Sevilla, Septiembre de 2013

ESCUELA SUPERIOR DE INGENIEROS DE LA UNIVERSIDAD DE SEVILLA

SIMULACIÓN NUMÉRICA DE LA CIRUGÍA ORTOGNÁTICA

PROYECTO FIN DE CARRERA

Presentado por: JUAN FRANCISCO MARTÍ MARTÍNEZ

Dirigido por: FRANCISCO JAVIER MARTÍNEZ REINA

Índice

Capítulo 1 – Introducción.

Capítulo 2 – Modelo de comportamiento hiperelástico para tejidos blandos.

Capítulo 3 – Modelo de elementos finitos.

Capítulo 4 – Simulaciones y resultados obtenidos; ajuste manual.

Capítulo 5 – Mejoras en el modelo.

Capítulo 6 – Simulación de la cirugía.

Capítulo 7 – Redes neuronales.

Capítulo 8 – Conclusiones.

Capítulo 1:

INTRODUCCIÓN

La simulación computacional de gran variedad de problemas tiene cada vez más importancia. El estudio teórico en muchos de estos casos es inabarcable, y el experimental suele ser muy costoso y de mucha dificultad de ejecución. Esto, unido a los grandes avances en computación (velocidad, coste, visualización gráfica…) que se han producido, hace del estudio computacional una herramienta de trabajo muy interesante.

1.1 Bioingeniería y Biomecánica.

La aplicación de los conocimientos de la ingeniería a problemas típicos de la medicina o la biología ha dado lugar a la Bioingeniería. Esta disciplina aplica técnicas e ideas típicas de la ingeniería a la biología humana. La parte de la Bioingeniería que se refiere a la medicina suele llamarse Ingeniería Biomédica.

El trabajo realizado en este proyecto se incluye dentro de lo anterior, más concretamente en una subdivisión denominada Biomecánica. Su definición según la Real Academia Española es:

"Ciencia que estudia la aplicación de las leyes de la mecánica a las estructuras y los órganos de los seres vivos."

Este trabajo se basa en un estudio mediante simulación computacional de un problema biomecánico. El empleo de programas de elementos finitos es una herramienta muy útil debido a la capacidad de abordar problemas de mucha complejidad, además de la gran aplicabilidad práctica que poseen.

1.2 Objetivo del proyecto.

La finalidad del proyecto es la simulación de una operación de cirugía ortognática en un modelo de elementos finitos de un paciente. Para ello es necesario obtener un modelo tridimensional de la cara en algún programa de cálculo de estructuras. En este caso el programa elegido ha sido ABAQUS. Este modelo 3D se crea a partir de la segmentación de un TAC de la cabeza del paciente. Una vez realizada la segmentación, hay que ajustar los parámetros de la ley de comportamiento por zonas.

Para validar el modelo se hará uso de resultados experimentales obtenidos de escaneados faciales tomados al paciente en diferentes posturas. Estas posiciones definen los estados de carga que se van a simular. Las propiedades por zonas de la piel se ajustan comparando los desplazamientos reales que se producen con los obtenidos de las simulaciones.

Lo que se intenta conseguir con la validación del modelo es que las deformaciones que aparezcan al simular la operación en el modelo de elementos finitos sean lo más parecidas posible a las que aparecen en la realidad. De esta forma se podría visualizar el resultado de la operación en estudio antes de llevarla a cabo.

Hasta ahora la planificación de este tipo de cirugía se ha realizado de manera algo rudimentaria, y a veces los resultados no son los esperados. El principal error que se

cometía es considerar los tejidos blandos de la cara como rígidos, sin considerar de forma realista las deformaciones que aparecen tras la intervención. En este trabajo se modelan los tejidos blandos como sólidos hiperelásticos, tratando así de mejorar el resultado de la planificación de operaciones de este tipo.

1.3 Resumen del trabajo realizado

El primer paso fue estudiar los fundamentos teóricos del comportamiento de materiales hiperelásticos. Esto sirve de ayuda para poder entender e interpretar los resultados. Dentro de este estudio, se prestó especial atención a la ley de comportamiento para materiales hiperelásticos desarrollada por Rubin y Bodner, ya que es la que se va a utilizar para caracterizar los tejidos blandos del modelo en todo el proyecto. Un resumen de todo esto se encuentra en el Capítulo 2: "Modelo de comportamiento hiperelástico para tejidos blandos".

Antes de empezar a trabajar, también fue necesario aprender a utilizar el programa SIMPLEWARE, que se utilizará para hacer la segmentación del TAC (Tomografía Axial Computerizada) y obtener un primer modelo de elementos finitos. Para ello fue de gran ayuda el manual realizado por José Manuel.

A partir de aquí se empezó el trabajo de ordenador. Una vez recibidos los archivos "Dicom" se segmentaron empleando el software SIMPLEWARE. Al acabar la segmentación se obtiene en primer modelo de elementos finitos. Después de una serie de modificaciones y considerando los casos de carga a estudiar, se obtiene el modelo final con el que empezar a correr simulaciones en ABAQUS. Todo este proceso se explica en el Capítulo 3: "Modelo de elementos finitos".

Antes de aplicar los métodos de ajuste es necesario conocer los desplazamientos reales que aparecen en los casos que se simulan. Se consiguen estos valores gracias a escaneados faciales realizados en el paciente.

Ahora se puede empezar a ajustar las propiedades de la ley de comportamiento del modelo. Para ello se han utilizado dos métodos distintos. El primero es un ajuste manual; se varían las propiedades fijándonos en los resultados previos y en los esperados. Tanto el cálculo de los desplazamientos reales (de los escaneados faciales) como este primer método de ajuste se recoge en el Capítulo 4: "Simulaciones y resultados obtenidos: ajuste manual".

El segundo método de ajustar las propiedades del material es mediante el uso de redes neuronales artificiales. Las entradas a la red son los desplazamientos que aparecen en el modelo, y las salidas las propiedades de los tejidos blandos. La red se

entrena con los resultados de las simulaciones previamente realizadas. Una vez entrenada, se simula introduciendo como entradas los valores reales de los desplazamientos. El resultado de esta simulación son las propiedades de la piel, que se introducirán en ABAQUS para comprobar la eficacia del ajuste. En el Capítulo 7: “Redes neuronales” se explica detalladamente cómo se ha realizado todo este proceso.

En el Capítulo 5: “Mejoras en el modelo” se comentan algunas modificaciones que se han realizado para mejorar el modelo, y las variaciones que estas suponen en los resultados.

Donde más problemas nos hemos encontrado ha sido a la hora de simular la cirugía. Se han llevado a cabo pruebas con distintas variaciones en el modelo, habiendo tenido hasta que crear un modelo diferente (mallando el sólido 3D con distintos elementos) para contrastar algunos resultados. En el Capítulo 6: “Simulación de una cirugía” se detallan todas las pruebas realizadas.

En el último capítulo (Capítulo 8: “Conclusiones”) se habla de cuáles son los resultados más importantes que se han obtenido en el proyecto y que lecciones podemos sacar de él.

Capítulo 2:

MODELO DE COMPORTAMIENTO HIPERELÁSTICO PARA TEJIDOS BLANDOS

Este capítulo comienza pretendiendo transmitir una idea general de la teoría de los medios continuos no lineal, así como la aplicación a la ley de comportamiento que se usará. La información sobre materiales no lineales y su implementación en el método de elementos finitos se ha obtenido principalmente del libro "*Nonlinear Solid Mechanics. A Continuum Approach for Engineering* (Gerhard A. Holzapfel)".

Posteriormente se realiza una búsqueda bibliográfica sobre la simulación de operaciones maxilofaciales, búsqueda de propiedades de tejidos blandos y modelos de comportamiento de los mismos. Destaca el artículo "Development and Validation of a Three-Dimensional Finite Element Model of the face" (Barbarino et al), ya que en él se recoge el proceso de elaboración de un modelo de elementos finitos de la cara. Además se usa la ley de comportamiento de Rubin-Bodner, que es la que se usa en este proyecto.

2.1 El problema elástico.

La representación del problema elástico sigue el siguiente esquema:

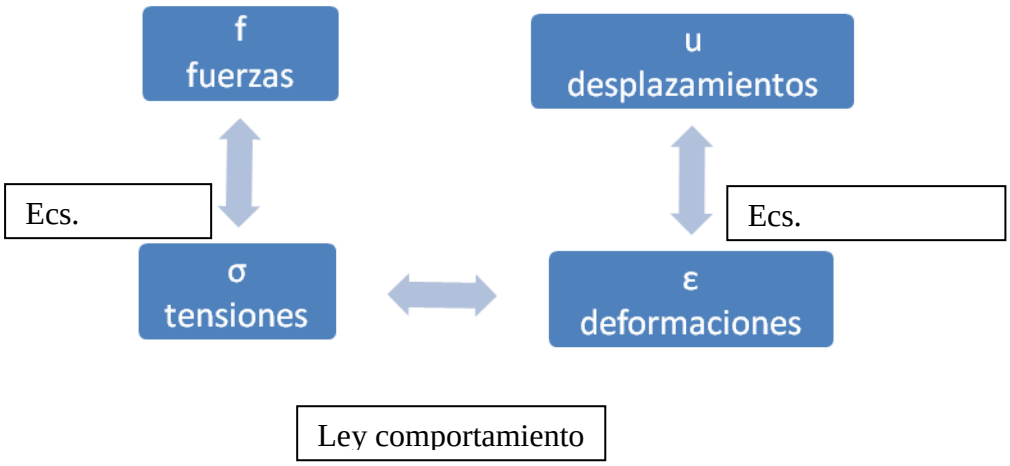

Figura 2.1: Problema elástico

El problema se puede plantear como la obtención de los desplazamientos originados por un sistema de fuerzas o cargas, o viceversa. Para ello se parte de las ecuaciones de equilibrio que deben cumplir el sistema de cargas con las tensiones internas que se originan en el interior del sólido deformable. Dichas tensiones son la causa de las deformaciones que se producen. La relación tensión-deformación se conoce como ley de comportamiento, y es principalmente función del material. El campo de deformaciones generado debe ser compatible con un campo de desplazamientos asociado, esta última imposición, junto con las condiciones de contorno, cierran el problema elástico.

Este tipo de problemas tan sólo tiene solución analítica para situaciones concretas de geometrías y cargas simples. De hecho, la geometría del problema aumenta la complejidad de tal manera que sólo exista una resolución a través de métodos numéricos como puede ser el empleo de elementos finitos (MEF). Por otra parte, la dificultad radica también en la elección de una ley de comportamiento adecuada para el material. Generalmente cualquier problema de elasticidad o de resistencia de materiales puede resolverse mediante procedimientos numéricos. Actualmente, la limitación para resolver este tipo de problemas radica en conocer la relación entre tensiones y deformaciones. Es por ello que gran parte de este proyecto tratará del ajuste de una ley de comportamiento.

La utilización del MEF para aplicaciones médicas precisa de un modelo matemático que reproduzca lo más fielmente posible el comportamiento del tejido biológico *in*

vivo. Para comenzar con el estudio de este tipo de tejidos, una visión de qué y cómo se componen, se expone a continuación.

2.2 Fisiología y propiedades mecánicas de los tejidos biológicos blandos.

Los tejidos blandos son los que forman parte de los seres vivos, como tendones, ligamentos, músculos, piel, etc. En mecánica de sólidos continuos las propiedades de un material dependen de su estructura y composición interna. Su composición principal es de fibras de colágeno y elastina, formando una red entrelazada, recubierta por una fina capa de proteoglicanos, llamada también "sustancia fundamental", su función es la de recubrir de manera adhesiva y lubricante.

En los vertebrados, el colágeno es la proteína más abundante, se trata del elemento estructural básico, proporcionando integridad mecánica y rigidez a los tejidos, así como de caracterizar un comportamiento anisótropo. En cuanto a la proteína de la elastina, posee una estructura más desorganizada y proporciona unas características de flexibilidad a los tejidos. En resumen, la presencia de un porcentaje mayor de colágeno en un tejido, daría a éste más rigidez, y la presencia de más elastina daría más flexibilidad.

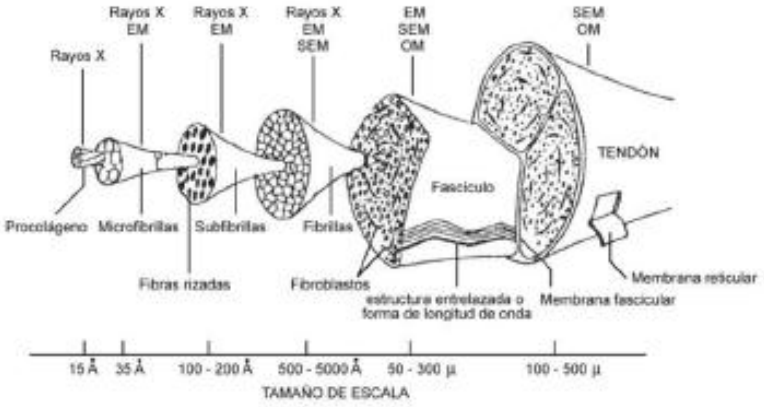

Figura 2.2: Estructura jerárquica del colágeno

En la figura anterior se observa la estructura jerárquica del colágeno formando un tendón. La molécula de colágeno está compuesta por tres cadenas helicoidales de aminoácidos con giro a izquierdas y a su vez enrolladas entre sí en una superhélice a derechas, se la conoce con el nombre de tropocolágeno. Cinco unidades de tropocolágeno constituyen una microfibrilla, que es el siguiente paso en la organización estructural. El siguiente paso es formado por una red tetragonal de cuatro microfibrillas para crear una subfibrilla y luego se forman la fibrillas. A este nivel interviene la sustancia fundamental, su asociación con agua une las fibrillas en fibras.

La agrupación en fascículos completa finalmente la estructura jerárquica del tendón mostrado.

El comportamiento mecánico particular de cada tejido depende también de la organización interna, de las fibras de colágeno y elastina, que presente. Así, un tejido con una organización preferencial unidireccional presentará mayor resistencia a la tracción en dicha dirección de alineación, mientras que una organización más irregular no daría tanta resistencia unidireccional pero proporcionaría una mejor resistencia biaxial.

Un tejido blando en general tiene un comportamiento anisótropo, a nivel microscópico se puede decir que es heterogéneo, ya que la estructura varía en cada punto. En cuanto a la no linealidad que presentan, se atribuye al patrón ondulado de las fibras. El alto contenido en agua que poseen les caracteriza también de incompresibles.

En la siguiente figura se expone la curva (tensión frente alargamiento) característica de un tejido blando:

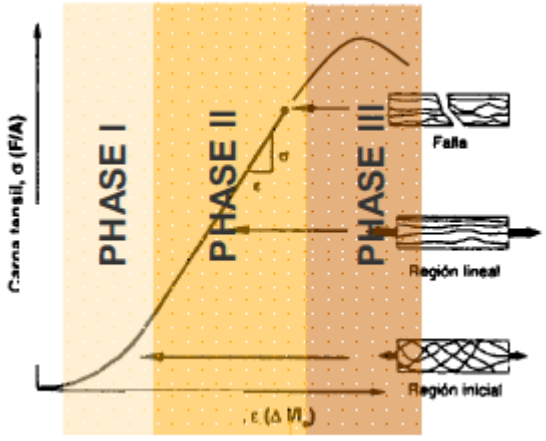

Figura 2.3: Representación de la curva tensión-deformación de un tejido blando

Como puede observarse la respuesta ante una carga de tracción presenta tres regiones de interés. La primera sería un tramo no lineal como consecuencia de deshacer el arrollamiento de las fibras, se observaría un gran alargamiento frente a pequeñas fuerzas. El segundo tramo lineal, es consecuencia de la rigidización de las fibras una vez estiradas, es necesario un aumento de la fuerza para producir un alargamiento. El último tramo también presenta no linealidad, consecuencia de la rotura de fibras.

Tal como se ha expuesto, el claro comportamiento no lineal que presentan estos tipos de materiales biológicos no es fácil de caracterizar. Junto a la complicación ya presente, se añade una complejidad aún mayor al depender sus propiedades de la edad, especie, topografía, velocidad de deformación, temperatura, etc. Contemplar alguno de estos factores en el modelo implicaría emplear términos de daño o viscoelásticos. En el presente proyecto el modelo de comportamiento escogido es capaz de representar las no linealidades (tensión-deformación) pero no puede reproducir los fenómenos viscoelásticos o de daño. Quedando abierto para futuros proyectos, la inclusión de un comportamiento distinto sobre el modelo obtenido que permita incorporar, por ejemplo, el daño en función de la edad, o cualquier otro modo que resulte de interés.

2.3 Introducción a la mecánica de sólidos no lineal.

La presencia en estos materiales de grandes deformaciones, comportamientos elásticos no lineales, anisotropía, etc. hace necesario un estudio del problema a través de la mecánica de sólidos no lineal.

El estudio bajo un punto de vista microscópico no será de nuestro interés. El enfoque de un medio continuo lo describirá un sistema macroscópico. La teoría de los medios continuos ha sido desarrollada independientemente de las teorías molecular o atómica, de manera que tiene en cuenta nuestras necesidades. Un sólido o cuerpo deformable β puede ser definido como una composición de partículas $P \epsilon \beta$. Decir que el concepto de partícula no tiene el significado de partícula atómica o punto masa, sino de una constituida por un número grande de moléculas, aunque suficientemente pequeño para ser considerado partícula.

2.3.1 Cinemática:

Para comenzar con esta ciencia, se definirán algunos matices sobre la notación que se empleará. Una variable vectorial se representará con una letra negrita. Si dicha variable está asociada a una situación material o indeformada, se escribirá con una letra mayúscula. Dejando las letras minúsculas para representar las variables de un estado deformado o espacial. Así, la posición de las partículas de un sólido en la situación material se denota con **X**, y en el caso espacial con **x**.

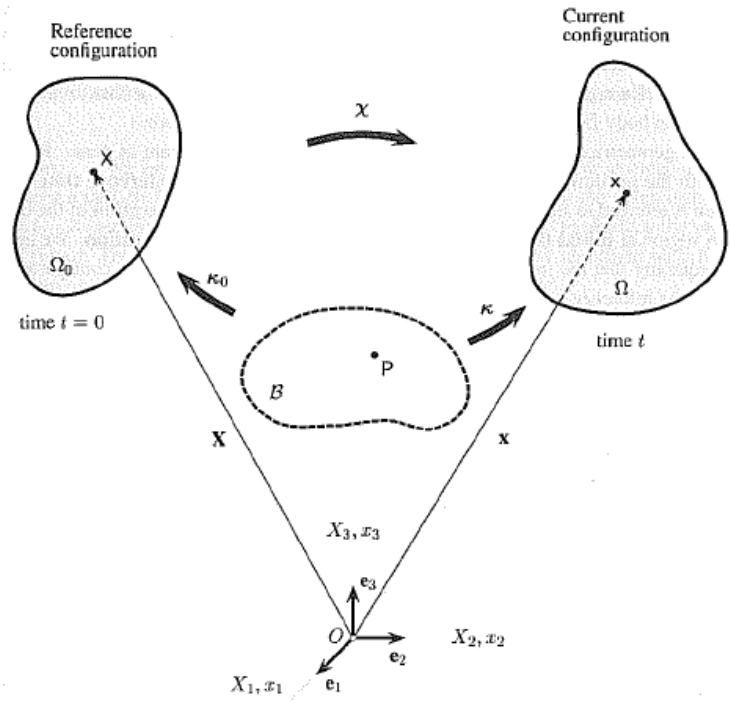

Figura 2.4: Configuraciones y movimiento de un cuerpo continuo.

Un cuerpo continuo que puede cambiar su forma se dice deformable. Todo punto del mismo en la situación Ω_o (estado inicial), tiene una relación biunívoca en la situación Ω (estado final). Se tiene la siguiente relación:

$$x = \kappa[\kappa_0^{-1}(X,t)] = \chi(X,t)$$

Donde $\chi(X,t)$ representa el campo vectorial que proporciona **x** de **X** en un tiempo dado. Se le conoce como movimiento (o deformada) del cuerpo β. Está describiendo el movimiento de las partículas (trayectoria) en el espacio. Se caracteriza por ser continua y diferenciable.

Las cantidades asociadas al movimiento del cuerpo como pueden ser: deformación, densidad o temperatura pueden describirse de acuerdo con la posición inicial o de referencia del cuerpo (descripción material o Lagrangiana) o con respecto a la posición actual del cuerpo (descripción espacial o Euleriana). Así el campo de desplazamiento puede expresarse como:

- $U(X,t) = x(x,t) - X$ (Forma Lagrangiana)
- $u(x,t) = x - X(x,t)$ (Forma Euleriana)

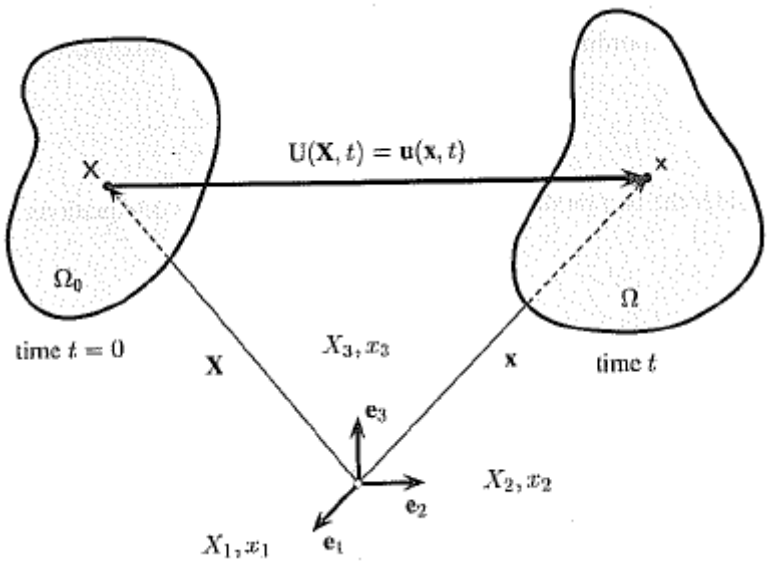

Figura 2.5: Campo de desplazamiento **U** de una partícula.

Ambas descripciones están relacionadas por la función χ del movimiento:

$$U(X,t) = U[\chi^{-1}(x,t),t] = u(x,t)$$

En mecánica de sólidos el movimiento y la deformación del cuerpo continuo es, en general, descrito en términos del campo de desplazamientos. Mientras que en mecánica de fluidos los principales campos son: el campo de velocidades y de aceleraciones. Las expresiones de velocidad y aceleración pueden obtenerse mediante la primera y segunda derivadas del movimiento respecto al tiempo, ya sea en descripción material o espacial.

A continuación vamos a realizar algunas matizaciones respecto a las derivadas espaciales y materiales. Como se ha dicho anteriormente, un campo material tiene como variables independientes la posición inicial **X** y el tiempo, y un campo espacial las variables independientes: posición final **x** y tiempo. Se define un campo material o espacial "suave" como una cantidad escalar, vectorial o tensorial que lo asocia con el movimiento χ. Considerando lo expuesto, comenzamos a definir las distintas derivadas:

- Derivada total o material de un campo material:

$$\dot{\mathcal{F}}(X,t) = \frac{\mathcal{D}\mathcal{F}(X,t)}{\mathcal{D}t} = \left(\frac{\partial \mathcal{F}(X,t)}{\partial t}\right)_X$$

- Gradiente material de un campo material:

$$Grad\mathcal{F}(X,t) = \frac{\partial \mathcal{F}(X,t)}{\partial X}$$

- Derivada y gradiente espacial de un campo espacial:

$$\dot{f}(x,t) = \frac{\partial f(x,t)}{\partial t} \qquad gradf(x,t) = \frac{\partial f(x,t)}{\partial x}$$

- Derivada material de un campo espacial:

$$\dot{f}(x,t) = \frac{\mathcal{D}f(x,t)}{\mathcal{D}t} = \left(\frac{\partial f[\chi(X,t),t]}{\partial t}\right)_{X=\chi^{-1}(x,t)}$$

Como aplicación de este último caso, considérese un campo espacial suave ϕ que asigna un escalar a cada punto **x** de t. Aplicando la regla de la cadena de la derivación se obtiene:

$$\dot{\phi} = \frac{\mathcal{D}\phi(x,t)}{\mathcal{D}t} = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \cdot v$$

2.3.2 Gradiente de deformación:

El siguiente concepto a introducir es el de deformación. Una deformación puede entenderse como un cambio de tamaño o forma. Considérese una curva en la situación indeformada: $X = \boldsymbol{\Gamma}(\xi) \subset \Omega o$ donde ξ representa una parametrización de la misma. Nótese que al estar asociada al estado inicial no es dependiente del tiempo. Durante cierto movimiento χ la curva se deforma, encontrándose en su estado deformado como $\mathbf{x}=\gamma(\xi,t) \subset \Omega$, en tiempo t. Esta también puede expresarse como:

$$\mathbf{x}=\gamma(\xi,t) = \chi(\Gamma(\xi),t)\,, \text{(1)}$$

Véase la siguiente figura:

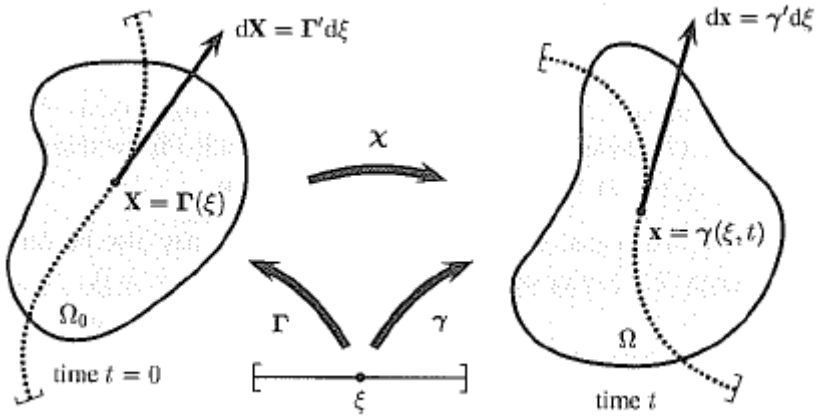

Figura 2.6: Deformación de una curva material Γ en su curva espacial $\gamma \subset \Omega$

Denotamos el vector tangente espacial a la curva deformada como d**x** y el vector tangente material a la curva indeformada como d**X**. Y se definen:

$$d\mathbf{x}=\gamma'(\xi,t)d\xi \quad (2) \qquad\qquad d\mathbf{X}=\Gamma'(\xi)d\xi$$

Usando la expresión (1) y aplicando la regla de la cadena de la derivación, tenemos que:

$$\gamma'(t)=(\partial\chi(\mathbf{X},t)/\partial\mathbf{X})\Gamma'(\xi) \quad (3)$$

Introduciendo la expresión (3) en (2):

$$d\mathbf{x} = \mathbf{F}(\mathbf{X},t)d\mathbf{X}, \quad \text{donde } \mathbf{F}(\mathbf{X},t)=\frac{\partial\chi(\mathbf{X},t)}{\partial\mathbf{X}}$$

La cantidad **F** es crucial en la mecánica de medios continuos no lineal y es una primera medida de la deformación, llamada gradiente de deformación. En general, **F** tiene nueve componentes para todo t, y caracteriza el comportamiento del movimiento en los alrededores de un punto. La expresión $d\mathbf{x} = \mathbf{F}(\mathbf{X},t)d\mathbf{X}$ se trata de una transformación lineal que genera un vector d**x** mediante la acción del tensor de segundo orden **F** sobre el vector d**X**. Nótese que el gradiente de deformación tiene bases en la situación de referencia y en la deformada, por esta particularidad, al tensor de deformación también se le denomina tensor bi-punto.

Se define el jacobiano *J*:

$$J=\det[\mathbf{F}]>0$$

De esta manera **F**, es un tensor de segundo orden no singular para todos los movimientos físicamente admisibles, así:

$$d\mathbf{X}=\mathbf{F}^{-1}d\mathbf{x}$$

2.3.3 Tensor de deformación:

El tensor de deformación es un tensor de segundo orden, que también puede escribirse en su configuración de referencia o deformada. Considérese el producto escalar entre los vectores $d\mathbf{X}_1$ y $d\mathbf{X}_2$ y veamos como se ve afectado este producto por el movimiento.

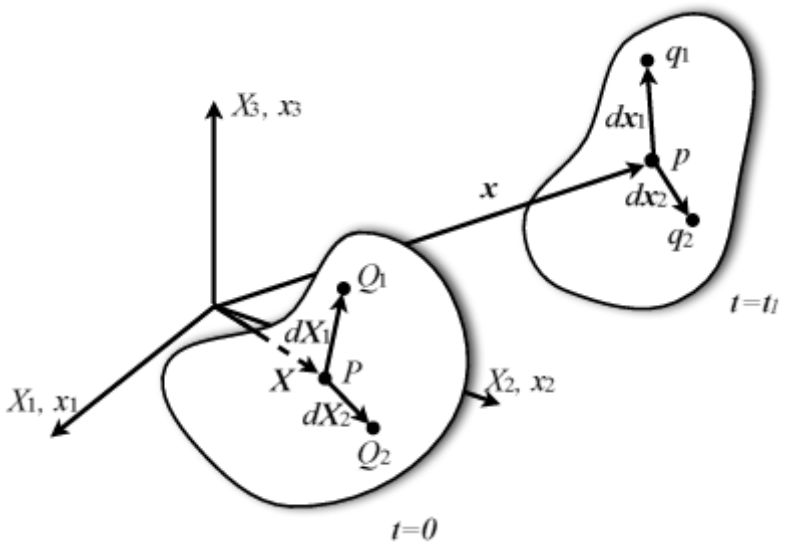

Figura 2.7: Gradiente de deformación

Debido al movimiento, d$\mathbf{X_1}$y d$\mathbf{X_2}$transforman a:

$$d\mathbf{x}_1 = \mathbf{F}d\mathbf{X}_1, \quad d\mathbf{x}_2 = \mathbf{F}d\mathbf{X}_2$$

Realizando el producto escalar entre d$\mathbf{x_1}$y d$\mathbf{x_2}$:

$$d\mathbf{x}_1 \cdot d\mathbf{x}_2 = d\mathbf{X}_1 \cdot \mathbf{C}d\mathbf{X}_2$$

Donde $\mathbf{C}=\mathbf{F}^T\mathbf{F}$ es el tensor de Cauchy-Green por la derecha. Este tensor es puramente material (opera sobre vectores en la configuración de referencia). De manera análoga, los vectores d$\mathbf{X_1}$y d$\mathbf{X_2}$ pueden expresarse en términos de los vectores en la configuración deformada, obteniendo: $\mathbf{b}=\mathbf{F}\mathbf{F}^T$ que es el tensor de Cauchy-Green por la izquierda. El tensor de deformaciones de Cauchy se define como $\mathbf{c}=\mathbf{F}^{-T}\mathbf{F}^{-1}$. Ambos tensores **b** y **c** son puramente espaciales (operan sobre vectores en la configuración espacial o deformada).

El tensor **C** ofrece una métrica adecuada para calcular la magnitud de vectores en la configuración deformada en términos de las componentes del vector material asociado al mismo a través del movimiento. Mientras que el tensor **b** tiene la métrica adecuada para calcular la magnitud de vectores en la configuración indeformada en términos de las componentes del vector espacial asociado al mismo a través del movimiento. Los tensores definidos son reales simétricos, por lo tanto son diagonalizables y poseen tres invariantes.

2.3.4 Concepto de tensión:

El movimiento y la deformación interacción entre la materia y su materia vecina en el interior de un cuerpo. Una de sus consecuencias es la tensión. La noción de tensión, la cual es responsable de la deformación de los materiales, es muy importante en la mecánica de medios continuos. Centremos la atención en un cuerpo continuo deformable β, que ocupa una región arbitraria Ω de un espacio físico con contorno superficial ∂Ω en t, como:

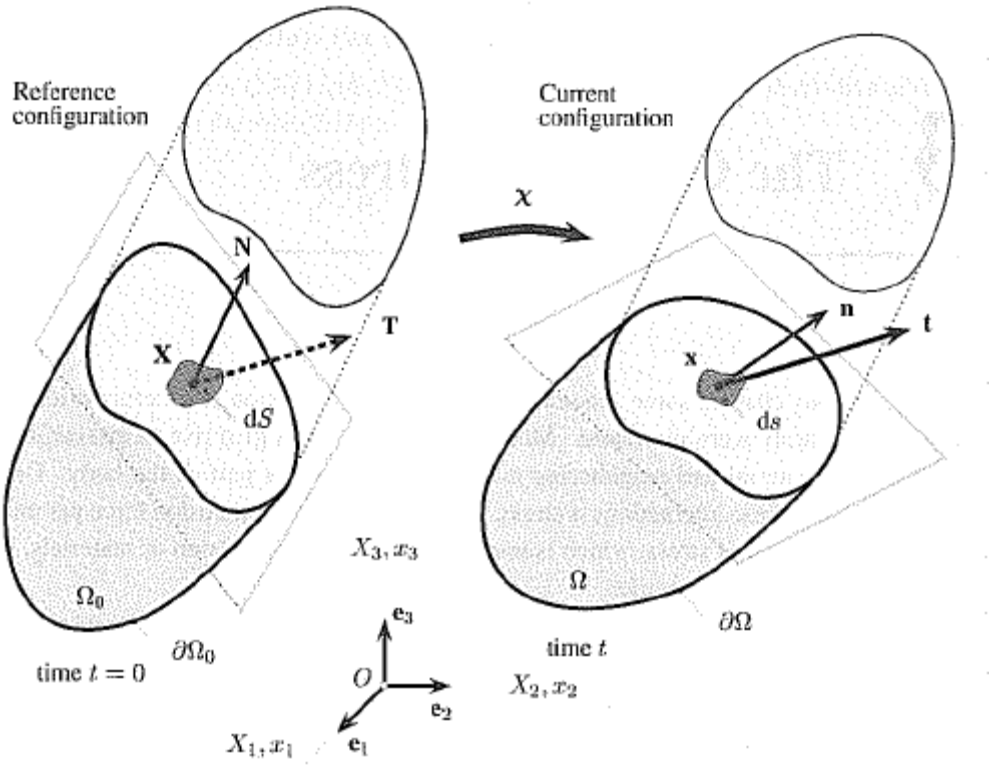

Figura 2.8: Vectores de tracción actuando sobre un elemento de superficie infinitesimal con normales unitarias exteriores.

Se postula que fuerzas arbitrarias actúan sobre las partes o las superficies de contorno (llamadas fuerzas exteriores), y en una superficie "imaginaria" dentro del interior del sólido (llamadas fuerzas internas).

Cortamos el cuerpo con una superficie plana, la cual pasa por un punto dado $\mathbf{x}\epsilon\Omega$ con coordinadas espacial en t. Como se observa, dicho plano divide el cuerpo en dos trozos. Aquí, **t** representa el vector tracción de Cauchy (fuerza medida por unidad de superficie de área definida en la situación deformada), ejercido en un ds con normal positiva **n**. Existen unos únicos tensores de segundo orden que:

$$\mathbf{t}(\mathbf{x}, \mathrm{t}, \mathbf{n}) = \sigma(\mathbf{x}, \mathrm{t}) \cdot \mathbf{n}$$
$$\mathbf{T}(\mathbf{X}, \mathrm{t}, \mathbf{N}) = \mathbf{P}(\mathbf{X}, \mathrm{t}) \cdot \mathbf{N}$$

Donde **σ** denota un tensor simétrico espacial llamado tensor de tensiones de Cauchy, mientras que **P** se conoce como primer tensor de tensiones de Piola-Kirchhoff. Son tensores bi-punto donde un índice describe las coordenadas espaciales y otro las materiales.

Antes de continuar se expone el concepto de objetividad, muy importante en mecánica de sólidos ya que establece la invarianza de las cantidades que describen la deformación del sólido a las rotaciones de cuerpo rígido (que no causan deformación en el cuerpo). Ante una rotación de cuerpo rígido, o de sistema de coordenadas, **Q**, las componentes de un vector y un tensor de segundo orden transforman de acuerdo a:

$$\hat{\mathbf{v}}=\mathbf{Q}\mathbf{v} \qquad \hat{\mathbf{S}}=\mathbf{Q}^{T}\mathbf{S}\mathbf{Q}$$

Las componentes son diferentes pero su magnitud (o autovalores) es idéntica (son idénticos). No todas las cantidades espaciales son objetivas. Por ejemplo, el vector velocidad no es objetivo ante rotaciones de sólido rígido. Mientras que, por ejemplo, los tensores de deformaciones si son invariantes respecto a rotaciones de cuerpo rígido, siendo por ende objetivos.

El tensor de Cauchy es una cantidad fundamental para establecer las ecuaciones de equilibrio y leyes de conservación, por esta razón es necesario establecer si se trata de una cantidad objetiva:

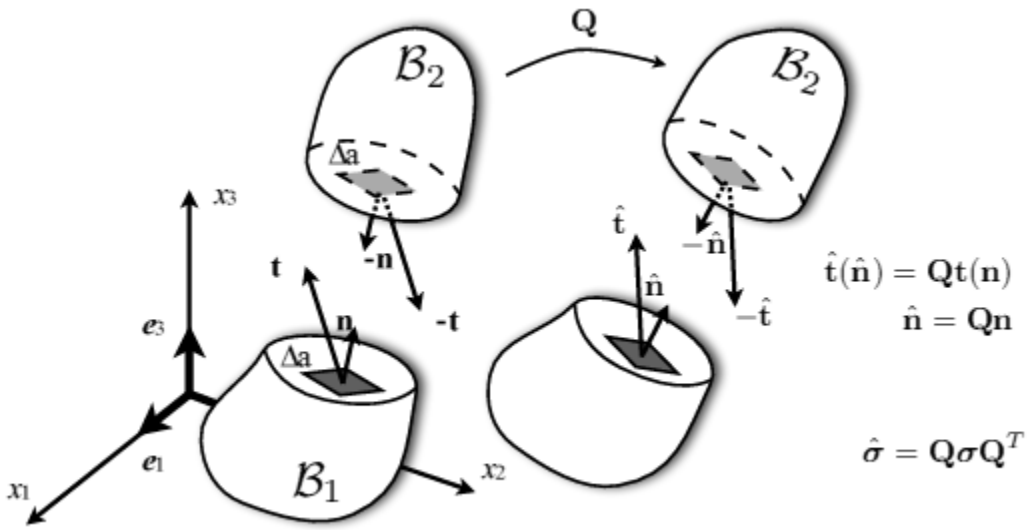

Figura 2.9: Vectores tensión. Tensor de Cauchy

Es por tanto, el tensor de Cauchy una cantidad objetiva. Es fundamental establecer que todas las cantidades empleadas en la obtención de un modelo constitutivo sean objetivas.

2.3.5 Principios de conservación:

Comenzaremos expresando el principio de conservación de la masa. Se considera que en el sólido no se produce la creación ni la destrucción de masa. Por tanto, la misma se conserva.

$$\mathrm{M}=\int_{\beta_o}\rho_o dV=\int_{\beta}\rho\, dv$$

Donde ρ_o es la densidad en la configuración de referencia y ρ es la densidad en la configuración actual. Operando sobre ella puede obtenerse la ecuación en derivadas, también conocida como ecuación de continuidad:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t}+\rho\nabla\cdot\mathbf{v}=0$$

El siguiente teorema que tratamos es el de la cantidad de movimiento. Primero se expondrá la conservación de la cantidad de movimiento lineal y seguidamente para el momento angular.

La conservación de la cantidad de movimiento implica que la variación en la cantidad de movimiento tiene que ser igual a la suma de las fuerzas externas. La cantidad de movimiento lineal se define como:

$$\mathbf{p}=\int_{\beta}\rho\mathbf{v}dv$$

La variación en la cantidad de movimiento es entonces:

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_{\beta}\rho\mathbf{v}dv=\int_{\partial\beta}\mathbf{t}ds+\int_{\beta}\rho\hat{\mathbf{b}}dv$$

Sustituyendo la relación entre el vector tracción y el tensor de Cauchy, aplicando el teorema de la divergencia a la integral de superficie y llevando a cabo la derivación temporal del término de la izquierda obtenemos:

$$\int_{\beta}\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t}dv=\int_{\beta}\left(\nabla\cdot\sigma+\rho\hat{\mathbf{b}}\right)dv$$

La forma diferencial se obtiene al poder considerarse dv arbitrario:

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t}=\nabla\cdot\sigma+\rho\hat{\mathbf{b}}$$

Conocida esta última como ecuación de equilibrio de Cauchy.

La conservación de la cantidad de movimiento angular implica que la variación en la cantidad de movimiento angular tiene que ser igual a la suma de las fuerzas externas. La cantidad de movimiento angular se define como:

$$\mathbf{h}=\int_{\beta}\mathbf{x}\times(\rho\mathbf{v})\mathrm{dv}$$

Considerando el equilibrio rotacional con respecto al origen del sistema coordenado:

$$\frac{\mathrm{D}}{\mathrm{Dt}}\int_{\beta}\mathbf{x}\times(\rho\mathbf{v})dv=\int_{\partial\beta}\mathbf{x}\times\mathbf{t}ds+\int_{\beta}\mathbf{x}\times(\rho\hat{\mathbf{b}})dv$$

Operando sobre esta última expresión lo que se deduce es que el tensor de tensiones es un tensor simétrico: $\sigma=\sigma^{T}$.

Las ecuaciones de cantidades de movimiento han sido expuestas en su formulación espacial. También podría llevarse a cabo una formulación material. De hecho, el resultado de formular la cantidad de movimiento en su configuración material, junto con la ecuación de equilibrio, es la obtención del primer tensor de tensiones de Piola. Se trata de un tensor de segundo orden, $\mathbf{P}=J\boldsymbol{\sigma}\mathbf{F}^{-T}$, que permite relacionar tracciones aplicadas en la superficie de la configuración deformada descritas en términos de la normal a la superficie en la configuración de referencia. Es el que se calcula normalmente en los ensayos de tracción. La fuerza se mide en la posición actual pero el área de la probeta es la inicial. Este tensor no es simétrico, lo que resulta incómodo para operar analíticamente con él. Se puede definir un tensor simétrico como: $\mathbf{S}=J\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T}$, llamado segundo tensor de tensiones de Piola. Este es un tensor puramente material, y no tiene interpretación física precisa. La formulación es por conveniencia matemática.

El siguiente principio es sobre la conservación de la energía, viene dado como:

$$\frac{D}{Dt}(K+\varepsilon)=W+U \qquad \text{donde:}$$

- K es la energía cinética: $k=\frac{1}{2}\int_{\beta}\rho\mathbf{v}\cdot\mathbf{v}dv$

- ε es la energía interna: $\varepsilon=\int_{\beta}\rho e dv$

- W es la potencia mecánica de las fuerzas externas: $W=\int_{\beta}(\rho\hat{\mathbf{b}})\cdot\mathbf{v}dv+\int_{\partial\beta}\mathbf{t}\cdot\mathbf{v}dv$

- *U* es la potencia térmica: $U = \int_{\beta} \rho r dv - \int_{\partial\beta} \mathbf{q} \cdot \mathbf{n} ds$

Sustituyendo cada expresión en la ecuación de conservación y operando:

$$\rho \dot{e} + \nabla \cdot \mathbf{q} = \rho r + \sigma : \mathbf{d}, \quad \text{con } \mathbf{d} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$$

Por último se expone el principio de entropía. La segunda ley de la termodinámica establece que la entropía total nunca es menor al flujo de entropía a través de la superficie del cuerpo y la entropía generada por fuentes en el cuerpo (por ejemplo, fuentes de calor). Sea η la entropía por unidad de volumen, entonces:

$$\frac{\mathrm{D}}{\mathrm{Dt}} \int_{\beta} \rho \eta dv \geq \int_{\beta} \rho \frac{r}{\theta} dv' \int_{\partial\beta} \frac{\mathbf{q} \cdot \mathbf{n}}{\theta} ds$$

Donde *r* es la generación de entropía y **q** el flujo de calor a través de la superficie del cuerpo. Aplicando el teorema de la divergencia a la integral de superficie y realizando la derivada temporal tenemos:

$$\int_{\beta} \rho \dot{\eta} dv \geq \int_{\beta} \left[\rho \frac{r}{\theta} - \nabla \cdot \left(\frac{\mathbf{q}}{\theta} \right) \right] ds$$

$$\rho \dot{\eta} \geq \frac{1}{\theta} \rho r - \nabla \cdot \left(\frac{\mathbf{q}}{\theta} \right)$$

Conocida esta última expresión como ecuación de Clausius-Duhem. Una forma más difundida de la anterior ecuación consiste en eliminar *r* de la ecuación anterior por medio de la ecuación de la energía

$$\rho(\theta \dot{\eta} - \dot{e}) \geq -\sigma : \mathbf{d} + \mathbf{q} \cdot \nabla \log \theta$$

Introduciendo la energía libre de Helmholtz por unidad de volumen material:

$$\psi = \rho_o (e - \theta \eta)$$

El principio de entropía se reduce a:

$$-\rho \left(\frac{\dot{\psi}}{\rho_o} + \eta \dot{\theta} \right) + \sigma : \mathbf{d} - \mathbf{v} \cdot \nabla \log \theta \geq 0$$

Expresión muy útil en el desarrollo de ecuaciones constitutivas de materiales deformables.

2.4 Materiales hiperelásticos.

Las ecuaciones fundamentales que se han realizado en el apartado anterior sobre cinemática, tensiones y balances, sirven para cualquier sólido continuo. Sin embargo no distingue un material de otro. Para sólidos deformables las ecuaciones mencionadas no son suficientes para determinar la respuesta del material. Es necesario establecer ecuaciones adicionales en la forma de leyes constitutivas que nos proporcionen el comportamiento, del tipo σ-ε (tensión – deformación).

Una ecuación constitutiva determina el estado de tensión en un punto **x**de un cuerpo continuo en un tiempo t y es necesariamente diferente para distintos tipos de sólidos continuos. Es la meta de las teorías constitutivas desarrollar modelos matemáticos que representen el comportamiento real de la materia.

Los materiales hiperelásticos postulan la existencia de una función de energía libre ψ, la cual es definida por unidad de volumen (inicial). En el caso de ψ=ψ(**F**) se la conoce como función de energía de deformación. Como se observa es una función que proporciona un escalar en dependencia de un tensor que debe ser continuo.

En materiales homogéneos donde la distribución interna constituyente es asumida como uniforme, la función ψ depende sólo del gradiente de deformación **F**. Los llamados materiales heterogéneos (material que no es homogéneo) tienen una función ψ dependiente además de **F**, de la posición del punto en el medio. Una ecuación constitutiva de un cuerpo elástico isotermo puede representarse de manera general como:

$$\sigma(\mathbf{x},\mathrm{t}) = \mathrm{g}(\mathbf{F}(\mathbf{X},\mathrm{t}),\mathbf{X})$$

Donde g se conoce como función de respuesta asociada con el tensor de tensiones de Cauchy. La generalización expuesta se refiere a un material heterogéneo, de ahí la dependencia de **X**. Como ya se había comentado anteriormente, un tejido blando es a nivel microscópico un material heterogéneo, pero nosotros estudiamos el comportamiento a una escala mayor, por lo que se asumirá como un material homogéneo. La función de respuesta puede considerarse:

$\sigma = \mathrm{g}(\mathbf{F}) = \mathrm{J}^{-1}\dfrac{\partial\psi(\mathbf{F})}{\partial\mathbf{F}}\mathbf{F}^{\mathrm{T}}$ a nivel espacial, y $\mathbf{P}=\mathrm{G}(\mathbf{F})=\dfrac{\partial\psi(\mathbf{F})}{\partial\mathbf{F}}$ a nivel material

Este tipo de ecuaciones es lo que conocemos como modelos constitutivos (puramente mecánicos). Establecen un modelo empírico como la base para aproximar el comportamiento del material real.

Por conveniencia se ha asignado un valor nulo a la función de energía de deformación en la configuración material, que expresamos como:

$$\psi = \psi(\mathbf{I}) = 0$$

Además, la experiencia de la observación física nos dice que la función de energía de deformación aumenta con la deformación:

$$\psi = \psi(\mathbf{F}) \geq 0$$

Resumiendo, estamos diciendo que la función de energía de deformación tiene un mínimo global para **F**=**I** en equilibrio termodinámico. Decimos que la configuración de referencia está libre de tensiones o que las tensiones residuales son cero.

Por último, definiremos el tensor elástico, representado en su forma material por $\mathbb{C}$ y en su configuración espacial por **c**.

$$\mathbb{C}=2\frac{\partial \mathbf{S}(\mathbf{C})}{\partial \mathbf{C}} \quad (4)$$

La cantidad $\mathbb{C}$ caracteriza el gradiente de la función **S** y relaciona el trabajo conjugado de la pareja de tensores de tensión y deformación. Mide el cambio en las tensiones respecto a un cambio en las deformaciones. Es un tensor de rango cuatro, por lo que tiene cuatro índices, siendo simétricos por parejas: primero y segundo, y tercero y cuarto.

$$C_{ABCD} = C_{BACD} = C_{ABDC}$$

Si se asume la existencia de una función energía ψ, entonces **S** puede ser derivado desde ψ de acuerdo con $\mathbf{S} = 2\partial\psi(\mathbf{C})/\partial\mathbf{C}$, y empleando la expresión 4 llegamos a relación:

$$\mathbb{C}=4\frac{\partial^2 \psi(\mathbf{C})}{\partial \mathbf{C}\partial \mathbf{C}}$$

El tensor elástico en su configuración espacial lo obtenemos a partir de $\mathbb{C}$ con la relación correspondiente:

$$\mathbb{c}=J^{-1}\chi_*(\mathbb{C})$$

Los tensores de cuarto orden $\mathbb{C}$ y **c** son muy importantes dentro del concepto de linealización, que debe llevarse a cabo para resolver el problema mediante técnicas de cálculo computacional.

2.5 Modelo de comportamiento.

El modelo de comportamiento material escogido de la literatura es debido a Rubin y Bodner, ha sido desarrollado como una ecuación constitutiva no lineal tridimensional para tejidos biológicos. Es un modelo que también ha sido empleado en la simulación del comportamiento de materiales no lineales de tejidos faciales por Barbarino et al. Este autor consideró sólo la respuesta continua del modelo, la cual usaremos nosotros también. Por esta razón las partes del modelo que describían la respuesta transitoria y disipativa del comportamiento fueron suprimidas.

El modelo de Rubin-Bodner empleado se caracteriza por una función de energía de deformación ψ (por unidad de masa), dada por:

$$\rho_o \psi = \frac{\mu_o}{2q}\left[e^{qg} - 1\right]$$

Donde ρ_o es la densidad de masa en la configuración material, y μ_o y q son parámetros del material. Además, la función g está definida como:

$$g = g_1 + g_2 = 2m_1\left[J - 1 - \ln(J)\right] + (1-w)m_2(\beta_1 - 3) \quad (5)$$

Con g_1 y g_2 caracterizando las respuestas de dilatación elástica y distorsión elástica, respectivamente. m_1 y m_2 son parámetros del material. Debe decirse que el modelo de Rubin-Bodner tiene la capacidad de caracterizar las respuestas de las fibras, así como la viscoplaticidad, asociadas con la componente disipativa de la respuesta del material. Aunque nosotros no hemos considerado los efectos de las fibras y la viscoplaticidad disipativa.

En la ecuación 5, J representa la dilatación, y β_1 es una medida de la distorsión elástica.

$$J = \det(\mathbf{F})$$
$$\beta_1 = \bar{\mathbf{B}} : \mathbf{I}$$

Como sabemos **F** es el gradiente de deformación, y $\bar{\mathbf{B}} = J^{-2/3}\mathbf{B} = J^{-2/3}\mathbf{F}\mathbf{F}^{\mathbf{T}}$ es el tensor izquierdo modificado de deformación de Cauchy-Green. El término w de edad o daño disminuye la rigidez de la respuesta elástica, disminuyendo la deformación distorsional. Finalmente el tensor de tensiones de Cauchy y el tensor elástico en la configuración espacial vienen descritos como:

$$\sigma = \mathrm{m}_1\mu\left[1 - \frac{1}{J}\right]\mathbf{I} + (1\text{-w})\mathrm{m}_2\mu J^{-1}\bar{\mathbf{B}}'$$

$$\mathbf{c} = \frac{2qJ}{\mu}\sigma\otimes\sigma + \mathrm{m}_1\mu\mathbf{I}\otimes\mathbf{I} - 2\mathrm{m}_1\mu\frac{J-1}{J}\mathbf{I}_4 + (1-w)\left\{\frac{2}{3}\frac{\mathrm{m}_2\mu\beta_1}{J}\left[\mathbf{I}_4 - \frac{1}{3}\mathbf{I}\otimes\mathbf{I}\right] - \frac{2}{3}\frac{\mathrm{m}_2\mu}{J}\left[\bar{\mathbf{B}}'\otimes\mathbf{I} + \mathbf{I}\otimes\bar{\mathbf{B}}'\right]\right\}$$

El operador $\otimes$ representa el producto tensorial, y **I** y $\mathbf{I}_4$ los tensores identidad de segundo y cuarto orden, respectivamente. $\bar{\mathbf{B}}'$ es la parte desviadora del tensor izquierdo modificado de deformación de Cauchy-Green:

$$\bar{\mathbf{B}}' = \bar{\mathbf{B}} - \frac{1}{3}(\bar{\mathbf{B}}:\mathbf{I})\mathbf{I}$$

Y μ viene dado por:

$$\mu = \mu_o e^{qg}$$

Respecto a las seis propiedades del material que definen el modelo, Barbarino propone valores para diferentes tejidos faciales, los cuales son recogidos por la siguiente tabla:

	Skin and Mucosa	SMAS	Deep fat	Muscles
ρ (g/cm^3)	1.1	1	1	1
μ_0 (MPa)	1.7	3.7	3.7	3.7
q	36	25	25	25
m_1	1294	595	595	595
m_2	$8\cdot10^{-4}$	$8\cdot10^{-4}$	$7\cdot10^{-5}$	0.00216
ω	0.9	0.9	0.9	0

Figura 2.10: Tabla representativa de las variables del modelo de comportamiento de Rubin-Bodner, para diferentes tejidos blandos.

Debido al alto contenido en agua que presentan los tejidos blandos, puede verse en la tabla que la mayor parte de los tejidos tienen una densidad de 1 g/cm^3, excepto la piel y la mucosa con 1,1 g/cm^3.

La función de energía de deformación que emplea Barbarino está en unidades de energía por unidad de masa. Según el libro de Holzapfel normalmente se utiliza un empleo de ψ con unidades de energía por unidad de volumen, que será la que nosotros utilizaremos. Es decir, Barbarino en emplea $\rho\psi_m (=\psi_V)$, y nosotros buscamos $\bar{\psi} = \psi_V$. Por lo que consideraremos ρ=1 para tener $\bar{\psi}$ con unidades de MPa. Por lo que finalmente la función de energía de deformación utilizada es:

$$\psi = \frac{\mu_o}{2q}\left[e^{qg} - 1\right]$$

Donde las variables tienen el significado anteriormente definido.

En las simulaciones que realicemos, necesitaremos que nuestro modelo esté definido con unidades del sistema internacional. Se hace para poder superponer las cargas gravitatorias con otras cargas: puntuales, presión, dinámicas, etc. Además, si empleásemos milímetros, el modelo de comportamiento hiperelástico daría muchos problemas de convergencia, porque las cargas saldrían irreales y se deformaría mucho.

Como consecuencia de lo expuesto, el valor de densidad que estableceremos de manera generalizada para todos los tejidos de la cara de nuestro modelo será de ρ=1000kg/m^3. El parámetro μ_o queda definido con unidades de Pascal (Pa).

El programa comercial ABAQUS no tiene definido el modelo de comportamiento de Rubin-Bodner, pero permite crear una subrutina de usuario (UHYPER) que defina el comportamiento escogido del material hiperelástico. Algunas características de la UHYPER son:

- Puede usarse para definir la función de energía de deformación del comportamiento hiperelástico de un material isótropo;

- Puede llamarse para el cálculo en todos los puntos de elementos para los cuales la definición del material contenga UHYPER;

- Puede incluir comportamientos que dependan de variables de campo o variables de estado; y

- Requiere que los valores de las derivadas de la función de energía de deformación del material hiperelástico estén definidas con respecto a los invariantes de deformaciones.

La UHYPER programada debe tener la siguiente estructura de encabezado y final:

```
SUBROUTINE UHYPER(BI1,BI2,AJ,U,UI1,UI2,UI3,TEMP,NOEL,
1 CMNAME,INCMPFLAG,NUMSTATEV,STATEV,NUMFIELDV,FIELDV,
2 FIELDVINC,NUMPROPS,PROPS)
C
INCLUDE'ABA_PARAM.INC'
C
CHARACTER*80 CMNAME
      REAL*4 MU,Q,M1,M2,W
DIMENSION U(2),UI1(3),UI2(6),UI3(6),STATEV(*),FIELDV(*),
2 FIELDVINC(*),PROPS(*)
```

→Código escrito←

```
RETURN
END
```

La estructura:

```
MU=PROPS(1)
Q=PROPS(2)
M1=PROPS(3)
M2=PROPS(4)
W=PROPS(5)
```

Representa los cinco parámetros de los cuales depende el comportamiento. Cuando se hace llamada a la UHYPER desde un archivo de carga, se realiza como:

```
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
```

Y la línea que sigue a la expuesta debe contener los valores de los cinco parámetros en el orden establecido: PROPS(1), PROPS(2), … , PROPS(5).

La siguiente estructura contiene las variables que deben ser definidas:

- *U(1)* : Es la función de energía de deformación.

- *U(2)*: Es la parte desviadora de la función de energía de deformación, aunque en nuestro caso es nula y por ello no aparece.

El resto son las distintas derivas de ψ respecto de los invariantes de deformaciones, $\bar{I}_1$, $\bar{I}_2$, $\bar{I}_3$. En nuestro caso $\bar{I}_1$=β_1, distorsión elástica, e $\bar{I}_3$=J, medida de la dilatación. El invariante $\bar{I}_2$ no aparece el modelo de Rubin-Bodner, por lo que las derivadas respecto al segundo invariante son todas nulas y no se definen.

- UI1(1) : $\partial\psi/\partial\bar{I}_1$:

$$\frac{\partial\psi}{\partial\beta_1}=\frac{\mu_o}{2}\left[e^{qg}-1\right](1-w)m_2$$

- UI1(3) : $\partial\psi/\partial J$:

$$\frac{\partial\psi}{\partial J}=\frac{\mu_o}{2}e^{qg}2m_1\left(1-\frac{1}{J}\right)$$

- UI2(1) : $\partial^2\psi/\partial\bar{I}_1^{\,2}$:

$$\frac{\partial^2\psi}{\partial\beta_1^{\,2}}=\frac{\mu_o}{2}e^{qg}q(1-w)^2m_2^{\,2}$$

- UI2(3) : $\partial^2\psi/\partial J^2$:

$$\frac{\partial^2\psi}{\partial J^2} = \frac{\mu_o}{2} e^{qg}\left[\frac{2m_1}{J^2} + q\left(2m_1\left(1-\frac{1}{J}\right)\right)^2\right]$$

- UI2(5) : $\partial^2\psi/\partial\bar{I}_1\partial J$

$$\frac{\partial^2\psi}{\partial\beta_1\partial J} = \frac{\mu_o}{2} e^{qg} q 2m_1\left(1-\frac{1}{J}\right) m_2(1-w)$$

- UI3(1) : $\partial^3\psi/\partial\bar{I}_1{}^2\partial J$:

$$\frac{\partial^3\psi}{\partial\beta_1{}^2\partial J} = \frac{\mu_o}{2} e^{qg} q^2 m_2{}^2(1-w)^2 2m_1\left(1-\frac{1}{J}\right)$$

- UI3(4) : $\partial^3\psi/\partial\bar{I}_1\partial J^2$:

$$\frac{\partial^3\psi}{\partial\beta_1\partial J^2} = \frac{\mu_o}{2} e^{qg} m_2(1-w)\left[\frac{2m_1 q}{J^2} + \left(q2m_1\left(1-\frac{1}{J}\right)\right)^2\right]$$

- UI3(6) : $\partial^3\psi/\partial J^3$

$$\frac{\partial^3\psi}{\partial J^3} = \frac{\mu_o}{2q} e^{qg}\left\{q2m_1\left(1-\frac{1}{J}\right)\left[\frac{2m_1 q}{J^2} + \left(q2m_1\left(1-\frac{1}{J}\right)\right)^2\right] + \left(-\frac{4m_1 q}{J^3} + \frac{8m_1{}^2 q^2}{J^2}\left(1-\frac{1}{J}\right)\right)\right\}$$

La UHYPER creada puede consultarse en el Anexo I.

Capítulo 3:

MODELO DE ELEMENTOS FINITOS

El objetivo de este capítulo es explicar el proceso que se sigue para crear un modelo de elementos finitos de la cara. Se empieza explicando el proceso de segmentación del TAC para obtener el modelo inicial. A continuación se habla de las modificaciones realizadas para conseguir el modelo final. Por último se describen las condiciones de contorno y cargas que se necesitan para conseguir los estados que se quieren simular.

3.1 Introducción.

Lograr un modelo de elementos finitos de la cara no supone una tarea sencilla. La geometría de la cara es muy compleja. Los programas más avanzados de diseño gráfico tampoco nos capacitan para obtener un modelo realista. Alcanzar un modelo lo más objetivo posible no podría lograrse sin los avances tecnológicos producidos en los últimos años sobre el campo de adquisición de imágenes médicas.

Figura 3.1: Escáner de tomografía axial computarizada.

El escáner de tomografía axial computarizada, como el de la fotografía, significó una auténtica revolución en el campo de la radiología. Se basa en el enfoque de un haz de rayos X sobre el paciente, donde la radiación remanente atenuada es medida por un detector cuya respuesta se transmite a un ordenador. El ordenador analiza la señal del detector, reconstruye la imagen y la presenta en un monitor de televisión. Mediante ecuaciones matemáticas (algoritmos) adaptadas al procesamiento informático se efectúa una reconstrucción por ordenador de vistas transversales de la región anatómica de interés.

Por otro lado, se tiene el software para la segmentación del TAC y obtención de un primer modelo de elementos finitos. La segmentación consiste en una selección de niveles de grises sobre el TAC, identificando las regiones anatómicas de interés, que en este caso serían los tejidos blandos (exteriores) de la cara y los huesos (mandíbula, maxilar superior y cráneo). Con ayuda de un programa que genera la malla se obtiene un primer modelo de elementos finitos. Éste no es definitivo, siendo necesaria la realización de operaciones como separar piel y hueso en el vestíbulo, crear superficies de contacto, etc.

La obtención de un modelo de cara tridimensional nace con necesidad de realizar un primer ajuste del comportamiento de los tejidos. Tras la validación del mismo, el objetivo es simular una operación de mandíbula sobre el modelo, atendiendo con

principal interés a las zonas principales definidas por los cirujanos, que comprende desde los labios hasta la punta de la nariz.

3.2 Segmentación.

Partiendo de las imágenes proporcionadas por los TAC, para obtener el modelo de elementos finitos sobre el que se va a trabajar, se usa el programa SIMPLEWARE. Este programa se divide en tres módulos independientes: ScanIP, ScanFE y ScanCAD.

Con el ScanIP se consigue, partiendo de una serie de imágenes, que pueden venir dadas en diferentes formatos, generar un sólido virtual en el que se distinguen las diferentes estructuras que formarán parte del modelo. Una vez realizado el sólido, éste se exporta a ScanFE.

Las funciones principales del ScanFE son suavizar y mallar la geometría del sólido proveniente del módulo anterior.

El ScanCAD proporciona las herramientas típicas de los programas de CAD. Permite crear y dibujar geometrías que podrán añadirse a modelos creados con los anteriores paquetes de SIMPLEWARE. Este módulo no se ha utilizado en este trabajo.

A continuación se describe la metodología empleada a la hora de obtener el modelo del proyecto.

Lo primero que se hace es cargar las imágenes del paciente proporcionadas pos los TAC en ScanIP. No es válido cualquier conjunto de imágenes, siendo necesarias unas ciertas condiciones a la hora poder trabajar con la tomografía. Hay que comprobar que el TAC no tenga imperfecciones. Estas imperfecciones son originadas normalmente por artefactos, como puede ser un empaste, que distorsiona las imágenes del TAC en dicha región. También es deseable que el paciente presente una cierta simetría sagital y que el TAC esté lo más completo posible, en el sentido de que el TAC no se corte inferiormente en el mentón, sino en el cuello, proporcionando una información completa de la parte inferior de la cara. El conjunto de imágenes que se tienen en este caso cumple todos estos requisitos, siendo por tanto un caso apto para su estudio.

La cantidad de información que contienen los TAC originales es muy grande. Esto dificulta el trabajo sobre ellos, con tiempos de espera largos en la realización de operaciones, a priori simples. Se puede comenzar a trabajar sobre el TAC original, pero conforme se avance en el segmentado, el archivo de trabajo irá aumentando de tamaño, pudiéndose ver interrumpido el trabajo realizado, por insuficiencia de recursos del ordenador para procesar el archivo. La gran cantidad de información que contienen los TAC originales se debe a que tienen un espaciado pequeño entre cortes transversales. Aumentando el espacio entre cortes, la información existente

disminuye, siendo interpolada al nuevo espaciado. Así mismo disminuye también el tamaño del archivo. Esta operación se llama *resample* y puede llevarse a cabo en ScanIP.

El fin del *resample* no es sólo disminuir el peso del archivo. También es necesario porque el mallador de ScanFE se basa en introducir un elemento hexaédrico sobre cada unidad elemental de volumen, o voxel. Un voxel es la unidad cúbica que compone un objeto tridimensional, es lo equivalente a un pixel en un objeto 3D. El tamaño del voxel dependerá de la distancia entre rebanadas del TAC. Tener un tamaño de voxel pequeño implicaría tener un número de elementos muy grande, que se traduce en un coste computacional mayor (tiempo de realización de una simulación mayor). Además, para realizar un suavizado es necesario mallar con elementos tetraédricos. Por lo que, en cada voxel se podrán introducir varios tetraedros. De manera que si se tuviera una distancia entre cortes muy pequeña, junto con el empleo de elementos tetraédricos, el número de elementos se dispararía a las decenas de millones. Resultando un problema inabordable computacionalmente para nuestros recursos. Como se muestra en la siguiente figura, para una serie de tamaños de voxel desde 0,5 mm de lado hasta 1,5 mm de lado, la evolución del número de elementos que tendría el modelo disminuye considerablemente.

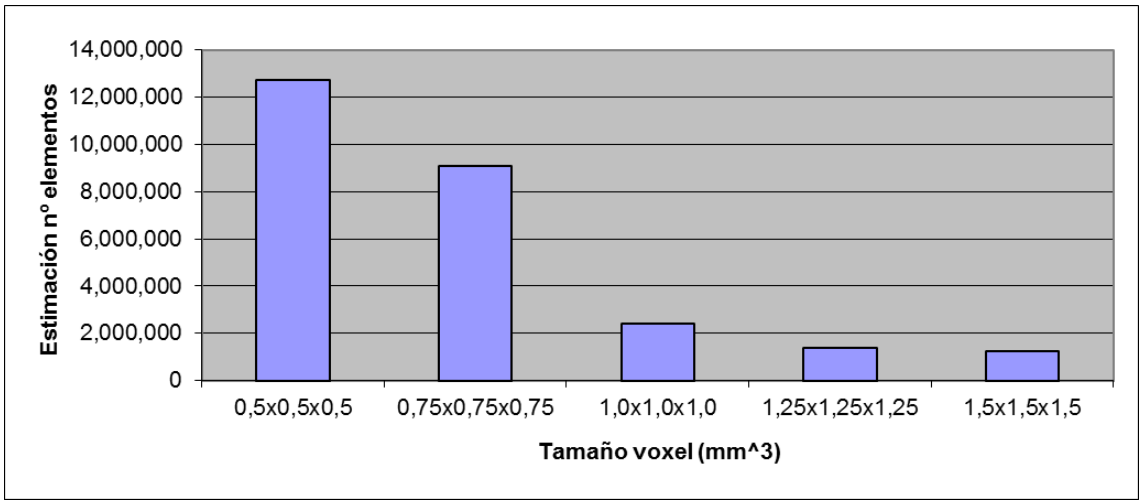

Figura 3.2: Estimación del número de elementos frente al tamaño de voxel

Una vez expuesta la necesidad e importancia del *resample*, decir que sobre el modelo definitivo se optó por un *resample* cúbico de 1,5 mm, que preveía un total de elementos tetraédricos (entre hueso y piel) de alrededor 1,25 millones. Un *resample* cúbico es aquel que tiene la misma distancia entre cortes a lo largo de los tres ejes.

Como paso previo a la segmentación, se realizó un giro alrededor del eje x de 340º de todas las imágenes. Esta operación es necesaria para hacer paralelos los planos perpendiculares al eje z con los planos anatómicos coronales. De esta forma la aplicación posterior de cargas y movimientos al modelo se simplifica bastante (se

aplican directamente en las direcciones de los ejes). A partir de ahora comienza realmente la segmentación del TAC.

Antes de realizar el r*esample* se delimita lo que es hueso y lo que es tejido blando. Para ello se usa la opción de segmentación por umbrales (*threshold*) de ScanIP, seleccionando el nivel de gris por umbrales, a modo de prueba y error, hasta tener unos valores de límites que proporcionen lo que es hueso y tejido blando de manera aproximada. Esta operación sería menos precisa si se realiza después del *resample* porque se perdería información. La forma de separar estructuras que tiene el programa es asociando cada una de ellas con una capa distinta. Estas capas, al exportar los resultados a ABAQUS, se convierten en set de elementos separados, de forma que se le pueden asignar facilmente distintas propiedades. Con la separación entre hueso y tejido blando más o menos definida, se realiza el *resample*. En la siguiente imagen se observa el resultado después de haber realizado estas dos operaciones (segmentación por umbrales más *resample*). En rojo se representa la capa de la piel y en blanco la capa del hueso. Se puede ver que hay bastantes imperfecciones en la capa de la piel y que los tejidos interiores al hueso están dentre de esa capa, por lo que, al no interesarnos, es necesario borrarlos como se comentará a continuación.

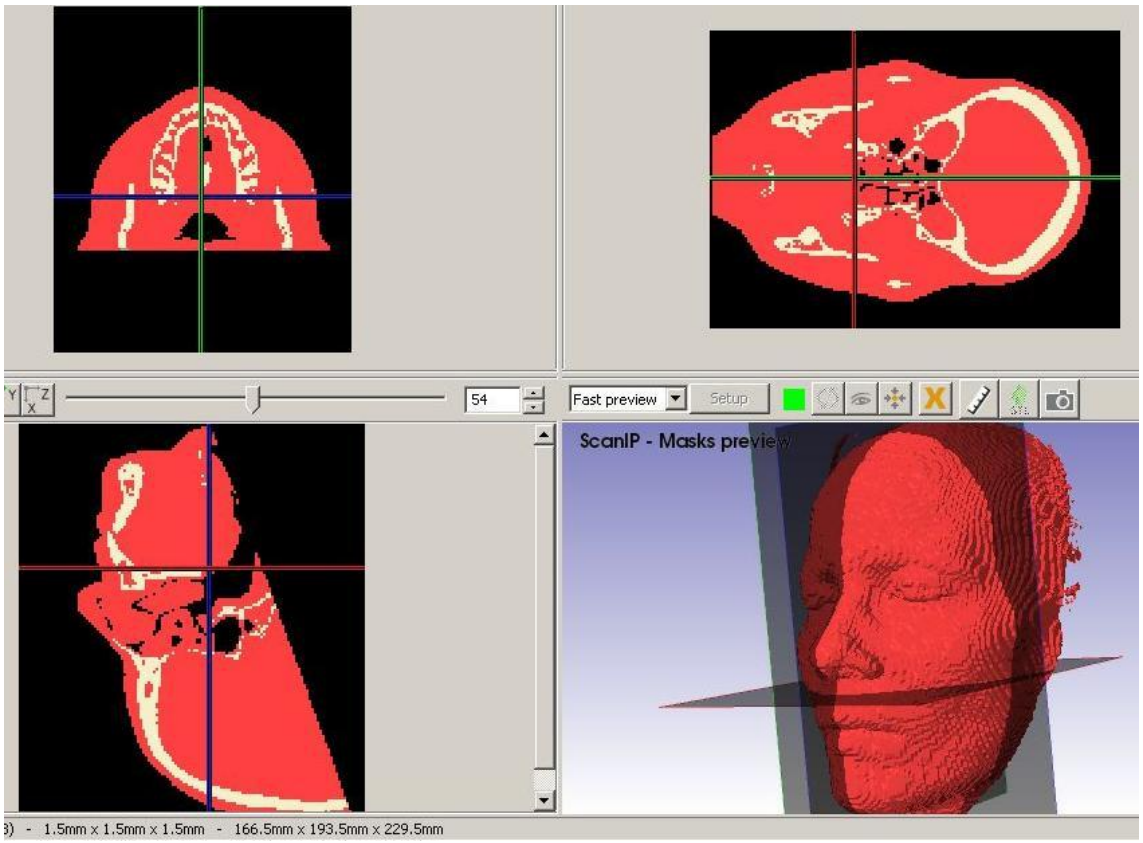

Figura 3.3: Ventana de trabajo de ScanIP

A partir de aquí, la manera de avanzar en la segmentación del TAC es mucho más trabajosa. Corte por corte hay que ir seleccionando con el ratón (pintando o despintando) la zona que pertenece a la piel y la que pertenece al hueso. Aunque, gracias a haber realizado el *resample*, la cantidad de cortes a disminuido cosiderablemente, el número de ellos aún es batante grande (pasa de alrededor de 450 a 160). La primera división hecha usando los niveles de gris por umbrales, tal como se explica en el párrafo anterior, no termina de ser del todo precisa. Por esta razón hay que afinar el segmentado, asegurándonos de que no existan zonas que no estén bien delimitadas y que no haya superposición de capas. Todos los tejidos blandos del interior (del hueso hacia dentro) se elimininan; sólo resultan de interés los tejidos exteriores al hueso.

A diferencia del modelo de José Manuel, en el momento de efectuar el TAC se le pidió al paciente que permaneciera con los labios ligeramente separados. Esto se ha tenido en cuenta al realizar la segmentación de esa zona, definiendo un pequeño espacio entre ellos que simula la comisura de los labios. De esta forma se evita tener que separarlos posteriormente.

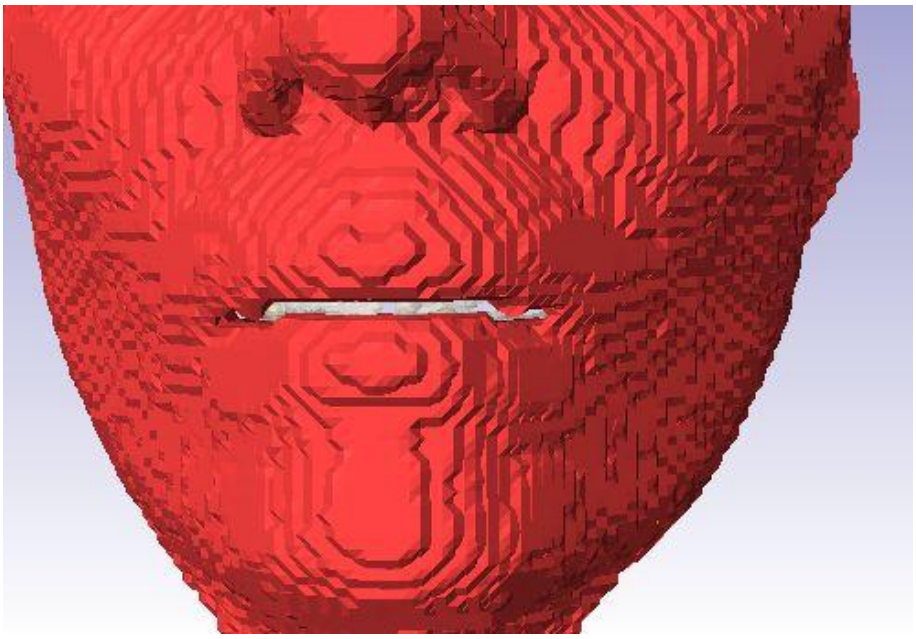

Figura 3.4: Detalle de los labios separados

Terminado el proceso de segmentación se tiene un sólido 3D compuesto por las dos capas definidas: piel y hueso. Piel contiene todos los tejidos blandos exteriores. Hueso es la selección que se hace de las zonas óseas y los dientes, que posteriormente será considerado como un sólido rígido en las simulaciones.

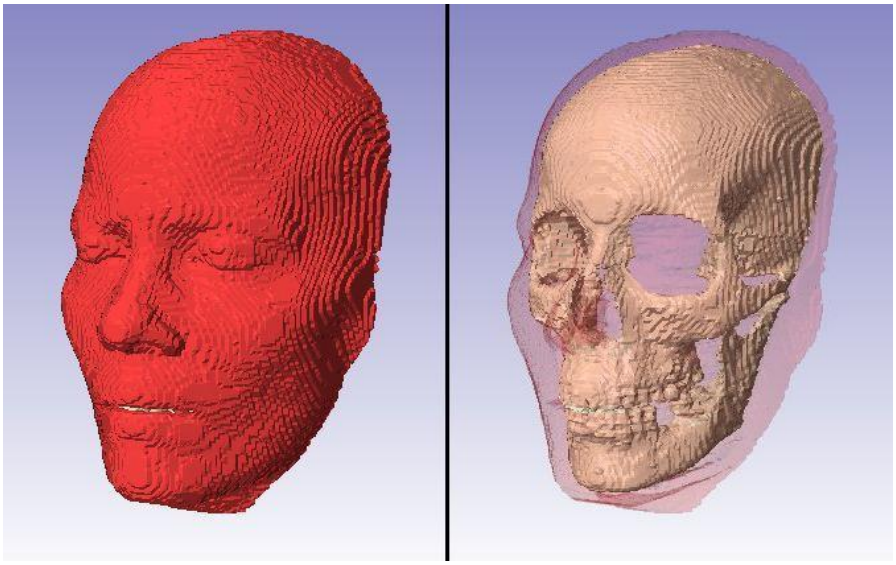

Figura 3.5: Modelo 3D obtenido de la segmentación en Scan IP

La exportación hacia ScanFE se hace con las opciones por defecto: pre-suavizado usando valores de escala de grises y permitiendo intercambio de partes.

Ya en el módulo de ScanFE, en las opciones de mallar, se optó por aplicar un suavizado a la malla inicial (de elementos hexaédricos), en la que todos los elementos fueran tetraedros. También se activó la pestaña de suavizado adicional con los parámetros por defecto del programa. Terminadas estas operaciones, el programa ya ha calculado una malla para la geometría. Basta exportarla (escribir la malla en formato .inp) para tener el primer modelo de elementos finitos en ABAQUS. Entre las opciones que aparecen a la hora de exportar el modelo están la elección de unidades y el tipo de elemento para el mallado. La definición de coordenadas nodales se pidió en metros para evitar problemas de unidades cuando se aplicaran cargas gravitatorias y el tipo de elemento escogido fue el tetraedro lineal.

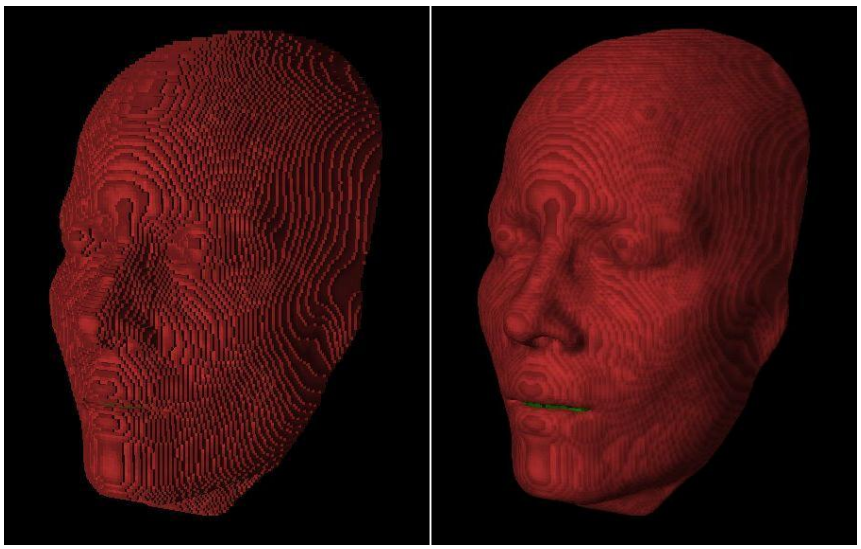

Figura 3.6: Comparación entre la superficie de la cara antes y después de aplicar el suavizado

Hasta aquí llega la utilización del programa SIMPLEWARE. A partir de ahora se trabajará principalmente con ABAQUS y algunas rutinas de FORTRAN.

Resumiendo hasta este punto, se tiene un primer modelo con 334598 nodos y 1580239 elementos. Los elementos se reparten en las diferentes capas como: PIEL (1003133) y HUESO (577106). Se debe aclarar que el término PIEL de la capa creada con dicho nombre no hace mención a la piel como tal, sino que se refiere al conjunto de los tejidos blandos. A continuación se representa una imagen de ABAQUS del modelo obtenido.

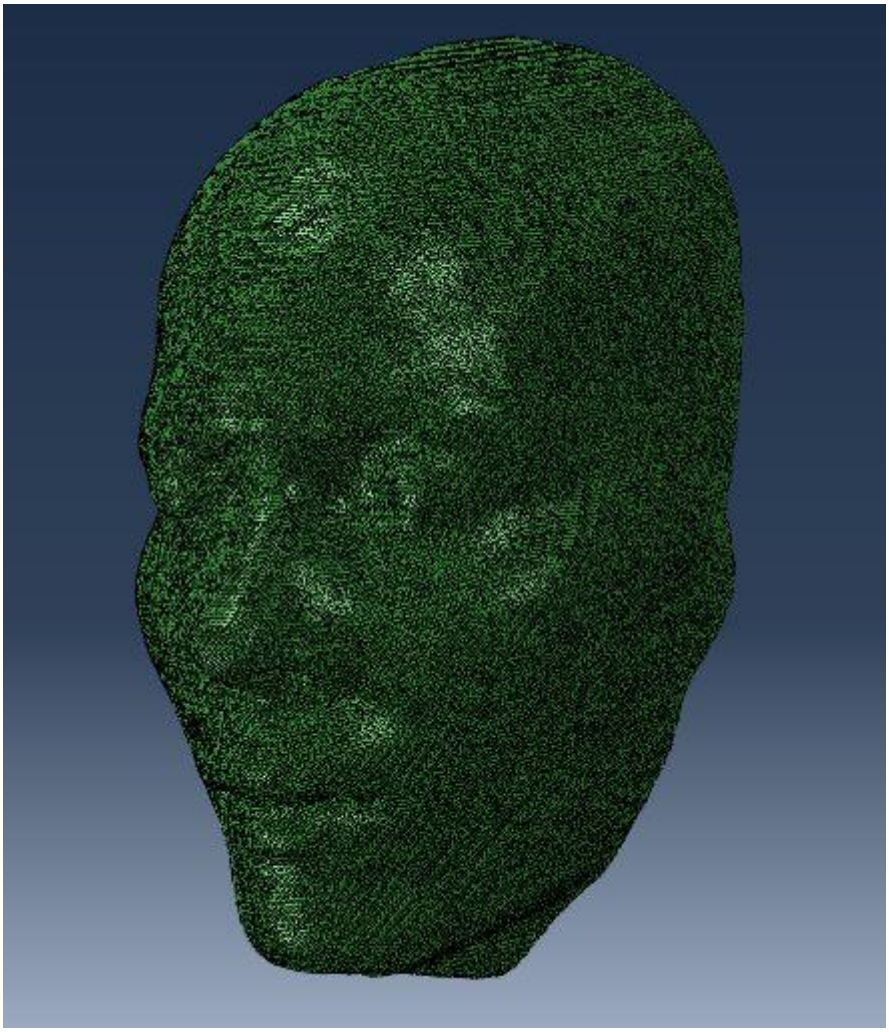

Figura 3.7: Modelo 3D una vez exportado a ABAQUS

No obstante, este modelo no es definitivo y será necesario realizar diferentes operaciones sobre el mismo, que se describirán en los siguientes apartados.

3.3 Creación del vestíbulo de la boca.

Las encías, junto con los dientes, forman las arcadas gingivodentarias, constituyendo la barrera que separa el vestíbulo de la cavidad bucal. El vestíbulo de la boca es el espacio limitado por delante por los labios, a los lados por las mejillas y por detrás por estas arcadas gingivodentarias. El modelo que se tiene no simula la separación existente en el vestíbulo. Durante la segmentación no fue posible separar la piel del hueso en esa zona. Para hacer más realista el modelo se separan estas capas tal y como se describe en este apartado.

La zona que se ha elegido para crear el vestíbulo está comprendida por un rectángulo de lados paralelos. Se extiende verticalmente desde el surco entre el labio y el mentón hasta la base de la nariz, y horizontalmente va desde un pómulo hasta el otro. En la siguiente imagen se representan los elementos seleccionados.

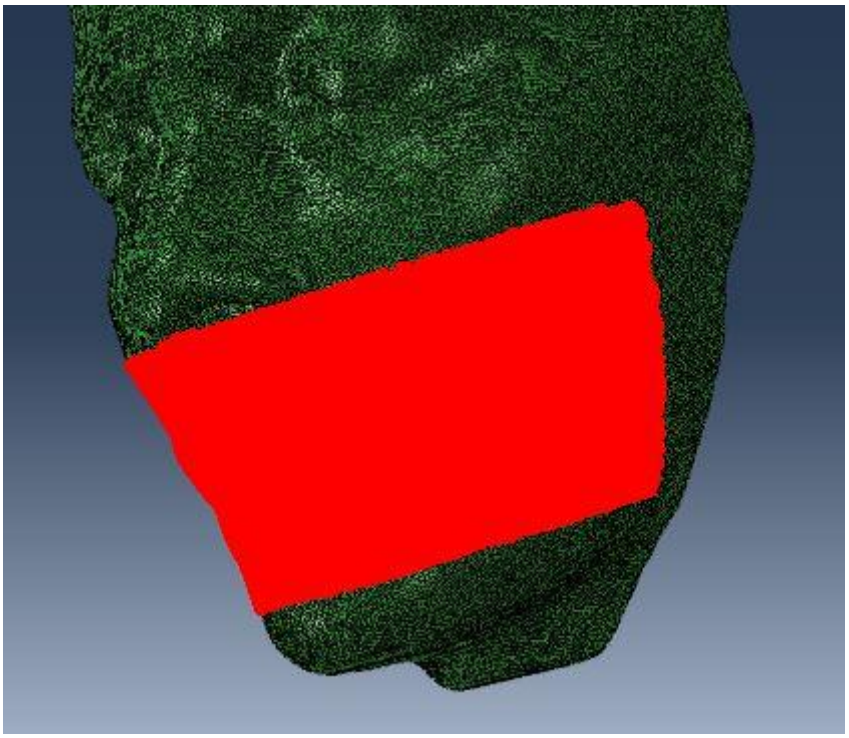

Figura 3.8: Selección de elementos del vestíbulo

A partir de este conjunto se crea en ABAQUS un set de nodos que contenga los nodos de los elementos anteriores. Con este set de nodos creado, se escribe el nuevo JOB del modelo (archivo con formato ".inp") del que se obtienen los nodos contenidos en la selección anterior.

Conviene, antes de continuar con la creación del vestíbulo, hacer una aclaración sobre los datos que se pueden extraer de los archivos .inp obtenidos al escribir un JOB desde ABAQUS. En el archivo inicial que crea ScanFE al exportar el modelo a ABAQUS

se encuentra la definición de todos los nodos y elementos del modelo, así como los set de elementos de piel y hueso. Los nodos se definen con cuatro números: el primero de ellos le da nombre al nodo y los otros tres son sus coordenadas espaciales dentro del modelo. Cada nodo se nombra por un solo número, pero en el mismo lugar (mismas coordenadas) puede haber más de un nodo. Los elementos se definen con 5 números: el primero nombra al elemento y los otros cuatro son los cuatro nodos que lo forman (caso de elementos tetraédricos). Cuando en ABAQUS CAE se escribe un nuevo JOB en el que se ha creado algún set de nodos o elementos, lo nuevo que aparece en el archivo ".inp" no es la definición de esos nodos o elementos, sino todos los nombres (número que los designa) de los nodos o elementos que pertenecen a ese grupo.

Los nodos del vestíbulo son los nodos del set creado que a su vez son comunes a los grupos PIEL y HUESO. Para separar lo que se hace es duplicar estos nodos comunes. De esta forma piel y hueso en esa zona pueden moverse independientemente. Hay que decirdir a que región se le asignan los nuevos nodos. En este caso interesa dárselos al grupo HUESO. El motivo de esta elección es porque a los nodos de este grupo se le restringirán los desplazamientos (en la gran mayoría de las simulaciones), estableciéndolo como un sólido rígido. En el montaje del sistema de ecuaciones que realiza el módulo de cálculo de ABAQUS, haber asignado los nodos duplicados a un grupo rígido se traduce en disminuir en cierta medida los tiempo de cálculo.

Esta operación se lleva a cabo con una rutina de FORTRAN escrita por José Manuel (*comunes_y_duplica.for*). Los datos de entrada son los nodos del set del vestíbulo, todos los nodos de la piel y el hueso, la definición de todos los nodos del modelo y la definición de todos los elementos del hueso. A partir de los datos de entrada, el programa realiza las siguientes operaciones:

- Identifica y almacena todos los nodos comunes a PIEL y HUESO.
- Dentro de los nodos anteriores, selecciona los que pertenecen al set de nodos del vestíbulo.
- Obtiene la definición de los nodos comunes en el vestíbulo.
- Se duplican los nodos comunes del vestíbulo.
- Se nombra a los nuevos nodos a partir del número 600000 (suficientemente alto para no interferir con los nodos del modelo inicial).
- Se redefinen los elementos del hueso sustituyendo los nodos que eran comunes en el vestíbulo por los nuevos.

Con el vestíbulo separado los nodos de piel y hueso del vestíbulo se mueven independientemente. Esto puede llevar a que, ante algunos casos de carga, aparezca superposición entre elementos en esa zona. Aún sabiendo que puede darse esta

situación, todas las simulaciones realizadas para la obtención de las propiedades de la cara se han hecho con este modelo. Se ha supuesto, y posteriormente comprobado, que dado los casos de carga que se iban a tratar, en el caso de producirse esta penetración su valor sería ínfimo.

Con el objetivo de evitar esta situación irreal, una de las mejoras que se aplican al modelo es la de definir un contacto entre las superficies del vestíbulo, que evite que la piel penetre en el hueso. Cómo afecta esto a los resultados y la comparación con el modelo inicial se discute en el Capítulo 5.

3.4 División de la piel en grupos de elementos.

Para poder definir distintas propiedades a la piel, dependiendo de la zona de la cara en que se encuentre, es necesario dividirla en diferentes grupos de elementos. A priori se desconoces cuantas zonas de la cara deberán considerarse. Teniendo en cuenta las conclusiones del proyecto previo (José Manuel) se ha optado por separarla en seis zonas: labio superior, labio inferior, mentón, punta de la nariz, papada más cuello y resto. Un número mayor de regiones complicaría el proceso de ajuste de las propiedas, que se realiza posteriormente, mientras que un número menor podría no ser suficiente.

La asignación de diferentes elementos a formar un grupo se realiza en el archivo de carga de ABAQUS mediante el comando **ELSET*. Anteriormente ha habido que seleccionar en ABAQUS CAE los elementos que forman parte de cada grupo. La creación de grupos debe hacerse con cuidado de no tener elementos compartidos entre grupos distintos y sin dejar ningún elemento suelto, es decir, sin grupo. Incumplirlo significaría tener problemas en el módulo preprocesador de ABAQUS, ya sea por asignación de propiedades diferentes a un mismo elemento, o por dejar algún elemento sin definir sus propiedades.

En la siguiente imagen pueden distinguirse, por colores, las zonas a las que se le asignan propiedades diferentes para conseguir el ajuste:

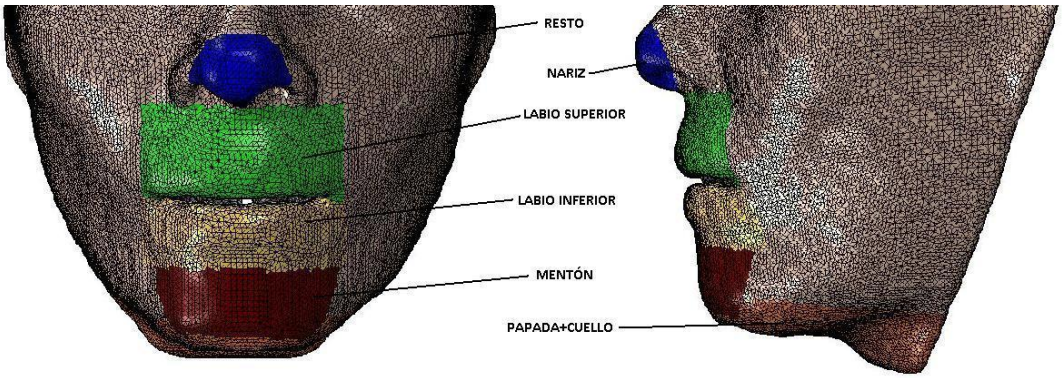

Figura 3.9: Zonas de ajuste del modelo 3D

Se observa que las divisiones están hechas principalmente en la zona central de la cara. Como se comenta en el siguiente capítulo, los desplazamientos que interesan son los que se producen en un corte sagital del modelo que divida la cara en dos mitades. Por esta razón la zona que más interesa es justamente esa zona central.

Aparte, también se creó un grupo denominado HUESO, con los elementos de la segmentación del hueso. Este grupo no es necesario subdividirlo porque todo él se considerará como un sólido rígido (desplazamientos nulos en todos sus puntos) en las simulaciones relizadas para obtener las propiedades de la piel.

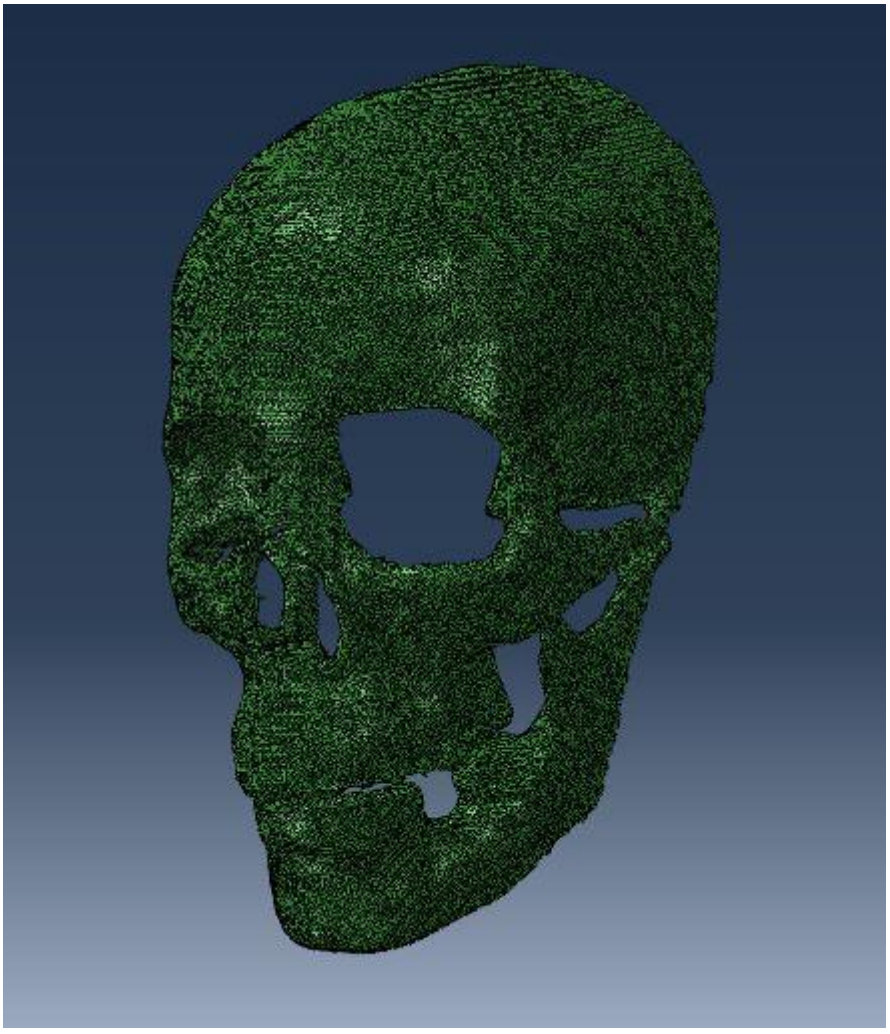

Figura 3.10: Elementos del grupo HUESO

3.5 Propiedades del material.

Como ya se ha comentado, la ley de comportamiento que se usa para los tejidos es la ley de comportamiento de Rubin-Bodner, cuya expresión viene dada por una función de energía de deformación:

$$\rho_o \psi = \frac{\mu_o}{2q}\left[e^{qg} \ -1\right]$$

Donde la función g se define como:

$$g = g_1 + g_2 = 2m_1\left[J - 1 - \ln(J)\right] + (1-w)m_2(\beta_1 - 3)$$

Si se considera que, por el alto contenido de agua en los tejidos, la densidad de estos es igual a la del agua, esta ley de comportamiento depende de cinco parámetros. Un ajuste de los cinco parámetros sería demasiado complicado, por lo que se realizó un estudio previo de la sensibilidad de todos ellos para determinar la importancia de cada parámetro y su influencia, rigidizadora o flexibilizadora al aumentarlo o disminuirlo. Este estudio permite simplificar el ajuste al parámetro más significativo, es decir, aquel que presenta una mayor sensibilidad ante una variación del mismo.

El estudio de sensibilidad, realizado por José Manuel, consistió en tomar los valores de los parámetros de la ley de comportamiento del artículo de Barbarino. Dichos valores son :

m_1	m_2	μ [MPa]	q	w
595	0.00213	3.7	25	0

Entorno a esos valores, para cada simulación, se aumentó o redujo uno de los parámetros, con el objetivo de ver el efecto sobre los desplazamientos producidos. De esta forma se sacó toda la casuística posible.

Los desplazamientos se midieron en cuatro nodos de cuatro zonas diferentes: labio superior, labio inferior, nariz y mejilla. Además se aplicaron tres pasos de carga distintos: el STEP1 correspondiente a la gravedad actuando en sentido anteroposterior, el STEP2 correspondiente a la gravedad actuando de arriba hacia abajo y el STEP3 una carga puntual de un Newton repartida sobre cuatro nodos de la mejilla tirando de ella hacia fuera.

El principal resultado de dicho estudio de sensibilidad fue encontrar al parámetro más significativo: m_2. Una variación de m_2 suponía una mayor variación de la magnitud analizada (desplazamiento) frente a los demás parámetros del modelo.

Además, se concluye que el aumento de m_2 rigidizaba el comportamiento, obteniendo desplazamientos menores, y por tanto la disminución de m_2 flexibilizaba el material. Así, cuando en capítulos posteriores se habla de ajuste de las propiedades por zonas, se refirie a ajustar el parámetro m_2 en las diferentes zonas.

Gracias al resultado obtenido por el estudio de sensibilidad, todas las simulaciones realizadas para el ajuste de propiedades se harán manteniendo constantes los parámetros m_1, q, w y μ_o en todas las zonas de la piel. El valor de m_2 se irá cambiando de simulación en simulación con el fin de obtener un amplio rango de resultados.

Para finalizar con las propiedades, hay que hablar también sobre las asignadas al hueso. En las simulaciones que se usan para el ajuste del m_2 por zonas (con la condición de contorno que se explica en siguiente punto), las propiedades asignadas al grupo de elementos del hueso no influyen en el resultado. Esto se debe a que en estas simulaciones el hueso se define como **RIGID BODY* en ABAQUS, impidiendo así las deformaciones en todo el conjunto. A la hora de simular la cirugía en la mandíbula inferior es necesario mover algunas partes de esa zona del hueso. En ABAQUS no se pueden imponer movimiento en un grupo de nodos si esos nodos pertenecen a un grupo de elementos definido como rígido. Así, mientras que la mandíbula superior y cráneo se siguen definiendo como **RIGID BODY*, sin importar que propiedades tengan, a la mandíbula inferior es necesario darle propiedades. A diferencia de los tejidos, para el hueso se emplea una ley de comportamiento elástica, lineal e isótropa obtenida de la literatura, pudiendo definirse con sólo dos parámetros: módulo de Young y coeficiente de Poisson. Las propiedades asignadas al hueso son: módulo de Young de 17 GPa y un coeficiente de Poisson de 0,3.

3.6 Condiciones de contorno y cargas.

Una vez hablado de las propiedades del modelo, hace falta determinar cuáles serán las condiciones de contorno y cargas necesarias para representar los estados que se intentan reproducir.

En la única zona donde hay que aplicar una condición de contorno es en la zona posterior del modelo. Para todas las simulaciones realizadas a la hora de ajustar las propiedades de la piel se ha optado por fijar todos los nodos de esa zona (tanto de piel como de hueso). Aunque para la piel no es una situación cien por cien realista, se considera que el error que se comete impidiendo los desplazamientos en esa zona tan alejada de la de interés es práctimente despreciable. Al restringir los desplazamientos de algún nodo del hueso (en este caso todos los de la parte poterior) junto con su definición como **RIGID BODY* antes comentada, se consigue que el modelo quede totalmente fijado ante desplazamientos y giros de sólido rígido.

Para determinar las cargas es necesario saber en qué posición se realizó el TAC y en qué situaciones se quieren evaluar las deformadas del modelo. Las situaciones en las que se evalúan las deformadas, tal y como se explica en el siguiente capítulo, son dos: de pie y decúbito prono (tumbado boca abajo).

El TAC se realiza en posición decúbito supino, es decir, acostado boca arriba. Si se le aplica la acción de la gravedad, tal y como muestra la siguiente figura, sobre un eje anteroposterior y en sentido contrario a la gravedad actuante sobre el paciente cuando se realizó el TAC, es como si estuviésemos anulando los efectos gravitatorios sobre la cara.

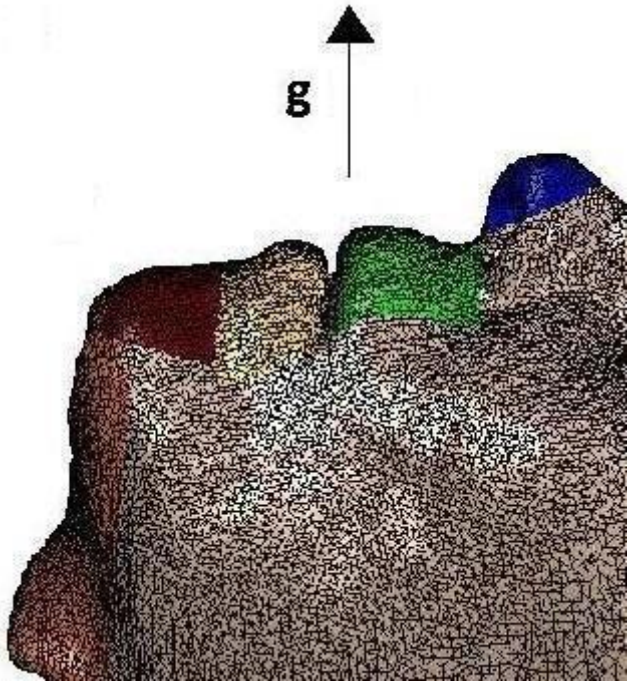

Figura 3.11: Aplicación de gravedad para llevar al modelo a la situación indeformada

Sobre el estado resultante, se deben aplicar dos cargas distintas. Una para simular la posición del paciente de pie, sobre el que actúa la gravedad sobre un eje vertical hacia sus pies:

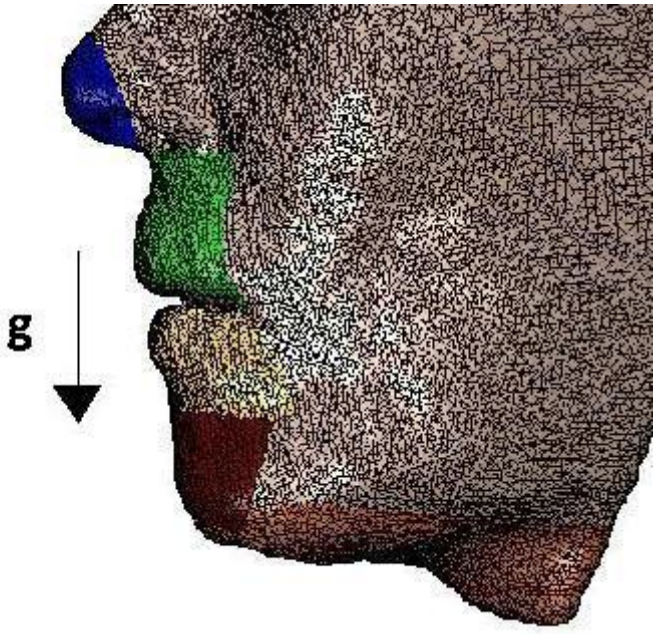

Figura 3.12: Aplicación de gravedad para poner al modelo de pie

La otra simulación necesaria es para representar la posición boca abajo:

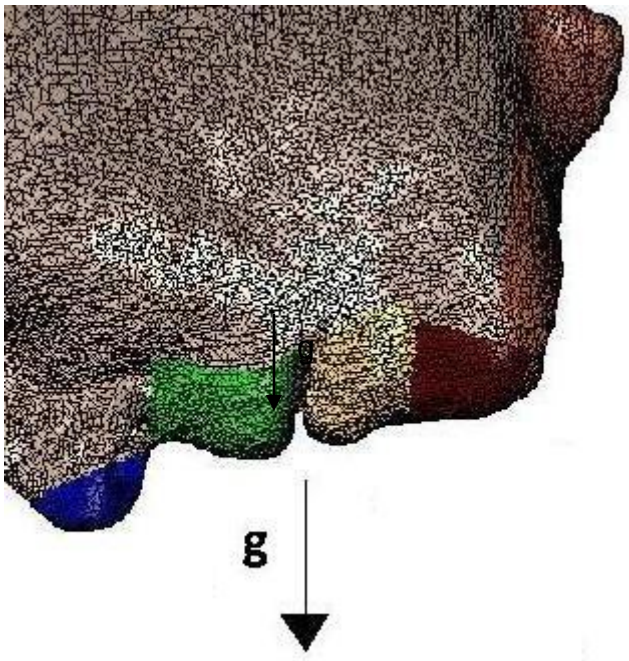

Figura 3.13: Aplicación de gravedad para poner al modelo boca abajo

En definitiva, se debe simular la carga de la primera figura y seguidamente la de la segunda, y luego, simular la carga de la primera figura, seguida de la tercera. Como el modelo de comportamiento es hiperelástico, y parte de la hipótesis de que la disipación mecánica de la función de energía de deformación es nula, el resultado final no depende del camino seguido. Esto significa que es lo mismo aplicar una carga q_1 y

luego otra q_2, que aplicar la suma de las dos directamente q_1+q_2 (principio de superposición), porque el resultado será igual.

De esta manera, se necesitarán dos pasos de carga. El primero de ellos, para conseguir la posición de pie:

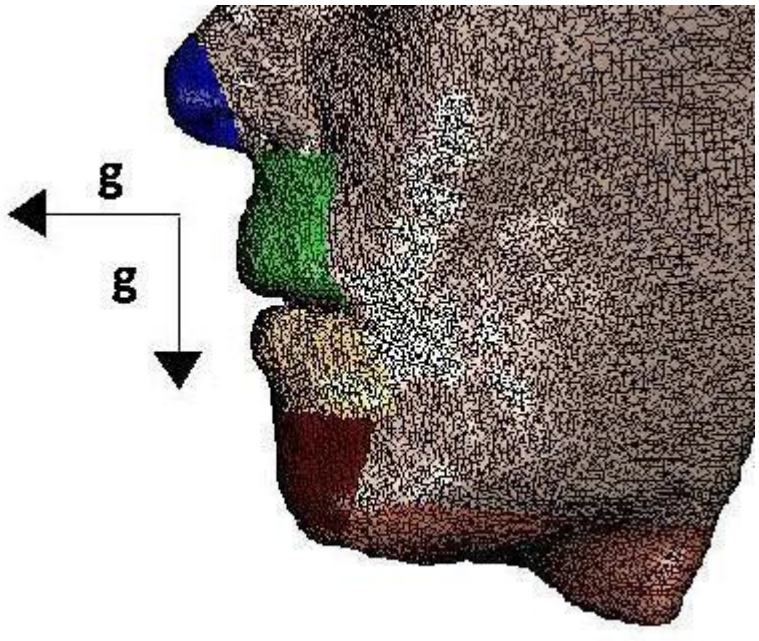

Figura 3.14: Paso de carga 1 (STEP1)

Y el segundo paso de carga, para representar la posición “boca abajo”:

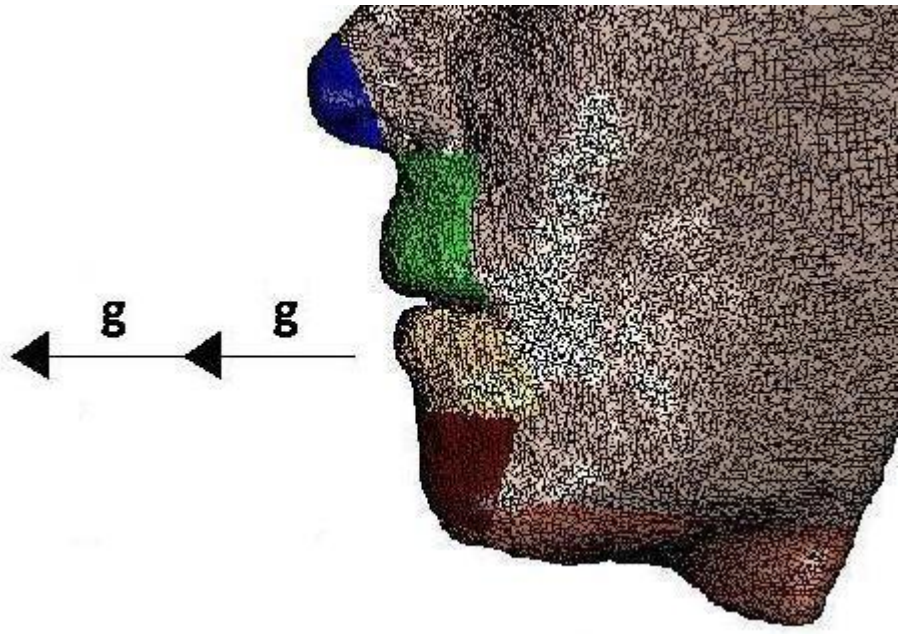

Figura 3.15: Paso de carga 2 (STEP2)

La aplicación de la carga gravitatoria en ABAQUS se hace mediante la opción *GRAV* del comando **DLOAD*. Para emplearlo, se debe definir sobre qué elementos actúa. En este caso, sobre los elementos del grupo creado llamado PIEL, que contiene todos los elementos pertenecientes a tejido blando.

En cuanto a las unidades de las variables con dimensiones que se introducen en los archivos de carga, decir que hay que expresarlas todas en el Sistema Internacional. Por esta razón la definición de los nodos se hizo en metros; la gravedad es definida en m/s^2 y el módulo de Young para el hueso y el parámetro μ de la ley de comportamiento de Rubin-Bodner en N/m^2 (Pa). El empleo de unidades del sistema internacional es necesario para poder superponer en ABAQUS cargas gravitatorias con otro tipo de cargas: puntuales, de presión, dinámicas, etc.

Capítulo 4:

SIMULACIONES Y RESULTADOS OBTENIDOS; AJUSTE MANUAL

En este capítulo se describen los pasos seguidos para ajustar las propiedades de la piel. Lo primero que se hace es obtener los desplazamientos reales que experimenta la piel de la cara en los casos que se van a simular. Para ello es necesario haber realizado, tal y como se explica a continuación, los escaneados faciales al paciente. Posteriormente se realizan una serie de simulaciones y, a partir de estas, se realiza un ajuste manual del parámetro m_2 por zonas. En el último aparatado del capítulo se comprueba cómo afecta a los resultados la variación de otro parámetro de la ley de comportamiento de Rubin-Bodner distinto de m_2. También se explican los motivos de por qué el parámetro elegido ha sido m_1.

4.1 Escaneados faciales.

Antes de comenzar a correr simulaciones de carga en ABAQUS, para poder comparar resultados y así validar el modelo 3D creado, es necesario conocer los desplazamientos reales que se producen con las cargas que se simulan. Para estimar estos desplazamientos se recurre a la técnica de los escaneados faciales.

Esta técnica consiste en una cámara que es capaz de obtener un modelo tridimensional para un programa de CAD, de una superficie real, que en este caso es la cara del paciente. Dicha imagen tridimensional viene en un formato conocido como ".stl" de estereolitografía. Este tipo de archivo utiliza una malla de pequeños triángulos sobre las superficies para definir la forma del objeto. El procesado de estos archivos se llevó a cabo sobre la versión gratuita del programa NETFABB, dedicado exclusivamente al trabajo sobre este tipo de archivos.

Una de las lecciones aprendidas del proyecto de José Manuel es que es necesario establecer un protocolo que normalice las condiciones de realización de los escaneados faciales. De esta forma se intentan disminuir los errores de partida en la medida de los desplazamientos. Los escaneados deben hacerse de la siguiente manera:

- Posición primera:
    - De pie.
    - Cuerpo y cuello rígidos.
    - Cabeza levantada y enderezada, con la mirada hacia el frente.
- Posición segunda:
    - Inclinar el cuerpo hacia delante, intentando formar un ángulo recto entre piernas y torso.
    - Cuello rígido y cabeza enderezada sobre los hombros, intentando mantener la línea recta que forma con el torso y paralela al suelo.
    - Igualmente la mirada hacia el frente, que en este caso sería mirando hacia el suelo.
- Para adoptar esta segunda posición es posible ayudarse de algún medio de apoyo, con el fin de conseguir que el plano que forma cabeza-cuello-torso sea paralelo al suelo.
- En ambas posiciones se debe evitarse estar en tensión, relajando, en la medida de lo posible, los músculos de la cara. Por ejemplo, para mantener la boca cerrada actúan varios músculos, aunque se haga de manera inconsciente. Para

evitar esta situación se pide al paciente que deje los labios ligeramente separados (de la misma manera que se hizo cuando se realizó el TAC).

- Se debe intentar adoptar la forma más natural posible como consecuencia de la acción gravitatoria.
- Para realizar la captura la cámara debe situarse en un plano paralelo a la superficie de la cara.

La siguiente imagen, sacada directamente del programa NETFABB, muestra los escaneados una vez girados de la posición de pie (en rojo) y boca abajo (en verde).

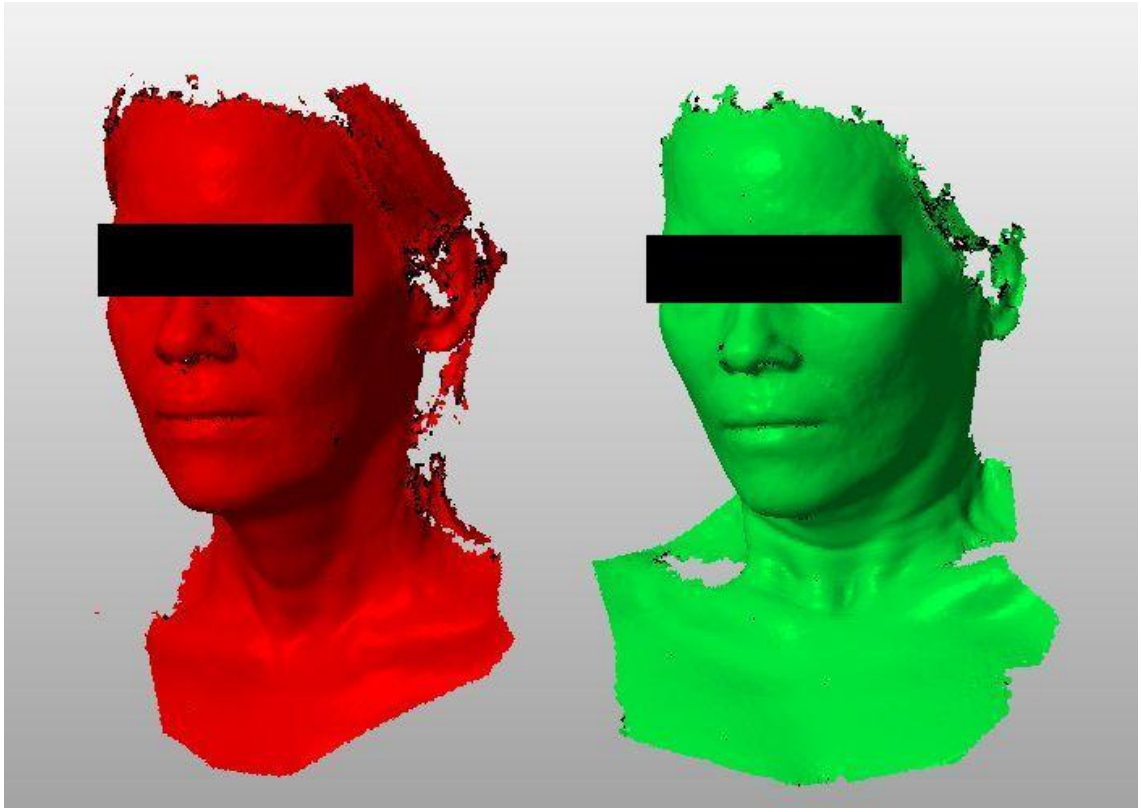

Figura 4.1: Escaneados faciales del paciente

Para poder medir las diferencias entre los desplazamientos que se producen en uno y otro caso, hay que superponer las dos imágenes anteriores, mediante una serie de giros y desplazamientos en cada una de ellas. En este proceso se hace la hipótesis de que los desplazamientos son nulos en la frente y parte superior de la nariz. Esto no es completamente cierto porque realmente sí hay desplazamientos en esas zonas, pero al ser mucho más pequeños que los que se dan, por ejemplo, en los labios, se desprecian frente a estos últimos. Por tanto, la superposición obtenida tiene que ser tal que se ajuste lo mejor posible a la zona donde se suponen casi nulos los desplazamientos, de forma que se puedan evaluar los desplazamientos producidos en

otras regiones. Una vez se tiene el archivo en este estado, se pasa a obtener el corte sagital.

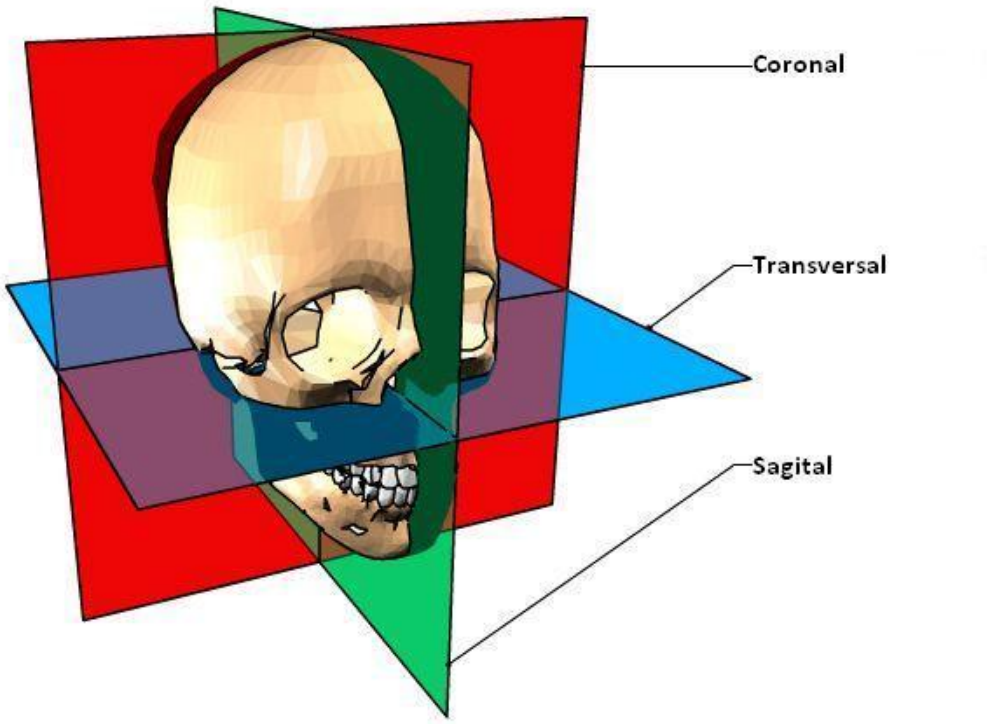

Figura 4.2: Planos anatómicos básicos

Un corte sagital es cualquiera que sea paralelo a un plano sagital. Como puede verse en la figura anterior, el plano sagital es aquel que divide al cuerpo en dos partes: derecha e izquierda. El corte sagital que se quiere es el que divide a la cara justo por la mitad. El corte se realiza directamente con el programa NETFABB, y lo que se obtiene son las dos líneas del perfil: la correspondiente a la situación de pie y boca abajo. En la siguiente la línea roja es la posición de pie y la línea negra la de boca abajo.

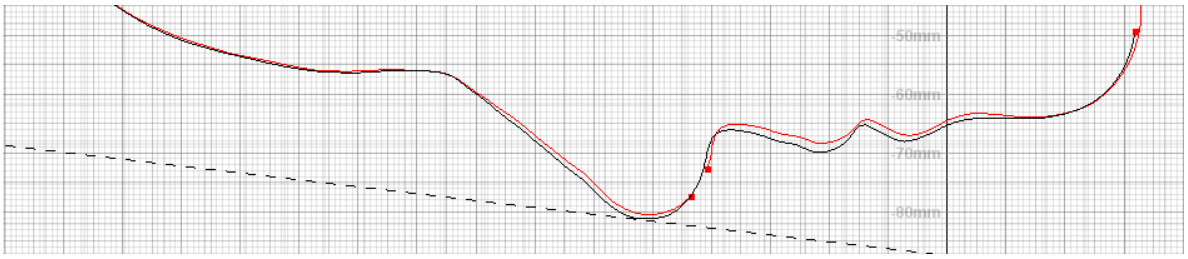

Figura 4.3: Corte sagital de los escaneados faciales superpuestos

Sobre esta imagen se miden las diferencias de desplazamientos en diferentes puntos del perfil. Las zonas donde se indica un punto de medición son: mentón, labio inferior, labio superior, justo debajo de la nariz y en la punta de la nariz. Los desplazamientos se han medido directamente en el programa, restando a los desplazamientos de la situación boca abajo los de la posición de pie.

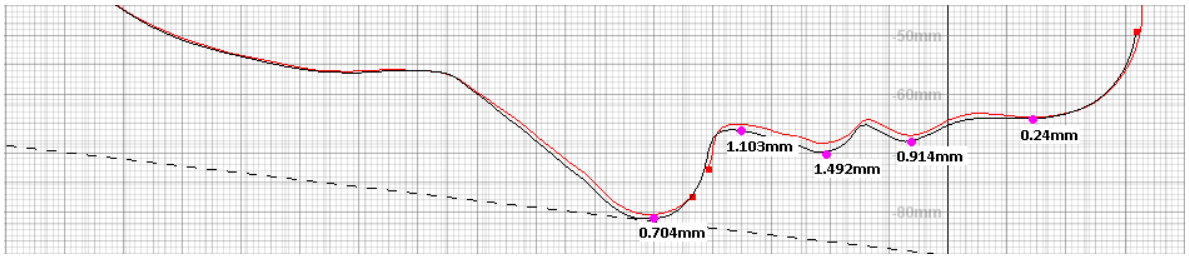

Figura 4.4: Medidas de la diferencia de desplazamientos en los cinco puntos que serán la referencia para el ajuste

Estos valores de desplazamientos obtenidos son los que se intentará ajustar posteriormente sobre los mismos puntos, situados estos, en el modelo tridimensional. Al intentar aproximar los desplazamientos en unos puntos análogos situados en el modelo 3D es lo que se ha llamado ajuste. Dicho ajuste se conseguirá al ir variando las propiedades de los diferentes grupos de elementos en que se ha dividido la piel.

4.2 Simulaciones realizadas; ajuste manual.

A partir del archivo JOB generado por ABAQUS una vez creado el vestíbulo y separada la piel en distintos grupos de elementos, se monta el archivo de carga. En él se incluye la definición de todos los nodos del modelo, la definición de los elementos de PIEL y HUESO, y los grupos de elementos de la piel. Las propiedades de los materiales por zonas, así como las condiciones de contorno y cargas también se especifican directamente en este archivo de carga. En el Anexo III aparecen los comandos del archivo de carga correspondiente a la simulación número 86 (caso importante como se verá a continuación).

Por defecto, la exportación desde ScanFE a ABAQUS define los elementos tetraédricos creados como C3D4. Para el conjunto de la piel este tipo de elementos no es adecuado. Al montar el archivo de carga para simular, hay que sustituirlos por elementos C3D4H. Esta modificación se realiza fácilmente con solo cambiar la nomenclatura en el grupo de elementos PIEL. La necesidad de este cambio se debe al hecho de considerar la piel como un material incompresible. En estos elementos, aparte de los desplazamientos, la presión también es una variable nodal.

Como paso previo antes de comenzar las simulaciones hay que escoger los nodos del modelo en los que se van a evaluar el desplazamiento. Estos nodos son equivalentes a los puntos donde se miden los desplazamientos sobre el corte sagital de los escaneados. Su elección se hace directamente en el modelo en ABAQUS CAE comparando con los puntos de los escaneados faciales donde se ha medido el desplazamiento. Como se puede intuir, no es fácil que los puntos coincidan exactamente, lo que añade una posible fuente de error. En el archivo de carga se incluyen estos nodos para facilitar el proceso de procesar los resultados, del que se habla a continuación. En la siguiente imagen están señalados estos nodos sobre el modelo tridimensional.

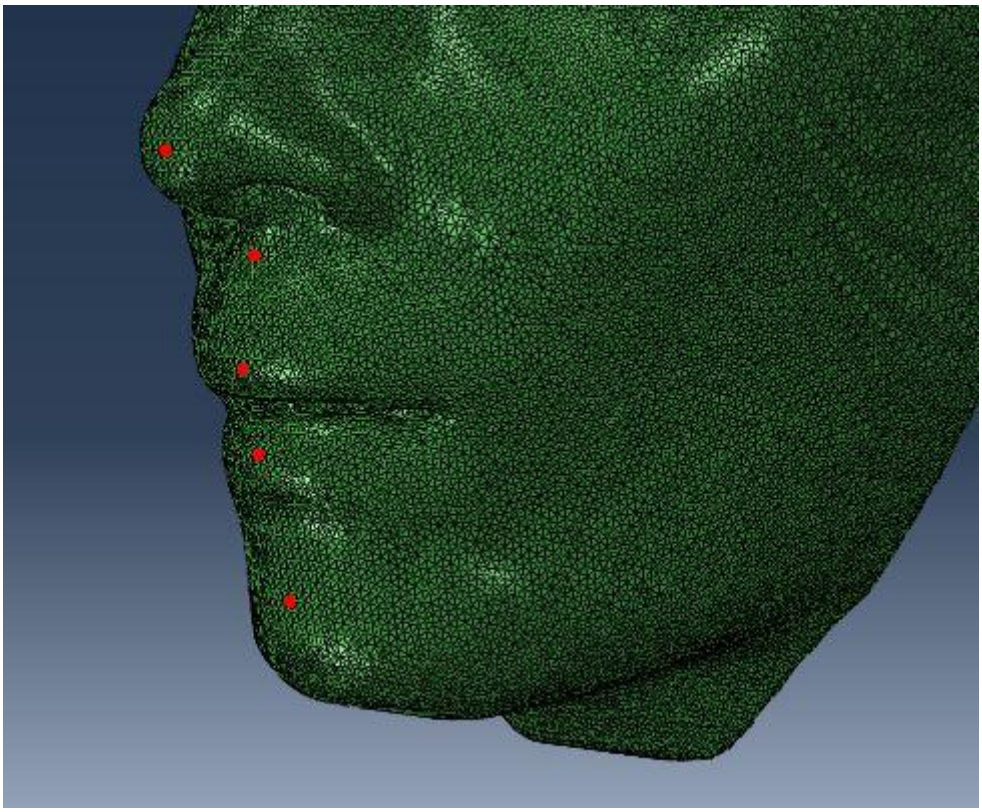

Figura 4.5: Nodos seleccionados para el ajuste del modelo 3D

Antes de hablar de las simulaciones en sí, se va a comentar la forma de extraer los datos de ellas. Los desplazamientos en el modelo de elementos finitos se obtienen como los desplazamientos proyectados en la dirección anteroposterior al final del STEP2 menos los desplazamientos proyectados en dicha dirección al final del STEP1. La dirección anteroposterior coincide con la dirección *y* del modelo, por tanto, los desplazamientos de las simulaciones se tienen directamente de la variable *U2* de ABAQUS.

Como el número de simulaciones que se va a analizar es muy grande, para no tener sacar del archivo de resultados ".odb" la diferencia de los desplazamientos de los

nodos en cada caso de carga manualmente, se hace uso de la opción *Run Script* de ABAQUS CAE. El proceso seguido para obtener los desplazamientos en cada simulación es el siguiente:

- Se abre en ABAQUS CAE el archivo ".odb" que contiene los resultados de la simulación.
- Navegando por las pestañas se pincha en: File/Run Script…
- Se abre el archivo ".py". En él se le está ordenando a ABAQUS que saque de los resultados obtenidos, el valor de los desplazamientos experimentados por el modelo en los nodos escogidos, para cada caso de carga.
- Se genera un archivo ".rpt" que contiene los desplazamientos obtenidos.
- Con una rutina de MatLab se abre el archivo anterior, se leen estos desplazamientos, se comparan con los desplazamientos reales y se calcula el error cometido.

Además de un para un ajuste por zonas manual de las propiedades de la cara, los resultados obtenidos de las simulaciones se van a usar como datos de entrada para entrenamiento en redes neuronales. Por este motivo, los valores de m_2 que se le van asignando a cada grupo de elementos de la piel varían de una forma más o menos regular. Se intenta así que los valores con los que se va a entrenar la red estén repartidos de manera uniforme por todo el rango de valores esperado.

Con el modelo obtenido tal y como se explicó en el capítulo anterior, se han realizado 99 simulaciones. En todas ellas se miden los desplazamientos que se producen, la diferencia que hay respecto a los desplazamientos reales (obtenidos de los escaneados faciales) y se calcula el error cuadrático medio. En el Anexo II se recoge una tabla con todos estos datos.

Con una simple inspección visual de estos datos se puede entender más o menos como afecta el valor de m_2 en cada una de las zonas a los desplazamientos producidos. Así, por ejemplo, rigidizar (aumentar el valor de m_2) el grupo de elementos denominado RESTO supone una disminución de los desplazamientos en todos los puntos del perfil considerados. Esto se debe a que de este grupo de elementos "cuelgan" los demás, y rigidizar este supone disminuir los desplazamientos del resto.

Para seleccionar el rango de valores que se iban a usar en estas simulaciones se tomó como referencia el valor de las propiedades de la ley de comportamiento del material para los tejidos de la cara que propone el artículo de Barbarino. En la siguiente tabla se recogen los valores de las variables que se consideran constantes:

m_1	μ [MPa]	q	w
595	3.7	25	0

El valor de m_2 según Barbarino varía bastante en función del tipo de tejido blando de la cara: desde 0.00007 para la grasa hasta 0.00216 para el músculo. Como se observa en la tabla del anexo, se ha considerado una variación de m_2 de entre 0.0001 y 0.002.

Una vez obtenidos todos estos resultados se realiza el ajuste manual de las propiedades por grupos. Se escogen como punto de partida los valores de m_2 de la simulación número 86. La justificación de esta elección es que se una de las que menos error con respecto a los desplazamientos reales presenta y, además, las rigideces relativas entre grupos de elementos son razonables; las zonas más flexibles de la cara son la papada y gran parte de las zonas que componen el grupo RESTO, como por ejemplo las mejillas.

Grupos de elementos	Valores de m2 de la simulación 86
MENTÓN	0.002
LABIO INFERIOR	0.002
LABIO SUPERIOR	0.002
NARIZ	0.002
PAPADA+CUELLO	0.001
RESTO	0.001

En la siguiente tabla se muestra el valor de los desplazamientos reales y el de los obtenidos en esta simulación.

Desplazamientos medidos [mm]	Desplazamientos en simulación 86 [mm]
0.24	0.180433
0.914	1.37698
1.492	1.933337
1.103	0.951666
0.704	0.503456

Como se ve en la tabla anterior, la mayor diferencia entre unos y otros aparece en la zona de los labios. Por eso, el ajuste manual que se ha llevado a cabo consiste en ir cambiando (aumentando) el valor de m_2 en los labios para ver cómo varía el error. Para el labio superior se encontró rápidamente un límite de m_2 en torno a 0.004 que hacía que los desplazamientos fueran muy parecidos a los reales. En el labio inferior se fue aumentando cada vez más la rigidez pero no se llegó a reducir el desplazamiento hasta un valor cercano al real. Se decidió dejar de seguir aumentando m_2 en esta zona por dos razones: su valor ya era muy grande y la disminución del error que se obtenía

era muy pequeña. Aunque esta discrepancia entre el modelo 3D y la realidad puede deberse a muchos factores, se cree que el que más influye es el error producido al medir los desplazamientos en los escaneados faciales. Tanto los valores de m_2 que se han usado como los resultados de las simulaciones se encuentran en el Anexo II.

El error mínimo alcanzado con el ajuste de las propiedades por zonas está en torno a 0.09 milímetros.

A continuación se representan dos imágenes del modelo tridimensional ajustado, donde se muestran los estados de desplazamientos en la dirección anteroposterior para los dos pasos de carga (de pie y tumbado hacia abajo).

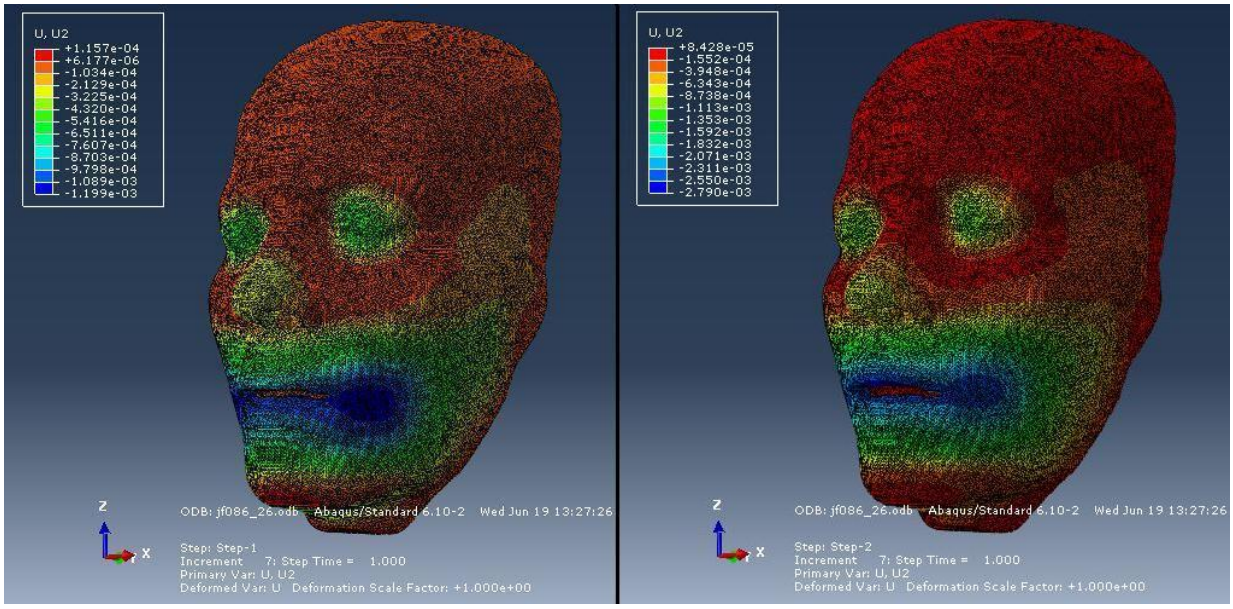

Figura 4.6: Desplazamientos en dirección anteroposterior en los dos casos de carga considerados

Los resultados que se han obtenido al usar estos datos para entrenar y simular redes neuronales se comentan en el Capítulo 7.

4.3 Variación de otro parámetro de la ley de comportamiento.

Gracias al estudio de la sensibilidad que se realizó, el análisis de las propiedades de la piel por zonas se redujo al estudio del parámetro m_2. Aún así, se analizó cómo afectaría a los resultados el variar otros parámetros de la ley de comportamiento de Rubin-Bodner. El parámetro que se decidió cambiar fue m_1, ya que es el que más influye después de m_2. Al igual que antes, se usan como base los valores de m_2 del la simulación número 86, y los demás parámetros se mantienen constantes. Se corrieron dos simulaciones, en una se redujo el valor de m_1 a 400 y en otra se aumento a 800 (el

m_1 en todas las simulaciones anteriores era igual a 595). Se uso el mismo valor de m_1 para todas las zonas de la piel. Los resultados obtenidos son los siguientes:

Punto en el que se evalúan los desplazamientos [mm]	m_1=595	m_1=400	m_1=800
Mentón	0.180433	0.180435	0.180432
Labio inferior	1.37698	1.37698	1.37697
Labio superior	1.933337	1.933334	1.933329
Debajo nariz	0.951666	0.951664	0.951666
Punta nariz	0.503456	0.503456	0.503455

Se observa claramente que al variar el valor de m_1 los desplazamientos son prácticamente idénticos. Esto pone de manifiesto que reducir el estudio de las propiedades del material al de la variable m_2 de la ley de comportamiento es una buena simplificación del problema.

Capítulo 5:

MEJORAS EN EL MODELO

Se recoge en este capítulo el proceso llevado a cabo para implementar algunas mejoras en el modelo de elementos finitos con el que se ha trabajado hasta ahora. Se exponen también los resultados obtenidos con estas modificaciones y se comparan estos resultados con los del modelo inicial. En la parte final del capítulo se describen brevemente varias variaciones, a la hora de crear el modelo de elementos finitos de la cara, que mejorarían los resultados (aunque añadirían complejidad al problema).

5.1 Mejoras en el modelo.

Con el objetivo de mejorar el modelo con el que se han realizado todas las simulaciones anteriores y que se parezca cada vez más a la realidad se han llevado a cabo dos modificaciones él: definición de un contacto entre superficies en el interior del vestíbulo y creación de una membrana en el exterior del grupo de elementos PIEL que simule la piel en sí.

5.1.1 Contacto en el vestíbulo de la boca.

Al haber separado (duplicado los nodos) en el modelo anterior la zona interior de la boca del hueso, podría aparecer en los resultados una superposición entre ellas. Se define un contacto con condición de no penetración en esa zona para evitar que se produzca la situación anterior, claramente irreal.

Para poder definir esta condición, se necesitan definir dos superficies: una en el interior de la boca y otra en la zona exterior de la mandíbula. Se han usado dos formas de crear superficies en ABAQUS: basadas en nodos o basadas en elementos. Cuando se define un contacto en ABAQUS entre dos superficies, cada una de ellas debe actuar o bien con superficie esclava (*slave*) o bien como *master*. La superficie esclava puede estar definida de cualquiera de las dos formas anteriores, mientras que la superficie *master* debe estar creada a partir de los elementos que la componen.

La superficie de la piel, que actúa como superficie *slave*, se define fácilmente, a partir de los nodos que se duplicaron, como una superficie basada en nodos. La superficie del hueso, al actuar como superficie *master*, no se puede crear como una superficie basada en nodos, sino que se tiene que crear como una superficie basada en elementos. La forma de proceder para crear una superficie de este tipo partiendo del modelo es la siguiente:

- Se abre el modelo (a partir de un archivo de carga) en ABAQUS CAE.
- En el módulo ASSEMBLY, navegando por las pestañas se pincha en: *Tools/Surface/Create*.
- Se crea la superficie “by angle”, eligiendo el ángulo más conveniente en cada caso para la superficie que interese.
- En el módulo JOB se crea y escribe el “input” de nuevo modelo.

- En el JOB generado por el programa está la lista con los elementos que forman parte de esa superficie y la cara del elemento que pertenece a ella (cuatro caras por elemento).

- En el archivo de carga se incluye un ELSET con los elementos de la superficie y se define la superficie, especificando en cada caso la cara del elemento que pertenece a ella.

La zona del hueso en contacto con la piel en el vestíbulo está separada en dos (dientes y encías de arriba y de abajo). Como para poder definir una superficie no se puede partir de regiones inconexas, hay que dividir la superficie del hueso en dos, una inferior y otra superior.

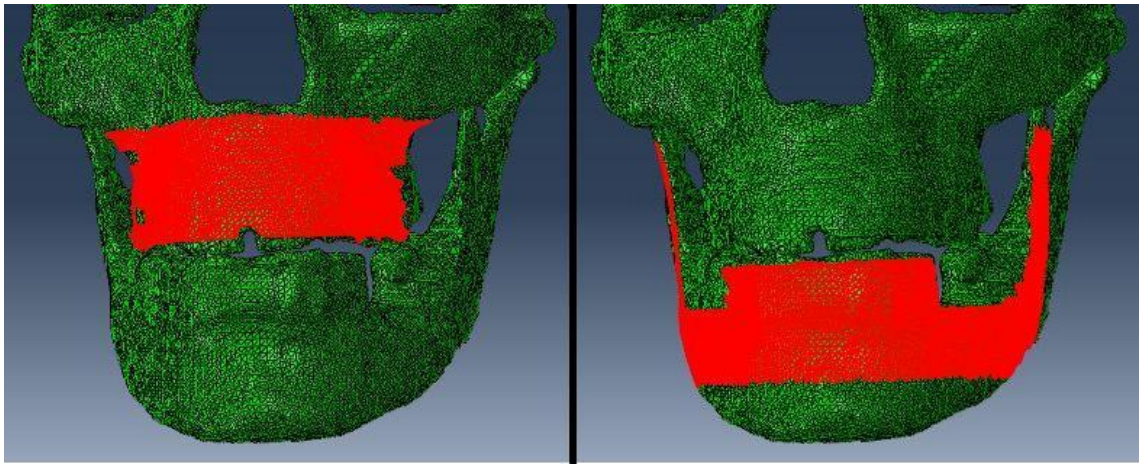

Figura 5.1: Superficies del hueso creadas para definir el contacto

Una vez creadas las superficies, el contacto entre ellas se define directamente en el archivo de carga con los siguientes comandos:

```
*CONTACT PAIR,INTERACTION=INT
SUP_PIEL,SUP_HUESO_SUP
*CONTACT PAIR,INTERACTION=INT
SUP_PIEL,SUP_HUESO_INF
*SURFACE INTERACTION,NAME=INT
*FRICTION
0
```

La interacción se crea sin fricción de forma que se desprecia el rozamiento que pudiera aparecer entre superficies.

5.1.2 Membrana de la piel.

Se crea para simular la rigidez exterior que aporta la piel (epidermis y dermis) al conjunto de tejidos blandos de la cara. Se modela como una capa de espesor constante sobre toda la cara. El espesor utilizado (2 mm) se ha escogido de acuerdo con los datos obtenidos de la literatura (artículo de Barbarino). Este espesor también

coincide con el estimado para la piel a partir de las imágenes de las resonancias magnéticas actuales. Esta membrana es como una máscara de la cara.

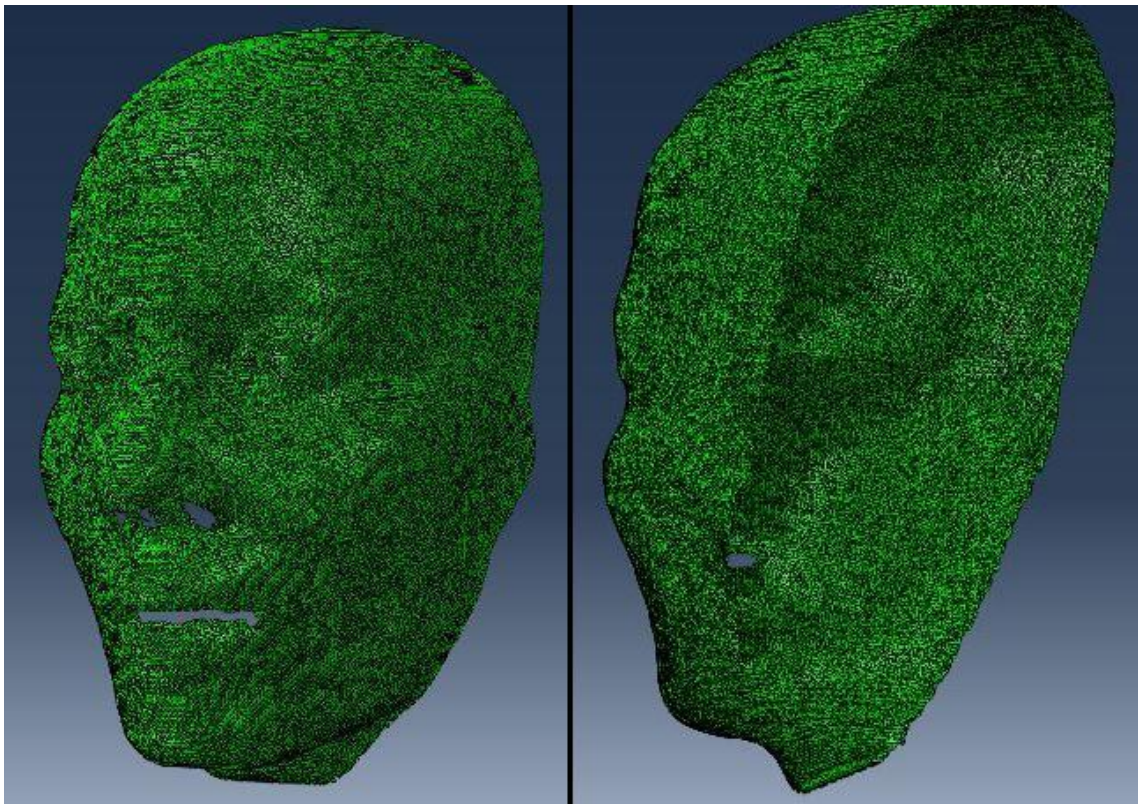

Figura 5.2: Elementos de la membrana de la piel

La creación de esta membrana conlleva añadir nuevos elementos al modelo; elementos tipo membrana. Para definir estos elementos se siguen los siguientes pasos:

- Se crea una superficie basada en elementos tal y como se explica en el apartado anterior.

- En esta superficie es necesario arreglar manualmente algunas imperfecciones que aparecen para obtener la superficie final.

- En este caso, se comprueba al abrir el archivo JOB generado, que la mayoría de los elementos tienen la cara S2 en la superficie (más de 70000 por 18 con otras caras). Por esta razón, y para simplificar, nos quedamos solo con estos elementos.

- Se usa la rutina de FORTRAN “Elset_elementos_membrana.for”, que funciona de la siguiente manera: lee los elementos que tienen la cara S2 en la superficie, lee la definición de todos los elementos de la piel, define los nuevos elementos. La numeración de estos elementos será la misma que tiene el elemento la que pertenece esa cara sumándole 2000000.

- Los nuevos elementos se definen como tipo S3: elementos triangulares definidos por tres nodos.

- En el archivo de carga se define como membrana un grupo de elementos de la siguiente forma:

```
*membrane section,elset=MEMBRANA_PIEL,material=MAT_MEMBRANA_PIEL
0.002
**************************************************************
*material,name=MAT_MEMBRANA_PIEL
*DENSITY
1100,
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
1.7e6,36,1294,0.0008,0.9
```

Las propiedades de la ley de comportamiento de Rubin-Bodner que se le han dado a esta membrana son las que aparecen en el artículo de Barbarino, con un espesor de 2 milímetros:

$\rho\ [\frac{kg}{m^3}]$	m_1	m_2	μ [MPa]	q	w
1100	1294	0.0008	1.7	36	0.9

5.2 Resultados obtenidos y comparación con el modelo inicial.

Igual que se hizo en el capítulo anterior, se coge como simulación base con la que comparar los resultados la número 86. Para apreciar más fácilmente como influye cada uno de los cambios anteriores en los resultados, estos se van a añadir al modelo primero de forma independiente y posteriormente juntos.

En la siguiente tabla se observa los desplazamientos obtenidos con el modelo base y con cada una de las mejoras aplicadas por separado.

Punto en el que se evalúan los desplazamientos [mm]	Modelo base (simulación 86)	Modelo base + Contacto vestíbulo	Modelo base + Membrana piel
Mentón	0.180433	0.176615	0.179935
Labio inferior	1.37698	1.35622	1.3729
Labio superior	1.933337	1.918726	1.924409
Debajo nariz	0.951666	0.945994	0.948116
Punta nariz	0.503456	0.501811	0.502022

Se comprueba que con ambas modificaciones se reduce el desplazamiento de la piel, aunque una cantidad muy pequeña (del orden de centésimas de milímetro). En el caso del contacto (condición de no penetración) se debe a que en el STEP1, si no se define contacto, se produce una pequeña penetración entre las superficies. Esto hace que la diferencia entre los desplazamientos de los dos STEP sea un poco mayor si no se define el contacto.

El efecto de la creación de una membrana en la piel es rigidizador, como se podía prever. Con el espesor de la membrana y los valores de las propiedades especificados en el punto anterior, la variación que se produce en los desplazamientos respecto al modelo base es muy pequeña. Hay dos opciones en el caso de que se quisiera rigidizar más esta capa exterior: aumentar el espesor o aumentar el parámetro m2 de las propiedades de la ley de comportamiento.

Al introducir en el modelo los dos cambios a la vez se obtienen los resultados de la siguiente tabla. Se observa que las diferencias respecto al modelo base siguen siendo muy pequeñas.

Punto en el que se evalúan los desplazamientos [mm]	Modelo base (simulación 86)	Modelo base + Contacto vestíbulo + Membrana piel
Mentón	0.180433	0.176084
Labio inferior	1.37698	1.352
Labio superior	1.933337	1.909752
Debajo nariz	0.951666	0.942548
Punta nariz	0.503456	0.500322

Esta variación ínfima de los desplazamientos al introducir las modificaciones en el modelo pone de manifiesto que los resultados de todas las simulaciones anteriores (sin haber aplicado ninguna de estas mejoras) son válidos. La razón principal por la que las simulaciones se realizaron sin estos cambios es el coste computacional. Al definir la membrana de la piel y, sobre todo, el contacto en el vestíbulo, el tiempo que se necesitaba para resolver el problema aumenta bastante. Puesto que se iban a correr muchas simulaciones, y sabiendo que la diferencia en los resultados obtenidos era mínima, se optó por no introducir estas mejoras en ellas.

5.3 Posibles mejoras a implementar en el futuro.

Aparte de las dos modificaciones anteriores, el modelo se puede seguir mejorando de muchas otras formas. A continuación se explican brevemente tres de ellas.

5.3.1 División de la cara en más zonas.

En nuestro caso, los tejidos blandos del modelo se dividieron en seis zonas (MENTÓN, LABIO INFERIOR, LABIO SUPERIOR, NARIZ, PAPADA+CUELLO y RESTO). Una división mayor aumentaría la precisión a la hora de ajustar las propiedades de cada zona. Un ejemplo claro de ello es la zona resto, pues está claro que las mejillas y la frente no presentan las mismas propiedades. La razón principal por la que se decidió escoger las seis zonas antes citadas y no más es la complejidad del problema. Aumentar las zonas significa aumentar los valores de m_2, lo que dificulta enormemente el ajuste. Puesto que la región que se pretende ajustar está situada alrededor de un corte sagital que divida la cara en dos mitades, se decidió que lo ideal era que las diferentes zonas estuvieran repartidas mayormente por esa zona. Además se intentó que estas regiones escogidas fueran lo más representativas posibles en el movimiento de la piel.

Como conclusión se puede decir que, aunque aumente la dificultad del problema, para un ajuste mejor de las propiedades de los tejidos blandos, es necesario dividir la cara en un mayor número de zonas.

5.3.2 Ajustar muestras de tejidos experimentalmente.

Cabría la posibilidad de ajustar muestras de tejidos experimentalmente, para asignar posteriormente propiedades a las diferentes regiones. El problema que se presenta es la validez de los valores obtenidos. Puesto que el propósito que tiene la creación del modelo, es el de simular una cirugía en un paciente, las propiedades que se quieren ajustar son las de ese paciente en particular. Por lo tanto muestras de tejidos muertos solo dan una indicación de por dónde van a estar los valores que se buscan.

5.3.3 Segmentación más detallada de la cara.

Otra manera alternativa de plantear el problema de ajuste, sería realizando la segmentación de los músculos, grasa, piel, etc. pero para ello sería necesaria una resonancia magnética, ya que con los niveles de grises del TAC no es posible realizar una segmentación de estas zonas, no se consiguen distinguir. El software utilizado (SIMPLEWARE) también tiene la posibilidad de trabajar con archivos de resonancia magnética. De esta manera el ajuste se haría conforme a estructuras reales.

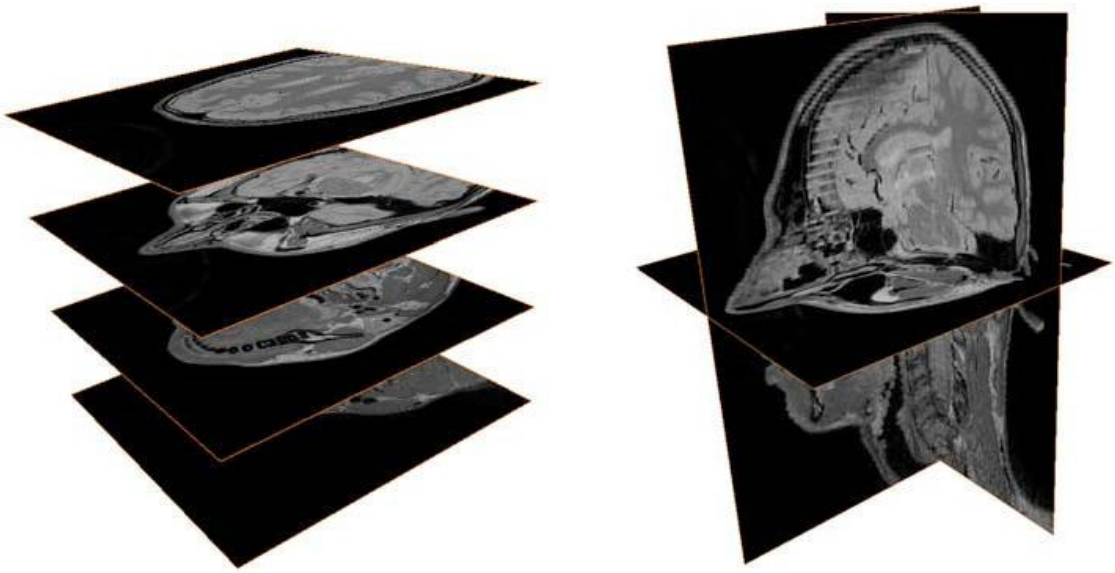

Figura 5.3: Ejemplos de imágenes de resonancias magnéticas

Capítulo 6:

SIMULACIÓN DE LA CIRUGÍA

El objetivo principal del proyecto es llegar a simular una cirugía ortognática al modelo de elementos finitos de un paciente. Antes de simular la operación es necesario tener un modelo lo más realista posible de la cara, razón por la cual se ha hecho todo lo anterior. Se describe en este capítulo en qué consiste una operación de cirugía ortognática, cuál es el caso en el que nos vamos a centrar y cuáles son las distintas pruebas y resultados que se han obtenido.

6.1 Cirugía ortognática.

La cirugía ortognática es el procedimiento realizado por cirujanos orales y maxilofaciales para colocar los huesos de la cara y los dientes en su posición óptima cuando estos se encuentran desplazados o alterados. De este modo, se mejora la apariencia de la cara, a la vez que se garantiza que los dientes funcionen de manera correcta y saludable. Se quiere lograr la mejor oclusión en la cara más bonita.

La cirugía ortognática no está dirigida sólo a aquellos pacientes que tienen una deformidad facial importante, sino también a todos aquellos que tienen una forma de morder alterada por una posición inadecuada de los dientes y huesos maxilares que el ortodoncista no puede resolver de forma aislada. Es decir, este tipo de cirugía permite solucionar todos aquellos problemas generados por un crecimiento irregular de los huesos de la cara y que ocasionan problemas a la hora de morder, hablar, dormir y en el aspecto estético.

Hay dos grandes motivos para decidir operarse: el estético, mejorando el aspecto general de la cara y de la sonrisa, y el funcional al lograr así una correcta forma de masticar y ayudar a mantener una buena salud de las piezas dentales, encías y articulaciones temporomandibulares. Esta cirugía también se ha mostrado muy eficaz a la hora de resolver otros problemas como la apnea obstructiva del sueño.

Antes de la cirugía, es necesaria la ortodoncia durante un periodo breve de tiempo, ya que a menudo los dientes están descolocados y no encajarían al colocar los maxilares en su sitio. Una vez finalizada la ortodoncia, el cirujano maxilofacial colocará los maxilares correctamente en oclusión, lo que mejorará la función oral y la masticación, pero además lo hará en el lugar más idóneo para que la cara y la sonrisa recuperen la armonía. Los resultados son visibles de forma inmediata. Esta cirugía no deja cicatrices visibles. La mayoría de los pacientes regresa a su casa dos o tres días después de la intervención, y pueden volver a su vida normal a los quince días aproximadamente, si bien es cierto que pueden hablar y comer desde el primer día.

Hasta ahora el planteamiento de la operación es tal y como se explica a continuación. De cada paciente se realiza un estudio radiológico en tres dimensiones, la toma de fotografías digitales y modelos de sus dientes. Toda esta información se digitaliza a fin de poder determinar qué parte de la cara no armoniza con el resto y qué tipo de tratamiento es el más adecuado. Finalmente se establece un plan combinado de ortodoncia y cirugía que consiga la colocación perfecta de los dientes.

Este proceso está basado básicamente en la experiencia de operaciones anteriores, y no hay manera de estar seguro de cómo va a quedar la cara del paciente después de la operación. El objeto de este proyecto es estudiar la forma de poder simular de manera virtual el tratamiento que se le quiere aplicar a cada paciente. Ya que las características de todas las pieles no son iguales y todas las caras no tienen la misma

estructura, la manera de deformarse de cada una de ellas será distinta. Antes de simular la operación es necesario encontrar las propiedades de la piel de cada paciente. Al aplicar estas propiedades al modelo 3D y simular el tratamiento que se ha decidido es el más adecuado, la piel se deformará de manera muy similar a como lo va a hacer en la realidad. De este modo se podrá ver el resultado de la operación en el paciente antes de llevar a cabo la cirugía.

En la siguiente imagen se representan algunos casos para los que se práctica este tipo de cirugía.

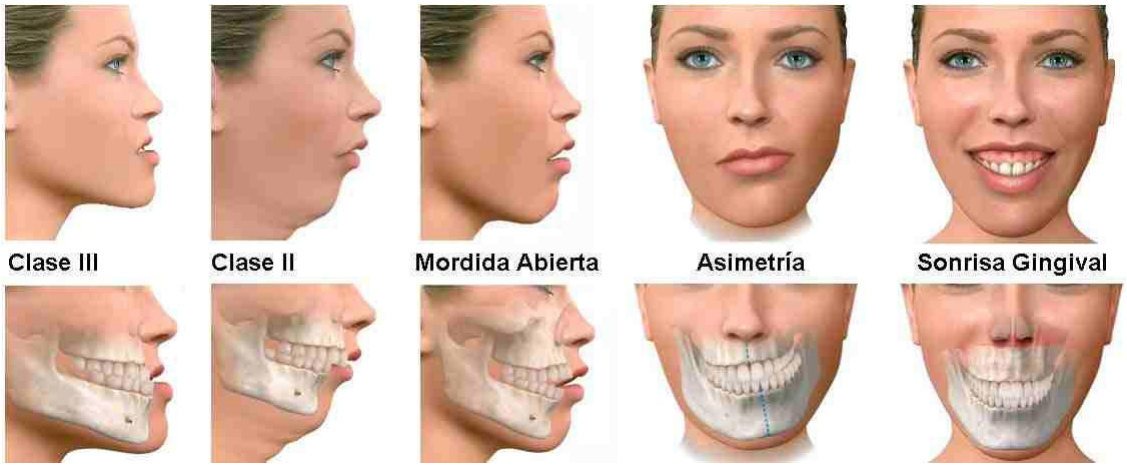

Figura 6.1: Casos de aplicación de la cirugía ortognática

A continuación se describen brevemente cuáles son los problemas más comunes:

- Mandíbula pequeña o retrasada: los dientes inferiores están retrasados con respecto a los superiores, al igual que el mentón. Suele haber problemas en la articulación de la mandíbula y puede haber dificultad de respirar durante el sueño.

- Maxilar superior retrasado: los pómulos están aplanados, los dientes superiores están retrasados y casi no se ven al sonreír.

- Maxilar superior alargado: la encía se ve de forma excesiva al sonreír. La cara es alargada, el mentón suele estar retrasado y cuesta trabajo cerrar los labios.

- Mandíbula grande o adelantada: los dientes inferiores están adelantados respecto a los superiores, la barbilla está adelantada y las muelas se estropean más de lo normal por el exceso de trabajo.

- Mordida abierta: los dientes superiores e inferiores no tocan al cerrar la boca, dejan un espacio. Hay que forzar los labios para poder cerrarlos y es difícil de comer. En casos extremos puede afectarse el habla con problemas para pronunciar algunas letras. Las muelas que sí contactan tienen un exceso de

trabajo, por lo que también se estropean antes y suele haber dolor en la articulación de la mandíbula.

- Asimetría mandibular: el mentón y los dientes se debían hacia un lado. De forma inconsciente el paciente tiende a girar la cabeza para disimular el efecto, por lo que, en ocasiones, hay dolor en el cuello. También puede haber problemas en la articulación.

El caso que se intenta simular en este proyecto es el primero de los puntos anteriores. La solución a este problema es un alargamiento de la mandíbula hasta hacer coincidir de manera correcta los dientes inferiores con los superiores. Esto se va a simular imponiendo a ciertos nodos de la mandíbula del modelo unos desplazamientos.

6.2 Pruebas con el modelo actual (elementos C3D4).

Se detallan a continuación los diferentes intentos que se llevaron a cabo (y en el orden que se hicieron) para simular la cirugía antes comentada.

- Con el modelo completo, separando la piel de interior de la boca y definiendo contacto entre las superficies internas.

Partiendo del modelo completo, con la condición de no penetración entre piel y hueso en el vestíbulo, antes de simular la cirugía se prepara el modelo, haciendo algunas modificaciones:

1. Se divide el hueso en varios grupos de elementos, al igual que se hizo con la piel. Se crean cuatro grupos de elementos: cráneo y mandíbula superior, zona de la articulación temporomandibular y parte posterior de la mandíbula, parte anterior de la mandíbula, y zona intermedia entre las dos anteriores. El primer y el segundo grupo de elementos se definen como RIGID BODY en ABAQUS, de tal manera que se impide cualquier movimiento en esos nodos. El cuarto grupo de elementos se define como ADAPTIVE MESH. Al grupo de elementos de la parte anterior de la mandíbula no se le define de ninguna manera especial, de forma que se permite su movimiento. La definición de un grupo de elementos como ADAPTIVE MESH hace que, según se va avanzando en la simulación, el programa puede remallar esa zona del modelo. En zonas del modelo en las que los elementos puedan distorsionarse mucho, ayuda a evitar problemas de incompatibilidad, definiendo una malla más fina. (Como después se comenta, la condición de ADAPTATIVE MESH no es válida en los elementos que aquí se tienen.)

La cirugía que se simula es un alargamiento de la mandíbula inferior. Se escoge el lugar por donde se parte y alarga la mandíbula tal que este dentro de la zona definida como ADAPTIVE MESH. Si no se hiciera así, al resolver en ABAQUS aparecerían problemas de incompatibilidades en esa zona.

2. Se "separa" la mandíbula inferior de la piel. Esto significa que los nodos comunes se duplican y la piel puede moverse independientemente de la mandíbula inferior.

Con esta separación se intenta conseguir que la piel se adapte de manera suave, y de una forma más parecida a la realidad, a la nueva mandíbula.

3. Se crean superficies en la zona de contacto entre mandíbula inferior y piel. Una vez creadas, se define un tipo de contacto entre ellas tal que no puedan traspasarse entre sí.

Esta condición de no penetración es básica. La parte de la mandíbula anterior, al desplazarse hacia delante, empuja la piel de la zona inferior del labio y esta arrastra al resto de la piel.

Una vez realizados estos cambios en el modelo se simuló la operación en ABAQUS, probando diferentes desplazamientos de los nodos de la mandíbula (desde 1 a 10 mm), y definiendo como **ADAPTATIVE MESH* distintas zonas de la mandíbula inferior. Todas las simulaciones daban error: "menor incremento de tiempo requerido". Observando con detenimiento los archivos creados por ABAQUS al simular el archivo de carga, se comprueba que al resolver, en ningún caso el programa ha tomado ninguna región como **ADAPTATIVE MESH*. Por esta razón se cree que el error que aparece se debe a los contactos definidos entre superficies.

6.2.1 Modelo antes de definir el vestíbulo.

Se prueba ahora con el modelo más básico, en el que ni siquiera se ha definido el vestíbulo. Se separa el hueso en dos partes: cráneo más mandíbula superior y mandíbula inferior. La primera de ellas se define como **RIGID BODY*. Se fijan dos nodos en la zona de la articulación temporomandibular y a la zona anterior de la mandíbula inferior se le impone un desplazamiento hacia delante de hasta 1 cm. Este caso de cargas sí que lo resuelve, pero, al estar la piel y el hueso unidos en el vestíbulo, no se produce un deslizamiento relativo y la piel no se deforma libremente como ocurre en la realidad. Además se deforma prácticamente toda la mandíbula, cuando lo ideal sería que la zona deformada sea lo más pequeña posible.

6.2.2 Modelo con vestíbulo pero sin contacto entre sus superficies.

Este caso es igual que el anterior solo que con el vestíbulo definido. Al no haber establecido un contacto entre las superficies del vestíbulo, aparece una penetración de

la mandíbula inferior en la piel. En este caso también resuelve desplazamientos de hasta 1 cm, con el mismo problema de que se deforma casi toda la mandíbula inferior. Tanto en este caso como en el anterior, si se considera que los desplazamientos que aparecen en un corte sagital que divida la cara en dos mitades están lo suficientemente alejados de la zona que no se ajusta a la realidad (zona posterior de la mandíbula inferior), se podría decir que la simulación ajusta bien esa zona de la cara. Que se puedan despreciar las distorsiones que aparecen en la parte de atrás de la mandíbula inferior esta por comprobar.

Para mejorar los dos casos anteriores, en lugar de fijar únicamente dos nodos de la articulación de la mandíbula inferior, se podrían fijar también algunos nodos de la zona inferior. De esta manera se intenta que la deformación que aparece en la mandíbula inferior se concentre en la parte delantera de la misma, acercándose cada vez más a la realidad. En la siguiente imagen están marcados tanto los dos nodos que se fijaban antes (justo en la articulación) como los dos nuevos en el inferior de la mandíbula.

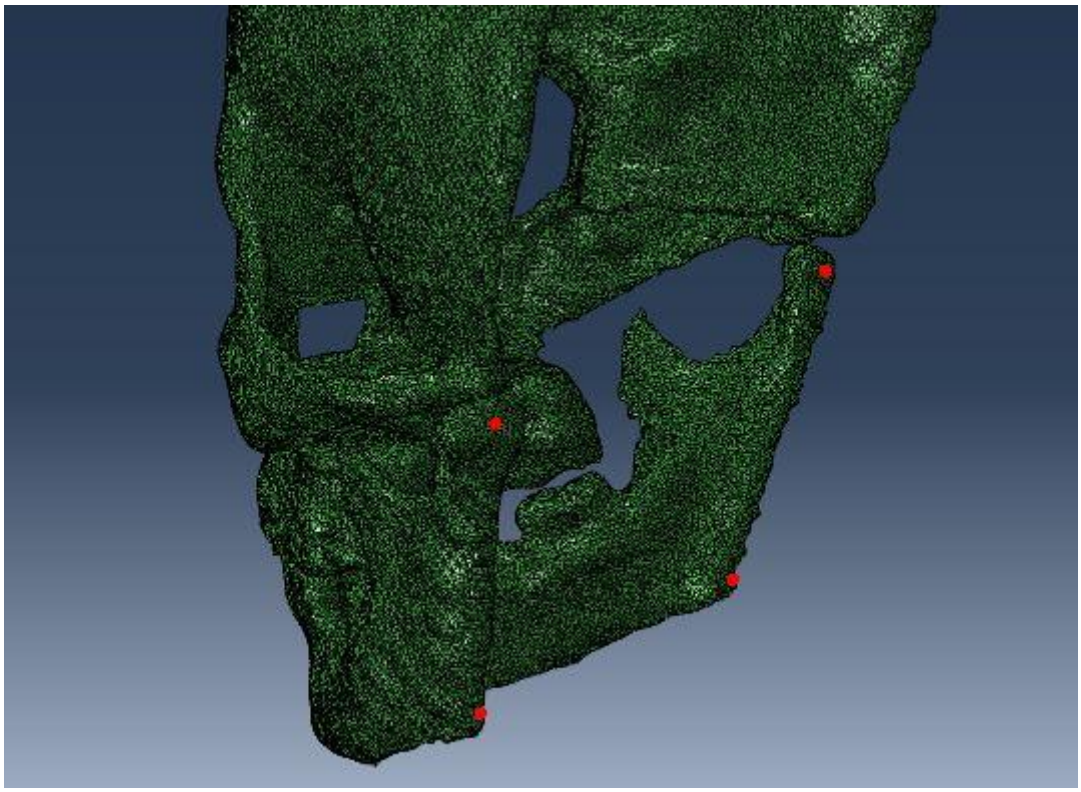

Figura 6.2: Nodos a los que se restringen los desplazamientos

Sin embargo, al fijar estos nuevos puntos el programa ya no es capaz de resolverlo. Aparece un error en ABAQUS que nos indica que se está produciendo una distorsión excesiva en los elementos del hueso. Este problema se intenta resolver definiendo como **ADAPTIVE MESH* la mandíbula inferior, de nuevo sin éxito. El programa no reconoce ningún elemento del modelo como **ADAPTIVE MESH*.

6.2.3 Modelo con vestíbulo y condición de no-penetración entre sus superficies.

Al modelo anterior se le añaden las superficies interiores del vestíbulo y se define el contacto entre ellas. De esta manera se evita que el hueso pueda penetrar en la piel. Al intentar resolver aparece el problema de "menor incremento de tiempo requerido", mismo problema que aparecía en el primer caso. No llega a resolver ni desplazamientos de 1 milímetro. Dado que sin definir el contacto entre superficies sí que resuelve y con contacto no, se deduce que debe haber algún error en la definición del contacto.

En todos los archivos de carga anteriores hay dos variables que se pueden incluir o no, el contacto en el vestíbulo y la zona de **ADAPTIVE MESH*. Dado que no se conoce cómo actúan exactamente ninguna de ellas ni cómo hay que tratarlas, a continuación se van a estudiar por separado para intentar aprender su funcionamiento en ABAQUS y cómo hay que trabajar con ellas.

6.3 Estudio del contacto.

El tipo de contacto que se ha definido hasta ahora, tal y como se dijo en el capítulo anterior, es un simple **CONTACT PAIR* sin fricción. Consultando con otros miembros del departamento cómo mejorar la definición del contacto se paso a la siguiente definición:

```
*CONTACT PAIR,INTERACTION=INT
SUP_PIEL,SUP_HUESO_SUP
*CONTACT PAIR,INTERACTION=INT
SUP_PIEL,SUP_HUESO_INF
*SURFACE INTERACTION,NAME=INT
*SURFACE BEHAVIOR,AUGMENTED LAGRANGE
*FRICTION
0.0001
```

Además de esta nueva definición, en los comandos que hay que escribir dentro del paso de carga se añade lo siguiente:

```
*CONTACT CONTROLS, ABSOLUTE PENETRATION TOLERANCE=0.0001
```

Se ha sustituido el valor de la fricción igual a cero por uno muy pequeño, pero distinto de cero. Esto, aunque en el resultado final no influye mucho, si que evita pequeños problemas que podrían surgir al trabajar con un valor igual a cero.

La otra modificación realizada es el cambio de la tolerancia de penetración que ABAQUS usa por defecto. Para que el programa tenga en cuenta este cambio de tolerancia es necesario definir el comportamiento de las superficies como *AUGMENTED LAGRANGE*. Una vez hecho esto, en el cada paso de carga se puede

variar esta tolerancia. Se puede variar tanto la tolerancia absoluta como la relativa (ABSOLUTE o RELATIVE). El programa por defecto usa una tolerancia relativa del 5%. Este parámetro hace referencia al cociente entre la penetración permitida y la dimensión característica de los elementos en contacto. En este caso lo que se quiere variar es la penetración absoluta, lo que se hace con el comando anterior. El valor que se escribe es la penetración permitida que se desea, en este caso 0.1 milímetro.

Con estas modificaciones en el contacto, el programa consigue resolver desplazamientos de hasta tres milímetros (fijando únicamente dos puntos en la zona de la articulación temporomandibular). Los problemas que aparecen ahora al imponer desplazamientos mayores ya no son errores por máxima penetración, sino errores por máxima fuerza de contacto.

Los desplazamientos que aparecen en este caso se ajustan más a la realidad, puesto que la penetración entre piel y hueso ya no aparece. El problema sigue siendo que los máximos desplazamientos que se consiguen son aún pequeños comparados con los de las cirugías que se pretenden conseguir (del orden de 1 centímetro).

En la mandíbula inferior siguen apareciendo las deformaciones de los casos anteriores. Este problema es el que se intenta solucionar con el remallado *(*ADAPTIVE MESH*).

6.4 Estudio del comando **ADAPTIVE MESH*.

Aunque en algunos casos de carga se definieran grupos de elementos como **ADAPTIVE MESH*, de los archivos que genera ABAQUS se comprueba que no ha tenido en cuenta esta propiedad. El mensaje que aparece para todos los elementos es el siguiente:

```
**WARNING: ELEMENT 1 IS NOT SUPPORTED FOR ADAPTIVITY, IT WILL BE
REMOVED FROM THE ADAPTIVE MESH
```

De aquí se deduce que la razón por la que no está funcionando como se espera puede ser que al tipo de elementos que se están usando (C3D4) no se pueda aplicar un remallado. Primero se busco en la documentación de ABAQUS ejemplos en los que se usara el mallado adaptativo, pero no se encontró ninguno con elementos C3D4. Se optó por crear dos geometrías sencillas: una con elementos C3D4 y otra con elementos C3D8 (para los que sí hay ejemplos). Al imponer unos desplazamientos en cada uno de los casos, con los elementos C3D4 sigue apareciendo lo mismo que antes. Sin embargo, con los desplazamientos C3D8 no sale esta frase y se comprueba, resolviendo una simulación con **ADAPTIVE MESH* y otra sin ella, que esta opción influye en el resultado final.

Con esto se comprueba que el problema por el que en el modelo que tenemos no funciona el **ADAPTIVE MESH* es el tipo de elementos del modelo.

6.5 Modelo de elementos C3D8: creación del modelo, simulaciones y problemas encontrados.

Para crear un modelo con elementos C3D8 (elementos hexaédricos) partimos de la segmentación del TAC que se hizo en ScanIP y la posterior exportación a ScanFE. En este último modulo de SIMPLEWARE, para obtener el modelo de elementos tetraédricos había que hacer un remallado y suavizado del modelo. Para obtener el modelo solo con elementos C3D8, en ScanFE no hay que hacer nada, solo exportar el modelo a ABAQUS sin escoger las opciones de "Elementos tetraédricos" ni "Elementos de mayor orden" que aparecen al exportar. Se consigue así un modelo con elementos cúbicos C3D8 tal y como se buscaba.

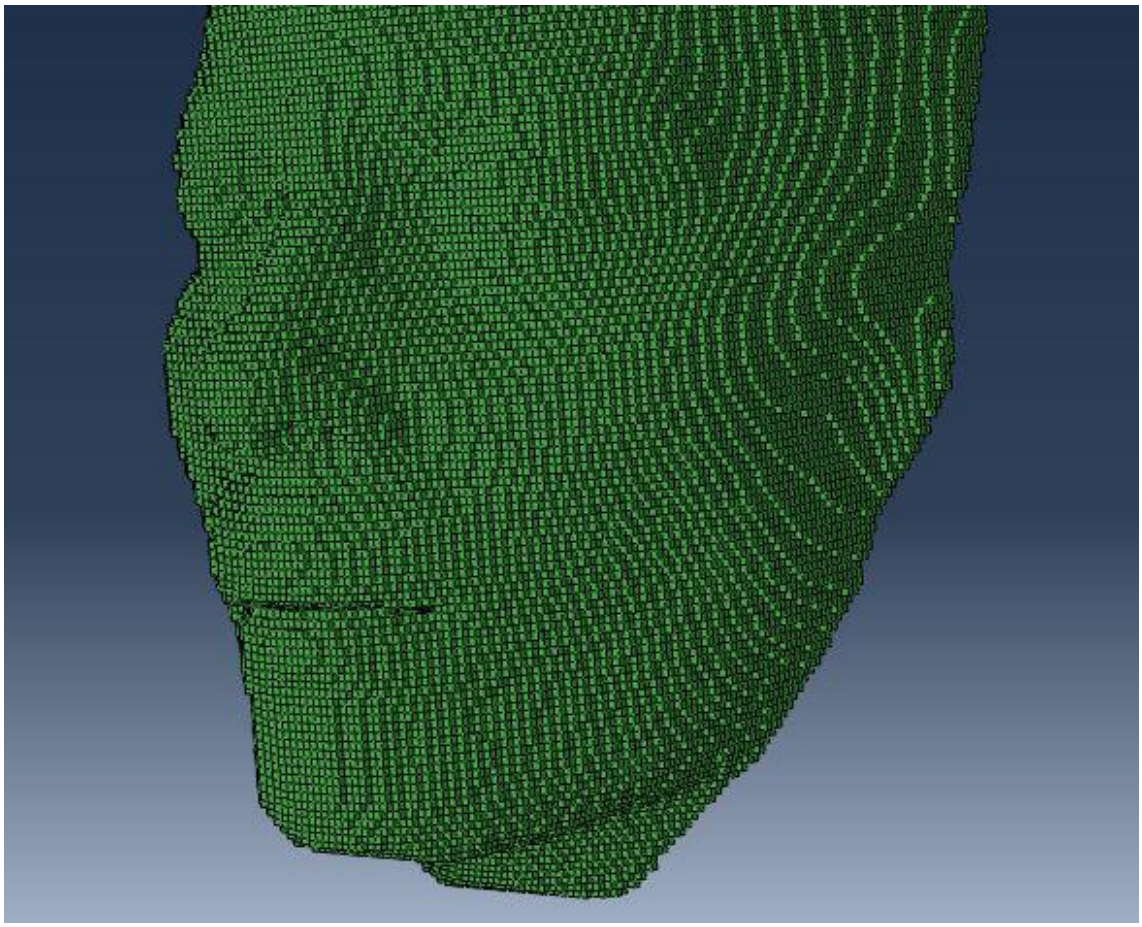

Figura 6.3: Modelo 3D con elementos C3D8 en ABAQUS

En la imagen anterior se observa claramente que las superficies del modelo son ahora mucho menos suaves que en el modelo con elementos tetraédricos, como consecuencia de no haber llevado a cabo el suavizado en ScanFE.

La primera dificultad que aparece al tratar con este modelo se da al separar cráneo y mandíbula superior de la mandíbula inferior. Al separar estas regiones en grupos de elementos distintos, en la zona de los dientes aparecen nodos que pertenecen tanto a la mandíbula inferior como a la superior (los dientes están juntos en algunos nodos). Esto hace que al ir a simular la cirugía e imponer desplazamientos a los elementos de la mandíbula inferior, también se le estén imponiendo movimientos a nodos pertenecientes a la mandíbula superior, que se define como sólido rígido, cosa que es incompatible. La solución adoptada ha sido buscar los nodos que pertenecían a ambos grupos de elementos (17 en nuestro caso), duplicarlos y dárselos a la mandíbula superior, redefiniendo cuidadosamente los elementos donde aparecían con los nuevos nodos.

Una vez solventado el problema anterior, antes siquiera de definir el vestíbulo de la boca, se hacen unas pruebas comprobar si el comando **ADAPTIVE MESH* funciona en este caso. Se divide el hueso en cuatro grupos de elementos (igual que se hizo en el primer caso explicado anteriormente). Se crea: uno para el cráneo y la mandíbula superior, otro para la zona posterior de la mandíbula inferior y articulación temporomandibular, otro para la parte anterior de la mandíbula y otro en una zona intermedia entre los dos anteriores. A todos los grupos de elementos se les dan las mismas propiedades del material. El primero y el segundo se definen como sólido rígido. El último de ellos es la zona en la que se va a utilizar el remallado. En la siguiente imagen, los grupos se han fijado como sólido rígido se representan en blanco, la zona delantera de la mandíbula (a cuyos nodos se va a imponer el desplazamiento) en rojo, y la zona con mallado adaptativo en azul.

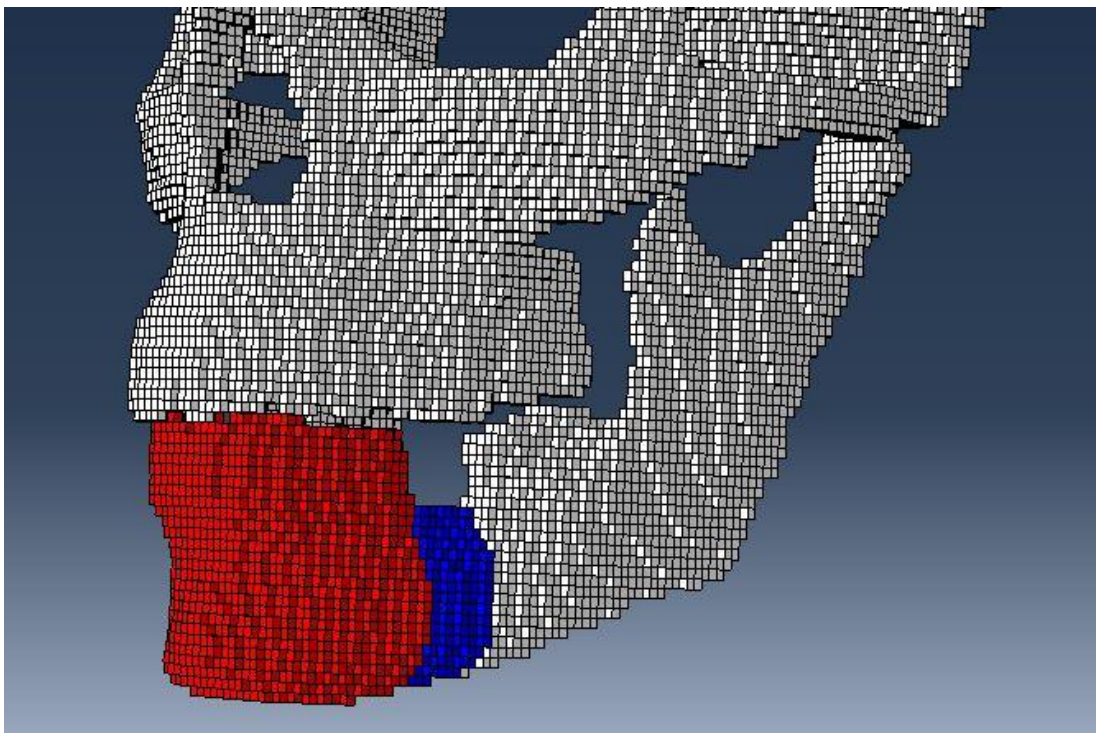

Figura 6.4: Zona de mallado adaptativo (azul) y zona a desplazar (rojo)

Si los nodos a los que se les impone el desplazamiento se escogen muy cerca de la zona definida como sólido rígido, al resolver aparecen problemas causados por la distorsión excesiva de los elementos. Por esta razón el grupo de nodos a los que se aplica el movimiento deseado se eligen justo por delante de la zona que remalla.

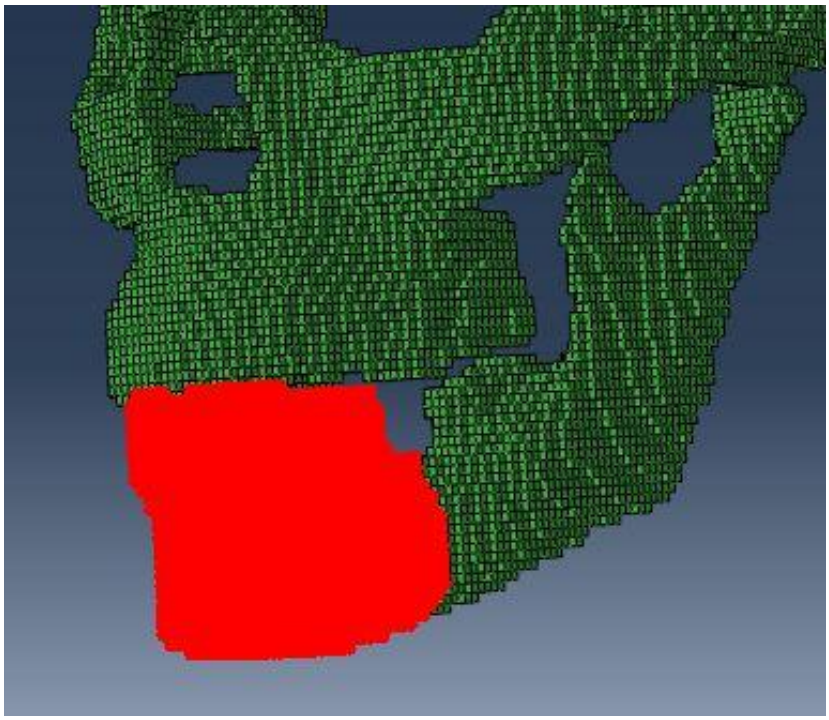

Figura 6.5: Grupo de nodos al que se le impone el desplazamiento

Con este modelo y estos grupos, se llegan a resolver desplazamientos de hasta casi 5 milímetros. En los archivos que crea ABAQUS al resolver ya no aparece la línea donde ponía qué elementos no eran validos para el **ADAPTIVE MESH*. Esto pone de manifiesto que ahora sí que se está remallando. Con valores mayores del desplazamiento aparecen problemas debidos a distorsiones excesivas en los elementos.

En este caso la mandíbula inferior no se deforma totalmente, como ocurría en los casos anteriores. Esto se acerca mucho más a la realidad, aunque el movimiento de la mandíbula que se consigue resolver aún no llega a los que se dan en las cirugías. Para solucionar el problema de la distorsión en los elementos basta con aumentar un poco el tamaño de la zona definida como **ADAPTIVE MESH*. En la siguiente imagen se observa como la única región del hueso que se ha deformado ha sido una parte de la zona donde se produce el remallado.

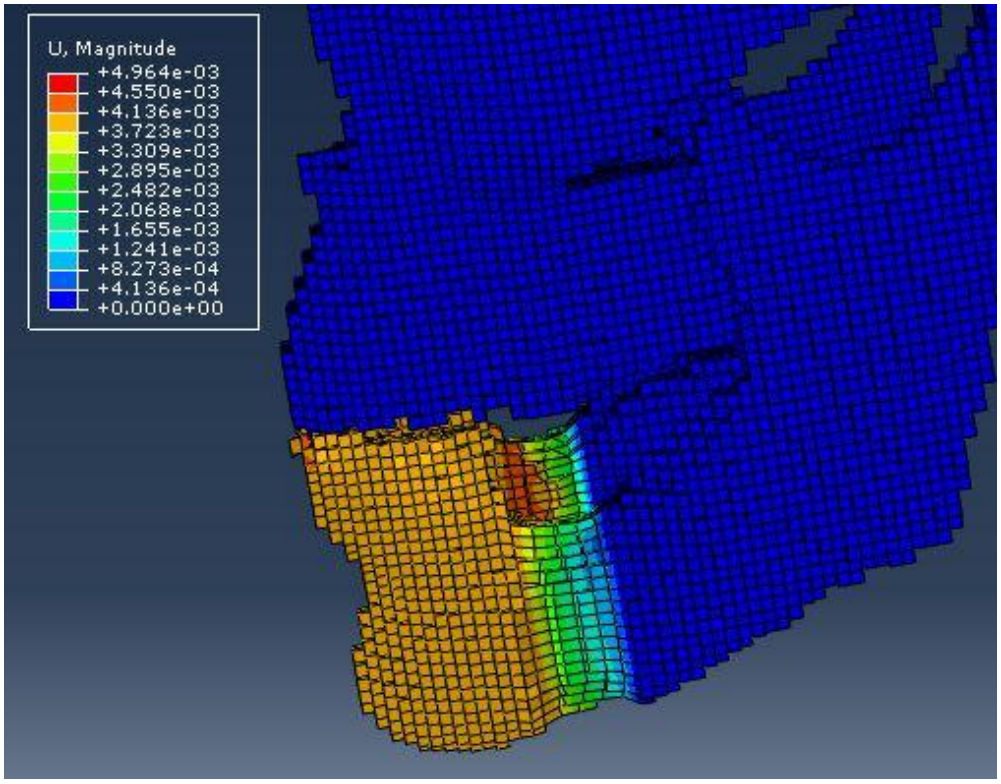

Figura 6.6: Deformaciones producidas en el hueso

Una vez comprobado que con este nuevo modelo el comando **ADAPTIVE MESH* sí que funciona, hay que crear el vestíbulo de la boca. Al tratarse ahora de elementos con más nodos que en el caso anterior, se tiene que modificar el programa de FORTRAN que se uso para el otro modelo. Hecha esta modificación, al igual que antes, se duplican los nodos comunes y se le asignan los nuevos nodos al hueso.

El siguiente paso es definir las superficies del vestíbulo e imponer la condición de no penetración entre ellas. La superficie interna de la piel se crea fácilmente como una superficie basada en nodos. El problema aparece en la definición de la superficie del hueso. Al actuar en el contacto como superficie *master* es necesario que sea creada como una superficie basada en elementos. Para poder definir una superficie de este tipo hay que conseguir un grupo de elementos tal que la cara que tengan en la superficie sea un conjunto continuo. Con la geometría actual, las superficies son muy discontinuas y resulta imposible encontrar un conjunto de este tipo.

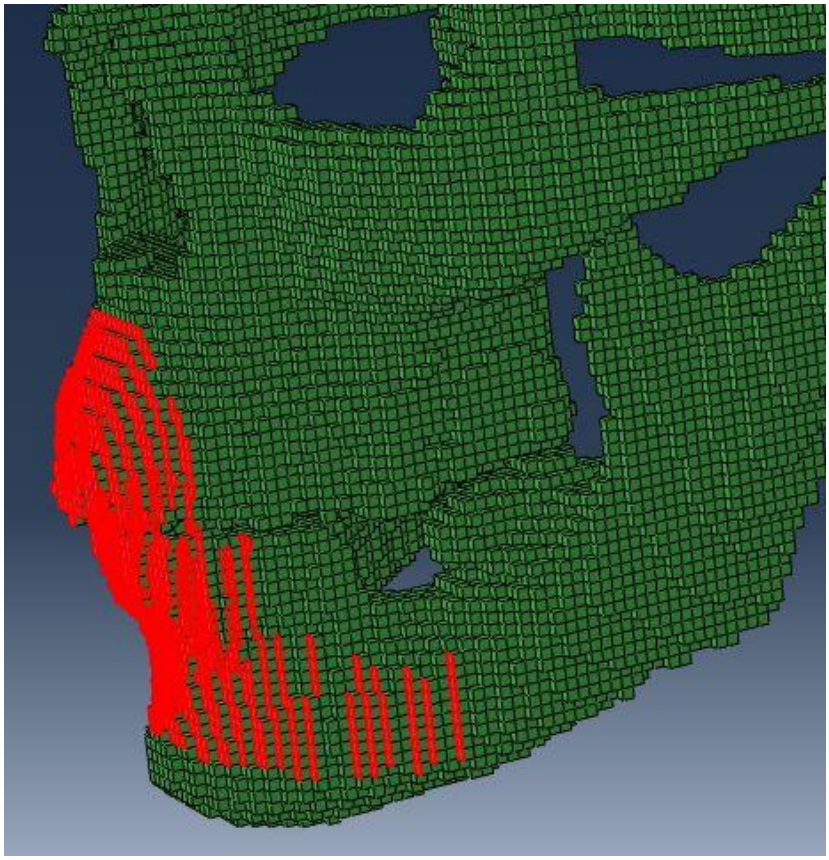

Figura 6.7: Superficies del hueso con el modelo de elementos C3D8

Al no poder definir esta superficies, no somos capaces de imponer la condición de no penetración en el vestíbulo. Aunque se consiguieran definir (seleccionando todos y cada uno de los elementos, lo cual es un trabajo muy laborioso), al ser una superficie tan irregular, resolver el problema del contacto sería prácticamente imposible. Por estas dos razones se decidió dejar de trabajar sobre este modelo.

6.6 Modelo final escogido; resultados.

Finalmente se decide que el modelo más adecuado con el que seguir trabajando es aquel en el que se definía el contacto en el interior de la mandíbula y solo se fijaban dos puntos en la parte posterior de la mandíbula. En este caso, al aplicar los desplazamientos en los nodos delanteros de la mandíbula, esta se deforma casi completamente. Aunque no sea el caso ideal (en el que las deformaciones se concentren en una zona pequeña de la mandíbula), al estar interesados principalmente en las deformaciones que se producen en un corte sagital que divide la cara en dos mitades, lo que ocurre en la zona posterior de la mandíbula no es tan importante.

Con este modelo, y con el contacto definido tal y como se explica en el apartado 3 de este capítulo, se entra más en profundidad en las causas de los errores que aparecen y en cómo solucionarlos. Abriendo el archivo ".odb" del caso que se ha conseguido resolver (desplazamiento de tres milímetros) se mira en qué zonas del hueso aparecen las presiones máximas. Se observa que las presiones se concentran en algunos puntos del modelo, por lo general en los extremos de la superficie de contacto definida. Esto se debe a la existencia de pequeños picos en el modelo (que actúan como si fueran punzones) y a las discontinuidades que aparecen en los bordes. Para solucionar estos problemas es necesario suavizar las superficies de contacto.

El suavizado de las superficies no es una tarea sencilla. Una vez encontrados los nodos en los que aparecen estas presiones anormales, lo que se hace es moverlos de tal forma que se suavice la superficie. Para saber la distancia y dirección en que moverlos se representan en Matlab, como una nube de puntos, el nodo en cuestión y los de su alrededor. Mediante el procedimiento de prueba y error se mueve el punto de manera que todos queden aproximadamente en una superficie suave.

Este procedimiento debe permitirnos resolver desplazamientos mayores, siempre que el error que nos lo impide sean esas presiones localizadas antes comentadas. Aunque a la hora de redactar este proyecto solo se han llegado a simular desplazamientos de tres milímetros con éxito, se va a seguir trabajando en él para intentar conseguir simular cirugías de hasta un centímetro.

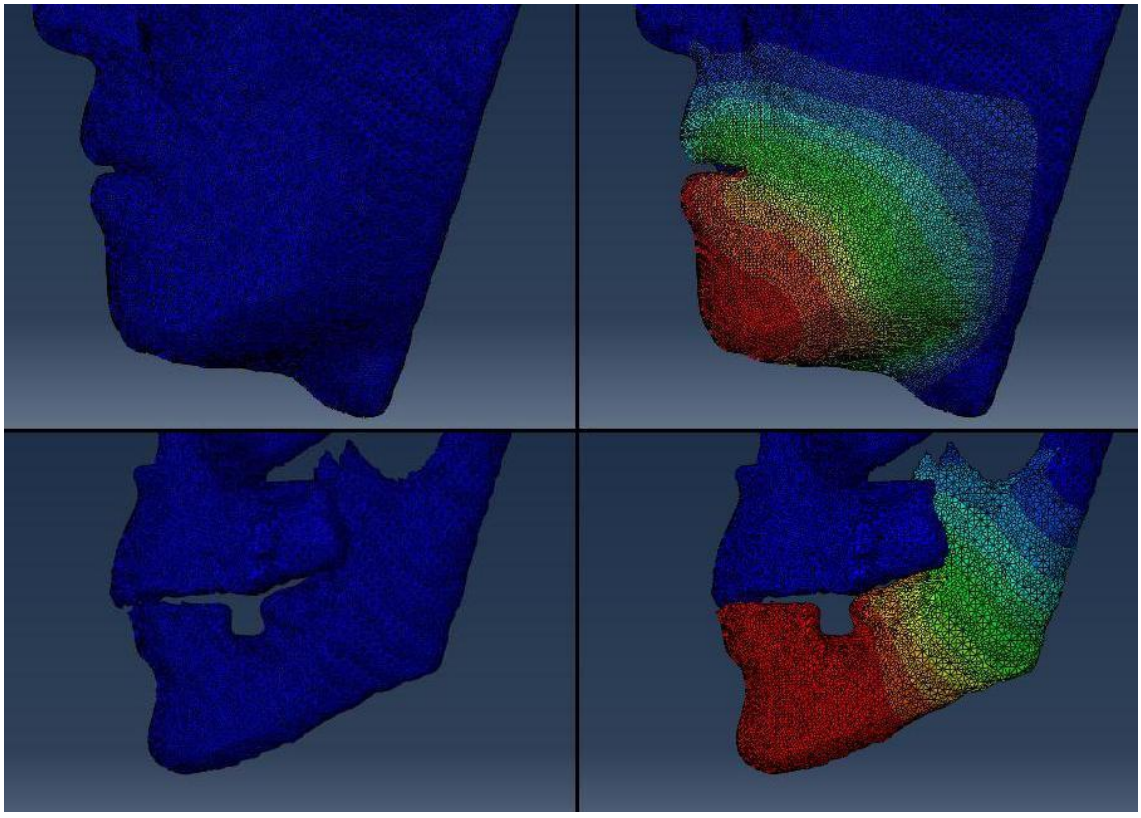

Figura 6.8: Resultados al simular un desplazamiento de 3 mm

Capítulo 7:

REDES NEURONALES

En este capítulo se usan los resultados de todas las simulaciones realizadas anteriormente para intentar ajustar los valores de m_2 con ayuda de redes neuronales artificiales. Estas redes actúan como una caja negra, sin información alguna del modelo con el que se está trabajando. Con los datos obtenidos de las simulaciones se entrena la red para que se comporte de manera similar a como lo hace el modelo de la cara. Lo que se pretende conseguir con la utilización de las redes neuronales es que, dados unos desplazamientos conocidos (los que se obtienen de los escaneados faciales), en los puntos de medición de la cara antes comentados, nos devuelva el valor de m_2 de las distintas regiones en las que se ha dividido la cara.

Se comienza definiendo las bases biológicas que han dado lugar a la implementación de las distintas redes neuronales artificiales. Posteriormente se explica que tipo de red es la que se ha empleado en este proyecto y que resultados se han obtenido de su utilización.

7.1 Breve introducción a las redes neuronales.

7.1.1 Fundamentos biológicos.

Las redes neuronales artificiales (RNA) tratan de emular el sistema neuronal del cerebro humano. Los conocimientos actuales de la estructura y composición de las neuronas se deben principalmente al científico D. Santiago Ramón y Cajal quien demostró que el sistema nervioso estaba compuesto por una red de neuronas individuales que se conectan entre sí formando redes.

A continuación se numeran las características más destacables del sistema nervioso central. Estas cualidades son las que se intentan simular con una RNA:

- Capacidad de adquirir conocimiento desde la experiencia (aprendizaje).
- Conocimiento almacenado en las conexiones entre neuronas (conexiones sinápticas).
- Gran capacidad de adaptación.
- Comportamiento altamente no lineal.
- Alta tolerancia a fallos.
- Apto para reconocimiento, percepción y control.

Las neuronas se componen de núcleo, axón, que es una ramificación de salida de la neurona, y de un gran número de ramificaciones de entrada llamadas dendritas. El axón puede ramificarse en su punto de arranque y suele presentar múltiples ramas en su extremo. La zona de contacto entre neuronas se denomina sinapsis o espacio sináptico.

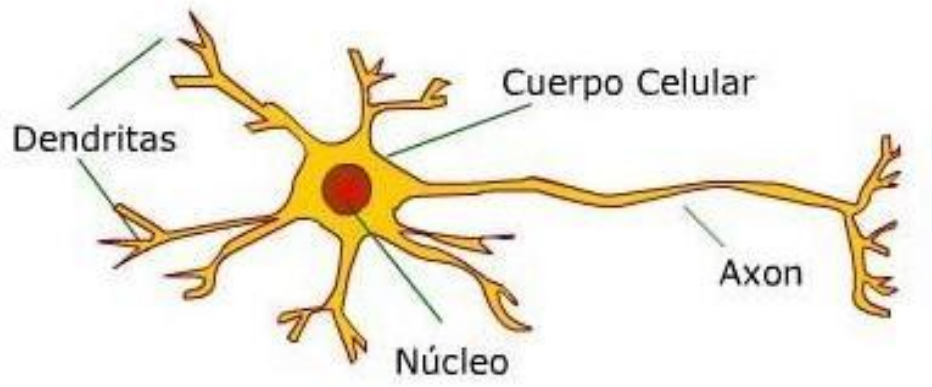

Figura 7.1: Esquema de una neurona

La sinapsis recoge la información que proviene de las células adyacentes. Esta información se transmite al núcleo de la neurona a través de las dendritas. En el núcleo

se procesa la información y se genera una respuesta que es posteriormente propagada por el axón.

El espacio sináptico es la zona de unión entre neuronas, y son en estos contactos donde se lleva a cabo la transmisión del impulso nervioso, de la información. La célula emisora de esa información, a través del axón, segrega un tipo de compuestos químicos (neurotransmisores) que se depositan en el espacio intersináptico. Estas sustancias segregadas son las encargadas de excitar o inhibir la acción de la célula receptora. Esta propiedad de poder alterar el peso de cada información en la red neuronal es la que otorga en cierta medida la capacidad de aprender.

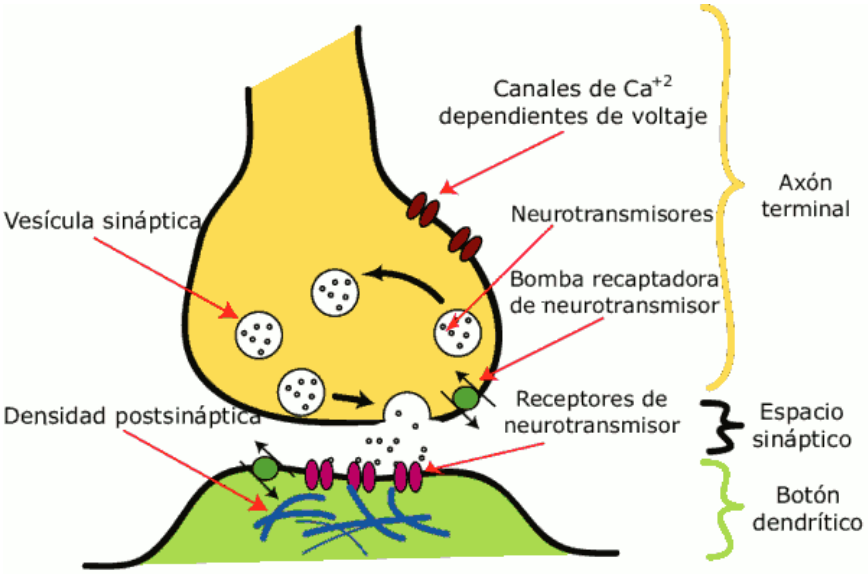

Figura 7.2: Detalle de la transmisión de información en el espacio intersináptico

7.1.2 Redes neuronales artificiales.

Existen numerosas formas de definir a las RNA entre las que cabe destacar:

"Una nueva forma de computación, inspirada en modelos biológicos."

"Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles."

(Hilera y Martínez)

"Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico."

(Kohonen, 1988)

"(…) un sistema de computación constituido por un gran número de elementos simples de procesamiento muy interconectados, que procesan la información por medio de su estado dinámico como respuesta a entradas externas."

(Hecht-Nielsen, 1988)

Las RNA pueden considerarse modelos de cálculo caracterizados por algoritmos muy eficientes que operan en paralelo y permiten desarrollar tareas como el aprendizaje de patrones, la clasificación o la optimización.

Una RNA puede definirse usando del concepto de grafo, objeto consistente en un conjunto de nodos (o vértices) más todas las conexiones establecidas entre ellos. Las neuronas de una red neuronal se identifican con los nodos y las dendritas y axones con las conexiones entre nodos. La propiedad que tiene el espacio sináptico de poder inhibir o excitar la señal de entrada a una determinada neurona, se consigue definiendo unos pesos en las conexiones entre neuronas.

"Una red neuronal artificial puede definirse como un grafo dirigido (todas las conexiones tienen asignado un sentido), con las siguientes propiedades:

- A cada nodo j se le asocia una variable de estado x_j.
- A cada conexión (i,j) de los nodos i y j se le asocia un peso $w_{ij} \in \mathbb{R}$.
- En muchas ocasiones a cada nodo se le asocia un umbral de disparo θ_j.
- Para cada nodo j se define una función $f_j(x_j, w_{ij}, \theta_j)$, que depende de los pesos de sus conexiones, del umbral y de los estados de los nodos i a él conectados. Esta función proporciona el nuevo estado del nodo."

(Müller y Reinhardt, 1990)

En la siguiente imagen se representa un esquema general de la arquitectura de una red neuronal artificial.

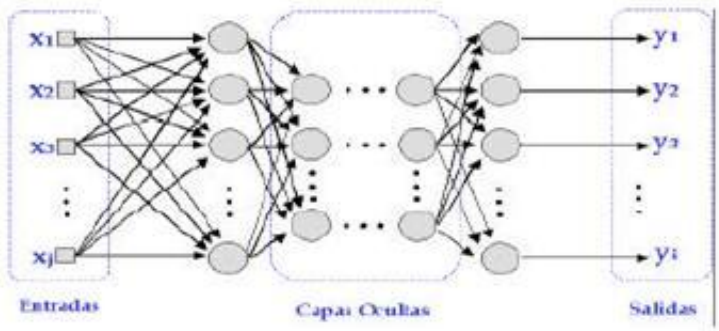

Figura 7.3: Arquitectura general de una red neuronal

Para finalizar, con el objetivo de establecer una analogía clara y directa entre las redes neuronales del cerebro humano y las redes neuronales artificiales, se incluye una tabla resumen:

Neurona biológica	Neurona artificial
- Señales que llegan a la sinapsis.	- Entradas a la neurona.
- Carácter excitador o inhibidor de las sinapsis de entrada.	- Pesos de entrada.
- Estímulo total de la neurona.	- $Net_j = \sum w_{ij}(t) \cdot x_i(t)$ (Regla de propagación)
- Activación o no de la neurona.	- Función de activación.
- Respuesta de la neurona.	- Función de salida.

Todos los elementos que forman parte de una RNA pueden variar de una red a otra. Hay distintos tipos de funciones de activación, funciones de salida y reglas de activación. Al no ser el objetivo de este proyecto las redes neuronales en sí, se insta al que lo desee a buscar en la bibliografía más información acerca de los distintos tipos de funciones que se pueden implementar.

La correcta elección de estos elementos es muy importante, y varía de un caso a otro. Hay algunos problemas en los que la forma de elegir estos elementos (y la arquitectura de la red) esta sobradamente documentada. En otros casos, como es el que nos ocupa, no se han encontrado referencias claras para definir la red de manera inequívoca. La forma de la red y las funciones elegidas que se han usado en este proyecto se basan en la experimentación con múltiples redes y funciones. En el siguiente punto de este capítulo se da una explicación más detallada de la metodología empleada en la elección de la red neuronal empleada.

Una característica importante de las RNA es su capacidad de aprendizaje. Se define como el proceso por el que una red neuronal crea, modifica o destruye sus conexiones (pesos) en respuesta a una información de entrada. Ante un conjunto de datos de entrenamiento (datos de entrada y datos de salida que se corresponden con cada entrada), la red modifica los pesos en cada neurona para, dada la entrada, obtener la salida adecuada. A la hora de entrenar una red hay muchos parámetros a tener en cuenta: método de aprendizaje, criterios a la hora de cambiar los pesos de entrada, problemas de sobreaprendizaje de la red… En la bibliografía referente a redes neuronales artificiales se puede encontrar esta información.

7.1.3 Arquitectura de una red neuronal artificial.

La arquitectura de una red hace referencia a la organización de las neuronas dentro de la misma, formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y la salida a dicha red. Los parámetros que define la topología de la red son: el número de capas, el número de neuronas por capa, la forma en la que están conectadas las distintas capas y neuronas y el tipo de conexión entre neuronas.

7.2 El perceptrón multicapa: base teórica y empleo en Matlab.

7.2.1 Base teórica.

La arquitectura que se ha usado en este caso es la arquitectura del perceptrón multicapa con conexiones hacia delante. Ha sido demostrado por algunos autores que este tipo de arquitectura es un aproximador universal de cualquier función en el espacio $\mathbb{R}_n$.

Este tipo de red se caracteriza porque tiene todas sus neuronas agrupadas en distintos niveles o capas. El primer nivel corresponde a la capa de entrada, que se encarga únicamente de propagar por el resto de la red las entradas recibidas. El último nivel es el de la capa de salida. Se encarga de proporcionar los valores de salida de la red. En las capas intermedias denominadas capas ocultas, se realiza un procesamiento no lineal de los patrones recibidos.

Las conexiones del perceptrón multicapa son hacia adelante. Generalmente todas las neuronas de un nivel se conectan con todas las neuronas de la capa inmediatamente posterior. A veces, dependiendo de la red, se encuentran conexiones de neuronas que no están en niveles consecutivos, o alguna de las conexiones entre dos neuronas de niveles consecutivos no existe, es decir, el peso asociado a dicha conexión es constante e igual a cero. Además, todas las neuronas de la red tienen un valor umbral asociado. Se suele tratar como una entrada cuyo valor es constante e igual a uno, y lo único que varia es el peso asociado a dicha conexión (que es el umbral realmente).

Por otro lado, las funciones de activación que se suelen utilizar son la función identidad, la función sigmoidal y la función tangente hiperbólica. A continuación se muestran sus respectivas expresiones.

Función identidad: $$f_1(x) = x$$

Función sigmoidal: $$f_2(x) = \frac{1}{1 + e^{-x}}$$

Función tangente hiperbólica: $$f_3(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

La principal diferencia entre la función sigmoidal y la función tangente hiperbólica es el rango de sus valores de salida. Mientras que para la primera su rango es [0,1], para la segunda es [-1,1]. De hecho existe una relación entre las dos. Se relacionan mediante la expresión:

$$f_3(x) = 2f_2(x) - 1$$

El método de aprendizaje que usa el perceptrón multicapa se basa en el algoritmo de retropropagación, también llamado "Regla Delta generalizada". Es el algoritmo mediante el cual se van adaptando todos los parámetros de la red.

El aprendizaje de la red se plantea como un problema de minimización de una determinada función de error. Se usa como función de error, el error medio cuadrático, es decir:

$$E = \frac{1}{N}\sum_{n=1}^{N} e(n)$$

$$e(n) = \frac{1}{2}\cdot\sum_{n=1}^{t}(s(n) - y(n))^2$$

Donde *s(n)* es la salida del patrón, *y(n)* la salida obtenida de la red, *t* el número de neuronas de salida y *N* el número de patrones.

El problema es no lineal y como tal, el problema de minimización de la función error se resuelve por técnicas de optimización no lineales que se basan en ajustar los parámetros siguiendo una determinada dirección. En este método, la dirección elegida es la negativa del gradiente de la función error.

A partir de aquí existen dos opciones. Podemos cambiar los parámetros cada vez que introducimos el patrón, o solamente cambiarlos cuando hayamos introducido todos los parámetros de entrenamiento por cada ciclo. En el primer caso, debemos minimizar *e(n)* y en el segundo se minimiza la función *E*. A continuación se presentará el desarrollo para el primer caso; la extensión al segundo es inmediata.

De acuerdo a lo que se ha dicho antes, todos los pesos deben variar según la dirección del gradiente del error. Matemáticamente esto se expresa de la siguiente forma:

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w}$$

Por lo tanto el problema consiste en evaluar la derivada. El parámetro α es la tasa de aprendizaje que influye en la magnitud del desplazamiento en la superficie de la función error. A continuación se evalúa el valor del gradiente para los pesos de las conexiones de la última capa oculta a las neuronas de la capa de salida.

Sea $w_{ji}^{c-1}(n)$ el peso de la conexión de la neurona *j* de la capa *C-1* a la neurona *i* de la capa de salida. El error *e(n)* solo se ve afectado por w_{ji}^{c-1} en el error de la salida de la neurona *i*. Por tanto:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = -(s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{ji}^{c-1}}$$

Por otro lado, la salida de la neurona *i* es igual a la suma de las entradas transformadas según su función de activación. Aplicando la regla de la cadena, y teniendo en cuenta que w_{ji}^{c-1} solo afecta a la entrada de la neurona *i* porque va multiplicando a_j se tiene que:

$$\frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} = f'\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c\right) a_j^{c-1}(n)$$

Se define *δ* asociado a la neurona *i* de la capa *C* del patrón *n*, $\delta_i^c(n)$, del siguiente modo:

$$\delta_i^c(n) = -\left(s_i(n) - y_i(n)\right) f'\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1}\, a_j^{c-1} + u_i^c\right)$$

De tal manera que el gradiente se exprese ahora mediante la siguiente expresión:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = \delta_i^c(n) a_j^{c-1}(n)$$

Esto es extensible para todos los pesos de las conexiones de las neuronas de la capa *C-1* con las neuronas de la capa de salida. Se procede de la misma forma para los valores umbrales obteniéndose una expresión análoga.

A continuación se procede a calcular la magnitud de la variación para el resto de conexiones que existan entre las capas ocultas. Se va a proceder de la misma forma que hasta ahora y se obtendrá una ley de recurrencia para la modificación de dichos pesos. Consideremos el peso w_{kj}^{c-2} de la conexión de la neurona *k* de la capa *C-2* con la neurona *j* de la capa *C-1*. El nuevo valor de dicho peso vendrá dado por una expresión análoga a la de $w(n)$, donde lo único que se desconoce es el valor del gradiente. En este caso w_{kj}^{c-2} influye en todas las salidas de la red, por lo que se tiene que:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = -\sum_{i=1}^{n_c}(s_i(n) - y_i(n))\frac{y_i(n)}{\partial w_{kj}^{c-2}}$$

Para calcular la derivada hay que tener en cuenta que la salida w_{kj}^{c-2} influye en la entrada de la neurona *j* (por lo que influye en su salida), que a su vez influye en la entrada de todas las neuronas de salida. Por tanto, aplicando la regla de la cadena tenemos:

$$\frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} = f'\left(\sum_{j=1}^{n_{c-1}} w_{ij}^{c-1}a_j^{c-1} + u_j^{c-1}\right)\frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}}$$

Por otro lado la salida de la neurona j, únicamente se ve afectada por w_{kj}^{c-2} que va multiplicando a a_k^{c-2}. Por lo tanto, y volviendo a aplicar la regla de la cadena tenemos:

$$\frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} = f'\left(\sum_{k=1}^{n_{c-2}} w_{kj}^{c-2}a_k^{c-2} + u_k^{c-2}\right)a_k^{c-2}(n)$$

Definiendo el valor de δ para las neuronas de las capas *C-1* de la siguiente forma:

$$\delta_j^{c-1}(n) = f'\left(\sum_{k=1}^{n_{c-2}} w_{kj}^{c-2}a_k^{c-2} + u_j^{c-2}\right)\sum_{i=1}^{n_c}\delta_i^c(n)w_{ji}^{c-1}$$

Se tiene que el valor de la derivada del error con respecto al peso w_{kj}^{c-2} es:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = \delta_j^{c-1}(n)a_k^{c-2}(n)$$

Por lo que el nuevo valor de dicho peso viene dado por la expresión:

$$w_{kj}^{c-2}(n) = w_{kj}^{c-2}(n-1) - \alpha\delta_j^{c-1}(n)a_k^{c-2}(n)$$

Se observa que la modificación de los pesos de las conexiones entre las neuronas de la capa *C-2* y las neuronas de la capa *C-1* se ve afectada por la salida de la neurona *k* de la capa *C-2* y el termino δ asociado a la neurona a la que llega la conexión. Lo único que varía es la expresión de δ. En este punto es posible generalizar la actualización de cualquiera de los pesos de cualquier capa dentro de la red, acorde a la ecuación anterior. Por ello, la actualización de uno de los pesos de la conexión que une la neurona *k* de la capa *h* con la neurona *j* de la capa *h+1* vendrá dado por la expresión:

$$w_{kj}^{h}(n) = w_{kj}^{h}(n-1) - \alpha\delta_{j}^{h+1}(n)a_{k}^{h}(n)$$

Donde le término δ viene dado por la siguiente ley de recurrencia:

$$\delta_{j}^{h+1} = f'\left(\sum_{k=1}^{n_h} w_{kj}^{h}a_{k}^{h} + u_{j}^{h+1}\right)\sum_{i=1}^{n_{h+2}}\delta_{i}^{h+2}(n)w_{ji}^{h+1}$$

El termino δ propaga los errores obtenidos a la salida hacia atrás. De esta manera, cada neurona oculta recibe un cierto error o valor δ de todas las neuronas a las cuales se conecta, y la suma de todos estos errores es una medida del error total que comete esta neurona. Para la actualización de los valores umbrales se procede de la misma forma llegando a una expresión para su actualización dada por la siguiente expresión:

$$u_{j}^{h+1}(n) = u_{j}^{h+1}(n-1) - \alpha\delta_{j}^{h+1}(n)$$

En este tipo de entrenamiento se define el número de épocas como el número de veces que se han comparado los ejemplos con las salidas de la red para realizar los justes en los pesos de las conexiones.

Normalmente se repite este proceso hasta alcanzar el error mínimo de entrenamiento, aunque se pueden establecer otros criterios de parada:

- Hasta que el error de entrenamiento se estabilice.
- Hasta que el error de validación se estabilice.
- Hasta que el error de validación aumente.

La siguiente imagen representa un esquema de una red de este tipo. La capa oculta, aunque en la imagen se representa como una sola, a su vez puede estar compuesta por una o varias capas.

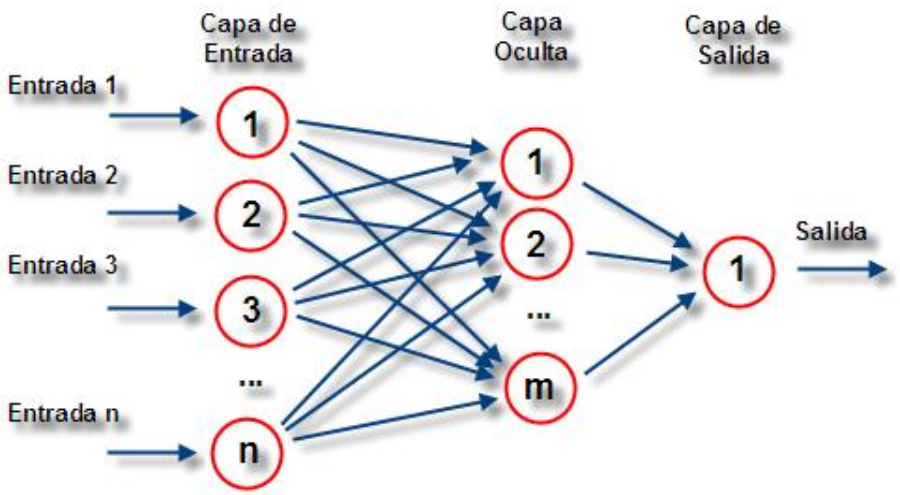

Figura 7.4: Esquema de red perceptrón multicapa

Este método de aprendizaje es el que usa Matlab por defecto en la arquitectura del perceptrón multicapa. Existen algunas mejoras que aceleran la convergencia del algoritmo. Estas mejoras se pueden emplear en Matlab modificando las opciones de la función que crea la red neuronal (tal y como se explica en el siguiente punto).

7.2.2 Empleo en Matlab.

- Creación de la red:

En Matlab se crea una red perceptrón multicapa con conexiones hacia delante utilizando la función *newff*. Esta función devuelve una variable que representa a la red.

Como aclaración, antes de comentar los argumentos de la función *newff*, es necesario definir qué valores son las entradas de la red y cuáles las salidas. El objetivo del problema es, dados unos desplazamientos conocidos (sacados de los escaneados faciales) obtener el valor de m_2 en las distintas zonas de la cara. Por tanto, para entrenar la red, las entradas serán los desplazamientos y las salidas los valores de m_2 usados para cada simulación. Las entradas son cinco valores (de los cinco puntos en los que se miden los desplazamientos), mientras que las salidas son seis (seis zonas en las que se han dividido los tejidos blandos de la cara).

Los argumentos de entrada de esta función (descritos en el orden en el que se colocan) se describen a continuación.

- *p* es una matriz cuyo número de filas es el número de entradas y el número de columnas es el número de casos para entrenar. Cada columna contiene los desplazamientos obtenidos en una simulación.

- *t* es un matriz cuyo número de filas es el número de salidas y el número de columnas es el número de casos para entrenar. Cada columna contiene los valores de m2 usados en las simulaciones.

- [*Capa1, Capa2,…*] es un vector que describe el número de neuronas que tienen todas las capas ocultas de la red. Asimismo el tamaño de dicho vector permite conocer al programa el número de capas que debe tener la red que se va a crear.

- {*Funciones*} es un vector de varias cadenas de caracteres en el que se señalan las funciones de activación que van a poseer todas las neuronas de una capa. Así, la primera cadena indicará la función de activación de las neuronas de la primera capa oculta, la segunda señalara la función de las neuronas de la segunda capa, y así sucesivamente. Por ello este vector debe tener tantas cadenas como capas ocultas vaya a tener nuestra red. Si se escriben menos cadenas de caracteres, el programa usa la función tangente hiperbólica por defecto. Además de las capas ocultas que se le indiquen, el programa crea una capa más, la capa de salida. Para esta capa la función por defecto es la lineal. En el caso de querer que la función de activación de la capa de salida no sea lineal, se puede poner una cadena más de caracteres en este vector indicando la función de las neuronas de la capa de salida. Para que la función de activación sea lineal, la cadena deber ser '*purelin*', para que ésta sea una función tangente hiperbólica se debe escribir '*tansig*', y para indicar que se quiere una función sigmoidal se escribirá '*logsig*'.

- En el siguiente argumento se indica el tipo de entrenamiento que va a seguir la red. Si no se indica nada el entrenamiento que seguirá es el explicado anteriormente (*trainlm*).

A continuación se representa una línea de código a modo de ejemplo:

```
net = newff(p,t,[10 30 20],{'purelin','purelin','purelin'},'trainlm')
```

La red creada se ha llamado *net*, tiene tres capas ocultas (con 10, 30 y 20 neuronas en cada capa respectivamente) más la capa de salida (con igual número de neuronas que salidas tiene el problema, en este caso seis), todas las funciones de activación son lineales y el método de entrenamiento que sigue es el de la regla delta generalizada. La siguiente imagen muestra la red creada por Matlab.

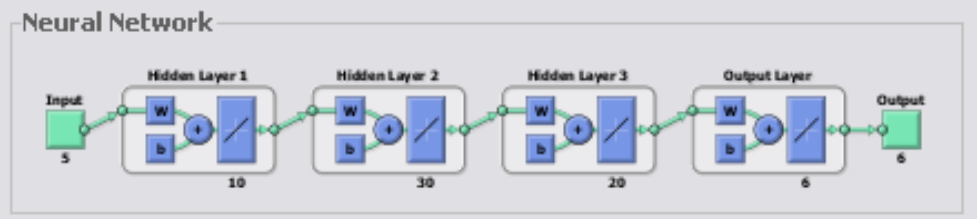

Figura 7.5: Esquema de la red creada por Matlab

- Entrenamiento.

El entrenamiento de la red consiste en presentarle unas entradas y sus correspondientes salidas, para que la red vaya reajustando su salida mediante la modificación de sus pesos y valores umbrales, de manera que el error de actuación de la red se minimice. La medida del error por defecto en Matlab es el error medio cuadrático.

Para entrenar la red en Matlab se utiliza la función *train*. Dicha función utiliza como argumentos el nombre de la red que se quiere entrenar (y que anteriormente a debido ser creada) y los patrones de entrenamiento, compuestos por unos vectores de entradas y sus correspondientes salidas (matrices *p* y *t* antes comentadas). De esta manera la línea de código que se debería escribir es:

```
net = train(net,p,t)
```

La variable net contiene a la red ya entrenada, es decir, con sus pesos y valores umbrales ajustados. Por otro lado, destacar que esta función utiliza una serie de variables para definir el entrenamiento y que pueden definirse con anterioridad. Estas variables varían según el tipo de entrenamiento utilizado. En el anexo IV se muestra un caso típico de las redes neuronales utilizadas en el que se han usado algunas de estas variables.

- Simulación de la red.

Para obtener las salidas de una determinada red ante unas ciertas entradas se utiliza la función *sim*. Esta función devuelve un vector cuyas componentes son las salidas que se obtienen de cada neurona de salida de la red. A dicha función se le introducen como argumentos el nombre que representa la red en Matlab y el vector de entradas a la red.

Se pueden obtener varias simulaciones a la vez introduciendo una matriz, cuyos vectores sea cada uno de los vectores de entrada de los que se quiere obtener la salida. En este proyecto, al ser las salidas objetivo únicas, no tiene sentido resolver varias simulaciones a la vez.

La línea de comandos que se usa para simular la red es:

```
sim (net,ppp)
```

- Métodos que se pueden usar en Matlab para mejorar la capacidad de generalización de la red.

Uno de los problemas que ocurre durante el entrenamiento de la red, como se citó anteriormente, es el sobreaprendizaje, que inhibe la capacidad de generalización de la red.

Uno de los métodos para evitar este problema es diseñar un entrenamiento con la extensión justa, pero es imposible saber de antemano cual debe ser el tamaño de la muestra para una aplicación específica. Por ello existen dos métodos que se están implementados en Matlab para mejorar la capacidad de generalización de la red. Estos son: método de regularización (modificar la función del error) y método de parada temprana.

Para aplicar el primero es necesario conocer bastante bien la red y el problema ante el que nos encontramos para saber cómo modificar la función de error. Ya que el conocimiento que se tiene es limitado, para entrenar a la red se usa el segundo de los métodos.

Esta técnica divide los datos disponibles en tres partes. La primera parte es el conjunto de entrenamiento que es usada para computar el gradiente y para la actualización de los pesos y valores umbrales. El segundo subconjunto es el de validación. El error de validación es monitorizado durante el proceso de entrenamiento. Normalmente este error va decreciendo a medida que transcurre el entrenamiento, pero en el caso de que se produzca sobreaprendizaje, este error comienza a crecer. Cuando el error de validación comienza a crecer en un especificado número de iteraciones (dato que se le introduce al programa), se detiene el entrenamiento y se utilizan los pesos y umbrales de la iteración de menor error de validación.

El tercer subconjunto es el de generalización o test. No es usado durante el entrenamiento pero es útil para comparar distintos modelos. Puede ser útil también graficar este error durante el entrenamiento. Si este error muestra un mínimo en un número de iteraciones significativamente diferente al de error de validación, puede mostrar una pobre división de los datos.

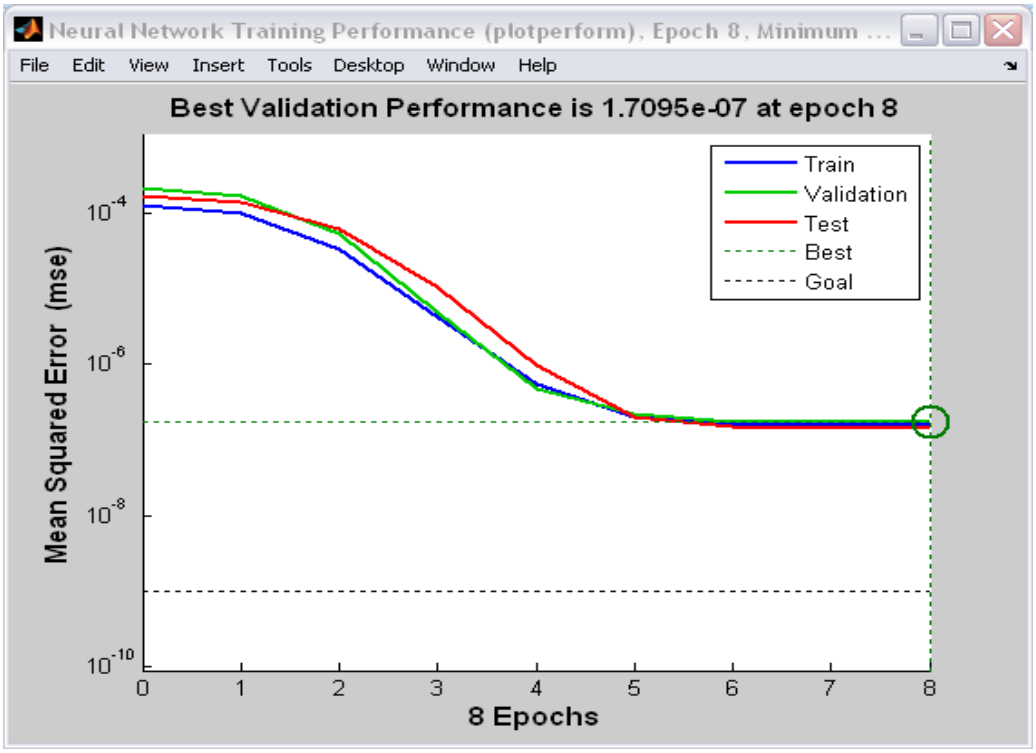

Figura 7.6: Evolución de los errores de entrenamiento, validación y test de la red

En el anexo IV, el caso que se representa usa un 80% de las muestras para entrenar, un 10% para validar y un 10% para verificar los resultados obtenidos.

7.3 Aplicación con los datos de las simulaciones: lecciones aprendidas y resultados.

Una vez realizadas las 99 primeras simulaciones, se montan las matrices *p* y *t* que se van a utilizar para entrenar la red. El programa usado para crear la red lee estas matrices, escritas en archivos independientes con extensión *.m* de Matlab.

Antes de empezar a trabajar con las redes neuronales en Matlab, se consulto a miembros del departamento para obtener algunas nociones prácticas de la forma de trabajar con las redes de Matlab. Gracias a ello se obtuvo la siguiente información:

- Las redes neuronales dan malos resultados con valores de entrada que estén fuera del rango de los valores de entrenamiento.
- Con dos capas intermedias suele bastar.
- Funcionan mejor si el número de neuronas en cada una de las capas intermedias va en orden decreciente.

- Aumentar el número de casos con los que entrenar la red, por lo general, mejora los resultados, aunque puede llegar un punto en que no lo haga (sobreaprendizaje).

- La dimensión de la red (número de capas y neuronas en cada capa) está limitada por la memoria del ordenador en que se trabaje.

Todo lo anterior se ha obtenido experimentalmente a base de trabajar con redes neuronales artificiales. Por esta razón puede ser que algo de lo arriba expuesto no sea totalmente cierto o haya que matizarlo. Aunque la dimensión de la red esté limitada por la memoria del ordenador, se cree que las redes que podemos crear son suficientes para nuestro caso.

A partir de aquí se empezaron a entrenar y simular redes con distintas estructuras y funciones de activación. Después de muchas pruebas se llegó a la conclusión de que lo que más influye es el tipo de función de activación. La función lineal es la que mejor resultados da al simular la red después de entrenarla. Además es la que menor tiempo de entrenamiento necesita, y sus resultados no varían mucho con la estructura de la red. Si se introduce una función tangente hiperbólica o una función sigmoidal, el tiempo de entrenamiento es mayor y los resultados son bastante peores.

Para decir si los resultados son buenos o malos nos basamos en los valores que se esperan obtener. Ya que las salidas de la red son los valores de m_2 de las distintas zonas en las que se ha dividido la piel, es necesario que estos valores sean positivos (un m_2 negativo no tiene sentido). También es de esperar que, por ejemplo, la piel en la zona que se ha llamado PAPADA+CUELLO sea menos rígida que en la NARIZ.

Introduciendo los resultados obtenidos al simular la red neuronal en archivos de carga de ABAQUS se obtienen errores un poco mayores de 0,1 milímetros (valores muy similar a los mejores resultados obtenidos con las simulaciones).

Como se mencionó anteriormente, las redes neuronales no funcionan bien con valores de entrada que no estén dentro del rango de valores de entrenamiento. Esto ocurre con la zona de los labios. Casi todas las simulaciones en ABAQUS presentan valores de desplazamiento en estas zonas más grandes que los valores objetivo (que son los que se introducen como entrada en la red neuronal). Por esta razón se simularon unos casos adicionales, rigidizando los labios, para que los valores objetivo estuvieran dentro del rango de valores de entrenamiento. Aún así, los resultados que se obtienen entrenando y simulando la red una vez añadidos los casos adicionales, no mejoran a los obtenidos anteriormente.

Capítulo 8:

CONCLUSIONES

Se comentan en este capítulo las conclusiones más importantes del trabajo realizado. En el primer apartado se enumeran algunas de los motivos que introducen errores en el modelo, en el ajuste de las propiedades de la ley de comportamiento del material hiperelástico, y a la hora de simular la cirugía. A continuación se comentan brevemente cuales han sido las principales dificultades que han aparecido a lo largo del proyecto.

8.1 Fuentes de error.

Se van comentar brevemente las posibles fuentes de error que aparecen en el ajuste de las propiedades por zonas. Se explican en orden cronológico según se presentan a lo largo del proyecto.

- Antes de comenzar la segmentación de los archivos del TAC, se realizó el *resample*. Esto reducía significativamente el tamaño del problema (tiempo de procesamiento, peso de los archivos), y reducía bastante el número de elementos que el modelo tendría posteriormente. La parte negativa es que al aumentar la distancia entre cortes en el TAC se pierde algo de precisión.

- A la hora de realizar los escaneados faciales al paciente, pueden aparecer errores aunque se haya seguido el protocolo que normaliza las condiciones en las que se realizan (descrito en el Capítulo 4), como por ejemplo: que la posición del paciente no sea totalmente vertical u horizontal, que el paciente no tenga los músculos de la cara relajados, que la posición de la cámara no sea la adecuada. A esto se le añade los fallos introduzca el equipo con el que se realizan los escaneados (aunque este es pequeño).

- Una vez abiertos los escaneaos con el programa NETFABB, es necesario realizarle un corte por un plano sagital que coincida por el plano medio. La elección de este plano es manual, por lo que el error que se puede cometer en este caso es grande.

- Para calcular la diferencia entre las deformadas de la cara en uno y otro caso, hay que superponer los cortes obtenidos. Para superponerlos se ha supuesto que la diferencia de desplazamientos entre ambos casos (de pie y tumbado boca abajo) en la frente y la parte superior de la nariz es nula. De esta manera se ha intentado que esa zona coincida, lo que no es totalmente cierto.

- Con los escaneados superpuestos se deben elegir una serie de puntos sobre ellos donde se van a calcular los desplazamientos. Aunque la elección de estos puntos se intenta que sea lo más representativa posible, coger unos u otros puntos siempre influirá mucho en el resultado final.

- Una vez exportado el modelo a ABAQUS, lo primero que se hacía era definir el vestíbulo de la cara (separando piel y hueso en una determinada zona). La elección de esta zona (definida en ABAQUS como un rectángulo) puede variar mucho.

- Con el modelo de elementos finitos ya completo, hay que seleccionar los puntos en los que se van a medir los desplazamientos. Teóricamente, estos puntos deberían coincidir con los puntos de medición que se seleccionaron en los escaneados faciales (puntos pertenecientes a un corte por un plano sagital que divida la cara en

dos mitades). En la práctica, que estos puntos sean exactamente los mismos es muy difícil, y su elección también condiciona el resultado final.

- Para poder ajustar las propiedades por zonas es necesario definir grupos de elementos a los que se les darán propiedades distintas. Estos grupos se eligen seleccionando directamente los elementos que los componen en ABAQUS CAE. Por esta razón puede haber gran variedad a la hora de elegirlos. Además las fronteras de los grupos que hay que seleccionar tampoco están totalmente definidas.

Una vez ajustadas las propiedades de los tejidos blandos se lleva a cabo la simulación de la cirugía. La forma en que se ha simulado la operación no es más que una simplificación del problema. El grupo de nodos al que se le ha impuesto el desplazamiento ha sido elegido manualmente, por lo tanto dicha elección no es fija. Además, en una operación de este tipo el corte que se realiza al paciente es mucho más complicado que el que se ha supuesto a la hora de simularlo. Aún así, la forma en que se alarga la mandíbula si es bastante similar.

8.2 Dificultades encontradas.

La mayor dificultad con la que nos hemos encontrado ha sido el contacto en la zona interior del vestíbulo. El problema del contacto en ABAQUS es uno de los más complicados de resolver. La principal razón de ello es que nunca se sabe exactamente en qué lugar está el fallo o qué es lo que no se está haciendo correctamente.

Otra de las dificultades apareció al usar las redes neuronales artificiales. Para poder aplicar de manera eficiente estos algoritmos en un problema tan complejo como el que nos ocupa (problema con cinco entradas y seis salidas) es necesario tener un amplio conocimiento de cómo funcionan estas redes. Este conocimiento no se refiere solo a la teoría de redes neuronales, sino también a su aplicación en Matlab (que ha sido la herramienta usada). Puesto que el proyecto no se basa en su uso, y que era la primera vez que trabajaba con redes neuronales artificiales, no se ha podido dedicar a su estudio el tiempo necesario que se requeriría para un aplicación óptima de las redes neuronales a nuestro problema.

8.3 Resultados.

Se describen brevemente en este apartado los principales resultados a los que se ha llegado con este proyecto.

- Ajuste manual.

Gracias a este procedimiento para encontrar las propiedades por zonas de la piel se consiguen los mejores resultados. Al ir variando los valores de m_2 en función de los desplazamientos obtenidos, se consigue que con cada simulación el error cometido sea menor. El inconveniente que presenta este sistema es que es muy costoso en tiempo, pues las propiedades se van afinando lentamente simulación a simulación. Como punto de partida se escogen los valores que da Barbarino para la ley de comportamiento de Rubin-Bodner.

- Redes neuronales.

El otro método usado para ajustar las propiedades de la piel es el uso de las redes neuronales, concretamente las que vienen implementadas en Matlab. Los mínimos errores obtenidos en este caso son algo mayores que en el caso anterior, aunque muy poco (del orden de décimas de milímetro). La ventaja principal que presenta esta forma de ajustar las propiedades de la cara frente a la anterior, es que no es necesario ir procesando los resultados de cada simulación individualmente. Basta con simular un determinado número de casos de carga, de forma que el parámetro m_2 en cada zona, cubra todo el rango de posibles valores que se esperan de él (asegurándonos que el m_2 buscado está dentro de los valores de las simulaciones).

- Simulación de la cirugía.

Debido a la gran complejidad a la hora de tratar el problema del contacto, hasta la fecha no se ha conseguido simular con éxito la cirugía buscada (desplazamiento de un centímetro en la zona delantera de la mandíbula inferior). Los mayores desplazamientos alcanzados han sido de tres milímetros. Se va a seguir trabajando para lograr simular la cirugía objetivo.

Otro resultado obtenido del trabajo realizado para simular la cirugía es que con el modelo inicial (modelo con elementos C3D4) obtenido de SIMPLEWARE, no se puede utilizar el mallado adaptativo en ABAQUS. Este tipo de elementos son incompatibles con esa opción del programa. Para poder utilizar el **ADAPTIVE MESH* es necesario exportar el modelo desde ScanFE con elementos hexaédricos C3D8. Esto lleva a que las superficies del modelo sean muy abruptas, lo que hace prácticamente imposible definir un contacto entre dos de ellas. Por estas razones, en el modelo final donde se simula la cirugía, no se ha usado el remallado. La no utilización de esta opción hace

que, al mover los nodos delanteros de la mandíbula, estos “arrastren” a todos los demás nodos de la mandíbula (salvó dos nodos que se fijan en la zona de la articulación temporomandibular para anclar el modelo), cuando lo ideal sería que esta deformación se concentrará en una región lo más pequeña posible de la mandíbula.

Bibliografía

1. ABAQUS Documentation.
2. Gerhard A. Holzapfel, Nonlinear Solid Mechanics: A Continuum Approach for Engineering.
3. G. G. Barbarino, Development and Validation of a Three - Dimensional Finite Element Model of the Face.
4. Rubin, M. B., and Bodner, S. R., 2002, "A Three-Dimensional Nonlinear Model for Dissipative Response of Soft Tissue," Int. J. Solids Struct., 39.19., pp. 5081–5099.
5. Las Redes Neuronales Artificiales – Fundamentos teóricos y aplicaciones prácticas. (Raquel Flores López y José Miguel Fernández Fernández).
6. Web del Laboratorio del Departamento de Informática (LDI) de la universidad Carlos III de Madrid.
7. PFC: "Desarrollo de una interfaz gráfica de redes neuronales usando Matlab."
8. Web de la clínica Universidad de Navarra: www.cun.es

Anexo I:

UHYPER: Modelo de comportamiento de Rubin-Bodner

```fortran
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C                                ANEXO I
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C      UHYPER DEL MODELO DE COMPORTAMIENTO HIPERELÁSTICO DE
C      RUBIN-BODNER.
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C
SUBROUTINE UHYPER(BI1,BI2,AJ,U,UI1,UI2,UI3,TEMP,NOEL,
1 CMNAME,INCMPFLAG,NUMSTATEV,STATEV,NUMFIELDV,FIELDV,
2 FIELDVINC,NUMPROPS,PROPS)
C
INCLUDE'ABA_PARAM.INC'
C
CHARACTER*80 CMNAME
      REAL*4 MU,Q,M1,M2,W
DIMENSION U(2),UI1(3),UI2(6),UI3(6),STATEV(*),FIELDV(*),
2 FIELDVINC(*),PROPS(*)

      MU=PROPS(1)
      Q=PROPS(2)
      M1=PROPS(3)
      M2=PROPS(4)
      W=PROPS(5)

C      user coding to define U,UI1,UI2,UI3,STATEV

            UI1(1)=(MU/(2.0))*M2*(1.0-W)*EXP(Q*(2.0*M1*(AJ-1.0-LOG(AJ))
      2     +(1.0-W)*M2*(BI1-3.0)))

            UI1(3)=(MU/(2.0))*2.0*M1*(1.0-(1.0/AJ))*EXP(Q*(2.0*M1*(AJ-
      2     1.0-LOG(AJ))+(1.0-W)*M2*(BI1-3.0)))

            UI2(1)=(MU/(2.0))*Q*(M2**2.0)*((1-W)**2.0)*EXP(Q*(2.0*M1*
      2     (AJ-1.0-LOG(AJ))+(1.0-W)*M2*(BI1-3.0)))

            UI2(3)=(MU/(2.0))*(2*M1/(AJ**2.0)+Q*(2*M1*(1-(1/AJ)))**2.0)
      2     *EXP(Q*(2.0*M1*(AJ-1.0-LOG(AJ))+(1.0-W)*M2*(BI1-3.0)))

            UI2(5)=(MU/(2.0))*Q*2.0*M1*(1-(1/AJ))*M2*(1-W)*EXP(Q*(2.0*
      2     M1*(AJ-1.0-LOG(AJ))+(1.0-W)*M2*(BI1-3.0)))

      UI3(1)=(MU/(2.0))*(Q**2.0)*(M2**2.0)*((1-W)**2.0)*2*M1*(1-
      2(1/AJ))*EXP(Q*(2.0*M1*(AJ-1.0-LOG(AJ))+(1.0-W)*M2*(BI1-3.0)))

            UI3(4)=(MU/(2.0))*M2*(1-W)*(2*M1*Q/(AJ**2.0)+(Q*2.0*M1*(1-
      2(1/AJ)))**2.0)*EXP(Q*(2.0*M1*(AJ-1.0-LOG(AJ))+(1.0-W)*M2*(BI1-
3     3.0)))

            UI3(6)=(Q*2.0*M1*(1-(1/AJ))*(2*M1*Q/(AJ**2.0)+(Q*2.0*M1*(1-
      2     (1/AJ)))**2.0)+(-4.0*M1*Q/(AJ**3.0)+8*(M1**2.0)*(Q**2.0)*
      3     (1-(1/AJ))/(AJ**2.0)))*(MU/(2.0*Q))*EXP(Q*(2.0*M1*(AJ-1.0-
      4     lOG(AJ))+(1.0-W)*M2*(BI1-3.0)))


            U(1)=(MU/(2.0*Q))*(EXP(Q*(2.0*M1*(AJ-1.0-LOG(AJ))+(1.0-W)*
      2     M2*(BI1-3.0)))-1)

RETURN
END
```

Anexo II:

TABLAS DE RESULTADOS DE LAS SIMULACIONES

Valores de m_2 por zonas y errores cometidos en las primeras simulaciones.

SIMULACIÓN	m_2 Mentón	m_2 Labio inferior	m_2 Labio superior	m_2 Nariz	m_2 Papada + Cuello	m_2 Resto	Error [mm]
1	0,001	0,001	0,001	0,001	0,001	0,001	0,24383898
2	0,001	0,001	0,0001	0,001	0,001	0,001	1,2319007
3	0,001	0,0001	0,001	0,001	0,001	0,001	0,32448283
4	0,0001	0,001	0,001	0,001	0,001	0,001	0,25396893
5	0,001	0,001	0,001	0,0001	0,001	0,001	0,24768804
6	0,001	0,001	0,001	0,001	0,0001	0,001	0,23749638
7	0,001	0,001	0,001	0,001	0,001	0,0005	0,54254133
8	0,001	0,0001	0,0001	0,001	0,001	0,001	1,30039924
9	0,0001	0,001	0,0001	0,001	0,001	0,001	1,22770881
10	0,001	0,001	0,0001	0,0001	0,001	0,001	1,23701736
11	0,001	0,001	0,0001	0,001	0,0001	0,001	1,23172811
12	0,001	0,001	0,0001	0,001	0,001	0,0005	1,45829309
13	0,0001	0,0001	0,001	0,001	0,001	0,001	0,43774995
14	0,001	0,0001	0,001	0,0001	0,001	0,001	0,32861594
15	0,001	0,0001	0,001	0,001	0,0001	0,001	0,30273804
16	0,001	0,0001	0,001	0,001	0,001	0,0005	0,58146056
17	0,0001	0,001	0,001	0,0001	0,001	0,001	0,25773078
18	0,0001	0,001	0,001	0,001	0,0001	0,001	0,2647628
19	0,0001	0,001	0,001	0,001	0,001	0,0005	0,57484456
20	0,001	0,001	0,001	0,0001	0,0001	0,001	0,2415975
21	0,001	0,001	0,001	0,0001	0,001	0,0005	0,55920415
22	0,001	0,001	0,001	0,001	0,0001	0,0005	0,53791408
23	0,0001	0,0001	0,0001	0,001	0,001	0,001	1,33082274
24	0,001	0,0001	0,0001	0,0001	0,001	0,001	1,30577682
25	0,001	0,0001	0,0001	0,001	0,0001	0,001	1,29906586
26	0,001	0,0001	0,0001	0,001	0,001	0,0005	1,52904393
27	0,0001	0,0001	0,001	0,0001	0,001	0,001	0,44091481
28	0,0001	0,0001	0,001	0,001	0,0001	0,001	0,42591227
29	0,0001	0,0001	0,001	0,001	0,001	0,0005	0,68081134
30	0,0001	0,001	0,001	0,0001	0,0001	0,001	0,26854562
31	0,0001	0,001	0,001	0,0001	0,001	0,0005	0,59063072
32	0,001	0,001	0,001	0,0001	0,0001	0,0005	0,55496794
33	0,0001	0,0001	0,0001	0,0001	0,001	0,001	1,33611851
34	0,0001	0,0001	0,0001	0,001	0,0001	0,001	1,33390009
35	0,0001	0,0001	0,0001	0,001	0,001	0,0005	1,56276579
36	0,0001	0,0001	0,001	0,0001	0,0001	0,001	0,42929235

37	0,0001	0,0001	0,001	0,0001	0,001	0,0005	0,6961941
38	0,0001	0,001	0,001	0,0001	0,0001	0,0005	0,59910747
39	0,001	0,0001	0,001	0,0001	0,0001	0,0005	0,58885965
40	0,0001	0,0001	0,001	0,001	0,0001	0,0005	0,68533189
41	0,0001	0,0001	0,001	0,0001	0,001	0,0005	0,6961941
42	0,001	0,001	0,0001	0,0001	0,0001	0,0005	1,47274239
43	0,0001	0,001	0,0001	0,001	0,0001	0,0005	1,45918885
44	0,0001	0,001	0,0001	0,0001	0,001	0,0005	1,469045
45	0,001	0,001	0,002	0,001	0,001	0,001	0,16791946
46	0,001	0,002	0,001	0,001	0,001	0,001	0,22241401
47	0,002	0,001	0,001	0,001	0,001	0,001	0,23687939
48	0,001	0,001	0,001	0,002	0,001	0,001	0,24368591
49	0,001	0,001	0,001	0,001	0,002	0,001	0,244443
50	0,001	0,001	0,001	0,001	0,001	0,002	0,13194147
51	0,001	0,002	0,002	0,001	0,001	0,001	0,14530379
52	0,002	0,001	0,002	0,001	0,001	0,001	0,1571932
53	0,001	0,001	0,002	0,002	0,001	0,001	0,16874259
54	0,001	0,001	0,002	0,001	0,002	0,001	0,16924241
55	0,001	0,001	0,002	0,001	0,001	0,002	0,13625397
56	0,002	0,002	0,001	0,001	0,001	0,001	0,21682482
57	0,001	0,002	0,001	0,002	0,001	0,001	0,22243394
58	0,001	0,002	0,001	0,001	0,002	0,001	0,22239834
59	0,001	0,002	0,001	0,001	0,001	0,002	0,12627823
60	0,002	0,001	0,001	0,002	0,001	0,001	0,23673027
61	0,002	0,001	0,001	0,001	0,002	0,001	0,23648704
62	0,002	0,001	0,001	0,001	0,001	0,002	0,13129566
63	0,001	0,001	0,001	0,002	0,002	0,001	0,24430734
64	0,001	0,001	0,001	0,002	0,001	0,002	0,13518202
65	0,001	0,001	0,001	0,001	0,002	0,002	0,13252988
66	0,002	0,002	0,002	0,001	0,001	0,001	0,13673313
67	0,001	0,002	0,002	0,002	0,001	0,001	0,14643016
68	0,001	0,002	0,002	0,001	0,002	0,001	0,14588722
69	0,001	0,002	0,002	0,001	0,001	0,002	0,1361091
70	0,002	0,001	0,002	0,002	0,001	0,001	0,15808795
71	0,002	0,001	0,002	0,001	0,002	0,001	0,15735963
72	0,002	0,001	0,002	0,001	0,001	0,002	0,13554753
73	0,001	0,001	0,002	0,002	0,002	0,001	0,17006991
74	0,001	0,001	0,002	0,002	0,001	0,002	0,13978597
75	0,001	0,001	0,002	0,001	0,002	0,002	0,13708174
76	0,002	0,002	0,001	0,002	0,001	0,001	0,21685741
77	0,002	0,002	0,001	0,001	0,002	0,001	0,21601675
78	0,002	0,002	0,001	0,001	0,001	0,002	0,12728235
79	0,001	0,002	0,001	0,002	0,002	0,001	0,22243467

80	0,001	0,002	0,001	0,002	0,001	0,002	0,12974411
81	0,001	0,002	0,001	0,001	0,002	0,002	0,12628985
82	0,002	0,001	0,001	0,002	0,002	0,001	0,23635449
83	0,002	0,001	0,001	0,002	0,001	0,002	0,13455389
84	0,002	0,001	0,001	0,001	0,002	0,002	0,13146025
85	0,001	0,001	0,001	0,002	0,002	0,002	0,13576115
86	0,002	0,002	0,002	0,002	0,001	0,001	0,13795607
87	0,002	0,002	0,002	0,001	0,002	0,001	0,13641913
88	0,002	0,002	0,002	0,001	0,001	0,002	0,13708007
89	0,001	0,002	0,002	0,002	0,002	0,001	0,14702533
90	0,001	0,002	0,002	0,002	0,001	0,002	0,13968056
91	0,001	0,002	0,002	0,001	0,002	0,002	0,13639802
92	0,002	0,001	0,002	0,002	0,002	0,001	0,15826853
93	0,002	0,001	0,002	0,002	0,001	0,002	0,13910142
94	0,002	0,001	0,002	0,001	0,002	0,002	0,13604826
95	0,001	0,001	0,002	0,002	0,002	0,002	0,14059507
96	0,002	0,002	0,001	0,002	0,002	0,001	0,21606514
97	0,002	0,002	0,001	0,002	0,001	0,002	0,13072469
98	0,002	0,002	0,001	0,001	0,002	0,002	0,1271414
99	0,002	0,002	0,002	0,002	0,002	0,002	0,14084137

Valores de m_2 por zonas y errores cometidos en las simulaciones empleadas para el ajuste manual de las propiedades de la piel (tomando como base la simulación número 86).

SIMULACIÓN	m_2 Mentón	m_2 Labio inferior	m_2 Labio superior	m_2 Nariz	m_2 Papada + Cuello	m_2 Resto	Error [mm]
86 - 1	0,002	0,003	0,003	0,002	0,001	0,001	0,11110091
86 - 2	0,002	0,0035	0,0035	0,002	0,001	0,001	0,10566114
86 - 3	0,002	0,004	0,004	0,002	0,001	0,001	0,10270069
86 - 4	0,002	0,004	0,0035	0,002	0,001	0,001	0,10393483
86 - 5	0,002	0,006	0,006	0,002	0,001	0,001	0,10142586
86 - 6	0,002	0,008	0,008	0,002	0,001	0,001	0,1054454
86 - 7	0,002	0,01	0,006	0,002	0,001	0,001	0,10014352
86 - 8	0,002	0,012	0,006	0,002	0,001	0,001	0,1000307
86 - 9	0,002	0,012	0,005	0,002	0,001	0,001	0,09685478
86 - 10	0,002	0,014	0,005	0,002	0,001	0,001	0,09658065
86 - 11	0,002	0,016	0,004	0,002	0,001	0,001	0,09307532
86 - 12	0,002	0,018	0,004	0,002	0,001	0,001	0,09269701
86 - 13	0,002	0,02	0,004	0,002	0,001	0,001	0,09240917
86 - 14	0,002	0,022	0,003	0,002	0,001	0,001	0,09030503
86 - 15	0,002	0,024	0,003	0,002	0,001	0,001	0,08982174
86 - 16	0,002	0,026	0,003	0,002	0,001	0,001	0,08940481

Anexo III:

COMANDOS DEL ARCHIVO DE CARGA DE LA SIMULACIÓN NÚMERO 86

```
*INCLUDE,INPUT=NPyH.inp
*INCLUDE,INPUT=PIEL.inp
*INCLUDE,INPUT=HUESO.inp
*INCLUDE,INPUT=LAB_SUP.inp
*INCLUDE,INPUT=LAB_INF.inp
*INCLUDE,INPUT=MENTON.inp
*INCLUDE,INPUT=NARIZ.inp
*INCLUDE,INPUT=PAPADA_CUELLO.inp
*INCLUDE,INPUT=RESTO.inp
*INCLUDE,INPUT=NODOS_CC_CUELLO.inp
**************************************************************
*ELSET,ELSET=ETODO
PIEL,HUESO
*NSET,NSET=NTODO
NODOS_PIELYHUESO
**************************************************************
**  Nodos donde se evalúan los desplazamientos producidos  **
**  Están situados en la superficie de la cara, sobre un   **
**  corte sagital imaginario que dividiría la cara por la mitad  **
**************************************************************
*NSET,NSET=n_menton
156298
*NSET,NSET=n_labioinf
172760
*NSET,NSET=n_labiosup
184687
*NSET,NSET=n_debajo_nariz
202726
*NSET,NSET=n_punta_nariz
225952
**************************************************************
**                MATERIALES                    **
**************************************************************
*solid section,elset=HUESO,material=MAT_HUESO
************solid section,elset=PIEL,material=MAT_PIEL************
*solid section,elset=LAB_SUP,material=MAT_LAB_SUP
*solid section,elset=LAB_INF,material=MAT_LAB_INF
*solid section,elset=MENTON,material=MAT_MENTON
*solid section,elset=NARIZ,material=MAT_NARIZ
*solid section,elset=PAPADA_CUELLO,material=MAT_PAPADA_CUELLO
```

```
***************Mismo material para PAPADA y CUELLO**************
*solid section,elset=RESTO,material=MAT_RESTO
*RIGID BODY,REF NODE=101588,ELSET=HUESO
**************************************************************
*material,name=MAT_HUESO
*ELASTIC,TYPE=ISOTROPIC
17E9,0.3
**************************************************************
*material,name=MAT_LAB_SUP
*DENSITY
1000,
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
3.7e6,25,595,0.002,0.0
*****************************************************
*material,name=MAT_LAB_INF
*DENSITY
1000,
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
3.7e6,25,595,0.002,0.0
*****************************************************
*material,name=MAT_MENTON
*DENSITY
1000,
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
3.7e6,25,595,0.002,0.0
*****************************************************
*material,name=MAT_NARIZ
*DENSITY
1000,
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
3.7e6,25,595,0.002,0.0
*****************************************************
*material,name=MAT_PAPADA_CUELLO
*DENSITY
1000,
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
3.7e6,25,595,0.001,0.0
*****************************************************
*material,name=MAT_RESTO
*DENSITY
1000,
```

```
*HYPERELASTIC,USER,TYPE=COMPRESSIBLE,PROPERTIES=5
3.7e6,25,595,0.001,0.0
**********************************************************************
**                  STEP  1    (Levantado)              **
**********************************************************************
*STEP,NLGEOM,INC=100
*STATIC
0.1,1.0,0.01,0.2
*BOUNDARY,OP=NEW
NODOS_CC_CUELLO,1,3,0.0
*DLOAD, OP=NEW
PIEL,GRAV,13.873435,0,-1,-1
*NODE PRINT,FREQUENCY=1
*EL PRINT,FREQUENCY=1
*OUTPUT,FIELD, FREQ=99999
*CONTACT OUTPUT,VARIABLE=PRESELECT
*ELEMENT OUTPUT, ELSET=ETODO
S,E
*NODE OUTPUT, NSET=NTODO
U,RF
*END STEP
**********************************************************************
**                 STEP  2           (Tumbado hacia abajo)      **
**********************************************************************
*STEP,NLGEOM,INC=100
*STATIC
0.1,1.0,0.01,0.2
*BOUNDARY,OP=NEW
NODOS_CC_CUELLO,1,3,0.0
*DLOAD,OP=NEW
PIEL,GRAV,19.62,0,-1,0
*NODE PRINT,FREQUENCY=1
*EL PRINT,FREQUENCY=1
*OUTPUT,FIELD, FREQ=99999
*CONTACT OUTPUT,VARIABLE=PRESELECT
*ELEMENT OUTPUT, ELSET=ETODO
S,E
*NODE OUTPUT, NSET=NTODO
U,RF
*END STEP
```

Anexo IV:

PROGRAMA DE MATLAB PARA LA RED NEURONAL

```
%%%% Redes neuronales
clc
clear all
format long g

%%%% Lee los datos para entrenar la red

%"p" contiene en columnas las diferencias de desplazamientos de cada
%simulación y en cada fila, las diferencias de desplazamientos según
%las zonas: mentón, labio inferior, labio superior, debajo nariz, pta
%nariz.

load 'p.m';

%t contienen en filas los valores del parámetro m2 por zonas, en este
%orden: mentón, labio inferior, labio superior, nariz, papada y
%resto. Por  columnas tiene los valores del parámetro m2 para cada
%simulación.
load 't.m';

%%%% Desplazamientos de los escaneados. Referencia a conseguir.
ppp=[0.24
    0.914
    1.492
    1.103
    0.704];



% Back Propagation - 10 neuronas en la primera capa, 30 en la segunda
% y 20 en la tercera

net = newff(p,t,[10 30 20],{'purelin','purelin','purelin'},'trainlm');

% Definición de algunas variables que el programa usa para entrenar la
% red

net.trainParam.goal = 1e-9;
net.trainParam.epochs = 100;
net.trainParam.min_grad = 1e-10;
net.trainParam.max_fail = 20;
net.trainParam.mu = 1e-3;

% Proporcion de casos para entrenar: 0.8, para validar: 0.1 y
% verificar: 0.1

net.divideParam.trainRatio = 0.8;
net.divideParam.valRatio = 0.1;
net.divideParam.testRatio = 0.1;
net = train(net,p,t);


%sim (net,ppp)
```

Anexo V:

RUTINAS DE FORTRAN

```
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     %                          Rutina 1                                    %
C     %                   "comunes_y_duplica.for"                            %
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C
C     Este programa duplica los nodos del vestíbulo comunes a los grupos
C     PIEL y HUESO. De esta forma los nodos se pueden mover de forma
C     independiente. Los nuevos nodos, que se le asignan al grupo HUESO,
C     se numeran a partir del 600000.
C



      PROGRAM LECTURA

      INTEGER*4,PARAMETER::   MAX=1000000
      INTEGER*4 I,J,K,L,AUX,AUX2,AUX3,EXITO,CONT_NODOS2,
     1INICIODEFINICION
      REAL*4 NODOS2_MAX(MAX,4),NODOS2[ALLOCATABLE](:,:),
     1DEF_NVEST[ALLOCATABLE](:,:),NDUPL[ALLOCATABLE](:,:)
      INTEGER*4 CONT_NPIEL,NPIEL_MAX(MAX,16),NPIEL[ALLOCATABLE](:,:),
     1CONT_NHUESO,NHUESO_MAX(MAX,16),NHUESO[ALLOCATABLE](:,:),
     2CONT_HUESO,HUESO_MAX(MAX,5),HUESO[ALLOCATABLE](:,:),
     3CONT_NAUX_VEST,NAUX_VEST_MAX(MAX,16),NAUX_VEST[ALLOCATABLE](:,:),
     3COMUNES(100000,1),
     4NVEST(100000,1),
     5HUESO2[ALLOCATABLE](:,:)

C     LEE LOS NODOS DEL ARCHIVO NAUX_VEST.txt Y LOS GUARDA EN LA
C     VARIABLE "NAUX_VEST".
      OPEN (unit=101,file='C:\FORTRAN\NAUX_VEST.txt',status='unknown')
      CONT_NAUX_VEST=0
      DO I=1,MAX
          READ(101,*,END=14) (NAUX_VEST_MAX(I,J),J=1,16)
      CONT_NAUX_VEST=CONT_NAUX_VEST+1
      ENDDO
   14 WRITE(*,*) 'Terminada lectura de nodos',CONT_NAUX_VEST,
     2'FILAS NAUX_VEST'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NAUX_VEST(CONT_NAUX_VEST,16),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_NAUX_VEST
          NAUX_VEST(I,1:16)=NAUX_VEST_MAX(I,1:16)
      ENDDO

C     LEE LOS NODOS DEL ARCHIVO NPIEL.txt Y LOS GUARDA EN LA
C     VARIABLE "NPIEL".
      OPEN (unit=101,file='C:\FORTRAN\NPIEL.txt',status='unknown')
      CONT_NPIEL=0
      DO I=1,MAX
          READ(101,*,END=15) (NPIEL_MAX(I,J),J=1,16)
      CONT_NPIEL=CONT_NPIEL+1
      ENDDO
   15 WRITE(*,*) 'Terminada lectura de nodos',CONT_NPIEL,'FILAS NPIEL'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NPIEL(CONT_NPIEL,16),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
```

```
      ENDIF

      DO I=1,CONT_NPIEL
          NPIEL(I,1:16)=NPIEL_MAX(I,1:16)
      ENDDO

C     LEE LOS NODOS DEL ARCHIVO NHUESO.txt Y LOS GUARDA EN LA
C     VARIABLE "NHUESO".
      OPEN (unit=101,file='C:\FORTRAN\NHUESO.txt',status='unknown')
      CONT_NHUESO=0
      DO I=1,MAX
          READ(101,*,END=16) (NHUESO_MAX(I,J),J=1,16)
      CONT_NHUESO=CONT_NHUESO+1
      ENDDO
   16 WRITE(*,*) 'Terminada lectura de nodos',CONT_NHUESO,'FILAS NHUESO'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NHUESO(CONT_NHUESO,16),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_NHUESO
          NHUESO(I,1:16)=NHUESO_MAX(I,1:16)
      ENDDO

C     LEE LOS NODOS DEL ARCHIVO NODOS2.txt Y LOS GUARDA EN LA
C     VARIABLE "NODOS2".
      OPEN (unit=101,file='C:\FORTRAN\NODOS2.txt',status='unknown')
      CONT_NODOS2=0
      DO I=1,MAX
          READ(101,*,END=17) (NODOS2_MAX(I,J),J=1,4)
      CONT_NODOS2=CONT_NODOS2+1
      ENDDO
   17 WRITE(*,*) 'Terminada lectura de nodos',CONT_NODOS2,'FILAS NODOS2'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NODOS2(CONT_NODOS2,4),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_NODOS2
          NODOS2(I,1:4)=NODOS2_MAX(I,1:4)
      ENDDO

C     LEE EL GRUPO DE ELEMENTOS DE LA PIEL DEL ARCHIVO
C     HUESO.txt Y LOS GUARDA EN LA VARIABLE "HUESO" Y
C     SE COPIA HUESO EN HUESO2.

      OPEN (unit=101,file='C:\FORTRAN\HUESO.txt',status='unknown')
      CONT_HUESO=0
      DO I=1,MAX
          READ(101,*,END=18) (HUESO_MAX(I,J),J=1,5)
      CONT_HUESO=CONT_HUESO+1
      ENDDO
   18 WRITE(*,*) 'Terminada lectura de nodos',CONT_HUESO,'FILAS HUESO'
      CLOSE(unit=101,status='keep')

      ALLOCATE (HUESO(CONT_HUESO,5),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF
```

```
      DO I=1,CONT_HUESO
          HUESO(I,1:5)=HUESO_MAX(I,1:5)
      ENDDO

      ALLOCATE (HUESO2(CONT_HUESO,5),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_HUESO
          HUESO2(I,1:5)=HUESO_MAX(I,1:5)
      ENDDO

C     ALMACENA EN UN VECTOR COLUMNA TODOS LOS NODOS COMUNES A NPIEL
C     Y NHUESO
      AUX=0
      DO I=1,CONT_NPIEL
          DO J=1,16
              IF(NPIEL(I,J).NE.0.0)THEN
                  DO K=1,CONT_NHUESO
                      DO L=1,16
                          IF(NPIEL(I,J).EQ.NHUESO(K,L))THEN
                          AUX=AUX+1
                          COMUNES(AUX,1)=NPIEL(I,J)
                          EXIT
                          ENDIF
                      ENDDO
                  ENDDO
              ENDIF
          ENDDO
      ENDDO

C     SELECCIONA LOS NODOS DEL VECTOR COMUNES, DENTRO DEL GRUPO
C     "NAUX_VEST". PARA OBTENER LOS NODOS COMUNES ENTRE PIEL Y HUESO
C     DEL VESTIBULO: "NVEST". Y SE GUARDA MEMORIA PARA DEF_NVEST Y NDUPL.
      AUX2=0
      DO I=1,AUX
          DO J=1,CONT_NAUX_VEST
              DO K=1,16
                  IF(COMUNES(I,1).EQ.NAUX_VEST(J,K))THEN
                      AUX2=AUX2+1
                      NVEST(AUX2,1)=NAUX_VEST(J,K)
                      EXIT
                  ENDIF
              ENDDO
          ENDDO
      ENDDO

      ALLOCATE (DEF_NVEST(AUX2,4),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      ALLOCATE (NDUPL(AUX2,4),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

C     SELECCIONA LOS NODOS DEL VECTOR "NVEST", DE ENTRE LA DEFINICIÓN
C     DE TODOS LOS NODOS EN "NODOS2". PARA OBTENER LA DEFINICIÓN DE
C     LOS NODOS COMUNES ENTRE PIEL Y HUESO DEL VESTIBULO
      DO I=1,AUX2
          DO J=1,CONT_NODOS2
              IF(NVEST(I,1).EQ.INT(NODOS2(J,1)))THEN
```

```
                DO K=1,4
                DEF_NVEST(I,K)=NODOS2(J,K)
                ENDDO
                EXIT
            ENDIF
        ENDDO
    ENDDO

C   DEFINE EL NUMERO DE NODO A PARTIR DEL CUAL SE COMIENZA LA DUPLICACIÓN
    INICIODEFINICION=600000

C   SE DUPLICAN LOS NODOS DE LA VARIABLE "DEF_NVEST", ALMACENANDOSE LOS
C   DUPLICADOS EN LA VARIABLE "NDUPL"
    AUX3=INICIODEFINICION+1
    DO I=1,AUX2
        NDUPL(I,1)=AUX3
        AUX3=AUX3+1
    ENDDO
    DO I=1,AUX2
        DO K=2,4
            NDUPL(I,K)=DEF_NVEST(I,K)
        ENDDO
    ENDDO



C   LECTURA ELTO A ELTO DE "HUESO"
C   SE BUSCAN DE ENTRE LOS NODOS QUE DEFINEN ESTOS ELEMENTOS
C   IGUALES A LOS DE "NVEST"
C   CDO SE ENCUENTRA SE VA ACTUALIZANDO LA VBLE "HUESO2" CON LOS NODOS
C   DUPLICADOS "NDUPL"
C

    DO I=1,CONT_HUESO
        DO K=1,AUX2
            IF(NVEST(K,1).EQ.HUESO(I,2))THEN
                HUESO2(I,2)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,3))THEN
                HUESO2(I,3)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,4))THEN
                HUESO2(I,4)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,5))THEN
                HUESO2(I,5)=INT(NDUPL(K,1))
            ENDIF
        ENDDO
    ENDDO



C   ESCRIBE EL VECTOR "COMUNES", CON LOS NODOS QUE PERTENECEN
C   A PIEL Y HUESO
    OPEN (unit=101,file='COMUNES.txt',status='unknown')
    DO I=1,AUX
        WRITE(101,12) COMUNES(I,1)
    ENDDO
    CLOSE(unit=101,status='keep')
12  FORMAT(I8)
C   ESCRIBE EL VECTOR "NVEST", CON LOS NODOS QUE PERTENECEN
C   A PIEL Y HUESO EN EL VESTIBULO
```

```
      OPEN (unit=101,file='NVEST.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,12) NVEST(I,1)
      ENDDO
      CLOSE(unit=101,status='keep')

C     ESCRIBE LA DEFINICIÓN DE LOS NODOS COMUNES A PIEL Y HUESO
C     DEL VESTIBULO, QUE SE ENCUENTRAN EN LA VBLE "DEF_NVEST"
      OPEN (unit=101,file='DEF_NVEST.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,11) INT(DEF_NVEST(I,1)),',',DEF_NVEST(I,2),',',
     1    DEF_NVEST(I,3),',',DEF_NVEST(I,4)
      ENDDO
      CLOSE(unit=101,status='keep')
11    FORMAT(I8,A1,E16.8,A1,E16.8,A1,E16.8)

C     ESCRIBE LA DEFINICIÓN DE LOS NODOS DUPLICADOS: "NDUPL"
      OPEN (unit=101,file='NDUPL.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,11) INT(NDUPL(I,1)),',',NDUPL(I,2),',',NDUPL(I,3),
     1    ',',NDUPL(I,4)
      ENDDO
      CLOSE(unit=101,status='keep')

C     ESCRIBE LA NUEVA DEFINICIÓN DE "HUESO" ("HUESO2")
      OPEN (unit=101,file='HUESO2.txt',status='unknown')
      DO I=1,CONT_HUESO
          WRITE(101,13) HUESO2(I,1),',',HUESO2(I,2),',',HUESO2(I,3),
     1    ',',HUESO2(I,4),',',HUESO2(I,5)
      ENDDO
      CLOSE(unit=101,status='keep')
13    FORMAT(I8,A1,I8,A1,I8,A1,I8,A1,I8)

C     ESCRIBE EL GRUPO DE NODOS DUPLICADOS: "GNDUPL"
      OPEN (unit=101,file='GNDUPL.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,12) INT(NDUPL(I,1))
      ENDDO
      CLOSE(unit=101,status='keep')


      END
```

```
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     %                          Rutina 2                                    %
C     %                    "comunes_y_duplica_C3D8.for"                      %
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C
C     Este programa hace lo mismo que el anterior pero para el modelo
C     de elementos C3D8.
C



      PROGRAM LECTURA

      INTEGER*4,PARAMETER::   MAX=1000000
      INTEGER*4 I,J,K,L,AUX,AUX2,AUX3,EXITO,CONT_NODOS2,
     1INICIODEFINICION
      REAL*4 NODOS2_MAX(MAX,4),NODOS2[ALLOCATABLE](:,:),
     1DEF_NVEST[ALLOCATABLE](:,:),NDUPL[ALLOCATABLE](:,:)
      INTEGER*4 CONT_NPIEL,NPIEL_MAX(MAX,16),NPIEL[ALLOCATABLE](:,:),
     1CONT_NHUESO,NHUESO_MAX(MAX,16),NHUESO[ALLOCATABLE](:,:),
     2CONT_HUESO,HUESO_MAX(MAX,9),HUESO[ALLOCATABLE](:,:),
     3CONT_NAUX_VEST,NAUX_VEST_MAX(MAX,16),NAUX_VEST[ALLOCATABLE](:,:),
     3COMUNES(100000,1),
     4NVEST(100000,1),
     5HUESO2[ALLOCATABLE](:,:)

C     LEE LOS NODOS DEL ARCHIVO NAUX_VEST.txt Y LOS GUARDA EN LA
C     VARIABLE "NAUX_VEST".
      OPEN (unit=101,file='C:\FORTRAN\NAUX_VEST.txt',status='unknown')
      CONT_NAUX_VEST=0
      DO I=1,MAX
          READ(101,*,END=14) (NAUX_VEST_MAX(I,J),J=1,16)
      CONT_NAUX_VEST=CONT_NAUX_VEST+1
      ENDDO
   14 WRITE(*,*) 'Terminada lectura de nodos',CONT_NAUX_VEST,
     2'FILAS NAUX_VEST'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NAUX_VEST(CONT_NAUX_VEST,16),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_NAUX_VEST
          NAUX_VEST(I,1:16)=NAUX_VEST_MAX(I,1:16)
      ENDDO

C     LEE LOS NODOS DEL ARCHIVO NPIEL.txt Y LOS GUARDA EN LA
C     VARIABLE "NPIEL".
      OPEN (unit=101,file='C:\FORTRAN\NPIEL.txt',status='unknown')
      CONT_NPIEL=0
      DO I=1,MAX
          READ(101,*,END=15) (NPIEL_MAX(I,J),J=1,16)
      CONT_NPIEL=CONT_NPIEL+1
      ENDDO
   15 WRITE(*,*) 'Terminada lectura de nodos',CONT_NPIEL,'FILAS NPIEL'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NPIEL(CONT_NPIEL,16),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF
```

```
      DO I=1,CONT_NPIEL
          NPIEL(I,1:16)=NPIEL_MAX(I,1:16)
      ENDDO

C     LEE LOS NODOS DEL ARCHIVO NHUESO.txt Y LOS GUARDA EN LA
C     VARIABLE "NHUESO".
      OPEN (unit=101,file='C:\FORTRAN\NHUESO.txt',status='unknown')
      CONT_NHUESO=0
      DO I=1,MAX
          READ(101,*,END=16) (NHUESO_MAX(I,J),J=1,16)
      CONT_NHUESO=CONT_NHUESO+1
      ENDDO
   16 WRITE(*,*) 'Terminada lectura de nodos',CONT_NHUESO,'FILAS NHUESO'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NHUESO(CONT_NHUESO,16),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_NHUESO
          NHUESO(I,1:16)=NHUESO_MAX(I,1:16)
      ENDDO

C     LEE LOS NODOS DEL ARCHIVO NODOS2.txt Y LOS GUARDA EN LA
C     VARIABLE "NODOS2".
      OPEN (unit=101,file='C:\FORTRAN\NODOS2.txt',status='unknown')
      CONT_NODOS2=0
      DO I=1,MAX
          READ(101,*,END=17) (NODOS2_MAX(I,J),J=1,4)
      CONT_NODOS2=CONT_NODOS2+1
      ENDDO
   17 WRITE(*,*) 'Terminada lectura de nodos',CONT_NODOS2,'FILAS NODOS2'
      CLOSE(unit=101,status='keep')

      ALLOCATE (NODOS2(CONT_NODOS2,4),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_NODOS2
          NODOS2(I,1:4)=NODOS2_MAX(I,1:4)
      ENDDO

C     LEE EL GRUPO DE ELEMENTOS DE LA PIEL DEL ARCHIVO
C     HUESO.txt Y LOS GUARDA EN LA VARIABLE "HUESO" Y
C     SE COPIA HUESO EN HUESO2.

      OPEN (unit=101,file='C:\FORTRAN\HUESO.txt',status='unknown')
      CONT_HUESO=0
      DO I=1,MAX
          READ(101,*,END=18) (HUESO_MAX(I,J),J=1,9)
      CONT_HUESO=CONT_HUESO+1
      ENDDO
   18 WRITE(*,*) 'Terminada lectura de nodos',CONT_HUESO,'FILAS HUESO'
      CLOSE(unit=101,status='keep')

      ALLOCATE (HUESO(CONT_HUESO,9),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_HUESO
          HUESO(I,1:9)=HUESO_MAX(I,1:9)
```

```
      ENDDO

      ALLOCATE (HUESO2(CONT_HUESO,9),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_HUESO
          HUESO2(I,1:9)=HUESO_MAX(I,1:9)
      ENDDO

C     ALMACENA EN UN VECTOR COLUMNA TODOS LOS NODOS COMUNES A NPIEL
C     Y NHUESO
      AUX=0
      DO I=1,CONT_NPIEL
          DO J=1,16
              IF(NPIEL(I,J).NE.0.0)THEN
                  DO K=1,CONT_NHUESO
                      DO L=1,16
                          IF(NPIEL(I,J).EQ.NHUESO(K,L))THEN
                          AUX=AUX+1
                          COMUNES(AUX,1)=NPIEL(I,J)
                          EXIT
                          ENDIF
                      ENDDO
                  ENDDO
              ENDIF
          ENDDO
      ENDDO

C     SELECCIONA LOS NODOS DEL VECTOR COMUNES, DENTRO DEL GRUPO
C     "NAUX_VEST". PARA OBTENER LOS NODOS COMUNES ENTRE PIEL Y HUESO
C     DEL VESTIBULO: "NVEST". Y SE GUARDA MEMORIA PARA DEF_NVEST Y NDUPL.
      AUX2=0
      DO I=1,AUX
          DO J=1,CONT_NAUX_VEST
              DO K=1,16
                  IF(COMUNES(I,1).EQ.NAUX_VEST(J,K))THEN
                      AUX2=AUX2+1
                      NVEST(AUX2,1)=NAUX_VEST(J,K)
                      EXIT
                  ENDIF
              ENDDO
          ENDDO
      ENDDO

      ALLOCATE (DEF_NVEST(AUX2,4),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      ALLOCATE (NDUPL(AUX2,4),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

C     SELECCIONA LOS NODOS DEL VECTOR "NVEST", DE ENTRE LA DEFINICIÓN
C     DE TODOS LOS NODOS EN "NODOS2". PARA OBTENER LA DEFINICIÓN DE
C     LOS NODOS COMUNES ENTRE PIEL Y HUESO DEL VESTIBULO
      DO I=1,AUX2
          DO J=1,CONT_NODOS2
              IF(NVEST(I,1).EQ.INT(NODOS2(J,1)))THEN
                  DO K=1,4
                  DEF_NVEST(I,K)=NODOS2(J,K)
```

```
                ENDDO
                EXIT
            ENDIF
        ENDDO
    ENDDO

C   DEFINE EL NUMERO DE NODO A PARTIR DEL CUAL SE COMIENZA LA DUPLICACIÓN
    INICIODEFINICION=400000

C   SE DUPLICAN LOS NODOS DE LA VARIABLE "DEF_NVEST", ALMACENANDOSE LOS
C   DUPLICADOS EN LA VARIABLE "NDUPL"
    AUX3=INICIODEFINICION+1
    DO I=1,AUX2
        NDUPL(I,1)=AUX3
        AUX3=AUX3+1
    ENDDO
    DO I=1,AUX2
        DO K=2,4
            NDUPL(I,K)=DEF_NVEST(I,K)
        ENDDO
    ENDDO



C   LECTURA ELTO A ELTO DE "HUESO"
C   SE BUSCAN DE ENTRE LOS NODOS QUE DEFINEN ESTOS ELEMENTOS
C   IGUALES A LOS DE "NVEST"
C   CDO SE ENCUENTRA SE VA ACTUALIZANDO LA VBLE "HUESO2" CON LOS NODOS
C   DUPLICADOS "NDUPL"
C

    DO I=1,CONT_HUESO
        DO K=1,AUX2
            IF(NVEST(K,1).EQ.HUESO(I,2))THEN
                HUESO2(I,2)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,3))THEN
                HUESO2(I,3)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,4))THEN
                HUESO2(I,4)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,5))THEN
                HUESO2(I,5)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,6))THEN
                HUESO2(I,6)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,7))THEN
                HUESO2(I,7)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,8))THEN
                HUESO2(I,8)=INT(NDUPL(K,1))
            ENDIF
            IF(NVEST(K,1).EQ.HUESO(I,9))THEN
                HUESO2(I,9)=INT(NDUPL(K,1))
            ENDIF
        ENDDO
    ENDDO
```

```
C     ESCRIBE EL VECTOR "COMUNES", CON LOS NODOS QUE PERTENECEN
C     A PIEL Y HUESO
      OPEN (unit=101,file='COMUNES.txt',status='unknown')
      DO I=1,AUX
          WRITE(101,12) COMUNES(I,1)
      ENDDO
      CLOSE(unit=101,status='keep')
12    FORMAT(I8)
C     ESCRIBE EL VECTOR "NVEST", CON LOS NODOS QUE PERTENECEN
C     A PIEL Y HUESO EN EL VESTIBULO
      OPEN (unit=101,file='NVEST.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,12) NVEST(I,1)
      ENDDO
      CLOSE(unit=101,status='keep')

C     ESCRIBE LA DEFINICIÓN DE LOS NODOS COMUNES A PIEL Y HUESO
C     DEL VESTIBULO, QUE SE ENCUENTRAN EN LA VBLE "DEF_NVEST"
      OPEN (unit=101,file='DEF_NVEST.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,11) INT(DEF_NVEST(I,1)),',',DEF_NVEST(I,2),',',
     1    DEF_NVEST(I,3),',',DEF_NVEST(I,4)
      ENDDO
      CLOSE(unit=101,status='keep')
11    FORMAT(I8,A1,E16.8,A1,E16.8,A1,E16.8)

C     ESCRIBE LA DEFINICIÓN DE LOS NODOS DUPLICADOS: "NDUPL"
      OPEN (unit=101,file='NDUPL.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,11) INT(NDUPL(I,1)),',',NDUPL(I,2),',',NDUPL(I,3),
     1    ',',NDUPL(I,4)
      ENDDO
      CLOSE(unit=101,status='keep')

C     ESCRIBE LA NUEVA DEFINICIÓN DE "HUESO" ("HUESO2")
      OPEN (unit=101,file='HUESO2.txt',status='unknown')
      DO I=1,CONT_HUESO
          WRITE(101,13) HUESO2(I,1),',',HUESO2(I,2),',',HUESO2(I,3),
     1    ',',HUESO2(I,4),',',HUESO2(I,5),',',HUESO2(I,6),',',HUESO2(I,7),
     2    ',',HUESO2(I,8),',',HUESO2(I,9)
      ENDDO
      CLOSE(unit=101,status='keep')
13    FORMAT(I8,A1,I8,A1,I8,A1,I8,A1,I8,A1,I8,A1,I8,A1,I8,A1,I8)

C     ESCRIBE EL GRUPO DE NODOS DUPLICADOS: "GNDUPL"
      OPEN (unit=101,file='GNDUPL.txt',status='unknown')
      DO I=1,AUX2
          WRITE(101,12) INT(NDUPL(I,1))
      ENDDO
      CLOSE(unit=101,status='keep')


      END
```

```
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     %                         Rutina 3                             %
C     %               "Elset_elementos_membrana.for"                 %
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C
C     Este programa define, a partir de los elementos de la superficie
C     de la cara, los nuevos elementos S3 del modelo que van a formar
C     la piel (elementos tipo membrana definidos por tres nodos).
C     Los numera sumándole 2000000 al elemento tetraédrico al que
C     pertenece.
C


      PROGRAM LECTURA

      INTEGER*4,PARAMETER::   MAX=2000000
      INTEGER*4 I,EXITO,AUX1,AUX2,AUX3,AUX4,AUX
      INTEGER*4 CONT_ELEMENTOS,ELEMENTOS_S2_MAX(MAX,16),PIEL_MAX(MAX,5),
     1CONT_PIEL,S2[ALLOCATABLE](:,:),PIEL[ALLOCATABLE](:,:)
      REAL*4 EL_S2_NEW[ALLOCATABLE](:,:)


C     LEE EL GRUPO DE ELEMENTOS DE LA MEMBRANA DEL ARCHIVO
C     EL_S2.txt Y LOS GUARDA EN LA VARIABLE "ELEMENTOS_S2_MAX"

      OPEN (unit=101,file='C:\FORTRAN\EL_S2.txt',status='unknown')
      CONT_ELEMENTOS=0
      DO I=1,MAX
          READ(101,*,END=18) (ELEMENTOS_S2_MAX(I,J),J=1,16)
      CONT_ELEMENTOS=CONT_ELEMENTOS+1
      ENDDO
   18 WRITE(*,*) 'Terminada lectura de elementos',CONT_ELEMENTOS,'F_S2'
      CLOSE(unit=101,status='keep')

      ALLOCATE (EL_S2_NEW(CONT_ELEMENTOS,16),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_ELEMENTOS
          EL_S2_NEW(I,1:16)=ELEMENTOS_S2_MAX(I,1:16)
      ENDDO


C     LEE LA DEFINICIÓN DE LOS ELEMENTOS DE LA PIEL

      OPEN (unit=101,file='C:\FORTRAN\PIEL.txt',status='unknown')
      CONT_PIEL=0
      DO I=1,MAX
          READ(101,*,END=19) (PIEL_MAX(I,J),J=1,5)
      CONT_PIEL=CONT_PIEL+1
      ENDDO
   19 WRITE(*,*) 'Terminada lectura de nodos',CONT_PIEL,'FILAS PIEL'
      CLOSE(unit=101,status='keep')

      ALLOCATE (PIEL(CONT_PIEL,5),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_PIEL
          PIEL(I,1:5)=PIEL_MAX(I,1:5)
      ENDDO
```

```
C     DEFINE LOS ELEMENTOS DE LA MEMBRANA

      AUX1=0
      AUX2=0
      AUX3=0
      AUX4=1
      AUX=1

      ALLOCATE (S2(MAX,4),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO AUX1=1,CONT_ELEMENTOS
          DO AUX2=1,16
              AUX3=EL_S2_NEW(AUX1,AUX2)
              IF (AUX3==0) THEN
                  GOTO 21
              ENDIF
20            IF (AUX3==PIEL(AUX,1)) THEN
                  S2(AUX4,1)=2000000+PIEL(AUX,1)
                  S2(AUX4,2)=PIEL(AUX,2)
                  S2(AUX4,3)=PIEL(AUX,3)
                  S2(AUX4,4)=PIEL(AUX,5)
                  AUX4=AUX4+1
              ELSE
                  AUX=AUX+1
                  GOTO 20
              ENDIF
          ENDDO
21    ENDDO

C     ESCRIBE LOS VECTORES CON LA NUEVA NUMERACIÓN DE LOS ELEMENTOS
C     DE LA MEMBRANA DE LA PIEL


      OPEN (unit=101,file='S2.txt',status='unknown')
      DO I=1,(AUX4-1)
      WRITE(101,12) INT(S2(I,1)),',',INT(S2(I,2)),',',
     1    INT(S2(I,3)),',',INT(S2(I,4))
      ENDDO
      CLOSE(unit=101,status='keep')
12    FORMAT(I8,A1,I8,A1,I8,A1,I8,A1,I8)


      END
```

```
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     %                          Rutina 4                                      %
C     %                  "Superficie_vestibulo.for"                            %
C     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C
C     Este programa lee los elementos del grupo HUESO que pertenecen
C     a la selección del vestíbulo hecha y selecciona cuales de ellos
C     tienen tres nodos en su superficie (basándose en que están
C     numerados a partir del 600000). El resultado son los elementos
C     que tienen una cara en el vestíbulo y que cara es.
C



      PROGRAM LECTURA

      INTEGER*4,PARAMETER::   MAX=1000000
      INTEGER*4 I,AUX,AUX1,AUX2,AUX3,AUX4,EXITO
      INTEGER*4 CONT_HUESO,HUESO_MAX(MAX,5),HUESO2[ALLOCATABLE](:,:),
     1S1(100000,1),S2(100000,1),S3(100000,1),S4(100000,1)


C     LEE EL GRUPO DE ELEMENTOS DEL HUESO DEL ARCHIVO
C     HUESO2.txt Y LOS GUARDA EN LA VARIABLE "HUESO2" Y
C     SE COPIA HUESO EN HUESO2.

      OPEN (unit=101,file='C:\FORTRAN\HUESO2.txt',status='unknown')
      CONT_HUESO=0
      DO I=1,MAX
          READ(101,*,END=18) (HUESO_MAX(I,J),J=1,5)
      CONT_HUESO=CONT_HUESO+1
      ENDDO
   18 WRITE(*,*) 'Terminada lectura de nodos',CONT_HUESO,'FILAS HUESO2'
      CLOSE(unit=101,status='keep')

      ALLOCATE (HUESO2(CONT_HUESO,5),STAT=EXITO)
      IF (EXITO.NE.0) THEN
        PRINT*,"ERROR"
      ENDIF

      DO I=1,CONT_HUESO
          HUESO2(I,1:5)=HUESO_MAX(I,1:5)
      ENDDO


C     BUSCAR ENTRE LOS ELEMENTOS DE HUESO2, LOS QUE TENGAN TRES NODOS
C     QUE PERTENEZCAN AL VESTÍBULO (MAYOR QUE 600000)


      AUX1=0
      AUX2=0
      AUX3=0
      AUX4=0


      DO AUX=1,CONT_HUESO

      IF (HUESO2(AUX,2)>600000) THEN
          IF (HUESO2(AUX,3)>600000) THEN
              IF (HUESO2(AUX,4)>600000) THEN
                  AUX1=AUX1+1
                  S1(AUX1,1)=HUESO2(AUX,1)
              ELSE
```

```
                IF (HUESO2(AUX,5)>600000) THEN
                    AUX2=AUX2+1
                    S2(AUX2,1)=HUESO2(AUX,1)
                ELSE
                    GOTO 20
                ENDIF
            ENDIF
        ELSE
            IF (HUESO2(AUX,4)>600000) THEN
                IF (HUESO2(AUX,5)>600000) THEN
                    AUX3=AUX3+1
                    S3(AUX3,1)=HUESO2(AUX,1)
                ELSE
                    GOTO 20
                ENDIF
            ELSE
                GOTO 20
            ENDIF
        ENDIF
    ELSE
        IF (HUESO2(AUX,3)>600000) THEN
            IF (HUESO2(AUX,4)>600000) THEN
                IF (HUESO2(AUX,5)>600000) THEN
                    AUX4=AUX4+1
                    S4(AUX4,1)=HUESO2(AUX,1)
                ELSE
                    GOTO 20
                ENDIF
            ELSE
                GOTO 20
            ENDIF
        ELSE
            GOTO 20
        ENDIF
20  ENDIF

    ENDDO

    WRITE(*,*) 'Terminada búsqueda de elementos',AUX1,'AUX1',
   1AUX2,'AUX2',AUX3,'AUX3',AUX4,'AUX4'


C   ESCRIBE LOS VECTORES CON LOS ELEMENTOS DEL HUESO QUE PERTENECEN
C   AL VESTÍBULO


    OPEN (unit=101,file='S1.txt',status='unknown')
    DO I=1,AUX1
        WRITE (101,12) S1(I,1)
    ENDDO
    CLOSE(unit=101,status='keep')
12  FORMAT(I8)

    OPEN (unit=101,file='S2.txt',status='unknown')
    DO I=1,AUX2
        WRITE (101,12) S2(I,1)
    ENDDO
    CLOSE(unit=101,status='keep')


    OPEN (unit=101,file='S3.txt',status='unknown')
    DO I=1,AUX3
        WRITE (101,12) S3(I,1)
    ENDDO
```

```
CLOSE(unit=101,status='keep')


OPEN (unit=101,file='S4.txt',status='unknown')
DO I=1,AUX4
    WRITE (101,12) S4(I,1)
ENDDO
CLOSE(unit=101,status='keep')

END
```