FEM Simulation*, Proceedings of the 1991 Powder Metallurgy Conf. Exhibition 91-V1, pp 111-122
- [15] Tomokazu Nakagawa y Masaki Sato [1992] *Simulation of Powder Densification in Die Compaction Process*, Proceedings of the 1992 Powder Metallurgy World Congress P2 (92-V2)
- [16] Oliver J. Oller S. y Cante J.C. [1992] *Numerical Simulation of Uniaxial Compaction Processes in Powder Materials*, Proceedings of the International Congress on Numerical Methods in Engineering and Applied Sciences, Chile, 16-20 Nov. pp 1277-1286
- [17] Riera, M.D. Miralles, A. Cugat, G. y Prado, J.M. [1993] *Modelización del Proceso de Compactación en Frío de Polvos Metálicos*, II Congreso de Métodos Numéricos en Ingeniería 1993, pp 1546-1555
- [18] Trasorras, J. Krauss, Tony M. y Ferguson, B.L. [1989] *Modeling of Powder Compaction Using the Finite Element Method*, Proceedings of the 1989 Powder Metallurgy Conf. Exhibition V1, pp 85-104
- [19] Weber, G.G. y Brown, S.B. [1990] *Simulation of the Compaction of Powder Components*, Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge
- [20] Ernst, E. y Barnekow D. *Pressure, Friction and Density During Axial Powder Compaction*,
- [21] Mosbah, P. Bouvard, D. Lanier, J. Quedraogo, E. y Stutz, P. *Rheological Characterization of Metal Powder Following Various Loading Paths*
- [22] Gethin, D.T. Lewis R.W. [1994] *Finite Element Modelling of Powder Compaction and its Experimental Validation*, Powder Metallurgy 94 World Congress, Paris, 4-10 June, pp 689-692
- [23] Cocoz, G. Bellet, M. Lécot, R. Ackermann, L. y Haggblad H. *Cold Compaction of Iron Powder: Experiments and Simulation*, Powder Metallurgy 94, World Congress, Paris, 4-10 June, pp 709-712
- [24] Turner, C.D. *A micromechanical model of multiaxial powder densification*, Powder Metallurgy 94, World Congress, Paris, 4-10 June, pp 713-716
- [25] Tamura, S. Mitsuno, T. Aizawa, T. *Powder Pressing Design Analysis by the Granular Model Simulation*, Powder Metallurgy 94, World Congress, Paris, 4-10 June, pp 753-756

- [26] Akisaya A.R. Cocks A.C.F. y Fleck N.A. *The Role of contact-contact Interaction in the Densification of Powders*, Powder Metallurgy 94, World Congress, Paris, 4-10 june, pp 757-760
- [27] Dorémus,P. Geindreau,C. Martin,A. Lécot,R. Dao,M. *High Pressure Tri-axial Apparatus for Investigating the Mechanical Behaviour of Metal Powders* Powder Metallurgy 94,World Congress, Paris, 4-10 june, pp 733-736
- [28] Bochini G.F. *Friction on Punch Faces and its Effects on the Green Density of P/M parts* Powder Metallurgy 94, World Congress, Paris, 4-10 june, pp 741-744
- [29] Bochini G.F. Fontanari,V. Molinari,A. *Friction and Boundary Layer Effects on the Apparent Compressibility of Iron Powders Determined on Thin Disks* Powder Metallurgy 94, World Congress, Paris, 4-10 june, pp 745-748
- [30] DiMaggio,F Sandler,I [1971] *Material Model for Granular Soils* Journal of the Engineering Mechanics Divison, Proceedings of ASCE 97,EM3 (1971) pp. 935-950
- [31] Shima,S. y Oyade,M. [1976] *Plasticity Theory for Porus Metals* INT. J. for Mechanical Science, Vol 18, pp. 285
- [32] Park,J.K. [1985] *Die compaction of powder: constitutive modelling and finite element calculations* Ph.D. Thesis, North Carolina State University
- [33] Kuhn, H.A. y Downey C.L. [1971] *Deformation Characteristics and Plasticity Theory of Sintered Powder Materials* International Journal of Powder Metallurgy 7(1), p. 15,1971
- [34] German R.M. *Powder metallurgy Science*, MPIF, Princeton
- [35] Abraham, R., J.E. Marsden and Ratiu, [1983], *Manifolds, Tensor Analysis and Applications*, Addison-Wesley, Mass.
- [36] Simo J.C. *Elastoplasticity and Viscoplasticity, Computational Aspects*, Stanford Univ. Dept. of Appl. Mech.
- [37] Simo J.C. *A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition:part 1. continuum formulation* Computer methods in applied mechanics and engineering 66 (1988), p. 199-219
- [38] Simo J.C. *A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition:part 2. computational aspects* Computer methods in applied mechanics and engineering 68 (1988), p. 1-31
- [39] T.A. Laursen y J.C. Simo [1993] *Algorithmic symmetrization of Coulomb frictional problems using augmented Lagrangians* Computer methods in applied mechanics and engineering, Vol. 68, pp 133-146

- [40] Jacob Lubliner [1990] *Plasticity Theory* Macmillan Publishing Company N.Y. 1990
- [41] Jacob Lubliner [1986] *Normality Rules in Large-Deformation Plasticity* Mechanics of Materials Vol. 5 pp 29-34
- [42] Kröner, E. [1960] *Arch Ration. Mech. Anal* 4, 273
- [43] Fox, N [1968] *Q. J. Mech. Appl. Math* 21, 67
- [44] Lee, E. H. [1950] Proc. 3rd Symp. Appl. Math. (Ann Arbor, 1949), McGraw-Hill, New York, p 213
- [45] Mandel J. [1972] *Plasticité Classique et Viscoplasticité* (CISM, Udine, 1971) Springer-Verlag, Vienna - New York.
- [46] Lee E.H. y Liu D.T. [1967] *J. Appl. Phys*, 38,19
- [47] Gurtin M.E. [1981] *An Introduction to Continuum Mechanics* Academic Press Inc. (1981)
- [48] Marsden J.E. y Hughes T.J.R. [1983] *Mathematical Foundations of Elasticity*, Prentice-Hall Englewood Cliffs, New Jersey
- [49] Duvaut G. *Mécanique des milieux continus* MASSON, Paris 1990
- [50] Sergio Oller [1988] *Un Modelo de "Daño Continuo" para Materiales Frictionales* Tesis Doctoral, Universidad Politécnica de Cataluña, mayo 1988
- [51] J. C. Simo y T. A. Laursen [1992] *An Augmented Lagrangian Treatment of Contact Problems Involving Friction*, Computers & Structures Vol 42, No 1, pp 97-116
- [52] J. Tinsley Oden [1986] *Qualitative Methods in Nonlinear Mechanics*, Prentice-Hall Englewood Cliffs, New Jersey
- [53] R. Glowinski [1982] *Numerical Methods for Nonlinear Variational Problems* Springer-Verlag
- [54] David E. Luenberger [1989] *Programación lineal y no lineal* Addison-Wesley Iberoamericana
- [55] Mokhatar S. Bazaraa, Hanif D. Sherali y C.M. Shetty *Nonlinear programming theory and Algorithms* New York, John Wiley & Sons Cop. 1993
- [56] Andreas H. y P. Alart Y E. [1993] *A Frictional Contact Element for Strongly Curved Contact Problems* submitted to International Journal for Numerical Methods in Engineering
- [57] P. Wriggers y J. C. Simo [1985] *A Note on Tangent Stiffness for Fully Non-linear Contact Problems* Communications in Applied Numerical Methods, Vol. 1, pp 199-203

- [58] Panayiotis Papadopoulos y R. L. Taylor [1992] *A Mixed Formulation for the Finite Element Solution of Contact Problems* Computer Methods in Applied Mechanics and Engineering, V. 94, pp 373-389
- [59] P. Wriggers, W. Wagner y E. Stein [1987] *Algorithms for Non-linear Contact Constraints with Application to Stability Problems of rods and Shells* Computational Mechanics, V. 2, pp 215-230
- [60] Bahram Nour-Omid y P. Wriggers [1987] *A note on the Optimum Choice for Penalty Parameters* Communications in Applied Numerical Methods, Vol 3, pp 581-585
- [61] Jiann-Wen Ju, R. L. Taylor y Louis Y. Cheng *A Consistent Finite Element Formulation of Nonlinear Frictional Contact Problems* Proceedings of the Int. Conf. on numerical Mech. in Engineering, theory and Applications, Swansea, 6-10 July 87. Ed. G.N. Pande & J. Middleton Martinus, Nijhoff Publishers
- [62] P. Wriggers, J. C. Simo y R. L. Taylor [1985] *Penalty and Augmented Lagrangian Formulations for Contact Problems* Proceedings of NUMETA'85 Conference/Swansea/7-11 January
- [63] E. Stein, P. Wriggers & T. van Van *Models of Friction, Finite-Element-Implementation and Application to Large Deformation Impact-Contact-Problems* Proceedings of 2nd International Conf. on Computational Plasticity, Part II, sep. 1989. Edited by Owen, Hinton, Oñate. Pineridge-Press UK.
- [64] T. Rodic y D.R.J. Owen *A Plasticity Theory of friction and Joint Elements* Proceedings of 2nd International Conf. on Computational Plasticity, Part II, sep. 1989. Edited by Owen, Hinton, Oñate. Pineridge-Press UK.
- [65] A. Curnier [1984] *A Theory of Friction* Int. J. Solids Structures Vol. 20, No 7, pp 637-647
- [66] A.G.K. Jinka y R.W. Lewis [1994], *Thermomechanical analysis of metal powder hot isostatic pressing process by the Finite Element Method* Powder Metallurgy 94, World Congress, Paris, 4-10 june, pp 801-804
- [67] D. Abondance, R. Baccino, F. Bernier, F. Moret, J.M. de Monicault, D. Guichard *Numerical Modelling of near-net-shape HIPing of TA6V powder* Powder Metallurgy 94, World Congress, Paris, 4-10 june, pp 797-800
- [68] M. Abouaf [1985], *Modélisation de la compactacion de poudres métalliques frittées* THESE présentée à L'Université Scientifique et Médicale et L'Institut National Polytechnique de Grenoble
- [69] S. Oller [1989], *Nuevos Materiales Estructurales: Cerámicos en Ingeniería* publicación interna: Centro Internacional de Metodos Numericos en Ingeniería

- [70] Hughes, T.J.R. [1987], *The Finite Element Method*, Prentice-Hall, Englewood-Cliffs, New Jersey
- [71] J.C. Simo y M. Ortiz [1985], *A Unified Approach to Finite Deformation Elastoplastic Analysis Based on the Use of Hyperelastic Constitutive Equations* Comp. Methods. Appl. Mech. Engrg. Vol 48, pp. 101-118
- [72] P.G. Ciarlet [1990], *Introduction à l'analyse numérique matricielle et à l'optimisation* MASSON ed.
- [73] E. Ernst y D. Barnekow [1994] *Pressure, Friction and Density During Axial Powder Compaction* Powder Metallurgy 94, World Congress, Paris, 4-10 June, pp 673-676
- [74] P. Mosbah, D. Bouvard, J. Lanier, E. Quedraogo y P. Stutz *Rheological Characterization of metal Powder Following Various Loading Paths* Powder Metallurgy 94, World Congress, Paris, 4-10 June, pp 677-680
- [75] O.C. Zienkiewicz *El Método de los Elementos Finitos* Editorial Reverté, s.a.
- [76] Roger A. Horn y Charles R. Johnson [1990] *Matrix Analysis* Cambridge University Press
- [77] Klaus-Jurgen Bathe, Ekkehard Ramm y Edward L. Wilson [1975] *Finite Element Formulations for Large Deformation Dynamic Analysis* International Journal for Numerical Methods in Engineering, Vol. 9, pp 353-386
- [78] Manfredo P. do Carmo *Differential geometry of Curves and Surfaces* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1976
- [79] Q.V. Dinh, R. Glowinski, B. Mantel, J. Periaux, P. Perrier *Sub-domain solutions of nonlinear problems in fluid dynamics of parallel processors*, Computing Methods in Applied Sciences and Engineering , V. R. Glowinski J.L. Lions (ed) North-Holland publishing Company
- [80] Kai Hwang y Fayé A. Briggs [1988] *Arquitectura de Computadoras y Procesamiento Paralelo* McGraw-Hill
- [81] Charbel Farhat [1990] *Which Parallel Finite Element Algorithm for Which Architecture and Which Problem ?* Eng. Comput., 1990, Vol 7, Sept.
- [82] Charbel Farhat [1989] *On the Mapping of Masively Parallel Processors Onto Finite Element Graphs* Computers & Structures Vol 32, No 2, pp 347-353
- [83] C. Farhat, N. Sobh y K.C. Park [1990] *Transient Finite Element Computations on 65536 Processors: the Connection Machine* International Journal for Numerical Methods in Engineering, Vol. 30, pp 27-55
- [84] S.L. Johnsson y K.K. Mathur [1990] *Data Structures and Algorithms for the Finite Element Method on a Data Parallel Supercomputer* International Journal for Numerical Methods in Engineering, Vol. 29, pp 881-908

- [85] T. Belytschko y Edward J. Plaskacz [1992] *SIMD Implementation of a non-linear Transient Shell Program With Partially Structured Meshes* International Journal for Numerical Methods in Engineering, Vol 33, pp 997-1026
- [86] T. Belytschko, Edward J. Plaskacz y J. M. Kennedy, *Finite Element Computations on SIMD Machines with Hybrid Communication Schemes*
- [87] Kapil K. Mathur [1990] *On the Use of Randomized Address Maps in Unstructured Three Dimensional Finite Element Simulations* Thinking Machines Corporation, Technical Reports TR-37 CS90-5
- [88] Zdenek Johan, Thomas J.R. Hughes, Kapil K. Mathur y S.L. Johnsson [1992] *A Data Parallel Finite Element Method for Computational Fluid Dynamics on the Connection Machine System* Comp. Methods. Appl. Mech. Engrg, Vol. 99, pp 113-134
- [89] T. Belytschko, L.P. Bindeman, H. Y. Chiang, E. J. Plaskacz, *Nonlinear Explicit Computations on Massively Parallel Computers*
- [90] Charbel Farhat, Loula Fezoui y Stéphane Lanteri [1993] *Two-dimensional viscous flow computations on the Connection Machine: Unstructured meshes, upwind schemes and massively parallel computations* Comp. Methods. Appl. Mech. Engrg, Vol. 102, pp 61-88
- [91] J. Stoer y R. Bulirsch [1980] *Introduction to Numerical Analysis* Springer-Verlag, N. Y. 1980
- [92] Thinking Machines Corporation. Scientific Software Library (CMSSL) for CM fortran. Version 3.0 June 1992, Cambridge, Massachusetts
- [93] O.C. Zienkiewicz, F.R.S. [1984] *Iterative Methods and Hierarchical Approaches; a Prospect for Future of the Finite Element Method* ASEA 100th Anniversary, Swaden, C/R/472/84, January
- [94] James Ortega [1988] *Introduction to parallel and Vector Solution of Linear System* Plenum Press. New York, 1988
- [95] Youcef Saad y Martin H. Schultz [1986] *GMRES: a Generalized Minimal RESidual Algorithm for solving nonsymmetric linear Systems* SIAM J. Sci. Stat. Comput. Vol 7, No 3, pp 856-869, July 1986
- [96] S. Lennart Johnsson y Kapil K. Mathur [1989] *Experiance with the Conjugate Gradient Method for Stress Analysis on a Data Parallel Supercomputer* International Journal for Numerical Methods in Engineering, Vol. 27, pp 523 -546
- [97] T. Belytschko, E. J. Plaskacz, James M. Kennedy y D. L. Greenwell [1990] *Finite Element Analysis on the Connection Machine* Comp. Methods. Appl. Mech. Engrg, Vol. 81, pp 229-254

- [98] J.C. Cante, Ilpide E., Oliver J., y S. Oller *Experiences on massive parallel computations in a Finite Element context*. Report of the International Center for Numerical Methods in Engineering. Barcelona, Spain. 1993.
- [99] A. Quarteroni [1992] *Mathematical Aspects of Domain Decomposition Methods* Proceedings of the 1rst European Congress of Mathematics, Paris.
- [100] A. Quarteroni [1991] *Domain Decomposition and Parallel Processing for the numerical Solution of Partial diferential equations*. In surveys on Mathematics for Industry, Springer-Verlag
- [101] T.N. Philips [1992] *Preconditioned iterative methods for elliptic problems on decomposed domains* International Journal Computer Math. Vol 44, pp 5-18
- [102] CONVEX Computer Corporation CONVEX Architecture and Optimization Training Course. Version 1.0 October 1992
- [103] CONVEX Computer Corporation CONVEX Fortran Optimization Guide. Fourth edition, November 1992
- [104] Thinking Machines Corporation. Programming in Fortran. Version 1.0, January 1991, Cambridge, Massachusetts
- [105] Y. Saad [1985] *Krylov Subspace Methods for Solving Unsymmetric Linear Systems* Mathematics of Computation Vol. 37, No 155, pp 105-126, July 1985
- [106] Y. Saad [1989] *Krylov Subspace Methods on Supercomputers* SIAM J. Sci. Stat. Comput. Vol 10, No 6, pp 1200-1232, Nov.
- [107] J.C. Cante, Ilpide E., Oliver J., y S. Oller *On the structure of finite element code for its implementation on a massively parallel SIMD computer* Report of the International Center for Numerical Methods in Engineering. Barcelona, Spain. 1994.
- [108] J.C. Cante, Joannas D., Oliver J., y S. Oller *Experiences on massive parallel computations in Finite Element context*. Report of the International Center for Numerical Methods in Engineering. Barcelona, Spain. 1993
- [109] S. Lennart Johnsson [1990] *Supercomputers: Past and Future* Thinking Machines Corporation, Technical Reports TR-56 DP90-1
- [110] Nahil Atef Sobh [1991] *Preconditioned Conjugate Gradient and Finite Element Methods for Massively data-parallel Architectures* Computer Physics Communications, Vol. 65, pp 253-267
- [111] C. Farhat, E.Pramono *Mapping, Solving and Visualizing Finite Elements Problems on the Connection Machine*, pp 95-107
- [112] E. Denning Dahl *Mapping and Compiled Communication on the Conection Machine System* Proceedings of the Fifth Distributed Memory Computing

- Conference, IEEE, DMCC 1990, Charleston, South Carolina, April 8-10, 1990
- [113] Gene H. Golub, Charles F. Van Loan [1989] *Matrix Computations*. Second edition, The Johns Hopkins University Press. 1989
- [114] Mathur, Kapil K. *Communication primitives for unstructured finite element simulations on data parallel architectures* Thinking Machines Corporation, Technical Reports TR-242
- [115] Hillis, W.D. *Data parallel Algorithms* Thinking Machines Corporation, Technical Reports TR-65 DP86-2

