



Universitat de les Illes Balears

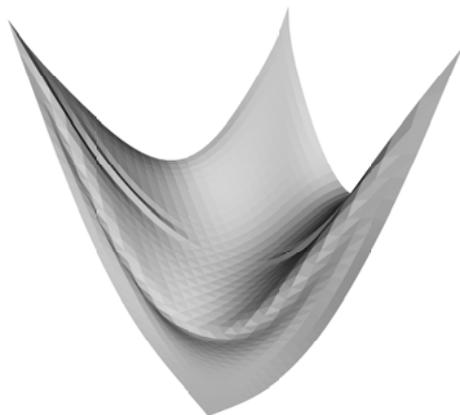
Departament de Ciències Matemàtiques i Informàtica

TESIS DOCTORAL

**SIMULACIÓN DINÁMICA Y DEFORMACIONES DE
SUPERFICIES PARAMÉTRICAS**

Autor: Pedro Antonio PALMER RODRÍGUEZ

Directores: Dr. Manuel GONZÁLEZ HIDALGO
Dr. Arnau MIR TORRES



marzo de 2014

D. Manuel González Hidalgo, Doctor en Informàtica per la Universitat de les Illes Balears i Professor Titular d'Universitat de l'àrea de coneixement de Ciències de la Computació i Intel·ligència Artificial del Departament de Ciències Matemàtiques i Informàtica de la Universitat de les Illes Balears, i

D. Arnau Mir Torres, Doctor en Matemàtiques per la Universitat de Barcelona i Professor Titular d'Universitat de l'àrea de coneixement de Ciències de la Computació i Intel·ligència Artificial del Departament de Ciències Matemàtiques i Informàtica de la Universitat de les Illes Balears,

FAN CONSTAR:

que la present memòria “SIMULACIÓN DINÁMICA Y DEFORMACIONES DE SUPERFICIES PARAMÉTRICAS” presentada per Pedro Antonio Palmer Rodríguez per optar al grau de Doctor en Informàtica, ha estat realitzada sota la seva direcció i reuneix suficient matèria original per ser considerada com a tesi doctoral.

Palma, març de 2014

Doctorand

Directors

Pedro Antonio Palmer Rodríguez Manuel González Hidalgo Arnau Mir Torres

*A na Peque,
la persona que
li ha donat sentit
a la meva vida.*

*A en Pere i en Miquel,
el sentit de la meva vida.*

Agradecimientos

Este trabajo empezó hace ya mucho tiempo, y parece que ha llegado a su fin. Desde que en su día empecé hasta ahora, multitud de vicisitudes han acontecido en mi vida. La principal de ellas es, seguramente, la que podría llamarse *experiencia vital*. Desde el jovencuelo presuntuoso que era hasta la persona respetuosa que — creo — soy ahora, dista un gran trecho. El camino recorrido ha sido el del aprendizaje, el de la búsqueda del conocimiento y el de buscar los defectos e intentar mejorarlos. En este largo, y no siempre agradable, camino he compartido experiencias con un gran número de personas, muchas de las cuales han aportado algo positivo. Desearía dedicar el resto de este apartado a agradecer a todas su aportación.

Querría especialmente agradecer la dedicación de mis dos directores de tesis, el Doctor Manuel González Hidalgo y el Doctor Arnau Mir Torres. Ellos me han enseñado el camino que se debe seguir para realizar un trabajo de investigación riguroso. Debo agradecerles también su fe en mi trabajo, su perseverancia y su apoyo, muy superior, seguro, al que muchos otros me habrían dado.

También querría destacar aquí a mis compañeros en el Departament de Ciències Matemàtiques i Informàtica, en especial a mi querido compañero Miquel Mascaró Portells, ¡Qué habría sido de mí sin él! También al resto de compañeros de *Unitat*, Antoni Bibiloni, *aaara!*, *ja era ben hora!* y Biel Fontanet. A mi buena amiga Mercè, que me ha ofrecido su amistad y su consejo en todas las circunstancias. ¡Cómo no! a Ricardo Alberich, gran amigo, a Margaret Miró y a Xavi Varona, el equipo directivo del Departamento, por haberme sufrido y ayudado tanto. También a Miquel Mascaró Oliver, *Tot arriba!*. Por supuesto, a Ramon Mas, que siempre ha estado dándome ánimos. Y a muchos, muchísimos más, de hecho a todos.

Por supuesto quiero agradecer además a mi gran amigo Miquel Garcies, y también a Cati y a Andreu, por darme tan buenos momentos, tantos consejos y tanta comprensión.

Y a mi familia, especialmente a mi madre y a mi padre, al que estoy seguro que le habría encantado leer estas líneas, a mi hermana Cati, a Juan, a Marc, a Fernando y a Carmen, y a todos y cada uno de sus hijos, mis cuñados, mis cuñadas y mis sobrinos, y a Julita. A todos ellos muchas gracias por aguantarme y comprenderme. *I molt especialment a tu Tega-tega, però és un secret entre tu i jo!*

Y, por encima de todos, a mi querida Peque, ella me ha aguantado, y me ha consolado, y me ha animado, y siempre me ha apoyado.

Finalmente, a Pere y Miquel, mis hijos, mi verdadero proyecto, mi ilusión.

Resumen

En el presente trabajo se desarrolla un modelo de representación de superficies parametrizadas en un dominio tetradimensional, basado en NURBS, al que se ha denominado BSpline4D. Como el objetivo es la representación y simulación dinámica de superficies deformables basadas en el modelo propuesto, ha sido realizado un estudio previo de las ecuaciones del movimiento, asociando un funcional de energía como potencial para medir la deformación de los objetos, para determinar la forma en que es posible adaptarlas al modelo, considerando los puntos de control 4D como coordenadas generalizadas. Se ha realizado un estudio riguroso sobre posibles métodos de integración, y de discretización tanto temporal como espacial, para determinar su grado de adecuación a la hora de resolver el sistema de ecuaciones diferenciales que se genera. El movimiento y la simulación dinámica de la deformación de las superficies parametrizadas se realiza exclusivamente con la determinación de los puntos de control 4D en cada paso temporal. La estabilidad numérica de la solución propuesta ha resultado ser excelente y también la eficiencia computacional en tiempo.

La determinación del modelo BSpline4D, para representar los objetos deformables, se ha realizado después de un estudio de los diferentes modelos existentes en la literatura para la representación en el ordenador de objetos deformables, señalando las ventajas e inconvenientes de cada uno de ellos.

Finalmente, el modelo se ha utilizado para desarrollar un modelo de deformaciones de formas libres (FFD), como una extensión del modelo *Scodef* de deformaciones geométricas basado en restricciones. Se han establecido las condiciones para que puedan aplicarse restricciones que sigan trayectorias no rectilíneas, representadas por curvas B-Spline 4D. Con la solución propuesta, la deformación de la superficie se adapta de forma precisa a la forma que describe la curva B-Spline. Al modelo se le ha denominado N-Scodef.

Palabras clave

Simulación, Dinámica, Deformación basada en física, NURBS, B-Splines, Sco-def, Deformación basada en restricciones, ecuaciones de movimiento de Lagrange, Mecánica, FFD, Free-Form Deformation

Resum

En el present treball es desenvolupa un model de representació de superfícies parametritzades en un domini tetradimensional, basat en NURBS, al que s'ha nomenat BSpline4D. Com que l'objectiu és la representació i simulació dinàmica de superfícies deformables basades en el model proposat, ha estat realitzat un estudi previ de les equacions del moviment, associant un funcional d'energia com potencial per mesurar la deformació dels objectes, per a determinar la forma en què és possible adaptar-les al model, considerant els punts de control 4D com coordenades generalitzades. S'ha realitzat un estudi rigorós sobre possibles mètodes d'integració, i de discretització tant temporal com espacial, per a determinar el seu grau d'adequació a l'hora de resoldre el sistema d'equacions diferencials que es genera. El moviment i la simulació dinàmica de la deformació de la superfície parametritzada es realitza exclusivament amb la determinació dels punts de control 4D en cada pas temporal. L'estabilitat numèrica de la solució proposada ha resultat ser excel·lent i també l'eficiència computacional en temps.

La determinació del model BSpline4D, per a representar els objectes deformables, s'ha realitzat després d'un estudi de les diferents formes existents a la literatura per a la representació en l'ordinador d'objectes deformables, assenyalant els avantatges i inconvenients de cada una d'elles.

Finalment, el model s'ha utilitzat per a desenvolupar un model de deformacions de formes lliures (FFD), com una extensió del model *Scodef* de deformacions geomètriques basat en restriccions. S'han establert les condicions per què puguin aplicar-se restriccions que segueixin trajectòries no rectilínies, representades per corbes B-Spline 4D. Amb la solució proposada, la deformació de la superfície s'adapta de forma precisa a la forma que descriu la corba B-Spline. Al model se l'ha nomenat N-Scodef.

Paraules clau

Simulació, Dinàmica, Deformació basada en física, NURBS, B-Splines, Sco-def, Deformació basada en restriccions, equacions de moviment de Lagrange, Mecànica, FFD, Free-Form Deformation

Abstract

This document presents a model for the representation of surfaces parameterized in a four-dimensional domain. The model, based on NURBS, has been called BSpline4D. The main goal is the representation and the dynamic simulation of deformable surfaces based on the proposed model. To determine how the model can be adapted, in order to achieve the main objective, it has been carried out a preliminary study of the equations of motion. An associated energy functional to express the potential to measure the deformation of the objects is attached to the equations of motion; taking into account that the 4D control points of the surfaces have to be considered as the generalized coordinates.

Also, there is a rigorous study of possible methods of integration and discretization, both temporal and spatial, to determine their suitability to solve the system of differential equations generated. The motion and dynamic deformation simulation of the parameterized surfaces is performed only with the determination of the 4D control points on each time step. The numerical stability of the proposed solution has proved to be excellent, and also the computational efficiency in time.

The model BSpline4D, to represent deformable objects, has been designed after a study of the different existing solutions, in the current state of the art, for a computer based representation of deformable objects, noting the advantages and disadvantages of each one.

Finally, the model has been used to develop a free-form deformation model (FFD) as an extension of the *Scodef* model of geometric constraint-based deformations. There have been established the conditions to define restrictions with non rectilinear trajectories, represented by 4D B-Spline curves. In the proposed solution, the deformation of the surface adjusts precisely to the form described by the B-Spline curve. The model has been called N-Scodef.

Keywords

Simulation, Dynamics, Physics based deformation, NURBS, B-Splines, Sco-def, Constraints based deformations, Lagrange's movement equations, Mechanics, FFD, Free-Form Deformation

Índice general

Índice general	I
Índice de figuras	v
Índice de tablas	xI
1. Introducción	1
2. Estudio físico	7
2.1. Modelización según la mecánica clásica	7
2.1.1. Principio de D'Alembert	10
2.1.2. Ecuaciones de Lagrange	12
2.1.3. El principio de Hamilton	15
2.1.4. Sistemas no conservativos	16
2.2. Conclusiones	20
3. Solución de la ecuación de equilibrio	21
3.1. Métodos basados en la integración directa	21
3.1.1. El método de las diferencias centradas	22
3.1.2. El método de las diferencias centradas implícito	23
3.1.3. El método de Houbolt	24
3.1.4. El método Wilson θ	25
3.1.5. El método Newmark β	27

3.2.	Comparación de los métodos de integración	29
3.2.1.	Estabilidad y precisión	30
3.3.	Estudio de los métodos de integración en un caso complejo	33
3.3.1.	Ecuación de un modelo deformable elásticamente	33
3.3.2.	Resolución numérica del modelo de deformación	35
3.3.3.	Discretización temporal	36
3.3.4.	Discretización espacial	37
3.3.5.	Estudio de la estabilidad	40
3.3.6.	Influencia de los coeficientes η y ξ sobre la deformación	84
3.3.7.	Coste computacional del comportamiento del modelo	87
3.3.8.	Ejemplos de simulación	91
3.4.	Resumen y conclusiones	92
4.	Representación de objetos deformables	99
4.1.	Modelos discretos	100
4.1.1.	Mallados poligonales	101
4.1.2.	Sistemas de partículas	105
4.1.3.	Modelos masa-resorte	107
4.2.	Modelos explícitos	109
4.2.1.	Cilindros generalizados	110
4.2.2.	Modelos de contornos activos	111
4.2.3.	Objetos continuos	112
4.3.	Modelos implícitos	114
4.3.1.	Superficies algebraicas	114
4.3.2.	Isosuperficies	119
4.3.3.	Conjuntos de nivel	122
4.3.4.	Representación de Objetos por Muestreo	124
4.4.	Modelos paramétricos	124
4.4.1.	Supercuádricas	126
4.4.2.	Descomposición modal	126

4.4.3. Superficies de subdivisión	127
4.4.4. Curvas de Bézier	130
4.4.5. B-Splines	133
4.4.6. NURBS	136
4.4.7. T-Splines	139
4.5. Conclusiones	141
5. BSpline4D un nuevo modelo de representación	143
5.1. Hipótesis	143
5.2. Desarrollo	143
5.2.1. Formulación	144
5.3. Deformación del modelo BSpline4D	157
5.3.1. Discretización temporal	157
5.3.2. Discretización espacial	157
5.3.3. Imposición de restricciones	162
5.4. Estabilidad	168
5.4.1. Influencia de los coeficientes en el funcional de energía . . .	176
5.5. Comparación con otros modelos de representación	179
5.5.1. Representación de superficies deformables basada en NURBS180	180
5.6. Estudio del coste computacional del modelo BSpline4D	184
5.7. Conclusiones	185
6. Deformaciones simples basadas en restricciones	187
6.1. Deformaciones de formas libres	189
6.2. <i>Scodef - Simple Constrained Deformations</i>	191
6.2.1. Extensiones al modelo Scodef	194
6.3. <i>N-Scodef</i>	196
6.3.1. <i>N-Scodef</i> una extensión de <i>Scodef</i>	197
6.3.2. Interacción entre diferentes restricciones	203
6.3.3. Particularidades de la implementación sobre BSpline4D . .	208
6.3.4. Restricciones sobre trayectorias	215

Índice general

6.3.5. Resultados	234
6.4. Conclusiones	235
7. Conclusiones	239
Bibliografía	245
Índice alfabético	269

Índice de figuras

1.1. Componentes de una deformación	5
2.1. Ejemplo explicativo de las diferencias en el movimiento de objetos rígidos y deformables	8
3.1. La aceleración constante es la base de algunos métodos de integración	26
3.2. La aceleración para el método Newmark β	27
3.3. Comparación de resultados para diferentes métodos de integración.	31
3.4. Regiones de discretización del contorno para poder calcular los operadores en diferencias finitas.	39
3.5. Radio espectral para el método de las diferencias centradas con $\Delta t = 0.01s$	49
3.6. Logaritmo del número de condición para el método de las diferen- cias centradas con $\Delta t = 0.01s$	49
3.7. Radio espectral para el método de las diferencias centradas implíci- to con $\Delta t = 0.01s$	50
3.8. Logaritmo del número de condición para el método de las diferen- cias centradas implícito con $\Delta t = 0.01s$	50
3.9. Logaritmo del radio espectral para el método de Houbolt con $\Delta t =$ $0.01s$	51
3.10. Logaritmo del número de condición para el método de Houbolt con $\Delta t = 0.01s$	51
3.11. Logaritmo del radio espectral para el método de Wilson θ con $\Delta t = 0.01s$	52
3.12. Número de condición para el método de Wilson θ con $\Delta t = 0.01s$.	52

3.13. Logaritmo del radio espectral para el método de Newmark β con $\Delta t = 0.01s$	53
3.14. Número de condición para el método de Newmark β con $\Delta t = 0.01s$	53
3.15. Radio espectral para el método de las diferencias centradas con $\Delta t = 0.001s$	53
3.16. Logaritmo del número de condición para el método de las diferencias centradas con $\Delta t = 0.001s$	54
3.17. Radio espectral para el método de las diferencias centradas implícito con $\Delta t = 0.001s$	54
3.18. Logaritmo del número de condición para el método de las diferencias centradas implícito con $\Delta t = 0.001s$	55
3.19. Logaritmo del radio espectral para el método de Houbolt con $\Delta t = 0.001s$	55
3.20. Logaritmo del número de condición para el método de Houbolt con $\Delta t = 0.001s$	55
3.21. Logaritmo del radio espectral para el método de Wilson θ con $\Delta t = 0.001s$	56
3.22. Número de condición para el método de Wilson θ con $\Delta t = 0.001s$	56
3.23. Logaritmo del radio espectral para el método de Newmark β con $\Delta t = 0.001s$	57
3.24. Número de condición para el método de Newmark β con $\Delta t = 0.001s$	57
3.25. Comparativa de los logaritmos del radio espectral y el número de condición para el método de las diferencias centradas	59
3.26. Comparativa de los logaritmos del radio espectral y el número de condición para el método de las diferencias centradas implícito	59
3.27. Comparativa de los logaritmos del radio espectral y el número de condición para el método de Houbolt	60
3.28. Comparativa de los logaritmos del radio espectral y el número de condición para el método de Wilson θ	60
3.29. Comparativa de los logaritmos del radio espectral y el número de condición para el método de Newmark β	60
3.30. Radio espectral para experimentos con contorno fijo, con el método de diferencias centradas.	74

3.31. Radio espectral para experimentos con contorno fijo, con el método de las diferencias centradas implícito.	75
3.32. Radio espectral para experimentos con contorno fijo, con el método de Houbolt.	76
3.33. Radio espectral para experimentos con contorno fijo, con el método de Newmark β	77
3.34. Radio espectral para experimentos con contorno fijo, con el método de Wilson θ	78
3.35. Radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas, con el método de las diferencias centradas. . .	79
3.36. Radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas, con el método de las diferencias centradas implícito.	80
3.37. Radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas, con el método de Houbolt.	81
3.38. Radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas, con el método de Newmark.	82
3.39. Radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas, con el método de Wilson θ	83
3.40. Medida de la resistencia a la elongación	86
3.41. Medida de resistencia al doblamiento	87
3.42. Medida de resistencia al doblamiento	88
3.43. Un ejemplo de simulación, el pañuelo colgado.	93
3.44. Un ejemplo de simulación, la mesa del restaurante.	94
4.1. Una clasificación de la representación geométrica de los objetos deformables	100
4.2. Referencias anteriores a la informática de los modelos discretos . .	101
4.3. Mallado poligonal y su función de conectividad	102
4.4. Dualidad entre triangulación de Delaunay y diagrama de Voronoi .	104
4.5. Dualidad entre una triangulación y un mallado 2-simple	106
4.6. Diferentes configuraciones de masas y resortes	108
4.7. Descripción de la forma en que se obtiene un cilindro generalizado	111
4.8. Representación de un objeto explícito	113

4.9. Esfera definida como una superficie supercuádrica	115
4.10. Un superelipsoide como ejemplo de superficie supercuádrica	117
4.11. Isosuperficie definida a partir de tres esferas equidistantes	120
4.12. Ejemplo de <i>blobs</i>	122
4.13. Ejemplo de <i>level set</i>	123
4.14. Ejemplo de subdivisión de una superficie	128
4.15. Influencia de los polinomios de Bernstein	131
4.16. Ejemplo de curva como forma polinomial a trozos	133
4.17. Curvas B-Spline y polígonos de control	135
4.18. Ejemplo de una superficie NURBS	136
4.19. Ejemplo de un mallado T-mesh	140
5.1. Polinomios de Legendre	160
5.2. Variación del radio espectral en función de los puntos de cuadratura de Gauss-Legendre	162
5.3. Ejemplo de números de condición para diferentes combinaciones de parámetros	177
5.4. Evolución de una lámina BSpline4D sujeta por un lado	178
5.5. Medida de resistencia al doblamiento de una lámina BSpline4D	179
5.6. Comparación de los modelos BSpline4D y NURBS	183
5.7. Diferencias entre los modelos BSpline4D y NURBS	183
6.1. Deformación de objetos mediante primitivas simples	189
6.2. Modelo de proyecciones Scodef	192
6.3. Un ejemplo de aplicación de deformaciones Scodef	198
6.4. Elementos de una deformación basada en restricciones y su acción sobre una superficie NURBS	199
6.5. Funciones de deformación aplicables	200
6.6. Ejemplo de cómo la forma de la función de deformación afecta a la manera en que la deformación se produce	201
6.7. El efecto del radio de influencia en las deformaciones basadas en restricciones	201

6.8. Diferentes formas para determinar las zonas de influencia y sus respectivas distancias	202
6.9. Comportamiento de un Scodef con dos restricciones	206
6.10. Separación de restricciones no disjuntas mediante hiperplanos	208
6.11. Deformación de una esfera	209
6.12. Las mismas deformaciones de la figura 6.11 pero aplicadas a un círculo.	210
6.13. Ejemplos de deformación de una superficie mediante la aplicación de una restricción lineal	214
6.14. Elementos a considerar para realizar una deformación sujeta a una trayectoria.	217
6.15. Triedro de Frenet	217
6.16. Orientación de una curva en el espacio	218
6.17. Orientación de la curva de perfil	222
6.18. Ejemplos de superficies de barrido	223
6.19. Más ejemplos de superficies de barrido	223
6.20. Superficies de barrido extremas	224
6.21. Corrección para curvas de trayectoria con excesiva torsión o curvatura	225
6.22. Restricción aplicada a una superficie BSpline4D	226
6.23. Sucesivos refinamientos de los puntos de control de una superficie	227
6.24. Refinamiento del extremo de las curvas de trayectoria.	230
6.25. El problema de las tangentes	233
6.26. Poliedro de control de una superficie deformada	235
6.27. Algunos resultados	236
6.28. El problema de la deformación de la primera curva de perfil	237

Índice de tablas

3.1. Comparativa de los valores obtenidos según el método de integración	32
3.2. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, caso 1)	61
3.3. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, caso 1)	62
3.4. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, caso 2)	63
3.5. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, caso 2)	64
3.6. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, caso 3)	65
3.7. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, caso 3)	66
3.8. Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 0.06667.	67
3.9. Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 0.100.	68
3.10. Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 0.200.	69
3.11. Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 1.000.	70
3.12. Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 1.429.	71

3.13. Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 2.0.	72
3.14. Comparativa del coste computacional de los métodos de integración	91
3.15. Metodología para la validación de métodos de integración	96
4.1. Una clasificación de los modelos de representación de objetos . . .	142
5.1. Efecto de los puntos de cuadratura sobre el resultado de la simulación	161
5.2. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, caso 1)	170
5.3. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, caso 1)	171
5.4. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, caso 2)	172
5.5. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, caso 2)	173
5.6. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, caso 3)	174
5.7. Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, caso 3)	175
5.8. Combinaciones de grado y puntos de control	176

Introducción

La comunidad científica ha encontrado en los ordenadores una herramienta extremadamente versátil. Gracias a su capacidad de proceso, acrecentada día a día, es posible resolver problemas que de otra forma serían inabordables. De entre todas las posibles aplicaciones, son de especial interés las que intentan representar el comportamiento de la realidad. Los contextos en los que se hace uso de estas aplicaciones pueden ser tan variados como la creación de una película de dibujos animados, el estudio de la estructura de un edificio o el análisis de los efectos de un terremoto. El objetivo buscado es representar con la mayor fidelidad posible el mundo real sin tener que recurrir a modelos y maquetas muchas veces difíciles de elaborar y a menudo muy costosos. Por ello, la utilización de sistemas informáticos para realizar simulaciones es un mecanismo muy práctico, siempre y cuando el comportamiento de la simulación sea suficientemente semejante a la realidad simulada.

Los objetos reales son siempre susceptibles de cambiar de forma en función de algunas influencias externas. Si se desea representar la realidad será preciso disponer de medios para poder representar tal característica. Esto hace que la complejidad a la hora de describir los objetos aumente, ya que además de la forma, posición y demás características que definen la apariencia del objeto es necesario indicar también su rigidez. El comportamiento de un objeto flexible o deformable será, pues, más complejo que el de un objeto rígido.

Hasta hace poco más de una década las limitaciones en la capacidad de proceso dificultaban en gran medida la posibilidad de crear, manipular y animar objetos deformables de manera eficiente. En cambio, en la actualidad existen entornos adecuados que permiten aplicar los modelos habituales para la generación de objetos deformables. Ello se debe en gran medida al aumento de la capacidad de proceso de los ordenadores, pero también a los avances en el estudio del comportamiento de los objetos deformables y a la mejora en las técnicas de re-

presentación de objetos. El objetivo del presente trabajo es estudiar una nueva forma de simular mediante un sistema informático la deformación de objetos, o más concretamente, de superficies parametrizadas, de forma que el resultado sea realista.

Partiendo de las siguientes definiciones obtenidas del Diccionario de la Real Academia Española de la Lengua:

Dinámica Parte de la mecánica que trata de las leyes del movimiento en relación con las fuerzas que lo producen.

Mecánica Parte de la física que trata del equilibrio y del movimiento de los cuerpos sometidos a cualquier fuerza.

Simulación Acción de simular.

Simular Representar algo, fingiendo o imitando lo que no es.

A partir de estas definiciones se puede interpretar que la simulación dinámica consiste en simular el movimiento, la deformación y la interacción entre objetos. Los puntos principales a considerar son:

- Representar el movimiento de un objeto en función de las fuerzas externas. Es posible plantear la solución desde dos puntos de vista diferentes: la *Dinámica directa*, consistente en calcular el movimiento de un objeto a partir de los momentos o de las fuerzas que inciden sobre él; y la *Dinámica inversa* que intenta determinar dichos momentos y fuerzas conociendo la trayectoria y las deformaciones que sufre un objeto.
- Representar el efecto de las fuerzas externas sobre la forma *geométrica* de un objeto.
- Detectar las colisiones entre objetos.
- Calcular las fuerzas que se derivan de dichas colisiones.
- Calcular las fuerzas de fricción, estáticas y cinéticas, entre dos objetos.

Para cada uno de estos aspectos existe una representación que es la más conveniente para su resolución. Es preciso encontrar una representación que sea apropiada para resolver todos estos apartados al mismo tiempo, o, alternativamente, diseñar un sistema que realice la combinación de todos estos aspectos de una forma rápida.

Los simuladores dinámicos, en general son programas que realizan una gran cantidad de operaciones. Para que un simulador dinámico sea útil en la práctica es preciso resolver los siguientes aspectos:

- Optimizar la complejidad del sistema para reducir el tiempo de ejecución.

-
- Minimizar el error cometido de forma que el resultado sea lo más parecido a la realidad que sea posible.
 - Encontrar un medio para poder describir el objeto real mediante una representación utilizable por el simulador.

La clasificación de los diferentes desarrollos se puede llevar a cabo según cual sea el problema a tratar (movimiento, deformación, interacción, ...) o según el dominio de la aplicación. Las técnicas de base que permiten construir un simulador dinámico pueden ser divididas en tres grupos:

- Técnicas que permiten calcular tanto el movimiento como las deformaciones de un objeto. Tanto si dichos cálculos se realizan conjuntamente como si lo hacen de forma separada.
- Técnicas que permiten detectar la colisión entre dos o más objetos.
- Técnicas que permiten calcular la fuerza de una colisión.

Estos tres grupos describen aspectos diferentes de una misma realidad.

Como ya se ha comentado, en el presente trabajo se abordarán aspectos relacionados con la deformación de objetos, dejando de lado aspectos tales como las colisiones que se puedan producir como consecuencia de las deformaciones entre diferentes objetos o de un objeto consigo mismo.

Dependiendo de la aplicación, la apariencia de la deformación puede ser más o menos importante que la precisión *física* del modelo considerado. En base a este criterio se han clasificado habitualmente las técnicas de modelización de objetos deformables en tres grandes categorías. Dichas categorías aparecen descritas en el estudio realizado por Ng y Grimsdale, [NG96]. Aunque el ámbito de dicho estudio se ciñe a la modelización de telas, por su generalidad esta clasificación es perfectamente aplicable a cualquier otro tipo de objeto deformable. Las tres categorías son:

Técnicas geométricas No incluyen las propiedades físicas del objeto. Se centran en la apariencia de la deformación y están basadas en ecuaciones geométricas. En consecuencia requieren un alto nivel de participación por parte del usuario (véase [Wei86, AGT93]).

Estas técnicas no tienen en cuenta nada más que la forma de los objetos. El resultado final parecerá tanto la realidad como pericia se tenga a la hora de definir la animación. Usualmente el ámbito de utilización de estas técnicas cubre aspectos en donde la realidad física es compleja o bien no existe, por ejemplo:

1. Generación de transiciones, *morphing*. Esta técnica consiste en transformar un objeto en otro (véase [Dec96]). El cambio se realiza mediante una correspondencia entre dos objetos, el inicial y el final, de manera que es posible realizar cambios en la forma del objeto inicial hasta que éste toma la forma del objeto final.
2. Deformaciones de formas libres, *Free Form Deformation (FFD)*. Técnicas consistentes en deformar el espacio en donde se halla un objeto y deformar el objeto según los cambios de dicho espacio ([HHK92, RM93, LW94, FvdPT97, CF01]).

Técnicas físicas La deformación del objeto se calcula en base a las ecuaciones físicas que controlan su comportamiento. El objeto se define teniendo en cuenta sus propiedades físicas (masa, rigidez, densidad, ...). El comportamiento del objeto se obtiene como resultado de la actuación de fenómenos externos (fuerzas, campos gravitatorios, tensiones, colisiones, ...) y de la repercusión de dichos fenómenos en las tensiones internas del objeto estudiado.

Las técnicas que se pueden considerar como físicas se pueden dividir a su vez en dos grupos, las técnicas basadas en energías y las técnicas basadas en fuerzas. En el primer grupo la idea es determinar la energía total del sistema de manera que se pueda establecer cómo debe comportarse el objeto para obtener un estado de mínima energía ([Fey86, TPBF87, TQ94, PMG00]). En el segundo, las fuerzas aplicadas provocan una serie de tensiones y fatigas que hacen que el sistema cambie ([MMP97, MMP99, MMP00, GMM⁺01, MMP02, Mas03]). Este segundo grupo sigue la línea establecida por las teorías clásicas de la elasticidad.

Técnicas híbridas Ésta última categoría agrupa a las técnicas que combinan modelos geométricos y físicos. Mediante estas técnicas se intenta reducir el coste computacional requerido por las técnicas físicas utilizando aproximaciones geométricas (véase [Rud90, Tso91, CGD97, RC02]).

Ésta no es la única clasificación posible, algunos autores proponen solamente dos categorías [GM97], eliminándose la que agrupa a las técnicas híbridas.

En general, para obtener una simulación de la deformación que sufre un objeto será necesario un *modelo computacional* que se obtendrá a partir de un modelo matemático, pudiéndose contemplar en él las propiedades físicas que caracterizan al objeto; y un modelo gráfico que permita poder construir una representación del objeto y de su entorno. Además puede ser necesaria un interfaz gráfica como medio para poder configurar la simulación. También es posible la utilización de secuencias de imágenes reales con objeto de obtener información necesaria para definir el comportamiento del objeto. Dichas imágenes reales pueden ser utilizadas

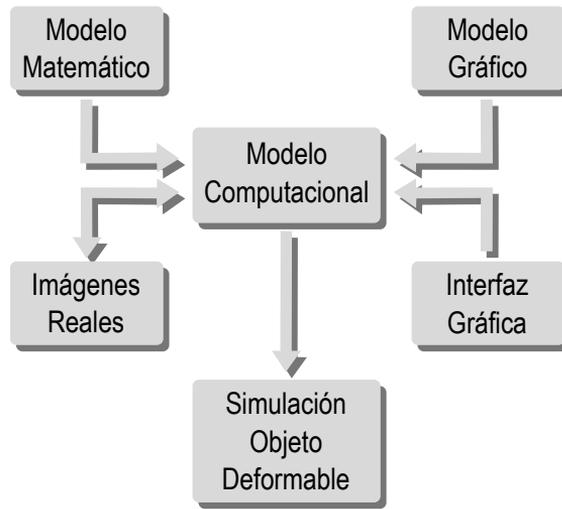


Figura 1.1: El modelo computacional se obtiene a partir de la información obtenida a partir de un modelo matemático. El objeto se representa en un modelo gráfico con el que se puede interactuar mediante una interfaz, usualmente gráfica. La información se puede complementar, o bien contrastar, mediante imágenes reales. El resultado obtenido del modelo computacional es la simulación del objeto deformable.

también para contrastar el comportamiento simulado de forma que se pueda determinar el grado de precisión alcanzado.

En el presente trabajo se aborda el estudio, diseño e implementación de modelos computacionales que permitan la representación y simulación de deformaciones de superficies parametrizadas. Para poder determinar la adecuación del modelo propuesto se han desarrollado una serie de programas informáticos que han permitido la creación de las imágenes con las que este documento está ilustrado. Además se ha utilizado un programa de manipulación algebraica para realizar comprobaciones numéricas relacionadas con los modelos matemáticos desarrollados.

Aparte del capítulo actual que sirve de introducción, el contenido del presente trabajo se organiza de la siguiente forma:

- En el capítulo 2 se abordan los fundamentos físicos necesarios para poder realizar la simulación de la deformación de objetos. El estudio describe los conceptos fundamentales de la mecánica clásica y de él se obtiene el modelo matemático que permitirá construir un modelo de deformaciones.
- Para poder tratar de forma adecuada y obtener los resultados esperados con un modelo de deformaciones, en el capítulo 3 se realiza un análisis

pormenorizado de los métodos de integración aplicables y se aplica dicho estudio al modelo de deformaciones planteado originalmente en [TPBF87]. El objetivo es determinar la adecuación tanto de los métodos de integración al modelo como del modelo mismo.

- La simulación de la deformación de objetos requiere representar los mismos como un modelo informático. En el capítulo 4 se examinan las posibles formas de representación de objetos deformables. El estudio de las ventajas e inconvenientes de cada una de ellas permitirá establecer el modelo de deformación que se utilizará en los siguientes capítulos.
- Los estudios realizados en los capítulos anteriores permitirán proceder, en el capítulo 5, a proponer el uso de un modelo alternativo de representación de superficies parametrizadas, buscando la facilidad de implementación, la sencillez de representación y de los cálculos; así como la versatilidad, que permitirá, adaptando las ecuaciones descritas en los capítulos 2 y 3, la simulación dinámica de la deformación de las mismas.
- La representación y el modelo propuestos en el capítulo anterior son válidos no solo para las técnicas descritas anteriormente, sino que son perfectamente aplicables a modelos geométricos. En el capítulo 6 se muestra cómo es posible aplicar el modelo para la obtención de deformaciones de formas libres, *FFD*.
- Finalmente, en el capítulo 7 se procede a realizar una revisión del trabajo presentado destacando las contribuciones realizadas y detallando posibles líneas de estudio que se pueden desarrollar a partir de los resultados obtenidos.

Estudio físico

El objetivo a conseguir en el presente trabajo es el de simular, mediante un sistema informático, el comportamiento de un objeto al cual se le aplican fuerzas de diversa índole de manera que dicho objeto cambie a lo largo del tiempo tanto de forma como de posición en el espacio. Interesa simular cómo se comportaría en la realidad el objeto simulado, de la manera más precisa posible.

El realismo puede ser conseguido de varias formas; por ejemplo, intentando mediante ensayos sucesivos obtener una apariencia realista de un objeto que aparentemente se parece al objeto real a base de ir ajustando una serie de parámetros. Sin embargo es posible decantarse por otro tipo de solución, posiblemente más compleja y difícil de atacar, pero también mucho más precisa en cuanto al resultado obtenido: utilizar un modelo físico para representar el objeto y realizar una simulación para determinar los cambios que se producen en el cuerpo a lo largo del tiempo.

Al tratarse del estudio del movimiento, y deformación, de objetos sujetos a la acción de las fuerzas que se les apliquen parece razonable buscar un modelo físico aplicable en el ámbito de la Mecánica clásica.

A continuación se describen los conceptos básicos relacionados con la Mecánica que permitirán establecer el modelo físico que regirá el movimiento y deformación de objetos tal y como se ha planteado, el desarrollo se basa en un estudio de los trabajos [Gol80] y [Sha88]([Sha13]).

2.1. Modelización según la mecánica clásica

El cuerpo, Ω , al que se aplicará la simulación puede ser considerado como un conjunto de m partículas situadas en el espacio:

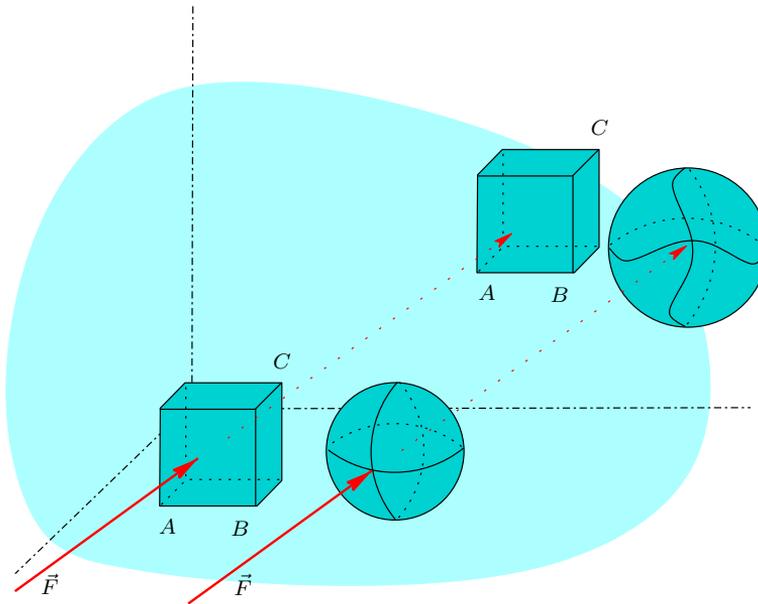


Figura 2.1: El movimiento en el espacio del cubo, rígido, como consecuencia de la aplicación de fuerzas no modifica las distancias \overline{AB} , \overline{AC} ni \overline{BC} . En cambio la esfera, deformable, sí que sufre cambios de forma al serle aplicadas fuerzas.

$$\Omega = \{ \mathbf{r}_i \in \mathbb{R}^3 \mid i = 1 \dots m \}.$$

Con esta consideración, si se desea determinar la posición del cuerpo será preciso determinar la posición de cada una de las partículas que lo forman, por lo tanto en un espacio \mathbb{R}^3 serían necesarias $3 \cdot m$ incógnitas, una para cada una de las componentes de las coordenadas de cada una de las partículas.

Si se considera al cuerpo como un sólido rígido, entonces la posición relativa de cada una de las partículas no cambiará a lo largo del tiempo. Por lo tanto una vez conocida la posición de una determinada partícula, para conocer la situación de cualquier otra será suficiente con determinar dos ángulos de rotación que indiquen, respecto de dos ejes arbitrarios, la ubicación de la segunda partícula. Una tercera partícula puede ser localizada mediante un tercer ángulo de rotación. Todas las demás partículas quedan ya localizadas.

Así pues, en el caso de un sólido rígido en realidad es suficiente con seis incógnitas para determinar su ubicación. Sin embargo, el hecho de tratarse de un sólido rígido, con una posición relativa de las partículas fija hace incompatible dicho modelo con el hecho de que se desea que el objeto a tratar pueda deformarse

durante la simulación.

En realidad al tratarse de un cuerpo, las diferentes partículas que lo forman no tienen un comportamiento independiente unas de otras, sino que tienen algún tipo de ligadura que las relaciona. Estas ligaduras pueden servir para definir la configuración del sistema con un número menor de incógnitas que las que se necesitarían en el caso de no considerarse dichas ligaduras. Dependiendo de la forma en que se manifiestan dichas ligaduras se pueden establecer dos grupos:

ligaduras holónomas: aquéllas que pueden ser expresadas en función de la posición de las partículas y, posiblemente, del tiempo.

$$f(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m, t) = 0 \quad (2.1)$$

ligaduras no holónomas: las que dependan de otros factores que no sean la posición de las partículas y, posiblemente, el tiempo.

En ambos casos puede hacer aparición una dependencia explícita del tiempo, en cuyo caso se habla de *ligaduras reónomas*, en caso contrario se trata de *ligaduras esclerónomas*.

Un ejemplo sencillo de ligadura holónoma es precisamente un sólido rígido, en el cual la posición de las partículas que lo forman permanece inalterable. Además, dicha ligadura no depende del tiempo, con lo que es también una ligadura esclerónoma. Una forma de representar una ligadura de este tipo es:

$$(\mathbf{r}_i - \mathbf{r}_j)^2 - C_{i,j}^2 = 0 \quad \forall i, j \in 1..n, i \neq j.$$

Las ligaduras establecen relaciones entre las diferentes partículas que forman el objeto, por lo tanto las diferentes ecuaciones que determinan el movimiento de cada partícula no son independientes. Si el sistema es tal que las ligaduras son holónomas, entonces es posible replantear el sistema de ecuaciones de manera que el comportamiento del mismo venga dado por un conjunto de parámetros diferente. Es decir, se expresa la posición de cada partícula en el espacio como una función de una serie de parámetros:

$$\mathbf{r}_i = \mathbf{r}_i(q_1, q_2, \dots, q_n, t) = \mathbf{r}_i(q_j, t) \quad \forall i = 1 \dots m, \quad j = 1 \dots n. \quad (2.2)$$

Los parámetros que sirven para determinar unívocamente la configuración del sistema se denominan *coordenadas generalizadas*. Dichos parámetros pueden

ser de cualquier magnitud, por ejemplo un radio y un ángulo pueden definir la posición de una partícula que se mueve en un plano alrededor de un punto fijo.

Se denomina *grado de libertad* a cada una de las incógnitas que aparecen en un sistema. En el caso de un sólido definido como un conjunto de m partículas habrá un total de $3 \cdot m$ grados de libertad, las coordenadas de cada partícula en \mathbb{R}^3 . Si existen ligaduras holónomas, que se expresen mediante k ecuaciones de la forma (2.1), entonces es posible utilizar dichas ecuaciones para reducir el número de grados de libertad a un total de $3 \cdot m - k$.

Un conjunto de coordenadas generalizadas se denomina libre si cada una puede variar de forma independiente de las demás. La ventaja de utilizar coordenadas generalizadas libres es que a partir de ellas se puede plantear un sistema de ecuaciones compatible determinado, es decir *resoluble*.

Además, la libertad que ofrece un planteamiento basado en coordenadas generalizadas permite escoger para cada caso la mejor manera de representar el objeto a simular de acuerdo con los criterios y restricciones que se puedan plantear.

Es usual denominar *configuración* a la solución de un sistema, por ello el espacio donde se definen dichas soluciones suele denominarse *espacio de configuraciones*. En el caso de utilizar coordenadas generalizadas, la configuración *instantánea*, es decir, para un instante de tiempo, será un punto en el espacio de configuraciones. Así como vaya transcurriendo el tiempo la configuración vendrá siendo siempre representada por un punto. Por lo tanto con el tiempo la configuración definirá una trayectoria. Es importante notar que cada uno de estos puntos representará la ubicación de todas las partículas del objeto. Por lo tanto el espacio real en donde se halla el objeto simulado es diferente del espacio de configuraciones, aunque están relacionados.

2.1.1. Principio de D'Alembert

Partiendo de la segunda ley de Newton, expresada para cada partícula, que postula que el movimiento viene descrito por la ecuación diferencial:

$$\mathbf{F}_i = \dot{\mathbf{p}}_i \quad \forall i = 1 \dots m. \quad (2.3)$$

Siendo \mathbf{F}_i la resultante de las fuerzas aplicadas sobre la partícula, y \mathbf{p}_i la cantidad de movimiento de dicha partícula; y por definición, la cantidad de movimiento de una partícula equivale al producto de su masa por su velocidad: $\mathbf{p}_i = m_i \cdot \mathbf{v}_i$. La ecuación (2.3) se puede escribir, para expresar un equilibrio dinámico, de la siguiente forma:

$$\mathbf{F}_i - \dot{\mathbf{p}}_i = 0 \quad \forall i = 1 \dots m,$$

si el sistema está en equilibrio, en un instante de tiempo infinitesimal (es decir, tan pequeño que la variación del tiempo se puede considerar despreciable) el desplazamiento también será infinitesimal, *virtual*, y el *trabajo virtual* realizado, δW , será nulo para el conjunto de las partículas:

$$\delta W = \sum_{i=1}^m (\mathbf{F}_i - \dot{\mathbf{p}}_i) \delta \mathbf{r}_i = 0$$

Debe tenerse en cuenta que un desplazamiento virtual de una partícula, $\delta \mathbf{r}_i$, representa una variación infinitesimal de las coordenadas que es compatible con las fuerzas y ligaduras que estén presentes en el sistema en un instante dado, t . Dicho desplazamiento no puede tomar en consideración el tiempo debido a que, a lo largo de éste, las fuerzas aplicadas y las ligaduras pueden variar, lo que contradice el planteamiento.

Considerando que las fuerzas que se manifiestan se dividen en dos grupos, las fuerzas externas, \mathbf{F}_i^e y las fuerzas de ligadura, \mathbf{F}_i^c :

$$\mathbf{F}_i = \mathbf{F}_i^e + \mathbf{F}_i^c,$$

se puede expresar δW como

$$\begin{aligned} \delta W &= \sum_{i=1}^m (\mathbf{F}_i - \dot{\mathbf{p}}_i) \cdot \delta \mathbf{r}_i \\ &= \sum_{i=1}^m (\mathbf{F}_i^e + \mathbf{F}_i^c - \dot{\mathbf{p}}_i) \cdot \delta \mathbf{r}_i \\ &= \sum_{i=1}^m (\mathbf{F}_i^e - \dot{\mathbf{p}}_i) \cdot \delta \mathbf{r}_i + \sum_{i=1}^m \mathbf{F}_i^c \cdot \delta \mathbf{r}_i = 0. \end{aligned}$$

Ahora bien, las fuerzas de ligadura no realizan ningún trabajo debido a que o bien aparecen de manera que la acción de algunas de ellas compensa la de otras, o bien se manifiestan de manera que su acción es perpendicular al desplazamiento virtual, y por lo tanto su producto será nulo.

$$\sum_{i=1}^m \mathbf{F}_i^c \cdot \delta \mathbf{r}_i = 0,$$

como consecuencia el trabajo virtual se puede expresar como

$$\delta W = \sum_{i=1}^m (\mathbf{F}_i^e - \dot{\mathbf{p}}_i) \cdot \delta \mathbf{r}_i = 0 \quad (2.4)$$

que es el conocido Principio de D'Alembert.

2.1.2. Ecuaciones de Lagrange

Sería interesante poder plantear el sistema de ecuaciones de (2.4) de manera que las ecuaciones sean independientes unas de otras, al mismo tiempo también sería deseable poder reducir el número de ecuaciones. Si se trata de un sistema holónimo, el desplazamiento infinitesimal de la coordenada \mathbf{r}_i , $\delta \mathbf{r}_i$, puede expresarse fácilmente en función de las coordenadas generalizadas. Así de (2.2) se obtiene:

$$\delta \mathbf{r}_i = \frac{\partial \mathbf{r}_i}{\partial q_1} \delta q_1 + \frac{\partial \mathbf{r}_i}{\partial q_2} \delta q_2 + \cdots + \frac{\partial \mathbf{r}_i}{\partial q_n} \delta q_n = \sum_{j=1}^n \frac{\partial \mathbf{r}_i}{\partial q_j} \delta q_j \quad \forall i \in 1 \dots m.$$

Evidentemente este desplazamiento virtual expresado en función de las coordenadas generalizadas no contempla el tiempo, es decir $\delta t = 0$.

Si se plantea el Principio de D'Alembert utilizando la expresión anterior,

$$\sum_{i=1}^m (\mathbf{F}_i^e - \dot{\mathbf{p}}_i) \cdot \sum_{j=1}^n \frac{\partial \mathbf{r}_i}{\partial q_j} \delta q_j = \sum_{j=1}^n \sum_{i=1}^m (\mathbf{F}_i^e - \dot{\mathbf{p}}_i) \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \delta q_j = 0$$

que, convenientemente arreglada, queda

$$\sum_{j=1}^n \left(\sum_{i=1}^m \mathbf{F}_i^e \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} - \sum_{i=1}^m \dot{\mathbf{p}}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \right) \delta q_j = 0, \quad (2.5)$$

entonces se ponen de manifiesto las *fuerzas generalizadas*, es decir, las fuerzas externas aplicadas en la base de las coordenadas generalizadas:

$$Q_j = \sum_{i=1}^m \mathbf{F}_i^e \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \quad \forall j = 1 \dots n. \quad (2.6)$$

Cabe remarcar que el concepto de *fuerzas generalizadas* no tiene por qué coincidir con el concepto habitual de fuerzas, ya que son expresiones que están relacionadas con las coordenadas generalizadas. Solamente en el caso en que las coordenadas generalizadas correspondan a coordenadas espaciales entonces sí que serán identificables como fuerzas. En todo caso, lo que sí es cierto siempre es que el producto de las *fuerzas generalizadas* con los desplazamientos virtuales de las coordenadas generalizadas debe dar como resultado un trabajo.

Para expresar la ecuación (2.5) de forma conveniente, se pueden tener en cuenta las siguientes consideraciones. Por una parte:

$$\dot{\mathbf{r}}_i = \sum_{j=1}^n \frac{\partial \mathbf{r}_i}{\partial q_j} \frac{\partial q_j}{\partial t} + \frac{\partial \mathbf{r}_i}{\partial t} = \sum_{j=1}^n \frac{\partial \mathbf{r}_i}{\partial q_j} \dot{q}_j + \frac{\partial \mathbf{r}_i}{\partial t},$$

de donde

$$\frac{\partial \dot{\mathbf{r}}_i}{\partial \dot{q}_j} = \frac{\partial}{\partial \dot{q}_j} \left(\sum_{k=1}^n \frac{\partial \mathbf{r}_i}{\partial q_k} \dot{q}_k + \frac{\partial \mathbf{r}_i}{\partial t} \right) = \sum_{k=1}^n \frac{\partial}{\partial \dot{q}_j} \left(\frac{\partial \mathbf{r}_i}{\partial q_k} \dot{q}_k \right) + \frac{\partial}{\partial \dot{q}_j} \left(\frac{\partial \mathbf{r}_i}{\partial t} \right) = \frac{\partial \mathbf{r}_i}{\partial q_j},$$

y considerando también que

$$\frac{d}{dt} \left(\frac{\partial \mathbf{r}_i}{\partial q_j} \right) = \frac{\partial}{\partial q_j} \left(\frac{d\mathbf{r}_i}{dt} \right) = \frac{\partial}{\partial q_j} \left(\sum_{k=1}^n \frac{\partial \mathbf{r}_i}{\partial q_k} \dot{q}_k + \frac{\partial \mathbf{r}_i}{\partial t} \right) = \sum_{k=1}^n \frac{\partial^2 \mathbf{r}_i}{\partial q_j \partial q_k} \dot{q}_k + \frac{\partial^2 \mathbf{r}_i}{\partial q_j \partial t} = \frac{\partial \dot{\mathbf{r}}_i}{\partial \dot{q}_j}$$

entonces, considerando que la energía cinética es la energía que posee un cuerpo debido a su movimiento y que representa el trabajo necesario para poder acelerar dicho cuerpo hasta que éste alcanza la velocidad que posee, y que se puede expresar dicha energía mediante la expresión:

$$T = \sum_{i=1}^m \frac{1}{2} m_i \dot{\mathbf{r}}_i^2, \quad (2.7)$$

se puede expresar el trabajo realizado por la variación en el tiempo de la cantidad de movimiento que aparece en (2.5) como

$$\begin{aligned} \sum_{i=1}^m \dot{\mathbf{p}}_i \frac{\partial \mathbf{r}_i}{\partial q_j} &= \sum_{i=1}^m m_i \ddot{\mathbf{r}}_i \frac{\partial \mathbf{r}_i}{\partial q_j} \\ &= \frac{d}{dt} \left(\sum_{i=1}^m m_i \dot{\mathbf{r}}_i \frac{\partial \mathbf{r}_i}{\partial \dot{q}_j} \right) - \sum_{i=1}^m m_i \dot{\mathbf{r}}_i \frac{\partial \dot{\mathbf{r}}_i}{\partial q_j} \\ &= \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j}, \end{aligned}$$

con lo que (2.5) se convierte en

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = Q_j. \quad (2.8)$$

A dichas ecuaciones se las denomina *ecuaciones de Lagrange*. Si las fuerzas provienen de un campo escalar: $F_i^e = -\nabla_i V$, las fuerzas generalizadas se pueden expresar de la forma:

$$Q_j = \sum_{i=1}^m F_i^e \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} = - \sum_{i=1}^m \nabla_i V \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} = - \frac{\partial V}{\partial q_j},$$

y la ecuación (2.8) puede escribirse como

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial(T - V)}{\partial q_j} = 0. \quad (2.9)$$

Dicha ecuación no impone que el sistema sea conservativo¹. Ahora bien, si V no depende de las velocidades generalizadas, \dot{q}_j , se podrá introducir V en el primer término de la expresión, y por lo tanto se puede reescribir (2.9) como

$$\frac{d}{dt} \left(\frac{\partial(T - V)}{\partial \dot{q}_j} \right) - \frac{\partial(T - V)}{\partial q_j} = 0$$

Si se define la *función lagrangiana*, $L(q, \dot{q}, t)$, como:

$$L = T - V. \quad (2.10)$$

¹Cuando el campo de fuerzas es tal que el trabajo requerido para trasladarse de un punto a otro es el mismo para todo camino posible que una los dos puntos, se dice que el sistema es conservativo

Entonces la ecuación (2.8) se puede escribir como:

$$L(q, \dot{q}, t) = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0. \quad (2.11)$$

Aún cuando las fuerzas externas no procedan de un potencial, siempre que se puedan expresar las fuerzas generalizadas en función de una función de *potencial generalizado*, $U(q_j, \dot{q}_j)$, de manera que

$$Q_j = -\frac{\partial U}{\partial q_j} + \frac{d}{dt} \left(\frac{\partial U}{\partial \dot{q}_j} \right),$$

se podrá obtener la ecuación (2.11) a partir de (2.8), cambiando (2.10) por

$$L = T - U.$$

2.1.3. El principio de Hamilton

Tanto el Principio de D'Alembert como la ecuaciones de Lagrange obtenidas parten de un planteamiento *diferencial* es decir, considerando estados instantáneos y desplazamientos virtuales. Es posible obtener los mismos resultados partiendo de un enfoque diferente, considerando el movimiento del sistema entre dos instantes de tiempo dados, es lo que se denomina un planteamiento *integral*.

Como se ha comentado anteriormente, en el espacio de configuraciones se define la trayectoria formada por el movimiento en dicho espacio de la configuración resultante para cada instante de tiempo. El tiempo se convierte entonces en un parámetro de dicha curva. Si en una situación tal como la descrita todas las fuerzas derivan de un potencial escalar, generalizado, que depende solamente de las coordenadas de posición, entonces el sistema será conservativo.

El *principio de Hamilton* postula que la trayectoria del sistema entre dos instantes de tiempo, t_1 y t_2 , es tal que la integral

$$I = \int_{t_1}^{t_2} L dt, \quad (2.12)$$

siendo $L = T - V$, tiene un valor estacionario para el camino del movimiento correcto. A I se le suele llamar *integral de acción*. Una forma equivalente de decir lo mismo es que el movimiento es tal que la variación de I para el intervalo de tiempo indicado es nula:

$$\delta I = \delta \int_{t_1}^{t_2} L(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n, t) dt = 0. \quad (2.13)$$

La importancia del principio de Hamilton reside en que si el sistema es holónomo entonces (2.13) es una condición necesaria y suficiente para las ecuaciones de Lagrange (2.9). Además, es posible deducir el principio de Hamilton a partir de las ecuaciones de Lagrange, por lo tanto es posible construir toda la Mecánica basándose en el principio de Hamilton en lugar de hacerlo en las leyes de Newton del movimiento. Hacerlo de esta forma conlleva dos ventajas:

- La integral de acción (2.12) es invariante al sistema de coordenadas generalizadas utilizadas para expresar L , (2.9). Las ecuaciones del movimiento deben tener siempre la forma de Lagrange independientemente de como se transformen las coordenadas generalizadas.
- La formulación en función de un principio variacional permite describir sistemas aparentemente no mecánicos con las formas matemáticas de la mecánica clásica.

Por lo tanto utilizar el principio de Hamilton en lugar de utilizar directamente las leyes de Newton o las ecuaciones de Lagrange permitirá poder obtener modelos más sencillos con los que será más fácil trabajar. Es bastante usual encontrar trabajos teóricos relacionados con la Mecánica Clásica que se basan en el principio de Hamilton, de la misma forma que también es usual encontrar expresiones en la forma de Lagrange para el desarrollo de modelos concretos.

2.1.4. Sistemas no conservativos

Si las fuerzas que se aplican al sistema no proceden de un potencial, las ecuaciones de Lagrange se pueden escribir como

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j. \quad (2.14)$$

Donde L contiene el potencial de todas las fuerzas conservativas del sistema, mientras que Q_j representa las fuerzas que no provienen de un potencial.

Este planteamiento permite contemplar situaciones en las que haya pérdidas o disipaciones de energía, por ejemplo debido a un rozamiento,

$$F_{\mathbf{r}_i}^r = -k(\mathbf{r}_i) \cdot \dot{\mathbf{r}}_i.$$

Es bastante usual que dichas disipaciones sean proporcionales a la velocidad de las partículas. Las fuerzas que se ajustan a dicho modelo se pueden deducir a partir de la *función de disipación de Rayleigh*:

$$F^R = \frac{1}{2} \sum_{i=1}^m \left(k(\mathbf{r}_i) \cdot \dot{\mathbf{r}}_i^2 \right). \quad (2.15)$$

Con lo que

$$F^r = -\nabla_{\dot{\mathbf{r}}_i} \cdot F^R,$$

de lo que se deduce que el trabajo realizado por el sistema para contrarrestar las fuerzas de rozamiento puede expresarse como

$$dW_{\mathbf{r}_i}^R = -F_{\mathbf{r}_i}^R \cdot d\mathbf{r}_i = -F_{\mathbf{r}_i}^R \cdot \dot{\mathbf{r}}_i dt = (k(\mathbf{r}_i) \cdot \dot{\mathbf{r}}_i^2) ddt.$$

La fuerza generalizada correspondiente a la fuerza de disipación es:

$$Q_j = \sum_{i=1}^m F_{\mathbf{r}_i}^r \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} = - \sum_{i=1}^m \nabla_{\dot{\mathbf{r}}_i} F^R \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} = - \sum_{i=1}^m \nabla_{\dot{\mathbf{r}}_i} F^R \cdot \frac{\partial \dot{\mathbf{r}}_i}{\partial q_j} = - \frac{\partial F^R}{\partial \dot{q}_j}.$$

Con lo que las ecuaciones de Lagrange quedan de la siguiente forma:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} + \frac{\partial F}{\partial \dot{q}_j} = 0. \quad (2.16)$$

Aplicando el mismo planteamiento pero a partir de la ecuación (2.8), se obtiene

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} + \frac{\partial F}{\partial \dot{q}_j} + \frac{\delta U}{\delta q_j} = Q_j. \quad (2.17)$$

De esta forma, si las coordenadas generalizadas se escogen de manera que formen un conjunto libre, se conseguirá el objetivo de obtener las ecuaciones que describen el movimiento de un cuerpo deformable sujeto a fuerzas de todo tipo de manera que el sistema de ecuaciones resultante será compatible determinado.

En particular, si se toma en consideración como coordenadas generalizadas que representan al objeto a tratar al conjunto de coordenadas, $p_i \in \mathbb{R}^n \forall i =$

$1, \dots, m$, de cada una de las m partículas que lo forman, entonces se puede expresar la energía cinética de forma parecida a la expresada en (2.7)

$$T = \sum_{i=1}^m \frac{1}{2} m_i \dot{p}_i^2, \quad (2.18)$$

asumiendo que cada partícula pueda tener una masa diferente.

A partir de (2.18) se pueden desarrollar los términos de (2.17) uno a uno:

- Primer término, $\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{p}_i} \right)$:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{p}_i} \right) &= \frac{d}{dt} \left(\frac{\partial}{\partial \dot{p}_i} \left(\sum_{i=1}^m \frac{1}{2} m_i \dot{p}_i^2 \right) \right) = \frac{d}{dt} \left(\frac{1}{2} \sum_{i=1}^m \left(\frac{\partial m_i}{\partial \dot{p}_i} \dot{p}_i^2 + m_i \frac{\partial \dot{p}_i^2}{\partial \dot{p}_i} \right) \right) \\ &= \frac{d}{dt} \frac{1}{2} \begin{pmatrix} 0 \\ \vdots \\ m_i \dot{p}_i \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \frac{d}{dt} (m_i \dot{p}_i) \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \frac{d m_i}{dt} \dot{p}_i + m_i \frac{d \dot{p}_i}{dt} \\ \vdots \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \vdots \\ m_i \ddot{p}_i \\ \vdots \\ 0 \end{pmatrix}, \end{aligned}$$

asumiendo que la masa de las partículas no varía. El resultado, expresado en forma matricial, para todo i es:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{p}_i} \right) = \mathbf{M} \ddot{\mathbf{p}}, \quad (2.19)$$

siendo \mathbf{M} una matriz diagonal que representa la masa asociada a cada punto y $\ddot{\mathbf{p}}$ el vector que representa la aceleración de cada punto.

- Segundo término, $\frac{\partial T}{\partial p_i}$:

$$\begin{aligned} \frac{\partial T}{\partial p_i} &= \frac{\partial}{\partial p_i} \left(\sum_{j=1}^m \frac{1}{2} m_j \dot{p}_j^2 \right) = \frac{1}{2} \sum_{j=1}^m \left(\frac{\partial m_j}{\partial p_i} \dot{p}_j^2 + m_j \frac{\partial \dot{p}_j^2}{\partial p_i} \right) \\ &= \frac{1}{2} \begin{pmatrix} 0 \\ \vdots \\ m_i \frac{\partial \dot{p}_i}{\partial p_i} \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ m_i \frac{\partial}{\partial p_i} \frac{d}{dt} p_i \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ m_i \frac{d}{dt} \frac{\partial p_i}{\partial p_i} \\ \vdots \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \vdots \\ m_i \frac{d}{dt} 1 \\ \vdots \\ 0 \end{pmatrix} = 0, \end{aligned}$$

es decir, el segundo término se anula.

- Tercer término, $\frac{\partial F}{\partial \dot{p}_i}$. Aplicando la función de disipación de Rayleigh, (2.15):

$$\begin{aligned} \frac{\partial F}{\partial \dot{p}_i} &= \frac{\partial}{\partial \dot{p}_i} \frac{1}{2} \sum_{j=1}^m (k(p_j) \dot{p}_j^2) = \frac{1}{2} \sum_{j=1}^m \left(\frac{\partial k(p_j)}{\partial \dot{p}_j} \dot{p}_j^2 + k(p_j) \frac{\partial \dot{p}_j^2}{\partial \dot{p}_i} \right) \\ &= \frac{1}{2} \begin{pmatrix} 0 \\ \vdots \\ k(\mathbf{p}_i) \dot{p}_i \\ \vdots \\ 0 \end{pmatrix}. \end{aligned}$$

Considerando $k(\mathbf{p}_i)$ constante, el resultado obtenido puede expresarse de forma matricial, como en el caso del primer término:

$$\frac{\partial F}{\partial \dot{p}_i} = \mathbf{D} \dot{\mathbf{p}}. \quad (2.20)$$

- Cuarto término, $\frac{\delta U}{\delta q_j}$: En este caso, a falta de una descripción concreta del funcional de energía que vaya a ser utilizado, no se hace ningún desarrollo, simplemente se expone que es de esperar que el funcional que en cada caso se utilice sea tal que la expresión pueda ser convenientemente desarrollada

para obtener una expresión compatible con las ya obtenidas para el primer y tercer término. Es decir:

$$\frac{\delta U}{\delta q_j} \approx \mathbf{Kp}. \quad (2.21)$$

De esta forma la expresión (2.17) puede expresarse como

$$\mathbf{M}\ddot{\mathbf{p}} + \mathbf{D}\dot{\mathbf{p}} + \mathbf{Kp} = \mathbf{f}_p \quad (2.22)$$

siempre y cuando las coordenadas generalizadas que se utilicen para representar el objeto sean como las descritas. Además, fijada las condiciones iniciales y si las matrices M , D y K están formadas por coeficientes con el grado de suavidad necesario se puede garantizar que existe una solución única, al estar trabajando en un dominio compacto.

2.2. Conclusiones

En este capítulo se ha obtenido la ecuación diferencial (2.17) que describe el movimiento y deformación de un objeto sujeto a un conjunto de fuerzas externas. A partir de ella, se ha desarrollado la expresión (2.22), que representa un sistema lineal de ecuaciones diferenciales de segundo orden. Este sistema es resoluble mediante las técnicas tradicionales de resolución de ecuaciones diferenciales. Obviamente, el término de la energía interna no desarrollado desempeñará un papel fundamental en la forma en que se comportará el objeto.

Por lo tanto, si es posible asumir que las matrices M y D no dependen del tiempo, y disponiendo de una representación del objeto deformable compatible con la descripción utilizada, mediante (2.22) se tiene la posibilidad de calcular su deformación a lo largo del tiempo mediante técnicas ya conocidas. Se trata ahora de estudiar y determinar cual es la mejor forma de abordar la solución del sistema de ecuaciones que representa (2.22), así como la forma más conveniente para representar los objetos deformables.

Solución de la ecuación de equilibrio

Cuando la solución de las ecuaciones diferenciales que describen el movimiento y deformación de un sistema dinámico, como el representado por (2.22), no puede ser obtenida de forma analítica es preciso utilizar un método numérico para poder obtener los resultados deseados. Tal es el caso de la ecuación (2.17), que solamente es resoluble de forma analítica para algunos casos simplificados. Existe una gran variedad de métodos de integración que permiten aproximar la solución, todos ellos comparten dos características:

- no satisfacen las ecuaciones para un valor t continuo, sino que lo hacen en incrementos discretos de tiempo, Δt , y
- en cada incremento Δt debe asumirse un tipo de variación del desplazamiento, \mathbf{p} , la velocidad, $\dot{\mathbf{p}}$ y la aceleración $\ddot{\mathbf{p}}$.

Así pues, es posible disponer de diferentes esquemas de integración numérica en función del tipo de variación asumida para \mathbf{p} , $\dot{\mathbf{p}}$ y $\ddot{\mathbf{p}}$ en cada intervalo de tiempo Δt .

De entre todas las posibles técnicas interesan aquéllas que ofrezcan una mayor eficacia [Bat82, GdJB94, Bat96, Cha04a, Cha04b, Cha09, SBS96].

3.1. Métodos basados en la integración directa

Los métodos basados en la integración directa se distinguen por utilizar un proceso de integración numérica paso a paso sin que se realice ninguna transformación a la ecuación para expresarla de otra forma.

Estos métodos buscan resolver la ecuación (2.22) no en cada instante de tiempo, t , sino en unos valores discretos de tiempo, $t_i = t_0 + i \cdot \Delta t$, $i = 1, \dots$

Por otra parte, en estos métodos se asume que en cada instante de tiempo considerado, t_i , existe una variación en los desplazamientos, las velocidades y las aceleraciones. Esta variación será la que establecerá la precisión así como la estabilidad y el coste del método.

En general se considera que debe calcularse la solución de la ecuación (2.22) entre los instantes de tiempo 0 y T , siendo T el tiempo total dividido en n intervalos de tiempo iguales entre sí, es decir $\Delta t = \frac{T}{n}$. Los métodos obtienen la solución para el siguiente intervalo de tiempo a partir de los datos previamente obtenidos para los intervalos anteriores. Es decir, se intenta determinar la solución para el intervalo de tiempo $t + \Delta t$ una vez conocida la solución para los instantes $0, \Delta t, 2\Delta t, 3\Delta t, \dots, t - \Delta t, t$, en un esquema claramente iterativo. Si el objetivo es la simulación de un proceso de deformación a lo largo del tiempo, este tipo de planteamiento es idóneo si es posible que los intervalos de tiempo, Δt , puedan ser ajustados para poder ofrecer una cierta continuidad temporal a la representación.

En esta categoría entran a formar parte todos aquellos métodos que contemplan a la ecuación (2.22) como una ecuación diferencial ordinaria con coeficientes constantes. Por lo tanto puede ser utilizada cualquier aproximación mediante diferencias finitas de las velocidades y aceleraciones. Por supuesto, se trata de determinar cuales de estas aproximaciones son las más adecuadas.

3.1.1. El método de las diferencias centradas

Este método consiste en considerar que la velocidad y la aceleración se expresan en términos de \mathbf{p} y del tiempo t como:

$$\begin{aligned}\ddot{\mathbf{p}}^t &\approx \frac{\mathbf{p}^{t+\Delta t} - 2\mathbf{p}^t + \mathbf{p}^{t-\Delta t}}{\Delta t^2}, \\ \dot{\mathbf{p}}^t &\approx \frac{\mathbf{p}^{t+\Delta t} - \mathbf{p}^{t-\Delta t}}{2\Delta t},\end{aligned}\tag{3.1}$$

siendo el error numérico del orden de Δt^2 .

Por lo tanto, la ecuación (2.22) para el instante $t + \Delta t$ se expresa a partir de la solución en el instante t :

$$\mathbf{M}\ddot{\mathbf{p}}^t + \mathbf{C}\dot{\mathbf{p}}^t + \mathbf{K}\mathbf{p}^t = \mathbf{f}_p^t,\tag{3.2}$$

y sustituyendo $\ddot{\mathbf{p}}^t$ y $\dot{\mathbf{p}}^t$ por sus valores en (3.1) se obtiene

$$\mathbf{M} \frac{\mathbf{p}^{t+\Delta t} - 2\mathbf{p}^t + \mathbf{p}^{t-\Delta t}}{\Delta t^2} + \mathbf{C} \frac{\mathbf{p}^{t+\Delta t} - \mathbf{p}^{t-\Delta t}}{2\Delta t} + \mathbf{K}\mathbf{p}^t = f_p^t$$

que, convenientemente arreglada, se puede expresar como

$$\left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t} \right) \mathbf{p}^{t+\Delta t} = f_p^t + \left(\frac{2\mathbf{M}}{\Delta t^2} - \mathbf{K} \right) \mathbf{p}^t + \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2} \right) \mathbf{p}^{t-\Delta t}. \quad (3.3)$$

Esta forma de plantear la solución al sistema se basa en determinar las condiciones de equilibrio en el instante de tiempo t . Los métodos que siguen este planteamiento se denominan *métodos de integración explícitos*. En este tipo de métodos, la matriz de rigidez, \mathbf{K} , no necesita ser factorizada ya que no forma parte de la matriz del sistema. Si se considera que la matriz de masas, \mathbf{M} , y de amortiguación, \mathbf{C} , no varían con el tiempo entonces esto significa que la matriz del sistema no cambiará y por lo tanto debe ser calculada una única vez para todo el proceso de simulación.

El principal inconveniente que aparece en el uso del método de las diferencias centradas es que el incremento de tiempo considerado, Δt , debe ser inferior a un cierto intervalo crítico, Δt_{cr} . En la práctica esta limitación hace que el número de iteraciones necesario para conseguir simular el comportamiento del objeto a tratar pueda llegar a ser excesivamente grande. Los esquemas de integración que requieren incrementos de tiempo menores que un cierto umbral se denominan *condicionalmente estables* ya que, para valores de Δt mayores que Δt_{cr} la integración es inestable obteniéndose generalmente un error excesivo en el resultado.

3.1.2. El método de las diferencias centradas implícito

Puede plantearse una variación del método anterior consistente en tomar en consideración las condiciones de equilibrio en el instante de tiempo $t + \Delta t$. Por lo tanto se plantea un *método de integración implícito* en el cual \mathbf{K} , tomada en tiempo t , debe ser factorizada ya que pasa a formar parte de la matriz del sistema al considerar que se aplica en el instante $t + \Delta t$. La expresión de (3.2) siguiendo un esquema basado en un método de integración implícito sería

$$\mathbf{M} \frac{\mathbf{p}^{t+\Delta t} - 2\mathbf{p}^t + \mathbf{p}^{t-\Delta t}}{\Delta t^2} + \mathbf{C} \frac{\mathbf{p}^{t+\Delta t} - \mathbf{p}^{t-\Delta t}}{2\Delta t} + \mathbf{K}\mathbf{p}^{t+\Delta t} = f_p^t,$$

es decir

$$\left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t} + \mathbf{K} \right) \mathbf{p}^{t+\Delta t} = f_p^t + \left(\frac{2\mathbf{M}}{\Delta t^2} \right) \mathbf{p}^t + \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2} \right) \mathbf{p}^{t-\Delta t}. \quad (3.4)$$

Para calcular la solución en el instante $t + \Delta t$ el método de las diferencias centradas implícito necesita las soluciones para los instantes t y $t - \Delta t$. Por ello en el inicio es preciso conocer \mathbf{p}^0 y $\mathbf{p}^{-\Delta t}$. Si se conoce la posición y la velocidad del objeto en el instante 0 (\mathbf{p}^0 y $\dot{\mathbf{p}}^0$ respectivamente) se puede inferir su aceleración, $\ddot{\mathbf{p}}^0$, directamente de la ecuación (2.22), y la posición en el instante $-\Delta t$ (es decir $\mathbf{p}^{-\Delta t}$) se puede obtener aplicando Taylor:

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{p}}^0 &= \mathbf{f}_p^0 - \mathbf{D}\dot{\mathbf{p}}^0 - \mathbf{K}\mathbf{p}^0, \\ \mathbf{p}^{-\Delta t} &= \mathbf{p}^0 - \Delta t\dot{\mathbf{p}}^0 + \frac{\Delta t^2}{2}\ddot{\mathbf{p}}^0.\end{aligned}$$

Este planteamiento representa suponer que el objeto a deformar ya se encuentra en movimiento en el instante inicial. Otra posibilidad es la de calcular los valores \mathbf{p}^0 y \mathbf{p}^1 así como $\ddot{\mathbf{p}}^1$ y $\dot{\mathbf{p}}^1$ con algún otro método y con estos valores continuar los cálculos. Una tercera opción consiste en considerar que el objeto está en reposo en el instante $t = 0$ y por lo tanto $\mathbf{p}^{-\Delta t} = \mathbf{p}^0$, $\ddot{\mathbf{p}} = 0$ y $\dot{\mathbf{p}} = 0$.

3.1.3. El método de Houbolt

El método de integración de Houbolt es muy parecido en su planteamiento al método de las diferencias centradas. En este caso la expresión de la velocidad y la aceleración en función de la posición y el tiempo se expresan de la siguiente forma

$$\begin{aligned}\ddot{\mathbf{p}}^{t+\Delta t} &= \frac{2\mathbf{p}^{t+\Delta t} - 5\mathbf{p}^t + 4\mathbf{p}^{t-\Delta t} - \mathbf{p}^{t-2\Delta t}}{\Delta t^2}, \\ \dot{\mathbf{p}}^{t+\Delta t} &= \frac{11\mathbf{p}^{t+\Delta t} - 18\mathbf{p}^t + 9\mathbf{p}^{t-\Delta t} - 2\mathbf{p}^{t-2\Delta t}}{6\Delta t}\end{aligned}\tag{3.5}$$

que, como en el caso del método de las diferencias centradas, tiene un error numérico del orden de Δt^2 . Cabe destacar el detalle que las expresiones de $\dot{\mathbf{p}}$ y $\ddot{\mathbf{p}}$ se evalúan en el instante de tiempo $t + \Delta t$. Por lo tanto el método de Houbolt es un método implícito. Así pues la solución para el instante $t + \Delta t$ se obtiene a partir de la ecuación (2.22) expresada en el instante $t + \Delta t$:

$$\mathbf{M}\ddot{\mathbf{p}}^{t+\Delta t} + \mathbf{C}\dot{\mathbf{p}}^{t+\Delta t} + \mathbf{K}\mathbf{p}^{t+\Delta t} = \mathbf{f}_p^{t+\Delta t}.\tag{3.6}$$

Sustituyendo (3.5) en la expresión anterior

$$\begin{aligned} \mathbf{M} \left(\frac{2\mathbf{p}^{t+\Delta t} - 5\mathbf{p}^t + 4\mathbf{p}^{t-\Delta t} - \mathbf{p}^{t-2\Delta t}}{\Delta t^2} \right) \\ + \mathbf{C} \left(\frac{11\mathbf{p}^{t+\Delta t} - 18\mathbf{p}^t + 9\mathbf{p}^{t-\Delta t} - 2\mathbf{p}^{t-2\Delta t}}{6\Delta t} \right) \\ + \mathbf{K}\mathbf{p}^{t+\Delta t} = f_p^{t+\Delta t}, \end{aligned}$$

es decir

$$\begin{aligned} \left(\frac{2\mathbf{M}}{\Delta t^2} + \frac{11\mathbf{C}}{6\Delta t} + \mathbf{K} \right) \mathbf{p}^{t+\Delta t} = f_p^{t+\Delta t} + \left(\frac{5\mathbf{M}}{\Delta t^2} + \frac{3\mathbf{C}}{\Delta t} \right) \mathbf{p}^t - \\ \left(\frac{4\mathbf{M}}{\Delta t^2} + \frac{3\mathbf{C}}{2\Delta t} \right) \mathbf{p}^{t-\Delta t} + \left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{3\Delta t} \right) \mathbf{p}^{t-2\Delta t}. \end{aligned} \quad (3.7)$$

El método de Houbolt es uno de los primeros esquemas implícitos de integración para la solución de problemas dinámicos. Es un esquema lineal multipaso incondicionalmente estable. De todas formas si el incremento de tiempo es muy grande la solución obtenida no se ajustará bien a la solución exacta.

Para conocer el valor de $\mathbf{p}^{t+\Delta t}$ el método de Houbolt requiere conocer los valores \mathbf{p}^t , $\mathbf{p}^{t-\Delta t}$ y $\mathbf{p}^{t-2\Delta t}$. Por lo tanto para empezar a utilizar este método será preciso empezar con un valor conocido para $\mathbf{p}^{-\Delta t}$ y $\mathbf{p}^{-2\Delta t}$. Con frecuencia se utilizan otros métodos de integración, principalmente el método de las diferencias centradas, para calcular los valores $\mathbf{p}^{\Delta t}$ y $\mathbf{p}^{2\Delta t}$ y con estos valores empezar a utilizar el método de Houbolt a partir del instante $t = 3\Delta t$ en adelante. Sin embargo también es razonable suponer que el objeto se encuentra en reposo hasta el instante anterior a t_0 , momento en el que empiezan a actuar las fuerzas definidas, es decir $\mathbf{p}^{-2} = \mathbf{p}^{-1} = \mathbf{p}^0$.

3.1.4. El método Wilson θ

Este método se basa en la suposición que la aceleración a lo largo del tiempo es lineal. Concretamente el método Wilson θ considera que la aceleración entre los instantes t y $t + \theta\Delta t$ es lineal, siendo $\theta \geq 1.0$.

El razonamiento sobre el que se basa este método es el siguiente. Sea τ el incremento de tiempo considerado, siendo $0 \leq \tau \leq \theta\Delta t$, entonces en el intervalo de tiempo $[t, t + \theta\Delta t]$ puede considerarse que la aceleración en el instante de tiempo τ es de la forma

$$\ddot{\mathbf{p}}^{t+\tau} = \ddot{\mathbf{p}}^t + \frac{\tau}{\theta\Delta t} \left(\ddot{\mathbf{p}}^{t+\theta\Delta t} - \ddot{\mathbf{p}}^t \right), \quad (3.8)$$

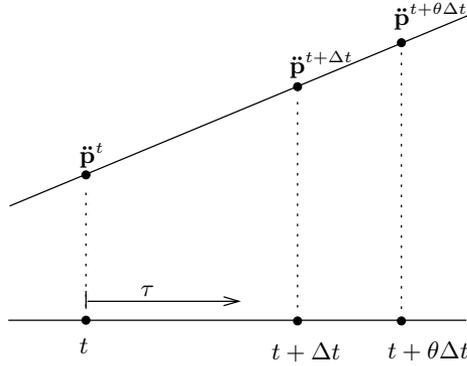


Figura 3.1: La asunción de una aceleración constante es la base de algunos métodos, entre ellos se encuentran el método Wilson θ y el método Newmark β .

expresión que, una vez integrada, permite obtener el valor de $\dot{\mathbf{p}}^{t+\tau}$ y de $\mathbf{p}^{t+\tau}$:

$$\begin{aligned}\dot{\mathbf{p}}^{t+\tau} &= \dot{\mathbf{p}}^t + \tau \ddot{\mathbf{p}}^t + \frac{\tau^2}{2\theta\Delta t} \left(\ddot{\mathbf{p}}^{t+\theta\Delta t} - \ddot{\mathbf{p}}^t \right), \\ \mathbf{p}^{t+\tau} &= \mathbf{p}^t + \tau \dot{\mathbf{p}}^t + \frac{\tau^2}{2} \ddot{\mathbf{p}}^t + \frac{\tau^3}{6\theta\Delta t} \left(\ddot{\mathbf{p}}^{t+\theta\Delta t} - \ddot{\mathbf{p}}^t \right)\end{aligned}\quad (3.9)$$

y, en consecuencia, en el instante de tiempo $t + \theta\Delta t$

$$\begin{aligned}\dot{\mathbf{p}}^{t+\theta\Delta t} &= \dot{\mathbf{p}}^t + \frac{\theta\Delta t}{2} \left(\ddot{\mathbf{p}}^{t+\theta\Delta t} + \ddot{\mathbf{p}}^t \right), \\ \mathbf{p}^{t+\theta\Delta t} &= \mathbf{p}^t + \theta\Delta t \dot{\mathbf{p}}^t + \frac{\theta^2\Delta t^2}{6} \left(\ddot{\mathbf{p}}^{t+\theta\Delta t} + 2\ddot{\mathbf{p}}^t \right),\end{aligned}$$

a partir de $\mathbf{p}^{t+\theta\Delta t}$ se puede obtener la expresión de la velocidad y de la aceleración en el instante $t + \theta\Delta t$:

$$\begin{aligned}\ddot{\mathbf{p}}^{t+\theta\Delta t} &= \frac{6}{\theta^2\Delta t^2} \left(\mathbf{p}^{t+\theta\Delta t} - \mathbf{p}^t \right) - \frac{6}{\theta\Delta t} \dot{\mathbf{p}}^t - 2\ddot{\mathbf{p}}^t, \\ \dot{\mathbf{p}}^{t+\theta\Delta t} &= \frac{3}{\theta\Delta t} \left(\mathbf{p}^{t+\theta\Delta t} - \mathbf{p}^t \right) - 2\dot{\mathbf{p}}^t - \frac{\theta\Delta t}{2} \ddot{\mathbf{p}}^t\end{aligned}\quad (3.10)$$

entonces, considerando que la aceleración varía linealmente, la ecuación de movimiento se convierte en

$$\mathbf{M}\ddot{\mathbf{p}}^{t+\theta\Delta t} + \mathbf{C}\dot{\mathbf{p}}^{t+\theta\Delta t} + \mathbf{K}\mathbf{p}^{t+\theta\Delta t} = \theta f_p^{t+\Delta t} + (1-\theta)f_p^t. \quad (3.11)$$

Aplicando a la ecuación (3.11) los resultados obtenidos en la ecuación (3.10) se obtiene el valor de $\mathbf{p}^{t+\theta\Delta t}$

$$\begin{aligned} \left(\frac{6\mathbf{M}}{\theta^2\Delta t^2} + \frac{3\mathbf{C}}{\theta\Delta t} + \mathbf{K} \right) \mathbf{p}^{t+\theta\Delta t} = & \theta f_p^{t+\Delta t} + (1-\theta)f_p^t + \left(\frac{6\mathbf{M}}{\theta^2\Delta t^2} + \frac{3\mathbf{C}}{\theta\Delta t} \right) \mathbf{p}^t + \\ & \left(\frac{6\mathbf{M}}{\theta\Delta t} + 2\mathbf{C} \right) \dot{\mathbf{p}}^t + \left(2\mathbf{M} + \frac{\theta\Delta t\mathbf{C}}{2} \right) \ddot{\mathbf{p}}^t, \end{aligned} \quad (3.12)$$

el valor obtenido para $\mathbf{p}^{t+\theta\Delta t}$ puede aplicarse a las ecuaciones (3.10) para obtener el valor de $\ddot{\mathbf{p}}^{t+\theta\Delta t}$ y $\dot{\mathbf{p}}^{t+\theta\Delta t}$. Una vez conocidos dichos valores se pueden aplicar a las ecuaciones (3.8) y (3.9) considerando $\tau = \Delta t$ para de esta forma obtener los valores de $\ddot{\mathbf{p}}^{t+\Delta t}$, $\dot{\mathbf{p}}^{t+\Delta t}$ y $\mathbf{p}^{t+\Delta t}$.

A diferencia de otros métodos, el método Wilson θ no necesita estimar valores para instantes de tiempo anteriores a $t = 0$ ya que todos los valores en un instante dado, $t + \Delta t$, se expresan exclusivamente en función de las cantidades calculadas para el instante anterior, t . Wilson θ es también un método de integración implícito incondicionalmente estable cuando $\theta \geq 1.37$.

3.1.5. El método Newmark β

Como el método Wilson θ , el esquema de integración de Newmark β puede ser considerado como una extensión del método de las diferencias centradas implícito ya descrito en el apartado 3.1.2.

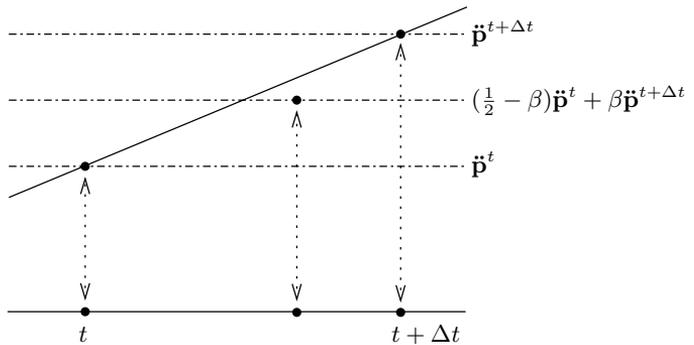


Figura 3.2: Según el planteamiento del método Newmark β la aceleración en un instante dado se plantea como una ponderación entre dos valores extremos.

El método parte de una expresión de la velocidad y de la posición como sigue:

$$\begin{aligned}\dot{\mathbf{p}}^{t+\Delta t} &= \dot{\mathbf{p}}^t + ((1 - \gamma)\ddot{\mathbf{p}}^t + \gamma\ddot{\mathbf{p}}^{t+\Delta t}) \Delta t, \\ \mathbf{p}^{t+\Delta t} &= \mathbf{p}^t + \dot{\mathbf{p}}^t \Delta t + \left(\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{p}}^t + \beta \ddot{\mathbf{p}}^{t+\Delta t} \right) \Delta t^2,\end{aligned}\quad (3.13)$$

donde β y γ son dos parámetros que permiten determinar la precisión y la estabilidad del método ($0 \leq \beta, \gamma \leq 1$). Los valores de γ y β representan la influencia que tienen $\ddot{\mathbf{p}}^t$ y $\ddot{\mathbf{p}}^{t+\Delta t}$ en el cálculo de $\dot{\mathbf{p}}^{t+\Delta t}$ y de $\mathbf{p}^{t+\Delta t}$ respectivamente. En particular si $\beta = \frac{1}{6}$ y $\gamma = \frac{1}{2}$ el método Newmark β equivale al método de la aceleración lineal :

$$\begin{aligned}\dot{\mathbf{p}}^{t+\Delta t} &= \dot{\mathbf{p}}^t + \left(\left(1 - \frac{1}{2} \right) \ddot{\mathbf{p}}^t + \frac{1}{2} \ddot{\mathbf{p}}^{t+\Delta t} \right) \Delta t \\ &= \dot{\mathbf{p}}^t + \frac{1}{2} (\ddot{\mathbf{p}}^t + \ddot{\mathbf{p}}^{t+\Delta t}) \Delta t, \\ \mathbf{p}^{t+\Delta t} &= \mathbf{p}^t + \dot{\mathbf{p}}^t \Delta t + \left(\left(\frac{1}{2} - \frac{1}{6} \right) \ddot{\mathbf{p}}^t + \frac{1}{6} \ddot{\mathbf{p}}^{t+\Delta t} \right) \Delta t^2 \\ &= \mathbf{p}^t + \dot{\mathbf{p}}^t \Delta t + \frac{1}{6} (2\ddot{\mathbf{p}}^t + \ddot{\mathbf{p}}^{t+\Delta t}) \Delta t^2.\end{aligned}$$

Para estos valores el método es condicionalmente estable, véase el apartado 3.2.1, su interés radica en que a partir de este planteamiento se derivó el método Wilson θ ([Bat82, Bat96]).

El caso en que se considera $\beta = \frac{1}{4}$ y $\gamma = \frac{1}{2}$ es conocido como el método trapezoidal, también conocido como método de la aceleración media . En este caso se considera que la aceleración en el intervalo $[t, t + \Delta t]$ es constante y su valor es $\frac{\ddot{\mathbf{a}}^t + \ddot{\mathbf{a}}^{t+\Delta t}}{2}$:

$$\begin{aligned}\dot{\mathbf{p}}^{t+\Delta t} &= \dot{\mathbf{p}}^t + \left(\left(1 - \frac{1}{2} \right) \ddot{\mathbf{p}}^t + \frac{1}{2} \ddot{\mathbf{p}}^{t+\Delta t} \right) \Delta t \\ &= \dot{\mathbf{p}}^t + \frac{1}{2} (\ddot{\mathbf{p}}^t + \ddot{\mathbf{p}}^{t+\Delta t}) \Delta t, \\ \mathbf{p}^{t+\Delta t} &= \mathbf{p}^t + \dot{\mathbf{p}}^t \Delta t + \left(\left(\frac{1}{2} - \frac{1}{4} \right) \ddot{\mathbf{p}}^t + \frac{1}{4} \ddot{\mathbf{p}}^{t+\Delta t} \right) \Delta t^2 \\ &= \mathbf{p}^t + \dot{\mathbf{p}}^t \Delta t + \frac{1}{4} (\ddot{\mathbf{p}}^t + \ddot{\mathbf{p}}^{t+\Delta t}) \Delta t^2.\end{aligned}$$

El método Newmark β parte de un planteamiento implícito, para la ecuación (3.2), en donde a partir de (3.13), se puede expresar $\ddot{\mathbf{p}}^{t+\Delta t}$ en función de \mathbf{p}^t y $\dot{\mathbf{p}}^t$

exclusivamente.

$$\begin{aligned}
 \left(\frac{1}{\beta \Delta t^2} \mathbf{M} + \frac{\gamma}{\beta \Delta t} \mathbf{C} + \mathbf{K} \right) \mathbf{p}^{t+\Delta t} &= f_p^{t+\Delta t} \\
 &+ \left(\left(\frac{1}{2\beta} - 1 \right) \mathbf{M} + \Delta t \left(\frac{\gamma}{2\beta} - 1 \right) \mathbf{C} \right) \ddot{\mathbf{p}}^t \\
 &+ \left(\frac{1}{\beta \Delta t^2} \mathbf{M} + \left(\frac{\gamma}{\beta \Delta t} - 1 \right) \mathbf{C} \right) \dot{\mathbf{p}}^t \\
 &+ \left(\frac{1}{\beta \Delta t^2} \mathbf{M} + \frac{\gamma}{\beta \Delta t} \mathbf{C} \right) \mathbf{p}^t.
 \end{aligned} \tag{3.14}$$

Acto seguido se puede volver a (3.13) sustituyendo $\ddot{\mathbf{p}}^{t+\Delta t}$ por la expresión obtenida, de esta forma se puede determinar el valor de $\dot{\mathbf{p}}^{t+\Delta t}$ y de $\mathbf{p}^{t+\Delta t}$. Para valores de $\gamma \neq \frac{1}{2}$ se introduce en el sistema una amortiguación superflua [Paz92], por ello es usual que el valor de γ sea siempre fijado en $\frac{1}{2}$. Considerando $\gamma = \frac{1}{2}$ y $\beta = \frac{1}{4}$ se obtiene el mayor grado de estabilidad y la ecuación (3.14) se expresa de la siguiente forma

$$\begin{aligned}
 \left(\frac{4}{\Delta t^2} \mathbf{M} + \frac{2}{\Delta t} \mathbf{C} + \mathbf{K} \right) \mathbf{p}^{t+\Delta t} &= f_p^{t+\Delta t} + \mathbf{M} \ddot{\mathbf{p}}^t + \left(\frac{4}{\Delta t^2} \mathbf{M} + \left(\frac{2}{\Delta t} - 1 \right) \mathbf{C} \right) \dot{\mathbf{p}}^t \\
 &+ \left(\frac{4}{\Delta t^2} \mathbf{M} + \frac{2}{\Delta t} \mathbf{C} \right) \mathbf{p}^t
 \end{aligned} \tag{3.15}$$

3.2. Comparación de los métodos de integración

La forma de construir la matriz del sistema para los distintos métodos de integración determina las ventajas e inconvenientes de cada uno de ellos. Generalmente un planteamiento explícito produce una matriz de sistema diagonal, con lo que su resolución es muy sencilla, ello es debido a que la matriz del sistema se construye con los valores de las matrices de masas, \mathbf{M} , y de amortiguación, \mathbf{C} . Esto es así siempre y cuando se considere que tanto la masa como el coeficiente de amortiguación se mantengan constantes en el tiempo.

El mayor inconveniente surge del mismo concepto, un planteamiento explícito no permite un análisis estático ya que si se plantea un análisis de este tipo, es decir sin considerar los componentes de la dinámica, se obtiene una matriz de sistema nula.

Por otro lado los métodos implícitos forman su matriz de sistema con las matrices de masas, \mathbf{M} , de amortiguación, \mathbf{C} , y de rigidez, \mathbf{K} . Este cambio de forma permite que este tipo de métodos sean útiles para un análisis estático. El inconveniente surge del hecho que la matriz del sistema ya no tiene por qué ser diagonal y por lo tanto el coste de la resolución del sistema será mayor. Además la matriz de rigidez puede variar a lo largo de la simulación con lo que puede ser preciso que tenga que ser recreada en cada paso de la simulación.

3.2.1. Estabilidad y precisión

Uno de los aspectos más importantes a la hora de escoger un método u otro es la estabilidad que ofrece cada uno. La estabilidad de un método de integración se determina en función del comportamiento de la solución numérica a partir de unas condiciones iniciales arbitrarias. Se dice que un método de integración es *incondicionalmente estable* si el error en la solución obtenida para cualquier condición inicial no crece sin límite para cualquier incremento de tiempo Δt , en particular cuando $\frac{\Delta t}{T}$ es grande. En cambio un método es solamente *condicionalmente estable* si lo anterior se cumple solamente para el caso en que $\frac{\Delta t}{T}$ sea menor o igual a un cierto valor, el denominado límite de estabilidad.

En [Bat82, Bat96] se realiza un estudio detallado de la estabilidad de los diferentes esquemas de integración. De dicho estudio se concluye que:

- El método de las diferencias centradas es condicionalmente estable. La estabilidad requiere que $\Delta t \leq \frac{T_n}{\pi}$ siendo T_n el valor del menor modo de vibración, el menor valor propio, del sistema.
- El método de Houbolt es incondicionalmente estable independientemente del valor del paso del tiempo.
- El método Wilson θ es incondicionalmente estable si $\theta \geq 1.37$.
- El método Newmark β es incondicionalmente estable si $\gamma \geq \frac{1}{2}$ y además $\beta \geq \frac{1}{4}(\gamma + \frac{1}{2})^2$.

Además en el método Newmark β según sea el valor de β puede producirse una amortiguación espuria proporcional a $(\beta - \frac{1}{2})$, de manera que si $\beta < \frac{1}{2}$ entonces se produce una amortiguación negativa, con lo que se aumenta la vibración del modelo, en cambio si $\beta > \frac{1}{2}$ entonces se produce una amortiguación mayor que la esperada [GdJB94].

De todas formas cabe remarcar que los métodos descritos como incondicionalmente estables pueden pasar a ser condicionalmente estables en determinadas circunstancias, como por ejemplo si se aplican a sistemas no lineales.

Para poder comparar los diferentes métodos estudiados se ha procedido primeramente a aplicarlos para resolver un pequeño problema (basado en un ejercicio enunciado en [Bat96]), que consiste en resolver el sistema representado por:

$$\mathbf{M} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} 6 & -2 \\ -2 & 1 \end{pmatrix} \text{ y } \mathbf{f}_p = \begin{pmatrix} 0 \\ 10 \end{pmatrix},$$

considerando $\mathbf{p}^0 = \{0, 0\}$ y $\dot{\mathbf{p}}^0 = \{0, 0\}$ y con $T = 14$ y $\Delta t = 0.28$. La solución exacta viene dada por la expresión:

$$\mathbf{p}(t) = \begin{pmatrix} \frac{5}{3} (1 - \cos(\sqrt{2}t)) + \frac{2}{3} (-1 + \cos(\sqrt{5}t)) \\ \frac{5}{3} (1 - \cos(\sqrt{2}t)) - \frac{4}{3} (-1 + \cos(\sqrt{5}t)) \end{pmatrix}. \quad (3.16)$$

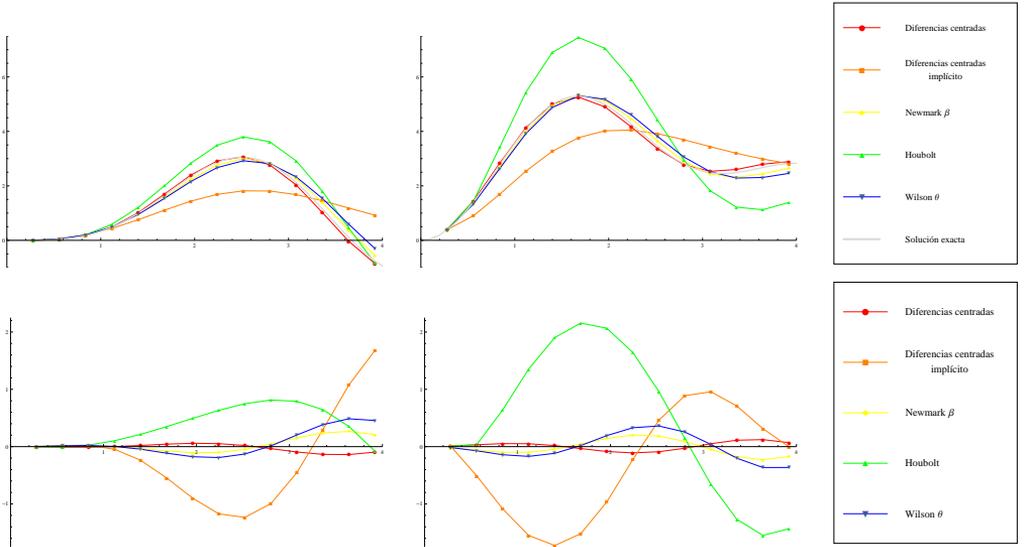


Figura 3.3: Visualización de los resultados obtenidos en la comparativa para cada componente. La primera columna muestra los valores correspondientes a la primera componente y la segunda columna hace lo propio con la segunda componente. En la primera fila se muestran los resultados para cada método, para poder comparar la exactitud del resultado obtenido también se muestra el valor exacto calculado a partir de (3.16). En la segunda fila se muestra la diferencia de los valores de cada método respecto del valor exacto calculado.

Los resultados obtenidos para cada método descrito se muestran en la tabla 3.1 y visualmente en la figura 3.3. Como puede observarse, para $\Delta t = 0.28$ el método que más se ajusta al valor exacto es el método de diferencias centradas. El

Solución de la ecuación de equilibrio

	Δt	$2\Delta t$	$3\Delta t$	$4\Delta t$	$5\Delta t$	$6\Delta t$	$7\Delta t$
exacta	0.0025	0.0381	0.1756	0.4860	0.9964	1.6570	2.3382
	0.3819	1.4116	2.7809	4.0936	4.9962	5.2905	4.9857
diferencias centradas	0.0000	0.0307	0.1675	0.4871	1.0170	1.7009	2.3969
	0.3920	1.4451	2.8338	4.1441	5.0152	5.2570	4.9009
diferencias centradas implícito	0.0000	0.0573	0.2003	0.4395	0.7571	1.1093	1.4385
	0.3920	0.9021	1.6958	2.5445	3.2703	3.7714	4.0210
Houbolt	0.0000	0.0307	0.1675	0.4659	0.9360	1.5298	2.1453
	0.3920	1.4451	2.8338	4.1657	5.1301	5.5522	5.4143
Wilson θ	0.0060	0.0525	0.1960	0.4896	0.9516	1.5425	2.1623
	0.3663	1.3393	2.6394	3.9235	4.8793	5.3093	5.1781
Newmark β	0.0067	0.0504	0.1894	0.4846	0.9613	1.5805	2.2328
	0.3637	1.3510	2.6833	3.9954	4.9497	5.3366	5.1296

	$8\Delta t$	$9\Delta t$	$10\Delta t$	$11\Delta t$	$12\Delta t$	$13\Delta t$	$14\Delta t$
exacta	2.8608	3.0517	2.8057	2.1306	1.1572	0.1095	-0.7583
	4.2766	3.4575	2.8062	2.4843	2.4888	2.6724	2.8224
diferencias centradas	2.9134	3.0714	2.7711	2.0368	1.0223	-0.0288	-0.8539
	4.1677	3.3683	2.7783	2.5354	2.6008	2.7949	2.8920
diferencias centradas implícito	1.6881	1.8175	1.8107	1.6788	1.4553	1.1868	0.9223
	4.0494	3.9181	3.6958	3.4414	3.1967	2.9858	2.8193
Houbolt	3.4942	3.8007	3.6193	2.9220	1.8026	0.4617	-0.8368
	5.9213	4.4211	2.9491	1.8259	1.2192	1.1230	1.3904
Wilson θ	2.6702	2.9226	2.8182	2.3340	1.5415	0.5956	-0.3018
	4.6064	3.8182	3.0605	2.5233	2.2862	2.3078	2.4588
Newmark β	2.7607	3.0035	2.8505	2.2840	1.3968	0.3752	-0.5474
	4.4781	3.6424	2.8967	2.4352	2.3129	2.4433	2.6483

Tabla 3.1: Comparativa de los valores obtenidos según el método de integración.

resultado para los demás métodos claramente diverge del valor real y es de esperar que para un tiempo T grande las diferencias puedan llegar a ser considerables.

Por otro lado, también es muy importante el grado de exactitud del método escogido. El grado de exactitud está ligado al coste requerido para alcanzarlo. Se tiene que conseguir un equilibrio entre el coste y la precisión.

En los métodos basados en la integración directa, el coste en tiempo para obtener la solución del sistema vendrá determinado por el número de iteraciones a realizar. Si se utiliza un sistema condicionalmente estable, como es el caso del método de las diferencias centradas, probablemente será necesario establecer cual es el mayor valor que puede tomar Δt . Dicho valor establecerá el número pasos de tiempo que deberán realizarse para poder calcular toda la simulación. El uso de sistemas condicionalmente estables se reservará entonces para aquellos casos en los que se pueda garantizar un valor para Δt suficientemente grande como para que el coste del cálculo de la solución sea aceptable.

En cambio, los métodos incondicionalmente estables no tienen tal dependencia del valor Δt , aunque para garantizar la estabilidad sea preciso que el valor de Δt sea menor que algunos de los modos de vibración mayores, es decir que los mayores valores propios de la matriz del sistema.

Una vez descritos los diferentes métodos de integración cabe analizar su adecuación para la simulación de objetos deformables elásticamente. Se ha escogido el trabajo descrito originalmente en [TPBF87] y que ha sido un referente en incontables trabajos. La idea es desarrollar algorítmicamente el planteamiento detallado en dicho artículo y realizar la integración numérica con cada uno de los métodos estudiados para poder determinar su adecuación para la simulación de objetos deformables elásticamente.

3.3. Estudio de los métodos de integración en un caso complejo

En la amplia bibliografía relacionada con el modelado y la animación de objetos deformables, se puede constatar que las técnicas basadas en la física pueden ser utilizadas para crear y animar objetos deformables para obtener resultados realistas. A tal fin, el modelo continuo que describe la evolución en el tiempo puede ser aproximado mediante métodos de discretización tales como el método de diferencias finitas o el método de elementos finitos (véase [SBS96], y además [MMP99] para este último método). Para el caso de la discretización mediante diferencias finitas, un objeto deformable es aproximado utilizando un mallado de puntos de control. En éste los puntos pueden moverse unos respecto a otros aunque manteniendo la conexión con sus vecinos. La forma en que los puntos pueden moverse determina las propiedades del objeto deformable. Dentro de estos métodos se incluyen aquéllos que intentan determinar el movimiento del objeto utilizando el modelo lagrangiano, ecuación (2.17), y definiendo para el objeto un potencial de energía que determine la forma en que puede ser deformado. Dicho funcional puede basarse en el teorema fundamental de las superficies de la Geometría Diferencial, tal y como se detalla en [TPBF87] y en [TW88] (véase también [GO94], [Gre94] y [KSS96]). El modelo escogido para evaluar los métodos de integración estudiados pertenece a esta misma categoría.

3.3.1. Ecuación de un modelo deformable elásticamente

Sea Ω un objeto. Supóngase que Ω es una superficie con coordenadas locales $p = (x_1, x_2)$ siendo $(x_1, x_2) \in [0, 1] \times [0, 1]$ (suponiendo que el dominio de para-

metrización de Ω es el cuadrado unitario). La posición de un punto del objeto en el tiempo t será

$$\mathbf{r}(p; t) := \mathbf{r}(x_1, x_2; t) = (r_1(x_1, x_2; t), r_2(x_1, x_2; t), r_3(x_1, x_2; t)),$$

y la posición inicial del objeto será

$$\mathbf{r}^0(p) := \mathbf{r}^0(x_1, x_2) = (r_1^0(x_1, x_2), r_2^0(x_1, x_2), r_3^0(x_1, x_2)).$$

Para que el objeto sea deformable el siguiente paso será introducir la energía de deformación. Para ello se deberá tener en cuenta el teorema fundamental de las superficies: dos superficies tienen la misma forma si su tensor métrico G , primera forma fundamental, y su tensor de curvatura B , segunda forma fundamental, son iguales, siendo

$$\begin{aligned} G_{ij} &= G_{ij}(\mathbf{r}(x_1, x_2; t)) = \frac{\partial \mathbf{r}}{\partial x_i} \cdot \frac{\partial \mathbf{r}}{\partial x_j}, \\ B_{ij} &= B_{ij}(\mathbf{r}(x_1, x_2; t)) = \vec{n} \cdot \frac{\partial^2 \mathbf{r}}{\partial x_i \partial x_j}, \end{aligned}$$

y \vec{n} es un vector unitario normal a la superficie. Véase [dC76] para una detallada discusión acerca de la geometría diferencial de curvas y superficies. Utilizando los tensores anteriores, se define la energía de deformación para el objeto Ω en el instante t como:

$$\varepsilon(\mathbf{r}) = \sum_{i,j=1}^2 \int_{\Omega} \left[\eta_{ij} (G_{ij} - G_{ij}^0)^2 + \xi_{ij} (B_{ij} - B_{ij}^0)^2 \right] dx_1 dx_2, \quad (3.17)$$

donde η_{ii} representa la resistencia del objeto a deformaciones longitudinales, $\eta_{12} = \eta_{21}$ representan la resistencia del objeto a la deformación por cizalladura, ξ_{ii} representa la resistencia del objeto a la deformación por doblamiento y $\xi_{12} = \xi_{21}$ representan la resistencia del objeto a la deformación por retorcimiento. Por lo tanto la energía vendrá dada como una ponderación de la diferencia de los tensores métrico y de curvatura en cada instante dado respecto del instante inicial.

Como ya se ha constatado en el capítulo 2, la ecuación del movimiento para un objeto deformable puede ser escrita con una formulación lagrangiana (véase [Gol80, Sha88, GdJB94]). La ecuación (2.22) puede ser completada haciendo uso del funcional antes descrito, como

$$\frac{\partial}{\partial t} \left(\mu \frac{\partial \mathbf{r}}{\partial t} \right) + \gamma \frac{\partial \mathbf{r}}{\partial t} + \frac{\delta \varepsilon(\mathbf{r})}{\delta \mathbf{r}} = f(\mathbf{r}, t), \quad (3.18)$$

en donde $\frac{\delta \varepsilon(\mathbf{r})}{\delta \mathbf{r}}$ es la primera derivada variacional para el funcional de deformación y $f(\mathbf{r}, t)$ es la resultante de las fuerzas externas aplicadas sobre el objeto. El primer

término representa la fuerza inercial debida a la distribución de masas, el segundo término es la fuerza de amortiguación debida a la disipación de la energía, y el tercer término representa la fuerza elástica debida a la deformación del objeto respecto de su forma natural.

Para poder crear animaciones mediante el modelo descrito, la ecuación diferencial de movimiento puede ser discretizada mediante la aplicación de aproximaciones basadas en métodos de diferencias finitas y resolviendo el sistema de ecuaciones diferenciales ordinarias encadenadas resultante que describe el movimiento. Aunque previamente es preciso simplificar la primera derivada variacional del funcional $\varepsilon(\mathbf{r})$. Respecto a la aproximación de $\frac{\delta\varepsilon(\mathbf{r})}{\delta\mathbf{r}}$ se puede comprobar en [TPBF87] que

$$\frac{\delta\varepsilon(\mathbf{r})}{\delta\mathbf{r}} \approx \sum_{i,j=1}^2 -\frac{\partial}{\partial x_i} \left(\alpha_{ij} \frac{\partial\mathbf{r}}{\partial x_j} \right) + \frac{\partial^2}{\partial x_i \partial x_j} \left(\beta_{ij} \frac{\partial^2\mathbf{r}}{\partial x_i \partial x_j} \right), \quad (3.19)$$

siendo $\alpha_{ij} = \eta_{ij}(p; t) (G_{ij} - G_{ij}^0)$ y $\beta_{ij} = \xi_{ij}(p; t) (B_{ij} - B_{ij}^0)$.

3.3.2. Resolución numérica del modelo de deformación

La ecuación (3.18), que representa tanto el movimiento de la superficie como su deformación, puede ser resuelta mediante el uso de diferencias finitas tanto en el dominio espacial como en el temporal. Para ello, se deben aproximar las derivadas espaciales para transformar la ecuación en derivadas parciales (3.18) en una ecuación diferencial ordinaria.

Mediante el uso de un método de integración directa, como los descritos en la sección 3.1, es posible expresar la función sobre el mallado $\mathbf{r}(m, n)$ como un vector de dimensión MN , \mathbf{r} simplificando la notación. Por lo tanto, se transforma la ecuación en derivadas parciales (3.18) en la siguiente ecuación diferencial:

$$\mathbf{M} \frac{d^2\mathbf{r}}{dt^2} + \mathbf{C} \frac{d\mathbf{r}}{dt} + \mathbf{K}(\mathbf{r})\mathbf{r} = \mathbf{f}_r, \quad (3.20)$$

en donde \mathbf{M} es la $MN \times MN$ matriz diagonal de masas, \mathbf{C} es la $MN \times MN$ matriz diagonal de amortiguación y $\mathbf{K}(\mathbf{r})$ es la $MN \times MN$ matriz de rigidez que contiene toda la información relativa a las propiedades de la deformación del objeto. Nótese que \mathbf{M} es la matriz de masas siendo $\mu(m, n)$ los elementos de la diagonal, y que \mathbf{C} es construida de forma semejante a partir de los valores de $\gamma(m, n)$. Como se podrá comprobar, la matriz $\mathbf{K}(\mathbf{r})$ es la más importante en la resolución numérica de la ecuación (3.20) ya que aporta la información de la

deformación. Cabe remarcar que la expresión obtenida es equivalente a (2.22) considerando un funcional concreto.

3.3.3. Discretización temporal

Con independencia del esquema de diferencias finitas utilizado para discretizar las coordenadas espaciales, para resolver la ecuación (3.20) se debe utilizar un método de integración temporal adecuado para la ecuación diferencial ordinaria:

$$\mathbf{A} \cdot \mathbf{r}(t + \Delta t) = \mathbf{b}(t), \quad (3.21)$$

en principio cualquiera de los métodos de integración descritos en este capítulo es aplicable. Por ejemplo, si se utiliza un esquema en diferencias centradas, para resolver la ecuación diferencial, se utilizan las siguientes aproximaciones temporales discretas:

$$\begin{aligned} \frac{d\mathbf{r}}{dt} &\approx \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t - \Delta t)}{2\Delta t}, \\ \frac{d^2\mathbf{r}}{dt^2} &\approx \frac{\mathbf{r}(t + \Delta t) - 2\mathbf{r}(t) + \mathbf{r}(t - \Delta t)}{\Delta t^2}. \end{aligned}$$

de manera que

$$\mathbf{A} = \frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t}$$

y

$$\mathbf{b}(t) = \mathbf{f}(t) - \left(\mathbf{K}(\mathbf{r}(t)) - \frac{2\mathbf{M}}{\Delta t^2} \right) \mathbf{r}(t) + \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2} \right) \mathbf{r}(t - \Delta t).$$

El sistema (3.21) es fácilmente resoluble toda vez que la matriz \mathbf{A} es diagonal. Puede escribirse

$$\mathbf{r}(t + \Delta t) = \mathbf{A}^{-1} \cdot \mathbf{b}(t) = \mathbf{S} \cdot \mathbf{b}(t),$$

en donde $S_{ij} = 0$, si $i \neq j$ y $S_{ii} = \left(\frac{m_{ii}}{\Delta t^2} + \frac{c_{ii}}{2\Delta t} \right)^{-1}$. De esta forma la ecuación en derivadas parciales no lineal original, (3.18), se ve reducida a una secuencia de sistemas algebraicos diagonales representados por (3.21).

El planteamiento original de [TPBF87] para la integración numérica a lo largo del tiempo se basa en un procedimiento de integración semi-implícito, véase [TPBF87, TW88, GO94], equivalente al método de las diferencias centradas implícito descrito en el apartado 3.1.2:

$$\mathbf{A}(t) \cdot \mathbf{r}(t + \Delta t) = \mathbf{g}_t, \quad (3.22)$$

siendo

$$\mathbf{A}(t) = \mathbf{K}(\mathbf{r}(t)) + \left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t} \right)$$

y

$$\mathbf{g}_t = \mathbf{f}_t + \frac{2\mathbf{M}}{\Delta t^2} \cdot \mathbf{r}(t) + \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2} \right) \cdot \mathbf{r}(t - \Delta t).$$

En lugar de aplicar solamente el modelo propuesto, en el presente estudio se procederá a utilizar cada uno de los métodos de integración descritos en la sección 3.1 para poder determinar la adecuación de cada uno de ellos a este tipo de problemas.

3.3.4. Discretización espacial

Para resolver numéricamente la ecuación (3.18) se han utilizado los operadores en diferencias finitas estándar. En primer lugar se ha utilizado un esquema basado en *diferencias finitas centradas*. De manera que para una función fijada, h , los operadores en diferencias para h son los siguientes

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial x_1} &\approx \frac{\mathbf{h}(x_1 + \Delta x_1, x_2) - \mathbf{h}(x_1 - \Delta x_1, x_2)}{2\Delta x_1}, \\ \frac{\partial \mathbf{h}}{\partial x_2} &\approx \frac{\mathbf{h}(x_1, x_2 + \Delta x_2) - \mathbf{h}(x_1, x_2 - \Delta x_2)}{2\Delta x_2}, \\ \frac{\partial^2 \mathbf{h}}{\partial x_1^2} &\approx \frac{\mathbf{h}(x_1 + \Delta x_1, x_2) - 2\mathbf{h}(x_1, x_2) + \mathbf{h}(x_1 - \Delta x_1, x_2)}{\Delta x_1^2}, \\ \frac{\partial^2 \mathbf{h}}{\partial x_2^2} &\approx \frac{\mathbf{h}(x_1, x_2 + \Delta x_2) - 2\mathbf{h}(x_1, x_2) + \mathbf{h}(x_1, x_2 - \Delta x_2)}{\Delta x_2^2}, \\ \frac{\partial^2 \mathbf{h}}{\partial x_1 \partial x_2} &\approx \frac{1}{\Delta x_1 \Delta x_2} (-\mathbf{h}(x_1 - \Delta x_1, x_2 + \Delta x_2) + \mathbf{h}(x_1 + \Delta x_1, x_2 + \Delta x_2) \\ &\quad + \mathbf{h}(x_1 - \Delta x_1, x_2) - \mathbf{h}(x_1 + \Delta x_1, x_2)). \end{aligned}$$

y así hasta la derivadas de cuarto orden. Con este planteamiento el error en la aproximación es de orden 2.

A continuación se ha utilizado un esquema en *diferencias finitas no centradas* con un error en la aproximación de orden 1. Éste último método fue el utilizado en [TPBF87] y [TW88] (véase también [GO94]). Se definen los siguientes primeros

operadores hacia adelante y hacia atrás:

$$\begin{aligned}
 D_1^+ \mathbf{h}(x_1, x_2) &= \frac{\mathbf{h}(x_1 + \Delta x_1, x_2) - \mathbf{h}(x_1, x_2)}{\Delta x_1}, \\
 D_1^- \mathbf{h}(x_1, x_2) &= \frac{\mathbf{h}(x_1, x_2) - \mathbf{h}(x_1 - \Delta x_1, x_2)}{\Delta x_1}, \\
 D_2^+ \mathbf{h}(x_1, x_2) &= \frac{\mathbf{h}(x_1, x_2 + \Delta x_2) - \mathbf{h}(x_1, x_2)}{\Delta x_2}, \\
 D_2^- \mathbf{h}(x_1, x_2) &= \frac{\mathbf{h}(x_1, x_2) - \mathbf{h}(x_1, x_2 - \Delta x_2)}{\Delta x_2}.
 \end{aligned}$$

A partir de estos operadores en diferencias se pueden definir los operadores de segundo orden:

$$\begin{aligned}
 D_{12}^+ \mathbf{h}(x_1, x_2) &= D_{21}^+ \mathbf{h}(x_1, x_2) = D_1^+(D_2^+ \mathbf{h}(x_1, x_2)), \\
 D_{11} \mathbf{h}(x_1, x_2) &= D_1^-(D_1^+ \mathbf{h}(x_1, x_2)), \\
 D_{12}^- \mathbf{h}(x_1, x_2) &= D_{21}^- \mathbf{h}(x_1, x_2) = D_1^-(D_2^- \mathbf{h}(x_1, x_2)), \\
 D_{22} \mathbf{h}(x_1, x_2) &= D_2^-(D_2^+ \mathbf{h}(x_1, x_2)).
 \end{aligned}$$

Ahora, partiendo de una malla de puntos que represente a la superficie continua, en la que se aplicará la deformación, se pueden discretizar los componentes de la ecuación (3.18) utilizando la aproximación de la derivada variacional descrita en (3.19).

Otro aspecto a tener en cuenta en el proceso de discretización de la ecuación (3.18) son las condiciones de contorno. Se han tenido en cuenta dos tipos de condiciones de contorno. Por una parte se han introducido unas *condiciones de contorno naturales* en los bordes de la superficie en donde los operadores de diferencias intentan acceder a variables nodales fuera del dominio discreto, asignando el valor cero a dichos operadores. Esto equivale a desplazar el contorno de manera que se amplíe el dominio cuadrado $[0, 1] \times [0, 1]$ y convertir los valores $\mathbf{r}(x_1, x_2)$ del contorno antiguo en nuevos valores $\mathbf{r}(x_1, x_2)$ del nuevo contorno. Por otra parte, se han considerado *regiones de discretización en el contorno*. Se ha dividido en cuadrado unitario $[0, 1] \times [0, 1]$ en nueve regiones, como puede observarse en la figura 3.4. En cada una de las regiones se ha definido un conjunto de operadores de diferencias de primer y de segundo orden específico. Por ejemplo, para la zona VIII de la figura las siguientes expresiones son las aproximaciones a

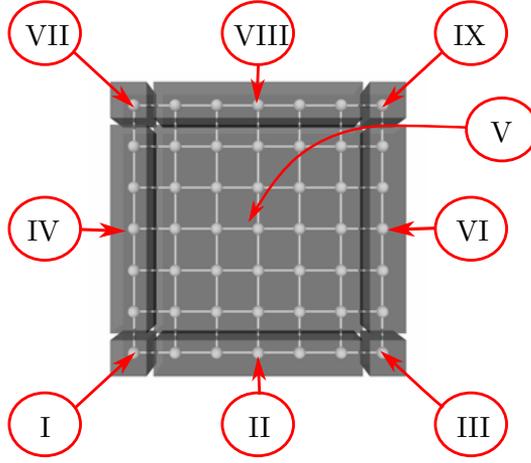


Figura 3.4: Regiones de discretización del contorno para poder calcular los operadores en diferencias finitas.

los operadores de diferencias:

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial x_1} &\approx \frac{3\mathbf{h}(x_1, x_2) - 4\mathbf{h}(x_1 - \Delta x_1, x_2) + \mathbf{h}(x_1 - 2\Delta x_1, x_2)}{2\Delta x_1}, \\ \frac{\partial \mathbf{h}}{\partial x_2} &\approx \frac{\mathbf{h}(x_1, x_2 + \Delta x_2) - \mathbf{h}(x_1, x_2 - \Delta x_2)}{2\Delta x_2}, \\ \frac{\partial^2 \mathbf{h}}{\partial x_1^2} &\approx \frac{1}{\Delta x_1^2} (2\mathbf{h}(x_1, x_2) - 5\mathbf{h}(x_1 - \Delta x_1, x_2) + 4\mathbf{h}(x_1 - 2\Delta x_1, x_2) \\ &\quad - \mathbf{h}(x_1 - 3\Delta x_1, x_2)), \\ \frac{\partial^2 \mathbf{h}}{\partial x_2^2} &\approx \frac{\mathbf{h}(x_1, x_2 + \Delta x_2) - 2\mathbf{h}(x_1, x_2) + \mathbf{h}(x_1, x_2 - \Delta x_2)}{\Delta x_2^2}, \\ \frac{\partial^2 \mathbf{h}}{\partial x_1 \partial x_2} &\approx \frac{1}{4\Delta x_1 \Delta x_2} (\mathbf{h}(x_1, x_2) - 2\mathbf{h}(x_1 - \Delta x_1, x_2) + 3\mathbf{h}(x_1 - \Delta x_1, x_2 - \Delta x_2) \\ &\quad - \mathbf{h}(x_1 - \Delta x_1, x_2 + \Delta x_2) - 2\mathbf{h}(x_1, x_2 - \Delta x_2) + \mathbf{h}(x_1, x_2 + \Delta x_2) \\ &\quad + \mathbf{h}(x_1 - 2\Delta x_1, x_2) - \mathbf{h}(x_1 - 2\Delta x_1, x_2 - \Delta x_2)), \end{aligned}$$

y así en cada una de las regiones y para cada una de las derivadas que aparecen en la ecuación (3.18). En el caso de tener en cuenta derivadas de tercer y cuarto orden se deberán contemplar un total de veinticinco regiones diferentes.

Así pues, desde el punto de vista de la discretización espacial, existen cuatro formas de resolver la ecuación en diferencias (3.18):

- (I) diferencias finitas centradas con contorno libre (*natural*),
- (II) diferencias finitas centradas con contorno organizado en zonas,
- (III) diferencias finitas no centradas con contorno libre (*natural*) y
- (IV) diferencias finitas no centradas con contorno organizado en zonas.

Para encontrar una aproximación numérica a la expresión (3.19), se puede escribir:

$$\mathbf{e}(x_1, x_2) = \sum_{i,j=1}^2 -D_i(\mathbf{p}_{ij})(x_1, x_2) + D_{ij}(\mathbf{q}_{ij})(x_1, x_2), \quad (3.23)$$

en donde

$$\mathbf{p}_{ij}(x_1, x_2) = \alpha_{ij}(x_1, x_2) \cdot D_j \mathbf{r}(x_1, x_2)$$

y

$$\mathbf{q}_{ij}(x_1, x_2) = \beta_{ij}(x_1, x_2) \cdot D_{ij}(\mathbf{r})(x_1, x_2)$$

y D_i , D_j y D_{ij} son operadores en diferencias finitas.

Tal y como se ha comentado anteriormente, utilizando cualquiera de estos cuatro métodos se puede transformar la ecuación (3.18) en la ecuación diferencial ordinaria (3.20), y, haciendo uso de la aproximaciones de las derivadas discretizadas temporalmente, se puede integrar dicha ecuación a lo largo del tiempo mediante un procedimiento paso a paso, resolviendo el sistema algebraico lineal (3.21). La solución dinámica puede evolucionar, lo que en definitiva corresponde a la simulación de la deformación de la superficie, desde unas condiciones iniciales para el instante $t = 0$ resolviendo el sistema lineal algebraico para las configuraciones instantáneas $r(t + \Delta t)$ utilizando las soluciones precedentes ya calculadas que sean necesarias según sea el método de integración escogido.

3.3.5. Estudio de la estabilidad

Para poder determinar cual es la mejor forma de resolver el sistema (3.21) es preciso estudiar la estabilidad del sistema en función de la matriz $\mathbf{K}(\mathbf{r}(t))$ que se esté utilizando y según cual sea el método de integración escogido.

En el proceso de simulación se realizan los cálculos en una serie de instantes de tiempo ($\Delta t, 2\Delta t, 3\Delta t, \dots, n\Delta t = T$) de manera que a lo largo del cálculo de la simulación se irán obteniendo una serie de valores: $\mathbf{r}_0 =$ el objeto sin deformar, $\mathbf{r}_1 = \mathbf{r}(\Delta t)$, \dots , $\mathbf{r}_{n-1} = \mathbf{r}((n-1)\Delta t)$, $\mathbf{r}_n = \mathbf{r}(n\Delta t) = \mathbf{r}(T)$, uno para cada instante de tiempo contemplado.

El radio espectral, ρ , está estrechamente relacionado con el comportamiento de la convergencia de la secuencia potencial de una matriz. Para cualquier

$x^0 \in \mathbb{R}^n$ la sucesión $\{x^k\}_{k \geq 0}$ definida por $x^k = Lx^{k-1} + c \forall k \geq 1$, siendo L un operador, converge a una solución única solamente si $\rho(L) < 1$. Para determinar la estabilidad del sistema, obtenido tras la aplicación de los diferentes métodos de integración estudiados, es preciso establecer algún tipo de estrategia que permita poder obtener una relación que exprese el valor de \mathbf{r}_n en función \mathbf{r}_0 :

$$\mathbf{r}_n = \mathbf{G}_n \mathbf{r}_0 + \mathbf{g}_n. \quad (3.24)$$

Considerando que $r_0 = r_{-1}$, toda vez que se supone que el objeto inicialmente está en reposo y sin que actúen sobre él fuerzas de ningún tipo, deberá encontrarse una expresión adecuada para la matriz \mathbf{G}_n y el vector \mathbf{g}_n en términos de los valores de las iteraciones anteriores y según cual sea el método de integración utilizado; es decir: definiendo una recurrencia. De esta forma se podrá expresar el valor de \mathbf{r}_n en función de los valores obtenidos para los pasos anteriores y de \mathbf{r}_0 .

Si es posible obtener los valores adecuados para \mathbf{G}_n y \mathbf{g}_n entonces la ecuación (3.24) indica que es posible representar los valores de cada instante a partir de los valores del instante inicial. Por lo tanto se desprende que la estabilidad del sistema se puede medir mediante el estudio del radio espectral de la matriz \mathbf{G}_n , $\rho(\mathbf{G}_n)$. Es decir, el sistema será estable si y solamente si $\lim_{n \rightarrow \infty} \rho(\mathbf{G}_n) < 1$.

En definitiva si el valor de \mathbf{G}_n se puede establecer en función de los valores de los pasos inmediatamente anteriores, entonces puede establecerse un mecanismo que permita determinar para cada iteración la estabilidad del sistema, así es posible comprobar la evolución de la estabilidad del sistema a lo largo de toda la simulación.

Estudio del modelo de las diferencias centradas

Según la ecuación (3.3) que describe este modelo, las operaciones que se pueden realizar se describen a continuación.

En primer lugar se definen

$$\begin{aligned} \mathbf{r}_n &:= \mathbf{r}(n\Delta t), \\ \mathbf{f}_n &:= \mathbf{f}(n\Delta t), \\ \mathbf{A} &:= \left(\frac{M}{\Delta t^2} + \frac{C}{2\Delta t} \right), \\ \mathbf{B}(\mathbf{r}_n) &:= \left(\frac{2\mathbf{M}}{\Delta t^2} - \mathbf{K}(\mathbf{r}_n) \right) \end{aligned}$$

y

$$\mathbf{D} := \left(\frac{C}{2\Delta t} - \frac{M}{\Delta t^2} \right),$$

de manera que la ecuación (3.3) puede escribirse

$$\mathbf{A}\mathbf{r}_{n+1} = \mathbf{f}_n + \mathbf{B}(\mathbf{r}_n)\mathbf{r}_n + \mathbf{D}\mathbf{r}_{n-1},$$

y por lo tanto

$$\mathbf{r}_{n+1} = \mathbf{A}^{-1}\mathbf{f}_n + \mathbf{A}^{-1}\mathbf{B}(\mathbf{r}_n)\mathbf{r}_n + \mathbf{A}^{-1}\mathbf{D}\mathbf{r}_{n-1},$$

o lo que es lo mismo

$$\mathbf{r}_n = \mathbf{A}^{-1}\mathbf{f}_{n-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{r}_{n-1})\mathbf{r}_{n-1} + \mathbf{A}^{-1}\mathbf{D}\mathbf{r}_{n-2},$$

de (3.24) se tiene

$$\begin{aligned}\mathbf{r}_{n-1} &= \mathbf{G}_{n-1}\mathbf{r}_0 + \mathbf{g}_{n-1}, \\ \mathbf{r}_{n-2} &= \mathbf{G}_{n-2}\mathbf{r}_0 + \mathbf{g}_{n-2}\end{aligned}$$

y, en consecuencia

$$\mathbf{r}_n = \mathbf{A}^{-1}\mathbf{f}_{n-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{r}_{n-1})(\mathbf{G}_{n-1}\mathbf{r}_0 + \mathbf{g}_{n-1}) + \mathbf{A}^{-1}\mathbf{D}(\mathbf{G}_{n-2}\mathbf{r}_0 + \mathbf{g}_{n-2}),$$

que convenientemente arreglado queda

$$\mathbf{r}_n = \mathbf{A}^{-1}(\mathbf{B}(\mathbf{r}_{n-1})\mathbf{G}_{n-1} + \mathbf{D}\mathbf{G}_{n-2})\mathbf{r}_0 + \mathbf{A}^{-1}(\mathbf{f}_{n-1} + \mathbf{B}(\mathbf{r}_{n-1})\mathbf{g}_{n-1} + \mathbf{D}\mathbf{g}_{n-2}),$$

entonces si se define

$$\mathbf{G}_n := \mathbf{A}^{-1}(\mathbf{B}(\mathbf{r}_{n-1})\mathbf{G}_{n-1} + \mathbf{D}\mathbf{G}_{n-2}), \quad (3.25a)$$

$$\mathbf{g}_n := \mathbf{A}^{-1}(\mathbf{f}_{n-1} + \mathbf{B}(\mathbf{r}_{n-1})\mathbf{g}_{n-1} + \mathbf{D}\mathbf{g}_{n-2}), \quad (3.25b)$$

se consigue obtener una expresión recurrente que permite expresar \mathbf{r}_n en función de \mathbf{r}_0 . Se puede considerar que \mathbf{G}_{-1} y \mathbf{G}_{-2} son la matriz identidad ya que se considera al objeto en reposo en el instante inicial.

Estudio para el método de las diferencias centradas implícito

Este modelo permite expresar la ecuación del movimiento (2.22) en la forma de la ecuación (3.4) que puede ser tratada de una forma semejante a como se ha

hecho en el apartado anterior, con la salvedad de que \mathbf{K} pasa a formar parte de la matriz del sistema. Por lo tanto, si se define

$$\mathbf{A} := \left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t} + \mathbf{K} \right),$$

$$\mathbf{B} := \left(\frac{2\mathbf{M}}{\Delta t^2} \right),$$

y

$$\mathbf{D} := \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2} \right),$$

se obtendrán unas expresiones para \mathbf{G}_n y \mathbf{g}_n equivalentes a las mostradas en (3.25a) y (3.25b) respectivamente.

Estudio para el método de Houbolt

Con el método de Houbolt se obtiene una expresión para la ecuación del movimiento como la mostrada en la ecuación (3.7). Para este método se definen

$$\mathbf{A} := \left(\frac{2\mathbf{M}}{\Delta t^2} + \frac{11\mathbf{C}}{6\Delta t} + \mathbf{K} \right),$$

$$\mathbf{B} := \left(\frac{5\mathbf{M}}{\Delta t^2} + \frac{3\mathbf{C}}{\Delta t} \right),$$

$$\mathbf{D} := - \left(\frac{4\mathbf{M}}{\Delta t^2} + \frac{3\mathbf{C}}{2\Delta t} \right),$$

y

$$\mathbf{E} := \left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{3\Delta t} \right),$$

de manera que de la ecuación (3.7) se obtiene la expresión

$$\mathbf{r}_{n+1} = \mathbf{A}^{-1}\mathbf{f} + \mathbf{A}^{-1}(\mathbf{B}\mathbf{r}_n + \mathbf{D}\mathbf{r}_{n-1} + \mathbf{E}\mathbf{r}_{n-2}),$$

por lo tanto

$$\mathbf{r}_n = \mathbf{A}^{-1}\mathbf{f} + \mathbf{A}^{-1}(\mathbf{B}\mathbf{r}_{n-1} + \mathbf{D}\mathbf{r}_{n-2} + \mathbf{E}\mathbf{r}_{n-3}),$$

si se expresa \mathbf{r}_{n-1} , \mathbf{r}_{n-2} y \mathbf{r}_{n-3} en la forma de (3.24)

$$\begin{aligned} \mathbf{r}_{n-1} &= \mathbf{G}_{n-1}\mathbf{r}_0 + \mathbf{g}_{n-1}, \\ \mathbf{r}_{n-2} &= \mathbf{G}_{n-2}\mathbf{r}_0 + \mathbf{g}_{n-2}, \\ \mathbf{r}_{n-3} &= \mathbf{G}_{n-3}\mathbf{r}_0 + \mathbf{g}_{n-3}, \end{aligned}$$

se puede escribir

$$\mathbf{r}_n = \mathbf{A}^{-1} \mathbf{f} + \mathbf{A}^{-1} (\mathbf{B}(\mathbf{G}_{n-1} \mathbf{r}_0 + \mathbf{g}_{n-1}) + \mathbf{D}(\mathbf{G}_{n-2} \mathbf{r}_0 + \mathbf{g}_{n-2}) + \mathbf{E}(\mathbf{G}_{n-3} \mathbf{r}_0 + \mathbf{g}_{n-3})),$$

que convenientemente arreglada se convierte en

$$\mathbf{r}_n = \mathbf{A}^{-1} (\mathbf{B}\mathbf{G}_{n-1} + \mathbf{D}\mathbf{G}_{n-2} + \mathbf{E}\mathbf{G}_{n-3}) \mathbf{r}_0 + \mathbf{A}^{-1} (\mathbf{f} + \mathbf{B}\mathbf{g}_{n-1} + \mathbf{D}\mathbf{g}_{n-2} + \mathbf{E}\mathbf{g}_{n-3}),$$

por lo tanto

$$\mathbf{G}_n = \mathbf{A}^{-1} (\mathbf{B}\mathbf{G}_{n-1} + \mathbf{D}\mathbf{G}_{n-2} + \mathbf{E}\mathbf{G}_{n-3}), \quad (3.26a)$$

$$\mathbf{g}_n = \mathbf{A}^{-1} (\mathbf{f} + \mathbf{B}\mathbf{g}_{n-1} + \mathbf{D}\mathbf{g}_{n-2} + \mathbf{E}\mathbf{g}_{n-3}), \quad (3.26b)$$

y en consecuencia para el método de Houbolt también se ha podido obtener una matriz y un vector expresados en forma recurrente que permiten expresar \mathbf{r}_n en función de \mathbf{r}_0 , teniendo presente que $\mathbf{G}_{-1} = \mathbf{G}_{-2} = \mathbf{G}_{-3} = I$, siendo I la matriz identidad.

Estudio para el método de Newmark β

Tal y como puede comprobarse en la ecuación (3.14), el método de Newmark β no solamente utiliza \mathbf{r} , sino que en la ecuación además aparecen términos relacionados con la velocidad $\dot{\mathbf{r}}$ y con la aceleración $\ddot{\mathbf{r}}$. Por ello no resulta adecuado un razonamiento como el seguido para los modelos anteriores.

Si se combina en un solo sistema de ecuaciones la ecuación (2.22) junto con (3.13) se obtienen

$$\begin{aligned} A\ddot{\mathbf{r}}_{n+1} + C\dot{\mathbf{r}}_{n+1} + K\mathbf{r}_{n+1} &= \mathbf{f}, \\ \dot{\mathbf{r}}_{n+1} - \dot{\mathbf{r}}_n - (1 - \gamma)\Delta t\ddot{\mathbf{r}}_n - \gamma\Delta t\ddot{\mathbf{r}}_{n+1} &= 0, \\ \mathbf{r}_{n+1} - \mathbf{r}_n - \Delta t\dot{\mathbf{r}}_n - \left(\frac{1}{2} - \beta\right)\Delta t^2\ddot{\mathbf{r}}_n - \beta\Delta t^2\ddot{\mathbf{r}}_{n+1} &= 0, \end{aligned}$$

que expresado de forma matricial quedan

$$\begin{pmatrix} A & C & K \\ -\gamma\Delta t I & I & 0 \\ -\beta\Delta t^2 I & 0 & I \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{r}}_{n+1} \\ \dot{\mathbf{r}}_{n+1} \\ \mathbf{r}_{n+1} \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ (1 - \gamma)\Delta t I & I & 0 \\ \left(\frac{1}{2} - \beta\right)\Delta t^2 I & \Delta t I & I \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{r}}_n \\ \dot{\mathbf{r}}_n \\ \mathbf{r}_n \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ 0 \end{pmatrix} \quad (3.27)$$

siendo I la matriz identidad. La expresión anterior para \mathbf{r}_n se puede expresar como

$$\tilde{A}\tilde{\mathbf{r}}_n - \tilde{B}\tilde{\mathbf{r}}_{n-1} = \tilde{\mathbf{f}}, \quad (3.28)$$

siendo

$$\tilde{\mathbf{r}}_i = \begin{pmatrix} \ddot{\mathbf{r}}_i \\ \dot{\mathbf{r}}_i \\ \mathbf{r}_i \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} A & C & K \\ -\gamma\Delta t I & I & 0 \\ -\beta\Delta t^2 I & 0 & I \end{pmatrix},$$

$$\tilde{\mathbf{f}} = \begin{pmatrix} \mathbf{f} \\ 0 \\ 0 \end{pmatrix} \text{ y } \tilde{B} = \begin{pmatrix} 0 & 0 & 0 \\ (1-\gamma)\Delta t I & I & 0 \\ (\frac{1}{2}-\beta)\Delta t^2 I & \Delta t I & I \end{pmatrix},$$

según lo cual

$$\tilde{\mathbf{r}}_n = \tilde{A}^{-1}\tilde{\mathbf{f}} + \tilde{A}^{-1}\tilde{B}\tilde{\mathbf{r}}_{n-1}. \quad (3.29)$$

Para poder establecer una relación de recurrencia, en los mismos términos que para los modelos anteriores, será preciso extender la ecuación (3.24) de manera que se contemple también la velocidad y la aceleración:

$$\tilde{\mathbf{r}}_n = \tilde{\mathbf{G}}_n\tilde{\mathbf{r}}_0 + \tilde{\mathbf{g}}_n, \quad (3.30)$$

de manera que se puede expresar (3.29) como

$$\tilde{\mathbf{r}}_n = \tilde{A}^{-1}\tilde{\mathbf{f}} + \tilde{A}^{-1}\tilde{B}(\tilde{\mathbf{G}}_{n-1}\tilde{\mathbf{r}}_0 + \tilde{\mathbf{g}}_{n-1}),$$

por lo tanto

$$\tilde{\mathbf{r}}_n = \tilde{A}^{-1}\tilde{\mathbf{f}} + \tilde{A}^{-1}\tilde{B}\tilde{\mathbf{g}}_{n-1} + \tilde{A}^{-1}\tilde{B}\tilde{\mathbf{G}}_{n-1}\tilde{\mathbf{r}}_0,$$

así que si se define

$$\tilde{\mathbf{G}}_n := \tilde{A}^{-1}\tilde{B}\tilde{\mathbf{G}}_{n-1}, \quad (3.31a)$$

$$\tilde{\mathbf{g}}_n := \tilde{A}^{-1}\tilde{\mathbf{f}} + \tilde{A}^{-1}\tilde{B}\tilde{\mathbf{g}}_{n-1}, \quad (3.31b)$$

se obtiene una expresión recurrente que permite expresar $\tilde{\mathbf{r}}_n$ en función de $\tilde{\mathbf{r}}_0$.

Estudio para el método de Wilson θ

Al igual que en el método de Newmark β , en el método de Wilson θ se debe tener en cuenta el valor de la aceleración y el de la velocidad en cada instante de tiempo para poder realizar los cálculos. Además, como se desprende de la ecuación (3.12), conocido el valor para un instante de tiempo t los valores se

calculan nuevamente para el instante de tiempo $t + \theta\Delta t$ para posteriormente calcular los valores deseados para $t + \Delta t$.

En primer lugar, de (3.8) y (3.9), fijando $\tau = \Delta t$ se obtiene

$$\begin{aligned}\ddot{\mathbf{r}}^{t+\Delta t} &= \frac{\theta-1}{\theta}\ddot{\mathbf{r}}^t + \frac{1}{\theta}\ddot{\mathbf{r}}^{t+\theta\Delta t}, \\ \dot{\mathbf{r}}^{t+\Delta t} &= \dot{\mathbf{r}}^t + \Delta t \left(1 - \frac{1}{2\theta}\right)\ddot{\mathbf{r}}^t + \frac{\Delta t}{2\theta}\dot{\mathbf{r}}^{t+\theta\Delta t}, \\ \mathbf{r}^{t+\Delta t} &= \mathbf{r}^t + \Delta t\dot{\mathbf{r}}^t + \Delta t^2 \left(\frac{3\theta-1}{6\theta}\right)\ddot{\mathbf{r}}^t + \frac{\Delta t^2}{6\theta}\dot{\mathbf{r}}^{t+\theta\Delta t},\end{aligned}$$

expresado en forma matricial

$$\underbrace{\begin{pmatrix} \ddot{\mathbf{r}}^{t+\Delta t} \\ \dot{\mathbf{r}}^{t+\Delta t} \\ \mathbf{r}^{t+\Delta t} \end{pmatrix}}_{\tilde{\mathbf{r}}^{t+\Delta t}} = \underbrace{\begin{pmatrix} \frac{1}{\theta}I & 0 & 0 \\ \frac{\Delta t}{2\theta}I & 0 & 0 \\ \frac{\Delta t^2}{6\theta}I & 0 & 0 \end{pmatrix}}_{\tilde{\mathbf{M}}} \underbrace{\begin{pmatrix} \ddot{\mathbf{r}}^{t+\theta\Delta t} \\ \dot{\mathbf{r}}^{t+\theta\Delta t} \\ \mathbf{r}^{t+\theta\Delta t} \end{pmatrix}}_{\tilde{\mathbf{r}}^{t+\theta\Delta t}} + \underbrace{\begin{pmatrix} \frac{\theta-1}{\theta}I & 0 & 0 \\ \Delta t \left(1 - \frac{1}{2\theta}\right)I & I & 0 \\ \Delta t^2 \frac{3\theta-1}{6\theta}I & \Delta t I & I \end{pmatrix}}_{\tilde{\mathbf{N}}} \underbrace{\begin{pmatrix} \ddot{\mathbf{r}}^t \\ \dot{\mathbf{r}}^t \\ \mathbf{r}^t \end{pmatrix}}_{\tilde{\mathbf{r}}^t} \quad (3.32)$$

es decir

$$\tilde{\mathbf{r}}^{t+\Delta t} = \tilde{\mathbf{M}}\tilde{\mathbf{r}}^{t+\theta\Delta t} + \tilde{\mathbf{N}}\tilde{\mathbf{r}}^t. \quad (3.33)$$

Para intentar obtener unos valores adecuados para \mathbf{G}_n y \mathbf{g}_n se puede proceder de forma semejante a como se ha hecho para el método de Newmark β , construyendo un sistema de ecuaciones en función de (3.10) y (3.11):

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{r}}^{t+\theta\Delta t} + \mathbf{C}\dot{\mathbf{r}}^{t+\theta\Delta t} + \mathbf{K}\mathbf{r}^{t+\theta\Delta t} &= \theta\mathbf{f}_r^{t+\theta\Delta t} + (1-\theta)\mathbf{f}_r^t, \\ \ddot{\mathbf{r}}^{t+\theta\Delta t} - \frac{6}{\theta^2\Delta t^2}\mathbf{r}^{t+\theta\Delta t} + \frac{6}{\theta^2\Delta t^2}\mathbf{r}^t + \frac{6}{\theta\Delta t}\dot{\mathbf{r}}^t + 2\ddot{\mathbf{r}}^t &= 0, \\ \dot{\mathbf{r}}^{t+\Delta t} - \frac{3}{\theta\Delta t}\mathbf{r}^{t+\theta\Delta t} + \frac{3}{\theta\Delta t}\mathbf{r}^t + 2\dot{\mathbf{r}}^t + \frac{\theta\Delta t}{2}\ddot{\mathbf{r}}^t &= 0,\end{aligned}$$

cuya representación matricial es

$$\underbrace{\begin{pmatrix} \mathbf{M} & \mathbf{C} & \mathbf{K} \\ I & 0 & \frac{-6}{\theta^2\Delta t^2}I \\ 0 & I & \frac{-3}{\theta\Delta t}I \end{pmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{pmatrix} \ddot{\mathbf{r}}^{t+\theta\Delta t} \\ \dot{\mathbf{r}}^{t+\theta\Delta t} \\ \mathbf{r}^{t+\theta\Delta t} \end{pmatrix}}_{\tilde{\mathbf{r}}^{t+\theta\Delta t}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 2I & \frac{6}{\theta\Delta t}I & \frac{6}{\theta^2\Delta t^2}I \\ \frac{\theta\Delta t}{2}I & 2I & \frac{3}{\theta\Delta t}I \end{pmatrix}}_{\tilde{\mathbf{B}}} \underbrace{\begin{pmatrix} \ddot{\mathbf{r}}^t \\ \dot{\mathbf{r}}^t \\ \mathbf{r}^t \end{pmatrix}}_{\tilde{\mathbf{r}}^t} = \underbrace{\begin{pmatrix} \theta\mathbf{f}_r^{t+\theta\Delta t} + (1-\theta)\mathbf{f}_r^t \\ 0 \\ 0 \end{pmatrix}}_{\tilde{\mathbf{f}}} \quad (3.34)$$

es decir

$$\tilde{\mathbf{A}}\tilde{\mathbf{r}}^{t+\theta\Delta t} + \tilde{\mathbf{B}}\tilde{\mathbf{r}}^t = \tilde{\mathbf{f}}. \quad (3.35)$$

Si en la expresión anterior se sustituye $\tilde{\mathbf{r}}^{t+\theta\Delta t}$ por su valor según (3.33) se obtiene

$$\tilde{\mathbf{r}}^{t+\theta\Delta t} = \tilde{\mathbf{A}}^{-1} \left(\tilde{\mathbf{f}} - \tilde{\mathbf{B}}\tilde{\mathbf{r}}^t \right), \quad (3.36)$$

y

$$\tilde{\mathbf{r}}^{t+\Delta t} = \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1} \left(\tilde{\mathbf{f}} - \tilde{\mathbf{B}}\tilde{\mathbf{r}}^t \right) + \tilde{\mathbf{N}}\tilde{\mathbf{r}}^t. \quad (3.37)$$

En general se puede obtener el valor para $\tilde{\mathbf{r}}^t$ a partir de la expresión anterior, y si además se sustituye el valor $\tilde{\mathbf{r}}^{t-\Delta t}$ por una expresión basada en $\tilde{\mathbf{r}}^0$

$$\tilde{\mathbf{r}}^{t-\Delta t} = \tilde{\mathbf{G}}^{t-\Delta t}\tilde{\mathbf{r}}^0 + \tilde{\mathbf{g}}^{t-\Delta t},$$

se obtiene

$$\tilde{\mathbf{r}}^t = \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{f}} + \left(\tilde{\mathbf{N}} - \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}} \right) \left(\tilde{\mathbf{G}}^{t-\Delta t}\tilde{\mathbf{r}}^0 + \tilde{\mathbf{g}}^{t-\Delta t} \right),$$

que se puede arreglar

$$\tilde{\mathbf{r}}^t = \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{f}} + \left(\tilde{\mathbf{N}} - \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}} \right) \tilde{\mathbf{g}}^{t-\Delta t} + \left(\tilde{\mathbf{N}} - \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}} \right) \left(\tilde{\mathbf{G}}^{t-\Delta t}\tilde{\mathbf{r}}^0 \right),$$

que una vez reescrito con la misma notación que la utilizada con los modelos anteriores queda

$$\tilde{\mathbf{r}}_n = \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{f}} + \left(\tilde{\mathbf{N}} - \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}} \right) \tilde{\mathbf{g}}_{n-1} + \left(\tilde{\mathbf{N}} - \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}} \right) \left(\tilde{\mathbf{G}}_{n-1}\tilde{\mathbf{r}}_0 \right),$$

así que si se define

$$\tilde{\mathbf{G}}_n = \left(\tilde{\mathbf{N}} - \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}} \right) \tilde{\mathbf{G}}_{n-1}, \quad (3.38a)$$

$$\tilde{\mathbf{g}}_n = \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{f}} + \left(\tilde{\mathbf{N}} - \tilde{\mathbf{M}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}} \right) \tilde{\mathbf{g}}_{n-1}, \quad (3.38b)$$

se concluye que con este método también es posible obtener una expresión recurrente que permita expresar $\tilde{\mathbf{r}}_n$ en función de $\tilde{\mathbf{r}}_0$.

Resultados

Una vez obtenidas las expresiones para calcular de forma recurrente el valor de los puntos en cada instante de tiempo considerado y para cada modelo, se dispone de un mecanismo que permite calcular la estabilidad de cada uno de los métodos estudiados, utilizando la estimación del radio espectral de la matriz \mathbf{G}_n en cada uno de los instantes de tiempo.

Otro valor que puede ser útil a la hora de determinar la bonanza de los métodos es el número de condición de la matriz \mathbf{G}_n , toda vez que el número de condición es una medida de la sensibilidad de la solución de un sistema de ecuaciones lineales a los errores presentes en los datos. Es decir, da una indicación de la precisión de los resultados de una matriz inversa y por lo tanto de la solución de un sistema de ecuaciones lineales. En definitiva, con el número de condición se podrá estimar el grado de exactitud de las soluciones que se obtengan a lo largo del tiempo.

Para comparar los diferentes métodos se han realizado tres simulaciones diferentes sobre un mismo objeto. Las características comunes a las tres simulaciones son:

- el objeto a deformar es un cuadrado unitario, plano,
- la discretización espacial del objeto se ha realizado de manera que éste se divide en 11 filas y 11 columnas de puntos,
- cada punto tiene un coeficiente de amortiguación y una masa unitarios,
- las fuerzas aplicadas corresponden a un campo gravitatorio normal a la superficie del objeto y que afecta a todo el objeto exceptuando aquellos puntos que hayan sido fijados,
- el campo empieza a actuar en el instante $t = 0s$; anteriormente no se aplica ningún tipo de fuerzas al objeto y por lo tanto éste permanece en reposo con velocidad y aceleración nulas.
- el tiempo total simulado es de 2 segundos.

Para poder determinar en qué medida afecta el incremento de tiempo en cada uno de los métodos cada una de las simulaciones se ha realizado tomando en consideración dos valores diferentes para Δt . En concreto se ha utilizado los valores $\Delta t = 0.01s$ y $\Delta t = 0.001s$.

Las simulaciones realizadas han sido las siguientes:

- 1) Deformación del objeto sujeto por sus cuatro esquinas con un coeficiente $\eta_{ij} = 1.0$ y $\xi_{ij} = 0.0 \forall i, j \in \{1, 2\}$. Los resultados pueden verse en la tabla 3.2 para $\Delta t = 0.01$ y 3.3 para $\Delta t = 0.001$.

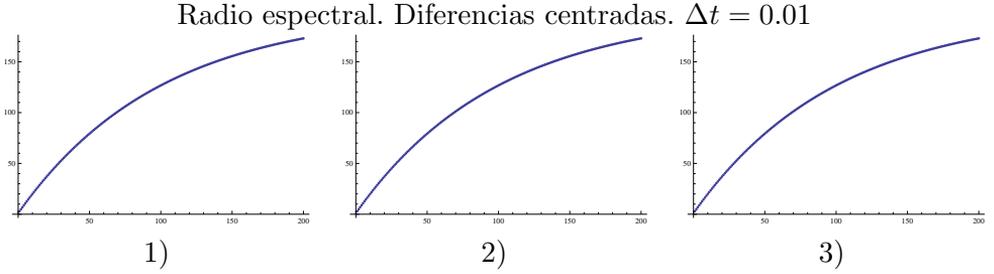


Figura 3.5: Gráficas del radio espectral en los experimentos basados en el método de las diferencias centradas con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n .

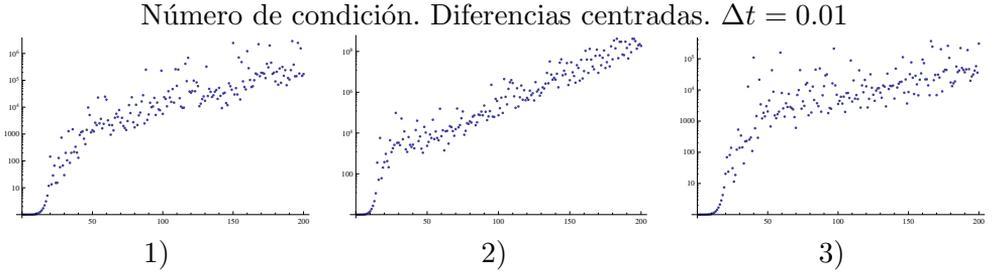


Figura 3.6: Gráficas del logaritmo del número de condición en los experimentos basados en el método de las diferencias centradas con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del número de condición de la matriz G_n .

- 2) Deformación del objeto sujeto por todo su contorno también con unos coeficientes $\eta_{ij} = 1.0$ y $\xi_{ij} = 0.0 \forall i, j \in \{1, 2\}$. Los resultados pueden verse en la tabla 3.4 para $\Delta t = 0.01$ y 3.5 para $\Delta t = 0.001$.
- 3) Deformación del objeto sujeto por todo su contorno con un coeficiente $\eta_{ij} = 10.0$ y $\xi_{ij} = 0.0 \forall i, j \in \{1, 2\}$. Los resultados pueden verse en la tabla 3.6 para $\Delta t = 0.01$ y 3.7 para $\Delta t = 0.001$.

Es decir, se ha intentado deformar un objeto comprobando su comportamiento en función de lo *sujeto* que esté (comparación de las simulaciones 1 y 2) y en función de su rigidez (comparación de las simulaciones 2 y 3).

En las figuras, desde la 3.5 hasta la 3.22, se muestra la evolución del radio espectral y del número de condición para cada uno de los métodos considerados y cada una de las tres simulaciones previamente descritas.

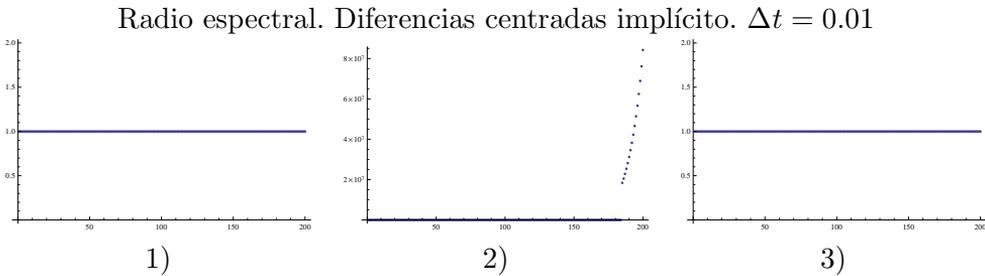


Figura 3.7: Gráficas del radio espectral en los experimentos basados en el método de las diferencias centradas implícito con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n .

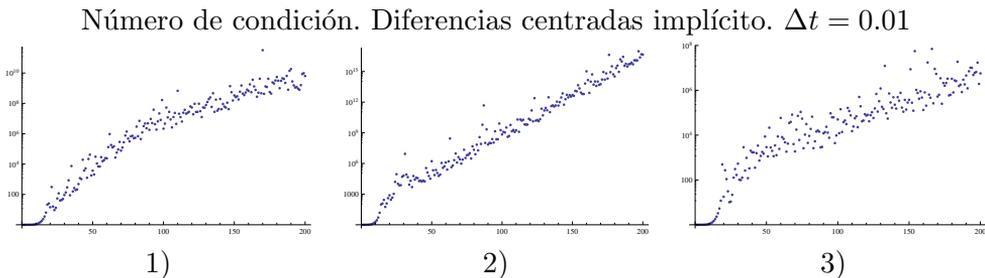


Figura 3.8: Gráficas del logaritmo del número de condición en los experimentos basados en el método de las diferencias centradas implícito con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del número de condición de la matriz G_n .

Como puede verse en la figura 3.5 el radio espectral para el método de las diferencias centradas no cambia en los tres experimentos. Esto es así debido a que el coeficiente de amortiguación y la masa no varían entre los diferentes experimentos, y al ser éstos los únicos elementos que forman la matriz \mathbf{G}_n , ésta no cambiará. El número de condición crece a medida que la simulación avanza en el tiempo, como puede verse en la figura 3.6, aun así los resultados obtenidos son adecuados.

Por otra parte, como se observa en la figura 3.7, el radio espectral para el método de las diferencias centradas implícito se mantiene prácticamente constante a lo largo de todo el tiempo de simulación, con lo que parece que este método es adecuado para realizar este tipo de simulaciones. Sin embargo el número de condición indica que los resultados obtenidos pueden divergir del resultado ideal. Aún así los valores obtenidos parecen aceptables.

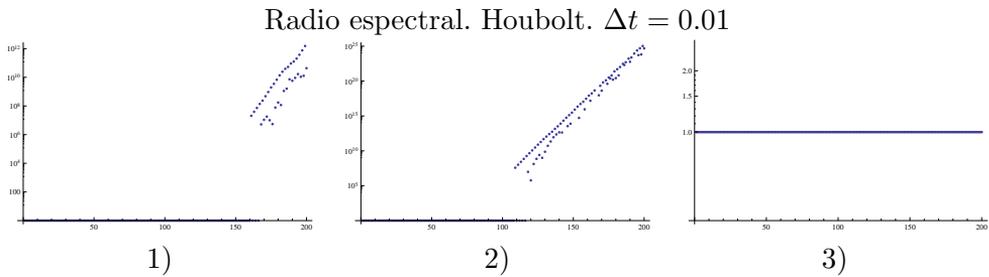


Figura 3.9: Gráficas del logaritmo del radio espectral en los experimentos basados en el método de Houbolt con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del radio espectral de la matriz G_n .

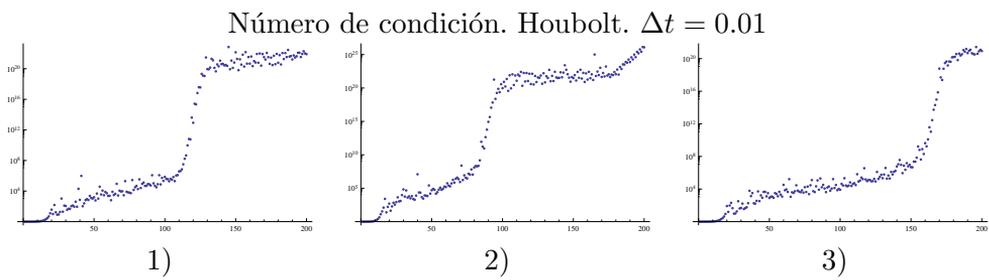


Figura 3.10: Gráficas del logaritmo del número de condición en los experimentos basados en el método de Houbolt con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del número de condición de la matriz G_n .

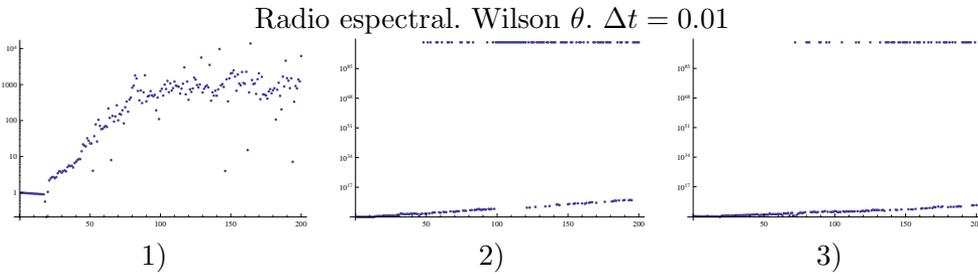


Figura 3.11: Gráficas del logaritmo del radio espectral en los experimentos basados en el método de Wilson θ con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del radio espectral de la matriz G_n .

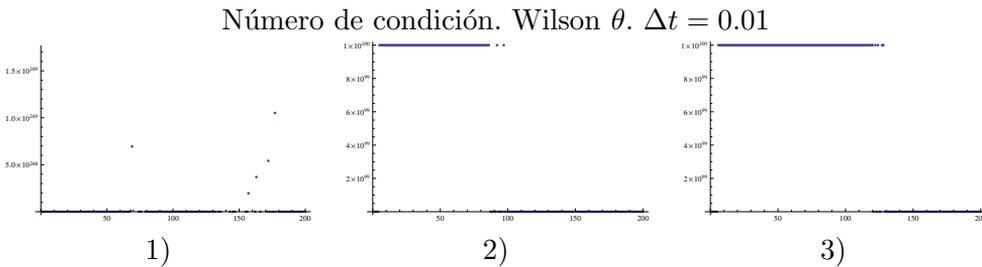


Figura 3.12: Gráficas del número de condición en los experimentos basados en el método de Wilson θ con $\Delta t = 0.01s$. En el eje horizontal, se representa la iteración n , y en el vertical, el número de condición de la matriz G_n .

Para la simulación con el método de Houbolt se obtienen unos valores para el radio espectral y el número de condición, figuras 3.9 y 3.10 respectivamente, que muestran claramente que este método no resulta adecuado para este tipo de simulación ya que el radio espectral crece linealmente mientras que el número de condición parece que llega a acotarse, aunque su valor es considerable.

Con el método de Wilson θ se obtiene un número de condición próximo a 1.0, figura 3.12, sin embargo el radio espectral 3.11 señala claramente que este método tampoco parece adecuado.

El último de los métodos estudiados, Newmark β , muestra un comportamiento similar al método de Wilson θ , con un número de condición razonable pero con un radio espectral sin acotar, figuras 3.13 y 3.14.

Para la simulación con $\Delta t = 0.001s$ el modelo de la diferencias centradas no muestra un comportamiento diferente al observado para $\Delta t = 0.01s$, como se constata al comparar las gráficas para este caso, figura 3.15 y figura 3.16, con las

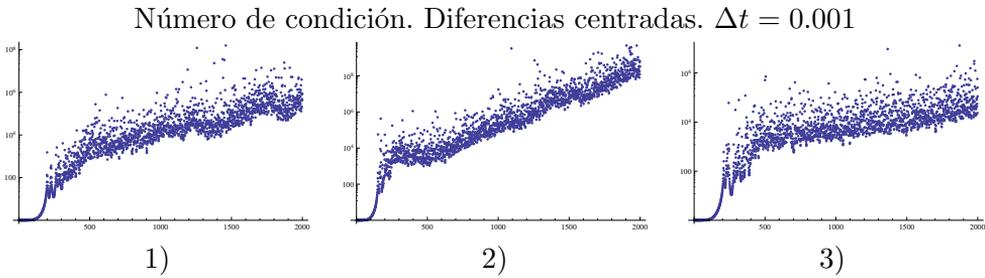


Figura 3.16: Gráficas del logaritmo del número de condición en los experimentos basados en el método de las diferencias centradas con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del número de condición de la matriz G_n .

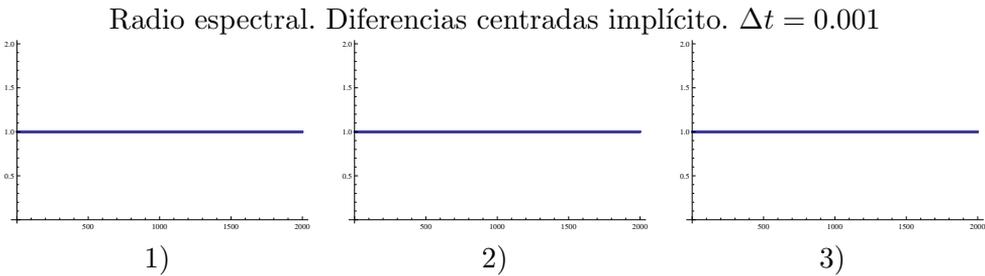


Figura 3.17: Gráficas del radio espectral en los experimentos basados en el método de las diferencias centradas implícito con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n .

anteriores, figura 3.5 y figura 3.6. Según esto no es necesario un incremento de tiempo tan pequeño para realizar simulaciones con éste método de integración.

En la figura 3.17 se muestra que para el método de integración de las diferencias centradas implícito el radio espectral se mantiene aun más cercano a 1 para un valor de $\Delta t = 0.001$, mientras que el número de condición se mantiene (véase la figura 3.18). Por lo tanto este método de integración puede ser utilizado con valores de Δt grandes o pequeños en función de la necesidad de precisión que se requiera en cada caso.

El método de Houbolt mejora sensiblemente con la aplicación de un incremento de tiempo menor, $\Delta t = 0.001s$. Sin embargo, como puede observarse en las figuras 3.19 y 3.20, el radio espectral obtenido sigue sin estar acotado y el número de condición sigue siendo elevado. Todo ello hace pensar que este método no ofrecería buenos resultados en caso de ser utilizado para simular largos periodos

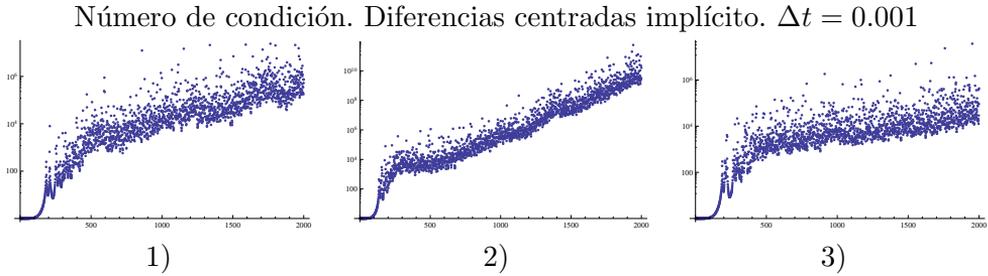


Figura 3.18: Gráficas del logaritmo del número de condición en los experimentos basados en el método de las diferencias centradas implícito con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del número de condición de la matriz G_n .

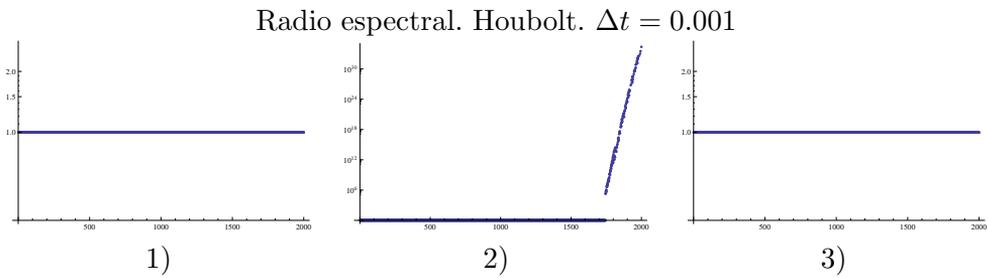


Figura 3.19: Gráficas del logaritmo del radio espectral en los experimentos basados en el método de Houbolt con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del radio espectral de la matriz G_n .

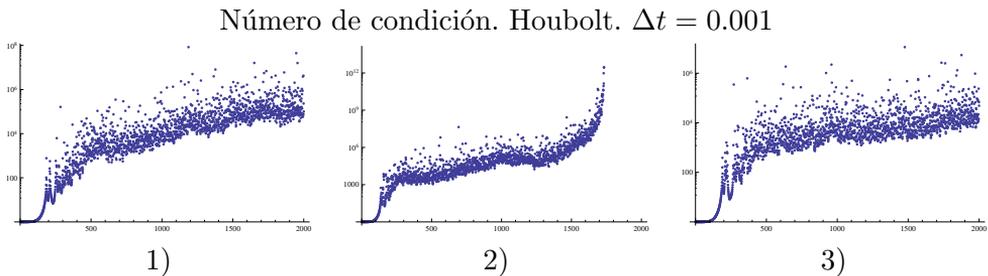


Figura 3.20: Gráficas del logaritmo del número de condición en los experimentos basados en el método de Houbolt con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del número de condición de la matriz G_n .

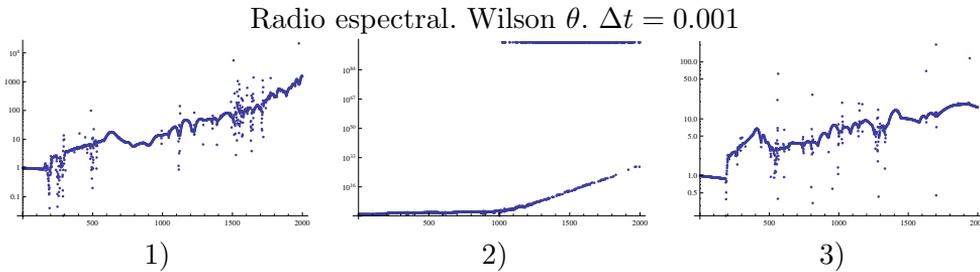


Figura 3.21: Gráficas del logaritmo del radio espectral en los experimentos basados en el método de Wilson θ con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del radio espectral de la matriz G_n .

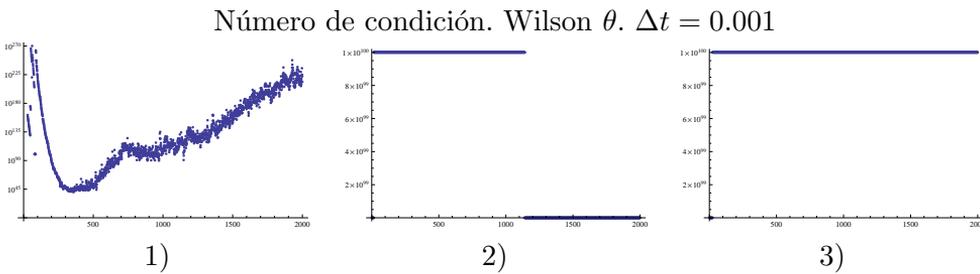


Figura 3.22: Gráficas del número de condición en los experimentos basados en el método de Wilson θ con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el número de condición de la matriz G_n . Para 1) el valor representado es del logaritmo del número de condición.

de tiempo.

El método de Wilson θ también ofrece mejores resultados con valores Δt pequeños, pero como en el caso del método de Houbolt no ofrece garantías de una estabilidad para simulaciones largas. Las figuras 3.21 y 3.22 justifican esta conclusión.

Para el método de Newmark β el comportamiento es análogo al de los métodos de Houbolt y de Wilson θ . El radio espectral obtenido, figura 3.23, indica poca estabilidad, aunque el número de condición, figura 3.24, se mantiene muy cercano a 1 o no crece indefinidamente.

De los experimentos realizados para $\Delta = 0.001s$ se puede concluir que los resultados obtenidos son mejores que cuando se considera $\Delta t = 0.01s$. Sin embargo, si el intervalo de tiempo a simular es considerable entonces los resultados serán aceptables solamente en el caso de que el método de integración utilizado sea el

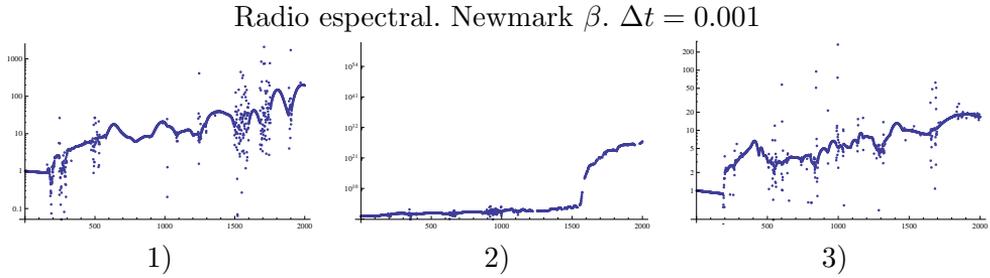


Figura 3.23: Gráficas del logaritmo del radio espectral en los experimentos basados en el método de Newmark β con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el logaritmo del radio espectral de la matriz G_n .

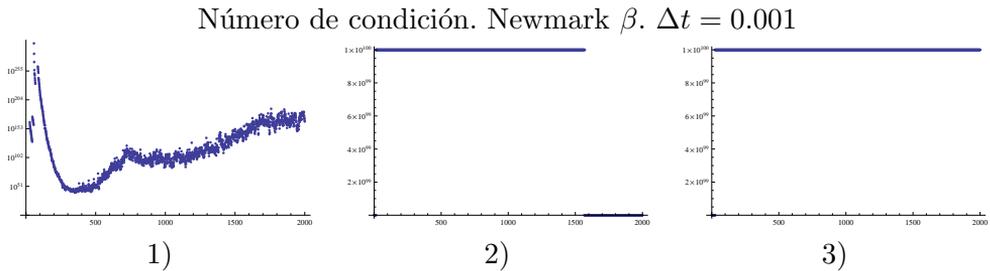


Figura 3.24: Gráficas del número de condición en los experimentos basados en el método de Newmark β con $\Delta t = 0.001s$. En el eje horizontal, se representa la iteración n , y en el vertical, el número de condición de la matriz G_n . Para 1) el valor representado es el del logaritmo del número de condición.

de las diferencias centradas o el de las diferencias centradas implícito. Se puede llegar a esta conclusión como resultado de la observación de las figuras 3.15, 3.17, 3.19, 3.21 y 3.23, en donde se comprueba que los radios espectrales obtenidos están acotados solamente en el caso del método de las diferencias centradas y en el de las diferencias centradas implícito.

Influencia de la discretización espacial

Parece extraño que los métodos de integración que *a priori* se postulan como los más adecuados cuando son aplicados resulten muy poco efectivos. El problema no puede hallarse en los modelos en sí, en el apartado 3.2.1 se ha constatado que se ajustan lo suficientemente bien al modelo teórico, sino que dicho problema debe estar en algún otro factor. Ahora bien, la única otra cosa que puede afectar al comportamiento del modelo es la discretización espacial, descrita en el apartado 3.3.4.

Para comprobar hasta qué punto afecta la discretización espacial al comportamiento de los métodos de integración se ha procedido a realizar un experimento en el que se ha realizado la misma simulación cambiando el objeto tratado y la discretización espacial. Así pues se ha procedido a realizar los experimentos con un objeto cuadrado plano de dimensión 1.0×1.0 discretizado con $M = \{16, 11, 6\}$, otro cuadrado plano de dimensión 4.0×4.0 con $M = 5$, otro más de dimensión 10.0×10.0 con $M = 8$ y, finalmente, un cuadrado plano de dimensión 20.0×20.0 con $M = 11$, por lo tanto las distancias entre dos puntos vecinos serán respectivamente $\{0.06667, 0.1000, 0.2000, 1.000, 1.429, 2.000\}$. Los datos del experimento, exceptuando los ya indicados, son los mismos que los ya citados para la simulación del objeto sujeto por las cuatro esquinas y $\Delta t = 0.01$. Los diferentes resultados pueden verse en las tablas 3.8, 3.9, 3.10, 3.11, 3.12 y 3.13.

Como puede observarse en la figura 3.25, el radio espectral para el método de las diferencias centradas no cambia significativamente en ninguno de los casos, exceptuando el caso en que la distancia entre puntos es muy pequeña. En definitiva, el comportamiento del método de las diferencias centradas no mejora su comportamiento sea cual sea la discretización espacial que se haya aplicado y solamente para distancias muy pequeñas no se obtienen resultados aceptables. En sí mismo, esto no es un problema, ya que los resultados obtenidos con este método se encuentran entre los mejores, además de ser el más rápido.

Para el método de las diferencias centradas implícito, figura 3.26, los resultados obtenidos son muy buenos, el radio espectral se mantiene constante y aunque el número de condición crece el resultado de las simulaciones siempre es aceptable.

El método de Houbolt, figura 3.27, tiene un comportamiento tal que el radio

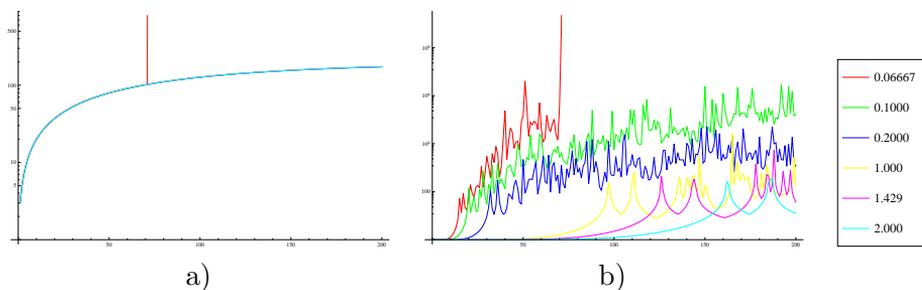


Figura 3.25: Gráficas de los logaritmos del a) radio espectral y b) el número de condición para los experimentos basados en el método de las diferencias centradas. Se han superpuesto los resultados para las diferentes distancias entre puntos.

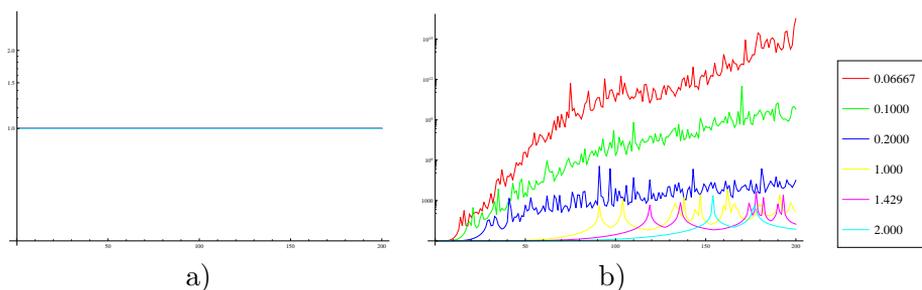


Figura 3.26: Gráficas de los logaritmos del a) radio espectral y b) el número de condición para los experimentos basados en el método de las diferencias centradas implícito. Se han superpuesto los resultados para las diferentes distancias entre puntos.

espectral crece linealmente y de forma indefinida a medida que avanza la simulación. Por lo tanto, con estos métodos no se obtiene ninguna mejora sea cual sea la forma en que se realiza la discretización espacial. Con los resultados obtenidos para el método de Houbolt se puede aseverar que dicho método no resulta adecuado para este tipo de simulaciones.

De la observación de las figuras 3.28 y 3.29 se puede destacar que tanto el método de Wilson θ como el de Newmark β mejoran mucho sus resultados cuando las distancias entre puntos aumentan.

De todo ello se puede concluir que el método de integración más adecuado para ser utilizado conjuntamente con un esquema de discretización espacial basado en diferencias, de la forma descrita en el apartado 3.3.4, es el método de las diferencias centradas implícito, ya que su comportamiento es el mejor con independencia de la distancia entre puntos y el incremento de tiempo utilizado.

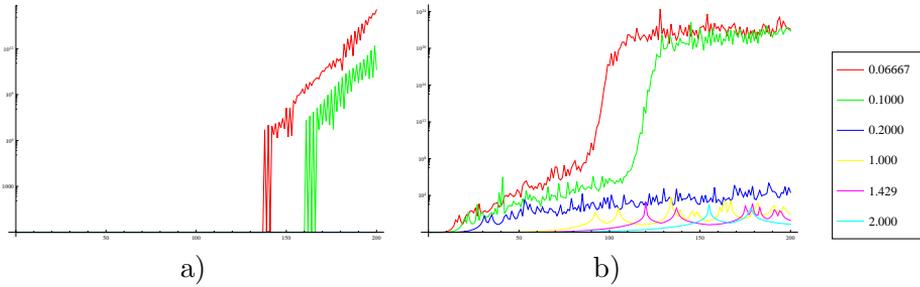


Figura 3.27: Gráficas de los logaritmos del a) radio espectral y b) el número de condición para los experimentos basados en el método de Houbolt. Se han superpuesto los resultados para las diferentes distancias entre puntos.

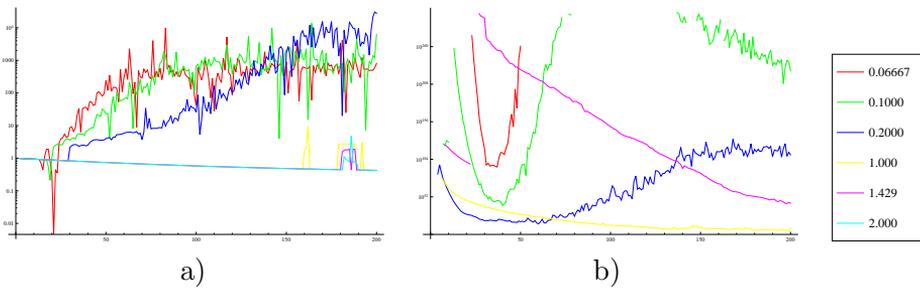


Figura 3.28: Gráficas de los logaritmos del a) radio espectral y b) el número de condición para los experimentos basados en el método de Wilson θ . Se han superpuesto los resultados para las diferentes distancias entre puntos.

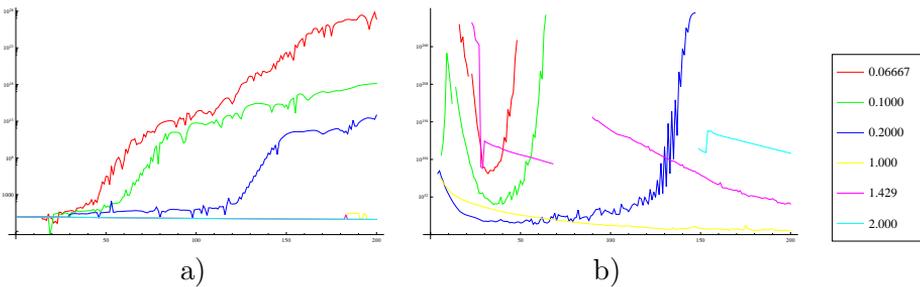


Figura 3.29: Gráficas de los logaritmos del a) radio espectral y b) el número de condición para los experimentos basados en el método de Newmark β . Se han superpuesto los resultados para las diferentes distancias entre puntos.

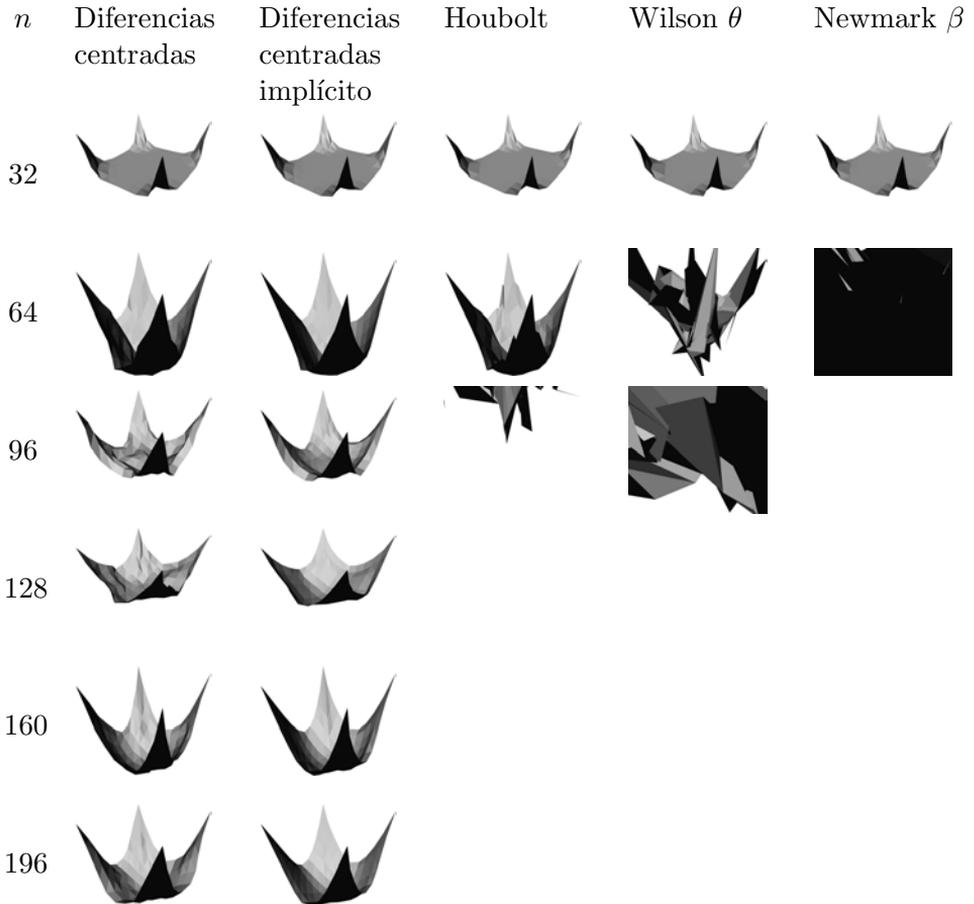


Tabla 3.2: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, resultados para el caso 1). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.3: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, resultados para el caso 1) En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

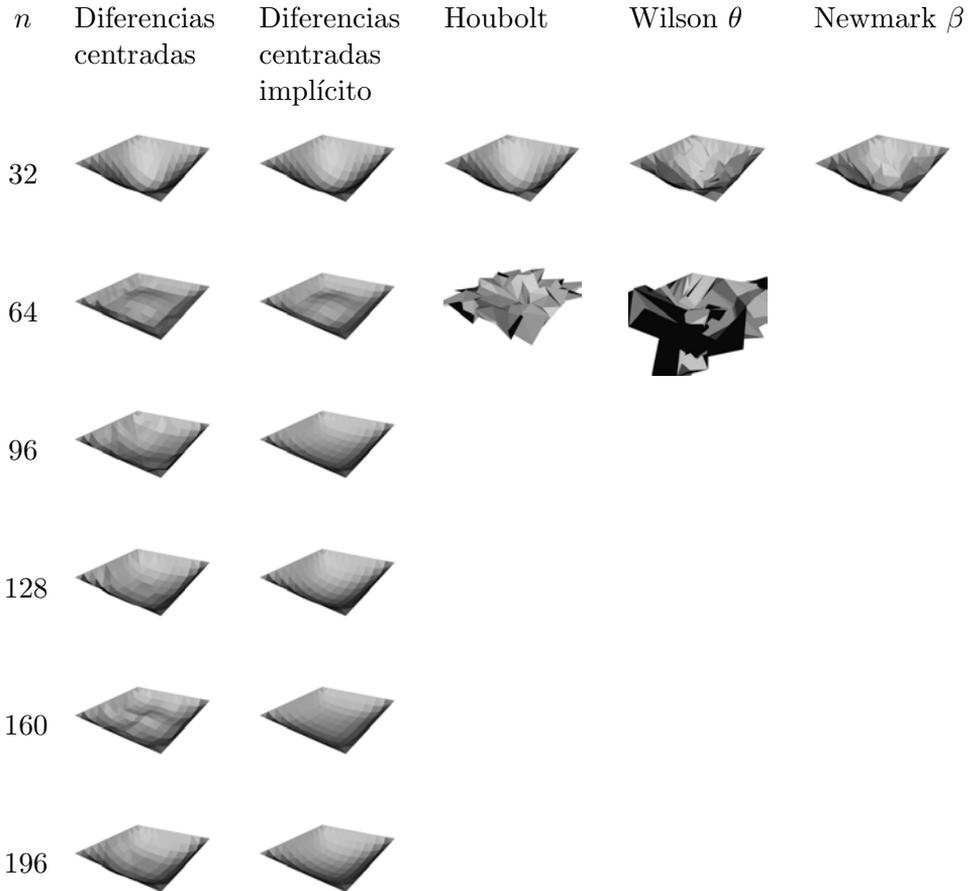


Tabla 3.4: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, resultados para el caso 2) En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

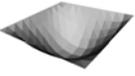
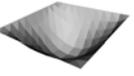
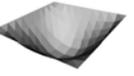
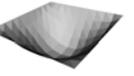
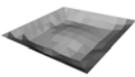
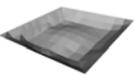
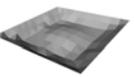
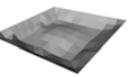
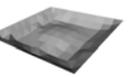
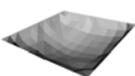
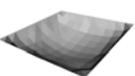
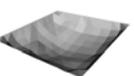
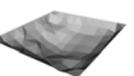
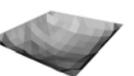
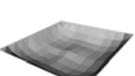
n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.5: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, resultados para el caso 2). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

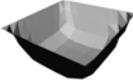
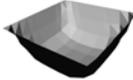
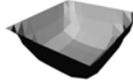
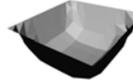
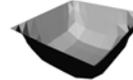
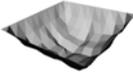
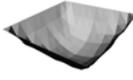
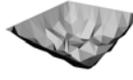
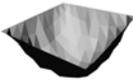
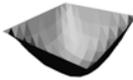
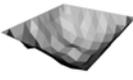
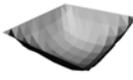
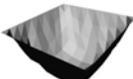
n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.6: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, resultados para el caso 3). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

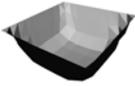
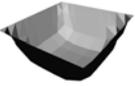
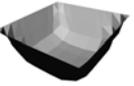
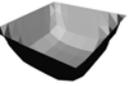
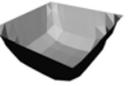
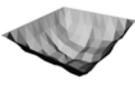
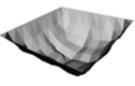
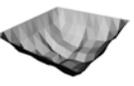
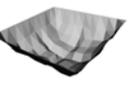
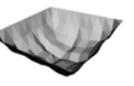
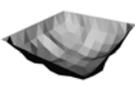
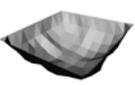
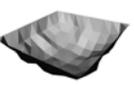
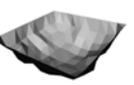
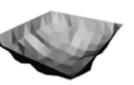
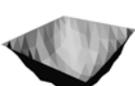
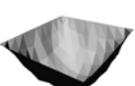
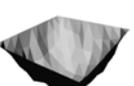
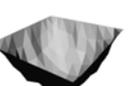
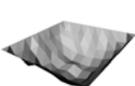
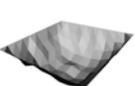
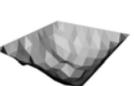
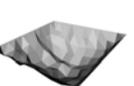
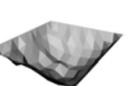
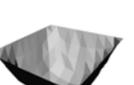
n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.7: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, resultados para el caso 3). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

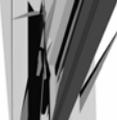
n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.8: Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 0.06667. En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.9: Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 0.100. En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.10: Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 0.200. En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$, si los resultados son excesivamente erróneos no se muestra la correspondiente imagen.

n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.11: Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 1.000.

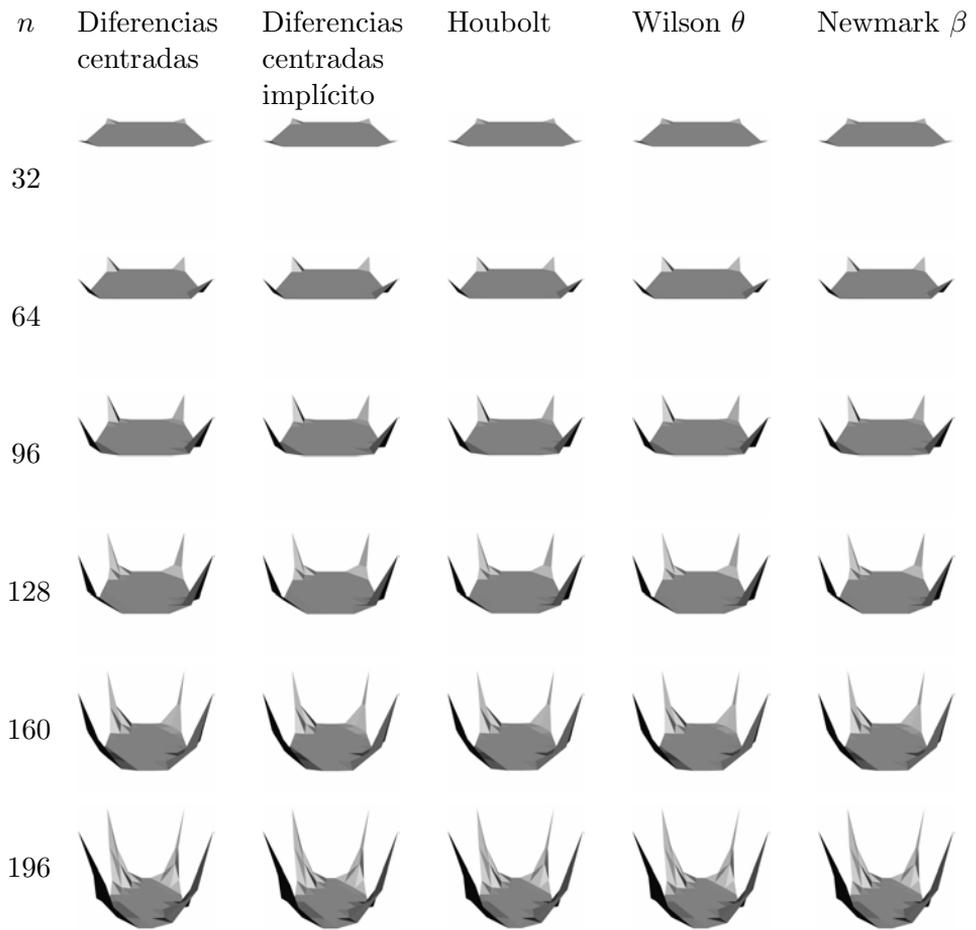


Tabla 3.12: Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 1.429.

n	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
32					
64					
96					
128					
160					
196					

Tabla 3.13: Evolución de las simulaciones para los diferentes modelos con la distancia entre puntos de 2.0.

Para probar la forma en que influye la discretización espacial en los resultados se debe comprobar también la forma en que se calcula dicha discretización.

Tal y como se describe en la sección 3.3.4 se han definido dos conjuntos de operadores en diferencias finitas, y también se han definido dos condiciones de contorno, con lo que existen cuatro posibles formas de calcular la discretización espacial.

Para determinar la adecuación de cada una de las combinaciones se han realizado dos tipos de experimentos. En ambos casos se han aplicado los cinco modelos de discretización temporal estudiados y se han utilizado modelos con diferentes distancias entre puntos. De esta forma se ha podido estudiar la mayor o menor adecuación de las cuatro formas planteadas para la discretización espacial.

Los experimentos realizados han consistido en la deformación de un plano con un contorno fijo y el mismo plano con solamente las cuatro esquinas fijas. Se ha realizado una simulación para un tiempo total T de 2 segundos con un incremento de tiempo de 0.01 segundos. En las figuras 3.30, 3.31, 3.32, 3.33 y 3.34 se muestran los resultados obtenidos para el experimento del plano con el contorno fijo y en las figuras 3.35, 3.36, 3.37, 3.38 y 3.39 los correspondientes para el caso del plano sujeto por las cuatro esquinas.

Se han definido tres variantes de cada experimento definidas sobre un cuadrado unitario, discretizado en M filas y en M columnas, siendo $M = \{16, 11, 6\}$, lo que ha determinado una distancia entre puntos de 0.0666, 0.1 y 0.2 respectivamente. Además se ha realizado una cuarta variante, esta vez sobre un cuadrado 10×10 con $M = 11$, determinándose así una distancia entre puntos de 1.0. En todos los casos se ha utilizado una masa y un coeficiente de amortiguación de 1.0 unidad en cada punto. La resistencia a la deformación horizontal (η_{11}), vertical (η_{22}) y de cizalladura (η_{12}) también tienen como valor 1 en cada punto en el caso del experimento en el que se sujeta el plano por sus cuatro esquinas y de 10.0 unidades para el experimento en el que el plano permanece sujeto en todo su contorno. En las figuras se muestra la evolución del radio espectral a lo largo de las simulaciones. En cada figura se muestra conjuntamente el comportamiento del radio espectral para cada una de las variantes.

Del estudio de los resultados de los experimentos se puede confirmar que la estabilidad de todos los métodos de integración mejora cuanto mayor es la distancia entre puntos, siendo esta mejora muy significativa si dicha distancia es mayor o igual a 1.

Otra conclusión a la que se llega es que el método de discretización espacial basado en diferencias centradas ofrece menor estabilidad que el método basado en diferencias no centradas. Además las condiciones de contorno libres, *naturales*,

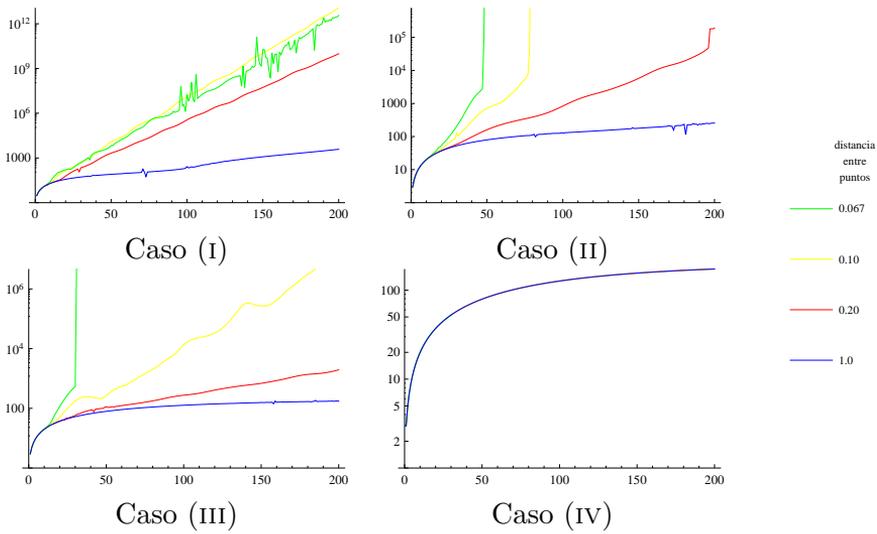


Figura 3.30: Gráficas del radio espectral en los experimentos en donde se fija el contorno. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de las diferencias centradas. En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

3.3. Estudio de los métodos de integración en un caso complejo

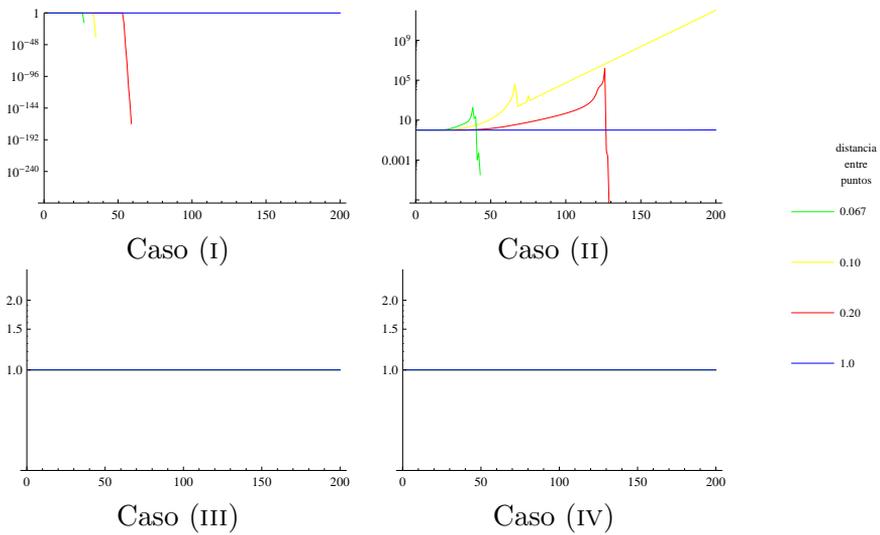


Figura 3.31: Gráficas del radio espectral en los experimentos en donde se fija el contorno. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método las diferencias centradas implícito. En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

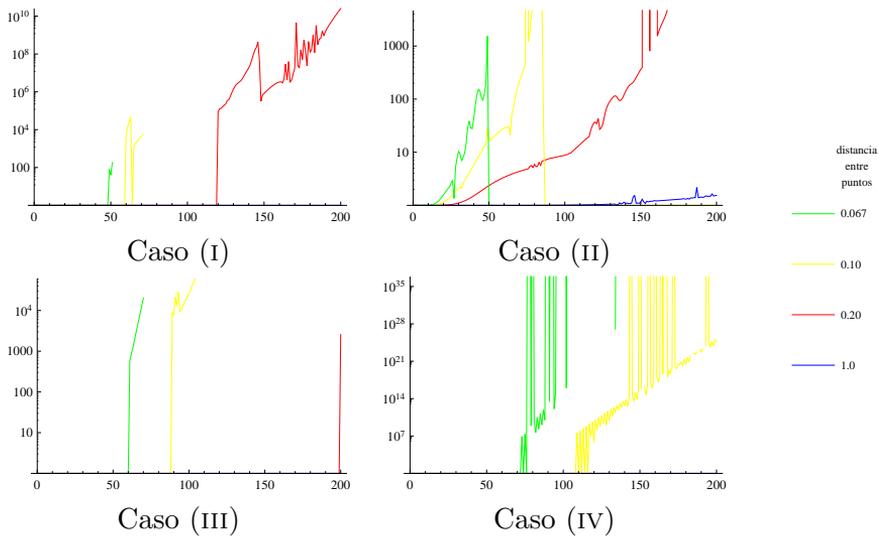


Figura 3.32: Gráficas del radio espectral en los experimentos en donde se fija el contorno. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de Houbolt. En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

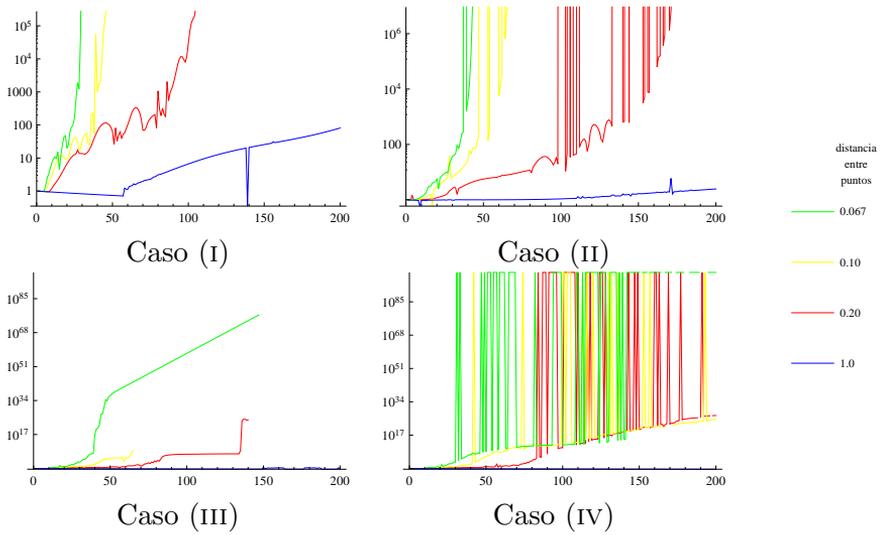


Figura 3.33: Gráficas del radio espectral en los experimentos en donde se fija el contorno. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de Newmark β . En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

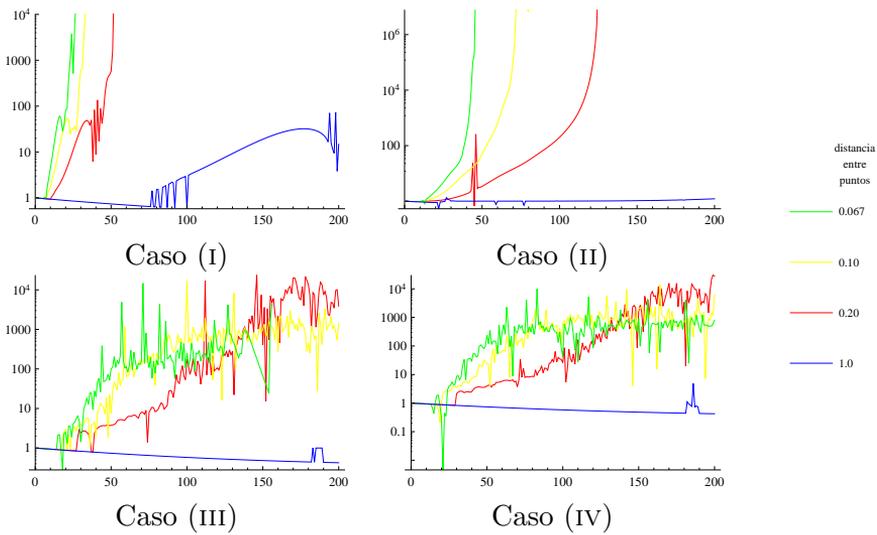


Figura 3.34: Gráficas del radio espectral en los experimentos en donde se fija el contorno. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de Wilson θ . En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

3.3. Estudio de los métodos de integración en un caso complejo

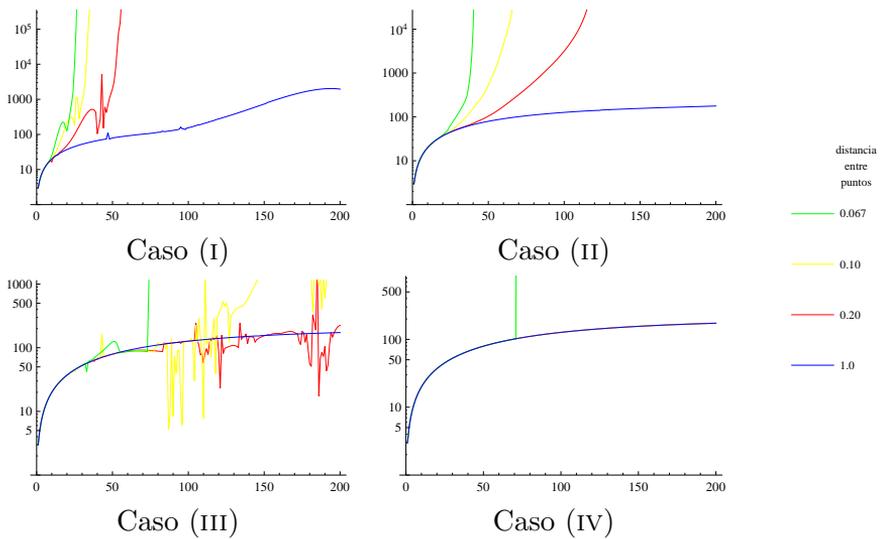


Figura 3.35: Gráficas del radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de las diferencias centradas. En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

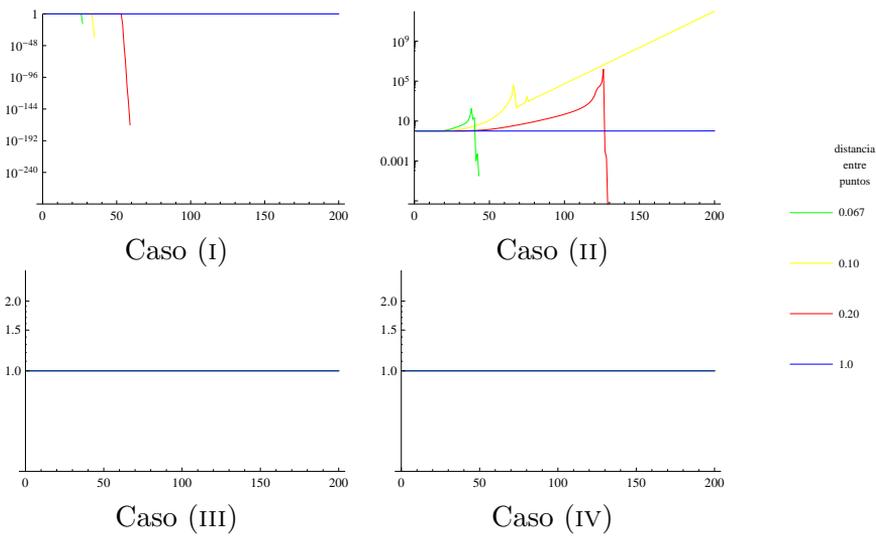


Figura 3.36: Gráficas del radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de las diferencias centradas implícito. En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

3.3. Estudio de los métodos de integración en un caso complejo

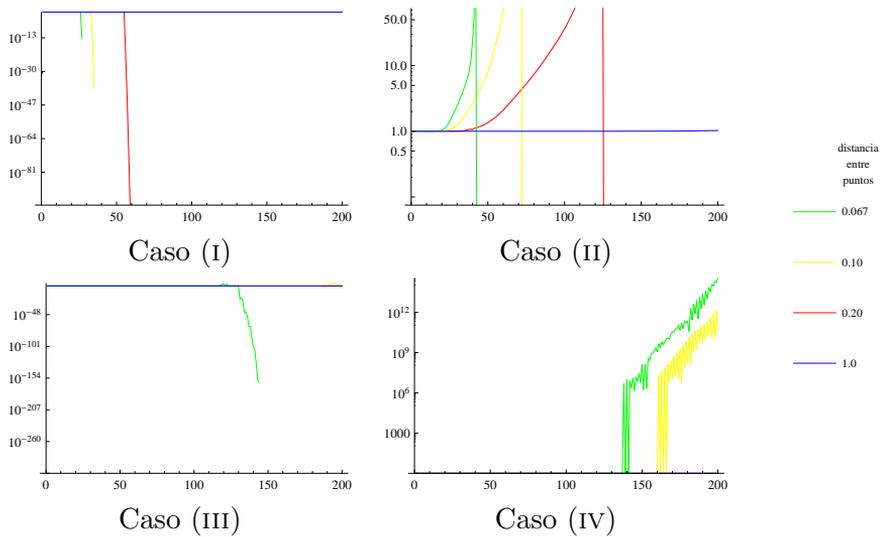


Figura 3.37: Gráficas del radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de Houbolt. En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

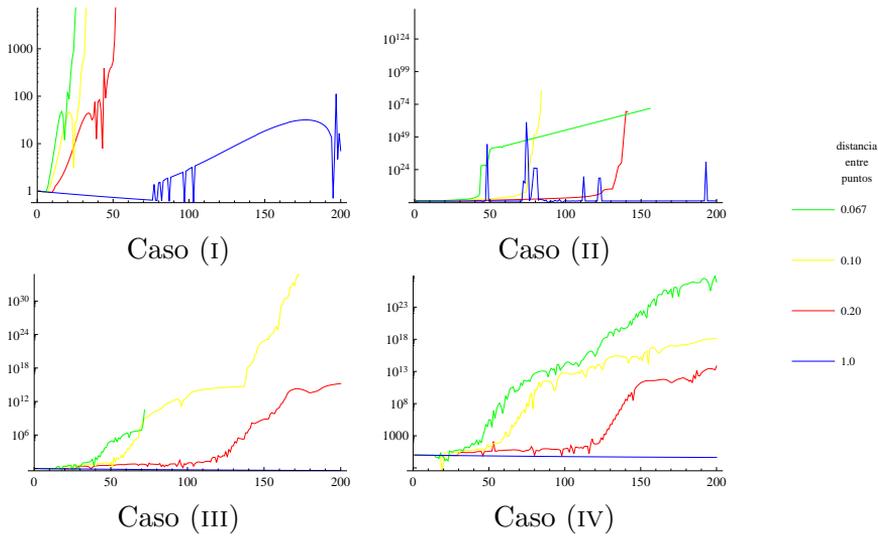


Figura 3.38: Gráficas del radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de Newmark β . En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

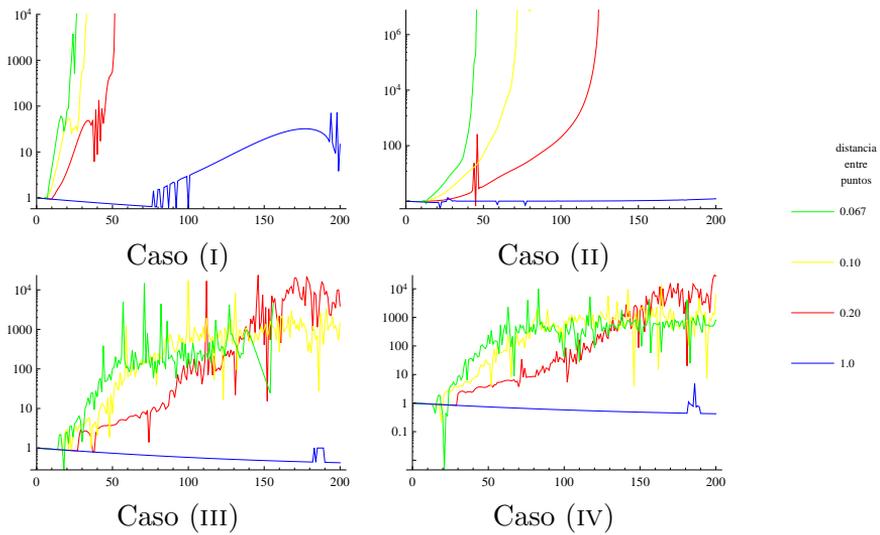


Figura 3.39: Gráficas del radio espectral en los experimentos en donde se fijan solamente las cuatro esquinas. En el eje horizontal, se representa la iteración n , y en el vertical, el radio espectral de la matriz G_n . En el proceso de simulación se ha utilizado el método de Wilson θ . En cada gráfico se superponen los resultados conseguidos en cuatro experimentos diferentes, cada uno con diferentes distancias entre puntos.

ofrecen un mejor comportamiento que el obtenido mediante las condiciones de contorno organizadas en zonas.

Además se confirma que, de los cinco métodos de integración estudiados, el que mejores resultados ofrece en cuanto a estabilidad es el método de las diferencias centradas implícito.

3.3.6. Influencia de los coeficientes η y ξ sobre la deformación

Un último aspecto a considerar es el de los valores que pueden tomar η y ξ , los coeficientes con los que se ponderan las diferencias respecto al objeto en reposo de los tensores métrico y de curvatura respectivamente. En definitiva, η representa la influencia que tiene la deformación por elongación del objeto mientras que ξ es la influencia que tiene la deformación por doblamiento.

Una primera conclusión a la que se puede llegar es que ambos valores no deben ser negativos. Efectivamente, un valor negativo representaría una energía negativa, es decir: cuanto más deformado estuviera el objeto menor resistencia (menor energía) tendría. Si el objeto representado debe semejar un objeto real, tal situación requeriría contemplar aspectos tales como que el objeto se derrita o que pierda la elasticidad y se convierta en una deformación plástica. En todo caso no parece que sea la forma adecuada en que se deba abordar este tipo de comportamiento.

Parece claro también que cuanto más próximos a 0 definan dichos valores menor será la influencia de la componente elástica en el sistema de ecuaciones. Esto conlleva considerar que cada elemento discreto sobre el que se realizan los cálculos se encuentra poco relacionado con los que se encuentran en su entorno, en definitiva cada uno se comportará como una partícula independiente, si los coeficientes son 0, o casi independiente, si no son 0 pero no son suficientemente grandes.

Otro aspecto a tener en consideración es la posible interrelación que pueda haber entre la masa, el coeficiente de amortiguación y los parámetros η y ξ .

Por último, un detalle que no puede ser obviado es el hecho de que se ha constatado que el comportamiento de la simulación está muy relacionado con la distancia entre puntos.

Para determinar el rango de valores aceptables para todos estos parámetros se ha procedido a la realización de un conjunto de simulaciones, variando los valores de η y ξ para poder determinar la influencia de ambos valores e intentando ajustar los parámetros de masa y coeficiente de amortiguación para intentar obtener resultados aceptables.

Estudio de la elongación de una lámina

El objetivo de este estudio es el de determinar si es posible conseguir representar una lámina que, estando sujeta por uno de sus lados y suspendida en un campo gravitatorio uniforme, se comporte de manera que quede colgando pero manteniendo sus proporciones; es decir, sin que cambie su tamaño.

Para conseguir dicho efecto η deberá ser suficientemente grande. En cambio ξ debe valer 0, toda vez que no interesa restringir el posible doblamiento de la lámina sino solamente su posible estiramiento o encogimiento.

El efecto obtenido varía en función de la distancia entre los puntos del mallado. Por ello se han estudiado dos casos, uno con una distancia entre puntos de 0.1 y el otro con una distancia de 1.0. Tanto en un caso como en el otro no se ha podido conseguir una rigidez absoluta, en la figura 3.40 puede observarse el porcentaje de elongación en función del valor de η . Se ha procedido a realizar un total de 100 simulaciones en cada caso. En cada una se ha variado el valor de η entre 0 hasta un valor máximo obtenido empíricamente. En cada simulación se ha determinado el valor de la máxima elongación y el valor de la elongación al cabo de 10 segundos.

Un aspecto a destacar es el hecho que los valores máximos para los que se han conseguido deformaciones aceptables varía en función de la distancia entre puntos. Así en el experimento con una distancia de 0.1 el valor máximo ha sido de 24.0 en cambio para el experimento con una distancia de 1.0 el valor máximo ha sido mayor que 1.05×10^9 .

Como puede observarse en la figura 3.40 el porcentaje de elongación disminuye en todos los casos, pero nunca llega a ser nulo, con lo que se puede concluir que siempre habrá un cierto grado de estiramiento en los objetos que se deformen.

Simulación de una lámina rígida

Se ha intentado obtener la simulación de una lámina cuadrada rígida sujeta por dos lados adyacentes. El objetivo de este experimento es intentar determinar si es posible obtener una combinación de valores que minimice el doblamiento de la lámina a la vez que no varíen sus dimensiones. Por lo tanto ξ deberá impedir que la lámina se doble y η que encoja. Cabe destacar que en ningún caso se ha conseguido un comportamiento semejante al de una lámina rígida. El hecho de contemplar una cierta resistencia al doblamiento hace que el sistema se vuelva inestable. Para una lámina de tamaño unitario con masa 1 y coeficiente de amortiguación 1 en cada punto, que como se ha indicado en el apartado anterior, admite un valor para η de 24.0, solamente permite un valor para ξ aproximadamente cuatro órdenes de magnitud menos que el valor de η . El resultado es que la lámina tiene

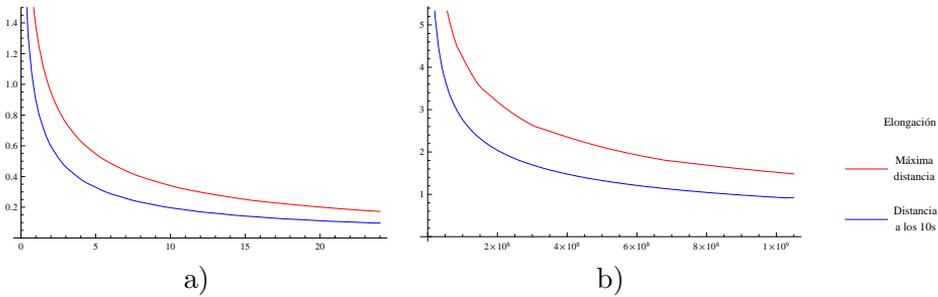


Figura 3.40: El experimento de dejar colgando una lámina permite concluir que siempre cabe esperar un cierto grado de elongación en los objetos deformados. En rojo se muestra la distancia máxima alcanzada y en azul la distancia tras una simulación de 10 segundos. El eje de abscisas representa los valores que toma η en cada simulación. En a) se muestra la elongación de un objeto suspendido con una distancia entre puntos de 0.1, la masa de cada punto es 1.0 y el coeficiente de amortiguación también es 1.0. En b) el objeto tiene una distancia entre puntos de 1.0 es tamaño es 10×10 , la masa de cada punto es 100 y el coeficiente de amortiguación también es 100. Como puede observarse el comportamiento es semejante en ambos casos, pero la magnitud de η cambia considerablemente.

una ligera resistencia al doblamiento, aunque dicha resistencia resulta totalmente insuficiente para impedir que éste se produzca. Para el caso de una distancia entre puntos mayor se observa que los valores de ξ pueden ser mayores, para valores de $\eta = 5.5 \times 10^6$ se pueden llegar a utilizar valores para ξ de aproximadamente un orden de magnitud menos, aunque sigue siendo insuficiente para impedir el doblamiento de la superficie.

Se ha realizado un experimento con una lámina de dimensión unitaria sujeta como se ha descrito, con los valores $\eta = 24.0$, $\xi = 0.0006$, $m = 1.0$ y $\gamma = 1.0$, se ha realizado una simulación de 20 segundos. La distancia entre puntos es de 0.1. Se ha procedido a medir la distancia del punto en la esquina del objeto que no está sujeta respecto de la posición original sin deformar. En la figura 3.41.a se muestra la diferencia entre la simulación con una lámina en la que el valor de ξ es 0, es decir, no tiene resistencia alguna al doblamiento. Como puede observarse hay una diferencia en el comportamiento. En la figura 3.41.b se muestra la distancia respecto a la posición sin deformar del punto estudiado.

Se ha realizado el mismo experimento con una lámina de dimensiones 10.0×10.0 y una distancia entre puntos de 1.0, con los valores $\eta = 5.5 \times 10^6$, $\xi = 1.18 \times 10^5$, $m = 100.0$ y $\gamma = 100.0$. En la figura 3.42.a se muestra la diferencia entre la simulación con una lámina que no tiene resistencia alguna al doblamiento, es decir, con $\xi = 0$. Como puede observarse hay una diferencia en el comportamiento. En la figura 3.42.b se muestra la distancia respecto a la posición sin deformar del

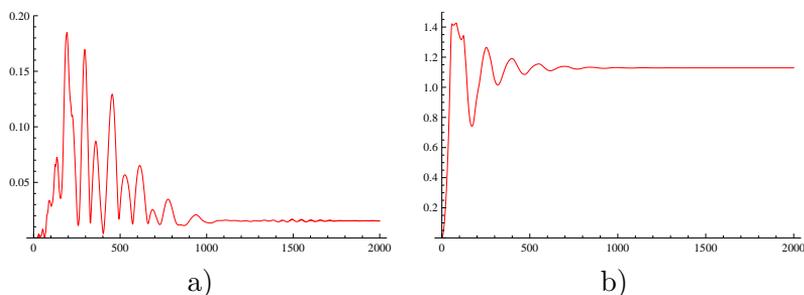


Figura 3.41: El experimento de sujetar una lámina por dos lados adyacentes y determinar la medida en que se mantiene rígida permite comprobar que con el modelo de deformación estudiado no cabe esperar tal comportamiento. Para el caso de una lámina de dimensión 1.0×1.0 con una distancia entre puntos de 0.1 una masa de 1.0 y un coeficiente de amortiguación de 1.0 en cada punto y $\eta = 24.0$ y $\xi = 0.0006$. En a) se muestra la diferencia en cuanto al comportamiento respecto de una lámina sin ninguna resistencia al doblamiento. En b) se muestra la distancia respecto a la posición del punto estudiado cuando éste no está deformado.

punto estudiado.

De todo este estudio se puede concluir que el modelo de deformación es aceptable para representar objetos que no opongan una gran resistencia a la deformación. Además, la dependencia de la distancia entre puntos hace que el rango de valores que se pueden aplicar a los coeficientes η y ξ varíe enormemente, dificultándose la selección de los valores adecuados para cada caso.

3.3.7. Coste computacional del comportamiento del modelo

Otro de los aspectos a considerar a la hora de determinar la adecuación de los diferentes modelos es el referente al coste que representa su implementación. Se tomará en consideración el problema de la obtención del coste computacional del comportamiento para cada uno de los modelos estudiados. Más precisamente, se considerará el estudio relacionado con la simulación en el caso de una superficie discretizada en M filas y N columnas, es decir $M \times N$ en el plano.

Previamente al proceso de simulación propiamente dicho es preciso construir los vectores y matrices necesarios para resolver numéricamente la ecuación diferencial ordinaria. Como el coste computacional para obtener cada uno de los vectores y/o matrices es $O(M \times N)$, entonces el coste computacional total de este proceso es también $O(M \times N)$.

Cabe recordar que en cada caso las ecuaciones relacionadas vienen dadas

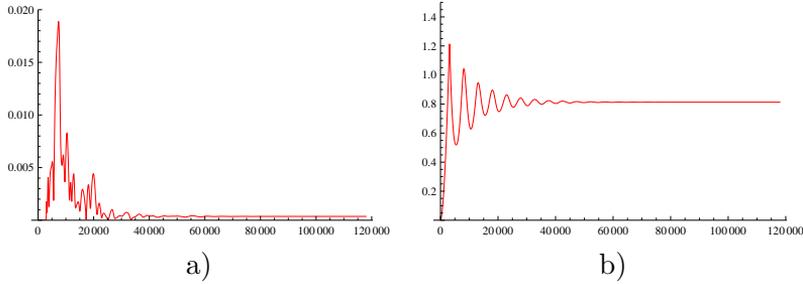


Figura 3.42: El experimento de sujetar una lámina por dos lados adyacentes y determinar la medida en que se mantiene rígida permite comprobar que con el modelo de deformación estudiado no cabe esperar tal comportamiento. Para el caso de una lámina de dimensión 10.0×10.0 con una distancia entre puntos de 1.0 una masa de 100.0 y un coeficiente de amortiguación de 100.0 en cada punto y $\eta = 5.5 \times 10^6$ y $\xi = 1.18 \times 10^5$. En a) se muestra la diferencia en cuanto al comportamiento respecto de una lámina sin ninguna resistencia al doblamiento. En b) se muestra la distancia respecto a la posición del punto estudiado cuando éste no esta deformado.

- para el método de las diferencias centradas por

$$\left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t} \right) \mathbf{p}^{t+\Delta t} = f_p^t + \left(\frac{2\mathbf{M}}{\Delta t^2} - \mathbf{K} \right) \mathbf{p}^t + \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2} \right) \mathbf{p}^{t-\Delta t},$$

- para el método de las diferencias centradas implícito por

$$\left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{2\Delta t} + \mathbf{K} \right) \mathbf{p}^{t+\Delta t} = f_p^t + \left(\frac{2\mathbf{M}}{\Delta t^2} \right) \mathbf{p}^t + \left(\frac{\mathbf{C}}{2\Delta t} - \frac{\mathbf{M}}{\Delta t^2} \right) \mathbf{p}^{t-\Delta t},$$

- para el método de Houbolt por

$$\begin{aligned} \left(\frac{2\mathbf{M}}{\Delta t^2} + \frac{11\mathbf{C}}{6\Delta t} + \mathbf{K} \right) \mathbf{p}^{t+\Delta t} &= f_p^{t+\Delta t} + \left(\frac{5\mathbf{M}}{\Delta t^2} + \frac{3\mathbf{C}}{\Delta t} \right) \mathbf{p}^t - \\ &\quad \left(\frac{4\mathbf{M}}{\Delta t^2} + \frac{3\mathbf{C}}{2\Delta t} \right) \mathbf{p}^{t-\Delta t} + \left(\frac{\mathbf{M}}{\Delta t^2} + \frac{\mathbf{C}}{3\Delta t} \right) \mathbf{p}^{t-2\Delta t}, \end{aligned}$$

- para el método Wilson θ por

$$\begin{aligned} \left(\frac{6\mathbf{M}}{\theta^2 \Delta t^2} + \frac{3\mathbf{C}}{\theta \Delta t} + \mathbf{K} \right) \mathbf{p}^{t+\theta \Delta t} &= \theta f_p^{t+\theta \Delta t} + (1 - \theta) f_p^t + \left(\frac{6\mathbf{M}}{\theta^2 \Delta t^2} + \frac{3\mathbf{C}}{\theta \Delta t} \right) \mathbf{p}^t + \\ &\quad \left(\frac{6\mathbf{M}}{\theta \Delta t} + 2\mathbf{C} \right) \dot{\mathbf{p}}^t + \left(2\mathbf{M} + \frac{\theta \Delta t \mathbf{C}}{2} \right) \ddot{\mathbf{p}}^t \end{aligned}$$

- y finalmente para el método Newmark β por

$$\begin{aligned} \left(\frac{1}{\beta\Delta t^2}\mathbf{M} + \frac{\gamma}{\beta\Delta t}\mathbf{C} + \mathbf{K} \right) \mathbf{p}^{t+\Delta t} = & f_p^{t+\Delta t} \\ & + \left(\left(\frac{1}{2\beta} - 1 \right) \mathbf{M} + \Delta t \left(\frac{\gamma}{2\beta} - 1 \right) \mathbf{C} \right) \ddot{\mathbf{p}}^t \\ & + \left(\frac{1}{\beta\Delta t^2}\mathbf{M} + \left(\frac{\gamma}{\beta\Delta t} - 1 \right) \mathbf{C} \right) \dot{\mathbf{p}}^t \\ & + \left(\frac{1}{\beta\Delta t^2}\mathbf{M} + \frac{\gamma}{\beta\Delta t}\mathbf{C} \right) \mathbf{p}^t \end{aligned}$$

estas ecuaciones se pueden expresar de una forma más breve:

- para el método de las diferencias centradas:

$$S_1\mathbf{p}^{t+\Delta t} = f_p^t + A_1\mathbf{p}^t + B_1\mathbf{p}^{t-\Delta t},$$

- para el método de las diferencias centradas implícito:

$$S_2\mathbf{p}^{t+\Delta t} = f_p^t + A_2\mathbf{p}^t + B_2\mathbf{p}^{t-\Delta t},$$

- para el método de Houbolt:

$$S_3\mathbf{p}^{t+\Delta t} = f_p^{t+\Delta t} + A_3\mathbf{p}^t - B_3\mathbf{p}^{t-\Delta t} + E_3\mathbf{p}^{t-2\Delta t},$$

- para el método Wilson θ :

$$S_4\mathbf{p}^{t+\theta\Delta t} = \theta f_p^{t+\Delta t} + (1 - \theta)f_p^t + A_4\mathbf{p}^t + F_4\dot{\mathbf{p}}^t + G_4\ddot{\mathbf{p}}^t$$

- y para el método Newmark β :

$$S_5\mathbf{p}^{t+\Delta t} = f_p^{t+\Delta t} + A_5\mathbf{p}^t + F_5\dot{\mathbf{p}}^t + G_5\ddot{\mathbf{p}}^t$$

A continuación se procederá a analizar el coste computacional, en tiempo, del proceso de simulación de los sistemas algebraicos lineales descrito. En el caso del método de las diferencias centradas, para obtener la matriz $\mathbf{K}(\mathbf{r}(t))$ primeramente se necesita calcular las expresiones implicadas en la ecuación (3.19), este proceso requiere un tiempo que es del orden de $O(M \times N)$. El proceso de construcción de la matriz requerirá un tiempo del orden de $O(M \times N)$, ya que no se debe rellenar la matriz entera, sino que solamente se toman en consideración los puntos que están relacionados en la deformación y movimiento de cada uno de los puntos de la superficie discreta. Por lo tanto el tiempo necesario para este proceso es del orden $O(M \times N)$.

Como las matrices \mathbf{S}_1 y \mathbf{B}_1 son diagonales, también se requiere un tiempo del orden de $O(M \times N)$ para calcularlas. Éste también es el orden del tiempo necesario para calcular la matriz \mathbf{A}_1 , ya que resulta de la diferencia entre $\mathbf{K}(\mathbf{r}(t))$ y $\frac{2\mathbf{M}}{\Delta t^2}$, que son matrices diagonales.

Por otro lado, el producto de un vector por una matriz requiere, en general, un tiempo que está en el orden de $O((M \times N)^2)$. Por lo tanto, éste es el tiempo necesario para calcular $\mathbf{A}_1 \cdot \mathbf{r}(t)$. Pero para calcular $\mathbf{B}_1 \cdot \mathbf{r}(t - \Delta t)$ es necesario solamente un tiempo del orden de $O(M \times N)$, debido a que \mathbf{B}_1 es diagonal. Por lo tanto se tiene un coste en tiempo del orden de $O((M \times N)^2)$ para obtener $\mathbf{r}_r - \mathbf{A}_1 \cdot \mathbf{r}(t) + \mathbf{B}_1 \cdot \mathbf{r}(t - \Delta t)$.

Finalmente debe resolverse el sistema algebraico lineal diagonal

$$\mathbf{S}_1 \cdot \mathbf{r}(t + \Delta t) = \mathbf{f}_r - \mathbf{A}_1 \cdot \mathbf{r}(t) + \mathbf{B}_1 \cdot \mathbf{r}(t - \Delta t).$$

Al ser \mathbf{S}_1 diagonal, es posible calcular su inversa en un tiempo lineal y a la vez se puede calcular su producto por $\mathbf{r}_r - \mathbf{A}_1 \cdot \mathbf{r}(t) + \mathbf{B}_1 \cdot \mathbf{r}(t - \Delta t)$, por lo que el coste computacional de todo este proceso es $O((M \times N)^2)$.

En el planteamiento del método de las diferencias centradas implícito se deben calcular las matrices \mathbf{A}_2 y \mathbf{B}_2 (que es igual a B_1), dicho proceso que es del orden de $O(M \times N)$, y se necesita el mismo orden de tiempo para obtener la matriz \mathbf{S}_2 .

Para resolver el sistema algebraico lineal

$$\mathbf{S}_2 \cdot \mathbf{r}(t + \Delta t) = \mathbf{r}_r + \mathbf{A}_2 \cdot \mathbf{r}(t) + \mathbf{B}_2 \cdot \mathbf{r}(t - \Delta t),$$

se necesita calcular el vector $\mathbf{r}_r + \mathbf{A}_2 \cdot \mathbf{r}(t) + \mathbf{B}_2 \cdot \mathbf{r}(t - \Delta t)$ y en este caso se requiere un tiempo de cálculo del orden de $O(M \times N)$ ya que la matriz es diagonal.

En cambio la matriz \mathbf{S}_2 no es diagonal y depende de $\mathbf{K}(\mathbf{r}(t))$. Por lo tanto es preciso invertirla en cada paso de integración. El tiempo requerido para ello es en general muy elevado. Por lo tanto es necesario utilizar un método de descomposición para obtener \mathbf{S}_2^{-1} . Estos procesos son de orden cúbico en el mejor de los casos, por lo que el coste computacional de todo el proceso es $O((M \times N)^3)$.

Los métodos de Houbolt, Wilson θ y Newmark β requieren un proceso de cálculo semejante al descrito para el caso del método de las diferencias centradas implícito, al fin y al cabo todos ellos son métodos implícitos y comparten la característica que la matriz \mathbf{K} forma parte de la matriz del sistema, por ello en todos estos casos el medio para resolver el sistema es del orden de $O((M \times N)^3)$.

A modo de resumen, en la tabla 3.3.7 se muestran los costes para cada uno de los modelos.

	Diferencias centradas	Diferencias centradas implícito	Houbolt	Wilson θ	Newmark β
A_i	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$
B_i	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$
E_i			$O(M \times N)$		
F_i				$O(M \times N)$	$O(M \times N)$
G_i				$O(M \times N)$	$O(M \times N)$
S_i	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$	$O(M \times N)$
total	$O((M \times N)^2)$	$O((M \times N)^3)$	$O((M \times N)^3)$	$O((M \times N)^3)$	$O((M \times N)^3)$

Tabla 3.14: Comparativa del coste computacional de los métodos de integración.

Con todo esto se puede concluir que el procedimiento de integración a lo largo del tiempo de la ecuación diferencial ordinaria representada por la ecuación (3.21) es más eficiente, asintóticamente hablando, que el procedimiento representado por la ecuación (3.22), en cuya categoría se englobarían los métodos de las diferencias centradas implícito, de Houbolt, de Wilson θ y de Newmark β .

3.3.8. Ejemplos de simulación

Una implementación de estos modelos ha sido realizada sobre diferentes plataformas utilizando el lenguaje de programación C++ y también la librería de código abierto Coin3D, basada en OpenInventor, una herramienta de modelado 3D que simplifica las tareas de visualización y composición de escenas.

Las siguientes dos figuras muestran animaciones *artísticas* de secuencias de objetos deformables. Las animaciones han sido obtenidas con el método explícito utilizando el esquema en diferencias finitas centradas con las condiciones de contorno libres (naturales).

La primera, figura 3.43, representa un pañuelo sujeto por sus cuatro esquinas y situado en un campo gravitatorio constante. Después de dos segundos, una de las esquinas es soltada, por lo que el punto fijo queda también sujeto al campo gravitatorio. El pie de figura indica los parámetros del experimento. Cabe destacar que la discretización de la tela está representada con un mallado de 21 filas y 21 columnas.

La segunda figura, figura 3.44, representa un mantel cayendo sobre una mesa. El mallado es el mismo que en el caso anterior. La colisión entre el mantel y la mesa no se ha simulado, sino que previamente se ha calculado el momento en el

que las dos superficies entran en contacto y se han establecido restricciones en los puntos del mallado que tocan la mesa a partir del instante en que esto sucede. Los parámetros del experimento se muestran en el pie de la figura.

Cabe destacar la diferente evolución de los bordes en los dos experimentos, en el primer caso (figura 3.43) es más elástico que en el segundo (figura 3.44).

3.4. Resumen y conclusiones

El modelado físico es preciso si se desea crear animaciones realistas. Los modelos físicos amplían los descriptores geométricos y añaden nuevas características a la descripción de los objetos, que son las utilizadas para la creación y desarrollo de los modelos. Para construir el modelo y la ecuación diferencial asociada con él, se pueden utilizar diferentes técnicas, como pueden ser las ecuaciones de movimiento de Lagrange utilizando una fuerza elástica interna, que se resista a la deformación, basada en geometría diferencial. Después de construir las ecuaciones de movimiento y deformación para los modelos, las ecuaciones deben ser resueltas lo más rápidamente posible utilizando métodos numéricos.

En los apartados anteriores se ha estudiado un sistema para animar modelos deformables utilizando una aproximación física. Se ha analizado como resolver la ecuación diferencial en derivadas parciales utilizando diferentes operadores en diferencias finitas, para poder estudiar el movimiento y la deformación de los objetos simulados. Se ha estudiado el desarrollo de cuatro esquemas de resolución en diferencias finitas (espaciales) que varían en la estimación del error y el tratamiento de los contornos y teniendo en consideración las ventajas e inconvenientes de cada formulación. Para realizar la integración a lo largo del tiempo se ha realizado un estudio de cinco modelos para intentar establecer cual es el más adecuado desde el punto de vista de la estabilidad numérica y la precisión de las simulaciones, así como también del coste computacional que requiere su implementación. De dicho estudio se han mostrado ejemplos numéricos.

También se ha realizado un estudio de cual es la mejor forma de abordar la discretización espacial según un planteamiento basado en diferencias finitas. Obviamente, un esquema en diferencias finitas con una aproximación de orden $O(h^2)$ depende directamente del punto implicado, y no se han estudiado todos ellos, solamente los más apropiados; es decir: los esquemas que utilizan el entorno del punto en consideración para calcular la derivada.

Es perfectamente posible desarrollar enfoques diferentes. Por ejemplo se podría realizar la discretización de la ecuación (3.19) utilizando una estrategia diferente a partir de la ecuación (3.23); desarrollando las expresiones implicadas en la

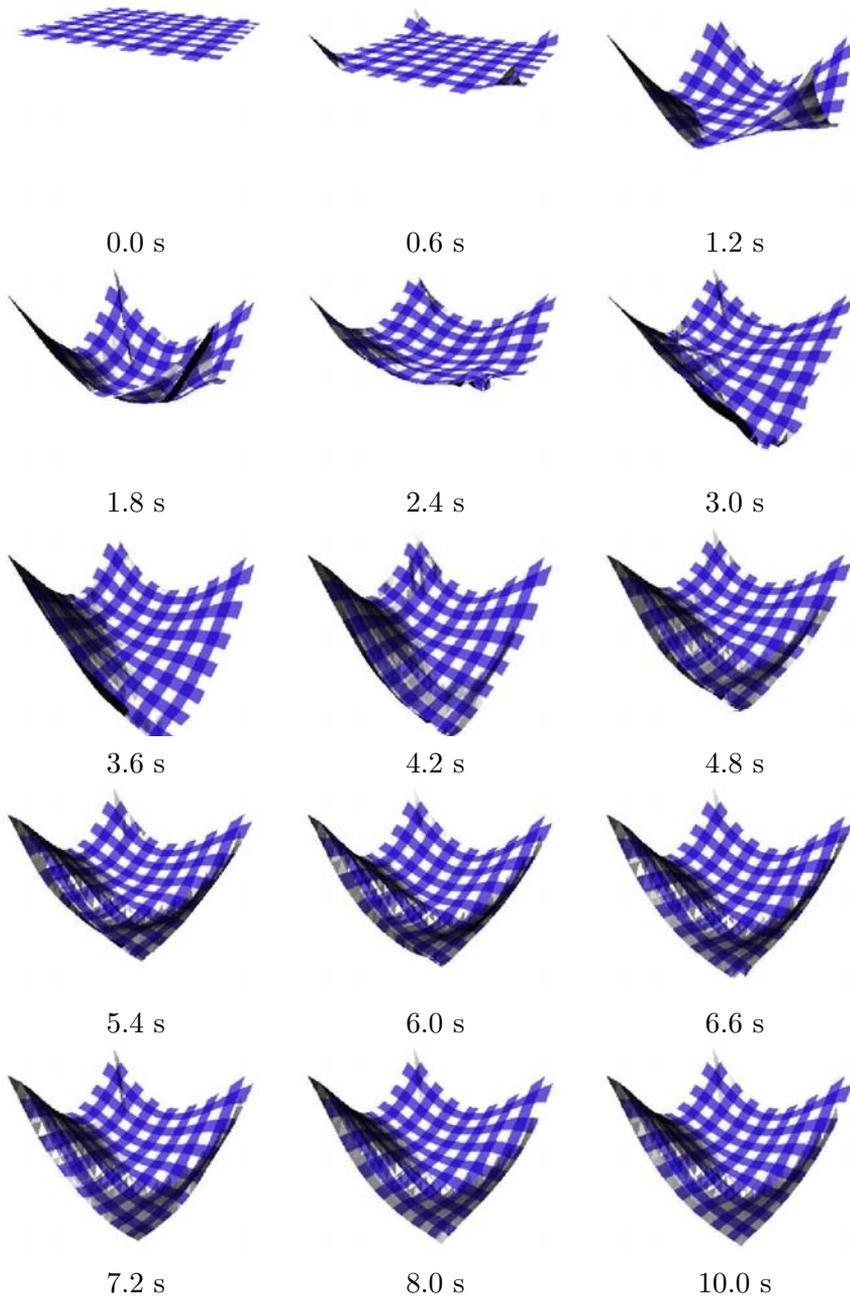


Figura 3.43: Un ejemplo de simulación, el pañuelo colgado. Un trozo de tela sujeto por sus cuatro esquinas, después de 2 s una esquina es soltada. Puede verse la evolución a lo largo de 10 s, los parámetros físicos de la tela son $\eta_{11} = 1$, $\eta_{12} = \eta_{21} = 1$, $\eta_{22} = 1$, $\xi_{i,j} = 0 \forall i, j$, la masa y el coeficiente de amortiguación son iguales a 1 en cada punto del mallado discreto, y el paso de simulación es de 0.01 s.

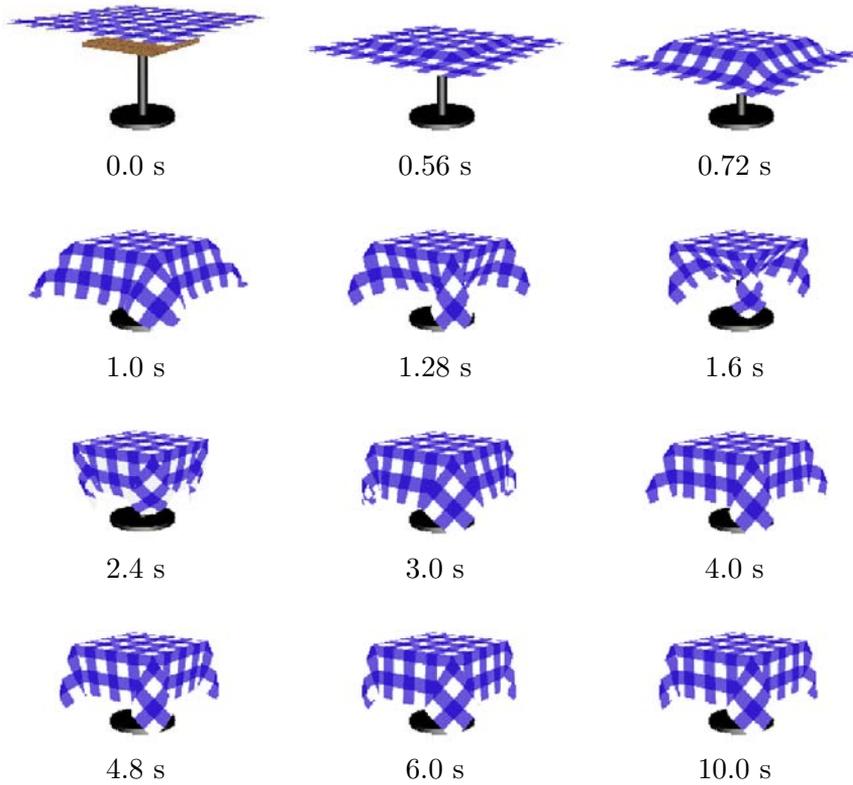


Figura 3.44: Un ejemplo de simulación, la mesa del restaurante. Un mantel cae sobre una mesa. El tiempo de simulación total es de 10 s dividido en incrementos de 0.01s, los parámetros físicos de la tela son $\eta_{11} = 4$, $\eta_{12} = \eta_{21} = 4$, $\eta_{22} = 4$, $\xi_{i,j} = 0 \forall i, j$, la masa y la densidad de amortiguación son iguales a 1 en cada punto del mallado discreto. El efecto de colisión ha sido realizado fijando los puntos afectados después de 0.5 s de tiempo de simulación, el tiempo requerido para que el mantel alcance la mesa.

ecuación (3.19) para poder obtener una nueva expresión para la aproximación numérica. Así pues, suponiendo η_{ij} constante, en el caso del primer término se obtendría:

$$\frac{\delta\varepsilon(\mathbf{r})}{\delta\mathbf{r}} \approx - \sum_{i,j=1}^2 -\eta_{ij} \left(\left(\frac{\partial^2\mathbf{r}}{\partial x_i^2} \cdot \frac{\partial\mathbf{r}}{\partial x_j} + \frac{\partial\mathbf{r}}{\partial x_i} \cdot \frac{\partial^2\mathbf{r}}{\partial x_i\partial x_j} \right) \frac{\partial\mathbf{r}}{\partial x_j} + \left(\frac{\partial\mathbf{r}}{\partial x_i} \cdot \frac{\partial\mathbf{r}}{\partial x_j} - G_{ij}^0 \right) \frac{\partial^2\mathbf{r}}{\partial x_i\partial x_j} \right),$$

obteniéndose una expresión similar para el segundo término. A continuación se aplicarían los operadores en diferencias finitas para obtener la correspondiente expresión (3.23) para el primer término:

$$\mathbf{e}(x_1, x_2) = - \sum_{i,j=1}^2 -\eta_{ij} \left((D_{ii}(\mathbf{r}) \cdot D_j(\mathbf{r}) + D_i(\mathbf{r}) \cdot D_{ij}(\mathbf{r})) D_j(\mathbf{r}) + (D_i(\mathbf{r}) \cdot D_j(\mathbf{r}) - G_{ij}^0) D_{ij}(\mathbf{r}) \right),$$

y una expresión equivalente para el segundo término. Las ventajas de este nuevo enfoque son que se mantendría el mismo orden de aproximación para los métodos (I) y (II), descritos en la página 39, pero en este caso se necesitarían menos puntos vecinos cerca del punto central (x_1, x_2) . Con lo que cabría esperar mejores resultados que los obtenidos. De todas maneras sería deseable un modelo no tan condicionado como el estudiado de manera que fuera aplicable con independencia de conceptos tales como la proximidad de los puntos.

Además se puede establecer que aunque cualquiera de los modelos de integración estudiados en principio resuelve el problema, su adecuación depende de otros factores. De todos ellos el mejor parece ser el método de las diferencias centradas implícito ya que ofrece los mejores resultados en todas las pruebas realizadas. Aun así cabe tener en consideración que el menor coste computacional del método de las diferencias centradas hace que pueda ser una alternativa interesante si las condiciones de la simulación permiten su uso.

Del estudio realizado se desprende que la discretización espacial es un componente esencial que no puede ser obviado. Para paliar los problemas derivados se podrían estudiar nuevas formas de realizar esta discretización, pero lo más probable es que no se consiguiera hallar un modelo aplicable en todas las situaciones. Sería interesante poder disponer de un modelo de representación que permitiera poder minimizar los efectos negativos detectados.

Por otro lado, del estudio del modelo descrito originalmente en [TPBF87] se comprueba que la aplicación de la Geometría Diferencial como mecanismo para estimar la rigidez del objeto a deformar da como resultado un comportamiento muy elástico, poco adecuado si se desea representar un objeto que tenga un cierto

1. Determinación de los métodos de integración numérica que se quieren utilizar para resolver la ecuación (2.22) asociada al funcional de energía considerado.
2. Discretización temporal.
3. Determinación de las discretizaciones espaciales.
4. Expresión del sistema de ecuaciones que se tiene que resolver en la forma dada en la sección 3.3.5 para cada uno de los modelos determinados en el paso 1.
5. Determinación de los experimentos necesarios para estudiar la influencia de las discretizaciones temporales y espaciales.
6. Estudio del número de condición y del radio espectral para cada experimento del paso 5 y cada expresión recurrente determinada en el paso 4.
7. Estudio del coste computacional para cada método.
8. Para los mejores métodos determinados por la intersección de 6 y de 7, evaluar la influencia de los coeficientes del funcional de energía, si los tuviese.

Tabla 3.15: Metodología para la validación de métodos de integración.

grado de rigidez. Sería interesante disponer de medios alternativos que permitan tal tipo de representación.

Otro aspecto a destacar es que se ha establecido una metodología para validar el método de integración elegido para resolver el sistema de ecuaciones diferenciales ordinarias derivado de la ecuación (2.22), o la ecuación (3.20), ya que el estudio realizado no depende del funcional de energía considerado, véase la ecuación (3.17), para medir la deformación del objeto y a partir del cual se obtiene la matriz K de rigidez del sistema. Esta metodología pasa por los pasos descritos en la tabla 3.15. Con esta metodología es posible comparar objetivamente qué métodos de integración son los más adecuados en cada caso.

Representación de objetos deformables

Para poder realizar la simulación de la deformación de objetos en un sistema informático es preciso poder representar la forma de dichos objetos. Para ello será necesario describir la geometría de dichos objetos y también determinar la evolución de dicha geometría durante el proceso de deformación, [MDSA00].

Entre las publicaciones relacionadas con modelos deformables han aparecido numerosos estudios comparativos, de entre los que cabe destacar algunos clásicos [MT96, GM97], y otros más recientes [GB08, NMK⁺06] así como algunos que tratan en profundidad algunos temas más específicos [Far92, Pie91, CF01, BS08, Ter11].

En este capítulo se estudiarán los enfoques, en cuanto a la representación de la geometría de los objetos deformables, que a lo largo del tiempo se han planteado. En cada caso se indicarán los aspectos para los que resultan más adecuados.

Una primera clasificación de los modelos de representación de objetos establece que dichos modelos pueden ser continuos o discretos. En el caso de la representación discreta la forma se infiere de un conjunto finito de puntos conocidos sobre su superficie. El conocimiento es incompleto, por lo que pueden surgir problemas numéricos a la hora de realizar determinados cálculos diferenciales (normales en un punto dado, derivadas, ...). Es bastante usual considerar como discretas las representaciones que solo tienen continuidad C^0 en la superficie. Por el contrario, en una representación continua se conoce toda la superficie, con lo que, en principio, todos los parámetros geométricos están bien definidos y son conocidos en cualquier punto de la superficie. Este segundo enfoque no está exento de inconvenientes, ya que si se desea un tratamiento computacional será precisa una discretización espacial del modelo.

Dentro de los modelos de representación continua a su vez se puede establecer una distinción entre los modelos paramétricos, los modelos explícitos y los modelos implícitos, véase la figura 4.1.

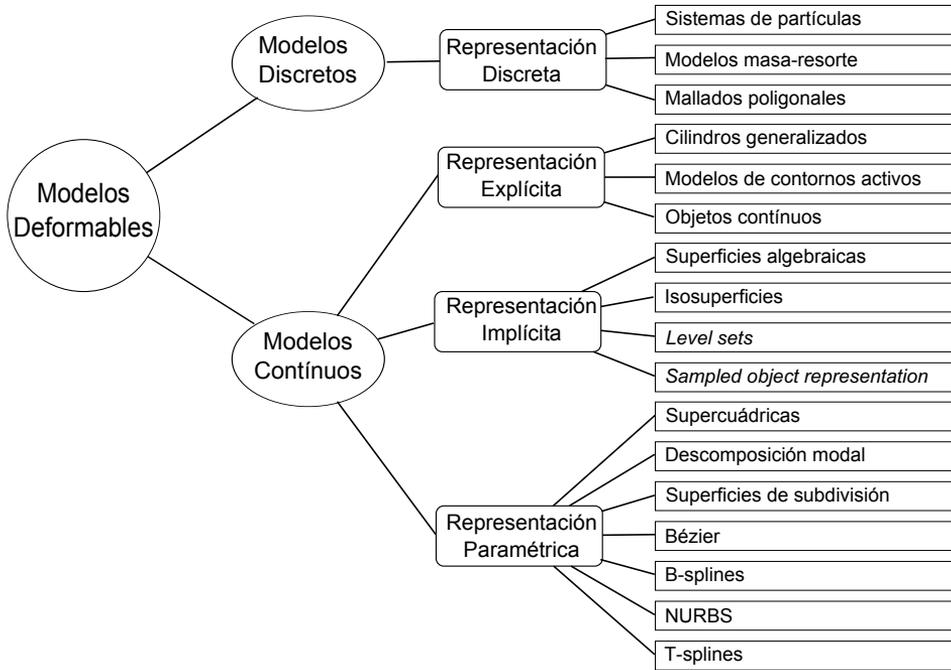


Figura 4.1: Una clasificación de la representación geométrica de los objetos deformables.

4.1. Modelos discretos

Este tipo de representación de objetos ha sido utilizado desde mucho antes de la aparición de los ordenadores. Durante el renacimiento, de la mano de Leonardo da Vinci, Luca Paccioli, o Paolo *Uccello*¹, ya aparecen representaciones geométricas de objetos tridimensionales en forma de lo que actualmente se denominarían mallados poligonales (véase la Figura 4.2).

Estos modelos permiten la representación de los objetos de una forma más sencilla que la mayoría de los otros modelos, ya que se prescinde de los parámetros presentes en los demás modelos y, por lo tanto, evitando también los inconvenientes que dichos parámetros conllevan. Sin embargo, no todo son ventajas ya

¹Una de cuyas obras aparece en la portada de la revista *Computer Aided Geometric Design*.

que aunque por un lado se solventan buena parte de las dificultades que tienen los otros modelos, también es cierto que estos modelos conducen a un esquema numérico explícito que puede acarrear problemas numéricos debido a que las ecuaciones de movimiento obtenidas no permiten la regularización de la superficie y puede ser necesario tener que imponer restricciones.

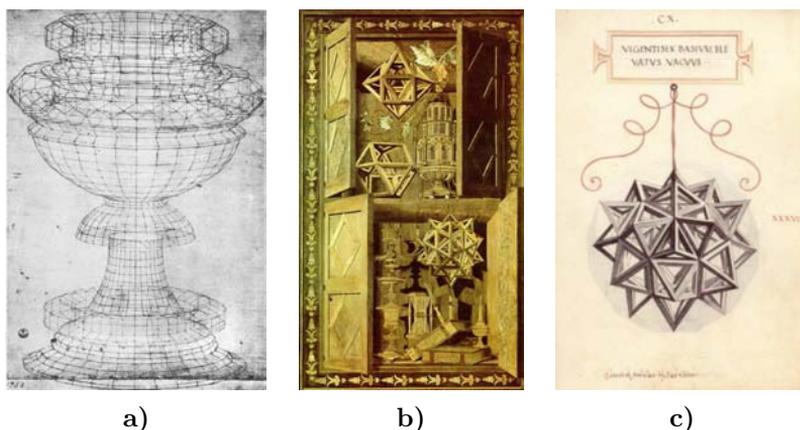


Figura 4.2: Los modelos discretos son, de hecho, anteriores a la informática. Se encuentra claramente documentada su utilización desde el siglo XV: **a)** Paolo Uccello, *Estudio de perspectiva de un cáliz*, alrededor de 1450, lápiz sobre papel, Galería de los Uffizzi, Florencia, Italia. **b)** Un ejemplo de *intarsia*, mosaico formado por madera taraceada, realizada por *Fra Giovanni Paciolli* para la catedral de Santa María en Organo, Verona, Italia, a principios del siglo XVI. En él se pueden apreciar algunas formas geométricas obtenidas a partir de dibujos realizados por Leonardo da Vinci, como el rombicuboctaedro elevado, **c)**, que aparece en *De Divina Proportione* del mismo Paciolli impreso en 1509.

4.1.1. Mallados poligonales

Un mallado poligonal, \mathcal{M} , permite describir un objeto a partir de un conjunto discreto de puntos y una relación de conectividad entre ellos de manera que para cada punto se tiene una relación de puntos vecinos. Por lo tanto un mallado se define como un par, $\mathcal{M} = \langle \mathcal{V}, \mathcal{N} \rangle$, de manera que

- \mathcal{V} es el conjunto de vértices, definidos en un espacio \mathbb{R}^d :

$$\mathcal{V} = \{P_i\}_{i \in I}.$$

- \mathcal{N} es la función conectividad del mallado:

$$\begin{aligned} \mathcal{N} : \mathcal{V} &\longrightarrow \mathcal{V}^* \\ P_i &\mapsto \{P_{i,1}, P_{i,2}, \dots, P_{i,m_i}\}, \end{aligned}$$

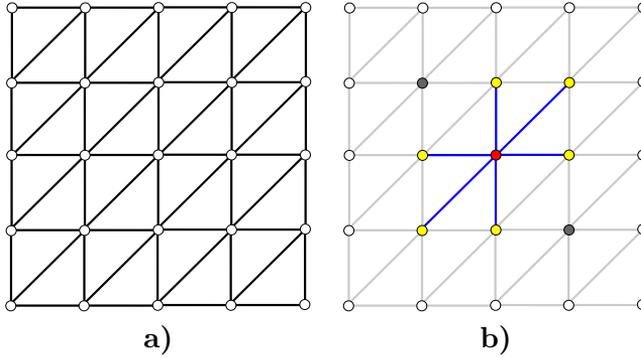


Figura 4.3: En **a** se muestra un mallado poligonal y en **b** se muestra la función de conectividad; mostrando, para el vértice de color rojo, sus vecinos (los vértices amarillos) y las aristas que los unen, (en azul). Nótese que solamente los vértices conectados mediante una arista son considerados como vecinos. En la imagen, los dos vértices grises no son considerados vecinos.

en donde \mathcal{V}^* es el conjunto de las partes finitas de \mathcal{V} y $m_i (m_i \geq 2)$ indica el número de vecinos del vértice P_i . La figura 4.3 aclara el concepto.

- Si dos vértices están unidos, lo están por una sola arista:

$$\forall i \in I, \quad P_{i,j} \neq P_{i,k} \forall j, k : 1 \leq j, k \leq m_i \wedge j \neq k.$$

- Un vértice no está unido a sí mismo:

$$\forall i \in I, \quad P_i \notin \mathcal{N}(P_i).$$

Se define el conjunto de aristas del mallado poligonal, \mathcal{E} , como:

$$\mathcal{E} = \{(P_i, P_{i,j}) \forall P_i \in \mathcal{V}, \forall P_{i,j} \in \mathcal{N}(P_i)\},$$

a partir de \mathcal{E} se define el conjunto de polígonos \mathcal{P} como:

$$\mathcal{P} = \{f = (e_{i_1, i_2}, e_{i_2, i_3}, \dots, e_{i_l, i_1}), e_{i_k, i_{k+1}} \in \mathcal{E}\} \subseteq \mathcal{E}^*. \quad (4.1)$$

Existen múltiples técnicas de generación de mallados poligonales a partir de objetos definidos mediante algún otro método de representación. En [MTG04] los objetos se definen con mallados cúbicos (es decir se divide el espacio en cubos regulares, de forma análoga a como se definiría una superficie) el mallado se genera automáticamente a partir del mallado poligonal de las superficies que definen el objeto

Si el valor de l en (4.1) es siempre 3, entonces todos los polígonos que definen mallado son triángulos, y al mallado se le suele llamar triangulación. Utilizar triángulos ofrece una serie de ventajas, por ejemplo la convexidad de cada uno de los polígonos y el detalle de que los tres vértices sean coplanarios, hace que los algoritmos de representación de superficies definidas mediante triangulación sean muy sencillos y eficientes. En [THMG04] se plantea un método de representación de objetos basado en mallados triangulares y tetraédricos (es decir, mallados triangulares que definen objetos, no superficies de objetos), la particularidad de su propuesta es que el modelo permite definir objetos complejos sin un coste computacional elevado.

Otras aproximaciones basadas en este modelo son, entre otras: [AS10] en donde se propone un mecanismo de segmentación de imágenes basado en mallados triangulares que adaptan su topología para ajustarse a los objetos de la imagen.

Los modelos explícitos son difíciles de utilizar cuando la topología de los objetos representados puede variar durante el proceso de deformación. Aunque hay algunos trabajos en esta dirección. En [SOG09] se presenta un modelo que utiliza mallados triangulares para representar objetos capaces de ser divididos o rotos. Además, en [WTGT09] los objetos también pueden ser divididos, y además pueden ser mezclados o combinados unos con otros.

Otra forma en la que el objeto deformable es representado es descrita en [KP11], en donde los objetos deformables están formados por un esqueleto sobre el que se dispone un mallado que determina el volumen y la forma.

Otra aplicación interesante de este modelo se encuentra en [NISA07], en donde se describe un herramienta gráfica, *FiberMesh*, con la que un usuario puede diseñar de forma fácil objetos con técnicas de deformaciones de formas libres (FFD), mediante la interpolación de curvas 3D y el uso de algunas operaciones básicas (creación de curvas, corte, extrusión, tunelado).

A pesar de la ventaja que representa su simplicidad, este tipo de representación tiene algunos inconvenientes. Entre ellos el ser dependiente de la escala o nivel de aproximación del objeto.

Triangulación de Delaunay

Una de las formas en que se puede establecer un mallado y que además computacionalmente es muy efectiva es la conocida como triangulación de Delaunay y su problema dual, los diagramas de Voronoi:

Dado un conjunto de puntos del plano, \mathcal{P} , una triangulación del conjunto es una subdivisión de su envolvente convexa en regiones triangulares mediante

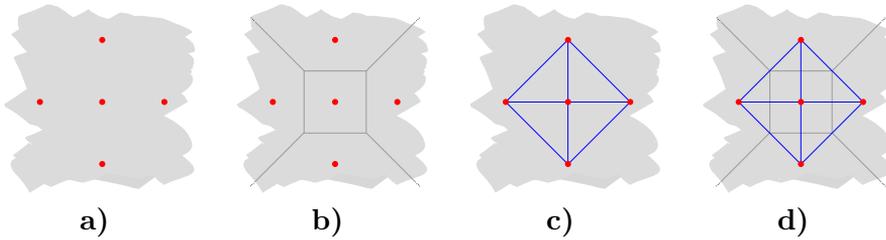


Figura 4.4: Para un conjunto de puntos en un espacio 2D, **a**, el espacio puede ser dividido en un diagrama de Voronoi **b**, es decir, dividido en regiones tales que los puntos de cada región están más próximos a su correspondiente punto del conjunto que al resto. A su vez, los puntos pueden ser interpretados como vértices de una triangulación de Delaunay, **c**. El diagrama de Voronoi y la triangulación de Delaunay están relacionados, como puede verse en **d**.

aristas cuyos vértices son puntos del conjunto. De todas ellas, la triangulación de Delaunay es la que verifica que ([She02], [She05]):

- Todos los triángulos son tales que la circunferencia que pasa por sus vértices no contiene a ningún otro punto de \mathcal{P} .
- Todas las aristas que forman la triangulación son tales que existe una circunferencia que pasa por sus extremos y que no contiene a ningún otro punto del conjunto.
- El mínimo ángulo entre aristas es maximal. Es decir, cualquier otra triangulación posible tiene aristas con un ángulo menor.

Además se cumple que la triangulación de Delaunay de un conjunto de puntos es el grafo dual del diagrama de Voronoi de dicho conjunto, véase la figura 4.4. Un diagrama de Voronoi es una teselación del plano basada en la distancia de los puntos del plano a los puntos del conjunto \mathcal{P} , ya que representa una partición del plano en regiones de manera que a cada punto del conjunto se le asocia el lugar geométrico de los puntos del plano que son más cercanos a él que a cualquier otro punto de \mathcal{P} .

Mallados simples

Los mallados simples son representaciones discretas de objetos caracterizadas por que la conectividad entre vértices es constante ([Del94, Del97]). Para la representación de superficies se utilizan los mallados 2-simple, véase la figura 4.5. En uno de estos mallados cada punto está unido mediante aristas a exactamen-

te tres puntos diferentes². Al tener una conectividad entre vértices constante la geometría es muy sencilla.

Un mallado k -simple, (M) , definido en \mathbb{R}^d representa un mallado de conectividad $k + 1$. La principal diferencia respecto a los otros los mallados poligonales reside en que la función de conectividad se define como:

$$\begin{aligned} N : \quad V &\longrightarrow \{V\}^{k+1} \\ P_i \in V &\longrightarrow (N_1(i), N_2(i), \dots, N_{k+1}(i)). \end{aligned}$$

Estos mallados constituyen un tipo particular de mallado en donde todos los vértices tienen exactamente el mismo número de vecinos. Lo que hace interesante este planteamiento es:

- Su generalidad: es posible representar cualquier tipo de superficie orientable, pudiéndose representar deformaciones de superficies k -dimensionales mediante mallados $k + 1$ -simples
- Su fácil y eficiente implementación.
- Su adaptabilidad: es posible refinar el mallado para aumentar el número de vértices en áreas con gran curvatura. Si la superficie a construir está basada en una serie de puntos dados, también es posible refinar el mallado para corregir valores o adaptarlo para crear contornos en donde los datos son incompletos.

Además se puede demostrar, [Del94], que una triangulación es equivalente a un mallado simple en el que las caras que forman las aristas son planas. También es posible definir un modelo con ambos enfoques, de esta forma es posible tener las mejores características de cada uno combinadas, [GJ10].

Se utiliza este tipo de representación en la segmentación de imágenes médicas [MDSA00, GPE⁺10]. Este modelo también puede ser utilizado en la segmentación de imágenes en general, en [Del99] se propone un algoritmo general de reconstrucción de imágenes volumétricas y de barrido basado en mallados simples deformables. Los objetos obtenidos pueden tener cualquier topología y, además, es posible dividir un objeto en múltiples partes.

4.1.2. Sistemas de partículas

Los sistemas de partículas parten del concepto de que cada objeto a representar se encuentra formado por un conjunto de *partículas* que se pueden caracteri-

²Existe una dualidad entre la representación mediante triangulaciones y la de mallados simples

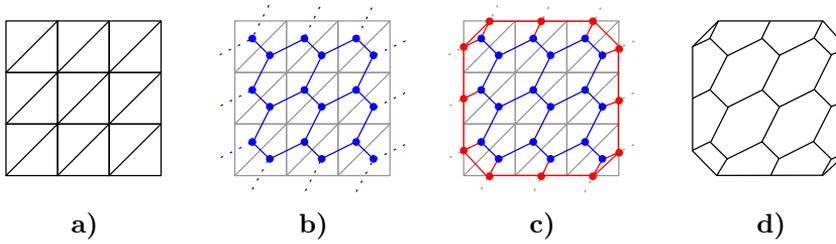


Figura 4.5: Dualidad entre una triangulación y un mallado 2-simple. **a)** representación de una superficies mediante una triangulación regular. **b)** mallado 2-simple, en azul, sobre la misma superficie, existe un problema en el contorno. **c)** se *corrige* el mallado incorporando el dual del contorno de la triangulación, en rojo. **d)** el mallado 2-simple acabado con el contorno.

zar por un conjunto de parámetros físicos (masa, posición, velocidad, aceleración, ...). El comportamiento de las partículas viene determinado por las leyes de la mecánica newtoniana. La interacción de las partículas se realiza por medio de fuerzas de atracción/repulsión mediante las que es posible obtener una cierta organización.

Por su concepción los sistemas de partículas son muy adecuados para la representación de fluidos o de objetos viscosos [CKO01]. Aunque también han sido utilizados para representar el comportamiento de telas [BHG91, BHW94a, BHW94b, EWS96].

En [ST92] se propone un modelo, el de partículas orientadas, basado en partículas a las que se les añaden funciones de potencial que favorecen la organización de las partículas en determinadas formas. Cada partícula tiene asociada una matriz de rotación respecto de un sistema de referencia. Con dicha matriz, juntamente con un vector normal asociado a cada partícula, se consiguen potenciales que inducen a las partículas a disponerse según unas restricciones predeterminadas de *coplanariedad*, *conormalidad* y *cocircularidad*. De este modo es posible conseguir la representación de superficies de forma fácil.

La utilización de los sistemas de partículas en combinación con otras técnicas ha sido utilizada para la representación de objetos a partir de datos obtenidos empíricamente. En [JSV98], se utilizan primeramente curvas paramétricas para definir el contorno de órganos obtenidos a partir de imágenes médicas. Una vez obtenido el contorno, el interior es creado utilizando partículas de diferentes tamaños y pesos que se organizan en capas. En [BN04] se utilizan partículas y superficies implícitas para la representación de nubes, de manera que cada nube se define mediante sucesivos niveles formados por partículas, sobre cada nivel se

definen las partículas del nivel siguiente. La forma final de la nube se obtiene mediante una formulación implícita.

Los sistemas de partículas combinados con otras técnicas han sido utilizados para representar objetos a partir de información obtenida empíricamente. En [JSV98], se utilizan primero curvas paramétricas para definir el contorno de órganos obtenidos a partir de imágenes médicas. Una vez obtenido el contorno, el interior es creado utilizando partículas, de diferentes tamaños y pesos, organizadas en capas. En [HLTC10] se presenta un proceso de modelado para producir la simulación de multitudes de personas basándose en un sistema de partículas. En particular, los fenómenos no deliberativos emergentes de multitudes, es decir: la auto-organización de un grupo de personas de manera que el colectivo muestra un comportamiento no detectado en los individuos por separado. Finalmente, en [MTBV09] los sistemas de partículas se utilizan como modelo para la representación de formas unidimensionales y bidimensionales, como pueden ser el pelo y la ropa.

El mayor inconveniente que plantean los modelos basados en partículas reside en la dificultad que representa obtener medidas precisas de la forma de los objetos representados (por ejemplo: la curvatura de una superficie). Otro de los inconvenientes es el coste computacional inherente, que suele ser bastante elevado.

4.1.3. Modelos masa-resorte

Los modelos masa-resorte son una de las técnicas físicas más utilizadas. Un objeto representado con este modelo consiste en una serie de masas puntuales conectadas mediante una serie de resortes, con masa despreciable y con una longitud determinada, que forman una estructura deformable. Los resortes suelen ser lineales, basándose en la Ley de Hooke, aunque es posible utilizar cualquier otro modelo de comportamiento para los mismos. El comportamiento de los objetos durante su deformación vendrá establecido por la aplicación de segunda ley de Newton a cada de las masas puntuales. Las fuerzas que actúen serán las definidas externamente por el usuario y también las que se generen por el estiramiento de los diferentes resortes como resultado del movimiento de las masas, [GOT97], [PDA00].

Las masas puntuales pueden estar conectadas mediante resortes de manera más o menos compleja, Figura 4.6. Por ejemplo, en [Pro95] se establecen tres tipos diferentes de resorte: los *resortes estructurales*, los *resortes de cizalladura* y los *resortes flexurales*; cada tipo de resorte actúa frente a un tipo de tensión determinado (tracción-compresión, cizalladura y doblamiento respectivamente).

Estos modelos son utilizados frecuentemente para representar tejidos ya que

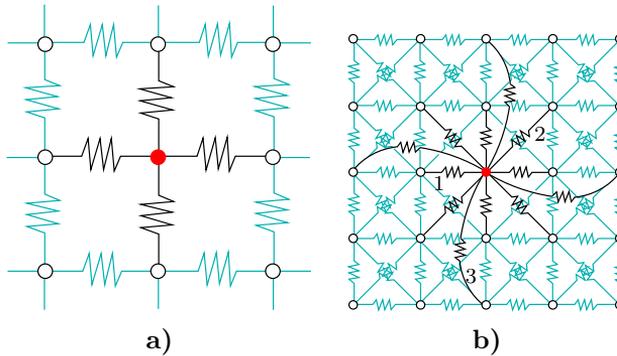


Figura 4.6: Diferentes configuraciones de masas y resortes. **a)** configuración simple: cada masa está conectada mediante resortes con sus vecinos inmediatos. **b)** configuración propuesta en [Pro95]: cada masa está conectada mediante los *resortes estructurales* (1), los *resortes de cizalladura* (2) y los *resortes flexurales* (3).

la estructura de las telas, formada por una malla y una urdimbre, hace fácil su descripción representando la superficie mediante una serie de puntos, el lugar en donde se entrecruzan las fibras del tejido, unidos cada uno con sus vecinos por una serie de ligaduras que confieren al conjunto una cierta elasticidad y flexibilidad ([DGPR97], [MDD01]).

No es ésta la única forma en que se utiliza este modelo. En [TV91, VT92] se utilizan masas y resortes, los denominados *mallados adaptativos*, para la reconstrucción de objetos a partir de imágenes. En la misma línea, en [NA93] se utiliza un modelo masa-resorte para realizar la segmentación de imágenes y el seguimiento de los objetos detectados en las mismas.

Lejos de ser abandonado, éste modelo se mantiene como un tópico de gran actualidad. Así, en [McD01] se utiliza, paradójicamente, para representar objetos rígidos con un alto grado de realismo en su movimiento. En [JL04], se parte de las partículas orientadas, [ST92], para plantear un sistema en el que las partículas mantienen unos vectores de referencia que permiten representar objetos con una gran estabilidad sin que sea necesario tener que añadir una gran cantidad de resortes para garantizar dicha estabilidad.

En general es muy difícil determinar los parámetros de elasticidad adecuados para representar el comportamiento real de un objeto, aunque en [NCC10] se propone un método automático para ello.

En [SIGGMT11] los objetos deformables se definen como mallados simples, aunque en realidad representan masas (por lo que se trata en realidad un modelo de masa-resorte). La evolución viene dirigida por una imagen y el resultado es la

segmentación de dicha imagen.

En un enfoque diferente, [SLF08] utiliza un modelo masa–resorte para representar cientos de miles de cabellos en una cabeza humana. Este modelo tiene en consideración las interacciones entre los diferentes cabellos mediante técnicas físicas.

4.2. Modelos explícitos

Los modelos explícitos parten de la base que los objetos pueden ser representados de manera que puede conocerse cualquier punto de los mismos. Para conseguir este objetivo es usual describirlos matemáticamente mediante una serie de parámetros que determinan su forma. Así un objeto, $S_{\mathbf{q}}$, se puede describir a partir de un vector de parámetros³ de forma, $\mathbf{q} = (q_1, \dots, q_{n_{\mathbf{q}}})^\top$, de manera que si Ω es el espacio paramétrico en el que se define el objeto, entonces la representación del objeto será:

$$\begin{aligned} S_{\mathbf{q}} : \mathbb{R}^{n_{\mathbf{q}}} \times \Omega &\longrightarrow \mathbb{R}^3 \\ (q_1, \dots, q_{n_{\mathbf{q}}}, \mathbf{u}) &\longrightarrow (\mathbf{x}(\mathbf{q}, \mathbf{u}), \mathbf{y}(\mathbf{q}, \mathbf{u}), \mathbf{z}(\mathbf{q}, \mathbf{u})). \end{aligned} \quad (4.2)$$

Las características de la forma obtenida, y de las deformaciones que podrá experimentar, serán unas u otras en función del tipo de parámetros escogidos. Así, por ejemplo, si los parámetros permiten controlar deformaciones locales, es decir, deformaciones que afectan a una pequeña parte de los objetos, será posible obtener objetos con formas muy elaboradas, pero a cambio de un esfuerzo computacional importante. En cambio si los parámetros controlan el objeto globalmente, entonces el coste computacional será menor y también lo será la complejidad de los objetos que se podrán obtener con ellos.

En general, este modelo tiene una serie de inconvenientes:

- Al ser modelos continuos necesitan una discretización en el espacio paramétrico, esta discretización conllevará un cierto grado de error.
- La descomposición del espacio paramétrico en intervalos regulares no conduce necesariamente a una descomposición regular de la superficie.
- La representación explícita requiere que se introduzcan condiciones de contorno para poder definir la topología del objeto.

³Bastantes autores consideran a los modelos explícitos dentro de grupo de los modelos paramétricos. Sin embargo, algunas de las características específicas permiten también considerarlos como un grupo independiente.

Por otra parte, la representación de los objetos tiende a ser sencilla. En las primeras aproximaciones a los objetos deformables era frecuente que el modelo de representación de dichos objetos siguiera un planteamiento que se puede clasificar como modelo explícito.

4.2.1. Cilindros generalizados

De entre las primeras representaciones de objetos deformables cabe destacar las que se basaron en la forma en que un contorno evolucionaba a lo largo de una curva, figura 4.7.

Los cilindros generalizados consisten en una curva en el espacio, el eje, y un conjunto de secciones transversales que se describen en este eje. Permiten describir de forma natural e intuitiva piezas que poseen elongación, o que tienen simetría axial. Los cilindros generalizados pueden ser unidos entre sí de diversas maneras para formar objetos complejos [Agi72]. La forma en que se obtiene el sólido es mediante el barrido de una curva a lo largo del eje, de este barrido se obtienen las secciones transversales que definen la forma. La curva se puede deformar mediante una función de escalado mientras evoluciona a lo largo del eje, esta función de escalado puede describirse como una curva secante y perpendicular a la curva que servirá de modelo para las secciones transversales, ver figura 4.7.

Las primeras referencias de trabajos que utilizan este modelo se remontan a principios de los años 70 del siglo XX, [Agi72]. En [AB76] los cilindros generalizados se utilizan para reconstruir la forma de objetos reales obtenidos mediante un escáner 3D primitivo. La recuperación de objetos a partir de los datos recogidos a partir de imágenes de objetos reales utilizando cilindros generalizados ha sido utilizado en otros trabajos con posterioridad, por ejemplo en [TWK88, CAL⁺04].

Mediante la combinación de varios cilindros generalizados es posible construir figuras complejas. Sin embargo, no siempre es fácil representar una forma mediante un eje y un contorno; tampoco es la forma más intuitiva de describir objetos, con lo que no parece que sea un modelo adecuado para ser utilizado con asiduidad.

El concepto de cilindros generalizados ha sido adoptado como base para otros modelos, tal es el caso de las denominadas *swung surfaces*, [QT95], y las superficies de barrido (*swept surfaces*, [PT97]) que en realidad son superficies paramétricas aunque en esencia siguen el planteamiento de los cilindros generalizados.

Recientemente han aparecido algunas nuevas aplicaciones de los cilindros generalizados. Por ejemplo, [FNMMG12] utiliza cilindros generalizados para modelar las microestructuras del suelo. El ámbito de los cilindros generalizados ha

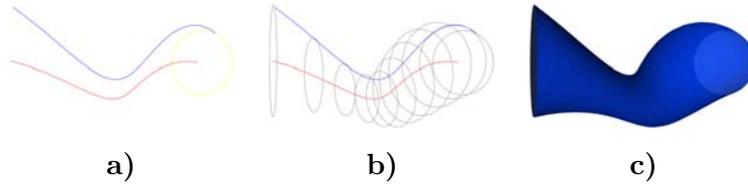


Figura 4.7: Un cilindro generalizado se forma a partir de una curva que sirve de eje, **a)**, sobre el que otra curva, el contorno, evoluciona, **b)**, para generar la forma del sólido, **c)**.

sido extendido para incluir aspectos tales como la representación de imágenes médicas. En [MC10] se utilizan los cilindros generalizados para segmentar y reconstruir árboles vasculares 3D. En el mismo campo de aplicación, en [WXXL12] los cilindros generalizados son utilizados para representar tendones, ligamentos y suturas quirúrgicas (en el mismo artículo, otros tejidos biológicos son modelados mediante mallados triangulares y cuadrangulares).

Una aproximación equivalente a los cilindros generalizados aparece en [LLC10], donde se presenta un modelo capaz de representar tubos huecos con secciones transversales deformables.

4.2.2. Modelos de contornos activos

Propuestos en [KWT88], los modelos de contornos activos, comúnmente conocidos como *snake*, son curvas que modifican su forma buscando ajustarse a una imagen dada. El mecanismo de ajuste dependerá de una serie de parámetros que impondrán restricciones. La evolución de los *snake* permite localizar los contornos y líneas así como otras características de objetos representados en imágenes. En principio los *snake* son curvas unidimensionales que pueden adaptarse a contornos, seguir movimientos, etc, respondiendo a las tensiones internas que se oponen al estiramiento y doblamiento, a las fuerzas que el usuario pueda imponer y a las que se puedan inferir de las imágenes. Otros autores, [CC93], han extendido los modelos de contornos activos de manera que pueden ser aplicados también en entornos 3D.

Los *snake* son utilizados asiduamente para el realzado y la detección de órganos en imágenes médicas [BN94, BN95, BNC96]. En [RAH97] se utilizan para extraer la información necesaria para reconstruir con un modelo paramétrico el ventrículo izquierdo del corazón. La segmentación de imágenes es pues el ámbito natural de este modelo.

En la misma línea, en [LLS⁺11] se propone un método alternativo para recons-

truir defectos craneales. Los autores combinan información médica, imágenes de baja resolución y los modelos de contornos activos para suprimir defectos de volumen parcial. El resultado es que partiendo de un cráneo con una parte defectuosa se obtiene una representación del mismo con la parte defectuosa reconstruida.

En general, los modelos de contornos activos son computacionalmente costosos. Este inconveniente se ve parcialmente reducido en [ZFT09], su propuesta consiste en analizar el flujo de gradiente de la curva para poder estudiar la evolución del contorno activo. Cuando la curva se encuentra alejada del objeto se le aplica un traslación global, y cuando se encuentra cerca del objeto se le aplica una deformación local. Con este enfoque la evolución del contorno activo es más eficiente y menor es el esfuerzo computacional requerido. Otro planteamiento se propone en [dVL09], consiste en discretizar el espacio, reemplazando la curva continua mediante un polígono, y definiendo la energía interna como una función basada en el polígono de menor longitud.

Otro problema de los modelos de contornos activos, en el campo de la segmentación de imágenes, es que no resulta fácil discriminar objetos adyacentes. En [BH12] se propone un modelo de contornos activos basado en grafos, los denominados *network snakes*. La idea principal consiste en tratar de identificar los objetos que aparecen en una imagen y también las relaciones que puedan existir entre ellos de manera que se pueden diferenciar objetos adyacentes de forma correcta.

4.2.3. Objetos continuos

Los modelos basados en una representación continua consideran el objeto definido en un subconjunto Ω del espacio 2D o 3D. El planteamiento es válido tanto para curvas como para superficies. Algunos de los artículos con mayor trascendencia en el contexto de la representación de objetos deformables han planteado modelos que siguen este planteamiento [TPBF87, TW88, TF88b].

Los objetos representados según este modelo se encuentran ubicados en el espacio euclídeo 3D según un marco de referencia, Φ . En un instante t la posición de cada punto $u \in \Omega$, $x(u, t)$, se define de diferentes formas.

En una primera formulación, [TPBF87], denominada *formulación primal* en [TF88a], la posición de cada punto material se define directamente como:

$$x(u, t) = (x_1(u, t), x_2(u, t), x_3(u, t))^T$$

Esta formulación lleva a obtener objetos que pueden deformarse y moverse libremente, aunque existe siempre un cierto riesgo de obtener un sistema mal

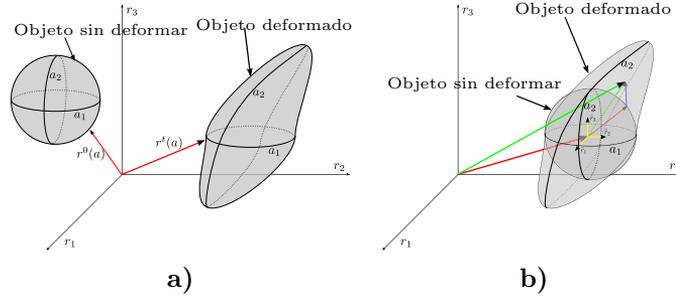


Figura 4.8: Representación de un objeto explícito: **a)** definido a partir de un sistema de referencia y **b)** definido como un desplazamiento rígido respecto a un sistema de referencia y una deformación.

condicionado tal y como se pone de manifiesto en [PMG00]. Ello hace que para la representación de objetos con un cierto grado de rigidez esta formulación no sea del todo adecuada.

Existe una segunda formulación, la *formulación híbrida*, descrita en [TW88], que plantea una combinación de conceptos de dinámica de cuerpos rígidos y de objetos deformables. En esta segunda formulación la posición, \mathbf{q} , de cada punto de un objeto vendrá definida a partir de un sistema de referencia local al objeto, ϕ , de manera que dichas posiciones se representan como

$$\mathbf{q}(u, t) = \mathbf{r}(u, t) + \mathbf{e}(u, t),$$

en donde $\mathbf{r}(u, t)$ es una componente de referencia, que define la forma del objeto sin que esté sujeto a deformaciones, y $\mathbf{e}(u, t)$ es la componente de deformación, que añade a la componente de referencia el desplazamiento relativo de cada punto en cada instante de tiempo. Tanto \mathbf{r} como \mathbf{e} se definen respecto a ϕ , que se ubica en el centro de masas del objeto, $c(u, t)$, cuya posición a su vez se define respecto al sistema de referencia inercial Φ (véase la figura 4.8). Dicho centro de masas se mueve de acuerdo a las leyes de la dinámica de los cuerpos rígidos.

Esta *formulación híbrida* no tiene los problemas de mal condicionamiento que aparecen en la *formulación primal*, pudiéndose representar con ella objetos con un alto grado de rigidez. Cuanto mayor sea la rigidez de los objetos, mejor condicionado estará el sistema. Por otro lado este sistema no obtiene buenos resultados a la hora de simular objetos flexibles (telas, objetos elásticos, ...). Sin embargo, en [TF88b] los autores consiguen representar el comportamiento de materiales viscoelásticos y plásticos así como fracturas mediante el uso de funcionales *spline generalizados*.

4.3. Modelos implícitos

Otra forma diferente de representar la forma de un objeto, S_f , es tomando los valores nulos de una función a valores reales dada, f :

$$f : \mathbb{R}^3 \longrightarrow \mathbb{R}$$

$$S_f = \{\mathbf{p} \in \mathbb{R}^3 \mid f(\mathbf{p}) = 0\} \tag{4.3}$$

De hecho, la ecuación (4.3) define la frontera del objeto, su superficie. Dependiendo de cual sea el tipo de función utilizada se tendrá más o menos capacidad de representación.

4.3.1. Superficies algebraicas

Si la función f de la ecuación (4.3) es un polinomio, entonces las superficies que se obtienen se denominan superficies algebraicas. Este tipo de superficies es muy utilizado aunque el uso de ciertos tipos de polinomio puede dar lugar a problemas. Por ejemplo, el determinar los ceros de una función polinomial no es un problema sencillo. Además el número de formas que pueden ser representadas no es tan extenso como sería deseable.

Supercuádricas

De entre todas las superficies algebraicas, la familia de las supercuádricas goza de bastante popularidad ya que con ellas se pueden obtener superficies simétricas con gran facilidad. Fue precisamente por este hecho por lo que, alrededor de 1965, se convirtieron en uno de los primeros modelos de superficie utilizados para definir fuselajes de aeronaves. El nombre de supercuádrica fue propuesto por Alan Barr, [Bar81], para definir a todo un conjunto de superficies definidas en \mathbb{R}^3 obtenidas como el producto esférico de dos curvas definidas en \mathbb{R}^2 ([JLS00]), véase la figura 4.9. Por ejemplo si se define el semicírculo $s(\phi)$, definido en el plano (y, z) :

$$s(\phi) = \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix}, \quad -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2},$$

y el círculo $c(\theta)$, definido en el plano (x, y) :

$$c(\theta) = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}, \quad -\pi \leq \theta \leq \pi,$$

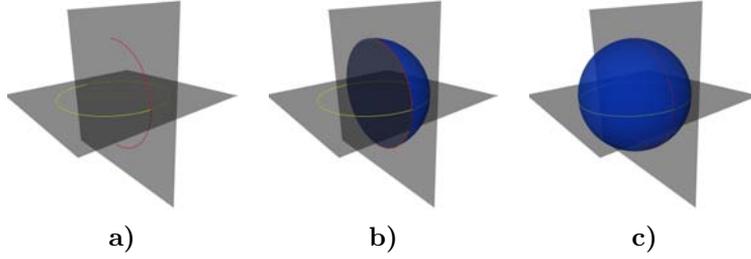


Figura 4.9: Esfera definida como una superficie supercuádrica. a) Dos curvas definidas en un plano. Del producto esférico de ambas se genera una superficie de revolución, b), que finalmente genera la esfera, c).

entonces del producto esférico⁴ de $s(\phi)$ y de $c(\theta)$ se obtiene la superficie esférica $r(\phi, \theta)$:

$$r(\phi, \theta) = s(\phi) \otimes c(\theta) = \begin{pmatrix} \cos \phi \cos \theta \\ \cos \phi \sin \theta \\ \sin \phi \end{pmatrix}, \quad \begin{matrix} -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}, \\ -\pi \leq \theta \leq \pi. \end{matrix}$$

Si en lugar de la ecuación de un semicírculo se utiliza la de una elipse, se obtiene un elipsoide en lugar de una esfera. También se puede hacer uso de una función de exponenciación con signo y utilizar diferentes parámetros para definir las dos curvas. En su trabajo Barr deduce que los superelipsoides son solo un tipo de supercuádrica. En general una superficie supercuádrica se define como:

$$Q(\phi, \theta) = s \begin{pmatrix} a_1 C_1^{\epsilon_1}(\phi) C_2^{\epsilon_2}(\theta) \\ a_2 C_1^{\epsilon_1}(\phi) S_2^{\epsilon_2}(\theta) \\ a_3 S_1^{\epsilon_1}(\phi) \end{pmatrix} \quad (4.4)$$

Siendo $a_1, a_2, a_3 \in \mathbb{R}$ y $s, \epsilon_1, \epsilon_2 \in \mathbb{R}^+ \cup \{0\}$; de manera que s es un factor de escala, a_1, a_2, a_3 son parámetros para controlar la proporcionalidad de la figura y ϵ_1 y ϵ_2 son los coeficientes que controlan la forma, de manera que para valores menores que 1.0 la forma tiende a ser rectangular con las esquinas más o menos redondeadas y para valores mayores que 1.0 la forma se asemeja más a la de la figura 4.10.

En general la forma de una supercuádrica vendrá dada por las funciones $C_1^\epsilon(\phi)$, $C_2^\epsilon(\theta)$, $S_1^\epsilon(\phi)$ y $S_2^\epsilon(\theta)$. Según cuales sean estas funciones se suelen establecer cuatro

⁴El producto esférico, \otimes , se define como: $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \otimes \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} X_1 Y_1 \\ X_1 Y_2 \\ X_2 \end{pmatrix}$.

tipos de supercuádricas:

1. Superelipsoides:

$$\begin{aligned} C_1^{\epsilon_1}(\phi) &= \cos \phi^{\epsilon_1}, & C_2^{\epsilon_2}(\theta) &= \cos \theta^{\epsilon_2}, \\ S_1^{\epsilon_1}(\phi) &= \sin \phi^{\epsilon_1}, & S_2^{\epsilon_2}(\theta) &= \sin \theta^{\epsilon_2}, \\ & & -\frac{\pi}{2} &\leq \phi \leq \frac{\pi}{2}, \\ & & -\pi &\leq \theta \leq \pi. \end{aligned} \tag{4.5}$$

2. Superhiperboloides de una hoja:

$$\begin{aligned} C_1^{\epsilon_1}(\phi) &= \sec \phi^{\epsilon_1}, & C_2^{\epsilon_2}(\theta) &= \cos \theta^{\epsilon_2}, \\ S_1^{\epsilon_1}(\phi) &= \tan \phi^{\epsilon_1}, & S_2^{\epsilon_2}(\theta) &= \sin \theta^{\epsilon_2}, \\ & & -\frac{\pi}{2} &\leq \phi \leq \frac{\pi}{2}, \\ & & -\pi &\leq \theta \leq \pi. \end{aligned} \tag{4.6}$$

3. Superhiperboloides de dos hojas:

$$\begin{aligned} C_1^{\epsilon_1}(\phi) &= \sec \phi^{\epsilon_1}, & C_2^{\epsilon_2}(\theta) &= \sec \theta^{\epsilon_2}, \\ S_1^{\epsilon_1}(\phi) &= \tan \phi^{\epsilon_1}, & S_2^{\epsilon_2}(\theta) &= \tan \theta^{\epsilon_2}, \\ & & -\frac{\pi}{2} &\leq \phi \leq \frac{\pi}{2}, \\ & & -\frac{\pi}{2} &\leq \theta \leq \frac{\pi}{2} \quad (\text{hoja 1}), \\ & & \frac{\pi}{2} &\leq \theta \leq \frac{3\pi}{2} \quad (\text{hoja 2}). \end{aligned} \tag{4.7}$$

4. Supertoroides:

$$\begin{aligned} C_1^{\epsilon_1}(\phi) &= a_4 + \cos \phi^{\epsilon_1}, & C_2^{\epsilon_2}(\phi) &= \cos \phi^{\epsilon_1}, \\ S_1^{\epsilon_1}(\theta) &= a_4 + \sin \theta^{\epsilon_2}, & S_2^{\epsilon_2}(\theta) &= \sin \theta^{\epsilon_2}, \\ & & -\pi &\leq \phi \leq \pi, \\ & & -\pi &\leq \theta \leq \pi, \\ & & a_4 &= \frac{R}{\sqrt{a_1^2 + a_2^2}}. \end{aligned} \tag{4.8}$$

Siendo la exponenciación una operación con signo definida de la siguiente forma:

$$\begin{aligned} a^b &= \operatorname{sgn}(a)|a|^b, \\ \operatorname{sgn}(a) &= \begin{cases} -1, & \text{si } a < 0, \\ 1, & \text{si } a \geq 0 \end{cases}. \end{aligned}$$

Ahora bien, una superficie definida mediante una supercuádrica es regular y simétrica, *a priori* no parece adecuada para representar un objeto deformable. Sin embargo, algunos trabajos ([TM91]) plantean supercuádricas a las que se añade

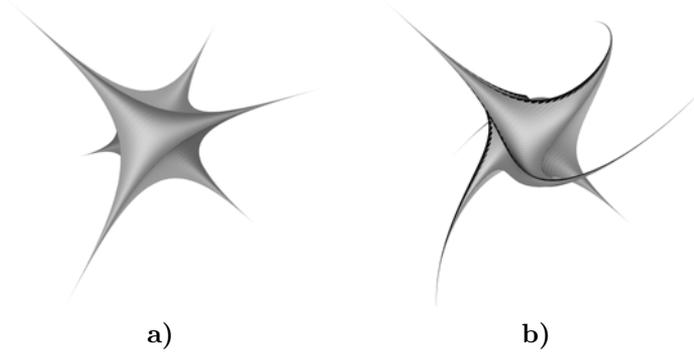


Figura 4.10: Un ejemplo de superficie supercuádrica: un superelipsoide con parámetros $s = a_1 = a_2 = a_3 = 1.0$, $\epsilon_1 = \epsilon_2 = 5.0$. **a)** Forma normal, **b)** Con una deformación.

una componente de perturbación local, de forma que un objeto vendrá expresado por:

$$\tilde{Q}(\phi, \theta) = \mathbf{c} + \mathbf{R}(Q(\phi, \theta) + \mathbf{d}(\phi, \theta)). \quad (4.9)$$

Siendo \mathbf{c} el centro de inercia de la supercuádrica $Q(\phi, \theta)$, \mathbf{R} una matriz de rotación, y $\mathbf{d}(\phi, \theta)$ un campo de desplazamientos sobre la superficie. El resultado obtenido es el de una forma regular excepto en los puntos en donde el campo de desplazamiento modifica la forma de la supercuádrica.

La representación de supercuádricas puede definirse también a partir de funciones implícitas⁵, [BCA95, BCA98]:

1. Superelipsoides:

$$F(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}. \quad (4.10)$$

2. Superhiperboloides de una hoja:

$$F(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} - \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}. \quad (4.11)$$

⁵También es posible representar a las supercuádricas como funciones paramétricas. Por tanto también se cuentan entre los modelos paramétricos.

3. Superhiperboloides de dos hojas:

$$F(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} - \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} - \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}. \quad (4.12)$$

4. Supertoroides:

$$F(x, y, z) = \left(\left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} - \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{2}} - a_4 \right)^{\frac{2}{\epsilon_1}} - \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}. \quad (4.13)$$

En [XY12] este modelo se utiliza juntamente con el modelo de *geones* (un modelo volumétrico que permite extraer propiedades cualitativas) para obtener *geones supercuádricos*, que pueden ser utilizados para definir partes de objetos 3D y además las relaciones topológicas entre las diferentes partes. Por otro lado, en [PSDS09] se propone una solución al problema de detectar contactos entre supercuádricas convexas mediante el uso de ecuaciones implícitas.

Aunque las superficies supercuádricas proporcionan una buena aproximación global, las formas que se pueden representar mediante esta familia de superficies están muy limitadas para poder representar de forma precisa formas complejas, [BCA95, BCA98].

El modelo de supercuádricas también se utiliza para extraer información a partir de imágenes [BVW10]. Véase también [ZYF⁺09], en donde se propone un modelo de segmentación 3D híbrida basado en deformaciones de formas libres (FFD) y supercuádricas.

Hipercuádricas

Las superficies hipercuádricas se proponen como una extensión de las supercuádricas, [Han88]. Una superficie hipercuádrica se define de la siguiente forma:

$$f_{\mathbf{q}}(\mathbf{p}) = \sum_{j=1}^{n \geq 3} |a_j p_x + b_j p_y + c_j p_z + d_j|^{\epsilon_j} = 1, \quad (4.14)$$

siendo $\epsilon_j > 0 \forall j$ y $\mathbf{q} = (a_1, b_1, c_1, d_1, \epsilon_1, \dots, a_n, b_n, c_n, d_n, \epsilon_n)^\top$. Para un valor de n mayor que 3 esta ecuación no tiene una representación explícita. La forma representada se encuentra inscrita en la envolvente convexa definida por los planos $a_j p_x + b_j p_y + c_j p_z + d_j = \pm 1$. El coeficiente ϵ_j es un parámetro de ajuste de la superficie a su envolvente. La cantidad de formas representables mediante

este tipo de superficies es mayor que la cantidad representable con superelipses, aunque solamente es posible definir formas homeomórficas con una esfera, [CC94].

4.3.2. Isosuperficies

Otro posible planteamiento es definir la superficie a partir de una función de potencial, es decir una función que a cada punto del espacio le hace corresponder un valor numérico:

$$F : \mathbb{R}^3 \longrightarrow \mathbb{R}.$$

Con dicha función se puede definir una superficie como el conjunto de puntos del espacio en donde la función toma un cierto valor, v .

$$S_v = \{\mathbf{p} \in \mathbb{R}^3 \mid F(\mathbf{p}) = v\} \quad (4.15)$$

Para una misma función se pueden obtener multitud de superficies simplemente cambiando el valor v , véase la figura 4.11. Las superficies que se generen dependerán de la función F utilizada. Por ejemplo, si se define $F(\mathbf{p}) = p_x^2 + p_y^2 + p_z^2$, entonces para cada posible valor de $v > 0$, S_v definirá la superficie de una esfera de radio \sqrt{v} .

A este tipo de superficies y según sea la función de potencial utilizada se las denomina:

1. *Blobs*: superficies descritas inicialmente en [Bli82] con una función de potencial que estaba inspirada en el comportamiento de los átomos al unirse entre sí para formar moléculas:

$$F(\mathbf{p}) = \sum_i b_i e^{-a_i r_i^2}, \quad (4.16)$$

siendo r_i la distancia del punto \mathbf{p} al centro de la molécula i -ésima. Para cada molécula, se define una campana de Gauss centrada en r_i con una altura a_i y una desviación estándar b_i . El potencial para cada punto del espacio vendrá expresado por la contribución de cada una de las partículas.

2. *Metaballs*: planteamiento algo posterior, [NHK⁺85], define la función de potencial para cada una de las *metaballs* como:

$$w_i(\mathbf{p}) = \begin{cases} d_i \left(1 - 3 \left(\frac{r_i}{b_i}\right)^2\right), & 0 \leq r_i < \frac{b_i}{3}, \\ \frac{3d_i}{2} \left(1 - \frac{r_i}{b_i}\right)^2, & \frac{b_i}{3} \leq r_i < b_i, \\ 0, & b_i \leq r_i, \end{cases} \quad (4.17)$$

en donde r_i es la distancia del punto \mathbf{p} al centro de la i -ésima *metaball*, b_i es su radio y d_i es un coeficiente de ponderación.

Si se define M como un conjunto de *metaballs* que se combinan, un *fusion cluster*, entonces la función de potencial será:

$$F(\mathbf{p}) = \sum_{m_i \in M} w_i(\mathbf{p}). \quad (4.18)$$

3. *Soft objects*: una evolución de las moléculas *blob* ([WMW86a, WMW86b]) en donde la función de potencial es aproximada con un polinomio cuyo valor es nulo a partir de una cierta distancia, el radio de influencia:

$$C_i(\mathbf{p}) = \begin{cases} -\frac{4}{9} \left(\frac{r_i}{R_i}\right)^6 + \frac{17}{9} \left(\frac{r_i}{R_i}\right)^4 - \frac{22}{9} \left(\frac{r_i}{R_i}\right)^2, & r_i \leq R_i, \\ 0, & r_i > R_i. \end{cases} \quad (4.19)$$

En este caso r_i es la distancia del punto \mathbf{p} a cada uno de los puntos clave y R_i es el radio de influencia del punto clave i -ésimo. Por lo tanto si se desea calcular el campo potencial que corresponde a un conjunto de puntos clave, la función de potencial será:

$$F(\mathbf{p}) = \sum_i C_i(\mathbf{p}) \quad (4.20)$$

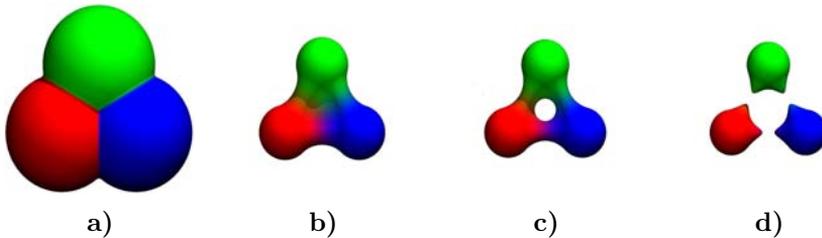


Figura 4.11: Isosuperficie definida a partir de tres esferas equidistantes con un un radio de influencia expresado como porcentaje de sus respectivos radios: en **a**) del 99 %, en **b**) del 45 % , en **c**) del 30 % y en **d**) del 25 %.

Los tres modelos descritos son muy parecidos, aunque existen diferencias notables entre unos y otros. Así, en el modelo de *blobs* para calcular la intensidad

del campo en cada punto del espacio es necesario calcular la aportación de cada una de las moléculas, lo que puede hacer que el cálculo sea demasiado costoso. Originalmente el problema se resolvió determinando una esfera de influencia para cada una de las moléculas, despreciando su efecto en las zonas externas a dicha esfera. En el modelo de *metaballs* y de *soft objects* el problema no es tal debido a la forma de las funciones, ya que éstas convierten el potencial de cada elemento en un valor nulo a partir de una cierta distancia. Sin embargo, una función polinomial, como es el caso de las ecuaciones (4.17) y (4.19), es menos costosa que una función exponencial, caso de (4.16). Los *soft objects* tienen como principal ventaja el hecho de que su función de potencial es la más sencilla de evaluar ya que incluso la raíz cuadrada implícita en el cálculo de las distancias puede verse simplificada gracias a los exponentes de la función.

Posteriormente han aparecido modelos definidos a partir de formas geométricas más complejas que una esfera. En realidad ya en los trabajos seminales se plantean variaciones y posibles extensiones. *A posteriori* se han planteado isosuperficies a partir de un *esqueleto* geométrico más o menos elaborado.

Existe otro aspecto asociado a las isosuperficies: la representación de las mismas. Uno de los algoritmos más utilizados es el de *marching cubes*, [LC87]. Básicamente el algoritmo consiste en dividir el espacio a tratar en pequeños cubos o *voxels*; cuanto más pequeños mayor *suavidad* tendrá la superficie. Entonces se recorren todos los vértices de los cubos determinando si el valor del campo escalar es mayor o menor que el umbral marcado. Una vez realizado el recorrido, cada uno de los ocho vértices de cada cubo tendrá un valor cierto o falso según haya resultado la comparación. En total existen 256 posibles combinaciones de valores cierto y falso de los ocho vértices⁶. Cada combinación determina una posible forma de la superficie buscada en ese cubo concreto. Por lo tanto, lo que se tiene que hacer es sustituir cada cubo por el trozo de superficie que determine su combinación.

El resultado del algoritmo es la isosuperficie de un determinado valor en el campo escalar. La suavidad dependerá del tamaño escogido para los cubos. En la figura 4.12 se muestran dos moléculas *blobs* representadas con el algoritmo descrito y con diferentes tamaños de cubo. Otros planteamientos en esta misma línea aparecen en [Blo88, BS91, She99].

Este tipo de superficies puede ser utilizado para definir contornos deformables, en [CD97] se utilizan isosuperficies definidas a partir de un *esqueleto*.

En trabajos recientes, [BN04], se utiliza esta técnica para modelar nubes a

⁶En la práctica es suficiente con 16 combinaciones diferentes, el resto se puede obtener a partir de transformaciones afines.

partir de un conjunto de partículas.

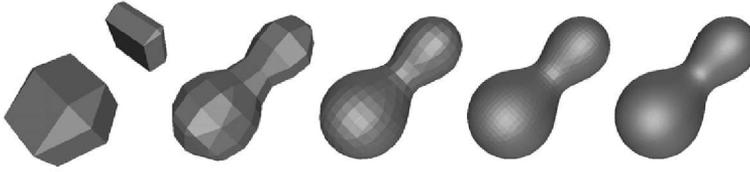


Figura 4.12: Dos *blobs* ([Bli82]) representadas, de izquierda a derecha, con un nivel de detalle mayor.

Otro de los aspectos en los que la utilización de este método ofrece ventajas, es en la representación de objetos a partir de datos de imágenes, o de mecanismos de medida que generan nubes de puntos con un cierto ruido. En [IF03] se hace uso de un mecanismo híbrido, parte modelo explícito y parte implícito, con el que se pueden generar superficies a la vez que se pueden deformar.

4.3.3. Conjuntos de nivel

Los conjuntos de nivel, también conocidos como *level sets*, fueron planteados originalmente por Osher y Sethian, [Set87, OS88, She99]. Su función es la de seguir el contorno de objetos que cambian de forma, este planteamiento es muy similar al propuesto en los modelos de contornos activos introducido en 4.2.2.

Básicamente su planteamiento consiste en ubicar el objeto a deformar en un espacio con un mayor número de dimensiones. Si el objeto cambia de forma, la función de movimiento deberá también proyectarse.

Así pues, si $\Gamma(t)$ representa el objeto, entendido como una hipersuperficie cerrada, en movimiento:

$$\Gamma(t) : [0, \infty) \rightarrow \mathbb{R}^n,$$

entonces sea F la velocidad de desplazamiento de Γ . Puede considerarse que el desplazamiento se realiza en dirección normal al objeto, sea $\pm d$ la distancia al frente de propagación. Suponiendo que el frente de propagación es el conjunto de nivel 0 de una función, ϕ , definida en el espacio de dimensión superior, se tiene que $\phi(x \in \mathbb{R}^n, t = 0) = \pm d$. Entonces se puede plantear una ecuación en derivadas parciales que describa la evolución de ϕ :

$$\phi_t + F|\nabla\phi| = 0,$$

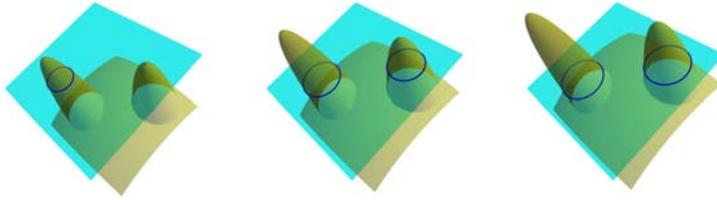


Figura 4.13: Los *level set* visualmente pueden ser interpretados como los contornos de la intersección de dos superficies. En la imagen, de izquierda a derecha puede verse, en la fila superior, cómo una superficie 3D genera una forma 2D (el *level set*) sobre un plano. Según sea el desplazamiento de la superficie 3D, la forma 2D generada cambia.

para un $\phi(x \in \mathbb{R}^n, t = 0)$ dado, F representa la velocidad de Γ , véase la Figura 4.13.

Una de las principales ventajas de este modelo es que el *level set* sigue siendo válido aunque la hipersuperficie cambie su topología. Los demás modelos descritos se basan en la determinación de la posición de algunos puntos del contorno y en calcular la forma de dicho contorno a partir de los puntos calculados. Este planteamiento no suele ser aplicable si se producen cambios en la topología de los objetos, como por ejemplo cuando se producen roturas.

El planteamiento es perfectamente aplicable a superficies. Sin embargo su ámbito de aplicación es el de la segmentación de imágenes, [MRYT08], y el de las imágenes médicas. Se trata pues de un modelo alternativo al de contornos activos. Un reciente estudio sobre los *level set* aplicados a imágenes médicas 2D/3D se encuentra en [SLS⁺02].

En el mismo campo de las imágenes médicas, [MRA04] utiliza los *level set* para identificar y rastrear células biológicas utilizando técnicas de microscopía electrónica. Otra aportación interesante se puede encontrar en [SNM07] en donde se propone un mecanismo automático para la segmentación de pulmones a partir de imágenes médicas.

Además [Act01, AM00a, AM00c, AM00b] utilizan los *level set* para la segmentación de imágenes y la detección de contornos. En estos trabajos se presenta una propuesta basada en operadores morfológicos implementados mediante algoritmos muy eficientes, de manera que se reduce significativamente el coste computacional asociado con este modelo, véase también [LPL07].

Aunque los *level set* son utilizados mayoritariamente en los ámbitos ya indicados, no se restringe a ellos su utilización. En [YXSN11] los *level set* son utilizados para reconstruir modelos 3D complejos utilizando un campo de fuerzas externo

que determina la evolución del sistema. Además, este modelo también puede ser utilizado para describir fluidos, y otros objetos complejos (humo, fuego, ...) [SS03].

4.3.4. Representación de Objetos por Muestreo

La representación de objetos a partir de muestra o SOR (*Sampled Object Representation*) consiste en la representación de objetos utilizando información obtenida a partir de un proceso de muestreo. En este proceso se obtiene un conjunto de muestras de posiciones discretas en el espacio con la intención de obtener información geométrica y física de uno o varios objetos reales, [CCI⁺07].

Esta técnica no permite disponer de la información topológica, geométrica o semántica que es necesaria de cara a poder manipular los objetos de una manera adecuada. Dicha información deberá ser obtenida de alguna otra forma.

Algunas propiedades de esta técnica son:

- Información geométrica limitada. Aunque no tiene por qué ser siempre así, es usual que los datos muestreados no contengan información geométrica y esto, aunque representa una cierta limitación, permite describir fácilmente objetos amorfos – humo, fuego, polvaredas – que son especialmente difíciles de modelar con otros modelos.
- Información topológica limitada. La única información topológica disponible es la de la ordenación espacial o temporal en que se han obtenido las muestras.
- Información semántica limitada. La información muestreada difícilmente puede contener información semántica de los objetos origen de las muestras.
- Canales de datos múltiples. Los datos obtenidos pueden ser de muy diversos tipos: temperaturas, imágenes, sonidos, ...
- Canales de datos multivaluados. Los valores de las muestras pueden dar pie a una cierta incertidumbre debido al tipo de datos que describen.

Según sea la información concreta a tratar este modelo hace uso de otros para poder generar los objetos.

4.4. Modelos paramétricos

Una forma tradicional de plantear la deformación de superficies consiste en determinar que dichas superficies deben ajustarse de acuerdo con ciertas restricciones. Por lo tanto, partiendo de una superficie inicial, más o menos ajustada a

la forma final deseada, deben realizarse las modificaciones oportunas para que la superficie se ajuste lo más posible al resultado final. Las restricciones se plantean en base a funciones matemáticas que se combinan con la definición de la superficies para obtener un sistema de ecuaciones. La definición de la superficie deberá tener suficientes grados de libertad para que dicho sistema de ecuaciones pueda ser resuelto ([Gre94]).

Con este modelo, la representación de los objetos es más restrictiva que la obtenida mediante los modelos explícitos. Ello es debido a las restricciones que deben imponerse. Sin embargo, mediante la modelización paramétrica se simplifica mucho la representación.

La representación paramétrica de un objeto, matemáticamente hablando, es aquella en que cada punto es representado separadamente mediante una función explícita de un conjunto de parámetros independientes. Así, una superficie paramétrica se representa como una aplicación, S , definida en un dominio $\Omega \subseteq \mathbb{R}^2$ de manera que a cada punto, (u, v) , del dominio se le hace corresponder un punto de \mathbb{R}^3 , $(x(u, v), y(u, v), z(u, v))$:

$$\begin{aligned} S : \Omega \subseteq \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ S(u, v) &\longrightarrow (x(u, v), y(u, v), z(u, v)). \end{aligned} \tag{4.21}$$

La representación no es única, de echo existen múltiples representaciones paramétricas para un objeto dado, de todas las posibles interesa la familia de representaciones que [PT97]:

- Sea capaz de representar con precisión todas las formas que el usuario que las utilice requiera.
- Se pueda implementar de forma eficiente en un sistema informático; es especialmente importante que:
 - el cálculo de puntos y derivadas sea eficiente,
 - las operaciones de cálculo sean robustas, sin problemas de error numérico, y
 - no requieran demasiado espacio de almacenamiento.
- Sean simples y bien entendidas desde el punto de vista matemático.

De entre las diferentes representaciones que se han planteado la más sencilla es la de los polinomios, aunque existen numerosos objetos que no son representables mediante éstos.

4.4.1. Supercuádricas

Las supercuádricas, descritas en el apartado 4.3.1, son susceptibles de ser representadas de forma paramétrica. Los coeficientes serán, en este caso, los valores que caracterizan a la supercuádrica en cuestión. En [TM91] se utilizan para representar objetos 3D deformables.

4.4.2. Descomposición modal

La esencia de estos modelos consiste en considerar el objeto a representar como incluido en un espacio donde se pueda expresar como combinación lineal de una base ortonormal de dicho espacio. Los coeficientes de la combinación lineal serán los parámetros que definan al objeto. La ventaja de esta representación es que es posible obtener una aproximación del objeto mediante un subconjunto finito de los coeficientes.

Así pues, si un objeto, f , puede ser considerado como un elemento de un espacio de Hilbert, H , un espacio con un producto escalar, $\langle \cdot, \cdot \rangle$. El objeto, f , se podrá expresar como:

$$f = \sum_{n \in \mathbb{Z}^+} \alpha_n \phi_n,$$

donde $\{\phi_n\}_{n \in \mathbb{Z}^+}$ es una base ortonormal de H y

$$\alpha_i = \langle f, \phi_i \rangle \quad \forall i \in \mathbb{Z}^+.$$

Para un cierto valor, $p \in \mathbb{Z}^+$, una aproximación de f será

$$f \approx \sum_{k=1}^p \alpha_k \phi_k,$$

de manera que el vector $(\alpha_1, \alpha_2, \dots, \alpha_p)$ forma el conjunto de parámetros que definen la forma del objeto con un cierto grado de aproximación. Evidentemente, a mayor número de parámetros mayor será la exactitud de la representación.

En [SD96] se plantea el uso de bases de Fourier, de manera que son los coeficientes de Fourier los que determinan la forma de los objetos. La esencia de este trabajo consiste en plantear el objeto en el espacio de frecuencias. Cada una de las ondas de frecuencia que describen un objeto está formada por un conjunto de armónicos. La representación de objetos de forma paramétrica siguiendo el

planteamiento de la descomposición requiere que los parámetros sean los pesos correspondientes a dichos armónicos. El objeto se podrá aproximar más o menos según el número de armónicos utilizado. En la práctica interesa reducir al máximo dicho número, siempre y cuando el modelo representado se ajuste suficientemente al objeto que se desea representar.

En [PW89] se utiliza este modelo para la representación de modelos dinámicos simplificados, consiguiendo la reducción de la complejidad en la representación y una simplificación gracias a la eliminación de los modos de vibración de mayor frecuencia. El precio a pagar es una cierta limitación en las deformaciones posibles.

Mediante el análisis modal es posible descomponer un objeto deformable en partes útiles, requiriendo solamente una única entrada, [HWAG09]. El análisis modal es capaz de determinar qué partes tienden a moverse rígidamente y es posible determinar el campo de deformaciones asociado al objeto.

Otra aplicación del análisis modal puede verse en [DRBR09], en donde se presenta un método que permite la animación en tiempo real de escenas complejas compuestas por miles de árboles sujetos a la acción de un viento controlado por el usuario. Los principales modos de la deformación son precalculados y utilizados posteriormente para deformar los árboles según la acción del viento. Esta propuesta puede ser implementada de forma eficiente sobre *hardware* gráfico y requiere un esfuerzo computacional reducido.

4.4.3. Superficies de subdivisión

La idea general de subdivisión de curvas y superficies es la siguiente: dado un conjunto inicial de puntos de control \mathbf{p}^0 formando una malla, donde

$$\mathbf{p}^0 = \begin{pmatrix} \vdots \\ p_{-2}^0 \\ p_{-1}^0 \\ p_0^0 \\ p_1^0 \\ p_2^0 \\ \vdots \end{pmatrix},$$

(los valores p_i^0 son puntos de \mathbb{R}^2 o \mathbb{R}^3 dependiendo de si se considera el plano o el espacio) y una matriz denominada matriz de subdivisión S , se halla el siguiente conjunto de puntos de control \mathbf{p}^1 de la forma siguiente: $\mathbf{p}^1 = S\mathbf{p}^0$. Y, en general, se halla la sucesión de puntos \mathbf{p}^k como $\mathbf{p}^{k+1} = S\mathbf{p}^k$.

Los puntos \mathbf{p}^k se consideran una aproximación de nivel k de la curva o superficie.

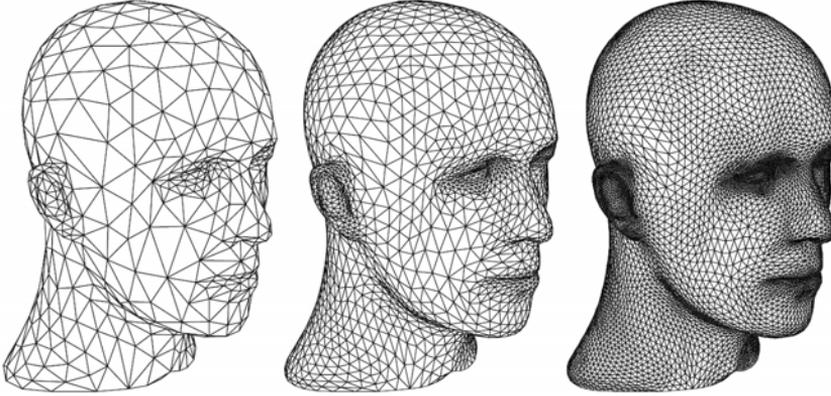


Figura 4.14: Ejemplo de subdivisión de una superficie, mostrando 3 niveles sucesivos de refinamiento [ZS00]. A la izquierda, un mallado triangular inicial como primera aproximación a la superficie. Cada triángulo se divide en 4 de acuerdo a una regla de subdivisión particular. A la derecha de la malla se subdivide una segunda vez siguiendo el mismo esquema.

Cuando se define una subdivisión hay que tener en cuenta:

- Los valores propios de la matriz S . Para que haya convergencia, todos los valores propios λ_i de S deben verificar: $|\lambda_i| \leq 1, \forall i$. Para que haya invariancia afín, S debe tener el valor propio 1 con vector propio $(1, \dots, 1)^\top$.
- En el caso de curvas, asumiendo que $\lambda_0 = 1 < \lambda_1 < \lambda_i$. Supóngase que $\mathbf{p}^0 = \sum_{i=0}^{n-1} a_i \mathbf{x}_i$ donde \mathbf{x}_i es el vector propio correspondiente al valor propio λ_i . Entonces el vector tangente de la curva en \mathbf{p}^k para k suficientemente grande se puede aproximar por a_1 . (a_i son vectores de \mathbb{R}^2 o \mathbb{R}^3).
- En el caso de superficies, si se supone que $\lambda_0 = 1 < \lambda_1 = \lambda_2 < \lambda_i$ y $\mathbf{p}^0 = \sum_{i=0}^{n-1} a_i \mathbf{x}_i$, los vectores que generan el plano tangente en \mathbf{p}^k para k suficientemente grande se pueden aproximar por a_1 y a_2 .
- Regularidad de la superficie generada por subdivisión. Como los puntos de control \mathbf{p}^k forman una malla, se denomina *valencia* del punto de control \mathbf{p}_i^k al número de aristas de la malla que inciden en él. Se considera que el punto de control \mathbf{p}_i^k es extraordinario si tiene una valencia distinta a los demás de la malla donde haya regularidad. En los puntos donde hay regularidad la superficie generada por subdivisión es \mathcal{C}^1 pero en los puntos extraordinarios,

la superficie puede dejar de ser C^1 .

- Mapa característico. A partir de los vectores propios \mathbf{x}_1 y \mathbf{x}_2 (que dependen del esquema de subdivisión) y de los vectores a_1 y a_2 (que dependen de la malla inicial \mathbf{p}^0 de la subdivisión) se puede hacer un estudio local de la superficie generada por subdivisión en \mathbf{p}^k para k suficientemente grande. A partir de dicho estudio, es posible definir el mapa característico que indicará la regularidad de la superficie en un entorno de \mathbf{p}^k .

Los orígenes de este modelo aparecen en [Cha74] en donde se plantea un algoritmo para representar curvas suaves a partir de un polígono de control. Básicamente se trataba de *recortar* las esquinas de dicho polígono de control para generar otro polígono, para cada vértice del polígono original se formaban dos en el nuevo que, a su vez, podía ser recortado nuevamente. El proceso recursivamente hacía que los polígonos de control convergieran hacia una curva cada vez más suave. El proceso se detenía cuando la distancia entre los vértices del polígono era menor que un cierto umbral, por ejemplo la resolución del dispositivo en donde se representaba la figura.

Las primeras referencias a este modelo son [CC78] y [DS78], aunque el concepto ha sido desarrollado ampliamente con posterioridad. De hecho ejemplos de este tipo de objetos pueden ser vistos asiduamente en cualquier película de animación actual, para un ejemplo véase la figura 4.14.

Los métodos basados en subdivisión se pueden clasificar en categorías según cuatro criterios [ZS00]:

- Tipo de regla de refinamiento: división de vértices o división de caras.
- Tipo de malla generada: triangular o cuadrilateral.
- Tipo de esquema: aproximación o interpolación.
- Continuidad de las superficies obtenidas (C^1, C^2, \dots).

A continuación se muestra una clasificación de algunos de los métodos más comúnmente utilizados:

División de caras

	Mallados triangulares	Mallados cuadrangulares
Aproximación	Loop [Loo87]: C^2	Catmull–Clark [CC78]: C^2
Interpolación	Butterfly [DLG90]: C^1	Kobbelt [Kob96]: C^1

División de vértices

Doo–Sabin [DS78]: C^1

Una aplicación de este modelo puede verse en [AWO⁺10], una aproximación a la representación de objetos deformables libre de mallados. Los objetos son representados como un conjunto desordenado de nodos cuya separación se describe de una forma similar al descrito como *fast marching method* [Set99]. La deformación se expresa mediante el uso de técnicas de descomposición modal, 4.4.2. Los resultados muestran que este modelo es capaz de manejar formas complejas en tiempos que permiten pensar en una interactividad con los usuarios y al mismo tiempo que permiten representar formas 3D deformables y en movimiento realistas.

Este modelo es comúnmente utilizado en el contexto de las deformaciones de formas libres (FFD), aunque debido a su elevado coste computacional requiere una conversión final a otro modelo poligonal. Especialmente cuando existe una restricción de tiempo real, como puede ser en el caso de los videojuegos. Para solventar este hándicap es posible utilizar *hardware* específico, como por ejemplo GPU (*Graphic Processing Units*), a pesar de que este tipo de procesadores están especialmente diseñados para acelerar cálculos polinomiales, aunque mucho se ha avanzado en este sentido [LSNC09].

4.4.4. Curvas de Bézier

Las curvas de Bézier fueron estudiadas independientemente por Bézier y por de Casteljaou en la misma época (finales de la década de 1950 para el caso de de Casteljaou y principios de la de 1960 para el de Bézier). Su solución, nunca hasta entonces planteada, se basó en el uso de los *polinomios de Bernstein*, cuyo origen se remonta a principios del siglo XX, y de los denominados polígonos de control. El planteamiento fue revolucionario: no se definía la curva a partir de los puntos que se situaban sobre la curva, sino de algunos puntos cercanos a la misma, el llamado polígono de control; en lugar de modificar la curva directamente se modificaban algunos puntos del polígono de control y, al hacerlo, la curva seguía el movimiento de una forma intuitiva. Los trabajos de ambos investigadores fueron desarrollados con posterioridad y se constató que eran equivalentes.

Una curva de Bézier de grado n se define como

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i \quad 0 \leq u \leq 1. \quad (4.22)$$

Siendo $B_{i,n}(u)$ los polinomios de Bernstein de grado n :

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i} = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}. \quad (4.23)$$

Todos los puntos de control influyen en mayor o menor medida en la forma de la curva, el grado de influencia viene dado por los polinomios de Bernstein, véase la figura 4.15.

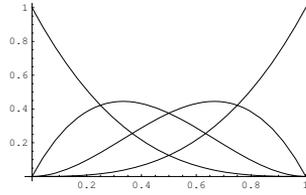


Figura 4.15: Los polinomios de Bernstein establecen la influencia que cada punto de control tiene en cada punto de la curva. En la imagen aparecen los cuatro polinomios de Bernstein que corresponderían a una curva de grado 3.

Los coeficientes geométricos, $\{\mathbf{P}_i \in \mathbb{R}^3, 0 \leq i \leq n\}$, son los denominados *puntos de control*. El polígono que se forma uniendo los puntos de control mediante líneas rectas es el denominado *polígono de control* y resulta ser una aproximación de la curva. Los puntos de control también definen una envolvente convexa de la curva. Otra característica interesante es que la curva pasa por el primer y último punto de control.

Dentro del ámbito de la representación de objetos deformables, en [VW96] se propone la utilización de parches triangulares de Bézier para la representación de superficies construidas a partir de restricciones según un planteamiento variacional.

De entre todas las formas que existen para realizar la representación de superficies paramétricas una de las más utilizadas es la basada en el producto tensorial. En este caso, las funciones básicas bivariantes, funciones de las variables u y v , se construyen como productos de las respectivas funciones básicas univariantes. Los coeficientes geométricos se ordenan en una red, y el producto de tensores tendrá la forma:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m f_i(u) \cdot g_j(v) \cdot \mathbf{b}_{ij}, \quad \begin{cases} \mathbf{b}_{ij} = (x_{ij}, y_{ij}, z_{ij}), \\ a \leq u \leq b, \\ c \leq v \leq d, \end{cases}$$

que expresado en forma matricial queda:

$$S(u, v) = [f_i(u)]^\top [\mathbf{b}_{ij}] [g_j(v)].$$

Siendo $[f_i(u)]^\top$ un vector fila, $[g_j(v)]$ un vector columna, representando ambos las bases univariantes. A su vez, $[\mathbf{b}_{ij}]$ es una matriz $(n + 1) \times (m + 1)$ de puntos tridimensionales.

El dominio de (u, v) es el rectángulo $[a, b] \times [c, d] \subseteq \mathbb{R}^2$, aunque es bastante usual utilizar el rectángulo $[0, 1] \times [0, 1]$ como dominio, ya que ello no supone ninguna restricción y en cambio permite realizar algunas simplificaciones.

En el caso concreto de superficies de Bézier el producto tensorial adoptará la forma:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,p}(u) \cdot B_{j,q}(v) \cdot \mathbf{P}_{ij} \quad 0 \leq u, v \leq 1. \quad (4.24)$$

Los puntos de control forman un poliedro, el poliedro de control, que resulta ser una aproximación lineal de la superficie. Las propiedades que satisfacen las superficies de Bézier son análogas a las descritas para las curvas, extendiéndose para espacios paramétricos de dos dimensiones.

Tanto las curvas como la superficies de Bézier, al ser representaciones polinomiales, tienen una serie de inconvenientes:

- Se requiere un grado muy elevado para el polinomio si la forma a representar tiene que ajustarse a un número elevado de restricciones (en general para satisfacer n restricciones se requiere un polinomio de grado $n - 1$). Lo mismo ocurre si la forma es compleja. Los algoritmos de representación son menos eficientes cuanto mayor es el grado. Además existe el problema que a mayor grado mayor es la inestabilidad numérica.
- El control de la forma no es local. La manipulación de cualquier punto de control afecta a la forma entera de forma poco intuitiva. Esto hace que sean poco atractivas para su utilización en entornos interactivos.
- Existe una gran conjunto de formas que no pueden ser representadas mediante una curva polinomial: las cónicas y cuádricas (círculos, conos, esferas, ...). Una extensión a las formas de Bézier son las formas de Bézier racionales propuestas en [Far83]. Mediante el uso de polinomios de Bézier racionales se solventan algunas de las limitaciones comentadas ya que es posible definir cónicas con este tipo de curvas.

4.4.5. B-Splines

Una solución a los problemas planteados por los esquemas de Bézier pasa por utilizar funciones polinomiales a trozos.

La idea consiste en construir una forma compleja mediante la unión de diferentes trozos con estructura polinomial de grado reducido, sin los inconvenientes derivados de la utilización de polinomios de grado elevado. Así pues, una curva se plantea como un conjunto de segmentos polinomiales unidos uno tras otro. Las uniones entre segmentos son los llamados puntos de ruptura, más conocidos como nudos. Cada uno de los segmentos es independiente de los otros de manera que entre dos nudos consecutivos se define cada uno de los segmentos polinomiales. Las uniones se realizan de manera que se pueda garantizar un cierto grado de continuidad a lo largo de toda la curva (véase la figura 4.16).

Las B-Splines se definen a partir de las denominadas funciones de base B-Spline. Dado un conjunto de valores reales $U = \{a = u_0, u_1, \dots, u_m = b\}$ tales que $u_i \leq u_{i+1} \forall i \in \{0 \dots m - 1\}$, se define recursivamente la i -ésima función de base B-Spline de grado p como⁷:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{si } u_i \leq u \leq u_{i+1}, \\ 0 & \text{en otro caso.} \end{cases} \\ N_{i,p}(u) &= \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u). \end{aligned} \quad (4.25)$$

Cada función de base, $N_{i,p}(u)$, está definida en \mathbb{R} , aunque el interés se centra en el intervalo $[u_0, u_m]$. Es usual que el conjunto de nudos se defina de manera que $u_0 = 0$ y $u_m = 1$.

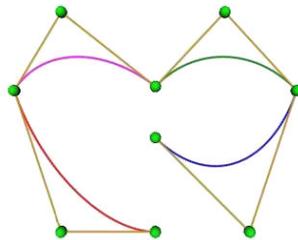


Figura 4.16: Las curvas definidas como formas polinomiales a trozos se forman como resultado de la unión de los segmentos de curva que cada forma polinomial representa. En la imagen aparecen los segmentos diferenciados.

⁷Existe la posibilidad de obtener un cociente de la forma $\frac{0}{0}$ o $\frac{\cdot}{0}$ que, a efectos del desarrollo algorítmico, se considera como 0.

A partir de las funciones de base B-Spline se definen las curvas y superficies B-Spline. En concreto, dado un conjunto de nodos:

$$U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\}, \quad (4.26)$$

y un conjunto de puntos, $\{\mathbf{P}_i \in \mathbb{R}^3, i = 0 \dots n\}$, se define la curva B-Spline de grado p como:

$$C(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i. \quad (4.27)$$

Los polinomios de Bernstein, (4.23), pueden obtenerse a partir de las funciones de base B-Spline, (4.25), si el conjunto de nudos se define como:

$$U = \{\underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1}\},$$

por lo que puede considerarse a las curvas y superficies de Bézier como un caso particular de las B-Spline. Sin embargo la capacidad de representación de las B-Spline es superior. Ello es debido a algunas de las características que las definen. De hecho las formas B-Spline resuelven los problemas comentados para las formas de Bézier:

1. La construcción de las formas B-Spline como formas polinomiales a trozos se realiza precisamente para reducir la complejidad y el grado de las formas de Bézier.
2. El soporte local proviene del hecho de que una determinada función de base, $N_{i,p}(u)$, tiene valores no nulos exclusivamente en el intervalo $[u_i, u_{i+p+1})$. En la práctica, esto hace que la influencia de cada uno de los puntos de control se reduzca a solo un intervalo de la forma. En concreto el punto de control \mathbf{P}_i afecta al intervalo $[u_i, u_{i+p+1})$ de la forma representada. Por lo tanto, el cambio en un punto de control solo afectará a una parte y no a la totalidad de la forma.
3. Otro de los problemas, el de la dificultad a la hora de definir formas complejas, se soluciona debido a que es posible definir formas con ángulos agudos e incluso sin continuidad *visual*, gracias a que la multiplicidad de los puntos de control puede ser mayor que 1. Véase la figura 4.17.

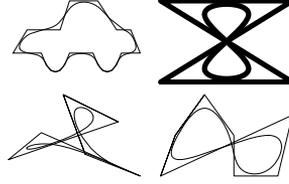


Figura 4.17: Diferentes curvas B-Spline y sus correspondientes polígonos de control.

De la misma forma que mediante el producto tensorial se obtienen superficies de Bézier a partir de curvas, (4.24), también se pueden obtener superficies B-Spline. Una superficie B-Spline, $S(u, v)$, de grados p en una dimensión y q en la otra, con los conjuntos de nudos

$$U = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{1, \dots, 1}_{p+1}\}, V = \{\underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_{q+1}\},$$

y con los puntos de control

$$\{\mathbf{P}_{ij} \in \mathbb{R}^3, i = 0 \dots n, j = 0 \dots m\}$$

de manera que $r = n + p + 1$ y $s = m + q + 1$, se define como:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) \cdot N_{j,q}(v) \cdot \mathbf{P}_{ij} \quad 0 \leq u, v \leq 1. \quad (4.28)$$

También es posible definir objetos sólidos utilizando B-Splines. La forma sería a partir de un producto tensorial tridimensional:

$$S(u, v, w) = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K N_{i,p}(u) \cdot N_{j,q}(v) \cdot N_{k,r}(w) \cdot \mathbf{P}_{ijk} \quad 0 \leq u, v, w \leq 1. \quad (4.29)$$

Este planteamiento es seguido en [RAH97] para la representación a partir de imágenes médicas del ventrículo izquierdo del corazón humano. Para más detalles relativos a curvas y superficies B-Spline véase por ejemplo [Far93] y [PT97].

Las formas B-Spline son utilizadas también en otros ámbitos, como puede ser en algoritmos de registro, es decir, clasificación. En [RAH⁺06] se propone

un método de registro difeomórfico basado en la composición de secuencias de deformaciones de formas libres (FFD) basadas en B-Splines.

Otros artículos interesantes relacionados con este modelo son [GHJCMNB08] en donde se propone una solución analítica basada en un modelo evolutivo para deformar superficies B-Spline. En [GHMN06] se introduce un modelo de evolución dinámico para deformar superficies paramétricas. Finalmente, en [GHJiCMNB10] se propone un método para deformar superficies paramétricas no planas basadas en B-Splines. Para desarrollar este método se hace uso de un funcional de energía expresada con una formulación variacional.

Aunque la variedad de objetos que se pueden representar mediante B-Splines es mucho más amplia que mediante el uso de los esquemas de Bézier, siguen sin poderse representar las cónicas y las cuádricas, formas que aparecen con asiduidad en entornos de simulación, animación, diseño, etc. Por este motivo es necesario buscar un esquema, que siendo más general, permita representar dichas formas.

4.4.6. NURBS

Los objetos NURBS aparecen como solución a una necesidad dentro del mundo del diseño asistido por ordenador, *CAD*, y de la manufactura asistida por ordenador, *CAM*: la necesidad de representación de elementos geométricos diferentes de una forma unificada.

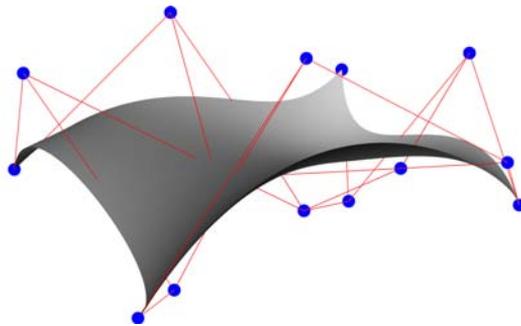


Figura 4.18: Una superficie NURBS, los puntos de control son los puntos azules, conectados por el mallado de control, las líneas rojas.

Considerando la formulación establecida en el apartado 4.4.5, una curva NURBS, $C(u)$, se define como:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) \cdot \omega_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(u) \cdot \omega_i}, \quad (4.30)$$

análogamente, una superficie NURBS, $S(u, v)$, se define como:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) \cdot N_{j,q}(v) \cdot \omega_{ij} \cdot \mathbf{P}_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) \cdot N_{j,q}(v) \cdot \omega_{ij}} \quad 0 \leq u, v \leq 1, \quad (4.31)$$

siendo ω_i y ω_{ij} valores reales denominados pesos.

Los objetos *NURBS* (*Non-Uniform Rational B-Spline*) se han convertido en un estándar *de facto* en el ámbito de la representación de objetos, véase la figura 4.18. Algunas de las características que han promovido este hecho son:

- Permiten representar un amplio rango de formas, desde formas analíticas hasta formas libres, proporcionando un modelo unificador. Esto hace que algoritmos que combinen diferentes objetos, tales como la intersección, unión y demás, sean generales.
- Los algoritmos son rápidos y numéricamente estables, [PT97].
- Se pueden representar discontinuidades tales como doblamientos, marcas, retorcimientos, etc.
- Son invariantes para las transformaciones afines.
- Tienen una semi invarianza proyectiva.
- Tanto las curvas y superficies de Bézier como las B-Spline son casos particulares de las NURBS.
- Son intuitivos, en el sentido en que las modificaciones que se puedan realizar sobre la forma a representar se corresponden con naturalidad a cambios en los elementos que los definen.

Las primeras referencias a las NURBS datan de mediados de los años 1970, de la mano de Versprille, [Ver75]. A principios de los años 1980 aparecen los primeros modeladores basados en NURBS. El interés que despiertan en el ámbito de la aeronáutica, del mundo académico y de las entidades relacionadas con estructuras en general es grande.

La incorporación de NURBS dentro de los estándares gráficos de la época (PHIGS+, IGES) así como sus características especiales han permitido una gran aceptación en el ámbito de CAD/CAM.

Algunas de las características que presentan las NURBS son las siguientes:

- Se sustentan en un modelo matemático que permite representar tanto formas libres (*FFD*) como formas analíticas (cónicas, cuádricas, ...). Al ser una extensión de las formas B-Spline, también es posible representar formas paramétricas de Bézier y B-Spline.
- Manipulando los puntos de control y los pesos se consigue poder representar una gran cantidad de formas. Además la manipulación es local, es decir el cambio en un peso o en un punto de control solo afecta a una parte de la figura, esto hace que la modificación de la forma sea fácil.
- En general, los algoritmos son estables numéricamente.
- La interpretación geométrica es intuitiva, lo que hace más fácil su utilización.
- Existe un extenso conjunto de operaciones que se pueden aplicar a las NURBS, con lo que es posible realizar ajustes y adaptarlas a múltiples formas de una manera sencilla.
- Son invariantes ante transformaciones afines y perspectivas, lo que las convierte en la herramienta ideal para el diseño.

Aunque las ventajas que representa trabajar con NURBS son muchas, este tipo de elementos tiene una serie de inconvenientes:

- La parametrización depende de los pesos asignados, una aplicación incorrecta de los mismos provoca deficiencias en la forma.
- No todas las operaciones geométricas se ajustan adecuadamente para trabajar con NURBS. Determinar la intersección de objetos, o la detección de contactos, son operaciones que requieren algoritmos computacionalmente bastante costosos.
- Algunos algoritmos, en particular el que determina los parámetros correspondientes a un punto de la superficie, son inestables numéricamente y requieren algoritmos con un coste computacional elevado, [PT97].
- Algunas representaciones elementales (círculos, cuadrados, ...) requieren una mayor cantidad de datos utilizando NURBS que utilizando otras formas de representación.

Este modelo ha sido ampliamente utilizado en diversas áreas, como por ejemplo obtención de formas a partir de imágenes [LW94, MBPD00, BARB01], deformaciones libres de forma [LW94, Juh99, LGRG07, CGMP08] y en muchas más [BS96, KML96, LC96], incluso en el ámbito de las imágenes biomédicas. Por ejemplo, en [CG11] se propone un método de representación paramétrica y de medidas funcionales de formas cardíacas 3D basadas en NURBS deformables.

Para mejorar el rendimiento, este modelo puede ser implementado para mostrar superficies NURBS en tiempo real mediante el uso de procesadores gráficos (GPU) [YY09].

4.4.7. T-Splines

Cuando se unen dos superficies NURBS, surge el problema de la combinación de los mallados de control de ambas. Mediante NURBS, es preciso añadir filas y columnas de puntos de control en el mallado en donde sea preciso. El nuevo mallado obtenido permite representar la forma de las dos superficies unidas, aunque es computacionalmente costoso y con frecuencia requiere un esfuerzo adicional a los usuarios. Además, aparece otro problema en el momento en el que se desea añadir un punto de control. En este caso se tiene que añadir al mallado de control una fila o una columna entera debido a la regularidad requerida en dicho mallado.

El modelo de T-Splines [SZBN03] permite representar superficies B-Spline no uniformes con *T-junctions*. Con este modelo es posible insertar puntos en el mallado de control sin que ello provoque que se tenga que añadir toda una fila o columna. Además es factible combinar dos T-Splines sin ningún esfuerzo adicional [SS10]. De esta forma, la regularidad impuesta al mallado de control en el modelo de NURBS puede ser evitada.

El principio de las T-Splines se basa en las denominadas B-Splines basadas en puntos, PB-Splines. La ecuación de una PB-Spline es

$$P(s, t) = \frac{\sum_{i=1}^n P_i \cdot B_i(s, t)}{\sum_{i=1}^n B_i(s, t)} \quad (s, t) \in D, \quad (4.32)$$

en donde P_i son los puntos de control, y $B_i(s, t)$ las funciones de base:

$$B_i(s, t) = N_{i0}^p(s) \cdot N_{i0}^q(t),$$

siendo $N_{i0}^p(s)$ la función de base de grado p asociada con el vector de nudos $s_i = s_{i0}, s_{i1}, \dots, s_{ip}, s_{ip+1}$ y $N_{i0}^q(t)$ la función de base de grado q asociada con el vector de nudos $t_i = t_{i0}, t_{i1}, \dots, t_{iq}, t_{iq+1}$. De esta manera, para cada punto de control se debe suministrar un vector de nudos.

El dominio D en (4.32) es la unión de los dominios de cada punto de control individual:

$$D \subseteq D_1 \cup D_2 \cup \dots \cup D_n = \cup_{i=1}^n D_i,$$

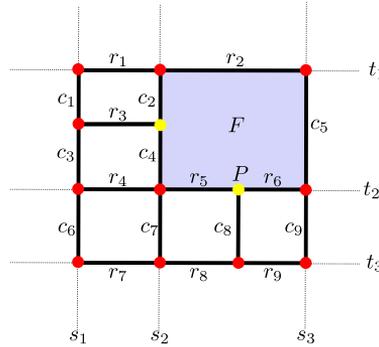


Figura 4.19: Un mallado T-mesh. Las líneas punteadas representan los s -edges y los t -edges. Los puntos son los puntos de control, de los que los de color rojo representan los puntos de control *regulares* y los amarillos son los T-junctions.

D no tiene por qué ser rectangular, solamente debe ser conexo y tal que:

$$\sum_{i=1}^n B_i(s, t) > 0 \quad \forall s, t \in D.$$

La T-Splines son PB-Splines con un mallado de control, T -mesh, que impone algún tipo de organización a los puntos de control. El mallado de control permite manipular más fácilmente a los puntos de control de lo que el planteamiento totalmente libre de las PB-Splines permite. Este mallado de control permite además tener un mecanismo de cálculo de los nudos de control t_i y s_i para cada función de base.

El aspecto fundamental es que cada punto de control en un mallado de control NURBS, exceptuando los puntos de los bordes de dicho mallado, tiene exactamente dos vecinos en cada dirección del mallado, tanto horizontal como verticalmente. Sin embargo, en un mallado de control T -mesh es posible tener un punto de control con solamente un vecino horizontal o vertical, este punto es conocido como unión en T , T -junction.

Un mallado T-mesh es básicamente un mallado rectangular que permite T-junctions. Cada arista en un T-mesh representa un segmento de línea para un valor constante s , un s -edge, o t , un t -edge. Un T-junction es un vértice compartido por un s -edge y dos t -edge o bien por dos s -edge y un solo t -edge, véase la figura 4.19.

Cada arista en un T-mesh está etiquetada en un intervalo de nudos, según las siguientes reglas:

1. La suma de los intervalos de aristas opuestas en cualquier cara debe ser igual. De esta forma, para la cara F de la figura 4.19: $r_2 = r_5 + r_6$ y $c_2 + c_4 = c_5$.
2. Si una T-junction en una arista de una cara puede ser conectada a una arista opuesta de la cara (dividiéndose de esta forma la cara en dos) sin que se transgreda la primera regla, entonces dicha arista puede ser incluida en el T-mesh.

Para determinar los vectores de nudos, s_i y t_i , de un punto de control P_i se utilizan dos líneas en el espacio paramétrico. Sean (s_{ij}, t_{ik}) las coordenadas de P_i en dicho espacio, entonces la línea $R(\alpha) = (s_{ij} + \alpha, t_{ik})$ intersecará con los s -edge y dicha intersección suministrará los nudos s_i . Los nudos t_i se obtienen de forma análoga. Por ejemplo, en la figura 4.19, para el punto de control P , y si el grado es 2, entonces los nudos en la dirección s son: $s = \{s_1, s_2, s_2 + r_5, s_3\}$.

En definitiva, las superficies T-Spline tienen las mismas ventajas que las NURBS, pero requieren un modelo de representación más versátil. Representan pues un modelo de representación muy interesante. En esta línea, en [NTNXBR11] se presenta una alternativa al análisis isogeométrico basado en NURBS que permite un refinamiento local. Además, en [LEB⁺10] se investigan funciones de grado elevado y continuidad en análisis estructural isogeométrico con grandes distorsiones del poliedro de control y con mallados físicos.

En [SZBN03] se propone una extensión de las T-Splines, las denominadas T-NURCCS. Básicamente son una extensión de las superficies NURBSS presentadas en [SZSS98]. Las T-NURCCS son superficies racionales no uniformes compatibles con el modelo de subdivisión de Catmull–Clark (NURCC) que permiten el uso de T-junctions en sus mallados de control, de una forma análoga en la descrita para las T-Splines.

Mediante este planteamiento es posible tener un modelo que combine NURBS y superficies de subdivisión, ya que ambos modelos serían casos particulares del modelo de superficies T-NURCC, y dicho modelo además tendría una formulación simple.

4.5. Conclusiones

Existen múltiples formas en las que los objetos pueden ser representados. Una clasificación visual de los modelos descritos se muestra en la tabla 4.1.

La respuesta a la cuestión acerca de qué modelo debe ser utilizado dependerá de cada caso en particular. Aunque es posible establecer una regla general:

Clasificación			Propiedades		
Categoría	Representación	Modelo	Reconstrucción a partir de datos	Generación sintética	Control local del objeto
Discretos	Discreta	Mallados poligonales	✓	✓	✓
		Sistemas de partículas		✓	
		Masa-resorte	✓	✓	✓
Continuos	Explícita	Cilindros generalizados	✓	✓	
		Contornos activos	✓		
		Objetos continuos		✓	
	Implícita	Superficies algebraicas		✓	
		Isosuperficies	✓	✓	✓
		<i>Level sets</i>	✓	✓	✓
		<i>Sampled Object Representation</i>	✓		
	Paramétrica	Supercuádricas		✓	
		Descomposición modal	✓	✓	
		Subdivisión		✓	✓
		Bézier	✓	✓	
		B-Splines	✓	✓	✓
NURBS		✓	✓	✓	
	T-Splines	✓	✓	✓	

Tabla 4.1: Una clasificación de los modelos de representación de objetos descritos.

si los objetos pueden ser modificados o cambiados para ajustarse a algún tipo de restricción, entonces es preferible tener un control local de la deformación. Además hay que tener presente el hecho de que los objetos puedan ser generados sintéticamente o, por el contrario, puedan ser generados a partir de datos obtenidos de objetos reales.

No todos los modelos pueden ser utilizados en la misma forma. Además cada uno puede ser utilizado con ventajas respecto de los demás para algunos casos particulares. Sin embargo, tal y como se resume en la tabla 4.1: los modelos paramétricos continuos, en particular los modelos basados en B-Splines, NURBS y T-splines son una buena elección y pueden ser considerados como modelos deformables de propósito general.

BSpline4D un nuevo modelo de representación

La representación de objetos mediante NURBS ha sido estudiada profusamente por multitud de investigadores. Aunque la representación de deformaciones basada en modelos físicos sobre una representación NURBS puede dar buenos resultados, el hecho de que las formas NURBS tengan una componente racional hace que su manipulación computacional pueda ser compleja, [QT95, QT96, QT97]. En este capítulo se desarrolla una hipótesis que permite reformular las formas NURBS para que dicha manipulación pueda simplificarse en gran medida.

5.1. Hipótesis

La utilización de una superficie NURBS descrita como una superficie B-Spline en un espacio \mathbb{R}^4 simplifica la representación de las ecuaciones que expresan la deformación de dicha superficie. A dicha representación se la denominará BSpline4D.

5.2. Desarrollo

La demostración de la hipótesis requerirá inicialmente la formulación de las superficies con las nuevas condiciones para acto seguido utilizar esta nueva formulación para expresar las ecuaciones del movimiento. Finalmente restará contrastar los resultados obtenidos con los de los demás modelos estudiados para poder determinar si la hipótesis es correcta.

5.2.1. Formulación

Una superficie NURBS en \mathbb{R}^3 se expresa como un cociente:

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n \mathbf{P}_{i,j} w_{i,j} N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{i,j} N_{i,k}(u) N_{j,l}(v)}, \quad (5.1)$$

siendo $\mathbf{P}_{i,j} = (P_{i,j_x}, P_{i,j_y}, P_{i,j_z}) \in \mathbb{R}^3$ los puntos de control. Los valores $w_{i,j}$ son los pesos asociados a dichos puntos de control ($w_{i,j} \in \mathbb{R}^+$). Por su parte, $N_{i,k}(u)$ y $N_{j,l}(v) \in \mathbb{R}$ son las funciones de base B-Spline de grados k y l respectivamente, véase (4.25).

Ahora bien, esta representación corresponde a la proyección en el espacio \mathbb{R}^3 de una superficie B-Spline definida en \mathbb{R}^4 :

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{P}_{i,j}^w N_{i,k}(u) N_{j,l}(v), \quad (5.2)$$

en donde $\mathbf{P}_{i,j}^w = (P_{i,j_x}^w, P_{i,j_y}^w, P_{i,j_z}^w, P_{i,j_w}^w) \in \mathbb{R}^4$ son los puntos de control. En esta ocasión los valores $w_{i,j}$ de (5.1) aparecen expresados como la cuarta coordenada de los puntos de control, mientras que las otras tres coordenadas se obtienen del producto del peso asociado a cada punto por la correspondiente coordenada; es decir:

$$\begin{aligned} P_{i,j_x}^w &= w_{i,j} \cdot P_{i,j_x}, \\ P_{i,j_y}^w &= w_{i,j} \cdot P_{i,j_y}, \\ P_{i,j_z}^w &= w_{i,j} \cdot P_{i,j_z}, \\ P_{i,j_w}^w &= w_{i,j}. \end{aligned}$$

Por otra parte, $N_{i,k}(u)$ y $N_{j,l}(v) \in \mathbb{R}$, siguen siendo las funciones de base de grados k y l respectivamente. Su valor dependerá exclusivamente del valor del parámetro, u o v , y del vector de nudos asociado. El vector de nudos de una curva B-Spline de grado k con n puntos de control tendría la siguiente forma:

$$U = \left\{ \underbrace{u_0, \dots, u_0}_{k+1}, u_1, u_2, \dots, u_m, \dots, u_m \right\}_{n+k+1} \quad u_i \leq u_{i+1} \quad \forall i \in 0, \dots, m-1. \quad (5.3)$$

Cabe remarcar que el número de nudos, $n + k + 1$ en el ejemplo, está relacionado con el grado y con el número de puntos de control, pero no con el valor de los puntos.

Si se desea obtener $\mathbf{P}_{i,j}$ a partir de $\mathbf{P}_{i,j}^w$, entonces se tiene que utilizar una aplicación, $H : \mathbb{R}^4 \rightarrow \mathbb{R}^3$:

$$H(\mathbf{P}_{i,j}^w) = \begin{cases} \left(\frac{P_{i,j_x}^w}{P_{i,j_w}^w}, \frac{P_{i,j_y}^w}{P_{i,j_w}^w}, \frac{P_{i,j_z}^w}{P_{i,j_w}^w} \right) & \text{si } P_{i,j_w}^w \neq 0, \\ \text{dirección} \left(P_{i,j_x}^w, P_{i,j_y}^w, P_{i,j_z}^w \right) & \text{si } P_{i,j_w}^w = 0. \end{cases} \quad (5.4)$$

Recuérdese que $w_{i,j}$ es directamente la cuarta componente de $\mathbf{P}_{i,j}^w$:

$$w_{ij} = P_{i,j_w}^w.$$

La forma de las superficies depende de los puntos de control que, a su vez, pueden cambiar a lo largo del tiempo.

$$\mathbf{p}(t) \equiv \mathbf{P}_{i,j}^w \quad \forall i \in 0..n, j \in 0..m \quad (5.5)$$

Para hacer explícita la dependencia de la superficie de los puntos de control, que dependen del tiempo t , se expresará la superficie dada en (5.2) de la forma $S(u, v, p)$, donde p es el vector obtenido al concatenar todos los puntos $P_{i,j}^w$ de la forma

$$P = (P_{0,0}^w, P_{0,1}^w, \dots, P_{m,n}^w)^\top,$$

que a su vez dependen del tiempo¹. Por lo tanto, la ecuación (5.2) puede expresarse como:

$$S(u, v, \mathbf{p}) = \mathbf{J} \cdot \mathbf{p}, \quad (5.6)$$

siendo \mathbf{J} la concatenación de las matrices jacobianas de $S(u, v)$, dada por la ecuación (5.2), con respecto a $P_{i,j}^w = (P_{i,j_x}^w, P_{i,j_y}^w, P_{i,j_z}^w, P_{i,j_w}^w)$.

Como $\mathbf{P}_{i,j}^w$ es un vector de cuatro componentes, el jacobiano para cada punto será a su vez una matriz 4×4 cuyos elementos serán todos nulos salvo en la diagonal, que tomará los valores:

¹Obsérvese que la dependencia tendría que ser expresada finalmente como $S(u, v, P_{0,0}^w, P_{0,1}^w, \dots, P_{m,n}^w, t)$. Sin embargo, al expresarla como $S(u, v, p)$ se tiene una forma más compacta y clara, lo que permitirá simplificar, como se verá más adelante, cálculos ulteriores.

$$\frac{\partial S}{\partial \mathbf{P}_{i,j}^w} = \frac{\partial}{\partial \mathbf{P}_{i,j}^w} \left(\sum_{i=0}^m \sum_{j=0}^n \mathbf{P}_{i,j}^w N_{i,k}(u) N_{j,l}(v) \right) = N_{i,k}(u) N_{j,l}(v).$$

De hecho, al ser $\mathbf{P}_{i,j}^w$ un vector de cuatro componentes:

$$\frac{\partial S}{\partial \mathbf{P}_{i,j}^w} = \left(\frac{\partial S}{\partial P_{i,j,x}^w}, \frac{\partial S}{\partial P_{i,j,y}^w}, \frac{\partial S}{\partial P_{i,j,z}^w}, \frac{\partial S}{\partial P_{i,j,w}^w} \right),$$

la expresión debe desarrollarse para cada una de ellas:

$$\frac{\partial S}{\partial P_{i,j}^w} = \frac{\partial S}{\partial P_{i,j,x}^w} = \frac{\partial S}{\partial P_{i,j,y}^w} = \frac{\partial S}{\partial P_{i,j,z}^w} = \frac{\partial S}{\partial P_{i,j,w}^w} = N_{i,k}(u) N_{j,l}(v).$$

Se denotará por $\mathbf{J}_{i,j}$ el elemento del jacobiano correspondiente a punto de control $\mathbf{P}_{i,j}^w$:

$$\mathbf{J}_{i,j} = \begin{pmatrix} \frac{\partial S}{\partial P_{i,j}^w} & 0 & 0 & 0 \\ 0 & \frac{\partial S}{\partial P_{i,j}^w} & 0 & 0 \\ 0 & 0 & \frac{\partial S}{\partial P_{i,j}^w} & 0 \\ 0 & 0 & 0 & \frac{\partial S}{\partial P_{i,j}^w} \end{pmatrix}.$$

el vector \mathbf{J} se obtiene concatenando los diferentes valores $\mathbf{J}_{i,j}$

$$\mathbf{J} = (\mathbf{J}_{0,0} \mathbf{J}_{0,1} \cdots \mathbf{J}_{m,n}).$$

El vector \mathbf{J} es en realidad la siguiente matriz de dimensión $4 \times 4(m+1)(n+1)$:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial S}{\partial P_{0,0}^w} & 0 & 0 & 0 & \cdots & \frac{\partial S}{\partial P_{i,j}^w} & 0 & 0 & 0 & \cdots & \frac{\partial S}{\partial P_{m,n}^w} & 0 & 0 & 0 \\ 0 & \frac{\partial S}{\partial P_{0,0}^w} & 0 & 0 & \cdots & 0 & \frac{\partial S}{\partial P_{i,j}^w} & 0 & 0 & \cdots & 0 & \frac{\partial S}{\partial P_{m,n}^w} & 0 & 0 \\ 0 & 0 & \frac{\partial S}{\partial P_{0,0}^w} & 0 & \cdots & 0 & 0 & \frac{\partial S}{\partial P_{i,j}^w} & 0 & \cdots & 0 & 0 & \frac{\partial S}{\partial P_{m,n}^w} & 0 \\ 0 & 0 & 0 & \frac{\partial S}{\partial P_{0,0}^w} & \cdots & 0 & 0 & 0 & \frac{\partial S}{\partial P_{i,j}^w} & \cdots & 0 & 0 & 0 & \frac{\partial S}{\partial P_{m,n}^w} \end{pmatrix}.$$

Ahora bien, ¿es cierto que $S(u, v, \mathbf{p}) = \mathbf{J} \cdot \mathbf{p}$? La respuesta es afirmativa, como

se puede comprobar en el siguiente desarrollo:

$$\begin{aligned}
 \mathbf{J} \cdot \mathbf{p} &= \begin{pmatrix} \frac{\partial S}{\partial P_{i,j}^w} & 0 & 0 & 0 \\ \dots & 0 & \frac{\partial S}{\partial P_{i,j}^w} & 0 & 0 \\ & 0 & 0 & \frac{\partial S}{\partial P_{i,j}^w} & 0 & \dots \\ & 0 & 0 & 0 & \frac{\partial S}{\partial P_{i,j}^w} \end{pmatrix} \cdot \begin{pmatrix} (P_{0,0,x}^w, P_{0,0,y}^w, P_{0,0,z}^w, P_{0,0,w}^w) \\ \vdots \\ \vdots \\ (P_{m,n_x}^w, P_{m,n_y}^w, P_{m,n_z}^w, P_{m,n_x}^w) \end{pmatrix} \\
 &= \begin{pmatrix} \sum_{i=0}^m \sum_{j=0}^n \frac{\partial S}{\partial P_{i,j}^w} P_{i,j_x}^w \\ \sum_{i=0}^m \sum_{j=0}^n \frac{\partial S}{\partial P_{i,j}^w} P_{i,j_y}^w \\ \sum_{i=0}^m \sum_{j=0}^n \frac{\partial S}{\partial P_{i,j}^w} P_{i,j_z}^w \\ \sum_{i=0}^m \sum_{j=0}^n \frac{\partial S}{\partial P_{i,j}^w} P_{i,j_w}^w \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^m \sum_{j=0}^n N_{i,k}(u) N_{j,l}(v) P_{i,j_x}^w \\ \sum_{i=0}^m \sum_{j=0}^n N_{i,k}(u) N_{j,l}(v) P_{i,j_y}^w \\ \sum_{i=0}^m \sum_{j=0}^n N_{i,k}(u) N_{j,l}(v) P_{i,j_z}^w \\ \sum_{i=0}^m \sum_{j=0}^n N_{i,k}(u) N_{j,l}(v) P_{i,j_w}^w \end{pmatrix} \\
 &= \sum_{i=0}^m \sum_{j=0}^n N_{i,k}(u) N_{j,l}(v) \mathbf{P}_{i,j}^w = S(u, v, \mathbf{p}).
 \end{aligned}$$

Sería también interesante poder demostrar

$$\dot{S}(u, v, p) = \mathbf{J} \cdot \dot{\mathbf{p}}. \quad (5.7)$$

ahora, gracias a la ecuación (5.6) la correspondencia es muy simple:

$$S(u, v, \mathbf{p}) = \mathbf{J} \cdot \mathbf{p} \implies \dot{S}(u, v, \mathbf{p}) = \dot{\mathbf{J}} \cdot \mathbf{p} + \mathbf{J} \cdot \dot{\mathbf{p}},$$

como $\mathbf{J} = N_{i,k}(u) \cdot N_{j,l}(v) \forall i = 0, \dots, m, j = 0, \dots, n$, no depende de t , su derivada respecto al tiempo será idénticamente nula. Por lo tanto:

$$\dot{\mathbf{J}} = \frac{d\mathbf{J}}{dt} = 0, \quad (5.8)$$

y entonces

$$\dot{S}(u, v, p) = \dot{\mathbf{J}} \cdot \mathbf{p} + \mathbf{J} \cdot \dot{\mathbf{p}} = 0 + \mathbf{J} \cdot \dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{p}}.$$

Así, una superficie NURBS se podrá expresar de la forma

$$S(u, v, p) = J \cdot p,$$

donde p es el vector dado por la concatenación de los puntos de control de \mathbb{R}^4 , y \mathbf{J} es la concatenación de las matrices jacobianas de $S(u, v)$ con respecto a $P_{i,j}^w$, que depende exclusivamente de (u, v) y de las funciones de base B-Spline. Además se tiene que

$$\dot{S}(u, v, p) = J \cdot \dot{p}.$$

de ahora en adelante, como ya se conocen las dependencias, se escribirá S en lugar de $S(u, v, p)$ y \dot{S} en lugar de $\dot{S}(u, v, p)$.

De lo anterior se concluye que es posible expresar una superficie NURBS en función de los puntos de control definidos en un espacio \mathbb{R}^4 . Por lo tanto, es razonable plantearse la representación de superficies deformables basadas en el modelo BSpline4D.

Ecuación del movimiento

El siguiente paso a dar es el de escribir las ecuaciones, que rigen el movimiento y deformación de las superficies descritas, en función de sus puntos de control definidos en \mathbb{R}^4 , es decir, las coordenadas generalizadas que se utilizarán serán los puntos de control de la superficie Spline4D.

Partiendo de la ecuación obtenida en 2.17:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{p}_i} \right) - \frac{\partial T}{\partial p_i} + \frac{\partial F}{\partial \dot{p}_i} + \frac{\partial U}{\partial p_i} = f_i, \quad (5.9)$$

se adapta dicha ecuación para contemplar las particularidades de las coordenadas generalizadas, que en definitiva son los puntos de control de la superficie BSpline4D. Los términos T , F y U tienen el siguiente desarrollo:

- Energía cinética T :

$$\begin{aligned} T &= \frac{1}{2} \iint \mu \dot{S}^\top \dot{S} dudv = \frac{1}{2} \iint \mu (\mathbf{J}\dot{\mathbf{p}})^\top (\mathbf{J}\dot{\mathbf{p}}) dudv \\ &= \frac{1}{2} \dot{\mathbf{p}}^\top \iint \mu \mathbf{J}^\top \mathbf{J} dudv \dot{\mathbf{p}} = \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \dot{\mathbf{p}}, \end{aligned} \quad (5.10)$$

siendo

$$\mathbf{M}(\mathbf{p}) = \iint \mu \mathbf{J}^\top \mathbf{J} dudv \quad (5.11)$$

- Disipación de Rayleigh F :

$$\begin{aligned} F &= \frac{1}{2} \iint \gamma \dot{S}^\top \dot{S} dudv = \frac{1}{2} \iint \gamma (\mathbf{J}\dot{\mathbf{p}})^\top (\mathbf{J}\dot{\mathbf{p}}) dudv \\ &= \frac{1}{2} \dot{\mathbf{p}}^\top \iint \gamma \mathbf{J}^\top \mathbf{J} dudv \dot{\mathbf{p}} = \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{D} \dot{\mathbf{p}} \end{aligned} \quad (5.12)$$

con

$$\mathbf{D}(\mathbf{p}) = \iint \gamma \mathbf{J}^\top \mathbf{J} du dv \quad (5.13)$$

- Energía de deformación, U . Del estudio realizado en 3.3 se concluye que utilizar el funcional de energía basado en la Geometría Diferencial conlleva serios problemas numéricos ([TPBF87], [WW92]). Una solución razonable es la de utilizar un funcional alternativo que presenta un mejor comportamiento, el denominado *thin plate under tension*, [WW92, TQ94], que puede ser interpretado como una simplificación del primero:

$$\varepsilon(\mathbf{r}) = \sum_{i,j=1}^2 \int_{\Omega} \left[\alpha_{ij} \frac{\partial \mathbf{r}}{\partial x_i} \cdot \frac{\partial \mathbf{r}}{\partial x_j} + \beta_{ij} \left(\frac{\partial \mathbf{r}^2}{\partial x_i \partial x_j} \right)^2 \right] dx_1 dx_2. \quad (5.14)$$

En dicho funcional los coeficientes α_{ij} representan la tendencia de la superficie a encogerse mientras que los valores β_{ij} representan la tendencia que tiene la superficie a mantenerse plana.

Por lo tanto la energía de deformación en base a (5.14) se desarrolla como sigue:

$$\begin{aligned} U &= \frac{1}{2} \iint \alpha_{1,1} \frac{\partial S^\top}{\partial u} \frac{\partial S}{\partial u} + \alpha_{2,2} \frac{\partial S^\top}{\partial v} \frac{\partial S}{\partial v} + \beta_{1,1} \frac{\partial^2 S^\top}{\partial u^2} \frac{\partial^2 S}{\partial u^2} \\ &\quad + \beta_{1,2} \frac{\partial^2 S^\top}{\partial u \partial v} \frac{\partial^2 S}{\partial u \partial v} + \beta_{2,2} \frac{\partial^2 S^\top}{\partial v^2} \frac{\partial^2 S}{\partial v^2} du dv \\ &= \frac{1}{2} \iint \alpha_{1,1} \frac{\partial (\mathbf{Jp})^\top}{\partial u} \frac{\partial (\mathbf{Jp})}{\partial u} + \alpha_{2,2} \frac{\partial (\mathbf{Jp})^\top}{\partial v} \frac{\partial (\mathbf{Jp})}{\partial v} \\ &\quad + \beta_{1,1} \frac{\partial^2 (\mathbf{Jp})^\top}{\partial u^2} \frac{\partial^2 (\mathbf{Jp})}{\partial u^2} + \beta_{1,2} \frac{\partial^2 (\mathbf{Jp})^\top}{\partial u \partial v} \frac{\partial^2 (\mathbf{Jp})}{\partial u \partial v} \\ &\quad + \beta_{2,2} \frac{\partial^2 (\mathbf{Jp})^\top}{\partial v^2} \frac{\partial^2 (\mathbf{Jp})}{\partial v^2} du dv \end{aligned} \quad (5.15)$$

y como \mathbf{p} no depende explícitamente ni de u ni de v entonces sus derivadas se anularán. Así pues

$$\frac{\partial (\mathbf{J}\mathbf{p})}{\partial u} = \frac{\partial \mathbf{J}}{\partial u} \mathbf{p} + \cancel{\mathbf{J} \frac{\partial \mathbf{p}}{\partial u}} = \frac{\partial \mathbf{J}}{\partial u} \mathbf{p}, \quad (5.16)$$

$$\frac{\partial (\mathbf{J}\mathbf{p})}{\partial v} = \frac{\partial \mathbf{J}}{\partial v} \mathbf{p} + \cancel{\mathbf{J} \frac{\partial \mathbf{p}}{\partial v}} = \frac{\partial \mathbf{J}}{\partial v} \mathbf{p}, \quad (5.17)$$

$$\frac{\partial^2 (\mathbf{J}\mathbf{p})}{\partial u^2} = \frac{\partial^2 \mathbf{J}}{\partial u^2} \mathbf{p} + \cancel{\mathbf{J}_u \frac{\partial \mathbf{p}}{\partial u}} = \frac{\partial^2 \mathbf{J}}{\partial u^2} \mathbf{p}, \quad (5.18)$$

$$\frac{\partial^2 (\mathbf{J}\mathbf{p})}{\partial u \partial v} = \frac{\partial^2 \mathbf{J}}{\partial u \partial v} \mathbf{p} + \cancel{\mathbf{J}_u \frac{\partial \mathbf{p}}{\partial v}} = \frac{\partial^2 \mathbf{J}}{\partial u \partial v} \mathbf{p}, \quad (5.19)$$

$$\frac{\partial^2 (\mathbf{J}\mathbf{p})}{\partial v^2} = \frac{\partial^2 \mathbf{J}}{\partial v^2} \mathbf{p} + \cancel{\mathbf{J}_v \frac{\partial \mathbf{p}}{\partial v}} = \frac{\partial^2 \mathbf{J}}{\partial v^2} \mathbf{p}, \quad (5.20)$$

y la expresión para U queda de la siguiente forma:

$$U = \frac{1}{2} \iint \alpha_{1,1} \mathbf{p}^\top \frac{\partial \mathbf{J}^\top}{\partial u} \frac{\partial \mathbf{J}}{\partial u} \mathbf{p} + \alpha_{2,2} \mathbf{p}^\top \frac{\partial \mathbf{J}^\top}{\partial v} \frac{\partial \mathbf{J}}{\partial v} \mathbf{p} + \beta_{1,1} \mathbf{p}^\top \frac{\partial^2 \mathbf{J}^\top}{\partial u^2} \frac{\partial^2 \mathbf{J}}{\partial u^2} \mathbf{p} \\ + \beta_{1,2} \mathbf{p}^\top \frac{\partial^2 \mathbf{J}^\top}{\partial u \partial v} \frac{\partial^2 \mathbf{J}}{\partial u \partial v} \mathbf{p} + \beta_{2,2} \mathbf{p}^\top \frac{\partial^2 \mathbf{J}^\top}{\partial v^2} \frac{\partial^2 \mathbf{J}}{\partial v^2} \mathbf{p} \, \text{p} \, \text{d}u \, \text{d}v, \quad (5.21)$$

es decir

$$U = \frac{1}{2} \mathbf{p}^\top \cdot \iint \alpha_{1,1} \frac{\partial \mathbf{J}^\top}{\partial u} \frac{\partial \mathbf{J}}{\partial u} + \alpha_{2,2} \frac{\partial \mathbf{J}^\top}{\partial v} \frac{\partial \mathbf{J}}{\partial v} + \beta_{1,1} \frac{\partial^2 \mathbf{J}^\top}{\partial u^2} \frac{\partial^2 \mathbf{J}}{\partial u^2} \\ + \beta_{1,2} \frac{\partial^2 \mathbf{J}^\top}{\partial u \partial v} \frac{\partial^2 \mathbf{J}}{\partial u \partial v} + \beta_{2,2} \frac{\partial^2 \mathbf{J}^\top}{\partial v^2} \frac{\partial^2 \mathbf{J}}{\partial v^2} \, \text{d}u \, \text{d}v \cdot \mathbf{p}, \quad (5.22)$$

si se define

$$\mathbf{K}(\mathbf{p}) = \iint \alpha_{1,1} \frac{\partial \mathbf{J}^\top}{\partial u} \frac{\partial \mathbf{J}}{\partial u} + \alpha_{2,2} \frac{\partial \mathbf{J}^\top}{\partial v} \frac{\partial \mathbf{J}}{\partial v} + \beta_{1,1} \frac{\partial^2 \mathbf{J}^\top}{\partial u^2} \frac{\partial^2 \mathbf{J}}{\partial u^2} \\ + \beta_{1,2} \frac{\partial^2 \mathbf{J}^\top}{\partial u \partial v} \frac{\partial^2 \mathbf{J}}{\partial u \partial v} + \beta_{2,2} \frac{\partial^2 \mathbf{J}^\top}{\partial v^2} \frac{\partial^2 \mathbf{J}}{\partial v^2} \, \text{d}u \, \text{d}v, \quad (5.23)$$

entonces

$$U = \frac{1}{2} \mathbf{p}^\top \mathbf{K} \mathbf{p}. \quad (5.24)$$

Cabe recordar que \mathbf{p} es el conjunto de puntos de control, que dependen de t , y que se expresa como un vector columna:

$$\mathbf{p}(t) = \begin{pmatrix} p_{0,0}(t) \\ \vdots \\ p_{m,n}(t) \end{pmatrix}, \dot{\mathbf{p}}(t) = \begin{pmatrix} \dot{p}_{0,0}(t) \\ \vdots \\ \dot{p}_{m,n}(t) \end{pmatrix},$$

por consiguiente, los vectores transpuestos serán:

$$\begin{aligned} \mathbf{p}(t)^\top &= (p_{0,0}(t) \ \cdots \ p_{m,n}(t)), \\ \dot{\mathbf{p}}(t)^\top &= (\dot{p}_{0,0}(t) \ \cdots \ \dot{p}_{m,n}(t)). \end{aligned}$$

de cara a simplificar la notación, cada coordenada generalizada del vector p se designará a partir de ahora como p_i .

Las fuerzas aplicadas a la superficie también deben expresarse en función de los puntos de control, así pues

$$f_p = \iint \mathbf{J}^\top f(u, v) du dv. \quad (5.25)$$

Ahora bien, cabe preguntarse cual es el valor que deberá tomar la cuarta componente de cada *punto de control* para cada fuerza, ya que éstas se describen en un espacio de tres dimensiones. Toda vez que no está prevista la variación de los nudos ni de otros aspectos geométricos de las superficies BSpline4D, parece razonable también suponer que las fuerzas no afecten a los puntos de control más allá de sus componentes \mathbb{R}^3 . En consecuencia, se puede fijar la cuarta componente de las fuerzas con el valor 0.0, lo que representa que no hay ninguna fuerza externa que afecte a los pesos de los puntos de control de la superficie. Si dichos pesos cambian a lo largo de la simulación será como resultado de la energía interna proveniente de las tensiones que se produzcan como consecuencia de la deformación.

Finalmente, la ecuación (5.9), que expresa el movimiento y deformación de un objeto, aplicada a una superficie parametrizada con las coordenadas generalizadas escogidas, los puntos de control de la superficie BSpline4D, y después de los cálculos realizados, puede ser escrita de la siguiente forma:

$$\frac{d}{dt} \frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \dot{\mathbf{p}})}{\partial \dot{p}_i} - \frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \dot{\mathbf{p}})}{\partial p_i} + \frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{D} \dot{\mathbf{p}})}{\partial \dot{p}_i} + \frac{\partial (\frac{1}{2} \mathbf{p}^\top \mathbf{K} \mathbf{p})}{\partial p_i} = f_p. \quad (5.26)$$

A continuación se desarrollan cada uno de los sumandos de la parte izquierda de la igualdad (5.26). Primero se desarrollarán las derivadas parciales respecto a p_i y a \dot{p}_i .

- Primer sumando, $\frac{\partial(\frac{1}{2}\dot{\mathbf{p}}^\top \mathbf{M}\dot{\mathbf{p}})}{\partial \dot{p}_i}$. Puede desarrollarse:

$$\frac{\partial(\frac{1}{2}\dot{\mathbf{p}}^\top \mathbf{M}\dot{\mathbf{p}})}{\partial \dot{p}_i} = \frac{1}{2} \frac{\partial \dot{\mathbf{p}}^\top}{\partial \dot{p}_i} \mathbf{M}\dot{\mathbf{p}} + \frac{1}{2} \dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial \dot{p}_i} \dot{\mathbf{p}} + \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \frac{\partial \dot{\mathbf{p}}}{\partial \dot{p}_i}$$

entonces, analizando el valor que toman $\frac{\partial \mathbf{M}}{\partial \dot{p}_i}$, $\frac{\partial \dot{\mathbf{p}}}{\partial \dot{p}_i}$ y $\frac{\partial \dot{\mathbf{p}}^\top}{\partial \dot{p}_i}$:

$$\begin{aligned} \frac{\partial \mathbf{M}}{\partial \dot{p}_i} &= 0, \\ \frac{\partial \dot{\mathbf{p}}}{\partial \dot{p}_i} &= \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{componente } i\text{-ésima}, \\ \frac{\partial \dot{\mathbf{p}}^\top}{\partial \dot{p}_i} &= (0 \quad \dots \quad 1 \quad \dots \quad 0), \\ &\quad \quad \quad \uparrow \\ &\quad \quad \quad \text{componente } i\text{-ésima} \end{aligned}$$

se obtiene:

$$\frac{\partial(\frac{1}{2}\dot{\mathbf{p}}^\top \mathbf{M}\dot{\mathbf{p}})}{\partial \dot{p}_i} = \frac{1}{2} (0 \quad \dots \quad 1 \quad \dots \quad 0) \mathbf{M}\dot{\mathbf{p}} + \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}.$$

Al ser \mathbf{M} una matriz simétrica, véase la ecuación (5.11), obtenida por el producto de un vector por sí mismo, entonces:

$$\begin{aligned} \forall i \in 0 \dots (m+1)(n+1), \\ (\mathbf{M}\dot{\mathbf{p}})_i &= \sum_{k=0}^{(m+1)(n+1)} M_{i,k} \dot{p}_k \\ &= \sum_{k=0}^{(m+1)(n+1)} \dot{p}_k M_{k,i} = \left(\dot{\mathbf{p}}^\top \mathbf{M} \right)_i \end{aligned}$$

y por lo tanto:

$$\frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \dot{\mathbf{p}})}{\partial \dot{p}_i} = (0 \quad \dots \quad 1 \quad \dots \quad 0) \mathbf{M} \dot{\mathbf{p}}.$$

- Segundo sumando, $\frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \dot{\mathbf{p}})}{\partial p_i}$. También tiene un desarrollo:

$$\frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \dot{\mathbf{p}})}{\partial p_i} = \frac{1}{2} \frac{\partial \dot{\mathbf{p}}^\top}{\partial p_i} \mathbf{M} \dot{\mathbf{p}} + \frac{1}{2} \dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial p_i} \dot{\mathbf{p}} + \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \frac{\partial \dot{\mathbf{p}}}{\partial p_i},$$

que se simplifica de la siguiente forma:

$$\frac{\partial \dot{\mathbf{p}}}{\partial p_i} = \frac{\partial}{\partial p_i} \frac{d\mathbf{p}}{dt} = \frac{d}{dt} \frac{\partial \mathbf{p}}{\partial p_i} = \frac{d}{dt} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = 0,$$

y otro tanto ocurre con $\frac{\partial \dot{\mathbf{p}}^\top}{\partial p_i}$. Entonces

$$\frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{M} \dot{\mathbf{p}})}{\partial p_i} = \frac{1}{2} \dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial p_i} \dot{\mathbf{p}}.$$

- Tercer sumando, $\frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{D} \dot{\mathbf{p}})}{\partial p_i}$. Se puede desarrollar de la misma forma con la que se procede con el primero, por lo tanto

$$\frac{\partial (\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{D} \dot{\mathbf{p}})}{\partial p_i} = (0 \quad \dots \quad 1 \quad \dots \quad 0) \mathbf{D} \dot{\mathbf{p}}.$$

- Cuarto sumando, $\frac{\partial (\frac{1}{2} \mathbf{p}^\top \mathbf{K} \mathbf{p})}{\partial p_i}$. Tiene un desarrollo algo diferente:

$$\frac{\partial (\frac{1}{2} \mathbf{p}^\top \mathbf{K} \mathbf{p})}{\partial p_i} = \frac{1}{2} \frac{\partial \mathbf{p}^\top}{\partial p_i} \mathbf{K} \mathbf{p} + \frac{1}{2} \mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} + \frac{1}{2} \mathbf{p}^\top \mathbf{K} \frac{\partial \mathbf{p}}{\partial p_i},$$

en este caso

$$\frac{\partial \mathbf{p}}{\partial p_i} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{componente } i\text{-ésima},$$

$$\frac{\partial \mathbf{p}^\top}{\partial p_i} = (0 \quad \dots \quad 1 \quad \dots \quad 0),$$

↑
componente i -ésima

y por lo tanto

$$\frac{\partial (\frac{1}{2}\mathbf{p}^\top \mathbf{K}\mathbf{p})}{\partial p_i} = \frac{1}{2} (0 \quad \dots \quad 1 \quad \dots \quad 0) \mathbf{K}\mathbf{p} + \frac{1}{2}\mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} + \frac{1}{2}\mathbf{p}^\top \mathbf{K} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix},$$

por el mismo razonamiento que el utilizado en los sumandos primero y tercero, al ser \mathbf{K} simétrica

$$\frac{\partial (\frac{1}{2}\mathbf{p}^\top \mathbf{K}\mathbf{p})}{\partial p_i} = (0 \dots 1 \dots 0) \mathbf{K}\mathbf{p} + \frac{1}{2}\mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p}.$$

Finalmente hay que considerar la derivada respecto a t del primer sumando:

$$\begin{aligned} \frac{d}{dt} \frac{\partial (\frac{1}{2}\dot{\mathbf{p}}^\top \mathbf{M}\dot{\mathbf{p}})}{\partial \dot{p}_i} &= \frac{d}{dt} ((0 \dots 1 \dots 0) \mathbf{M}\dot{\mathbf{p}}) \\ &= (0 \dots 1 \dots 0) \dot{\mathbf{M}}\dot{\mathbf{p}} + (0 \dots 1 \dots 0) \mathbf{M}\ddot{\mathbf{p}} \end{aligned}$$

Por lo tanto, la ecuación que rige el movimiento para una coordenada generalizada, p_i , será:

$$\begin{aligned} (0 \dots 1 \dots 0) \mathbf{M}\ddot{\mathbf{p}} + (0 \dots 1 \dots 0) \dot{\mathbf{M}}\dot{\mathbf{p}} - \frac{1}{2}\dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial p_i} \dot{\mathbf{p}} \\ + (0 \dots 1 \dots 0) \mathbf{D}\dot{\mathbf{p}} + (0 \dots 1 \dots 0) \mathbf{K}\mathbf{p} + \frac{1}{2}\mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} = f_{p_i}. \end{aligned}$$

De donde se puede inferir la representación matricial siguiente

$$\mathbf{M}\ddot{\mathbf{p}} + \dot{\mathbf{M}}\dot{\mathbf{p}} - \left[\dots \frac{1}{2}\dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial p_i} \dot{\mathbf{p}} \dots \right]^\top + \mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\mathbf{p} + \left[\dots \frac{1}{2}\mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} \dots \right]^\top = f_p,$$

o lo que es lo mismo:

$$\mathbf{M}\ddot{\mathbf{p}} + \mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\mathbf{p} = f_p + \left[\dots \frac{1}{2} \dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial p_i} \dot{\mathbf{p}} \dots \right]^\top - \dot{\mathbf{M}}\dot{\mathbf{p}} - \left[\dots \frac{1}{2} \mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} \dots \right]^\top. \quad (5.27)$$

La ecuación (5.27) posee algunos términos que deben ser estudiados. Primeramente, la expresión

$$\left[\dots \frac{1}{2} \mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} \dots \right]^\top$$

se anula, como se puede ver a continuación. Primeramente, hay que recordar que $\mathbf{p}=\mathbf{p}(t)$ no depende de los puntos (u,v) . Además, $\mathbf{K}(\mathbf{p})$ toma su valor según lo expresado en la ecuación (5.23)

$$\frac{1}{2} \mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} = \frac{1}{2} \mathbf{p}^\top \frac{\partial}{\partial p_i} \left(\iint \left(\alpha_{1,1} \mathbf{J}_u^\top \mathbf{J}_u + \alpha_{2,2} \mathbf{J}_v^\top \mathbf{J}_v + \beta_{1,1} \mathbf{J}_{uu}^\top \mathbf{J}_{uu} + \beta_{1,2} \mathbf{J}_{uv}^\top \mathbf{J}_{uv} + \beta_{2,2} \mathbf{J}_{vv}^\top \mathbf{J}_{vv} \right) dudv \right) \mathbf{p}. \quad (5.28)$$

Como formalmente se demostrará, cada uno de los sumandos de la integral se anula.

$$\frac{1}{2} \mathbf{p}^\top \frac{\partial \iint \alpha_{1,1} \mathbf{J}_u^\top \mathbf{J}_u dudv}{\partial p_i} \mathbf{p} = \frac{1}{2} \iint \alpha_{1,1} \mathbf{p}^\top \frac{\partial (\mathbf{J}_u^\top \mathbf{J}_u)}{\partial p_i} \mathbf{p} dudv,$$

$$\begin{aligned} \mathbf{p}^\top \frac{\partial (\mathbf{J}_u^\top \mathbf{J}_u)}{\partial p_i} \mathbf{p} &= \mathbf{p}^\top \frac{\partial \mathbf{J}_u^\top}{\partial p_i} \mathbf{J}_u \mathbf{p} + \mathbf{p}^\top \mathbf{J}_u^\top \frac{\partial \mathbf{J}_u}{\partial p_i} \mathbf{p} = \mathbf{p}^\top \left(\frac{\partial \mathbf{J}_u}{\partial p_i} \right)^\top \mathbf{J}_u \mathbf{p} + \mathbf{p}^\top \mathbf{J}_u^\top \frac{\partial \mathbf{J}_u}{\partial p_i} \mathbf{p} \\ &= \left(\frac{\partial \mathbf{J}_u}{\partial p_i} \mathbf{p} \right)^\top \mathbf{J}_u \mathbf{p} + \mathbf{p}^\top \mathbf{J}_u^\top \frac{\partial \mathbf{J}_u}{\partial p_i} \mathbf{p}, \end{aligned}$$

ahora bien, \mathbf{J} no depende de p_i para ningún i , por lo tanto

$$\frac{\partial \mathbf{J}}{\partial p_i} = 0, \quad (5.29)$$

entonces

$$\frac{\partial \mathbf{J}_u}{\partial p_i} \mathbf{p} = \frac{\partial}{\partial p_i} \frac{\partial \mathbf{J}}{\partial u} \mathbf{p} = \frac{\partial}{\partial u} \frac{\partial \mathbf{J}}{\partial p_i} \mathbf{p} = 0,$$

y, así,

$$\begin{aligned}
 \mathbf{p}^\top \frac{\partial (\mathbf{J}_u^\top \mathbf{J}_u)}{\partial p_i} \mathbf{p} &= 0 \\
 \Rightarrow \frac{1}{2} \iint \alpha_{1,1} \mathbf{p}^\top \frac{\partial (\mathbf{J}_u^\top \mathbf{J}_u)}{\partial p_i} \mathbf{p} \, dudv &= 0 \\
 \Rightarrow \frac{1}{2} \mathbf{p}^\top \frac{\partial}{\partial p_i} \left(\iint \alpha_{1,1} \mathbf{J}_u^\top \mathbf{J}_u \, dudv \right) \mathbf{p} &= 0,
 \end{aligned}$$

y lo mismo ocurre con los otros sumandos, en consecuencia:

$$\frac{1}{2} \mathbf{p}^\top \frac{\partial \mathbf{K}}{\partial p_i} \mathbf{p} = 0.$$

Los otros dos sumandos correspondientes a la parte derecha de la ecuación (5.27)

$$\left[\dots \frac{1}{2} \dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial p_i} \dot{\mathbf{p}} \dots \right]^\top - \dot{\mathbf{M}} \dot{\mathbf{p}},$$

también se anulan. En primer lugar, de la definición de M vista en la ecuación (5.11) se deduce que, siempre que μ no dependa de t :

$$\dot{\mathbf{M}} = \iint \mu \dot{\mathbf{J}}^\top \mathbf{J} \, dudv + \iint \mu \mathbf{J}^\top \dot{\mathbf{J}} \, dudv,$$

siendo la derivada de \mathbf{J} con respecto al tiempo idénticamente nula, ecuación (5.8), se determina que también lo será $\dot{\mathbf{M}}$ por lo que

$$\dot{\mathbf{M}} \dot{\mathbf{p}} = 0,$$

el sumando restante también se anula, si se tiene en consideración que

$$\frac{\partial \mathbf{M}}{\partial p_i} = \frac{\partial}{\partial p_i} \left(\iint \mu \mathbf{J}^\top \mathbf{J} \, dudv \right) = \iint \mu \frac{\partial \mathbf{J}^\top}{\partial p_i} \mathbf{J} \, dudv + \iint \mu \mathbf{J}^\top \frac{\partial \mathbf{J}}{\partial p_i} \, dudv,$$

y por (5.29):

$$\frac{\partial \mathbf{M}}{\partial p_i} = \iint \mu \mathbf{0} \mathbf{J} \, dudv + \iint \mu \mathbf{J}^\top \mathbf{0} \, dudv = 0.$$

Eliminando los términos que se anulan en la ecuación que rige el movimiento y deformación de la superficie, (5.27), dicha ecuación puede expresarse como

$$\mathbf{M}\ddot{\mathbf{p}} + \mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\mathbf{p} = \mathbf{f}_p. \quad (5.30)$$

Como puede observarse, la ecuación (5.30) tiene una forma equivalente a la ecuación (2.22). Por consiguiente, todos los estudios y desarrollos que se han mostrado en el capítulo 3 aplicados a la ecuación (2.22) son aplicables a la ecuación (5.30).

5.3. Deformación del modelo BSpline4D

Una vez determinada la forma en que se pueden representar las superficies mediante el modelo de BSpline4D y obtenida la ecuación que representa el movimiento y la deformación de dichas superficies, cabe estudiar la forma en que todo ello se puede utilizar para obtener la simulación de la deformación a lo largo del tiempo de dichas superficies.

5.3.1. Discretización temporal

En el capítulo 3 se han estudiado algunos de los métodos de integración que pueden ser aplicados para la resolución numérica de sistemas de ecuaciones diferenciales como la obtenida en el apartado anterior, (5.30). En dicho estudio se concluye que, de entre todos los modelos estudiados, el modelo de las diferencias centradas implícito es el más aconsejable. La cuestión que surge es si para un modelo de representación como el descrito en el presente capítulo seguirá siendo así. La respuesta requerirá estudiar el comportamiento de los métodos de integración con el nuevo modelo.

5.3.2. Discretización espacial

El modelo de representación basado en BSpline4D comparte las propiedades de las formas B-Spline. Por ello, cada punto de una superficie B-Spline corresponde a una combinación lineal de las funciones de base. En consecuencia la superficie es infinitamente diferenciable en cualquier punto de la superficie que se corresponda con unas coordenadas del espacio paramétrico tales que no coincidan con un nodo en ninguna de las dos dimensiones paramétricas. Además, en aquellos puntos que coincidan con un nodo, la superficie es diferenciable $p - k$ veces en la dimensión a la que pertenece el nodo, siendo p el grado correspondiente y k la multiplicidad de dicho nodo. Así pues, es posible asegurar la continuidad

necesaria en todo punto de la superficie. A tal fin será preciso establecer que la superficie tenga un grado suficientemente alto en cada dimensión.

Como se ha visto en (5.6) la expresión de las superficies se puede representar como el producto del jacobiano, formado exclusivamente a partir de las funciones de base B-Spline, y los puntos de control. En el desarrollo de la ecuación del movimiento se ha obtenido la expresión (5.30) utilizando como coordenadas generalizadas los puntos de control de la superficie.

De cara a la resolución de la expresión (5.30) es precisa la resolución de las integrales definidas descritas en (5.11), (5.13), (5.23) y (5.25). Las expresiones que definen las funciones de base son tales que no permiten determinar una forma explícita para las integrales, por lo tanto se hace preciso utilizar métodos de integración numérica para poder resolver las integrales.

Cuadratura de Gauss–Legendre

Los métodos de cuadratura más precisos son los que encajan en la fórmula de cuadratura de Gauss [DB74, BF10]. La principal característica de este tipo de métodos es que los puntos que se toman en consideración no se encuentran repartidos uniformemente a lo largo del intervalo de integración. La forma general que describe la cuadratura de Gauss se puede representar mediante la siguiente expresión:

$$\int_{-1}^1 f(x)dx \approx \sum_{i=1}^n w_i f(x_i), \quad (5.31)$$

para un conjunto discreto x_i de valores y un conjunto w_i de pesos asociados.

El primer aspecto a considerar es que, si el intervalo sobre el que se debe realizar la integración no es $[-1, 1]$, es preciso realizar un cambio de la variable de integración que permita ajustar el intervalo. Así pues, si la integral a calcular se expresa como

$$\int_a^b f(x)dx,$$

se puede realizar el cambio

$$x = \frac{b-a}{2}z + \frac{a+b}{2},$$

de manera que se obtiene

$$dx = \frac{b-a}{2}dz, \quad (5.32)$$

de manera que

$$\int_a^b f(x)dx = \int_{-1}^1 \frac{b-a}{2} f\left(\frac{b-a}{2}z + \frac{a+b}{2}\right) dz, \quad (5.33)$$

con lo que (5.31) se puede reescribir

$$\int_{-1}^1 f(x)dx \approx \frac{b-a}{2} \sum_{i=1}^n w_i f\left(\frac{b-a}{2}z_i + \frac{a+b}{2}\right). \quad (5.34)$$

Existen varias opciones para determinar los valores x_i y los pesos asociados, w_i . La utilización de las raíces de los polinomios de Legendre es una forma común de hacerlo. Un polinomio de Legendre de grado n se puede expresar como

$$P_n(x) = \frac{1}{2^n n!} \frac{d(x^2 - 1)^n}{dx^n} \quad (5.35)$$

Estos valores también pueden ser obtenidos de forma recurrente mediante la siguiente expresión:

$$P_n(x) = \begin{cases} 1, & \text{si } n = 0, \\ x, & \text{si } n = 1, \\ \frac{(2n-1)x}{n} P_{n-1}(x) - \frac{n-1}{n} P_{n-2}(x), & \text{si } n \geq 2. \end{cases} \quad (5.36)$$

En la figura 5.1 se puede observar la representación de los polinomios de Legendre hasta el grado 7. Para cada uno de los puntos, z_i , es preciso calcular el valor del correspondiente peso, w_i , para ello se utiliza la expresión:

$$w_i = \frac{2}{(1 - z_i^2) (P'_n(z_i))^2}. \quad (5.37)$$

El número de puntos y pesos utilizado para realizar la aproximación de las integrales determinará la precisión del resultado obtenido. En general, una cuadratura de Gauss–Legendre de $n + 1$ puntos es exacta para funciones polinomiales de grado menor o igual a $2n + 1$. Así pues, si la deformación representada requiere una cierta continuidad C^p , $p > 0$, entonces será suficiente con establecer un número de puntos, n , proporcional:

$$p = 2n + 1 \quad \Rightarrow \quad n = \left\lceil \frac{p - 1}{2} \right\rceil$$

A modo de ejemplo, si la superficie debe tener una continuidad C^4 sería suficiente con 2 puntos para calcular la cuadratura. Cabe destacar aquí que lo descrito

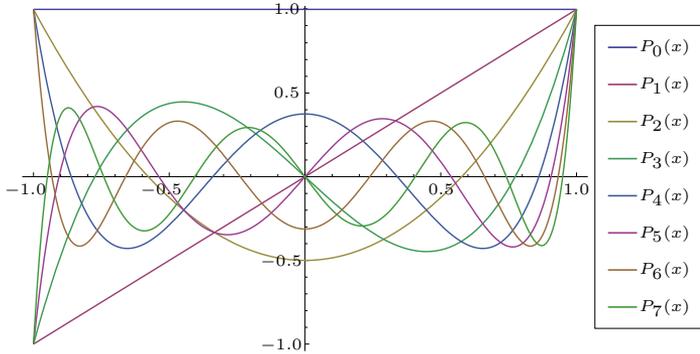


Figura 5.1: Representación de los polinomios de Legendre hasta el grado 7. Puede observarse la simetría respecto al origen de los polinomios y también la propiedad que tienen consistente en que entre dos raíces consecutivas de un polinomio de grado n existe una única raíz del polinomio de grado $n - 1$.

respecto del grado hace referencia al *grado* de la expresión polinómica que describe la superficie y no al concepto de grado de la representación BSpline4D. En definitiva, la forma de la superficie determina el número de puntos de cuadratura que deben ser utilizados para poder calcular la deformación. Idealmente, en cada paso del cálculo de la simulación se podría estimar cual es el número de puntos a utilizar para que el resultado sea el más exacto posible, otra opción es la de permitir que el usuario establezca el número de puntos de cuadratura que desea utilizar.

Para probar el comportamiento del modelo en función del número de puntos de cuadratura se ha diseñado un experimento en el que se simula la deformación de un plano, sujeto por sus cuatro esquinas y sujeto a un campo gravitatorio perpendicular. Para unos mismos valores ($\alpha_{11} = \alpha_{22} = 1.0, \beta_{ij} = 0.5$), se ha probado una simulación de 2 segundos, con un incremento de tiempo Δt de 0.01 segundos. El plano se ha representado mediante una superficie BSpline4D de grado 3 y 11 puntos de control en cada dimensión. Se han realizado tres pruebas, cada una con un número de puntos de cuadratura diferente: 4×4 , 8×8 y 11×11 puntos. En las tres pruebas el comportamiento ha sido estable, aunque el resultado ha sido diferente; a mayor número de puntos mejor ha sido el resultado. En la figura 5.2 se muestra el radio espectral de las tres pruebas y en la tabla 5.1 se muestra la evolución a lo largo del tiempo de los tres experimentos.

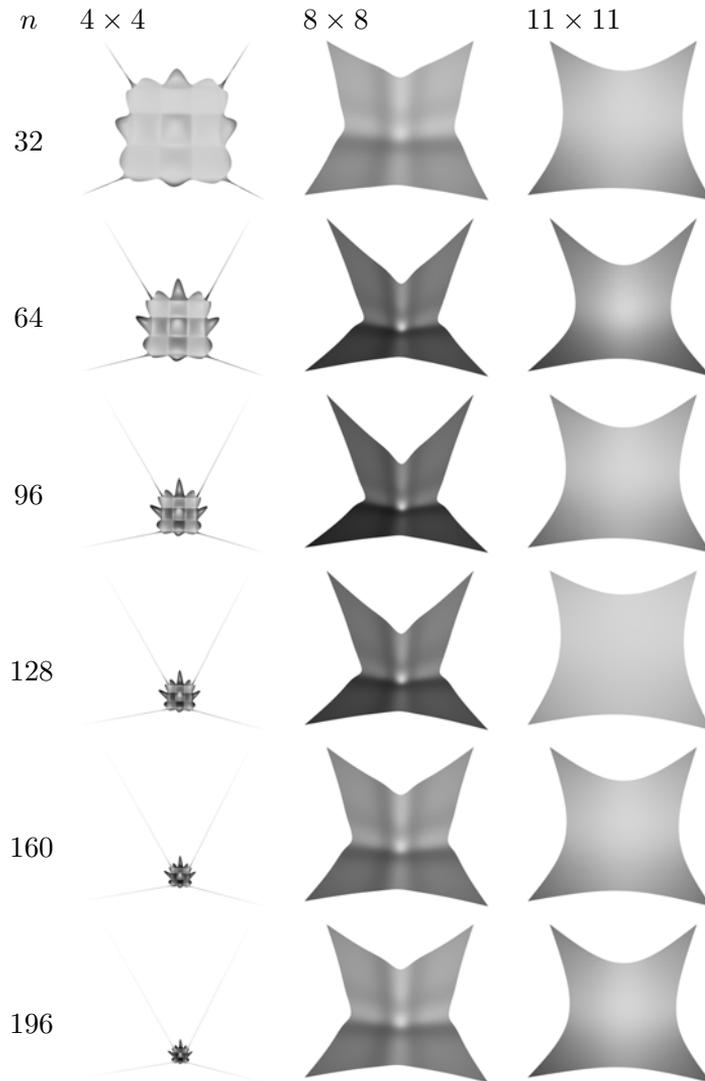


Tabla 5.1: El resultado de la simulación puede verse afectado por el número de puntos de cuadratura de Gauss–Legendre que se utilicen. En las imágenes se muestra el resultado obtenido para una misma simulación en función del número de puntos de cuadratura utilizados, de izquierda a derecha: 4×4 , 8×8 y 11×11 puntos. Puede verse que los mejores resultados se obtienen para este último caso.

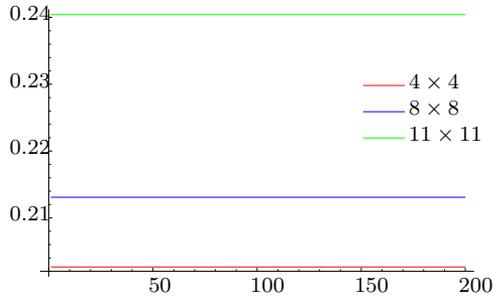


Figura 5.2: En función del número de puntos de cuadratura utilizados, el radio espectral es uno u otro. En cualquier caso, el radio espectral se mantiene constante a lo largo de toda la simulación. En la imagen se muestra el radio espectral de los tres experimentos descritos en el texto.

5.3.3. Imposición de restricciones

A la hora de resolver problemas de ingeniería es frecuente tener que imponer algún tipo de restricción a algunas de las variables que forman el espacio de configuraciones del problema a tratar. Usualmente las restricciones suelen representar la imposición de un determinado valor o el establecimiento de algún tipo de relación entre ciertas variables de manera que la solución del sistema de ecuaciones obtenida sea tal que dichas condiciones se mantengan.

En el presente trabajo las restricciones se han utilizado para fijar la posición de algunos puntos de control de manera que no queden afectados por la deformación. Esto ha permitido desarrollar los experimentos descritos en los que se ha fijado uno o más puntos.

La idea es representar el sistema mediante la expresión

$$\frac{1}{2}p^{\top}Hp = p^{\top}f, \quad (5.38)$$

y el conjunto de restricciones como

$$Ap = b. \quad (5.39)$$

Las restricciones geométricas descritas en (5.39) se imponen como funciones lineales sobre los grados de libertad explícitos del sistema original sin restricciones. Cada función representa una restricción, y expresa una combinación lineal de los valores de las coordenadas generalizadas. El correspondiente valor de b representa el valor que dicha combinación lineal debe tomar. El sistema no tiene por qué tener una solución única, aunque de hecho es deseable que así sea, toda vez que este hecho será el que permita que el método sea más útil.

En definitiva, la idea es expresar (5.38) como

$$\frac{1}{2}\tilde{p}^\top \tilde{H}\tilde{p} = \tilde{p}^\top \tilde{f}, \quad (5.40)$$

de manera que se puedan obtener los valores de p a partir de los valores obtenidos para \tilde{p} minimizando la expresión (5.38) de forma que se cumpla la igualdad (5.39) al mismo tiempo. Lo interesante es que (5.40) representa un sistema de ecuaciones sin restricciones, que será mínimo cuando su derivada sea 0, por tanto

$$\frac{\partial \tilde{p}^\top \tilde{H}\tilde{p} - \tilde{p}^\top \tilde{f}}{\partial \tilde{p}} = \tilde{H}\tilde{p} - \tilde{f} = 0. \quad (5.41)$$

Existen diferentes formas de obtener \tilde{p} , y existen una gran cantidad de trabajos las describen ([Bat96, Cel90, CW92, FB97, MT92, MT93, MT96, PB88, QT95, QT96, QT96, Sha13, TQ94, TWK88, WFB87] entre muchos otros). A continuación se describen algunos de los más utilizados.

Los multiplicadores de Lagrange

El método de los multiplicadores de Lagrange es probablemente el método más general que puede ser aplicado. La idea es crear un nuevo funcional que combine de alguna forma (5.38) y (5.39). Para ello se añade para cada restricción, representada por las filas de A , lo que se conoce como un multiplicador de Lagrange, λ_i ; que, en definitiva, viene a ser una nueva variable del sistema, es decir un nuevo grado de libertad. Por lo tanto se trata de minimizar:

$$\frac{1}{2}p^\top Hp - p^\top f + (Ap - b)^\top \lambda, \quad (5.42)$$

por lo tanto su derivada respecto de p y de λ deber ser igual a 0 y, en consecuencia,

$$\begin{aligned} Hp + A^\top \lambda &= f \\ Ap &= b \end{aligned} \quad (5.43)$$

que expresado de forma matricial queda

$$\left(\begin{array}{c|c} H & A^\top \\ \hline A & 0 \end{array} \right) \begin{pmatrix} p \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ b \end{pmatrix} \quad (5.44)$$

La resolución del sistema obtenido es tal que asegura unos valores de p compatibles a la vez con (5.38) y con (5.39).

Los principales inconvenientes de este planteamiento, [Met96], son que las restricciones se deben cumplir en el instante inicial y, principalmente, que el sistema obtenido en un instante dado puede tener una matriz que no sea definida positiva, con lo que la resolución del sistema se podría complicar. Además, al sistema se le añaden tantas incógnitas como restricciones haya.

Por otra parte, el resultado es preciso y además permite contemplar todo tipo de restricciones, tanto lineales como no lineales.

El método de penalización

Éste es un método parecido al de los multiplicadores de Lagrange, pero a diferencia de éste, no se añaden nuevas variables sino que se fuerza cada restricción imponiendo un peso de valor mucho mayor que cualquier valor relacionado con la variable en cuestión. El sistema a minimizar queda de la siguiente forma:

$$\frac{1}{2}p^\top Hp - p^\top f + \frac{\alpha}{2}(Ap - b)^2 \quad (5.45)$$

Con lo que el mínimo se consigue si

$$Hp - f + \alpha Ap - \alpha b = 0$$

o, lo que es lo mismo:

$$(H + \alpha A)p = f + \alpha b \quad (5.46)$$

Cuanto mayor sea el valor de α tanto mejor se ajustará el resultado obtenido a las restricciones que se hayan impuesto. El problema surge del hecho que aumentar el valor de α puede provocar que el sistema resultante esté mal condicionado y si el valor no es suficientemente elevado entonces no tiene por qué obtenerse un resultado que cumpla estrictamente con las restricciones. A su favor tiene el hecho que no se añaden más variables.

El método de proyección sobre el espacio nulo

Otra posible forma de imponer restricciones lineales mediante la proyección del sistema de ecuaciones en el subespacio que contiene las soluciones que son compatibles con las restricciones impuestas, [Bjö96], usado por algunos autores, [Cel90, CW92, TQ94]. El método de proyección sobre el espacio nulo trata de encontrar la solución a un problema de mínimos cuadrados con restricciones de igualdad, [Bjö96]. El problema a tratar se considerará escrito en la forma (5.41):

$$\min_p \|Hp - f_p\|_2 \quad (5.47a)$$

$$\text{sujeto a } Ap = b, \quad (5.47b)$$

donde $H \in \mathbb{R}^{n \times n}$ y $A \in \mathbb{R}^{r \times n}$, siendo r el número de restricciones. Como el sistema debe tener algún movimiento se tiene que $r < n$. El problema tiene solución solamente en el caso en que (5.47b) sea consistente. Una condición suficiente para asegurar la consistencia es que las filas de A sean linealmente independientes. La solución de (5.47) es única si y solo si se dan los siguientes rangos:

$$\begin{aligned} \mathcal{R}(A) &= r \\ \mathcal{R} \begin{pmatrix} H \\ A \end{pmatrix} &= n. \end{aligned} \quad (5.48)$$

Asumiendo que el rango de A es r ($\mathcal{R}(A) = r$), se puede calcular una matriz ortogonal $Q_A \in \mathbb{R}^{n \times n}$ tal que

$$Q_A^\top A^\top = \begin{pmatrix} R_A \\ 0 \end{pmatrix}, \quad (5.49)$$

donde $R_A \in \mathbb{R}^{r \times r}$ es una matriz triangular superior y no singular. Sea

$$Q_A = (Q_1, Q_2) \quad Q_1 \in \mathbb{R}^{n \times r}, \quad Q_2 \in \mathbb{R}^{n \times (n-r)},$$

entonces el espacio nulo de A es igual al rango de Q_2 , $\mathcal{N}(A) = \mathcal{R}(Q_2)$, es decir, Q_2 proporciona una base ortogonal para el espacio nulo de A . Cualquier vector que satisfaga (5.47b) puede entonces ser representado como

$$p = p_1 + Q_2 p_2, \quad p_1 = A^\dagger b = Q_1 \left(R_A^\top \right)^{-1} b, \quad (5.50)$$

Al ser A una matriz no cuadrada no es posible obtener su matriz inversa, con A^\dagger se representa la pseudoinversa de A :

$$A^\dagger = (A^\top A)^{-1} A^\top$$

En definitiva, p_1 es una solución particular de (5.47b) y p_2 es libre y su valor puede ser calculado a partir de (5.47a). Por lo tanto

$$Hp - f_p = Hp_1 + HQ_2 p_2 - f_p, \quad p_2 \in \mathbb{R}^{n-r}, \quad (5.51)$$

con lo que el sistema que debe ser resuelto es el sistema reducido

$$\min_{p_2} \| (HQ_2) p_2 - (f_p - Hp_1) \|_2. \quad (5.52)$$

Sea p_2 la mínima distancia de (5.52):

$$p_2 = (HQ_2)^\dagger (f_p - Hp_1), \quad (5.53)$$

o, lo que es lo mismo

$$p_2 = ((HQ_2)^T(HQ_2))^{-1}(HQ_2)^T(f_p - Hp_1),$$

y sea p definido como (5.50). Al ser p_1 ortogonal a Q_2p_2 , se comprueba que

$$\|p\|_2^2 = \|p_1\|_2^2 + \|Q_2p_2\|_2^2 = \|p_1\|_2^2 + \|p_2\|_2^2,$$

y por lo tanto p es la solución mínima del problema de mínimos cuadrados con restricciones de igualdad planteado.

Si el rango de la matriz

$$C = \begin{pmatrix} A \\ H \end{pmatrix} Q_A = \begin{pmatrix} R_A^\top & 0 \\ HQ_1 & HQ_2 \end{pmatrix}$$

es igual a n , entonces todas las columnas de C deben ser linealmente independientes. Por lo tanto el rango de HQ_2 es $n - r$. En tal caso la condición expresada en (5.48) se cumple y, en consecuencia, la solución obtenida es única.

En definitiva, A determina dos subespacios complementarios. El primero es aquél formado por los vectores de dimensión r obtenidos por las combinaciones lineales de las columnas de A y el segundo será el formado por los vectores ortonormales a las columnas de A para un cierto valor y , es decir

$$A^0y = 0$$

siendo A^0 la matriz del subespacio nulo de A .

La idea es que el vector de estados, las variables, representado por p debe ser *atraído* al subespacio representado por (5.47b) y deberá *desplazarse* por dicho subespacio hasta determinar el menor valor de (5.47a) en el subespacio. Las soluciones se restringen a un conjunto de las soluciones posibles para el problema planteado sin restricciones.

Con este método se obtiene un sistema bien condicionado aunque el radio espectral de H sea grande, siempre y cuando el radio espectral de A y de HQ_2 sean pequeños, [Bjö96]. Experimentalmente se ha constatado que es así.

Como el sistema representado por (5.39) es no homogéneo toda solución x será de la forma

$$x = A^0y + x_0 \quad (5.54)$$

siendo x_0 una solución particular, y un conjunto reducido de valores, de dimensión r , y A^0 el espacio nulo de A , de dimensión $n \times r$.

Para obtener el valor de A^0 se puede proceder aplicando Gauss–Jordan, se trata de convertir $Ap = b$ en $\tilde{A}p = \tilde{b}$ siendo

$$\tilde{A} = (Id_{r \times r} | R_{r \times (n-r)}) \quad (5.55)$$

Sea p_p una solución particular. Entonces toda solución p de $\tilde{A}p = \tilde{b}$ será de la forma $p = p_p + y$ siendo $y \in \mathcal{N}(A)$ ya que

$$\tilde{A}(p_p + y) = \tilde{A}p_p + \tilde{A}y = \tilde{b}.$$

Como $\tilde{A}y = 0$ entonces

$$(Id_{r \times r} | R) \begin{pmatrix} y_r \\ y_{n-r} \end{pmatrix} = 0,$$

siendo y_r las r primeras componentes de y y y_{n-r} las últimas $n - r$ componentes de y . Por lo tanto

$$y_r + Ry_{n-r} = 0$$

y así

$$y_r = -Ry_{n-r}.$$

Así pues,

$$y = \begin{pmatrix} -Ry_{n-r} \\ y_{n-r} \end{pmatrix} = \begin{pmatrix} -R \\ Id \end{pmatrix} y_{n-r} = A^0 y_{n-r},$$

siendo A^0 una matriz $n \times (n - r)$.

Por lo tanto, toda solución de $\tilde{A}p = \tilde{b}$ será de la forma $p = p_p + A^0 y_{n-r}$, siendo y_{n-r} libre.

A continuación se deberá resolver el problema

$$\min_{p \text{ solución de } \tilde{A}p = \tilde{b}} \|Hp - f_p\|_2, \quad (5.56)$$

que es lo mismo que

$$\min_{y_{n-r}} \|HA^0 y_{n-r} - (f_p - Hp_p)\|_2, \quad (5.57)$$

cuya solución es

$$(HA^0)^\top (HA^0) y_{n-r} = (HA^0)^\top (f_p - Hp_p),$$

o lo que es lo mismo

$$y_{n-r} = (HA^0)^\dagger (f_p - Hp_p). \quad (5.58)$$

Así pues el valor de p será:

$$p = p_p + A^0 (HA^0)^\dagger (f_p - Hp_p) \quad (5.59)$$

Este método tiene la ventaja de reducir el número de grados de libertad, y por lo tanto de incógnitas a calcular. Además se preserva la estabilidad numérica y da resultados exactos.

Un inconveniente del método es que las restricciones no deben influir las unas sobre las otras. Con todo y con eso este método es preferible al método de penalización, [Cel90].

Este modelo ha sido el escogido como método de representación de restricciones lineales para el desarrollo de las pruebas del modelo de representación de superficies BSpline4D.

Además, si las restricciones (5.39) se mantienen, el valor de A^0 permanece inalterado, el valor de p_p tampoco cambiará, a no ser que manteniendo el valor de A en (5.39) se modifique el valor de b . Por lo tanto se puede plantear una implementación que tenga precalculado el valor de A^0 y p_p con lo que aumentaría el rendimiento.

5.4. Estabilidad

De cara a determinar la estabilidad numérica del modelo propuesto se ha procedido a evaluarlo mediante un conjunto de experimentos análogo al descrito en el apartado 3.3.5. Con el conocimiento previo del comportamiento de los diferentes modelos de integración se ha reducido el número de modelos estudiado, en este caso se ha procedido a evaluar el modelo de deformación solamente con el método de las diferencias centradas, el método de las diferencias centradas implícito y con el método de Newmark β .

Se ha realizado un experimento equivalente en la medida de lo posible al descrito en el apartado 3.3.5. Por lo tanto las características comunes a las tres simulaciones son:

- el objeto a deformar es un cuadrado unitario, plano,
- el objeto se ha descrito como una superficie BSpline4D de grado 3 en cada dimensión con 4×4 puntos de control,

- cada punto tiene un coeficiente de amortiguación y una masa unitarios,
- las fuerzas aplicadas corresponden a un campo gravitatorio normal a la superficie del objeto y que afecta a todo el objeto exceptuando aquellos puntos que hayan sido fijados,
- el campo empieza a actuar en el instante $t = 0s$; anteriormente no se aplica ningún tipo de fuerzas al objeto y por lo tanto éste permanece en reposo con velocidad y aceleración nulas.
- el tiempo total simulado es de 2 segundos.

De cara a mantener los mismos criterios cada una de las simulaciones se ha realizado con dos valores diferentes para Δt . Los valores han sido $\Delta t = 0.01s$ y $\Delta t = 0.001s$.

Las tres simulaciones realizadas han sido:

- 1) deformación del objeto sujeto por sus cuatro esquinas con un coeficiente $\alpha_{11} = \alpha_{22} = 1.0$, $\alpha_{12} = 0.0$ y $\beta_{ij} = 0.5 \forall i, j \in \{1, 2\}$. Los resultados pueden verse en la tabla 5.2 para $\Delta t = 0.01$ y 5.3 para $\Delta t = 0.001$.
- 2) Deformación del objeto sujeto por todo su contorno con un coeficiente $\alpha_{11} = \alpha_{22} = 1.0$, $\alpha_{12} = 0.0$ y $\beta_{ij} = 0.5 \forall i, j \in \{1, 2\}$. Los resultados pueden verse en la tabla 5.4 para $\Delta t = 0.01$ y 5.5 para $\Delta t = 0.001$.
- 3) Deformación del objeto sujeto por todo su contorno con un coeficiente $\alpha_{11} = \alpha_{22} = 1.0$, $\alpha_{12} = 0.0$ y $\beta_{ij} = 0.0 \forall i, j \in \{1, 2\}$. Los resultados pueden verse en la tabla 5.6 para $\Delta t = 0.01$ y 5.7 para $\Delta t = 0.001$.

Una vez mas, se ha intentado deformar un objeto comprobando el comportamiento en función de lo *sujeto* que esté (comparación de las simulaciones 1 y 2) y en función de su rigidez (comparación de las simulaciones 2 y 3).

Como puede observarse en las figuras, el resultado obtenido es el mismo con independencia del intervalo de tiempo considerado y del método de discretización temporal utilizado.

Para determinar la estabilidad numérica del método descrito se ha procedido a realizar un conjunto de pruebas. Concretamente se ha estudiado el comportamiento variando el grado de la superficie, el número de puntos de control y el número de puntos de cuadratura de Gauss–Legendre utilizados. En las tablas 5.8 se muestran las diferentes combinaciones de grado y puntos de control que se han probado. Para los tres métodos de integración utilizados se han realizado pruebas con diferente número de puntos de cuadratura de Gauss–Legendre, concretamente se han utilizado las combinaciones de 4×4 , 6×6 y 8×8 puntos de cuadratura.

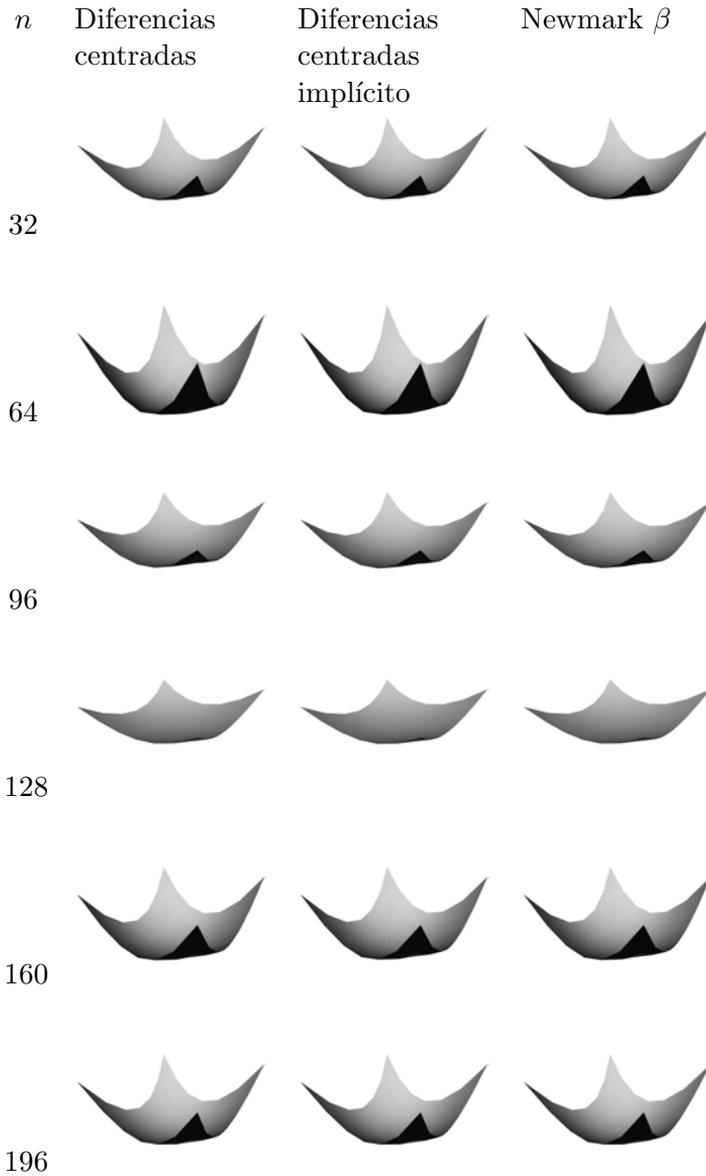


Tabla 5.2: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, resultados para el caso 1). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$.

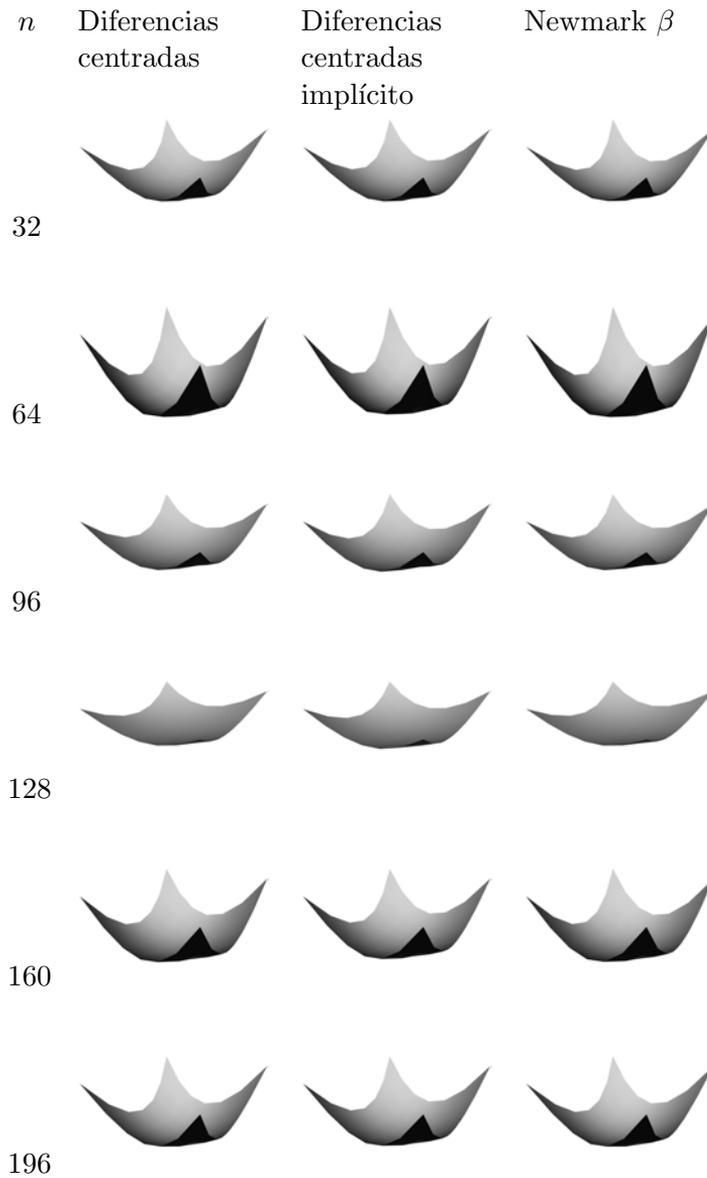


Tabla 5.3: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, resultados para el caso 1). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$.

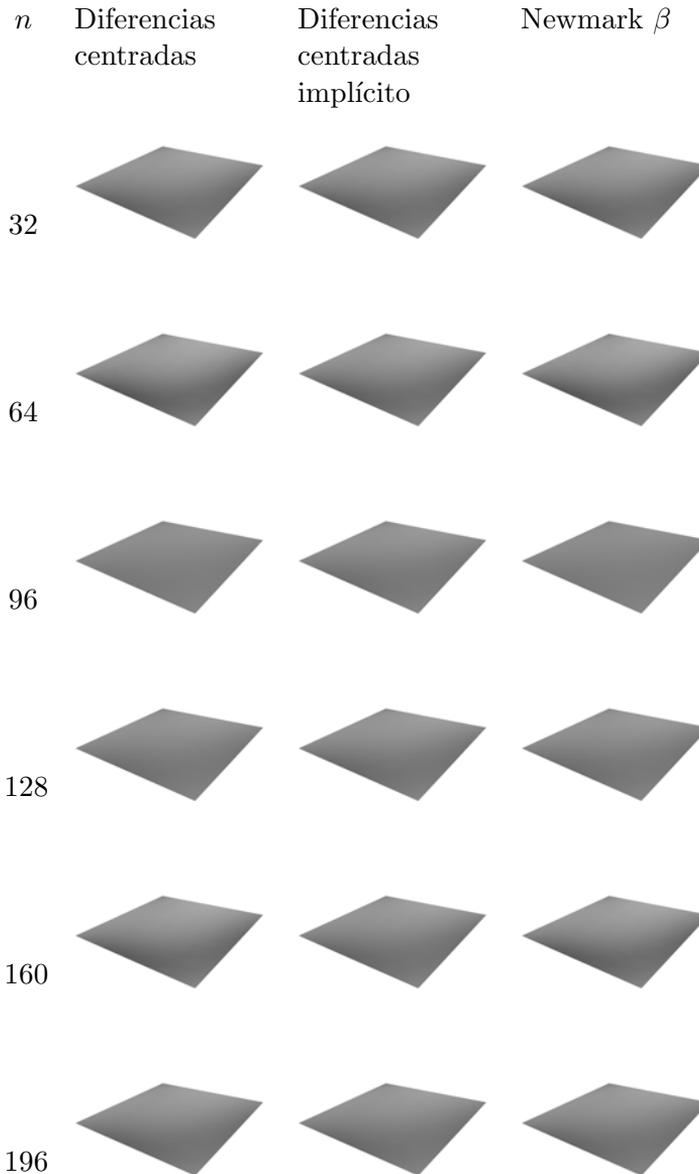


Tabla 5.4: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, resultados para el caso 2). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$.

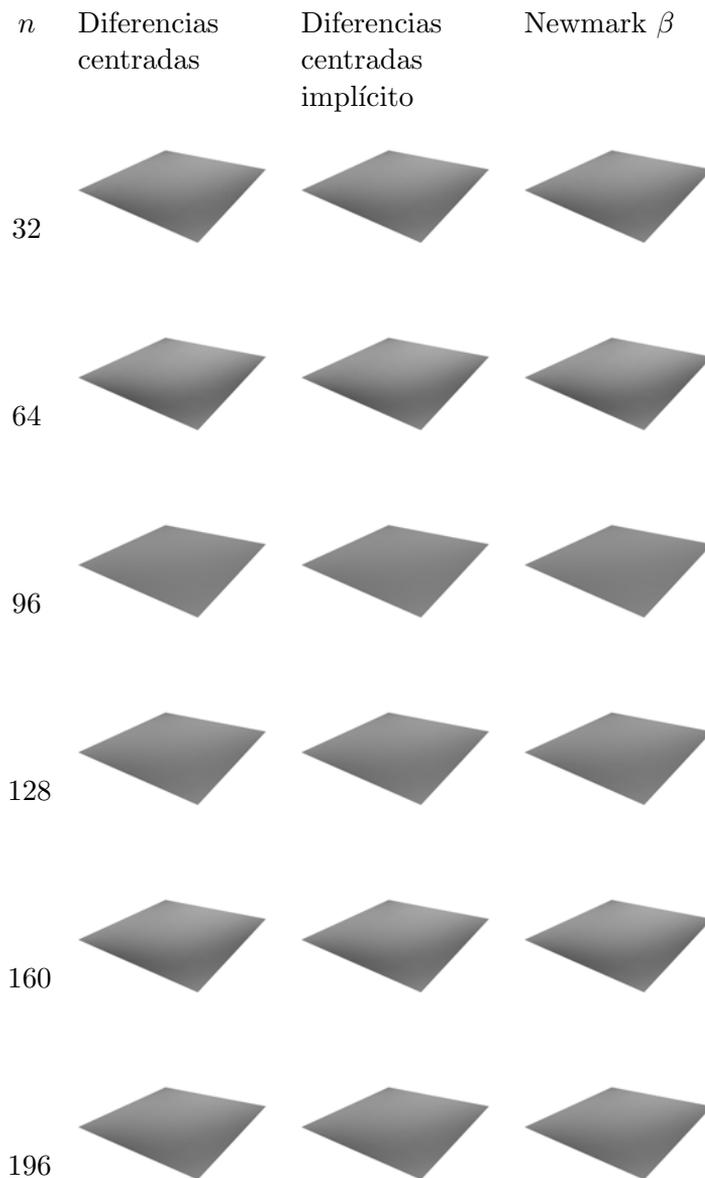


Tabla 5.5: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, resultados para el caso 2). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$.

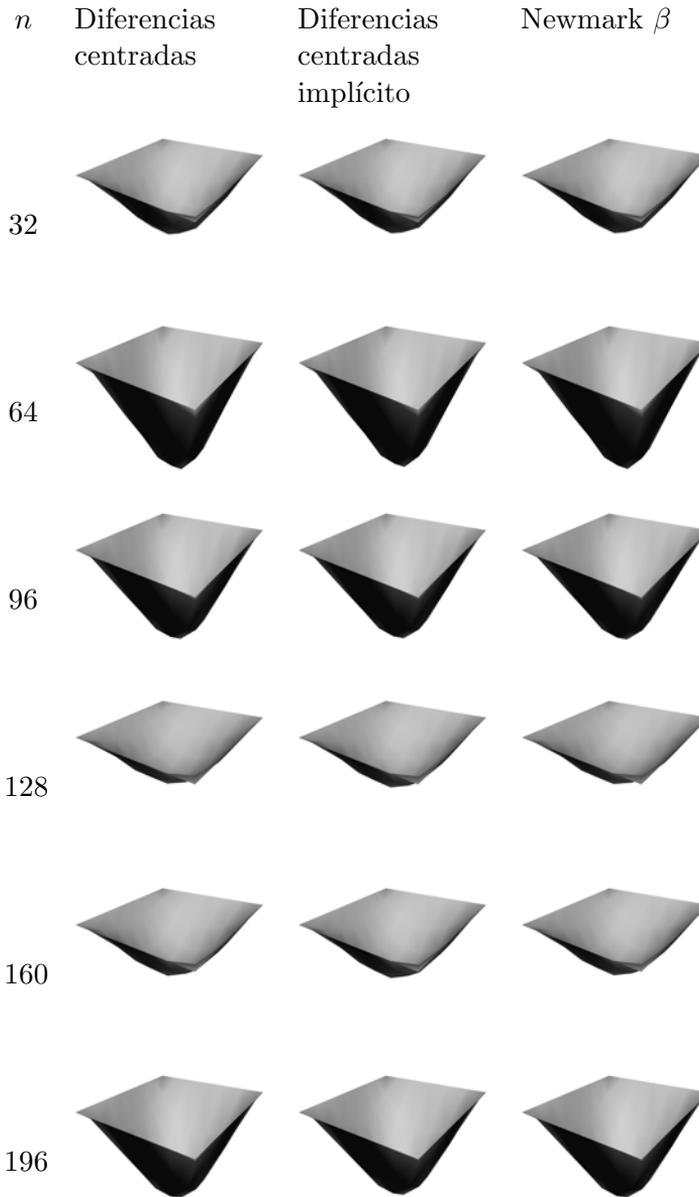


Tabla 5.6: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.01$, resultados para el caso 3). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$.

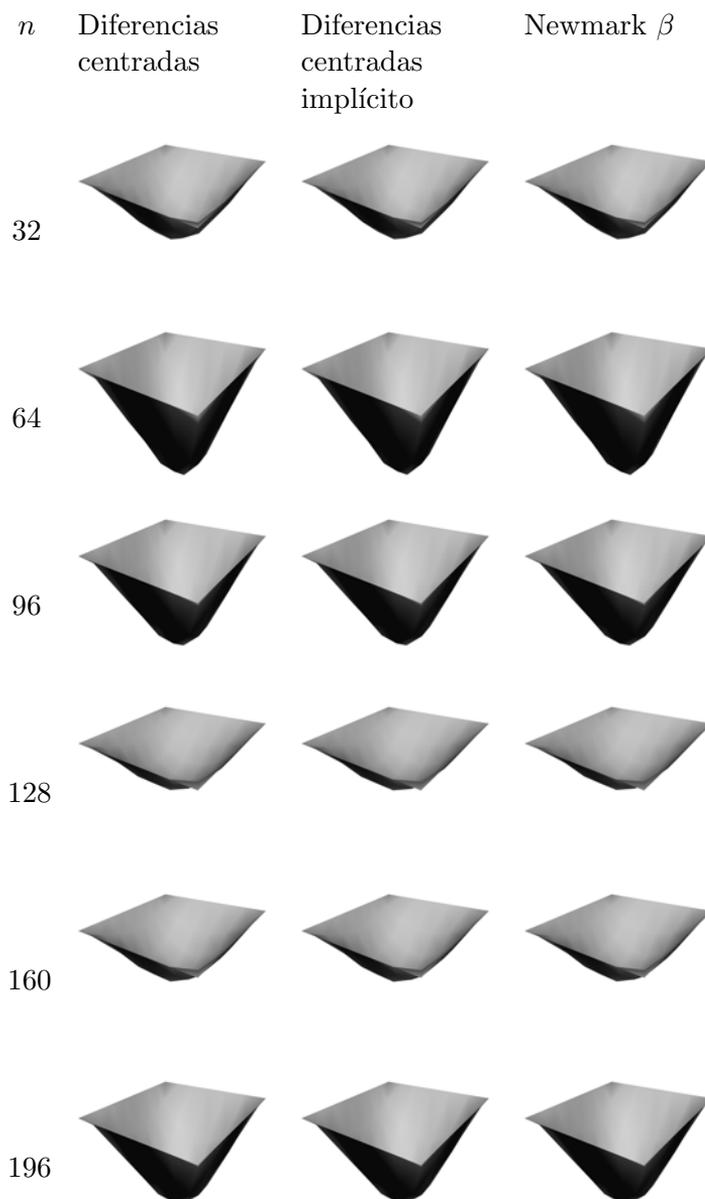


Tabla 5.7: Evolución de las simulaciones para los diferentes modelos con $\Delta t = 0.001$, resultados para el caso 3). En cada fila, para cada modelo, se muestra el resultado de la simulación en el instante $n\Delta t$.

		Puntos de control		
grado	3	4 × 4	6 × 6	8 × 8
	4	6 × 6	8 × 8	10 × 10
	5	6 × 6	8 × 8	10 × 10

Tabla 5.8: En esta tabla se muestran las diferentes combinaciones de grado, puntos de control que se han utilizado para constatar el grado de adecuación del modelo.

El comportamiento desde el punto de vista numérico es más que aceptable, el radio espectral y el número de condición se mantienen constantes a lo largo de todo el proceso de simulación. A modo de ejemplo, en la figura 5.3 se muestra el radio espectral del experimento consistente en deformar una superficie de grado 5 sujeta por los cuatro extremos. La deformación se produce al someter la superficie a un campo gravitatorio perpendicular a la superficie. Los parámetros que determinan el comportamiento elástico son $\alpha_{11} = \alpha_{22} = 1.0$ y $\beta_{ij} = 0.5$. Se han realizado un total de 9 experimentos diferentes, variando el número de puntos de control y el número de puntos de cuadratura de Gauss–Legendre según los valores descritos en la tabla 5.8, para los tres métodos de discretización temporal utilizados las combinaciones de puntos de cuadratura de Gauss–Legendre indicadas anteriormente.

En todos los casos se ha constatado que el radio espectral y el número de condición se ha mantenido inalterado a lo largo de todo el proceso de cálculo. Sin embargo, no todos los experimentos han resultado satisfactorios. El método de diferencias centradas se ha mostrado como poco fiable, el número de condición obtenido en los experimentos con grado 4 y 5 ha sido siempre muy elevado, indicando de esta forma que el sistema obtenido estaba mal condicionado. En cambio tanto el método de diferencias centradas implícito como el de Newmark β han dados unos resultados adecuados.

Aun en los casos peor condicionados, el modelo BSpline4D ha mostrado un comportamiento robusto. Es una muestra más de la adecuación del modelo. A modo de ejemplo, en la figura 5.3 se muestra el número de condición para el caso de la simulación de una superficie de grado 5. Cabe destacar que, como se observa en la figura citada, a mayor número de puntos de control y de puntos de cuadratura de Gauss–Legendre utilizados, mejor condicionado resulta el sistema.

5.4.1. Influencia de los coeficientes en el funcional de energía

Como ya se ha comentado, el funcional de energía utilizado (5.14) busca representar el modelo descrito como *thin plate under tension*, este modelo describe

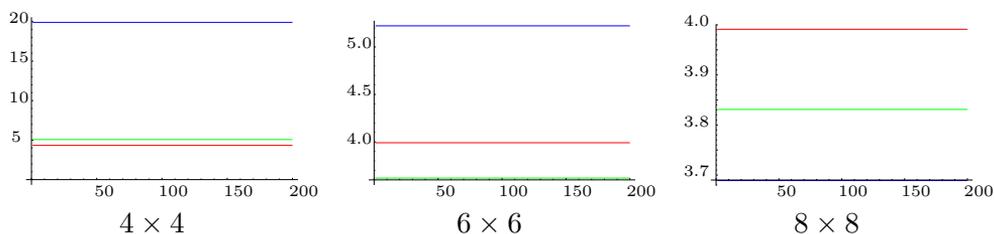


Figura 5.3: El número de condición cambia con cada combinación de grado, número de puntos de control y número de puntos de cuadratura. Sin embargo en todos los casos se mantiene constante a lo largo de toda la simulación. En las imágenes se muestran los resultados para una superficie sujeta por los cuatro extremos, colgando en un campo gravitatorio perpendicular a la superficie, $\alpha_{11} = \alpha_{22} = 1.0$ y $\beta_{ij} = 0.5$. En la imagen aparecen los resultados para la superficie de grado 5 con los diferentes puntos de control descritos en la tabla 5.8, en rojo para 6×6 puntos, en amarillo para 8×8 puntos y en azul para 10×10 puntos. La simulación en los tres casos se basa en el modelo de integración de diferencias centradas implícito. Cada imagen tiene indicada la cantidad de puntos de cuadratura de Gauss–Legendre utilizados. Nótese que el número de condición disminuye cuanto mayor es el número de puntos de cuadratura.

un comportamiento parecido al de una membrana elástica.

El comportamiento de una membrana elástica es tal que ofrece poca resistencia al doblamiento, solamente ofrece una resistencia considerable a los esfuerzos de tracción, es decir de estiramiento.

El funcional descrito contempla dos tipos de parámetros.

- α_{ij} . Determina la tensión de la superficie. A mayor valor, mayor tendencia a minimizar su superficie, ofrece por lo tanto mayor resistencia a la elongación.
- β_{ij} . Determina la resistencia al doblamiento. A mayor valor, mayor rigidez tendrá la superficie.

Los valores de α_{ij} y β_{ij} determinaran el comportamiento elástico de las superficies. Como en el caso ya estudiado en el apartado 3.3.6, tampoco ahora es razonable que estos coeficientes tengan valores negativos. El valor mínimo que ambos parámetros pueden tener es 0.0, en cuyo caso se anularía la aportación correspondiente, y por lo tanto la resistencia a la deformación que pudiera tener la superficie sería también nula.

En el estudio realizado para el modelo basado en Geometría Diferencial se concluye que con el modelo basado en mallados poligonales y con el funcional basado en la Geometría Diferencial no era posible conseguir una superficie que,

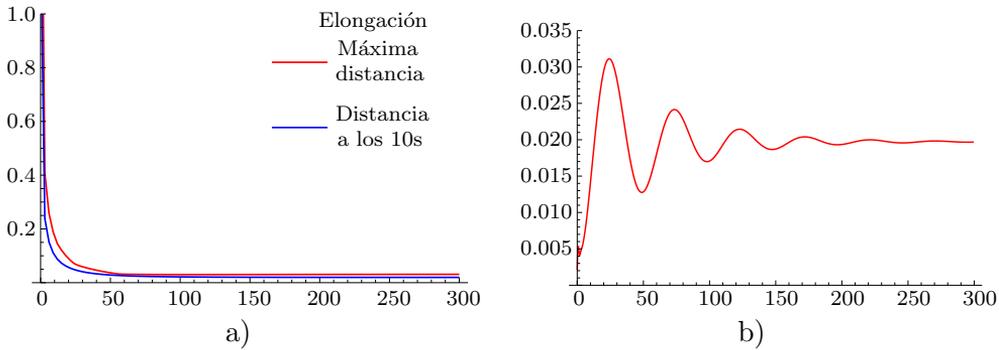


Figura 5.4: En el estudio de la evolución de una lámina sujeta por un lado se puede comprobar que no se consigue una rigidez completa, aunque es posible alcanzar un grado de rigidez notable. El valor de α_{11} y α_{22} se han fijado en 5.0 y el de β_{ij} varía desde 0.0 hasta 300. La superficie tiene grado 3 en cada dimensión y está formada por 4×4 puntos de control. En a) se muestra en color rojo la elongación máxima por cada valor de β_{ij} , en azul se muestra el valor alcanzado tras 10 segundos de simulación. En b) se muestra la variación de la forma de la superficie respecto de la forma ideal inalterada.

colgando por un lado, se mantuviera vertical sin sufrir ninguna elongación. También se ha intentado determinar unos coeficientes η_{ij} y ξ_{ij} que permitan mantener una lámina rígida, sin doblamientos perceptibles. Ambos escenarios han resultado imposibles. Se trata ahora de evaluar el comportamiento del modelo BSpline4D para determinar si es posible conseguir los objetivos descritos y no alcanzados anteriormente. En todos los casos la superficie tratada es de tamaño unitario con masa 1 y coeficiente de amortiguación 1 por unidad de superficie. El grado de la superficie es 3 en cada dimensión y con un total de 4×4 puntos de control.

Estudio de la elongación de una lámina BSpline4D

Se ha procedido a sujetar una lámina por uno de sus lados y se la ha dejado colgando. Fijando el valor de α_{11} y α_{22} a 5.0, valor empíricamente calculado. Variando los valores de β_{ij} desde 0 hasta 300, se ha procedido a medir la deformación longitudinal de la superficie y se ha podido constatar que para $\beta_{ij} = 300$ la elongación de la superficie tras 10 segundos de simulación es de aproximadamente un 1.97%. Para valores de β_{ij} mayores no se ha conseguido una mejora apreciable en el resultado. En la figura 5.4 se muestra la evolución en la elongación de la lámina en función del valor de β_{ij} .

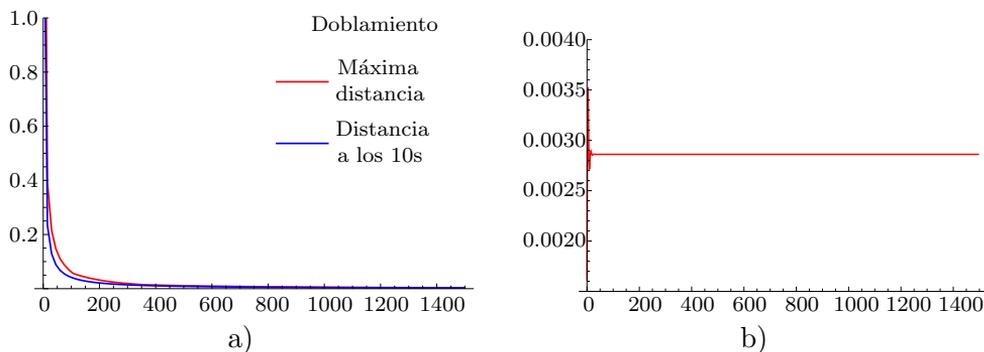


Figura 5.5: El experimento de sujetar una lámina por dos lados adyacentes y determinar la medida en que se mantiene rígida es plenamente satisfactorio. En a) se muestra en color rojo la elongación máxima por cada valor de β_{ij} , en azul se muestra el valor alcanzado tras 10 segundos de simulación. En b) se muestra la variación de la forma de la superficie respecto de la forma ideal inalterada.

Simulación de una lámina BSpline4D rígida

También se ha intentado obtener la simulación del comportamiento de una lámina cuadrada rígida sujeta por dos lados adyacentes. A diferencia del resultado descrito en el apartado 3.3.6, en este caso el comportamiento de la lámina ha sido estable. Ha sido posible simular el comportamiento de una lámina rígida mediante los valores de $\alpha_{11} = \alpha_{22} = 5.0$ y $\beta_{ij} = 1500$. El doblamiento de la lámina ha resultado ser prácticamente imperceptible, aproximadamente un 0.28 % de variación en el extremo libre.

Los resultados obtenidos, como se puede observar en la figura 5.5, permiten establecer que el modelo permite realmente representar superficies con un grado de rigidez muy elevado.

5.5. Comparación con otros modelos de representación

El modelo BSpline4D parece ser una buena forma de representar superficies susceptibles de ser deformadas. Se podría aducir que el resultado no es comparable con el estudio realizado en 3.3 toda vez que el funcional de energía utilizado en los dos casos no es el mismo y por tanto las deformaciones obtenidas no pueden ser numéricamente equiparables.

Como ya se ha comentado, el funcional de energía basado en la Geometría

Diferencial puede presentar problemas numéricos. Por ello se ha descartado el intentar obtener una expresión del funcional (3.17) coherente con las coordenadas generalizadas utilizadas en el modelo BSpline4D. En su lugar, lo que se ha buscado es otro modelo de representación de superficies alternativo sobre el que implementar el funcional de energía *thin plate under tension*, (5.14). El modelo escogido se basa en una representación NURBS de las superficies y se detalla en [TQ94]. A continuación se describe cómo es dicho modelo.

5.5.1. Representación de superficies deformables basada en NURBS

Como ya se ha indicado en (5.1) una superficie NURBS de grados k y l se representa como:

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n \mathbf{P}_{i,j}(t) w_{i,j}(t) N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^m \sum_{j=0}^n w_{i,j} N_{i,k}(u) N_{j,l}(v)} \quad (5.60)$$

En donde $\mathbf{P}_{i,j}(t) \in \mathbb{R}^3$ representa los puntos de control y $w_{i,j}(t) \in \mathbb{R}$ son los pesos asociados a cada uno de estos puntos, en cada instante de tiempo. Combinando ambos valores:

$$\tilde{P}_{i,j}(t) = (P_{i,j}(t) \mid w_{i,j}(t)) = \left(P_{i,j_x}(t), P_{i,j_y}(t), P_{i,j_z}(t), w_{i,j}(t) \right) \in \mathbb{R}^4,$$

se obtendrán valores que pueden servir como coordenadas generalizadas. Así, se puede expresar S en términos de \tilde{P} , expresando de esta forma la dependencia de la superficie NURBS de los puntos de control y los pesos asociados:

$$S(u, v, \tilde{p}) = \mathbf{J}\tilde{\mathbf{p}}, \quad (5.61)$$

$$\dot{S}(u, v, \tilde{p}) = \mathbf{J}\dot{\tilde{\mathbf{p}}}. \quad (5.62)$$

Siendo \mathbf{J} el Jacobiano, cuya forma es:

$$J_{i,j}^p(u, v, \tilde{p}) = \frac{\partial S}{\partial \mathbf{p}_{i,j}} = \frac{w_{i,j} N_{i,k}(u) N_{j,l}(v)}{\sum_{c=0}^m \sum_{d=0}^n w_{c,d} N_{c,k}(u) N_{d,l}(v)}, \quad (5.63)$$

$$J_{i,j}^w(u, v, \tilde{p}) = \frac{\partial S}{\partial w_{i,j}} = \frac{\sum_{c=0}^m \sum_{d=0}^n (p_{i,j} - p_{c,d}) w_{c,d} N_{c,k}(u) N_{d,l}(v) N_{i,k}(u) N_{j,l}(v)}{\left(\sum_{c=0}^m \sum_{d=0}^n w_{c,d} N_{c,k}(u) N_{d,l}(v) \right)^2}. \quad (5.64)$$

Entonces concatenando $J_{i,j}^p$ y $J_{i,j}^w$, se puede plantear

$$\mathbf{J}(u, v, \tilde{p}) = \left[J_{0,0}^p | J_{0,0}^w \cdots J_{m,n}^p | J_{m,n}^w \right]. \quad (5.65)$$

Es decir \mathbf{J} es una matriz de $3 \times 4(m+1)(n+1)$ componentes. Más precisamente, es un *vector* de $N = (m+1)(n+1)$ componentes cada uno de los cuales es una matriz 3×4 :

$$\mathbf{J}_{i,j} = \begin{pmatrix} \left(\frac{\partial s}{\partial P_{i,j}} \right)_x & 0 & 0 & \left(\frac{\partial s}{\partial w_{i,j}} \right)_x \\ 0 & \left(\frac{\partial s}{\partial P_{i,j}} \right)_y & 0 & \left(\frac{\partial s}{\partial w_{i,j}} \right)_y \\ 0 & 0 & \left(\frac{\partial s}{\partial P_{i,j}} \right)_z & \left(\frac{\partial s}{\partial w_{i,j}} \right)_z \end{pmatrix}. \quad (5.66)$$

Con una expresión como la obtenida es perfectamente posible realizar un desarrollo equivalente al realizado para expresar (2.22) en función de las coordenadas generalizadas del modelo BSpline4D. En este caso, las coordenadas generalizadas serán los valores de los puntos de control expresados en la forma \tilde{p} indicada.

Las expresiones obtenidas son las siguientes:

- Primer término: $\frac{d}{dt} \frac{\partial T}{\partial \dot{\tilde{p}}_i}$.

$$\begin{aligned} \frac{dT}{dt} &= \frac{1}{2} \dot{\tilde{p}}^\top M \dot{\tilde{p}}, \\ \mathbf{M} &= \int \int \mu \mathbf{J}^\top \mathbf{J} du dv, \\ \frac{dT}{dt} &= M \ddot{\tilde{p}}. \end{aligned}$$

- Segundo término: $\frac{\partial T}{\partial \tilde{p}_i}$. Se anula.
- Tercer término: $\frac{\partial F}{\partial \tilde{p}_i}$.

$$\begin{aligned} F &= \frac{1}{2} \dot{\tilde{p}}^\top D \dot{\tilde{p}}, \\ D &= \int \int \gamma \mathbf{J}^\top \mathbf{J} du dv, \\ \frac{\partial F}{\partial \tilde{p}_i} &= D \dot{\tilde{p}}. \end{aligned}$$

- Cuarto término: $\frac{\partial U}{\partial \mathbf{p}_i}$. Aplicando el concepto de *thin plate under tension* ya descrito.

$$\begin{aligned}
 U &= \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{K} \dot{\mathbf{p}}, \\
 K &= \int \int (\alpha_{1,1} \mathbf{J}_u^\top \mathbf{J}_u + \alpha_{2,2} \mathbf{J}_v^\top \mathbf{J}_v + \\
 &\quad \beta_{1,1} \mathbf{J}_{uu}^\top \mathbf{J}_{uu} + \beta_{1,2} \mathbf{J}_{uv}^\top \mathbf{J}_{uv} + \beta_{2,2} \mathbf{J}_{vv}^\top \mathbf{J}_{vv}) dudv, \\
 \frac{\partial U}{\partial \mathbf{p}_i} &= \frac{\partial}{\partial \mathbf{p}_i} \left(\frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{K} \dot{\mathbf{p}} \right) = \\
 &\quad \frac{1}{2} \left(0 \cdots \overset{i}{1} \cdots 0 \right)^\top \mathbf{K} \dot{\mathbf{p}} + \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{K} \dot{\mathbf{p}} + \frac{1}{2} \dot{\mathbf{p}}^\top \mathbf{K} \left(0 \cdots \overset{i}{1} \cdots 0 \right).
 \end{aligned}$$

- Término independiente:

$$f_p = \int \int \mathbf{J}^\top f(u, v, t) dudv.$$

Por lo tanto (2.22) se puede expresar como

$$\mathbf{M} \ddot{\mathbf{p}} + \mathbf{D} \dot{\mathbf{p}} + \mathbf{K} \mathbf{p} = f_i + \left[\cdots \frac{1}{2} \dot{\mathbf{p}}^\top \frac{\partial \mathbf{M}}{\partial \mathbf{p}_i} \dot{\mathbf{p}} \cdots \right]^\top - \dot{\mathbf{M}} \dot{\mathbf{p}} - \left[\cdots \frac{1}{2} \dot{\mathbf{p}}^\top \frac{\partial \mathbf{K}}{\partial \mathbf{p}_i} \dot{\mathbf{p}} \cdots \right]^\top \quad (5.67)$$

que también se puede escribir

$$\begin{aligned}
 \mathbf{M} \ddot{\mathbf{p}} + \mathbf{D} \dot{\mathbf{p}} + \mathbf{K} \mathbf{p} &= f_p - \mathbf{I} \dot{\mathbf{p}} \\
 \mathbf{I}(\mathbf{p}) &= \int \int \mu \mathbf{J}^\top \dot{\mathbf{J}} dudv
 \end{aligned} \quad (5.68)$$

siendo f_p un vector de N componentes, cada uno de los cuales es un punto en coordenadas generalizadas. Por su parte, $\mathbf{I}(\mathbf{p})$ es una matriz $N \times N$ cuyos elementos son matrices 4×4 .

Para poder comparar los dos modelos se ha procedido a aplicar la misma deformación a una superficie con idénticas características utilizando los dos modelos como único elemento diferenciador. La simulación realizada es la de una superficie plana de grado 3 con 4×4 puntos de control y 4×4 puntos de cuadratura de Gauss. La superficie ha sido sujeta por las cuatro esquinas y se la ha sometido a un campo gravitatorio perpendicular a la superficie, los coeficientes aplicados son $\alpha_{11} = \alpha_{22} = 1.0$ y $\beta_{ij} = 0.5$, la duración de la simulación es de 2 segundos con $\Delta t = 0.01$. Se ha utilizado el método de integración de diferencias centradas implícito. En la figura 5.6 se muestra el radio espectral y el número de condición calculado para cada iteración. Puede observarse que la estabilidad del modelo basado en BSpline4D es claramente mayor, por lo tanto, aunque visualmente los resultados obtenidos mediante los dos modelos son muy semejantes, véase la figura 5.7, numéricamente el modelo BSpline4D se muestra mucho más sólido.

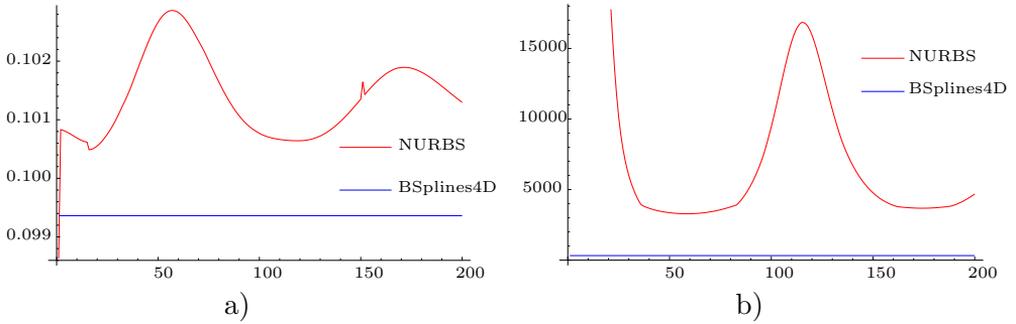


Figura 5.6: Al comparar el modelo de representación basado en BSpline4D con el basado en NURBS se llega a la conclusión que el modelo BSpline4D es mejor desde el punto de vista numérico. Para el experimento descrito en el texto se muestra, en a), el radio espectral de las dos simulaciones y, en b), el correspondiente número de condición.

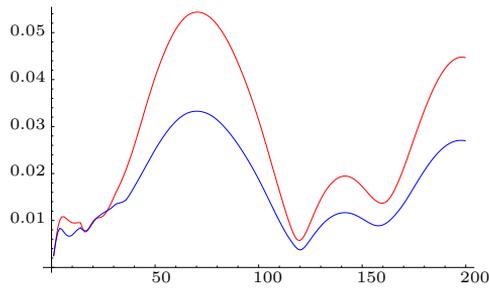


Figura 5.7: Aunque el resultado obtenido con cada modelo no es exactamente el mismo, la diferencia es poca. Para el experimento descrito se ha procedido a determinar para cada iteración la distancia euclídea entre el valor de cada punto de control calculado con los dos modelos, en la gráfica se muestra en azul la media de los valores y en rojo la desviación estándar.

5.6. Estudio del coste computacional del modelo BSpline4D

En el apartado 3.3.7 se ha realizado el estudio del coste computacional, en tiempo, de los diferentes modelos de integración estudiados. Al ser los mismos modelos los que se han aplicado con el modelo BSpline4D, los razonamientos planteados siguen siendo aplicables. De todas formas, con el modelo BSpline4D existen algunas particularidades que se detallarán a continuación.

- Primeramente, el uso como coordenadas generalizadas de los puntos de control. El número de puntos de control requeridos para representar una superficie BSpline4D no tiene por qué ser muy elevado. Por lo tanto el número total de incógnitas será reducido en comparación con la cantidad de incógnitas necesaria para representar una superficie mediante un mallado poligonal que tenga una apariencia semejante.
- El valor de \mathbf{J} no varía si no cambian las propiedades geométricas de la superficie. Por lo tanto su valor, junto con el de todas las derivadas que sean precisas pueden ser calculadas una única vez.
- Para poder determinar los valores de las matrices M , D y K y del vector f que aparecen en (5.30) se utiliza el proceso de cuadratura de Gauss-Legendre. Para poder establecer cada uno de los valores de será preciso realizar los cálculos para cada uno de los puntos de cuadratura que se hayan establecido.
- La propiedad de soporte local de las superficies BSpline4D permite que para cada punto de cuadratura no sea preciso calcular los valores correspondientes para todas las coordenadas generalizadas, sino solamente para un subconjunto de ellas, en función del grado de la superficie.
- El método escogido para el establecimiento de restricciones permite reducir el rango del sistema que deberá resolverse, pero a cambio de tener que realizar más cálculos para obtener el resultado.

En definitiva, el rendimiento computacional, en tiempo, esperado para el modelo de deformaciones basado en BSpline4D es mucho mejor que el rendimiento de un sistema basado en mallados poligonales, como el descrito en la sección 3.3.7. Además, el modelo basado en NURBS introducido en la sección 5.5.1, que podría ser considerado como equivalente al modelo BSpline4D tiene un coste computacional mayor al tener una representación racional de la superficie. Es decir, el modelo BSpline4D también es más eficiente que el modelo basado en NURBS. Por lo tanto, se puede considerar al modelo BSpline4D como una alternativa muy interesante para representar superficies deformables.

5.7. Conclusiones

La ecuación (5.30), que describe el movimiento y deformación de una superficie BSpline4D basada en los puntos de control definidos en \mathbb{R}^4 , ha mostrado un comportamiento numérico muy satisfactorio. Su forma es más sencilla que la correspondiente ecuación, (5.68), obtenida para una representación basada en NURBS. Por añadidura, y como se ha constatado, figura 5.6, también es más estable.

Además, como se deduce de la función de proyección (5.4) las superficies NURBS definidas en \mathbb{R}^3 pueden expresarse siempre como superficies B-Spline definidas en \mathbb{R}^4 . Por lo tanto, toda superficie NURBS deformable puede ser representada mediante BSpline4D.

Finalmente, computacionalmente es un modelo muy eficiente, con lo que su utilización permitirá poder representar modelos más complejos que los que se podrían elaborar basándose en otros medios.

En definitiva, el modelo BSpline4D se presenta como una alternativa viable para la representación de superficies deformables.

Deformaciones simples basadas en restricciones

Una vez desarrollado el modelo de representación basado en B-Splines definidas en \mathbb{R}^4 y comprobada su adecuación para la representación de objetos deformables cabe preguntarse si este modelo de representación resulta adecuado para su utilización con otros modelos de deformación. Este capítulo se dedica a dar respuesta a dicha cuestión.

Posiblemente una de las maneras más sencillas que hay para crear un objeto, infográficamente hablando, sea partir de una forma más o menos simple a la que se van efectuando cambios, deformaciones, hasta conseguir la forma deseada para dicho objeto. La deformación simple de objetos puede ser entendida como una forma computacionalmente poco costosa de deformar objetos de manera que se adapten a un patrón determinado. Los procesos que permiten realizar tal tarea se dividen en dos grupos:

- Métodos iterativos que aplican transformaciones específicas a los objetos a tratar. El proceso de deformación se realiza repetidamente hasta que los objetos tratados alcanzan un grado de estabilidad tal que los cambios que sufren en su forma son tan pequeños que se puede considerar que los objetos ya no cambian. Los métodos de esta categoría suelen asociarse con los modelos basados en las características de los objetos, [TPBF87], [QT95], [QT96], [MMP00]. En definitiva en esta categoría se pueden ubicar los desarrollos planteados en el capítulo 3.
- Métodos independientes de los objetos a tratar. La deformación se aplica a todo el espacio en donde se encuentran los objetos. Al ser todo el espacio el que se deforma, también se deforman los objetos que se encuentran ubicados en este espacio. Dentro de esta categoría se encuentran, entre otros muchos, los métodos propuestos en [Bar84], en donde se aplican transformaciones en

\mathbb{R}^3 tanto afines (escalado, rotación, sesgado) como no afines (retorcimiento, doblamiento, aplastamiento) o como en [SP86] en donde se plantea por primera vez el método conocido como *Free-Form Deformation*. Es en esta categoría para la que se intentará determinar la adecuación del modelo BSpline4D.

Esta segunda categoría tiene como principal ventaja el relativamente bajo coste computacional que requiere su aplicación respecto de otras soluciones. Ahora bien, en general todos los métodos de esta categoría tienen el inconveniente de la dificultad a la hora de manipular los objetos y el espacio circundante para conseguir el resultado deseado, principalmente cuando los objetos a deformar tienen una forma compleja y tienen que ser deformados por partes. Sin embargo existen algunas propuestas que permiten soslayar en parte el problema de la manipulación.

Generalmente la manera en que se pueden realizar dichas deformaciones depende del modelo de representación de los objetos. En la actualidad, la mayor parte de los sistemas se basa en un sistema de mallado; sin embargo, existen alternativas igualmente válidas, como pueden ser las superficies paramétricas.

Los objetos basados en modelos discretos, como son los mallados, pueden deformarse con facilidad seleccionando un vértice del objeto y, desplazándolo a continuación, el objeto se deforma en consecuencia. Esto es así ya que se supone que se mantienen las conexiones con los vértices vecinos, los que definen las mismas caras que el seleccionado. También es posible definir unos pesos asociados a cada vértice y otras muchas variantes que facilitan, en mayor o menor medida, el proceso de deformación. El inconveniente de este sistema es que aunque la modificación de un vértice pueda parecer sencilla, conseguir la deformación de un objeto para conseguir que tenga una cierta forma concreta más o menos compleja puede ser una tarea hartamente tediosa.

También existe la opción consistente en aplicar primitivas simples (rotación, torsión, ...) a objetos aplicando una o más matrices de transformación a sus vértices, [Bar84], para ver un ejemplo véase la figura 6.1.

El resto de este capítulo se dedicará al estudio de la forma en que se puede aplicar el modelo de BSpline4D a las deformaciones libres de forma, en particular para las deformaciones basadas en restricciones. La estructura del capítulo sigue con una primera introducción a las deformaciones de formas libres, para luego pasar al estudio de las deformaciones basadas en restricciones para posteriormente elaborar un modelo de deformaciones basado en restricciones compatible con el modelo de BSpline4D. Finalmente se procederá a extender el modelo para contemplar trayectorias no lineales en las restricciones.

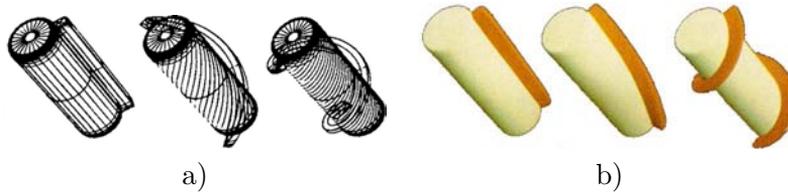


Figura 6.1: Deformación de objetos por aplicación de matrices de deformación a los vértices utilizando primitivas simples. Ejemplos extraídos de [Bar84].

6.1. Deformaciones de formas libres

Las deformaciones de formas libres (*FFD* o *Free-Form Deformations*) permiten modelar objetos de una manera sencilla. Según este modelo, la deformación se basa en la definición de espacios descritos mediante mallados volumétricos basados generalmente en las ecuaciones de Bernstein-Bézier; dentro de estos espacios se sitúan los objetos a deformar. El proceso de deformación se aplica al espacio volumétrico y los objetos que en él se han situado se deforman de manera coherente. Los objetos susceptibles de ser deformados con esta técnica pueden ser representados tanto con series de vértices, es decir con objetos poliédricos o mallados (en definitiva con modelos discretos) como con formas paramétricas e incluso con formas implícitas, aunque con la pérdida de la función matemática que las define. En esencia es posible utilizar gran parte de las representaciones descritas en el capítulo 4.

En general debe tenerse en cuenta que las propiedades o características intrínsecas de los objetos pueden perderse en el proceso de deformación. Un ejemplo claro es que partes que originalmente son interiores pueden convertirse en exteriores, otro ejemplo es el de las autointersecciones, en donde parte del objeto deformado se cruza consigo mismo. Es preciso pues tener presente que la utilización de estos modelos de deformación puede permitir una modelización visualmente aceptable aunque requerirá de un elevado grado de responsabilidad por parte de quien los utilice a la hora de realizar la manipulación para que la deformación sea adecuada.

Los objetos poliédricos, descritos en razón de sus vértices y caras, son los más utilizados con estos modelos debido a que al estar su estructura definida por la relación de los vértices formando aristas, y éstas a su vez formando caras, es fácil mantener la relación de vecindad entre los vértices y por tanto mantener de alguna forma la esencia del objeto pese a su deformación.

El principio de estos modelos es simple, los vértices del objeto a deformar se definen en un sistema de coordenadas que determina el entorno a deformar. Dicho entorno suele representarse como un mallado regular. La modificación de

este sistema de coordenadas, su deformación, provocará que deba recalcularse la posición de los vértices del objeto a tratar y, en consecuencia, éste es a su vez deformado. A partir de los trabajos de Sederberg y Parry [SP86] ha habido una considerable aportación por parte de diversos investigadores en el área ([DHI91], [CF01], [RGLG07]).

En [Coq90] se propone una extensión al modelo FFD (*Extended Free-Form Deformation* o EFFD). Su propuesta consiste en modificar la forma de una superficie a través de su adaptación a un espacio definido por un mallado con formas arbitrarias. El espacio viene definido por un volumen Bézier. El usuario define la forma del mallado que a su vez determina la forma de la deformación. El proceso es equivalente al definido para el modelo FFD a excepción de la forma del mallado original. De forma parecida, en [LW94] el espacio en el que se inscribe el objeto a deformar se describe como un volumen NURBS. En cambio en [CR94] se propone una variante en la que la deformación se plantea en base a una generalización del algoritmo de *de Casteljou*, por la aplicación iterativa de transformaciones afines, en donde el objeto a deformar se adapta a una curva de Bézier que sirve como referencia a la deformación. Desde el punto de vista computacional su planteamiento resulta más eficiente que el modelo original de [SP86].

En [MQ00] se plantea la utilización de un espacio volumétrico basado en el esquema de subdivisión de Catmull-Clark, [CC78], la deformación se obtiene mediante la aplicación recursiva de la subdivisión sobre un entramado de vértices de control especificados por el usuario.

Un enfoque diferente es el que aparece en [Cel90] y [CG91] en donde se propone un modelo basado en elementos finitos. El método de elementos finitos es utilizado para generar primitivas que permitan la construcción de objetos deformables a los que aplicar un nuevo estilo de modelización de formas libres. Las primitivas se deforman autónomamente para minimizar un funcional de energía sujeto a las restricciones geométricas impuestas por el usuario y a las cargas que se definan. Esta aproximación requiere una menor interacción por parte del usuario debido a que el objeto manipulado puede ser parametrizado independientemente de los grados de libertad necesarios para describir su forma.

En algunos casos, [HHK92], el entorno en el que se integra el objeto a deformar queda oculto y se plantea la deformación a partir de la manipulación directa de dicho objeto, de manera que los puntos del espacio a deformar cambian gracias a una aproximación por mínimos cuadrados.

En [RM93] y [RNM95] se utilizan las FFD para aproximar objetos a partir de nubes de puntos obtenidas mediante técnicas de digitalización, como por ejemplo tomografías. Algo parecido se propone en [MMT97] en donde se describe un

modelo de representación de la musculatura, en particular del movimiento de una mano. En [FvdPT97] se plantea un modelo dinámico, es decir, se dota de movimiento a los objetos de manera que la deformación que se les aplique no sea instantánea sino que se produzca dicha deformación durante un periodo de tiempo. El resultado busca dotar a los objetos de un movimiento al estilo de los dibujos animados.

Dentro de lo que se entiende como Free-Form Deformations es en donde se encuadran los modelos simples de deformaciones basadas en restricciones, *Scodef*, [BR94], [RND98].

En este capítulo se realiza un estudio de un modelo simple de deformaciones basado en restricciones geométricas. Dicho modelo se basa en el propuesto en [BR94] extendiendo el mismo para contemplar deformaciones a lo largo de trayectorias no rectilíneas. El estudio puede ser considerado como una extensión del trabajo descrito en [CGMP08].

6.2. *Scodef - Simple Constrained Deformations*

En el trabajo descrito en [BB91], que puede ser considerado como un trabajo seminal, se propone un modelo que permite representar deformaciones basadas en restricciones establecidas por los usuarios. El sistema implementado, denominado DOGME por los autores, ha seguido evolucionando con el tiempo con la contribución de otros trabajos. El planteamiento parte de la consideración de que tanto el espacio original como el deformado son espacios n -dimensionales, \mathbb{R}^n , que en realidad son proyecciones de un espacio \mathbb{R}^m de dimensión superior (es decir $m \geq n$), algunos autores citan al modelo como nD -*deformations*. La deformación se define como la composición de una función $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, que se encarga de transformar los puntos del espacio original en \mathbb{R}^n en puntos del espacio \mathbb{R}^m , con una proyección ($T : \mathbb{R}^m \rightarrow \mathbb{R}^n$, con una matriz de proyección M) del espacio \mathbb{R}^m al espacio deformado \mathbb{R}^n , véase la figura 6.2. Es decir, si n es la dimensión del espacio en donde se encuentra el objeto a deformar S , entonces la deformación de S , $\mathcal{D}(S)$, se caracteriza por la función de deformación $\mathcal{D} : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$\mathcal{D}(S) = S + d(S) = \tilde{S}, \quad (6.1)$$

siendo \tilde{S} el objeto deformado y $d : \mathbb{R}^n \rightarrow \mathbb{R}^n$ la composición de una función $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, que especifica cómo se aplica la deformación a cada uno de los puntos del objeto a deformar, con la transformación lineal $M : \mathbb{R}^m \rightarrow \mathbb{R}^n$, representada por su matriz y calculada de manera que se satisfagan todas las restricciones establecidas, combinándolas de manera adecuada. La matriz de proyección puede

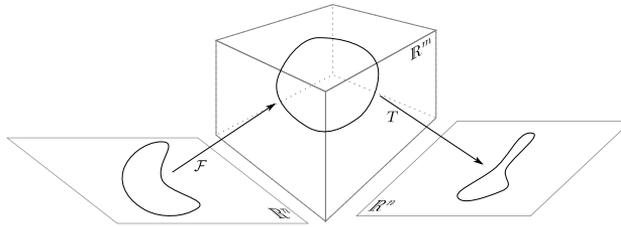


Figura 6.2: Modelo de proyecciones Scodef. Un objeto definido en un espacio \mathbb{R}^n es proyectado a un espacio \mathbb{R}^m de dimensión superior mediante una función \mathcal{F} y luego es proyectado nuevamente a \mathbb{R}^n mediante otra función, T .

calcularse de manera que se cumplan ciertas restricciones, previamente impuestas, en el desplazamiento de algunos puntos respecto del espacio original. Así pues

$$\tilde{S} = S + (M \circ \mathcal{F})(S) = S + M \mathcal{F}(S),$$

y por consiguiente

$$d(S) = M \mathcal{F}(S), \tag{6.2}$$

en donde se representa con M la matriz de la aplicación lineal M .

Con este planteamiento se consigue una deformación con las siguientes características:

- Se deforma todo el espacio en el que se inscribe el objeto.
- La deformación se puede definir interactivamente con facilidad, simplemente desplazando algunos puntos, las restricciones.
- Las restricciones impuestas pueden dar lugar a más de una deformación, para calcular la matriz de proyección M será preciso utilizar algún mecanismo de optimización, como puede ser el de minimizar la distancia a un punto dado o bien minimizar los desplazamientos.
- La forma de la función de deformación, \mathcal{F} , puede permitir deformaciones globales, es decir que afectan a todo el espacio, o bien a deformaciones locales, que restringen su ámbito de aplicación a una parte del espacio y de los objetos o partes de los mismos que en ella se encuentren. Si \mathcal{F} se define como una función de base B-Spline puede conseguirse fácilmente el efecto de deformación local.
- El modelo es aplicable a objetos de cualquier dimensión.

Este planteamiento puede ser considerado como perteneciente a la categoría de FFD, en donde todo el espacio es deformado, pero en este caso solamente se realizan cálculos para los puntos de interés.

Este mismo concepto se desarrolla en [BR94] que puede ser considerado como el primer trabajo sobre Scodef. En él se formaliza el concepto de Scodef tal y como es conocido en la actualidad. Las funciones de deformación se definen como funciones de base B-Spline y se introduce el concepto de radio de influencia de las deformaciones, es decir: las deformaciones se aplican afectando a un punto del objeto y a un entorno del mismo en el espacio en el que está inmerso. Además se plantea la posibilidad de aplicar dos o más restricciones que actúen simultáneamente sobre el objeto. La función de deformación $d : \mathbb{R}^n \rightarrow \mathbb{R}^n$ para cada una de las restricciones se describe como:

$$f_i(Q) = B_i \left(\frac{\|Q - C_i\|}{R_i} \right), \quad (6.3)$$

siendo f_i la función de deformación que se aplica al punto Q del objeto a deformar, R_i es el radio de influencia, C_i es la restricción y B_i es una función de base B-Spline centrada en 0 y tal que $B_i(0) = 1$ y $B_i(1) = 0$. De esta forma los puntos que estén a una distancia de C_i mayor que R_i no quedarán afectados y los puntos que estén más cerca sufrirán una deformación proporcional a su distancia al punto C_i . Surge un problema si dos restricciones aplicadas son tan próximas entre sí que la distancia entre sus puntos de aplicación es menor que el radio de influencia de al menos una de ellas, fenómeno que puede dar lugar a singularidades numéricas que pueden provocar efectos no deseados en la deformación resultante. Para minimizar el problema se plantea la definición de un segundo radio de influencia para poder limitar el efecto negativo del solapamiento de las restricciones.

Además se proponen una serie de extensiones al modelo básico como son:

1. La aplicación de la restricción no se realiza solamente sobre un punto del objeto sino que se aplica a una región del mismo, todos los puntos que estén a una distancia menor que r_i del punto C_i se deforman igual. La forma de la función de deformación cambia:

$$f_i(Q) = \begin{cases} 1, & \|Q - C_i\| \leq r_i, \\ B_i \left(\frac{\|Q - C_i\| - r_i}{R_i} \right), & \|Q - C_i\| > r_i. \end{cases}$$

2. La definición de zonas de influencia con diferentes parámetros para cada dimensión:

$$f_i(Q) = \sqrt{\sum_{j=1}^n \frac{(Q_j - C_{ij})^2}{R_{ij}^2}}.$$

3. Utilizar una parametrización local de las deformaciones, definiéndose un sistema de referencia de coordenadas para cada restricción. De esta forma

el resultado de las deformaciones es mucho más preciso aunque el coste computacional es mucho más elevado.

6.2.1. Extensiones al modelo Scodef

Trabajos posteriores han extendido la funcionalidad original de las deformaciones basadas en restricciones. En [RND98] se describen una serie de extensiones para las Scodef, las e-Scodef. En dicho trabajo se propone un modelo general para la función de deformación, f . Estableciéndose tres restricciones que debe cumplir dicha función:

1. $f_i(C_i) = 1$ para satisfacer la restricción de C_i .
2. $f_i(C_i + R_i) = 0$ según la definición del radio de influencia, que puede ser considerado como un desplazamiento.
3. f debe ser continua para asegurar la continuidad de la deformación.

En el trabajo se define también una extensión del radio de influencia, definiéndose, en lugar de una región circular centrada en el punto de aplicación de la restricción, lo que se denomina un contorno de influencia que virtualmente puede ser de cualquier forma. En particular, en el trabajo se propone la utilización de supercuádras y también de formas estrelladas¹. Este planteamiento permite que si una restricción C_i engloba en su contorno de influencia el punto de aplicación de otra de las restricciones, C_j , es decir si ambas restricciones son no disjuntas, entonces se puedan recortar los contornos de influencia mediante un hiperplano ortogonal a $\overline{C_i C_j}$ de manera que se mantenga la convexidad de los contornos y a la vez se elimine el problema del solapamiento de las restricciones.

Hasta el trabajo de [RNJ99] no se plantea la posibilidad de que la deformación aplicada por la restricción siga una trayectoria. En este trabajo se propone modificar el modelo de *e-Scodef* para que también se pueda definir una trayectoria que deberán seguir las partes del objeto sujetas a deformación. El nuevo modelo de desplazamiento se formula de la siguiente forma:

$$d(U) = Mg(U) \quad \forall U \in \mathbb{R}^n, \quad (6.4)$$

siendo $g(U)$ una nueva función de deformación que combina el efecto de una función de desplazamiento, T , con la función de deformación (6.3):

$$g(U) = T(f(U)) \quad \forall U \in \mathbb{R}^n.$$

¹Un objeto n -dimensional A tiene una forma estrellada respecto a un punto H si para cualquier punto M de A el segmento MH pertenece a A .

T representa la trayectoria que deberá seguir el punto asociado a la restricción. En el trabajo se propone que T sea una curva de Bézier, de manera que el primer punto de control se asocia con el punto C_i del objeto y el último punto con el lugar en donde debe ir a parar dicho punto como resultado de la deformación; los puntos intermedios permiten describir la trayectoria y por tanto la forma de la deformación. Este planteamiento permite combinar muy fácilmente T y f .

Según como se defina la trayectoria, los objetos deformados con este planteamiento pueden resultar aplastados. En [RNJ00] se propone una solución que palía en parte el problema mediante el desplazamiento del lugar al que va a parar cada punto según el vector normal al punto correspondiente de la curva de la trayectoria. Además se contempla la posibilidad de la aplicación de múltiples restricciones mediante la reescritura de (6.4), aplicando a cada punto del objeto el resultado de la influencia de todas las restricciones definidas. Así, si hay r restricciones, el desplazamiento del punto Q será:

$$d(Q) = \sum_{i=1}^r T_i(M_i f_i(Q)).$$

En [BG03] se refina el mismo planteamiento de deformaciones siguiendo trayectorias curvas ampliando el concepto para definir deformaciones *curva-curva*, en donde, en lugar de definir un punto al que se aplica una restricción, se define una curva sobre el objeto a deformar que determina el conjunto de puntos a los que se aplica la restricción. Dicha restricción puede seguir una trayectoria curva. También se describe la forma de definir volúmenes de influencia, que se corresponden con el interior de cualquier poliedro. El modelo presentado, una extensión de DOGME, ha sido diseñado e implementado para su uso en entornos virtuales, [BG04].

Otras extensiones del modelo se han centrado en la definición de diferentes métodos para representar el área de influencia, [JLP00], o sobre diferentes modelos de representación de las superficies, por ejemplo en [LRN06] se aplican Scodef sobre superficies de subdivisión. Otro planteamiento muy interesante es el propuesto en [LG05] y en [LGRG07] en donde se aplican Scodef sobre superficies NURBS estableciendo el desplazamiento, $m(i, j)$ de cada punto de control de la superficie, $P(i, j)$ según la expresión:

$$m(i, j) = \frac{f(i, j)}{\sum_{k=0}^n \sum_{l=0}^m R_{k,l}(u, v) f(k, l)} \mathbf{e}, \quad (6.5)$$

siendo $\mathbf{e} = M\hat{M}$ el vector que une el punto M , el lugar original de la restricción, con el punto \hat{M} , el destino de dicho punto. La función f representa el medio para

determinar qué puntos de control quedan afectados por la restricción, $R_{k,l}(u, v)$ se define como:

$$R_{k,l}(u, v) = \frac{N_{k,p}(u)N_{l,q}(v)w_{k,l}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)w_{i,j}},$$

siendo N las funciones de base B-Spline de la superficie NURBS y $w_{i,j}$ el peso asociado al punto de control $P_{i,j}$. Para el caso de deformaciones compatibles con el ideal Scodef, $f(k, l) = R_{k,l}(u, v)$.

En el mismo trabajo se propone también la deformación de superficies NURBS siguiendo una trayectoria realizando el proceso de deformación a lo largo de una serie de pasos y en cada uno de ellos calculando una restricción puntual adecuada para realizar una deformación utilizando (6.5). La trayectoria viene definida por una curva NURBS, $C(t)$, $t \in [0, 1]$ de manera que el punto M , lugar en donde se fija la restricción marca el lugar del primer punto de la curva (y por lo tanto del primer punto de control) y el lugar a donde debe ir a para dicho punto corresponde con el último punto la curva (y por lo tanto del último punto de control). La curva es recorrida iterativamente en una serie, N , de pasos discretos (los autores proponen como valor de N diez veces el producto del grado de la curva por el número de puntos de control de la misma), a cada paso se define una restricción temporal para hacer que los puntos de control cambien para ajustar la superficie. Surge el problema de la cantidad de puntos de control necesarios para poder ajustar de una forma adecuada la deformación de la superficie a la curva de trayectoria. Además, la superficie resultante deja de ser una superficie NURBS.

6.3. *N-Scodef*

A continuación se detalla un método generalizado de deformaciones basado en Scodef que permite su aplicación a superficies BSpline4D, permitiendo que éstas mantengan sus propiedades. El método propuesto permite la utilización de un elevado número de funciones de deformación que podrán ser aplicadas al conjunto de restricciones impuestas por el usuario con una gran cantidad de diferentes zonas de influencia. Además se realizará el estudio de un conjunto de normas y distancias para determinar su adecuación a la hora de calcular la deformación y las zonas de influencia de las restricciones, de manera que se pueda conseguir representar el mayor número posible de formas de aplicar las deformaciones.

6.3.1. *N-Scodef* una extensión de *Scodef*

Partiendo de la notación expresada originalmente en [BB91] y después utilizada en [BR94] a la hora de definir las deformaciones simples basadas en restricciones (*Scodef*) la deformación de un objeto se define mediante la manipulación de las denominadas restricciones, puntos seleccionados por el usuario que deben tener un determinado comportamiento durante la deformación del objeto a tratar.

A partir de la función de deformación escogida y de las restricciones definidas por el usuario se calcula la matriz de proyección, M , con la que se podrá asegurar que las restricciones impuestas se cumplan.

La zona de influencia de una restricción se define como el conjunto de puntos del modelo de referencia, el objeto original, cuya distancia a la restricción es menor que un cierto valor, el denominado radio de la zona de influencia. Evidentemente, la forma de la zona de influencia cambiará dependiendo de la forma en que se calcule dicha distancia y de la manera en que se determine el radio de influencia.

El proceso de deformación se descompone en tres fases:

1. Selección por parte del usuario de los desplazamientos, D_i , del conjunto de restricciones, C_i , así como la definición de la zona de influencia de cada restricción. Elección del radio de influencia, R_i , y de la formulación de \mathcal{F} . De esta forma, dado cualquier punto $Q \in \mathbb{R}^n$ se puede determinar su desplazamiento respecto de las restricciones definidas mediante:

$$d(Q) = \sum_{i=1}^r M_i f_i(Q), \quad (6.6)$$

en principio $f_i(Q)$ puede ser tal y como se define en (6.3) o bien cualquiera de sus extensiones. La ecuación (6.6) puede ser reescrita como $d(Q) = M\mathcal{F}(Q)$ siendo M una matriz de n filas y r columnas, una para cada una de las restricciones, $\mathcal{F}(Q)$ es un vector de dimensión r formado por las funciones f_i para cada una de las restricciones.

2. Cálculo de la matriz de proyección, M , de manera que se satisfagan las restricciones impuestas por el usuario. Es decir, se debe resolver el sistema de ecuaciones lineal; $d(C_i) = M\mathcal{F}(C_i)$, $1 \leq i \leq r$, siendo r el número de restricciones y $\mathcal{F} = (f_1, f_2, \dots, f_r)$. La influencia que una restricción pueda tener sobre otra dependerá principalmente de la forma de la región de influencia y del radio de influencia. Si existe un solapamiento en las regiones de influencia puede darse el caso que la deformación no se desarrolle como es de esperar, entonces deberá establecerse un mecanismo que permita paliar el problema, este aspecto se estudia con mayor detalle en el apartado 6.3.2.

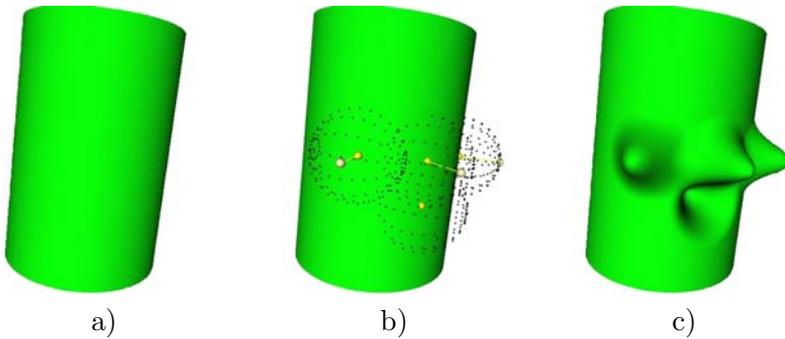


Figura 6.3: Aplicación de una deformación simple basada en restricciones. a) El objeto a deformar: un cilindro. b) Las restricciones mostradas como un punto sobre la superficie del cilindro y el lugar al que debe ir a parar el punto, también se muestra la zona de influencia como una esfera de puntos. c) El cilindro una vez aplicada la deformación.

3. Con la matriz de proyección M ya calculada se puede proceder a deformar el objeto a manipular aplicando las ecuaciones (6.1) y (6.2).

En la figura 6.3 se muestra un ejemplo en el que se aplican varias deformaciones sobre un cilindro.

En el trabajo presentado en [BR94] las restricciones corresponden a puntos definidos sobre la superficie del objeto a deformar o cercanos a ésta. Para cada uno de estos puntos se define un desplazamiento, de manera que se indica cual será la posición que dicho punto ocupará una vez que la deformación del objeto se haya realizado. El desplazamiento se expresa como un vector lineal, pero también podría ser especificado mediante una curva ([RNJ99], [RNJ00], [BG03], [LGRG07]).

El objetivo es deformar un objeto (en principio una curva o superficie) BSpline4D, S , de una forma interactiva y siguiendo el esquema descrito previamente. El usuario tiene que definir los puntos sobre el objeto sujetos a restricciones y sus desplazamientos, que pasarán a formar parte de la función de deformación que se aplicará sobre S .

La deformación se expresa, de acuerdo con (6.2) de la siguiente forma: sea $d : \mathbb{R}^n \rightarrow \mathbb{R}^n$ (usualmente $n = 2, 3$ o 4) la función de deformación que representa el desplazamiento de los puntos de una superficie BSpline4D, $d(S(u, v))$ siendo $(u, v) \in [0, 1]^2$. La función es de la forma:

$$\forall (u, v) \in [0, 1] \times [0, 1] \quad d(S(u, v)) = \sum_{i=1}^r M_i f_i(S(u, v)),$$

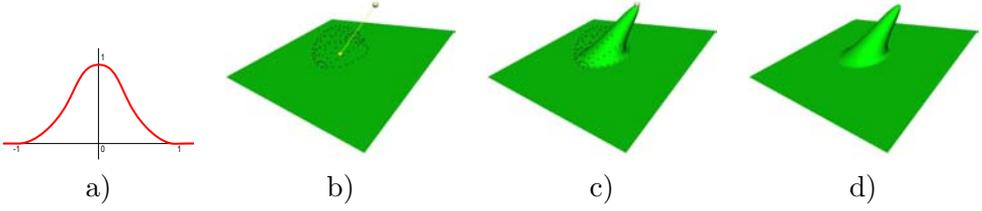


Figura 6.4: Elementos de una deformación basada en restricciones y su acción sobre una superficie NURBS. a) La función de deformación, b) Restricción seleccionada (punto amarillo sobre la superficie) y desplazamiento de dicha restricción, la esfera de puntos muestra la zona de influencia, c) muestra la superficie deformada con la restricción superpuesta, y d) muestra la deformación tal cual resulta.

siendo

- r el número de restricciones impuestas por el usuario.
- M_i la i -ésima columna de la matriz M .
- $\tilde{f}_i : \mathbb{R} \rightarrow \mathbb{R}$ una función escalar monótonamente decreciente definida de manera que $\tilde{f}_i(0) = 1$ y $\tilde{f}_i(x) = 0 \forall x : |x| \geq 1$. Además está asociada a la restricción C_i . Dicha función determina la influencia de la restricción sobre cada punto de la superficie $S(u, v)$. De esta forma la función $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ dependiente de las restricciones C_i y de los radios de influencia R_i de cada una de ellas, se define como:

$$f_i(S(u, v)) = \tilde{f}_i \left(\frac{\|S(u, v) - C_i\|}{R_i} \right). \quad (6.7)$$

De manera que para cada punto $Q = S(u, v)$ se cumple

$$f_i(Q) = \begin{cases} 1, & \text{si } \|Q - C_i\| = 0, \\ 0, & \text{si } \|Q - C_i\| \geq R_i, \\ 0 \leq f_i(Q) \leq 1, & \text{en otro caso.} \end{cases} \quad (6.8)$$

De cara a asegurar la continuidad de la superficie deformada, \tilde{f}_i debe a su vez ser continua. En este caso la función $\mathcal{F} = (f_1, f_2, \dots, f_r)$ es una función vectorial cuyas componentes son funciones continuas.

En la figura 6.4 se muestran gráficamente los componentes descritos. Tanto la zona de influencia como la función de deformación son calculadas utilizando la distancia euclídea. Si un punto de la superficie se encuentra dentro de la zona de influencia de la restricción entonces dicho punto es deformado con lo que su ubicación en el espacio cambia. La deformación se aplica exclusivamente sobre los puntos que se encuentran originariamente en la zona de influencia de una

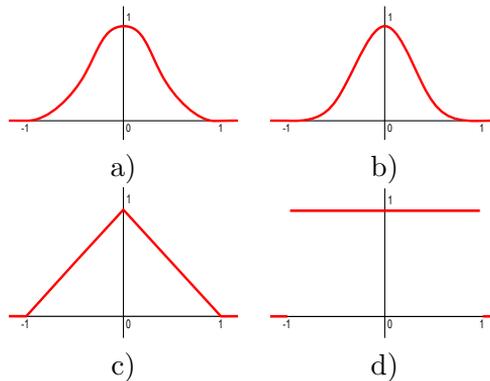


Figura 6.5: Algunas funciones de deformación aplicables a las deformaciones basadas en restricciones. Se definen a partir de una función de base B-Spline de grado 3, 2, 1 y 0 respectivamente.

restricción quedando el resto inalterados. La deformación de los puntos sigue la forma de la función de deformación escogida, tal y como se puede ver en la figura 6.4.

Es posible definir una gran variedad de funciones de deformación que satisfagan las condiciones descritas. Véase la figura 6.5 en donde se muestran algunas funciones de deformación basadas en funciones de base B-Spline, en particular la figura 6.5.a se corresponde con la función de deformación introducida en [BR94] y denominada *radial deformation* ya que permite una propagación isotrópica de la deformación. Obviamente la forma de cada función de deformación, \tilde{f}_i , influye directamente en la forma en que se deforma el objeto, tal y como se puede observar en la figura 6.6. De cara a ofrecer una mejor funcionalidad, un usuario podría escoger la función deseada dentro de un conjunto de funciones ya definidas.

Sin embargo no es solamente la forma de las funciones, \tilde{f}_i , lo que determina la forma en que se deformará el objeto sino que, teniendo en cuenta la ecuación (6.7), la norma escogida y el radio de influencia de la región son también aspectos a tener en cuenta. El usuario debe poder cambiar ambos de manera interactiva ya que de esta forma se consigue aumentar considerablemente el rango de posibles deformaciones aplicables.

El concepto de radio de influencia, tal y como es introducido en [BR94], es de gran ayuda en el proceso de deformación ya que controla la localidad de las deformaciones, limitando la influencia espacial de la restricción [RNJ00]. El efecto del radio de influencia puede observarse en la figura 6.7 en donde aparece una superficie BSpline4D a la que se aplica una deformación utilizando una región de influencia esférica con diferentes radios y utilizando una distancia euclídea.

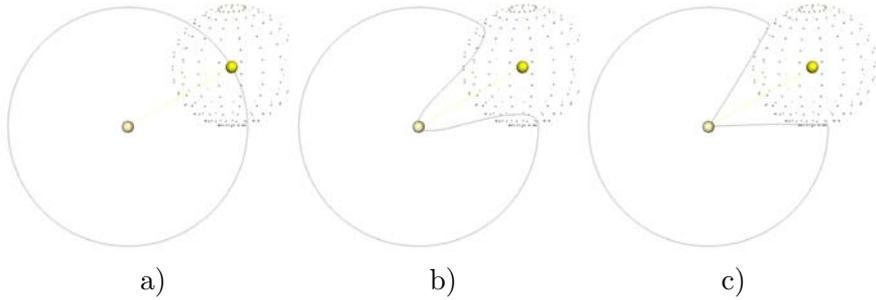


Figura 6.6: La forma de la función de deformación afecta a la manera en que la deformación se produce. En a) se puede apreciar el mismo objeto a deformar, un círculo, en donde se ha seleccionado un punto y con la misma zona de influencia (esfera de puntos alrededor). El resultado difiere: en b) la función de deformación es la que aparece en la figura 6.5.a y en c) la función es la que aparece en la figura 6.5.c.

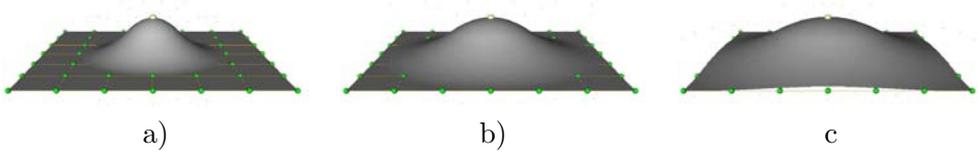


Figura 6.7: El efecto del radio de influencia: una superficie NURBS es deformada en el punto central con un radio de influencia esférico, en a) el radio es menor y en c) mayor.

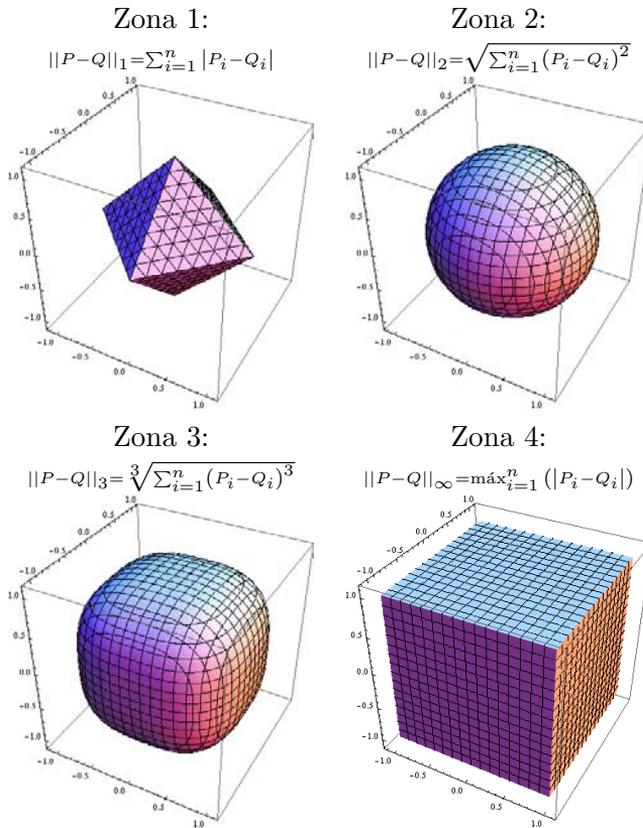


Figura 6.8: Diferentes formas para determinar las zonas de influencia y sus respectivas distancias. Estas formas son las que han sido utilizadas en las figuras 6.11 y 6.12.

Es posible utilizar zonas de influencia con formas no esféricas, y también es posible utilizar distancias no euclidianas [RND98, RNJ99, RNJ00, JLP00, LGRG07, CGMP08]. En la figura 6.8 se muestran diferentes formas posibles para determinar las zonas de influencia así como las correspondientes distancias que pueden ser aplicadas para calcular (6.7). Es posible utilizar envolventes definidas por supercuádricas en la misma forma en que se detalla en [BCA98] de forma que se puedan definir fácilmente diferentes envolventes. El usuario puede modificar la forma de la superficie simplemente cambiando unos pocos parámetros. De esta forma pueden conseguirse con suma facilidad restricciones con zonas de influencia anisotrópicas aplicables sobre los objetos a deformar.

6.3.2. Interacción entre diferentes restricciones

Las deformaciones descritas según el modelo detallado en el apartado anterior son continuas ya que se definen mediante un aplicación $\mathbb{R}^n \rightarrow \mathbb{R}^n$ definida como una combinación lineal de funciones continuas, definidas por la ecuación (6.7). De hecho el grado de suavidad de la superficie deformada depende del grado de suavidad de las funciones definidas en (6.7). Así, para un punto Q de la superficie a deformar, la función dada en (6.7) puede expresarse, por ejemplo, como:

$$f_i(Q) = B_i \left(\frac{\|Q - C_i\|}{R_i} \right), \quad (6.9)$$

siendo Q el punto considerado, C_i una restricción, R_i el radio de influencia de la deformación y B_i una función de base B-Spline centrada en 0 y que es igual a 0 en -1 y en 1. Es decir, f_i es una función de distancia que vale 1 si $Q = C_i$ y que vale 0 si $\|Q - C_i\| \geq R_i$, de acuerdo con la ecuación (6.8).

Cabe recordar que las restricciones imponen el lugar en donde deben ir a parar los puntos, C_i , sobre las que son aplicados los vectores de desplazamiento D_i . Para ello deberá determinarse cual es la forma de la matriz M . Para satisfacer la restricción C_i , M debe ser tal que:

$$D_i = d(C_i) = M \mathcal{F}(C_i) \quad \forall i \in 1..r,$$

y así el desplazamiento de C_i se corresponderá con la restricción definida. Si n es la dimensión del espacio considerado y r el número de restricciones definidas, entonces M es una matriz $n \times r$ y $\mathcal{F}(C_i)$ es un vector columna de dimensión r ; y por lo tanto D_i es un vector columna de dimensión n , siendo cada componente el valor del desplazamiento en la correspondiente coordenada del espacio.

Sea D_{ij} la j -ésima coordenada de D_i y sea M_j la j -ésima fila de M , entonces de la expresión anterior se obtiene que

$$D_{ij} = M_j \mathcal{F}(C_i) = \mathcal{F}^\top(C_i) \cdot M_j^\top \quad \forall i \in 1, \dots, r,$$

y esto para cada una de las restricciones. Combinando todas las restricciones en una ecuación; se obtiene para la j -ésima coordenada que

$$D_j(C) = \begin{pmatrix} D_{1j} \\ \vdots \\ D_{rj} \end{pmatrix} = \begin{pmatrix} \mathcal{F}^\top(C_1) \\ \vdots \\ \mathcal{F}^\top(C_r) \end{pmatrix} M_j^\top = X M_j^\top, \quad (6.10)$$

donde X es una matriz $r \times r$ obtenida a partir del producto de los valores de $\mathcal{F} = (f_1, f_2, \dots, f_r)$ por las restricciones. Como puede observarse, el valor de X

no depende de j en absoluto; lo que permite definir un sistema de ecuaciones lineales para cada coordenada, siendo la solución la correspondiente fila de M . Por lo tanto, la solución para la coordenada j -ésima será la fila M_j .

En caso de que se decida utilizar más de una restricción, dependiendo del radio de influencia de cada una de ellas pueden darse dos casos, el primero es que las restricciones no interfieran entre sí:

$$\forall i, j \in 1, \dots, r, i \neq j \quad f_j(C_i) = 0,$$

con lo que $X = I$ y por lo tanto

$$d(Q) = \sum_{i=1}^r D_i f_i(Q),$$

es decir, que el desplazamiento de un punto Q se corresponde con la media ponderada de los desplazamientos de las restricciones. El peso o influencia de cada restricción será inversamente proporcional a su distancia respecto a Q .

El segundo caso es el que contempla que las restricciones interfieran las unas con las otras:

$$\exists i, j \in 1, \dots, r, i \neq j \mid f_j(C_i) \neq 0$$

lo que en definitiva significa que las restricciones se aplican a puntos afectados por la zona de influencia de otras restricciones:

$$d(Q) = \sum_{i=1}^r M_i f_i(Q).$$

Mientras que el primer caso no representa ningún problema añadido, no sucede lo mismo con el segundo. En general, el desplazamiento de un conjunto de r restricciones no disjuntas puede expresarse según la siguiente expresión:

$$(d(C_1) \dots d(C_r)) = M \begin{pmatrix} f_1(C_1) & \dots & f_1(C_r) \\ \vdots & \ddots & \vdots \\ f_r(C_1) & \dots & f_r(C_r) \end{pmatrix}, \quad (6.11)$$

que representa un sistema de ecuaciones siendo $M = (M_1, M_2, \dots, M_r)$ las incógnitas. Suponiendo el caso de una Scodef con solamente dos restricciones que tengan el mismo radio de influencia, que éste sea mayor que la distancia que separa ambas restricciones y que por lo demás sean idénticas; entonces se puede expresar:

$$f_2(C_1) = f_1(C_2) = a$$

y por lo tanto la ecuación (6.11) se reduce a:

$$(d(C_1) d(C_2)) = M \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix} = (M_1 M_2) \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix},$$

de manera que para conocer el valor de $M = (M_1 M_2)$ será preciso resolver el sistema:

$$(d(C_1) d(C_2)) \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}^{-1} = (M_1 M_2) \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix} \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}^{-1} = (M_1 M_2) I = (M_1 M_2)$$

siendo

$$\begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{1-a^2} & \frac{-a}{1-a^2} \\ \frac{-a}{1-a^2} & \frac{1}{1-a^2} \end{pmatrix},$$

entonces el valor de $M = (M_1 M_2)$ es:

$$M_1 = \frac{1}{1-a^2} d(C_1) - \frac{a}{1-a^2} d(C_2),$$

$$M_2 = \frac{-a}{1-a^2} d(C_1) - \frac{1}{1-a^2} d(C_2),$$

suponiendo que $d(C_1) = -d(C_2)$ entonces

$$M_1 = -M_2 = \frac{1}{1-a} d(C_1)$$

y por lo tanto

$$d(Q) = \frac{d(C_1)}{1-a} (f_1(Q) - f_2(Q))$$

De este desarrollo se deduce que:

- Si la distancia entre las restricciones tiende a 0, entonces a tiende a 1, los valores de M pueden crecer enormemente.
- Si $d(C_1) \neq d(C_2)$ y las restricciones coinciden ($a = 1$) entonces no existe ninguna M que pueda satisfacer las restricciones, ya que lo que representa tal situación es que un mismo punto acaba en dos lugares diferentes una vez aplicada la deformación, figura 6.9.f.
- Como la magnitud de $d(C_1)$ es independiente de $f_1(Q) - f_2(Q)$, un punto Q próximo a las restricciones, puede resultar desplazado muy lejos. Lo que visualmente puede apreciarse como una *rotura* del espacio, esta situación puede producirse cuando la distancia entre dos restricciones es mucho menor que la magnitud de los radios de influencia de las mismas. Véase la figura 6.9.e.

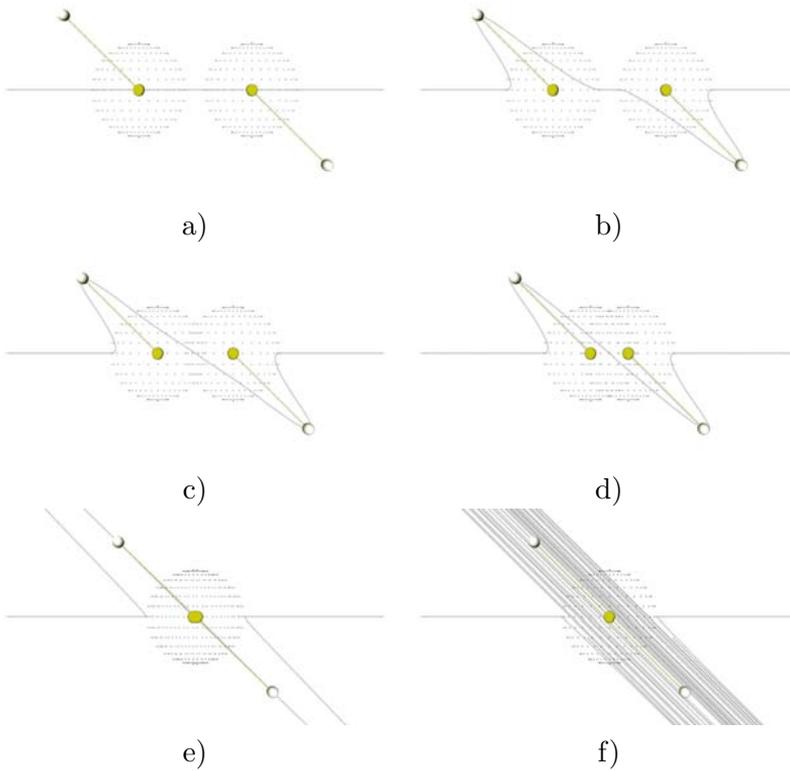


Figura 6.9: Comportamiento de un Scodef con dos restricciones. a) las restricciones, disjuntas, antes de aplicar la deformación y después, b). En c) y d) las restricciones están cada vez más cerca la una de la otra. En e) las dos restricciones se aplican a puntos muy próximos entre sí. Finalmente en f) las dos restricciones se aplican al mismo punto, no existe ninguna M que pueda satisfacer las restricciones.

El sistema planteado en (6.10) tiene una solución única, si no fuera así el usuario tendría la posibilidad de escoger de entre todas las posibilidades la que se ajustara más a sus intereses. Por ello, en [BR94] se propone un duplicado de las restricciones, con lo que se incrementan los grados de libertad de (6.10). El resultado de ello es que para cada restricción se dispone de dos radios de influencia que el usuario puede manipular de forma independiente y así conseguir ajustar la deformación en las zonas donde puedan surgir solapamientos. Para cada restricción el *radio mayor* determina el radio de influencia de la restricción y el *radio menor* determina hasta qué punto la restricción influye sobre los puntos. El inconveniente de esta solución es que la matriz X se convierte en singular y para la solución del sistema de ecuaciones se hace preciso utilizar el cálculo de su pseudoinversa.

En cambio, en [RND98] el problema se soluciona definiendo unas nuevas funciones de deformación, parametrizando la zona de influencia de las restricciones y añadiendo una regla de no influencia de las restricciones unas respecto de las otras. Cada zona de influencia se delimita mediante una *envolvente de influencia*.

Para evitar las singularidades se separan las restricciones no disjuntas estableciendo que las *envolventes de influencia* puedan ser cortadas por un hiperplano, manteniéndose así la convexidad. Para ello se realiza una comprobación para cada restricción para determinar si existe otra que esté afectando a su punto de aplicación, es decir, que la intersección de las envolventes de influencia es no nula, y en el caso de que así sea la envolvente es cortada por un hiperplano ortogonal al segmento que une las dos restricciones afectadas, de manera que cada punto quede afectado solamente por una única restricción, véase la figura 6.10 para ver un ejemplo en \mathbb{R}^2 . El método es el siguiente:

Para cada i, j , $i \neq j$, sean C_i y C_j dos restricciones, y sean $E(C_i)$ y $E(C_j)$ las correspondientes envolventes. Si $C_i \in E(C_j)$:

- a) Se determina el hiperplano P que pasa por C_i y es ortogonal a la recta $C_i + \lambda C_i C_j$.
- b) Si $p \in E(C_i) \cap E(C_j)$ y además p se encuentra en el semiespacio que no contiene a C_j entonces p queda en la zona de influencia de C_i .
- c) Si $p \in E(C_i) \cap E(C_j)$ y además p se encuentra en el semiespacio que contiene a C_j entonces p queda en la zona de influencia de C_j .

Con este método se consigue que la restricción C_j no afecte a C_i y, en consecuencia, se puede asegurar que todas las restricciones serán disjuntas. La ventaja de este planteamiento es que no requiere el cálculo de pseudoinversas y por lo tanto cabe esperar que sea más rápido.

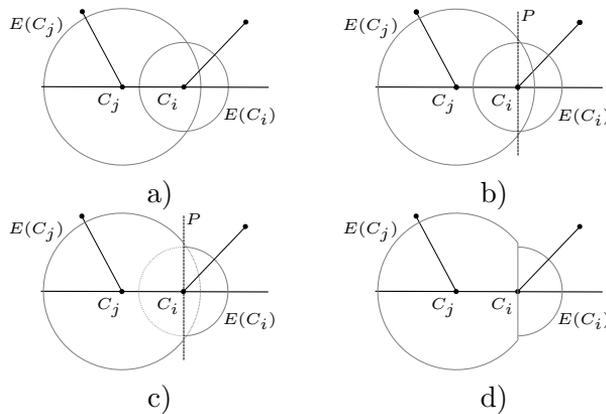


Figura 6.10: Proceso para la separación de restricciones no disjuntas mediante hiperplanos. En a) se muestran dos restricciones R_i y R_j aplicadas a los puntos C_i y C_j respectivamente y cuyas envolventes de influencia son $E(C_i)$ y $E(C_j)$, puede observarse que $C_i \in E(C_j)$ por lo que las restricciones son no disjuntas. En b) se determina la recta P que pasa por C_i y es ortogonal a C_iC_j , los puntos de $E(C_i) \cap E(C_j)$ que se encuentran en el semiplano que contiene a C_i pertenecen a $E(C_i)$, mientras que los que se encuentran en el semiplano que no contiene a C_i pertenecen a $E(C_j)$, c). Finalmente, en d) puede verse como quedan definidos $E(C_i)$ y $E(C_j)$.

En [LG05] se plantea una solución iterativa que converge hacia la solución óptima. En dicho trabajo se comenta, aunque no se demuestra, que en el caso que haya restricciones no disjuntas el resultado es que los puntos afectados tienen un comportamiento equivalente a la sustitución de todas las restricciones por una única situando el punto de destino en el baricentro de todas las restricciones afectadas.

6.3.3. Particularidades de la implementación sobre BSpline4D

Todo lo descrito hasta aquí es perfectamente aplicable a cualquier tipo de objeto sin tener en cuenta el modelo de deformación utilizado. El inconveniente de utilizar modelos basados en BSpline4D es que la manipulación de los mismos a través de Sodef provoca la pérdida de la relación entre el objeto representado y su estructura, tal y como se puede observar en la figura 6.7 en donde la superficie cambia, pero los puntos de control de la misma permanecen inalterados. El objetivo último de este apartado es el de la aplicación de Sodef sobre superficies BSpline4D con la condición de que éstas mantengan sus propiedades.

Para que una deformación basada en restricciones de una superficie BSpline4D

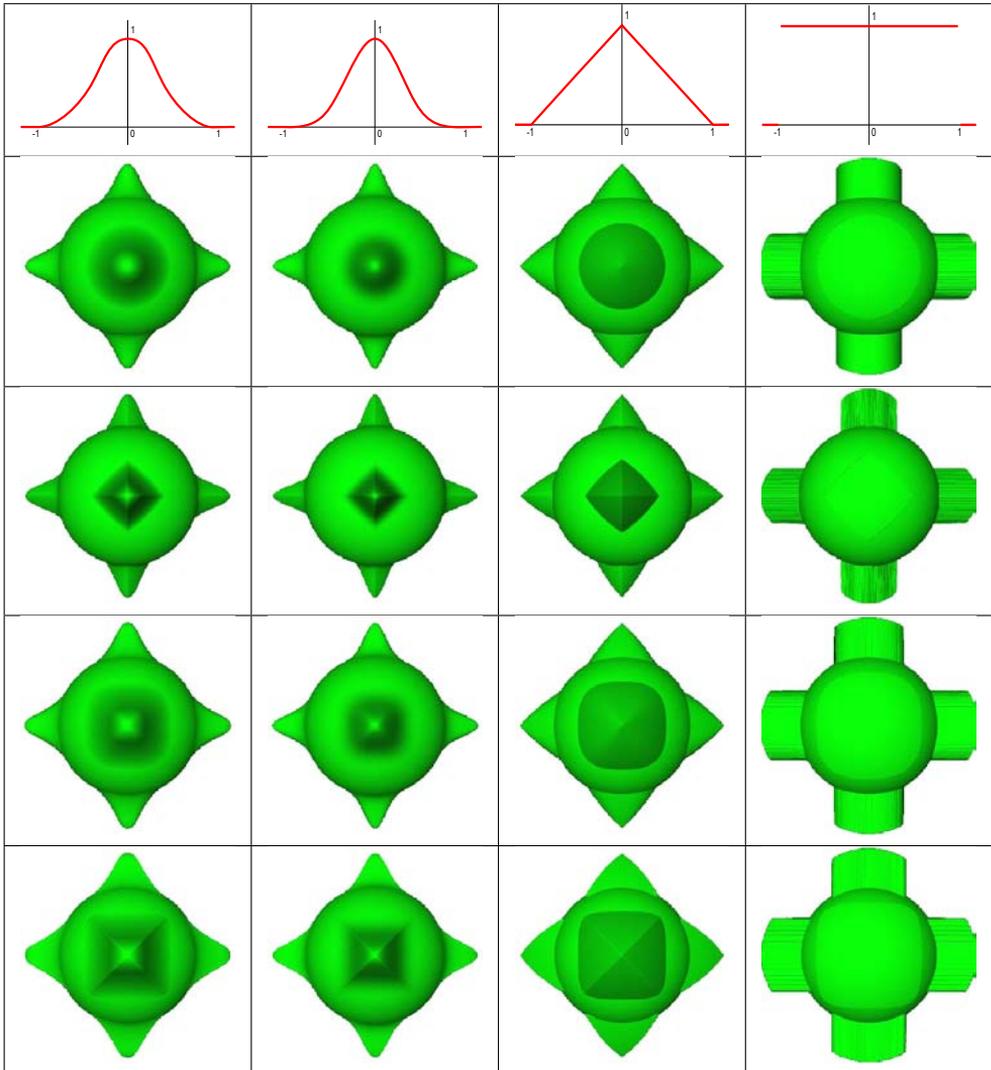


Figura 6.11: Deformación de una esfera.

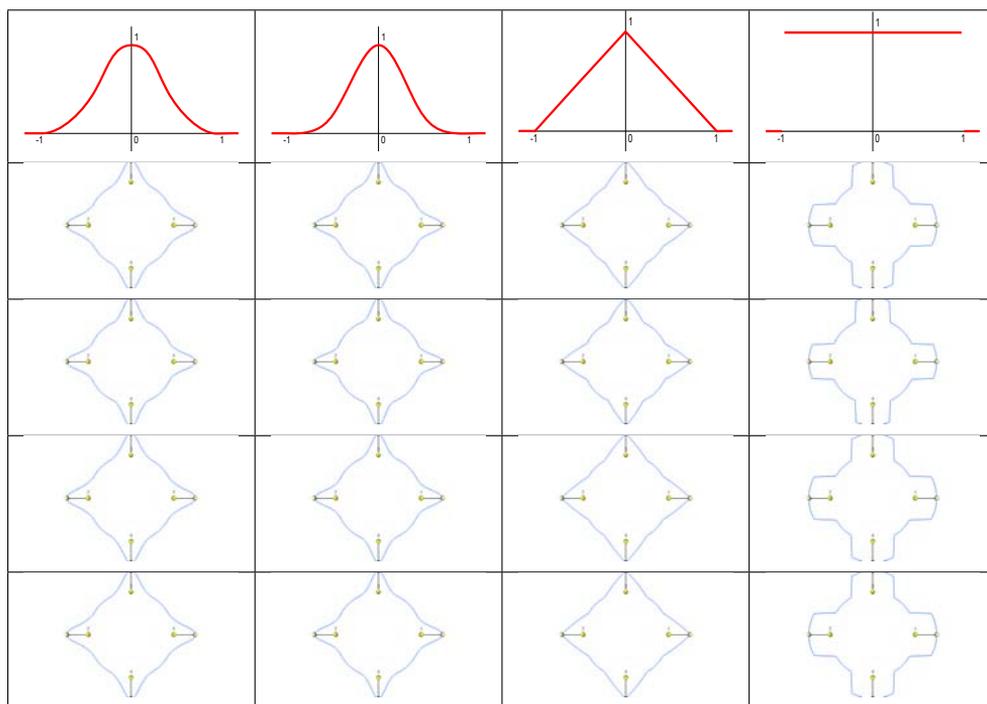


Figura 6.12: Las mismas deformaciones de la figura 6.11 pero aplicadas a un círculo. En esta figura se muestran también las restricciones aplicadas.

sea tal, un punto de la superficie deberá ser trasladado a otro lugar del espacio, el resto de puntos que se encuentren dentro del entorno de influencia de la restricción deberán ser trasladados según sea la función de deformación elegida. Si además la superficie resultante debe seguir siendo una superficie BSpline4D, entonces los puntos de control deberán ser modificados de forma coherente. Si la deformación es puntual, es decir el punto en donde se aplica la restricción se traslada a otro lugar del espacio *directamente*, entonces el planteamiento descrito en [LG05] y en [LGRG07] es más que suficiente.

Para una superficie BSpline4D $S(u, v)$ de grado p y q con $m \times n$ puntos de control a la que se desea aplicar una deformación basada en restricciones, sea C el punto de la superficie en donde se aplica la restricción y sea \tilde{C} el punto al que debe ir a parar, de manera que $D = \tilde{C} - C$ es el desplazamiento de dicho punto como resultado de la deformación. La superficie resultante $\tilde{S}(u, v)$ será una superficie BSpline4D de grados p y q y con $m \times n$ puntos de control de manera que si $S(u, v) = C$ entonces $\tilde{S}(u, v) = \tilde{C}$.

En general la ubicación de cada punto de una forma BSpline4D no queda afectada por todos los puntos de control de la misma. Supóngase el caso de una superficie BSpline4D, al punto del espacio paramétrico (u, v) le corresponde el punto del espacio $S(u, v)$, el valor de dicha función (véase la ecuación (4.31), en la página 137) depende de las funciones de base B-Spline (4.25), que, en definitiva, determinan el grado de influencia de cada punto de control en la ubicación de (u, v) . El valor de una función de base B-Spline, $N_{i,p}(u)$, es 0 si $u \notin [U_i, U_{i+p+1})$. Por lo tanto para un valor dado $u \in [U_i, U_{i+1})$ las únicas funciones de base con valor no nulo son $N_{i-p,p}(u) \dots N_{i,p}(u)$. Por lo tanto, para el caso de una superficie, si el punto del espacio paramétrico (u, v) pertenece al intervalo $[U_i, U_{i+1}) \times [V_j, V_{j+1})$, entonces su ubicación dependerá solamente de los puntos de control P_{kl} , $i-p \leq k \leq i$, $j-q \leq l \leq j$ siendo p y q los grados de la superficie.

Así pues, para convertir S en \tilde{S} , se deberá modificar cada punto de control original, P_{ij} , añadiéndole un cierto desplazamiento, $m(i, j)$, para obtener un nuevo punto de control desplazado \tilde{P}_{ij} :

$$\tilde{P}_{ij} = P_{i,j} + m(i, j). \quad (6.12)$$

Por lo tanto

$$\begin{aligned}
 \tilde{C} = \tilde{S}(u, v) &= \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \tilde{P}_{ij} \\
 &= \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) (P_{ij} + m(i, j)) \\
 &= \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{ij} + \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) m(i, j) \\
 &= S(u, v) + \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) m(i, j) \\
 &= C + D,
 \end{aligned}$$

de donde se deduce que

$$D = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) m(i, j). \quad (6.13)$$

Queda por definir cómo debe ser $m(i, j)$ para que la igualdad (6.13) se verifique.

De forma general, es preciso determinar qué puntos de control quedan afectados por la restricción, y por lo tanto deberán modificarse. Cada restricción tiene una influencia sobre el punto $C = S(u, v)$ y los puntos de control que le afectan. Para definir esta influencia se utilizará lo que pasará a denominarse *función de restricción de localización*, $\hat{f}(i, j)$, que controlará la influencia de la restricción sobre el conjunto de puntos de control que afectan a C . Es decir se define la función $\hat{f}(i, j)$ para que, a la hora de definir la superficie resultado de la deformación, solamente los puntos de control que sea preciso modificar realmente cambien. La función se define de manera que cada punto de control que influye sobre C puede quedar más o menos afectado por la deformación:

$$\hat{f} : \{0, \dots, n\} \times \{0, \dots, m\} \longrightarrow \mathbb{R}^+. \quad (6.14)$$

Existe además un condicionante adicional, consistente en que al menos algunos puntos de control que cambien deberán pertenecer a la envolvente convexa del punto, es decir:

$$\exists (i, j) \in \{0, \dots, n\} \times \{0, \dots, m\} \mid N_{i,p}(u) N_{j,q}(v) \hat{f}(i, j) \neq 0. \quad (6.15)$$

En definitiva, $\hat{f}(i, j)$ indica qué puntos de control son afectados por una restricción y en qué medida lo son. En principio cualquier función que satisfaga las

condiciones anteriores puede servir, en [LG05] se proponen varias, por ejemplo:

$$\begin{aligned}\hat{f}(i, j) &= \cos(i)\cos(j), \\ \hat{f}(i, j) &= \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{\sigma^2}}.\end{aligned}$$

Una forma sencilla de determinar el valor de esta función es estableciendo que dicha restricción venga dada por las propias características de la superficie a tratar, de manera que las propias funciones de base utilizadas para expresar la superficie sean utilizadas para determinar la función de localización, es decir:

$$\hat{f}(i, j) = N_{i,p}(u)N_{j,q}(v). \quad (6.16)$$

Incorporando el concepto de restricción de localización definido, se puede definir el desplazamiento de los puntos de control, $m(i, j)$, de la siguiente forma:

$$m(i, j) = \frac{\hat{f}(i, j)}{\sum_{k=0}^n \sum_{l=0}^m N_{kp}(u)N_{lq}(v)\hat{f}(k, l)} D, \quad (6.17)$$

demostrándose la igualdad (6.13):

$$\begin{aligned}D &= \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)m(i, j) \\ &= \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)f(i, j)}{\sum_{k=0}^n \sum_{l=0}^m N_{kp}(u)N_{lq}(v)f(k, l)} D \\ &= D.\end{aligned}$$

Si se utiliza (6.16) entonces

$$m(i, j) = \frac{N_{i,p}(u)N_{j,q}(v)}{\sum_{k=0}^n \sum_{l=0}^m (N_{kp}(u)N_{lq}(v))^2} D. \quad (6.18)$$

Observando las expresiones (6.16) y (6.17) se puede comprobar la exigencia de que la función de restricción de localización deba satisfacer (6.14) y (6.15). En la figura 6.13 pueden verse algunos ejemplos de deformaciones obtenidas mediante el método descrito, en la figura se puede observar el resultado de aplicar

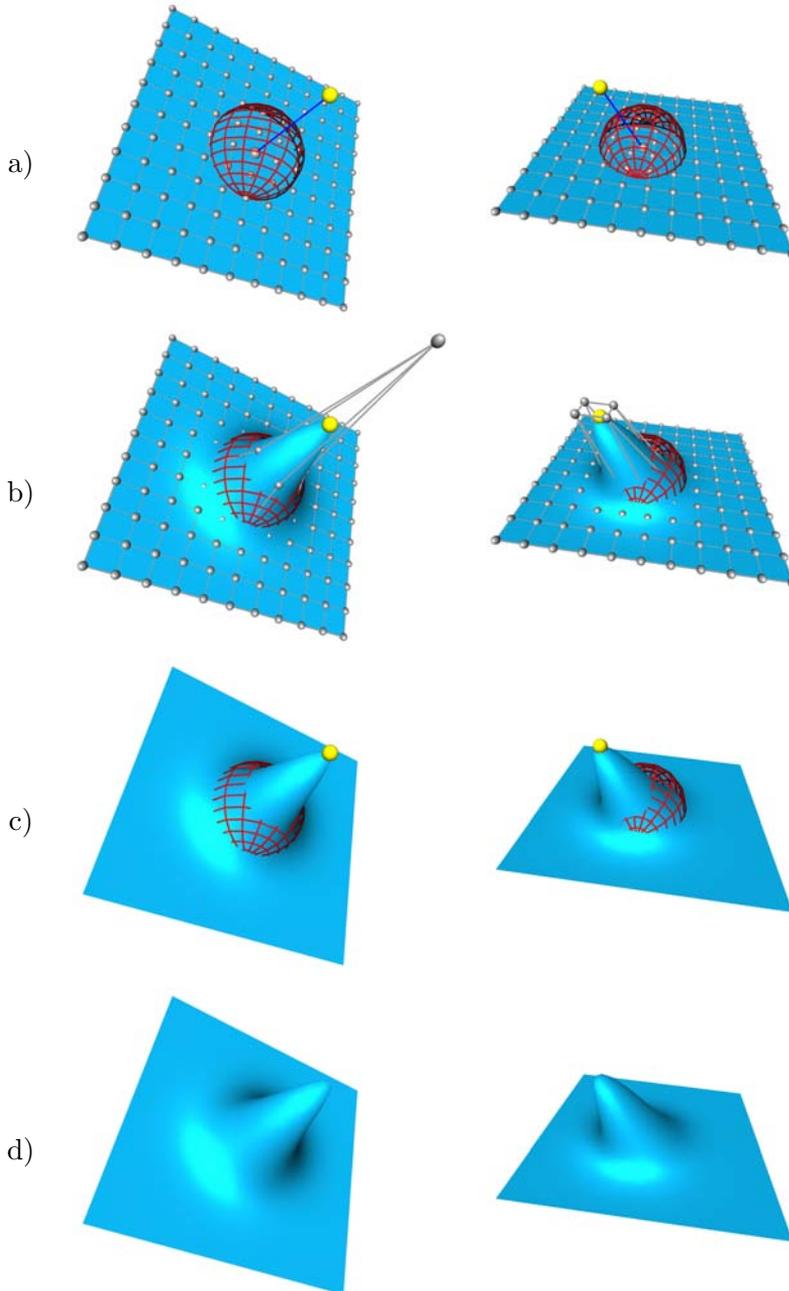


Figura 6.13: Dos ejemplos de deformación de una superficie mediante la aplicación de una restricción lineal. a) la superficie con la restricción definida; b) la deformación mostrándose los puntos de control y la restricción; c) la superficie deformada con la restricción; d) la superficie tal cual resulta.

la restricción en un punto coincidiendo con un punto de control, imágenes de la izquierda, y también en un punto arbitrario, imágenes de la derecha.

En definitiva, mediante la utilización de la función de restricción de localización, $\hat{f}(i, j)$, se consigue que para cada punto de control, $P_{i,j}$, cuya imagen paramétrica² caiga fuera de la zona paramétrica de influencia³ se mantengan inalterados ya que $\hat{f}(i, j) = 0$. En cambio, para los puntos de control cuya imagen paramétrica caiga dentro de la zona paramétrica de influencia, $f(i, j)$ representará la medida en que $P_{i,j}$ deberá cambiar, tanto más cuanto más cerca se encuentre la imagen paramétrica de $P_{i,j}$ del punto del espacio paramétrico al que se aplica la restricción. Si en la zona paramétrica de influencia no hay suficientes puntos de control como para realizar la deformación, previamente a su cálculo se deberá proceder a añadir los suficientes puntos de control para poder asegurar que la deformación se podrá realizar de forma correcta.

6.3.4. Restricciones sobre trayectorias

La definición de una deformación basada en restricciones siguiendo una trayectoria sobre una superficie BSpline4D de manera que ésta preserve sus propiedades es un trabajo no del todo resuelto. Un planteamiento como el descrito en [LG05] y en [LGRG07] no parece suficiente. En esta sección se propone una forma alternativa para realizar esta operación.

Los principales elementos a considerar para poder definir una deformación basada en restricciones según una trayectoria son los siguientes:

- La curva de trayectoria, C^t , que determinará la trayectoria que deberá seguir la deformación. Vendrá dada por una curva BSpline4D.
- La curva de perfil, C^p , que determinará el contorno de la deformación. En lugar de definir un entorno de influencia mediante algún tipo de función matemática, se puede definir mediante una curva BSpline4D, en principio cerrada, que determinará el área a deformar. La curva bien puede ser indicada mediante un proceso interactivo con el usuario, en el que se dibujaría dicha curva sobre el propio objeto, o bien puede ser seleccionada una curva ya definida que sería aplicada sobre el objeto en la zona a deformar.

Las curvas implicadas pueden tener el espacio paramétrico definido entre cualesquiera dos valores reales a y b , con la única restricción que $a < b$ (aunque por

²La imagen paramétrica de un punto de control $P_{i,j}$ se define como el punto (u, v) del espacio paramétrico de la superficie para el que la influencia de $P_{i,j}$ es máxima.

³La región del espacio paramétrico cuyos valores tienen una imagen que se encuentra en la zona de influencia de la restricción.

simplicidad se supondrá $a = 0$ y $b = 1$), con cualquier grado, número de nodos y número de puntos de control.

La idea es que la curva de perfil seguirá el camino definido por la curva de trayectoria orientándose según el plano normal, véase la figura 6.15, en cada punto y escalada según una función análoga a la función de deformación (6.3), de forma parecida a la que se muestra en la figura 6.16. El recorrido será desde el punto $C^t(0)$ hasta el punto $C^t(1)$. En aras de la generalidad se supondrá que tanto las curvas suministradas como la superficie a deformar pueden ubicarse en cualquier lugar del espacio. Por lo tanto la curva de trayectoria puede que no esté en contacto con la superficie. El punto, x , de aplicación de la restricción sobre la superficie S será el correspondiente al de mínima distancia entre $C^t(0)$ y S .

Al ser una curva cualquiera, la curva de perfil no tiene por qué tener un tamaño acorde con la región sobre la que se va a realizar la deformación ni tampoco tiene por qué estar situada en el lugar deseado. Por ello, para que la curva de perfil determine el contorno de deformación, primeramente deberá ser orientada de manera que se sitúe sobre S en el lugar en donde se vaya a realizar la deformación, el punto x antes descrito. Sin pérdida de generalidad se procederá a trasladar C^p de manera que el baricentro de los puntos de control coincida con x , aunque podría utilizarse cualquier otro criterio.

De cara a ajustar la dimensión de C^p se definirá también una función de distancia que determinará el área de la superficie que quedará afectada por la deformación. Esta distancia fijará un factor de escala para C^p de manera que ésta se ajuste de forma adecuada. Así, si se determina que el radio de influencia de la restricción es r , y x se encuentra a una distancia l de $C^t(0)$, entonces la distancia efectiva que determina la escala de la curva será $d = \sqrt{r^2 - l^2}$. Por lo tanto, C^p quedará inscrita en la circunferencia de radio d centrada en x y orientada según el plano rectificante de S en x . Si la distancia l es mayor que el radio r la superficie no se verá afectada por la restricción. En la figura 6.14 se muestran los elementos indicados, en esta figura se representa C^p como un círculo.

Por lo tanto para calcular la deformación se deberá proceder a ajustar los puntos de control de la superficie de manera que la región que se encuentre a una distancia menor o igual a d del punto x , de mínima distancia entre S y C^t , se ajuste a C^p . La distancia que separe cada uno de los puntos del punto x servirá para determinar el lugar al que debe ir a parar, para ello se utilizaran funciones del estilo de la descrita en (6.3).

La forma que adoptará la superficie en la zona deformada y la manera en que lo hará recuerda la técnica descrita en [PT97] para la generación de superficies de

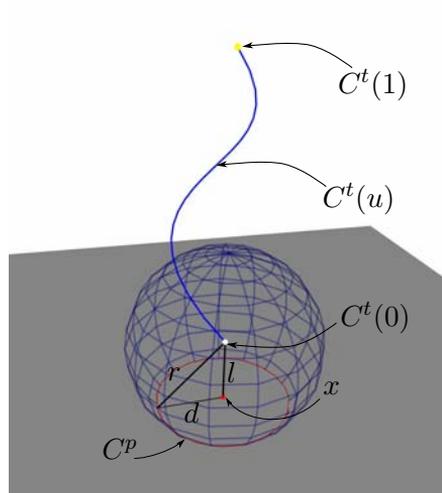


Figura 6.14: Elementos a considerar para realizar una deformación sujeta a una trayectoria.

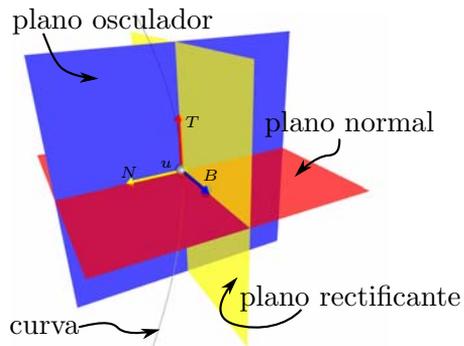


Figura 6.15: Para cada punto, u de una curva definida en el espacio se definen tres vectores ortogonales, T , el vector tangente a la curva, N , el vector normal y B el vector binormal.

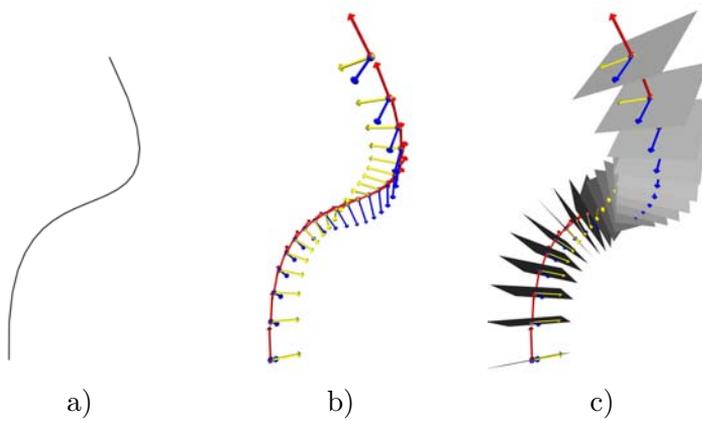


Figura 6.16: Para una curva en el espacio, en cada punto es posible determinar cuales son los vectores tangente (en rojo), normal (en amarillo) y binormal (en azul) y por ello es posible determinar el plano normal en cada punto, c). De esta forma es posible conocer la orientación de la curva en cada punto.

barrido (*swept surfaces*). Por ello se realizará primeramente un estudio de dicha técnica para determinar su adecuación al modelo BSpline4D y a la deformación de las superficies basada en restricciones.

Superficies de barrido

Una superficie de barrido se define a partir de una curva de trayectoria, $C^t(v)$, y una curva de perfil, $C^p(u)$, cualesquiera:

$$S(u, v) = C^t(v) + A(v)E(v)C^p(u) \quad (6.19)$$

siendo

- $E(v)$ una matriz de escalado que permite ajustar el tamaño de $C^p(u)$ en función de v ,
- $A(v)$ es una matriz de transformación que permite pasar de un sistema de referencia global a uno local orientado según sea conveniente.

Si se expresa

$$C^t(v) = \sum_{j=0}^n N_{j,p}(v)P_j^t, \quad (6.20)$$

y

$$C^p(u) = \sum_{i=0}^m N_{i,q}(u) P_i^p, \quad (6.21)$$

entonces

$$\begin{aligned} S(u, v) &= \sum_{j=0}^n N_{j,p}(v) P_j^t + A(v) E(v) \sum_{i=0}^m N_{i,q}(u) P_i^p \\ &= \sum_{j=0}^n N_{j,p}(v) P_j^t + \sum_{i=0}^m N_{i,q}(u) A(v) E(v) P_i^p, \end{aligned} \quad (6.22)$$

teniendo en cuenta la propiedad partición de la unidad de las funciones de base B-Spline:

$$\begin{aligned} \sum_{i=0}^m N_{i,q}(u) &= 1, \\ \sum_{j=0}^n N_{j,p}(v) &= 1, \end{aligned}$$

se puede arreglar la expresión:

$$\begin{aligned} &\sum_{j=0}^n N_{j,p}(v) P_j^t + \sum_{i=0}^m N_{i,q}(u) A(v) E(v) P_i^p = \\ &\sum_{i=0}^m N_{i,q}(u) \sum_{j=0}^n N_{j,p}(v) P_j^t + \sum_{j=0}^n N_{j,p}(v) \sum_{i=0}^m N_{i,q}(u) A(v) E(v) P_i^p = \\ &\sum_{i=0}^m \sum_{j=0}^n N_{i,q}(u) N_{j,p}(v) P_j^t + \sum_{j=0}^n \sum_{i=0}^m N_{j,p}(v) N_{i,q}(u) A(v) E(v) P_i^p, \end{aligned}$$

y por lo tanto

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,q}(u) N_{j,p}(v) (P_j^t + A(v) E(v) P_i^p). \quad (6.23)$$

Por lo tanto los puntos de control de $S(u, v)$ se obtienen a partir de los puntos de control de $C^t(v)$ (P_j^t $j = 0, \dots, m$) y de $C^p(u)$ (P_i^p $i = 0, \dots, n$):

$$P_{i,j} = P_j^t + A(v) E(v) P_i^p \quad i = 0, \dots, n \quad j = 0, \dots, m. \quad (6.24)$$

Si el desarrollo se realiza para superficies NURBS, partiendo de curvas NURBS la expresión que se obtiene es

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,q}(u) N_{j,p}(v) w_i w_j \left(P_j^t + A(v) E(v) P_i^p \right)}{\sum_{i=0}^m \sum_{j=0}^n N_{i,q}(u) N_{j,p}(v) w_i w_j}. \quad (6.25)$$

En este caso, los pesos NURBS de la superficie se corresponden con el producto de los pesos de $C^t(v)$ (w_j^t $j = 0, \dots, m$) y de $C^p(u)$ (w_i^p $i = 0, \dots, n$):

$$w_{i,j} = w_j^t w_i^p \quad i = 0, \dots, n \quad j = 0, \dots, m. \quad (6.26)$$

Dependiendo de cuales sean las curvas elegidas, la función (6.23) puede dar como resultado superficies con formas no deseadas. Según sea la forma de $A(v)$ es posible incluso que la forma obtenida no sea NURBS. Sin embargo con el método que se describirá en este apartado la generación de superficies produce resultados razonables.

Evidentemente, el resultado dependerá de como sean las curvas elegidas, si la torsión o la curvatura de la curva de trayectoria son grandes es posible que el resultado no tenga la continuidad que sería deseable, además según sea la trayectoria, es posible que se produzcan autointersecciones. Todos estos efectos no deseables pueden ser subsanados por el usuario que defina la superficie modificando convenientemente los parámetros de las curvas.

La idea es determinar una serie de curvas de interpolación $\tilde{C}_i^p(u)$ obtenidas a partir de $C^p(u)$ desplazadas y orientadas convenientemente a lo largo de $C^t(v)$. La superficie $S(u, v)$ se obtendrá a partir de la interpolación de las curvas $\tilde{C}_i^p(u)$ calculadas.

Para el cálculo de $A(v)$ será preciso determinar para cada punto de control de C^t cual es su proyección sobre la propia curva C^t de manera que se pueda determinar cual es el triedro de Frenet para dicho punto. El valor de los puntos de $C^t(v)$ para cada una de las curvas $\tilde{C}_i^p(u)$ será el de mínima distancia entre la curva y el correspondiente punto de control. Por lo tanto será la misma curva de trayectoria, con sus características, la que determinará la cantidad de curvas de interpolación necesarias. Así pues, para cada punto de control P_j^t siendo $0 \leq j \leq m$ se deberá determinar el parámetro $v_j \in [V_{j-q}, V_{j+1})$ tal que:

$$\forall v \in [V_{j-q}, V_{j+1}) \quad \|P_j^t - C^t(v_j)\| \leq \|P_j^t - C^t(v)\|.$$

Conocido v_j es posible determinar el triedro de Frenet para el punto $C^t(v_j)$, y de esta forma es posible conocer el plano normal, definido por los vectores normal

y binormal, y el vector tangente; y con ellos se podrá conocer la orientación que debe tener la curva \tilde{C}_j^p . Si $T(C(v_j))$, $N(C(v_j))$ y $B(C(v_j))$ son, respectivamente, los vectores tangente, normal y binormal de la curva C^t en el punto v_j , entonces el valor de la matriz de transformación $A(v_j)$ para $\tilde{C}_j^p(u)$ será:

$$A(v_j) = \begin{pmatrix} T(v_j)_x & N(v_j)_x & B(v_j)_x \\ T(v_j)_y & N(v_j)_y & B(v_j)_y \\ T(v_j)_z & N(v_j)_z & B(v_j)_z \end{pmatrix}^{-1}. \quad (6.27)$$

Esta expresión es válida para orientar puntos definidos en \mathbb{R}^3 . Teniendo en cuenta que los puntos de las formas BSpline4D son de \mathbb{R}^4 y que la cuarta dimensión representa los pesos de los puntos de control, que no deben variar, la expresión (6.27) debe ser modificada para contemplar puntos de \mathbb{R}^4 de manera que la cuarta dimensión no quede alterada:

$$A(v_j) = \begin{pmatrix} T(v_j)_x & N(v_j)_x & B(v_j)_x & 0 \\ T(v_j)_y & N(v_j)_y & B(v_j)_y & 0 \\ T(v_j)_z & N(v_j)_z & B(v_j)_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1}. \quad (6.28)$$

Si el objetivo es orientar la curva de perfil según el plano normal, la aplicación de $A(v_j)$ a los puntos de control de $C^p(u)$ directamente no es suficiente ya que si el baricentro de los puntos de control de $C^p(u)$ no se encuentra en el origen de coordenadas el resultado no es el esperado. Por lo tanto previamente a su orientación, los puntos de control deberán ser trasladados de forma que el baricentro se sitúe en el origen de coordenadas. A tal fin, resulta conveniente modificar $E(v)$ para que además de escalar la curva de perfil proceda a trasladarla de manera que el baricentro, P_c , de la misma coincida con el origen de coordenadas. Si el baricentro de los puntos de control se puede expresar como

$$P_c = \frac{1}{m+1} \sum_{i=0}^m P_i^p, \quad (6.29)$$

entonces, teniendo presente las consideraciones que han llevado a (6.28):

$$E(v_j) = \begin{pmatrix} S_{j,x} & 0 & 0 & P_{c,x} \\ 0 & S_{j,y} & 0 & P_{c,y} \\ 0 & 0 & S_{j,z} & P_{c,z} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (6.30)$$

Una vez orientados, será preciso trasladar nuevamente los puntos para conseguir que el baricentro se sitúe en el lugar en donde se encuentra el punto de

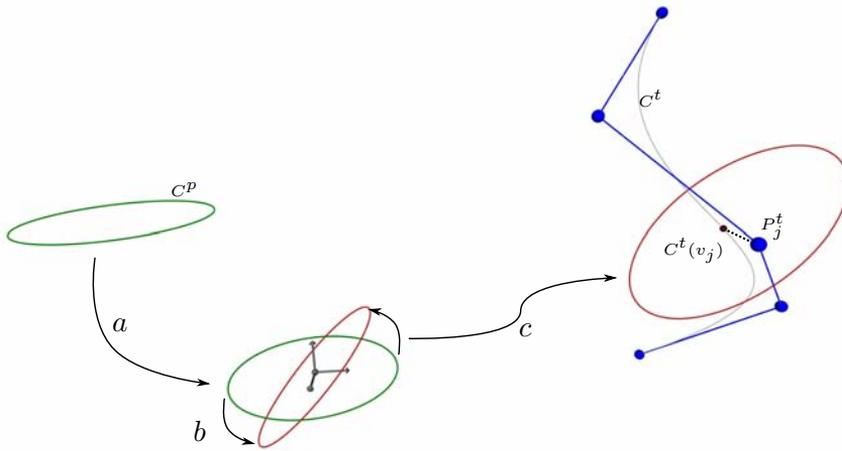


Figura 6.17: Orientación de la curva de perfil. Una curva de perfil es trasladada al origen de coordenadas, *a*. A continuación se procede a cambiar su orientación según los valores calculados para P_j^t , *b*. Finalmente se traslada la curva para que su baricentro coincida con $C^t(v_j)$, *c*.

control P_j^t . Así pues:

$$P_{i,j} = P_j^t + A(v_j) \left(P_i^p - \frac{\sum_{k=0}^n P_k^p}{n+1} \right), \quad \begin{matrix} 0 \leq j \leq m, \\ 0 \leq i \leq n. \end{matrix}$$

En la figura 6.17 puede verse de forma gráfica este proceso.

La generación de superficies con este método permite obtener como resultado superficies suaves. En las imágenes 6.18, 6.19 y 6.20 pueden verse ejemplos de superficies generadas con este modelo.

Como puede observarse en la figura 6.20 la curvatura y la torsión de la curva de trayectoria pueden ser determinantes a la hora de generar la superficie. Es posible que si dichos valores son grandes, entonces no sea suficiente con establecer que haya tantas curvas \tilde{C}_j^p como puntos de control tenga C^t . En la imagen de la izquierda, la curva de trayectoria es suficientemente suave como para que la superficie generada sea a su vez suave, en cambio, en la imagen de la derecha esto no es así. La curva de trayectoria tiene un segmento con un importante cambio en la torsión que provoca que la superficie generada tenga un *aplastamiento* en la región correspondiente. Es posible compensar dicho efecto mediante el proceso de añadir uno o más puntos de control en las zonas de la curva de trayectoria

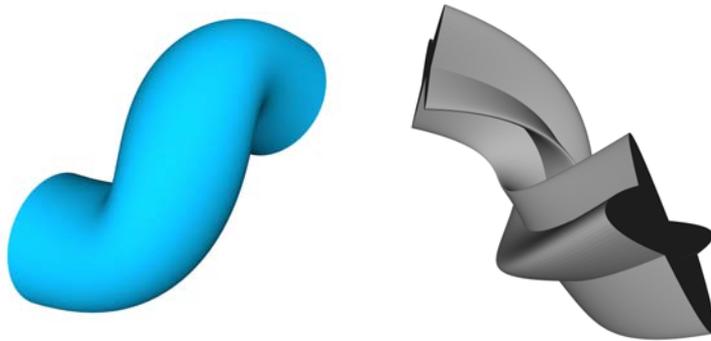


Figura 6.18: Dos superficies obtenidas a partir de la misma curva de trayectoria pero con diferentes curvas de perfil. La curva de perfil mantiene sus dimensiones desde el principio al final.

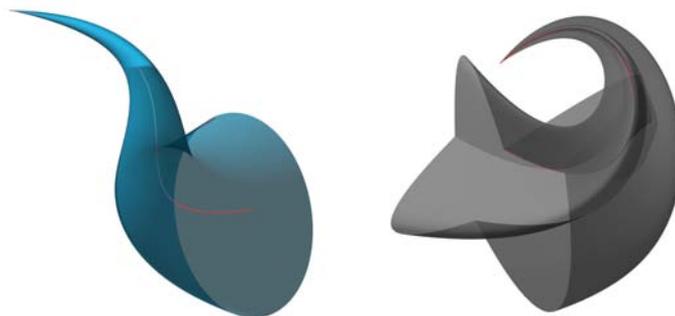


Figura 6.19: Dos superficies obtenidas a partir de la misma curva de trayectoria pero con diferentes curvas de perfil. La curva de perfil disminuye su tamaño de acuerdo con (6.3). Las superficies se han generado con un cierto grado de transparencia para que se pueda apreciar la curva de trayectoria sobre la que se han generado.

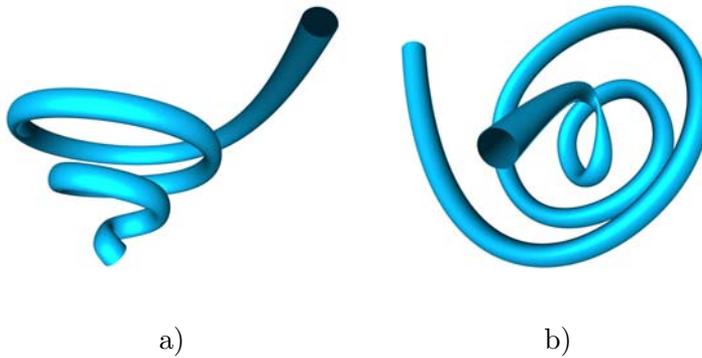


Figura 6.20: En general si la forma de la curva de trayectoria es suficientemente suave los resultados son siempre razonables, a). Sin embargo, si la curva de trayectoria tiene cambios grandes en su curvatura o torsión se pueden producir efectos como el del *aplastamiento* de la curva de perfil que se puede apreciar en la figura b).

que tengan una curvatura o una torsión excesivos. De esta forma el algoritmo no cambia, simplemente se precisa establecer un preprocesamiento previo para determinar si tal modificación es precisa, figura 6.21.

Una vez desarrollada la técnica de generación de superficies de barrido, falta ahora plantear la posibilidad de extender el método para poder obtener la deformación de una superficie previamente definida en lugar de tener la libertad de ubicar puntos de control sin ninguna restricción adicional.

Deformación de superficies mediante restricciones con trayectorias

Para definir las superficies según el modelo basado en curvas de barrido anteriormente descrito, y de acuerdo con (6.23), se ha establecido que para cada punto de control de la curva de trayectoria se debe obtener una curva de interpolación a partir de la curva de perfil. Tomando en consideración que no se trata ahora de generar una superficie completamente nueva sino que lo que se busca es modificar una superficie ya existente, es muy posible que la superficie no disponga de suficientes puntos de control en la región a deformar para poder ajustar dicha región a la forma deseada. Por lo tanto será preciso asegurar la existencia de suficientes puntos de control de cara a obtener un resultado correcto.

En la figura 6.22 se muestra un ejemplo ilustrativo: una superficie BSpline4D a la que se desea aplicar una deformación basada en una restricción con relativamente pocos puntos de control, concretamente se trata de una superficie de

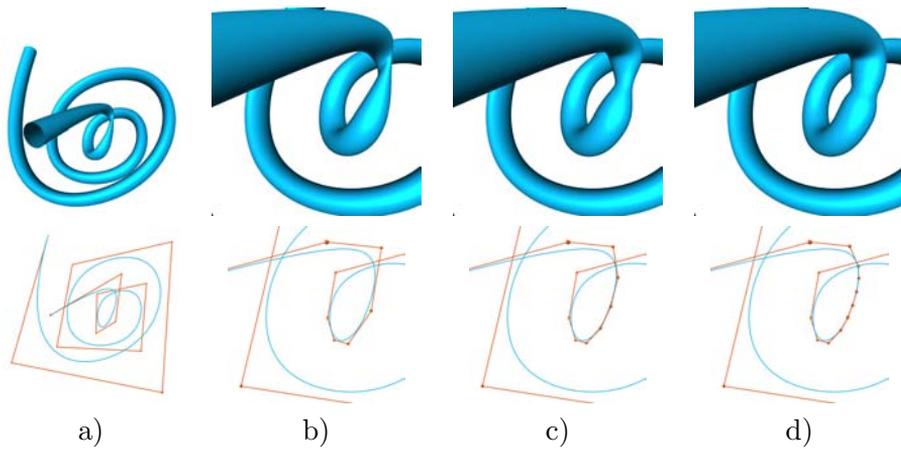


Figura 6.21: Para aquellas curvas de trayectoria cuya torsión o curvatura provoquen efectos visualmente poco deseables, a), es posible realizar un preprocesamiento de las mismas para añadir puntos de control adicionales que permitan paliar, al menos en parte, el efecto no deseado. En la fila superior se muestran superficies generadas con las mismas curvas de trayectoria y perfil, a medida que se añaden puntos de control en la zona afectada el efecto se reduce. En la fila inferior se muestran las correspondientes curvas de trayectoria con su polígono de control.

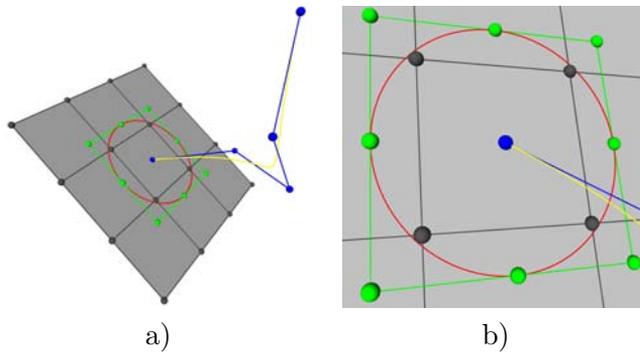


Figura 6.22: Restricción aplicada a una superficie BSpline4D. En a) se muestra la superficie BSpline4D con sus puntos de control (esferas grises) y una restricción definida mediante una curva de trayectoria B-Spline, en azul, y una región de influencia definida mediante una curva de perfil B-Spline, en rojo. En b) se muestra el detalle de la zona de influencia, en su interior no aparecen suficientes puntos de control de la superficie como para conseguir que ésta se ajuste a la curva de trayectoria.

grado 3 en ambas dimensiones paramétricas y con 4×4 puntos de control. La deformación debe seguir la trayectoria definida por una curva BSpline4D de grado 3 con 5 puntos de control. El problema surge con la zona que debe quedar afectada por la restricción, el círculo de color rojo en la figura, que resulta ser tal que no incluye suficientes puntos de control de la superficie (las esferas de color gris) en su interior, escasamente 4 puntos de control se encuentran en la región definida por C^p , con los que no es posible obtener una forma tan elaborada como la que sugiere la curva de trayectoria. Si el objetivo es conseguir una superficie BSpline4D, necesariamente deberán añadirse más puntos de control.

Todos los intentos para determinar un modelo que permita insertar uno o más puntos de control en una superficie sin cambiar su forma han resultado infructuosos toda vez que siempre se ha obtenido un sistema de ecuaciones indeterminado. Ello parece lógico toda vez que, en definitiva, se trata de añadir puntos de control situados en literalmente cualquier lugar del espacio y pretender que dicha operación no afecte en absoluto a la forma de la superficie.

Descartado un método directo para realizar la inserción de los puntos de control necesarios se constata que el método para realizar dicha operación deberá ser mediante un planteamiento indirecto como puede ser el de la obtención de nuevos puntos de control a través del proceso de refinamiento de nodos, [PT97]. Utilizar este método para obtener nuevos puntos de control tiene el problema de que los puntos de control obtenidos no tienen por qué ajustarse exactamente al lugar previsto, sino que pueden verse desplazados respecto del lugar en el cual sería

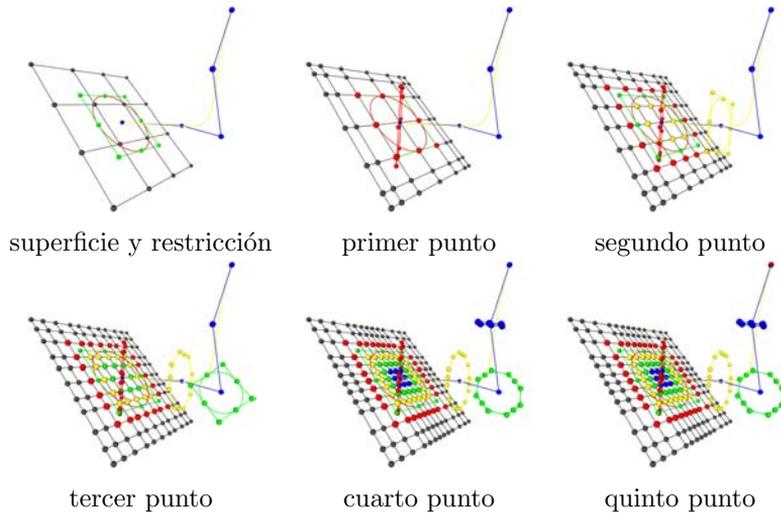


Figura 6.23: Sucesivos refinamientos de los puntos de control de una superficie. Los puntos de control originales de la superficie son modificados para poder disponer de suficientes puntos de control para satisfacer la restricción y según sean la curva de trayectoria y de perfil.

deseable que estuvieran.

En cuanto a la forma en que dicho proceso puede realizarse, de entrada podría parecer razonable añadir a la vez todos los puntos de control que sean precisos para representar toda la deformación. Sin embargo este modo de actuar no es aconsejable toda vez que el modelo propuesto busca la generalidad. Como no es previsible la forma que pueda tener la curva de perfil, resulta difícil determinar *a priori* qué puntos deben ser añadidos y cuales no. Descartada esta opción, la estrategia adoptada se basa en un planteamiento iterativo, insertando primeramente los puntos de control necesarios para poder ajustar la superficie según el primer punto de control de la curva de trayectoria, una vez insertados los puntos necesarios se procede a determinar qué puntos son precisos para el segundo punto de control para poder insertar aquéllos para los que no se disponga de puntos en la superficie. Se procederá de esta forma para todos y cada uno de los puntos de control de la curva de trayectoria. En la figura 6.23 se puede apreciar el resultado.

El refinamiento sucesivo de los puntos de control provoca una serie de efectos no esperados y que deben ser tenidos en consideración:

- Los puntos de control se obtienen como consecuencia del refinamiento del vector de nodos correspondiente a uno de los dos parámetros de la super-

ficie. Al añadirse un nodo es preciso añadir también un punto de control para mantener el grado⁴. Al tratarse de una superficie el punto de control no puede ser único debido a que la forma del poliedro de control de la superficie BSpline4D, al igual que el resto de superficies B-Spline, requiere un número igual de elementos en cada fila y en cada columna. Por lo tanto deberá añadirse una fila o una columna completas de puntos de control.

- El proceso de refinamiento de nodos, además de crear un punto de control, afecta a la ubicación de un cierta cantidad de puntos control ya existentes, en número igual al grado del parámetro de la superficie para el cual se crea el nodo. Inevitablemente esto provoca un desplazamiento de los puntos de control que deberá ser tenido en cuenta si es preciso identificar lo puntos según sea su ubicación esperada.
- El método iterativo descrito hace que la cantidad de puntos de control implicados en la deformación aumente de forma considerable. Para cada paso de la iteración se añaden los puntos de control previstos para deformar la superficie de acuerdo con un determinado punto de control de la curva de trayectoria, y además pueden aumentar los puntos destinados a deformar la superficie según los puntos de control de la curva de trayectoria previamente calculados. Esto es así también como consecuencia de tener que añadir toda una fila o columna de puntos de control.

Por lo tanto el número de puntos de la superficie puede ser superior al previsto. Esto hace que la correspondencia entre los puntos de control de la superficie y los puntos de control de la curva de perfil, objetivo de todo este proceso, deje de existir. Este hecho requerirá que, a la hora de deformar convenientemente los puntos, sea preciso realizar un refinamiento de los puntos de control de la curva de perfil para que nuevamente sea posible realizar la correspondencia entre éstos y los de la superficie. En la figura 6.23 se muestran las curvas de perfil asociadas a cada paso, escaladas convenientemente y orientadas según el plano normal de la curva de trayectoria en el punto de la misma de mínima distancia respecto del correspondiente punto de control. La curva de perfil original es una curva NURBS de grado 2 con 9 puntos de control (el primero y el último se solapan), sin embargo puede verse claramente que a medida que se calculan nuevas curvas, el número de puntos de control para las anteriores debe aumentar. El número de puntos de cada una se corresponde con el número de puntos de control de la superficie asociados a esa curva en concreto.

- La forma en que se realice el refinamiento de C^p puede provocar que la superficie obtenida tenga unas características o unas otras. Por ejemplo, si a la hora de insertar los nuevos nodos se opta por repartirlos por todo

⁴Gracias a este efecto *colateral* es por lo que esta técnica resulta útil en este caso.

el rango de valores sin tener en consideración la situación de los nodos ya existentes, entonces es posible que se obtenga un vector de nodos con algunos de ellos solapados. Como es sabido, dada una curva NURBS de grado q , dicha curva es continua C^q en los nodos de multiplicidad 1 y C^{q-k} en los nodos de multiplicidad k , la curva es infinitamente diferenciable en todo punto del espacio paramétrico que no coincida con un nodo. Por lo tanto si el refinamiento de nodos provoca el solapamiento de uno o más de ellos, en dichos nodos la continuidad de la superficie obtenida disminuirá, con lo que el resultado de la deformación obtenida puede mostrar efectos inesperados.

El último punto de control de la curva de trayectoria requiere un tratamiento diferente debido a que en principio se corresponde con el lugar en donde debe ir a parar el punto de la superficie original asociado a la restricción.

Considerando que en una superficie NURBS los puntos de control se encuentran en cualquier lugar del poliedro de control y considerando también la propiedad de envolvente convexa que tienen los poliedros de control ([PT97]), no es de esperar que, de forma natural, la superficie pase por un punto de control; a no ser que se trate de uno situado en los extremos del poliedro de control. Por lo tanto el lugar a donde debe ir a parar el último punto de control a deformar, en principio, no tiene por qué ser el lugar en dónde se encuentra el último punto de control de la curva de trayectoria.

Para conseguir ajustar la superficie, no parece adecuado un planteamiento parecido al propuesto para el caso de restricciones lineales aplicando (6.17) para determinar el desplazamiento de los puntos de control, toda vez que no es posible desplazar los puntos de control ya fijados sin que la superficie cambie de forma. Existen dos posibilidades para modificar solamente un punto de control y conseguir cambiar la forma de la superficie:

1. Modificar la ubicación del punto de control alejándolo siguiendo el sentido definido por la tangente de la curva de trayectoria en el último punto de la misma hasta que la superficie alcance el punto deseado. Con esta solución no se consigue un resultado adecuado, toda vez que los puntos de control ya calculados ejercen una influencia que no consigue eliminarse.
2. Modificar el peso del punto de control para hacer que su influencia sea mucho mayor que la del resto de puntos de control. Esta solución permite conseguir que la superficie pase por el lugar deseado, pero si la distancia entre el último punto de control y los anteriores es suficientemente grande entonces el último tramo de la deformación puede divergir de la forma de la curva de trayectoria. Esto se podría paliar refinando C^t de manera que se

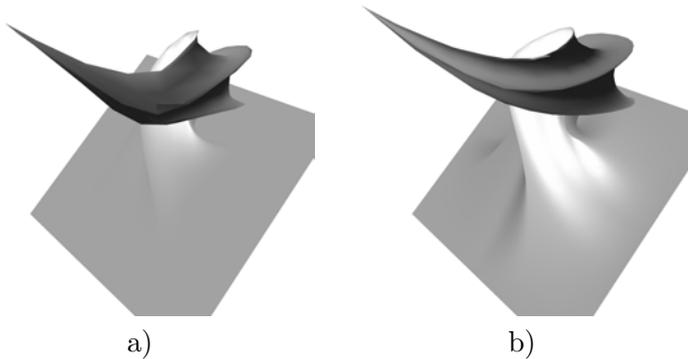


Figura 6.24: El extremo de una deformación puede mejorarse si se procede a añadir un punto de control próximo al extremo final de la curva de trayectoria. En a) se puede observar una deformación sin modificar la curva de trayectoria, en b) la misma deformación, pero añadiendo un punto de control adicional en el extremo de la curva de trayectoria. Puede observarse que la forma de b) es mucho mas suave.

añadiera un punto de control en un lugar próximo al último. Empíricamente, se ha estimado que, para obtener un buen resultado, el peso del último punto de control, w_{ij} , debe ser al menos igual a la suma de los pesos de los puntos de control que afectan al punto de aplicación de la restricción, $S(u, v)$:

$$(u, v) \in [U_{k-p}, U_k] \times [V_{l-q}, V_l]$$

$$w_{ij} \geq \sum_{k_1=k-p}^k \sum_{l_1=l-q}^l w_{k_1 l_1}$$

Aunque la segunda opción se ha mostrado como la más adecuada, no deja de ser una aproximación que puede dar un resultado más o menos satisfactorio en función de las características de la curva de trayectoria. Si la distancia paramétrica entre los dos últimos puntos de control de la curva de trayectoria es suficientemente grande, la forma resultante es muy *lineal* en el extremo, si la distancia entre los dos últimos puntos es pequeña este efecto se minimiza. Por ello una buena forma de suavizar el resultado final es añadir un punto de control adicional a la curva de trayectoria muy cerca de su extremo final. Para añadir un punto de control, ya se ha indicado que el método utilizado es el de refinamiento de nodos. En este caso se ha optado por añadir un nuevo nodo en una posición distante en el espacio paramétrico un 1% del extremo final de la curva de trayectoria. En la figura 6.24 se puede apreciar la diferencia en una deformación con y sin la adición de un punto de control al final.

Orientación de los puntos de la superficie

La cuestión de como ubicar los puntos de control de la superficie se resuelve determinando para cada punto de control de C^t , P_i^t , cual es su imagen paramétrica, u_m , que es el punto para el que $C^t(u_m)$ es el valor de C^t más próximo a P_i^t que se encuentra dentro del entorno local del punto de control. Así si la curva de trayectoria es de grado p se cumple:

$$\forall u \in [U_i^t \dots U_{i+p+1}^t) \quad p_i^t - C_t(u_m) < p_i^t - C_t(u), \quad u_m \in [U_i^t \dots U_{i+p+1}^t),$$

siendo U^t el vector de nodos correspondiente a la curva de trayectoria C^t .

Una vez conocido $C^t(u_m)$ es preciso conocer cuales son los vectores tangente, T^t , normal, N^t , y binormal, B^t , a dicho punto:

$$\begin{aligned} T^t &= \frac{C^{t'}(u_m)}{\|C^{t'}(u_m)\|}, \\ B^t &= \frac{C^{t'}(u_m) \times C^{t''}(u_m)}{\|C^{t'}(u_m) \times C^{t''}(u_m)\|}, \\ N^t &= B^t \times T^t, \end{aligned} \tag{6.31}$$

siendo $C^{t'}(u_m)$ y $C^{t''}(u_m)$ la primera y segunda derivadas de $C^t(u)$ evaluadas en el punto u_m . El plano normal es el definido por B^t y N^t y es ortogonal a T^t . Los vectores T^t , N^t y B^t forman una base ortonormal.

Para realizar un cambio de base y pasar del sistema de referencia a otro se pueden utilizar las expresiones (6.27) o (6.28). Además, si se desea orientar un conjunto de puntos según el plano definido por dos vectores, v y w , de forma que n sea el vector normal de la superficie será preciso multiplicar cada punto de los puntos por una *matriz de orientación* en \mathbb{R}^3 :

$$m_o(n, v, w) = \begin{pmatrix} n_x & v_x & w_x \\ n_y & v_y & w_y \\ n_z & v_z & w_z \end{pmatrix}^{-1}, \tag{6.32}$$

o para el caso de \mathbb{R}^4

$$m_o(n, v, w) = \begin{pmatrix} n_x & v_x & w_x & 0 \\ n_y & v_y & w_y & 0 \\ n_z & v_z & w_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1}, \tag{6.33}$$

que tienen la misma forma que (6.27) y (6.28). Por lo tanto a la hora de crear la matriz de cambio de base, escogiendo el orden en que se colocarán los valores de los vectores tangente, normal y binormal se puede conseguir el resultado deseado.

Para situar la curva de perfil, C_p sobre una superficie, S , centrada en un punto $p = S(u, v)$ será preciso determinar el plano rectificante y el vector normal a la superficie en dicho punto:

$$\begin{aligned} T_s &= \frac{S_u(u, v)}{\|S_u(u, v)\|} \\ B_s &= \frac{S_v(u, v)}{\|S_v(u, v)\|} \\ N_s &= B_s \times T_s \end{aligned} \tag{6.34}$$

siendo $S_u(u, v)$ y $S_v(u, v)$ las derivadas parciales de S respecto de los parámetros u y v respectivamente evaluadas en el punto p . Los vectores T_s y B_s determinan el plano rectificante y N_s el vector normal. El baricentro de los puntos de control se trasladará a p y cada uno de los puntos de control se trasladará y rotará en consecuencia.

Así pues, la curva de perfil \tilde{C}_p , orientada y trasladada al punto p , será obtenida modificando los puntos de control de C_p :

$$\tilde{P}_i = m_o(N_s, B_s, T_s)(P_i - p_c) + p, \tag{6.35}$$

siendo

$$p_c = \frac{1}{n+1} \sum_{i=0}^n P_i$$

el baricentro de los puntos de control.

El problema de las rectas

Uno de los casos particulares que se pueden dar con relativa facilidad es el de que la curva de trayectoria sea realmente una recta, bien sea en toda su extensión o bien sea solo en una parte de la misma, en cuyo caso tres o más puntos de control consecutivos son colineales. El problema surge a la hora de determinar cual es el plano normal en esos puntos de control, ya que el producto vectorial de la primera y segunda derivadas es nulo y por lo tanto no es posible determinar cuales son los vectores normal y binormal a partir de las ecuaciones (6.34) directamente. En la figura 6.25 se ilustra el problema así como la solución planteada.

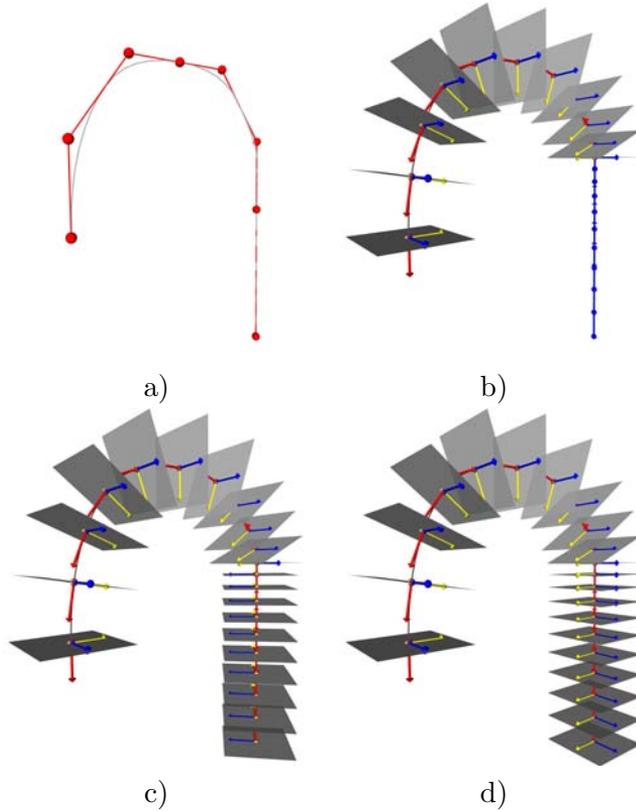


Figura 6.25: El problema de las tangentes. Cuando una curva tiene tres o más puntos de control consecutivos y colineales, a), aparece un problema a la hora de calcular los vectores normal y binormal si se aplican directamente las expresiones (6.34), b). Una solución es determinar arbitrariamente un vector perpendicular al vector tangente, pero existe la dificultad añadida de la discrepancia de la orientación que surge respecto de otros segmentos de la curva, c). La solución escogida busca cual es el primer segmento con una curvatura adecuada para determinar cual es la dirección que deberán tomar los vectores normal y binormal, d).

Una primera solución, consiste en establecer que el vector binormal sea un vector cualquiera perpendicular al vector tangente. El problema radica en que los vectores normal y binormal obtenidos no tienen por qué tener una orientación compatible con la del resto de vectores obtenidos para la parte de la curva en donde se puedan calcular de forma directa ambos vectores, véase la figura 6.25.c. Una aproximación mucho mejor es la de buscar en la curva el primer punto de control no colineal y a partir de él determinar el primer punto de la curva en el que influye este punto de control, conocido dicho punto se procede a calcular el vector binormal correspondiente y con él se aproxima el vector binormal para los puntos problemáticos. Conocido el vector binormal se procede a determinar el vector normal como ya se ha indicado en (6.34), el resultado obtenido puede apreciarse en la figura 6.25.d.

6.3.5. Resultados

La aplicación de este modelo es factible para cualquier superficie y cualesquiera curvas de trayectoria y de perfil. En la figura 6.27 se muestran algunos resultados obtenidos utilizando diferentes curvas de trayectoria y de perfil.

Una vez descritos todos los aspectos que han sido tenidos en consideración cabe solamente pasar a analizar los resultados obtenidos. En la figura 6.26 se muestran dos imágenes: una superficie deformada y los puntos de control resultantes. Se puede observar claramente que la cantidad de puntos de control generados es considerable, la cantidad de puntos dependerá del número de puntos de control que tengan las curvas de trayectoria y de perfil.

La forma concreta que resulta de la aplicación del modelo depende en buena medida del método descrito para realizar la deformación. Quedan aspectos no contemplados que pueden dejarse perfectamente a la discreción del usuario que pueda utilizar el modelo. Entre ellos, el grado de ajuste de la zona deformada con la zona de influencia, como puede observarse en las imágenes mostradas, la influencia de los puntos de control que afecta a la zona a deformar tienen también cierta influencia en la zona que queda fuera de dicha zona. Para evitar este efecto se puede ofrecer al usuario la opción de modificar la cardinalidad de los puntos limítrofes de manera que la influencia externa de los puntos de control quede truncada.

Según sea la curva de trayectoria es posible que la deformación de la superficie se introduzca en la propia superficie. Ello es debido a que la curvatura y la torsión de la curva de trayectoria así como su posición respecto de la superficie a deformar hagan que la forma en que se deforma la superficie se introduzca en sí misma, para evitar el efecto se puede optar por no orientar la primera de las curvas de

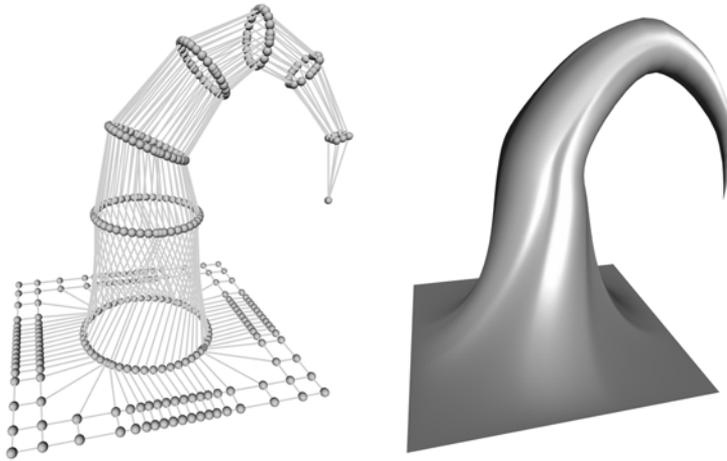


Figura 6.26: El poliedro de control de una superficie deformada puede tener un número considerable de puntos de control, según sean las curvas de superficie y de control.

interpolación, \tilde{C}_0^p . Si la primera de las curvas de interpolación se sitúa en su lugar, pero se orienta según el plano rectificante de la superficie en lugar de hacerlo con el plano normal de la curva de trayectoria, se consigue que los puntos control correspondientes queden sobre la superficie.

En definitiva es posible utilizar el modelo aplicándole a conveniencia los matices que se puedan juzgar necesarios para ajustar de forma precisa la manera en que se producirá la deformación.

6.4. Conclusiones

En este capítulo se ha procedido al estudio de la adecuación del modelo de representación BSpline4D para la deformación de objetos basada en restricciones. Para ello se ha realizado un estudio detallado del modelo original y se ha adaptado el mismo para su aplicación al modelo BSpline4D. Además se ha extendido el modelo de deformaciones para que sea posible plantear la deformación de superficies basada en restricciones no lineales.

Como todo modelo geométrico la aplicación del mismo resulta sencilla a la par que computacionalmente muy eficiente. El modelo es estático, aunque es posible plantear una animación en base a realizar interpolaciones de la deformación en diferentes estadios. El problema del ajuste de las curvas para obtener mejores

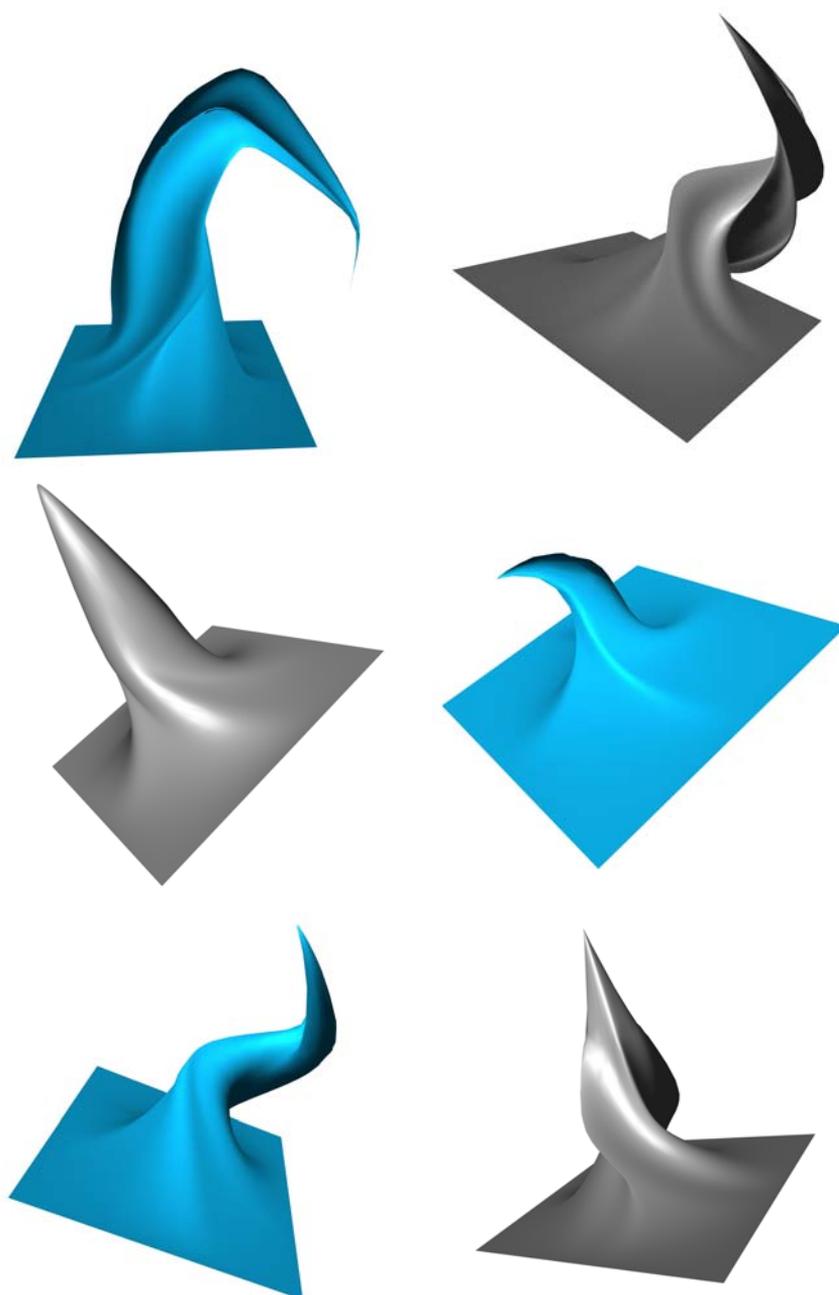


Figura 6.27: El modelo planteado no impone ninguna restricción en la forma que puedan tener las curvas utilizadas.

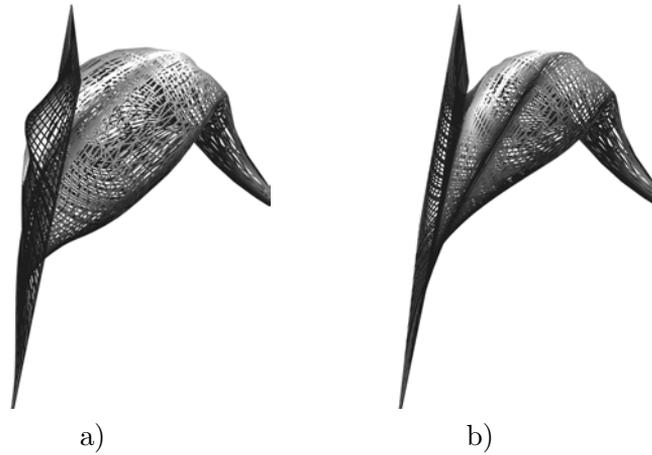


Figura 6.28: Según sea la curva de trayectoria es posible que la deformación de la superficie se introduzca en la propia superficie. En a) no se establece ningún control, en b) la primera curva de interpolación no se ha orientado según la curva de trayectoria sino que se orienta según el plano rectificante de la propia superficie.

resultados, o, en todo caso, los resultados que el usuario desee, es considerablemente menos complejo que el que pueden presentar otros modelos que pertenecen a la misma categoría de deformaciones libres de forma.

Se puede concluir por tanto que la utilización de BSpline4D conjuntamente con el modelo de deformaciones basadas en restricciones no lineales es una alternativa viable para la representación de objetos deformables.

Conclusiones

Acabada la tarea de proponer ideas, conjeturar hipótesis, desarrollar conceptos, realizar demostraciones y analizar resultados, llega el momento de realizar un repaso al trabajo realizado.

Una vez establecido el modelo físico que permite realizar la animación y simulación de objetos deformables mediante la ecuación (2.17), se ha desarrollado un estudio de los métodos de integración aplicables. De resultados de este estudio se ha desarrollado una metodología de validación aplicable de forma general a todo modelo de deformación que, usando la ecuación (2.17) en combinación con un funcional de energía, que describa la forma en que se puede deformar la superficie, pueda ser descrito mediante una ecuación de la forma (2.22).

Tras el estudio de los métodos de integración y conocida su adecuación se ha procedido a examinar las diferentes formas de representación de objetos deformables. Del estudio se ha concluido que los métodos basados en formas parametrizadas son los que ofrecen mejores cualidades de cara a poder desarrollar un modelo de representación de superficies deformables.

De las diferentes representaciones parametrizadas, las formas NURBS son las más versátiles. Sin embargo, el hecho de que sean formas racionales añade una componente de complejidad a la hora de realizar los cálculos que sería deseable poder evitar.

Por este motivo se ha propuesto el uso de un modelo alternativo interpretando las formas NURBS mediante una representación B-Spline en 4 dimensiones. El modelo, al que se ha denominado BSpline4D, ha demostrado tener las mismas capacidades de representación que un modelo basado en NURBS. Además, la adecuación de la ecuación (2.17) al nuevo modelo ha resultado tener unas propiedades numéricas mucho mejores. Para poder realizar la simulación dinámica se ha desarrollado la ecuación (2.17) para poder ser aplicada con los puntos de control de las superficies BSpline4D considerados como coordenadas generalizadas. La

estabilidad numérica de la ecuación obtenida, (5.30), ha resultado ser excelente. Además, la eficiencia computacional del modelo presentado es mejor que la de los otros modelos estudiados.

Del estudio realizado en el capítulo 3 y en el capítulo 5 se infiere que, para resolver la ecuación del movimiento y deformación de una superficie parametrizada, en combinación con un funcional de energía que mida la deformación de la superficie, de entre los métodos numéricos estudiados los mejores son el método de diferencias centradas implícito y el método de Newmark β .

Evidentemente, el ámbito de aplicación del modelo BSpline4D no se limita a la representación de superficies deformables mediante la aplicación de la ecuación (5.30). El modelo es perfectamente aplicable a cualquier ámbito. En particular se ha realizado una adaptación del modelo de deformaciones simples basadas en restricciones geométricas, *Scodef*, para que dicho modelo sea aplicable sobre una representación BSpline4D, a dicho modelo se le ha denominado N-Scodef. También se ha realizado una extensión del modelo para poder realizar deformaciones basadas en restricciones que siguen una trayectoria arbitraria, lo que constituye una aportación completamente original. Se han resuelto diversos problemas relacionados con la interpretación BSpline4D y, en particular, se han propuesto soluciones en el caso en que la deformación de la superficie siga una trayectoria descrita mediante una curva B-Spline. Se ha probado la conveniencia de alguna de ellas y para otras se ha dejado como una posibilidad de trabajo futuro. El objetivo era que la superficie se adaptase lo mejor posible a la forma de la curva, y se ha alcanzado de forma muy satisfactoria.

En definitiva, se ha descrito un modelo de representación que puede ser considerado como una mejora frente a los otros modelos que actualmente están siendo utilizados.

El presente trabajo es original; es decir, no ha sido publicado en ningún ámbito con anterioridad. Sin embargo, se han publicado algunos resultados preliminares. Por supuesto dichos trabajos han sido sometidos a la evaluación y crítica de los diferentes revisores, y el hecho de que hayan sido publicados es un primer indicio de la calidad del trabajo realizado.

Las publicaciones son las siguientes:

- [PMG00]: Este trabajo describe una primera aproximación al estudio de los métodos de integración; en él ya se perfilan algunos de los resultados a los que se ha llegado en el capítulo 3.
- [GMM⁺01]: Este artículo contiene parte de los primeros desarrollos realizados.

-
- [CGMP08]: Se muestran aquí las primeras aproximaciones a las deformaciones N-Scodef.
 - [PMGH13]: Es un capítulo de un libro, es básicamente el estudio de los diferentes modelos de representación detallados en el capítulo 4.

El objetivo principal de toda investigación científica es la de profundizar en el conocimiento dentro del ámbito en el que se realiza. Pero tal esfuerzo puede resultar inútil si no se puede sacar provecho de él. Si nadie aprovecha la investigación realizada para sustentar su propia actividad investigadora, entonces las aportaciones que se hayan podido realizar no habrán servido para nada.

En este sentido debe señalarse que las publicaciones antes mencionadas han sido citadas por terceras personas como justificación y fundamento de sus propios trabajos. Cabe destacar que se citarán a continuación exclusivamente aquellos trabajos en los que no existe ningún tipo de relación personal con los autores. Por lo tanto se puede inferir que sus citas se basan exclusivamente en el provecho que han podido obtener en base al contenido de las publicaciones.

- Publicaciones que citan a [PMG00]
 - [Zar03]: Es una tesis doctoral que trata acerca de la aplicación de algoritmos paralelos para el cálculo y representación de telas basándose en modelos físicos, utilizando modelos basados en sistemas de partículas y masa-resorte. Utilizan [PMG00] como referencia a la posibilidad de utilizar otros medios de representación, en este caso continuos.
 - [Dob05]: Otra tesis doctoral, esta busca la representación de tejidos blandos y su animación realista para representar cuerpos humanos en movimiento. Plantean un modelo basado en diferencias finitas y utilizan [PMG00] para justificar la elección de los operadores en diferencias. Utilizan nuevamente al artículo para confirmar el hecho del mal condicionamiento del modelo.
 - [DF06]: Publicación derivada del trabajo anterior, utilizan los mismos argumentos
 - [HGB11]: Este trabajo propone un modelo para la representación de telas, utilizando como funcional una expresión equivalente a (3.17). Utilizan [PMG00] citando las conclusiones principales a las que se llega en el artículo.
- Publicaciones que citan a [CGMP08]
 - [CSJ10]: En este trabajo los autores proponen un modelo de representación basado en *metaballs* con restricciones hápticas. En este trabajo se cita a [CGMP08] como referencia a la extensión del modelo *Scodef* a superficies NURBS.

El haber podido aportar algo de conocimiento al trabajo de investigación de otros, incluso de algunas tesis doctorales, es un motivo más para poder determinar la calidad del trabajo realizado.

Pero no se trata solamente de determinar qué es lo que se ha hecho, ni siquiera si está bien hecho, tampoco quién lo está utilizando. También es importante poder determinar qué líneas de trabajo se pueden seguir a partir de lo que hasta el momento se ha conseguido.

El modelo BSpline4D desarrollado puede ser extendido y ampliado en múltiples facetas, a continuación se mencionan aquéllas que se han planteado hasta el momento.

- Desarrollar el modelo de simulación de deformaciones para poder ser utilizado en entornos de computación de alto rendimiento. Mediante la utilización de las cada vez mayores facilidades para la supercomputación, es posible poder abordar el problema de la simulación de formas de gran complejidad. Aunque el rendimiento del modelo BSpline4D es muy bueno, su eficiencia asintótica sigue marcada por el número de coordenadas generalizadas que deben ser utilizadas para representar las superficies. Por ello, si la complejidad en la forma de dichas superficies exige un elevado número de puntos de control, necesariamente el tiempo de cálculo para realizar la simulación será proporcionalmente elevado. Replantear el algoritmo para poder utilizar algún tipo de multiprocesamiento, utilizando procesadores con múltiples núcleos, basándose en GPU o utilizando las nuevas tendencias en computación, *cloud computing*, *cluster computing*; será la forma de poder minimizar el coste en tiempo. Existen varios trabajos en esta línea, [Mac97, Zar03, YY09, SIGGMT11] y muchos otros, que habrá que evaluar.
- De cara a reducir el número puntos de control para el desarrollo de Scodef basado en BSpline4D sería interesante intentar adaptar el modelo a otros conceptos, como por ejemplo T-Splines, [SZBN03, NTNBR11], de manera que el número de puntos de control que forman el poliedro de control de la superficie resultante no sean tan elevado.
- Los modelos de representación de la energía de la deformación, bien sea el basado en la Geometría Diferencial, (3.17), o el modelo *thin plate under tension*, (5.14), no son las únicas opciones posibles. Una línea de trabajo en la que, de hecho, ya se está trabajando es la de la aplicación de otros funcionales de energía al modelo BSpline4D para poder simular otros comportamientos [Cia80, Gre94].
- El desarrollo formal del modelo BSpline4D puede ser dado a conocer mediante la publicación de trabajos de investigación, pero además sería muy interesante buscar una forma alternativa de difusión del modelo que per-

mitiera poner al alcance de más gente los resultados obtenidos. A tal fin, sería de gran interés poder desarrollar módulos que implementen el modelo y que puedan ser integrados en herramientas de modelización ya existentes.

En resumen, el modelo BSpline4D es una solución muy interesante y cabe esperar grandes resultados de su aplicación. Es hora de seguir desarrollando el concepto.

Bibliografía

- [AB76] G.J. Agin and T.O. Binford. Computer description of curved objects. *Computers, IEEE Transactions on*, C-25(4):439–449, april 1976.
- [Act01] S.T. Acton. Fast algorithms for area morphology. *Digital Signal Processing*, 11(3):187–203, 2001.
- [Agi72] G. J. Agin. *Representation and Description of Curved Objects*. PhD thesis, Stanford University of California, Department of Computer Science, 1972.
- [AGT93] L. Álvarez, F. González, and A. Trujillo. Deformación de superficies usando ecuaciones parciales. *III Congreso Español de Informática Gráfica*, pages 1–15, 1993.
- [AM00a] S.T. Acton and D.P. Mukherjee. Area operators for edge detection. *Pattern Recognition Letters*, 21(8):771–777, 2000.
- [AM00b] S.T. Acton and D.P. Mukherjee. Image edges from area morphology. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 6, pages 2239–2242, 2000.
- [AM00c] S.T. Acton and D.P. Mukherjee. Scale space classification using area morphology. *IEEE Transactions on Image Processing*, 9(4):623–635, apr 2000.
- [AS10] J. Abhau and O. Scherzer. A combinatorial method for topology adaptations in 3d deformable models. *International Journal of Computer Vision*, 87:304–315, 2010. 10.1007/s11263-009-0282-5.

- [AWO⁺10] B. Adams, M. Wicke, M. Ovsjanikov, M. Wand, H.-P. Seidel, and L. J. Guibas. Meshless shape and motion design for multiple deformable object. *Computer Graphics Forum*, 29(1):43–59, 2010.
- [Bar81] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
- [Bar84] A. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3):21–30, 1984.
- [BARB01] D. Brujic, I. Ainsworth, M. Ristic, and V. Brujic. Efficient shape description using NURBS. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *IWVF-4: Proceedings of the 4th International Workshop on Visual Form*, volume 2059, pages 643–653, London, UK, 2001. Springer-Verlag.
- [Bat82] K.-J. Bathe. *Finite element procedures in engineering analysis*. Prentice-Hall Engineering and Engineering Mechanics Series. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [Bat96] K.-J. Bathe. *Finite element procedures*. Prentice-Hall, New Jersey, 1996.
- [BB91] P. Borrel and D. Bechmann. Deformation of n-dimensional objects. In *SMA '91: Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications*, pages 351–369, New York, NY, USA, 1991. ACM.
- [BCA95] E. Bardinet, L. D. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. Technical Report RR-2617, INRIA Institut National de Recherche en Informatique et en Automatique, 1995.
- [BCA98] E. Bardinet, L. D. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding*, 71(1):39–54, July 1998.
- [BF10] R. L. Burden and J. D. Faires. *Numerical Analysis: 9th Ed.* Cengage Learning, 2010.
- [BG03] D. Bechmann and D. Gerber. Arbitrary shaped deformations with DOGME. *The Visual Computer*, 19:175–186, 2003. 10.1007/s00371-002-0191-x176.

-
- [BG04] D. Bechmann and D. Gerber. DogmeRV: A geometric deformation modeller for the workbench. In *6th Virtual Reality international conference IEEE-VRIC 2004*, pages 165–171, 2004.
- [BH12] M. Butenuth and C. Heipke. Network snakes: graph-based object delineation with active contour models. *Machine Vision and Applications*, 23:91–109, 2012. 10.1007/s00138-010-0294-8.
- [BHG91] D. Breen, D. House, and P. Getto. A particle-based computational model for cloth draping behaviour. In N. M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena*, pages 113–134. Springer-Verlag, 1991.
- [BHW94a] D. Breen, D. House, and M. J. Wozny. A particle-based model for simulating the draping behaviour of woven cloth. *Textile Research Journal*, 64(11):663–685, november 1994.
- [BHW94b] D. E. Breen, D. H. House, and M. J. Wozny. Predicting the drape of woven cloth using interacting particles. *Computer Graphics*, 28(Annual Conference Series):365–372, 1994.
- [Bjö96] A. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, SIAM, 1996.
- [Bli82] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, July 1982.
- [Blo88] J. Bloomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1988.
- [BN94] M. Bro-Nielsen. Active nets and cubes. Technical Report 94–13, Institute of Mathematical Modelling. Technical University of Denmark, 1994.
- [BN95] M. Bro-Nielsen. Modelling elasticity in solids using active cubes – application to simulated operations. In *Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine (CVRMed’95)*, volume 905 of *Lecture Notes in Computer Science*, pages 535–541. Springer Verlag, 1995.
- [BN04] A. Bouthors and F. Neyret. Modeling clouds shape. In M. Alexa and E. Galin, editors, *Eurographics ’04 (short papers)*, August 2004.
-

- [BNC96] M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum*, 15(3):57–66, 1996.
- [BR94] P. Borrel and A. Rappoport. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics*, 13(2):137–155, april 1994.
- [BS91] J. Bloomenthal and K. Shoemake. Convolution surfaces. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 251–256. ACM Press, 1991.
- [BS96] C. Blanc and C. Schlick. Accurate parametrization of conics by NURBS. *IEEE Computer Graphics and Applications*, 16(6):64–71, November 1996.
- [BS08] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, jan.–feb. 2008.
- [BVW10] G. Biegelbauer, M. Vincze, and W. Wohlkinger. Model-based 3D object detection. *Machine Vision and Applications*, 21:497–516, 2010. 10.1007/s00138-008-0178-3.
- [CAL⁺04] J.-H. Chuang, N. Ahuja, C.-C. Lin, C.-H. Tsai, and C.-H. Chen. A potential-based generalized cylinder representation. *Computers & Graphics*, 28(6):907–918, 2004.
- [CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978.
- [CC93] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, November 1993.
- [CC94] I. Cohen and L. D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. Technical Report 2188, INRIA. Institut Nationale de Recherche en Informatique et en Automatique, 1994.
- [CCI⁺07] M. Chen, C. Correa, S. Islam, M. W. Jones, P.-Y. Shen, D. Silver, S. J. Walton, and P. J. Willis. Manipulating, deforming and

- animating sampled object representations. *Computer Graphics Forum*, 26(4):824–852, 2007.
- [CD97] M.-P. Cani and M. Desbrun. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3(1), march 1997. Published under the name Marie–Paule Cani–Gascuel.
- [Cel90] G. Celniker. *Shape Write: Finite Element Based Free-Form Shape Design*. PhD thesis, Massachusetts Institute of Technology, 1990.
- [CF01] R. J. Campbell and P. J. Flynn. A survey of free–form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81:166–210, 2001.
- [CG91] G. Celniker and D. Gossard. Deformable curve and surface finite–elements for free–form shape design. *Computer Graphics – SIGGRAPH Proceedings 91*, 25(4):257–266, 1991.
- [CG11] S. Y. Chen and Q. Guan. Parametric shape representation by a deformable nurbs model for cardiac functional measurements. *IEEE Transactions on Biomedical Engineering*, 58(3):480–487, march 2011.
- [CGD97] M.-P. Cani-Gascuel and M. Desbrun. Animation of deformable models using implicit surfaces. *IEEE/Transactions on visualization and Computer Graphics*, 3(1), 1997.
- [CGMP08] M. Clapés, M. González, A. Mir, and P. A. Palmer. Interactive constrained deformations of NURBS surfaces: N–SCODEF. In F. J. Perales and R. B. Fisher, editors, *Articulated Motion and Deformable Objects*, number 5098 in Lecture Notes in Computer Science, pages 359–369. Springer Berlin / Heidelberg, 2008.
- [Cha74] G. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3(4):346–349, december 1974.
- [Cha04a] S.-Y. Chang. Studies of newmark method for solving nonlinear systems: (i) basic analysis. *Journal of the Chinese Institute of Engineers*, 27(5):651–662, 2004.

- [Cha04b] S.-Y. Chang. Studies of newmark method for solving nonlinear systems: (ii) verification and guideline. *Journal of the Chinese Institute of Engineers*, 27(5):663–675, 2004.
- [Cha09] S.-Y. Chang. Numerical characteristics of constant average acceleration method in solution of nonlinear systems. *Journal of the Chinese Institute of Engineers*, 32(4):519–529, 2009.
- [Cia80] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North–Holland Publishing Company, 1980.
- [CKO01] Y. Chikazawa, S. Koshizuka, and Y. Oka. A particle method for elastic and visco–plastic structures and fluid–structure interactions. *Computational Mechanics*, 27:97–106, 2001.
- [Coq90] S. Coquillart. Extended free–form deformation: a sculpturing tool for 3d geometric modeling. *SIGGRAPH Computer Graphics*, 24:187–196, September 1990.
- [CR94] Y.-K. Chang and A. P. Rockwood. A generalized de casteljau approach to 3d free–form deformation. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 257–260, New York, NY, USA, 1994. ACM.
- [CSJ10] H. Chen, H. Sun, and X. Jin. Haptic–constraint modeling based on interactive metaballs. *Computer Animation and Virtual Worlds*, 21(5):485–497, 2010.
- [CW92] G. Celniker and W. Welch. Linear constraints form deformable b–spline surfaces. In *Proceedings of the 1992 symposium on Interactive 3D graphics. March 29 – April 1, 1992, Cambridge, MA USA*, pages 165–170, 1992.
- [DB74] G. Dahlquist and A. Bjorck. *Numerical Methods*. Prentice Hall, 1974.
- [dC76] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice–Hall, traducción al castellano alianza editorial, 1994 edition, 1976.
- [Dec96] P. Decaudin. Geometric deformation by merging a 3D–object with a simple shape. *Graphics Interface '96*, pages 21–26, may 1996.

-
- [Del94] H. Delingette. Simplex meshes: a general representation for 3D shape reconstruction. Technical Report 2214, INRIA Institut National de Recherche en Informatique et en Automatique, 1994.
- [Del97] H. Delingette. General object reconstruction based on simplex meshes. Technical Report 3111, INRIA Institut National de Recherche en Informatique et en Automatique, 1997.
- [Del99] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):111–146, 1999.
- [DF06] M. Dobšík and M. Frydrych. Biomechanically based muscle model for dynamic computer animation. In *Computer Vision and Graphics*, volume 32 of *Computational Imaging and Vision*, pages 448–455. Springer Netherlands, 2006.
- [DGPR97] J. M. S. Dias, R. Galli, P. Palmer, and J. M. Rebordão. Deformable objects with real-time realistic behaviour for virtual scenarios. *The internet in 3D*. Academic Press, pages 179–199, 1997.
- [DHI91] H. Delingette, M. Hebert, and K. Ikeuchi. Deformable surfaces: A free-form shape representation. In *Geometric Methods in Computer Vision*, volume 1570, pages 21–30. SPIE, 1991.
- [DLG90] N. Dyn, D. Levine, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions Graphics*, 9(2):160–169, apr 1990.
- [Dob05] M. Dobšík. *Computer Animation of Soft Tissues*. PhD thesis, Brno University of Technology. Faculty of Information Technology, 2005.
- [DRBR09] J. Diener, M. Rodriguez, L. Baboud, and L. Reveret. Wind projection basis for real-time animation of trees. *Computer Graphics Forum*, 28(2):533 – 540, 2009.
- [DS78] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, 1978.
-

- [dVL09] F. de Vieilleville and J.-O. Lachaud. Digital deformable model simulating active contours. In Srecko Brlek, Christophe Reutenauer, and Xavier Provençal, editors, *Discrete Geometry for Computer Imagery*, volume 5810 of *Lecture Notes in Computer Science*, pages 203–216. Springer Berlin / Heidelberg, 2009.
- [EWS96] B. Eberhardt, A. Weber, and W. Strasser. A fast, flexible, particle–system model for cloth draping. *IEEE Computer Graphics and Applications*, 16(5):52–59, September 1996.
- [Far83] G. Farin. Algorithms for rational bézier curves. *Computer-Aided Design*, 15(2):73–77, March 1983.
- [Far92] G. Farin. From conics to nurbs: A tutorial and survey. *IEEE Computer Graphics and Applications*, 12(5):78–86, September 1992.
- [Far93] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 3 edition, 1993.
- [FB97] P. Fua and C. Brechbühler. Imposing hard constraints on deformable models through optimization in orthogonal subspaces. *Computer Vision and Image Understanding*, 65(2):148–162, February 1997.
- [Fey86] R. Feynman. Modeling the appearance of cloth. Master’s thesis, Department of Electrical Engineering and Computer Science, MIT, 1986.
- [FNMMG12] N. Fatou Ngom, O. Monga, M. M. O. Mohamed, and P. Garnier. 3d shape extraction segmentation and representation of soil microstructures using generalized cylinders. *Computers & Geosciences*, 39(0):50–63, 2012.
- [FvdPT97] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Dynamic free–form deformations for animation synthesis. *IEEE Transactions on Visualization & Computer Graphics*, 3(3):201–214, 1997.
- [GB08] J. Gain and D. Bechmann. A survey of spatial deformation from a user–centered perspective. *ACM Transactions on Graphics*, 27:1–21, November 2008.

-
- [GdJB94] J. García de Jalón and E Bayo. *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge*. Springer-Verlag, 1994.
- [GHJCMNB08] M. González-Hidalgo, A. Jaume Capó, A. Mir, and G. Nicolau-Bestard. Analytical simulation of B-spline surfaces deformation. In Francisco Perales and Robert Fisher, editors, *Articulated Motion and Deformable Objects*, volume 5098 of *Lecture Notes in Computer Science*, pages 338–348. Springer Berlin / Heidelberg, 2008.
- [GHJiCMNB10] M. González-Hidalgo, A. Jaume-i Capó, A. Mir, and G. Nicolau-Bestard. Analytical simulation of non-planar B-spline surfaces deformation. In Francisco Perales and Robert Fisher, editors, *Articulated Motion and Deformable Objects*, volume 6169 of *Lecture Notes in Computer Science*, pages 213–223. Springer Berlin / Heidelberg, 2010.
- [GHMN06] M. González-Hidalgo, A. Mir, and G. Nicolau. An evolution model of parametric surface deformation using finite elements based on B-splines. In *Proceedings of CompImage'2006 Conference, Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*, Coimbra, Portugal, 2006.
- [GJ10] F. J. Galdames and F. Jaillet. From triangulation to simplex mesh: a simple and efficient transformation. Technical Report RR-LIRIS-2010-021, LIRIS Laboratoire d'Informatique en Image et Systèmes d'information, UMR 5205 CNRS / INSA de Lyon / Université Claude Bernard Lyon 1 / Université Lumière Lyon 2 / École Centrale de Lyon, nov 2010.
- [GM97] S. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical Report MERL-TR-97-19, Mitsubishi Electric Research Laboratory, Cambridge, Massachusetts, USA, 1997.
- [GMM⁺01] M. González, M. Mascaró, A. Mir, P. Palmer, and F. Perales. Modeling and animating deformable objects. In J. S. Sánchez and F. Pla, editors, *Pattern Recognition and Image Analysis, Vol. I*, number 6 in *Treballs d'informàtica i tecnologia*, pages 279–290. AERFAI, Publicacions de la Universitat Jaume I, 2001.

- [GO94] U. Gdkbay and B. zgc. Animation of deformable models. *Computer-Aided Design*, 26(12):868–875, 1994.
- [Gol80] H. Goldstein. *Classical Mechanics*. Addison–Wesley, traduccin al castellano en ed. revert, 1988 edition, 1980.
- [GOT97] U. Gdkbay, B. zgc, and Y. Tokad. A spring force formulation for elastically deformable models. *Computer Graphics*, 21(3):335–346, 1997.
- [GPE⁺10] F.J. Galdames, C.A. Prez, P.A. Estvez, C.M. Held, F. Jaiilet, G. Lobo, G. Donoso, and C. Coll. Registration of renal SPECT and 2.5D US images. Technical Report RR–LIRIS–2010–022, LIRIS Laboratoire d’Informatique en Image et Systmes d’information, UMR 5205 CNRS / INSA de Lyon / Universit Claude Bernard Lyon 1 / Universit Lumiere Lyon 2 / cole Centrale de Lyon, nov 2010.
- [Gre94] G. Greiner. *Surface Construction Based on Variational Principles*, pages 277–286. Laurent, Le Mhaut & Schumaker (Eds.), 1994.
- [Han88] A. J. Hanson. Hyperquadrics: smoothly deformable shapes with convex polyhedral bounds. *Computer Vision, Graphics, and Image Processing*, 44(2):191–210, 1988.
- [HGB11] H. Hedfi, A. Ghith, and H. BelHadjSalah. Dynamic fabric modelling and simulation using deformable models. *Journal of the Textile Institute*, 102(8):647–667, 2011.
- [HHK92] W. M. Hsu, J. F. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. *Computer Graphics*, 26(2):177–184, 1992.
- [HLTC10] L. Heigas, A. Luciani, J. Thollot, and N. Castagn. A physically–based particle model of emergent crowd behaviors. *CoRR*, abs/1005.4405, 2010.
- [HWAG09] Q.-X. Huang, M. Wicke, B. Adams, and L. Guibas. Shape decomposition using modal analysis. *Computer Graphics Forum*, 28(2):407 – 416, 2009.
- [IF03] S. Ilic and P. Fua. From explicit to implicit surfaces for visualization, animation and modeling. In *ISPRS workshop on Visualization and Animation of Reality-based 3D Models*, 2003.

- [JL04] I.-K. Jeong and I. Lee. An oriented particle and generalized spring model for fast prototyping deformable objects. In *Proceedings of the 25th Annual Conference of the European Association for Computer Graphics*, volume 23. Eurographics, 2004.
- [JLP00] X. Jin, Y. Li, and Q. Peng. General constrained deformations based on generalized metaballs. *Computer & Graphics*, 24:200–0, 2000.
- [JLS00] A. Jaklic, A. Leonardis, and F. Solina. *Superquadrics and Their Geometric Properties*, volume 20 of *Computational Imaging and Vision*, chapter 2. Kluwer Academic Publishers, 2000.
- [JSV98] F. Jaillet, B. Shariat, and D. Vandorpe. Deformable object reconstruction with particle systems. *Computer & Graphics*, 22(2–3):189–194, 1998.
- [Juh99] I. Juhász. Weight-based shape modification of NURBS curves. *Computer Aided Geometric Design*, 16:377–383, 1999.
- [KML96] S. Kumar, D. Manocha, and A. Lastra. Interactive display of large NURBS models. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):323–336, December 1996.
- [Kob96] L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum*, 15(3):409–420, august 1996.
- [KP11] J. Kim and N. S. Pollard. Fast simulation of skeleton-driven deformable body characters. *ACM Transactions Graphics*, 30(5):121:1–121:19, October 2011.
- [KSS96] M. Kimura, T. Saito, and M. Shinya. Surface deformation with differential geometric structures. *Computer Aided Geometric Design*, 13:243–256, 1996.
- [KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal on Computer Vision*, 1(4):321–331, 1988.
- [LC87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169. ACM Press, 1987.

- [LC96] W. L. Luken and F. Cheng. Comparison of surface and derivative evaluation methods for the rendering of nurbs surfaces. *ACM Transactions on Graphics*, 15(2):153–178, April 1996.
- [LEB⁺10] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, and T.J.R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357–373, 2010. Computational Geometry and Analysis.
- [LG05] R. La Gréca. *Approche déclarative de la modélisation de surfaces*. PhD thesis, Université de la Méditerranée Aix–Marseille II, 2005.
- [LGRG07] R. La Gréca, R. Raffin, and G. Gesquière. Punctual constraint resolution and deformation path on NURBS. In *Graphicon'07. International Conference on Computer Graphics and Vision*, 2007.
- [LLC10] H. Li, W. K. Leow, and I.-S. Chiu. Elastic tubes: Modeling elastic deformation of hollow tubes. *Computer Graphics Forum*, 29(6):1770 – 1782, 2010.
- [LLS⁺11] Y.-L. Liao, C.-F. Lu, Y.-N. Sun, C.-T. Wu, J.-D. Lee, S.-T. Lee, and Y.-T. Wu. Three-dimensional reconstruction of cranial defect using active contour model and image registration. *Medical and Biological Engineering and Computing*, 49:203–211, 2011. 10.1007/s11517-010-0720-0.
- [Loo87] C. T. Loop. *Smooth Subdivision Surfaces Based on Triangles*. PhD thesis, University of Utah, 1987.
- [LPL07] S. Li, F. Pu, and D. Li. An improved edge detection algorithm based on area morphology and maximum entropy. In *Proceedings of the Second International Conference on Innovative Computing, Information and Control, ICICIC '07*, pages 536–539, Washington, DC, USA, 2007. IEEE Computer Society.
- [LRN06] S. Lanquetin, R. Raffin, and M. Neveu. Generalized SCODEF deformations on subdivision surfaces. In *Articulated motion and deformable objects*, LNCS 4069, pages 132–142, july 2006. proceedings of 4th International Conference AMDO Spain.

- [LSNC09] C. Loop, S. Schaefer, T. Ni, and I. Castaño. Approximating subdivision surfaces with gregory patches for hardware tessellation. In *ACM SIGGRAPH Asia 2009 papers*, SIGGRAPH Asia '09, pages 151:1–151:9, New York, NY, USA, 2009. ACM.
- [LW94] H. J. Lamousin and W. N. Waggenspack. NURBS-based free-form deformations. *IEEE Computer Graphics and Applications*, 14(6):59–65, november 1994.
- [Mac97] J. M. MacLaren. Parallelising serial codes: A comparison of three high performance parallel programming methods. Master Thesis. Department of Computer Science, University of Manchester., 1997.
- [Mas03] M. Mascaró. *Modelo de Simulación de Deformaciones de Objetos Basado en la Teoría de la Elasticidad*. PhD thesis, Departament de Ciències Matemàtiques i Informàtica. Universitat de les Illes Balears, 2003.
- [MBPD00] G. V. Moustakides, D. Briassoulis, E. Z. Psarakis, and E. Dimas. 3D image acquisition and NURBS based geometry modelling of natural objects. *Advances in Engineering Software*, 31(12):955–969, decembre 2000.
- [MC10] J Mille and L. Cohen. 3D CTA image segmentation with a generalized cylinder-based tree model. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro (IS-BI)*, pages 1045–1048, 2010.
- [McD01] J. McDonald. On flexible body approximations of rigid body dynamics. In V. Skala, editor, *WSCG 2001 Conference Proceedings*, 2001.
- [MDDB01] M. Meyer, G. DeBunne, M. Desbrun, and A. H. Barr. Interactive animation of cloth-like objects in virtual reality. *The Journal of Visualization and Computer Animation*, 12(1):1–12, 2001.
- [MDSA00] J. Montagnat, H. Delingette, N. Scapel, and N. Ayache. Representation, shape, topology and evolution of deformable surfaces. application to 3D medical image segmentation. Technical Report 3954, INRIA. Institut Nationale de Recherche en Informatique et en Automatique, 2000.

- [Met96] D. N. Metaxas. *Physics-Based Deformable Models. Applications to Computer Vision, Graphics and Medical Imaging*, volume 389 of *The Kluwer international Series in engineering and computer science*. Kluwer Academic Publishers, Boston, Hardbound, November 1996.
- [MMP97] M. Mascaró, A. Mir, and F. Perales. Modelado y animación de objetos deformables basado en la teoría de la elasticidad. *VII Congreso Español de Informática Gráfica*, pages 95–110, 1997.
- [MMP99] M. Mascaró, A. Mir, and F. Perales. Visualización de deformaciones dinámicas mediante elementos finitos triangulares y cuadrangulares. *IX Congreso Español de Informática Gráfica*, pages 47–61, 1999.
- [MMP00] M. Mascaró, A. Mir, and F. Perales. Elastic deformations using finite element methods in computer graphic applications. *AMDO'2000. First International Workshop on Articulated Motion and Deformable Objects. Lecture Notes in Computer Science, 1899*, pages 38–47, 2000.
- [MMP02] M. Mascaró, A. Mir, and F. Perales. P³DMA: A physical 3D deformable modelling and animation system. *AMDO'2002. Second International Workshop on Articulated Motion and Deformable Objects. Lecture Notes in Computer Science, 2492*, pages 68–79, 2002.
- [MMT97] L. Moccozet and N. Magnenat-Thalmann. Dirichlet free-form deformations and their application to hand simulation. In *Computer Animation '97*, pages 93–102, 1997.
- [MQ00] K. T. McDonnell and H. Qin. Dynamic sculpting and animation of free-form subdivision solids. *Proceedings of IEEE Computer Animation 2000*, pages 126–133, 2000.
- [MRA04] D.P. Mukherjee, N. Ray, and S.T. Acton. Level set analysis for leukocyte detection and tracking. *IEEE Transactions on Image Processing*, 13(4):562–572, april 2004.
- [MRYT08] J. G. Malcolm, Y. Rathi, A. Yezzi, and A. R. Tannenbaum. Fast approximate surface evolution in arbitrary dimension. In J. M. Reinhardt and J. P. W. Pluim, editors, *Medical Imaging 2008: Image Processing*, volume 6914. SPIE–The International Society

-
- for Optical Engineering, Georgia Institute of Technology, Society of Photo–Optical Instrumentation Engineers, 2008.
- [MT92] D. Metaxas and D. Terzopoulos. Dynamic deformation of solid primitives with constraints. *Computer Graphics*, 26(2):309–312, July 1992.
- [MT93] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics–based synthesis. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 15(6):580–591, June 1993.
- [MT96] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108, 1996.
- [MTBV09] N. Magnenat-Thalmann, U. Bonanni, and P. Volino. Physical behavior of deformable hair and clothes: What is common? In *CAD/Graphics*, pages 12–18. IEEE, 2009.
- [MTG04] M. Mueller, M. Teschner, and M. Gross. Physically–based simulation of objects represented by surface meshes. In *Proceedings of Computer Graphics International CGI’04*, pages 26–33, 2004.
- [NA93] C. Nastar and N. Ayache. Fast segmentation, tracking, and analysis of deformable objects. In *Proceedings of Fourth International Conference on Computer Vision, ICCV’93*, pages 275–279, 1993.
- [NCÇ10] S. Natsupakpong and M. Cenk Çavuşoğlu. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graph. Models*, 72(6):61–73, November 2010.
- [NG96] H. N. Ng and R. L. Grimsdale. Computer graphics techniques for modelling cloth. *IEEE Computer Graphics and Applications*, pages 28–41, 1996.
- [NHK⁺85] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Object modeling by distribution function and a method of image generation. *Transactions of the Institute of Electronics and Communication Engineers of Japan, (traducido al inglés por T. Fujiwara, Advanced Studies in Computer Aided Art and Design, Middlesex Polytechnic, England, 1989)*, J68-D(4):718–725, 1985.
-

- [NISA07] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Fibermesh: designing freeform surfaces with 3D curves. *ACM Trans. Graph.*, 26(3), July 2007.
- [NMK⁺06] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.
- [NTNXBR11] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordas, and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical t-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 200:1892–1908, 2011.
- [OS88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [Paz92] M. Paz. *Dinámica Estructural: Teoría y Cálculo*. Ed. Reverté, 3 edition, 1992.
- [PB88] J. C. Platt and A. H. Barr. Constraint methods for flexible models. In *Proceedings of the 15th annual conference on Computer graphics*, pages 279–288, Atlanta, Georgia, United States, august 1988.
- [PDA00] G. Picinbono, H. Delingette, and N. Ayache. Real-time large displacement elasticity for surgery simulation: Non-linear tensor-mass model. In *Third International Conference on Medical Robotics, Imaging And Computer Assisted Surgery: MICCAI 2000*, pages 643–652, october 2000.
- [Pie91] L. Piegl. On NURBS: a survey. *IEEE Computer Graphics and Applications*, 11(1):55–71, 1991.
- [PMG00] P. Palmer, A. Mir, and M. González. Stability and complexity study of animated elastically deformable objects. *AMDO'2000. First International Workshop on Articulated Motion and Deformable Objects. Lecture Notes in Computer Science, 1899*, pages 58–71, 2000.
- [PMGH13] P. Palmer, A. Mir, and M. González-Hidalgo. Deformable objects representation. In M. González Hidalgo et al., editor, *Deformation Models. Tracking, Animation and Applications*, volu-

-
- me 7 of *Lecture Notes in Computational Vision and Biomechanics*, pages 3–47. Springer Netherlands, 2013.
- [Pro95] X. Provot. Deformation constraints in a mass–spring model to describe rigid cloth behaviour. In Wayne A. Davis and Przemyslaw Prusinkiewicz, editors, *Graphics Interface '95*, pages 147–154. Canadian Human–Computer Communications Society, 1995.
- [PSDS09] R. Portal, L. Sousa, J. Dias, and N. Santos. Contact detection of convex superquadric using optimization techniques with graphical user interface. In *Proceedings of the 7th EUROMECH Solid Mechanics Conference (ESMC2009)*, 2009.
- [PT97] L. Piegl and W. Tiller. *The NURBS Book*. Monographs in visual communications. Springer–Verlag, 2 edition, 1997.
- [PW89] A. Pentland and J. Williams. Good vibrations: model dynamics for graphics and animation. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 215–222. ACM Press, 1989.
- [QT95] H. Qin and D. Terzopoulos. Dynamic NURBS swung surfaces for physics–based shape design. *Computer-aided Design*, 27(2):111–127, 1995.
- [QT96] H. Qin and D. Terzopoulos. D–NURBS: A physics–based framework for geometric design. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):85–96, March 1996.
- [QT97] H. Qin and D. Terzopoulos. Triangular NURBS and their dynamic generalizations. *Computer Aided Geometric Design*, 14(4):325–347, 1997.
- [RAH97] P. Radeva, A. A. Amini, and J. Huang. Deformable b–solids and implicit snakes for 3D localization and tracking of SPAMM MRI data. *Computer Vision and Image Understanding*, 66(2):163–178, May 1997.
- [RAH⁺06] D. Rueckert, P. Aljabar, R. Heckemann, J. Hajnal, and A. Hamers. Diffeomorphic registration using B–splines. In R. Larsen, M. Nielsen, and J. Sporring, editors, *Medical Image Computing and Computer–Assisted Intervention–MICCAI 2006*, volume 4191 of *Lecture Notes in Computer Science*, pages 702–709. Springer Berlin / Heidelberg, 2006.
-

- [RC02] I. Rudomín and J. L. Castillo. Real-time clothing: Geometry and physics, 2002.
- [RGLG07] R Raffin, G. Gesquière, and R. La Gréca. Déformations de modèles géométriques. Technical Report LSIS.RR.2007.001, LSIS, 2007.
- [RM93] D. Ruprecht and H. Müller. *Free Form Deformation with Scattered Data Interpolation Methods*, chapter (Computer Suppl. 8), pages 267–281. Springer-Verlag, 1993.
- [RND98] R. Raffin, M. Neveu, and B. Derdouri. Constrained deformation for geometric modeling and object reconstruction. In *WSCG 98 – International Conference in Central Europe on Computer Graphics, Visualization’98*, volume 2, pages 299–306, 1998.
- [RNJ99] R. Raffin, M. Neveu, and F. Jaar. Extended constrained deformations: a new sculpturing tool. *Shape Modeling and Applications, 1999. Proceedings. Shape Modeling International ’99. International Conference on*, pages 219–224, mar. 1999.
- [RNJ00] R. Raffin, M. Neveu, and F. Jaar. Curvilinear displacement of free-form-based deformation. *The Visual Computer*, 16:38–46, 2000. 10.1007/s003710050005.
- [RNM95] D. Ruprecht, R. Nagel, and H Müller. Spatial free-form deformation with scattered data interpolation methods. *Computers and Graphics*, 19(1):63–71, 1995.
- [Rud90] I. J. Rudomín. Simulating cloth using a mixed geometry-physical method. *Doctoral dissertation, Dep. of Computer and Information Science, Univ. of Pennsylvania*, 1990.
- [SBS96] F. J. Serón, J. Badal, and F. J. Sabadell. A numerical laboratory for simulation and visualization of seismic wavefields. *Geophysical Prospecting*, 44:603–642, 1996.
- [SD96] L. H. Staib and J. S. Duncan. Model-based deformable surface findind for medical images. *IEEE Transactions on Medical Imaging*, 15(5):720–731, October 1996.
- [Set87] J. A. Sethian. Numerical methods for propagating fronts. In P. Concus and R. Finn, editors, *Variational Methods for Free Surface Interfaces. Proceedings of the Sept, 1985 Vallambrosa Conference*, pages 155–164. Springer-Verlag, 1987.

-
- [Set99] J. A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, 1999.
- [Sha88] A. A. Shabana. *Dynamics of Multibody Systems*. John Wiley & Sons, 1 edition, 1988.
- [Sha13] A. A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, 4 edition, 2013.
- [She99] J. A. Shetian. *Level Set Methods and Fast Marching Methods*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 2nd edition, 1999.
- [She02] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22(1–3):21–74, 2002. 16th ACM Symposium on Computational Geometry.
- [She05] J. R. Shewchuk. Theoretically guaranteed delaunay mesh generation – in practice. In *14th International Meshing Roundtable*, volume Short Course, 2005.
- [SIGGMT11] J. Schmid, J. Iglesias Guitián, E. Gobbetti, and N. Magnenat-Thalmann. A GPU framework for parallel segmentation of volumetric images using discrete deformable models. *The Visual Computer*, 27:85–95, 2011. 10.1007/s00371-010-0532-0.
- [SLF08] A. Selle, M. Lentine, and R. Fedkiw. A mass spring model for hair simulation. *ACM Trans. Graph.*, 27(3):64:1–64:11, August 2008.
- [SLS⁺02] J. S. Suri, K. Liu, S. Singh, S. Laxminarayan, X. Zeng, and L. Reden. Shape recovery algorithms using level sets in 2-D/3-D medical imagery: a state-of-the-art review. *IEEE Transactions on Information Technology in Biomedicine*, pages 8–28, 2002.
- [SNM07] M. Silveira, J. Nascimento, and J. Marques. Automatic segmentation of the lungs using robust level sets. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4414–4417, aug. 2007.
- [SOG09] D. Steinemann, M. A. Otaduy, and M. Gross. Splitting meshless deforming objects with explicit surface tracking.
-

- Graphical Models*, 71(6):209–220, 2009. 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2006).
- [SP86] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160, 1986.
- [SS03] J. A. Sethian and P. Smereka. Level set methods for fluid interfaces. *Fluid Mechanics*, 35:341–372, 2003.
- [SS10] M. T. Sederberg and T. W. Sederber. T-splines: A technology for marine design with minimal control points. Technical report, Chesapeake Powerboat Symposium, 2010.
- [ST92] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. *Computer Graphics*, 26(2):185–194, 1992.
- [SZBN03] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. *ACM Transactions on Graphics*, 22(3):477–484, July 2003.
- [SZSS98] T. W. Sederberg, J. Zheng, D. Sewell, and M. Sabin. Non-uniform recursive subdivision surfaces. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 387–394, New York, NY, USA, 1998. ACM.
- [Ter11] D. Terzopoulos. Deformable and functional models. In João Manuel R. S. Tavares and R. M. Natal Jorge, editors, *Computational Vision and Medical Image Processing*, volume 19 of *Computational Methods in Applied Sciences*, pages 125–143. Springer Netherlands, 2011.
- [TF88a] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4:306–331, 1988.
- [TF88b] D. Terzopoulos and K. Fleischer. Modelling inelastic deformation: Viscoelasticity, plasticity, fracture. *Computer Graphics*, 21(4):269–278, july 1988.
- [THMG04] M. Teschner, B. Heidelberger, M. Mueller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Proceedings of Computer Graphics International CGI'04*, pages 312–319, 2004.

- [TM91] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.
- [TPBF87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (Proceedings SIGGRAPH)*, 21(4):205–214, 1987.
- [TQ94] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, 1994.
- [Tso91] N. Tsoelas. Animating the crumpling behaviour of garments. *Proc. 2nd Eurographics Workshop on Animation and Simulation, Blackwell, UK*, pages 11–24, 1991.
- [TV91] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 70–75, Maui, Hawaii, June 1991. IEEE Computer Society Press.
- [TW88] D. Terzopoulos and A. Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics & Applications*, pages 41–51, 1988.
- [TWK88] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.
- [Ver75] K. Versprille. *Computer-aided design applications of the rational b-spline approximation form*. PhD thesis, Syracuse University, 1975.
- [VT92] M. Vasilescu and D. Terzopoulos. Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pages 829–832, Champaign, IL, June 1992. IEEE Computer Society Press.
- [VW96] R. C. Veltkamp and W. Wesselink. Variational modeling of triangular bezier surfaces, 1996.

- [Wei86] J. Weil. The synthesis of cloth objects. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 49–54. ACM Press, 1986.
- [WFB87] A. Witkin, K. Fleischer, and A. H. Barr. Energy constraints on parameterized models. *SIGGRAPH Computer Graphics*, 21(4):225–232, aug 1987.
- [WMW86a] G. Wyvill, C. McPheeters, and B. Wyvill. Animating soft objects. *The Visual Computer*, 2(4):235–242, 1986.
- [WMW86b] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, 1986.
- [WTGT09] C. Wojtan, N. Thürey, M. Gross, and G. Turk. Deforming meshes that split and merge. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 76:1–76:10, New York, NY, USA, 2009. ACM.
- [WW92] W. Welch and A. Witkin. Variational surface modelling. *Computer Graphics, SIGGRAPH'92 Proceedings*, 26(2):157–166, 1992.
- [WXXL12] Y. Wang, Y. Xiong, K. Xu, and D. Liu. vKASS: a surgical procedure simulation system for arthroscopic anterior cruciate ligament reconstruction. *Computer Animation and Virtual Worlds*, 2012.
- [XY12] W. Xing and B. Yuan. 3D part based structural description extracting and modeling. In *Proceedings of the International Multiconference of Engineers and Computer Scientists, 2012*, IMECS'2012, 2012.
- [YXSN11] S. Y. Yeo, X. Xie, I. Sazonov, and P. Nithiarasu. Geometrically induced force interaction for three-dimensional deformable models. *IEEE Transactions on Image Processing*, 20(5):1373–1387, may 2011.
- [YY09] Y.-K. Yau, H.-T. Lin and C.-T. Yeh. A new approach to accelerate NURBS surface rendering on GPU. *Computer-Aided Design and Applications*, 6(4):529 – 538, 2009.
- [Zar03] F. Zara. *Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles*. PhD thesis, Institut National Polytechnique de Grenoble, 2003.

- [ZFT09] L. Zhu, B. Fan, and Y. Tang. Active contour method with separate global translation and local deformation. In Ming Xie, Youlun Xiong, Caihua Xiong, Honghai Liu, and Zhencheng Hu, editors, *Intelligent Robotics and Applications*, volume 5928 of *Lecture Notes in Computer Science*, pages 876–884. Springer Berlin / Heidelberg, 2009.
- [ZS00] D. Zorin and P. Schröder. Subdivision for modeling and animation. Technical report, SIGGRAPH 2000 – Course Notes, 2000.
- [ZYF⁺09] H. Zhang, L. Yang, D. J. Foran, J. L. Noshier, and P. J. Yim. 3D segmentation of the liver using free-form deformation based on boosting and deformation gradients. In *Proceedings of the Sixth IEEE international conference on Symposium on Biomedical Imaging: From Nano to Macro*, ISBI'09, pages 494–497, Piscataway, NJ, USA, 2009. IEEE Press.

Índice alfabético

- B-Spline, 133
 - función de base, 133
- blobs, 119
- cilindros generalizados, 110
- coeficiente
 - influencia, 84
- colisiones, 2
- configuración, 10
 - espacio de, 10
 - instantánea, 10
- conjunto de vértices, 101
- conjuntos de nivel, 122
- control
 - polígono de, 130, 131
 - poliedro de, 132
 - punto de, 131
 - imagen paramétrica, 215
- coordenadas generalizadas, 9
 - libres, 10
- coste computacional, 4, 87, 95, 96, 103, 107, 109, 123, 184, 188
- cuadratura de Gauss–Legendre, 158
- curva
 - de Bézier, 130
 - de perfil, 215
 - de trayectoria, 215
 - simple, 187
- de formas libres, 4, 189
- doblamiento, 84, 85
- elongación, 84, 85
- siguiendo trayectoria, 215
- siguiendo una trayectoria, 194
- Deformación temporal, 157
- deformaciones
 - restricciones geométricas, 191
- deformaciones basadas en restricciones, 188
- Delaunay
 - triangulación, 103
- descomposición modal, 126
- dinámica, 2
 - directa, 2
 - inversa, 2
 - simulación, 2
- discretización
 - condiciones de contorno
 - naturales, 38
 - regiones de, 38
 - diferencias finitas centradas, 37
 - espacial, 37, 157
 - influencia, 58
- DOGME, 191
- energía
 - cinética, 13
 - técnicas basadas en, 4
- envolvente de influencia, 207
- estabilidad, 40, 168

- número de condición, 48, 176
- radio espectral, 40, 48, 176
- FFD, 4, 189
- formas estrelladas, 194
- formulación híbrida, 113
- formulación primal, 112
- free-form deformation, 4, 189
- fuerzas
 - externa, 11
 - externas, 2
 - generalizadas, 12
 - ligadura, 11
 - técnicas basadas en, 4
- función
 - de base B-Spline, 133
 - de conectividad del mallado, 101
 - de deformación, 191
 - de disipación de Rayleigh, 17, 19
 - de potencial, 15, 119
 - de restricción de localización, 212
 - lagrangiana, 14
 - polinomial a trozos, 133
- generación de transiciones, 4
- geometría diferencial
 - primera forma fundamental, 34
 - segunda forma fundamental, 34
- geones, 118
 - supercuádricos, 118
- grado de libertad, 10
- Hamilton
 - principio de, 15
- integración
 - esquemas, 23
 - condicionalmente estables, 23
 - incondicionalmente estables, 25
- integral de acción, 15
- isosuperficies, 119
- límite de estabilidad, 30
- Lagrange
 - ecuaciones, 12, 14
 - multiplicadores de, 163
- level sets, 122
- ligadura
 - esclerónoma, 9
 - fuerzas, 11
 - holónoma, 9
 - no holónoma, 9
 - reónoma, 9
- métodos de integración
 - aceleración lineal, 28
 - aceleración media, 28
 - diferencias centradas, 22
 - diferencias centradas implícito, 23
 - directa, 21
 - estabilidad, 30
 - explícitos, 23
 - grado de exactitud, 32
 - Houbolt, 24
 - implícitos, 23
 - Newmark β , 27
 - precisión, 30
 - trapezoidal, 28
 - Wilson θ , 25
- mallado
 - 2-simple, 104
 - k -simple, 105
 - adaptativo, 108
 - diagramas de Voronoi, 103
 - poligonal, 100
 - conjunto de aristas, 102
 - conjunto de polígonos, 102
 - mallados cúbicos, 102
 - triangulación, 103
 - simple, 104
 - triangulación
 - Delaunay, 103
- marching cubes

-
- algoritmo, 121
 - matriz
 - de orientación, 231
 - de proyección, 191
 - mecánica, 2, 7
 - membrana elástica, 177
 - metaballs, 119
 - modelo
 - computacional, 4
 - de contornos activos, 111
 - basado en grafos, 112
 - de objetos continuos, 112
 - de subdivisión de Catmull–Clark, 141
 - discreto, 100
 - explícito, 109
 - gráfico, 4
 - implícito, 114
 - masa–resorte, 107
 - paramétrico, 124
 - representación de objetos por muestreo, 124
 - morphing, 4
 - N–Scodef, 196
 - nD–deformations, 191
 - Newton
 - segunda ley, 10
 - Non–Uniform Rational B-Spline, 137
 - NURBS, 136
 - partículas orientadas, 106
 - PB–Splines, 139
 - primera derivada variacional, 35
 - producto esférico, 115
 - radial deformation, 200
 - radio de influencia, 193
 - Rayleigh
 - función de disipación de, 17, 19
 - resistencia, 34
 - deformación longitudinal, 34
 - deformación por cizalladura, 34
 - deformación por doblamiento, 34
 - deformación por retorcimiento, 34
 - doblamiento, 177
 - elongación, 177
 - resortes
 - de cizalladura, 107
 - estructurales, 107
 - flexurales, 107
 - restricciones
 - geométricas, 162
 - imposición de, 162
 - interacción, 203
 - método de penalización, 164
 - método de proyección en el espacio nulo, 164
 - multiplicadores de Lagrange, 163
 - sampled object representation, 124
 - scodef, 191
 - simple constrained deformations, 191
 - simulación, 2
 - simuladores dinámicos, 2
 - simular, 2
 - sistema
 - conservativo, 14, 15
 - de partículas, 105
 - no conservativo, 16
 - snake, 111
 - network snakes, 112
 - soft objects, 120
 - supercuádricas
 - como superficies algebraicas, 114, 115
 - como superficies paramétricas, 126
 - superficies
 - algebraicas, 114
 - supercuádricas, 114, 115
 - superelipsoides, 115
 - de barrido, 218
 - de subdivisión, 127
 - hipercuádricas, 118
-

- paramétricas
 - supercuádricas, 126
- surfaces
 - swept, 110, 218
 - swung, 110
- T-junctions, 139
- T-Splines, 139
- T-junction, 140
- T-mesh, 140
- T-NURCCS, 141
- técnicas
 - geométricas, 3
 - híbridas, 4
- tensor
 - métrico, 34
 - de curvatura, 34
- thin plate under tension, 149, 176, 180,
182
- virtual
 - desplazamiento, 11
 - trabajo, 11
- Voronoi
 - diagramas, 103
- zona de influencia, 193
 - paramétrica, 215

