



Universitat d'Alacant
Universidad de Alicante

Esta tesis doctoral contiene un índice que enlaza a cada uno de los capítulos de la misma.

Existen asimismo botones de retorno al índice al principio y final de cada uno de los capítulos.

[Ir directamente al índice](#)

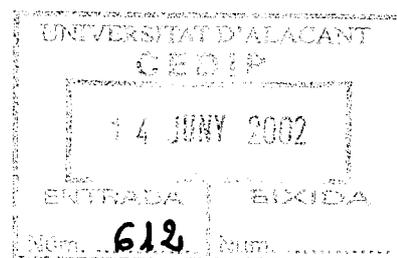
Para una correcta visualización del texto es necesaria la versión de [Adobe Acrobat Reader 7.0](#) o posteriores

Aquesta tesi doctoral conté un índex que enllaça a cadascun dels capítols. Existeixen així mateix botons de retorn a l'índex al principi i final de cadascun dels capítols .

[Anar directament a l'índex](#)

Per a una correcta visualització del text és necessària la versió d' [Adobe Acrobat Reader 7.0](#) o posteriors.

Universidad de Alicante
*Dpto. de Ciencia de la Computación
e Inteligencia Artificial*



Tesis Doctoral

Modelo Superficie-Trayectoria.

Un modelo geométrico para el diseño y la fabricación de objetos tridimensionales

Autor

Rafael Molina Carmona

Director

Dr. Ramón Rizo Aldeguer

Julio de 2002





Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

A mi padre



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

Agradecimientos

A Ramón Rizo, por su apoyo y sus consejos y por ser mucho más que un director de tesis; a mis compañeros del grupo VGIA y del departamento, y en especial a Juan Antonio Puchol y Juan Manuel Sáez, por su ayuda y su amistad; a INESCOP y los que forman parte de esta organización, por introducirme en el mundo del diseño y la fabricación asistidos por ordenador; en fin, gracias a todos.



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

ÍNDICES

ÍNDICE GENERAL

	Agradecimientos	5
	Índice general.....	8
	Índice de figuras	13
	Índice de tablas	17
Capítulo 1	Introducción.....	19
	1.1 Los sistemas CAD/CAM.....	20
	1.2 Los modelos geométricos.....	20
	1.3 El modelo Superficie-Trayectoria	21
	1.4 Estructura del documento	21
Capítulo 2	Motivación y objetivos.....	23
	2.1 Motivación	24
	2.1.1 La industria y el CAD/CAM	24
	2.1.2 Limitaciones de los modelos actuales	24
	2.2 Objetivos	26
	2.2.1 Objetivos para facilitar el diseño	26
	2.2.2 Objetivos para mejorar la fabricación	26
	2.2.3 Objetivos para disminuir los costes.....	27
Capítulo 3	Estado del arte	29
	3.1 Modelos Geométricos.....	30
	3.1.1 Introducción	30
	3.1.2 Taxonomía de los modelos geométricos.....	31
	3.1.3 Modelo alámbrico	31
	3.1.4 Modelos superficiales.....	32
	3.1.4.1 Modelo poligonal	33
	3.1.4.2 Modelo de superficies curvas.....	34
	3.1.4.3 Otros modelos superficiales	35
	3.1.5 Modelos sólidos	36
	3.1.5.1 Sólidos generados por barrido	36
	3.1.5.2 Geometría sólida constructiva (CSG).....	37
	3.1.5.3 Enumeración de la ocupación espacial	38
	3.1.6 Otros modelos	40
	3.1.6.1 Modelo de geometría fractal.....	40
	3.1.6.2 Modelo gramatical	41
	3.1.6.3 Sistemas de partículas	42
	3.1.6.4 Modelos basados en características físicas	43
	3.2 Mecanizado.....	44
	3.2.1 Introducción	44
	3.2.2 Tipos de máquinas y herramientas	45
	3.2.3 El problema del mecanizado	46
	3.2.4 Estrategias de mecanizado	48

3.2.5	Cálculo de trayectorias	51
3.2.5.1	Mecanizado a altura constante	52
3.2.5.2	Mecanizado a altura variable.....	53
Capítulo 4	El modelo Superficie-Trayectoria	57
4.1	Introducción	58
4.2	Fundamentos del modelo.....	59
4.2.1	Superficies	59
4.2.2	Trayectorias.....	59
4.2.3	Herramientas virtuales.....	60
4.2.4	Características del modelo	61
4.3	Definición formal del modelo.....	63
4.3.1	Sintaxis.....	63
4.3.2	Semántica	64
Capítulo 5	Constructores de curvas y superficies	67
5.1	Introducción	68
5.2	Curvas y superficies paramétricas	68
5.2.1	Decisiones de diseño sobre el modelo	69
5.2.2	Formulación de curvas y superficies Spline	69
5.2.2.1	Curvas y superficies Bézier	71
5.2.2.2	Curvas y superficies BSpline	73
5.2.2.3	Conversión entre representaciones	75
5.3	Constructores de superficies.....	76
5.3.1	Extrusión de curvas planas.....	76
5.3.2	Revolución de una curva alrededor de un eje.....	76
5.3.3	Cruce de curvas planas	77
5.3.4	Definición por secciones.....	78
5.4	Constructores de curvas sobre superficies.....	79
5.4.1	Discretización de curvas	79
5.4.2	Proyección de curvas sobre superficies	80
5.4.2.1	La transformación del espacio (x,y,z) al t	82
5.4.2.2	La transformación del espacio (x,y,z) al (u,v)	82
5.4.2.3	Proyección de una curva sobre una superficie.....	83
5.4.3	Despliegue de curvas sobre superficies	85
Capítulo 6	Constructores de trayectorias	87
6.1	Introducción	88
6.2	Cálculo de trayectorias de mecanizado.....	88
6.2.1	Parámetros de mecanizado	88
6.2.2	Cálculo de <i>offsets</i>	89
6.2.2.1	Cálculo del <i>offset</i> por discretización	91
6.2.2.2	Operaciones booleanas entre curvas cerradas.....	95
6.2.2.3	Cálculo del <i>offset</i> de varias curvas.....	96
6.2.3	Perfilado	97
6.2.4	Desbaste.....	97
6.2.4.1	Desbaste en zig-zag.....	98
6.2.4.2	Desbaste en espiral.....	101

6.2.5	Mecanizado a altura constante.....	104
6.2.5.1	Obtención de capas.....	104
6.2.5.2	Espaciado entre capas	105
6.2.5.3	Recortado del contorno	105
6.2.5.4	Cálculo de la trayectoria	106
6.2.6	Mecanizado a altura variable	106
6.2.6.1	Trayectoria de los puntos de contacto.....	107
6.2.6.2	Trayectoria de los centros de herramienta	107
Capítulo 7	Constructores de herramientas	111
7.1	Introducción.....	112
7.2	Definición de las herramientas virtuales	112
7.2.1	Herramientas reales disponibles.....	113
7.2.2	Diseño del perfil de las herramientas virtuales	113
7.3	Emulación de las herramientas virtuales.....	114
7.3.1	Posicionamiento de las herramientas reales	114
7.3.2	Emulación de una herramienta cilíndrica con otra cilíndrica de radio inferior	115
7.3.3	Emulación de una herramienta esférica con otra esférica de radio inferior	115
7.3.4	Emulación de una herramienta cónica con otra cónica de radio inferior	117
7.3.5	Emulación de una herramienta tórica.....	119
7.3.5.1	Emulación mediante herramientas esféricas.....	120
7.3.5.2	Emulación mediante herramientas tóricas	120
7.4	Cálculo de trayectorias.....	121
7.4.1	Herramientas de traslación	121
7.4.2	Herramientas de rotación	122
Capítulo 8	Constructores de objeto.....	123
8.1	Introducción.....	124
8.2	El algoritmo <i>Marching Cubes</i>	126
8.2.1	Características del algoritmo.....	126
8.2.2	Método	126
8.2.3	Limitaciones y mejoras	127
8.3	Poligonalización del modelo Superficie-Trayectoria	128
8.3.1	Algoritmo.....	128
8.3.2	Tamaño del cubo y del paso máximo de herramienta	133
8.3.3	Prueba de interioridad de los vértices y cálculo de los puntos de corte.....	133
8.3.3.1	Herramientas esféricas	133
8.3.3.2	Herramientas cilíndricas.....	134
8.3.3.3	Herramientas cónicas	136
8.3.3.4	Herramientas tóricas	137
8.3.4	Coherencia espacial	138
8.3.5	Incrementalidad del método	138

		Índices
		Índice general
Capítulo 9	Experimentos y resultados	141
	9.1 Introducción.....	142
	9.2 Diseño de los experimentos	142
	9.2.1 Elección de los modelos más representativos	142
	9.2.2 Parámetros de medida	143
	9.2.3 Banco de pruebas	143
	9.3 Parámetros cualitativos	143
	9.3.1 Parámetro 1: Facilidades para definir el interfaz de usuario ..	144
	9.3.1.1 Modelo CSG	144
	9.3.1.2 Modelo BRep.....	144
	9.3.1.3 Modelo S-T	145
	9.3.2 Parámetro 2: Adecuación para describir superficies libres	146
	9.3.2.1 Modelo CSG	146
	9.3.2.2 Modelo BRep.....	146
	9.3.2.3 Modelo S-T	146
	9.3.3 Parámetro 3: Facilidad para realizar operaciones booleanas ..	147
	9.3.3.1 Modelo CSG	147
	9.3.3.2 Modelo BRep.....	147
	9.3.3.3 Modelo S-T	147
	9.3.4 Parámetro 4: Adecuación para la visualización realista	148
	9.3.4.1 Modelo CSG	148
	9.3.4.2 Modelo BRep.....	148
	9.3.4.3 Modelo S-T	148
	9.4 Parámetros cuantitativos.....	149
	9.4.1 Cálculo del error al mecanizar superficies	149
	9.4.1.1 Error al mecanizar superficies planas.....	150
	9.4.1.2 Error al mecanizar superficies esféricas	150
	9.4.2 Ejemplo 1.....	152
	9.4.2.1 Diseño del objeto	152
	9.4.2.2 Definición de las trayectorias.....	154
	9.4.2.3 Cálculo del error	156
	9.4.3 Ejemplo 2.....	160
	9.4.3.1 Diseño del objeto	160
	9.4.3.2 Definición de las trayectorias.....	161
	9.4.3.3 Cálculo del error	163
Capítulo 10	Conclusiones y líneas futuras de investigación	167
	10.1 Conclusiones	168
	10.1.1 Consecución de los objetivos	168
	10.1.2 Características del modelo Superficie-Trayectoria	169
	10.1.3 Principales aportaciones	170
	10.2 Líneas futuras de investigación.....	171
	10.2.1 Optimización de las trayectorias.....	171
	10.2.2 Mecanizado con cinco ejes	172
	10.2.3 Cálculo de <i>offsets</i> y mecanizado en espiral	172
	10.2.4 Algoritmos de visualización.....	173

Anexo	Implementación.....	175
	A.1 Implementación	176
	A.1.1 Plataforma y software utilizados	176
	A.1.2 Interfaz de usuario.....	176
	A.1.2.1 Vista de proyecto	176
	A.1.2.2 Vista de dimensiones.....	177
	A.1.2.3 Vista de diseño	177
	A.1.2.3 Vista de escalado.....	179
	A.1.3 Jerarquía de clases.....	180
	Bibliografía	183

ÍNDICE DE FIGURAS

Figura 3.1	Tetraedro en modelo alámbrico.....	31
Figura 3.2	Almacenamiento para modelo alámbrico	31
Figura 3.3	Objeto en modelo BRep	32
Figura 3.4	Estructura de datos para almacenar el modelo BRep	32
Figura 3.5	Almacenamiento del modelo BRep mediante aristas aladas.....	33
Figura 3.6	Ejemplo de superficie BSpline.....	34
Figura 3.7	Elipsoide.....	35
Figura 3.8	Objeto generado mediante isosuperficies según el modelo Blob.....	36
Figura 3.9	Ejemplos de sólidos generados por barrido.....	37
Figura 3.10	Árbol que representa el proceso de construcción de un objeto por Geometría Sólida Constructiva.....	37
Figura 3.11	Objetos definidos según un modelo de enumeración de la ocupación espacial, almacenados en un árbol octal.....	39
Figura 3.12	Objetos definidos según un modelo de enumeración de la ocupación espacial, almacenados en un árbol BSP	39
Figura 3.13	Vegetal obtenido por geometría fractal	40
Figura 3.14	Objetos generados por geometría fractal.....	40
Figura 3.15	Planta generada con un modelo gramatical.....	41
Figura 3.16	Objeto generado con un sistema L	41
Figura 3.17	Ejemplos de sistemas de partículas para modelar las plumas de un ave y un géiser.....	43
Figura 3.18	Tejido modelado según sus características físicas.....	43
Figura 3.19	Ejemplos de fresas para mecanizado por control numérico	46
Figura 3.20	Esquemas de mecanizado.....	49
Figura 3.21	Mecanizados según la disposición de las trayectorias	50
Figura 3.22	Error producido por la interpolación lineal y una distancia excesiva entre puntos sucesivos	51
Figura 3.23	Montes y valles producidos por una distancia excesiva entre pasadas	51
Figura 3.24	Problemas de interferencias	51
Figura 3.25	Problema del compensado	52
Figura 3.26	Problemas del mecanizado isoparamétrico	54
Figura 4.1	Herramientas virtuales	61
Figura 4.2	Emulación de herramientas virtuales mediante herramientas reales	61
Figura 5.1	Curva paramétrica con los espacios paramétrico y de representación	70
Figura 5.2	Superficie paramétrica con los espacios paramétrico y de representación....	71
Figura 5.3	Superficie generada por extrusión de una curva plana.....	76

Índices

Índice de figuras

Figura 5.4	Superficie generada por revolución de una curva plana alrededor de un eje.....	77
Figura 5.5	Superficie generada por cruce de dos curvas planas	78
Figura 5.6	Superficie generada a partir de secciones definidas por curvas planas.....	78
Figura 5.7	Proyección de una curva sobre una superficie.....	80
Figura 5.8	Superficie trivariada para un valor (x,y)	83
Figura 5.9	Proyección de una curva sobre una superficie.....	84
Figura 5.10	Efecto conseguido al proyectar una curva y al desplegarla	85
Figura 5.11	Despliegue de una curva sobre una superficie	86
Figura 6.1	Diagrama de Voronoi para un contorno cerrado.....	90
Figura 6.2	Cálculo del <i>offset</i> por discretización	92
Figura 6.3	Eliminación de tramos no válidos entre autointersecciones	93
Figura 6.4	Operaciones booleanas entre curvas cerradas.....	95
Figura 6.5	Esquema de resolución para el preproceso, la unión, la intersección y la diferencia de curvas.....	95
Figura 6.6	<i>Offset</i> de una curva exterior con dos islas	97
Figura 6.7	Puntos singulares	99
Figura 6.8	Detección de singularidades	99
Figura 6.9	Funcionamiento del algoritmo de desbaste en zig-zag	101
Figura 6.10	Funcionamiento del algoritmo recursivo de desbaste en espiral	103
Figura 6.11	Vista superior y lateral de la intersección de una superficie S con una capa L_i	104
Figura 6.12	Espaciado constante entre capas	105
Figura 6.13	Espaciado variable entre capas.....	105
Figura 6.14	Cálculo de las áreas accesibles al ir profundizando en las sucesivas capas	105
Figura 6.15	Superficie con zonas no accesibles	107
Figura 6.16	Cálculo de la trayectoria de los centros de una herramienta esférica	107
Figura 6.17	Cálculo de la trayectoria de los centros de una herramienta cilíndrica	108
Figura 6.18	Cálculo de la trayectoria de los centros de una herramienta cónica	108
Figura 6.19	Cálculo de la trayectoria de los centros de una herramienta tórica	109
Figura 7.1	Forma y perfil de las herramientas reales disponibles.....	113
Figura 7.2	Emulación de dos herramientas virtuales con tres reales.....	114
Figura 7.3	Emulación de una herramienta cilíndrica	115
Figura 7.4	Emulación de una herramienta esférica	116
Figura 7.5	Cálculo del área sin desbastar al emular una herramienta esférica	116
Figura 7.6	Emulación de una herramienta cónica con otra de igual relación altura/radio	117

Figura 7.7	Emulación de una herramienta cónica con otra de mayor relación altura/radio.....	118
Figura 7.8	Emulación de una herramienta cónica con otra de menor relación altura/radio.....	118
Figura 7.9	Perfil de una herramienta tórica, con su zona de corte	119
Figura 7.10	Emulación de una herramienta tórica con una esférica.....	120
Figura 7.11	Parámetros para la emulación de una herramienta tórica con una esférica.....	120
Figura 7.12	Cálculo de las trayectorias reales para una herramienta de traslación	122
Figura 7.13	Cálculo de las trayectorias reales para una herramienta de rotación.....	122
Figura 8.1	Algoritmo de trazado de rayos	125
Figura 8.2	Casos posibles de intersección de un cubo con una superficie	127
Figura 8.3	Codificación de los vértices y de algunas aristas del cubo para <i>Marching Cubes</i>	130
Figura 8.4	Vista bidimensional de la inicialización de la matriz de vóxeles.....	130
Figura 8.5	Discretización de una curva c_i con un paso máximo de herramienta de p_{max}	130
Figura 8.6	Determinación del tamaño máximo del cubo.....	133
Figura 8.7	Parámetros y prueba de interioridad para una herramienta esférica.....	134
Figura 8.8	Parámetros de una herramienta cilíndrica	134
Figura 8.9	Prueba de interioridad para una herramienta cilíndrica.....	135
Figura 8.10	Parámetros de una herramienta cónica	136
Figura 8.11	Prueba de interioridad para una herramienta cónica	136
Figura 8.12	Parámetros de una herramienta tórica	137
Figura 8.13	Aprovechamiento de la coherencia espacial en el caso general	138
Figura 8.14	Ejemplos de poligonalización con el algoritmo modificado de <i>Marching Cubes</i>	139
Figura 9.1	Entorno de usuario de la aplicación diseñada basada en el modelo S-T	145
Figura 9.2	Efecto de montes y valles.....	150
Figura 9.3	Cálculo del área de la sección del monte.....	150
Figura 9.4	Cálculo del área del error al mecanizar una superficie esférica convexa.....	151
Figura 9.5	Objeto del ejemplo 1	152
Figura 9.6	Cálculo del área de la sección	152
Figura 9.7	Árbol de operaciones para generar el objeto del ejemplo 1 mediante un modelado CSG	153
Figura 9.8	Diseño del objeto por barrido con el modelo BRep	153
Figura 9.9	Grupos de planos para un mecanizado a altura constante	154
Figura 9.10	Mecanizado en paralelo	155

Figura 9.11	Mecanizado concéntrico	155
Figura 9.12	Trayectorias para el diseño y el mecanizado según el modelo S-T	156
Figura 9.13	Vista superior del mecanizado del cilindro por el exterior	157
Figura 9.14	Vista superior del mecanizado del cilindro por el interior	158
Figura 9.15	Vista superior del mecanizado de parte de la pieza inferior	158
Figura 9.16	Objeto del ejemplo 2	160
Figura 9.17	Diseño del objeto por barrido con el modelo BRep	160
Figura 9.18	Diseño del objeto con el modelo S-T	161
Figura 9.19	Intersección del objeto con los planos de mecanizado a altura constante ...	161
Figura 9.20	Herramienta virtual emulada con una herramienta esférica	163
Figura 9.21	Poligonalización del ejemplo 2 mediante el algoritmo modificado de <i>Marching Cubes</i>	165
Figura A.1	Vista de proyecto	176
Figura A.2	Vista de dimensiones	177
Figura A.3	Vista de diseño	177
Figura A.4	Asistente para la definición y emulación de herramientas virtuales	178
Figura A.5	Asistente para la generación de trayectorias	179
Figura A.6	Vista de simulación bidimensional	180
Figura A.7	Jerarquía de clases de la aplicación	181

ÍNDICE DE TABLAS

Tabla 4.1	Sintaxis de la gramática del modelo.....	64
Tabla 5.1	Algoritmo para la discretización de curvas en forma de polilíneas	80
Tabla 5.2	Algoritmo para la proyección de curvas planas sobre superficies 3D	84
Tabla 5.3	Algoritmo para el despliegue de curvas planas sobre superficies 3D.....	86
Tabla 6.1	Algoritmo de cálculo de la paralela a una polilínea	92
Tabla 6.2	Algoritmo de eliminación de tramos no válidos entre autointersecciones....	94
Tabla 6.3	Algoritmo de cálculo de operaciones booleanas entre curvas cerradas.....	96
Tabla 6.4	Algoritmo de desbaste en zig-zag	100
Tabla 6.5	Algoritmo recursivo de desbaste en espiral.....	102
Tabla 6.6	Algoritmo iterativo de desbaste en espiral	103
Tabla 8.1	Acciones a realizar dependiendo de la configuración de los vértices	131
Tabla 8.2	Algoritmo de poligonalización del modelo S-T.....	132
Tabla 9.1	Adecuación de los distintos modelos geométricos a las características cualitativas estudiadas	149
Tabla 9.2	Comparación de los resultados de cada modelo para el ejemplo 1	160
Tabla 9.3	Comparación de los resultados de cada modelo para el ejemplo 2	165
Tabla A.1	Características de la plataforma hardware utilizada para las pruebas	176



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 1

INTRODUCCIÓN

1.1 LOS SISTEMAS CAD/CAM

Un sistema de diseño y fabricación asistidos por ordenador CAD/CAM (*Computer Aided Design/Computer Aided Manufacturing*) es una aplicación o un conjunto de aplicaciones informáticas que facilitan el diseño y posterior fabricación de objetos industriales. Este tipo de sistemas engloban desde simples aplicaciones para diseñar objetos sencillos, hasta complejos sistemas informáticos que permiten no sólo diseñar, sino también obtener programas de control numérico para fabricar el objeto automáticamente, realizar complejos cálculos sobre las propiedades físicas de éstos, prever su comportamiento ante los agentes externos a que estén sometidos, visualizarlos de manera realista, obtener prototipos a bajo coste, calcular costes de producción, etc. Por esta razón, muchas veces se habla de sistemas CAD/CAM/CAE (*Computer Aided Design/Manufacturing/Engineering*) para hacer hincapié en que cubren todo el proceso ingenieril, desde el diseño hasta el producto final con todas sus características. La utilización de estos sistemas CAD/CAM en la industria ha supuesto una revolución en los métodos de producción.

En general, los sistemas CAD/CAM utilizados hasta ahora, presentan una dificultad clave cuando se quieren adaptar a industrias pequeñas, con personal poco especializado: Se trata de sistemas de propósito general, con una alta complejidad de uso y de aprendizaje, lo que requiere su manejo por ingenieros muy especializados. El modelo geométrico subyacente es un modelo de amplio propósito, en el que se consigue generalidad a costa de la sencillez de manejo.

Por estas razones se plantea la necesidad de realizar productos específicamente diseñados para un sector determinado, que se adapten a la forma de trabajo de los profesionales del ramo, sin necesidad de unos conocimientos muy profundos de ingeniería.

1.2 LOS MODELOS GEOMÉTRICOS

Conocemos por modelo geométrico a todo el conjunto de métodos y estructuras de datos para diseñar, manipular y guardar objetos geométricos. Habitualmente estos objetos van a ser tridimensionales y, además, serán el resultado de su creación con un sistema CAD/CAM.

El modelo geométrico es en buena medida responsable de las características del sistema de diseño y fabricación que lo incorpora. A lo largo de los últimos años se han ido definiendo modelos geométricos cada vez más complejos y con mayores capacidades. Los modelos más utilizados son los BRep, que definen los objetos a partir de sus fronteras, y los modelos basados en geometría sólida, que crean objetos complejos operando otros objetos más sencillos.

Los dos problemas fundamentales para los que el modelo geométrico debe aportar una solución son el diseño de los objetos y su fabricación automática.

En el problema del diseño se ha avanzado notoriamente en los últimos años: Son problemas relacionados en gran medida con la interfaz de usuario del sistema, aunque también interviene de forma crucial la filosofía del modelo geométrico, que permitirá o no al usuario plasmar de forma intuitiva su creatividad sobre el objeto diseñado.

El otro problema fundamental es la fabricación automática de los objetos diseñados utilizando máquinas herramienta. Esto supone la generación de las trayectorias que las herramientas deben seguir para dar lugar al objeto correctamente. Esta generación es un problema no resuelto completamente y las características del modelo geométrico inciden de manera fundamental en la facilidad de generación de las trayectorias.

La mayoría de los modelos separan completamente el diseño de la fabricación. Esta opción tiene la ventaja de que posibilita el diseño de objetos a usuarios no familiarizados con los procesos de fabricación. A cambio, la calidad del mecanizado de las piezas se resiente porque las estrategias de generación de trayectorias no se adaptan suficientemente bien al objeto diseñado. Incluso en ocasiones algunos objetos diseñados no pueden ser fabricados.

1.3 EL MODELO SUPERFICIE-TRAYECTORIA

Presentamos con este trabajo un nuevo modelo geométrico para la definición de objetos tridimensionales. Con ello se pretende dar solución a los problemas que entrañan los modelos geométricos clásicos para el diseño de determinadas piezas dentro del sector industrial. Se trata de un modelo que relaciona íntimamente el proceso de diseño y el de fabricación. Así incrementamos la calidad de las piezas obtenidas, al guiar en parte la generación de las trayectorias. Esta filosofía conlleva, no obstante, el peligro de dificultar y restringir demasiado los procesos de diseño. Con el objetivo de no menguar estas capacidades, se definen nuevas herramientas de diseño, una de las principales aportaciones del sistema descrito.

En el modelo Superficie-Trayectoria el diseño se basa en la definición de superficies y de trayectorias sobre estas superficies por las que se hace pasar herramientas virtuales de diseño, concepto éste que es una generalización del concepto de herramienta de mecanizado. El sistema proporciona así una guía sobre la que calcular las trayectorias reales de mecanizado, con lo que este proceso se simplifica. Además, las trayectorias obtenidas son más precisas que las generadas por otros métodos con lo que, en definitiva, el objeto fabricado es de mayor calidad.

1.4 ESTRUCTURA DEL DOCUMENTO

En el capítulo 2 de esta memoria profundizamos en las motivaciones que nos han llevado a realizar este trabajo y en los objetivos marcados.

El capítulo 3 incluye un recorrido por el estado del arte de los modelos geométricos y del problema del mecanizado. Se presentan aquí las principales características de los modelos más utilizados, sus usos y su formulación básica. En una segunda parte, se trata el mecanizado con la diferente problemática a la que debemos hacer frente para abordar con éxito esta cuestión.

El modelo Superficie-Trayectoria se presenta en el capítulo 4, inicialmente de una forma intuitiva para pasar a continuación a presentar un formalismo que utiliza el concepto de gramática para describir el modelo. De esta manera queda caracterizado el problema a partir de la sintaxis y proponiendo un modelo operacional para la semántica que se expone en capítulos posteriores.

En los siguientes capítulos se presentan los constructores semánticos para las diferentes reglas sintácticas. Así, en el capítulo 5 se tratan los constructores de curvas y superficies; en el capítulo 6 se presentan los constructores de trayectorias; el capítulo 7 aborda los constructores de herramientas y, por último, se tratan los constructores de objeto en el capítulo 8.

Una vez abordados los principales métodos propuestos, se ponen a prueba en el capítulo 9, que trata la cuestión de los experimentos más adecuados para probar el sistema y los resultados que se obtienen al llevarlos a cabo.

En el capítulo 10 se presentan las conclusiones que se han extraído, tanto de los experimentos como del propio desarrollo de la memoria. Se constata el grado de consecución de los objetivos marcados al inicio, las principales aportaciones que se han hecho y, para terminar, se plantea un conjunto de problemas que dan lugar a las líneas futuras de investigación.

Por último, se ofrece un anexo que recoge la información más técnica sobre el desarrollo y la implementación de una aplicación que utiliza el modelo objeto de esta tesis.



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 2

MOTIVACIÓN Y OBJETIVOS

2.1 MOTIVACIÓN

2.1.1 La industria y el CAD/CAM

La necesidad de definir nuevos modelos geométricos ha ido surgiendo, generalmente, a remolque de las crecientes necesidades de la industria moderna. Conforme los sistemas CAD/CAM se han ido implantando en la industria, sus requerimientos se han ido ampliando y con ellos los del modelo geométrico que los soporta. De los primitivos modelos de tipo alámbrico que no permitían más que un diseño burdo de objetos simples, hemos pasado a definir curvas y superficies libres en el espacio y a construir objetos nuevos a partir de otros mediante operaciones booleanas.

El caso del modelo Superficie-Trayectoria no es una excepción, y surge como respuesta a las necesidades que plantea la pequeña industria. La mayoría de los sistemas CAD/CAM actuales tienen un propósito general, es decir, permiten el diseño de prácticamente cualquier objeto que se desee fabricar. Son sistemas muy grandes, con innumerables herramientas y opciones y que requieren una alta especialización en el personal que los maneja. Frente a ellos, la pequeña industria se encuentra en desventaja, ya que no dispone de medios necesarios para poder implantar sistemas de este tipo. Esta es la razón principal por la que la implantación de los sistemas de diseño y fabricación automáticos en las PYMES es todavía testimonial en muchos sectores.

Un caso paradigmático es el de la industria del calzado. Se trata de un sector atomizado en el que la inmensa mayoría de las empresas tienen un escaso tamaño y un muy bajo nivel tecnológico. Con el objetivo de abrirles la posibilidad de acceder a tecnologías modernas de producción, nos hemos planteado la definición de un modelo geométrico adaptado a sus necesidades y que permita la construcción de sistemas de diseño y de manufactura de manejo sencillo y bajo coste.

Aunque el presente proyecto se planteó para resolver las dificultades existentes a la hora del diseño de componentes para calzado y se basa, por ello, en los procesos que se siguen en este sector, se ha comprobado que muchos otros sectores industriales siguen procesos similares. Para todos ellos el modelo Superficie-Trayectoria se presenta como un modelo útil.

2.1.2 Limitaciones de los modelos actuales

El modelo geométrico utilizado en un sistema de diseño y fabricación asistidos por computador, condiciona de manera importante las características del mismo. Se han detectado carencias en los modelos descritos hasta el momento, que hacen difícil su adaptación a algunos problemas industriales:

- Son modelos geométricos generales que sacrifican la sencillez en aras de una mayor generalidad. Se trata de sistemas dirigidos a ingenieros o a usuarios con amplios conocimientos

en diseño industrial. No obstante, en numerosos sectores industriales, especialmente en los más pequeños, es notable la ausencia de personal con este tipo de formación.

- Algunos modelos se encuentran totalmente orientados al diseño de los objetos, lo que los aleja de los procesos de fabricación. Como consecuencia, la obtención automática de la pieza (por mecanizado, utilizando máquinas de control numérico) es costosa y de baja calidad.
- Otros modelos se encuentran muy cercanos a los procesos de fabricación, basándose casi exclusivamente en el diseño de las trayectorias que deben seguir las herramientas de mecanizado. Su utilización es escasamente intuitiva, muy supeditada a la forma en que trabajan las máquinas, dificultando la reutilización de diseños, el uso de formas complejas, la visualización, y otros aspectos importantes de un sistema CAD/CAM moderno.

En concreto, los dos modelos geométricos más utilizados en la actualidad para el diseño de objetos tridimensionales con sistemas de diseño y fabricación asistidos por computador son el modelo de Geometría Sólida Constructiva (CSG) y los modelos superficiales (BRep). Ambos modelos poseen características que los hacen recomendables, pero existen algunos inconvenientes importantes que son, fundamentalmente:

- El diseño de objetos complejos resulta complicado: En el caso del modelo CSG el limitado número de primitivas con que suelen contar hace que no sea posible diseñar superficies libres. En el caso del modelo BRep, podemos diseñar estas superficies pero su modificación y ajuste (generalmente manipulando los puntos de control de las superficies) es escasamente intuitivo.
- En cuanto a la fabricación de los objetos, el proceso de obtención de las trayectorias de mecanizado a partir de las formas geométricas no es trivial. De hecho, este proceso en muchos casos se convierte en el cuello de botella de los sistemas de diseño y fabricación. Esto se debe a que el usuario tiene total libertad a la hora del diseño, lo que no se corresponde con las restricciones que impone la fabricación con máquinas de control numérico. Algunos problemas frecuentes al obtener la trayectoria de las herramientas son:
 - No es posible obtener una trayectoria que dé lugar al objeto diseñado, bien porque éste es imposible, bien porque las características de la máquina no lo permiten.
 - La trayectoria que se obtiene no es correcta: cuando las formas son complejas las estrategias de mecanizado no siempre son adecuadas a la precisión que se desea obtener.

Todos estos inconvenientes nos llevan a plantear la hipótesis de un modelo híbrido, basado en la generación de formas libres y de trayectorias de mecanizado de una máquina abstracta, que incorpora herramientas de diseño. Esta máquina sería posteriormente emulada

por la máquina real de que se disponga. La máquina abstracta proporciona una serie ilimitada de capacidades de diseño, ya que sería configurable por el usuario y, a la vez, mediante el proceso de emulación, proporcionaría unos procesos de fabricación adecuados.

2.2 OBJETIVOS

Para tratar de paliar las limitaciones de los modelos geométricos actuales y facilitar su implantación en el sector de la pequeña industria, nos planteamos construir un modelo geométrico con tres objetivos generales:

- Facilitar el procedimiento de diseño de los objetos y hacerlo más versátil incorporando nuevas funcionalidades adaptadas a las necesidades de los usuarios.
- Obtener objetos mecanizados con alta calidad sin aumentar la complejidad de los métodos.
- Construir un modelo que permita alcanzar los objetivos anteriores a través de métodos de uso sencillo e intuitivo, para limitar los costes de programación e implantación de sistemas CAD/CAM basados en el modelo propuesto.

Los objetivos generales planteados se pueden desglosar en un conjunto de objetivos concretos que planteamos a continuación.

2.2.1 Objetivos para facilitar el diseño

Para tratar de alcanzar el objetivo de facilitar el diseño de los objetos, nos planteamos los siguientes objetivos concretos:

- Nuestro modelo debe ser híbrido, incorporando los procedimientos de diseño más avanzados y de mayor utilidad de los modelos existentes. Nos estamos refiriendo a la posibilidad de definir superficies libres (típicas del modelo BRep) y de realizar operaciones entre ellas (procedentes del modelo CSG).
- Debemos añadir nuevas funcionalidades que faciliten el diseño y cubran las carencias de otros modelos. Para ello incorporaremos las herramientas virtuales de diseño como una de las principales aportaciones.
- Debemos contar con una visualización realista de calidad suficiente, para que el usuario pueda tener una idea ajustada a la realidad del objeto diseñado.

2.2.2 Objetivos para mejorar la fabricación

El segundo objetivo general es el de mejorar la calidad de la fabricación. Podremos conseguirlo si alcanzamos los siguientes objetivos particulares:

- Puesto que los métodos totalmente automáticos no proporcionan en ocasiones trayectorias de buena calidad, utilizaremos las directrices que el usuario proporciona durante el diseño para mejorar el mecanizado de las piezas.

- Debemos asegurar la integridad del modelo, de tal manera que los objetos que se puedan diseñar también se puedan fabricar.

2.2.3 Objetivos para disminuir los costes

Al implantar un nuevo sistema informático en una industria se incurre en diferentes costes: desarrollo del producto, implantación, etc. Nos marcamos los siguientes objetivos concretos para limitar estos costes:

- El modelo debe ser sencillo e intuitivo y no debe requerir el uso de grandes sistemas ni de periféricos especiales.
- Debemos perseguir la adecuación del modelo geométrico al problema y a la forma de trabajar del profesional del sector, para que esté adaptado a las necesidades de la pequeña industria.

Como fin último, se persigue introducir de manera más generalizada los sistemas de CAD/CAM en industrias de pequeño tamaño, lo que redundaría en un aumento de la productividad y en una mejora en la calidad del producto final



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 3

ESTADO DEL ARTE

3.1 MODELOS GEOMÉTRICOS

3.1.1 Introducción

Un modelo geométrico es el conjunto de estructuras de datos, algoritmos de manejo de esas estructuras y operaciones disponibles entre los datos, que dan soporte a la información geométrica correspondiente a un objeto. Es, en definitiva, todo aquello necesario para crear, mantener, manipular y almacenar objetos geométricos en los sistemas de diseño automáticos.

Los modelos geométricos deben permitir a los programas CAD/CAM computar algunos de estos aspectos [Bowyer, 1996]:

- Volumen
- Masa
- Área de las superficies
- Momentos de inercia
- Fuerza
- Flexibilidad
- Flujo de calor y de fluidos
- Integridad y ajuste mecánicos
- Fabricación automática
- Representación visual

Los cuatro primeros apartados (cálculo del volumen, la masa, el área y los momentos de inercia) constituyen lo que se denomina cálculo de **propiedades integrales**; la computación de fuerzas, flexibilidad, y flujo de calor y fluidos se encuentra relacionada con las técnicas de cálculo por **elementos finitos** y la **dinámica de fluidos**; la fabricación automática (el cálculo de las trayectorias que deben seguir las herramientas y su planificación) es un problema de **inteligencia artificial**; por último, la representación visual del modelo es fundamental para comprobar la corrección del mismo y obtener una representación fácilmente interpretable por el ser humano.

Las características deseables para todos los modelos geométricos son [Requicha, 1992], [Bowyer, 1996]:

- Preciso: debe representar los objetos sin aproximaciones.
- No ambiguo: los objetos no deben estar sujetos a más de una posible interpretación.
- Único: un objeto no debe permitir más que una representación.
- Compacto: debe utilizar poco espacio para codificar los objetos.
- Rápido: las operaciones con los objetos deben consumir poco tiempo.
- Útil : debe ser adecuado para los objetos que se quieran representar.

3.1.2 Taxonomía de los modelos geométricos

Realizar una clasificación de los modelos geométricos no es una tarea sencilla. Aunque la mayoría de los autores [Hearn, 1995], [Foley, 1990], [Bowyer, 1996], [Sanz, 2002] coinciden en distinguir entre modelos superficiales y sólidos, hay un conjunto de representaciones difícilmente clasificables. Nuestra propuesta es:

- Modelo alámbrico
- Modelos superficiales
 - Modelo poligonal
 - Superficies cuádricas y supercuádricas
 - Isosuperficies
 - Superficies libres
- Modelos sólidos
 - Objetos generados por barrido
 - Geometría Sólida Constructiva
 - Enumeración espacial
- Otros modelos
 - Modelo de geometría fractal
 - Modelo gramatical
 - Sistemas de partículas
 - Modelos basados en características físicas

3.1.3 Modelo alámbrico

Se trata del primer modelo que aparece en la literatura. En este modelo se representan únicamente los vértices y las aristas de los objetos, con lo que no se almacena ninguna información superficial o sobre los sólidos.

Para almacenar los objetos en este modelo existen dos opciones:

- Mediante una lista de aristas
- Mediante una lista de vértices y otra de aristas

Entre las ventajas del modelo destacamos:

- Es muy sencillo
- La visualización es muy rápida

LISTA DE ARISTAS		
Arista	Inicio	Fin
A ₁	(x ₁ , y ₁ , z ₁)	(x ₂ , y ₂ , z ₂)
A ₂	(x ₂ , y ₂ , z ₂)	(x ₃ , y ₃ , z ₃)
A ₃	(x ₃ , y ₃ , z ₃)	(x ₁ , y ₁ , z ₁)
A ₄	(x ₂ , y ₂ , z ₂)	(x ₄ , y ₄ , z ₄)
A ₅	(x ₁ , y ₁ , z ₁)	(x ₄ , y ₄ , z ₄)
A ₆	(x ₄ , y ₄ , z ₄)	(x ₃ , y ₃ , z ₃)

(a)

LISTA DE VERTICES	
Vértice	Coordenadas
V ₁	(x ₁ , y ₁ , z ₁)
V ₂	(x ₂ , y ₂ , z ₂)
V ₃	(x ₃ , y ₃ , z ₃)
V ₄	(x ₄ , y ₄ , z ₄)

(b)

LISTA DE ARISTAS		
Arista	Inicio	Fin
A ₁	V ₁	V ₂
A ₂	V ₂	V ₃
A ₃	V ₃	V ₁
A ₄	V ₂	V ₄
A ₅	V ₁	V ₄
A ₆	V ₄	V ₃

Figura 3.2

Almacenamiento para el modelo alámbrico. Tetraedro de la figura 3.1 guardado en una sola lista de aristas (a) y en una lista de vértices y otra de aristas (b)

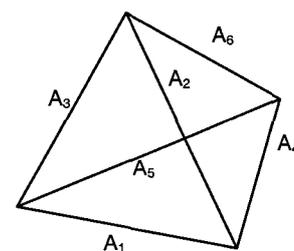


Figura 3.1
Tetraedro en modelo alámbrico

Por otro lado, los inconvenientes son numerosos:

- La visualización es ambigua (dos objetos diferentes pueden visualizarse de la misma manera)
- No permite visualizar las superficies ni la representación realista
- No guarda información sobre los puntos que son interiores o exteriores al objeto
- No es posible realizar un mecanizado del objeto puesto que no existen las superficies
- Es un modelo aproximado y no permite definir objetos complejos
- Al visualizarlo, puede haber pérdidas de líneas de silueta

Este modelo no se emplea en la actualidad, ya que existen otros modelos mucho más completos y que permiten también una visualización muy rápida.

3.1.4 Modelos superficiales

En los modelos superficiales o BRep (*Boundary Representation*), los objetos vienen representados por sus vértices, las aristas que los unen y las caras que forman estas aristas. Las caras pueden ser planas (hablamos entonces de modelo poligonal) o curvas (modelo de superficies curvas). En cualquier caso, existe una fórmula, denominada de Euler-Poincaré, que relaciona el número de vértices, aristas, caras y agujeros, y que asegura que el objeto definido es un sólido topológicamente correcto [Bowyer, 1996]. Esta fórmula es:

$$F + V - E - R = 2(S - H) \quad \text{Ecuación 3.1}$$

donde F es el número de caras, V el número de vértices, E el número de aristas, R el número de anillos, S el número de objetos independientes y H el número de agujeros.

En cuanto a las estructuras de almacenamiento posibles, una alternativa muy habitual es utilizar listas de dos tipos:

- Listas de geometría, en las que se almacenan los datos geométricos, es decir, los vértices.
- Listas de atributos, en las que se guardan características como el color de las caras, el grado de transparencia, las propiedades del material, etc.

LISTA DE VÉRTICES			LISTA DE ARISTAS				LISTA DE CARAS		
Vértice	Coordenadas	Atrib.	Arista	Inicio	Fin	Atrib.	Cara	Aristas	Atrib.
V ₁	(x ₁ , y ₁ , z ₁)	AtrV ₁	A ₁	V ₁	V ₂	AtrA ₁	C ₁	A ₁ , A ₂ , A ₃	AtrC ₁
V ₂	(x ₂ , y ₂ , z ₂)	AtrV ₂	A ₂	V ₂	V ₃	AtrA ₂	C ₂	A ₂ , A ₄ , A ₆	AtrC ₂
V ₃	(x ₃ , y ₃ , z ₃)	AtrV ₃	A ₃	V ₃	V ₁	AtrA ₃	C ₃	A ₃ , A ₆ , A ₅	AtrC ₃
V ₄	(x ₄ , y ₄ , z ₄)	AtrV ₄	A ₄	V ₂	V ₄	AtrA ₄	C ₄	A ₁ , A ₅ , A ₄	AtrC ₄
			A ₅	V ₁	V ₄	AtrA ₅			
			A ₆	V ₄	V ₃	AtrA ₆			

Figura 3.4

Estructura de datos para almacenar el modelo BRep. Almacenamiento del objeto de la figura 3.3 con una lista de vértices, otra de aristas y otra de caras

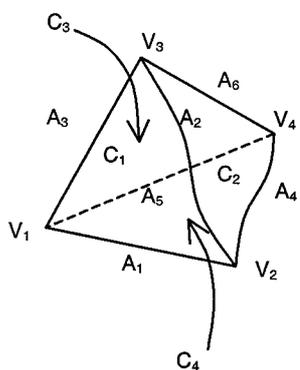


Figura 3.3
Objeto en modelo BRep

Es posible utilizar únicamente una lista de caras para almacenar los objetos BRep, aunque las opciones más extendidas manejan dos listas (de caras y vértices) o tres (de caras, aristas y vértices). La utilización de tres listas tiene la ventaja de que reduce la redundancia al máximo, permite comprobar fácilmente la consistencia de los datos y proporciona la posibilidad de asociar atributos tanto a las caras, como a las aristas y a los vértices.

Para facilitar el acceso a los elementos de las listas es posible añadir referencias cruzadas. Así conocemos a qué cara es adyacente cada arista, o qué aristas son incidentes a cada vértice, por ejemplo.

Entre los atributos que se pueden almacenar en las listas, se encuentra, por ejemplo, la ecuación del plano que forma cada polígono (en el caso del modelo poligonal) o la ecuación de la superficie curva (en el caso del modelo de superficies curvas).

Otra representación muy utilizada es la denominada "aristas aladas". En este caso, el objeto se almacena en forma de grafo no dirigido, cuyos nodos representan los vértices y las aristas del objeto y las uniones las relaciones de incidencia entre las aristas y los vértices.

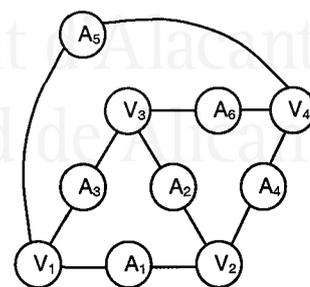


Figura 3.5
Almacenamiento del modelo BRep mediante aristas aladas. En la imagen, estructura de datos para el objeto de la figura 3.3

3.1.4.1 Modelo poligonal

Se trata del caso particular del modelo BRep en el que las caras son polígonos planos [Hearn, 1995] [Bowyer, 1996]. Entre los atributos asociados a las caras se suele incluir la normal de cada polígono y la ecuación del plano. Si tenemos la precaución de ordenar los vértices de los polígonos en sentido antihorario vistos desde el exterior del objeto, la normal se puede calcular utilizando el producto vectorial y siempre se encuentra dirigida hacia la parte externa del objeto. En cuanto a la ecuación del plano puede obtenerse fácilmente a partir de las coordenadas de los vértices.

El hecho de poder calcular la normal y la ecuación de manera tan sencilla, facilita algunas de las operaciones básicas que pueden realizarse con los objetos poligonales: pruebas de interioridad/exterioridad, cálculo de intersecciones con rectas o con otros planos, cálculo del sombreado de los polígonos y su visualización realista, etc.

El principal inconveniente de este modelo es la imprecisión, por ser aproximado. Sin embargo, se trata de uno de los modelos más utilizados, especialmente como modelo secundario para visualización. Su adaptación al mecanizado cuenta con el inconveniente de que la calidad viene limitada por la precisión del modelo.

Los polígonos que forman un objeto poligonal pueden tener cualquier número de aristas. Sin embargo los casos más habituales son los de los polígonos de tres y cuatro aristas. La utilización de triángulos tiene la ventaja de que los vértices siempre son coplanares.

Las ventajas del modelo poligonal son, en resumen:

- Tiene una definición sencilla.
- La visualización es muy rápida. De hecho existen multitud de desarrollos tanto software como hardware, que resuelven la visualización realista de manera muy eficiente y con muy buenos resultados estéticos.
- Los polígonos planos se describen mediante ecuaciones lineales.
- Puede representar objetos poliédricos de manera exacta.
- La visualización es no ambigua.
- La prueba de interioridad/exterioridad es muy sencilla.

Los inconvenientes son:

- Se trata de una representación aproximada para objetos no poliédricos.
- En objetos curvos, para conseguir una buena aproximación, es necesario utilizar un número muy elevado de polígonos.
- El mecanizado depende de la precisión del modelo poligonal.
- Algunos cálculos son complejos.

3.1.4.2 Modelo de superficies curvas

Se trata de una generalización del modelo BRep, en el que las caras son superficies no necesariamente planas. El caso más habitual es el de las superficies paramétricas (Bézier, Splines, BSplines, NURBS, etc).

Las superficies paramétricas se definen partiendo de una nube de puntos, a partir de la cual se traza una superficie por interpolación o por aproximación. La nube de puntos se encuentra ordenada y se denomina poliedro de control. La disposición de los puntos del poliedro de control determina la forma de la superficie.

Las superficies se definen mediante una ecuación paramétrica, generalmente a partir de dos parámetros (u,v) . Aunque la superficie se puede determinar mediante procedimientos de interpolación (Splines, Hermite, ...) o aproximación (Bézier, BSplines, NURBS, ...) son los métodos de aproximación los más adecuados para los sistemas CAD/CAM, por sus buenas propiedades de continuidad y suavidad de las formas, especialmente las superficies de grado cúbico.

Un ejemplo de superficie BSpline es:

$$S(u, v) = \sum_{j=0}^m \sum_{k=0}^n p_{j,k} B_{j,e}(v) B_{k,d}(u) \quad \text{Ecuación 3.2}$$

para $(m+1) \times (n+1)$ puntos de control $p_{j,k}$ y donde B son las denominadas funciones *blending* o de combinación de Bernstein.

Incorpora las ventajas del modelo poligonal, eliminando la imprecisión al ser un modelo exacto. No obstante, supone un aumento de la complejidad. Es uno de los modelos más implantados

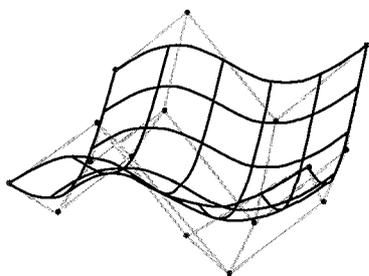


Figura 3.6
Ejemplo de superficie BSpline. También se muestran los puntos y el poliedro de control

en los sistemas CAD/CAM, y su visualización se puede abordar mediante la adaptación de los algoritmos de sombreado de polígonos a las superficies paramétricas, o bien utilizando un modelo secundario, generalmente el poligonal. Se trata de uno de los modelos mejor adaptados para el mecanizado.

Sus ventajas fundamentales son:

- Es un modelo exacto para superficies curvas.
- Las superficies paramétricas basadas en las funciones de Bernstein son muy estables numéricamente.
- Estas superficies son invariantes a transformaciones afines.
- Cumple una propiedad que facilita el cálculo de colisiones: las superficies se encuentran dentro del casco convexo formado por el polígono de control.
- Es un modelo muy utilizado y existe una gran cantidad de algoritmos desarrollados.

Tiene, no obstante, algunos inconvenientes:

- El cálculo de las curvas tiene complejidad cuadrática.
- El polígono de control es complicado de manejar.
- El cálculo de intersecciones es muy complejo.
- Las superficies con topología no rectangular tienen una formulación muy complicada.

3.1.4.3 Otros modelos superficiales

Superficies cuádricas y supercuádricas

En estos modelos los objetos vienen descritos mediante ecuaciones [Hearn, 1995]. Tienen la gran ventaja de que su definición es exacta. Algunos de estos objetos son: esferas, elipsoides, paraboloides, toros y superelipsoides. La definición de las ecuaciones suele hacerse de forma implícita o paramétrica.

Por ejemplo, en el caso de un elipsoide, la ecuación implícita es:

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1 \quad \text{Ecuación 3.3}$$

La ecuación paramétrica es

$$\left. \begin{array}{l} x = r_x \cos \phi \cos \theta \\ y = r_y \cos \phi \sin \theta \\ z = r_z \sin \phi \end{array} \right\} \begin{array}{l} -\pi \leq \phi \leq \pi \\ -\pi \leq \theta \leq \pi \end{array} \quad \text{Ecuación 3.4}$$

Las ventajas de este modelo son:

- Es una representación exacta.
- El mecanizado es muy preciso al tratarse de representaciones exactas y tener información sobre la superficie.
- Al estar descrito mediante ecuaciones, permite operaciones como la intersección con otros objetos.

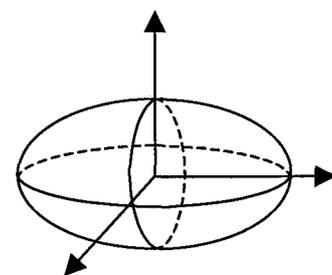


Figura 3.7
Elipsoide

Tiene inconvenientes, sin embargo, que restringen mucho su uso:

- Es un modelo muy limitado, ya que sólo algunas figuras muy concretas pueden representarse mediante una ecuación.
- La visualización necesita de un modelo secundario (generalmente el modelo poligonal) o la utilización de métodos avanzados no tan rápidos (trazado de rayos).

Isosuperficies

Los objetos se definen a partir de un conjunto de primitivas en el espacio que definen un campo de fuerza. La superficie exterior que define el campo de fuerza se denomina isosuperficie y viene dada por una función de densidad [Puchol, 1997]. Existen varios modelos, según la definición de la función de densidad:

- Blobs:

$$f(x, y, z) = \sum_{k=1}^N b_k e^{-a_k r_k^2} - T = 0 \quad \text{Ecuación 3.5}$$

- Metaballs:

$$f(r) = \begin{cases} b \left(\frac{1 - 3r^2}{d^2} \right) & 0 < r \leq \frac{d}{3} \\ \frac{3}{2} b \left(\frac{1 - r^2}{d^2} \right) & \frac{d}{3} < r \leq d \\ 0 & r > d \end{cases} \quad \text{Ecuación 3.6}$$

- Soft objects:

$$f(r) = \begin{cases} 1 - \frac{22r^2}{9d^2} + \frac{17r^4}{9d^4} - \frac{4r^6}{9d^6} & 0 < r \leq d \\ 0 & r > d \end{cases} \quad \text{Ecuación 3.7}$$

Las isosuperficies se utilizan muy frecuentemente para modelar objetos orgánicos.

3.1.5 Modelos sólidos

3.1.5.1 Sólidos generados por barrido

Los objetos se generan a partir de una forma bidimensional que recorre una trayectoria [Hearn, 1995]. Podemos distinguir dos tipos:

- Por extrusión: La trayectoria es de traslación, siendo en general una curva.
- Por revolución: La trayectoria es de rotación, es decir, la forma bidimensional gira alrededor de un eje.

Hablamos de método de barrido generalizado cuando la forma varía durante el recorrido.

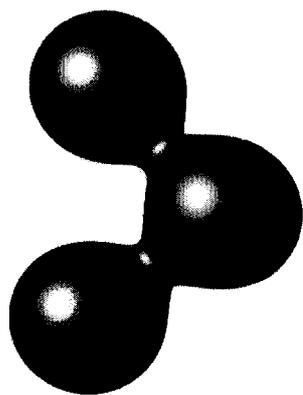


Figura 3.8
Objeto generado mediante isosuperficies según el modelo Blob

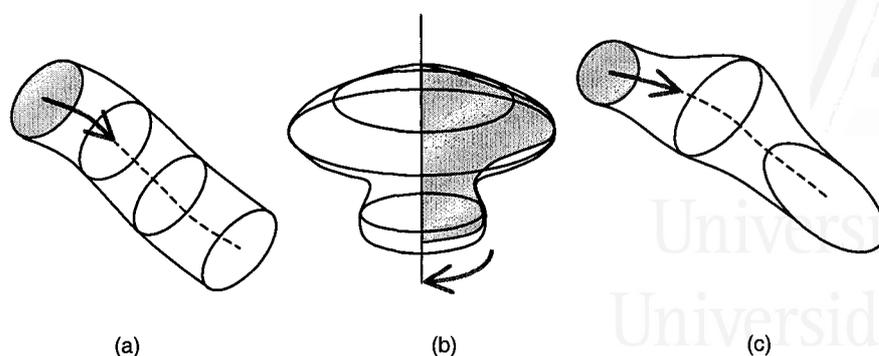


Figura 3.9
Ejemplos de sólidos generados por barrido. Por extrusión (a), por revolución (b) y por barrido generalizado (c).

Algunos aspectos a tener en cuenta son:

- El sólido se calcula incorporando cortes a pasos discretos de la trayectoria. La precisión del objeto depende de la separación entre cortes, y de si esa distancia es constante o dependiente de la curvatura.
- Otro factor es la orientación de la sección con respecto a la trayectoria. Generalmente, la trayectoria es normal a la sección.

3.1.5.2 Geometría sólida constructiva (CSG)

Los objetos están formados por primitivas geométricas simples (esferas, conos, cilindros, cubos, ...) unidos mediante operaciones booleanas (unión, intersección y diferencia) [Hearn, 1995].

Los objetos se representan mediante árboles binarios en los que las ramas son primitivas y los nodos internos operaciones booleanas.

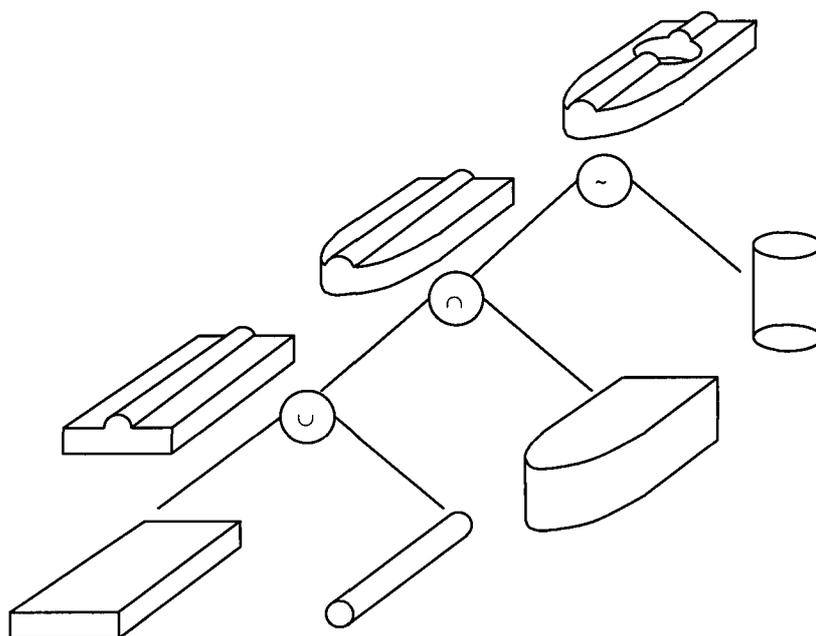


Figura 3.10
Árbol que representa el proceso de construcción de un objeto por Geometría Sólida Constructiva

Dichas primitivas pueden definirse mediante instanciación de un conjunto de primitivas predefinido o utilizando cualquiera de los métodos anteriores. Se trata de un modelo exacto (si lo es la definición inicial de las primitivas), pero su visualización requiere la utilización de un modelo secundario. No obstante es, junto con el modelo BRep, el más extendido en los sistemas actuales de diseño.

Las ventajas que hacen de este modelo uno de los más utilizados son:

- Incorpora información sobre el sólido, por lo que es un modelo muy adecuado para la extracción de características físicas.
- Si las primitivas son exactas, se trata de un modelo exacto.
- Es adecuado para el mecanizado.
- Si conocemos el test de interioridad/exterioridad de las primitivas (por ejemplo, cuando están definidas mediante ecuaciones) el test del objeto completo se calcula de forma inmediata, dada la equivalencia entre las operaciones de unión e intersección y las de disyunción y conjunción.

Tiene, no obstante, algunas limitaciones:

- La visualización es complicada: debe utilizarse un modelo secundario (por ejemplo, poligonal) o métodos avanzados (por ejemplo, trazado de rayos, que es poco eficiente).
- Está limitado por las primitivas que se hayan definido.

3.1.5.3 Enumeración de la ocupación espacial

El espacio se divide en regiones y cada región se etiqueta como interior o exterior al objeto [Hearn, 1995]. Existen dos tipos de representaciones básicas: árboles octales (*octrees*) y árboles de partición binaria del espacio (árboles BSP).

Árboles octales

El objeto es aproximado mediante cubos sólidos o vóxeles, almacenados en una estructura jerárquica en forma de árbol. Inicialmente se define un cubo que engloba a todo el objeto. Este cubo se divide en ocho cubos de igual tamaño, y cada uno de ellos se etiqueta de la siguiente manera:

- Si el cubo no contiene ninguna porción del objeto, se etiqueta como exterior.
- Si el cubo se encuentra completamente en el interior del objeto, se etiqueta como interior.
- Si el cubo contiene parte de un objeto, se divide en otros ocho cubos y se repite el proceso.

El proceso de división termina cuando todos los cubos son completamente interiores o exteriores, o bien se ha alcanzado una profundidad máxima en el árbol.

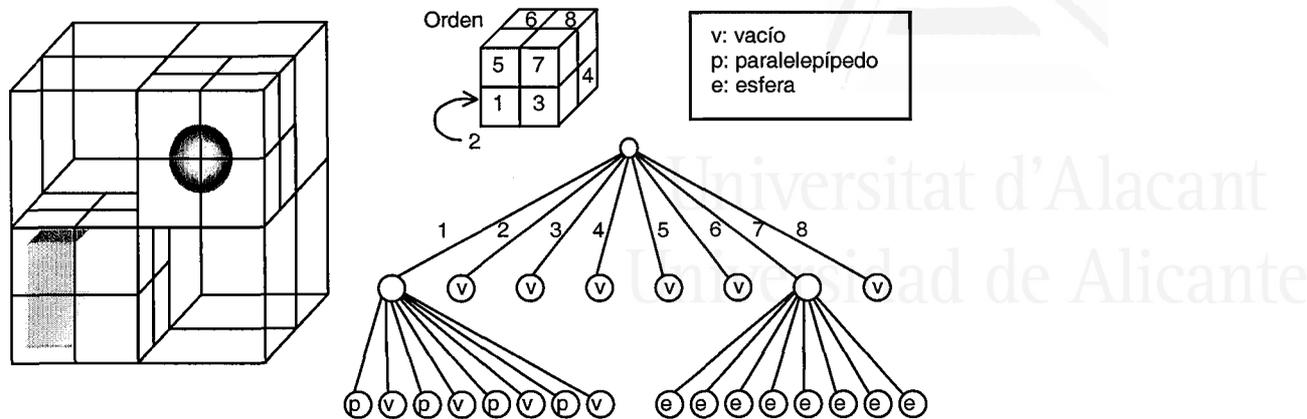


Figura 3.11
Objetos definidos según un modelo de enumeración de la ocupación espacial, almacenados en un árbol octal

Árboles BSP

El proceso es similar, pero el espacio se divide a cada paso en dos mediante un plano. Estos planos se pueden definir en cualquier orientación, de manera que los árboles BSP se adaptan mucho mejor a la forma del objeto y la división del espacio es menor.

Los métodos de enumeración espacial facilitan las operaciones lógicas entre objetos, pero tienen problemas de falta de precisión y las operaciones globales tienen una mayor complejidad (traslación, rotación, etc). Por otro lado, la representación es aproximada para objetos con caras no planas.

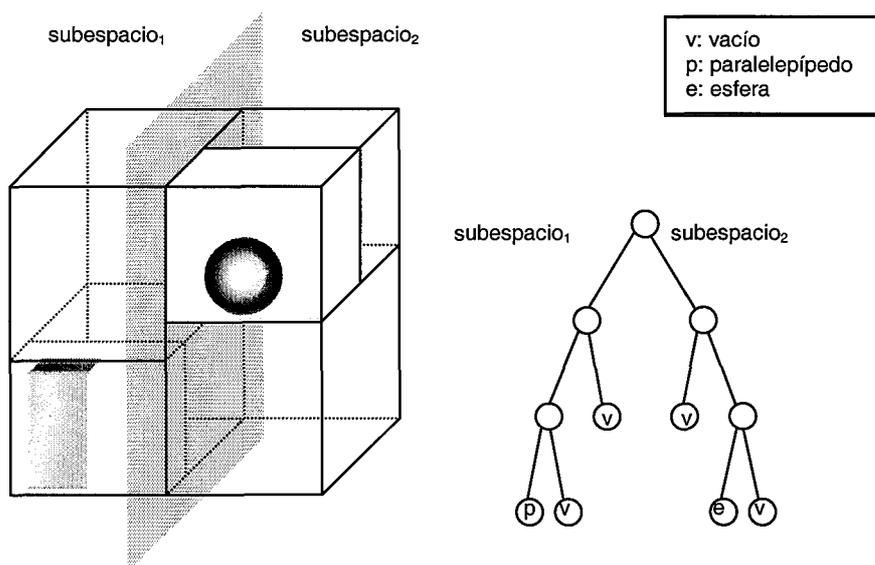


Figura 3.12
Objetos definidos según un modelo de enumeración de la ocupación espacial, almacenados en un árbol BSP

3.1.6 Otros modelos

Existe un conjunto heterogéneo de modelos geométricos cuyo uso en sistemas CAD/CAM es más testimonial.

3.1.6.1 Modelo de geometría fractal

A diferencia de la geometría euclídea, en que los objetos se definen mediante ecuaciones, en la geometría fractal los objetos se representan mediante procedimientos [Hearn, 1995]. De forma simplificada, un objeto fractal se especifica mediante una operación repetida infinitamente para producir detalle de las partes del objeto. Esto proporciona a los objetos fractales dos características fundamentales:

- Conforme nos acercamos a los objetos fractales seguimos teniendo el mismo detalle que el objeto completo, es decir, contamos con infinito detalle.
- Existe una cierta autosimilitud entre las partes del objeto y el todo, es decir, las estructuras globales del objeto se repiten en las distintas partes a diferente escala.

Un fractal se genera al aplicar de forma repetida una función de transformación llamada generador a las primitivas o iniciadores que forman el objeto (puntos, rectas, curvas, superficies, poliedros...). Las funciones de transformación pueden ser de diferente tipo (traslaciones, escalados, rotaciones, transformaciones no lineales, ...) y se pueden aplicar de manera determinista o incorporando algún procedimiento aleatorio. La aplicación de las funciones se extiende hasta el infinito, aunque en la práctica se detiene cuando el detalle es menor que un cierto umbral, determinado generalmente por el tamaño del píxel de la pantalla en que se va a representar. Sea, por ejemplo, un iniciador I_0 y una función de transformación o generador G . El objeto se obtiene al aplicar sucesivamente G a I_0 .

$$I_1 = G(I_0) \rightarrow I_2 = G(I_1) \rightarrow \dots \rightarrow I_i = G(I_{i-1}) \rightarrow \dots$$

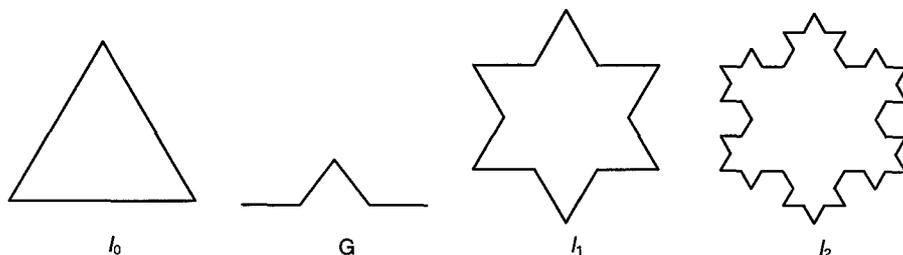


Figura 3.14

Objetos generados por geometría fractal. Se parte de un iniciador I_0 y un generador G . Se muestran los dos primeros pasos de la generación I_1 e I_2

La geometría fractal se utiliza en gráficos por ordenador para representar objetos naturales, fundamentalmente: rocas, montañas, plantas, nubes, agua, plumas, etc.

3.1.6.2 Modelo gramatical

En este modelo los objetos se generan utilizando una gramática con reglas de producción [Hearn, 1995]. Los objetos que se pueden definir son palabras del lenguaje que define la gramática. Este modelo es un método de definición de objetos complejos, cuyas primitivas (que serían los símbolos terminales del lenguaje) deben estar descritas en algún otro modelo de los vistos. El modelo CSG puede verse como un caso particular de modelo gramatical.

Un ejemplo de una gramática para generar árboles es:

```

árbol → TRONCO {rama}
rama → TALLO {rama} | TALLO {hoja}
hoja → PECIOLO | SUPERFICIE

```

Un árbol concreto será una palabra del lenguaje definido por esta gramática.

Dentro de los modelos gramaticales destacan los Sistemas L (sistemas de Lindenmayer) concebidos como sistemas de reescritura de cadenas [McCormack, 1993] [Mock, 1998]. La reescritura es una técnica para definir objetos complejos mediante la sustitución sucesiva de partes de una cadena inicial utilizando un conjunto de reglas de reescritura. Estas reglas pueden verse como una gramática. La interpretación de un sistema de este tipo suele hacerse a través de lo que se ha dado en llamar gráficos de tortuga: se trata de una tortuga que al desplazarse va dejando un rastro. Un dibujo de la tortuga puede verse como una cadena de la gramática que define sus movimientos. Los cuatro símbolos permitidos son:

- F: movimiento hacia delante pintando
- f: movimiento hacia delante sin pintar
- +: giro a la izquierda un ángulo predeterminado
- -: giro a la derecha un ángulo predeterminado

Por ejemplo, en la figura 3.16 se representa la figura obtenida en varios pasos de reescritura para una cadena inicial

$$F-F-F-F-$$

siendo la única regla de reescritura

$$F \rightarrow F-F+F-F-F$$

Una variación de estos sistemas son los sistemas L estocásticos, en los que se introduce un conjunto de probabilidades asociadas a cada regla de reescritura. Otro sistema alternativo son los sistemas L paramétricos que incluyen la posibilidad de pasar parámetros a las reglas.

Los sistemas L se han utilizado especialmente para modelar el crecimiento de las plantas y de objetos fractales. De hecho, un sistema L puede verse también como un caso particular de sistema fractal.



Figura 3.15
Planta generada con un modelo gramatical

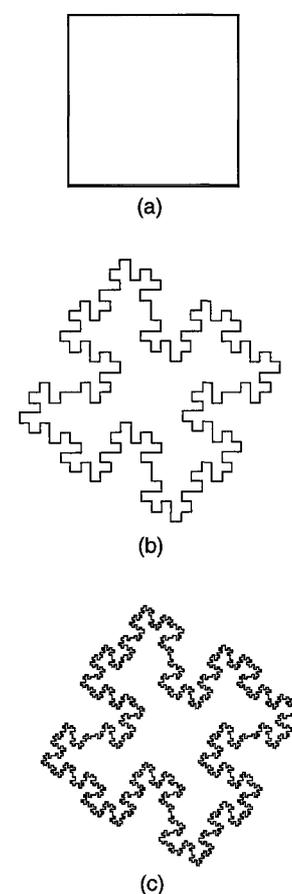


Figura 3.16
Objeto generado con un sistema L. Se muestra la cadena inicial (a), y el objeto en dos momentos de la reescritura (b) y (c)

3.1.6.3 Sistemas de partículas

Los sistemas de partículas son una manera de modelar objetos difusos como el fuego, las nubes, el humo o el agua. Los objetos se definen como conjuntos de partículas que siguen algún tipo de ley física, como las leyes de la mecánica de Newton. Estas partículas siguen una trayectoria u ocupan una posición dependiendo de esta ley, que contiene alguna componente aleatoria [Reeves, 1983] [Hearn, 1995].

A diferencia de los modelos clásicos, un objeto descrito por un sistema de partículas no está representado por un conjunto de superficies o sólidos estáticos, sino por una nube de partículas que definen su volumen y que pueden cambiar de forma o posición. Además pueden crearse o destruirse partículas durante el proceso. Por otro lado, el proceso de definición es no determinista, es decir, se usan procesos estocásticos para crear y cambiar la forma y apariencia de los objetos. En definitiva, los sistemas de partículas son un ejemplo de modelado procedimental estocástico y, por lo tanto, están emparentados con los fractales o con los sistemas L estocásticos. Como ellos, permiten definir objetos complejos con poco esfuerzo y ajustar el nivel de detalle a la vista particular que se tenga de la escena.

Dado el carácter dinámico del modelo, estas técnicas se utilizan especialmente para crear animaciones. El procedimiento de creación de una escena con este modelo puede resumirse en los siguientes pasos:

1. Crear partículas nuevas
2. Asociar a cada partícula nueva un conjunto de atributos
3. Destruir todas las partículas que hayan superado su tiempo máximo de vida.
4. Transformar las partículas restantes según sus atributos dinámicos y las leyes físicas introducidas
5. Visualizar las partículas que queden
6. Volver al paso 1

Durante el procedimiento puede introducirse alguna componente probabilística. Por ejemplo, las partículas pueden crearse siguiendo un modelo estocástico, añadiendo una cierta aleatoriedad en el número de partículas creadas y en las características de cada una. Esta misma técnica puede aplicarse durante el proceso dinámico de transformación. Por último cada partícula puede tener una probabilidad de ser destruida. Esta probabilidad aumentará conforme aumente su tiempo de vida.

Existen varias posibilidades para la visualización del sistema de partículas: podemos utilizar la partícula para definir una primitiva centrada en ella. Por ejemplo, podemos dibujar una esfera alrededor de cada partícula. Ajustando los parámetros de color, transparencia o material, podemos modelar de esta manera la lluvia, la nieve, el humo, etc.

Otra posibilidad es situar fuentes de luz en las partículas. De esta manera puede definirse el fuego o los fuegos artificiales.

También podemos utilizar las partículas para definir una nube de puntos y trazar una superficie que los interpole o los aproxime. De esta manera es sencillo simular diferentes fluidos, tejidos, plumas, cabellos, etc.

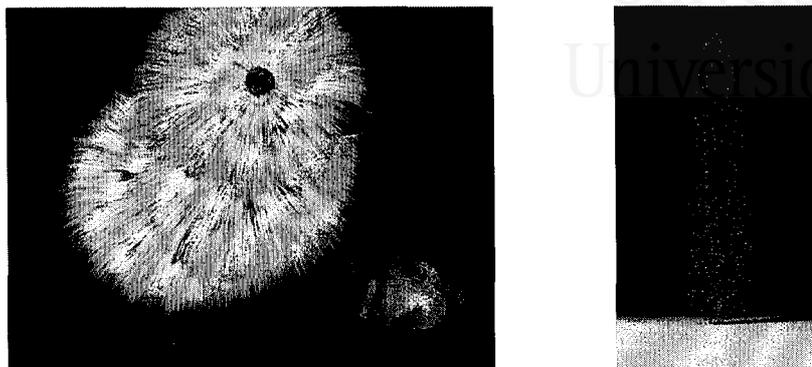


Figura 3.17
Ejemplos de sistemas de partículas para modelar las plumas de un ave y un géiser

3.1.6.4 Modelos basados en características físicas

Un método común para modelar objetos no rígidos es aproximarlos mediante una red de nodos con conexiones flexibles [Hearn, 1995]. Sobre los nodos actúan fuerzas que producen deformaciones en las conexiones que suelen modelarse utilizando muelles o resortes. La compresión o estiramiento de las conexiones viene dada por la constante k del muelle. El desplazamiento x de un nodo bajo la influencia de una fuerza F_x (aplicada en sentido x) viene dado por la siguiente expresión, conocida como ley de Hooke:

$$-F_x = -k \cdot x \quad \text{Ecuación 3.8}$$

La relación en los demás sentidos es idéntica. En la figura 3.18 encontramos un modelo bidimensional de un tejido y su aspecto una vez visualizado.

Los objetos homogéneos se modelarán utilizando resortes idénticos en todos los nodos. Por el contrario, la utilización de resortes con distinta constante nos permite modelar objetos cuyas características sean distintas en cada dirección.

Este tipo de modelos se utilizan para definir telas, tejidos animales, objetos flexibles, gomas, estructuras, etc. En CAD/CAM se emplean frecuentemente estas técnicas como modelo secundario para calcular fuerzas, presiones, etc., mediante descomposición del objeto inicial en elementos finitos.

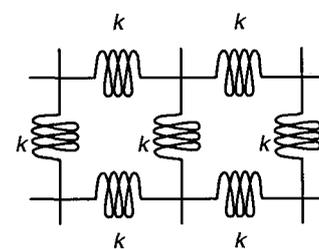


Figura 3.18
Tejido modelado según sus características físicas

3.2 MECANIZADO

3.2.1 Introducción

Por mecanizado entendemos la fabricación automática de piezas utilizando máquinas de Control Numérico. Textualmente, una máquina de Control Numérico (CN) es aquella controlada por números. El término se aplicaba originalmente a las máquinas programables mediante tarjetas perforadas, pero que no iban equipadas con ordenadores. Más tarde, debido a la reducción de precios y tamaños de los procesadores, las máquinas empezaron a dotarse con ordenadores que las controlaban, dando lugar a las máquinas CNC (*Computer Numerally Controlled*) para diferenciarlas de las NC (*Numerally Controlled*) que no iban equipadas con un ordenador. En la actualidad se utiliza generalmente la denominación de NC o CN para denominar todas las máquinas de control numérico, tengan o no ordenador [Lasheras, 1989], [IMT, 1992].

Los inicios de las máquinas de CN se remontan a principios del siglo XIX, con la aparición de los primeros telares gobernados por tarjetas perforadas. Pero hasta los años 50 del siglo XX no aparecen las primeras máquinas controladas numéricamente, tal y como las conocemos actualmente. Se trataba de las primeras fresadoras controladas por ordenador y que se utilizaron en la industria armamentística. El uso de estas máquinas se generaliza ya a partir de los años 70, habiendo llegado en estos momentos a ser imprescindibles incluso en industrias de pequeño tamaño.

El uso de sistemas de CN aporta numerosas ventajas, mejorando considerablemente, entre otros aspectos:

- La productividad: las operaciones se realizan en las condiciones óptimas, sin tiempos muertos y con gran rapidez en los posicionamientos.
- La flexibilidad: el cambio de programa para trabajar en una serie diferente es casi inmediato.
- La precisión: se minimizan las desviaciones de la pieza fabricada con respecto a la diseñada.
- La fiabilidad: se reducen al máximo los errores y la necesidad de realizar muchos controles.
- La repetibilidad: el automatismo con que se repiten las operaciones hace que los resultados sean idénticos.

No obstante, también supone cambios en los métodos de fabricación que pueden llevar al desastre si no se afrontan adecuadamente: los operarios deben adquirir una formación muy especializada, los métodos de producción deben cambiar, las inversiones iniciales son muy importantes, el mantenimiento y actualización de los sistemas es imprescindible, etc.

En esta sección trataremos el problema del mecanizado, entendido como el problema del cálculo de las trayectorias que debe

seguir una máquina de Control Numérico para generar de manera física una pieza lo más parecida posible al objeto diseñado.

3.2.2 Tipos de máquinas y herramientas

Las máquinas de CN pueden clasificarse, atendiendo al trabajo que realizan, en tres clases:

- Máquinas de control punto a punto: en estas máquinas sólo se pueden programar los puntos finales de cada desplazamiento, pero no las trayectorias, cuyo trazado y velocidad no se controlan. Las herramientas no realizan ningún mecanizado en los desplazamientos, sino solamente en los puntos programados. Son máquinas de este tipo las punteadoras, taladradoras, máquinas de soldar por puntos, etc.
- Máquinas de control paraxial: en este caso, pueden programarse tanto los desplazamientos a puntos determinados como el mecanizado durante estos desplazamientos, aunque en este caso sólo si son paralelos a los ejes. Un ejemplo son los tornos.
- Máquinas de contorneado: además de las operaciones que realizan las máquinas anteriores pueden programarse para mecanizar en desplazamientos rectos y curvos sin limitación. Esta característica hace que el control de los motores deba cuidarse más que en los casos anteriores, ya que una desincronización en alguno de los motores que mueven los ejes de la máquina provocaría un error en el mecanizado de la pieza.

Las máquinas de contorneado son las que centran hoy en día la mayoría de los trabajos sobre mecanizado de piezas.

De todos los procesos de mecanizado que pueden realizarse con estas máquinas (mandrinado, taladrado, roscado, amolado, fresado, etc) nos vamos a centrar en el fresado, proceso que consiste en el arranque progresivo de material mediante un tipo de herramienta llamada fresa que está provista de elementos o aristas cortantes. Cuando el proceso de fresado está en marcha, la fresa avanza lentamente sobre la materia prima arrancando el material sobrante en su movimiento. Para conseguir eliminar este material la fresa debe girar a gran velocidad (valores típicos son de 3.000 a 10.000 rpm dependiendo del tipo de material y masa de la fresa).

La geometría de la fresa se convierte en un factor determinante a la hora de conseguir un determinado acabado en la pieza a fabricar. La máquina que porta la herramienta sólo puede controlar un punto fijo de dicha herramienta normalmente colocado sobre su eje de rotación. Esto produce uno de los problemas más importantes a la hora de controlar la fabricación con este tipo de máquinas; se denomina *compensado* de herramienta y consiste en saber dónde colocar el centro de la herramienta para poder mecanizar cierto punto del objeto, teniendo en cuenta, además, que para ello no se debe incidir en una superficie ya mecanizada.

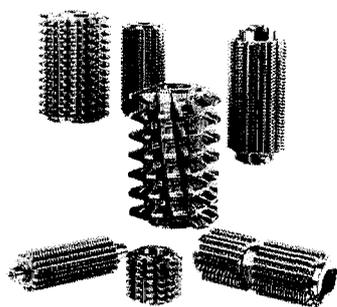


Figura 3.19
Ejemplos de fresas para
mecanizado por control
numérico

Los tipos de geometría de herramienta que podemos encontrar en la industria son muy variados: fresas cónicas, cilíndricas, esféricas, tóricas, etc. Cada una de ellas presenta una problemática diferente en cuanto al control del compensado.

Las máquinas fresadoras son máquinas de contorneado, muy versátiles y que pueden realizar también otros procesos de mecanizado típicos como el mandrinado, taladrados, etc. En la mayoría de fresadoras la fresa, que gira a gran velocidad, permanece fija mientras que el movimiento sobre los distintos ejes lo realiza el objeto. Sin embargo, también son numerosas las fresadoras donde los movimientos de traslación los realiza el brazo que porta la fresa.

Existen fresadoras de 2 a 5 ejes:

- Máquinas de 2 ejes: la máquina sólo puede moverse en dos ejes.
- Máquinas de 2 y $\frac{1}{2}$ ejes: la herramienta puede moverse en dos ejes simultáneamente y en un tercero cuando está estática. Se utiliza para mecanizados a profundidad constante.
- Máquinas de 3 ejes: la herramienta puede moverse en los tres ejes X, Y, Z simultáneamente.
- Máquinas de 5 ejes: la herramienta puede cambiar no sólo su posición sino también su orientación. Tienen una gran precisión y pueden mecanizar piezas que no es posible fabricar con otras máquinas.

Como vemos la mayoría incorporan ejes de traslación longitudinal sobre los ejes cartesianos X, Y y Z. El resto de ejes suelen estar asociados a movimientos de rotación sobre alguno de los ejes principales citados. La adición de ejes otorga versatilidad pero dificultad en la generación de la trayectoria de la fresa. Hoy en día existe un gran esfuerzo investigador en este sentido, y numerosos autores presentan trabajos sobre generaciones automáticas de trayectorias sobre máquinas de 5 ejes [Jerard, 1989b], [Li, 1994].

Otras máquinas de uso más particular son los tornos, que permiten trabajar una pieza haciéndola girar sobre un eje y arrancándole el material periféricamente hasta transformarla en la pieza deseada. Esta operación recibe el nombre de torneado. La materia a transformar es colocada sobre la parte giratoria llamada plato (eje Y). La herramienta cortadora (habitualmente de tipo tórico) está fija a un eje roscado (husillo) que provoca un desplazamiento transversal del brazo que sujeta la fresa. Además, este brazo puede desplazarse bien longitudinal o angularmente a través de un tercer eje (eje Z). Todos ellos en movimiento definen una trayectoria helicoidal sobre el objeto que va a ser mecanizado.

3.2.3 El problema del mecanizado

Los sistemas de Control Numérico se programaban en un principio directamente: el operario programaba las trayectorias que debían seguir las herramientas en un lenguaje poco intuitivo, muy cercano al lenguaje máquina. En la actualidad, los sistemas CAD

generan automáticamente las trayectorias de las herramientas y los programas que manejan el Control Numérico. Este proceso permite disminuir los errores, el tiempo de generación de los programas y facilitan la modificación sencilla de los mismos. Por eso los sistemas actuales reciben el nombre de CAD/CAM o CAD/CAM/CAE: el ordenador asiste al usuario no sólo en el diseño, sino también en la fabricación del objeto e incluso en la ingeniería [Chu, 1998]. Sin embargo, la generación completamente automática de los programas de mecanizado por CN a partir de los modelos geométricos de las piezas, requiere una enorme síntesis de conceptos y métodos matemáticos, ingenieriles, geométricos, computacionales, de inteligencia artificial y de práctica manual que no siempre es factible incorporar [Jerard, 1989].

De estas afirmaciones se desprende que uno de los problemas básicos del CAD/CAM es la obtención de las trayectorias de mecanizado que dan lugar a una pieza lo más perfecta posible, fabricada, además, en el menor tiempo de que seamos capaces. Independientemente de cómo abordemos el problema, las características que debe cumplir este proceso de generación de trayectorias son [Li, 1994] [Arkin, 2000]:

- **Validez:** El resultado de mecanizar el material siguiendo la trayectoria descrita debe ser la pieza diseñada, de manera que no se elimine material en las zonas que no lo requieren, ni que se dejen áreas sin desbastar adecuadamente.
- **Precisión:** La mayoría de los objetos que se fabrican utilizando controles numéricos son piezas de alta precisión, por lo que es imprescindible limitar los errores que se pueden producir al generar las trayectorias de mecanizado. Además, el nivel de calidad debe ser adecuado a la pieza.
- **Eficiencia:** Hay tres medidas fundamentales de eficiencia: el incremento de productividad que se produce al acelerar el proceso de desarrollo y fabricación del producto; la eficiencia del algoritmo generador de trayectorias en términos de tiempo de proceso y de espacio de almacenamiento; y el tiempo de mecanizado requerido para obtener un objeto terminado. Aunque la eficiencia de las trayectorias es un concepto difícil de medir, existen algunos requerimientos mínimos que deben cumplirse:
 - Ninguna región debe mecanizarse repetidamente ya que, además del incremento del tiempo de mecanizado, se pueden producir mermas en la calidad de la pieza final.
 - Debe evitarse un excesivo número de retracciones de la herramienta, que producen, evidentemente, un aumento del tiempo de mecanizado, así como problemas en la entrada de la herramienta en el material, pudiendo dejar marcas.
- **Versatilidad:** El sistema debe ser versátil ante la gran variedad de formas de los objetos a producir, permitiendo el uso de diferentes estrategias de generación de trayectorias dependiendo de las características del objeto.

La problemática del mecanizado puede verse desde dos niveles de amplitud: la determinación de la estrategia de mecanizado y el cálculo de las trayectorias. Para resolver adecuadamente el problema, es necesario abordar ambos niveles y obtener soluciones para los dos.

3.2.4 Estrategias de mecanizado

Por estrategia de mecanizado entendemos el conjunto de decisiones que tomamos para abordar el mecanizado de una pieza: por dónde deben pasar las trayectorias, si deben ser o no paralelas entre sí, cómo ir profundizando en la pieza, etc.

El modelo geométrico que utiliza el sistema CAD condiciona de manera importante la generación de trayectorias de mecanizado. En la mayoría de los sistemas, durante el proceso de diseño de las piezas, el usuario no proporciona información sobre las estrategias que se deben seguir en la fabricación. Esto hace que el sistema afronte la generación de las trayectorias de mecanizado de dos maneras:

- Automáticamente: el sistema genera las trayectorias del CN de manera automática, siguiendo una estrategia definida por el sistema. Este procedimiento puede ser útil para usuarios no familiarizados con las máquinas, pero da lugar a piezas de baja calidad, debido a la dificultad de determinar automáticamente la mejor estrategia a seguir.
- Manualmente: Es el usuario el que determina la estrategia a seguir, aunque el sistema pueda generar automáticamente la trayectoria concreta partiendo de las indicaciones del usuario. En este caso, o el diseñador tiene unos conocimientos muy amplios de mecanización, o es el personal de mecanizado el que debe determinar la estrategia adecuada.

La estrategia que utilicemos para generar las trayectorias depende de los siguientes factores:

- La geometría de la pieza.
- El tipo de máquina con que contemos, es decir, su número de ejes: 2, 2 y $\frac{1}{2}$, 3 ó 5.
- La forma de la herramienta que utilicemos para mecanizar: esférica, cilíndrica, cónica o tórica.
- El material que debemos mecanizar.
- El tipo de mecanizado que queramos obtener:
 - Vaciado: mecanizado rápido, de poca precisión, para eliminar de manera rápida y burda el grueso del material.
 - Acabado: mecanizado más lento, de precisión alta, cuyo fin es terminar la pieza.
 - Refinado: mecanizado muy lento, con una herramienta muy pequeña y de muy alta precisión, para refinar aquellas partes con mucho detalle que las herramientas habituales no pueden terminar.
- Restricciones temporales, referidas al tiempo máximo de fabricación de la pieza.

Todo sistema debe tener en cuenta estos factores: si la estrategia se define manualmente, será el usuario el que deba tener una formación suficiente para tomar las decisiones adecuadas; si la selección de la estrategia es automática, será el sistema el que tome las decisiones, con las dificultades que supone tratar este tipo de cuestiones.

Las estrategias que podemos aplicar en el mecanizado pueden clasificarse según diferentes criterios.

Atendiendo al espacio geométrico sobre el que se trabaja, distinguimos:

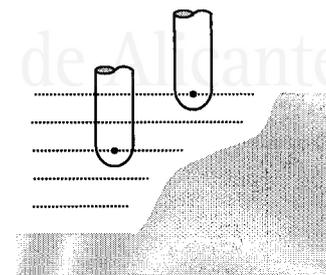
- Métodos que utilizan el espacio paramétrico: estos métodos tratan de obtener las coordenadas (x,y,z) del espacio cartesiano a partir de las coordenadas paramétricas (u,v) de la superficie. El método más evidente es el mecanizado isoparamétrico, que veremos más adelante, caracterizado porque el proceso de cálculo de las trayectorias es inmediato aplicando la ecuación de la superficie, aunque no es adecuado en la mayoría de los casos.
- Métodos que utilizan el espacio cartesiano de representación: el punto de partida son las coordenadas del objeto en el espacio cartesiano y se trata de obtener las coordenadas paramétricas que se corresponden con aquellas. El problema a resolver es, por lo tanto, obtener las coordenadas (u,v) a partir de coordenadas (x,y,z) del espacio de representación. Este paso no es, sin embargo, inmediato y requiere la resolución de sistemas de ecuaciones no lineales o de discretizaciones. No obstante, estos son los métodos más utilizados.

Atendiendo a la forma en que se profundiza en el material, distinguimos (Figura 3.20):

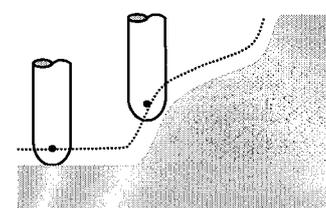
- Mecanizado a altura constante: El mecanizado se realiza por planos, con un valor de profundidad constante en cada uno. Está especialmente indicado para máquinas de 2 y $\frac{1}{2}$ ejes y para vaciados.
- Mecanizado a altura variable y orientación constante: El mecanizado se realiza cambiando la altura de la herramienta para seguir la superficie que queremos obtener pero sin alterar su orientación. Se utiliza para acabados y para refinados si la máquina de que se dispone es de 3 ejes.
- Mecanizado a altura variable y orientación normal: Durante el mecanizado se cambia tanto la altura como la orientación de la herramienta para incidir en el material de manera ortogonal a la superficie de mecanizado. Este mecanizado produce acabados y refinados de gran precisión, pero sólo es posible en máquinas de 5 ejes.

Atendiendo a la disposición de las trayectorias, podemos clasificar los mecanizados en las siguientes categorías (Figura 3.21):

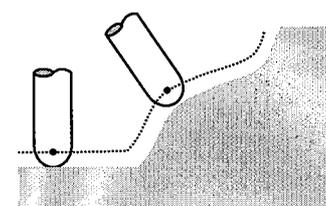
- Zig-zag: La herramienta se mueve en sucesivas capas, generalmente planos horizontales. En cada plano, la herramienta traza líneas rectas paralelas, por lo que también



(a)



(b)



(c)

Figura 3.20
Esquemas de mecanizado. A altura constante (a), a altura variable y orientación constante (b) y a altura variable y orientación normal (c)

se le llama mecanizado paralelo a una dirección. El movimiento de la herramienta es bidireccional. Se utiliza sobre todo para mecanizados de altura constante, con poca precisión.

- Espiral: La trayectoria de la herramienta se obtiene con *offsets* (paralelas) sucesivos de la curva de contorno en la capa actual. Por esto también se le conoce como mecanizado paralelo al contorno. Como en el caso anterior se suele utilizar para mecanizados en altura constante.
- Radial: La herramienta se mueve a lo largo de los radios de un círculo. Se utiliza para mecanizados con altura variable.
- Contorneado: La trayectoria sigue los bordes de la pieza. Se utiliza para acabados y refinados, por ser de alta precisión.
- Ortogonal: Se utiliza para acabados y refinados. La herramienta se mueve ortogonalmente en las concavidades donde no ha podido entrar una herramienta grande.
- Isoparamétrico: Es una opción sencilla para mecanizar superficies libres. Las trayectorias se trazan sobre la superficie coincidiendo con líneas isoparamétricas: En cada pasada de la herramienta, uno de los parámetros de la superficie permanece constante mientras el otro va incrementándose en pasos discretos constantes. Para cada valor de los parámetros (u,v) se obtiene la posición sobre la superficie en coordenadas cartesianas. El gran problema de este método es que dos rectas paralelas en el espacio paramétrico no lo son, en general, en el espacio cartesiano.

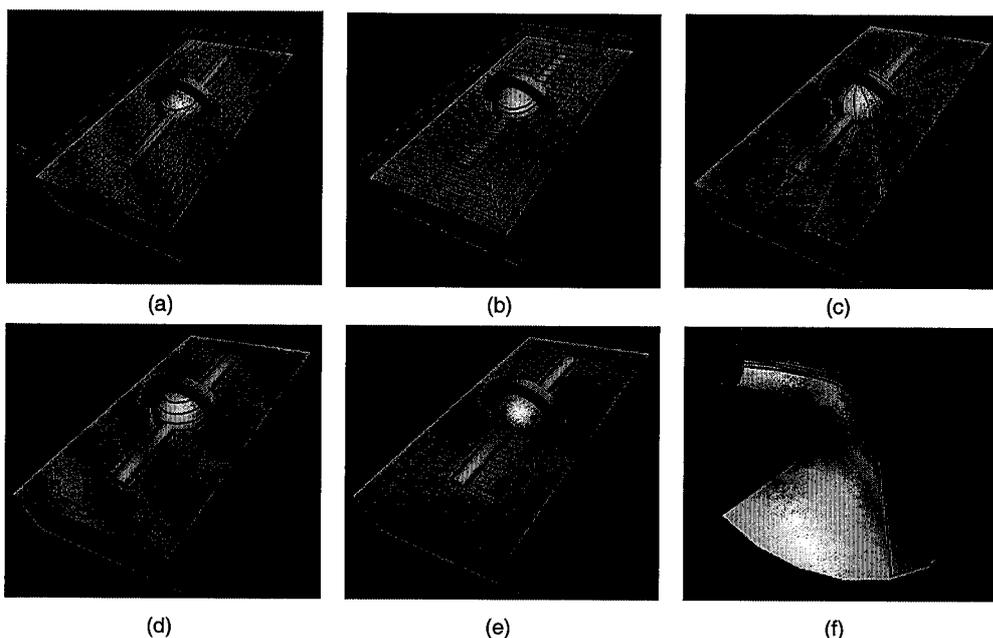


Figura 3.21
Mecanizados según la disposición de las trayectorias. En zig-zag (a), en espiral (b), radial (c), por contorneado (d), ortogonal (e) e isoparamétrico (f)

3.2.5 Cálculo de trayectorias

Una vez tomada la decisión de con qué estrategia abordar el problema del mecanizado, debemos obtener las trayectorias concretas que debe seguir la herramienta, según las decisiones adoptadas. Los métodos existentes para estos cálculos dependen del tipo de estrategia elegida.

Antes de tratar los distintos métodos de generación de trayectorias, veremos algunos de los problemas generales propios del problema de la generación de trayectorias, para a continuación ver como los resuelven estos métodos.

- Problemas de precisión: Estos problemas se resuelven controlando dos distancias:
 - Distancia entre puntos sucesivos: Dado que las máquinas de control numérico se basan en una interpolación lineal de posiciones sucesivas de la herramienta, se pueden producir problemas entre pasos contiguos si la distancia entre puntos es excesiva (Figura 3.22)
 - Distancia entre pasadas: La distancia entre pasadas sucesivas puede producir montes y valles, especialmente cuando se utilizan herramientas no cilíndricas o la geometría del objeto es muy compleja. Se debe llegar a un compromiso entre el error producido y la velocidad de mecanizado (Figura 3.23).
- Problemas por tamaño o tipo inadecuado de la herramienta: Puede producirse una interferencia y un arranque de material no deseado debido a que la curvatura de la superficie sea menor que el radio de la herramienta, el tipo de la herramienta sea inadecuado o existan superficies múltiples (Figura 3.24).
- Problemas de inconsistencia topológica en el objeto: Pueden producirse errores cuando los objetos no son consistentes topológicamente: Objetos no cerrados, autointersecciones, superficies superpuestas, etc. El modelo geométrico a utilizar debe evitar la existencia de inconsistencias.
- Problema del compensado. Este es, quizás, el problema más importante en la generación de trayectorias. Dado que en los programas de control numérico la posición que se indica es la del centro de la herramienta, es necesario calcular dónde debe encontrarse ese centro para que la herramienta mecanice el punto deseado de la superficie. Este problema no está totalmente resuelto y depende del tipo de herramienta que se utilice (Figura 3.25).
- Problema de la planificación de las trayectorias. Consiste en la planificación del orden en que deben recorrerse las trayectorias para que no se produzcan errores y para que el tiempo de mecanizado sea el menor posible.

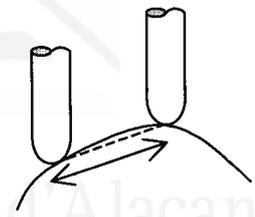


Figura 3.22
Error producido por la interpolación lineal y una distancia excesiva entre puntos sucesivos

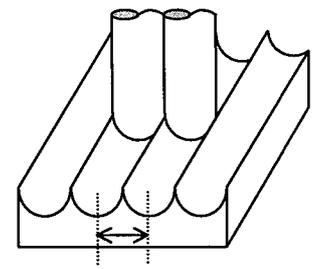


Figura 3.23
Montes y valles producidos por una distancia excesiva entre pasadas

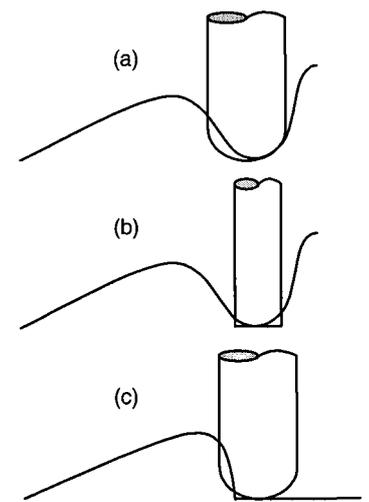
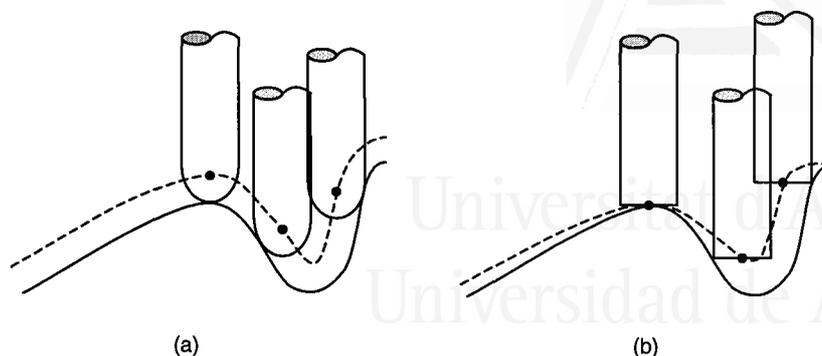


Figura 3.24
Problemas de interferencias. Producidas por una herramienta demasiado grande (a), por una herramienta de tipo inadecuado (b) y por la existencia de superficies múltiples (c)

**Figura 3.25**

Problema del compensado. Con una herramienta esférica (a) y con una herramienta cilíndrica (b)

3.2.5.1 Mecanizado a altura constante

En el mecanizado a altura constante, el problema se divide en dos etapas: cálculo del contorno de la pieza para cada uno de los planos de mecanizado y obtención de las trayectorias dentro de cada plano.

El cálculo del contorno de la pieza se aborda calculando la intersección de la misma con un plano de altura (Z) constante. El cálculo de esta intersección no es trivial y depende del modelo geométrico con que hallamos definido la pieza:

- Si se trata de un modelo poligonal, tendremos que calcular la intersección de cada polígono con el plano, obtener los puntos y aristas de corte y ordenar los resultados para obtener el contorno.
- Si es un modelo algebraico, debemos calcular la intersección igualando las ecuaciones del objeto y del plano. El sistema resultante puede no tener solución o, aunque exista, el método de resolución puede ser desconocido. Como mínimo, estos sistemas son muy complejos de resolver y suelen utilizarse métodos iterativos.
- Para la mayoría de los modelos, se pueden plantear métodos de resolución por subdivisiones sucesivas. Estos son los métodos más utilizados cuando el modelo geométrico es complejo. Este tipo de soluciones proporciona resultados aproximados y generalmente requieren gran número de costosas iteraciones.

Una vez calculado el contorno de la pieza para el plano dado, el cálculo de las trayectorias va a depender de la estrategia adoptada: mecanizados en zig-zag y en espiral.

Cálculo de las trayectorias para mecanizados en zig-zag

Los mecanizados en zig-zag utilizan métodos de *scanline*: trazan rectas paralelas que intersectan con el contorno. En ocasiones aparecen islas, es decir, zonas interiores al contorno que no deben mecanizarse. En esos casos, deben tratarse también esos contornos

interiores evitándolos y determinando cómo acceder de unas áreas a otras. Una vez calculados los tramos rectos de trayectoria, tenemos que determinar el orden en que deben seguirse para minimizar el recorrido de la herramienta y el número de retracciones de la misma. Para ello en la literatura encontramos diferentes métodos [Arkin, 2000]:

- Utilización de heurísticas para determinar el recorrido. Estos métodos suelen ser útiles sólo para determinados tipos de piezas.
- Caracterización de los diferentes casos y resolución ad-hoc de cada uno.
- Obtención de un grafo de mecanizado que modeliza las posibles conexiones entre los tramos de trayectorias, convirtiendo el problema en el clásico del Viajante de Comercio.

Cálculo de las trayectorias para mecanizados en espiral

Los mecanizados en espiral se basan en el cálculo de *offsets* (curvas paralelas) sucesivos y concéntricos. El cálculo del *offset* de un contorno es un problema complejo que se puede abordar de diversas maneras:

- Mediante la discretización de la curva, transformando el problema en el cálculo de paralelas a rectas, controlando en cada paso los bucles o autointersecciones que puedan producirse.
- Otra opción es utilizar mapas de proximidad, entre los que destaca el diagrama de Voronoi sobre los que existe numerosa bibliografía y cuyo cálculo puede abordarse mediante algoritmos Divide y Vencerás, de Propagación o *Wavefront Propagation*, etc. [Held, 1998] [O'Rourke, 1993]

3.2.5.2 Mecanizado a altura variable

El mecanizado a altura variable tiene especial relevancia en el acabado de superficies libres. En este tipo de mecanizado existe una problemática mucho más amplia que en el caso del mecanizado a altura constante y puede, en general, abordarse de dos maneras:

- Cálculo de los puntos de localización a partir de los puntos de contacto: Se parte de la trayectoria situada sobre la superficie a mecanizar, que define una serie de puntos de contacto, y posteriormente se calculan los puntos de localización, es decir, se halla la posición del centro de la herramienta para obtener los puntos de contacto sobre la superficie.
- Cálculo de los puntos de contacto a partir de los puntos de localización sobre la paralela a la superficie: En este caso se parte de la paralela u *offset* de una superficie, se sitúa la trayectoria sobre esta paralela (los puntos de localización) y se calculan los puntos de contacto.

Ninguno de estos dos problemas está resuelto completamente. Veremos algunas aproximaciones.

Mecanizado isoparamétrico

La forma más sencilla de abordar el mecanizado de una superficie libre es utilizar líneas isoparamétricas, es decir, las líneas que obtenemos al dejar uno de los dos parámetros (u,v) constante [Li, 1994]. En este caso debemos generar los puntos de contacto incrementando uno de los dos parámetros en valores discretos y dejando constante el otro parámetro.

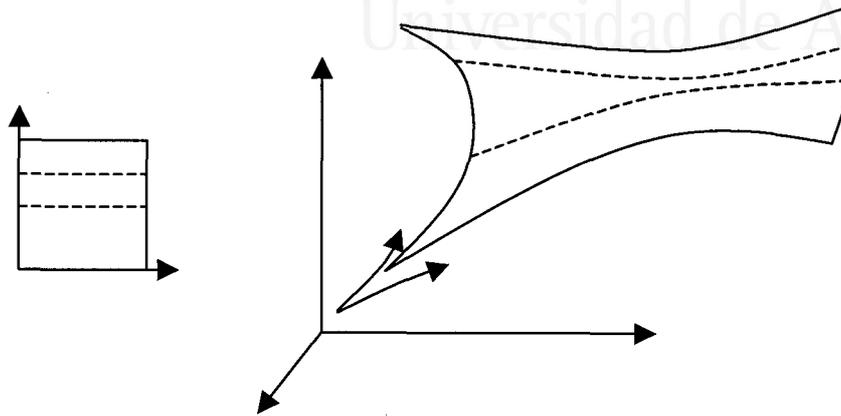


Figura 3.26
Problemas del mecanizado isoparamétrico

Aunque el principio es sencillo, llevarlo a la práctica supone tener en cuenta numerosos aspectos:

- El problema de la distancia entre puntos y pasadas, por otro lado común a todos los tipos de mecanizado.
- El hecho de que dos rectas paralelas en el espacio paramétrico no lo sean, en general, en el espacio de representación.
- El problema de las interferencias, especialmente complicado a la hora de mecanizar superficies libres.
- El problema de la intersección de superficies, generando bordes que describen curvas en el espacio paramétrico que no son paralelas a los ejes.

Mecanizado no isoparamétrico

El mecanizado no isoparamétrico tiene el problema añadido de que no disponemos de una manera sistemática y sencilla de recorrer la superficie, que permita solucionar algunos de los problemas del mecanizado isoparamétrico. En concreto, el problema fundamental que pretende solucionar este tipo de mecanizados es trazar trayectorias que sean paralelas en el espacio de representación.

Las aproximaciones son numerosas, y entre otras incluyen [Li, 1994]:

- Cálculo exacto del punto de tangencia sobre la superficie de la herramienta para hacerlo coincidir con el punto de contacto. Se suelen utilizar métodos iterativos.
- Cálculo de la superficie *offset*.

- Poligonalización de la superficie, para calcular el punto de contacto sobre esta superficie discretizada y posteriormente el punto de localización de la herramienta.
- Discretización de la superficie en un conjunto de puntos de superficie, para encontrar posteriormente la localización de la herramienta para el punto de contacto marcado por el vértice más alto.

La gran mayoría de los métodos exactos se basan en el cálculo iterativo de soluciones, con los problemas de ineficiencia y de inestabilidad numérica que presentan estos métodos. Por ejemplo, algunos de estos métodos hacen uso de algoritmos como el de Newton para el cálculo de soluciones, con lo que se introducen las siguientes restricciones:

- Para que el método converja adecuadamente es necesario que la solución inicial esté próxima a la solución correcta.
- Si hay superficies solapadas, es necesario detectar los cambios entre estas superficies para que la solución se encuentre en la adecuada.
- Las interferencias entre superficies adyacentes son difícilmente resolubles.

Un método alternativo es calcular la superficie *offset* o paralela a la superficie que queremos mecanizar, y localizar el centro de la herramienta sobre este *offset*. Sin embargo, esta superficie paralela puede contener autointersecciones que deben ser detectadas y eliminadas. Además, este método sólo es válido para herramientas esféricas, siendo su extensión a herramientas de forma general muy compleja. Algunas aproximaciones válidas son:

- Cálculo del *offset* de la superficie por discretización en polígonos, controlando los posibles bucles. Es una generalización del caso de las curvas.
- Utilización de mapas de proximidad en tres dimensiones: diagrama de Voronoi en 3D.
- Digitalización virtual [Jimeno, 2001].

Algunos sistemas comerciales utilizan una discretización de la superficies en forma de malla de polígonos. Es lo que se denomina mecanizado poliédrico. Existen en la literatura sobre algoritmos robustos de poligonalización que permiten discretizar la superficie continua con una tolerancia máxima [Li, 1994]. Estos métodos facilitan, además, el sombreado y visualización de la superficie utilizando la malla poligonal como modelo secundario.

Los métodos de discretización en conjuntos de puntos son útiles cuando el modelo discreto de la superficie sólo se va a utilizar para el mecanizado. En este caso no es necesario describir los polígonos de la malla y basta con utilizar puntos para calcular las trayectorias. Algunos de estos métodos utilizan mallas de puntos equidistantes, lo que genera problemas de precisión en las zonas de mayor curvatura. En otros casos, los puntos no son necesariamente equidistantes para asegurar una tolerancia máxima [Li, 1993].



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 4

EL MODELO SUPERFICIE- TRAYECTORIA

4.1 INTRODUCCIÓN

El modelo Superficie-Trayectoria (modelo S-T) nace como respuesta a las necesidades específicas de los sistemas CAD/CAM orientados a la pequeña industria, con el propósito de suplir las carencias que otros modelos más genéricos presentan para este tipo de sistemas.

La hipótesis de partida es la de fundamentar el diseño de los objetos tridimensionales en la definición de trayectorias por las que se hace pasar herramientas de mecanizado. De esta manera tendríamos las siguientes ventajas

- Se evita el costoso cálculo de trayectorias de mecanizado.
- La calidad del resultado final es mayor.
- Se obtienen siempre objetos que se pueden fabricar mediante una máquina de control numérico.

Desafortunadamente, un sistema de este estilo tiene unas capacidades de diseño muy limitadas.

Para aprovechar las ventajas de simplicidad, calidad y corrección de un planteamiento como el expuesto sin renunciar a procedimientos avanzados y flexibles de diseño, se introduce el concepto de máquina-herramienta virtual. Para ello, en el modelo S-T proponemos también realizar el diseño basado en trayectorias por las que se hacen pasar herramientas que denominaremos *virtuales*, con unas características muy especiales:

- Utilizaremos superficies como base para el diseño de las trayectorias, incorporando constructores específicos para facilitar esa tarea.
- El concepto de herramienta es mucho más amplio que el de una simple fresa de mecanizado: La herramienta virtual se concibe como el elemento fundamental de diseño, dejando al usuario total libertad para definir tanto su forma como su comportamiento.
- Se establecen mecanismos sencillos para emular las herramientas virtuales con herramientas reales de mecanizado y para obtener las trayectorias reales a partir de las virtuales, con lo que se asegura que la fabricación de la pieza es factible.

El objeto final que diseñemos se definirá como el resultado de mecanizar una pieza de material siguiendo las trayectorias descritas con las herramientas definidas.

En este capítulo nos aproximaremos al modelo planteado desde dos puntos de vista. En primer lugar presentaremos el modelo de manera intuitiva, introduciendo sus fundamentos y adelantando los métodos que veremos en capítulos posteriores. En segundo lugar se formaliza el modelo para darle cuerpo teórico y cohesión.

4.2 FUNDAMENTOS DEL MODELO

La base del modelo S-T, como ya hemos adelantado, son las trayectorias y herramientas virtuales. Para poderlas definir de forma sencilla y correcta el modelo contará con tres elementos constructivos fundamentales:

- Las superficies, como mecanismo para dar una forma básica al objeto y poder definir las trayectorias sobre ellas.
- Las trayectorias, trazadas en general sobre las superficies, que determinarán el recorrido que deben seguir las herramientas virtuales.
- Las herramientas virtuales que pasan por cada trayectoria y que son el mecanismo básico de diseño.

Veremos a continuación, con más detalle, cada uno de estos elementos y qué características aportan al modelo.

4.2.1 Superficies

El primer paso en el diseño de un objeto según nuestro modelo es la definición de un elemento inicial básico, formado por superficies. Estas superficies en principio son libres, y pueden ser definidas de cualquier manera.

Aunque el modo en que se describan las superficies es ajeno a lo que es el modelo en sí, para la realización del presente trabajo se han utilizado superficies de aproximación e interpolación tipo Spline, definidas por un poliedro de control y las funciones de aproximación e interpolación conocidas. Se recomienda utilizar este modelo, por ser exacto, compacto y bien conocido. Estas superficies son generadas de cuatro formas:

- Por extrusión de una curva plana, siguiendo una dirección.
- Por revolución de una curva plana (generatriz) alrededor de un eje cualquiera.
- Por cruce de dos curvas planas cualesquiera, definidas en dos planos diferentes, que determinan la frontera de la superficie.
- Por definición de sucesivos cortes de la superficie mediante curvas planas, para formar después una superficie reglada.

Además, se ha incluido la posibilidad de editar independientemente cualquier punto del poliedro que define la superficie para tener un control total de la forma. Se podrían, de igual manera, definir nuevos procedimientos de creación de superficies, según las necesidades de cada aplicación particular.

4.2.2 Trayectorias

El diseño de las superficies básicas nos permite obtener un diseño inicial del objeto. Sin embargo, el objeto no está realmente definido hasta que se han descrito las trayectorias y las herramientas.

De alguna manera el modelo planteado puede verse como una aproximación a las técnicas que utilizaría un artesano: Se esculpe una forma inicial y sobre ella se van introduciendo herramientas que arrancan el material hasta obtener el objeto apropiado. En nuestro caso pasamos herramientas siguiendo trayectorias sobre la superficie.

De nuevo, la forma en que hagamos la definición de las trayectorias es independiente del modelo. Sin embargo podemos aprovechar las formas básicas que nos proporcionan las superficies para facilitar el diseño de las trayectorias tridimensionales, que de otra manera sería muy complicado. En nuestro caso, como constructores de trayectorias, se han utilizado, con buenos resultados, la definición de curvas bidimensionales sobre un plano y la posterior proyección o despliegue de las curvas sobre la superficie tridimensional elegida. Estas operaciones suponen una problemática matemática subyacente, que ha sido resuelta mediante la utilización de algunos algoritmos ya conocidos y de otros nuevos, desarrollados para este modelo.

4.2.3 Herramientas virtuales

Las herramientas virtuales, concebidas como herramientas de diseño, nos van a permitir no sólo mecanizar las superficies iniciales de forma sencilla y precisa, sino también incorporar elementos de detalle, que es donde generalmente el diseñador plasma su creatividad. Podemos pensar en estos elementos como en formas que se esculpen sobre la superficie.

La herramienta virtual es la que dota de verdadero interés a nuestro modelo. Podemos definirla como un objeto geométrico sencillo que se utiliza para generar otros objetos nuevos más complejos. Su forma, por tratarse de una herramienta virtual, es libre, sin que sea necesaria la correspondencia con una herramienta real disponible. La forma en que esta herramienta virtual esculpe el material tampoco tiene porqué corresponderse con el funcionamiento de una herramienta real. Será en un paso posterior en el que se plantee la forma de emular la herramienta virtual con herramientas reales disponibles. Este concepto proporciona dos grandes ventajas sobre otro tipo de sistemas basados también en el mecanizado:

- Independencia del modelo con las herramientas disponibles. En un modelo basado exclusivamente en la definición de trayectorias se presenta el problema de que para modelar un mismo objeto con dos máquinas de control numérico distintas, las trayectorias que deben seguir las herramientas son diferentes. Con una herramienta virtual independizamos el diseño del objeto de la emulación de la herramienta.
- Una gama mucho mayor de posibilidades para el diseño de piezas, ya que la forma y el comportamiento de las herramientas son definidas por el usuario conforme a sus necesidades.

Sin ser un tema cerrado, se plantean dos tipos de herramientas, ilustradas en la figura 4.1:

- Herramientas de rotación: su funcionamiento es igual al de las reales: desbastan el material al girar sobre su eje. Por ello son simétricas con respecto al eje de giro.
- Herramientas de barrido: su funcionamiento no se corresponde con el de las reales pero pueden ser una buena forma de abstracción. Actúan arrancando el material mediante un barrido. Esto les permite no ser simétricas respecto a su eje longitudinal.

Para que la utilización de herramientas virtuales sea realmente útil y permita no sólo el diseño de la pieza, sino también su fabricación, es necesario que cada herramienta virtual sea definida posteriormente en función de los instrumentos reales de que se disponga. Se trata de emular cada herramienta virtual mediante herramientas reales o virtuales ya definidas. Este proceso hace que la relación entre el diseño de la pieza y el proceso de mecanizado sea muy estrecha, favoreciendo el proceso final de fabricación de la pieza. En la figura 4.2. se muestra de manera intuitiva en qué consiste la emulación de herramientas virtuales con herramientas reales.

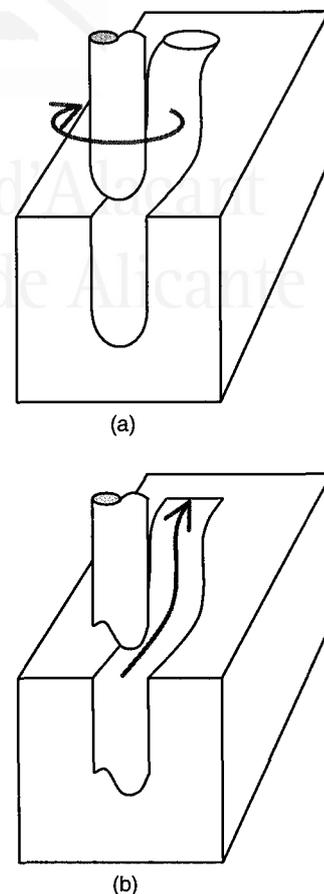


Figura 4.1
Herramientas virtuales. De rotación (a) y de barrido (b)

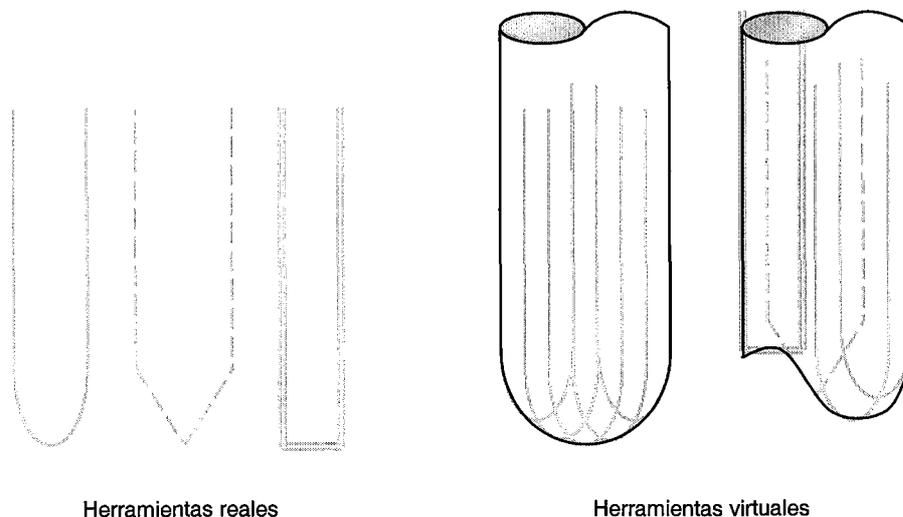


Figura 4.2
Emulación de herramientas virtuales mediante herramientas reales

4.2.4 Características del modelo

Un planteamiento como el descrito en los apartados anteriores proporciona al modelo Superficie-Trayectoria características especiales que lo distinguen de otros modelos.

En primer lugar, la definición de las superficies y de las trayectorias está basada en modelos matemáticos exactos, lo que confiere a los objetos que podamos diseñar una representación exacta. Evidentemente, como en cualquier otro modelo, el

mecanizado final mediante una máquina de control numérico y algunos procedimientos constructivos que iremos viendo suponen una discretización de las formas, lo que producirá ciertos errores que debemos acotar. Sin embargo, la representación intrínseca del objeto seguirá siendo exacta.

Por otro lado, a diferencia de otros modelos clásicos, es un modelo orientado a la fabricación y no a la visualización. Esto entraña sus ventajas y sus inconvenientes. Entre las primeras están:

- Resulta cercano a la forma habitual de trabajo del especialista en mecanizado.
- La obtención de las trayectorias de mecanizado y de las herramientas es inmediata, con lo que el resultado es de mayor calidad y más rápido.

Entre los inconvenientes se encuentran:

- Requiere el conocimiento básico de los procesos de fabricación. No obstante, no es necesario un conocimiento exhaustivo, gracias a la utilización de las herramientas virtuales.
- La visualización realista o *rendering* del objeto es costosa y supone la definición de nuevos procedimientos.

También es importante señalar que la utilización de herramientas virtuales amplía las posibilidades de diseño. Además permite la separación entre diseño y fabricación, al no limitar el uso de herramientas a las disponibles físicamente. A pesar de ello, se realiza un enlace entre los dos procesos gracias a la emulación de las herramientas virtuales con las reales, facilitando también el proceso de fabricación.

El uso de las herramientas virtuales también permite la reutilización de los diseños ya realizados: Pueden variarse las trayectorias sobre un mismo conjunto de superficies básicas; puede proyectarse una misma trayectoria sobre distintas superficies; puede recorrerse una trayectoria con herramientas diferentes; pueden utilizarse herramientas virtuales ya emuladas para emular otras más complejas, etc.

Otro de los aspectos más importantes es que se garantiza la corrección de los objetos obtenidos: Todo lo que se puede diseñar se puede fabricar, dado que nos estamos basando en trayectorias de herramientas, siempre que podamos definir un procedimiento de emulación de las herramientas virtuales con otras reales.

Por último cabe señalar que, a diferencia de otros métodos, el nuestro se basa en la eliminación de volúmenes a partir de un volumen inicial, y no en la adición de los mismos. Esta característica hace que se encuentre mucho más cerca de los procedimientos reales de fabricación.

4.3 DEFINICIÓN FORMAL DEL MODELO

Una vez descritos a grandes rasgos los fundamentos del modelo Superficie-Trayectoria vamos a aportar una definición formal que nos permita proporcionar cohesión a todos los elementos que conforman el modelo. Para ello vamos a introducir una gramática que define un lenguaje de descripción de objetos tridimensionales, de tal manera que un objeto definido en el modelo es válido si es una cadena del lenguaje definido por la gramática.

La sintaxis del lenguaje, expresada en forma de reglas de producción, permitirá determinar qué objetos están bien definidos. La semántica, descrita de forma operacional, comprenderá todas las operaciones necesarias para obtener un objeto real a partir de la cadena que lo describe.

4.3.1 Sintaxis

La sintaxis de un lenguaje permite determinar qué palabras están bien construidas y, por lo tanto, pertenecen a él. Habitualmente la sintaxis se presenta describiendo una gramática [Davis, 1994] [Llorens, 1999].

Una gramática G es una tupla $\langle \Sigma, N, P, S \rangle$ donde:

- Σ es un conjunto finito de *símbolos terminales* o *alfabeto* del lenguaje, que unidos formarán sus cadenas.
- N es un conjunto finito de *símbolos no terminales* o *variables*, que representan subcadenas de las cadenas del lenguaje.
- P es un conjunto finito de *reglas sintácticas* o *de producción* que describen cómo un símbolo no terminal se define en función de símbolos terminales y no terminales. Una regla de producción es una aplicación $p:W^* \rightarrow W^*$, donde $W = \Sigma \cup N$ y $p \in P$.
- $S \in N$ es el *símbolo inicial* o *axioma* de la gramática.

Una cadena $w \in \Sigma^*$ es generada por la gramática si se puede obtener aplicando sucesivamente reglas de producción a partir del símbolo inicial S . Al conjunto de cadenas que se pueden obtener por este método a partir de S lo llamaremos lenguaje generado por la gramática G :

$$L(G) = \{w \in \Sigma^* \mid S \rightarrow^* w\}$$

donde $S \rightarrow^* w$ indica que w se deriva de S en un número cualquiera de pasos.

La gramática propuesta $G = \langle \Sigma, N, P, S \rangle$ se presenta en la tabla 4.1 y consta de las reglas de producción (P) que ahí se describen, donde los símbolos terminales (Σ) se denotan con mayúsculas, los no terminales (N) con minúsculas y el símbolo inicial es S =objeto. El símbolo ϵ representa la cadena vacía.

```

objeto → OBJETO material cjtTrayectorias FOBJETO
material → MATERIAL punto3D punto3D FMATERIAL
cjtTrayectorias → trayectoria cjtTrayectorias | ε
trayectoria → TRAYECTORIA curva herramienta FTRAYECTORIA
curva → curva2D | curva3D
curva2D → CURVA2D cjtPuntos2D tipoCurva superficie tipoProyección
FCURVA2D
curva3D → CURVA3D cjtPuntos3D tipoCurva FCURVA3D
tipoCurva → CAPROXIMACION | CINTERPOLACION | CPOLILINEA
tipoProyección → CPROYECTADA | CDESPLEGADA
superficie → SUPERFICIE poliedro tipoSuperficie FSUPERFICIE
poliedro → cjtPuntos3D ; poliedro | ε
cjtPuntos3D → punto3D cjtPuntos3D | punto3D
cjtPuntos2D → punto2D cjtPuntos2D | punto2D
punto3D → real , real , real
punto2D → real, real
tipoSuperficie → SAPROXIMACION | SINTERPOLACIÓN | SPOLIGONAL
herramienta → HERRAMIENTA cjtSimulaciones tipoHerramienta
FHERRAMIENTA | ε
cjtSimulaciones → simulación cjtSimulaciones | ε
simulación → SIMULACIÓN herramienta punto3D FSIMULACIÓN |
SIMULACIÓN herramReal punto3D FSIMULACIÓN
herramReal → punto3D descripHerramReal
tipoHerramienta → HROTACION | HBARRIDO
descripHerramReal → HRCILINDRICA punto3D real real | HRCONICA
punto3D real real | HRESFERICA punto3D real | HRTORICA punto3D
real real

```

Tabla 4.1

Sintaxis de la gramática del modelo

Se trata de una gramática independiente del contexto o de tipo 2, por lo tanto existe un algoritmo que determina si una cadena pertenece o no al lenguaje definido por la gramática.

4.3.2 Semántica

Considerando la semántica de un lenguaje como el significado que tienen las cadenas que pertenecen a él y que una cadena define una computación sobre una máquina, podemos ver la semántica como la función computada por la cadena [Davis, 1994] [Llorens, 1999].

En el caso de un lenguaje de programación, la función computada por un programa (una cadena del lenguaje) es la salida que se obtiene

para la entrada del programa. En el caso de un lenguaje de definición de objetos como el que planteamos, la semántica es el objeto definido por la cadena del lenguaje.

La semántica de un lenguaje puede describirse utilizando diversos métodos, entre los que destacan tres: semántica operacional, semántica denotacional y semántica axiomática.

Cuando se utiliza una descripción *operacional*, la semántica del lenguaje se denota mediante operaciones en un lenguaje de una máquina abstracta. Por ejemplo, la semántica de un programa en un lenguaje de programación puede verse como el conjunto de computaciones que realiza la máquina al ejecutarse el programa para cada entrada posible, donde una computación es una secuencia finita de descripciones instantáneas, es decir, una secuencia de los estados por los que pasa el programa.

La semántica *denotacional* utiliza funciones en el sentido matemático del término para describir el significado de una cadena del lenguaje. La función asociada a la cadena se deriva por métodos matemáticos como composición de funciones más simples.

La semántica *axiomática* define el significado de las cadenas del lenguaje en términos de fórmulas lógicas, dando precondiciones (condiciones que deben cumplirse al principio) y postcondiciones (condiciones finales).

En el caso de nuestro modelo geométrico, utilizaremos un modelo operacional para describir el objeto resultado de la cadena del lenguaje. Mediante esta descripción semántica se asocia el significado que adquiere cada símbolo de la gramática por aparecer en una regla de producción u otra. Para que los símbolos puedan adquirir significados, se les asociará información geométrica y a las reglas de producción les serán añadidas acciones semánticas, que serán métodos geométricos que permitan construir el objeto.

En definitiva, veremos el proceso semántico como un conjunto de constructores geométricos que permiten obtener un objeto tridimensional en el espacio a partir de una cadena del lenguaje definido por la gramática del apartado anterior.

Las reglas de producción de la gramática del modelo podemos dividir las en cuatro grupos: las que hacen referencia al objeto, las que tratan curvas y superficies, las que se relacionan con las trayectorias y las que hacen referencia a las herramientas. Para cada grupo de reglas definiremos una serie de constructores geométricos que iremos tratando en los capítulos siguientes.



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 5

**CONSTRUCTORES DE CURVAS Y
SUPERFICIES**

5.1 INTRODUCCIÓN

La semántica asociada a las reglas de producción que permiten definir curvas y superficies viene dada por un conjunto de operadores geométricos que funcionan como constructores. Estos constructores dan lugar a superficies libres sobre las que se sitúan curvas que, después, se tomarán como base para hacer pasar por ellas a las herramientas virtuales de diseño. Por lo tanto, el proceso de diseño de un objeto en nuestro modelo parte de un conjunto de superficies básicas y de una serie de trayectorias curvas, situadas tomando como base esas superficies.

En este apartado trataremos los constructores semánticos que se han incorporado al modelo para el diseño de superficies y de curvas sobre éstas, introduciendo la notación que se ha empleado y describiendo los procedimientos y algoritmos desarrollados para permitir un uso intuitivo y sencillo del modelo.

5.2 CURVAS Y SUPERFICIES PARAMÉTRICAS

El tipo de curvas y superficies más empleado actualmente en los sistemas CAD/CAM son las curvas y superficies libres paramétricas. Se debe, fundamentalmente, a las siguientes razones:

- Su diseño es intuitivo y sencillo, porque se basan en la definición de un conjunto de puntos ordenados, por los cuales debe pasar la curva o superficie (interpolación) o que deben ser aproximados (aproximación).
- Son modelos exactos por estar descritos mediante ecuaciones y no con aproximaciones.
- Su formulación es numéricamente estable, relativamente sencilla y posibilita la visualización realista.
- Cumplen la propiedad de encontrarse definidas dentro del casco convexo formado por el polígono de control (en el caso de las curvas) o por el poliedro de control (para superficies).
- Permiten definir prácticamente cualquier forma que se desee.

En este modelo las curvas y superficies pueden ser definidas de cualquiera de las maneras conocidas para formas libres paramétricas. No obstante, para la realización del presente trabajo, se han utilizado curvas y superficies de aproximación e interpolación tipo Spline, de grado cúbico, definidas por una malla de control y las funciones de aproximación e interpolación conocidas. Se recomienda utilizar este modelo, por ser exacto, compacto y bien conocido.

Los constructores que se han incorporado para curvas 3D son:

- Proyección de una curva plana sobre una superficie.
- Despliegue de una curva plana siguiendo una superficie.

En cuanto a las superficies, son generadas de cuatro formas:

- Por extrusión de una curva plana, siguiendo una dirección.
- Por revolución de una curva plana alrededor de un eje.

- Por cruce de dos curvas planas cualesquiera.
- Por definición de secciones sucesivas mediante curvas planas.

Como facilidad adicional para el diseño, se ha incluido la posibilidad de editar independientemente cualquier punto de la malla que define la curva o la superficie para tener un control total de la forma. Se podrían, de igual manera, definir nuevos procedimientos de creación de este tipo de objetos geométricos, según las necesidades de cada aplicación particular.

En esta sección comenzaremos viendo qué restricciones planteamos para el sistema y cuál es la formulación elegida para las curvas y superficies.

5.2.1 Decisiones de diseño sobre el modelo

Planteamos inicialmente un conjunto de decisiones de diseño que nos permiten fijar un marco sobre el que desarrollar los métodos. Como veremos, estas decisiones no suponen restricciones de importancia, dado que en la mayor parte de los casos las particularizaciones son fácilmente generalizables.

- Utilizaremos superficies y curvas de grado cúbico. Esta restricción no afecta de forma importante a los sistemas CAD actuales, ya que son las de uso más extendido por sus buenas propiedades y su formulación sencilla. No obstante, es posible generalizar los diferentes métodos para superficies y curvas de cualquier grado.
- Las curvas y superficies se encuentran en forma Bézier o BSpline. Realmente, los dos modelos son equivalentes, ya que utilizan dos formulaciones distintas para un tipo de figura común. Utilizaremos la forma más adecuada en cada caso, implementando los algoritmos de conversión entre ellas.
- En general, tomaremos valores paramétricos en el intervalo $[0,1] \subset \mathcal{R}$, lo que no supone una pérdida de generalidad, bastando con una reparametrización si fuera necesaria.

5.2.2 Formulación de curvas y superficies Spline

En los sistemas CAD/CAM se conoce por curvas Spline a toda curva compuesta por secciones polinómicas que satisfacen ciertas condiciones de continuidad entre secciones consecutivas. Cuando los polinomios son cúbicos, hablamos de curvas cúbicas. Además, hablamos de curvas paramétricas cuando la formulación de las curvas se hace mediante polinomios escritos en forma paramétrica, esto es, cuando se hace una transformación o *mapping* desde un espacio de una única dimensión en el que toma valores un parámetro t , a otro espacio en dos o tres dimensiones, correspondiente al espacio cartesiano de representación.

Del mismo modo, una superficie de tipo Spline se define como dos conjuntos de curvas ortogonales de tipo Spline, de manera que el espacio paramétrico es bidimensional, y el de representación

tridimensional. Si los polinomios son cúbicos, hablaremos de superficies paramétricas bicúbicas [Hearn, 1995].

Las curvas y superficies Spline tienen diferentes formulaciones, dependiendo del tipo de polinomio que utilizemos para definir las. De entre todas ellas, destacan las curvas y superficies Bézier, BSpline y NURBS. Los dos primeros modelos representan la misma familia de curvas y superficies, pudiendo pasar de una representación a otra mediante un algoritmo sin pérdidas [Farin, 1993], [Hearn, 1995]. Las curvas y superficies NURBS son una generalización de las BSpline no existiendo, en general, algoritmos de conversión entre los modelos.

En el presente trabajo se han elegido las representaciones BSpline y Bézier. Para el diseño se ha optado por la formulación BSpline, debido a las siguientes características [Hearn, 1995]:

- El grado del polinomio se puede establecer de manera independiente de la cantidad de puntos de control.
- Esta representación permite un control local sobre la forma.
- La curva o superficie se trata de manera global, con un polinomio único, no necesitando el manejo de secciones por separado.

Por el contrario, para algunos cálculos, se ha utilizado la representación Bézier. Esto no supone pérdida de capacidad, puesto que son modelos equivalentes. Por el contrario, la formulación Bézier trabaja por secciones independientes, cada una de ellas representada por un polinomio de grado cúbico o bicúbico (según hablemos de curvas o superficies, respectivamente), y todos los polinomios homogéneos, es decir, del mismo tipo. Esta característica facilita enormemente los cálculos.

Independientemente del modelo de representación elegido, las curvas cúbicas paramétricas se representan de la siguiente manera: Sea $C(t)$ una curva en forma paramétrica, con un parámetro t definido sobre un espacio \mathfrak{R} (en lo sucesivo, espacio paramétrico), y cuya imagen (x,y,z) se encuentra definida en un espacio \mathfrak{R}^3 , que llamaremos espacio de representación. Sea $P=\{p_1, p_2, \dots, p_n\}$ un conjunto de puntos tridimensionales en el espacio de representación, que vamos a llamar puntos de control. Las curvas se definen a partir de estos puntos y de una serie de polinomios denominados funciones de combinación o *blend functions*. Estas funciones y la forma en que se combinan con los puntos de control cambian de un modelo a otro y son las que caracterizan a la formulación utilizada.

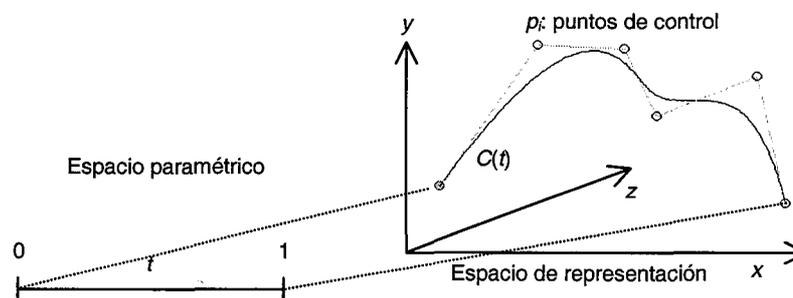


Figura 5.1
Curva paramétrica con los espacios paramétrico y de representación

Con respecto a las superficies bicúbicas paramétricas, tienen una formulación que es una ampliación a dos dimensiones del modelo de las curvas: Definimos una superficie paramétrica como una función $S(u,v)$, con parámetros (u,v) definidos en este caso sobre un espacio paramétrico \mathbb{R}^2 , y cuya imagen (x,y,z) está en el espacio de representación \mathbb{R}^3 . Como en el caso de las curvas se define un conjunto de puntos de control tridimensionales, en esta ocasión dispuestos en forma de malla $P=\{p_{00},p_{01},\dots,p_{mn}\}$, y una serie de polinomios de combinación iguales a los de las curvas.

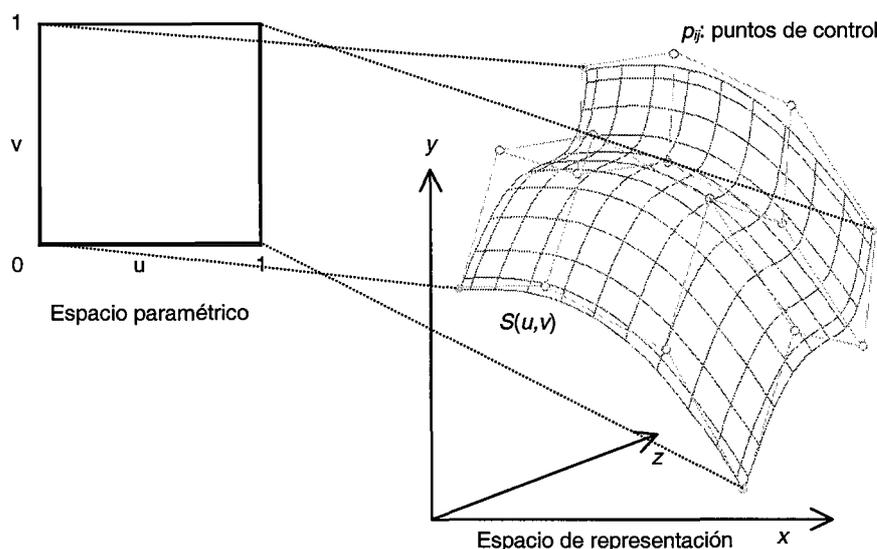


Figura 5.2

Superficie paramétrica con los espacios paramétrico y de representación

5.2.2.1 Curvas y superficies Bézier

Sea un conjunto de $n+1$ puntos de control tridimensionales $P=\{p_0,p_1,\dots,p_n\}$. Podemos definir una curva Bézier de grado n en función de un parámetro t de la siguiente manera

$$C(t) = \sum_{k=0}^n p_k B_{k,n}(t) \quad \text{Ecuación 5.1}$$

con $0 \leq t \leq 1$ y donde $B_{k,n}(t)$ son polinomios de combinación. En concreto, los polinomios de combinación que se utilizan son los polinomios de Bernstein, que se calculan de la siguiente manera:

$$B_{k,n}(t) = \frac{n!}{k!(n-k)!} t^k (1-t)^{n-k} \quad \text{Ecuación 5.2}$$

En el caso de las curvas Bézier cúbicas (grado del polinomio $n=3$, descritas mediante 4 puntos de control), es posible obtener una notación equivalente en forma matricial, que es la que se utiliza habitualmente [Watt, 1992]. En este caso, la curva $C(t)$ se describe como:

$$C(t) = T \cdot M_{\text{Bézier}} \cdot P \quad \text{Ecuación 5.3}$$

donde T es un vector fila $T=[t^3 \ t^2 \ t \ 1]$, $M_{\text{Bézier}}$ la matriz característica de las curvas Bézier, que tiene orden 4×4 , y P es un vector columna con los cuatro puntos de control $P^T=[p_0 \ p_1 \ p_2 \ p_3]$. La matriz $M_{\text{Bézier}}$ tiene la forma:

$$M_{\text{Bézier}} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{Ecuación 5.4}$$

En cuanto a las superficies Bézier, son una generalización de las curvas, teniendo la siguiente formulación: Sea ahora un conjunto de puntos de control $p_{j,k}$ dispuestos en forma de malla cuadrada de tamaño $(m+1) \times (n+1)$ puntos. La superficie se define en función de dos parámetros u y v :

$$S(u, v) = \sum_{j=0}^m \sum_{k=0}^n p_{j,k} B_{j,m}(v) B_{k,n}(u) \quad \text{Ecuación 5.5}$$

donde $0 \leq u \leq 1$ y $0 \leq v \leq 1$, y los polinomios de combinación B son los polinomios de Bernstein ya descritos.

En una superficie Bézier, dejando el valor de uno de los parámetros constante y dando valores al otro parámetro obtenemos una curva Bézier, situada sobre la superficie. Así, las cuatro curvas de la frontera de una superficie Bézier se obtienen para valores constantes de $u=0$, $u=1$, $v=0$ y $v=1$.

Como pasa con las curvas, para superficies Bézier bicúbicas se suele utilizar una notación matricial equivalente:

$$S(u, v) = \mathbf{U} \cdot M_{\text{Bézier}} \cdot \mathbf{P} \cdot M_{\text{Bézier}}^T \cdot \mathbf{V} \quad \text{Ecuación 5.6}$$

donde \mathbf{U} es un vector fila $\mathbf{U}=[u^3 \ u^2 \ u \ 1]$, \mathbf{V} un vector columna $\mathbf{V}^T=[v^3 \ v^2 \ v \ 1]$, \mathbf{P} una matriz cuadrada de orden 4×4 que contiene los 16 puntos de control $p_{j,k}$ y $M_{\text{Bézier}}$ la matriz característica de Bézier que hemos descrito para las curvas.

Las principales propiedades de curvas y superficies Bézier son:

- Las curvas (superficies) interpolan los puntos de control extremos p_0 y p_n ($p_{0,0}$, $p_{0,n}$, $p_{m,0}$ y $p_{m,n}$).
- La tangente a la curva (superficie) en los puntos extremos viene dada por la línea (plano) que une el punto extremo con el punto de control contiguo. Esto hace que sea muy sencillo conseguir la continuidad entre secciones con sólo alinear el primer tramo de cada sección con el último de la sección anterior.
- Toda curva (superficie) Bézier se encuentra contenida dentro del casco convexo que definen sus puntos de control. Esto hace que pueda utilizarse el casco convexo como paso previo en numerosos cálculos: intersecciones, proyecciones, pruebas de interioridad/exterioridad, etc.

5.2.2.2 Curvas y superficies BSpline

La definición de una curva o superficie BSpline es muy similar a la de las curvas y superficies Bézier.

En el caso de la curvas partimos también de un conjunto de $n+1$ puntos de control tridimensionales $P=\{p_0, p_1, \dots, p_n\}$. La curva se define también en función de un parámetro t de la siguiente manera

$$C(t) = \sum_{k=0}^n p_k B_{k,d}(t) \quad \text{Ecuación 5.7}$$

pero ahora $t_{\min} \leq t \leq t_{\max}$ y $2 \leq d \leq n+1$. $B_{k,d}(t)$ siguen siendo los polinomios de combinación, pero ahora se definen de manera diferente, mediante la fórmula recursiva de Cox-deBoor. Además, estas funciones dependen del valor de d , que marca el grado de los polinomios:

$$B_{k,1}(t) = \begin{cases} 1, & \text{si } t_k \leq t \leq t_{k+1} \\ 0, & \text{en otro caso} \end{cases} \quad \text{Ecuación 5.8}$$

$$B_{k,d}(t) = \frac{t - t_k}{t_{k+d-1} - t_k} B_{k,d-1}(t) + \frac{t_{k+d} - t}{t_{k+d} - t_{k+1}} B_{k+1,d-1}(t)$$

Obsérvese, por otro lado, que el parámetro t se define en varios intervalos, marcados por pares de valores t_k y t_{k+1} . Esos valores, entre los que se produce una transición de un intervalo de definición del parámetro a otro, se denominan nodos. Podemos colocar estos valores en forma de vector, de manera que $t_{\min} = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_{n+d} = t_{\max}$, denominado vector de nodos. Cada nodo indica una transición entre secciones sucesivas de la curva.

También es posible utilizar una notación matricial para el caso de curvas BSpline cúbicas. En este caso $d=4$ y cada sección de la curva viene determinada por cuatro puntos de control. Cada sección i puede representarse mediante la siguiente expresión:

$$C(t) = T \cdot M_{BSpline} \cdot P \quad \text{Ecuación 5.9}$$

con T un vector fila $T=[t^3 \ t^2 \ t \ 1]$, $M_{BSpline}$ la matriz 4×4 característica de las curvas BSpline, y P es un vector columna con los cuatro puntos de control que afectan a esa sección $P^T=[p_i \ p_{i+1} \ p_{i+2} \ p_{i+3}]$. La matriz $M_{BSpline}$ es:

$$M_{BSpline} = \frac{1}{6} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad \text{Ecuación 5.10}$$

Las superficies BSpline son una extensión de las curvas del mismo tipo. La función que las define es:

$$S(u, v) = \sum_{j=0}^m \sum_{k=0}^n p_{j,k} B_{j,e}(v) B_{k,d}(u) \quad \text{Ecuación 5.11}$$

con $0 \leq u \leq 1$ y $0 \leq v \leq 1$, y los polinomios de combinación B los que ya hemos definido. Además, para el caso de las superficies, los puntos de control $p_{i,k}$ forman una malla cuadrada de dimensiones $(m+1) \times (n+1)$ puntos, y los valores e y d representan el grado de los polinomios en sentido u y v de los parámetros respectivamente.

Como ocurre con la notación Bézier, en una superficie BSpline obtenemos curvas sobre la superficie también en notación BSpline, dejando uno de los parámetros constante y dando valores al otro. De igual manera, obtenemos las cuatro curvas de la frontera dando los valores constantes $u=0$, $u=1$, $v=0$ y $v=1$.

En una superficie BSpline bicúbica e y d valen 4 y cada sección de la superficie viene determinada por una malla de 4×4 puntos de control. En este caso es fácil obtener una notación matricial como la siguiente para cada sección ij :

$$S(u, v) = \mathbf{U} \cdot \mathbf{M}_{BSpline} \cdot \mathbf{P} \cdot \mathbf{M}_{BSpline}^T \cdot \mathbf{V} \quad \text{Ecuación 5.12}$$

\mathbf{U} es un vector fila $\mathbf{U} = [u^3 \ u^2 \ u \ 1]$, \mathbf{V} un vector columna $\mathbf{V}^T = [v^3 \ v^2 \ v \ 1]$ y \mathbf{P} una matriz cuadrada de orden 4×4 que contiene los puntos de control que afectan a la sección ij , dispuestos de la siguiente manera:

$$\mathbf{P} = \begin{bmatrix} p_{i,j} & p_{i,j+1} & p_{i,j+2} & p_{i,j+3} \\ p_{i+1,j} & p_{i+1,j+1} & p_{i+1,j+2} & p_{i+1,j+3} \\ p_{i+2,j} & p_{i+2,j+1} & p_{i+2,j+2} & p_{i+2,j+3} \\ p_{i+3,j} & p_{i+3,j+1} & p_{i+3,j+2} & p_{i+3,j+3} \end{bmatrix} \quad \text{Ecuación 5.13}$$

$\mathbf{M}_{BSpline}$ es la matriz característica de BSpline que ya hemos visto.

Las características que diferencian a las curvas y superficies de tipo BSpline son:

- El grado del polinomio es independiente del número de puntos de la curva (superficie): En consecuencia, la curva (superficie) no se define por tramos, sino que su ecuación da lugar a una única figura, independientemente del número de puntos de control que definamos.
- Aunque no existen tramos como tal, sí podemos determinar secciones de la curva (superficie), cuya transición viene marcada por los nodos. En cualquier caso, la continuidad entre secciones está asegurada por la forma en que se define el polinomio.
- Permiten modificaciones locales, es decir, el cambio en uno de los puntos de control afecta solamente a un pequeño número de secciones de la curva (superficie). Este número de secciones depende del grado de la figura.

Su formulación es más complicada que la de las Bézier, por lo que convertiremos las figuras a este modelo para ciertos cálculos complejos.

5.2.2.3 Conversión entre representaciones

En ocasiones es muy útil poder cambiar de una representación de Spline a otra. En nuestro caso, las curvas BSpline se adaptan muy bien a las necesidades de diseño que tenemos, pero su formulación es demasiado complicada para ciertos cálculos. En estos casos se adapta mejor el modelo Bézier, que maneja polinomios cúbicos y bicúbicos.

Por simplicidad, vamos a partir de la notación matricial de las curvas. Sea una curva $C(t)$ descrita con una formulación genérica de un tipo $Spline_1$ que queremos convertir a un segundo tipo $Spline_2$. Sea la notación matricial de la curva en el primer tipo de representación de la siguiente manera:

$$C(t) = T \cdot M_{Spline_1} \cdot P_{Spline_1} \quad \text{Ecuación 5.14}$$

donde T es el vector fila conocido $T=[t^3 \ t^2 \ t \ 1]$, que es común a todas las notaciones, M_{Spline_1} es la matriz característica de la notación $Spline_1$ y P_{Spline_1} es el vector columna en que se disponen los puntos de control de la manera que la notación $Spline_1$ necesita.

La misma curva en la notación $Spline_2$ se describe, de manera análoga, con las matrices M_{Spline_2} y P_{Spline_2} adaptadas a las características de esta notación:

$$C(t) = T \cdot M_{Spline_2} \cdot P_{Spline_2} \quad \text{Ecuación 5.15}$$

Si queremos que ambas notaciones representen la misma curva, debemos igualar la expresiones, obteniendo:

$$\begin{aligned} T \cdot M_{Spline_1} \cdot P_{Spline_1} &= T \cdot M_{Spline_2} \cdot P_{Spline_2} \\ P_{Spline_2} &= M_{Spline_2}^{-1} \cdot M_{Spline_1} \cdot P_{Spline_1} \end{aligned} \quad \text{Ecuación 5.16}$$

La matriz que permite convertir la representación $Spline_1$ en $Spline_2$ la representaremos $M_{Spline_1 \rightarrow Spline_2}$ y es:

$$M_{Spline_1 \rightarrow Spline_2} = M_{Spline_2}^{-1} \cdot M_{Spline_1} \quad \text{Ecuación 5.17}$$

En el caso de las transformaciones entre representación Bézier y BSpline bicúbicas y viceversa, estas matrices son:

$$\begin{aligned} M_{BSpline \rightarrow Bézier} &= \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1} \cdot \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 4 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix} \\ M_{Bézier \rightarrow BSpline} &= \left(\frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 6 & -7 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -7 & 6 \end{bmatrix} \end{aligned}$$

5.3 CONSTRUCTORES DE SUPERFICIES

Como hemos visto en el apartado anterior, el diseño de una curva o una superficie se realiza partiendo de un conjunto de puntos de control que debe definir el usuario. Estos puntos son los que nos permiten dar la forma deseada a las figuras dibujadas. La definición directa de los puntos de control es útil en el caso de las curvas, pero no así cuando tratamos con superficies, ya que los puntos se encuentran formando una malla tridimensional de edición muy complicada. Hemos de pensar que en una aplicación de diseño se cuenta con una pantalla plana, por lo que para dar profundidad a un punto hemos de manejar vistas diferentes u otros métodos artificiales poco intuitivos. Por esa razón, para el diseño de superficies se suele partir de una o varias curvas que se toman como base para definir una superficie.

En este apartado veremos algunas técnicas que permiten diseñar superficies libres de manera muy sencilla: superficies por extrusión, por revolución, por cruce y por secciones.

5.3.1 Extrusión de curvas planas

El método más sencillo de obtención de una superficie es por extrusión de una curva plana. El concepto intuitivo es muy sencillo: dibujamos una curva plana que se desplaza en el espacio siguiendo una recta y que forma, de esta manera, una superficie.

De manera más formal, sea $C(u)$ una curva paramétrica Bézier o BSpline definida en un espacio paramétrico unidimensional, $u \in [0,1] \subset \mathcal{R}$, y con un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$. Podemos obtener una extrusión de la curva para generar una superficie, incorporando una tercera dimensión al espacio de representación, marcada por una recta también en forma paramétrica $R(v)$, $v \in [0,1] \subset \mathcal{R}$, de manera que $C(u)$ determina la forma de la superficie en el sentido del parámetro u y $R(v)$ en el sentido de v .

Para que la notación sea homogénea, hemos de definir la recta con la misma formulación que la curva. Para ello basta recordar que tanto en el caso de las curvas Bézier como BSpline, cuatro puntos de control colineales dan lugar a una recta.

La superficie obtenida será del mismo tipo que las curvas. Tendrá un espacio paramétrico bidimensional $(u,v) \in [0,1] \times [0,1] \subset \mathcal{R}^2$, y un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$.

5.3.2 Revolución de una curva alrededor de un eje

Otro método habitual para la obtención de superficies es la revolución de una curva plana alrededor de un eje. Para ello hemos de definir una curva denominada generatriz que gira alrededor de un eje de revolución, dando lugar de esta manera a una superficie.

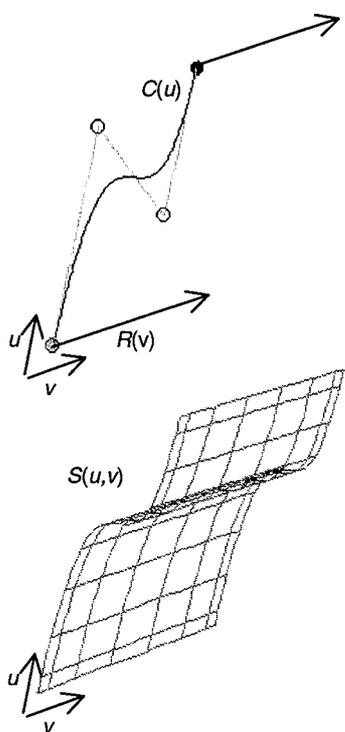


Figura 5.3
Superficie generada por
extrusión de una curva plana

En este caso partimos de una generatriz formada por una curva paramétrica $G(u)$ definida en un espacio paramétrico unidimensional, $u \in [0,1] \subset \mathcal{R}$, y con un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$. Para realizar la rotación definimos una circunferencia cuyo radio es la distancia del eje de rotación al extremo de la generatriz y que está situada en un plano normal al eje de revolución, de tal manera que el eje pasa por el centro de la circunferencia. El problema que aparece en este caso, es que no es posible representar circunferencias exactas utilizando curvas BSpline o Bézier. Sin embargo, si el número de puntos de control es suficiente, la aproximación es lo bastante buena como para bastar en los casos más exigentes [Farin, 1995], por lo que definimos la circunferencia como una curva $C(v)$ con una parametrización $v \in [0,1] \subset \mathcal{R}$ y que es una aproximación suficientemente buena. La superficie obtenida tendrá un espacio paramétrico bidimensional $(u,v) \in [0,1] \times [0,1] \subset \mathcal{R}^2$, y un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$.

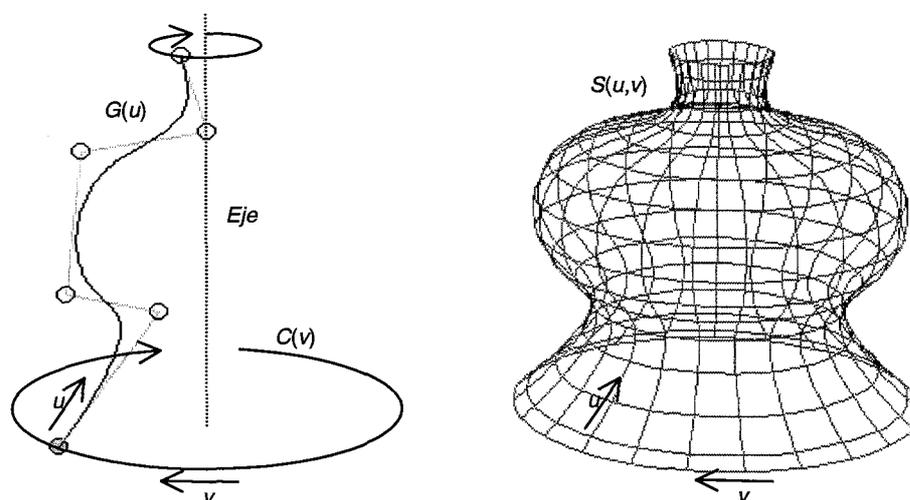


Figura 5.4
Superficie generada por revolución de una curva plana alrededor de un eje

5.3.3 Cruce de curvas planas

El tercer método de modelado de superficies parte de dos curvas planas definidas en dos planos diferentes, que determinan la frontera de la superficie. Obsérvese que se trata de una extrusión generalizada de una curva siguiendo la trayectoria de otra. Cada curva marca la forma de la superficie en su dirección, mientras que la posición viene dada por la primera curva.

Así, sean $C_1(u)$ y $C_2(v)$ dos curvas paramétricas definidas en sus espacios paramétricos unidimensionales, u y $v \in [0,1] \subset \mathcal{R}$, y ambas con un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$. Podemos generar una superficie por extrusión generalizada, cruzando las dos curvas, de manera que la superficie obtenida tendrá un espacio paramétrico bidimensional $(u,v) \in [0,1] \times [0,1] \subset \mathcal{R}^2$, y un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$.

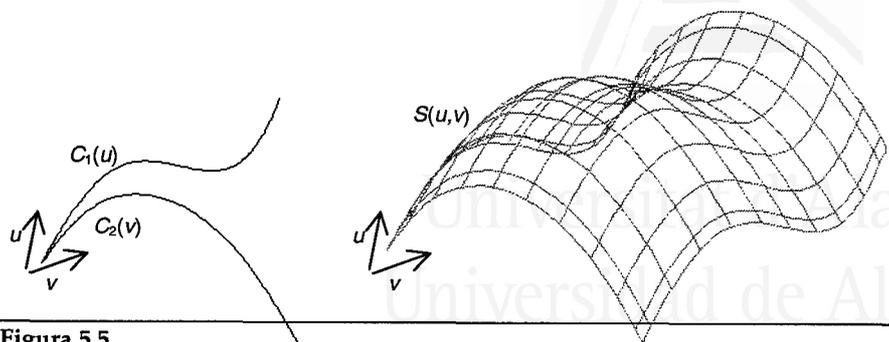


Figura 5.5
Superficie generada por cruce de dos curvas planas

5.3.4 Definición por secciones

El último método de creación de superficies propuesto es la definición mediante secciones. En este caso partimos de un conjunto de curvas ordenadas, que conforman las secciones sucesivas de la superficie que queremos diseñar.

Sea un conjunto ordenado de n curvas paramétricas $C=\{C_i(u)\}$, $1 \leq i \leq n$, definida cada una en su espacio paramétrico unidimensional, $u \in [0,1] \subset \mathcal{R}$, y todas con un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$. Sea, además un vector $V = [v_1, v_2, \dots, v_n]$ de n valores en el intervalo $[0,1]$. Podemos generar una superficie utilizando las curvas del conjunto C como secciones, haciendo que la superficie adopte el perfil de la curva $C_i(u)$ para un valor del parámetro $v=v_i$. La superficie obtenida tendrá un espacio paramétrico bidimensional $(u,v) \in [0,1] \times [0,1] \subset \mathcal{R}^2$, y un espacio de representación tridimensional $(x,y,z) \in \mathcal{R}^3$.

Puesto que para la representación que utilizamos la malla de puntos de control debe ser topológicamente cuadrada, es necesario que el número de puntos de control de cada curva C_i sea el mismo. Esta restricción supone que en el caso de distinto número de puntos de control debemos recurrir a algoritmos de incremento de puntos de control para curvas paramétricas [Farin, 1995].

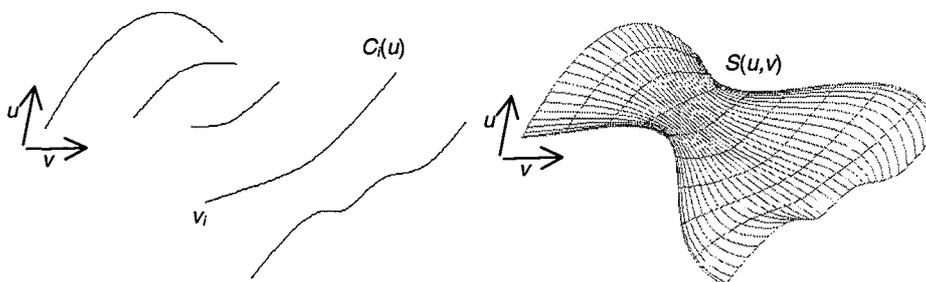


Figura 5.6
Superficie generada a partir de secciones definidas por curvas planas

5.4 CONSTRUCTORES DE CURVAS SOBRE SUPERFICIES

En un sistema como el planteado, debemos disponer de herramientas que nos permitan colocar curvas sobre las superficies diseñadas. Estas curvas nos servirán para definir trayectorias sobre las que hacer pasar las herramientas virtuales.

La obtención de una curva situada sobre una superficie debe facilitarse de alguna manera, ya que resulta prácticamente imposible conseguir de manera manual que una curva se asiente perfectamente sobre la superficie. Presentamos dos mecanismos de fácil uso para conseguir este propósito: Proyección y despliegue de curvas sobre superficies.

Como paso previo, proponemos un método de discretización de curvas, necesario para poder definir los dos constructores de curvas sobre superficies.

5.4.1 Discretización de curvas

Vamos a definir discretización de una curva como la obtención de una polilínea que se aproxime a la curva con una determinada precisión, de tal manera que pueda sustituirla en determinados procedimientos en los que no es posible tratar con curvas continuas. Por lo tanto, en estos métodos es fundamental que el error esté acotado, para asegurar que la polilínea sea lo más cercana posible a la curva original.

Proponemos un método adaptativo basado en la curvatura de la curva. Sea $C(t)$ la curva en forma Bézier que queremos discretizar. Si la curva estuviera en forma BSpline la conversión a forma Bézier no presenta problemas como ya hemos visto. Su discretización es una polilínea o un conjunto $S=\{s_1, s_2, \dots, s_n\}$ de n segmentos rectos ordenados, de tal manera que la polilínea S aproxima a la curva $C(t)$ con un error máximo en cada tramo e_{max} .

Puesto que $C(t)$ es una curva polinómica es muy sencillo calcular las tangentes en cada punto. Para acotar el error máximo e_{max} permitido consideramos la diferencia entre las tangentes en dos puntos consecutivos y fijamos una diferencia angular máxima entre ambas α_{max} .

El algoritmo propuesto se basa en el cálculo inicial de un segmento de longitud paramétrica fija (Δt_{max}) y la posterior subdivisión de éste hasta que el error cometido sea menor que el máximo permitido (e_{max}) o la longitud paramétrica del segmento menor que un valor mínimo (Δt_{min}). En la tabla 5.1 se presenta el algoritmo.

<p>ENTRADA $C(t)$: Curva a discretizar e_{\max}: error máximo permitido en cada tramo</p> <p>SALIDA S: polilínea o conjunto de segmentos rectos que aproxima $C(t)$</p> <p>VARIABLES p_0, p_1: Puntos</p> <p>ALGORITMO</p> <pre> t=0 p₀=C(t) S.Añadir(p₀) Δt=Δt_{max} Mientras t<1 p₀=C(t) p₁=C(t+Δt) Si (error(C,p₀,p₁)<e_{max} o Δt<Δt_{min}) entonces S.Añadir(p₁) Δt=Δt_{max} t=t+Δt Si no Δt=Δt/2 Fin Si Fin Mientras p₁=C(1) S.Añadir(p₁) </pre>

Tabla 5.1

Algoritmo para la discretización de curvas en forma de polilíneas

5.4.2 Proyección de curvas sobre superficies

Podemos ver la proyección de una curva sobre una superficie como la curva resultante de la intersección de la superficie con una superficie por extrusión obtenida a partir de la curva en la dirección normal a su plano de definición. Alternativamente, también se podría definir como la proyección paralela ortogonal de la curva sobre la superficie, de ahí que hablemos de proyección.

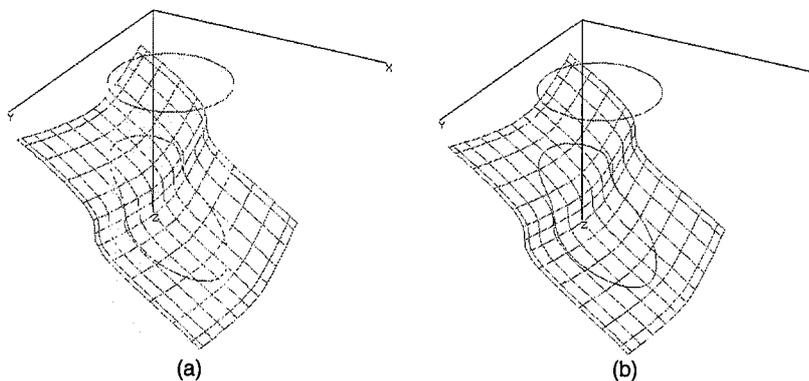


Figura 5.7

Proyección de una curva sobre una superficie. Puede verse como la intersección de dos superficies (a) o como la proyección paralela ortogonal de la curva sobre la superficie (b)

La proyección de curvas planas sobre superficies libres entraña una serie de dificultades que describimos a continuación [Molina, 1997]. Para simplificar el problema, se ha tomado la decisión de trabajar a nivel interno con superficies Bézier por dos razones:

- Su manejo separado de las secciones facilita mucho los cálculos.
- No introduce restricciones en cuanto al diseño puesto que, como hemos visto, la conversión de forma BSpline a forma Bézier es inmediata.

También vamos a suponer que la curva a proyectar se ha definido sobre uno de los planos coordenados. En el caso de estar definida sobre otro plano, bastará con realizar las transformaciones pertinentes para hacerlo coincidir con uno de los coordenados.

Sea, por lo tanto, una superficie Bézier bicúbica paramétrica $S(u,v)=(x,y,z)$. Estas superficies se definen en un espacio paramétrico (u,v) para obtener, a partir de su formulación, el punto tridimensional correspondiente en el espacio de representación (x,y,z) .

Sea también una curva cúbica paramétrica $C(t)=(x,y,z)$, también en forma Bézier, sin perder generalidad como en el caso anterior. Esta función, de igual manera, representa el paso de un espacio paramétrico t al de representación (x,y,z) . La curva se encuentra sobre uno de los planos coordenados, por lo que una de las coordenadas (x,y,z) será siempre 0.

Con estas definiciones podemos observar que mientras los espacios paramétricos de las figuras anteriores (o de cualquier figura que se defina) son diferentes entre sí, los espacios de representación son los mismos. Como consecuencia, cualquier figura puede intersectar con otra en el espacio (x,y,z) sin que en los espacios paramétricos se pueda detectar esto. Por lo tanto, todo cálculo relacionado con la intersección de figuras ha de realizarse sobre el espacio de representación. Sin embargo, los cálculos necesarios para la visualización se realizan sobre el espacio paramétrico. Este hecho entraña algunas complicaciones adicionales.

La situación planteada es la siguiente: Tenemos un conjunto de figuras (superficies, curvas o cualquier otra) definidas cada una de ellas en su espacio paramétrico propio, pero cuyo espacio de representación es común. Las interacciones entre dichas figuras sólo es posible tratarlas en el espacio de representación, mientras que los cálculos para cada figura se realizan en su propio espacio paramétrico. Para facilitar la utilización conjunta de ambos espacios habrá que utilizar algún tipo de mecanismo que nos permita el paso no sólo de los espacios paramétricos al de representación, sino también al revés. La transformación de los espacios paramétricos t -para curvas- y (u,v) -para superficies- al espacio común de representación (x,y,z) es inmediata, sin más que aplicar las formulaciones conocidas para curvas y superficies Bézier. Sin embargo, el paso del espacio de representación a cada uno de los espacios paramétricos requiere invertir el proceso. En los apartados siguientes veremos como realizar estos pasos.

5.4.2.1 La transformación del espacio (x,y,z) al t

Sea la curva paramétrica $C(t)=(x,y,z)$, de grado cúbico y calculada a partir de un conjunto de puntos de control $P=\{p_i\}$. Como sabemos, la formulación en forma paramétrica de una curva cúbica Bézier es:

$$C(t) = T \cdot M_{\text{Bézier}} \cdot P \quad \text{Ecuación 5.18}$$

donde T es el vector fila $T=[t^3 \ t^2 \ t \ 1]$, $M_{\text{Bézier}}$ la matriz 4x4 característica de las curvas Bézier y P es un vector columna con los cuatro puntos de control $P^T=[p_0 \ p_1 \ p_2 \ p_3]$. Operando esta expresión, obtenemos la ecuación polinómica:

$$C(t) = at^3 + bt^2 + ct + d \quad \text{Ecuación 5.19}$$

donde a , b , c y d son coeficientes vectoriales que se pueden calcular a partir de los puntos de control p_i . La fórmula anterior da lugar a tres ecuaciones cúbicas, con una incógnita t , resolubles mediante métodos algebraicos o numéricos. Cada una de las ecuaciones tendrá tres soluciones, reales o complejas.

El valor del parámetro t para un punto (x,y,z) del espacio de representación será aquel punto o puntos, en el caso de que existan, que cumplan:

- Es una solución real común a las tres ecuaciones.
- Su valor se encuentra en el intervalo de definición del parámetro (en nuestro caso entre 0 y 1).

5.4.2.2 La transformación del espacio (x,y,z) al (u,v)

Sea la superficie paramétrica $S(u,v)=(x,y,z)$ definida sobre un espacio paramétrico (u,v) y un espacio de representación (x,y,z) , y calculada a partir de un conjunto de puntos de control p_{ij} dispuestos en forma de malla. Podemos escribirla como:

$$S(u,v) = U \cdot M_{\text{Bézier}} \cdot P \cdot M_{\text{Bézier}}^T \cdot V \quad \text{Ecuación 5.20}$$

donde $U=[u^3 \ u^2 \ u \ 1]$ es un vector fila, $V^T=[v^3 \ v^2 \ v \ 1]$ es un vector columna, P la matriz cuadrada de orden 4x4 que contiene los 16 puntos de control $p_{j,k}$ y $M_{\text{Bézier}}$ la matriz característica de Bézier. Operando obtenemos la siguiente ecuación vectorial

$$\begin{aligned} S(u,v) = & a_{00} u^3 v^3 + a_{01} u^3 v^2 + a_{02} u^3 v + a_{03} u^3 + \\ & a_{10} u^2 v^3 + a_{11} u^2 v^2 + a_{12} u^2 v + a_{13} u^2 + \\ & a_{20} u v^3 + a_{21} u v^2 + a_{22} u v + a_{23} u + \\ & a_{30} v^3 + a_{31} v^2 + a_{32} v + a_{33} = (x,y,z) \quad \text{Ecuación 5.21} \end{aligned}$$

donde a_{ij} son coeficientes vectoriales que se pueden calcular a partir de los puntos de control p_{ij} del poliedro de control. Esto da lugar a tres ecuaciones bicúbicas con dos incógnitas. Podemos plantear, por lo tanto tres sistemas de ecuaciones con dos incógnitas cada uno (los correspondientes a las componentes XY, XZ e YZ). Estos sistemas no son resolubles mediante métodos algebraicos, por lo que hemos de recurrir a métodos iterativos aproximados (por ejemplo, Newton-

Raphson, que es el que se ha implementado). Así se obtienen tres conjuntos de tres soluciones cada uno de ellos.

Los valores de los parámetros (u,v) para un punto (x,y,z) del espacio de representación, serán aquellos pares (u,v) que cumplan:

- Son solución real común a los tres sistemas.
- Su valor se encuentra en el intervalo de definición de los parámetros (típicamente entre 0 y 1).

5.4.2.3 Proyección de una curva sobre una superficie

Sea $C(t)$ una curva cúbica paramétrica y $S(u,v)$ una superficie bicúbica paramétrica. La curva ha sido definida sobre un plano: sin pérdida de generalidad, supongamos que es el XY ; si se utiliza otro plano, basta con hacer las transformaciones necesarias para llevar dicho plano al plano XY . La superficie $S(u,v)$, por su lado, se encuentra definida sobre el espacio (x,y,z) .

Para proyectar la superficie debemos, en primer lugar, calcular el valor (x_i, y_i) para cada valor t_i del parámetro t , puesto que la curva está definida para el plano XY . No hay más que aplicar las ecuaciones de cálculo de las curvas paramétricas (Bézier). Para evitar una pérdida de precisión importante, la obtención de los sucesivos valores t_i del parámetro debe realizarse según el método de discretización descrito en el apartado 5.4.1 de este capítulo.

A continuación, a partir de cada punto (x_i, y_i) del espacio de representación, tenemos que obtener el valor (u_i, v_i) del espacio paramétrico de la superficie correspondiente a dicho punto. Para ello, hay que aplicar la transformación detallada en la sección anterior (5.4.2.2), pero con una particularidad: al ser la curva plana, desconocemos el valor de una de las componentes (en este caso la z), por lo que sólo es posible obtener uno de los sistemas de ecuaciones de que hablábamos. Esto, sin embargo, no causa ningún problema: ahora cualquier solución real que satisfaga el segundo requisito del apartado anterior es válida.

A partir del valor de (u_i, v_i) obtenido al resolver el sistema de ecuaciones, podemos calcular el punto (x_i, y_i, z_i) de la superficie buscado, sin más que aplicar las ecuaciones de cálculo de superficies paramétricas en forma Bézier. Obsérvese que si este punto (x_i, y_i, z_i) es una proyección del punto (x_i, y_i) de la curva, los valores de x e y deben ser los mismos, habiendo calculado por este método el valor de z adecuado. Nótese, además, que para un valor de (x,y) puede haber hasta tres valores de z , es decir, hasta tres soluciones. Serán soluciones válidas aquellas soluciones reales que se encuentren dentro del intervalo de definición del parámetro $[0,1]$. Pueden existir hasta tres soluciones, ya que las superficies paramétricas bicúbicas pueden ser trivariadas para un único valor de (x,y) (Figura 5.8). De las posibles soluciones del sistema, debemos seleccionar aquella más cercana al punto de la curva que estamos proyectando, en la dirección de proyección.

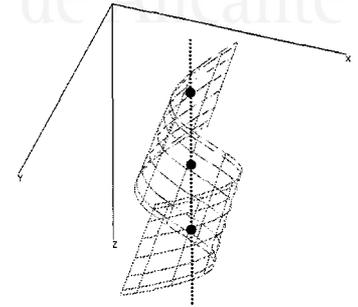
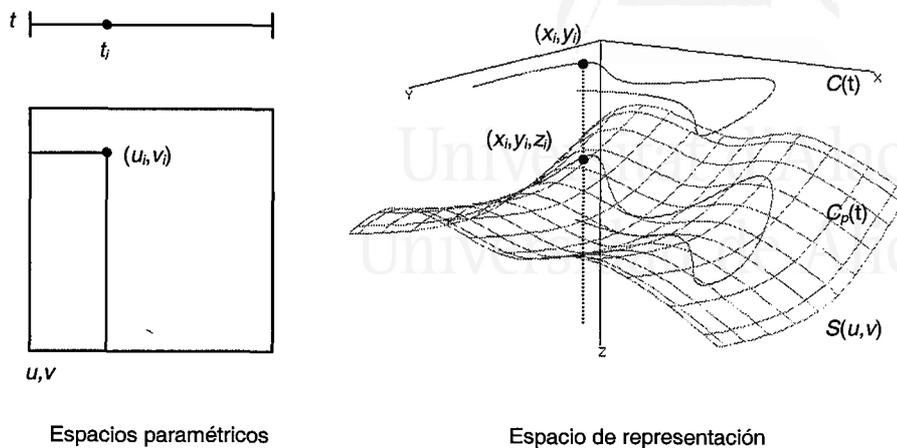


Figura 5.8
Superficie trivariada para un
valor (x,y)

**Figura 5.9**

Proyección de una curva sobre una superficie. Obsérvese las transformaciones llevadas a cabo entre los diferentes espacios paramétricos y de representación

El algoritmo planteado para proyectar una curva plana sobre una superficie tridimensional se puede ver en la tabla 5.2.

ENTRADA
Sup(u,v): Superficie sobre la que queremos proyectar
C(t): Curva definida sobre un plano (XY)
e _{max} : error máximo para discretización
SALIDA
C'(t): Curva resultado de proyectar C(t) sobre Sup(u,v)
VARIABLES
p, p': Puntos
P: Conjunto de puntos
S: Polilínea que discretiza la curva C
ALGORITMO
S=C.Discretizar (e _{max})
Para i=0 hasta S.NumeroPuntos() hacer
p = S(i)
SistemaXY = PlantearSistema (Sup, p)
P = SistemaXY.Resolver ()
Si (P ≠ ∅) entonces
p' = q ∈ P / (q ∈ [0,1] ∧
(¬∃q' ∈ P / distancia (q', p') < distancia (q, p'))
C'.Añadir (p')
FinSi
FinPara

Tabla 5.2

Algoritmo para la proyección de curvas planas sobre superficies 3D

5.4.3 Despliegue de curvas sobre superficies

La otra herramienta desarrollada para situar curvas sobre superficies se ha denominado despliegue de curvas, por colocar las curvas como si se desplegaran sobre la superficie. De modo intuitivo, podemos decir que la curva se diseña sobre un plano que se deja caer sobre la superficie y se adapta a ella como si fuera de papel o de algún tejido. En la figura 5.10 se establece una comparación entre los métodos de proyección y de despliegue.

Se parte de una discretización de la curva como la planteada anteriormente, y se va desplegando cada tramo de forma consecutiva: se hace coincidir el primer punto de la curva con un punto sobre la superficie, para pasar a posicionar el plano que contiene a la curva de forma tangente a la superficie en el punto inicial. El segundo punto del tramo ya está prácticamente sobre la superficie. No obstante, para asegurarnos, se proyecta sobre ella. Este procedimiento se repite para todos los tramos.

Más formalmente, sea $C(t)$ una curva cúbica paramétrica definida sobre un espacio \mathcal{R} para el parámetro t , y con un espacio de representación \mathcal{R}^2 , diseñada sobre un plano. Supongamos que ese plano sea el plano coordenado XY . Sea p_a un punto de aplicación de la curva sobre la superficie, y D un vector de dirección de despliegue, por ejemplo $(1,0,0)$, es decir, la curva se despliega en sentido X .

El primer paso es discretizar la curva con una precisión suficiente. A continuación debemos hacer coincidir el punto de aplicación p_a con el primer punto p_1 de la curva. Alternativamente, podemos prescindir del punto de aplicación y tomar ese punto como la proyección del primer punto de la curva. En cualquier caso, el punto de aplicación p_a será el punto inicial de la curva desplegada. A continuación debemos trasladar la curva al punto de aplicación.

Sean dos vectores perpendiculares definidos sobre el plano que contiene a la curva, uno de ellos coincidente con el vector de dirección de despliegue y el otro perpendicular a éste. Sin pérdida de generalidad, podemos suponer que el plano es el XY y los vectores son los vectores paralelos a los ejes. Llamemos V y H (vertical y horizontal) a estos vectores, definidos como $V=(0,1,0)$ y $H=(1,0,0)$. Sea α el ángulo que forma el vector V con el segmento que une los dos primeros puntos de la curva. Sea N el vector normal a la superficie en el punto de aplicación p_a . Podemos calcular un vector $V'=N \times H$, por lo tanto perpendicular a N y a H , que coincidiría con el vector V si transformáramos el plano de manera que N y la normal al plano coincidieran, manteniendo invariable la dirección de despliegue.

Sea, por otro lado, d la distancia entre los dos primeros puntos de la curva p_1 y p_2 . Podemos crear un nuevo punto q como el punto de aplicación trasladado una distancia d en la dirección marcada por V' . Este punto q debemos rotarlo un ángulo α alrededor de la normal, es decir, sobre el plano tangente a la superficie en el punto de aplicación. Ahora sólo queda proyectar el punto q sobre la superficie, para asegurarnos de que se encuentra sobre ella y repetir el proceso para el siguiente par de puntos, tomando q como punto de aplicación.

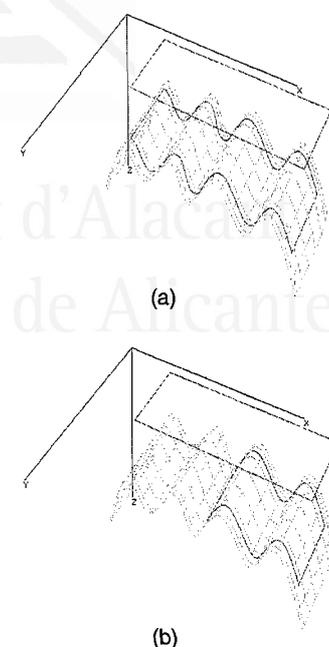


Figura 5.10
Efecto conseguido al proyectar una curva y al desplegarla. En (a) se muestra la proyección y en (b) el despliegue

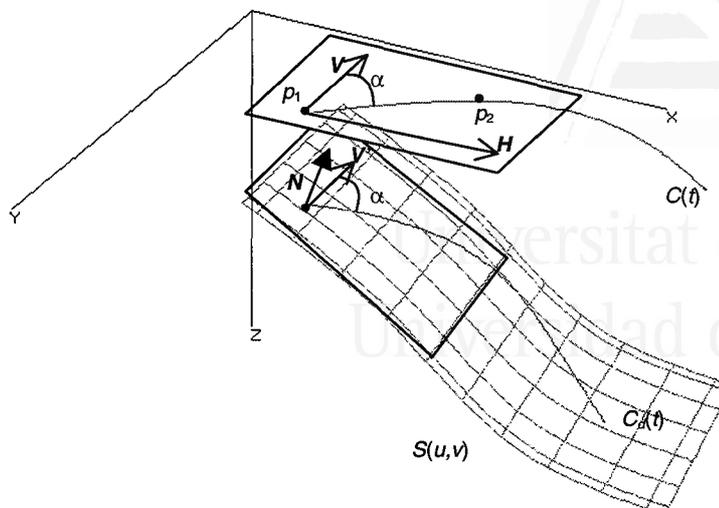


Figura 5.11

Despliegue de una curva sobre una superficie

La figura 5.11 presenta el método de forma gráfica, mientras que en la tabla 5.3 podemos ver el algoritmo de despliegue propuesto.

ENTRADA
Sup(u, v): Superficie sobre la que queremos desplegar
C(t): Curva definida sobre un plano (XY)
p_a : Punto de aplicación sobre la superficie
e_{max} : error máximo para discretización
SALIDA
C'(t): Curva resultado de desplegar C(t) sobre Sup(u, v)
VARIABLES
D, V, H, N, V': Vectores
p, p_0 , p_1 : Puntos
P: Conjunto de puntos
α : Angulo
d: distancia
S: Polilínea que discretiza la curva C
ALGORITMO
D = (1, 0, 0)
V = (0, 1, 0)
H = (1, 0, 0)
C.Trasladar ($p_a - C(0)$)
P.Añadir (p_a)
S=C.Discretizar (e_{max})
Para i=0 hasta S.NumeroPuntos()-1 hacer
$p_0 = S(i)$
$p_1 = S(i+1)$
d = Distancia (p_0, p_1)
α = Angulo ($p_1 - p_0, V$)
p = Sup.Proyectar (p_0)
N = Sup.Normal (p)
V' = N x H
p' = p
p'.Trasladar (V', d)
p'.Rotar (α, p)
P.Añadir (Sup.Proyectar(p'))
Fin para

Tabla 5.3

Algoritmo para el despliegue de curvas planas sobre superficies 3D



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 6

CONSTRUCTORES DE TRAYECTORIAS

6.1 INTRODUCCIÓN

El segundo grupo de constructores geométricos que definen la semántica de nuestro modelo aborda el cálculo de las trayectorias que debe seguir una herramienta para mecanizar una pieza. Debemos tener en cuenta una cuestión fundamental: a diferencia de los demás modelos, las trayectorias en nuestro caso forman parte también del diseño de la pieza, es decir, el objeto no está completamente definido hasta que las trayectorias y las herramientas que pasan por ellas se han descrito por completo. Por el contrario, en los demás modelos, las trayectorias persiguen fabricar adecuadamente el objeto que ya se ha diseñado.

El hecho que evidencia esta relación entre el diseño y las trayectorias en nuestro modelo, es que no es posible ni tan siquiera visualizar un objeto acabado hasta que se han completado todas las trayectorias de mecanizado y se han definido las herramientas.

6.2 CÁLCULO DE TRAYECTORIAS DE MECANIZADO

Como paso previo a la descripción de los métodos de cálculo de trayectorias de mecanizado vamos a acotar el problema planteando la restricción de utilizar solamente máquinas de 3 ejes. La ampliación a 5 ejes no supone a nivel conceptual ninguna complicación bastando con permitir que la dirección de incidencia de la herramienta pueda ser libre. Sin embargo, sí supone un aumento de complejidad al tratar los métodos concretos o constructores que facilitan el diseño con el modelo. Determinados procedimientos (por ejemplo, el compensado o la emulación de las herramientas) tiene una problemática diferente y, a menudo, más compleja, cuando utilizamos 5 ejes. No obstante, son conceptos que tienen que ver más con un interfaz de diseño intuitivo y sencillo que con el modelo en sí.

6.2.1 Parámetros de mecanizado

En nuestra propuesta, hemos contemplado diferentes posibilidades para el cálculo de las trayectorias. En concreto, las opciones referentes al tipo de trayectoria son:

- Trayectoria de perfilado o contorneado
 - Sobre la curva (sin compensado)
 - Interior/Exterior para curvas cerradas
 - Izquierda/Derecha para curvas abiertas
- Trayectoria de desbaste o vaciado
 - Paralelo o en zig-zag
 - Y constante
 - X constante
 - Paralelo a una dirección genérica

- Concéntrico o en espiral
 - Hacia el interior
 - Hacia el exterior

Por otro lado, también podemos distinguir los procesos de mecanizado según cómo se lleve a cabo la profundización en el material:

- Mecanizado a altura constante
- Mecanizado a altura variable

6.2.2 Cálculo de *offsets*

Antes de proceder a describir el procedimiento de mecanizado, debemos explicar cómo se calcula el *offset* o paralela de una curva. Este método se va a utilizar frecuentemente en los algoritmos siguientes de cálculo de trayectorias.

Sea $C(t)=(C_x(t),C_y(t))$ una curva cúbica paramétrica definida sobre un espacio \mathfrak{R} para el parámetro t , y con un espacio de representación \mathfrak{R}^2 , diseñada sobre un plano. Supongamos que ese plano sea el plano coordenado XY .

Definimos su curva *offset* $C_d(t)$ con un radio constante d como [Elber, 1997] [Li, 1998]:

$$C_d(t) = C(t) + d \cdot N(t) \quad \text{Ecuación 6.1}$$

donde $N(t)$ es el vector normal unitario de $C(t)$, que se define según la siguiente expresión:

$$N(t) = \frac{(C_y'(t), -C_x'(t))}{\sqrt{C_x'(t)^2 + C_y'(t)^2}} \quad \text{Ecuación 6.2}$$

La condición de regularidad de $C(t)$, garantiza que la derivada $C'(t)=(C_x'(t),C_y'(t)) \neq (0,0)$, con lo que $N(t)$ está bien definido para cualquier curva $C(t)$ que cumpla esta condición. En concreto, es fácil conseguir que las curvas Bézier y BSpline la cumplan, con lo que podremos asegurar el cálculo de esta curva *offset*. Cualquier curva Bézier o BSpline es regular sin no existe ningún punto en el que su derivada sea 0. Si esto ocurriera, bastaría con dividir la curva en ese punto por cualquiera de los métodos conocidos [Farin, 1993].

Desafortunadamente, la raíz en el denominador de la ecuación 6.2 hace que la curva *offset* de una curva $C(t)$ no sea polinómica o racional, aun cuando $C(t)$ sí lo sea. Esto hace que debamos recurrir a aproximaciones racionales o polinómicas del *offset*, y no a soluciones exactas.

Entre las aproximaciones existentes en la literatura destacamos [Elber, 1997]:

- **Métodos basados en el polígono de control:** Estos métodos se basan en la traslación de los puntos de control de la curva (Cobb) o de las aristas del polígono de control (Tiller y Hanson). Otros autores ha mejorado estos métodos tratando

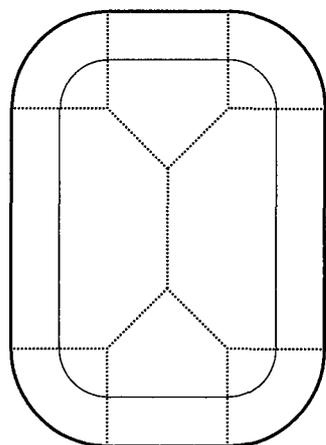


Figura 6.1
Diagrama de Voronoi para un contorno cerrado. En trazo fino se representa el *offset* y en trazo discontinuo el diagrama

los puntos de la curva más cercanos a cada punto de control (Coquillart) o utilizando una estimación del error en un entorno de vecindad para cada punto de control y ajustar así su traslación (Elber y Cohen).

- **Métodos de interpolación:** Algunos autores han utilizado las curvas cúbicas de Hermite para aproximar curvas *offset* (Klass). En otros casos se han preferido soluciones basadas en métodos de mínimos cuadrados (Hoschek). También se ha sugerido la utilización de un método de interpolación simple con BSplines, generando un conjunto finito de puntos pertenecientes al *offset* y haciendo pasar por ellos una curva cúbica de tipo BSpline (Pham).
- **Métodos de aproximación por círculos:** La curva paralela puede computarse de manera exacta utilizando arcos circulares. Estos métodos sugieren aproximar estos arcos mediante curvas racionales. La precisión de la solución dependerá de la aproximación a los círculos que se utilice (Lee *et al.*)
- **Métodos basados en aproximaciones de series:** Li y Hsu proponen un método basado en la aproximación de la curva paralela mediante una serie. En concreto, utilizan la serie de Legendre que ofrece ventajas sobre la de Taylor en problemas como éste [Li, 1998].
- **Métodos basados en el diagrama de Voronoi:** El diagrama de Voronoi de un conjunto de elementos se define como una división del plano en áreas disjuntas denominadas áreas de Voronoi. Cada una de estas áreas va asociada a un elemento del conjunto, y está formada por aquellos puntos del plano que están más cercanos a ese elemento que a ningún otro del conjunto [O'Rourke, 1993]. Si tomamos como elementos los tramos de la curva y calculamos las paralelas a cada tramo, el *offset* estará formado por cada sección de las paralelas que se encuentre en su área de Voronoi (figura 6.1). Lambregts [Lambregts, 1996] propone en su artículo un algoritmo para calcular el diagrama de Voronoi basado en una técnica de división y unión (*split-and-merge*). Por su parte, Held [Held, 1998] presenta un método de propagación. Pueden encontrarse, además, procedimientos basados en la intersección de los planos intermedios, en técnicas de divide y vencerás, etc. [O'Rourke, 1993]. Los métodos basados en el diagrama de Voronoi suelen utilizarse para figuras formadas por tramos rectos y arcos circulares.
- **Métodos basados en una discretización de la curva:** Algunos métodos de obtención de paralelas parten de una discretización de la curva en forma de polilínea. Estos métodos tienen la ventaja de ser muy sencillos pero, a cambio, presentan dos problemas importantes: La discretización genera pérdida de precisión y, además, es necesario resolver el problema de la autointersección.

En el artículo de Elber, Lee y Kim [Elber, 1997] encontramos una interesante comparación entre buena parte de estos métodos.

En el modelo Superficie-Trayectoria, es posible utilizar cualquiera de ellos, siempre que la aproximación sea suficientemente buena. En nuestro caso, hemos optado por un método de aproximación basado en una discretización de las curvas en forma de polilíneas.

6.2.2.1 Cálculo del *offset* por discretización

Hemos elegido un sistema de cálculo del *offset* por discretización de la curva por ser una de las opciones más sencillas. Estos métodos producen una pérdida de precisión por basarse en una aproximación. Sin embargo, si esta discretización es adecuada, las pérdidas son mínimas y obtenemos un método sencillo y rápido.

Nuestro método se realiza en tres pasos: el primero consiste en discretizar la curva con una precisión adecuada. A continuación, se realiza el proceso de cálculo de las paralelas. Por último, es necesario detectar los puntos de autointersección y eliminar los tramos no válidos entre ellos.

Discretización

El problema de la discretización de una curva se trató en el capítulo 5. Utilizaremos este mismo procedimiento para obtener una aproximación adaptativa de una curva paramétrica $C(t)$ mediante segmentos rectos. Llamemos $S=\{s_1, s_2, \dots, s_n\}$ a la polilínea o conjunto de n segmentos rectos ordenados que aproximan la curva $C(t)$.

Cálculo de las paralelas

Una vez reducido el problema del cálculo de la paralela a una curva al del cálculo de la paralela a una polilínea, vamos a proceder a efectuar este proceso.

Sea S el conjunto de tramos resultado de la discretización descrita en el apartado anterior, y sea d la distancia a la que debe ser trazada la paralela. El procedimiento recorre todos los tramos de la polilínea S . Para cada uno de ellos s_i , calcula un segmento paralelo s'_i a una distancia d . Este segmento es finito, siendo su longitud la misma que la del segmento s_i .

Se calcula a continuación el punto de intersección p_i entre el segmento paralelo s'_i y el correspondiente al tramo anterior s'_{i-1} . Si el ángulo que forman los tramos s_i y s_{i-1} (y por lo tanto, el que forman s'_i y s'_{i-1}) es menor de 180° la intersección se encontrará dentro de los segmentos, es decir, si describimos los segmentos s'_i y s'_{i-1} en forma paramétrica, el valor en s'_{i-1} del parámetro en el punto de intersección debe ser menor o igual que 1. En ese caso debemos añadir ese punto de intersección p_i como punto del *offset*.

Cuando los segmentos s_i y s_{i-1} (y s'_i y s'_{i-1}) forman un ángulo mayor de 180° , el punto de intersección p_i se encuentra fuera de los segmentos. Entonces debemos añadir al *offset* un arco de circunferencia entre los extremos de los segmentos s'_i y s'_{i-1} en lugar de un único punto. Esto hace que el *offset* se adapte mejor como

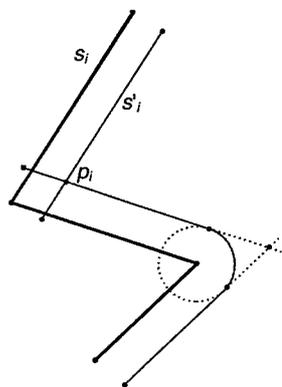


Figura 6.2
Cálculo del *offset* por discretización

trayectoria para el mecanizado. En la figura 6.2 mostramos el comportamiento del algoritmo.

El proceso debe repetirse para todos los tramos de S , suponiendo que este conjunto de segmentos es cíclico si la curva es cerrada. En la tabla 6.1 se presenta el algoritmo de cálculo de la paralela a una polilínea.

<p>ENTRADA S: Polilínea para la que queremos calcular su paralela d: distancia a la que queremos trazar la paralela</p> <p>SALIDA S': Polilínea paralela a S</p> <p>VARIABLES p: Punto $s(t), s_0'(t), s_1'(t)$: Tramos de polilínea en forma paramétrica</p> <p>ALGORITMO</p> <p>$s_0' = \emptyset$ $i = 0$ Repetir</p> <p style="padding-left: 20px;">$s = S.Tramo(i)$ $s_1' = s.CalcularParalela(d)$ $p = s_1'.Intersectar(s_0')$ Si $(\exists p)$ entonces</p> <p style="padding-left: 40px;">Si $(s_0'.Parámetro(p) \leq 1)$ entonces $S'.Añadir(p)$</p> <p style="padding-left: 40px;">Si no $S'.Añadir(s_0'(1))$ $S'.Añadir(Arco(s(0), s_0'(1), s_1'(0)))$ $S'.Añadir(s_1'(0))$</p> <p style="padding-left: 20px;">Fin Si</p> <p style="padding-left: 20px;">Fin Si $s_0' = s_1'$ $i++$</p> <p>Hasta $i=UltimoTramo()$ Si $(EsCerrada(S))$ entonces $S'.Añadir(s_0'(1))$</p> <p>Fin Si</p>

Tabla 6.1

Algoritmo de cálculo de la paralela a una polilínea

Eliminación de tramos no válidos entre autointersecciones

Una vez calculado el *offset* inicial de la polilínea, debemos eliminar los ciclos o tramos no válidos entre autointersecciones. Sea S la polilínea inicial resultado de discretizar la curva $C(t)$ y sea S' el *offset* obtenido tras aplicar el algoritmo del apartado anterior. S' será una lista de segmentos rectos y de arcos de circunferencia ordenados. Si la curva original era cerrada, la lista será cíclica.

El primer paso consiste en calcular todas las autointersecciones de S' , hallando los puntos de corte de cada tramo con los demás. Teniendo en cuenta que todos los tramos son rectas o arcos, será muy sencillo calcular estas intersecciones: recta-recta, recta-arco, arco-arco. Debemos añadir los puntos que indican las autointersecciones en la

lista del *offset* S' , marcados con un *flag* para poderlos distinguir de los puntos que delimitan los segmentos. Puesto que cada punto de intersección pertenece a dos segmentos de la lista, debe añadirse dos veces en las posiciones que le correspondan para mantener la ordenación inicial.

A continuación realizamos un proceso repetitivo basado en una pila. Debemos empezar eligiendo un punto p_0 cualquiera de S' que sea válido, es decir, que no pertenezca a un ciclo no válido. Creamos una lista de puntos nueva y guardamos el punto p_0 . Seguidamente elegimos el punto contiguo a p_0 , que llamaremos p y realizamos la siguiente prueba:

- Si p es un punto marcado con el *flag* de intersección, lo apilamos en la pila y recorremos la lista de puntos S' hasta localizar la otra ocurrencia del punto p , a partir de la cual continuaremos ahora. Guardamos p en la lista nueva.
- Si p no está marcado con el *flag*, lo añadimos a la lista nueva y continuamos.

Debemos continuar este proceso hasta que, recorriendo la lista S' , lleguemos de nuevo al punto p_0 si la lista es cíclica o al final si no lo es. En este punto habremos generado una nueva lista que representa uno de los ciclos del *offset*, de tal manera que:

- Si nos encontramos en una iteración impar el ciclo es válido y forma parte del *offset* final.
- Si la iteración en la que hemos obtenido el ciclo es par, debemos rechazarlo por no ser válido.

En este momento creamos un nuevo ciclo, desapilamos un nuevo punto p_0 de la pila, y procedemos otra vez hasta que la pila se encuentre vacía.

En la figura 6.3 se muestra gráficamente el funcionamiento del algoritmo. Por otro lado, la tabla 6.2 representa el algoritmo detallado del método.

Analizando el método descrito podemos ver que es fácil generalizarlo para definir la paralela a una curva sin pasar necesariamente por una discretización. Basta con realizar el cálculo de la paralela directamente sobre la curva por un método, por ejemplo, basado en el polígono de control y calcular las intersecciones entre tramos sobre esa curva. El proceso de eliminación de autointersecciones no sufrirá modificaciones.

Sobre el método de cálculo de un *offset* se ha diseñado también un algoritmo que permite calcular el *offset* de varias curvas. Este algoritmo es la base para el cálculo adecuado de las trayectorias de mecanizado. Como paso previo se han implementado también algoritmos para realizar operaciones booleanas entre curvas cerradas.

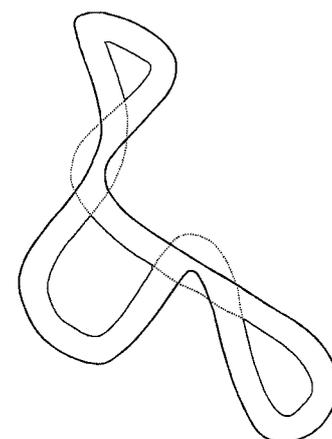


Figura 6.3
 Eliminación de tramos no válidos entre autointersecciones. En líneas discontinuas los tramos eliminados

ENTRADA

S': Polilínea para la que queremos eliminar tramos no válidos

SALIDA

C: Conjunto de polilíneas o ciclos válidos

VARIABLESp₀, p: Puntoss(t), s₀'(t), s₁'(t): Tramos de polilínea en forma paramétrica

c: polilínea o ciclo

ALGORITMO

```

Para i=0 hasta S'.NumeroTramos() hacer
  s0'=S'.Tramo(i)
  Para j=1 hasta S'.NumeroTramos() hacer
    s1'=S'.Tramo(j)
    p = s1'.Intersectar(s0')
    Si (∃ p) entonces
      p.FlagIntersección=Cierto
      S'.Añadir(p, s0')
      S'.Añadir(p, s1')
    Fin Si
  Fin Para
Fin Para

p0=q∈S'/Valido(q)
Pila.Apilar(p0)
iteración = 1

Repetir
  p0=Pila.Desapilar()

  Repetir
    p=S'.PuntoSiguiente(p0)
    Si (p.FlagIntersección=Cierto) entonces
      Pila.Apilar(p)
      p1=S'.PuntoSiguiente(p)
      Mientras p1≠p hacer
        p1=S'.PuntoSiguiente(p1)
      Fin Mientras
      p= p1
    Fin Si
    c.Añadir(p)
  Hasta (p= p0)

  Si (Impar(i)) entonces
    C.Añadir(c)
  Fin Si

  i++
  c=∅

Hasta Pila.Vacia()

```

Tabla 6.2

Algoritmo de eliminación de tramos no válidos entre autointersecciones

6.2.2.2 Operaciones booleanas entre curvas cerradas

Se definen tres operaciones entre curvas cerradas: unión, intersección y diferencia. En la figura 6.4 se muestra gráficamente el significado de estas operaciones para este tipo de figura.

En las tres operaciones se plantea un preproceso común: en primer lugar debemos calcular las intersecciones entre las dos curvas. A continuación, se ordenan los valores paramétricos de esos puntos de intersección en orden creciente para ambas curvas.

Unión

Para cada par de parámetros ordenados, extraer los tramos de la primera curva comprendido entre esos dos parámetros. Tomar cualquier punto del tramo y ver si es interior a la otra curva. Si lo es, desechar el tramo; si no lo es, incorporarlo a una lista de tramos.

A continuación, repetir el proceso para la segunda curva, comprobando qué tramos son interiores o exteriores a la primera, y quedarnos con los exteriores.

Por último, debemos ordenar todos los tramos de la lista, para obtener una única figura.

Intersección

El procedimiento de intersección es muy similar: Se recorren igualmente los tramos de ambas curvas, pero en esta ocasión nos quedamos con los tramos de cada curva que son interiores a la otra.

Diferencia

Tras hacer el mismo preproceso que para la unión y la intersección, debemos quedarnos con los tramos de la primera curva que son exteriores a la segunda, y con los tramos de la segunda que son interiores a la primera.

La tabla 6.3 presenta el preproceso y los tres métodos para realizar las operaciones lógicas.

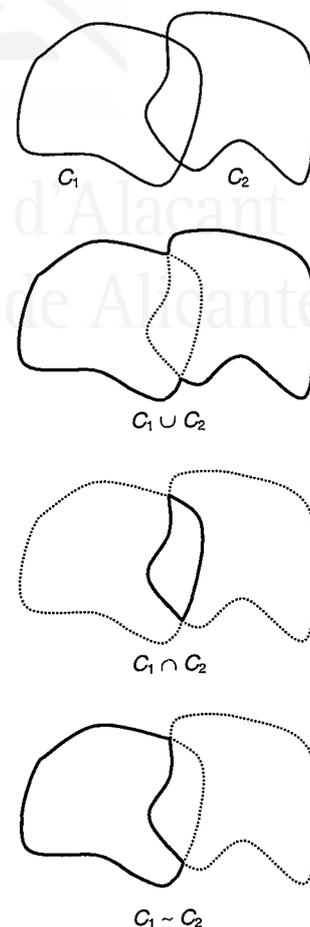


Figura 6.4
Operaciones booleanas entre curvas cerradas

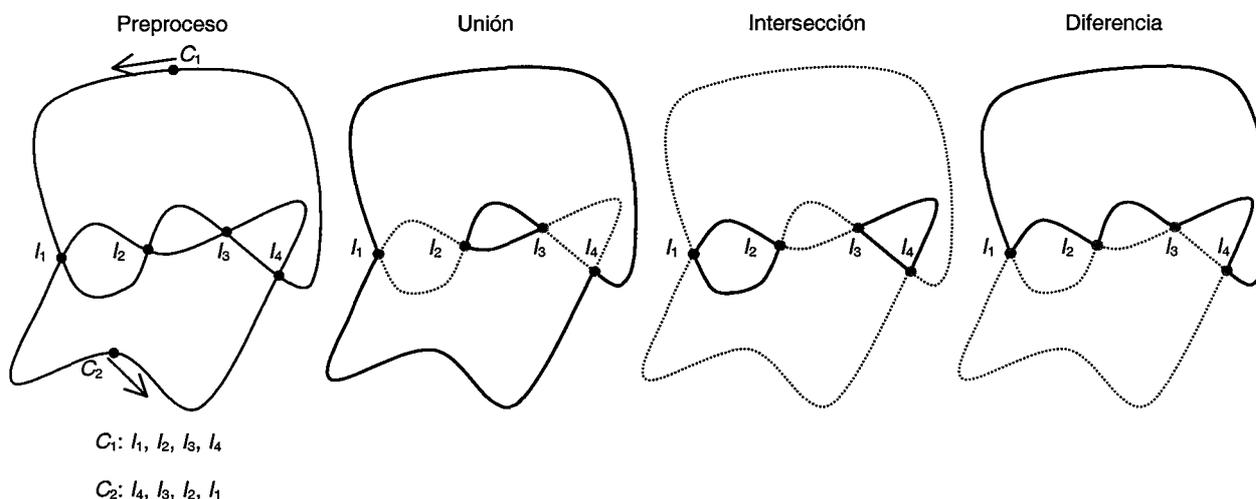


Figura 6.5
Esquema de resolución para el preproceso, la unión, la intersección y la diferencia de curvas

<p>ENTRADA C_1, C_2: Curvas con las que queremos calcular la operación Op: Operación: Unión, Intersección o Diferencia</p> <p>SALIDA C': Conjunto de curvas resultantes de la operación</p> <p>VARIABLES I: Conjunto de puntos de intersección p: Puntos t_1, t_2: Tramos de curva</p> <p>ALGORITMO</p> <pre> //preproceso I=Intersectar(C_1, C_2) Para i=0 hasta I.NumeroPuntos() hacer p=I(i) C_1.InsertarEnOrden(p) C_2.InsertarEnOrden(p) Fin Para Para i=0 hasta I.NumeroPuntos() hacer $t_1=C_1$.ExtraerTramo(i) $t_2=C_2$.ExtraerTramo(i) //unión Si (Op=Unión) entonces Si (C_2.Exterior(t_1)) entonces C.Añadir(t_1) Fin Si Si (C_1.Exterior(t_2)) entonces C.Añadir(t_2) Fin Si Fin Si //intersección Si (Op=Intersección) entonces Si (C_2.Interior(t_1)) entonces C.Añadir(t_1) Fin Si Si (C_1.Interior(t_2)) entonces C.Añadir(t_2) Fin Si FinSi //diferencia Si (Op=Diferencia) entonces Si (C_2.Exterior(t_1)) entonces C.Añadir(t_1) Fin Si Si (C_1.Interior(t_2)) entonces C.Añadir(t_2) Fin Si FinSi </pre>
--

Tabla 6.3

Algoritmo de cálculo de operaciones booleanas entre curvas cerradas

6.2.2.3 Cálculo del *offset* de varias curvas

El problema del cálculo de la paralelas a varias curvas surge al tratar el mecanizado de curvas cerradas que encierran islas, es decir, porciones de material que no deben ser mecanizadas. Estas islas se definen a partir de curvas cerradas incluidas en las curvas exteriores.

Por ello el cálculo del *offset* de varias curvas sólo tiene sentido si son cerradas.

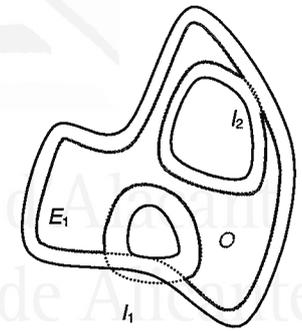
Hemos de distinguir entre *offsets* exteriores (paralelas trazadas por dentro de las curvas exteriores) e interiores (paralelas por fuera de las islas contenidas en las curvas exteriores).

Definimos el *offset* de varias curvas cerradas como la diferencia entre la unión de todas las paralelas exteriores y la unión de todas las paralelas interiores, es decir

$$O = \cup E_i \sim \cup I_j \quad \text{Ecuación 6.3}$$

donde E_i son las paralelas exteriores y I_j las paralelas interiores.

En la figura 6.6 podemos observar el resultado de realizar el *offset* de varias curvas.



$$O = E_1 \sim (I_1 \cup I_2)$$

Figura 6.6
Offset de una curva exterior con dos islas

6.2.3 Perfilado

La operación de perfilado o contorneado se define como el seguimiento de una curva mediante una herramienta de mecanizado. El perfilado puede realizarse con o sin compensado.

En el perfilado sin compensado la herramienta se desplaza directamente sobre la curva, es decir, el centro de la herramienta se sitúa sobre la curva. Una operación de este tipo dejará un surco sobre el material cuya anchura coincidirá con el diámetro de la herramienta, dejando a cada lado de la curva una distancia equivalente al radio.

Otra posibilidad es perfilar la curva realizando un compensado, es decir, calculando la posición del centro de la herramienta para que el borde del surco siga exactamente la curva. La curva que describe el centro de la herramienta coincide con un *offset* calculado a una distancia igual al radio de la herramienta.

6.2.4 Desbaste

Conocemos por desbaste al vaciado de material dentro de un área determinada por una curva o contorno. Existen, básicamente, dos tipos principales de trayectorias para realizar un desbaste [Arkin, 2000]:

- Desbaste en zig-zag o paralelo a una dirección: la herramienta sigue una trayectoria paralela a una dirección inicial. Para minimizar el número de retracciones de la herramienta, las trayectorias se recorren alternativamente de izquierda a derecha y viceversa, resultando un recorrido de tipo zig-zag.
- Desbaste en espiral, concéntrico o paralelo al contorno: las trayectorias son paralelas (*offsets*) al contorno del área a desbastar. Resulta, de esta manera, una trayectoria de tipo espiral, donde cada pasada de la herramienta se realiza a una distancia del contorno constante.

Evidentemente las curvas a desbastar deben ser cerradas. No obstante, nosotros vamos a extender el concepto para definir también la operación de desbaste para superficies: en este caso consistirá en el desbaste del contorno de la superficie.

En el interior de los contornos pueden existir otras curvas cerradas que no queramos desbastar. Esas curvas se conocen como islas y los algoritmos de desbaste propuestos deben contemplar la posibilidad de que existan una o varias dentro de la curva cerrada.

En las secciones siguientes presentamos dos algoritmos para el desbaste de curvas cerradas en zig-zag y en espiral. Previamente calculamos el offset del contorno y de las islas si las hubiere. Esto evita que la herramienta invada el área que no debe mecanizarse.

6.2.4.1 Desbaste en zig-zag

Para el desbaste de un área en zig-zag se ha diseñado un algoritmo basado en el método de relleno de polígonos por línea de rastreo o *scanline* [Hearn, 1995] [Foley, 1990]. En estos métodos de relleno, se traza un conjunto de rectas paralelas al eje X , denominadas líneas de rastreo o *scanlines*, que se hacen intersectar con las aristas del polígono a rellenar. Cuando una línea de rastreo pasa por un vértice, se produce un caso singular que habrá que tratar adecuadamente. Después se ordenan los puntos de intersección de izquierda a derecha y cada par de puntos consecutivos determinan, alternativamente, una porción de recta que debe rellenarse, y otra porción que debe quedar con el color del fondo. Estos algoritmos están descritos para polígonos con aristas rectas, por lo que es muy sencillo aprovechar la coherencia espacial y desarrollar algoritmos incrementales de alta eficiencia.

En nuestro caso, se propone una serie de generalizaciones que permiten desarrollar un algoritmo para aplicarlo al problema de la generación de trayectorias de desbaste en zig-zag. Estas generalizaciones son:

- Líneas de rastreo no necesariamente paralelas a un eje coordinado. Su dirección vendrá dada por un vector director.
- Los objetos a desbastar no son, necesariamente, polígonos de aristas rectas. El método se generaliza para curvas paramétricas cerradas.
- Debemos seguir tratando los casos singulares, representados por aquellas intersecciones en que la dirección de la tangente a la curva y el vector director de las líneas de rastreo coinciden.
- La ordenación de los puntos variará, alternativamente, de derecha a izquierda y de izquierda a derecha para cada línea de rastreo. De esta manera se produce un recorrido en zig-zag, evitando un exceso de recorridos a altura de seguridad.

Veamos detenidamente el algoritmo y cada uno de estos aspectos. Sea D el vector director de las trayectorias en zig-zag, que puede ser cualquier vector definido sobre el plano de diseño de la curva a desbastar. Sin pérdida de generalidad, podemos suponer que ese vector es paralelo al eje de las X , ya que si no lo fuera podemos

aplicar la transformación adecuada a toda la figura para que el vector coincida con esa dirección. D define un conjunto de líneas de rastreo paralelas $L=\{l_i\}$, separadas una distancia d , definida por el usuario.

Tras calcular el *offset* de los contornos e islas que queremos desbastar, el algoritmo recorre cada línea de rastreo l_i realizando este proceso: Intersectamos l_i con todas las curvas de la figura, incluyendo las islas si las hubiera. Todos los puntos de intersección p_j se añaden a una lista, en orden creciente de X si el índice i es par, y decreciente si es impar (o a la inversa, si el usuario lo requiere). En este punto es necesario distinguir el caso de que la intersección se produzca en un punto de pendiente 0, es decir, en un punto de tangente paralela al vector director D por si se da una singularidad (figura 6.7):

- El punto es un máximo o un mínimo y, por lo tanto, los tramos de curva contiguos a la intersección se encuentran ambos al mismo lado de la línea de rastreo. En estos casos existe singularidad y debemos eliminar el punto de la lista de intersecciones. Obsérvese que se produce aquí una diferencia con respecto al algoritmo de relleno de polígonos: en este algoritmo los puntos singulares se añaden dos veces a la lista. En nuestro caso también podría plantearse esta solución. Sin embargo se producirían retracciones de la herramienta innecesarias en algunos casos (como en el punto b de la figura 6.7) o incidencias de la herramienta en un solo punto (como en el punto a de esta misma figura), situación poco recomendable porque puede producir marcas en el material. Se ha optado por eliminar el punto singular con el objetivo de obtener trayectorias más eficientes y eliminar posibles errores.
- La intersección es un punto de inflexión, es decir, los tramos contiguos a la intersección se encuentran en lados diferentes de la línea de rastreo. Debemos tratar la intersección de forma normal y hemos de añadir el punto a la lista de intersecciones.

Una forma sencilla de determinar si se produce el caso singular es comprobar para cada intersección si a ambos lados los tramos consecutivos hasta la siguiente intersección y hasta la anterior, se encuentran en el mismo semiplano marcado por la línea de rastreo. Para ello basta con elegir un punto cualquiera de cada tramo y comprobar si el valor de la coordenada Y en esos puntos está por encima o por debajo de la del punto de intersección (puesto que el vector D es paralelo al eje X). Si ambos están por encima o por debajo, el punto no se añade a la lista, si no, debemos añadirlo (figura 6.8).

Después de este procedimiento tenemos almacenados en la lista todos los puntos de intersección I_j convenientemente ordenados empezando en 0. Teniendo en cuenta las singularidades, el número de puntos de la lista será par. Debemos entonces tomarlos dos a dos, determinando en cada caso cómo recorrer la trayectoria. Sea (I_j, I_{j+1}) un par de puntos de la lista. Pueden darse cuatro posibilidades.

- **Caso a:** El índice j es par. Añadimos I_j e I_{j+1} a la trayectoria y recorreremos la distancia entre ambos a la altura de desbaste.
- **Caso b:** El índice j es impar y ambos puntos están en la misma línea de rastreo (tienen el mismo valor de coordenada Y).

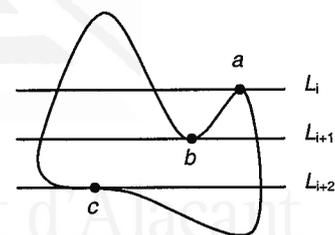


Figura 6.7
Puntos singulares. En a y b se produce una singularidad (por ser máximo y mínimo respectivamente) mientras que en c no se produce (es un punto de inflexión)

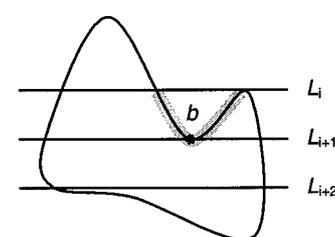


Figura 6.8
Detección de singularidades. Para determinar si en b existe una singularidad basta con comprobar si dos puntos cualesquiera a la derecha y a la izquierda de b (de la zona en gris) se encuentran por encima o por debajo de L_{i+1}

Hemos de añadir I_j e I_{j+1} a la trayectoria y recorrer la distancia entre ambos a la altura de seguridad.

- **Caso c:** El índice j es impar, los puntos están en líneas de rastreo distintas y existe algún punto de intersección sobre la figura entre los puntos I_j e I_{j+1} . Añadimos los puntos I_j e I_{j+1} a la trayectoria y recorreremos su distancia a altura de seguridad.
- **Caso d:** El índice j es impar, los puntos están en líneas de rastreo distintas y no existe ningún punto de intersección sobre la figura entre los puntos I_j e I_{j+1} . En este caso debemos añadir el tramo de la figura entre I_j e I_{j+1} a la trayectoria y recorrer todo el tramo a la altura de desbaste. De esta manera evitamos una salida de herramienta innecesaria.

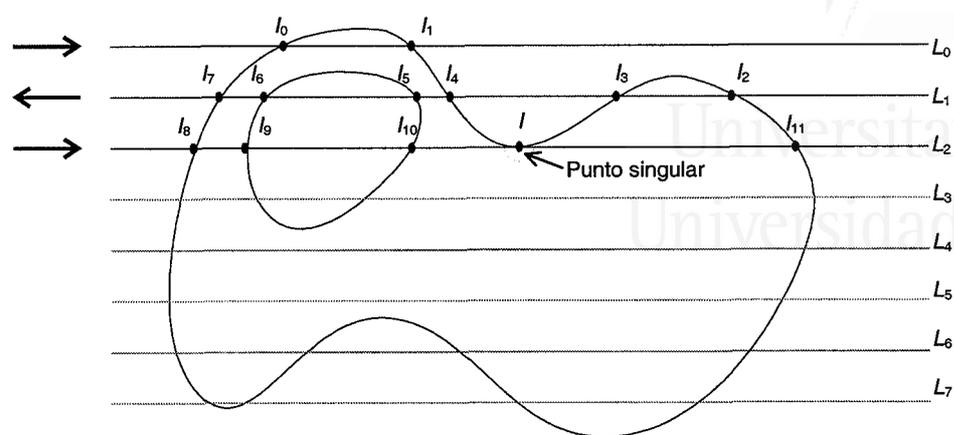
A continuación, en la tabla 6.4 se ofrece el algoritmo.

<p>ENTRADA F: Figura. Cjto de curvas a desbastar. Las interiores son islas</p> <p>SALIDA T: Trayectoria de desbaste en zig-zag para F.</p> <p>VARIABLES l: Línea de rastreo de coordenada y constante p: Punto de intersección I: Conjunto de puntos de intersección L: Lista ordenada de puntos de intersección d: Distancia entre pasadas (p,q): Segmento formado por un par de puntos</p> <p>ALGORITMO i=0; l.y = Ymax Mientras (l.y > Y min) hacer I = F.Intersectar (l) Si ($\exists q \in I / q$ es singular) entonces I.Eliminar (q) Fin si Si (par(i)) entonces L.AñadirEnOrden (I, XCrecente) Si no L.AñadirEnOrden (I, XDecreciente) Fin si l.y = l.y+d i = i+1 Fin Mientras Para j=0 hasta L.NumeroPuntos() hacer (p,q) = (L[j], L[j+1]) Si (par(j)) entonces //caso a T.Añadir (p,q) Si no si (p.y = q.y) entonces //caso b T.AñadirAlturaSeguridad (p,q) Si no Tramo = F.TramoEntre(p,q) Si ($L \cap \text{Tramo} \neq \emptyset$) entonces //caso c T.AñadirAlturaSeguridad(p,q) Si no //caso d T.Añadir(Tramo) Fin si Fin si Fin Para</p>

Tabla 6.4

Algoritmo de desbaste en zig-zag

En la figura 6.9 se representa el funcionamiento del algoritmo de manera gráfica.



Lista de intersecciones

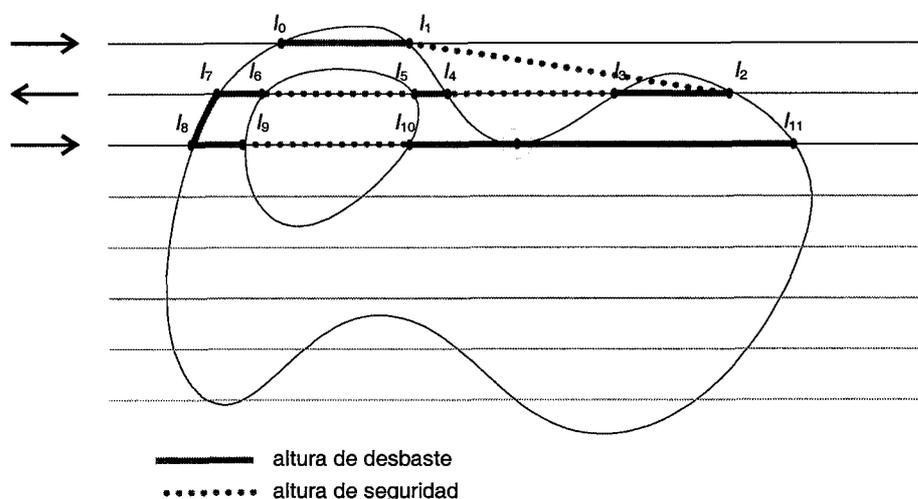
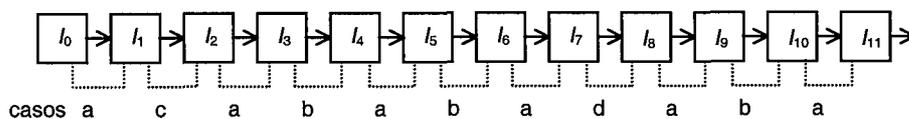


Figura 6.9
Funcionamiento del algoritmo de desbaste en zig-zag

6.2.4.2 Desbaste en espiral

El otro método de desbaste incorporado al modelo es el desbaste en espiral. La mayoría de estos métodos se basa en el cálculo de los diagramas de Voronoi [O'Rourke, 1993], [Lambregts, 1996], [Held, 1998]. En nuestro caso, hemos optado por un algoritmo basado en el cálculo recursivo de paralelas u *offsets* a la curva inicial, aprovechando los métodos de cálculo de paralelas para una y varias curvas descritos en el apartado 6.2.2 de este capítulo.

Conceptualmente, el algoritmo puede plantearse como un algoritmo recursivo en el que se van calculando paralelas interiores sucesivas hasta que la paralela a la curva actual no existe por ser el paso de la espiral demasiado grande. El algoritmo empieza calculando la paralela al contorno, que puede estar formada por una, varias o ninguna curva. El caso base se produce cuando la paralela a la curva actual no existe. En el caso genérico, se llama recursivamente al algoritmo para cada una de las curvas que forman la paralela y se añaden a la trayectoria final. El algoritmo puede verse en la tabla 6.5.

<p>FUNCION Espiral (C, I, T)</p> <p>ENTRADA</p> <p>C: Contorno (curva exterior) I: Conjunto de islas contenidas en el contorno. T: trayectoria calculada hasta el momento.</p> <p>SALIDA</p> <p>T: Trayectoria tras añadirle la última paralela.</p> <p>FUNCIONES AXILIARES</p> <p>Paralela (C,I): Función que calcula la paralela a una curva C exterior que puede contener islas I</p> <p>VARIABLES</p> <p>Off: Conjunto de curvas offset off: offset</p> <p>ALGORITMO</p> <p>Off = Paralela (C, I) Si (Off=∅) entonces //caso base Terminar Si no //caso general Para i=0 hasta Off.NumeroOffsets() hacer off = Off(i) Espiral (off, I, T) T.Añadir (off) Fin para Fin si</p>
--

Tabla 6.5
Algoritmo recursivo de desbaste en espiral

En la figura 6.10 se presenta el funcionamiento del algoritmo de forma gráfica, así como el árbol de recursiones.

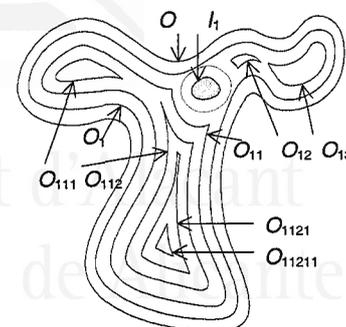
Un aspecto importante en el cálculo de las trayectorias es el control de los puntos que deben recorrerse a altura de seguridad. En este algoritmo este control es complejo. Por esa razón y por cuestiones de eficiencia, se presenta una implementación iterativa del algoritmo basada en un pila.

Se utiliza una pila en la que se van almacenando las diferentes paralelas calculadas, desapilando y agregando la paralela a la lista de trayectorias cuando la curva ya no tiene paralela (es demasiado pequeña para el paso de la espiral). Obsérvese que este caso coincide con el caso base del algoritmo recursivo de manera que la acción de

desapilar simula el procedimiento de vuelta hacia atrás de la recursión.

Además se utiliza un *flag* o marcador booleano en las curvas. Este marcador vale verdadero cuando se ha de continuar calculando paralelas a esa curva, y falso cuando sus paralelas ya se han desarrollado. De esta manera se simula el árbol de recursiones y se posibilita el recorrido a altura de seguridad de aquellos puntos que lo requieran.

El algoritmo se presenta en la tabla 6.6.



ENTRADA

C: Contorno (curva exterior)
I: Conjunto de islas contenidas en el contorno.

SALIDA

T: Trayectoria final en espiral.

FUNCIONES AXILIARES

Paralela (C,I): Función que calcula la paralela a una curva C exterior que puede contener islas I

VARIABLES

Off: Conjunto de curvas offset
off: offset
P: pila
C': curva

ALGORITMO

C' = C

Repetir

Off = Paralela (C', I)

Si (Off ≠ ∅) entonces

Para i=0 hasta Off.NumeroOffsets() hacer

off=Off(i)

off.FlagContinuar = Cierto

P.Apilar (off)

Fin para

Si no

C' = P.Desapilar()

Si (C'.FlagContinuar = Falso) entonces

T.AñadirAlturaSeguridad (C'(0))

Fin Si

Mientras (C'.FlagContinuar = Falso) hacer

T.Añadir (C')

C' = P.Cima()

Si (C'.FlagContinuar = Falso) entonces

C' = P.Desapilar()

Fin si

Fin mientras

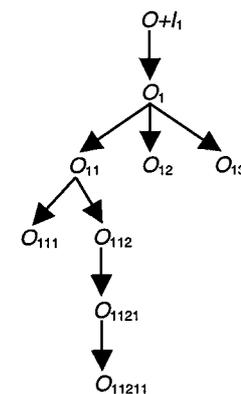
T.AñadirAlturaSeguridad (T.UltimoPunto())

Fin si

C' = P.Cima()

C'.FlagContinuar = Falso

Hasta P.Vacia()



Trayectoria final para $O+h_1$:
 $O_{111} O_{11211} O_{1121} O_{112} O_{11} O_{12} O_{13} O_1$

Figura 6.10

Funcionamiento del algoritmo recursivo de desbaste en espiral.

Obsérvese que la trayectoria final coincide con el recorrido en preorden del árbol de recursiones

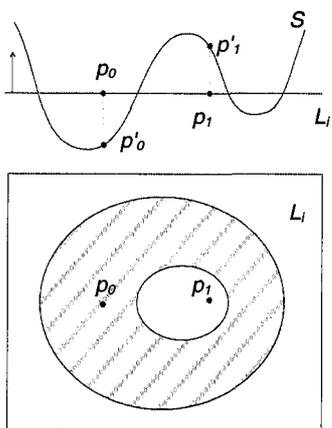
Tabla 6.6

Algoritmo iterativo de desbaste en espiral

6.2.5 Mecanizado a altura constante

El mecanizado a altura constante con una herramienta cuya dirección de incidencia no varía se denomina mecanizado 2D y $\frac{1}{2}$ o *pocket machining*. La forma más directa de abordarlo es trazar una serie de capas y mecanizar una de ellas en cada paso [Mahadevan, 1997]. El método propuesto se basa en tres pasos:

- Obtención de las capas y de los contornos de mecanizado en cada una de ellas, como intersección con la superficie de un plano normal a la dirección de incidencia de la herramienta.
- Determinación del espaciado más adecuado entre capas sucesivas para optimizar el mecanizado.
- Recortado del contorno de la capa con la proyección del contorno de la capa anterior, para evitar el mecanizado de zonas no accesibles.
- Cálculo de la trayectoria para cada capa.



6.2.5.1 Obtención de capas

El primer paso para un mecanizado a altura constante es la obtención de las capas 2D y $\frac{1}{2}$. Para ello, sea S la superficie que queremos mecanizar y sea $L=\{L_i\}$ un conjunto de planos perpendicular al vector de incidencia de la herramienta a los que llamaremos capas. Supondremos, a efectos de calcular las caras interiores/exteriores, que la normal a cada una de las capas tiene sentido contrario al vector de incidencia.

Calcularemos la intersección de S y cada capa L_i , que será un conjunto de contornos cerrados $C_i=\{C_{ij}\}$, cada uno de los cuales deberá ser mecanizado o no, dependiendo de si la superficie que encierra se encuentra por debajo o por encima del plano de la capa, respectivamente. Determinar aquellos contornos que deben ser mecanizados se resuelve mediante el siguiente procedimiento: dado el contorno C_{ij} , sea p un punto interior a ese contorno pero exterior a todos los contornos que se encuentran dentro de C_{ij} . Sea p' la proyección del punto p sobre la superficie S en la dirección marcada por el vector de incidencia de la herramienta. Si p' es un punto interior a la capa L_i , el contorno C_{ij} debe mecanizarse; si p' es exterior a esa capa, el contorno no debe mecanizarse. La figura 6.11 ilustra este procedimiento.

Para el cálculo de la intersección, disponemos de dos posibilidades:

- Intersectar directamente la superficie con la capa [Farin, 1992], utilizando algún algoritmo exacto (si la definición de la superficie lo permite) o aproximado (para superficies Bézier o BSpline, basados en divisiones sucesivas).
- Realizar un mallado de la superficie, generalmente en triángulos, para simplificar el cálculo mediante intersecciones entre planos [Mahadevan, 1997].

Figura 6.11
Vista superior y lateral de la intersección de una superficie S con una capa L_i . El área rayada es la región que debe mecanizarse

6.2.5.2 Espaciado entre capas

La forma más directa de espaciar las capas es seguir un espaciado constante. Sin embargo, esta opción puede resultar ineficiente cuando nos encontramos con superficies horizontales y verticales, tal y como se ilustra en la figura 6.12.

Para determinar un espaciado más adecuado debemos determinar qué tramos de la superficie a mecanizar (o qué triángulos, si la superficie está mallada) son horizontales, verticales o inclinados con respecto a la dirección de incidencia de la herramienta. A continuación calcularemos el espaciado de manera que:

- Los tramos verticales se pueden mecanizar en un único paso, aunque en ocasiones, si la profundidad de estos tramos supera a la capacidad de desbaste de la herramienta, debemos realizar este mecanizado en varios pasos.
- Debemos incluir una capa coincidiendo con cada tramo horizontal.
- En los tramos inclinados las capas se deben trazar con un espaciado adecuado a la precisión que queramos obtener. El mecanizado de estos tramos proporcionará un efecto escalera.

En la figura 6.13 se muestra el resultado de mecanizar el objeto de la figura 6.12 mediante un espaciado variable y ajustado a la inclinación de cada tramo de la superficie.

6.2.5.3 Recortado del contorno

El área a mecanizar en una capa no depende sólo de los contornos C_i obtenidos para esa capa sino también de los de la capa anterior C_{i-1} . Esto ocurre porque en un mecanizado con una máquina de 2 y $\frac{1}{2}$ o 3 ejes, la dirección de incidencia de la herramienta es constante y, por lo tanto, sólo se puede acceder al área común a la capa actual y a la anterior, que denominaremos área accesible. De no hacerlo así, podríamos eliminar material de una capa al mecanizar la siguiente cuando existe alguna concavidad (figura 6.14). Por lo tanto, en un mecanizado con una máquina con dirección de incidencia única el área a mecanizar sólo puede decrecer conforme vamos avanzando a capas más profundas.

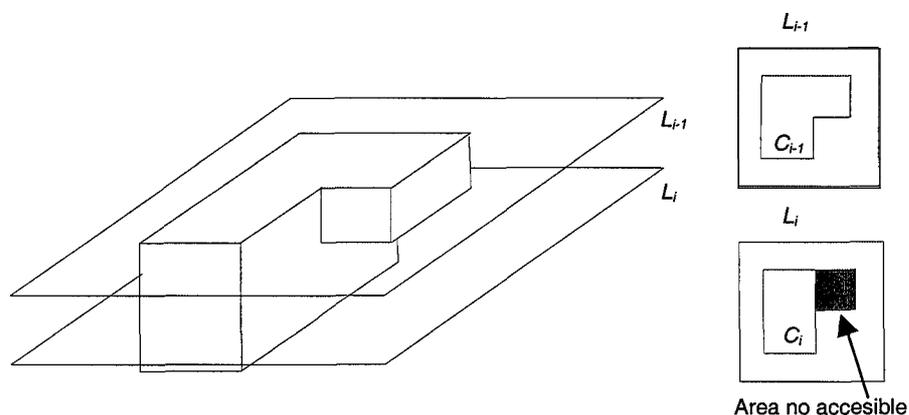


Figura 6.14
Cálculo de las áreas accesibles al ir profundizando en las sucesivas capas

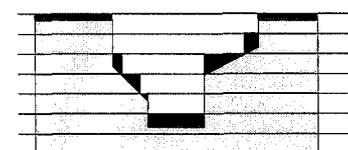


Figura 6.12
Espaciado constante entre capas. En oscuro aparecen las zonas sin mecanizar

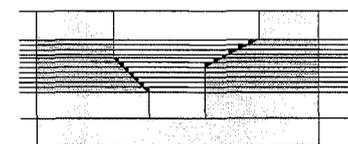


Figura 6.13
Espaciado variable entre capas. Produce resultados mucho mejores

La determinación de la accesibilidad de un área puede verse como la proyección de la sombra de la capa anterior en la dirección de incidencia de la herramienta, es decir, perpendicularmente a las capas. Esta área accesible la calculamos como la intersección de los contornos C_i de una capa con los de la capa anterior C_{i-1} , siguiendo el algoritmo descrito en la sección 6.2.2.2.

6.2.5.4 Cálculo de la trayectoria

Una vez calculadas las áreas accesibles en cada capa, debemos obtener la trayectoria para cada una de ellas. En nuestro modelo hemos contemplado dos tipos de trayectorias: en zig-zag y en espiral. En la sección 6.2.4 hemos tratado ampliamente el problema del cálculo de trayectorias para desbaste.

6.2.6 Mecanizado a altura variable

Llamaremos mecanizado a altura variable o mecanizado 3D a aquél que se realiza con una herramienta cuya dirección de incidencia no varía pero cuya posición puede variarse simultáneamente en sus tres componentes (X , Y , Z). Estas máquinas se conocen como máquinas de 3 ejes y permiten mecanizar superficies no planas siguiendo su curvatura, pero con una única dirección de incidencia. Las máquinas que permiten cambiar también la dirección de incidencia se denominan máquinas de 5 ejes y su problemática es distinta.

En el mecanizado a altura variable el cálculo de las trayectorias es dependiente de la geometría de la herramienta. Habitualmente, sin embargo, se suelen utilizar herramientas esféricas ya que la distancia entre el centro de la herramienta (punto de aplicación) y el punto de contacto con la superficie (punto donde se realiza el corte) es constante.

En este tipo de mecanizados se pueden plantear dos aproximaciones:

- Cálculo de la trayectoria de los puntos de contacto sobre la superficie a mecanizar y posterior obtención de la trayectoria del centro de la herramienta.
- Cálculo del *offset* de la superficie a mecanizar y posterior obtención de la trayectoria del centro de la herramienta sobre la superficie *offset* calculada.

Nuestra propuesta opta por una estrategia del primer tipo, que nos permite adaptar fácilmente algunos algoritmos ya descritos. La segunda opción implica complejos cálculos de superficies paralelas, que pueden consultarse en la literatura [Jerard, 1989], [Li, 1993], [Li, 1994], [Jimeno, 2001].

Debemos hacer notar que el método propuesto es válido siempre que todas las zonas de la superficie sean accesibles, es decir, cuando se cumplan estas dos condiciones (figura 6.15):

- El ángulo entre el vector de incidencia de la herramienta y la normal a la superficie en cualquier punto, debe ser mayor de 90° . Este es un requerimiento para cualquier sistema de mecanizado con 3 ejes, ya que no es posible acceder con una herramienta cuya dirección de incidencia es constante a regiones que presentan alguna concavidad con respecto a esa dirección.
- Todas las concavidades son de tamaño mayor que el diámetro de la herramienta.

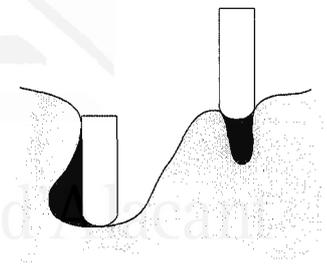


Figura 6.15
Superficie con zonas no accesibles (en color oscuro)

6.2.6.1 Trayectoria de los puntos de contacto

El primer paso en nuestra propuesta es calcular la trayectoria que deben seguir los puntos de contacto de la herramienta con la superficie real que queremos mecanizar. Sea S la superficie a mecanizar y sea C una curva cerrada definida sobre la superficie, obtenida, por ejemplo, por proyección o por despliegue.

En primer lugar aplicaremos una de las estrategias conocidas (zig-zag o concéntrica) para calcular una trayectoria plana de desbaste del contorno C . A continuación debemos situar la trayectoria sobre la superficie de una de estas dos formas:

- Si el contorno C se ha obtenido por proyección de una curva plana, se recomienda calcular el desbaste del contorno plano antes de la proyección y, a continuación, proyectar la trayectoria de desbaste junto con C .
- Si por el contrario C proviene de un despliegue, la forma más adecuada de proceder es obtener la trayectoria de desbaste de C antes del despliegue para proceder a continuación a desplegar las dos curvas, C y la trayectoria.

Como consecuencia de este proceso, se obtiene una trayectoria situada sobre la superficie, que indica los puntos en los que la herramienta debe hacer contacto.

6.2.6.2 Trayectoria de los centros de herramienta

Una vez obtenida la trayectoria de los puntos de contacto, debemos calcular en qué posición ha de estar el centro de la herramienta en cada caso para que se produzca el punto de contacto donde deseamos.

Puesto que la trayectoria de partida es, en realidad, una polilínea, pues proviene de una discretización de la curva original, podemos proceder vértice a vértice. Sea T la trayectoria de los puntos de contacto, $p_i \in T$ cada uno de sus puntos, N_i la normal de S en cada p_i y V el vector que indica la dirección de incidencia de la herramienta. La forma de calcular el centro de la herramienta para cada p_i dependerá de su geometría:

- Si la herramienta es esférica de radio r , el centro de la herramienta c_i para el punto p_i se calcula como un punto a una distancia r de p_i , situado sobre una recta que pase por p_i y cuyo vector director sea N_i (figura 6.16).

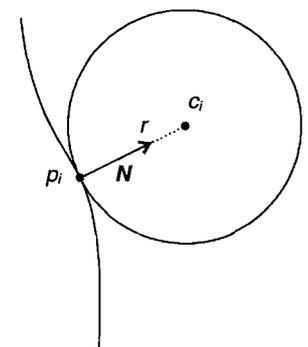
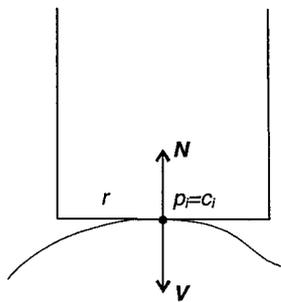
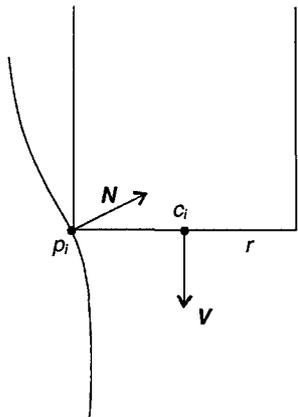


Figura 6.16
Cálculo de la trayectoria de los centros de una herramienta esférica



(a)

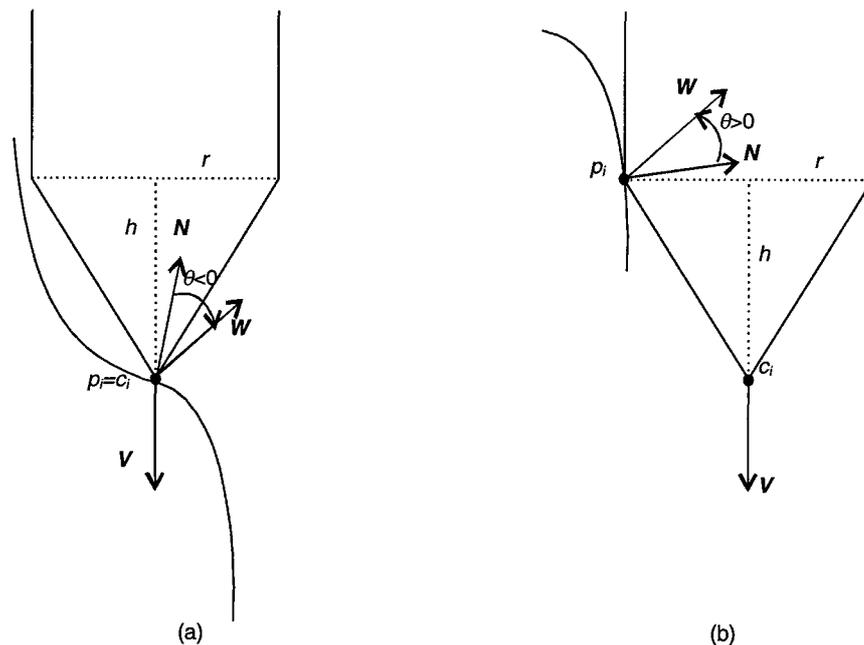


(b)

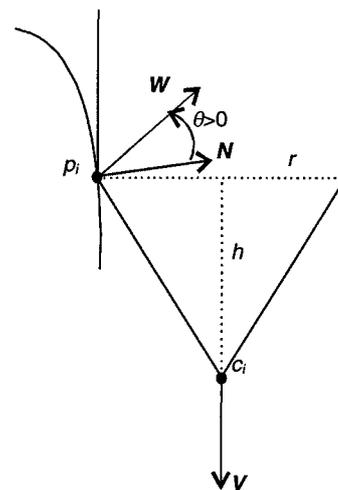
Figura 6.17
Cálculo de la trayectoria de los centros de una herramienta cilíndrica. En (a) las direcciones de N y V coinciden, en (b) no

- Si la herramienta es cilíndrica de radio r , el centro de la herramienta c_i para el punto p_i se calculará de la siguiente manera (figura 6.17):
 - Si N_i tiene la misma dirección que el vector de incidencia V pero sentido inverso, el punto c_i coincide con el punto p_i .
 - En otro caso, c_i es un punto a una distancia r de p_i situado sobre una recta que pase por p_i y cuyo vector director sea perpendicular a V .
- Si la herramienta es cónica de radio r y altura h y W es el vector normal al lateral del cono por su parte interior, el centro de la herramienta c_i para el punto p_i se calculará de la siguiente manera (figura 6.18):
 - Si el ángulo θ que forma N_i con W es ≤ 0 , el punto c_i coincide con el punto p_i .
 - Si el ángulo θ que forma N_i con W es > 0 , c_i es un punto a una distancia $\sqrt{(r^2+h^2)}$ de p_i situado sobre una recta que pase por p_i y cuyo vector director sea perpendicular a W .
- Si la herramienta es tórica de radio principal r y radio del tubo r' , procedemos de forma similar a como hacíamos con la herramienta esférica: Para cada p_i se calcula un punto p'_i a una distancia r' de p_i situado sobre una recta que pase por p_i y cuyo vector director sea N_i . El centro de la herramienta c_i para el punto p_i se sitúa sobre el plano que contiene a V y a N_i a una distancia r de p'_i en dirección perpendicular a V (Figura 6.19).

Los puntos c_i obtenidos en cada caso deben almacenarse ordenados formando una polilínea T' , que será la trayectoria de los centros de herramienta.



(a)



(b)

Figura 6.18
Cálculo de la trayectoria de los centros de una herramienta cónica. En (a) $\theta < 0$ y en (b) $\theta > 0$

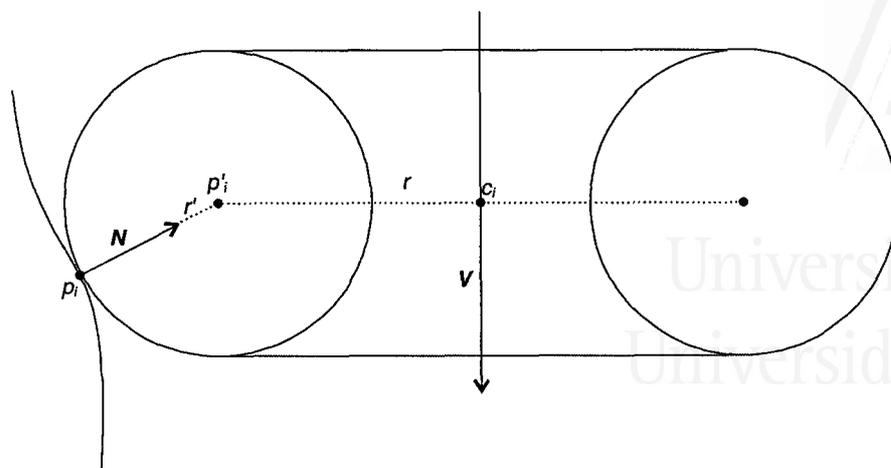


Figura 6.19
Cálculo de la trayectoria de los centros de una herramienta tórica

Puesto que la mayoría de máquinas y de herramientas limitan la profundización que es posible realizar en cada pasada para evitar roturas debido a un exceso de fuerzas sobre la herramienta, estos mecanizados suelen realizarse en varias etapas:

- En primer lugar se realiza una serie de pasadas a baja precisión (puede ser mediante un mecanizado a altura constante o variable). Para calcular pasadas sucesivas de un mecanizado a altura variable basta con sumarle al radio de herramienta r un valor constante en cada pasada y llevar a cabo los cálculos como hemos descrito.
- A continuación se realiza el acabado de la pieza con un mecanizado a altura variable.



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 7

CONSTRUCTORES DE HERRAMIENTAS

7.1 INTRODUCCIÓN

Una de las aportaciones más interesantes del modelo planteado es la utilización de herramientas virtuales como forma de hacer mucho más versátil el uso de herramientas para el diseño de objetos. Los métodos geométricos que permiten definir estas herramientas y que facilitan su emulación mediante herramientas reales constituyen los constructores semánticos asociados a las reglas de producción de la gramática que se refieren a herramientas.

En los capítulos anteriores hemos descrito los mecanismos para obtener trayectorias basándonos en curvas que a su vez se construyen utilizando otras curvas y superficies. En este capítulo vamos a describir la forma en que se definen las herramientas virtuales y qué correspondencia tienen con las herramientas de que se disponga físicamente.

7.2 DEFINICIÓN DE LAS HERRAMIENTAS VIRTUALES

En nuestro modelo, una herramienta virtual es un objeto geométrico que se desplaza siguiendo una curva denominada trayectoria. Conforme avanza, se produce una disminución del volumen del objeto inicial en la cantidad y forma determinada por la herramienta. Desde la perspectiva del modelo geométrico, puede verse este proceso como una secuencia de diferencias booleanas sucesivas entre el objeto original y la herramienta. Desde el punto de vista de la fabricación del modelo, lo interpretamos como el desbaste que produciría, si fuera real, la herramienta sobre el material al desplazarse según la trayectoria. El fin último del modelo, como puede deducirse fácilmente, es la obtención de trayectorias reales que sean seguidas por herramientas reales a partir de las trayectorias virtuales que son seguidas por herramientas virtuales.

Una herramienta virtual puede tener o no correspondencia con una herramienta real. Si es así, la obtención de la trayectoria real de la herramienta en forma de programa de control numérico es prácticamente inmediata. No ocurre lo mismo si la herramienta no existe en la realidad. En tal caso será necesario emular su comportamiento utilizando herramientas que sí estén disponibles.

Sea una herramienta y un plano que la corta longitudinalmente según un eje dado. Llamaremos perfil de la herramienta a la curva que resulta de la intersección del plano con la superficie de la herramienta. La definición de las herramientas la realizaremos basándonos en ese perfil. Además, para cada herramienta definiremos un punto de aplicación, que será el punto que se desplaza siguiendo la trayectoria, y un eje de incidencia por el que se une la herramienta a la máquina y que es su eje de rotación.

7.2.1 Herramientas reales disponibles

Las herramientas reales concretas de que se dispone van a depender del sistema concreto sobre el que estemos trabajando. No obstante, podemos definir un conjunto estándar de herramientas que son las de uso más extendido: herramientas esféricas, cilíndricas, cónicas y tóricas.

Una herramienta *esférica* se puede definir mediante una tupla $h_{esf}=(r,a,v)$, y es una esfera de radio r unida a la máquina mediante un eje que sigue una dirección v y alrededor del cual gira. Su perfil es una circunferencia de radio r y su punto de aplicación a se encuentra situado en el centro de la esfera.

Las herramientas *cilíndricas* $h_{cil}=(r,h,a,v)$ están formadas por un cilindro de radio r y altura h , que gira respecto a su eje central longitudinal definido por el vector director v . El punto de aplicación a se encuentra en el centro de la base circular opuesta a donde se encuentra el eje de giro. Su perfil es un rectángulo de dimensiones $2rxh$.

En una herramienta *cónica* $h_{con}=(r,a,v,h)$ la figura que la define es un cono de altura h y radio r . Como las demás herramientas, ésta gira con respecto a su eje v . El punto de aplicación coincide con su vértice, mientras que su perfil es un triángulo isósceles de base $2r$ y altura h .

Por último una herramienta *tórica* $h_{tor}=(r,r',a,v)$ es un toro de radio principal r y radio del tubo r' , con punto de aplicación a (coincidente con el centro del toro) y orientada según la dirección de incidencia v que es el eje de rotación. El perfil está formado por dos circunferencias de radio r' , separadas una distancia r cada una del punto de aplicación en dirección normal al eje.

En la figura 7.1 se muestra gráficamente la forma de estos tipos de herramientas y sus correspondientes secciones.

7.2.2 Diseño del perfil de las herramientas virtuales

En el caso de las herramientas virtuales el perfil puede corresponderse o no con el de herramientas reales que existan en el mercado. No sólo eso, sino que también es posible que, aún teniendo una forma estándar, no dispongamos en realidad de una herramienta con las dimensiones de la herramienta virtual utilizada. En cualquier caso debemos determinar la forma en que la herramienta va a ser emulada con las herramientas reales.

Por otro lado, el perfil de las herramientas virtuales puede o no ser simétrico con respecto al eje, dependiendo del tipo de comportamiento que tenga la herramienta, tal y como veremos en apartados posteriores.

El primer paso para definir una herramienta es determinar su perfil utilizando una curva sobre el plano de corte. Esta curva puede adaptarse a una de las formas estándar de herramienta (cilíndrica, esférica, cónica o tórica) o tratarse de una curva con forma libre. Si la

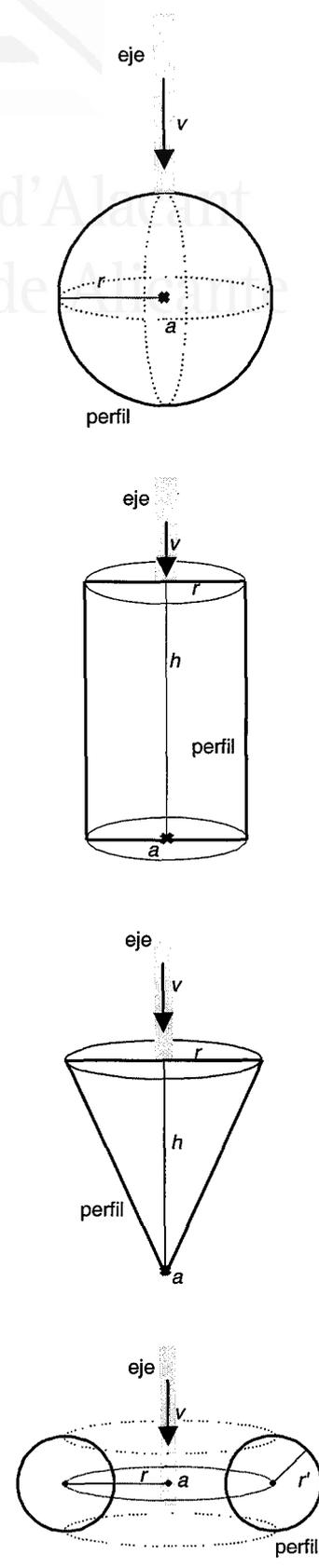


Figura 7.1
Forma y perfil de las
herramientas reales
disponibles

herramienta es simétrica, sólo será necesario especificar la mitad de la curva.

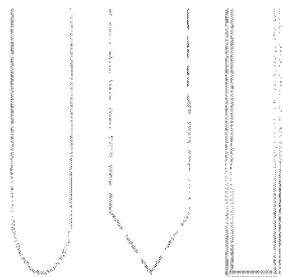
7.3 EMULACIÓN DE LAS HERRAMIENTAS VIRTUALES

Una vez definido el perfil de la herramienta, debemos determinar de qué manera va a ser emulada con una herramienta real. Para ello debemos establecer qué herramientas reales utilizaremos durante la fabricación y qué posición relativa ocupan con respecto al punto de aplicación de la herramienta virtual.

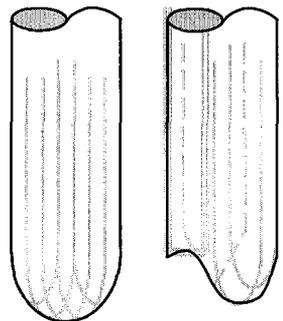
Las herramientas virtuales no están sujetas a las restricciones que marcan el comportamiento de las herramientas reales. Su comportamiento puede ser cualquiera, aunque si pretendemos fabricar posteriormente el objeto diseñado debemos tener presente que es necesario que pueda establecerse una forma físicamente posible de emular ese comportamiento.

En nuestra propuesta contemplamos dos formas básicas de comportamiento:

- Herramientas de rotación: la herramienta realiza un movimiento de giro alrededor de su eje conforme se va desplazando. Su comportamiento se corresponde con el real.
- Herramientas de traslación: en este caso no existe la rotación, por lo que su comportamiento es distinto del real. La herramienta realiza un barrido al desplazarse sobre la trayectoria.



Herramientas reales



Herramientas virtuales

7.3.1 Posicionamiento de las herramientas reales

Dada una herramienta virtual, emularla significa sustituir su acción sobre el objeto por la de una o varias herramientas, reales o también virtuales a su vez, de tal manera que el resultado de la acción de éstas sea lo más cercano posible al de aquella. La forma más sencilla de describir la emulación es colocar las herramientas emuladoras de tal manera que su perfil coincida en la medida de lo posible con el de la herramienta emulada (figura 7.2).

El sistema permite utilizar herramientas virtuales para emular otras herramientas, con lo que se consigue mucha mayor potencialidad. Este procedimiento tiene características recursivas, siendo el caso base el de las herramientas de más bajo nivel, cuya emulación debe ser, evidentemente, a través de herramientas reales.

El posicionamiento de las herramientas emuladoras para que sus perfiles se ajusten al perfil de la herramienta virtual puede hacerse manual o automáticamente.

El posicionamiento manual deja al usuario libertad para situar las herramientas. La calidad de la emulación dependerá en este caso de la habilidad del usuario para definir la posición de las herramientas.

Figura 7.2
Emulación de dos
herramientas virtuales con
tres reales

El posicionamiento automático resulta más complejo. El problema a resolver es el del recubrimiento de un área (el perfil de la herramienta virtual) con un conjunto de áreas más pequeñas y de formas diversas (el perfil de las herramientas emuladoras). Además, debemos introducir una restricción adicional al problema que es la de evitar colisiones del material con el eje de la herramienta.

La obtención de un método general para obtener el posicionamiento de herramientas para simular el comportamiento de otra es una tarea altamente compleja. Por esa razón se ha procedido al diseño de un conjunto de heurísticas que resuelvan los casos particulares más habituales.

7.3.2 Emulación de una herramienta cilíndrica con otra cilíndrica de radio inferior

Este es quizás el caso más sencillo de resolver y que mejores resultados proporciona. De hecho, con un número adecuado de herramientas emuladoras, la simulación puede ser exacta.

Sea una herramienta virtual cilíndrica H_{cil} de radio R y sea otra herramienta cilíndrica h_{cil} de radio r , tal que $r < R$. Sea s el solapamiento mínimo deseado entre dos pasadas de la herramienta h_{cil} . El mínimo número de pasadas p para emular la herramienta H_{cil} con la h_{cil} es

$$p = \left\lceil \frac{2R - s}{2r - s} \right\rceil \quad \text{Ecuación 7.1}$$

El solape real s' que se produce entre dos pasadas es:

$$s' = \frac{2pr - 2R}{p - 1} \quad \text{Ecuación 7.2}$$

Si suponemos el origen de coordenadas situado sobre el punto de aplicación A de la herramienta H_{cil} , la disposición de los puntos de aplicación a_i de la herramienta h_{cil} en cada pasada $0 \leq i \leq p-1$ son:

$$a_i = (-R + r + i(2r - s'), 0) \quad \text{Ecuación 7.3}$$

En la figura 7.3 podemos observar la emulación de una herramienta cilíndrica con otra de inferior radio.

7.3.3 Emulación de una herramienta esférica con otra esférica de radio inferior

En este caso se plantea la emulación situando la herramienta emuladora radialmente con respecto a la herramienta virtual, tal y como puede apreciarse en la figura 7.4.

El error cometido en este caso, depende de los radios de las herramientas y del número de pasos de emulación que utilizemos. Sea una herramienta esférica H_{esf} de radio R y sea otra herramienta también esférica h_{esf} de radio r , tal que $r < R$. En este caso fijaremos el número de pasos de emulación p , de tal manera que el reparto de la

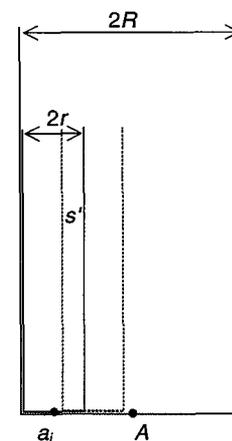


Figura 7.3
Emulación de una
herramienta cilíndrica

herramienta h_{esf} se realiza radialmente sobre la herramienta H_{esf} con una separación angular de α , donde α es

$$\alpha = \frac{\pi}{p} \tag{Ecuación 7.4}$$

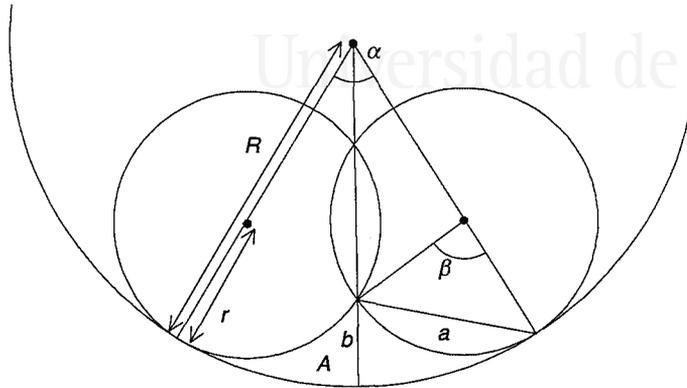
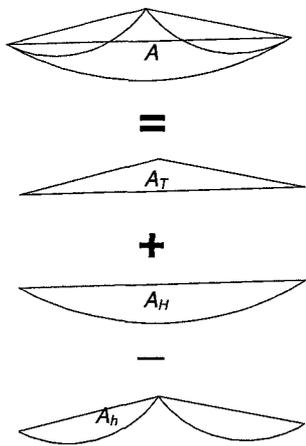


Figura 7.4 Emulación de una herramienta esférica



El error que se produce en el emulación viene dado por el número de pasos de emulación p , el radio de la herramienta virtual R , y el radio de la herramienta emuladora r . En concreto ese error es:

$$e = (p - 1) \cdot A \tag{Ecuación 7.5}$$

donde A es el área sin mecanizar que queda entre pasadas sucesivas de la herramienta emuladora, tal y como se muestra en la figura 7.4. Esta área puede calcularse según la siguiente expresión

$$A = A_T + A_H - 2 \cdot A_h \tag{Ecuación 7.6}$$

siendo A_T el área del triángulo, A_H el área del arco de circunferencia de la herramienta H_{esf} y A_h el área del arco de circunferencia de la herramienta h_{esf} (figura 7.5).

Estas áreas vienen dadas por las expresiones

$$A_T = R \operatorname{sen} \frac{\alpha}{2} \sqrt{a^2 - \left(R \operatorname{sen} \frac{\alpha}{2} \right)^2} \tag{Ecuación 7.7}$$

$$A_H = \frac{R^2}{2} (\alpha - \operatorname{sen} \alpha) \tag{Ecuación 7.8}$$

$$A_h = \frac{r^2}{2} (\beta - \operatorname{sen} \beta) \tag{Ecuación 7.9}$$

El segmento a puede calcularse como:

$$a = \sqrt{b^2 \operatorname{sen}^2 \frac{2\pi - \alpha}{4} + \left(2R \operatorname{sen} \frac{\alpha}{4} - b \cos \frac{2\pi - \alpha}{4} \right)^2} \tag{Ecuación 7.10}$$

Figura 7.5 Cálculo del área sin desbastar al emular una herramienta esférica

donde b es:

$$b = R - (R - r) \cos \frac{\alpha}{2} - \sqrt{r^2 - \left[(R - r) \sin \frac{\alpha}{2} \right]^2} \quad \text{Ecuación 7.11}$$

Por otro lado, el ángulo β se calcula como:

$$\beta = 2 \arcsen \frac{a}{2r} \quad \text{Ecuación 7.12}$$

Si suponemos el origen de coordenadas situado sobre el punto de aplicación de la herramienta H_{esf} , la disposición de los puntos de aplicación a_i de la herramienta h_{esf} en cada pasada $0 \leq i \leq p-1$ son:

$$a_i = ((R - r) \cos i\alpha, -(R - r) \sin i\alpha) \quad \text{Ecuación 7.13}$$

7.3.4 Emulación de una herramienta cónica con otra cónica de radio inferior

En este caso cabe distinguir dos situaciones: cuando la relación entre el radio y la altura es igual en las dos herramientas y cuando es distinta.

Si la relación entre radio y altura es igual, la emulación no produce errores. Sea una herramienta cónica H_{con} de radio R y altura H y sea otra herramienta cilíndrica h_{con} de radio r y altura h , tal que $r < R$ y $H/R = h/r$. Sea s el solapamiento mínimo deseado entre dos pasadas de la herramienta h_{con} . El mínimo número de pasadas p para emular la herramienta H_{con} con la h_{con} es

$$p = 2 \left\lceil \frac{\sqrt{R^2 + H^2} - s}{\sqrt{r^2 + h^2} - s} \right\rceil - 1 \quad \text{Ecuación 7.14}$$

El solape real que se produce entre dos pasadas es:

$$s' = \frac{(p+1)\sqrt{r^2 + h^2} - 2\sqrt{R^2 + H^2}}{p-1} \quad \text{Ecuación 7.15}$$

Si suponemos el origen de coordenadas situado sobre el punto de aplicación (vértice) de la herramienta H_{con} , la disposición de los puntos de aplicación a_i de la herramienta h_{con} en cada pasada $0 \leq i \leq p-1$ son:

$$a_i = \left(-\left(\frac{p-1}{2} - i \right) \left(r - \frac{rs'}{\sqrt{r^2 + h^2}} \right), \left(\frac{p-1}{2} - i \right) \left(h - \frac{hs'}{\sqrt{r^2 + h^2}} \right) \right) \quad \text{Ecuación 7.16}$$

En la figura 7.6 podemos observar la emulación de una herramienta cónica con otra de inferior radio pero igual relación entre radio y altura.

Cuando la herramienta simuladora tiene distinta relación altura/radio que la emulada, la emulación producirá errores de precisión. Sea una herramienta H_{con} de radio R y altura H y una herramienta emuladora h_{con} de radio r y altura h , de tal manera que $r < R$ y $H/R < h/r$, y sea p el número de pasos de emulación. Este valor

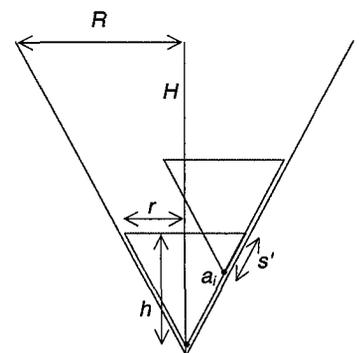


Figura 7.6
Emulación de una
herramienta cónica con otra
de igual relación altura/radio

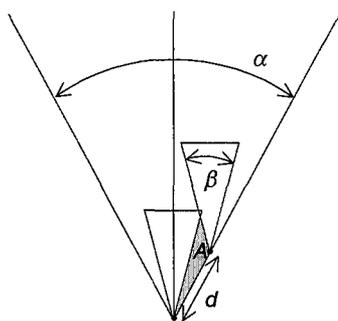


Figura 7.7
Emulación de una
herramienta cónica con otra
de mayor relación
altura/radio

p debe ser un número impar para que la herramienta emuladora h_{con} se aplique sobre el punto de aplicación de H_{con} . En este caso el error viene dado por la expresión:

$$e = p \cdot A \quad \text{Ecuación 7.17}$$

donde A es el área sin mecanizar que queda entre pasadas sucesivas de la herramienta emuladora, tal y como se muestra en la figura 7.7. Esta área puede calcularse según la siguiente expresión

$$A = \frac{d^2 (\cos \beta - \cos \alpha)}{4 \operatorname{sen} \beta} \quad \text{Ecuación 7.18}$$

donde α y β con los ángulos en el vértice de las herramientas H_{con} y h_{con} respectivamente y d es la distancia entre dos puntos de aplicación sucesivos. Estas cantidades vienen dadas por las expresiones:

$$d = \frac{2\sqrt{R^2 + H^2}}{p+1} \quad \text{Ecuación 7.19}$$

$$\cos \alpha = \frac{H^2 - R^2}{H^2 + R^2} \quad \text{Ecuación 7.20}$$

$$\cos \beta = \frac{h^2 - r^2}{h^2 + r^2} \quad \text{Ecuación 7.21}$$

$$\operatorname{sen} \beta = \frac{2rh}{\sqrt{r^2 + h^2}} \quad \text{Ecuación 7.22}$$

Si suponemos el origen de coordenadas situado sobre el punto de aplicación (vértice) de la herramienta H_{con} , la disposición de los puntos de aplicación a_i de la herramienta h_{con} en cada pasada $0 \leq i \leq p-1$ son:

$$a_i = \left(-\left(\frac{p-1}{2} - i \right) \frac{2R}{p+1}, \left| \frac{p-1}{2} - i \right| \frac{2H}{p+1} \right) \quad \text{Ecuación 7.23}$$

Existe una segunda posibilidad cuando la herramienta simuladora tiene distinta relación altura/radio que la emulada. Sea ahora una herramienta H_{con} de radio R y altura H y una herramienta emuladora h_{con} de radio r y altura h , tal que $r < R$ y $H/R > h/r$, y sea p un número impar de pasos de emulación. En este caso también se produce un error en el mecanizado que viene dado por la expresión:

$$e = (p-1) \cdot A + A_0 \quad \text{Ecuación 7.24}$$

siendo A , como antes, el área sin mecanizar que queda entre pasadas sucesivas de la herramienta emuladora y A_0 el área que queda en el vértice. La figura 7.8 muestra esta nueva situación. Estas áreas A y A_0 pueden calcularse según las siguientes expresiones

$$A = \frac{d^2 \operatorname{sen} \frac{\beta - \alpha}{2} \cos \frac{\alpha}{2}}{2 \cos \frac{\beta}{2}} \quad \text{Ecuación 7.25}$$

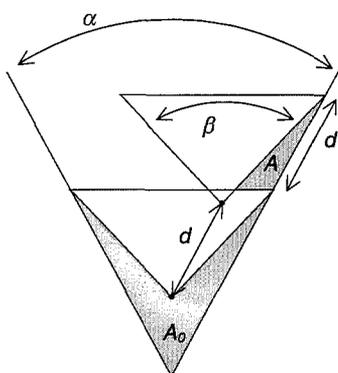


Figura 7.8
Emulación de una
herramienta cónica con otra
de menor relación
altura/radio

$$A_0 = \frac{r^2 H - rRh}{2R} \quad \text{Ecuación 7.26}$$

donde α y β son los ángulos en el vértice de las herramientas H_{con} y h_{con} respectivamente y d es la distancia entre dos puntos de aplicación sucesivos. Estas cantidades vienen dadas por las expresiones:

$$d = \frac{2(R-r)\sqrt{R^2 + H^2}}{R(p+1)} \quad \text{Ecuación 7.27}$$

$$\cos \frac{\alpha}{2} = \frac{H}{\sqrt{R^2 + H^2}} \quad \text{Ecuación 7.28}$$

$$\cos \frac{\beta}{2} = \frac{h}{\sqrt{r^2 + h^2}} \quad \text{Ecuación 7.29}$$

$$\text{sen} \frac{\beta - \alpha}{2} = \frac{rH - hR}{\sqrt{r^2 + h^2} \sqrt{R^2 + H^2}} \quad \text{Ecuación 7.30}$$

Si suponemos el origen de coordenadas situado sobre el punto de aplicación (vértice) de la herramienta H_{con} , la disposición de los puntos de aplicación a_i de la herramienta h_{con} en cada pasada $0 \leq i \leq p-1$ son:

$$a_i = \left(-\left(\frac{p-1}{2} - i \right) \frac{2(R-r)}{p+1}, \left| \frac{p-1}{2} - i \right| \frac{2(R-r)H}{R(p+1)} + \frac{r(H-R)}{R} \right) \quad \text{Ecuación 7.31}$$

7.3.5 Emulación de una herramienta tórica

Las herramientas tóricas tienen particularidades que hacen que los métodos de emulación sean específicos para cada caso. Debemos tener en cuenta que estas herramientas se utilizan habitualmente para mecanizar con máquinas de 3 ejes objetos que tienen partes cóncavas con respecto a la dirección de incidencia de la herramienta. Este tipo de trabajos son inabordables con otro tipo de herramientas.

El perfil de una herramienta tórica puede simplificarse, dando lugar a una curva como la de la figura 7.9. En esta curva, la zona de corte se localiza sobre el contorno exterior del toro. Esto permite mecanizar concavidades laterales en el material sin alterar la dirección de incidencia.

Dependiendo del tipo de desbaste que queramos realizar, la emulación de la herramienta debe hacerse de una manera u otra: mediante herramientas esféricas si no hay riesgo de colisión del eje con el material o mediante otras herramientas tóricas o cambiando la dirección de incidencia si existe ese riesgo.

A continuación presentamos las expresiones de cálculo de los puntos de aplicación y del error para la emulación de las herramientas tóricas con herramientas esféricas y tóricas. La tercera posibilidad apuntada requiere el cambio de posición del objeto con respecto al eje de la herramienta, posibilidad factible sólo en algunos casos y que requiere un tratamiento específico para cada uno.

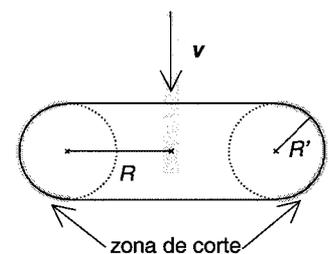


Figura 7.9
Perfil de una herramienta
tórica, con su zona de corte

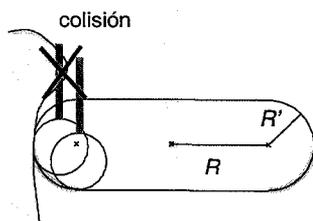


Figura 7.10
Emulación de una herramienta tórica con una esférica. Existe riesgo de colisiones

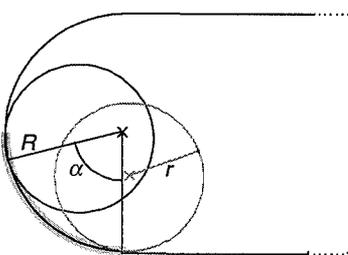


Figura 7.11
Parámetros para la emulación de una herramienta tórica con una esférica

7.3.5.1 Emulación mediante herramientas esféricas

La emulación de una herramienta tórica mediante herramientas esféricas cuyos vectores de incidencia coinciden sólo permite emular la mitad de la superficie de corte, tal y como se muestra en la figura 7.10. Además, tampoco será posible la emulación si existe el riesgo de colisión del eje de la herramienta con la superficie.

En estos casos la posición de los puntos de aplicación y el cálculo del error cometido son muy similares a los de la emulación de herramientas esféricas: Sea una herramienta tórica H_{tor} de radio principal R y radio del tubo R' y sea otra herramienta esférica h_{esf} de radio r , tal que $r < R'$. Como en el caso de las herramientas esféricas, dado el número de pasos de emulación p , el reparto de la herramienta h_{esf} se realiza radialmente sobre la herramienta H_{tor} , con una separación angular de α , donde α es:

$$\alpha = \frac{\pi}{p} \tag{Ecuación 7.32}$$

El error que se produce en el emulación viene dado por la misma expresión que en el caso de la emulación de herramientas esféricas, sustituyendo el valor de R en aquellas expresiones por el radio del tubo R' . Si suponemos el origen de coordenadas situado sobre el punto de aplicación de la herramienta H_{tor} , los puntos de aplicación a_i de la herramienta h_{esf} en cada pasada $0 \leq i \leq p-1$ se encuentran en la posición dada por esta ecuación:

$$a_i = \begin{cases} (R + (R' - r)\cos i\alpha, -(R' - r)\sin i\alpha) & \text{si } i\alpha \leq \frac{\pi}{2} \\ (-R + (R' - r)\cos i\alpha, -(R' - r)\sin i\alpha) & \text{si } i\alpha > \frac{\pi}{2} \end{cases} \tag{Ecuación 7.33}$$

7.3.5.2 Emulación mediante herramientas tóricas

La utilización de herramientas tóricas con otras herramientas tóricas de menor tamaño permite solucionar algunos de los problemas descritos en el apartado anterior. Ahora ya será posible emular toda el área de corte de la herramienta virtual, aunque sigue existiendo el problema de la colisión del eje de la herramienta con el material.

Como en el caso anterior, vamos a basarnos en los cálculos de la emulación de herramientas esféricas para deducir las expresiones para el caso de las herramientas tóricas. Sea ahora una herramienta tórica H_{tor} de radio principal R y radio del tubo R' y sea otra herramienta también tórica h_{tor} de radios r y r' , tal que $r < R$ y $r' < R'$. Si el número de pasos de emulación es p , la separación angular α entre sucesivas posiciones de la herramienta h_{tor} sobre el perfil de H_{tor} viene dada por la expresión siguiente.

$$\alpha = \frac{2\pi}{p} \tag{Ecuación 7.34}$$

Hemos de hacer notar que esta emulación repetida se producirá a ambos lados del tubo del toro que forma la herramienta virtual.

De nuevo podemos reutilizar las expresiones de cálculo del error deducidas para la emulación de herramientas esféricas, sin más que cambiar los valores de los radios R y r por los de los radios de los tubo de nuestras herramientas R' y r' respectivamente. Suponiendo el origen de coordenadas en el punto de aplicación de la herramienta H_{tor} , los puntos de aplicación a_i de la herramienta h_{esf} en cada pasada $0 \leq i \leq p-1$ vienen dados por la siguiente ecuación.

$$a_i = \begin{cases} ((R-r) + (R'-r') \cos i\alpha, (R'-r') \sin i\alpha) & \text{si } 0 < i\alpha \leq \frac{\pi}{2} \text{ ó } \frac{3\pi}{2} < i\alpha \leq 2\pi \\ ((r-R) + (R'-r') \cos i\alpha, (R'-r') \sin i\alpha) & \text{si } \frac{\pi}{2} < i\alpha \leq \frac{3\pi}{2} \end{cases}$$

Ecuación 7.35

7.4 CÁLCULO DE TRAYECTORIAS

La finalidad principal de todos los métodos que hemos ido describiendo para la definición y emulación de las herramientas virtuales es doble: por un lado facilitar el diseño de los objetos y por el otro mejorar la calidad del mecanizado.

En este último apartado de este capítulo trataremos el problema de convertir las trayectorias de herramientas virtuales en trayectorias reales para recorrerlas con herramientas reales. De esta manera realizaremos el paso completo de la fase de diseño a la fase de fabricación del objeto.

Una de las ventajas del uso de herramientas virtuales es la posibilidad de describir herramientas cuyo comportamiento no se corresponda con el real. Nosotros contemplamos aquí dos comportamientos básicos, herramientas de traslación y de rotación, aunque en cada aplicación concreta podrían describirse otros adaptados a las necesidades que surjan.

7.4.1 Herramientas de traslación

Definimos una herramienta virtual de traslación como aquella que se desplaza siguiendo su trayectoria sin rotar. Evidentemente este comportamiento no se corresponde con el de una herramienta real, pero permite definir formas geométricas mucho más generales que utilizando únicamente herramientas de rotación.

Sea una herramienta virtual de traslación H con punto de aplicación A y vector de incidencia V . Sea un conjunto de herramientas h_i que la emulan, cada una de ellas con su punto de aplicación a_i y un vector de incidencia v_i . Sea una trayectoria T para la herramienta virtual, definida como una lista ordenada de puntos T_j . Para cada herramienta h_i debemos calcular una nueva trayectoria t_i de tal manera que el resultado de desbastar el material utilizando las herramientas reales h_i sobre las trayectorias t_i sea el mismo que se obtendría en el hipotético caso de poder hacer pasar la herramienta virtual H sobre la trayectoria T .

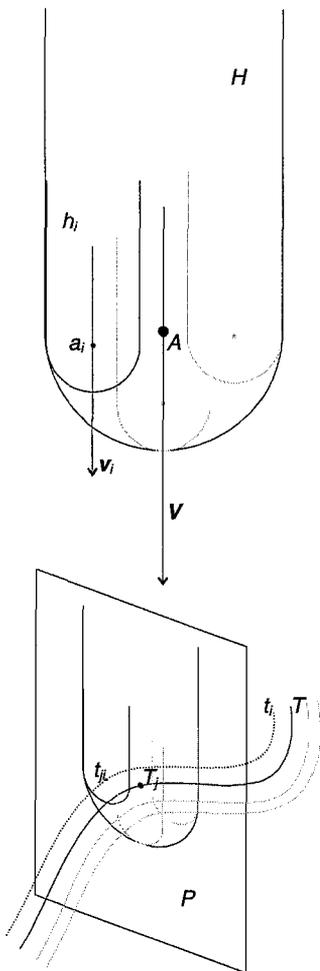


Figura 7.12
Cálculo de las trayectorias
reales para una herramienta
de traslación

Para cada punto $T_i \in T$ de la trayectoria inicial definimos el plano P de posición de la herramienta original como el plano que contiene ese punto y es normal a la trayectoria. Para cada una de las herramientas virtuales h_i calculamos la posición de su punto de aplicación a_i transformándolo para llevarlo sobre el plano P , siguiendo la orientación marcada por el vector de incidencia V de la herramienta virtual H . Cada punto de aplicación a_i transformado será un nuevo punto t_{ij} de la trayectoria real t_i , correspondiente a su herramienta real h_i . En la figura 7.12 se muestra gráficamente el funcionamiento del método.

7.4.2 Herramientas de rotación

Una herramienta virtual es de rotación si gira conforme se desplaza sobre su trayectoria. En este caso el comportamiento sí se corresponde con el de una herramienta real.

El cálculo de la posición de los puntos de aplicación de las herramientas emuladoras será el mismo que en el caso de las herramientas de traslación. Aprovechando la simetría de este tipo de herramientas en cada paso podremos calcular sólo la mitad de los puntos y obtener los otros mediante una transformación de reflexión con respecto a la dirección de incidencia V de la herramienta virtual.

Una vez calculadas todas las trayectorias de las herramientas emuladoras, debemos realizar un cálculo adicional en los extremos de las trayectorias para simular el procedimiento de rotación. Sea t_i la trayectoria real para una herramienta real h_i , y sea t'_i la trayectoria correspondiente a la posición simétrica de h_i al otro lado del vector V . Debemos unir el último punto de t_i con el de t'_i con una semicircunferencia de radio igual a la distancia perpendicular que separa el punto de aplicación a_i de la herramienta h_i del vector de incidencia V y en dirección hacia el exterior de la trayectoria. Este procedimiento puede verse en la figura 7.13.

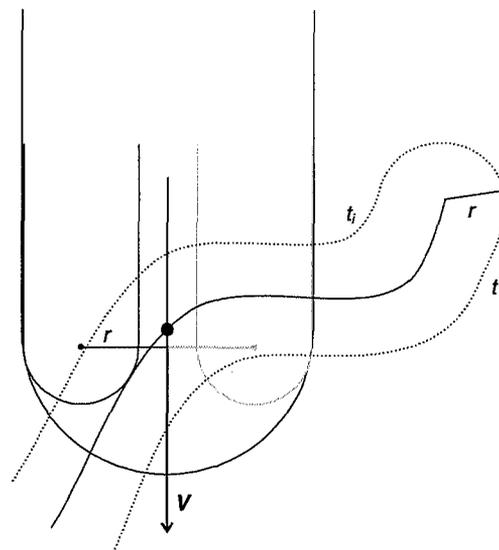


Figura 7.13
Cálculo de las trayectorias reales para una herramienta de rotación



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 8

CONSTRUCTORES DE OBJETO

8.1 INTRODUCCIÓN

Para que un modelo geométrico sea realmente útil es fundamental una visualización lo más realista posible de los objetos diseñados. Este proceso es tan importante por diversos motivos:

- Disponer de una vista previa del modelo antes de la fabricación permite detectar errores que no es posible determinar con la representación matemática del modelo.
- Los usuarios no técnicos comprenden con mucha mayor facilidad una información visual realista.
- Es posible realizar catálogos para mostrar los productos antes de fabricarlos. Esto es especialmente importante en objetos de diseño.

Representar mediante una visualización realista un objeto obtenido utilizando el modelo Superficie-Trayectoria tiene algunas particularidades. Hemos de tener en cuenta que no disponemos de las superficies que forman el objeto, con lo que no es posible realizar un mallado directo. En nuestro modelo se trata, de alguna manera, de realizar una simulación del mecanizado de la pieza y visualizar el resultado final.

Por otro lado, este proceso nos permite construir el objeto completo, por lo que todos estos métodos constituyen los constructores semánticos de objeto.

Dos aproximaciones son las más adecuadas: visualización directa del modelo y obtención de un modelo secundario poligonal adecuado para la visualización.

Entre las técnicas de visualización directa destaca el trazado de rayos. Dada una escena, una posición para el observador y una rejilla de píxeles, coincidente con la pantalla de visualización o *raster*, se traza un rayo desde el observador a través de cada píxel de la rejilla y se calcula la primera intersección con un objeto de la escena. Ese es el píxel que debe representarse en la pantalla. El color en que debe visualizarse el píxel está formado por tres componentes:

- Componente local, calculada según el modelo de iluminación de Phong.
- Componente reflexiva, que da lugar a un nuevo rayo que se traza en la dirección de reflexión de la luz.
- Componente refractiva, si el objeto tiene algún grado de transparencia, que también da lugar a un nuevo rayo, en este caso trazado en la dirección de refracción de la luz.

Cada componente viene ponderada por un factor que determina el peso de ese tipo de iluminación en el resultado final. Así, podemos hacer que los objetos sean completamente opacos y sin reflejos si el peso de las componentes reflexiva y refractiva es 0. Un valor alto para la componente reflexiva da lugar a objetos que reflejan a los de su entorno, mientras que un valor alto de la componente refractiva hace que los objetos sean transparentes. Los nuevos rayos a que dan lugar la reflexión y la refracción se tratan de igual manera al rayo inicial.

Se trata de una técnica que proporciona imágenes de gran realismo, pero que tiene un alto coste computacional, debido a que el cálculo de las intersecciones no es sencillo. Además, al ser un método punto a punto, un cambio en la posición del observador conlleva el recálculo de todos los píxeles de la escena. En la figura 8.1, encontramos una representación gráfica del funcionamiento de este algoritmo. Este técnica proporciona resultados de alta calidad.

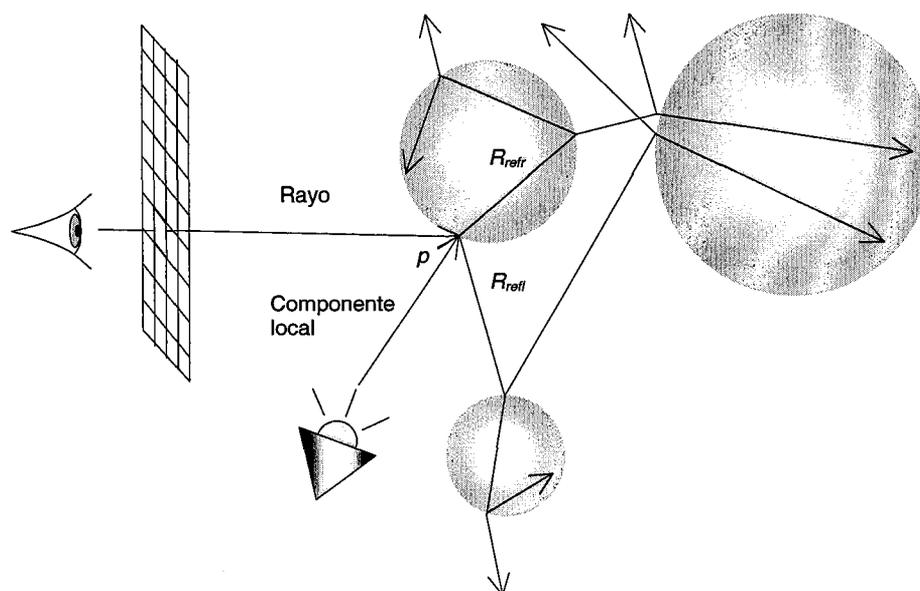


Figura 8.1

Algoritmo de trazado de rayos. En el punto p , la iluminación es la suma ponderada de la componente local (Phong), de la reflexiva (R_{refl}) y de la refractiva (R_{refr})

La obtención de un modelo secundario de tipo poligonal tiene dos grandes ventajas sobre el trazado de rayos:

- Una vez obtenido el modelo poligonal, su visualización es muy rápida debido a que las tarjetas gráficas actuales disponen de aceleradores especialmente dedicados a representar polígonos.
- El modelo poligonal es independiente de la vista, es decir, una vez calculado no varía con la posición del observador.

A cambio, la obtención de un modelo poligonal a partir del modelo Superficie-Trayectoria no resulta una tarea sencilla.

En el presente trabajo hemos optado por desarrollar un método de obtención de un modelo secundario poligonal a partir del objeto diseñado en nuestro modelo. Para ello se ha partido de un algoritmo bien conocido como puede ser *Marching Cubes* [Lorenson, 1987], adaptado a las características de nuestro problema.

8.2 EL ALGORITMO *MARCHING CUBES*

La idea general del algoritmo *Marching Cubes*, diseñado por Lorensen y Cline [Lorensen, 1987], consiste en dividir el volumen de un paralelepípedo que encierra al objeto a representar en un número de cubos de igual tamaño. A continuación, se hacen intersectar los cubos con la superficie del objeto de manera que, dependiendo de la posición de los vértices de cada cubo (exteriores o interiores) y de los cortes, se poligonaliza (utilizando triángulos) el objeto de una u otra forma, de entre las catorce posibilidades básicas de corte de un cubo con una superficie.

8.2.1 Características del algoritmo

El algoritmo de *Marching Cubes* aparece descrito en dos artículos de Wyvill y McPheeters [Wyvill, 1986] y de Lorensen y Cline [Lorensen, 1987]. Se trata de un algoritmo diseñado para obtener una malla poligonal a partir de un volumen en el que no existe información superficial. En el caso del artículo de Wyvill y McPheeters, el método se utiliza para visualizar una isosuperficie formada por un campo definido por un conjunto de puntos, como pueden ser los objetos *Blob* o los *Metaballs*. Por el contrario, Lorensen y Cline lo desarrollaron inicialmente con el objeto de mostrar datos médicos tridimensionales. En nuestro trabajo proponemos utilizar una extensión de este algoritmo para obtener un modelo superficial de un objeto generado utilizando el modelo Superficie-Trayectoria.

El algoritmo original de *Marching Cubes* presenta algunas ventajas sobre otros métodos a la hora de poligonalizar volúmenes, como son su sencillez de implementación, el relativo bajo coste computacional y el hecho de que la malla resultante sea igualmente sencilla. También tiene, no obstante algunos inconvenientes: no se puede mejorar la precisión localmente, puede provocar huecos en la malla y, en ocasiones, no preserva la conectividad de los objetos. Se han presentado, sin embargo, algunas mejoras que se comentan en apartados posteriores.

8.2.2 Método

El primer paso en este algoritmo es determinar el paralelepípedo que encierra al objeto. Seguidamente se divide dicho paralelepípedo en cubos de tamaño constante. Este tamaño será definido por el usuario dependiendo de la precisión que desee para la poligonalización final.

A continuación se analiza cada cubo por separado. Debemos determinar qué vértices del cubo son interiores al objeto y cuáles exteriores. Para ello se suele utilizar un vector de 8 bits que define la posición de cada cubo con respecto al objeto. Cada elemento del vector hace referencia a un vértice del cubo (lo que implica que debemos hacer una ordenación siempre igual de los vértices del cubo) y almacena un 1 si el vértice es interior o un 0 si es exterior.

Una vez obtenida la información sobre la colocación del cubo con respecto al objeto, debe procederse a estudiar qué forma tendrá la porción de superficie asociada a dicho cubo. De las 256 posibles maneras en que un cubo puede intersectar con una superficie (por la definición del vector tenemos $2^8=256$ combinaciones de los valores binarios), son sólo 16 los casos distintos, si eliminamos los casos de colocación físicamente imposibles, además de las simetrías y las rotaciones [Lorensen, 1987].

Cada uno de esos casos se poligonaliza según el esquema de la figura 8.2. La unión de los distintos triángulos obtenidos forma la malla completa el objeto.

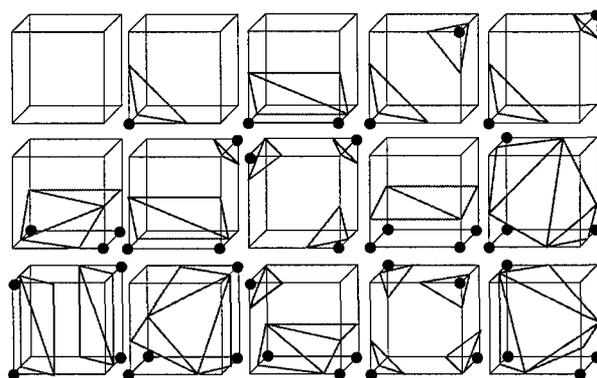


Figura 8.2

Casos posibles de intersección de un cubo con una superficie. Los vértices en negro son vértices interiores al objeto; el resto son exteriores. Se indica, además, la triangulación asociada en cada caso

8.2.3 Limitaciones y mejoras

Marching Cubes y la mayoría de los algoritmos derivados del mismo tienen el problema común de que en todo momento se analizan cubos de igual tamaño sea cual sea la región que se esté estudiando. Esto produce que el algoritmo obtenga un conjunto de triángulos que puede no ser óptimo para el objeto a representar. Algunas variantes del método tratan de mejorar este aspecto. Por ejemplo, Payne y Toga [Payne, 1990] proponen descomponer los cubos que cortan a la superficie en cinco tetraedros, que a su vez vuelven a cortar a la superficie, con lo que se obtiene una malla más suave que con la técnica original. El problema de dicha técnica es que es equivalente a disminuir el tamaño de cubo, aumentando de forma global el conjunto de triángulos y por tanto generando concentraciones excesivas en zonas donde no es necesario. Otra propuesta [Puchol, 1997], [Puchol, 1999], [Sáez, 2000] define una medida de precisión local que estudia cada región de la superficie (en este caso se trata de isosuperficies tipo *blob*) con diferente nivel de detalle, determinando el modelo de fronteras más aproximado a dicha forma.

Otro problema característico del algoritmo inicial, es la posible existencia de huecos en la malla poligonal en determinados casos.

Varios autores proponen una solución añadiendo algunos casos más de posibles intersecciones del cubo con la superficie [Payne, 1990].

Por otro lado, también se puede presentar un problema de coherencia entre el volumen real y el poligonalizado, provocando que en algunos casos la conectividad del objeto mallado no sea la misma que la del sólido original. Una solución aportada es el algoritmo *Volume preserving Marching Cubes* [Kaneko, 1997].

Un aspecto crucial en este algoritmo es la determinación del tamaño óptimo para el cubo, de manera que se alcance una precisión adecuada sin aumentar en exceso el número de polígonos. Para ello se suele determinar este tamaño en base a la información geométrica del objeto original, tomando dicho tamaño en función del objeto más pequeño que tenga la escena. De esta forma se asegura que ningún objeto se queda totalmente incluido en un cubo con lo que no obtendríamos intersección y perderíamos la porción de volumen.

Por último, cabe remarcar que este tipo de algoritmos ha sido pensado para isosuperficies en las que es muy sencillo determinar si un punto en el espacio se encuentra dentro o fuera de la superficie. En nuestro caso, determinar la posición no es tan sencillo y requiere adaptar convenientemente el algoritmo a nuestro caso particular. En el siguiente apartado desarrollamos este método.

8.3 POLIGONALIZACIÓN DEL MODELO SUPERFICIE-TRAYECTORIA

El algoritmo de obtención del modelo secundario poligonal para visualización de objetos definidos utilizando el modelo Superficie-Trayectoria, es una variación del algoritmo *Marching Cubes*. Igual que éste, su funcionamiento parte de una división del paralelepípedo que engloba al objeto (en este caso el material en el que se va a mecanizar el objeto) en un conjunto de cubos de igual tamaño, calculando la interioridad/exterioridad de los vértices de cada cubo y poligonalizando cada cubo según los 14 casos ya estudiados.

La mayor diferencia con el algoritmo básico es que el modelo poligonal se va obteniendo de forma incremental, es decir, se obtiene un modelo completo para cada posición discreta de la herramienta, y no un modelo al final del proceso como en el caso original.

8.3.1 Algoritmo

Sea un objeto generado utilizando el modelo Superficie-Trayectoria. Podemos definir un objeto de este tipo como un par $O=(M,T)$, donde M es un paralelepípedo de material y T un conjunto de trayectorias, definidas como hemos descrito en capítulos anteriores.

Por simplificar y sin pérdida de generalidad, el material vendrá definido como un par de puntos en el espacio $M=(m_{min},m_{max})$, correspondientes a los puntos de coordenadas máxima y mínima que determinan el paralelepípedo. Si originalmente este paralelepípedo

no tuviera sus lados paralelos a los planos coordenados, bastaría con una rotación adecuada para hacer que así ocurriera.

El conjunto de trayectorias vendrá dado por un conjunto de pares $T=\{(c_1,h_1),(c_2,h_2),\dots,(c_n,h_n)\}$ donde cada c_i es una curva y cada h_i la herramienta que pasa por esa curva. La definición de la curva dependerá del modelo utilizado para diseñarla (una polilínea, una curva BSpline, una curva Bézier, ...), mientras que la herramienta se define como una tupla con sus parámetros, que dependerán del tipo de herramienta. Por ejemplo, una herramienta esférica vendrá dada por la tupla $h_i=(r_i,a_i,v_i)$, donde r_i es el radio de la herramienta, a_i el punto de aplicación y v_i su vector de incidencia. En cualquier caso, supondremos que un punto de la trayectoria marca la posición del punto de aplicación de la herramienta correspondiente.

El modelo Superficie-Trayectoria puede verse como un modelo CSG en el que la operación a realizar es:

$$O = M \sim \bigcup_{i=1}^n (c_i, h_i) \quad \text{Ecuación 8.1}$$

donde M es el paralelepípedo de material y cada (c_i, h_i) lo podemos ver como el objeto formado al desplazar la herramienta h_i a través de la curva c_i .

El problema que tratamos de resolver, dadas estas definiciones, es el de construir un modelo poligonal formado por una malla de triángulos que aproxime el objeto resultado de mecanizar el paralelepípedo de material haciendo pasar por las trayectorias las herramientas correspondientes, o lo que es lo mismo, que aproxime el objeto CSG de la ecuación 8.1.

El algoritmo tendrá como entrada el objeto O y como salida una malla triangular L .

El primer paso será dividir el paralelepípedo M en un conjunto disjunto de cubos de igual tamaño $V=\{v_j\}$, a los que llamaremos vóxeles, de manera que

$$(M = \bigcup v_j) \wedge (\bigcap v_j = \phi) \quad \text{Ecuación 8.2}$$

Cada vóxel se define mediante un par $v_j=(A_j, V_j)$, siendo A_j un conjunto de 12 aristas y V_j un conjunto de 8 vértices. Cada vértice es un valor binario, de manera que 0 indica que el vértice es exterior al objeto y 1 que es interior. Por su lado, cada arista almacena un valor en el intervalo $[0,100]$, que indica el porcentaje de interioridad de la arista, es decir, aquellas aristas cuyos vértices sean interiores almacenarán el valor 100%; aquellas cuyos vértices sean exteriores tendrán un valor de 0% y aquellas cuyos vértices sean uno interior y el otro exterior, guardarán un valor intermedio que indique qué tanto por ciento de esa arista es interior al objeto. La figura 8.3 representa gráficamente los valores de vértices y aristas.

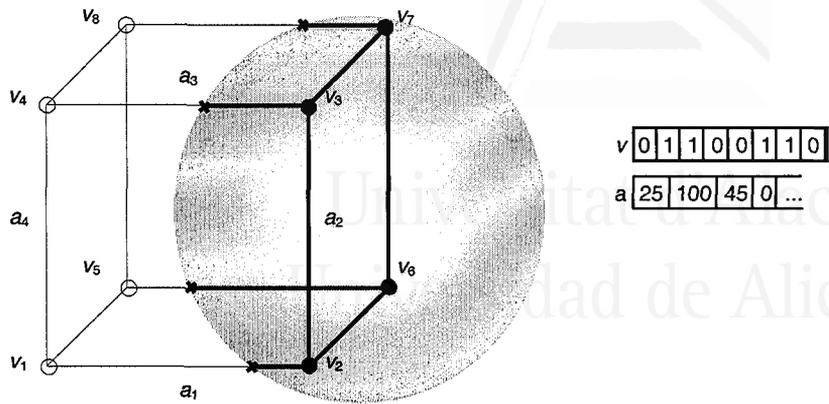


Figura 8.3
Codificación de los vértices y de algunas aristas del cubo para *Marching Cubes*. En negro se muestran los vértices interiores

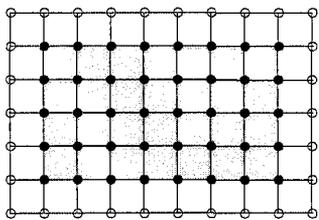


Figura 8.4
Vista bidimensional de la inicialización de la matriz de vóxeles. Se añade una fila de cubos en cada lado del paralelepípedo

Los vóxeles se guardan en una matriz tridimensional, añadiendo una capa más de vóxeles por cada lado del paralelepípedo. A continuación, la matriz se inicializa con los valores adecuados para los vértices y aristas: así, todos los vértices interiores al paralelepípedo (todos los que lo forman) se etiquetan con 1, mientras que los exteriores (los de las capas adicionales exteriores) se etiquetan con 0. De igual modo se asigna valor a las aristas: 100% las interiores y 0% las exteriores. La figura 8.4 muestra la inicialización de los vóxeles.

A continuación debemos comenzar el proceso con cada una de las trayectorias, restando al paralelepípedo inicial la herramienta en la posición marcada por la curva. El proceso es incremental, es decir, la ecuación 8.1 se aplica restando a cada paso al paralelepípedo la herramienta en la posición dada. La curva c_i por donde pasa la herramienta es una curva continua, por lo que es necesaria una discretización adecuada que permita realizar el proceso en pasos discretos (figura 8.5). Sea p_{max} el paso máximo de trayectoria. Discretizaremos c_i en un conjunto de puntos p_k tal que se cumpla

$$\forall k, |p_k - p_{k+1}| < p_{max} \tag{Ecuación 8.3}$$

donde $|p_k - p_{k+1}|$ es la distancia euclídea entre p_k y p_{k+1} . Si p_{max} está convenientemente escogido, evitaremos que la herramienta quede totalmente interior o exterior a uno de los cubos. En un apartado posterior trataremos el cálculo de p_{max} .

Llamemos c'_i a la discretización de la curva c_i . Para cada punto $p_k \in c'_i$, colocamos el centro de la herramienta en la posición p_k y procedemos a realizar la resta booleana del volumen del paralelepípedo y del volumen de la herramienta. Para ello comprobamos qué vértices de cada vóxel de la matriz son interiores o exteriores al volumen de la herramienta y los etiquetamos convenientemente. Evidentemente, conociendo el radio de la herramienta, podemos determinar un entorno de vecindad centrado en p_k y de radio el radio de la herramienta, de tal manera que sólo será necesario comprobar los vértices de aquellos vóxeles situados en ese entorno.

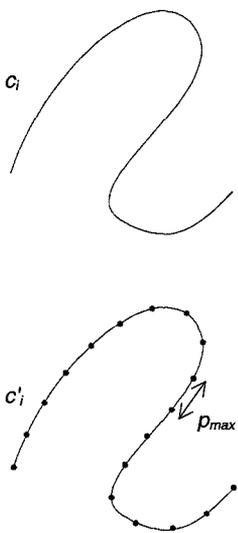


Figura 8.5
Discretización de una curva c_i con un paso máximo de herramienta de p_{max}

Una vez calculados y etiquetados los vértices interiores y exteriores, debemos asignar a cada arista el valor de corte adecuado. Se pueden dar los siguientes casos, dependiendo de las transformaciones que sufran los vértices:

- Si ambos vértices eran exteriores antes del procedimiento, no debe realizarse ninguna acción, puesto que ningún vértice puede pasar de ser exterior a ser interior.
- Si ambos vértices eran interiores antes del procedimiento pueden darse tres casos:
 - Si después del procedimiento siguen siendo interiores, no realizamos ninguna acción adicional.
 - Si después del procedimiento ambos pasan a ser a exteriores, tampoco debe realizarse acción alguna.
 - Si uno pasa a ser exterior, debemos intersectar la arista con el volumen de la herramienta y asignarle el porcentaje de interioridad correspondiente.
- Si un vértice era interior y otro exterior antes del procedimiento, pueden darse dos casos:
 - Si después del procedimiento ambos vértices son exteriores, no debe realizarse ninguna acción adicional.
 - Si después del procedimiento sigue habiendo uno interior y otro exterior, debemos calcular la nueva intersección de la arista con el volumen de la herramienta y asignar a la arista el mínimo entre el porcentaje de interioridad que ya tenía asignado y el recién calculado.

En la tabla 8.1 se resumen los casos posibles y las acciones a realizar. Determinar la interioridad/exterioridad de los vértices así como el punto de corte de las aristas con el volumen de la herramienta, va a depender de la geometría de ésta. En un apartado posterior trataremos estos temas.

Situación inicial	Situación final	Acción
		Acción 1: Intersectar arista-volumen y asignar porcentaje
		Ninguna
		Acción 1: Intersectar arista-volumen y asignar porcentaje
		Ninguna
		Ninguna
		Ninguna
		Acción 2: Intersectar arista-volumen y asignar porcentaje menor
		Acción 2: Intersectar arista-volumen y asignar porcentaje menor
		Ninguna

Tabla 8.1

Acciones a realizar dependiendo de la configuración de los vértices. En negro se muestran los vértices interiores, en blanco los exteriores

Este procedimiento debe repetirse para todos los puntos de todas las trayectorias que conforman el objeto.

Por último debemos generar la lista de triángulos asociada a cada vóxel siguiendo los 14 casos posibles de *Marching Cubes*.

En la tabla 8.2 se presenta el algoritmo descrito.

<p>ENTRADA O: Objeto a poligonalizar definido en el modelo S-T</p> <p>SALIDA L: Malla triangular</p> <p>VARIABLES M: paralelepípedo de material T: conjunto de trayectorias t: trayectoria V: conjunto de vóxeles almacenados en forma de matriz v: vóxel h: herramienta a: arista p: punto b1, b2, b1', b2': valores lógicos para indicar interior/exterior inter: entero</p> <p>ALGORITMO</p> <pre> M= O.Paralelepípedo() T = O.Trayectorias() V = M.DividirVóxeles() V.Inicializar() Para cada trayectoria t∈T hacer h=t.Herramienta() Para cada punto p∈t hacer Para cada vóxel v∈V ∧ v∈Entorno(p,h.Radio()) hacer Para cada arista a∈v hacer b1=a.PuntoInicial().EsInterior() b2=a.PuntoFinal().EsInterior() a.AplicarHerramienta(h) b1'= a.PuntoInicial().EsInterior() b2'= a.PuntoFinal().EsInterior() Si (b1=0 y b2=1) entonces Si (b2'=1) entonces a.Interioridad=PuntoCorte(a,h) Fin si Si no si (b1=1 y b2=0) entonces Si (b1'=1) entonces a.Interioridad=PuntoCorte(a,h) Fin si Si no si (b1=1 y b2=1) entonces Si (b1'≠b2') entonces inter=PuntoCorte(a,h) Si (inter<a.Interioridad) entonces a.Interioridad=inter Fin si Fin si Fin si Fin si Fin para Fin para Fin para </pre>
--

Tabla 8.2
Algoritmo de poligonalización del modelo S-T

8.3.2 Tamaño del cubo y del paso máximo de herramienta

La determinación del tamaño del cubo es fundamental en un algoritmo de *Marching Cubes*. Es necesario llegar a un compromiso entre precisión y eficiencia. Un cubo de un tamaño demasiado pequeño, genera una malla muy tupida pero excesivamente grande.

Por el contrario, si el cubo es demasiado grande, la malla será eficiente pero podemos perder detalles del objeto.

En el caso que nos ocupa, el detalle de menor tamaño en un objeto es, como mínimo, del tamaño de la menor de las herramientas utilizadas para mecanizarlo. Tal y como se ve en la figura 8.6, el tamaño máximo t_{max} del cubo, si r es el radio de la herramienta más pequeña, debe ser:

$$t_{max} = \frac{2r}{\sqrt{2}} \quad \text{Ecuación 8.4}$$

Este tamaño de cubo, no obstante, resulta demasiado grande. Empíricamente se ha determinado que un valor adecuado para el tamaño es $r/2$.

El otro aspecto importante para obtener una malla adecuada es el paso máximo de herramienta. De nuevo es necesario definir un paso adecuado. En este caso el paso no influye en el tamaño de la malla, pero sí en la calidad de la visualización de la trayectoria. Si el paso es demasiado grande puede visualizarse el efecto de la discretización de la curva. Un valor demasiado pequeño ralentizará el procedimiento en exceso. Se ha determinado empíricamente que un valor de $r/4$ es adecuado.

8.3.3 Prueba de interioridad de los vértices y cálculo de los puntos de corte

La prueba de interioridad de los vértices y el cálculo del punto de intersección de la herramienta con la arista depende del tipo de herramienta que utilicemos para mecanizar. En esta sección veremos cómo realizar la prueba y cómo calcular el punto de corte para cada tipo de herramienta [Hearn, 1995] [Hoff, 1996] [Dodgson, 2000] [Zorin, 2000] [Eberly, 2002].

En todos los casos, utilizaremos la ecuación paramétrica de la arista. Sea la arista que une los vértices p y q . Su ecuación paramétrica es:

$$\left. \begin{aligned} x &= p_x + (q_x - p_x)t \\ y &= p_y + (q_y - p_y)t \\ z &= p_z + (q_z - p_z)t \end{aligned} \right\} 0 \leq t \leq 1 \quad \text{Ecuación 8.5}$$

8.3.3.1 Herramientas esféricas

Sea una herramienta esférica definida por una tupla $h_{esf} = (r, a, v)$. Una herramienta de este tipo se compone de una esfera de radio r

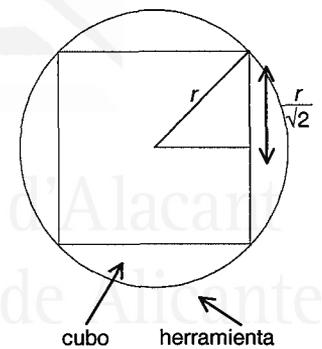
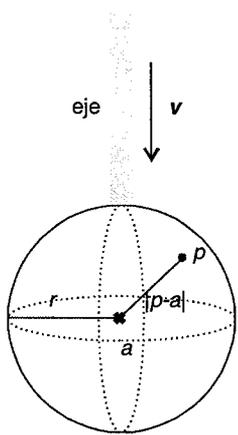


Figura 8.6
Determinación del tamaño máximo del cubo



cuyo centro coincide con el punto de aplicación de la herramienta a . Aunque evidentemente la herramienta se encuentra unida a la máquina a través de un eje (cuya dirección viene marcada por el vector de incidencia v) este eje no es necesario considerarlo durante la visualización, puesto que no contiene elementos de corte. La disposición de la esfera y del eje puede verse en la figura 8.7.

En una herramienta esférica, un punto p será interior a la herramienta si es interior a la esfera que forma la herramienta.

$$p \notin O \Leftrightarrow p \in h_{esf} \Leftrightarrow p \in \text{Esfera} \quad \text{Ecuación 8.6}$$

El punto p es interior a la esfera, y por lo tanto a la herramienta, si la distancia euclídea entre el punto y el centro de la esfera, que se corresponde con el punto de aplicación a , es menor que el radio de la herramienta r (figura 8.7). La prueba de interioridad para herramientas esféricas es:

$$p \notin O \Leftrightarrow p \in h_{esf} \Leftrightarrow |p-a| < r \quad \text{Ecuación 8.7}$$

Obsérvese que esta prueba de interioridad es equivalente a sustituir las coordenadas del punto en la ecuación de la esfera. La ecuación implícita de una esfera de radio r y centrada alrededor de un punto a es:

$$(x-a_x)^2 + (y-a_y)^2 + (z-a_z)^2 = r^2 \quad \text{Ecuación 8.8}$$

El punto de intersección entre una arista $p-q$ y la esfera se calcula sustituyendo la ecuación de la arista (ecuación 8.5) en esta ecuación. Esta sustitución da lugar a una ecuación de segundo grado con dos posibles soluciones. En tal caso, la solución válida es aquella cuyo valor para el parámetro t está en el intervalo $[0,1]$, ya que en otro caso el punto de corte se encuentra sobre la recta pero fuera de la arista finita. En el caso de no existir solución o de encontrarse ésta fuera del intervalo, no existe el punto de intersección.

8.3.3.2 Herramientas cilíndricas

Sea ahora una herramienta cilíndrica definida por la tupla $h_{cil}=(r,h,a,v)$. Este tipo de herramientas está formada por un cilindro de radio r y altura h , cuya base se encuentra situada en el punto de aplicación a y cuyo eje central se encuentra en la dirección de incidencia de la herramienta v (figura 8.8).

En una herramienta cilíndrica, un punto p será interior a la herramienta y, por lo tanto, exterior al objeto, si es interior al cilindro que forma la herramienta.

$$p \notin O \Leftrightarrow p \in h_{cil} \Leftrightarrow p \in \text{Cilindro} \quad \text{Ecuación 8.9}$$

La ecuación implícita de un cilindro infinito de radio r cuyo eje coincide con el eje Z es:

$$x^2 + y^2 = r^2 \quad \text{Ecuación 8.10}$$

Sea T la matriz de transformación que sitúa un cilindro genérico haciendo coincidir su eje con el eje Z y su punto de aplicación con el

Figura 8.7
Parámetros y prueba de interioridad para una herramienta esférica

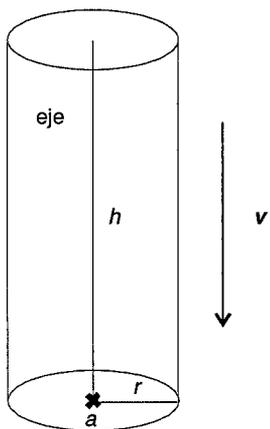


Figura 8.8
Parámetros de una herramienta cilíndrica

origen de coordenadas, es decir, la matriz que realiza la conversión al sistema de coordenadas local del objeto. Sea p' el punto resultante de aplicar la transformación T a un punto p , es decir, $p' = T \cdot p$. Sustituyendo las coordenadas de p' en la ecuación 8.10 podemos saber si el punto es interior al cilindro:

$$p \notin O \Leftrightarrow p \in h_{cil} \Leftrightarrow (p' = T \cdot p) \wedge (p_x'^2 + p_y'^2 < r^2) \wedge (0 < p_z' < h) \quad \text{Ecuación 8.11}$$

Obsérvese que además de comprobar que se cumple la inecuación, tenemos que asegurarnos de que la coordenada z del punto se encuentre entre el valor máximo y mínimo del cilindro, esto es, entre 0 y la altura h .

Alternativamente, podemos realizar también la siguiente prueba, que es equivalente: el punto p es interior al cilindro, si la distancia euclídea entre el punto y el eje del cilindro es menor que el radio de la herramienta r y el punto se encuentra entre los dos planos que contienen cada base del cilindro. Dicho de otra manera, sea v' un vector con la misma dirección que v pero en sentido inverso, el punto p es interior al cilindro si el ángulo α que forman v' y el vector w que une p con a es menor de 90° , la distancia perpendicular de p al eje es menor que el radio y la distancia sobre el eje desde p al punto de aplicación a es menor que la altura h . Por lo tanto, la prueba de interioridad para herramientas cilíndricas es:

$$\begin{aligned} p \notin O \Leftrightarrow p \in h_{cil} &\Leftrightarrow \left(\alpha < \frac{\pi}{2} \right) \wedge (dist_{perpendicular}(p, eje) < r) \wedge (dist_{sobreje}(p, a) < h) \\ &\Leftrightarrow \left(0 < \cos \alpha < \frac{h}{|p-a|} \right) \wedge (|p-a| \cdot \text{sen} \alpha < r) \quad \text{Ecuación 8.12} \end{aligned}$$

En la figura 8.9 se explica gráficamente el significado de la prueba. El ángulo α se calcula de manera sencilla utilizando el producto escalar de v' y w .

$$\cos \alpha = \frac{v' \cdot w}{|v'| \cdot |w|} \quad \text{Ecuación 8.13}$$

En cuanto al cálculo del punto de intersección, sólo se realizará para aristas que tienen un vértice interior y otro exterior a la herramienta, por lo que podemos asegurar que existe un único punto de intersección. Sea la arista $p-q$. Obtenemos la arista transformada a coordenadas del cilindro $p'-q'$ aplicando a los puntos p y q la matriz de transformación T definida anteriormente. Para calcular la intersección, hemos de sustituir la ecuación paramétrica de la arista (ecuación 8.5) en la ecuación del cilindro (ecuación 8.10) obteniendo una ecuación cuadrática cuyas soluciones son los puntos de intersección de la recta definida por la arista con el cilindro infinito. Para calcular el punto de corte final habrá que tener en cuenta que el cilindro es finito y la arista también, y desecharemos aquellas soluciones que se salgan del rango de las dos figuras, es decir:

- De las dos soluciones, elegiremos aquella cuyo valor para el parámetro t está en el intervalo $[0,1]$, es decir, la que se encuentra en el segmento de recta que forma la arista.

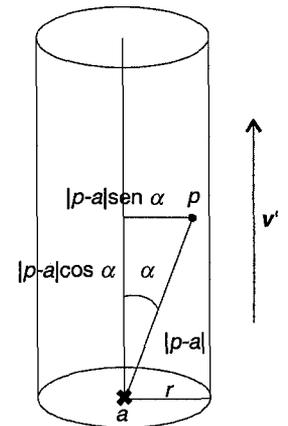


Figura 8.9
Prueba de interioridad para una herramienta cilíndrica

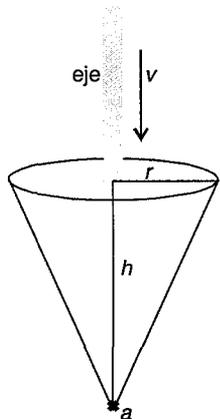


Figura 8.10
Parámetros de una herramienta cónica

- Si la coordenada z del punto está en el intervalo $[0, h]$ esa es la solución correcta ya que nos aseguramos de que está en la parte del cilindro infinito acotada por las dos bases.
- Si la coordenada z está fuera del intervalo o t no pertenece a $[0, 1]$ hemos de calcular la intersección con las bases. Si z es inferior a 0, la intersección debe realizarse con el plano $z=0$ y si es superior a h , con el plano $z=h$.

8.3.3.3 Herramientas cónicas

Una herramienta cónica viene definida por la tupla $h_{con}=(r,a,v,h)$, siendo r el radio, a el punto de aplicación, v la dirección de incidencia y h la altura del cono. Está compuesta por un cono cuyo vértice coincide con el punto de aplicación a y que tiene altura h y cuyo eje tiene la dirección v . Como en el caso de las herramientas esféricas, el eje que une al cono con la máquina no se considera a efectos de cálculo de puntos interiores/exteriores. La figura 8.10 representa una herramienta cónica.

En el caso de las herramientas cónicas, un punto p es interior a ésta si lo es al cono que la forma.

$$p \notin O \Leftrightarrow p \in h_{con} \Leftrightarrow p \in \text{Cono} \tag{Ecuación 8.14}$$

La ecuación implícita de un cono doble, infinito, con vértice situado en el origen de coordenadas y eje coincidente con el eje Z es:

$$x^2 + y^2 = \left(\frac{r}{h}z\right)^2 \tag{Ecuación 8.15}$$

donde r/h es la relación entre el radio y la altura del cono, si fuera finito. Como en el caso del cilindro, llamamos T a la matriz de transformación al espacio local del objeto, es decir, aquella que sitúa un cono genérico haciendo coincidir su eje con el eje Z y su vértice con el origen de coordenadas. Sea p' el resultado de aplicar la matriz T a un punto p , esto es, $p'=T \cdot p$. Ahora basta con sustituir p' en la ecuación 8.15 y comprobar que la altura del punto está entre 0 y la altura del cono h para determinar la interioridad de p :

$$p \notin O \Leftrightarrow p \in h_{con} \Leftrightarrow (p'=T \cdot p) \wedge \left(p'^2_x + p'^2_y < \left(\frac{r}{h}p'_z\right)^2 \right) \wedge (0 < p'_z < h) \tag{Ecuación 8.16}$$

También existe una prueba alternativa para determinar la interioridad del punto al cono: sea v' el vector de igual dirección que v pero con sentido inverso, p es interior al cono si el ángulo α que forman v' y el vector w que une p con a es menor que β y la distancia sobre el eje desde p al punto de aplicación a es menor que la altura h . β es el ángulo que forma el eje con el lado del cono, como podemos ver en la figura 8.11. La prueba queda:

$$p \notin O \Leftrightarrow p \in h_{con} \Leftrightarrow (\alpha < \beta) \wedge (dist_{sobre eje}(p,a) < h) \Leftrightarrow (\cos \alpha > \cos \beta) \wedge (|p-a| \cdot \cos \alpha < h) \tag{Ecuación 8.17}$$

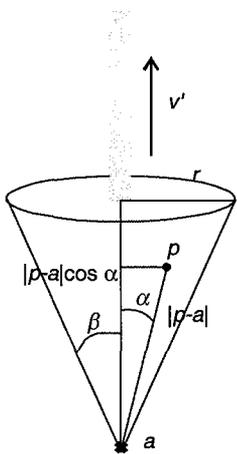


Figura 8.11
Prueba de interioridad para una herramienta cónica

El cálculo del punto de intersección entre la arista y el cono se resuelve de manera similar al caso del cilindro. Si la arista es $p-q$ obtenemos la arista transformada a coordenadas del cono $p'-q'$ aplicando T a los puntos extremos p y q . La intersección se calcula sustituyendo la ecuación 8.5 de la arista en la ecuación 8.15 del cono. La ecuación de segundo grado resultante tiene un máximo de dos soluciones, siendo válida aquella que se encuentre en el rango tanto de la recta ($0 \leq t \leq 1$) como del cono (el valor de z del punto de intersección debe estar entre 0 y la altura h). Si estas condiciones no se cumplen, entonces la intersección se encuentra sobre la base del cono, por lo que es necesario calcular el punto de corte de la arista con el plano $z=h$.

8.3.3.4 Herramientas tóricas

Las herramientas tóricas se definen como una tupla $h_{tor}=(r,r',a,v)$, y están formadas por un toro de radio principal r y radio del tubo r' , con punto de aplicación a (coincidente con el centro del toro) y orientada según la dirección de incidencia v . Supondremos que el eje que une a la herramienta con la máquina no interfiere en el proceso de visualización. La configuración de este tipo de herramientas se presenta en la figura 8.12.

Un punto p es interior a una herramienta tórica definida como en el párrafo anterior si lo es al toro que la forma.

$$p \notin O \Leftrightarrow p \in h_{tor} \Leftrightarrow p \in \text{Toro} \quad \text{Ecuación 8.18}$$

En este caso, para determinar si el punto es interior, vamos a utilizar la ecuación implícita de un toro. Un toro de radio principal r y radio el tubo r' , centrado en el origen de coordenadas y cuyo eje principal es el eje Z se puede definir mediante la siguiente ecuación:

$$\left(r - \sqrt{x^2 + y^2}\right)^2 + z^2 = r'^2 \quad \text{Ecuación 8.19}$$

Basta con sustituir el punto en la ecuación, de manera que p es interior a h_{tor} si:

$$p \notin O \Leftrightarrow p \in h_{tor} \Leftrightarrow \left(r - \sqrt{p_x^2 + p_y^2}\right)^2 + p_z^2 < r'^2 \quad \text{Ecuación 8.20}$$

Como ya hemos descrito para las herramientas anteriores, el cálculo del punto de intersección entre la arista y el cono se resuelve utilizando la ecuación de la figura. Sea una arista $p-q$ de manera que la arista, tras aplicar la transformación a coordenadas del toro T , se convierte en $p'-q'$. Debemos sustituir la ecuación de la arista (ecuación 8.5) en la ecuación 8.20 del toro. En este caso resulta una ecuación de cuarto grado con cuatro posibles soluciones. Será válida la solución cuyo valor para el parámetro t se encuentre en el intervalo $[0,1]$.

La solución a una ecuación de orden cuarto no es tan inmediata como en el caso de las cuadráticas. En este caso podemos realizar el procedimiento de dos maneras posibles:

- De forma analítica, transformando la ecuación en otra de orden 3 mediante un cambio de variable y resolviendo esta

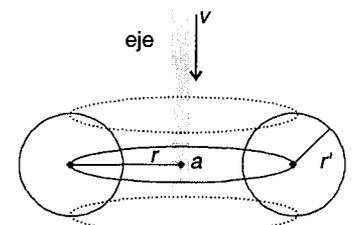


Figura 8.12
Parámetros de una herramienta tórica

última. Podemos encontrar el método en [Dodgson, 2000] y [Eberly, 2002].

- De forma aproximada, mediante un algoritmo de punto medio, hasta encontrar un intervalo suficientemente pequeño en el que se encuentre la solución.

8.3.4 Coherencia espacial

El concepto de coherencia espacial se refiere a la posibilidad de utilizar las restricciones espaciales de un problema para disminuir el coste del mismo y aprovechar operaciones ya realizadas en pasos sucesivos. Un algoritmo de *Marching Cubes* con las características que hemos descrito en este capítulo puede aprovechar esta coherencia en dos sentidos.

En primer lugar, los vóxeles en que dividimos el paralelepípedo inicial son todos adyacentes entre sí, de manera que comparten los vértices y las aristas. Así, a la hora de tratar un cubo, aquellos vértices y aristas que ya hayan sido tratados anteriormente no necesitarán ningún procesamiento adicional. Obsérvese que, de esta manera, el número de vértices y aristas procesados en cada caso son:

- Sólo en el primer cubo tratado se procesan 8 vértices y 12 aristas.
- Los cubos de los bordes del paralelepípedo supondrán procesar 4 vértices y 8 aristas para cada uno.
- En los cubos de los lados, sólo se tratarán 2 vértices y 5 aristas.
- En el caso general los cubos necesitan un procesamiento de 1 vértice y 3 aristas (figura 8.13).

En segundo lugar, tal y como ya se ha comentado previamente, a la hora de procesar los cubos sólo es necesario tratar aquellos que se encuentran en el área de influencia de la herramienta, área que vendrá marcada como un entorno de vecindad alrededor del punto de aplicación de radio el de la herramienta.

8.3.5 Incrementalidad del método

El método de poligonalización que hemos desarrollado es un procedimiento incremental, en el sentido de que cada paso del algoritmo da lugar a un modelo poligonal completo que es utilizado en el paso siguiente. Esta incrementalidad se da en dos aspectos:

- Tras cada paso discreto de herramienta conocemos la configuración de todos los vóxeles del objeto.
- Es posible en cada paso poligonalizar estos vóxeles para obtener el modelo poligonal.

Además, aprovechando la coherencia espacial, cada paso de ese proceso incremental supone modificar solamente los vóxeles y los polígonos situados en el entorno marcado por el radio de herramienta. Esta característica, además de reducir el coste del método, también posibilita dos procesos posteriores:

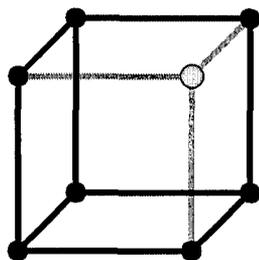


Figura 8.13
Aprovechamiento de la coherencia espacial en el caso general. En gris se muestran el vértice y las aristas que se han de procesar. Las demás ya han sido procesadas al tratar otros cubos (en negro)

- La simulación del mecanizado, es decir, podemos visualizar la progresión del modelo conforme va avanzando, lo que nos mostrará el procedimiento de mecanizado paso a paso. Este tipo de visualizaciones dinámicas son de gran ayuda para el usuario porque le permiten detectar fácilmente muchos de los posibles errores de diseño o cálculo de las trayectorias.
- La reutilización de la poligonalización, de manera que, una vez acabado el proceso, si añadimos nuevas trayectorias sobre el objeto no es necesario reiniciar el proceso desde el principio sino que podemos basarnos en la poligonalización previa.

La figura 8.14 muestra tres ejemplos de poligonalización obtenida mediante nuestro algoritmo.

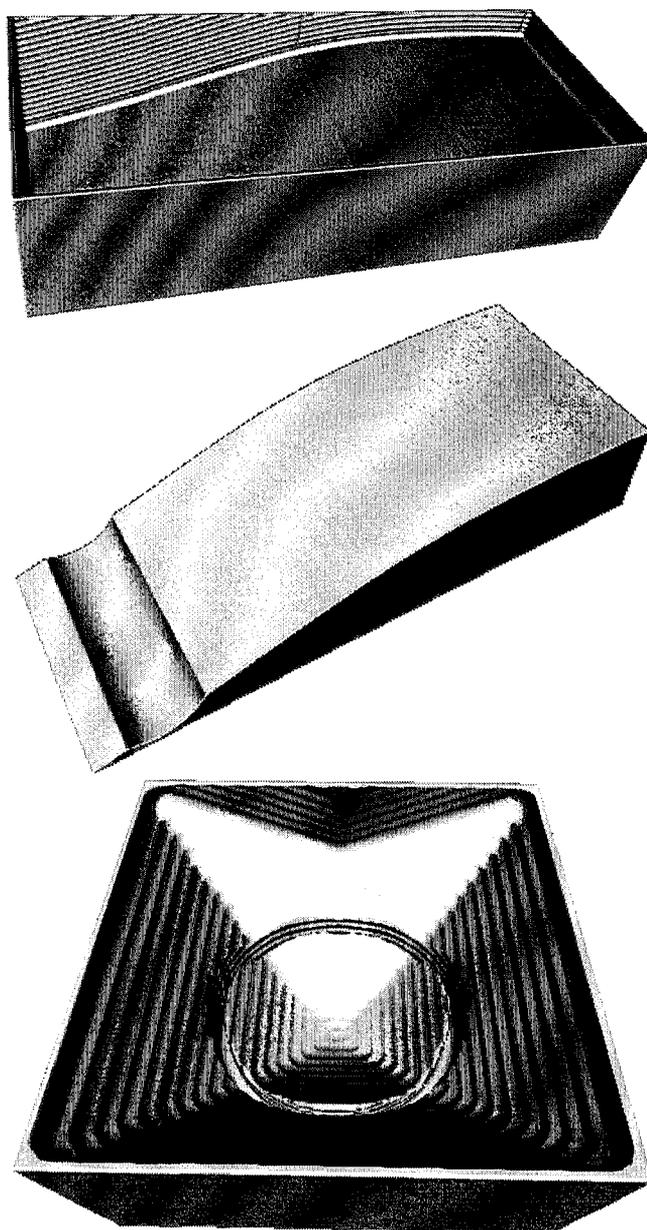


Figura 8.14
Ejemplos de poligonalización con el algoritmo modificado de *Marching Cubes*



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 9

EXPERIMENTOS Y RESULTADOS

9.1 INTRODUCCIÓN

La experimentación con el modelo propuesto nos va a permitir comprobar su adecuación a los objetivos que nos habíamos marcado al principio de esta tesis. Con este modelo pretendíamos conseguir dos metas principales:

- Facilitar el diseño de determinados objetos tridimensionales.
- Aumentar la calidad de mecanizado de dichos objetos.

Nuestra propuesta se basa en el uso de herramientas virtuales como forma intuitiva y sencilla de diseñar objetos y que permite a la vez conseguir un mecanizado de calidad. En este capítulo compararemos nuestro modelo con aquellos de uso más frecuente.

9.2 DISEÑO DE LOS EXPERIMENTOS

El diseño de los experimentos en este contexto es una cuestión realmente compleja, en la que es necesario determinar:

- Un conjunto de métodos a comparar con el propio, que sean representativos y estén disponibles.
- Un conjunto de parámetros a medir que nos den una idea adecuada de la bondad del método.
- Un banco de pruebas representativo de la casuística que pueda darse.

Esta labor se complica aún más en un caso como el nuestro, en el que se dan las siguientes circunstancias:

- Los modelos a comparar son muy heterogéneos y también, por lo tanto, los parámetros que es posible medir.
- Las características a comparar son más cualitativas que cuantitativas.

No obstante, se han tomado las decisiones que se justifican en los apartados siguientes.

9.2.1 Elección de los modelos más representativos

Quizás es ésta una de las pocas cuestiones en las que la elección no deja lugar a dudas: una inmensa mayoría de los sistemas CAD/CAM actuales utilizan el modelo de Geometría Sólida Constructiva (CSG), el modelo superficial (BRep) de superficies libres o modelos híbridos CSG-BRep. Una descripción de estos modelos y de sus características puede consultarse en el capítulo 3 "Estado del Arte" y en [Hearn, 1995], [Foley, 1990], [Bowyer, 1996], [Farin, 1993] y [Faux, 1979]

Se han analizado los productos comerciales CAD/CAM de uso más extendido: Catia, Unigraphics, SolidEdge, Parasolids y SolidWorks. Todos ellos utilizan uno de estos dos modelos, y en la

mayoría de los casos, permiten definir objetos híbridos generados utilizando ambos modelos.

Por lo tanto haremos una comparación de nuestro modelo con estos dos.

9.2.2 Parámetros de medida

La elección de los parámetros de medida es mucho más complicada debido a la heterogeneidad de los modelos y al carácter cualitativo de las características que queremos medir. Dado que lo que pretendemos medir es la facilidad de diseño y la mejora en la calidad del mecanizado, proponemos dos grupos de parámetros: parámetros cualitativos y parámetros cuantitativos.

Utilizaremos los parámetros cualitativos para comparar la facilidad de diseño, relacionándolos con las operaciones de diseño más habituales y compararemos qué dificultad tienen para los tres modelos (CSG, BRep y Superficie-Trayectoria). Evidentemente la valoración de estos parámetros tiene una componente de subjetividad inevitable.

Los parámetros cuantitativos permiten una visión más objetiva. Nos serán útiles para determinar la calidad del mecanizado y se referirán, especialmente, al número de operaciones que es necesario definir y al error cometido durante el mecanizado.

9.2.3 Banco de pruebas

Otro de los problemas fundamentales al diseñar los experimentos en un sistema como el nuestro es la elección del banco de pruebas. Cada modelo se adapta mejor al diseño y mecanizado de determinados objetos. Por ello, nos hemos decantado por generar dos tipos de objetos: los que son en teoría más adecuados para ser diseñados con el modelo CSG y los que se adaptan mejor a las características del modelo BRep, con el objetivo de comprobar cómo estos dos modelos y el S-T tratan cada uno de los casos. Estas pruebas se utilizarán para medir los parámetros cuantitativos propuestos.

9.3 PARÁMETROS CUALITATIVOS

Determinar la dificultad de uso de un sistema informático es una tarea eminentemente subjetiva. Cada usuario tiene una visión distinta de lo que le resulta fácil o difícil de realizar. No obstante, hemos tratado de identificar los aspectos que determinan de manera más habitual la facilidad de diseño con un determinado sistema CAD/CAM y la relación de estos aspectos con el modelo geométrico subyacente. Hemos llegado a la conclusión de que cuatro de las cuestiones fundamentales son:

- Parámetro 1: Facilidades que el modelo ofrece para describir un interfaz de usuario de uso sencillo.
- Parámetro 2: Adecuación para describir objetos formados por superficies complejas.

- Parámetro 3: Facilidad para realizar operaciones booleanas entre objetos.
- Parámetro 4: Adecuación para la visualización realista del objeto diseñado.

A continuación tratamos estos aspectos para cada uno de los modelos objeto de estudio.

9.3.1 Parámetro 1: Facilidades para definir el interfaz de usuario

La facilidad de uso de un interfaz de usuario depende fundamentalmente de la habilidad de su diseñador y de su constructor para determinar qué acciones resultan más intuitivas para un usuario tipo. No obstante, en un caso como el que nos ocupa, el modelo geométrico que constituye el motor del sistema y la información que necesita el modelo para ser descrito pueden condicionar de manera importante la facilidad de uso.

Un buen interfaz de usuario debe ser consistente, adaptado a diferentes niveles de destreza de los usuarios, retroalimentado, intuitivo, preciso, fácil de implementar e independiente de los periféricos disponibles.

Quizás uno de los aspectos que más dificulta el diseño de un interfaz gráfico de usuario intuitivo y de fácil manejo es la necesidad de introducir directamente datos tridimensionales. Esto se deriva del hecho evidente de que los periféricos, tanto los de entrada (ratón, tableta gráfica, ...) como los de salida (pantalla, impresora, plotter, ...), trabajan sobre un espacio bidimensional. La tercera dimensión debe añadirse utilizando algún tipo de artificio: vistas múltiples, introducción directa del valor, etc. Por lo tanto, cuantos menos datos tridimensionales sea necesario definir, tanto más sencillo será el interfaz de manejar y de implementar.

9.3.1.1 Modelo CSG

Este modelo se basa en la combinación de primitivas que se combinan mediante operaciones booleanas. Dado que las primitivas ya se encuentran descritas, es el modelo que menos datos tridimensionales directos necesita: para cada primitiva incorporada al objeto, sólo será necesario introducir sus dimensiones, la posición que ocupa y la orientación. Por otro lado, las operaciones entre objetos primitivos no necesitan información tridimensional adicional. El modelo CSG, por lo tanto, facilita la construcción del interfaz.

9.3.1.2 Modelo BRep

La inexistencia de primitivas y la utilización de las superficies libres que son la base del modelo, condicionan de manera importante el interfaz. En este caso es necesario introducir datos tridimensionales de manera masiva, en forma de puntos de control de las superficies que forman los objetos. Además, un sistema versátil debe permitir que esos puntos puedan ser modificados interactivamente de forma

sencilla. Ello hace que sea necesario ser muy cuidadoso a la hora de construir el interfaz para facilitar la tarea. Suelen utilizarse técnicas que facilitan la introducción y modificación de los puntos:

- Si se dispone de un modelo real del objeto, puede ser de gran utilidad los digitalizadores tridimensionales que permiten introducir los datos geométricos de un objeto tridimensional de forma automática.
- El sistema debe incorporar herramientas que permitan obtener formas tridimensionales a partir de objetos bidimensionales mucho más sencillos de definir: construcción de superficies por barrido, por rotación, etc. También suelen diseñarse interfaces con varias vistas en dos dimensiones, cada una de las cuales permite modificar dos de las tres coordenadas de cada punto.

9.3.1.3 Modelo S-T

El modelo S-T basa su forma de diseño en la construcción de superficies libres. Por lo tanto, comparte la misma problemática de generación del interfaz con el modelo BRep. No obstante, en muchos casos el uso de trayectorias y herramientas virtuales reduce considerablemente el número de superficies libres que se han de definir para diseñar un objeto, por lo que también se reducen las operaciones que el usuario debe realizar con el interfaz.

Por otro lado, los constructores de superficies descritos en el capítulo 5 facilitan de manera importante la definición de puntos tridimensionales al basarse en formas bidimensionales a partir de las cuales se construyen los objetos en tres dimensiones.

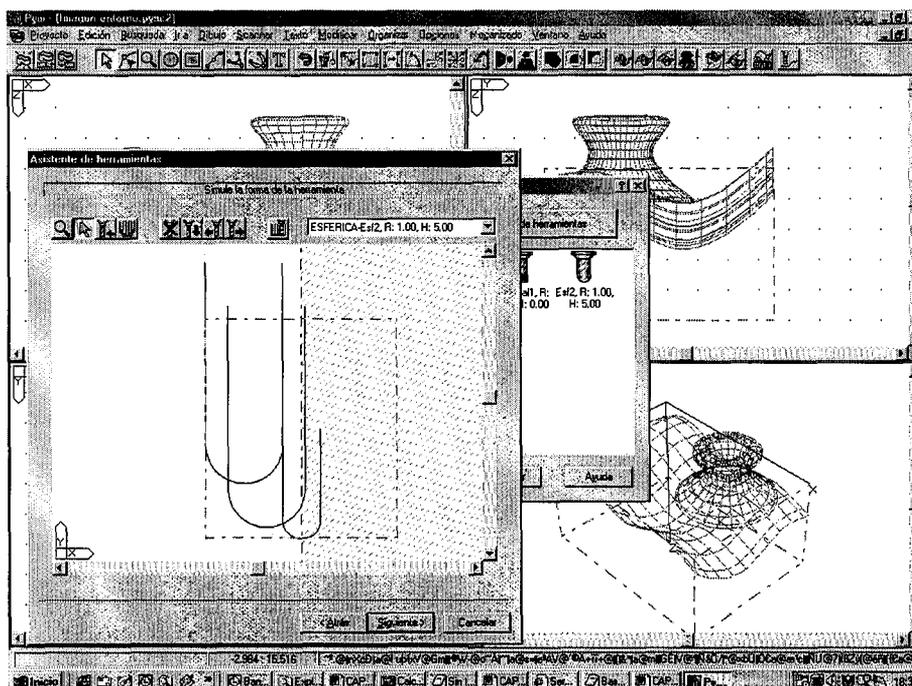


Figura 9.1
Interfaz de usuario de la aplicación diseñada basada en el modelo S-T

9.3.2 Parámetro 2: Adecuación para describir superficies libres

Las superficies libres se utilizan para diseñar formas curvas complejas. Prácticamente la totalidad de los sistemas CAD/CAM actuales permiten la posibilidad de definir curvas y superficies mediante alguno de los modelos Spline existentes. Como ya se ha comentado anteriormente, las curvas y superficies de este tipo se caracterizan por su versatilidad, su fácil diseño para el usuario y las buenas características geométricas (modificación local, estabilidad numérica, etc).

Por esta razón, el modelo debe incorporar la posibilidad de describir esta clase de objetos geométricos. Algunos modelos se basan precisamente en ellos; otros no los tienen en cuenta y los toman prestados de modelos superficiales.

9.3.2.1 Modelo CSG

Un modelo CSG puro se basa en un conjunto de primitivas geométricas estándar entre las que no caben las superficies libres, por lo que presenta grandes dificultades para diseñar objetos que contengan superficies complejas. Esto hace que la mayoría de los sistemas que utilizan este modelo incorporen también la posibilidad de crear superficies libres. Por esta razón, los modelos CSG aparecen casi siempre complementados con modelos BRep, formando modelos híbridos.

Sin embargo, las superficies paramétricas no se adaptan bien a la filosofía de trabajo de este modelo sólido, por ser muy difícil la implementación de las operaciones booleanas sobre formas libres. Por esa razón los sistemas basados en un modelo CSG suelen utilizar el modelo BRep como herramienta de diseño sobre la que construir posteriormente un objeto secundario con características de sólido, generalmente por poligonalización de la superficie libre.

9.3.2.2 Modelo BRep

El modelo BRep se basa, precisamente, en el diseño a partir de elementos superficiales en general, y de superficies libres paramétricas en particular. Esto hace que esté especialmente indicado para construir objetos con superficies complejas, sobre todo si son suaves. Sin embargo, cuando las superficies son planas, contienen aristas o son cuádricas, las superficies libres no son adecuadas. Esto hace que el modelo se complemente muy bien con el CSG, más apropiado para representar formas de estos últimos tipos.

9.3.2.3 Modelo S-T

El modelo S-T comparte con el BRep la utilización de superficies libres como elemento básico de diseño. Sin embargo, incorpora también el uso de trayectorias para herramientas virtuales. Esto hace que se adecue perfectamente a la definición de superficies complejas de tipo paramétrico sin renunciar a la posibilidad de definir

superficies de otro tipo, con aristas y con formas que son difíciles de obtener con superficies libres.

9.3.3 Parámetro 3: Facilidad para realizar operaciones booleanas

Los sistemas gráficos modernos implementan en su mayoría operaciones de combinación de formas geométricas sencillas para obtener objetos más complejos. Estas operaciones son de tipo booleano: unión, intersección y diferencia. Este mecanismo de construcción resulta muy versátil y permite generar objetos complejos de una forma sencilla e intuitiva.

Los sistemas que se basan en estas operaciones como forma de construcción las incorporan de manera natural. En otros casos, las operaciones supondrán un alto coste computacional.

9.3.3.1 Modelo CSG

Las operaciones booleanas de unión, intersección y diferencia entre objetos geométricos son precisamente los métodos básicos de construcción en el modelo CSG. Estas operaciones hacen de este modelo uno de los más extendidos entre los sistemas CAD/CAM.

9.3.3.2 Modelo BRep

En principio este tipo de operaciones no forman parte del modelo BRep puro, por lo que no resulta posible realizarlas. En todo caso, pueden simularse con grandes dificultades o utilizar un modelo secundario de tipo poligonal que permita implementarlas. Hemos de tener en cuenta que las operaciones booleanas entre volúmenes suponen el cálculo de intersecciones entre ellos, y este es un proceso de alta complejidad cuando entran en juego superficies libres.

Esta es otra de las razones por las que el modelo BRep suele ir acompañado de métodos propios del modelo CSG.

9.3.3.3 Modelo S-T

Comparte con el modelo BRep el uso de superficies libres, por lo que algunas limitaciones también le son aplicables. Sin embargo, el uso de trayectorias como elemento de diseño del modelo añade una nueva operación que puede verse como una sustracción booleana de la herramienta virtual al modelo superficial inicial. Esto hace que se encuentre incluida la operación de diferencia de forma implícita.

En este punto debemos resaltar una diferencia fundamental con el modelo BRep: en éste, los objetos se encuentran definidos por su frontera, de manera que para definir un objeto formado por un único volumen, debemos obtener una única frontera que englobe al volumen. Esto implica la necesidad de implementar métodos que permitan unir y cortar superficies complejas. En el modelo S-T, sin embargo, el objeto se define como el resultado de restarle a un volumen inicial las herramientas virtuales que pasan por las trayectorias diseñadas. Esto hace que no sea necesario cortar o unir

superficies puesto que se utilizan sólo como soporte para definir posteriormente las trayectorias. De esta manera, la obtención de objetos mediante operaciones de tipo booleano es mucho más sencilla, al no tener que intersectar las superficies libres.

9.3.4 Parámetro 4: Adecuación para la visualización realista

La visualización realista del modelo es otro de los aspectos que resultan determinantes en un sistema moderno. Cada vez más los usuarios demandan un aspecto visual más atractivo y cercano a la realidad de los objetos diseñados.

De nuevo el modelo geométrico subyacente puede determinar la facilidad y calidad de la visualización.

9.3.4.1 Modelo CSG

El modelo CSG se basa en aspectos volumétricos, mientras que la visualización de un objeto depende de su superficie. Por ello debemos recurrir a técnicas que nos permitan obtener la superficie del modelo a partir del volumen. Las dos posibilidades más utilizadas son:

- Visualización directa del modelo: debemos recurrir a técnicas avanzadas de visualización como puede ser el trazado de rayos. Estos métodos trabajan directamente con el volumen, alcanzan una alta calidad y presentan un modelo exacto del objeto. A cambio, tienen un alto coste computacional y no permiten, en general, una visualización interactiva.
- Visualización mediante un modelo secundario: estos métodos se basan en una poligonalización del objeto para visualizarlo mediante las técnicas clásicas de sombreado de polígonos. Una vez obtenido el modelo poligonal el proceso de visualización es muy rápido. En contrapartida, las técnicas de obtención de este modelo secundario son costosas, además de ofrecer un modelo aproximado y no exacto.

9.3.4.2 Modelo BRep

Los modelos de superficies tienen la ventaja, a la hora de la visualización de no tener que calcular su frontera. Esto hace que sea fácil poligonalizar los objetos y visualizarlos utilizando las técnicas conocidas de sombreado de polígonos. El proceso resulta rápido y la visualización tiene suficiente calidad, aunque se trate de una aproximación de superficies curvas mediante polígonos planos.

9.3.4.3 Modelo S-T

La visualización en nuestro modelo comparte los mismos problemas que el modelo CSG. No disponemos de la frontera del objeto (hemos de tener en cuenta que las superficies que se diseñan sirven de base para definir las trayectorias pero no tienen por qué formar parte del objeto final) con lo que es necesario visualizar el

volumen directamente u obtener un modelo secundario poligonal. En el capítulo 8 se han presentado las técnicas propuestas, basadas en la obtención del modelo poligonal mediante una variación del algoritmo *Marching Cubes*.

En la tabla 9.1 se muestra una comparativa de las características cualitativas de los modelos analizados. Se cuantifica la adecuación de cada modelo al aspecto considerado atendiendo a la "facilidad" que presenta el modelo según el estudio realizado. Se manejan cuatro grados de adecuación: Ninguna, Alguna, Buena y Excelente. Como vemos, el modelo S-T resulta bastante equilibrado en todos los aspectos.

	Parámetro 1: Interfaz de usuario	Parámetro 2: Superficies complejas	Parámetro 3: Operaciones booleanas	Parámetro 4: Visualización realista
CSG	Excelente	Ninguna	Excelente	Alguna
BRep	Alguna	Excelente	Ninguna	Buena
S-T	Buena	Excelente	Buena	Alguna

Tabla 9.1
Adecuación de los distintos modelos geométricos a las características cualitativas estudiadas

9.4 PARÁMETROS CUANTITATIVOS

Los parámetros cuantitativos permiten una visión más objetiva. Nos serán útiles para determinar la calidad del mecanizado y se referirán, especialmente, al número de operaciones de diferentes tipos que es necesario realizar y al error cometido durante el mecanizado. Para ello, obtendremos una cota del error producido al mecanizar el objeto con cada modelo. Ese error será la diferencia entre el volumen del objeto real y el volumen del objeto mecanizado. También compararemos la forma en que es posible generar las trayectorias para cada modelo y qué consecuencias tiene esto en la calidad final de la pieza.

Para poder evaluar el comportamiento de los diferentes modelos en la generación de trayectorias y en el mecanizado de la pieza, vamos a fijar un conjunto de parámetros con el que se ha realizado el banco de pruebas:

- El mecanizado se realiza con una máquina de 3 ejes.
- Se utilizará una herramienta esférica, por ser el tipo de herramienta más común y versátil.

Para cada caso, calcularemos el error que se comete al mecanizar la pieza con respecto al volumen del objeto original.

9.4.1 Cálculo del error al mecanizar superficies

Presentamos aquí las expresiones para calcular el error al fabricar superficies planas y esféricas con herramientas esféricas. Estas ecuaciones se utilizarán varias veces a lo largo del capítulo. Otras expresiones más específicas se deducen en cada apartado concreto.

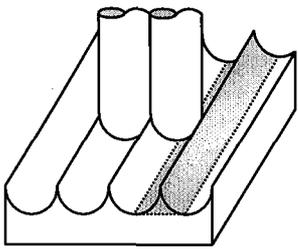


Figura 9.2
Efecto de montes y valles. El error es el volumen de cada monte

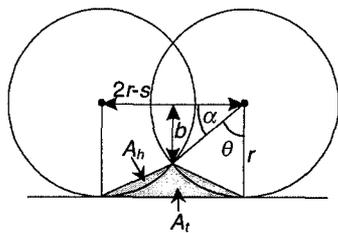


Figura 9.3
Cálculo del área de la sección del monte

9.4.1.1 Error al mecanizar superficies planas

Las superficies planas son unas de las que más corrientemente nos podemos encontrar. Al mecanizar una superficie de este tipo con una herramienta esférica se produce el efecto de montes y valles que se mostraba en la sección 3.2.5 del capítulo 3. El error que se comete puede cuantificarse hallando el volumen del monte (figura 9.2). Para ello, calculamos como paso previo el área de la sección del monte.

Sea una superficie plana a mecanizar con un herramienta esférica de radio r y con un solape entre pasadas de la herramienta s . El área A que queda sin mecanizar puede cuantificarse mediante la siguiente expresión (figura 9.3):

$$A = A_t - 2A_h \quad \text{Ecuación 9.1}$$

La áreas del triángulo A_t y del segmento de circunferencia de la herramienta A_h se obtienen con las ecuaciones:

$$A_t = \frac{(2r-s)(r-b)}{2} \quad \text{Ecuación 9.2}$$

$$A_h = \frac{1}{2}r^2(\theta - \text{sen}\theta) \quad \text{Ecuación 9.3}$$

donde el segmento b y el ángulo θ se calculan con las siguientes expresiones:

$$b = r \text{sen} \alpha \quad \text{Ecuación 9.4}$$

$$\theta = \frac{\pi}{2} - \alpha \quad \text{Ecuación 9.5}$$

$$\alpha = \arccos \frac{2r-s}{2r} \quad \text{Ecuación 9.6}$$

El error que se produce es el volumen generado por esta sección al desplazarse sobre la trayectoria. Dependiendo de las características de la trayectoria, el cálculo del volumen se hará de una manera o de otra.

9.4.1.2 Error al mecanizar superficies esféricas

Otro caso típico que nos vamos a encontrar es el error que se comete al mecanizar una superficie esférica, convexa o cóncava. En estos casos también se produce el efecto de montes y valles.

Superficies convexas

Sea ahora una superficie esférica convexa de radio R a mecanizar con un herramienta esférica de radio r y con un solape entre pasadas de la herramienta s . De manera similar a como ocurría en el caso anterior, el área A sin mecanizar se calcula con una expresión como la siguiente:

$$A = A_t - A_s - 2A_h \quad \text{Ecuación 9.7}$$

donde A_t , A_s y A_h son, respectivamente, las áreas del triángulo, de la sección de circunferencia de la superficie y de la sección de

circunferencia de la herramienta, tal y como se muestran en la figura 9.4. Las ecuaciones que permiten calcular estas áreas son:

$$A_t = \frac{a \cdot b}{2} \quad \text{Ecuación 9.8}$$

$$A_s = \frac{1}{2} R^2 (\alpha - \text{sen } \alpha) \quad \text{Ecuación 9.9}$$

$$A_h = \frac{1}{2} r^2 (\beta - \text{sen } \beta) \quad \text{Ecuación 9.10}$$

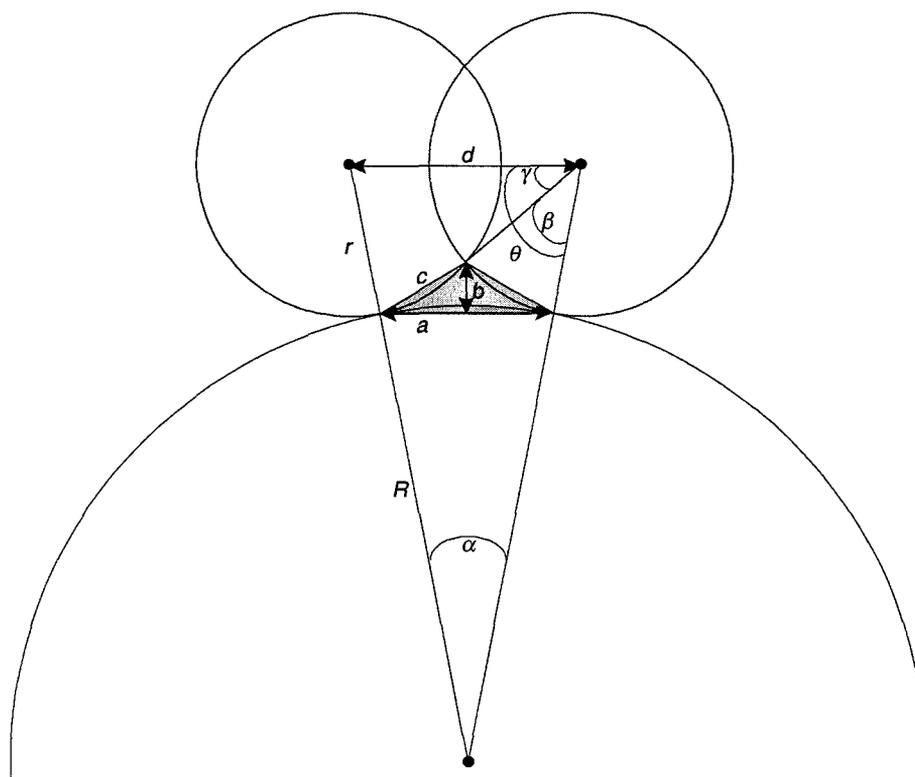


Figura 9.4
Cálculo del área del error al mecanizar una superficie esférica convexa

El ángulo α se calculará en función del caso específico que estemos analizando. Si suponemos α conocido, los segmentos a , b y c de la figura 9.4 se obtienen de la siguiente manera:

$$a = 2R \text{sen } \frac{1}{2} \alpha \quad \text{Ecuación 9.11}$$

$$b = \sqrt{c^2 - \left(\frac{a}{2}\right)^2} \quad \text{Ecuación 9.12}$$

$$c = 2r \text{sen } \frac{1}{2} \beta \quad \text{Ecuación 9.13}$$

Por otro lado, el ángulo β es:

$$\beta = \theta - \gamma = \frac{\pi - \alpha}{2} - \arccos \frac{d}{2r} \quad \text{Ecuación 9.14}$$

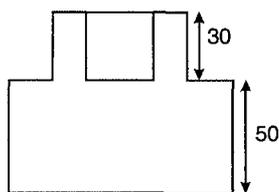
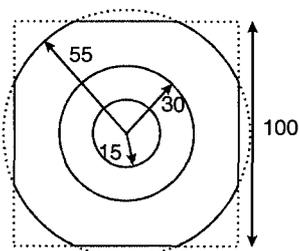
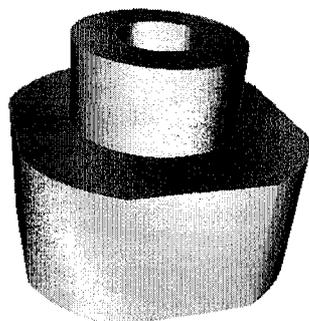
Por último, el segmento d se obtiene a partir del ángulo α con la siguiente ecuación:

$$d = 2(R+r)\text{sen}\frac{1}{2}\alpha \quad \text{Ecuación 9.15}$$

Superficies cóncavas

En este caso podemos utilizar las expresiones que obtuvimos en el apartado 7.3.3, del capítulo 7 cuando calculábamos el error al emular una herramienta esférica utilizando otra herramienta esférica de menor radio.

Una vez calculada el área sin mecanizar, el volumen que constituye el error se hallará de una manera o de otra dependiendo de la trayectoria concreta.



9.4.2 Ejemplo 1

El primer ejemplo que vamos a realizar se muestra en la figura 9.5. Se trata de un objeto sencillo cuyas características se adecuan especialmente al modelo CSG. Veremos todo el proceso de diseño y cálculo de las trayectorias para este objeto según cada uno de los modelos analizados.

Para el cálculo del error, hallaremos previamente el volumen del objeto utilizando la siguiente expresión:

$$V = V_a + V_b - V_c \quad \text{Ecuación 9.16}$$

donde V_a es el volumen de la parte inferior, V_b el del cilindro exterior y V_c el del cilindro interior.

V_a puede calcularse a partir del área de la sección de la pieza inferior, multiplicada por su altura. Tal y como se aprecia en la figura 9.6, el área de esta sección A_s se calcula con la siguiente expresión:

$$A_s = A_a + 2A_b + 4A_c \quad \text{Ecuación 9.17}$$

donde $A_a = 100 \cdot 45,82 = 4582 \text{ mm}^2$, $A_b = 27,09 \cdot 45,82 = 1241,2638 \text{ mm}^2$ y $A_c = 412,99675 \text{ mm}^2$, luego $A_s = 8716,5146 \text{ mm}^2$. Por lo tanto, el volumen de la pieza inferior es de $V_a = A_s \cdot h = 8716,5146 \cdot 50 = 435825,73 \text{ mm}^3$.

El volumen total del objeto será $V = V_a + V_b - V_c = 435825,73 + 84823 - 21205,75 = 499442,98 \text{ mm}^3$.

9.4.2.1 Diseño del objeto

Modelo CSG

El diseño de este objeto siguiendo el modelo CSG resulta muy sencillo utilizando primitivas básicas (cilindros y paralelepípedos) y las tres operaciones booleanas (unión, intersección y diferencia). El árbol de operaciones que da lugar al objeto se muestra en la figura 9.7.

Figura 9.5
Objeto del ejemplo 1

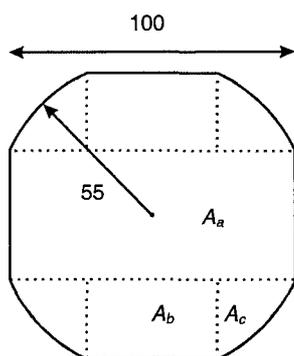


Figura 9.6
Cálculo del área de la sección

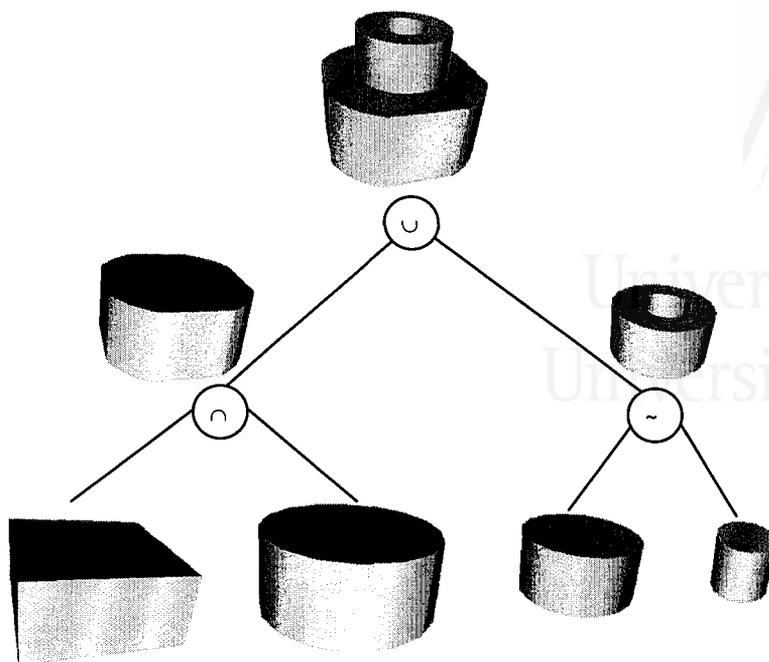


Figura 9.7
 Árbol de operaciones para generar el objeto del ejemplo 1 mediante un modelado CSG

Modelo BRep

El diseño del objeto propuesto con el modelo BRep se complica. Tratar de diseñar de manera directa las superficies que componen la figura resulta impracticable. Por ello debemos recurrir a herramientas que nos asistan en el diseño. En un caso como este la herramienta más útil puede ser la construcción de superficies por barrido. En este ejemplo concreto dos barridos, uno lineal y otro por rotación pueden dar lugar a dos objetos que unidos forman el objeto final (figura 9.8). Para que el objeto sea único (esté formado por un único volumen, con una única frontera) debemos contar con herramientas para unir las superficies que resultan de los barridos.

Modelo S-T

En el modelo S-T el diseño del objeto y la definición de las trayectorias se encuentran íntimamente ligados, ya que las trayectorias también forman parte del diseño del objeto.

Una forma sencilla de diseñar el objeto propuesto es trazar tres figuras planas:

- El contorno exterior de la pieza inferior, que podría obtenerse como intersección de un cuadrado de 100×100 y una circunferencia de radio 55.
- El contorno exterior del cilindro, que es una circunferencia de radio 30.
- El contorno interior del cilindro, que es otra circunferencia de radio 15.

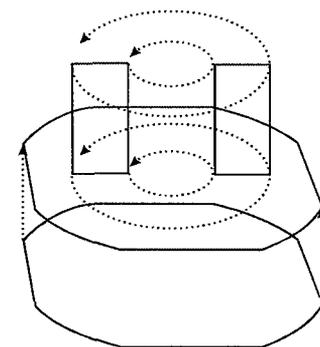


Figura 9.8
 Diseño del objeto por barrido con el modelo BRep

Como vemos los objetos que debemos diseñar son muy sencillos, además de planos, por lo que evitamos el problema de tener que indicar una tercera dimensión.

El diseño del objeto se completará cuando se definan las trayectorias de mecanizado.

9.4.2.2 Definición de las trayectorias

La obtención automática de las trayectorias de mecanizado se puede realizar según dos esquemas: trayectorias paralelas a uno de los ejes o en zig-zag, y trayectorias concéntricas o en espiral.

Para mecanizar esta figura utilizaremos trayectorias de mecanizado a altura constante, ya que los planos principales del objeto son normales al vector de incidencia de la herramienta. En el cálculo de trayectorias a altura constante, el sistema debe obtener, para cada plano de mecanizado, su intersección con el objeto y, a continuación, el mecanizado de ese plano, dejando las partes sólidas del objeto como islas. La obtención de la intersección del plano con el objeto será un tema crucial y su facilidad dependerá del modelo elegido. Como veremos, el modelo S-T no necesitará del cálculo de estas intersecciones.

En cuanto a los planos de mecanizado que debemos obtener, podemos distinguir tres casos:

- En los primeros planos, hasta alcanzar la parte superior del objeto, se realiza un mecanizado de todo el plano, sin islas.
- En los planos situados entre la parte superior del objeto y una profundidad de 30, se mecaniza el cilindro superior. En este caso, en cada plano tendremos dos islas circulares concéntricas que delimitan el cilindro por su parte exterior e interior.
- Los últimos planos a mecanizar, desde una profundidad de 30 hasta alcanzar la parte inferior del objeto, tienen una única isla cuya forma coincide con el contorno de la parte inferior del objeto.

En la figura 9.9 se muestran los planos que debemos mecanizar.

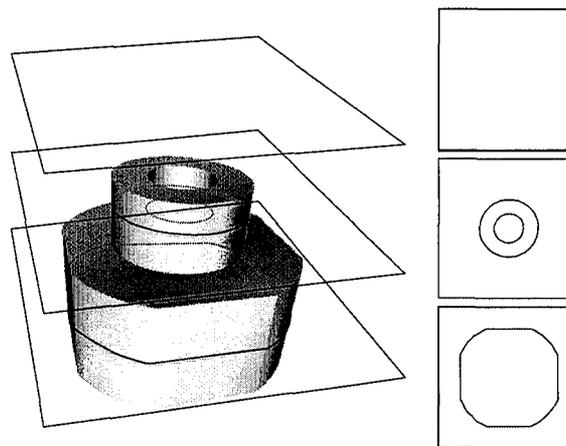


Figura 9.9

Grupos de planos para un mecanizado a altura constante

Para cada tipo de plano, debemos realizar el cálculo de la trayectoria según cada estrategia.

Para un mecanizado paralelo a un eje, las trayectorias obtenidas se muestran en la figura 9.10. Para un mecanizado concéntrico, se obtienen las trayectorias de la figura 9.11.

El cálculo automático de las trayectorias que se ha presentado penaliza en general la calidad del mecanizado. Sin embargo, cuando el usuario guía el proceso, como ocurre en el modelo S-T, se puede obtener un mecanizado más adecuado si el usuario es experto, a cambio de un aumento en el número de tareas que debe desarrollar.

Veamos ahora las particularidades que presenta la definición de las trayectorias para cada modelo.

Modelo CSG

A la hora de calcular las trayectorias que deben seguir las herramientas, una de las diferencias fundamentales entre los modelos es la obtención de las intersecciones entre los planos de mecanizado y el objeto. En el modelo CSG esta intersección puede resultar sencilla si tratamos con primitivas básicas (poliedros, cuádricas o supercuádricas, etc.) pero se complica con otro tipo de objetos con los que será necesario definir los métodos particulares de intersección.

Una vez determinados los planos y las curvas resultantes de la intersección de éstos con el objeto, podemos emplear para cada uno cualquiera de los dos esquemas de mecanizado propuestos (paralelo o concéntrico).

Modelo BRep

Para realizar el cálculo automático de las trayectorias de mecanizado podemos seguir las estrategias en zig-zag o en espiral ya señaladas. Igual que en aquel caso, debemos calcular para cada plano de mecanizado su intersección con el objeto para mecanizar luego el plano respetando las islas. En este caso la intersección de un plano con una superficie genérica no es sencilla. Sólo para ciertos casos especiales podrán utilizarse métodos analíticos; en la mayoría tendremos que emplear algoritmos basados en subdivisión de la superficie, métodos que resultan costosos y aproximados.

Modelo S-T

En el modelo S-T la definición de las trayectorias de mecanizado completa al proceso de diseño del objeto. Esto hace que esta definición sea guiada por el usuario, lo que, en la mayoría de los casos, redundará en una mejora de la calidad de la pieza final. Además, por la forma en que se definen las trayectorias, se evita el costoso cálculo de las intersecciones plano-objeto.

Es posible diseñar y fabricar de forma adecuada un objeto como el propuesto, definiendo las siguientes trayectorias de mecanizado (figura 9.12):

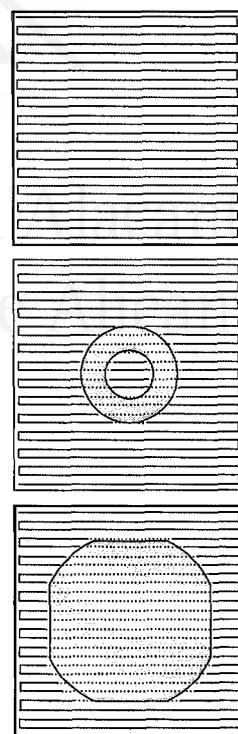


Figura 9.10
Mecanizado en paralelo. En trazos discontinuos se señalan las pasadas a altura de seguridad

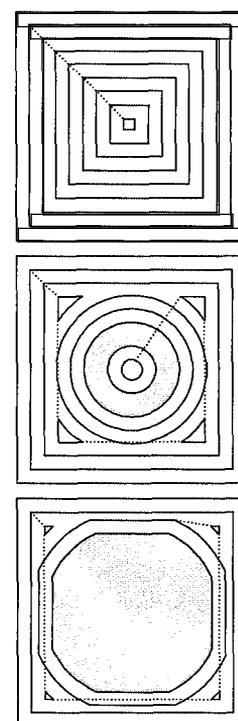


Figura 9.11
Mecanizado concéntrico. En trazos discontinuos se señalan las pasadas a altura de seguridad

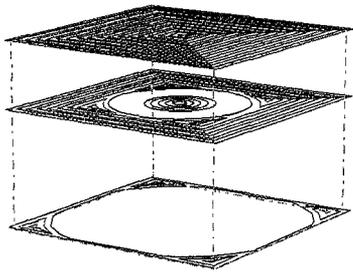


Figura 9.12
Trayectorias para el diseño y
el mecanizado según el
modelo S-T

- Trayectoria 1: Mecanizar el material hasta una altura de 80 mm, es decir, hasta la parte superior del objeto, mediante una trayectoria paralela o concéntrica.
- Trayectoria 2: Mecanizar la circunferencia central de radio 15 mm por el interior hasta una altura de 50 mm con una trayectoria de desbaste concéntrica.
- Trayectoria 3: Mecanizar todo el material hasta una altura de 50 mm con una trayectoria de desbaste concéntrica, tomando la circunferencia de 30 mm como isla.
- Trayectoria 4: Mecanizar todo el material hasta una altura de 0 mm con una trayectoria de desbaste concéntrica, tomando como isla a la sección de la parte inferior del objeto.

En este caso se generan cuatro trayectorias. Aunque el procedimiento requiere la intervención del usuario las trayectorias que debe definir son sencillas y, además, se producen las siguientes ventajas:

- El hecho de definir las trayectorias facilita y acorta la fase de diseño.
- La calidad del mecanizado es mayor pues permite al usuario guiar el proceso. El error cometido lo veremos en el siguiente apartado.

9.4.2.3 Cálculo del error

Tomando como base las ecuaciones obtenidas en el apartado 9.4.1, calculamos el error que se produce al mecanizar según cada una de las estrategias.

Obsérvese que en el error final de la pieza sólo interviene el mecanizado del último de los planos de cada tipo, es decir, cuando la herramienta se desplaza sobre la superficie del objeto. Por lo tanto, supondremos que todas las capas anteriores han sido desbastadas y tendremos en cuenta el error producido en la última.

Trayectorias paralelas a un eje

En el mecanizado paralelo, distinguimos dos tipos de errores: los producidos sobre las superficies mecanizadas directamente (volumen de los montes que se dejan, tal y como ya hemos comentado) y los errores en las paredes perpendiculares.

Los primeros errores pueden calcularse como el área del monte multiplicada por la longitud lineal de la trayectoria, puesto que es recta. Podemos simplificar el cálculo, considerando únicamente el área de la isla del tercero de los planos de la figura 9.9, equivalente a la suma de las áreas mecanizadas en cada uno de los planos. Para una herramienta de radio $r=2$ mm y un solapamiento entre pasadas de la herramienta de $s=1$ mm, la separación entre los puntos de aplicación de la herramienta entre pasadas sucesivas es de $2r-s=3$ mm. Puesto que la isla del segundo plano tiene unas dimensiones de 100×100 mm, tenemos un total de $100/3=33$ pasadas de herramienta. En la zona central, la longitud de cada pasada es de 100 mm. Sin embargo, en las

zonas superior e inferior la longitud de la pasada va disminuyendo conforme nos alejamos del centro de la pieza. Una buena aproximación para calcular la longitud de esas pasadas es calcular el porcentaje de cuadrado que ocupa el área de la zona y utilizar ese porcentaje como aproximación para reducir la longitud de las pasadas.

Tal y como se obtuvo al principio del apartado 9.4.2 de este capítulo, el área de la sección de la pieza inferior del objeto es $A_3=8716,5146 \text{ mm}^2$, lo que supone un 87,17% del área del cuadrado. Si la longitud total de la trayectoria es de $33 \cdot 100=3300 \text{ mm}$, la longitud de la trayectoria sobre la sección es de $l=2876,61 \text{ mm}$.

Siguiendo las ecuaciones 9.1 a 9.6, el área de la sección del monte que se produce entre pasadas es $A=0,6234 \text{ mm}^2$. El error cometido será, por lo tanto de $e_1=A \cdot l= 1793,2787 \text{ mm}^3$.

En cuanto al error producido sobre las paredes perpendiculares, debemos tener en cuenta tres casos: error al mecanizar el cilindro superior por el interior, error al mecanizarlo por el exterior y error al mecanizar la parte inferior del objeto por el exterior.

Al mecanizar el cilindro por el exterior se produce un error sobre la pared vertical, como consecuencia de un conjunto de montes longitudinales, iguales a los que se producen al mecanizar una superficie esférica convexa. Por lo tanto podemos utilizar las ecuaciones 9.7 a 9.15 de la sección 9.4.1.2 para estimar este error. Para ello debemos conocer cuál es el ángulo α al que se hace referencia en estas expresiones, y que se corresponde con los ángulos α_i que pueden observarse en la figura 9.13. Estos ángulos pueden obtenerse con las ecuaciones siguientes:

$$\alpha_1 = \arcsen \frac{2r-s}{R+r} \quad \text{Ecuación 9.18}$$

$$\alpha_i = \arcsen \frac{i(2r-s)}{R+r} - \sum_{j=1}^{i-1} \alpha_{j-1} \quad \text{Ecuación 9.19}$$

En el caso que nos ocupa, el cilindro tiene radio $R=30 \text{ mm}$, con lo que el número de pasadas será de $60/3=20$. Puesto que es simétrico con respecto al diámetro horizontal, calcularemos los 10 ángulos superiores: $\alpha_1=0,09389$, $\alpha_2=0,09473$, $\alpha_3=0,09648$, $\alpha_4=0,09930$, $\alpha_5=0,10348$, $\alpha_6=0,10953$, $\alpha_7=0,11843$, $\alpha_8=0,13222$, $\alpha_9=0,15617$ y $\alpha_{10}=0,21115$. Los ángulos α_8 , α_9 y α_{10} sobrepasan el ángulo crítico a partir del cual no se produce la intersección entre las fronteras de las herramientas. Tomaremos como valor aproximado para ellos el ángulo crítico: $\alpha_8=\alpha_9=\alpha_{10}=0,12508$. Para cada uno de estos ángulos hallamos el área de la sección del monte siguiendo las ecuaciones 9.7 a 9.15, obteniendo los siguientes valores: $A_1=0,55742$, $A_2=0,57394$, $A_3=0,6097$, $A_4=0,67109$, $A_5=0,7713$, $A_6=0,93858$, $A_7=1,2463$, $A_8=1,55516$, $A_9=1,55516$ y $A_{10}=1,55516$. Para obtener el volumen total de error debemos multiplicar cada área por la longitud del valle (que es la altura del cilindro), sumarlos todos y multiplicarlos por cuatro. Este error es $e_2= 1200,4572$.

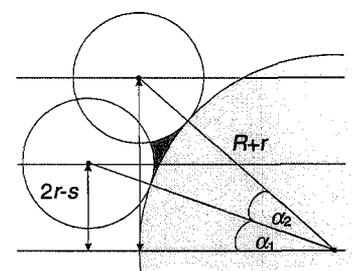


Figura 9.13
Vista superior del
mecanizado del cilindro por
el exterior

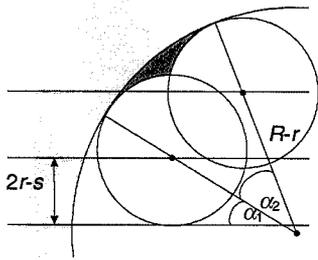


Figura 9.14
Vista superior del
mecanizado del cilindro por
el interior

Cuando se calculan las trayectorias para mecanizar el cilindro por el interior las ecuaciones adecuadas son las del cálculo del error al mecanizar una superficie esférica cóncava. De forma análoga a como hemos procedido en el caso anterior, el ángulo α de las ecuaciones 7.4 a 7.13 del capítulo 7, se corresponde con los ángulos α_i de la figura 9.14. Las expresiones que nos permiten calcular esos ángulos son:

$$\alpha_1 = \arcsen \frac{2r-s}{R-r} \quad \text{Ecuación 9.20}$$

$$\alpha_i = \arcsen \frac{i(2r-s)}{R-r} - \sum_{j=1}^{i-1} \alpha_{j-1} \quad \text{Ecuación 9.21}$$

En este caso el radio del cilindro es $R=15$, con lo que el número de pasadas es $30/3=10$, aunque al ser simétrico sólo debemos calcular cinco ángulos: $\alpha_1=0,23287$, $\alpha_2=0,24686$, $\alpha_3=0,28495$, $\alpha_4=0,41132$ y $\alpha_5=\pi/2$. Los ángulos α_4 y α_5 sobrepasan el ángulo crítico a partir del cual no se produce la intersección entre las fronteras de las herramientas. Tomaremos como valor aproximado para ellos el ángulo crítico: $\alpha_4=\alpha_5=0,30892$. Las áreas de las secciones de los montes que se producen se calculan a partir de estos ángulos y son $A_1=0,82848$, $A_2=0,99915$, $A_3=1,60943$, $A_4=2,161912$ y $A_5=2,161912$. Por último, el volumen de estos montes será el producto de estas áreas por su longitud. Tras el cálculo de cada uno, su suma y su multiplicación por cuatro (pues al ser simétrico sólo hemos calculado la mitad y, además son dos caras) obtenemos el siguiente volumen de error $e_3=931,30608 \text{ mm}^3$.

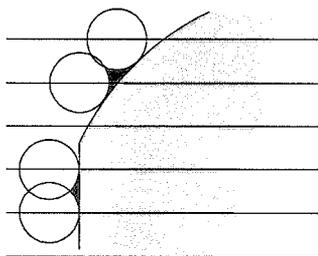


Figura 9.15
Vista superior del
mecanizado de parte de la
pieza inferior

En el error al mecanizar la parte inferior del objeto por el exterior debemos distinguir tres casos especiales (figura 9.15):

- En el mecanizado de las paredes paralelas a la dirección de la trayectoria no se produce error.
- En el mecanizado de las paredes perpendiculares a la dirección de la trayectoria el error viene dado por las mismas ecuaciones que en el mecanizado de superficies planas.
- En el mecanizado de las secciones cilíndricas se produce un error como el del mecanizado del cilindro por el exterior.

Tratemos, por lo tanto estos dos últimos casos: al mecanizar la pared plana perpendicular se producen un total de $45,82/3=15$ montes a cada uno de los dos lados, cada uno de ellos con una longitud de 50 mm (la altura de la pieza) y una sección de área $A=0,6234 \text{ mm}^2$, y por lo tanto el error es de $e_4=935,1 \text{ mm}^3$.

El último caso es el error producido en cada una de las cuatro secciones cilíndricas. Como hemos hecho en el caso del mecanizado del cilindro exterior, debemos calcular los ángulos α_i , que en este caso son 9: $\alpha_1=0,04218$, $\alpha_2=0,06004$, $\alpha_3=0,06225$, $\alpha_4=0,06499$, $\alpha_5=0,06845$, $\alpha_6=0,07291$, $\alpha_7=0,07887$, $\alpha_8=0,08725$ y $\alpha_9=0,10006$. Los cuatro últimos ángulos superan el ángulo crítico, por lo que tomamos su valor que es de 0,07019 como una aproximación. El error que se producen este caso es $e_5=2277,908 \text{ mm}^3$.

En total se produce un error para el objeto de $e=e_1+e_2+e_3+e_4+e_5=7138,04998 \text{ mm}^3$, que en términos relativos es un error del 1,429%.

Trayectorias concéntricas

En el mecanizado en espiral, podemos observar que no se va a producir error en el mecanizado de las paredes verticales, por lo que podemos centrarnos sólo en el error producido sobre la superficie superior de cada parte del objeto.

En el primero de los planos de la figura 9.11 puede observarse que la longitud de la trayectoria viene dada por la expresión

$$l = \sum_{i=1}^{16} 4 \cdot (100 - 6i) \quad \text{Ecuación 9.22}$$

ya que son 16 los cuadrados concéntricos que forman la trayectoria, y sus perímetros disminuyen 6 mm cada uno con respecto al anterior. La longitud total es, por lo tanto $l=3136 \text{ mm}$.

Podemos aproximar el error si calculamos el porcentaje de superficie que cubre el objeto como en el caso anterior. Ya vimos que este porcentaje era del 87,17% del área del cuadrado por lo que la longitud de la trayectoria sobre la sección es de $l=2733,6512 \text{ mm}$.

Como ocurría en el caso anterior, el área de la sección del monte que se produce entre pasadas es $A=0,6234 \text{ mm}^2$, luego el error cometido será de $e=A \cdot l= 1704,15816 \text{ mm}^3$. Este es el único error, y supone, en términos relativos, un error del 0,341%, como vemos, considerablemente inferior al del mecanizado paralelo.

Modelo CSG

En el modelo CSG, para realizar un cálculo automático de las trayectorias podemos recurrir a cualquiera de las dos estrategias propuestas. Por lo tanto, los errores que se producen son:

- Trayectorias paralelas: 1,429%
- Trayectorias concéntricas: 0,341%

Modelo BRep

Puesto que las estrategias de mecanizado son las mismas que en el modelo anterior, si partimos de los planos de mecanizado y las intersecciones de éstos con el objeto, el error debe ser el mismo, es decir:

- Trayectorias paralelas: 1,429%
- Trayectorias concéntricas: 0,341%

Modelo ST

En este caso las trayectorias han sido trazadas con la ayuda del usuario. En cualquier caso, debemos hacer notar que el resultado de las trayectorias propuestas para este modelo coincide básicamente con las trayectorias concéntricas que ya hemos calculado, por lo que el error es el mismo:

- Trayectorias del usuario: 0,341%

	Facilidad de diseño	Facilidad de cálculo de trayectorias	Error al mecanizar		
			Paralelo	Concéntrico	Usuario
CSG	Excelente	Regular	1,429%	0,341%	-
BRep	Excelente	Regular	1,429%	0,341%	-
S-T	Excelente	Excelente	-	-	0,341%

Tabla 9.2

Comparación de los resultados de cada modelo para el ejemplo 1

9.4.3 Ejemplo 2

Para este segundo ejemplo hemos elegido un objeto más adecuado a la forma de proceder de los modelos superficiales. El objeto, con sus medidas y características se presenta en la figura 9.16.

Como en el caso anterior, hallaremos el volumen del objeto para poder calcular a continuación el error que se produce. El volumen total de este objeto se ha aproximado de manera sencilla: En primer lugar se ha discretizado la curva que representa el perfil superior de del objeto según el algoritmo de discretización que presentamos en el capítulo 5. A partir de la discretización es fácil calcular el área de la sección longitudinal vertical como la suma de los trapecoides que se obtienen. Basta ahora con multiplicar esa área por la profundidad del objeto (100 mm) para obtener un volumen total de $1060496,431 \text{ mm}^3$.

9.4.3.1 Diseño del objeto

Modelo CSG

El diseño de este objeto con el modelo sólido CSG se complica porque las superficies que presenta no se corresponden con las primitivas habituales de que se dispone. Por ello hemos de recurrir a utilizar superficies libres y convertirlas después a modelo sólido o utilizar una técnica de barrido a partir del perfil de la pieza.

Modelo BRep

Los modelos superficiales se adaptan mucho mejor a este tipo de objetos. Para obtener la superficie que forma el objeto podemos recurrir a las superficies libres directamente o utilizar el barrido como método de construcción.

Una buena opción sería diseñar dos curvas cerradas como las de la figura 9.17 y después realizar la operación de diferencia entre ellas, tal y como se presentó en el apartado 6.2.2.2 del capítulo 6. A continuación el perfil obtenido se puede utilizar para barrer una recta de longitud 100 mm dando lugar a las superficies que forman el objeto.

Modelo S-T

El modelo S-T, por sus características en común con el modelo BRep, nos permitiría realizar un diseño siguiendo el mismo esquema:

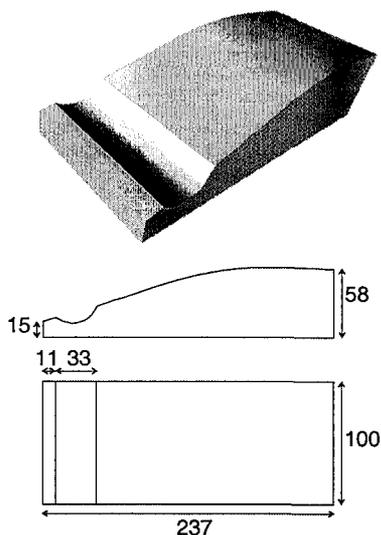


Figura 9.16
Objeto del ejemplo 2

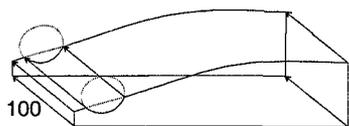


Figura 9.17
Diseño del objeto por barrido con el modelo BRep

diseño inicial del perfil como diferencia de dos curvas cerradas y barrido sobre una recta.

Sin embargo, podemos utilizar algunas de las características especiales del modelo para facilitar este diseño: se propone como esquema de diseño el siguiente (figura 9.18):

- Diseño de una única superficie por barrido de una curva abierta a lo largo de una recta, es decir, una extrusión.
- Diseño de una herramienta virtual que siga una trayectoria recta sobre la superficie anterior.

El proceso de diseño quedará completado cuando se definan completamente las trayectorias y herramientas.

9.4.3.2 Definición de las trayectorias

De nuevo podemos seguir dos esquemas para realizar el cálculo de las trayectorias de mecanizado: trayectorias paralelas a uno de los ejes y trayectorias concéntricas. Sin embargo, una diferencia importante separa a este ejemplo del anterior: en este caso podemos optar entre realizar el mecanizado a altura constante o a altura variable. En general, el primero se utilizará para un desbaste rápido inicial, mientras que en la fase de acabado de la pieza se utilizará un mecanizado a altura variable, por ofrecer mucha mejor calidad. Por lo tanto, a efectos del cálculo del error, tendremos en cuenta sólo el mecanizado final a altura variable.

Para mecanizar a altura constante es necesario calcular la intersección de cada plano de mecanizado con el objeto. Estas intersecciones dan lugar a regiones como las que se muestran en la figura 9.19 que debemos desbastar. Suponiendo que en cada plano profundizaremos 2 mm, necesitamos un total de 29 planos.

El mecanizado a altura variable nos permite reducir drásticamente el error producido. En este caso, trazaremos una superficie *offset* a la superficie real del objeto sobre la que se situarán las trayectorias. De esta manera evitamos mecanizar partes que no debemos.

A continuación pasamos a describir las peculiaridades de cada modelo en el cálculo de las trayectorias.

Modelo CSG

En la primera fase de desbaste rápido, el aspecto que diferencia a unos métodos de otros es la obtención de las intersecciones entre los planos de mecanizado y el objeto. Puesto que para este ejemplo no estamos tratando con primitivas básicas, debemos proporcionar los métodos de cálculo de la intersección. Como se parte de superficies libres, la opción más extendida es implementar métodos aproximados basados en la subdivisión de la superficie, técnicas que tienen un alto coste computacional.

Tras el cálculo de los planos, debemos mecanizar la región rectangular obtenida por uno de los dos esquemas descritos: zig-zag o espiral.

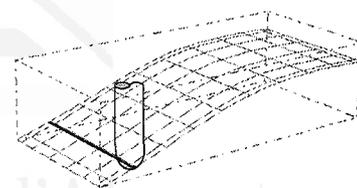


Figura 9.18
Diseño del objeto con el modelo S-T

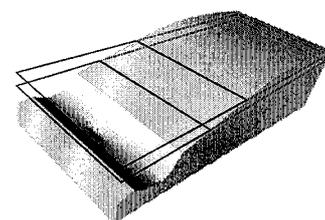


Figura 9.19
Intersección del objeto con los planos de mecanizado a altura constante

En cuanto al mecanizado a altura variable requiere el cálculo de una superficie *offset*, métodos que tienen un alto coste computacional. Este es un tema central en el ámbito del CAD/CAM durante los últimos años [Jimeno, 2001].

A partir de la superficie paralela obtenida, las trayectorias de mecanizado consistirán en recorrer la superficie en zig-zag o en espiral. Para ello puede recurrirse a mecanizados isoparamétricos (ya vimos en el capítulo 3 que los resultados no son buenos para todo tipo de superficies) o utilizar técnicas como las de proyección y despliegue descritas en el capítulo 5.

En resumen, las trayectorias con las que construiremos el objeto son:

- Trayectoria de desbaste inicial, en zig-zag o en espiral.
- Trayectoria de acabado, en zig-zag o en espiral.

Modelo BRep

En este caso el modelo superficial comparte buena parte de su problemática para el cálculo de las trayectorias con el modelo sólido. Dos son los problemas existentes:

- Cálculo de la intersección de los planos de mecanizado con el objeto.
- Obtención de una superficie paralela u *offset* a una dada.

Aunque las filosofías de diseño sean distintas, en ambos casos se terminan representando los objetos mediante superficies libres, lo que hace que compartan estos problemas a la hora de obtener las trayectorias.

Para el modelo BRep, las trayectorias son, de nuevo:

- Trayectoria de desbaste inicial, en zig-zag o en espiral.
- Trayectoria de acabado, en zig-zag o en espiral.

Modelo S-T

Un objeto como el del ejemplo 2 presenta algunas características que hacen que su obtención mediante un modelo S-T sea más sencilla que para los otros modelos.

En primer lugar, puesto que el objeto final en nuestro modelo se define a partir de las trayectorias y las herramientas y no a partir de las superficies, no es necesario tratar aquellas superficies por las que no pasan trayectorias, como ocurre en este caso con los laterales y la base del objeto. Además, como ya hemos visto, para definir la parte superior sólo es necesario construir una superficie.

La superficie superior nos servirá para calcular las intersecciones con los planos de mecanizado para el desbaste a altura constante. Puesto que la superficie es más simple que en los dos modelos anteriores, el cálculo de la intersección es más sencillo.

En cuanto al acabado a altura variable, en este caso proponemos utilizar el método de compensado en tres dimensiones descrito en el

apartado 6.2.6 del capítulo 6. De esta manera evitamos el costoso cálculo del *offset* de la superficie.

Por otro lado, la forma cóncava que presenta el objeto sobre su parte superior puede obtenerse de manera sencilla mediante una herramienta virtual cuya sección coincida con la sección de esta forma cóncava (figura 9.20). A continuación, basta con hacer pasar esta herramienta sobre una recta situada sobre las superficie, gracias a los procesos de proyección o despliegue.

Por lo tanto, las trayectorias que nos permiten terminar de diseñar y mecanizar este objeto son:

- Trayectoria de desbaste inicial, en zig-zag o en espiral, sobre una única superficie.
- Trayectoria de acabado, en zig-zag o en espiral, compensada según el método propuesto.
- Trayectoria recta de la herramienta virtual para construir la forma cóncava.

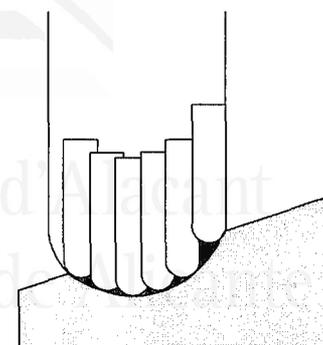


Figura 9.20
Herramienta virtual emulada
con una herramienta esférica

9.4.3.3 Cálculo del error

Como ya hemos comentado, vamos a considerar sólo el error que produce la última trayectoria de acabado a altura variable. El objeto de este ejemplo, a diferencia del anterior, no es simétrico, por lo que la orientación de las trayectorias en el caso de ser paralelas va a influir en el error final. Por eso distinguiremos tres posibles estrategias: trayectorias paralelas al eje X, trayectorias paralelas al eje Y y trayectorias concéntricas.

Trayectorias paralelas al eje X

Como ya vimos para el ejemplo 1, se producen dos tipos de errores: sobre las superficies mecanizadas directamente y sobre las paredes perpendiculares.

En el primer caso, debemos obtener la longitud lineal de la trayectoria. Partiendo de la discretización de la curva del perfil, calculamos su longitud, que es, aproximadamente, la suma de la longitud de los tramos rectos. La curva del perfil, en su parte superior, tiene una longitud de $l=248,54522$ mm. Suponiendo una herramienta de radio $r=2$ mm y un solapamiento entre pasadas de la herramienta de $s=1$ mm necesitamos 33 pasadas de herramienta para cubrir los 100 mm de anchura del objeto. Esto hace que la longitud total de las trayectorias sea de $33 \cdot 248,54522=8201,8926$ mm. Por otro lado el área de la sección del monte que se produce entre pasadas es $A=0,6234$ mm², si utilizamos las ecuaciones 9.1 a 9.6. El error en la superficie superior será, por lo tanto, de $e_1=A \cdot l= 5113,05985$ mm³.

En cuanto al error cometido sobre las paredes perpendiculares, éste sólo se producirá en los extremos de las trayectorias. La longitud total de los montes será de $33 \cdot (15+58)=2409$ mm. Puesto que el área de los montes es la misma de antes, el error en las paredes es de $e_2=A \cdot l= 1501,7706$ mm³.

Por último, el error total cometido es $e=e_1+e_2= 6614,83045 \text{ mm}^3$, lo que supone un 0,62375% del volumen total.

Trayectorias paralelas al eje Y

En este caso también debemos distinguir entre el error cometido al mecanizar las superficies mecanizadas directamente y el que se produce sobre las paredes perpendiculares.

Para calcular el primero de los errores partiremos de la longitud total de la superficie superior en sentido longitudinal. Debemos distinguir entre las dos regiones que la forman: Por un lado la región convexa, más grande, y por el otro la región cóncava transversal. Basándonos en la discretización de la curva, la longitud de la sección de la región convexa es de 208,52617 mm y la de la cóncava de 40,01905.

Para mecanizar la región convexa debemos utilizar un total de 69 pasadas, cada una con longitud 100, es decir, una trayectoria de longitud $l=6900 \text{ mm}$. Aunque la curva que forma esta región es una curva libre, podemos aproximarla fácilmente mediante una circunferencia de radio 400 mm, para poder utilizar así las ecuaciones 9.7 a 9.15 y obtener el área del monte que se produce. Esta área es de $A=0,62573 \text{ mm}^2$, de tal manera que el error que se produce al mecanizar esta superficie es $e_1=l \cdot A=4317,537 \text{ mm}^3$.

En cuanto a la región cóncava, podemos aproximar su perfil mediante una circunferencia de radio 22,5 mm, de tal manera que el área del monte que se produce lo calculamos con las ecuaciones 7.4 a 7.13 y su valor es $A=0,58432 \text{ mm}^2$. El error cometido, teniendo en cuenta que necesitamos 13 pasadas de 100 mm de longitud, es de $e_2=759,61937 \text{ mm}^3$.

Sólo se produce error sobre las paredes perpendiculares situadas en los extremos de las trayectorias. La longitud de estos montes será de 3669,22832 mm y su área de $0,6234 \text{ mm}^2$, por lo que el error en las paredes es de $e_3=A \cdot l= 2287,39693 \text{ mm}^3$.

Para terminar calculamos el error total que será $e=e_1+e_2+e_3= 7364,5533 \text{ mm}^3$, es decir, un 0,69444% del volumen total.

Trayectorias concéntricas

El uso de trayectorias concéntricas tiene la ventaja de que no se producen errores sobre las paredes perpendiculares. Por ello, nos basamos solamente en las trayectorias situadas sobre la superficie superior. El error será, en este caso, de $5095,10811 \text{ mm}^3$, que supone un 0,48044% sobre el volumen total del objeto.

Modelo CSG

Las tres estrategias analizadas son aplicables en el modelo CSG. Los errores cometidos son los que se han calculado, que mostramos a continuación a modo de resumen:

- Trayectorias paralelas al eje X: 0,62375%
- Trayectorias paralelas al eje Y: 0,69444%

- Trayectorias concéntricas: 0,48044%

Modelo BRep

Una vez construido el objeto, el cálculo automático de las trayectorias nos lleva a tres posibilidades coincidentes con las analizadas para el modelo anterior. Por lo tanto, los errores cometidos son equivalentes:

- Trayectorias paralelas al eje X: 0,62375%
- Trayectorias paralelas al eje Y: 0,69444%
- Trayectorias concéntricas: 0,48044%

Modelo S-T

En la propuesta presentada el usuario se encarga de trazar tres trayectorias, tal y como ya hemos visto: desbaste inicial a altura constante (esta trayectoria no interviene en el error final), acabado a altura variable y trayectoria recta con herramienta virtual. En la figura 9.21 se muestra el resultado de la poligonalización de este objeto mediante el algoritmo modificado de *Marching Cubes* descrito en el capítulo 8.

Para el acabado a altura variable calculamos, a partir de la discretización de la curva del perfil, su longitud como la suma de la longitud de los tramos rectos, sin contar con la región cóncava. Esta longitud es de $l=208,52617$ mm. Si necesitamos 69 pasadas de 100 mm cada una, la longitud total de las trayectorias será de $69 \cdot 208,52617=14388,30573$ mm. Puesto que el área de la sección del monte es $A=0,6234$ mm², el error en la superficie superior es $e_1=A \cdot l=4317,537$ mm³.

Por otro lado, podemos calcular el error que se produce al emular la herramienta de la figura 9.20 con una herramienta esférica de radio $r=2$ mm con solapamiento entre pasadas de $s=1$ mm siguiendo las ecuaciones 7.4 a 7.13. El área de este error es de $7,59616$ mm², que al multiplicarlo por la longitud total de la trayectoria nos da un error en volumen de $e_2=759,616$ mm³. El error total cometido será, por lo tanto, de $e=e_1+e_2=5077,153$ mm³, que es un 0,47875%.

En resumen, para la estrategia propuesta, el error es:

- Trayectorias del usuario: 0,47875%

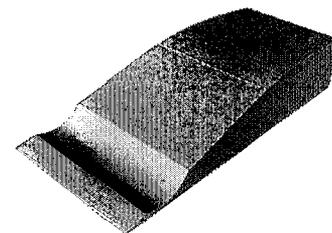


Figura 9.21
Poligonalización del ejemplo 2 mediante el algoritmo modificado de *Marching Cubes*

	Facilidad de diseño	Facilidad de cálculo de trayectorias	Error al mecanizar			Usuario
			Paralelo X	Paralelo Y	Concéntrico	
CSG	Regular	Regular	0,62375%	0,69444%	0,48044%	-
BRep	Excelente	Regular	0,62375%	0,69444%	0,48044%	-
S-T	Excelente	Buena	-	-	-	0,47875%

Tabla 9.3

Comparación de los resultados de cada modelo para el ejemplo 2



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

CAPÍTULO 10

**CONCLUSIONES Y LÍNEAS FUTURAS
DE INVESTIGACIÓN**

10.1 CONCLUSIONES

Presentamos aquí las conclusiones que se derivan de los capítulos anteriores y, especialmente, de los experimentos realizados.

Analizando los resultados en cuanto a los parámetros cualitativos de que hablábamos en el capítulo 9, se deduce que el modelo Superficie-Trayectoria se muestra bastante equilibrado en todos los aspectos considerados, frente a los otros dos modelos que destacan sólo en algunos aspectos. Así, el modelo S-T es comparable con el BRep en cuanto a la definición de superficies libres, pero lo supera en la facilidad de construcción del entorno de usuario y en la posibilidad de realizar operaciones entre objetos. Comparado con el CSG, aunque no llega a alcanzar su idoneidad en el manejo de las operaciones booleanas y en el diseño del interfaz, tiene unos niveles aceptables y, sobre todo, soluciona el gran problema del diseño de las superficies libres.

En cuanto a los parámetros cuantitativos, se ha comprobado en las pruebas realizadas que el modelo S-T facilita en general el diseño y permite obtener unas trayectorias más adecuadas. Como consecuencia de esto, el error que se produce en la pieza final es, cuanto menos, comparable con el mejor de los resultados posibles obtenidos utilizando otros modelos y, en la mayoría de los casos, es inferior.

10.1.1 Consecución de los objetivos

En este punto vamos a retomar los objetivos que nos habíamos marcado en el capítulo 2, para constatar en qué grado se han alcanzado.

Nuestro primer objetivo general era facilitar el diseño de objetos tridimensionales y hacerlo más versátil. Para conseguir esta meta, nos proponíamos tres subobjetivos más concretos:

- Definir un modelo que incorporara los procedimientos de diseño más avanzados: superficies libres y operaciones entre ellas. En el modelo presentado no sólo se utilizan las superficies libres como elemento base inicial para el diseño sino que, además, se ha definido un conjunto de constructores que facilitan el diseño de estas superficies (extrusión, rotación, cruce y por secciones) y de curvas situadas sobre ellas (proyección y despliegue). En cuanto a las operaciones entre superficies, el hecho de que el objeto se defina en función de la trayectorias y no de las propias superficies evita tener que realizar esas operaciones.
- Añadir nuevas funcionalidades para facilitar el diseño. En este sentido, las herramientas virtuales suponen una aportación importante, pues permiten obtener formas complejas a partir de objetos geoméricamente sencillos.
- Visualizar los objetos de forma realista. Se ha diseñado y construido un método que permite visualizar con una calidad

aceptable el objeto diseñado. El método se basa en el algoritmo de *Marching Cubes* y resulta fácil de implementar.

El segundo de los objetivos era mejorar la calidad del mecanizado conteniendo la complejidad de los métodos. Este objetivo se debía alcanzar a través de dos metas específicas:

- Utilizar las directrices dadas por el usuario durante la fase de diseño. De nuevo, las herramientas virtuales suponen un avance en este sentido: el usuario las define para hacer el diseño pero, a la vez, está guiando el proceso de mecanizado. De esta manera la obtención de las trayectorias reales de mecanizado es prácticamente inmediata y la calidad del resultado es mayor.
- Asegurar la integridad del modelo. El modelo propuesto asegura de manera intrínseca que los objetos diseñados también se puedan fabricar. De hecho, se utilizan las trayectorias como elemento de diseño, de tal manera que si la emulación de las herramientas virtuales es posible y correcta, el objeto se podrá fabricar.

Como objetivo final, se planteaba limitar los costes de programación e implantación del sistema a través de estos objetivos concretos:

- Construir un modelo sencillo e intuitivo sin grandes requerimientos en cuanto a tipo de ordenador ni a periféricos especiales. Tal y como se plantea el modelo, los métodos propuestos no tienen alta complejidad y pueden ser implementados sin problemas sobre un ordenador personal de características medias. En cuanto a periféricos, no se plantea ninguno más allá de los habituales.
- Adecuar el modelo geométrico al problema y a la forma de trabajar del profesional del sector. La filosofía de trabajo del modelo S-T se aproxima bastante a la de un mecánico de máquinas herramientas, aunque incorpora muchas facilidades. Por eso no debe ser muy difícil el implantar esta tecnología en empresas que no cuenten con grandes expertos.

Por último señalar que el modelo puede suponer una aportación importante para introducir en las pequeñas industrias sistemas CAD/CAM específicos que resuelvan los problemas de forma adecuada.

10.1.2 Características del modelo Superficie-Trayectoria

A modo de resumen, presentamos a continuación las principales características del modelo propuesto.

- Frente a otros sistemas, la fase de diseño y la de fabricación entendida como la generación de trayectorias reales de mecanizado se encuentran íntimamente relacionadas. Esto permite que la obtención de las trayectorias finales sea prácticamente inmediata a partir del diseño.

- El uso de herramientas virtuales supone añadir versatilidad a las capacidades de diseño. En este sentido, en el capítulo 9 se ha constatado que el diseño con nuestro modelo es intuitivo y sencillo, superando, en la mayoría de las ocasiones, a los demás modelos en cuanto a facilidad de uso.
- La filosofía de diseño del modelo facilita poder utilizar el conocimiento del usuario para guiar todo el proceso ingenieril, mejorando así la calidad final. De hecho, tal y como se ha comprobado en el capítulo 9, el error que se produce es al menos tan pequeño como el de la mejor de las demás posibilidades.
- Los diseños realizados son fácilmente reutilizables. Así, entre otras posibilidades, podemos utilizar trayectorias ya calculadas con diferentes herramientas o emular las mismas herramientas virtuales con otras herramientas reales, lo que independiza el diseño de la máquina concreta.

10.1.3 Principales aportaciones

Para terminar con las conclusiones, resaltamos a continuación las aportaciones que, a nuestro juicio, son más importantes en nuestra propuesta.

El concepto de **herramienta virtual** es, sin duda, la principal aportación de la tesis. En este sentido, se ha presentado una herramienta que puede ser muy útil tanto para el diseño como para la fabricación de calidad de los objetos. Además, permite independizar el diseño de la máquina concreta de mecanizado.

Para apoyar el concepto de herramienta virtual se ha desarrollado un conjunto completo de constructores geométricos adaptados al problema:

- En cuanto a los constructores para curvas y superficies, las mayores aportaciones se refieren al método adaptativo de discretización de curvas y a los constructores de curvas por proyección y por despliegue.
- Los constructores de trayectorias que, a nuestro juicio, suponen un mayor avance son el método de cálculo del *offset* por discretización, las operaciones booleanas entre curvas cerradas, los procedimientos de desbaste en zig-zag y en espiral y el mecanizado a altura variable.
- Entre los constructores geométricos de herramientas destacan los procedimientos para emulación de las herramientas virtuales con herramientas reales y el cálculo de trayectorias para herramientas de rotación y de traslación.

Por último, aunque no menos importante, resaltar los constructores de objetos, en especial, la adaptación del algoritmo de *Marching Cubes* a nuestro problema y la solución incremental propuesta, junto con el uso de la coherencia espacial para reducir costes y los métodos de prueba de interioridad de los vértices y el punto de corte de la herramienta con las aristas.

También debemos destacar las publicaciones a que ha dado lugar este trabajo hasta el momento.

- [Molina, 1997] Molina Carmona, Rafael; Puchol García, Juan A. y Salas Pérez, Faustino. *Definición y visualización de superficies recortadas*. Actas del IX Congreso Internacional de Ingeniería Gráfica, págs. 465-474. 1997.
- [Puchol, 1997] Puchol García, Juan A.; Molina Carmona, Rafael y García Quintana, Candelaria. *Conversión paramétrica de Blobs a modelo de fronteras en tiempo real*. Actas del IX Congreso Internacional de Ingeniería Gráfica, págs. 465-474. 1997.
- [Puchol, 1999] Puchol García, Juan A.; Sáez Martínez, Juan M. y Molina Carmona, Rafael. *Postproceso de rectificación de metapolígonos*. Actas del IX Congreso Español de Informática Gráfica, CEIG99, págs. 331-342. 1999.
- [Sáez, 2000] Sáez Martínez, Juan M.; Jimeno Morenilla, Antonio M.; Puchol García, Juan A. y Molina Carmona, Rafael. *Adaptación del PRM sobre superficies libres. Aplicación a entornos CAD/CAM*. Actas del X Congreso Español de Informática Gráfica, CEIG00. 2000.

10.2 LÍNEAS FUTURAS DE INVESTIGACIÓN

Vamos a presentar un conjunto de líneas futuras de investigación que permitirán avanzar en las aportaciones del modelo.

10.2.1 Optimización de las trayectorias

En un sistema de fabricación asistida por computador, se persigue que las trayectorias calculadas sean *válidas* (que no existan invasiones en el material que no deba ser desbastado ni sobrantes de material), que generen objetos de *calidad* (que el efecto de montes y valles sea mínimo) y que sean *eficientes* (que el mecanizado de la pieza consuma el menor tiempo posible). A lo largo de la tesis, se han cubierto los aspectos de validez y calidad. Aunque por la forma en que se construyen los objetos en nuestro modelo las trayectorias obtenidas van a ser más eficientes que en buena parte de los otros modelos, es posible mejorar estos aspectos.

Es difícil cuantificar la eficiencia de una trayectoria, pues depende de muchos factores ajenos a la propia trayectoria, tales como el material que vamos a desbastar, la herramienta utilizada, la máquina que porta la herramienta, etc. Sin embargo, existen dos requerimientos que sí dependen de las trayectorias generadas:

- No debemos mecanizar repetidamente una misma región ya que aumenta el tiempo de mecanizado y puede menguar la calidad de la superficie obtenida.

- Debemos evitar un número excesivo de retracciones de la herramienta. Evidentemente será necesario un número mínimo, pero la planificación adecuada de las trayectorias hará disminuir drásticamente el tiempo de mecanizado.

Para planificar el orden de las trayectorias podemos recurrir a la construcción de grafos de mecanizado que, convenientemente tratados, reducen el problema al clásico del Viajante de Comercio [Arkin, 2000]. Esta sería una línea de trabajo futura que permitiría disminuir los tiempos de mecanizado.

10.2.2 Mecanizado con cinco ejes

En el mecanizado con cinco ejes la herramienta tiene dos grados de libertad adicionales que controlan su orientación angular [Jerard, 1989b], [Li, 1994]. Así, podemos situarla en la posición que deseemos y orientarla de la forma más favorable.

Este tipo de mecanizado tiene grandes ventajas: puede favorecer la velocidad de fabricación, la calidad del acabado o ambas a la vez. Habitualmente la orientación normal a la superficie es la más conveniente por velocidad y calidad, lo que hace que para superficies libres el uso de cinco ejes sea muy adecuado.

A pesar de las ventajas de este esquema de mecanizado, su uso no está completamente extendido debido a la dificultad de generar trayectorias complejas evitando colisiones con el material sin desbastar. Esto hace que el problema del compensado con superficies tenga una complejidad muy alta.

Como futura línea de trabajo se plantea el estudio de este tipo de métodos y de otros nuevos y su incorporación al modelo S-T.

10.2.3 Cálculo de *offsets* y mecanizado en espiral

Ya vimos en el capítulo 6 que, además del método de cálculo de *offsets* basado en la discretización de la curva, existen varios métodos cuyos resultados son curvas continuas [Elber, 1997]. Ciertas máquinas de control numérico permiten realizar interpolaciones y aproximaciones basadas en Splines, por lo que puede ser interesante obtener paralelas en esta representación.

Se propone como futura investigación, desarrollar estos métodos como base para otros procedimientos. Obsérvese que, si partimos de una curva continua, los métodos de eliminación de tramos no válidos entre autointersecciones (sección 6.2.2.1), de operaciones booleanas entre curvas cerradas (sección 6.2.2.2) y de cálculo del *offset* de varias curvas (sección 6.2.2.3) siguen siendo válidos, sin cambiamos los métodos de intersección entre rectas por la intersección entre curvas. Por otro lado, el algoritmo de cálculo del desbaste en espiral (sección 6.2.4.2) no necesita ninguna modificación.

10.2.4 Algoritmos de visualización

Hemos presentado un algoritmo basado en *Marching Cubes* cuya principal característica es su incrementalidad. En el futuro, proponemos realizar algunas mejoras en este algoritmo:

- Estudio detallado del tamaño del cubo y del paso de mecanizado, así como la determinación de los valores más adecuados para estos parámetros, teniendo en cuenta su influencia en el coste computacional del algoritmo y en la calidad de la poligonalización obtenida.
- Desarrollo de nuevas estructuras de datos de tipo árbol para permitir que la precisión sea adaptativa, es decir, para aumentar el número de polígonos en las zonas de alta curvatura y disminuirlo donde no sea necesario.
- Adaptación al modelo S-T de diversos métodos de mejora de la malla poligonal, tema en el que hemos presentado diversas propuestas para otros modelos [Puchol, 1999] [Sáez, 1999] [Sáez, 2000].



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

ANEXO

IMPLEMENTACIÓN

A.1 IMPLEMENTACIÓN

Se ha desarrollado una aplicación CAD/CAM que utiliza el modelo Superficie-Trayectoria para el diseño y la fabricación de pisos de calzado y moldes para estos pisos. Esta aplicación se enmarca en una colaboración con el Instituto Tecnológico del Calzado y Conexas (INESCOP). Presentamos en este anexo las principales cuestiones técnicas referentes a esta aplicación.

A.1.1 Plataforma y software utilizados

A lo largo del proyecto de desarrollo de la aplicación, se han venido utilizando distintas máquinas. Todas ellas han sido ordenadores personales tipo PC con procesador Pentium a distintas velocidades y sistema operativo Microsoft Windows versiones 95 y 98. Las características de la última plataforma utilizada, con las que se han efectuado las pruebas finales se muestran en la tabla A.1.

Sistema operativo	Microsoft Windows 98
CPU	Pentium III 800 MHz
RAM	512 Mb
Disco Duro	40 Gb
Tarjeta gráfica	NVIDIA TNT2 Model 64
RAM	32 Mb
Bus	AGP

Tabla A.1
Características de la plataforma hardware utilizada para las pruebas

En cuanto al lenguaje de programación, nos hemos inclinado por C++, por ser un lenguaje que aúna la potencialidad de C con las ventajas de un lenguaje orientado a objetos: encapsulación, herencia, reusabilidad, etc.

El compilador elegido es Borland C++ en sus versiones 5.0 y 5.2. Este compilador destaca por incorporar algunas capacidades visuales y la potente librería de clases OWL (Object Windows Library), lo que facilita el prototipado rápido.

A.1.2 Interfaz de usuario

Para facilitar el diseño y la fabricación de pisos y moldes de calzado se ha construido un interfaz de usuario adaptado a las necesidades de este tipo de objetos. En concreto, el entorno se estructura en cinco tipos de vistas, con las que el usuario puede acceder fácil y rápidamente a las diferentes partes del piso: vista de proyecto, vista de dimensiones, vista de diseño, vista de escalado y vista de simulación.

A.1.2.1 Vista de proyecto

Permite el acceso a las otras cuatro vistas. Se estructura en forma de árbol desplegable. En la figura A.1 se muestra la vista de proyecto con el árbol desplegado.

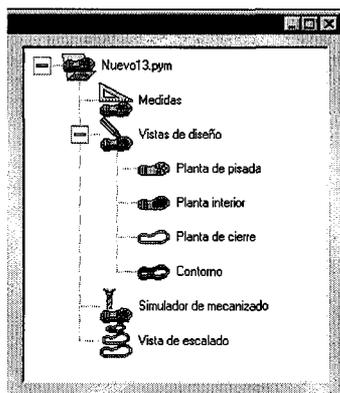


Figura A.1
Vista de proyecto

A.1.2.2 Vista de dimensiones

Permite determinar las medidas básicas del objeto, que se toman como dimensiones del material a desbastar. El cuadro de diálogo que permite introducir estos valores se muestra en la figura A.2.

A.1.2.3 Vista de diseño

Existen en realidad cuatro vistas de este tipo para acceder fácilmente a las cuatro partes que componen un piso: interior, pisada, cierre y contorno. Todas ellas tienen las mismas características, cambiando sólo el punto de vista del usuario.

Este tipo de vista es la vista principal del objeto y sobre la que se realiza tanto el diseño, como la generación de las trayectorias y la definición y emulación de herramientas virtuales. Como puede observarse en la figura A.3 consta de cuatro subvistas: tres bidimensionales (planta, alzado y perfil) y una tridimensional perspectiva. Las tres subvistas bidimensionales permiten la edición simultánea de cualquier figura. La tridimensional permite cambiar el punto de vista de la cámara de forma interactiva.

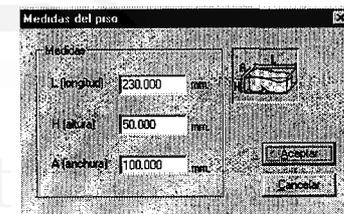


Figura A.2
Vista de dimensiones

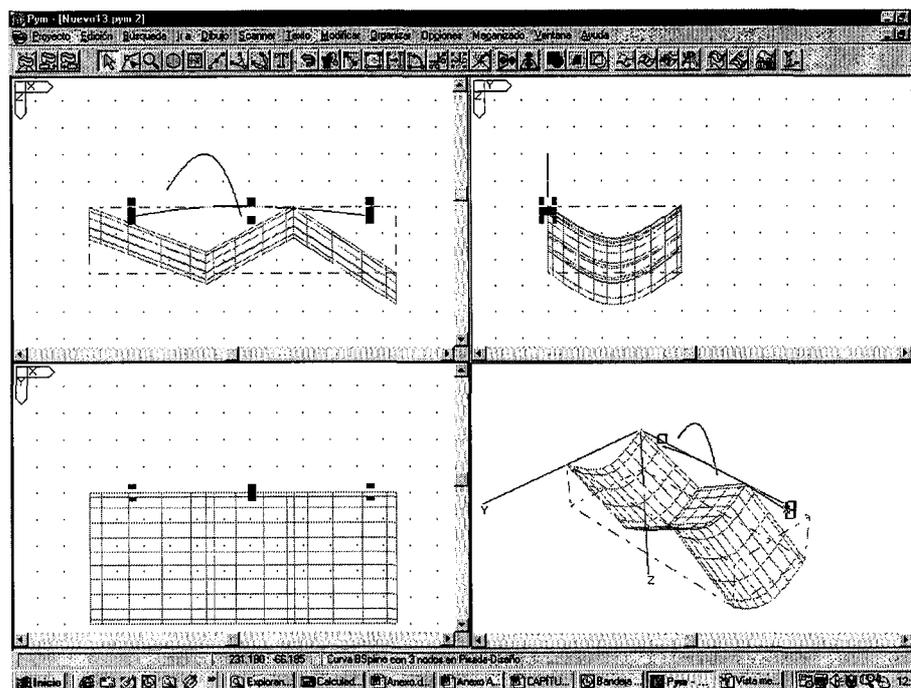


Figura A.3
Vista de diseño

Diseño del objeto

Se han incluido todos los constructores de curvas y superficies descritos en los capítulos anteriores: curvas BSpline de aproximación y de interpolación, superficies por extrusión, rotación, cruce y secciones, proyección y despliegue de curvas, etc. Se han incluido también otras funcionalidades: operaciones booleanas entre curvas cerradas, cálculo de paralelas, manejo de capas, etc.

Definición y emulación de herramientas virtuales

Se ha desarrollado un asistente que facilita el proceso de definición de las herramientas. En él se incluyen los diferentes procesos de emulación descritos a lo largo del documento. En la figura A.4 se presentan las diferentes etapas del asistente.

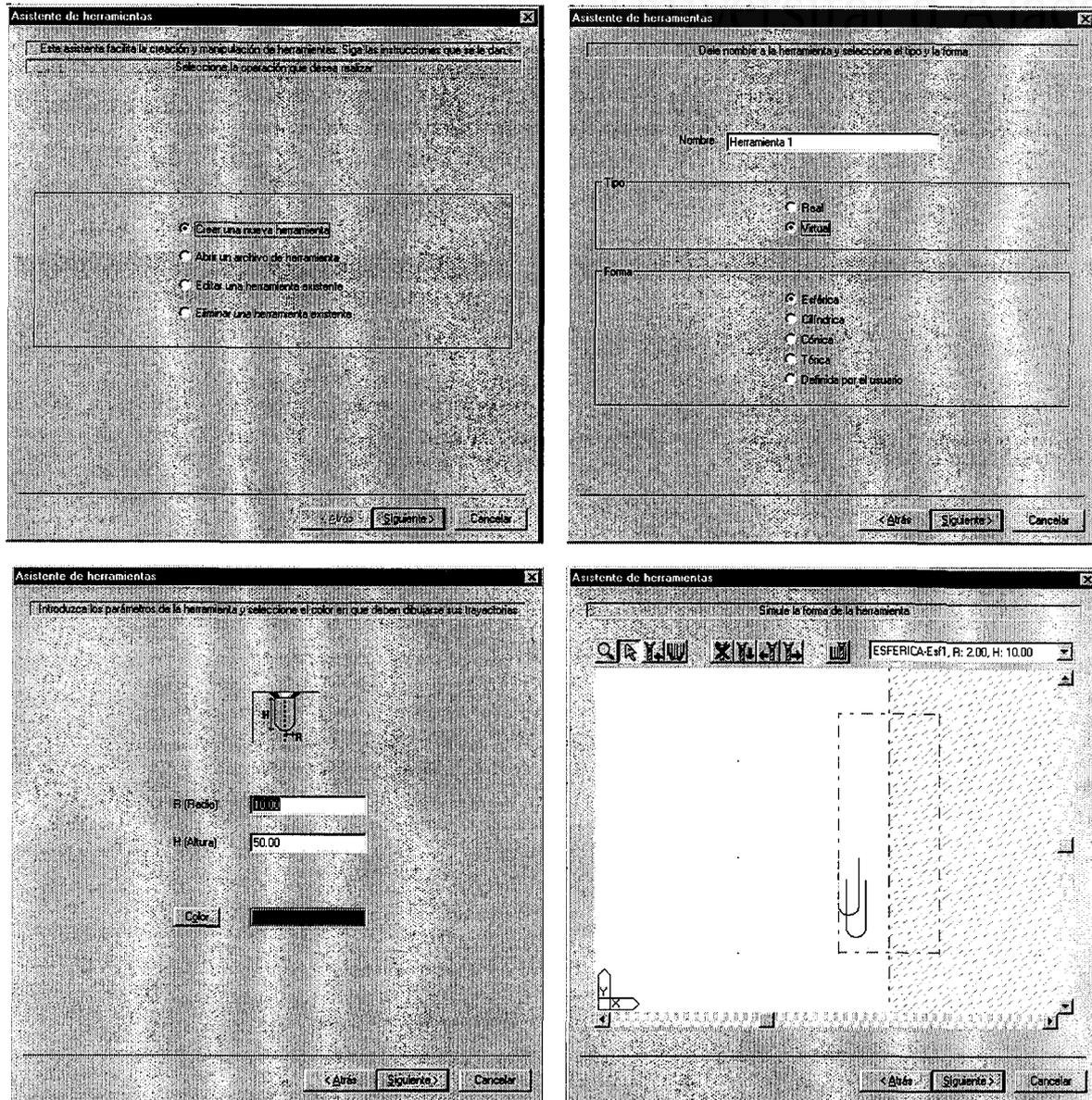


Figura A.4
Asistente para la definición y emulación de herramientas virtuales

Generación de trayectorias

Para la generación de las trayectorias de mecanizado se ha desarrollado otro asistente que permite definir trayectorias de desbaste o perfilado de diferentes tipos (zig-zag o concéntricas) y con distintos modos de profundización. El asistente se muestra en la figura A.5.

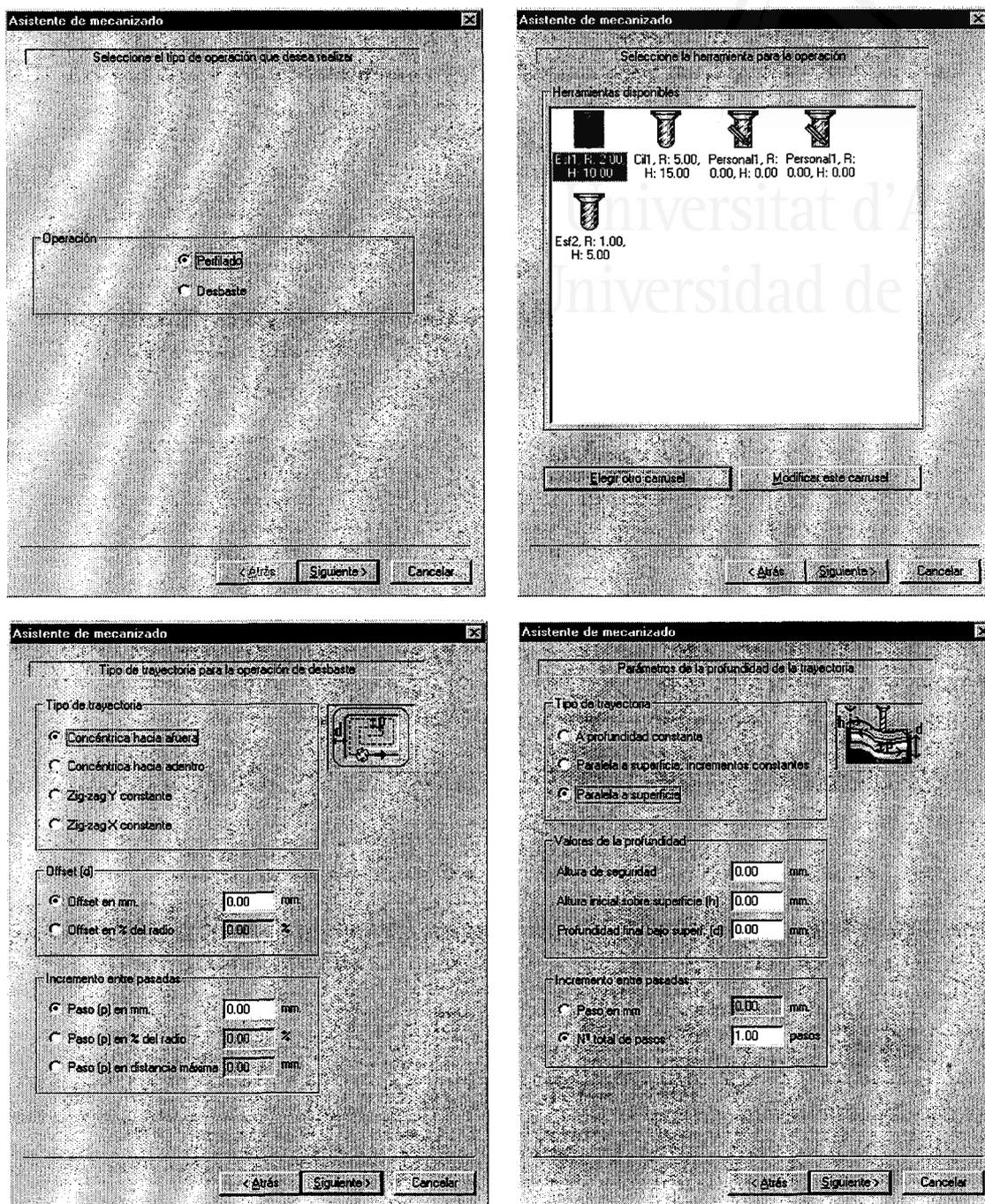


Figura A.5
Asistente para la generación de trayectorias

A.1.2.4 Vista de escalado

En un futuro esta vista contendrá los diferentes métodos automáticos de cálculo de las escalas para obtener las diferentes tallas de calzado a partir de un modelo base. Este tipo de métodos quedan fuera del alcance de esta tesis.

A.1.2.5 Vista de simulación

Mediante esta vista se facilita el proceso de verificación de la corrección de las trayectorias. Se ha contemplado una simulación en dos dimensiones (figura A.6) y otra en tres dimensiones basada en el método de poligonalización incremental descrito en el capítulo 8. Por el momento, la aplicación incluye sólo la simulación plana. El procedimiento de poligonalización para la simulación y para la vista tridimensional genera un archivo que es mostrado a continuación mediante un visualizador en OpenGL de libre distribución (GLView).

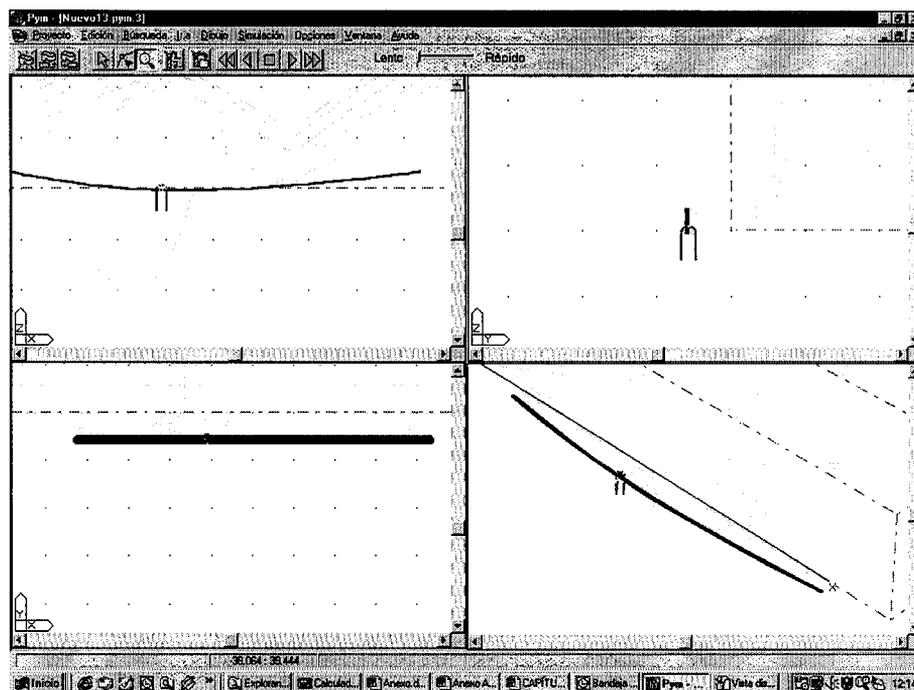


Figura A.6
Vista de simulación bidimensional

A.1.3 Jerarquía de clases

Para la construcción de la aplicación se ha partido de una jerarquía de clases previa que funciona como un núcleo gráfico 2D. Para este proyecto se ha generalizado este núcleo a tres dimensiones y se han generado nuevas clases para implementar el modelo Superficie-Trayectoria. Las principales clases de que consta el sistema son:

- TPad y T3DPad: Son contenedores de figuras. TPad es genérico y de él deriva T3DPad, un contenedor especializado de figuras en tres dimensiones.
- T2DPoint y T3DPoint: Son puntos en dos y tres dimensiones respectivamente.
- TCadShape: Es la clase de las figuras, de la cual derivan un conjunto de clases de comportamiento especializado que implementan cada una de las figuras tridimensionales que es posible diseñar. Todas ellas estarán formadas por puntos tridimensionales. Las clases de las figuras son:

- TCadLine: Recta.
 - TCadRect: Rectángulo.
 - TCadRountRect: Rectángulo de esquinas redondeadas.
 - TCadPolyline: Polilínea.
 - TCadBSpline: Curva BSpline (de aproximación).
 - TCadIBSpline: Curva BSpline (de interpolación).
 - TCadEllipse: Circunferencia y elipse.
 - TCadBezier: Curva Bézier.
 - TCadSurface: Superficie BSpline.
 - TCadToolPath: Trayectoria de herramienta.
- TGuidelines: Son las líneas guía o dimensiones del material a desbastar.
 - TCarousel: Es el carrusel o conjunto de herramientas y, por lo tanto, es un contenedor de herramientas.
 - TTool: Es la clase de las herramientas de mecanizado. Incluirá todos los parámetros propios de la herramienta.

Cada una de estas clases tiene un conjunto de métodos que permiten construirlas utilizando los algoritmos desarrollados.

En la figura A.7 se muestra la jerarquía de las clases. Las flechas simples indican la relación de herencia, mientras que las dobles representan atributos.

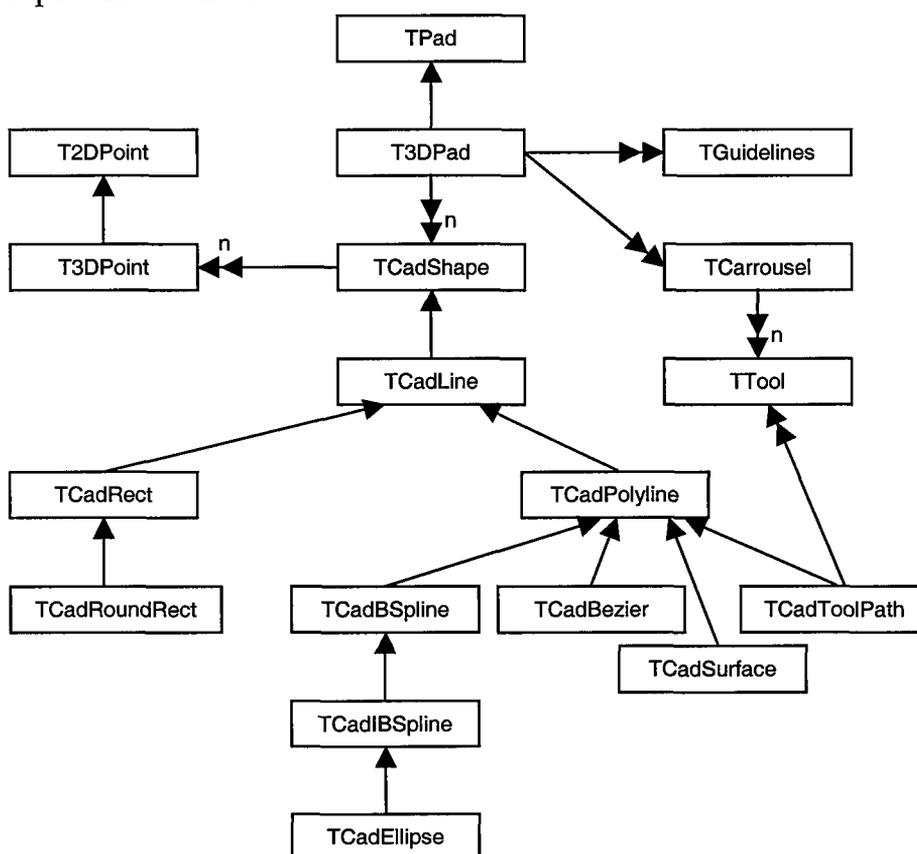


Figura A.7
Jerarquía de clases de la aplicación



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

BIBLIOGRAFÍA

- [Arkin, 2000] Arkin, Esther M.; Held, Martin y Smith, Christopher L. *Optimization Problems Related to Zigzag Pocket Machining*. Algorithmica vol. 26, num. 2, págs. 197-236. 2000.
- [Bowyer, 1999] Bowyer, Adrian. *The Bath final-year undergraduate Geometric Modelling course*. University of Bath. Department of Mechanical Engineering. 1999.
- [Chu, 1998] Chu, Chih-Hsing y You, Chun-Fong. *Operation Planning in NC Programming Based on CAD Systems*. Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan. 1998.
- [Davis, 1994] Davis, Martin D.; Sigal, Ron y Weyuker, Elaine J. *Computability, Complexity and Languages. Fundamentals of Theoretical Computer Science. Second Edition*. Academic Press. 1994.
- [Dodgson, 2000] Dodgson, Neil A. *Some Mathematical Elements of Graphics*, University of Cambridge, Computer Laboratory. 2000.
<http://www.cl.cam.ac.uk/Teaching/2000/AGraphHCI/SMEG/node1.html>
- [Eberly, 2002] Eberly, David. *Roots of Polynomial*, Magic Software Documentation, 2002.
<http://www.magic-software.com/Documentation/algebra.pdf>
- [Elber, 1997] Elber, Gershon; Lee, In-Kwon y Kim, Myung-Soo. *Comparing Offset Curve Approximation Methods*. IEEE Computer Graphics & Applications, págs. 62-71. May-June 1997.
- [Farin, 1993] Farin, Gerald. *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide*. Academic Press, Inc. 1993.
- [Faux, 1979] Faux, I.D. y Pratt, M.J. *Computational Geometry for Design and Manufacture*. Ellis Horwood. 1979.
- [Foley, 1990] Foley, James D.; van Dam, Andries; Feiner, Steven K. y Hughes, John F. *Computer Graphics. Principles and Practice*. Addison-Wesley. 1990.
- [GSS, 1999] *Multi Surface Machining Library User Guide*. Geometric Software Solutions Co. Ltd. 1999.
- [Hearn, 1995] Hearn, Donald y Baker, M. Pauline. *Gráficas por Computadora*. Prentice Hall. 1995.
- [Held, 1998] Held, Martin. *Voronoi Diagrams and Offset Curves of Curvilinear Polygons*. Computer-Aided Design, vol. 30, num. 4, págs. 287-300. 1998.

- [Hoff, 1996] Hoff, Kenny. *Ray Intersection with the Unit Cylinder*, University of North Carolina at Chapel Hill. 1996.
<http://www.cs.unc.edu/~hoff/projects/comp236/raytrace2/cylinder.html>
- [IMT, 1992] Instituto Madrileño de Tecnología. *Introducción de la Máquina-Herramienta de Control Numérico en las PME*. 1992.
- [Jerard, 1989] Jerard, R.B., Drysdale, R.L., Hauck, K., Schaudt, B. y Magewick, J. *Sculptured Surfaces: Methods for Detecting Errors in Numerically Controlled Machining of Sculptured Surfaces*. IEEE Computer Graphics & Applications, págs. 26-39. January 1989.
- [Jerard, 1989b] Jerard, R.B., Hussaini, S.Z., Drysdale, R.L. y Schaudt, B. *Approximate Methods for Simulation and Verification of Numerically Controlled Machining Programs*. The Visual Computer, vol. 5, págs. 329-348. Springer-Verlag 1989.
- [Jimeno, 2001] Jimeno Morenilla, Antonio; García Chamizo, Juan Manuel y Salas Pérez, Faustino. *Shoe Last machining using Virtual Digitising*. International Journal of Advanced Manufacturing Technology, vol. 17, no. 10, págs. 744-750. 2001.
- [Kaneko, 1997] Kaneko, Toyohisa y Yamamoto, Yuuki. *Volume-Preserving Surface Reconstruction from Volume Data*. Proceedings of the 1997 International Conference on Image Processing (ICIP'97). IEEE. 1997.
- [Lambregts, 1996] Lambregts, C. A. H.; Delbressine, F. L. M.; de Vries, W. A. H. y van der Wolf, A. C. H. *An efficient automatic tool path generator for 2 1/2 D free-form pockets*. Computers in Industry, vol. 29, num. 3, págs. 151-157. 1996.
- [Lasheras, 1989] Lasheras, José M. *Control Numérico y Robótica*. Ed. Cedel, 1989.
- [Li, 1993] Li, Susan X. y Jerard, Robert B. *Non-isoparametric Three Axis NC Tool Path Generation for Finish Machining of Sculptured Surfaces*. Geometric Modeling for Product Realization. P. R. Wilson, M. J. Wozng and M. J. Pratt Publishers, págs. 251-265. 1993.
- [Li, 1994] Li, Susan X. y Jerard, Robert B. *5-axis machining of sculptured surfaces with a flat-end cutter*. Computer Aided Design, vol. 26, num. 3, págs. 165-178. 1994.

- [Li, 1998] Li, Y. M. y Hsu, V. Y. *Curve offsetting based on Legendre series*. Computer Aided Geometric Design, vol. 15, págs 711-720. 1998.
- [Llorens, 1999] Llorens Largo, Faraón; Molina Carmona, Rafael; Rizo Aldeguer, Ramón y Satorre Cuerda, Rosana. *Programación: Formalización, análisis y reutilización de algoritmos matemáticos*. Publicaciones de la Universidad de Alicante. 1999.
- [Lorensen, 1987] Lorensen, W.E. and Cline, H.E. *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, ACM Computer Graphics, vol 21, num. 24, págs. 163-169. 1987.
- [Mahadevan, 1997] Mahadevan, B.; Putta, L.; Sarma, S. y Shaikh, Z. *A feature free approach to tool selection and path planning in 3-axis rough cutting*. Proceedings of First International Conference on Responsive Manufacturing, págs. 47-60, Nottingham. September 1997.
- [McCormack, 1993] McCormack, Jon. *Interactive Evolution of L-System Grammars for Computer Graphics Modelling*. Complex Systems: from Biology to Computation. Capítulo 21, págs. 118-130. IOSPress, Amsterdam. 1993.
- [Mock, 1998] Mock, Kenrick J. *Wildwood: The Evolution of L-System Plants for Virtual Environments*. Proceedings of the 1998 IEEE World Congress on Computational Intelligence, págs. 476-480. 1998.
- [Molina, 1997] Molina Carmona, Rafael; Puchol García, Juan A. y Salas Pérez, Faustino. *Definición y visualización de superficies recortadas*. Actas del IX Congreso Internacional de Ingeniería Gráfica, págs. 465-474. 1997.
- [O'Rourke, 1993] O'Rourke, Joseph. *Computational Geometry in C*. Cambridge University Press. 1993.
- [Payne, 1990] Payne, Bradley A. y Toga, Arthur W. *Surface mapping brain function on 3D models*. IEEE Computer Graphics and Applications, págs. 33-41. September 1990.
- [Puchol, 1997] Puchol García, Juan A.; Molina Carmona, Rafael y García Quintana, Candelaria. *Conversión paramétrica de Blobs a modelo de fronteras en tiempo real*. Actas del IX Congreso Internacional de Ingeniería Gráfica, págs. 465-474. 1997.
- [Puchol, 1999] Puchol García, Juan A.; Sáez Martínez, Juan M. y Molina Carmona, Rafael. *Postproceso de rectificación de metapolígonos*. Actas del IX Congreso Español de Informática Gráfica, CEIG99, págs. 331-342. 1999.

- [Reeves, 1983] Reeves, William T. *Particle Systems – A Technique for Modelling a Class of Fuzzy Objects*. Computer Graphics, vol. 17, num. 3, págs. 359-376. 1983.
- [Requicha, 1992] Requicha, A. A. G. y Rossignac, J.R. *Solid Modelling and beyond*. IEEE Computer Graphics & Applications, págs. 31-44. September 1992.
- [Sáez, 1999] Sáez Martínez, Juan M.; Molina Carmona, Rafael y Puchol García, Juan A. *Poligonalización de blobs con rectificación de precisión*. Actas del IX Congreso Español de Informática Gráfica, CEIG99, págs. 19-32. 1999.
- [Sáez, 2000] Sáez Martínez, Juan M.; Jimeno Morenilla, Antonio M.; Puchol García, Juan A. y Molina Carmona, Rafael. *Adaptación del PRM sobre superficies libres. Aplicación a entornos CAD/CAM*. Actas del X Congreso Español de Informática Gráfica, CEIG00. 2000.
- [Sanz, 2002] Sanz Adán, Félix y Blanco Fernández, Julio. *CAD-CAM. Gráficos, Animación y Simulación por Computador*. Thomson-Paraninfo. 2002.
- [Watt, 1992] Watt, Alan y Watt, Mark. *Advanced Animation and Rendering Techniques. Theory and Practice*. Addison-Wesley. 1992.
- [Wyvill, 1986] Wyvill, G; McPheeters, C. y Wyvill, B. *Data Structure for Soft Object*. Visual Computer, vol. 2, num. 4, págs. 227-34. 1986.
- [Zorin, 2000] Zorin, Denis. *Topics in Computer Graphics Image Synthesis*. New York University, Media Research Laboratory. 2000.
<http://mrl.nyu.edu/~dzorin/rendering/>