



Universidad
Zaragoza

Trabajo Fin de Grado

Simulación en tiempo real del comportamiento
mecánico de la vesícula biliar

Autor

Ismael Sanz Martínez

Director

David González Ibáñez

Escuela de Ingeniería y Arquitectura
2014/2015

Resumen

En este proyecto se procederá a la simulación computacional del comportamiento mecánico de la vesícula biliar bajo acciones externas de carga.

Este proyecto se realizará para acoplar los resultados obtenidos con los de otro TFG, que se encargará de la simulación del hígado, para poder generar una simulación en tiempo real de una colecistectomía laparoscópica.

La metodología del proyecto será la siguiente. En un primer apartado se procederá a la caracterización mecánica del órgano, así como las condiciones de contorno de la simulación y las cargas a las que estará sometida la vesícula biliar. Después se procederá a la explicación del método y la aplicación usada para los cálculos (en nuestro caso el método PGD). Una vez explicado esto, se realizarán varios cálculos elásticos lineales para aportar una primera aproximación de los desplazamientos de la vesícula.

A continuación se realiza una simulación de la vesícula biliar en comportamiento hiperelástico, como consideración del tejido de un modo más genérico, con el que obtendremos la solución de desplazamientos para la simulación de la colecistectomía.

Finalmente para concluir este proyecto desarrollaremos el cálculo dinámico de la vesícula y se mostrará el potencial que ofrece la técnica propuesta mostrando los resultados de la simulación en tiempo real en un entorno web.

En los anexos se puede consultar un desarrollo del método y la aplicación utilizada.

INDICE

Resumen

1	Introducción	1
	1.1 Biomecánica computacional.....	1
	1.2 Fisiología de la vesícula biliar.....	4
	1.3 Cirugía laparoscópica: <i>Colecistectomía</i>	6
	1.3.1 Colecistectomía.....	7
	1.4 Evolución histórica de la simulación quirúrgica en tiempo real....	8
	1.4.1 Primera generación de simuladores.....	9
	1.4.2 Segunda generación de simuladores: <i>Perfeccionamiento de la apariencia</i>	10
	1.4.3 Tercera a quinta generación de simuladores: adicción de los detalles fisiológicos	11
	1.5 Modelos de orden reducido (MOR)	12
2	Descomposición propia Generalizada: Método y Aplicación (software)	13
	2.1 Aplicación PGD de Matlab.....	15
3	Ejemplos de aplicación técnicas PGD para obtener soluciones en tiempo real : vesícula Biliar	16
	3.1 Definición del problema computacional.....	16
	3.1.1 Definición de condiciones de contorno del modelo	16
	3.1.2 Definición de cargas del modelo	17
	3.2 Cálculo estático lineal.....	18
	3.2.1 Cálculo con solo presión	19
	3.2.2 Cálculo con un solo punto de carga	21
	3.2.3 Cálculo con varias cargas	23
	3.2.4 Variación del error relativo en función del numero de modos de carga	26
	3.3 Calculo estático no lineal.....	27
4	Cálculo de la vesícula hiperelástica en régimen dinámico.....	31
	4.1 Generación de la solución dinámica en Abaqus para generación de base reducida mediante técnicas POD.....	31
	4.2 Cálculo dinámico con la aplicación PGD de Matlab	32
	4.2.1 Generación del cálculo elástico dinámico de la vesícula.....	32
5	Conclusiones	35

Anexo I: método del punto Fijo.....	36
Anexo II: Aplicación PGD de Matlab	38
Anexo III: Generación de un modelo de base reducida POD en Abaqus.....	43
BIBLIOGRAFÍA	45

INDICE DE FIGURAS

Figura 1. Estudios biomecánicos en el cuerpo humano.	2
Figura 2. Vesícula biliar en su localización dentro del sistema digestivo (Fuente: radiolugo.info)	4
Figura 3. Anatomía vesícula biliar (Fuente: wikipedia.org).....	5
Tabla 1. Propiedades físicas vesícula biliar	5
Figura 4. Instrumental laparoscópico (trocares y cánulas).....	6
Figura 5. Procedimiento de introducción de los instrumentos en el abdomen. .	7
Figura 6. Ejemplo de simulador desarrollado en el grupo AMB de la Universidad de Zaragoza equipado con un brazo háptico.	9
Figura 7. Ejemplo de una viga en voladizo usando las leyes de la elasticidad lineal (figura superior) y las de geometría no lineal (figura inferior). Fuente: [4]	10
Figura 8. Vesícula biliar interaccionando con el resto de órganos.	16
Figura 9. Condición de apoyo de la vesícula biliar	17
Figura 10. Contacto entre hígado y vesícula.....	17
Figura 11. Ejemplo de algunas cargas definidas para el cálculo estático	18
Figura 12. Simulación solo presión con modelo PGD (resultado y modelo sin deformar)	20
Figura 13. Simulación solo presión con Abaqus	21
Figura 14. Pesos sacados por la aplicación para una carga de 0.1N en situación estática lineal	22
Figura 15. Desplazamientos en vesícula biliar bajo condiciones de carga de 0.1N en situación estática lineal.	22
Figura 16. Modelo Abaqus con una sola carga en situación elástica lineal.....	23
Figura 17. Modos con peso de simulación estática lineal con cargas definitivas.	24
Figura 18. Particularización de la solución para carga en nodo 23 (solución final y sin deformar).	24
Figura 19. Situación de varias cargas para modelo Abaqus.	25
Figura 20. Variación del error relativo en función de los modos PGD de carga. ...	26
Figura 21. Ejemplo de problema no lineal	27
Figura 22. Discretización del problema no lineal en suma de problemas lineales	28

Figura 23. Solución final del problema estático no lineal (solución final y sin deformar).....	29
Figura 24. Modo 1 del cálculo estático no lineal (solución final y sin deformar)... ..	29
Figura 25. Modo 2 del cálculo elástico no lineal (solución final y sin deformar). ..	30
Figura 26. Postproceso de la vesícula imagen 1	32
Figura 27. Postproceso de la vesícula biliar imagen 2	33
Figura 28. Directorio de la aplicación PGD con todas sus carpetas.....	38
Figura 29. Ejemplo ficheros dentro de la carpeta Bladder.....	39
Figura 30. Ejemplo DefaultBladder2.....	40
Figura 31. Reports del XYData para desplazamientos.....	43
Figura 32. Ficheros .rpt y .inp obtenidos del cálculo dinámico con Abaqus. ...	44

1 Introducción

1.1 Biomecánica computacional

La Biomecánica es un conjunto de conocimientos derivados de la física que tienen como objetivo estudiar los efectos de las fuerzas mecánicas sobre los sistemas orgánicos de los seres vivos y sus estructuras. Así se podrá predecir las posibles alteraciones que sufran los mismos y proponer así soluciones de manera artificial a las mismas.

El objetivo de la biomecánica es resolver los problemas que surgen de las diversas condiciones a las que puede verse sometido nuestro cuerpo en distintas situaciones. En la Figura 1 podemos observar una de las aplicaciones de la biomecánica al estudio del cuerpo humano.

En la actualidad, la biomecánica se halla presente en tres ámbitos fundamentales de actuación:

- ***Biomecánica médica:*** encargada de apreciar las patologías que aquejan al cuerpo humano para forjar soluciones capaces de valorarlas, repararlas o disminuirlas.
- ***Biomecánica deportiva:*** establece un análisis de la actividad deportiva, con el objeto de aumentar: su rendimiento, sus resultados y desarrollar nuevas formas de entrenamiento e instrumentos que mejoren las prestaciones de los anteriores.
- ***Biomecánica ocupacional:*** cuya misión es estudiar la interacción del cuerpo humano con nuestro entorno más inmediato y adaptarlo a nuestras necesidades y capacidades. En este terreno, la Biomecánica está muy relacionada con otra disciplina, conocida como ergonomía.



Figura 1. Estudios biomecánicos en el cuerpo humano.

En este proyecto se enfoca dentro de la rama de la biomecánica conocida como biomecánica computacional y por tanto tendremos que tener en cuenta como se ha desarrollado el estudio de los aspectos del cuerpo humano tales como los tejidos o la circulación sanguínea.

La **Biomecánica computacional** se refiere a las simulaciones computerizadas de sistemas biomecánicos, tanto para poner a prueba modelos teóricos y refinarlos, como para las aplicaciones técnicas. Usualmente se usan tanto modelos de sólidos para simular comportamientos cinemáticos, como modelos de elementos finitos para simular propiedades de deformación y resistencia de los tejidos y elementos biológicos. La biomecánica computacional ha sido aplicada principalmente a los siguientes aspectos dentro de la anatomía humana:

- **Circulación sanguínea:** Fue uno de los primeros enfoques en los que se aplicó la biomecánica a la medicina. Se intentó aplicar las ecuaciones de Navier-Stokes (ecuaciones en derivadas parciales no lineales que describen el movimiento de un fluido) a la compresión del riego sanguíneo (que aunque se considera que sea un fluido newtoniano incomprensible, en las zonas de capilares o arteriolas no es exactamente así).

- **Huesos:** Son estructuras anisótropas (diferentes propiedades en direcciones longitudinales y transversales). Se aplicaron las leyes de Hooke para el desarrollo de las relaciones tensión-deformación.
- **Tejido muscular:** Donde aparecen tres tipos de tejidos: **liso** (se mueve involuntariamente, por ejemplo el estomago), **miocárdico** (también conocidos como cardiomiocitos, están situados en la pared del corazón se contraen involuntariamente y son los responsables de los latidos del corazón) y **esquelético** (desarrolla esfuerzo sostenido y generalmente voluntario).
- **Tejidos Blandos:** Cuyos cálculos constitutivos fueron desarrollados desde la década de los 70, se buscaba dar una caracterización del comportamiento de este tipo de tejidos. Los primeros trabajos se centraron en los tendones, ligamentos y el cartílago. Todos ellos están formados por fibras de colágeno (diferente cantidad y calidad según su función). La función de cada uno será la siguiente:
 - **Tendones:** sujetan el musculo al hueso y están sometidos a cargas a tracción. Deben ser fuertes y flexibles para el buen funcionamiento del musculo.
 - **Ligamentos:** conectan huesos entre sí; son más rígidos que los tendones.
 - **Cartílago:** solicitado principalmente a compresión y actúa como almohadilla en las articulaciones.

Actualmente se han desarrollado modelos para la piel y para órganos internos (como es el caso desarrollado en este TFG).

1.2 Fisiología de la vesícula biliar

La vesícula biliar es una víscera hueca en forma de ovoide de un tamaño relativamente pequeño, aproximadamente unos 5 cm de diámetro mayor. Esta se encuentra alojada por debajo de del hígado, al cual se encuentra pegado por una fina membrana de gran elasticidad (como se puede observar en la Figura 2). La vesícula se conecta al intestino delgado en la zona del duodeno a través de un conducto de tamaño nimio llamado vía biliar o colédoco.

La vesícula biliar es una fuente muy frecuente de complicaciones médicas, ya que es aquí donde se producen los cálculos biliares que llevan a la vesícula a inflamarse produciendo un dolor agudo, y cuya aparición es una de las posibles fuentes de la aparición de una pancreatitis.

Los problemas más típicos que se pueden producir en ella son: la inflamación de la vesícula biliar (también llamada colecistitis) y la presencia de cálculos biliares en dichas vías (también llamado colelitiasis).

La función principal de la vesícula es la acumulación de la bilis que libera al duodeno a través de los conductos antes citados, y entra en el duodeno a través de la papila o ampolla de Váter.

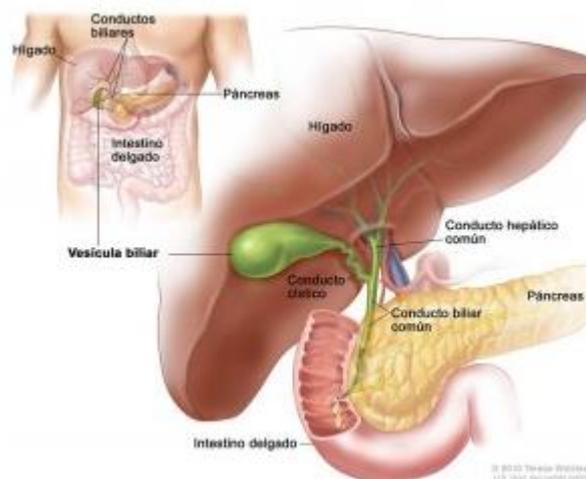


Figura 2. Vesícula biliar en su localización dentro del sistema digestivo (Fuente: radiolugo.info)

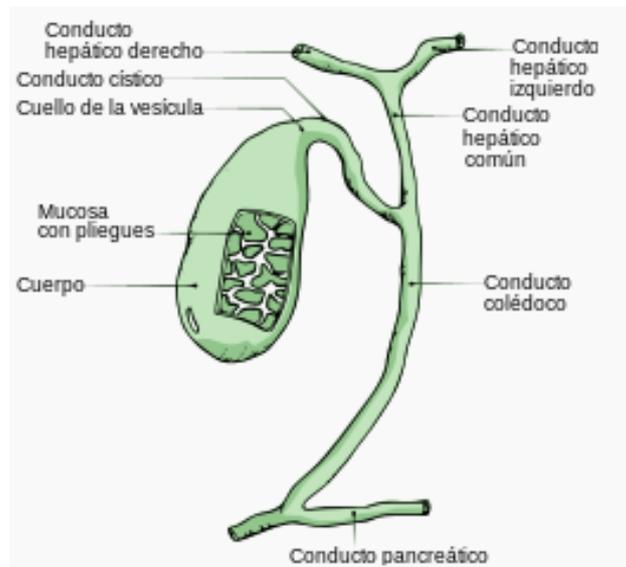


Figura 3. Anatomía vesícula biliar (Fuente: wikipedia.org)

Para el correcto desarrollo de este TFG, es necesario determinar las propiedades físicas de la vesícula biliar, que serán necesarias para la simulación computacional que se va a llevar a cabo. Estas características serán: presión (en el caso de la vesícula habrá dos tipos de presiones diferentes, una cuando esté llena completamente que será la presión máxima y otra cuando este vacía que será la mínima), modulo de Young, coeficiente de Poisson y densidad del órgano. Estas se expresan en la siguiente tabla:

Propiedades necesarias de la vesícula biliar	
Densidad de la vesícula biliar	1200Kg/m ³
Coeficiente de Poisson	0,48
Módulo de Young	170000Pa = 170000N/m ²
Presión interna	Dos presiones a diferenciar <ul style="list-style-type: none"> • Máxima: 1961,33 Pa • Mínima : 980,665 Pa

Tabla 1. Propiedades físicas vesícula biliar

Cabe destacar que para la simulación quirúrgica vacían la vesícula biliar, por lo que no aporta apenas información la suposición de esta presión interna en el caso de querer simular la cirugía laparoscópica.

1.3 Cirugía laparoscópica: Colecistectomía

La laparoscopia es una técnica que permite la visualización de la cavidad pélvica-abdominal con ayuda de una lente óptica. Por medio de una fibra óptica, se introduce la luz para iluminar a la cavidad a la vez que obtienen imágenes del interior por medio de una cámara colocada en la misma lente.

Este mismo método se utiliza en intervenciones quirúrgicas por lo que se considera, hoy en día un método de cirugía mínimamente invasiva para corregir o curar enfermedades.

El instrumental que se requiere para realizar las operaciones será: telescopios, fuentes de luz, imagen y vídeo (cámaras, monitor, procesador de video, videograbadora), insufladores, trocares, cánulas de aspiración/irrigación, equipo para electrocirugía, instrumental quirúrgico (sujeción, disección, retracción, tijeras, porta agujas, aplicadores de grapas, engrapadoras, otros).

El método consiste en que a través de incisiones se introducen trocares y cánulas; la punta del trocar se retira y queda la cánula en la incisión de tal manera que permite una vía de acceso del instrumental quirúrgico a utilizar hasta el interior del abdomen.



Figura 4. Instrumental laparoscópico (trocares y cánulas)

Así pues el cirujano tendrá que operar mientras observa una pantalla de unos 30 cm (en dos dimensiones); con la habilidad suficiente pues el entorno real de trabajo es obviamente tridimensional.

1.3.1 Colecistectomía

Se conoce como colecistectomía laparoscópica a la extracción de la vesícula que se encuentra adherida al hígado por una fina membrana a través de incisiones mínimas a través de las cuales se introducirá el instrumental y la cámara.

El proceso a seguir será el siguiente:

- Se introduce un delgado trocar a través del abdomen por el ombligo, a través del cual se introducirá una pequeña cámara (laparoscopio) con la que el cirujano podrá ver una visión ampliada de los órganos en una pantalla de televisión.
- A través de pequeñas incisiones o cortes se introducirán el resto de instrumental necesario para la operación; después se procederá a separar la vesícula y a extraerla a través de una de las aberturas.
- Una vez sacada se suturan las incisiones y se cosen las abdominales.

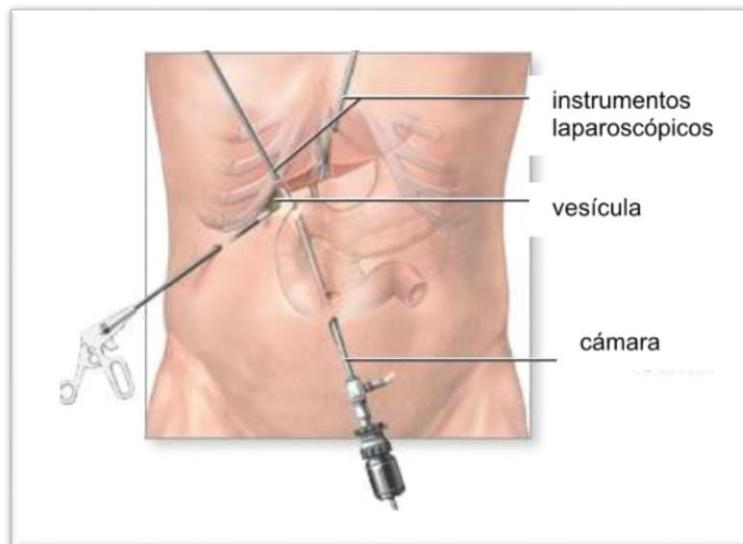


Figura 5. Procedimiento de introducción de los instrumentos en el abdomen.

El desarrollo de esta nueva forma de operar de manera tan poco invasiva ha sido toda una revolución debido a que se han reducido los tiempos de rehabilitación desde varias semanas a días.

Este tipo de operaciones requieren un alto grado de especialización por parte del cirujano, por lo que el entrenamiento y la preparación de estos son

factores muy importantes. El objetivo de este TFG es el de añadir el comportamiento de la vesícula biliar al desarrollo que se está llevando a cabo (dentro del grupo AMB de la Universidad de Zaragoza) de un simulador quirúrgico, que pueda ofrecer el suficiente realismo, para que la formación de los futuros cirujanos sea correcta. Es necesario (con vistas a una mejor aplicación en el futuro) seguir desarrollando los métodos de interacción en los órganos para alcanzar niveles de preparación óptimos.

1.4 Evolución histórica de la simulación quirúrgica en tiempo real

En los últimos años el incremento de técnicas mínimamente invasivas en los campos de la Gastroenterología y de la Cirugía, así como los efectos de las curvas de aprendizaje sobre la aparición de complicaciones, han suscitado un debate acerca de los estándares de formación.

A partir del aporte de la publicación de los doctores Elías Cueto y Francisco Chinesta en [4], se desarrolla a continuación y de manera breve la evolución histórica que ha sufrido la simulación computacional de procedimientos médicos y simulación de órganos.

R. Satava (en su estudio *Medical Virtual Reality: the current status of the future.*) propuso hace unos quince años una taxonomía de anatomía virtual que se divide en cinco generaciones diferentes.

La primera generación estuvo conformada por sistemas que representan con exactitud la geometría de los órganos a nivel macroscópico. La segunda generación sería incluir una descripción de la dinámica física del cuerpo. Si bien es todavía difícil más de una década después, encontrar un simulador quirúrgico en tiempo real con esa precisión para los tejidos blandos actuales.

Desde la tercera a la quinta, las descripciones de estos pacientes virtuales deben incluir, respectivamente, descripciones precisas de la fisiología, anatomía microscópica (en un nivel neurovascular, por ejemplo) y, por último, sistemas bioquímicos.



Figura 6. Ejemplo de simulador desarrollado en el grupo AMB de la Universidad de Zaragoza equipado con un brazo háptico.

1.4.1 Primera generación de simuladores

Los orígenes de los llamados “modelos deformables” en el mundo de la informática datan de los años ochenta. Estos simuladores tenían que lidiar con el problema de deformación de un sólido siguiendo las leyes de mecánica de medios continuos.

Sin embargo, las restricciones impuestas por los requisitos de tiempo no permitieron, en ese momento, llevar a cabo estas deformaciones en tiempo real.

En estos momentos, la mayoría de los enfoques se basaban en la elasticidad lineal. Particularmente notable fue el intento de Monserrat C., et al. en [7] de resolver la elasticidad lineal empleando métodos de elementos de contorno, evitando así a la malla en el interior de los dominios.

En general, esta primera generación simuladores quirúrgicos carece hoy de interés, ya que algunos prototipos en la actualidad incorporan, por lo menos, medidas de deformación no lineal en la simulación.

1.4.2 Segunda generación de simuladores: Perfeccionamiento de la apariencia

Durante la evolución de la primera generación de simuladores se comprobó que estos no podían proporcionar una sensación realista debido obviamente, a los pobres modelos constitutivos que las restricciones en tiempo real permitían resolver hasta entonces.

Además al resolver sistemas lineales de grandes deformaciones se observaban grandes ganancias de volumen que se produce cuando es empleado para resolver grandes deformaciones.

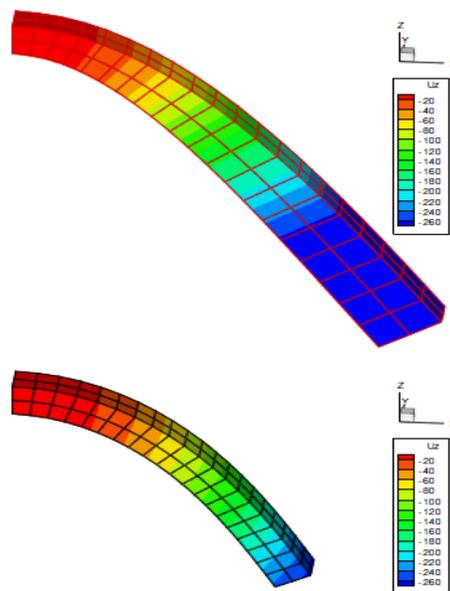


Figura 7. Ejemplo de una viga en voladizo usando las leyes de la elasticidad lineal (figura superior) y las de geometría no lineal (figura inferior). Fuente: [4]

Por tanto se empezaron a considerar los modelos de Saint Venant-Kirchhoff (SVK) como un medio para introducir por lo menos una buena descripción visual de grandes deformaciones. También se utilizaron métodos de elementos finitos con este mismo propósito.

Un enfoque aún más sorprendente se puede encontrar en la obra de Delingette, “*Biquadratic and quadratic springs for modeling st venant kirchhoff materials*”. En ella, un modelo no-lineal (cuadrático) de muelle-masa se desarrolla de modo que sea capaz de reproducir modelos SVK. La formulación resultante, además, elimina las típicas inestabilidades en la compresión de este modelo constitutivo.

Sin embargo, se sabe que los tejidos blandos a menudo están compuestos de fibras de colágeno e incluyen considerable cantidad de agua, y por lo tanto puede ser modelado como cuasi-incompresible. Esta composición compleja no puede ser resuelta solo con modelos geoméricamente no lineales, y por tanto se deben buscar leyes constitutivas para una mejor aproximación del comportamiento de estos tejidos (como por ejemplo, el modelo propuesto por *Holzapfel* y las funciones de energía de tensión-neo *Hookean*).

En general, estos métodos no funcionan bien para grandes tiempos de integración, como es bien conocido por elementos finitos explícitos clásicos. Además este tipo de simulaciones, para resolverse en tiempo real no pueden superar más de 20.000 elementos (según los modelos propuestos por *Holzapfel*).

1.4.3 Tercera a quinta generación de simuladores: adicción de los detalles fisiológicos

Entender y simular cuantitativamente la fisiología del ser humano viene siendo un reto en los últimos años. Aunque la introducción de la fisiología no es estrictamente necesaria para muchas aplicaciones (como los procedimientos de cirugía mínimamente invasiva, por ejemplo), la simulación basada en la planificación médica o quirúrgica lo necesita. En otros problemas es simplemente inevitable (como por ejemplo la predicción del resultado del injerto de vena, que combina la necesidad de la simulación a nivel macroscópico con la simulación a nivel de la red de regulación de los genes).

Los algoritmos Montecarlo (como el llamado algoritmo de simulación estocástica) son a menudo el enfoque preferido para resolución de estos problemas.

En términos generales, añadir detalles fisiológicos a las simulaciones en tiempo real para planificar la cirugía está todavía fuera de alcance en muchos casos. Se necesitan aun esfuerzos adicionales de investigación para incorporar con éxito muchos aspectos tales como sangrado, detalles microscópicos tales como la anatomía glandular o, por último, los aspectos bioquímicos.

1.5 Modelos de orden reducido (MOR)

A la vista de lo anterior, parece que la única posibilidad hasta la fecha para considerar leyes constitutivas diferentes a SVK, es poner en práctica los modelos de Elementos Finitos, en el que la ley constitutiva elegida se evalúa de forma explícita.

La principal limitación de este enfoque se deriva del hecho de que la dinámica del órgano se evalúa explícitamente durante grandes intervalos de tiempo (del orden de horas de tiempo de los usuarios), por lo tanto carecen de estabilidad en conservación de energía, por ejemplo.

Una atractiva posibilidad para resolver este problema es el uso de las técnicas de reducción de modelos. La clave de ello sería el uso de técnicas cuyo mayor tiempo de cálculo se realizará offline (fuera de la interacción en el simulador) de tal manera que en la fase online del simulador los cálculos se realicen de manera instantánea. En este TFG se hace uso de algunos métodos de reducción de modelos bastante conocidos, como son:

- METODO POD: (descomposición Propia Ortogonal), se basa en la descomposición de los modelos complejos en sus valores singulares de tal manera que simplifiquemos los cálculos y así poder obtener soluciones con buenas aproximaciones de los problemas. Este es un método a-posteriori.
- METODO PGD: (Método de Descomposición Propia Generalizada), método a-priori que generaliza el anterior y que será herramienta de simulación de este TFG.

2 Descomposición propia Generalizada: Método y Aplicación (software)

El método de la Descomposición Propia Generalizada está basado en la reducción de modelos *a priori* basada en el uso de representaciones separadas, como se desarrolla en [3], entre otros. La característica principal de este método es que la solución exacta del problema se aproxima imponiendo separación de variables. De esta manera, si $u(\mathbf{x})$, $\mathbf{x} \in R_d$, es la solución del problema, ésta se aproxima en la PGD como:

$$u(\mathbf{x}) \approx \sum_{i=1}^N F_1^i(x_1) \cdot F_2^i(x_2) \cdot \dots \cdot F_d^i(x_d), \quad (2.1)$$

Donde d es el número de dimensiones del problema a resolver. A diferencia de otras metodologías (como POD, Descomposición propia Ortogonal), las funciones $F_{sn}(X_n)$ son funciones desconocidas a priori.

Para determinar estas soluciones el método usa un algoritmo voraz (Greedy algorithm), de tal manera, que para la determinación de cada sumando se resuelve un problema no lineal. Cada valor de estos sumandos estará compuesto de varias funciones que suelen aproximarse por elementos finitos, cuyos valores nodales deben determinarse.

Para el cálculo de estos elementos se puede usar cualquiera de los métodos de resolución existentes (Newton, cuasi-Newton, etc.) pero se prefiere por su sencillez conceptual y de programación, el método de punto fijo. En el cual por cada paso del algoritmo se va obteniendo una de las funciones separadas, de tal manera que se va reduciendo el residuo del problema.

Debido al carácter multidimensional que ofrece la técnica, en este trabajo se procederá al uso del método PGD para la simulación del comportamiento mecánico de la vesícula biliar ante cualquier estado de carga, teniendo en cuenta la posición de la carga, como una extra coordenada del problema.

Enriquecimiento de la solución

Como ya se comenta en [1], y a modo de resumen, dada una aproximación a la solución de la forma que se observa en la ecuación 2.1 (se considera un problema bidimensional por simpleza); es necesario enriquecer la solución mediante la adicción de dos nuevos sumandos $R(x)$ y $S(y)$ que se irán calculando mediante el método de punto fijo.

Así por ejemplo, tomando la ecuación de Poisson:

$$\Delta\varphi_{(x,y)}=f_{(x,y)} \quad (2.2)$$

En un dominio bidimensional cuyas condiciones de contorno se establecerán como Dirichlet ($\Omega= \Omega_x \times \Omega_y=(0, L) \times (0, H)$) la forma débil del problema diferencial:

$$\int_{\Omega_x \times \Omega_y} u^* \cdot \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - f \right) dx \cdot dy = 0. \quad (2.3)$$

Como se ha comentado anteriormente, la solución propuesta por la técnica se puede representar de forma separada como:

$$u(x, y) = \sum_{i=1}^N X_i(x) \cdot Y_i(y). \quad (2.4)$$

De tal manera, si suponemos que son conocidos $n-1$ términos de la solución, estos serán de la forma:

$$u^{n-1}(x, y) = \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y). \quad (2.5)$$

Por lo tanto para calcular el siguiente término se denotará como:

$$u^n(x, y) = u^{n-1}(x, y) + R(x) \cdot S(y) = \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y) + R(x) \cdot S(y). \quad (2.6)$$

Esta expresión se introduce en la forma débil y se obtiene el valor de las funciones $R(x)$ y $S(y)$ que son desconocidas en el paso n del enriquecimiento. Estas funciones las podremos obtener a través del método del punto fijo (que se desarrolla en el Anexo I).

Este método permite calcular $R(x)$ de la interacción actual p a partir de la $S(y)$ de la interacción anterior $p-1$, para a continuación, obtener la $S(y)$ de la interacción p a través de $R(x)$. Las interacciones de este proceso se sucederán hasta que converjan:

$$\frac{\|R_p(x) \cdot S_p(y) - R_{p-1}(x) \cdot S_{p-1}(y)\|}{R_{p-1}(x) \cdot S_{p-1}(y)} < \epsilon,$$

Donde $|\cdot|$ es una norma adecuada.

2.1 Aplicación PGD de Matlab.

La técnica aplicada anteriormente ha sido implementada en Matlab y es la que ha sido utilizada para el desarrollo de este TFG.

Esta aplicación de Matlab usa el método PGD (Descomposición Propia Generalizada) para la resolución de problemas elástico lineales y no lineales bajo las condiciones de la estática o la dinámica (la mayor diferencia entre ambas radica en que en el uso de las condiciones dinámicas se tienen en cuenta a parte de las coordenadas de desplazamiento en el instante inicial, las coordenadas de desplazamiento en el instante inmediatamente anterior; así se desarrolla una solución más completa del problema a tratar). El usuario de la aplicación podrá decidir en cada caso el tipo de problema que quiere realizar y ajustar las condiciones del mismo.

Esta aplicación se ha desarrollado dentro del grupo AMB (*Applied Mechanics and Bioengineering*) de la Universidad de Zaragoza por el profesor titular David González.

3 Ejemplos de aplicación técnicas PGD para obtener soluciones en tiempo real : vesícula Biliar

3.1 Definición del problema computacional

Antes de comenzar con cualquier cálculo es necesario generar tanto las condiciones de carga, como las condiciones de contorno que definirán el problema.

3.1.1 Definición de condiciones de contorno del modelo

Para establecer correctamente el problema es necesario determinar las condiciones de contorno correctamente. En este caso se tienen en cuenta las correspondientes a una operación laparoscópica en el que la vesícula interaccionara con el resto de órganos como se puede apreciar en la imagen 8(Para el caso de la simulación de una colecistectomía, interaccionará con el hígado).



Figura 8. Vesícula biliar interaccionando con el resto de órganos.

Para definir las condiciones de apoyo hay que tener en cuenta que la vesícula se encuentra pegada en su cuello, al extremo del conducto cístico, y también esta adherida al hígado por medio de una membrana (esta última no la tendremos en cuenta debido a que consideraremos esta unión como parte de la transmisión de las cargas generadas por el peso del hígado).

Para esta situación se supondrá que una restricción de empotramiento (“*Encastre*”) en los nudos de la parte superior de la vesícula. De esta manera el apoyo queda así:

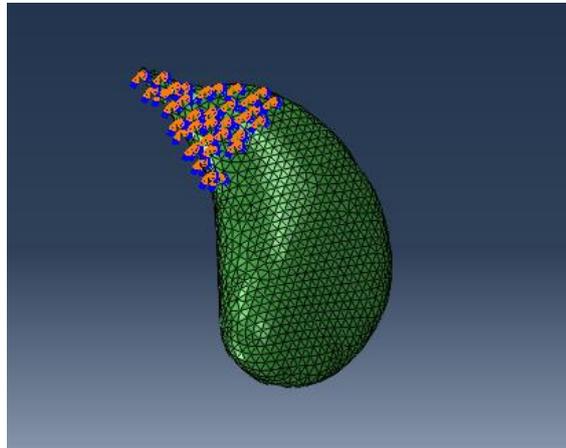


Figura 9. Condición de apoyo de la vesícula biliar

3.1.2 Definición de cargas del modelo

Una vez definidas las condiciones de contorno, establecemos las cargas que influirán en la vesícula. Como ya se ha comentado antes estas cargas serán los resultados de la interacción con el hígado suprimiendo la membrana intermedia.

En la Figura 10 se puede observar la zona de contacto entre ambos órganos, que corresponde a la situación de carga que se va a simular.

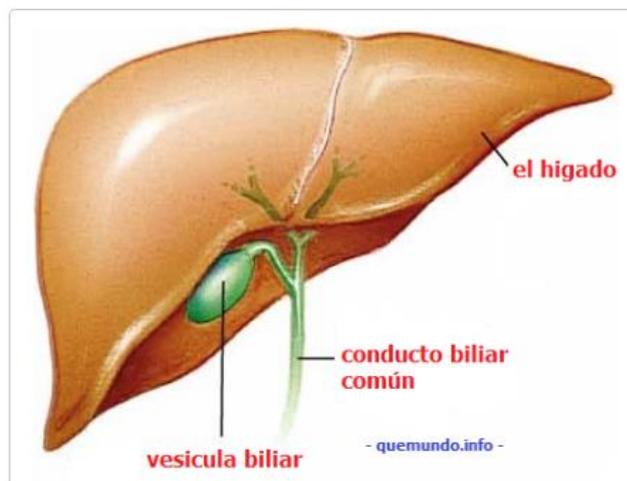


Figura 10. Contacto entre hígado y vesícula

Así pues extrapolando estas condiciones para la simulación quedará la siguiente distribución de cargas de la Figura 11. Cada una de las cargas propuesta representa un estado posible de cargas a la hora de la simulación en tiempo real. La solución que proporciona la técnica PGD permite establecer a priori cualquier estado de cargas que se pueda dar en la fase on-line, determinando en la fase off-line una solución multiparamétrica que dependerá de la geometría y de cualquier posición de carga, que en este caso abarcará la zona de contacto entre el hígado y la vesícula.

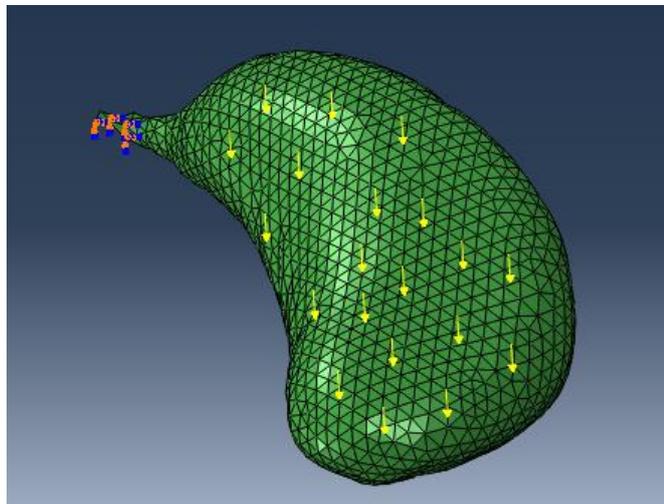


Figura 11. Ejemplo de algunas cargas definidas para el cálculo estático

3.2 Cálculo estático lineal

A continuación se presentan 3 ejemplos de aplicación: simulación de la vesícula sometida a presión interna exclusivamente, simulación de la vesícula sometida a una carga puntual y simulación de la vesícula ante un estado de carga cualquiera. Con los dos primeros estudios, comprobaremos que la técnica ofrece resultados satisfactorios comparándolos con programas comerciales que servirán de referencia. Con el último de los cálculos se presenta la gran ventaja que ofrece esta técnica, pues en un solo cálculo (una única solución) nos permite obtener el resultado de la vesícula ante cualquier estado de cargas particularizando la solución obtenida, en un tiempo de respuesta háptico (1Mhz-mil veces por segundo).

3.2.1 Cálculo con solo presión

Como ya se he comentado antes este cálculo se realizará tanto en Abaqus como por la aplicación PGD para comparar los resultados y comprobar la bonanza de los mismos.

En Abaqus se puede introducir la presión directamente (dado que este permite la aplicación de una presión en todo el contorno del órgano); teniendo en cuenta las diferentes bibliografías que se han consultado, el valor considerado es de 1961.33 Pa.

En el caso de la aplicación PGD es más complicado. Al ser un órgano de dimensiones tan pequeñas, el tamaño de las áreas de los elementos es menor, y por tanto hay que encontrar una presión que se ajuste a él de tal manera que provoque el mismo efecto que la presión total que se aplica en Abaqus. Así pues se procedió al cálculo de la superficie efectiva que se obtenía para cada nodo interior de la vesícula.

Este cálculo se realizó a partir de la suposición de que la superficie interna del órgano estaba formada por elementos triangulares (debido a que esta es la forma que tienen las caras exteriores de un tetraedro).

Para cada elemento de la zona interior de la vesícula, se extraen 4 puntos (uno de ellos se tomara como origen) y se desarrollan las coordenadas de los vectores que forman los tres puntos con el que hemos cogido como origen. Una vez hecho se aplica la siguiente fórmula para hallar el volumen equivalente:

$$V = \frac{1}{6} \begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ z_A & z_B & z_C \end{vmatrix} \quad (3.1)$$

Y se obtiene un volumen de $6.13 \cdot 10^{-10} \text{ m}^3$, por tanto y sabiendo que:

$$V = \frac{\sqrt{2}}{12} \cdot a^3 \approx 0,1179 \cdot a^3 \quad (3.2)$$

Se obtiene una arista equivalente $a = 0.00173 \text{ m}$ y con la fórmula del cálculo de área de una de las caras del tetraedro por su arista que es:

$$A = 4 \cdot A_c = 4 \cdot \frac{\sqrt{3}}{4} \cdot a^2 = \sqrt{3} \cdot a^2 \approx 1,732 \cdot a^2 \quad (3.3)$$

Se obtiene que el área efectiva del triangulo, que forma una cara del tetraedro, es de $5.2 * 10^{-6}m^2$ y multiplicando este valor por la presión de $1961.33 Pa = 1961.33 N/m^2$ se obtiene que la fuerza que hay que aplicar por cada nodo interior es de $0.0102 N$.

Al realizar el cálculo con la aplicación PGD se obtiene un único modo, como es de esperar debido a que solo hay un estado de carga. El resultado que entrega la aplicación es el siguiente:

```
Elastic
Static
Bladder2
Computing PGD solution...
Mode 1, ITERS 2, Weight 0.210589
Mode 2, ITERS 22, Weight 0.000000
```

Los resultados de ambos métodos comparados en los que se observa la geometría y su distribución de desplazamientos se pueden ver en las figuras 12 y 13.

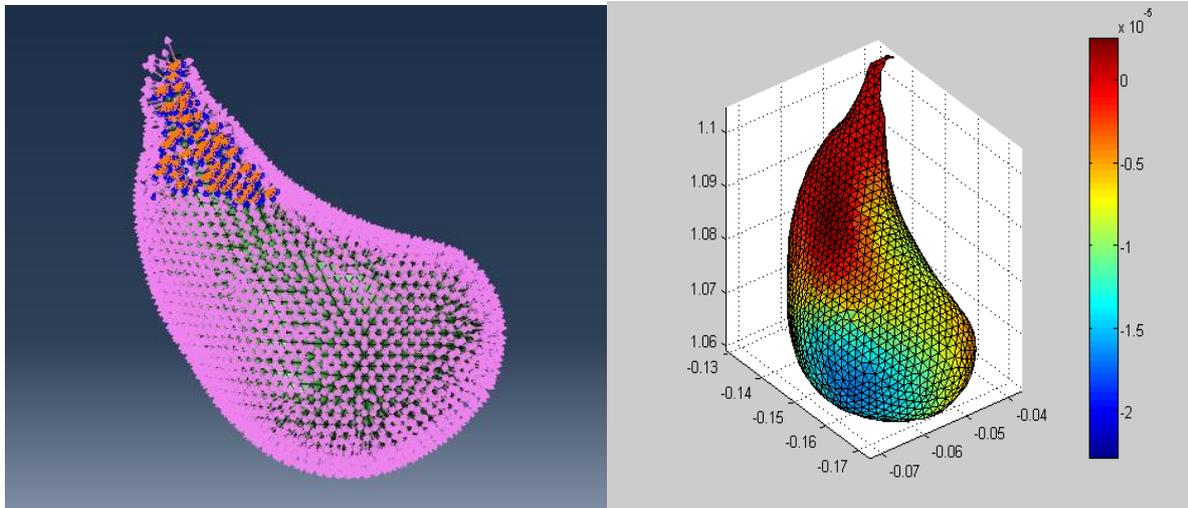


Figura 12. Simulación solo presión con modelo PGD (resultado y modelo sin deformar)

En este caso a través de la aplicación de matlab obtenemos que el desplazamiento máximo se da en el nodo inferior mas alejado de la zona de condiciones de contorno.

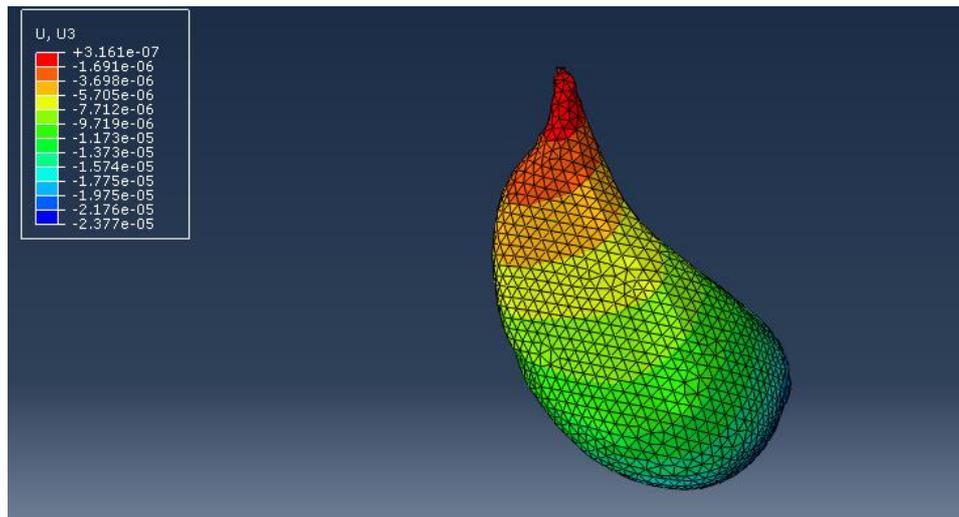


Figura 13. Simulación solo presión con Abaqus

Se observa como ambos rangos de valores son muy similares y no se aprecian diferencias significativas en ambos. También cabe remarcar la poca influencia de la presión en los cálculos de desplazamientos, del orden de $10^{-5} m$.

Si aplicamos pues una formula sencilla, para calcular el error relativo entre ambos soluciones:

$$\text{Error (\%)} = \frac{|U_{max\text{Matlab}} - U_{max\text{Abaqus}}|}{U_{max\text{Abaqus}}} \times 100 \quad (3.4)$$

Obtenemos un error cercano al 1,304348% (cogiendo en ambos casos el desplazamiento máximo alcanzado por ambos métodos).

Durante la operación la vesícula suele ser vaciada para evitar complicaciones por lo que la presión interna no debería considerarse para la simulación de la colecistectomía.

3.2.2 Cálculo con un solo punto de carga

Una vez establecida la relación entre presiones el siguiente paso es obtener una comparación entre los resultados en ambos software, en un cálculo con un solo punto cargado con 0.1N.

Este tipo de cálculo será el más rápido que podamos hacer debido a que su tiempo de computación es muy reducido, y que los valores que se obtienen nos dotan de una primera pequeña aproximación de los valores reales que se podrían producir en una situación de interacción real.

En cuanto a las deformaciones como se puede observar en la Figura 15 han sufrido un aumento considerable con respecto a los cálculos ejecutados solo con presión en los que el órgano apenas se deformaba y ahora da lugar a un desplazamiento de 6 mm. Hay que tener en cuenta que con un Módulo de Young de 170000 MPa estamos hablando de un material con poca rigidez.

De nuevo basta con un solo sumando (o modo PGD) para obtener la solución final.

```
Elastic
Static
Bladder2
Computing PGD solution...
Mode 1, ITERS 2, Weight 0.435757
Mode 2, ITERS 19, Weight 0.000000
```

Figura 14. Pesos sacados por la aplicación para una carga de 0.1N en situación estática lineal

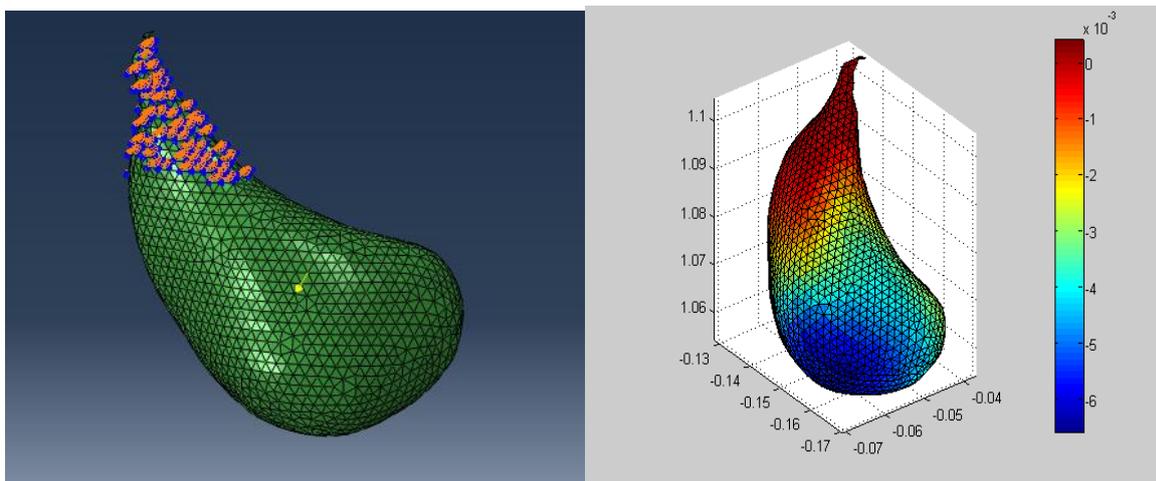


Figura 15. Desplazamientos en vesícula biliar bajo condiciones de carga de 0.1N en situación estática lineal.

Como ya se ha hecho antes ejecutaremos un cálculo en Abaqus con las mismas condiciones de contorno y carga que las usadas en el cálculo anterior. Los resultados pueden observarse en la imagen 16.

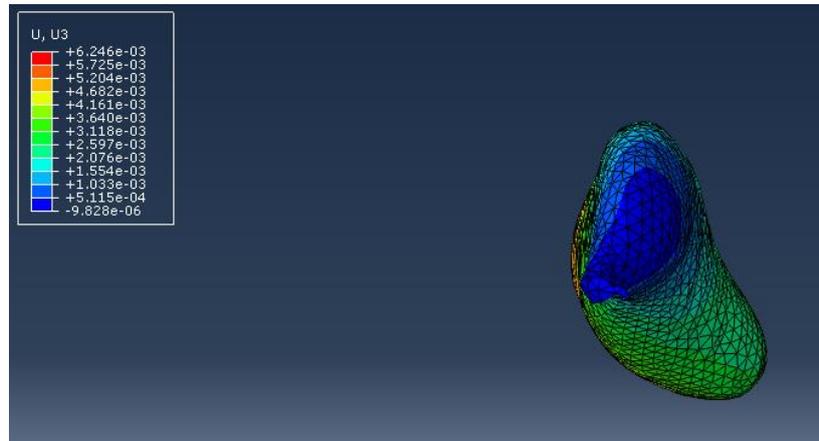


Figura 16. Modelo Abaqus con una sola carga en situación elástica lineal.

Una vez ejecutado como hemos hecho antes calculamos el error relativo entre los desplazamientos entre ambos métodos a través de la fórmula 3.4.

Y obtenemos un error del 2.33 %. Se observa un aumento del error respecto al anterior pero sigue siendo una buena aproximación para el entorno de la simulación biomecánica.

3.2.3 Cálculo con varias cargas

Para terminar con el cálculo elástico lineal supondremos una situación de varias cargas (como se definieron en el apartado 3.1.1) solución que solo permite la técnica PGD propuesta.

Así pues ejecutamos el cálculo y obtenemos la siguiente solución con sus pesos de modos característicos y cuántos de ellos necesita para converger:

```
Elastic
Static
Bladder2
Computing PGD solution...
Mode 1, ITERS 3, Weight 2.656668
Mode 2, ITERS 9, Weight 0.366536
Mode 3, ITERS 6, Weight 0.183832
Mode 4, ITERS 19, Weight 0.011012
Mode 5, ITERS 22, Weight 0.007124
Mode 6, ITERS 22, Weight 0.004228
Mode 7, ITERS 22, Weight 0.006535
Mode 8, ITERS 22, Weight 0.004445
Mode 9, ITERS 22, Weight 0.002099
Mode 10, ITERS 22, Weight 0.004761
Mode 11, ITERS 22, Weight 0.001724
Mode 12, ITERS 22, Weight 0.001837
Mode 13, ITERS 22, Weight 0.001245
Mode 14, ITERS 22, Weight 0.001067
Mode 15, ITERS 22, Weight 0.002046
Mode 16, ITERS 22, Weight 0.002796
Mode 17, ITERS 22, Weight 0.003838
Mode 18, ITERS 22, Weight 0.003327
Mode 19, ITERS 22, Weight 0.001809
Mode 20, ITERS 22, Weight 0.004366
Mode 21, ITERS 22, Weight 0.002385
Mode 22, ITERS 2, Weight 0.005861
Mode 23, ITERS 13, Weight 0.000000
```

Figura 17. Modos con peso de simulación estática lineal con cargas definitivas.

En la figura 18 puede verse el campo de desplazamientos para la aplicación de la carga en el nodo 23.

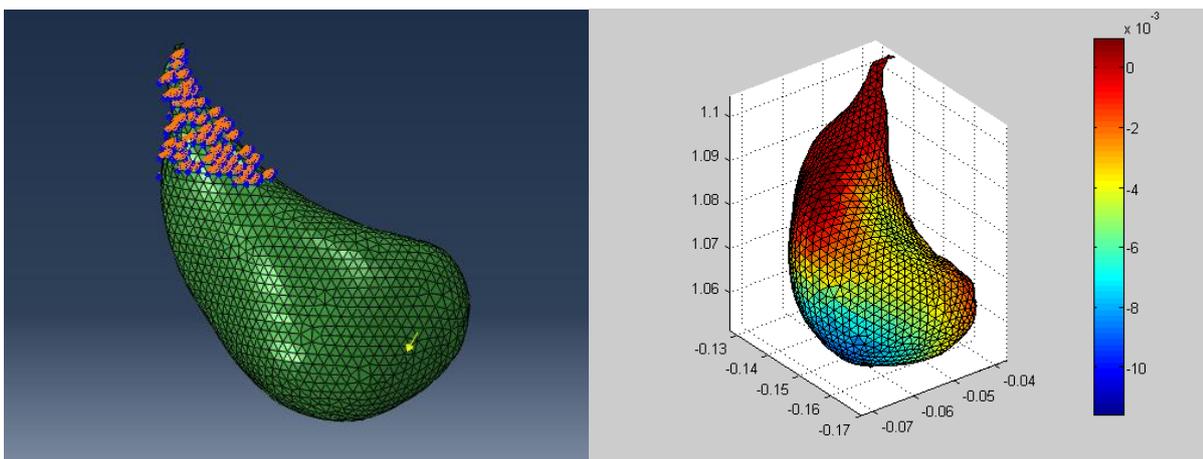


Figura 18. Particularización de la solución para carga en nodo 23 (solución final y sin deformar).

Por último para poder comparar la calidad de los resultados usando la técnica PGD al incorporar numerosos casos de carga y obtener una solución con esta técnica de reducción de modelos, se comparan los resultados considerando el resultado obtenido por Abaqus como resultado de referencia. Para el cálculo de Abaqus se simula la misma condición de cargas para Abaqus imponiendo el cálculo lineal e igualando las condiciones de contorno y las cargas (para ello se colocara una sola carga alojada en el nudo número 3727). La solución obtenida se representa en la figura 19.

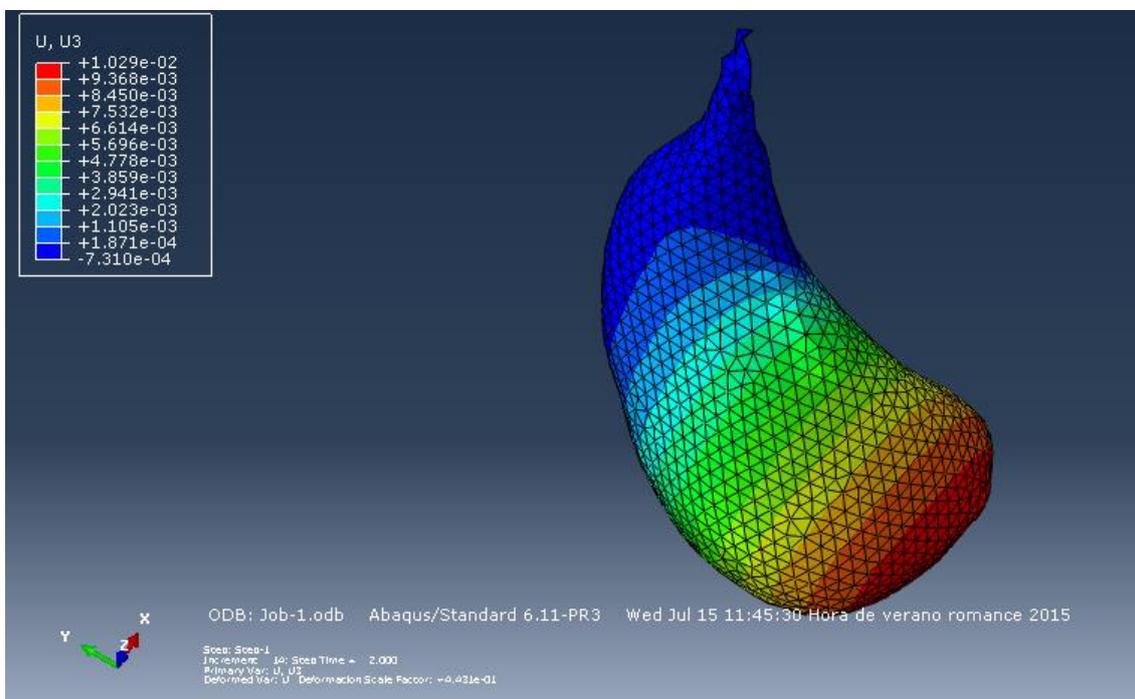


Figura 19. Situación de varias cargas para modelo Abaqus .

La diferencia entre las imágenes 18 y 19 radica en que mientras la imagen de Matlab corresponde a la carga aplicada en el extremo más alejado, dentro de todas las cargas definidas en las condiciones de contorno, el de abaqus es el modelo con todas las cargas alicadas a la vez.

En ella se puede apreciar como los desplazamientos no varían mucho entre un modelo y otro. Pero para profundizar más se procederá al cálculo del error entre ambas soluciones mediante la siguiente fórmula:

$$\text{Error (\%)} = \frac{|Valor_{Matlab} - Valor_{Abaqus}|}{Valor_{Matlab}} \times 100$$

En ella para simplificar los cálculos se introducirán solo los valores máximos obtenidos para cada método. Siendo el mayor desplazamiento medido con la aplicación PGD de Matlab de $10 \cdot 10^{-3} \text{m}$ y con Abaqus de $10,29 \cdot 10^{-3} \text{m}$.

Así pues queda que el Error relativo entre los dos métodos es de un 2.9% un error más que aceptable para la comunidad de la simulación biomecánica.

3.2.4 Variación del error relativo en función del numero de modos de carga

Una vez ejecutados los cálculos elásticos procederemos al cálculo del error relativo variando el número de modos de carga de la solución (este error será el referido a la norma L_2 , que es la integral de la suma de los desplazamientos de todo el dominio). Como se observa en la figura 20, una vez llegados a los 50 modos PGD el error cometido entre el método PGD y el cálculo con Abaqus es inferior al 0.5%. Al ejecutar el cálculo más allá de los 200 modos PGD observamos como el error es prácticamente inexistente (del orden de 0.05%).

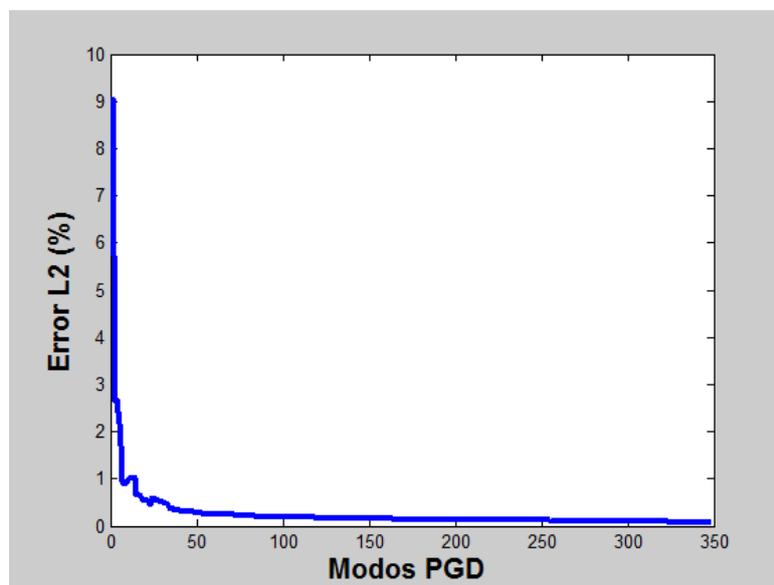


Figura 20. Variación del error relativo en función de los modos PGD de carga.

3.3 Calculo estático no lineal

En este apartado el material se simula bajo un comportamiento hiperelástico. Esto implica que el material está sometido a grandes deformaciones pero no presenta una deformación permanente (sin que haya disipación interna de energía). Estos materiales no tienen en consideración las deformaciones pasadas y su comportamiento no presenta histéresis. En ellos la relación esfuerzo deformación se deriva de una función de energía de deformación.

El cálculo elástico no lineal que realiza la aplicación es diferente al lineal. Este será la suma de los casos lineales en los que se podrá descomponer el problema. Así pues para un dominio no lineal como el de la figura 21 ,en el que la solución no toma un comportamiento lineal, se puede descomponer la solución en una sucesión de problemas lineales, a través de unos incrementos de fuerza, de tal manera que la solución sea parecida, como ocurre en la figura22.

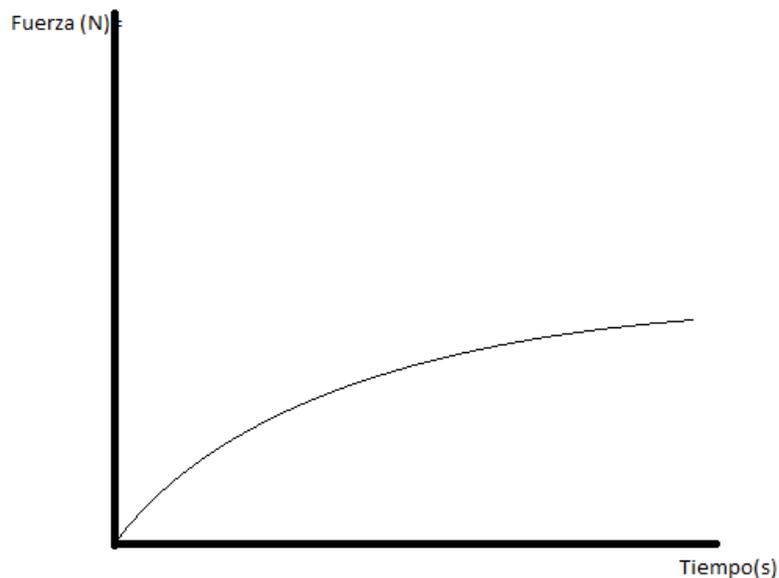


Figura 21. Ejemplo de problema no lineal

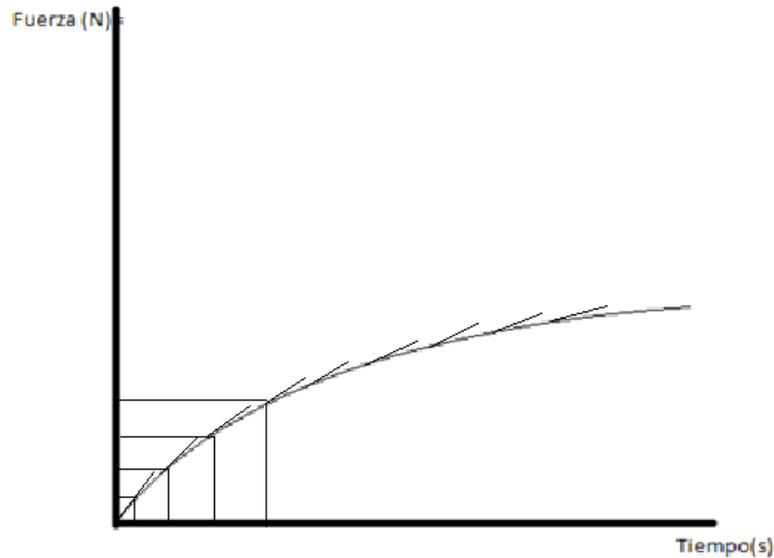


Figura 22. Discretización del problema no lineal en suma de problemas lineales

Una vez establecidas las condiciones de contorno y de cargas se obtiene la solución PGD, de nuevo, como una única solución que se puede particularizar a varios casos de carga distintos. En el caso de simulaciones de materiales hiperelásticos, la técnica utilizada hace una linealización de las ecuaciones y aplica la carga total del problema como una secuencia de incrementos de carga resolviendo cada vez, problemas elásticos lineales más sencillos. En [2] se puede consultar una descripción más detallada de la técnica.

Así pues se puede observar en la figura 23 como queda el estado final de desplazamientos en la vesícula una vez se ejecuta la aplicación, particularizando la solución para un caso de cargas concreto. Se observa como con respecto al estático lineal se produce una subida de aproximadamente unos 2 mm en el máximo desplazamiento de la vesícula.

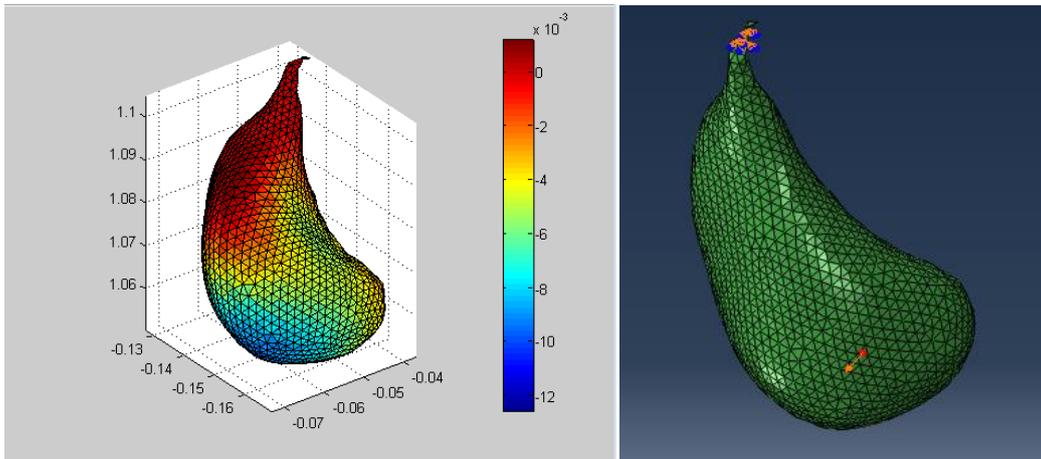


Figura 23. Solución final del problema estático no lineal (solución final y sin deformar).

En las figuras 24 y 25 se puede observar el comportamiento de la vesícula a la hora de particularizar la solución en distintos nodos, obteniendo esta solución (postproceso) en tiempo real, una vez calculada la solución multiparmétrica en fase off-line previa.

Aplicación del caso de carga 1

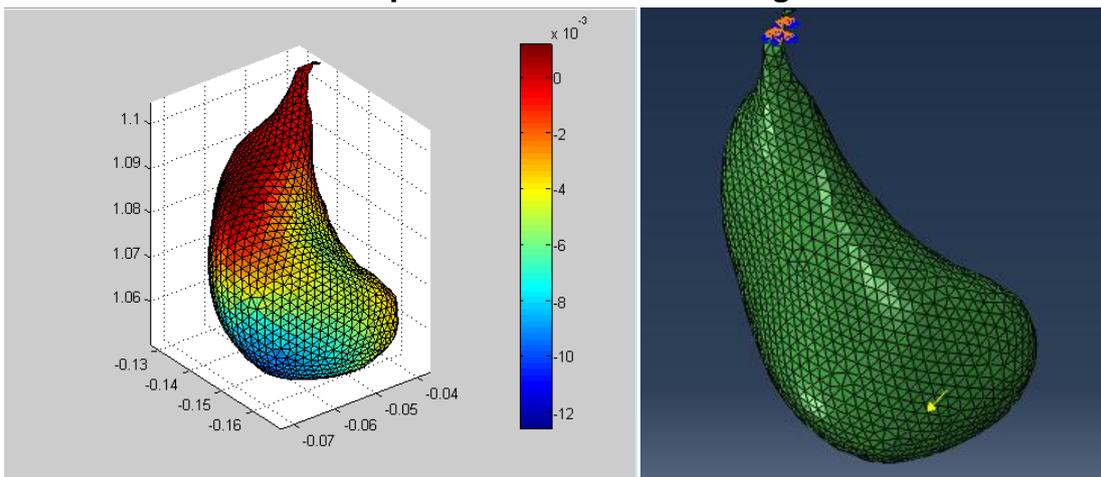


Figura 24. Modo 1 del cálculo estático no lineal (solución final y sin deformar).

Aplicación del caso de carga 2

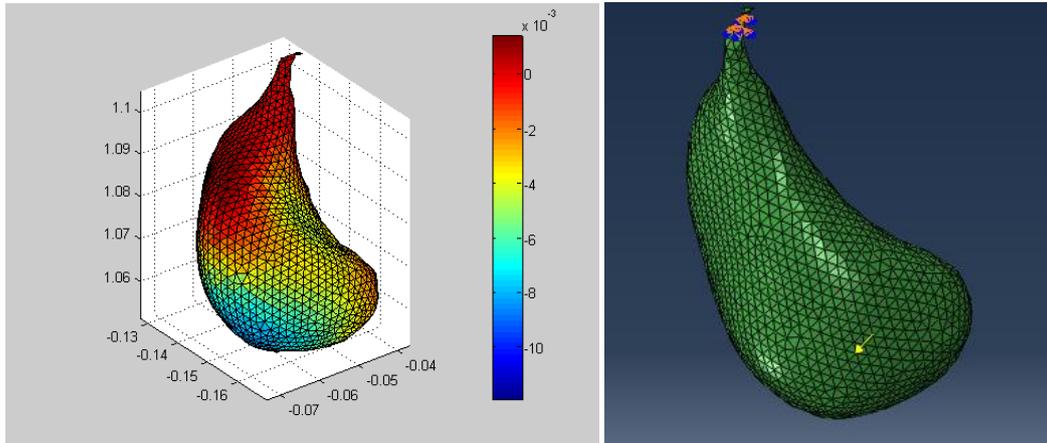


Figura 25. Modo 2 del cálculo elástico no lineal (solución final y sin deformar).

Estos cálculos permitirán en un futuro acoplarlos con otras soluciones similares de otros órganos para la simulación del contacto entre órganos.

4 Cálculo de la vesícula hiperelástica en régimen dinámico

Tal y como se comenta y explica en [3], se puede proceder mediante PGD al cálculo de una solución multiparamétrica dinámica para la vesícula.

Antes de iniciar los cálculos con la aplicación de Matlab será necesario generar un fichero Workspace a través las variables de desplazamiento que se obtendrán del cálculo dinámico en Abaqus. Esto es debido a que para el cálculo dinámico la aplicación PGD necesita nutrirse de los modos POD de una simulación previa para el cálculo dinámico.

4.1 Generación de la solución dinámica en Abaqus para generación de base reducida mediante técnicas POD

Para obtener una base reducida POD (modos de mayor energía) hay que realizar un cálculo dinámico en Abaqus. Por tanto para poder obtenerlo es necesario obtener los desplazamientos para cada instante de tiempo de la simulación, del cálculo dinámico generado por Abaqus.

Por tanto es necesario generar un modelo de la vesícula en Abaqus que servirá para generar una base reducida POD del comportamiento de la vesícula y que será la base en la que proyectar la solución PGD que se creará tal y como se desarrolla en [5].

El cálculo dinámico se ejecutará con un step dinámico (en el que se asignará un tiempo de cálculo máximo de 2 segundos con incrementos de tiempo de 0.015 segundos) y se impondrán las cargas y condiciones de contorno que aparecen en los apartados 3.1.1 y 3.1.2.

Mediante un simple postproceso (Anexo III), será posible capturar la base POD óptima que servirá como base reducida para la solución multiparamétrica que ofrece la PGD.

4.2 Cálculo dinámico con la aplicación PGD de Matlab

Antes de empezar a ejecutar el cálculo dinámico es necesario efectuar el desarrollo de las condiciones de contorno y de cargas que sean idénticas a las que hemos introducido en el modelo de Abaqus; hay que tener en cuenta que ambos tienen los mismos ficheros para las coordenadas y las conectividades de los nodos que forman los elementos.

4.2.1 Generación del cálculo elástico dinámico de la vesícula

Una vez obtenido el modelo POD reducido mediante Abaqus (como se explica en el anexo III), se ejecutará el cálculo dinámico elástico de la vesícula biliar. El resultado de este cálculo se llevará a un postproceso donde se podrán visualizar la deformación de la vesícula, como se observa en las figuras 26 y 27.

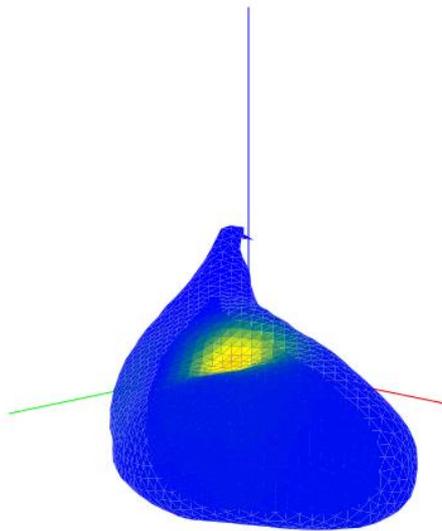


Figura 26. Postproceso de la vesícula imagen 1

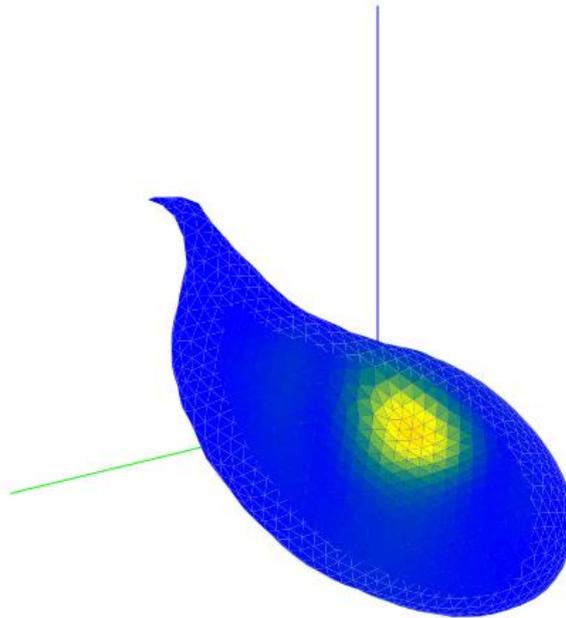
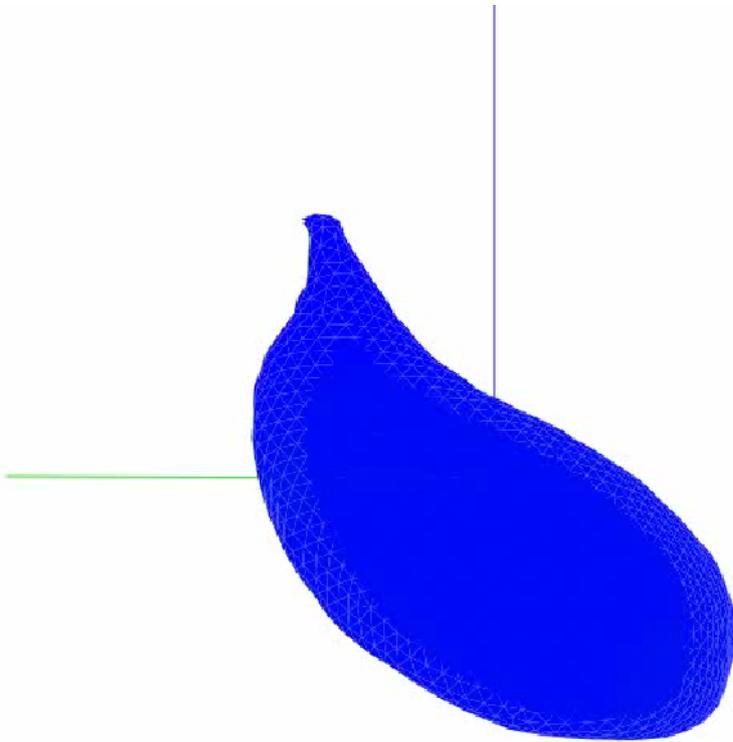
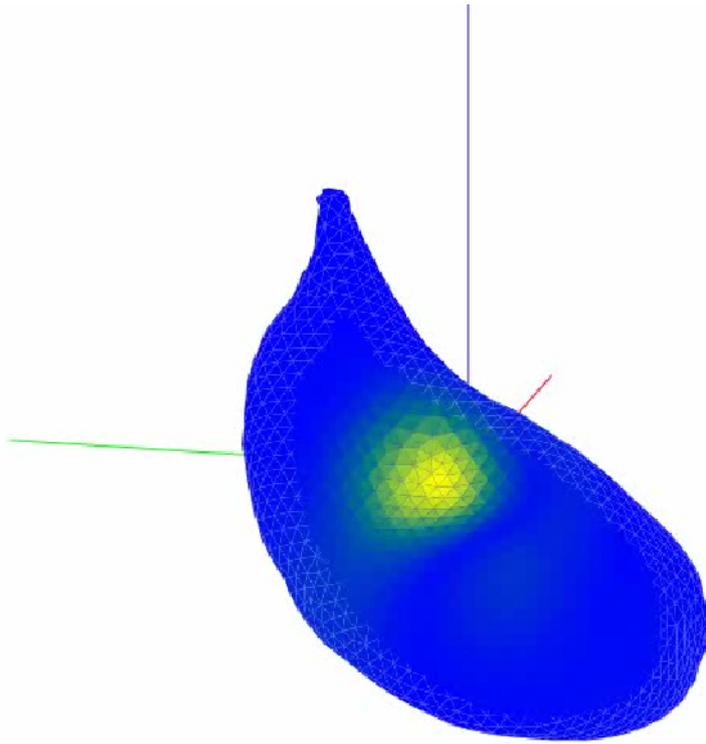


Figura 27. Postproceso de la vesícula biliar imagen 2

Cabe comentar que para este cálculo para hacer los desplazamientos más visibles se ha presentado una zona de cargas más grande que las explicadas en el apartado 3.1.2 (Evitando así interpolar los casos de carga). Esto es debido a que así en el postproceso se intuyen mejor los desplazamientos y se puede hacer interaccionar mayor parte del órgano.

A continuación se podrán observar unos videos en los que se aprecia el postproceso y la influencia de los modos PGD en la deformación del órgano.



5 Conclusiones

Ahora expondré las conclusiones obtenidas del desarrollo de este TFG y los cálculos realizados.

Para empezar la mayoría de software simuladores llegan a tardar varias horas en realizar los cálculos debido a que estos los desarrollan on-line (o sea los cálculos se realizan en el mismo momento que se imponen las condiciones del problema). La aplicación PGD realiza los cálculos en dos fases: una off-line (antes de empezar a simular, en la que se realizan los cálculos de los desplazamientos para cada caso de carga) y otra on-line (en la que se ejecuta directamente la solución según las condiciones de carga que nos encontremos) de tal manera que el tiempo se ve muy reducido y pueden hacerse cambios instantáneamente.

La precisión de la aplicación hace que para cada posición de carga, la aplicación devuelve un campo de desplazamientos muy cercanos a los que tendría el órgano realmente (con errores entorno al 2%).

Posteriormente con los cálculos dinámicos se ha comprobado la respuesta del órgano como una buena aproximación al comportamiento dinámico de la vesícula biliar dentro de los márgenes de tiempo de simulación en tiempo real que concluyen que la técnica parece una herramienta muy apropiada para este tipo de simulaciones.

En un futuro se estudiara el contacto entre órganos para poder simular operaciones de cirugía en tiempo real.

ANEXOS

Anexo I: método del punto Fijo

Como ya se ha comentado en el punto 2.1 del proyecto, en el método de punto fijo cada interacción cuenta con los siguientes pasos:

- Calculo de la función $R(x)$ actual, siendo conocida $S(y)$ anterior.
- Calculo de la función $S(y)$ conocida $R(x)$.

Calculo de la función $R(x)$

Recordamos pues que la función solución del sistema es:

$$u^n(x, y) = \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y) + R(x) \cdot S(y),$$

Donde todas las funciones son conocidas excepto $R(x)$. Elegimos pues una función de ponderación u^* de tal manera que la solución del sistema pasa a ser:

$$u^*(x, y) = R^*(x) \cdot S(y) + R(x) \cdot S^*(y), \quad (2.7)$$

Que es equivalente a seleccionar la forma residual ponderada de Galerkin en la ecuación de Poisson. De tal manera que la función $S(y)$ es conocida y $S^*(y) = 0$. Por lo tanto la ecuación queda reducida a:

$$u^*(x, y) = R^*(x) \cdot S(y). \quad (2.8)$$

De tal manera que sustituyendo la ecuación 2.7 en la 2.3 obtenemos que:

$$\begin{aligned} \int_{\Omega_x \times \Omega_y} R^*(x) \cdot S(y) \cdot \left(\frac{d^2 R(x)}{dx^2} \cdot S(y) + R(x) \cdot \frac{d^2 S(y)}{dy^2} \right) dx \cdot dy = \\ = - \int_{\Omega_x \times \Omega_y} R^*(x) \cdot S(y) \cdot \sum_{i=1}^{n-1} \left(\frac{d^2 X_i}{dx^2} \cdot Y_i + X_i \cdot \frac{d^2 Y_i}{dy^2} \right) dx \cdot dy + \\ + \int_{\Omega_x \times \Omega_y} R^*(x) \cdot S(y) \cdot f(x, y) dx \cdot dy. \end{aligned} \quad (2.9)$$

Donde se podrán usar variables separadas (como SVD para bidimensional o HOSVD para alto orden), para así simplificar las dimensiones de cálculo y obtener la solución de $R(x)$.

Calculo de S(y) a partir de R(x)

El proceso será idéntico al anterior en el que simplemente se intercambiarán los valores de las funciones relevantes x e y. De tal manera que ahora la ecuación 2.8 quedaría de la siguiente manera:

$$u^*(x, y) = R(x) \cdot S^*(y).$$

Y al sustituir en 2.2 quedaría de la siguiente manera:

$$\begin{aligned} \int_{\Omega_x \times \Omega_y} R(x) \cdot S^*(y) \cdot \left(\frac{d^2 R(x)}{dx^2} \cdot S(y) + R(x) \cdot \frac{d^2 S(y)}{dy^2} \right) dx \cdot dy = \\ = - \int_{\Omega_x \times \Omega_y} R(x) \cdot S^*(y) \cdot \sum_{i=1}^{n-1} \left(\frac{d^2 X_i}{dx^2} \cdot Y_i + X_i \cdot \frac{d^2 Y_i}{dy^2} \right) dx \cdot dy + \\ + \int_{\Omega_x \times \Omega_y} R(x) \cdot S^*(y) \cdot f(x, y) dx \cdot dy. \end{aligned}$$

Que como antes basta con utilizar cualquiera de los métodos de variables separadas para simplificar la ecuación y obtener el valor de S (y).

Anexo II: Aplicación PGD de Matlab

Generación de una solución PGD

La aplicación consta de tres pasos fundamentales como la mayoría de programas computacionales:

- Preproceso: Se generan los documentos iniciales del análisis del problema (geometría, comportamiento, tipo análisis, propiedades del problema a tratar, fuerzas, etc.).
- Solución: Puede llegar a tardar un gran tiempo en obtenerse (depende del tipo de análisis y la cantidad de condiciones de contorno). La concepción de esta de esta solución multidimensional se realiza una única vez (etapa off-line) y permite que la siguiente etapa (el postproceso) obtener la solución de manera instantánea para cualquiera de las posibles posiciones de carga que se puedan imaginar.
- Postproceso: una vez desarrollada la solución por el método PGD esta se almacena en forma vectorial (esta incluirá todas las posibles situaciones de carga a las que haga frente nuestro problema). La solución se almacenará en el directorio de *Outputs* que se muestra en la Figura 28 y podrá ser exportado a varios tipo de formato (como por ejemplo java, C++, TecPlot...).

Nombre	Fecha de modifica...	Tipo	Tamaño
Outputs	11/06/2015 17:56	Carpeta de archivos	
Problems	02/06/2015 11:33	Carpeta de archivos	
metadata	01/06/2015 16:51	Carpeta de archivos	
WS_	11/06/2015 18:32	Microsoft Office A...	667 KB
PGD0x2D1App.m	01/06/2015 16:51	MATLAB Code	3 KB
PGD.mlappinstall	01/06/2015 16:51	MATLAB App Inst...	536.597 KB
main.m	17/04/2015 9:51	MATLAB Code	1 KB
elemstiffNL_estatico.p	17/04/2015 9:40	MATLAB P-code	1 KB
fem3DHiperelastico.p	17/04/2015 9:40	MATLAB P-code	2 KB
pgdlauncher.p	13/04/2015 17:10	MATLAB P-code	5 KB
elemstiff_3D.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimiento.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimiento_explicito_cell.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimiento_nolin.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimiento_substep1.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimiento_substep2.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimientoNL_explicito_cell.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimientoNL_substep1.p	13/04/2015 16:31	MATLAB P-code	1 KB
enriquecimientoNL_substep2.p	13/04/2015 16:31	MATLAB P-code	1 KB
evalua_fforma.p	13/04/2015 16:31	MATLAB P-code	1 KB
fea.p	13/04/2015 16:31	MATLAB P-code	1 KB
fem3D.p	13/04/2015 16:31	MATLAB P-code	2 KB
fem3DHiperelasticoNL.p	13/04/2015 16:31	MATLAB P-code	2 KB

Figura 28. Directorio de la aplicación PGD con todas sus carpetas

Preproceso

Antes de ejecutar cualquier tipo de cálculo es necesario tener una serie de archivos iniciales para resolver el tipo de problema que vayamos a hacer (Figura 29). Así pues los archivos necesarios son los siguientes:



Nombre	Fecha de modifica...	Tipo	Tamaño
defaultBladder.m	03/06/2015 19:27	MATLAB Code	3 KB
Forcefinales.m	21/05/2015 17:53	MATLAB Code	1 KB
bcnodesfinales.m	21/05/2015 17:52	MATLAB Code	1 KB
bcnodesCreated_2014_Oct_01_10_16.m	17/04/2015 9:50	MATLAB Code	1 KB
xtrascript.m	17/04/2015 9:50	MATLAB Code	1 KB
ForceAll.m	17/04/2015 9:50	MATLAB Code	1 KB
ForceNoForce.m	17/04/2015 9:50	MATLAB Code	1 KB
ForceOneNode.m	17/04/2015 9:50	MATLAB Code	1 KB
InternalPressure.m	17/04/2015 9:50	MATLAB Code	63 KB
.DS_Store	22/12/2014 12:41	Archivo DS_STORE	7 KB
Force_Generator.p	04/11/2014 16:50	MATLAB P-code	1 KB
conecBladder.dat	25/09/2014 14:56	Archivo DAT	943 KB
conecHoleBladder.dat	25/09/2014 14:55	Archivo DAT	584 KB
gcoordBladder.dat	25/09/2014 14:52	Archivo DAT	338 KB
bcnodes_Generator.p	23/09/2014 17:08	MATLAB P-code	1 KB
ShellBladder.dat	17/09/2014 9:28	Archivo DAT	214 KB
WorkSpaceBladder_NoValido_POD	22/01/2014 13:55	Microsoft Office A...	5.483 KB

Figura 29. Ejemplo ficheros dentro de la carpeta Bladder

- gcoord<Nombre del Problema>.dat: En este archivo aparecen las coordenadas del objeto o problema que vayamos a resolver. Las coordenadas serán en 3D y estarán organizadas en columnas.
- conect<Nombre del Problema>.dat: En este fichero se definirán los elementos del dominio a partir de los puntos espaciales que los conforman.
- Default<Nombre del Problema>.dat: En este fichero contiene los datos por defecto que aparecerán en las diferentes ventanas de ejecución (ejemplo Figura 30).

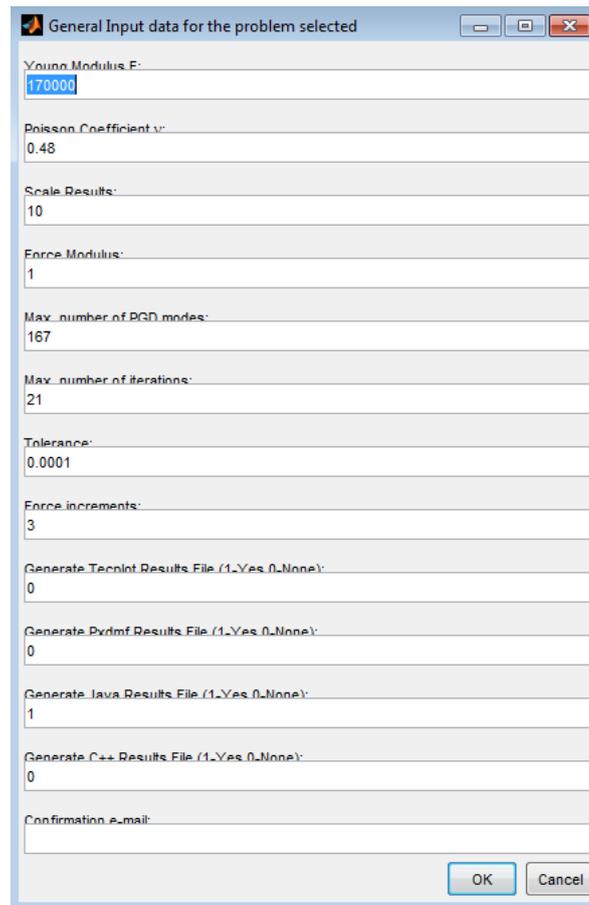


Figura 30. Ejemplo DefaultBladder2

- Workspace.mat: Estos ficheros son soluciones de regencia obtenidas con otro tipo de software de elementos finitos (como por ejemplo en este proyecto usaremos Abaqus, el modo de generación de esto y del fichero POD del proyecto se explicarán más tarde). Puede haber de dos tipos: una solución POD (en el que se extraen los modos POD para el cálculo dinámico) y la solución MODAL (en la que se usará un análisis modal para el cálculo de los modos dinámicos).
- bcnodes*.m (.p): Este fichero encierra las condiciones de contorno del problema a través de un vector en el que aparecen los números de los nodos seleccionados. Entre estos puede haber, como pasara también con las fuerzas, un fichero bcnodes_Generator.p que permitirá al usuario introducir manualmente las condiciones de contorno dentro del propio dominio de la aplicación.
- Force*.m: Este al igual que el anterior será un archivo en el que aparezca un vector pero esta vez de cargas del problema. Y como antes, también

habrá un Force_Generator.p para introducir manualmente las cargas dentro del dominio de la aplicación.

- xtrascript.m: En este fichero se podrán escribir sentencias y funciones que se consideren convenientes para crear o modificar datos dentro de la aplicación.

Por tanto con el desarrollo de estas carpetas tendremos definido un modelo o problema para su resolución mediante la aplicación PGD.

Solución

Al ejecutar la aplicación se obtiene la solución del cálculo PGD. Este es un cómputo iterativo en el que la solución va generando modos PGD, formados por productos de funciones que dependen de variables que se obtiene de la solución del problema, de tal manera que al superponerlas se obtenga la solución final.

En el caso de los cálculos estáticos lineales la solución dependerá tanto del espacio (3D) como del estado de cargas. Mientras que en el cálculo dinámico dependerá además: del tiempo en T-1 y T-2 en el caso explícito y de la velocidad y aceleración en el caso implícito (como ya se comenta [5]).

Para cada tipo de cálculo se dará una salida distinta que a continuación se explica:

1. Cálculo estático lineal: en el cual el producto de las variables del modo anterior se le suma el producto de las variables del siguiente modo y se hacen converger. A medida que se añaden nuevos modos al problema la aplicación calcula el número de interacciones que le lleva converger.
2. Cálculo estático no lineal (hiperelástico): la aplicación genera una serie de soluciones lineales para incrementos de la carga final, o sea, aplica problemas más sencillos con una parte de la carga. La solución final será la suma de los problemas individuales.
3. Cálculo dinámico lineal: los modos se almacenan en una variable en celdas denominada F. La primera de estas celdas será la geometría, luego las variables internas necesarias para la ejecución del esquema de integración temporal y las dos últimas las relativas a cargas.
4. Cálculo dinámico no lineal: igual que el anterior aportando comportamiento no lineal a las ecuaciones.

Postproceso

Es el último paso de la simulación; recordemos que las variables de salida están guardadas en un entorno Workspace de Matlab.

Para la ejecución la aplicación permite un rango muy limitado de métodos de visualización entre los que destacamos: java, C++, Tecplot o archivos pxdmf (estos últimos permiten su compatibilidad con smartphones y tablets, por lo que son de gran utilidad porque confieren al método de gran portabilidad y facilidad de visualización).

Anexo III: Generación de un modelo de base reducida POD en Abaqus.

Como ya se ha explicado brevemente en el apartado 4.1, una vez generada la solución con el cálculo dinámico será necesario exportar las soluciones de los desplazamientos de Abaqus a Matlab. Para ello empezaremos generando la lista de desplazamiento de todos los nodos de la vesícula simulada. Para realizar esto habrá que generar un fichero XY Data en la ventana de postproceso de Abaqus, desde un ODB Field Output. En el habrá que añadir todos los nodos del dominio seleccionado manualmente toda la geometría de la vesícula. El resultado será una lista de desplazamientos para cada nodo como se puede observar en la figura 31.

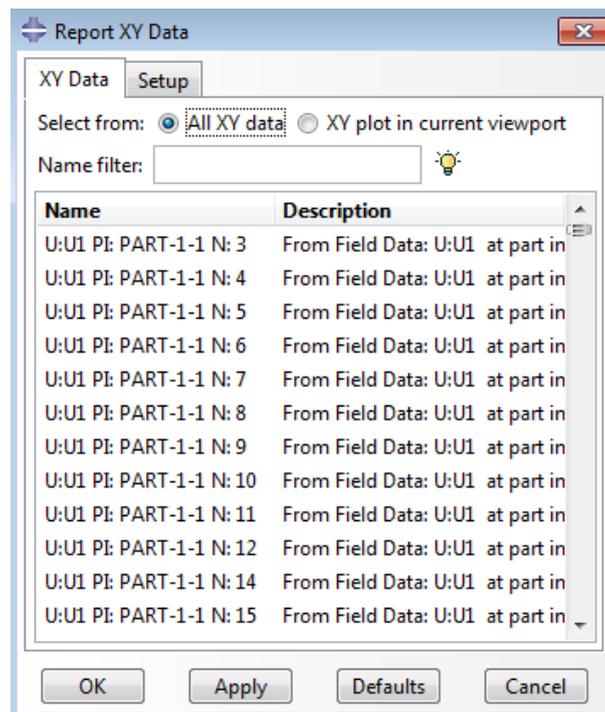
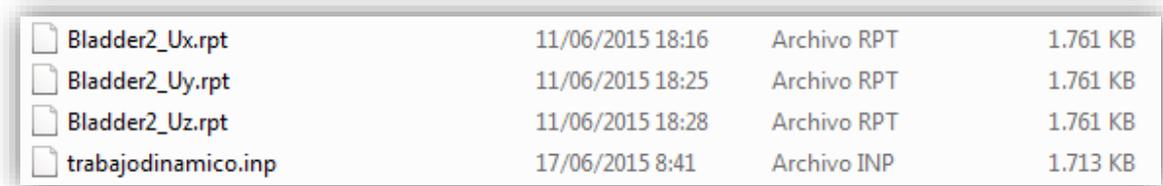


Figura 31. Reports del XYData para desplazamientos.

Una vez obtenida la lista con los nodos y desplazamientos estos se seleccionan y se guardan en formato.rpt con el nombre que le hemos puesto a la carpeta en la que guardamos los archivos de la vesícula. Una vez obtengamos los desplazamientos para las tres direcciones ya habremos acabado nuestra tarea con Abaqus.

Es útil generar también un fichero .inp de Abaqus en el que estén definidas todas las situaciones del cálculo dinámico que hemos ejecutado, dado que así podremos obtener los ficheros gcoord, conect, cargas y las condiciones de contorno, necesario para el cálculo dinámico en la carpeta de la aplicación PGD.



Bladder2_Ux.rpt	11/06/2015 18:16	Archivo RPT	1.761 KB
Bladder2_Uy.rpt	11/06/2015 18:25	Archivo RPT	1.761 KB
Bladder2_Uz.rpt	11/06/2015 18:28	Archivo RPT	1.761 KB
trabajodinamico.inp	17/06/2015 8:41	Archivo INP	1.713 KB

Figura 32. Ficheros .rpt y .inp obtenidos del cálculo dinámico con Abaqus.

A continuación para terminar hay que crear la carpeta Workspace que servirá como base para el cálculo dinámico con Abaqus. Para ello la aplicación dispone de un fichero situado en la carpeta PGD llamo PODscript.p ejecutando este fichero y con el resto de fichero contenidos en la carpeta necesarios para el cálculo (tales como: el gcoordBladder, el conectBladder y los ficheros.rpt que se obtienen del inp de Abaqus) obtendremos un Workspace para el cálculo dinámico con la aplicación.

BIBLIOGRAFÍA

[1] A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part ii: transient simulation using spacetime separated representations. Ammar, A., Mokdad, B., Chinesta, F., , and Keunings., R. (2007). *J. Non-Newtonian Fluid Mech.*, 144:98--121.

[2] Real-time simulation of biological soft tissues: a PGD approach. S. Niroomandi, D. Gonzalez, I. Alfaro, F. Bordeu, A. Leygue, E. Cueto, F. Chinesta. *International Journal for Numerical Methods in Biomedical Engineering*, 29(5), 586-600, 2013.

[3] Computational vademecums for the real-time simulation of haptic collision between nonlinear solids. D. Gonzalez, I. Alfaro, C. Quesada, E. Cueto, F. Chinesta. *Comput. Method Appl. Mech. Engrg.*, (283), 210-223, 2015. DOI: 10.1016/j.cma.2014.09.029

[4] Real time simulation for Computational Surgery. E. Cueto, F. Chinesta. CICYT-DPI2011-27778-C02-01.

[5] Real-time direct integration of reduced solid dynamics equations. D. González, E. Cueto , F. Chinesta. *International Journal for Numerical Methods in Biomedical Engineering*, Int. J. Numer. Meth. Engng 2012; 00:1–28.

[6] Un método de Descomposición Propia Generalizada para operadores de alto orden. C. Quesada.

[7] Real-time deformable models for surgery simulation: a survey. M. Meier, U., Lopez, O., Monserrat, C., Juan, M., Alcaniz. *Computer Methods and Programs in Biomedicine*.77(3), 183-197(2005).DOI10.1016/j.cmpb.2004.11.002.

[8] [www.http://scielo.isciii.es/scielo.php?pid=S021271992004000900009&script=sci_arttext](http://www.scielo.isciii.es/scielo.php?pid=S021271992004000900009&script=sci_arttext).

[9] <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4285607/>

[10] <http://www.bondisalud.com.ar/47.html>

[11] Un método de Descomposición Propia Generalizada para operadores de alto orden. Carlos Quesada Granja.