

ANIMAÇÃO DE FLUIDOS VIA TÉCNICAS DE  
VISUALIZAÇÃO CIENTÍFICA E MECÂNICA  
COMPUTACIONAL

GILSON A. GIRALDI

ANTONIO L. APOLINÁRIO JR.

ANTONIO A. F. OLIVEIRA

RAUL A. FEIJÓO

LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA

COORDENAÇÃO DE CIÊNCIA DA COMPUTAÇÃO

FEVEREIRO - 2005

## AGRADECIMENTOS.

Agradeço ao PRONEX-LNCC (2004) e ao CNPq pelo apoio financeiro a este projeto.

# Índice

<b>ÍNDICE .....</b>	<b>I</b>
<b>CAPÍTULO 1 INTRODUÇÃO .....</b>	<b>1</b>
<b>CAPÍTULO 2 EQUAÇÕES BÁSICAS DA DINÂMICA DE FLUIDOS.....</b>	<b>5</b>
2.1. INTRODUÇÃO.....	5
2.2. LEIS DE CONSERVAÇÃO PARA CAMPOS ESCALARES.....	6
2.3. LEI DE CONSERVAÇÃO VETORIAL .....	7
2.4. EQUAÇÃO DE CONSERVAÇÃO DA MASSA .....	7
2.5. EQUAÇÃO DE CONSERVAÇÃO DO MOMENTO (EQUAÇÃO DE MOVIMENTO) .....	7
2.6. EQUAÇÃO DE VORTICIDADE .....	9
2.7. EQUAÇÃO DE CONSERVAÇÃO PARA A ENERGIA .....	10
2.8. VORTICIDADE E ROTAÇÃO .....	11
2.9. LINHAS DE VORTICIDADE, TUBOS E VÓRTICES .....	13
2.10. REFERENCIAIS EM ROTAÇÃO .....	14
2.11. VÓRTICES EM FLUIDOS BIDIMENSIONAIS .....	16
2.12. DECOMPOSIÇÃO DE HELMHOLTZ.....	18
<b>CAPÍTULO 3 TOPOLOGIA DE CAMPOS VETORIAIS .....</b>	<b>21</b>
3.1. INTRODUÇÃO.....	21
3.2. EXISTÊNCIA E UNICIDADE.....	22
3.3. DEPENDÊNCIA COM RELAÇÃO ÀS CONDIÇÕES INICIAIS E PARÂMETROS .....	23
3.4. CAMPOS VETORIAIS ESTACIONÁRIOS E FLUXOS .....	25
3.4.1 <i>Retrato de Fase de um Campo Vetorial</i> .....	25
3.4.2 <i>Equivalência e Conjugação de Campos Vetoriais</i> .....	26
3.4.3 <i>Teorema do Fluxo Tubular</i> .....	27
3.4.4 <i>Estrutura Local dos Pontos Singulares Hiperbólicos</i> .....	28
3.4.5 <i>Estrutura local das Órbitas Periódicas</i> .....	34
3.4.6 <i>Teorema de Poincaré-Bendixson</i> .....	36
<b>CAPÍTULO 4 MÉTODOS NUMÉRICOS PARA ANIMAÇÃO DE FLUIDOS .....</b>	<b>41</b>
4.1. INTRODUÇÃO.....	41
4.2. ANÁLISE NUMÉRICA .....	41
4.3. DIFERENÇAS FINITAS .....	44
4.4. ELEMENTOS FINITOS .....	46
4.5. MÉTODO DE APROXIMAÇÃO DE GALERKIN .....	49
4.6. RUNGE KUTTA .....	51
4.7. SMOOTHED PARTICLE HYDRODYNAMICS .....	54
<b>CAPÍTULO 5 TÉCNICAS PARA VISUALIZAÇÃO DE FLUIDOS.....</b>	<b>58</b>
5.1. INTRODUÇÃO.....	58
5.2. VISUALIZAÇÃO DE CAMPOS ESCALARES .....	61
5.2.1 <i>Isosuperfícies</i> .....	61
5.2.2 <i>Volume Rendering</i> .....	65
Mapeamento das Características dos Dados nos parâmetros de visualização .....	66
Termos Fonte .....	66
Termo de absorção.....	68
Termos de Espalhamento.....	69
5.3. SPLATTING .....	70
5.3.1 <i>Volume Splatting</i> .....	71
5.3.2 <i>Surface Splatting</i> .....	74

5.4.	VISUALIZAÇÃO DE CAMPOS VETORIAIS .....	81
5.4.1	<i>Integrais de Campo</i> .....	81
5.4.2	<i>Line Integral Convolution</i> .....	86
5.4.3	<i>Visualização da Topologia de Campos Vetoriais</i> .....	88
5.5.	TÉCNICAS PARA VISUALIZAÇÃO DE VÓRTICES .....	91
5.5.1	<i>Técnicas Baseadas em Isosuperfícies</i> .....	92
	Isosuperfícies de Baixa Pressão .....	92
	Autovalores do Tensor de Deformação .....	93
5.5.2	<i>Técnicas Baseadas na Identificação do Core</i> .....	94
	Algoritmo para time tracking de tubos de vorticidade em Turbulência .....	94
	Descrição do Algoritmo .....	94
	Extraindo o Esqueleto do Tubo .....	94
	Tracking: Janela Fixa ou Janela Lagrangiana .....	95
	Busca em Espiral .....	95
	Preditor-Corretor .....	96
	Pontos Sementes .....	96
	Construção do Core .....	96
	Encontrando pontos Extremos do Core .....	97
	Filamentos Conectando Vórtices .....	98
	Detalhes da Implementação para os Pontos Semente .....	99
	Detalhes dos Métodos Numéricos .....	100
	Encontrando as Seções Transversais .....	101
5.6.	TÉCNICA BASEADA EM EXTRAÇÃO DE CARACTERÍSTICAS .....	102
5.6.1	<i>Extraindo Regiões de Interesse</i> .....	102
5.6.2	<i>Atributos dos Objetos</i> .....	103
5.6.3	<i>Tracking</i> .....	104
5.6.4	<i>Interações</i> .....	104
5.6.5	<i>Visualização e Tracking</i> .....	106
<b>CAPÍTULO 6 ANIMAÇÃO COMPUTACIONAL DE FLUIDOS .....</b>		<b>108</b>
6.1.	INTRODUÇÃO .....	108
6.2.	ANIMAÇÃO VIA MODELO PARA GASES AQUECIDOS .....	109
6.2.1	<i>Modelagem do Ambiente (Cena)</i> .....	110
6.2.2	<i>Discretização das Equações</i> .....	111
6.2.3	<i>Conservação da Massa</i> .....	112
6.2.4	<i>Estabilidade Numérica</i> .....	114
6.2.5	<i>Condições de Contorno e Iniciais</i> .....	114
6.2.6	<i>Rendering</i> .....	115
6.2.7	<i>Algoritmo para Animação de Gases</i> .....	115
6.2.8	<i>Resultados</i> .....	116
6.3.	FLUIDOS ESTÁVEIS .....	117
6.3.1	<i>Esquema Numérico</i> .....	118
6.3.2	<i>Condições de Contorno Periódicas</i> .....	120
6.3.3	<i>Advecção e Difusão de Substâncias</i> .....	121
6.3.4	<i>Renderização e Resultados</i> .....	121
6.4.	FLUIDOS COMPRESSÍVEIS .....	124
6.4.1	<i>Fenômenos Incorporados</i> .....	125
6.4.2	<i>Solução Numérica</i> .....	126
6.4.3	<i>Resultados</i> .....	126
6.5.	ANIMAÇÃO VIA SPH .....	128
6.5.1	<i>Simulação</i> .....	131
6.5.2	<i>Visualização e Resultados</i> .....	132
<b>CAPÍTULO 7 PERSPECTIVAS E CONCLUSÕES .....</b>		<b>135</b>
<b>APÊNDICE A .....</b>		<b>138</b>
A.1	POLINÔMIOS DE BERNSTEIN .....	138

A.2 SPLINES .....	139
A.3 SUPERFÍCIES SPLINES .....	141
A.4 OPERADORES .....	142
A.5 DEFINIÇÕES .....	143
<b>REFERÊNCIAS .....</b>	<b>145</b>

# Capítulo 1 Introdução

Esta monografia tem como tema central a animação de fluidos via técnicas para visualização de dados científicos (Visualização Científica) empregando métodos de Mecânica Computacional.

Por “animação de fluidos” entendemos a geração de uma seqüência de imagens digitais ilustrando a evolução do fluido ao longo de um intervalo de tempo. Este conjunto de imagens deve ser capaz de reproduzir o movimento do fluido de forma convincente, ou seja, o fluido deve escoar com o grau de realismo necessário para o contexto da animação que está sendo gerada.

Dentro desta perspectiva, a animação de fluidos tem natureza multidisciplinar, e seu desenvolvimento depende da interação entre profissionais das áreas de computação e engenharia. Objetivamente, para atingir o grau de realismo necessário, é preciso que as equações de fluidos sejam convenientemente tratadas e que as técnicas de visualização de dados utilizadas possam gerar imagens com a aparência realista. As técnicas utilizadas no processo de visualização são oriundas da computação gráfica. As equações de fluidos são derivadas de conceitos da mecânica clássica. É oportuno fazer a seguir alguns comentários gerais destas áreas. Nos capítulos seguintes, estes temas serão devidamente detalhados.

A aplicação de técnicas em mecânica do continuum para a descrição matemática do escoamento de fluidos é a origem da chamada Dinâmica de Fluidos Computacional (DFC). Esta é uma área de intensa pesquisa em função de suas aplicações tecnológicas e científicas. O comportamento de estruturas mecânicas sujeitas a fortes correntes de ar ou ao movimento das marés, o estudo de fenômenos atmosféricos tais como ciclones e tornados, o estudo do comportamento do plasma no interior dos reatores de fusão, a análise do comportamento do fluxo sanguíneo no interior das artérias do corpo humano e, particularmente, a geração de efeitos visuais para indústria de entretenimento, são alguns exemplos da importância das pesquisas nesta área.

Para estudar os fenômenos em dinâmica de fluidos são utilizados princípios físicos juntamente com recursos de análise numérica [Hirsch (1988)]. Basicamente, o comportamento do fluido é modelado por um sistema de equações diferenciais parciais sujeito a certas condições iniciais e de contorno. Dificilmente este sistema apresenta solução analítica, o que obriga os especialistas a lançarem mão dos recursos da análise numérica e de métodos numéricos para discretizar o problema e obter uma solução com um nível de aproximação aceitável.

Uma simulação numérica em Dinâmica de Fluidos Computacional pode empregar malhas 3D com um nível de refinamento da ordem de  $256^3$  pontos para cada instante de tempo simulado [Samtaney et al. (1994)]. O volume de dados gerado (da ordem de dezenas de GigaBytes) é em geral constituído por campos escalares (pressão e densidade) e vetoriais (velocidade e vorticidade entre outros).

Esses campos podem ser estacionários ou não-estacionários (dependentes explicitamente do tempo ou não-dependentes, respectivamente) e a sua análise depende do desenvolvimento de técnicas específicas que possam filtrar o montante de dados realçando apenas as informações realmente relevantes. Dentre estas técnicas, as técnicas de visualização vêm se estabelecendo como um recurso de particular importância na análise dos dados. Técnicas para visualização de campos escalares (*rendering* de volume, iso-superfícies, *splatting*, etc.), campos vetoriais (linhas de corrente, superfícies de corrente, traçado de partículas, texturas, etc.) e campos tensoriais vêm sendo aplicadas para auxiliar a análise dos campos gerados nas simulações numéricas [Rosemblum et al.(1994)].

O desenvolvimento das técnicas de visualização científica em DFC, aliado aos próprios métodos de modelagem e simulação de fluidos, propiciou as pesquisas em animação computacional de fluidos.

Animação computacional é uma sub-área da computação gráfica com intensas pesquisas pelas suas aplicações na indústria de entretenimento, interação homem-computador e educação [Kerlow (1996)], [Taylor (1996)], [Watt (1991)]. Animações envolvendo modelos rígidos e articulados, tipicamente relacionados a representação de figuras humanas e de animais, têm recebido especial atenção nas últimas décadas [Foster(1997)] [Goktekin (2004)], tanto da comunidade científica quanto de empresas ligadas a cinema e televisão.

Em particular, animações onde os elementos de uma cena obedecem a princípios físicos têm despertado muito interesse da comunidade pelo nível de realismo que estes métodos podem obter. Esta área, também denominada Modelagem Baseada em Física – *Physics Based Modeling* – é multidisciplinar por natureza, uma vez que envolve tanto conhecimentos de modelagem computacional quanto de computação gráfica [Kerlow (1996)], [Taylor (1996)], [Watt (1991)] [Metaxas (1996)].

No caso particular de cenários envolvendo animações de fluidos, a utilização de métodos em Dinâmica de Fluidos Computacional (DFC) tem se mostrado um recurso valioso para os animadores, diminuindo o custo, bem como o tempo de produção de filmes, tais como *The Prince of Egypt* [Witting (1999)] ou *Shrek* [Enright (2002)].

A necessidade de se buscar métodos em DFC para auxiliar a geração de efeitos visuais envolvendo fluidos vem da dificuldade para executar tal tarefa, com o grau de realismo necessário, com métodos puramente geométricos. Fluidos são sistemas com um número muito elevado de graus de liberdade, o que torna impraticável animá-los com realismo sem fazer uso de princípios físicos para guiar o trabalho de designers experientes.

Os métodos encontrados na literatura para animação de fluidos, baseados em modelos de DFC, são fundamentados nas equações de Navier-Stokes, com técnicas de discretização baseadas em diferença finitas implícitas [Stam (1999)] e explícitas [Foster-Metaxas (1997)], bem como em métodos Lagrangeanos tais como Método das Características [Stam (1999)], *Smoothed Particle Hydrodynamics (SPH)*

[Desbrun-Gascuel (1998)] e *Moving-Particle Semi-Implicit* (MPS) [Premoze et al. (2003)].

O método *SPH* é independente de malhas (assim como o *MPS*) e vem sendo aplicado com sucesso em problemas de engenharia, tais como simulação de fluidos compressíveis, para a descrição da dinâmica de materiais, estudo de explosões e fenômenos de transporte [Liu et al. (2003)]. Implementações deste método para arquiteturas paralelas têm sido desenvolvidas com o objetivo de melhorar a performance do mesmo.

Do ponto de vista da computação gráfica, o fato do *SPH* não depender de uma malha previamente definida é uma vantagem, se comparado com os métodos clássicos. As cenas em um filme são em geral dinâmicas, com fronteiras variáveis em função de novos objetos que entram na cena à medida que esta se desenvolve. Desta forma, métodos que não fazem uso de malhas, podem ser vantajosos por evitar o custo extra de re-gerar a malha sempre que tais alterações ocorrem. Estas vantagens vêm despertando interesse por pesquisadores em computação gráfica, e propostas enfocando a utilização do *SPH* para a geração de efeitos visuais já podem ser encontradas na literatura [Müller et al. (2003)], [Amada(2004)].

As pesquisas em animação de fluidos se dividem basicamente em três etapas. Primeiramente, temos a busca de novos modelos em DFC que sejam mais eficientes do ponto de vista da computação gráfica. Esta etapa envolve tanto a pesquisa de novos modelos físicos quanto o ajuste de modelos já conhecidos, sem perder de vista o fato de que o objetivo final é a geração de efeitos visuais, e não a descrição de fenômenos naturais.

Uma vez resolvidas às equações de fluidos numericamente, passa-se à fase de *rendering*, onde técnicas de visualização são aplicadas sobre os campos gerados, com o objetivo de criar efeitos visuais, tais como transparência, imagens refletidas na superfície de um líquido, ou mesmo, efeitos especiais que incluem deformação de paisagens, incêndios, etc. Nesta etapa, tem-se usado o método clássico dos Cubos-Marchantes (*Marching Cubes*) [Lorensen (1987)], métodos dos conjuntos de níveis [Premoze et al. (2003)], *volume rendering* [Foster-Metaxas (1997)] e *LIC* [Witting (1999)].

Finalmente, as técnicas utilizadas devem ser incorporadas a um software, com interface gráfica conveniente, a qual permita a utilização destes recursos (modelos de fluidos e computação gráfica) por designers e animadores (ver [Witting (1999)], como exemplo).

Neste trabalho vamos nos concentrar nas duas primeiras etapas: Modelos de fluidos e técnicas de *rendering*. Neste sentido temos como objetivos a apresentação de modelos matemáticos de fluidos do ponto de vista da computação gráfica e a descrição de técnicas de visualização focadas para *rendering* de cenas envolvendo fluidos.

Este texto está organizado da seguinte forma. No Capítulo 2, fazemos uma revisão dos conceitos básicos em mecânica de fluidos. No capítulo 3 discutimos resultados em sistemas dinâmicos que são básicos para o estudo da topologia dos



campos gerados pelas soluções das equações de Navier-Stokes. No Capítulo 4, desenvolvemos os esquemas numéricos encontrados na literatura para resolver as equações de fluidos. A seguir, de posse da teoria necessária, discutimos as principais técnicas para visualização em DFC e sua fundamentação matemática (Capítulo 5).

No Capítulo 6 apresentamos uma revisão dos trabalhos mais significativos em animação de fluidos. O objetivo principal deste Capítulo é apresentar as linhas de pesquisa que julgamos mais significativas nesta área, sem, no entanto, fazer uma revisão exaustiva de todos os métodos. O Capítulo 7 apresenta conclusões e perspectivas futuras para esta área. A bibliografia citada encontra-se no capítulo de referências. Com o objetivo de facilitar a leitura deste trabalho, reservamos o Apêndice A para fórmulas e definições que podem não ser de uso comum ao leitor, considerando o caráter multidisciplinar da presente abordagem.

## Capítulo 2

# Equações Básicas da Dinâmica de Fluidos

### 2.1. Introdução

Um modelo matemático para o comportamento de um sistema físico, em particular para fluidos em movimento, pode ser definido somente após considerações sobre o nível de aproximação necessário para o estudo das propriedades de interesse. Esta etapa da formulação do modelo tem por objetivo a definição de um conjunto de variáveis dependentes e independentes necessária para descrever o sistema.

O primeiro nível de aproximação a ser definido é a escala em que se pretende estudar o sistema. No estudo de fluidos distinguem-se dois níveis de escalas: microscópico (átomos e moléculas) e macroscópico.

Para gases, por exemplo, a descrição no nível microscópico consiste de uma análise estatística baseada nas leis da teoria cinética e física estatística [Liboff (1990)]. Esta descrição leva à definição das variáveis em mecânica de fluidos, tais como temperatura e pressão, como sendo médias estatísticas de grandezas microscópicas.

Segundo Hirsch [Hirsch (1988)], a dinâmica de fluidos propriamente dita tem início quando a interação entre um número suficientemente alto de partículas (átomos e moléculas) afeta e domina, pelo menos parcialmente, o movimento individual de cada partícula. Neste caso, estamos na escala macroscópica onde a densidade de partículas é alta o suficiente para que possamos considerar o sistema (fluido) como um contínuo. É neste nível de aproximação que estamos interessados.

Aqui podemos distinguir duas abordagens para formular as equações do modelo: a **Lagrangeana** e a **Euleriana**. A formulação Lagrangeana da mecânica de fluidos pode ser (pelo menos teoricamente) colocada precisamente como a solução de um problema do tipo:

$$\begin{aligned} F(\dot{x}, \dot{y}, \dot{z}; \ddot{x}, \ddot{y}, \ddot{z}; t) &= 0; \\ x(t_0) &= x_0, y(t_0) = y_0, z(t_0) = z_0, \quad (2.1) \\ \dot{x}(t_0) &= \dot{x}_0, \dot{y}(t_0) = \dot{y}_0, \dot{z}(t_0) = \dot{z}_0. \end{aligned}$$

A solução deste problema é dada por uma terna da forma:

$$\left( x(x_0, y_0, z_0; \dot{x}_0, \dot{y}_0, \dot{z}_0; t), y(x_0, y_0, z_0; \dot{x}_0, \dot{y}_0, \dot{z}_0; t), z(x_0, y_0, z_0; \dot{x}_0, \dot{y}_0, \dot{z}_0; t) \right), \quad (2.2)$$

a qual representa a posição num instante  $t$  da partícula de fluido que em  $t = t_0$  satisfazia as condições iniciais dadas em (2.1).

Na escala macroscópica uma partícula de fluido é uma porção de fluido pequena se comparada com as dimensões das escalas consideradas, mas contendo um número suficientemente alto de átomos (ou moléculas) para definir sem ambigüidade médias de grandezas microscópicas (velocidade, energia, etc) das moléculas e átomos individuais.

Certamente, encontrar a função  $F$  em (2.1) não é uma tarefa simples, o que torna a formulação Lagrangeana de difícil utilização direta.

Um modo mais conveniente de formular as equações do fluido é abandonar a tentativa de especificar a história de cada partícula de fluido e, ao invés, especificar grandezas macroscópicas, tais como densidade, pressão e a velocidade do fluido, em cada ponto do espaço, a cada instante de tempo. Este método, chamado de descrição Euleriana será o método seguido na apresentação que será feita neste capítulo.

Apresentaremos aqui a formulação Euleriana via leis de conservação. Para tal, desenvolveremos inicialmente as expressões gerais destas leis para campos escalares e vetoriais para em seguida aplicá-las ao caso específico da mecânica de fluidos.

## 2.2. Leis de Conservação para Campos Escalares.

Seja  $U$  uma quantidade escalar por unidade de volume, definida num volume arbitrário  $\Omega$ , fixo no espaço e limitado por uma superfície fechada  $S$ . A intensidade local de  $U$  varia devido a atuação do fluxo que expressa a contribuição do meio externo a  $\Omega$  e das fontes  $Q$ . A forma geral da lei de conservação para a quantidade escalar  $U$  é expressa assumindo-se que a variação de  $U$  dentro do volume  $\Omega$  por unidade de tempo é dada por:

$$\frac{\partial}{\partial t} \int_{\Omega} U d\Omega, \quad (2.3)$$

e deve ser igual a contribuição do meio externo devido ao fluxo através da superfície  $S$ ,

$$- \oint_S \vec{F} \cdot d\vec{S}, \quad (2.4)$$

mais as contribuições das fontes volumétricas ( $Q_v$ ) e superficiais ( $Q_s$ ) de  $U$ , expressas por:

$$\int_{\Omega} Q_v d\Omega + \oint_S \vec{Q}_s \cdot d\vec{S}. \quad (2.5)$$

Assim, a forma geral da lei de conservação para a quantidade escalar  $U$  pode ser escrita como [Hirsch (1988)]:

$$\frac{\partial}{\partial t} \int_{\Omega} U d\Omega + \oint_S \vec{F} \cdot d\vec{S} = \int_{\Omega} Q_v d\Omega + \oint_S \vec{Q}_s \cdot d\vec{S}, \quad (2.6)$$

Supondo continuidade do fluxo e das fontes superficiais, podemos usar o Teorema de Gauss e reescrever a ultima expressão como:

$$\int_{\Omega} \frac{\partial U}{\partial t} d\Omega + \int_{\Omega} \vec{\nabla} \cdot \vec{F} = \int_{\Omega} Q_V d\Omega + \int_{\Omega} \vec{\nabla} \cdot Q_S d\Omega, \quad (2.7)$$

Esta equação é a forma integral da lei de conservação. A forma diferencial é obtida diretamente desta e tem a forma:

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = Q_V + \vec{\nabla} \cdot Q_S \Leftrightarrow \frac{\partial U}{\partial t} + \vec{\nabla} \cdot (\vec{F} - Q_S) = Q_V \quad (2.8)$$

### 2.3. Lei de Conservação Vetorial

Se a quantidade conservada é descrita por uma quantidade vetorial  $\vec{U}$  então o fluxo e o termo relativo às fontes superficiais se tornam tensores,  $\vec{\bar{F}}$  e  $\vec{\bar{Q}}$  respectivamente, o termo correspondente às fontes volumétricas se torna um vetor  $\vec{Q}_V$  e as equações (2.7)-(2.8) assumem, respectivamente, a forma [Hirsch (1988)]:

$$\int_{\Omega} \frac{\partial \vec{U}}{\partial t} d\Omega + \int_{\Omega} \vec{\nabla} \cdot \vec{\bar{F}} d\Omega = \int_{\Omega} \vec{Q}_V d\Omega + \int_{\Omega} \vec{\nabla} \cdot \vec{\bar{Q}}_S d\Omega, \quad (2.9)$$

$$\frac{\partial \vec{U}}{\partial t} + \vec{\nabla} \cdot (\vec{\bar{F}} - \vec{\bar{Q}}_S) = \vec{Q}_V, \quad (2.10)$$

### 2.4. Equação de Conservação da Massa

Vamos agora usar a lei de conservação dada por (2.7) para o caso particular em que U é a densidade volumétrica de massa ( $\rho$ ). Neste caso, a equação (2.7) toma a forma:

$$\int_{\Omega} \frac{\partial \rho}{\partial t} d\Omega + \int_{\Omega} \vec{\nabla} \cdot (\rho \vec{v}) = \int_{\Omega} Q_V d\Omega + \int_{\Omega} \vec{\nabla} \cdot Q_S d\Omega, \quad (2.11)$$

onde  $\vec{v}$  é a velocidade do fluido. Na forma diferencial, teremos a chamada equação de continuidade, dada por:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) = Q_V + \vec{\nabla} \cdot Q_S.$$

### 2.5. Equação de Conservação do Momento (Equação de movimento)

O momento linear ( $\rho \vec{v}$ ) é uma quantidade vetorial e portanto sua lei de conservação tem a forma geral dada pelas equações (2.9)-(2.10) acima. A questão central agora é determinar os tensores  $\vec{\bar{F}}$  e  $\vec{\bar{Q}}_S$  e o vetor  $\vec{Q}_V$  que aparecem nessas equações.

Pela segunda lei de Newton, sabemos que os agentes responsáveis pela variação do momento linear de um sistema físico são as forças que agem no mesmo.

No presente caso, temos uma porção do fluido limitada pela superfície  $S$ . Sobre esta região do fluido, agem forças de natureza externa ao fluido resultantes da presença de campos de força tais como gravitacional e eletromagnético. Seja  $\vec{f}_e$  a densidade volumétrica da resultante destas forças.

Além das forças externas ao sistema (fluido), temos também as forças de natureza interna decorrentes de deformações e tensões internas do mesmo. Nós assumiremos que o fluido é Newtoniano, ou seja, que as tensões internas são dadas pela expressão:

$$\overline{\overline{\sigma}} = -p\overline{\overline{I}} + \overline{\overline{\tau}}, \quad (2.12)$$

onde  $\overline{\overline{I}}$  é um tensor unitário,  $p$  é a pressão e  $\overline{\overline{\tau}}$  é o tensor de tensão cujas componentes são dadas por:

$$\tau_{ij} = \nu \left[ (\partial_i v_j + \partial_j v_i) - \frac{2}{3} (\nabla \cdot \vec{v}) \delta_{ij} \right], \quad (2.13)$$

sendo  $\nu$  é a viscosidade cinemática.

Integrando (2.12) ao longo de  $S$  teremos a resultante das forças internas ao fluido que atuam sobre o volume  $\Omega$ .

Quanto ao segundo tensor  $\overline{\overline{F}}$  este será dado por:

$$\overline{\overline{F}} = \rho \vec{v} \otimes \vec{v}, \quad (2.14)$$

onde:

$$\vec{v} \otimes \vec{v} = \begin{pmatrix} v_1 v_1 & v_1 v_2 & v_1 v_3 \\ v_2 v_1 & v_2 v_2 & v_2 v_3 \\ v_3 v_1 & v_3 v_2 & v_3 v_3 \end{pmatrix},$$

com  $\vec{v} = (v_1, v_2, v_3)$ .

Assim, a expressão (2.9) toma a forma:

$$\int_{\Omega} \frac{\partial(\rho \vec{v})}{\partial t} d\Omega + \int_{\Omega} \vec{\nabla} \cdot (\rho \vec{v} \otimes \vec{v}) d\Omega = \int_{\Omega} \rho \vec{f}_e d\Omega + \int_S \overline{\overline{\sigma}} \cdot d\vec{S}. \quad (2.15)$$

Substituindo  $\overline{\overline{\sigma}}$  pela expressão (2.12) obtemos a forma integral da lei de conservação do momento linear para um fluido Newtoniano. A expressão diferencial correspondente é obtida diretamente da forma integral e é dada por:

$$\frac{\partial}{\partial t}(\rho \bar{v}) + \bar{\nabla} \cdot (\rho \bar{v} \otimes \bar{v} + p \bar{I} - \bar{\tau}) = \rho \bar{f}_e. \quad (2.16)$$

Subtraindo do lado esquerdo desta equação a equação (2.11) na forma diferencial, multiplicada por  $\bar{v}$ , e supondo-se os termos fontes nulos, temos:

$$\rho \frac{d\bar{v}}{dt} = -\bar{\nabla} p + \bar{\nabla} \cdot (\bar{\tau}) + \rho \bar{f}_e. \quad (2.17)$$

Quando a expressão (2.13) para o tensor de tensão de um fluido Newtoniano viscoso é substituída em (2.17) obtemos as chamadas Equações de Navier-Stokes do movimento. Para coeficientes de viscosidade constantes, estas equações reduzem-se a:

$$\rho \frac{d\bar{v}}{dt} = -\bar{\nabla} p + \nu \left[ \Delta \bar{v} + \frac{1}{3} \bar{\nabla} (\bar{\nabla} \cdot \bar{v}) \right] + \rho \bar{f}_e. \quad (2.18)$$

Para fluidos incompressíveis ( $\bar{\nabla} \cdot \bar{v} = 0$ ), a equação acima reduz-se a:

$$\rho \frac{d\bar{v}}{dt} = -\bar{\nabla} p + \nu \Delta \bar{v} + \rho \bar{f}_e. \quad (2.19)$$

No caso de um fluido ideal sem tensões internas, e portanto com viscosidade  $\nu = 0$ , a equação acima reduz-se à chamada Equação de Euler:

$$\rho \frac{d\bar{v}}{dt} = \rho \frac{\partial \bar{v}}{\partial t} + \rho (\bar{v} \cdot \bar{\nabla}) \bar{v} = -\bar{\nabla} p + \rho \bar{f}_e.$$

## 2.6. Equação de Vorticidade

A equação de movimento (2.17) pode ser reescrita em varias formas, uma das quais, introduzindo-se o vetor vorticidade  $\bar{\xi}$ :

$$\bar{\xi} = \bar{\nabla} \times \bar{v}, \quad (2.20)$$

(ver A.4) e usando a identidade vetorial:

$$(\bar{v} \cdot \bar{\nabla}) \bar{v} = \bar{\nabla} \left( \frac{\bar{v} \cdot \bar{v}}{2} \right) - \bar{v} \times (\bar{\nabla} \times \bar{v}), \quad (2.21)$$

para obter:

$$\frac{\partial \bar{v}}{\partial t} - (\bar{v} \times \bar{\xi}) = -\frac{1}{\rho} \bar{\nabla} p - \bar{\nabla} \left( \frac{v^2}{2} \right) + \frac{1}{\rho} \bar{\nabla} \cdot \bar{\tau} + \bar{f}_e. \quad (2.22)$$

Uma equação importante para a vorticidade pode ser obtida tomando-se o rotacional da equação (2.22) e usando-se em seguida a identidade vetorial:

$$\bar{\nabla} \times (\bar{v} \times \bar{\xi}) = (\bar{v} \cdot \bar{\nabla}) \bar{\xi} + (\bar{\xi} \cdot \bar{\nabla}) \bar{v}$$

juntamente com o fato do rotacional do gradiente de uma função ser identicamente nulo.

Desta forma temos a equação de Helmholtz:

$$\frac{\partial \vec{\xi}}{\partial t} + (\vec{v} \cdot \vec{\nabla}) \vec{\xi} = -(\vec{\xi} \cdot \vec{\nabla}) \vec{v} + \vec{\nabla} p \times \vec{\nabla} \left( \frac{1}{\rho} \right) + \vec{\nabla} \times \left( \frac{1}{\rho} \vec{\nabla} \cdot \vec{\tau} \right) + \vec{\nabla} \times \vec{f}_e. \quad (2.23)$$

Para um fluido Newtoniano, incompressível  $\vec{\nabla} \cdot \vec{v} = 0$  com coeficiente de viscosidade cinemática constante  $\nu$ , o termo de correspondente às tensões internas reduz-se ao Laplaciano da vorticidade:

$$\vec{\nabla} \times \left( \frac{1}{\rho} \vec{\nabla} \cdot \vec{\tau} \right) = \nu \Delta \vec{\xi}, \quad (2.24)$$

onde  $\Delta$  é o Laplaciano (ver seção A.4), o que simplifica a equação final.

### 2.7. Equação de Conservação para a Energia

Em um fluido a energia total a ser considerada na equação de conservação é a soma da energia interna com a energia cinética por unidade de massa. Nós indicaremos por  $E$  a energia total por unidade de massa, a qual será dada por:

$$E = e + \frac{\|\vec{v}\|^2}{2}, \quad (2.25)$$

onde  $\|\vec{v}\|$  é o módulo do vetor velocidade e representa a energia interna.

A primeira lei da termodinâmica estabelece que os agentes responsáveis pela variação da energia total são o trabalho das forças atuando no sistema juntamente com o calor transmitido para o mesmo.

Pela lei de condução de calor de Fourier, a difusão de calor é expressa pela lei:

$$\vec{F}_D = -\kappa \vec{\nabla} T, \quad (2.26)$$

onde  $T$  é a temperatura absoluta e  $\kappa$  é o coeficiente de condutividade térmica.

As demais fontes de variação da energia interna podem ser separadas em:

- Fontes volumétricas devido a atuação das forças de volume:  $\vec{f}_e$ ,
- Outras fontes volumétricas de calor (radiação, reações químicas, etc):  $q_H$ ,
- Fontes superficiais devida tensões internas:  $\vec{Q}_s = \vec{\sigma} \cdot \vec{v}$ .

Agrupando todos estes elementos, temos a forma integral da equação de conservação da energia de um fluido dada via aplicação da lei de conservação (2.6):

$$\int_{\Omega} \frac{\partial(\rho E)}{\partial t} d\Omega + \oint_S \rho E \vec{v} \cdot d\vec{S} = \int_S \kappa \vec{\nabla} T \cdot d\vec{S} + \int_{\Omega} (\rho \vec{f}_e \cdot \vec{v} + q_H) d\Omega + \oint_S (\vec{\sigma} \cdot \vec{v}) d\vec{S}. \quad (2.27)$$

u sua versão diferencial (2.8):

$$\frac{\partial}{\partial t}(\rho E) + \vec{\nabla} \cdot (\rho \vec{v} E) = \vec{\nabla} \cdot (\kappa \vec{\nabla} T) + \vec{\nabla} \cdot (\vec{\sigma} \cdot \vec{v}) + \rho \vec{f}_e \cdot \vec{v} + q_H. \quad (2.28).$$

## 2.8. Vorticidade e Rotação

A dinâmica da vorticidade pode ser considerada um capítulo especial da dinâmica de fluidos. Fluidos com vorticidade não nula têm comportamento mais complexo e seu estudo pode ser feito a partir de elementos derivados da noção de vorticidade. Tais elementos podem ser úteis para a animação de fluidos, e por isso serão desenvolvidos nesta seção.

Primeiramente, observemos que a interpretação cinemática da vorticidade pode ser obtida pela análise do movimento relativo próximo a um ponto. A velocidade relativa  $\delta u_i, i = 1, 2, 3$ , de duas partículas de fluido separadas por uma distância  $\delta x_i$  pode ser escrita como:

$$\delta v_i = \frac{\partial v_i}{\partial x_j} \delta x_j = e_{ij} \delta x_j + \Omega_{ij} \delta x_j, \quad i, j = 1, 2, 3, \quad (2.29)$$

onde:

$$e_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right), \quad \Omega_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i} \right) e \left( e_{ij} + \Omega_{ij} \right) = \left( \frac{\partial v_i}{\partial x_j} \right) \quad (2.30)$$

são componentes dos Tensores de Tensão, Rotação e Deformação, respectivamente.

Observemos que:

$$\Omega_{ij} = -\frac{1}{2} \varepsilon_{ijk} \xi_k, \quad (2.31)$$

onde:

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{se } i, j, k \text{ formam permutação par de } 1, 2, 3, \\ -1 & \text{se } i, j, k \text{ formam permutação ímpar de } 1, 2, 3, \\ 0 & \text{cc.} \end{cases} \quad (2.32)$$

A expressão (2.31) justifica o nome comumente dado ao tensor de componentes  $\Omega_{ij}$  de tensor de vorticidade.

As duas contribuições do lado direito de (2.29) podem ser interpretadas como dois movimentos distintos. No primeiro termo nós observamos que elementos de linha são dilatados ou contraídos e esferas são deformadas em quádras com eixos



principais ao longo dos auto-vetores do tensor  $e_{ij}$ . No segundo termo observamos que os elementos de linha permanecem com comprimentos inalterados (matriz  $(\Omega_{ij})$  é anti-simétrica) e portanto este termo corresponde a um movimento de corpo rígido. Analisando esse fato em detalhe, observamos que ele nos conduz a uma distinção significativa entre fluidos onde a vorticidade é identicamente nula (irrotacionais) e fluidos com vorticidade não nula (rotacionais).

Seja o momento angular de uma porção infinitesimal de fluido com volume  $\delta V$  em torno do centróide  $x$  da região:

$$\delta A_i = \rho \sum_{j,k} \int_{\delta V} \varepsilon_{ijk} \delta x_j \delta u_k dV. \quad (2.33)$$

Substituindo (29) nesta expressão obteremos [Saffman (1995)]:

$$\delta A_i = \sum_{j,k,l} \varepsilon_{ijk} e_{kl} I_{jl} + \frac{1}{2} (\delta_{ij} I_{kh} - I_{ij}) \bar{\xi}_j, \quad (2.34)$$

onde

$$I_{ij} = \rho \int_{\delta V} \delta x_i \delta x_j dV, \quad (2.35)$$

é o tensor de inércia da porção de fluido e  $e_{kl}$  está definido em (2.30).

O segundo termo de (2.34) é o momento angular de uma região de fluido rodando como um corpo rígido com velocidade angular  $\frac{1}{2} \bar{\xi}$  [Symon (1982)].

Consideremos um corpo com simetria esférica (cubos e sólidos regulares, por exemplo). O primeiro termo de (2.34) se anula e o mesmo acontecerá com o segundo termo caso  $\bar{\xi} \equiv 0$ . Uma vez que a taxa de variação temporal do momento angular é igual ao torque total atuando sobre o corpo, fica claro que o torque total será nulo neste caso. Logo, se o corpo não tiver uma velocidade angular inicial, este permanecerá sem movimento de rotação se  $\bar{\xi} \equiv 0$ . Caso  $\bar{\xi}$  não seja identicamente nulo o segundo termo de (2.34) não se anula em geral e assim o corpo poderá iniciar movimento de rotação. Esta é a distinção básica entre os fluidos rotacionais e irrotacionais.

A expressão (2.34) fornece também a interpretação da vorticidade como sendo proporcional ao momento angular de uma porção infinitesimal de fluido cujo tensor de inércia  $I_{ij}$  em (2.35) tem simetria esférica.

## 2.9. Linhas de Vorticidade, Tubos e Vórtices

Em regiões do fluido onde a vorticidade não é identicamente nula, podemos definir as chamadas linhas de vorticidade como sendo as soluções do seguinte problema de Cauchy:

$$\frac{dx}{ds} = \vec{u}, \quad x(0) = x_0, \quad (2.36)$$

para um conjunto de posições iniciais  $x_0$ , onde  $x_0 = (x_1, x_2, x_3)$ ,  $\vec{u} = \frac{\vec{\xi}}{|\vec{\xi}|}$ , e  $s$  é o comprimento de arco. Se o campo de vorticidade for estacionário então pode-se usar o próprio tempo  $t$  como parâmetro e a própria vorticidade  $\vec{\xi}$  no lugar de  $\vec{u}$ . As condições para existência e unicidade da solução de (2.36) são estabelecidas via a diferenciabilidade dos campos em questão e serão discutidas precisamente nas seções 3.2 e 3.3.

As linhas de vorticidade que passam através de uma curva fechada simples, definem um volume chamado tubo vórtice. Este tubo, em geral, pode ser reduzido à forma cilíndrica via aplicação de uma transformação contínua.

Observemos que, pela sua definição, a vorticidade num ponto da superfície do tubo é sempre paralela a esta superfície. Com este fato em mente e a definição de vorticidade, vamos considerar o comportamento da circulação ao longo de uma seção transversal  $A_1$  do tubo a qual define uma curva fechada  $C$  sobre o mesmo. Esta grandeza (circulação) é definida por:

$$\Gamma = \oint_C \vec{v} \cdot \vec{T} ds, \quad (2.37)$$

onde  $\vec{T}$  é o vetor tangente unitário à curva  $C$ . Pelo Teorema de Stokes podemos converter (2.37) em uma integral da vorticidade sobre a área  $A_1$  dada por:

$$\Gamma = \int_{A_1} \vec{\nabla} \times \vec{v} \cdot \vec{n} ds, \quad (2.38)$$

onde  $\vec{n}$  é o vetor normal unitário sobre  $A_1$ .

Assim a circulação mede o fluxo da vorticidade através da seção transversal  $A_1$ . Sejam então duas seções transversais  $A_1$  e  $A_2$  do mesmo tubo. Consideremos o fluxo da vorticidade ao longo da superfície definida por estas seções transversais e pela superfície ( $W$ ) do tubo entre estas seções. Este fluxo é dado por:

$$\int_S \vec{\xi} \cdot \vec{n} dS = \int_{A_1} \vec{\xi} \cdot \vec{n} dS + \int_{A_2} \vec{\xi} \cdot \vec{n} dS + \int_W \vec{\xi} \cdot \vec{n}_W dS. \quad (2.39)$$

O terceiro termo desta expressão se anula pois  $\vec{\zeta}$  e  $\vec{n}_w$  (normal à superfície do tubo) são ortogonais pela definição do tubo dada acima. Pelo teorema da divergência, a integral do lado direito pode ser convertida numa integral de volume da divergência de  $\vec{\zeta}$ , a qual é identicamente nula pela definição de vorticidade dada em (2.20). Assim (2.39) é equivalente a:

$$\int_{A_1} \vec{\zeta} \cdot \vec{n} dS - \int_{A_2} \vec{\zeta} \cdot \vec{n} dS = 0, \quad (2.40)$$

o que nada mais é que a expressão de uma lei de conservação para a circulação. Esta grandeza é denominada *strength* do tubo sendo a medida mais natural de sua intensidade, o que a torna um objeto de pesquisas no campo da vorticidade e turbulência [Saffman (1995)].

Um fenômeno importante e de utilidade para os métodos de visualização é o alinhamento médio observado para as linhas de vorticidade em regiões de alta vorticidade [Constantin-Procaccia (1995)]. Fora destas regiões, este alinhamento torna-se tênue o que dificulta a utilização prática das linhas de vorticidade nestas regiões.

O termo vórtice refere-se a estruturas de natureza mais geral que os tubos. Grosseiramente, pode-se dizer que vórtices são porções do fluido que mantêm um movimento de rotação por um certo intervalo de tempo. Queremos com isto dizer que o movimento de rotação da porção de fluido é dominante sobre os outros movimentos e deformações. Pela expressão (2.31) isto implica dizer que a vorticidade nestas porções do fluido é suficientemente alta para que o tensor de Deformação em (2.30) tenha autovalores complexos.

Este argumento é usado em [Chong-Perry (1990)] como uma possível definição de vórtices. Apesar de não ser a única definição e de não haver um consenso entre os especialistas sobre como definir um vórtice uma questão fica aparente por esta argumentação: precisamos de vorticidade alta (localmente) para termos um vórtice.

Identificada uma região de alta vorticidade (um provável vórtice) a linha de vorticidade dada pela solução de (2.36) que passa pelo máximo local de vorticidade da região e que é paralela ao vetor vorticidade neste ponto é denominada *core* e tem um papel importante neste estudo. Grosseiramente, esta linha pode ser pensada como um eixo de rotação do vórtice e fornece uma boa idéia da localização do mesmo no espaço. Estes fatos tornam o *core* um elemento importante para a detecção e visualização de vórtices em dinâmica de fluidos computacional.

## 2.10. Referenciais em Rotação

Na discussão acima não consideramos a pressão. Neste caso uma heurística comumente usada na detecção de vórtices postula que: pontos do vórtice possuem alta vorticidade e baixa pressão. Nosso objetivo nesta seção é apresentar uma justificativa para esta heurística.

Sejam dois sistemas de referência  $(\hat{x}, \hat{y}, \hat{z})$  e  $(\hat{x}^*, \hat{y}^*, \hat{z}^*)$  onde o referencial \* tem origem coincidente com a origem do primeiro referencial mas não é fixo. Então, um vetor  $\vec{A}$  pode ser expresso nos dois sistemas pelas expressões:

$$\vec{A} = A_x \hat{x} + A_y \hat{y} + A_z \hat{z} = A_x^* \hat{x}^* + A_y^* \hat{y}^* + A_z^* \hat{z}^*. \quad (2.41)$$

Supondo o vetor dependente do tempo, teremos as seguintes expressões para a derivada total de  $\vec{A}$  para um observador fixo no sistema \* e para um observador fixo no sistema sem \*, respectivamente:

$$\frac{d^* \vec{A}}{dt} = \frac{dA_x^*}{dt} \hat{x}^* + \frac{dA_y^*}{dt} \hat{y}^* + \frac{dA_z^*}{dt} \hat{z}^*. \quad (2.42)$$

$$\frac{d\vec{A}}{dt} = \frac{dA_x^*}{dt} \hat{x}^* + \frac{dA_y^*}{dt} \hat{y}^* + \frac{dA_z^*}{dt} \hat{z}^* + A_x^* \frac{d\hat{x}^*}{dt} + A_y^* \frac{d\hat{y}^*}{dt} + A_z^* \frac{d\hat{z}^*}{dt}. \quad (2.43)$$

Vamos supor que o sistema \* gire com uma velocidade angular  $\vec{\omega}$  constante e que:

$$\vec{\omega} = \hat{z}^*. \quad (2.44)$$

Nestas condições pode-se mostrar que:

$$\frac{d\hat{x}^*}{dt} = \vec{\omega} \times \hat{x}^*. \quad (2.45)$$

obtendo relações análogas para as outras derivadas. Assim, a equação (2.43) pode ser reescrita na forma:

$$\frac{d\vec{A}}{dt} = \frac{d^* \vec{A}}{dt} + \vec{\omega} \times \vec{A}, \quad (2.46)$$

Se  $\vec{A} = \vec{r}$  (vetor posição) teremos:

$$\frac{d\vec{r}}{dt} = \frac{d^* \vec{r}}{dt} + \vec{\omega} \times \vec{r}, \quad (2.47)$$

Analisemos a equação de continuidade no referencial em rotação. Seja  $\rho^* = \rho^*(x^*, y^*, z^*, t)$  a densidade de massa vista do referencial \*. Então a forma diferencial da equação (2.11), desconsiderando-se os termos fontes, conduz a:

$$\frac{\partial \rho^*}{\partial t} + \vec{\nabla}^* (\rho^* \vec{v}^*) = 0, \quad (2.48)$$

onde:  $\vec{v}^* = \frac{d^* \vec{r}}{dt}. \quad (2.49)$

Por outro lado, derivando a equação (2.47) em relação a t teremos:

$$\begin{aligned} \frac{d^2 \vec{r}}{dt^2} &= \frac{d}{dt} \left( \frac{d^* \vec{r}}{dt} \right) + \frac{d}{dt} (\vec{\omega} \times \vec{r}) = \\ \frac{d^{*2} \vec{r}}{dt^2} + \vec{\omega} \times \frac{d^* \vec{r}}{dt} + \vec{\omega} \times \left( \frac{d^* \vec{r}}{dt} + \vec{\omega} \times \vec{r} \right) + \frac{d\vec{\omega}}{dt} \times \vec{r} &= \\ \frac{d^{*2} \vec{r}}{dt^2} + 2\vec{\omega} \times \frac{d^* \vec{r}}{dt} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) + \frac{d\vec{\omega}}{dt} \times \vec{r}. \end{aligned} \quad (2.50)$$

A equação (2.15) no referencial \* deve ser escrita substituindo-se o campo  $\vec{f}_e$  pela expressão da derivada  $\frac{d^{*2} \vec{r}}{dt^2}$  obtida via (2.50), tomando-se  $\frac{d\vec{\omega}}{dt} = 0$ , uma vez que no presente caso temos velocidade angular constante. Assim (2.15) toma a forma:

$$\int_{\Omega^*} \frac{\partial(\rho^* \vec{v}^*)}{\partial t} d\Omega^* + \int_{\Omega^*} \vec{\nabla}^* \cdot (\rho^* \vec{v}^* \otimes \vec{v}^*) d\Omega^* = \int_{\Omega^*} \rho^* \frac{d^{*2} \vec{r}}{dt^2} d\Omega^* + \int_{S^*} \vec{\sigma}^* \cdot d\vec{S}^*, \quad (2.51)$$

a qual, apos desenvolvermos o segundo membro, assume a forma final:

$$\begin{aligned} \frac{\partial}{\partial t} (\rho^* \vec{v}^*) + \vec{\nabla}^* \cdot (\rho^* \vec{v}^* \otimes \vec{v}^*) = \\ \rho^* \vec{f}_e - \vec{\nabla}^* \cdot (\rho^* \vec{I}^* - \vec{\tau}^*) - 2\rho^* \vec{\omega} \times \vec{v}^* - \rho^* \vec{\omega} \times (\vec{\omega} \times \vec{r}). \end{aligned} \quad (2.52)$$

Se  $\vec{v}^* = 0$ , a equação acima reduz-se a:

$$\vec{\nabla}^* p^* = \rho^* \vec{f}_e - \rho^* \vec{\omega} \times (\vec{\omega} \times \vec{r}), \quad (2.53)$$

onde o segundo termo do lado direito corresponde é a aceleração centrípeta Assim, desconsiderando a força externa  $\vec{f}_e$  nós vemos pela expressão (2.53) que o gradiente da pressão no referencial \* aponta para o exterior do vórtice o que justifica o idéia de que pontos do vórtice possuem baixa pressão.

Uma vez que a vorticidade está relacionada com velocidade angular por (2.34) fica também justificada a idéia de que os pontos do vórtice devem possuir alta vorticidade.

### 2.11. Vórtices em Fluidos Bidimensionais

Fluidos bidimensionais ocorrem, por exemplo, no estudo de cilindros imersos em fluidos viscosos. Nestes sistemas, devido a simetrias, nós podemos formular as

equações dinâmicas do fluido usando apenas duas variáveis espaciais juntamente com o tempo. Em outras palavras o fenômeno que se deseja estudar não depende da outra variável espacial. Assim, o fluido pode ser visualizado em um plano o que simplifica muito a análise. Apesar disso, quando a dinâmica do fluido se torna complexa, particularmente quando o número de vórtices se torna alto e estes começam a interagir entre si, a identificação e acompanhamento dos vórtices no tempo tornam-se um processo bem mais complexo. Portanto, o uso de fluidos 2D pode ser muito atrativo no processo de teste de viabilidade para novas metodologias de visualização.

Em [Yeung et al. (1993)] encontramos um estudo do escoamento em torno de cilindros, para um fluido bidimensional e Newtoniano, viscoso ( $\nu \neq 0$ ) e incompressível ( $\vec{\nabla} \cdot \vec{v} = 0$ ). O sistema está sujeito ao campo gravitacional constante suposto da forma  $\vec{g} = g\vec{k}$  onde  $\vec{k}$  é um vetor unitário normal ao escoamento. As equações dinâmicas para o campo de velocidades são obtidas a partir da equação (2.18) e são dadas por:

$$\rho \left( \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \vec{\nabla}) \vec{v} \right) = -\vec{\nabla} p + \nu \Delta \vec{v} + \rho g \vec{k}. \quad (2.61)$$

$$\vec{\nabla} \cdot \vec{v} = 0, \quad (2.62)$$

onde a densidade  $\rho$  é suposta constante.

As equações para o campo de vorticidade correspondente podem ser obtidas diretamente a partir da equação (2.23) e (2.24) e da observação que para fluidos 2D tem-se:

$$(\vec{\xi} \cdot \vec{\nabla}) \vec{v} \equiv 0. \quad (2.63)$$

Obtemos assim a seguinte equação para vorticidade do fluido:

$$\frac{\partial \vec{\xi}}{\partial t} + (\vec{v} \cdot \vec{\nabla}) \vec{\xi} = \nu \Delta \vec{\xi}. \quad (2.64)$$

Para fluidos incompressíveis bidimensionais podemos fazer uso ainda da chamada função de corrente para simplificar a formulação. Esta função pode ser definida da forma a seguir.

Da equação (2.62) obtemos que:

$$\frac{\partial v_1}{\partial x_1} = -\frac{\partial v_2}{\partial x_2}, \quad (2.65)$$

e portanto podemos escrever:

$$v_2 = -\int \frac{\partial v_1}{\partial x_1} dx_2 = -\frac{\partial}{\partial x_1} \left( \int v_1 dx_2 \right), \quad (2.66)$$

pois estamos supondo  $v_1$  suficientemente diferenciáveis. A função  $\phi$  dada por:

$$\phi(x_1, x_2) = \int v_1 dx_2, \quad (2.67)$$

Esta função  $\phi$  é denominada função de corrente, e, pela sua definição, é tal que:

$$v_1 = \frac{\partial \phi}{\partial x_2}, \quad v_2 = -\frac{\partial \phi}{\partial x_1}, \quad (2.68)$$

As expressões (2.68) mostram uma propriedade fundamental da função de corrente: suas isolinhas são tangentes ao vetor velocidade do fluido. Isto pode ser demonstrado observando-se que o gradiente da função de corrente é ortogonal ao vetor velocidade dado por (2.68). Esta propriedade da função de corrente é básica para a técnica apresentada em [Knight-Mallinson (1996)] a qual faz uso de funções de corrente para escoamentos estacionários 3D.

Podemos agora escrever a vorticidade em função de  $\phi$ . Para tal basta tomar a definição (2.20) e usar (2.68) para obter-se:

$$\Delta \phi = -\xi_3. \quad (2.69)$$

A solução do problema em questão se resume a resolver as equações (2.64) e (2.69) sob condições iniciais e de contorno convenientes.

## 2.12. Decomposição de Helmholtz

Neste caso, procuramos decompor o campo de velocidades em duas partes:

$$\vec{v} = \vec{v}^{(\xi)} + \vec{v}^{(\phi)}, \quad (2.70)$$

onde  $\vec{v}^{(\xi)}$  é a componente rotacional, da qual a vorticidade é derivada, e  $\vec{v}^{(\phi)}$  é a componente irrotacional. Assim teremos [Chorin-Marsden (1990)]:

$$\vec{\nabla} \times \vec{v}^{(\xi)} = \vec{\xi}, \quad (2.71)$$

$$\vec{\nabla} \times \vec{v}^{(\phi)} = 0. \quad (2.72)$$

Um fluido que satisfaz a equação (2.72) tem a propriedade de que existe uma função (potencial)  $\phi$  tal que:

$$\vec{v}^{(\phi)} = \vec{\nabla} \phi. \quad (2.73)$$

Observemos que a decomposição dada pela expressão (2.70) não é única, pois qualquer campo vetorial da forma  $\vec{\nabla} f$ , onde  $f$  é uma função escalar suave, fornece uma nova decomposição da forma (2.70), dada por:

$$\vec{v} = \left( \vec{v}^{(\xi)} - \vec{\nabla} f \right) + \left( \vec{v}^{(\phi)} + \vec{\nabla} f \right), \quad (2.74)$$

pois as expressões (2.71)-(2.72) continuam satisfeitas.

Uma forma de diminuir a quantidade de escolhas para a decomposição é impor a condição adicional:

$$\vec{\nabla} \cdot \vec{v}^{(\xi)} = 0, \quad (2.75)$$

Usando agora a equação (2.70), observamos que a condição (2.75) implica:

$$\vec{\nabla} \cdot \vec{v} = \vec{\nabla} \cdot \vec{v}^{(\xi)} + \vec{\nabla} \cdot \vec{v}^{(\phi)} = \vec{\nabla} \cdot \vec{v}^{(\phi)}, \quad (2.76)$$

Usando agora a propriedade (2.73) obtemos:

$$\vec{\nabla} \cdot \vec{v} = \Delta \phi, \quad (2.77)$$

Uma vez que conhecemos o campo  $\vec{v}$ , podemos calcular o lado direito desta equação de Poisson e, então, calcular o potencial  $\phi$  pela expressão [Chorin-Marsden (1990)]:

$$\phi(x) = \Phi(x) - \frac{1}{4\pi} \int \frac{\vec{\nabla} \cdot \vec{v}(x')}{\|x - x'\|} dx', \quad (2.78)$$

onde  $\Phi$  (função harmônica) é a solução da equação de Laplace associada  $\Delta \phi = 0$ . Esta função pode depender das condições de contorno impostas no problema. Assim, a componente irrotacional  $\vec{v}^{(\phi)}$  pode ser calculada pela expressão:

$$\vec{v}^{(\phi)}(x) = \vec{\nabla} \Phi(x) + \frac{1}{4\pi} \int \frac{(x - x') \Delta \phi(x')}{\|x - x'\|^3} dx'. \quad (2.79)$$

Com relação à componente rotacional do campo  $\vec{v}$ , vamos fazer uso da propriedade segundo a qual qualquer campo que satisfaça (2.75) admite um potencial vetor, que denotaremos por  $\vec{B}$ ; ou seja:

$$\vec{\nabla} \times \vec{B} = \vec{v}^{(\omega)}. \quad (2.79)$$

Relembrando agora a identidade vetorial:

$$\Delta \vec{B} = -\vec{\nabla} \times (\vec{\nabla} \times \vec{B}) + \vec{\nabla} (\vec{\nabla} \cdot \vec{B}), \quad (2.80)$$

a qual, impondo-se a condição:

$$\vec{\nabla} \cdot \vec{B} = 0, \quad (2.81)$$

pode ser simplificada para:

$$\Delta \vec{B} = -\vec{\xi}, \quad (2.82)$$

onde usamos a definição de vorticidade dada por (2.20) e as expressões (2.71)-(2.72). Pode-se demonstrar que a solução de (2.81) é [Chorin-Marsden (1990)], [Butkov (1968)]:



$$\vec{B} = \frac{1}{4\pi} \int \frac{\vec{\xi}(x')}{\|x - x'\|} dx'. \quad (2.83)$$

Isto que nos leva à Lei de Biot-Savat [Chorin-Marsden (1990)], bem conhecida nos textos de teoria eletromagnética, segundo a qual, a componente rotacional  $\vec{v}^{(\xi)}(x)$  pode ser calculada pela equação:

$$\vec{v}^{(\xi)}(x) = -\frac{1}{4\pi} \int \frac{(x - x') \times \vec{\xi}(x')}{\|x - x'\|^3} dx'. \quad (2.84)$$

Esta expressão tem implicitamente a condição  $\vec{\nabla} \cdot \vec{B} = 0$ . Assim, uma vez conhecida a vorticidade, encontra-se a componente rotacional  $\vec{v}^{(\xi)}(x)$  pela expressão acima. As expressões (2.79) e (2.82), juntamente com as condições (2.75) e (2.81), completam a decomposição proposta em (2.70).

## Capítulo 3

### Topologia de Campos Vetoriais

#### 3.1. Introdução

As equações (2.18) e (2.23) envolvem campos vetoriais (velocidade e vorticidade) que, são dependentes da posição no espaço e podem depender explicitamente do tempo (campos não-estacionários) ou não (campos estacionários). Uma vez resolvidas estas equações para certas condições iniciais e de contorno, os campos obtidos podem ser vistos como uma função:

$$X: I \times \Delta \rightarrow \mathfrak{R}^3, \quad (3.1)$$

onde  $I$  é um intervalo aberto da reta (eixo do tempo) e  $\Delta$  um subconjunto aberto de  $\mathfrak{R}^3$ .

Uma forma de obtermos informações quantitativas e qualitativas do campo  $X$  é estudando as soluções da equação diferencial:

$$\frac{dx}{dt} = X(t, x). \quad (3.2)$$

Num certo sentido, este ponto de vista retoma a idéia básica da formulação Lagrangeana (seção 2.1): acompanhar o movimento de partículas de fluido ao longo do tempo.

Uma partícula de fluido (ou ponto material) seria uma porção do fluido deslocando-se numa trajetória dada por uma solução da equação (3.2) [Saffman (1995)]. Mais precisamente, o fluido seria descrito pelas soluções do problema de Cauchy:

$$\frac{dx}{dt} = X(t, x). \quad x(t_0) = x_0, \quad (3.3)$$

para um conjunto conveniente de posições iniciais  $x_0$ .

Este ponto de vista é muito interessante para a visualização científica pois os métodos para integrar (3.3) são bem estabelecidos e a teoria analítica correspondente (sistemas dinâmicos) possui resultados de natureza geométrica, e portanto, com um apelo visual natural.

Assim, nas próximas duas seções apresentaremos resultados básicos referentes ao problema (3.3). Nas seções seguintes analisaremos (3.3) para campos estacionários chegando ao teorema de Hartman (que caracteriza localmente a estrutura de pontos singulares) e ao teorema de Poincaré-Bendixson que, de certa maneira, formaliza as idéias básicas inerentes aos métodos de visualização de campos vetoriais baseados em conceitos topológicos.

### 3.2. Existência e Unicidade

Uma vez que os campos envolvidos nas equações (2.18) e (2.23) são supostos contínuos e com  $D_2 X$  também contínua (Apêndice A, seção A.4), apresentaremos estas questões nas suas versões mais fortes. Em [Sotomayor (1979)] encontramos as demonstrações destes resultados e de suas versões mais fracas.

Durante toda a exposição que segue, estaremos identificando o espaço euclidiano  $n$ -dimensional com o espaço  $\mathfrak{R}^n$  e utilizando o espaço  $\mathfrak{R} \times \mathfrak{R}^n$  com norma dada por:

$$\|(t, x)\| = \max\{|t|, |x|\}, \quad (3.4)$$

onde  $|x| = (x_1^2 + x_2^2 + \dots + x_n^2)^{1/2}$ .

Em geral, não vamos nos restringir ao caso  $n=3$  (ou  $n=2$ ) uma vez que tal restrição não simplifica a exposição dos resultados mais importantes.

**Teorema 1:**

Seja  $\Omega$  um aberto de  $\mathfrak{R} \times \mathfrak{R}^n$  e seja  $X: \Omega \rightarrow \mathfrak{R}^n$  contínuo com  $D_2 X$  contínua também. Para todo ponto  $(t_0, x_0)$  em  $\Omega$  existe uma vizinhança  $V = I(t_0) \times B(t_0)$  tal que o problema de Cauchy (3.3) tem uma única solução  $\varphi$  (denominada trajetória ou curva integral de  $X$ ) em  $I(t_0)$ . Além disso, o gráfico desta solução, isto é,  $\{(t, \varphi(t)); t \in I(t_0)\}$ , está contido em  $V$ .

**Demonstração:** [Sotomayor (1979)].

Este resultado, aqui colocado como um teorema, é na verdade um corolário do teorema de Picard. Do ponto de vista da visualização, este resultado é importante pois do contrário a aplicação de um método numérico para resolver (3.3) poderia conduzir a resultados sem qualquer sentido. Do ponto de vista físico, o teorema acima é uma necessidade uma vez que um modelo físico na mecânica clássica, sujeito a condições iniciais realistas, deverá ter uma e somente uma possibilidade de evolução no tempo.

Se tomarmos um ponto  $(t_1, x_1)$  pertencente ao gráfico da solução acima, poderíamos, pelas condições de continuidade e diferenciabilidade do campo  $X$ , aplicar o teorema acima novamente, e assim estender a solução o máximo possível. Tal possibilidade dá origem à idéia de solução máxima; ou seja, uma solução  $\varphi$  definida num intervalo  $I$  tal que se  $\psi$  é uma outra solução num intervalo  $J$  com  $J \supseteq I$  e  $\varphi = \psi|_I$ , então  $I=J$ . O intervalo  $I$  é chamado intervalo máximo e seus extremos denotados por  $w_+$  e  $w_-$ .

A unicidade da solução máxima para o problema (3.3) é consequência direta do Teorema 1. Este é mais um fato que fornece consistência à utilidade prática de (3.3).

### 3.3. Dependência com Relação às Condições Iniciais e Parâmetros

Na descrição do comportamento do escoamento via o problema de Cauchy (3.3) precisamos resolver a equação diferencial para um conjunto conveniente de condições iniciais. Por exemplo, poderíamos estudar a evolução de uma linha material (uma curva composta por partículas de fluido) ao longo do tempo. Novamente, devemos nos perguntar sob que condições tal problema admite solução e quais as características desta solução. Os teoremas a seguir são fundamentais para esta resposta.

Teorema 2:

Seja  $X$  contínua num aberto  $\Omega \subset \mathfrak{R} \times \mathfrak{R}^n \times \Lambda$  onde  $\Lambda$  é o intervalo de variação de um parâmetro  $\lambda$ . Para todo  $(t_0, x_0, \lambda) \in \Omega$  suponha que o problema de valores iniciais, com  $\lambda$  fixo,

$$\frac{dx}{dt} = X(t, x, \lambda). \quad x(t_0) = x_0, \quad (3.5)$$

tenha uma única solução  $\varphi$  definida no seu intervalo máximo  $(w_-, w_+)$ ,  $w_{\pm} = w_{\pm}(t_0, x_0, \lambda)$ .

Então:

$$D = \left\{ (t, t_0, x_0, \lambda); (t_0, x_0, \lambda) \in \Omega, t \in (w_-, w_+) \right\}$$

é aberto em  $\mathfrak{R} \times \Omega$  e  $\varphi$  é contínua em  $D$ .

**Demonstração:** [Sotomayor (1979)].

Para mostrar uma consequência prática deste teorema retomemos a questão colocada acima de acompanhar o movimento de uma curva material no interior do fluido. Esta curva, no instante  $t = t_0$  poder ser escrita como:

$$c(t_0, s) = x_0(s), \text{ onde } s \text{ é um parâmetro.}$$

O problema em questão pode ser colocado na forma:

$$\frac{dx}{dt} = X(t, x). \quad x(t_0) = x_0(s). \quad (3.6)$$

Observemos que, de acordo com o Teorema 1, para cada valor de  $s$  este problema tem solução máxima única e portanto podemos aplicar o teorema acima para garantir que a curva inicial  $c(t_0, s)$  será deslocada continuamente pela ação do campo (escoamento). A solução  $c(t, s)$  do problema (3.6) é comumente denominada

linha de tempo na literatura [Rosemblum et al. (1994)]. Demais possibilidades inspiradas na idéia de variação de parâmetros presente no Teorema 2 acima são apresentadas a seguir e admitem uma análise análoga.

Linhas de Corrente (Streamlines):

Funções  $\varphi(s, s_0, x_0)$  soluções de:

$$\frac{dx}{ds} = u(t_0, x), \quad x(s_0) = x_0, \quad (3.7)$$

onde

$$u = \begin{cases} \frac{X(t_0, x)}{|X(t_0, x)|}, & \text{se } X(t_0, x) \neq 0, \\ 0, & \text{cc.} \end{cases}$$

Linhas de Trajetória (Path Lines):

$\varphi(t, t_0, x_0)$  tais que:

$$\frac{dx}{dt} = X(t, x), \quad x(t_0) = x_0, \quad (3.8)$$

com condição inicial fixa.

Linhas de Emissão (Streaklines):

$$\frac{dx}{dt} = X(t, x), \quad x(\lambda) = x_0, \quad (3.9)$$

onde  $\lambda$  varia num intervalo  $t_0 \leq \lambda \leq t_1$ .

Na definição (3.7) para linhas de corrente, a idéia é estudar o campo congelado num instante de tempo  $t = t_0$ . Assim, as linhas obtidas não corresponderão a trajetórias de partículas materiais, mas conterão informações do comportamento do escoamento (mais especificamente, da topologia do campo) no instante  $t = t_0$ . Este é um recurso explorado pela técnica descrita em [Helman-Hesselink (1991)] para visualização de campos não-estacionários.

Devemos acrescentar também que as linhas de corrente e as linhas de trajetórias são equivalentes no caso de campos estacionários.

A diferenciabilidade da função  $\varphi$  com relação à variável  $x$  é estabelecida pelo seguinte resultado.

Teorema 3:

Seja  $X$  e  $D_2 X$  contínuas num aberto  $\Omega \subset \mathfrak{R} \times \mathfrak{R}^n \times \Lambda$  onde  $\Lambda$  é o intervalo de variação de um parâmetro  $\lambda$ . Então, para  $\lambda$  fixo a solução  $\varphi = \varphi(t, t_0, x_0, \lambda)$  de

$$\frac{dx}{dt} = X(t, x, \lambda). \quad x(t_0) = x_0, \quad (3.10)$$

é única e admite derivada parcial  $D_3\varphi$  com relação a  $x_0$ .

Além disso, a aplicação  $(t, t_0, x_0, \lambda) \rightarrow D_3\varphi(t, t_0, x_0, \lambda)$  é contínua no seu domínio dado por:

$$D = \left\{ (t, t_0, x_0, \lambda), (t_0, x_0, \lambda) \in \Omega, t \in (w_-(t_0, x_0, \lambda), w_+(t_0, x_0, \lambda)) \right\}.$$

**Demonstração:** [Sotomayor (1979)].

### 3.4. Campos Vetoriais Estacionários e Fluxos

O estudo das soluções da equação (3.2) ganha em complexidade pela dependência explícita do campo com relação ao tempo. Esta dependência poderá provocar mudanças globais na topologia das trajetórias de tal ordem a tornar inviável o estudo do campo a partir desta equação. Uma alternativa prática nestes casos é estudar a topologia do campo em vários instantes e em seguida analisar a evolução da topologia no tempo. Mais especificamente, tudo se passa como se considerássemos o campo  $X(t, x)$  como uma família de campos estacionários e estivássemos analisando o comportamento desta família via o estudo de uma seqüência discreta de elementos da mesma, por exemplo:

$$\left\{ X(t_i, x); t_i = t_0 + i\Delta t, i = 1, \dots, N \right\}.$$

Assim, os conceitos relativos a campos estacionários assumem um caráter fundamental nesta apresentação. A equação diferencial em questão será:

$$\frac{dx}{dt} = X(x), \quad (3.11)$$

Nas próximas seções passaremos a analisar os resultados relativos às soluções desta equação (dita autônoma) e do comportamento destas soluções nas vizinhanças de pontos singulares de  $X$  (pontos onde  $X$  se anula).

De início podemos garantir existência e unicidade das soluções do problema de Cauchy correspondente se  $X$  for o campo de velocidades ou de vorticidade, pelo que foi exposto na seção 3.2. A aplicação  $\varphi: D \rightarrow \mathfrak{R}^n$ , onde  $D$  é o conjunto do teorema 2 e  $\varphi$  é definida por:  $\varphi(t, x) = \varphi_x(t)$  com  $\varphi$  dada pelo mesmo teorema, é diferenciável com relação às variáveis  $t$  e  $x$ . Esta aplicação é denominada fluxo gerado por  $X$  e será utilizada a seguir.

#### 3.4.1 Retrato de Fase de um Campo Vetorial

O retrato de fase é um recurso importante para a representação gráfica das propriedades topológicas de um campo vetorial. O retrato de fase de um campo  $X$  pode ser representado por um conjunto de curvas denominadas órbitas. A órbita do campo  $X$  pelo ponto  $p$  é a imagem da curva integral de  $X$  pelo ponto  $p$ , ou seja:

$$\gamma_p = \left\{ \varphi(t, p); \quad t \in I_p \right\}.$$

Pela unicidade das curvas integrais (Teorema 1) para os campos de interesse, tem-se que duas órbitas de  $X$  coincidem ou são disjuntas. Mais ainda, pode-se demonstrar [Sotomayor (1979)] que o conjunto  $\Delta$  fica decomposto em numa união disjunta de curvas diferenciáveis, podendo cada uma ser:

- imagem biunívoca de um intervalo da reta,
- um ponto, ou
- fechada (periódica).

O conjunto  $\Delta$ , munido da decomposição em órbitas de  $X$  é o retrato de fase de  $X$ . Na representação gráfica do retrato de fase costuma-se orientar as órbitas no sentido das curvas integrais do campo  $X$  (crescimento de  $t$ ).

A grande utilidade do retrato de fase como instrumento para representação gráfica de um campo vetorial está na possibilidade de visualização das propriedades essenciais (topológicas) do mesmo, via a utilização de algumas órbitas conectando os pontos singulares do campo. Este recurso ficará mais claro com os exemplos apresentados na seção 3.4.6 onde discutimos o teorema de Poincaré-Bendixson, clássico nesta área.

### 3.4.2 Equivalência e Conjugação de Campos Vetoriais

Uma vez que dispomos de um recurso para representação de propriedades de um campo vetorial, seu retrato de fase, devemos procurar meios para comparar duas representações e dizer quando são equivalentes (possuem as mesmas propriedades essenciais). Introduziremos então as noções de equivalência e conjugação topológica entre dois campos vetoriais, as quais permitem comparar seus retratos de fase.

Equivalência:

Sejam  $X_1$  e  $X_2$  campos vetoriais definidos nos abertos  $\Delta_1$  e  $\Delta_2$  de  $\mathfrak{R}^n$ , respectivamente. Diz-se que  $X_1$  é topologicamente equivalente (respectivamente  $C^r$ -equivalente) a  $X_2$  quando existe um homeomorfismo (resp. um difeomorfismo de classe  $C^r$ , (ver A.5))  $h: \Delta_1 \rightarrow \Delta_2$  levando as órbitas de  $X_1$  em órbitas de  $X_2$  e preservando a orientação.

Conjugação:

Sejam  $\varphi_1: D_1 \rightarrow \mathfrak{R}^n$  e  $\varphi_2: D_2 \rightarrow \mathfrak{R}^n$  os fluxos gerados pelos campos  $X_1: \Delta_1 \rightarrow \mathfrak{R}^n$  e  $X_2: \Delta_2 \rightarrow \mathfrak{R}^n$  respectivamente. Diz-se que  $X_1$  é topologicamente conjugado (resp.  $C^r$ -conjugado) a  $X_2$  quando existe um homeomorfismo (resp. um difeomorfismo de classe  $C^r$  (ver A.5))  $h: \Delta_1 \rightarrow \Delta_2$  tal que  $h(\varphi_1(t, x)) = \varphi_2(t, h(x))$  para todo  $(t, x) \in D_1$ . O homeomorfismo  $h$  chama-se conjugação topológica (resp.  $C^r$ -conjugação) entre  $X_1$  e  $X_2$ .

Estas duas noções definem relações de equivalência entre campos vetoriais definidos em abertos do  $\mathfrak{R}^n$ . Além disso, toda conjugação é uma equivalência, mas o inverso não é verdadeiro, o que pode ser mostrado pelo exemplo a seguir.

Sejam os campos vetoriais  $X$  e  $Y$  em  $\mathfrak{R}^2$  dados por:

$$X(x_1, x_2) = (ax_2, -ax_1), \quad Y(x_1, x_2) = (bx_2, -bx_1), \quad (3.12)$$

Para  $a, b$  não nulos as órbitas correspondentes serão circunferências de centro na origem cujas curvas integrais têm período  $2\pi/a$  e  $2\pi/b$ , respectivamente. A aplicação identidade é uma  $C^\infty$ -equivalência entre os campos, mas estes não são topologicamente conjugados.

Um aspecto essencial destas relações é a preservação de pontos singulares e órbitas periódicas. No caso da conjugação, o período das órbitas periódicas também é preservado. São estas características que tornam estas relações úteis para a comparação de retratos de fase uma vez que os pontos singulares e órbitas periódicas são elementos essenciais na caracterização da topologia de um campo.

A importância dos pontos singulares para a caracterização da topologia pode ser medida pelo teorema que segue. Este mostra que o comportamento do fluxo na vizinhança de um ponto não-singular (regular) é topologicamente simples (“tubular”), o que de certa forma aponta para os pontos singulares como os grandes responsáveis por comportamentos topológicos mais complexos. O resultado seguinte (teorema de Hartman) fornece uma noção da extensão desta complexidade. Em seguida o mapa de Poincaré e suas conseqüências mostram a importância das órbitas periódica na definição da topologia.

### 3.4.3 Teorema do Fluxo Tubular

Sejam  $X: \Delta \rightarrow \mathfrak{R}^n$  um campo vetorial de classe  $C^r$ ,  $r \geq 1$ , com  $\Delta \subset \mathfrak{R}^n$  e aberto e  $A \subset \mathfrak{R}^{n-1}$  também aberto. Uma aplicação diferenciável  $f: A \rightarrow \Delta$  de classe  $C^r$  chama-se seção transversal local de  $X$  (de classe  $C^r$ ) quando, para cada  $a \in A$ ,  $Df(a)(\mathfrak{R}^{n-1})$  e  $X(f(a))$  geram o espaço  $\mathfrak{R}^n$ . Seja  $\Sigma = f(A)$  munido da topologia induzida. Se  $f: A \rightarrow \Delta$  for um homeomorfismo, diz-se que  $\Sigma$  é uma seção transversal de  $X$ .

**Teorema 4** (do fluxo tubular):

Seja  $p$  um ponto não singular de um campo vetorial  $X: \Delta \rightarrow \mathfrak{R}^n$  de classe  $C^r$  ( $X(p) \neq 0$ ) e  $f: A \rightarrow \Delta$  uma seção transversal local de  $X$  de classe  $C^r$  com  $f(0) = p$ . Então existe uma vizinhança  $V$  de  $p$  em  $\square$  e um difeomorfismo  $h: V \rightarrow (-\varepsilon, \varepsilon) \times B$  de classe  $C^r$ , onde  $\varepsilon > 0$  e  $B$  é uma bola aberta em  $\mathfrak{R}^{n-1}$  de centro na origem  $0 = f^{-1}(p)$  tal que:

$$h(\Sigma \cap V) = \{0\} \times B;$$



$h$  é uma  $C^r$ -conjugação entre  $X|_V$  e o campo constante  $Y:(-\varepsilon, \varepsilon) \times B \rightarrow \mathbb{R}^n$ ,  $Y = (1, 0, 0, \dots, 0) \in \mathbb{R}^n$ .

**Demonstração:** [Sotomayor (1979)].

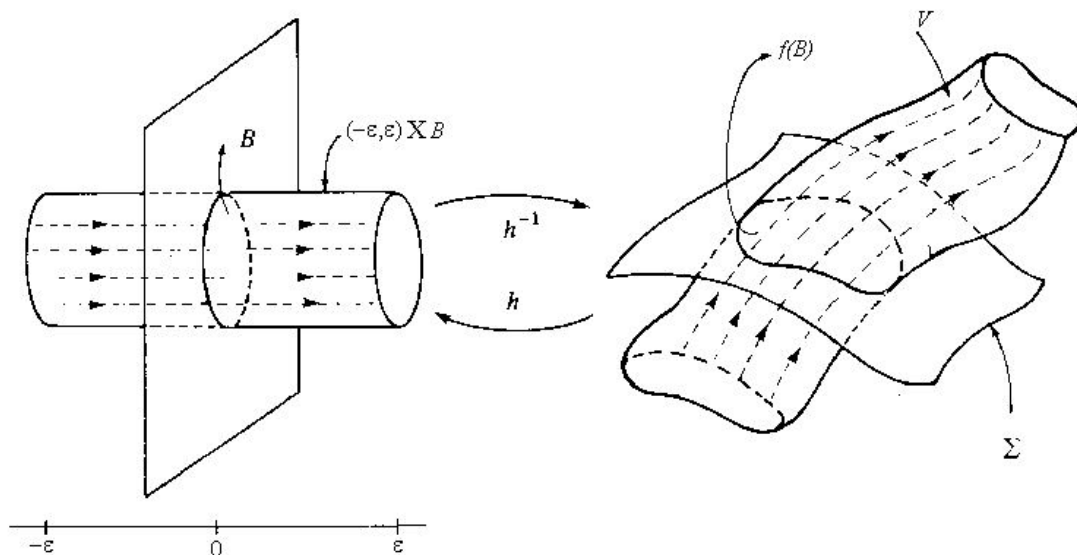


Figura 3.1 – Fluxo na vizinhança de um ponto regular.

Assim, seja  $p$  um ponto não singular de um campo vetorial  $X$  de classe  $C^r$ ,  $r \geq 1$ . Pelo teorema acima, sabemos que existe um difeomorfismo de classe  $C^r$  que conjugua  $X$  com o campo constante  $Y = (1, 0, 0, \dots, 0)$  em uma vizinhança de  $p$ . Do ponto de vista topológico este conhecimento é satisfatório pois deixa claro a existência de apenas uma classe de conjugação diferenciável (a conjugação é uma relação de equivalência) local em torno de pontos regulares.

Se  $p$  é um ponto singular a situação é bem mais complexa e será tratada no próximo item.

### 3.4.4 Estrutura Local dos Pontos Singulares Hiperbólicos

Seja  $p$  um ponto singular de um campo vetorial  $X$  de classe  $C^r$  ( $X(p) = 0$ ). Os autovalores da matriz  $DX(p)$  são fundamentais para a caracterização do comportamento do campo na vizinhança de  $p$ . Particularmente, se os autovalores de  $DX(p)$  têm parte real diferente de zero ( $p$  é ponto *singular hiperbólico*) o teorema a seguir permitirá estabelecer as diferentes classes de conjugação topológica para o campo  $X$  na vizinhança de  $p$ .

**Teorema 5** (de Hartman):

Sejam  $X : \Delta \rightarrow \mathfrak{R}^n$  um campo vetorial de classe  $C^1$  e  $p$  um ponto singular hiperbólico. Existem vizinhanças  $V$  de  $p$  em  $\Delta$  e  $W$  de  $0$  em  $\mathfrak{R}^n$  tais que  $X|_V$  é topologicamente conjugado a  $DX(p)|_W$ .

**Demonstração:** [Sotomayor (1979)].

Assim, classificar os pontos singulares hiperbólicos estabelecendo as diferentes classes de conjugação remete ao problema de classificação topológica dos sistemas lineares hiperbólicos, ou seja, sistemas do tipo:

$$\frac{dx}{dt} = Ax$$

onde  $A$  é uma matriz constante com autovalores com parte real diferente de zero.

O número  $s=s(A)$  de autovalores de  $A$ , contando suas multiplicidades, que têm parte real negativa, chama-se índice de estabilidade do sistema (3.13). Notemos que esta definição depende apenas da classe de similaridade da matriz  $A$ , ou equivalentemente, da classe de conjugação linear do sistema (3.13).

Chama-se subespaço estável de (3.13) o subespaço maximal  $E^s$ , invariante por  $A$  (isto é,  $Av \in E^s$  se  $v \in E^s$ ) tal que  $A|_{E^s}$  tem todos seus autovalores com parte real negativa. Analogamente, define-se o subespaço instável de (3.13) como o subespaço maximal e invariante  $E^u$  onde  $A|_{E^u}$  tem todos seus valores próprios com parte real positiva.

Estamos agora em condições de apresentar dois teoremas fundamentais neste contexto.

**Teorema 6:**

Seja um sistema linear hiperbólico da forma (3.13) com índice de estabilidade  $\beta$ .

$\varphi(t) = \varphi(t, p)$  e  $t \geq 0$  são invariantes pelo sistema, isto é, para todo  $x \in E^i$ ,  $i = s, u$ , a trajetória do sistema,  $\gamma^+(p)$ , pertence a  $K \subset \Delta$  para todo  $\omega(p)$ . A dimensão de  $\omega(p)$  é igual a  $\beta$ .

Existem  $\omega(p)$  e  $\omega(p)$  tais que:

$\omega(p)$  para  $t \rightarrow \pm\infty$  e  $\omega(p)$

$\omega(p)$  para  $x \in E^u$  e  $t \leq 0$ .

**Demonstração:** [Sotomayor (1979)].

**Teorema 7:**

Dois sistemas lineares hiperbólicos do tipo (3.13) em  $\mathfrak{R}^n$  são topologicamente conjugados se e somente se ambos têm o mesmo índice de estabilidade.

**Demonstração:** [Sotomayor (1979)].

Pelo segundo teorema vemos que cada classe de conjugação dos sistemas lineares hiperbólicos em  $\mathfrak{R}^n$  é caracterizada por um índice de estabilidade. No caso tridimensional, que nos interessa mais de perto, os possíveis comportamentos desses sistemas podem ser caracterizados como na Figura 3.2.

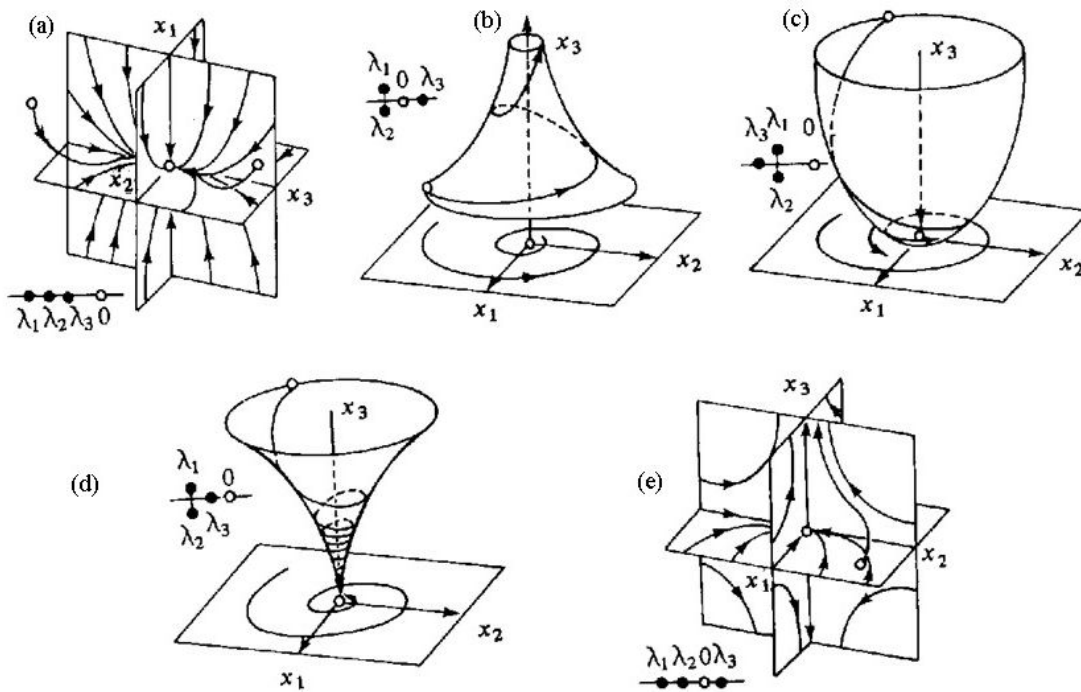


Figura 3.2 – Sistemas lineares hiperbólicos no  $\mathfrak{R}^3$ .

Seja, por exemplo o caso (b) representado na Figura 3.2 onde temos dois autovalores complexos ( $\omega(\gamma)$  e  $\overline{\omega(\gamma)}$ ) ambos com parte real negativa e um autovalor real e positivo ( $X: \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$ ). O primeiro item do penúltimo teorema implica que  $C^1$  e  $\omega(p)$  são subespaços de dimensão um e dois respectivamente. Assim, se  $x \in E^s$  (plano  $x_1, x_2$ ) então a trajetória passando por  $x$  pertence também a  $E^s$  o que está representado pela espiral da Figura 3.2. Por outro lado, se  $x \in E^u$  (eixo  $x_3$ ) a trajetória será o próprio eixo. A trajetória no espaço representada na Figura 3.2 procura mostrar o fato de que sua projeção sobre o plano  $x_1, x_2$  caminha em espiral para o ponto singular enquanto que sua projeção sobre o eixo  $x_3$  se afasta deste ponto à medida que  $t$  cresce.

A Figura 3.2 também é interessante por indicar como a topologia do fluxo vai mudando com a mudança do conjunto de autovalores. Em [Chong-Perry (1990)] encontramos uma caracterização detalhada destas mudanças via o estudo da equação de autovalores da matriz  $A$ , mais precisamente, dos coeficientes desta equação (invariantes por similaridade de matrizes). A proposta desta última referencia é oferecer recursos para a interpretação de escoamentos complexos em termos de padrões elementares de fluxo dados pelo comportamento do fluido na vizinhança de pontos singulares. Vejamos alguns detalhes deste trabalho.

Seja a equação de autovalores da matriz  $A$  do sistema (3.13) dada por

$$\det[A - \lambda I] = 0. \quad (3.14)$$

Para o caso tridimensional, esta equação pode ser expandida e colocada na forma:

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0, \quad (3.15)$$

onde:

$$P = -(a_{11} + a_{22} + a_{33}) = -\text{tr}[A], \quad (3.16)$$

$$Q = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}, \quad (3.17)$$

$$R = -\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = -\det[A], \quad (3.18)$$

A equação característica pode ter (i) todas as raízes reais e distintas, (ii) todas as raízes reais com pelo menos duas raízes iguais ou (iii) uma raiz real com um par de raízes complexas conjugadas. Os coeficientes da equação (3.15) são invariantes por transformações de similaridade entre matrizes e por isso a caracterização das possibilidades (i)-(iii) via estes coeficientes não possuirá ambigüidades.

A equação (3.15) é uma equação cúbica em  $\lambda$ . Uma vez que seus coeficientes são reais, pode-se mostrar que o discriminante:

$$D = W^3 + Y^2, \quad (3.19)$$

é tal que a equação (3.15) terá [Spiegel (1973)]:

uma raiz real e duas complexas conjugadas se  $D > 0$ .

todas as raízes reais e no mínimo duas iguais se  $D = 0$ .

todas as raízes reais e distintas se  $D < 0$ .

Com uma manipulação algébrica simples pode-se mostrar que a equação  $D = 0$  é equivalente a:

$$27R^2 + (4P^3 - 18PQ)R + (4Q^3 - P^2Q^2) = 0. \quad (3.20)$$

o que define uma superfície no espaço  $P$ - $Q$ - $R$  que denominaremos superfície  $S_1$ .

A Figura 3.3 mostra a interseção desta superfície com os planos  $P=\text{constante}$ .

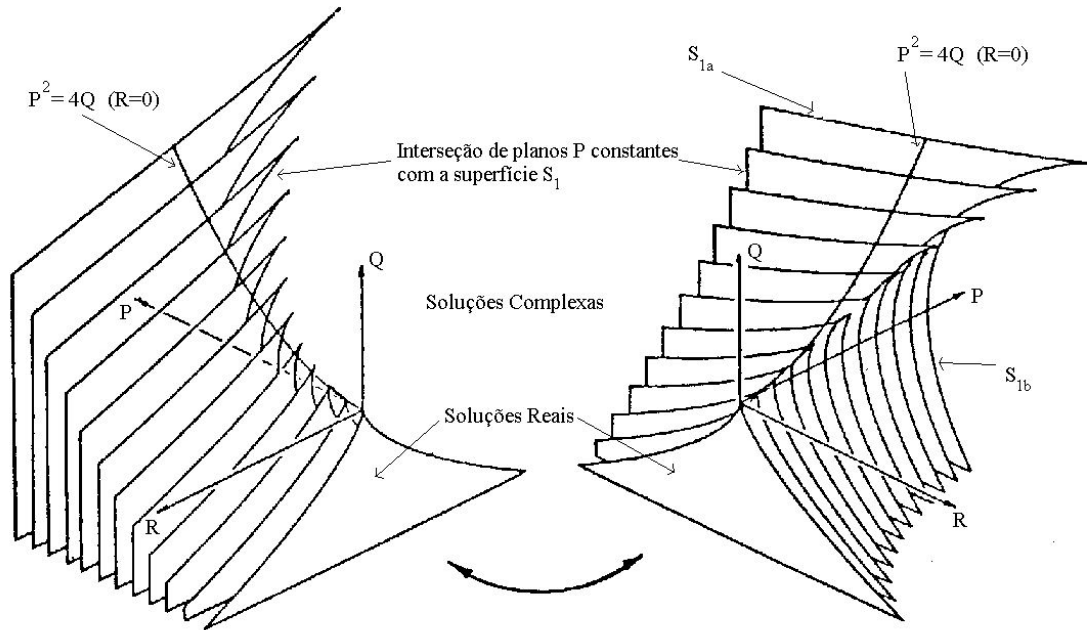


Figura 3.3 – interseções de  $S_1$  com planos  $P=\text{const}$ . No espaço  $P$ - $Q$ - $R$ .

De acordo com (b) acima, sobre  $S_1$  os autovalores são reais e pelo menos dois deles são iguais. Resolvendo (3.20) para  $R$  obteremos as superfícies  $S_{1a}$  e  $S_{1b}$  definidas, respectivamente, pelas equações:

$$\frac{1}{3}P\left(Q - \frac{2}{9}P^2\right) - \frac{2}{27}(-3Q + P^2)^{3/2} - R = 0, \quad (3.21)$$

$$\frac{1}{3}P\left(Q - \frac{2}{9}P^2\right) + \frac{2}{27}(-3Q + P^2)^{3/2} - R = 0. \quad (3.22)$$

A Figura 3.4 mostra um perfil típico do corte de  $S_1$  por um plano  $P$  constante, indicando os cortes de  $S_{1a}$  e  $S_{1b}$  correspondentes a (3.21) e (3.22) acima.

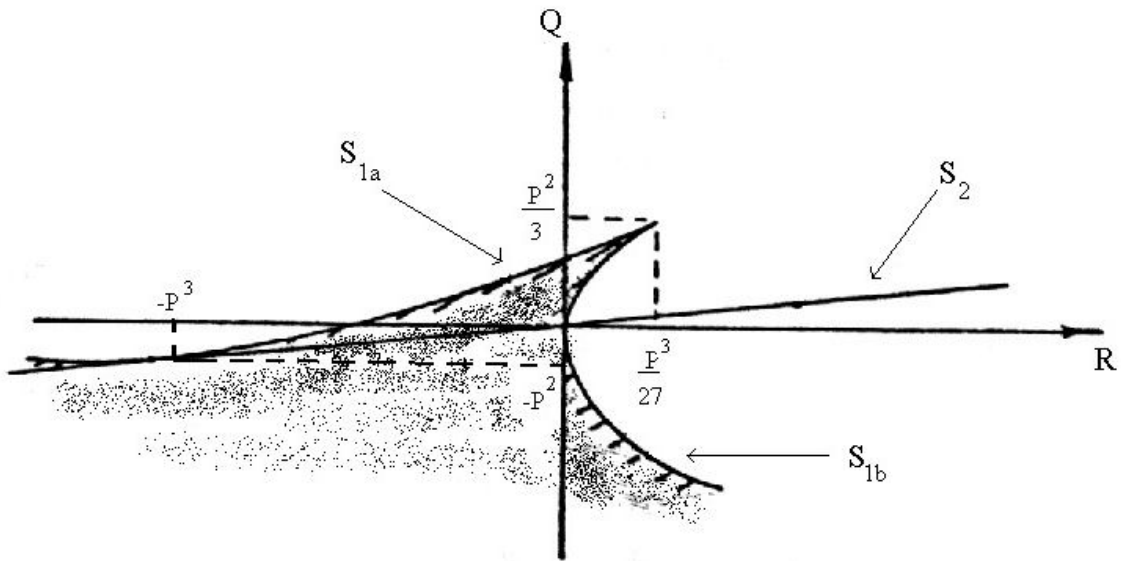


Figura 3.4 – Perfil dos cortes de  $S_1$  por  $P=\text{const.}$  no plano  $QR$ .

De (3.21) e (3.22) vemos que o discriminante  $D$  pode ser fatorado na forma:

$$D = (R - R_a)(R - R_b), \quad (3.23)$$

onde:

$$R_a = \frac{1}{3}P\left(Q - \frac{2}{9}P^2\right) - \frac{2}{27}(-3Q + P^2)^{3/2}, \quad (3.24)$$

$$R_b = \frac{1}{3}P\left(Q - \frac{2}{9}P^2\right) + \frac{2}{27}(-3Q + P^2)^{3/2}. \quad (3.25)$$

Portanto, de acordo com (a)-(c) acima e (3.23)-(3.25) teremos:

uma raiz real e duas complexas conjugadas se:

$$Q > P^2/3, \text{ ou}$$

$$Q < P^2/3 \text{ e } R < R_a \text{ ou}$$

$$R > R_b. \quad (3.26)$$

todas as raízes reais e no mínimo duas iguais se:

$$R = R_a \text{ ou } R = R_b \text{ e } Q < P^2/3, \quad (3.27) \text{ ou}$$

$$R = P^3/27 \text{ e } Q = P^2/3, \quad (3.28) \text{ (três raízes reais e iguais).}$$

todas as raízes reais e distintas se:

$$Q < P^2/3 \text{ e } R_a < R < R_b, \quad (3.29) \text{ (região destacada na Figura 3.4).}$$

Se os autovalores complexos são puramente imaginários (sistema não-hiperbólico), pode-se mostrar que estes devem satisfazer à equação:

$$PQ - R = 0, \quad (3.30)$$

que define a superfície  $S_2$ , cujo corte por P constante é uma reta no plano QR (Figura 3.4).

Quando p é um ponto singular não-hiperbólico temos que lançar mão de recursos mais atuais para o estudo do comportamento local das soluções. Por exemplo, para pontos singulares parcialmente hiperbólicos (pelo menos um dos autovalores com parte real não nula) pode-se, sob certas condições, generalizar o teorema de Hartman e simplificar o campo numa vizinhança da singularidade [Takens (1970)]. Estes resultados saem do escopo clássico e sua utilidade para questões de visualização é um tema ao que sabemos ainda em aberto.

### 3.4.5 Estrutura local das Órbitas Periódicas

Como já mencionado, as órbitas periódicas também são elementos importantes para a caracterização da topologia de um campo. Uma maneira de descrevermos o comportamento de um campo em uma vizinhança de uma órbita periódica é através da transformação de Poincaré a qual definiremos a seguir.

Seja  $\gamma = \{\varphi(t, p), 0 \leq t \leq \tau_0\}$  uma órbita periódica de período  $\tau_0$  de um campo X de classe  $C^r$ ,  $r \geq 1$ , definido em um aberto  $\Delta \subset \mathfrak{R}^n$ . Seja  $\Sigma$  uma seção transversal (seção 3.4.3 acima) de X em p. Em virtude da continuidade do fluxo  $\varphi$  de X, para todo  $q \in \Sigma$  próximo de p a trajetória  $\varphi(t, q)$  permanece próxima a  $\gamma$ , com t em um intervalo compacto pré-fixado, por exemplo,  $[0, 2\tau_0]$ . Define-se a Transformação de Poincaré  $\pi$  tal que  $\pi(q)$  seja o primeiro ponto onde esta órbita intercepta  $\Sigma$ . Seja  $\Sigma_0$  o domínio de  $\pi$ . Naturalmente  $p \in \Sigma_0$  e  $\pi(p) = p$ .

Pode-se demonstrar que  $\pi$  é um difeomorfismo de classe  $C^r$  (A.5) sobre sua imagem (denotada por  $\Sigma_1$ ) e mesmo tornar mais precisa a definição acima usando o teorema do fluxo tubular e suas conseqüências.

Embora não diretamente útil para a representação geométrica de propriedades do campo, a Transformação de Poincaré é um instrumento valioso na demonstração de resultados como o teorema a seguir; este sim, com um forte apelo para a representação geométrica do fluxo. Vejamos algumas definições preliminares

Seja um campo vetorial  $X: \Delta \rightarrow \mathfrak{R}^2$ , de classe  $C^1$ , no plano. Uma órbita periódica  $\gamma$  de X chama-se ciclo limite se existe uma vizinhança V de  $\gamma$  tal que  $\gamma$  é a única órbita fechada de X que intercepta V.

Teorema 8:

Existem apenas os seguintes tipos de ciclos limites (diminuindo  $V$  se necessário):

Estável:

$$\lim_{t \rightarrow \infty} d(\varphi(t, q), \gamma) = 0$$

quando  $t \rightarrow \infty$  para todo  $q \in V$ ;

Instável:

$$\lim_{t \rightarrow -\infty} d(\varphi(t, q), \gamma) = 0$$

quando  $t \rightarrow -\infty$  para todo  $q \in V$ ;

Semi-estável:

$$\lim_{t \rightarrow \infty} d(\varphi(t, q), \gamma) = 0$$

$$\lim_{t \rightarrow -\infty} d(\varphi(t, q), \gamma) = 0$$

quando  $t \rightarrow \infty$  para todo  $q \in V \cap \text{Ext}\gamma$  e  $t \rightarrow -\infty$  para todo  $q \in V \cap \text{Int}\gamma$ , ou o contrário.

**Demonstração:** [Sotomayor (1979)].

A Figura 3.5 ajuda esclarece o teorema. Nesta figura representamos um ciclo limite  $\gamma$ , uma seção transversal  $\Sigma$  e indicamos os sentidos positivos do fluxo ao longo de  $\gamma$  da órbita que passa por um ponto  $q$  dentro da vizinhança  $V$  do teorema acima.

Uma vez que temos garantida a existência e unicidade das curvas integrais a curva integral passando por  $q$  não pode interceptar o círculo, logo, ou a imagem  $\pi(q)$  pela Transformação de *Poincaré* ficará entre  $p$  e  $q$  (Estável) ou ficará antes de  $q$  (Instável), assumindo-se a orientação indicada na Figura 3.5 para a seção transversal  $\Sigma$ . Para um ponto  $q$  localizado no interior de  $\gamma$  teremos uma órbita confinada que não poderá cortar o ciclo limite ficando limitada ao comportamento dado pelo item a) ou b) do teorema.



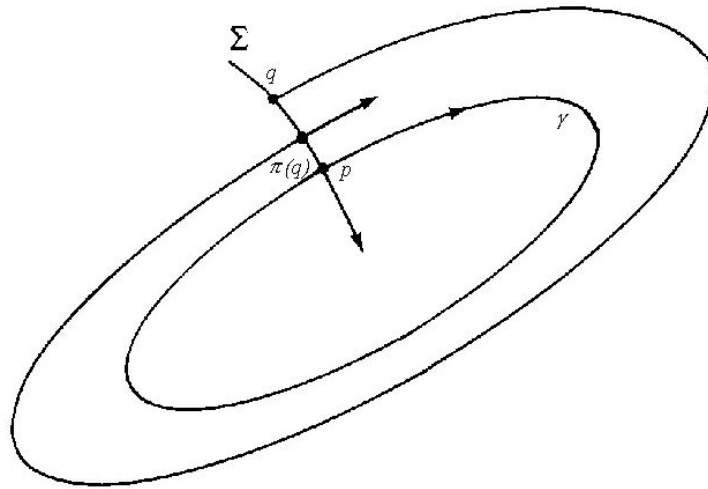


Figura 3.5 – Ciclo limite e Transformação de Poincaré.

### 3.4.6 Teorema de Poincaré-Bendixson

Certas técnicas para visualização fazem uso da idéia de conectar convenientemente os pontos singulares de um campo vetorial para obter um “esqueleto” que caracteriza a topologia do campo. A questão que colocamos é, do ponto de vista da teoria de sistemas dinâmicos, qual o significado deste esqueleto?

Mostraremos na seção 4.3.3 do próximo capítulo, quando tais técnicas serão apresentadas, que o esqueleto obtido tem uma estreita relação com o conjunto  $\omega$ -limite de órbitas convenientes, o qual passaremos a definir e analisar nesta seção. Por final, apresentaremos o Teorema de Poincaré-Bendixson que caracteriza estes conjuntos para o caso bidimensional.

Sejam  $\Delta$  um subconjunto aberto do espaço Euclidiano  $\mathfrak{R}^n$  e  $X:\Delta \rightarrow \mathfrak{R}^n$  um campo vetorial de classe  $C^k$ ,  $k \geq 1$ .

Seja  $\varphi(t) = \varphi(t, p)$  a curva integral de  $X$  passando pelo ponto  $p$ , definida no seu intervalo máximo  $I_p$ ,  $I_p = (w_-(p), w_+(p))$ . Se  $w_+(p) = \infty$  define-se o conjunto:

$$\omega(p) = \{q \in \Delta; \exists(t_n) \text{ com } t_n \rightarrow \infty \text{ e } \varphi(t_n) \rightarrow q, \text{ quando } n \rightarrow \infty\}.$$

Analogamente, se  $w_-(p) = -\infty$ , define-se o conjunto:

$$\alpha(p) = \{q \in \Delta; \exists(t_n) \text{ com } t_n \rightarrow -\infty \text{ e } \varphi(t_n) \rightarrow q, \text{ quando } n \rightarrow \infty\}.$$

Os conjuntos  $\omega(p)$  e  $\alpha(p)$  são chamados respectivamente de conjunto  $\omega$ -limite e  $\alpha$ -limite de  $p$ .

Podemos definir estes conjuntos para todo ponto de uma órbita. O conjunto  $\omega$ -limite de uma órbita  $\gamma$  é o conjunto  $\omega(p)$ , para qualquer  $p \in \gamma$ . Analogamente define-se o conjunto  $\alpha$ -limite de uma órbita  $\gamma$ .

Durante esta seção vamos nos restringir ao estudo do conjunto  $\omega$ -limite uma vez que dados  $\varphi(t) = \varphi(t, p)$  a curva integral do campo  $X$  pelo ponto  $p$  e  $\psi(t) = \psi(t, p)$  a curva integral do campo  $-X$  pelo ponto  $p$ , então  $\psi(t, p) = \varphi(-t, p)$ .

Vejamos algumas propriedades básicas do conjunto  $\omega(p)$ .

**Teorema 9:**

Sejam  $X: \Delta \rightarrow \mathbb{R}^n$  um campo de classe  $C^k$ ,  $k \geq 1$ , definido num aberto  $\Delta \subset \mathbb{R}^n$  e  $\gamma^+(p) = \{\varphi(t, p); t \geq 0\}$  a semi-órbita positiva do campo  $X$  pelo ponto  $p$ . Se  $\gamma^+(p)$  está contida num subconjunto compacto  $K \subset \Delta$ , então:

$$\omega(p) \neq \emptyset,$$

$\omega(p)$  é compacto,

$\omega(p)$  é invariante por  $X$ , isto é, se  $q \in \omega(p)$  então a curva integral de  $X$  por  $q$  está contida em  $\omega(p)$ ,

$\omega(p)$  é conexo.

**Demonstração:** [Sotomayor (1979)].

Observações:

Se  $p$  é um ponto singular de  $X$  então  $\omega(p) = \{p\}$  pois neste caso  $\varphi(t, p) = p$ , para todo  $t \in \mathbb{R}$ .

Se  $\gamma$  é uma órbita fechada do campo  $X$  então  $\omega(p) = \gamma$ ,  $\forall p \in \gamma$ . Além disso, se  $\gamma$  é um ciclo limite estável, existe uma vizinhança  $V$  de  $\gamma$  tal que  $\omega(p) = \gamma$ ,  $\forall p \in V$ .

Propriedades análogas podem ser estabelecidas para os outros dois tipos de ciclos limites.

Assim, existem órbitas do campo  $X$  para as quais os conjuntos  $\omega$ -limite contêm elementos essenciais para a caracterização da topologia de um campo bidimensional: pontos singulares e ciclos limites.

Como exemplo, seja um campo  $X: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  de classe  $C^1$ , cuja órbitas são espirais exteriores e interiores à circunferência  $C$  de centro na origem (também uma órbita do campo). A Figura 3.6 representa o retrato de fase (seção 3.4.1) em questão:

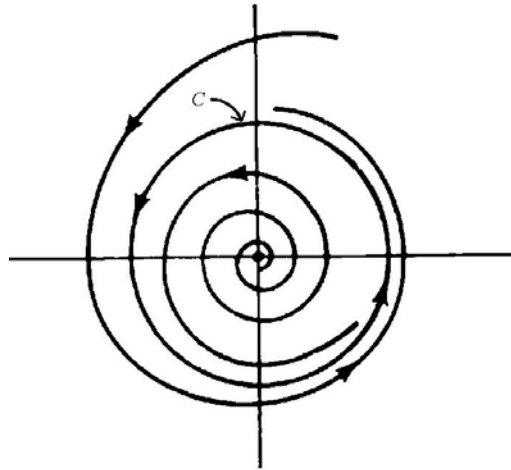


Figura 3.6 – Retrato de fase com ciclo limite  $C$  e foco instável na origem.

Neste caso, temos:

- i.  $\omega(p) = C, \forall p \neq (0,0)$
- ii.  $\omega(p) = \{(0,0)\}, \text{ se } p = (0,0).$

Pelo comportamento das espirais exteriores e interiores podemos ver que  $C$  é um ciclo limite do campo e a origem é um foco instável.

Suponha agora, que as únicas informações que temos do campo  $X: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  é que seu único ponto singular é um foco instável na origem e que a única órbita fechada é a circunferência  $C$  com a orientação da Figura 3.6. Seria possível existir uma órbita contida num subconjunto fechado e limitado  $\Delta \subset \mathbb{R}^2$ , contendo a região representada na Figura 3.6, tal que  $\omega(\gamma) \not\subset C \cup \{(0,0)\}$ ?

O teorema 10 a seguir, e suas conseqüências, mostram que isto não é possível o que mostra a importância do conjunto  $\omega$ -limite na determinação da topologia.

No teorema que segue, estaremos supondo  $\Delta$  um subconjunto do  $\mathbb{R}^2$  e  $X$  um campo vetorial de classe  $C^k$  ( $k \geq 1$ ) em  $\Delta$ . A vantagem do caso bidimensional é a validade do Teorema da Curva de Jordan que estabelece: Se  $J$  é uma curva fechada, contínua e simples, então  $\mathbb{R}^2 - J$  tem duas componentes conexas, uma limitada e outra não limitada, as quais têm  $J$  como fronteira comum.

**Teorema 10** (Poincaré-Bendixson):

Seja  $\varphi(t) = \varphi(t, p)$  uma curva integral de  $X: \Delta \rightarrow \mathbb{R}^2$ , definida para todo  $t \geq 0$  tal que  $\gamma^+(p)$  esteja contida num compacto  $K \subset \Delta$ . Suponha que o campo  $X$  tenha um número finito de singularidades em  $\omega(p)$ . Temos as seguintes alternativas:

Se  $\omega(p)$  possui somente pontos regulares, então  $\omega(p)$  é uma órbita periódica,

Se  $\omega(p)$  contém pontos regulares e singulares, então  $\omega(p)$  consiste de um conjunto de órbitas, cada uma das quais tende a um desses pontos singulares quando  $t \rightarrow \pm\infty$ ,

Se  $\omega(p)$  não contém pontos regulares, então  $\omega(p)$  é um ponto singular.

**Demonstração:** [Sotomayor (1979)].

A Figura 3.7 exemplifica as alternativas deste teorema.

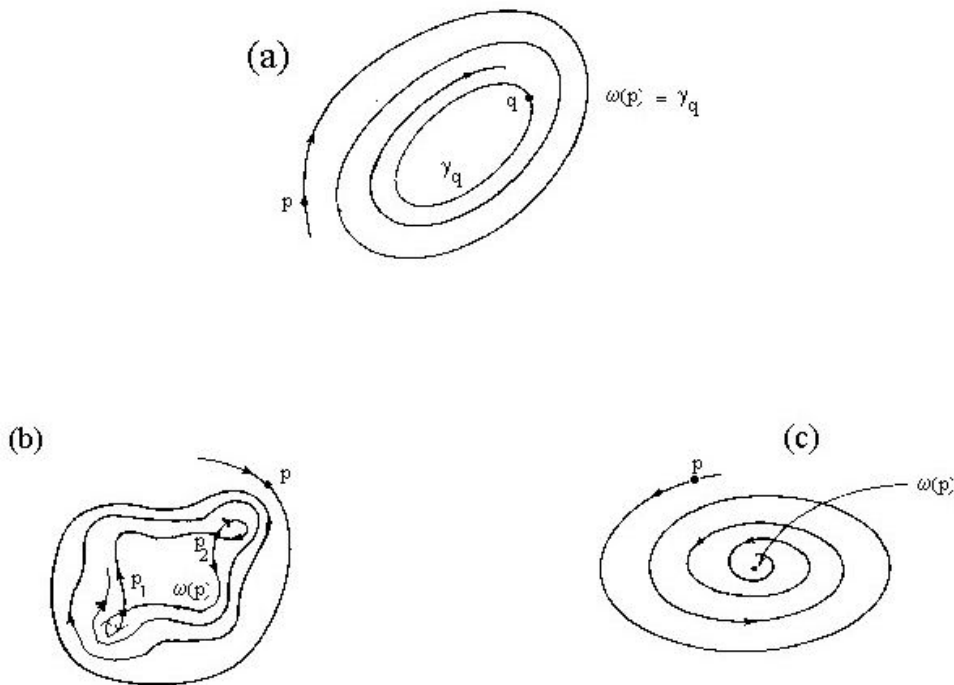


Figura 3.7 – (a)  $\omega(p)$  possui apenas pontos regulares. (b)  $\omega(p)$  possui pontos regulares e singulares. (c)  $\omega(p)$  não contém pontos regulares.

Voltemos ao campo da Figura 3.6 acima. Suponha que exista uma curva integral  $\gamma$  cuja semi-órbita positiva satisfaz as condições do teorema acima e tal que  $\omega(\gamma) \not\subset C \cup \{(0,0)\}$ . Então, uma vez que não existem outros pontos singulares, e o conjunto  $\omega(\gamma)$  é conexo (Teorema 9), a única possibilidade para  $\omega(\gamma)$  seria dada pelo item a) do teorema acima.

Portanto, teríamos a existência de uma curva fechada contida na região  $o$  que fica descartado, se sabemos que a única curva integral fechada de  $X$  é  $C$ . Além disso, uma consequência do teorema acima é a ocorrência de pontos singulares no

interior de qualquer órbita fechada de um campo  $X: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  de classe  $C^1$ . Tal fato, por sua vez, demonstra a importância, no caso bidimensional, das órbitas fechadas na caracterização da topologia de um campo bidimensional.

Os fluxos encontrados na natureza são em geral 3D(onde não temos as simplificações introduzidas do Teorema da Curva de Jordan), podendo apresentar pontos singulares não isolados, saindo, portanto, do escopo do teorema de Poincaré-Bendixon.

Apesar disto, os exemplos acima mostram como a detecção do conjunto  $\omega(p)$ , para um ponto  $p$  conveniente, é suficiente para caracterizar a topologia de toda uma região do fluido.

Observemos que, se ocorrer a hipótese b, este conjunto (esqueleto) é obtido conectando-se convenientemente os pontos singulares do campo o que dará origem a trajetórias que separam regiões com topologias distintas. É exatamente esta a idéia explorada na técnica apresentada discutida na seção 4.3.3 do próximo capítulo.

## Capítulo 4

### Métodos Numéricos para Animação de Fluidos

#### 4.1. Introdução

Neste capítulo desenvolvemos métodos numéricos para solução das equações de Navier-Stokes, sob o ponto de vista da animação de fluidos. Assim, é importante ressaltar que não se trata de um texto tradicional no assunto. Por exemplo, discutimos o método de Runge-Kuta, que não é usualmente utilizado para simulação computacional de fluidos em física e engenharia, mas que pode fornecer bons resultados para a animação de fluidos. Por outro lado, fazemos uma breve introdução aos métodos de elementos finitos, os quais são largamente utilizados em engenharia, mas com poucas referências para a animação computacional de fluidos.

Tais diferenças de abordagens se devem ao seguinte aspecto: o principal objetivo da animação computacional de fluidos é a geração de efeitos visuais enquanto que na física e engenharia se procura analisar com precisão fenômenos físicos envolvendo fluidos. Assim, na animação de fluidos estamos “livres” para usar métodos que, embora comprometam a precisão dos resultados gerados, fornecem o efeito visual com o realismo desejado.

Tendo em vista este aspecto, passemos à descrição dos métodos de interesse para esta monografia. Inicialmente, vamos fazer algumas considerações gerais sobre métodos numéricos.

#### 4.2. Análise Numérica

Os modelos matemáticos obtidos usando-se as equações das seções 2.2-2.4 precisam ser discretizados se quisermos efetuar simulações numéricas. O resultado desta discretização é a substituição do meio contínuo inicial do modelo por um conjunto discreto e finito de pontos no espaço e no tempo nos quais as derivadas que aparecem nas equações serão aproximadas. Nos métodos tradicionais de Elementos Finitos e Diferenças Finitas, este conjunto de pontos é conectado seguindo uma topologia conveniente o que dará origem a uma malha que pode ser de um dos três tipos:

- Malha Estruturada: cada ponto tem o mesmo número de pontos vizinhos.
- Malha Não-Estruturada: O número de pontos vizinhos é variável.
- Malha Mista: Parte da malha é estruturada e parte é não-estruturada.

O método de elementos finitos possui características mais gerais que o segundo método por permitir subdivisões do domínio em células de forma e

tamanho arbitrários. Este método tem por base uma formulação variacional para as equações de campo (seção 4.4).

O método de diferenças finitas é baseado nas propriedades da série de Taylor e na aplicação direta das definições de derivadas. É um método mais simples, embora tenha a desvantagem de requerer um alto grau de regularidade da malha.

Três propriedades são fundamentais para o comportamento de um esquema numérico: consistência, estabilidade e convergência.

A condição de consistência define uma relação entre a equação diferencial e sua formulação discreta; a condição de estabilidade define uma relação entre a solução exata do sistema linear resultante da discretização e solução numérica deste sistema; finalmente, a condição de convergência conecta a solução numérica à solução exata da equação diferencial.

Estas questões nos serão úteis no capítulo 5 e por isso serão analisadas com algum detalhe.

Consistência expressa o fato de que a equação discretizada deve tender à equação diferencial correspondente quando  $\Delta t \rightarrow 0$  e  $\Delta \rightarrow 0$ , onde  $\Delta t$  é o passo no tempo e  $\Delta$  é o tamanho característico das células da malha.

O conceito de estabilidade é um pouco mais delicado. Para formular precisamente este conceito, vamos denotar por  $\varepsilon_i^n$  a diferença entre a solução computada ( $u_i^n$ ) e a solução exata ( $\bar{u}_i^n$ ) da equação discretizada:

$$\varepsilon_i^n = u_i^n - \bar{u}_i^n, \quad (2.54)$$

então uma forma de escrever a condição de estabilidade é a seguinte:

$$\lim_{n \rightarrow \infty} |\varepsilon_i^n| \leq K, \text{ para } \Delta t \text{ fixo,} \quad (2.55)$$

com K independente de n. Esta condição de estabilidade tem um defeito: ela não garante que o erro não se torna muito alto para instantes intermediários  $t_n = n\Delta t$ .

Assim, devemos procurar uma condição de estabilidade mais geral. Essa nova condição estabelece que qualquer componente da solução inicial não deve ser amplificado sem um limite superior finito.

Seja então o vetor  $V^n$  o vetor formado pelas quantidades desconhecidas, em cada ponto da malha, em um dado instante  $n\Delta t$ . O esquema de discretização, seja ele implícito ou explícito, pode ser escrito na forma matricial:

$$V^{n+1} = C \cdot V^n, \quad (2.56)$$

onde o operador C é representado por uma matriz que depende do passo de tempo  $\Delta t$  e do tamanho  $\Delta$  da malha .

Se  $V^0$  representa a solução inicial para  $t=0$  então, por repetidas aplicações do operador C nós teremos:

$$\begin{aligned}
V^1 &= C \cdot V^0, \\
V^2 &= C \cdot V^1 = C^2 \cdot V^0, \\
&\vdots \\
&\vdots \\
&\vdots \\
V^n &= C^n \cdot V^0.
\end{aligned}
\tag{2.57}$$

A nova condição estabilidade pode agora ser colocada na seguinte forma:

Para que todo  $V^n$  permaneça limitado e o esquema definido pelo operador  $C$  permaneça estável, o conjunto infinito de operadores  $C^n$  deve ser uniformemente limitado. Isto é, existe uma constante  $K$  tal que:

$$\|C^n\| < K, \quad 0 < \Delta t < \tau \text{ e } 0 \leq n\Delta t \leq T, \tag{2.58}$$

para valores fixos de  $\tau$  e  $T$  e para todo  $n$ . Esta condição implica na definição de alguma norma no espaço de funções em questão, a saber, a norma de Hilbert no espaço das funções quadrado integráveis [Hirsch (1988)].

A condição de convergência pode ser estabelecida dizendo-se que a solução numérica  $u_i^n$  deve aproximar-se da solução exata  $\tilde{u}_i^n$  da equação diferencial em qualquer ponto  $P_i$  da malha e para qualquer instante  $t_n = n\Delta t$  quando  $\Delta t \rightarrow 0$  e  $\Delta x \rightarrow 0$ . Esta condição pode ser colocada em termos do operador  $C$  como segue:

$$\lim_{\substack{\Delta t \rightarrow 0 \\ n \rightarrow \infty}} \| [C(\Delta t)]^n V^0 - \tilde{V}(t) \| = 0, \tag{2.59}$$

com  $n\Delta t = t$  fixado. Em termos do erro  $\tilde{\varepsilon}_i^n = u_i^n - \tilde{u}(i\Delta x, n\Delta t)$ , a condição de convergência pode ser escrita:

$$\lim_{\substack{\Delta t \rightarrow 0 \\ \Delta x \rightarrow 0}} |\tilde{\varepsilon}_i^n| = 0, \tag{2.60}$$

para qualquer ponto fixo  $P_i$  da malha e qualquer instante fixo  $t_n = n\Delta t$ .

Claramente, as condições de consistência, estabilidade e convergência estão relacionadas entre si, e a relação precisa está contida no *Teorema Fundamental de Equivalência de Lax* [Hirsch (1988)]:

“Para um problema de valor inicial bem posto e um esquema de discretização consistente, a condição de estabilidade é necessária e suficiente para convergência”.

Este teorema fundamental mostra que a convergência é uma decorrência das condições de consistência e estabilidade quando a existência e a unicidade da solução exata estiverem garantidas (problema bem posto).



A discussão acima está focada em métodos dependentes de malhas, onde a quantidade  $\Delta$  (tamanho característico das células da malha) faz sentido. No método SPH, não se faz uso de uma subdivisão celular (ou simplicial) do domínio de interesse (uma malha). Os fundamentos do método SPH estão na teoria de interpolação, tendo como ponto de partida a representação das quantidades de interesse via convolução com um núcleo de suavização. Contudo, a análise acima pode ser aplicada mesmo neste caso, substituindo-se  $\Delta$  pelo suporte do núcleo.

### 4.3. Diferenças Finitas

O método de diferenças finitas está fundamentado na representação de uma função suave  $f$  em série de Taylor, na vizinhança de um ponto  $x$ :

$$f(x+h) = f(x) + f^{(1)}(x)h + \frac{1}{2!} f^{(2)}(x)h^2 + \frac{1}{3!} f^{(3)}(x)h^3 + \dots, \quad (4.61)$$

onde  $f^{(i)}(x)$  representa a derivada de ordem  $i$ , calculada no ponto  $x$ .

A partir desta representação, pode-se obter as expressões usuais para a aproximação da primeira derivada de  $f$  em  $x$ . Seja então,  $x - x_0 = h$ , com  $h > 0$ . Neste caso, lembrando a série de Taylor com resto [Chapra-Canale(1988)], teremos:

$$f(x+h) = f(x) + f^{(1)}(x)h + O(h^2), \quad (4.62)$$

onde  $O(h^2)$  representa o resto decorrente do truncamento da série em (4.61), o qual tem a propriedade:

$$\lim_{h \rightarrow 0} \frac{O(h^2)}{h} = \lim_{h \rightarrow 0} O(h) = 0. \quad (4.63)$$

Assim, escrevendo  $x = x_0 + h$ , e isolando  $f^{(1)}(x)$  na expressão (4.62) podemos escrever:

$$f^{(1)}(x) = \frac{f(x+h) - f(x)}{h} + O(h), \quad (4.64)$$

Se desprezarmos o resto no segundo membro desta expressão, obtemos a aproximação  $f^{(1)}(x)$  em diferenças finitas avançadas:

$$f^{(1)}(x) \approx \frac{f(x+h) - f(x)}{h}. \quad (4.65)$$

De maneira semelhante podemos obter a aproximação de  $f^{(1)}(x)$  em diferenças finitas retardadas:

$$f^{(1)}(x) \approx \frac{f(x) - f(x-h)}{h}. \quad (4.66)$$

Aproximação de ordem mais alta para a primeira derivada podem ser obtidas por argumentos semelhantes. De maneira geral, quando procuramos representar a d-ésima derivada  $f^{(d)}(x)$  estamos procurando uma expressão do tipo:

$$\frac{h^d}{d!} f^{(d)}(x) + O(h^{d+p}) = \sum_{i=i_{\min}}^{i=i_{\max}} C_i f(x+ih), \quad (4.67)$$

para alguma escolha dos extremos  $i_{\min}, i_{\max}$  e dos coeficientes  $C_i$ , sendo  $p$  a ordem da aproximação desejada.

Uma maneira elegante de obter tais quantidades é desenvolvida em [Eberly (2003)] e reproduzida abaixo.

Formalmente, podemos escrever a série de Taylor para  $f(x+ih)$  da forma:

$$f(x+ih) = \sum_{n=0}^{\infty} i^n \frac{h^n}{n!} f^{(n)}(x), \quad (4.68)$$

Substituindo esta expressão na equação (4.67), obteremos:

$$\frac{h^d}{d!} f^{(d)}(x) + O(h^{d+p}) = \sum_{i=i_{\min}}^{i=i_{\max}} C_i \left( \sum_{n=0}^{\infty} i^n \frac{h^n}{n!} f^{(n)}(x) \right), \quad (4.69)$$

a qual, trocando o ordem dos somatórios no segundo membro, assume a forma:

$$\frac{h^d}{d!} f^{(d)}(x) + O(h^{d+p}) = \sum_{n=0}^{\infty} \left( \sum_{i=i_{\min}}^{i=i_{\max}} i^n C_i \right) \frac{h^n}{n!} f^{(n)}(x). \quad (4.70)$$

Finalmente, podemos isolar a d-ésima derivada no primeiro membro dividindo a equação por  $h^d / d!$  e usando o série de Taylor com resto:

$$f^{(d)}(x) = \frac{d!}{h^d} \sum_{n=0}^{d+p-1} \left( \sum_{i=i_{\min}}^{i=i_{\max}} i^n C_i \right) \frac{h^n}{n!} f^{(n)}(x) + O(h^p), \quad (4.71)$$

onde usamos o fato que:

$$\frac{d!}{h^d} O(h^{d+p}) = O(h^p).$$

Para que a equação (4.71) seja satisfeita, é suficiente que:

$$\sum_{i=i_{\min}}^{i=i_{\max}} i^n C_i = \begin{cases} 0, & \text{se } 0 \leq n \leq d+p-1 \text{ e } n \neq d, \\ 1, & n = d. \end{cases} \quad (4.72)$$

Este é um sistema formado por  $d+p$  equações lineares com  $i_{\min} + i_{\max} + 1$  incógnitas. Impondo  $d+p = i_{\min} + i_{\max} + 1$  teremos solução única. A Tabela a seguir mostra o resultado obtido para algumas escolhas de  $p$  e  $q$  e o correspondente tipo de diferenças finitas (avançada, central, retardada).

Derivada	d	p	Tipo	$i_{\min}$	$i_{\max}$
$f^{(1)}$	1	1	Avançada	0	1
$f^{(1)}$	1	1	Retardada	-1	0
$f^{(1)}$	1	2	Central	-1	1
$f^{(1)}$	1	2	Avançada	0	2
$f^{(1)}$	1	2	Retardada	-2	0
$f^{(1)}$	1	4	Central	-2	2
$f^{(2)}$	2	1	Avançada	0	2
$f^{(2)}$	2	2	Retardada	-1	1
$f^{(2)}$	2	4	Central	-2	2

Tabela 4.1: Tipos mais comuns de diferenças finitas para aproximação de derivadas de primeira e segunda ordens.

#### 4.4. Elementos Finitos

Embora existam poucas referências da utilização explícita de Elementos Finitos para animação computacional de fluidos [Enright (2002)], é importante acrescentar a este material uma introdução objetiva a este tema, uma vez que os esquemas numéricos derivados desta técnica podem ser mais estáveis e não existe a necessidade de malhas regulares, como no caso das diferenças finitas.

Nesta apresentação, seguiremos a referência [Hughes(1987)]. Os principais elementos do método de elementos finitos para solução de um problema de valor de contorno são:

A formulação variacional (ou fraca) do problema;

A solução aproximada da equação variacional através do uso de funções de elementos finitos.

Como exemplo, consideremos o problema a seguir:

Dado  $f : [0,1] \rightarrow \mathfrak{R}$  e constantes reais  $g, h$  encontre  $u : [0,1] \rightarrow \mathfrak{R}$ , tal que:

$$(S) \quad \begin{cases} u_{,xx} + f = 0, \\ u(1) = g, \\ -u_{,x}(0) = h, \end{cases}$$

onde  $u_{,x}$  e  $u_{,xx}$  significam  $du/dx$  e  $d^2u/dx^2$ , respectivamente.

Para desenvolver a formulação fraca, ou variacional, de (S), precisamos caracterizar duas classes de funções. A primeira classe é composta por funções candidatas à solução. Considerando as restrições de (S), fica claro que esta classe de funções deverá satisfazer a condição de contorno  $u(1) = g$ . A outra restrição não será imposta nesta definição. Vamos assumir também que as derivadas de primeira ordem das funções em questão são quadrado integráveis, ou seja:

$$\int_0^1 (u_{,x})^2 dx < \infty \quad (4.73).$$

Funções  $u: [0,1] \rightarrow \Re$  que satisfazem (4.73) são denominadas funções  $H^1([0,1])$ . Portanto, o conjunto de funções candidatas à solução, denotado por  $\zeta$ , de (S) é dado por:

$$\zeta = \{u \mid u \in H^1([0,1]), \quad u(1) = g\} \quad (4.74)$$

O segundo conjunto de funções é chamado de funções peso, ou variações. Este conjunto é similar ao anterior, com a diferença de que as funções deste conjunto devem satisfazer  $w(1) = 0$ . Este espaço de funções será denotado por  $\mathcal{G}$  e definido da seguinte forma:

$$\mathcal{G} = \{w \mid w \in H^1([0,1]), \quad w(1) = 0\} \quad (4.75)$$

Nestas condições, a forma fraca do problema (S) é dada por: Dados  $f, g$  e  $h$  como anteriormente, encontrar  $u \in \zeta$  tal que para todo  $w \in \mathcal{G}$  tenhamos:

$$(W) \quad \int_0^1 w_{,x} u_{,x} dx = \int_0^1 w f dx + w(0)h$$

Formulações deste tipo são freqüentemente chamadas princípio do trabalho virtual ou do deslocamento virtual. As funções  $w$  são também denominadas deslocamentos virtuais. A equação acima é denominada equação variacional, ou equação do trabalho virtual. A solução do problema (W) é denominada fraca, ou solução generalizada.

A relação entre os problemas (S) e (W) é estabelecida pela seguinte proposição:

Proposição:

Uma função  $u \in \zeta$  é solução de (S) se e somente se  $u$  é solução de (W).

Demonstração:

(a) Se  $u \in \zeta$  é solução de (S) então, multiplicando a equação diferencial em (S) por  $w \in \mathcal{G}$  e integrando teremos:

$$\int_0^1 w(u_{,xx} + f) dx = 0 \quad (4.76).$$

Usando agora a relação:

$$(wu_{,x})_{,x} = w_{,x}u_{,x} + wu_{,xx} \Rightarrow wu_{,xx} = (wu_{,x})_{,x} - w_{,x}u_{,x}, \quad (4.77),$$

podemos re-escrever a equação (4.76) na forma:

$$\int_0^1 w_{,x}u_{,x} dx - \int_0^1 w f dx - wu_{,xx} \Big|_1^0. \quad (4.78)$$

Re-arranjando os termos e usando as restrições  $u_{,x}(0) = h$  e  $w(1) = 0$  obtemos:

$$\int_0^1 w_{,x}u_{,x} dx = \int_0^1 w f dx + w(0)h. \quad (4.79)$$

Finalmente, uma vez que, por hipótese  $u$  é solução de (S), este satisfaz  $u(1) = g$ , e, portanto,  $u \in \zeta$ . Além disso, uma vez que  $u$  satisfaz para todo  $w \in \mathcal{G}$ , temos que  $u$  é solução fraca do problema.

(b) Supondo agora que  $u$  é solução fraca. Então, por definição,  $u \in \zeta$  e, conseqüentemente,  $u(1) = g$ , e:

$$\int_0^1 w_{,x}u_{,x} dx = \int_0^1 w f dx + w(0)h. \quad (4.80)$$

para todo  $w \in \mathcal{G}$ . Integrando por partes, pela aplicação da equação (4.77), e usando o fato  $w(1) = 0$  obteremos:

$$0 = \int_0^1 w(u_{,xx} + f) dx + w(0)[u_{,x}(0) + h] \quad (4.81)$$

Para provar que  $u$  é solução de (S) é suficiente mostrar que:

$$u_{,xx} + f = 0 \text{ em } [0,1],$$

$$u_{,x}(0) + h = 0.$$

Para mostrar (i) basta escolher um elemento conveniente de  $\mathcal{G}$ . Se usarmos  $w$  da forma:

$$w = \phi(u_{,xx} + f), \quad (4.82)$$

com  $\phi$  suave,  $\phi(x) > 0$ , para todo  $x \in ]0,1[$  e  $\phi(0) = \phi(1) = 0$ , teremos que a função  $w$  assim definida é tal que  $w(1) = 0$ , e, portanto,  $w \in \mathcal{G}$  (por exemplo, fazendo  $\phi(x) = x(1-x)$  teremos o resultado desejado). Além disso,  $w(0) = 0$ , o que anula o termo constante do segundo membro da equação (4.81). Portanto, substituindo (4.82) em (4.81), obteremos:

$$0 = \int_0^1 \phi(u_{,xx} + f)^2 dx + 0. \quad (4.83)$$

Uma vez que  $\phi > 0$  em  $]0,1[$  é imediato da equação acima que (i) é verdadeiro. Portanto, a equação (4.81) fica restrita a:

$$0 = w(0)[u_{,x}(0) + h] \quad (4.84)$$

Uma vez que não há restrições para o valor de  $w$  em  $x = 0$ , podemos assumir que  $w(x) \neq 0$ , o que implica que (ii) é verdadeira. De (a) e (b) vem que a demonstração está completa.

#### 4.5. Método de Aproximação de Galerkin

Descreveremos agora um método para obter-se soluções aproximadas para problemas de valores de contorno baseado na formulação fraca. O primeiro passo será a construção de aproximações de dimensão finita para os espaços  $\zeta$  e  $\mathcal{G}$ . Estes espaços funcionais serão denotados por  $\zeta^h$  e  $\mathcal{G}^h$ , respectivamente, onde "h" refere-se à dimensão característica de uma malha (discretização do domínio) associada. O papel desta malha no método é permitir a definição de uma base de funções, com suporte limitado, para o espaço (vetorial)  $\mathcal{G}^h$ . Estes espaços devem satisfazer as condições:

$$\zeta^h \subset \zeta \Rightarrow u^h(1) = g, \quad (4.86)$$

$$\mathcal{G}^h \subset \mathcal{G} \Rightarrow w^h(1) = 0. \quad (4.87)$$

Método de (Bubnov-) Galerkin: Suponha que  $\mathcal{G}^h$  é conhecido. Então, para cada elemento  $v^h \in \mathcal{G}^h$ , nós construímos uma função  $u^h \in \zeta^h$  dada por:

$$u^h = v^h + g^h, \quad (4.88)$$

onde  $g^h$  é uma função que satisfaz a condição de contorno

$$g^h(1) = g. \quad (4.89)$$

A expressão (4.88) será tomada como a definição do espaço  $\zeta^h$ . É importante observar que, a menos da definição de  $g^h$ , os espaços  $\zeta^h$  e  $\mathcal{G}^h$  são compostos por funções da mesma natureza.

O próximo passo será escrever o problema variacional W, em termos das funções  $u^h \in \zeta^h$  e  $v^h \in \mathcal{G}^h$ . Utilizando a notação adicional:

$$a(w^h, u^h) = \int_0^1 w_{,x} u_{,x} dx, \quad (4.90)$$

$$(w, f) = \int_0^1 w f dx, \quad (4.91)$$

A partir do Problema (W) obtemos a seguinte expressão:

$$a(w^h, u^h) = (w^h, f) + w^h(0)h. \quad (4.92)$$

Substituindo a definição (4.88) nesta expressão, e observando que  $a(\cdot, \cdot)$  é bilinear, teremos:

$$a(w^h, v^h) = (w^h, f) + w^h(0)h - a(w^h, g^h). \quad (4.93)$$

Esta equação será usada para encontrar  $v^h$ , a parte desconhecida de  $u^h$ . Para isto, o método de Galerkin define o espaço  $\mathcal{G}^h$  como sendo aquele formado pelas possíveis combinações lineares de funções pré-definidas  $N_A : [0,1] \rightarrow \mathfrak{R}$ , onde  $A = 1, 2, \dots, n$ . Desta forma teremos:

$$w^h \in \mathcal{G}^h \Leftrightarrow w^h = \sum_{A=1}^{A=n} c_A N_A. \quad (4.94)$$

As funções  $N_A$  são denominadas, funções de forma, funções de base, ou funções de interpolação. Uma vez que  $N_A \in \mathcal{G}^h$ , devemos ter:

$$N_A(1) = 0, \quad A = 1, 2, \dots, n, \quad (4.95)$$

o que implica diretamente que  $w^h(1) = 0$ .

Para definir os elementos  $u^h \in \zeta^h$  precisamos especificar a função  $g^h$  usada na definição (4.88). Para isto, será introduzida uma nova função de forma  $N_{n+1} : [0,1] \rightarrow \mathfrak{R}$  satisfazendo:

$$N_{n+1}(1) = 1. \quad (4.96)$$

É importante notar que  $N_{n+1} \notin \mathcal{G}^h$ . Assim,  $g^h$  pode ser escrita na forma:

$$g^h = g N_{n+1}, \quad g^h(1) = g. \quad (4.97)$$

Assim, podemos definir os elementos  $u^h \in \zeta^h$  pela expressão:

$$u^h = v^h + g^h = \sum_{A=1}^{A=n} d_A N_A + g N_{n+1}, \quad (4.98)$$

onde  $d_A \in \mathfrak{R}$ ,  $A = 1, 2, \dots, n$  são constantes e, por (4.95)-(4.96), fica evidente que  $u^h(1) = g$ .

Substituindo-se as expressões (4.94), (4.97) e (4.98) na equação (4.93), e rearranjando convenientemente os termos resultantes, obteremos finalmente:

$$0 = \sum_{A=1}^n c_A G_A, \quad (4.99)$$

onde:

$$G_A = \sum_{B=1}^n a(N_A, N_B) d_B - (N_A, f) - N_A(0)h + a(N_A, N_{n+1})g. \quad (4.100)$$

Uma vez que a expressão (4.99) deve ser satisfeita para todo  $w^h \in \mathcal{G}^h$ , ou seja, para todo  $c_A \in \mathfrak{R}$ ,  $A = 1, 2, \dots, n$ , é necessário que os coeficientes  $G_A$  sejam identicamente nulos, o que nos leva ao sistema de equações:

$$\sum_{B=1}^n a(N_A, N_B) d_B = (N_A, f) + N_A(0)h - a(N_A, N_{n+1})g, \quad (4.101)$$

onde as variáveis desconhecidas são os coeficientes  $d_B \in \mathfrak{R}$  da expressão (4.98).

Podemos observar que o segundo membro da equação acima pode ser calculado a partir dos dados do problema e das definições das funções de forma. Usualmente, escreve-se o sistema (4.101) na forma:

$$\sum_{B=1}^n K_{AB} d_B = F_A, \quad A = 1, 2, \dots, n, \quad (4.102)$$

onde:

$$K_{AB} = a(N_A, N_B), \quad (4.103)$$

$$F_A = (N_A, f) + N_A(0)h - a(N_A, N_{n+1})g, \quad (4.104)$$

ou, de forma compacta:

$$Kd = F, \quad (4.105)$$

onde  $K = [K_{AB}]$  é a matriz dos coeficientes do sistema (4.102),  $d = (d_B)$  e  $F = (F_A)$  são, respectivamente, o vetor coluna formado pelas incógnitas e o vetor coluna formado pelo segundo membro da equação (4.102). Para aplicações em mecânica do contínuo, costuma-se denominar a matriz K de matriz de rigidez, o vetor F de vetor de forças e o vetor d de vetor de deslocamentos.

#### 4.6. Runge Kutta

O método de Runge-Kutta é muito utilizado para solução numérica de problemas de valor inicial do tipo:



$$\begin{cases} \frac{dx}{dt} = X(t, x), \\ x(t_0) = x_0, \end{cases} \quad (4.106)$$

onde  $x_0 \in \mathfrak{R}^n$  é um ponto do espaço Euclidiano n-dimensional e  $X : I \times \Delta \subset \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  é um campo vetorial ( $I \subset \mathfrak{R}^+$ ). No Capítulo 3 analisamos analiticamente problemas do tipo acima.

O método de Runge-Kutta é um método iterativo baseado na expressão:

$$\begin{cases} x_{i+1} = x_i + \phi(t_i, x_i, h), \\ x(t_0) = x_0, \end{cases} \quad (4.107)$$

onde a função  $\phi(t_i, x_i, h)$ , denominada função de incremento, pode ser escrita na forma geral:

$$\phi(t_i, x_i, h) = a_1 k_1 + a_2 k_2 + \dots + a_n k_n,$$

onde  $a_1, a_2, \dots, a_n$  são constantes e  $k_1, k_2, \dots, k_n$  são dados por:

$$\begin{cases} k_1 = X(t_i, x_i), \\ k_2 = X(t_i + p_1 h, x_i + q_{11} k_1 h), \\ k_3 = X(t_i + p_2 h, x_i + q_{21} k_1 h + q_{22} k_2 h), \\ \dots \\ k_n = X(t_i + p_{n-1} h, x_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + \dots + q_{n-1,n-1} k_{n-1} h). \end{cases} \quad (4.108)$$

Uma justificativa simples do esquema acima pode ser obtida observando-se a série de Taylor da solução  $x$  do problema (4.106). Neste caso, temos:

$$x(t+h) = x(t) + \frac{dx}{dt} h + \frac{1}{2} \frac{d^2 x}{dt^2} h^2 + O(h^2), \quad (4.109).$$

Pela expressão (4.106) temos que:

$$\frac{dx(t)}{dt} = X(t, x), \quad \frac{d^2 x(t)}{dt^2} = \frac{dX}{dt} = \frac{\partial X}{\partial t} + \frac{\partial X}{\partial x} \frac{dx(t)}{dt} \quad (4.110).$$

Assim, substituindo estas expressões em (4.109) obteremos:

$$x(t+h) = x(t) + X(t, x)h + \frac{1}{2} \left( \frac{\partial X}{\partial t} + \frac{\partial X}{\partial x} \frac{dx(t)}{dt} \right) h^2 + O(h^2), \quad (4.111).$$

Finalmente, agrupando convenientemente os termos, obtemos:

$$x(t+h) = x(t) + \left[ X(t, x) + \frac{1}{2} \left( \frac{\partial X}{\partial t} + \frac{\partial X}{\partial x} \frac{dx(t)}{dt} \right) h \right] h + O(h^2). \quad (4.112)$$

Tomando agora a série de Taylor para o campo  $X(t, x)$ :

$$X(t+h, x+hX(t, x)) = X(t, x) + \frac{\partial X}{\partial t} h + \frac{\partial X}{\partial x} hX(t, x) + O(h^2), \quad (4.113)$$

A expressão (4.112) pode ser escrita na forma:

$$x(t+h) = x(t) + X(t+h, x+hX(t, x))h + O(h^2), \quad (4.114)$$

Generalizando este resultado, podemos escrever:

$$\begin{aligned} x(t+h) &= x(t) + \left[ (a_1 + a_2)X(t, x) + a_2 \frac{\partial X}{\partial t} p_1 h + a_2 \frac{\partial X}{\partial x} X(t, x) q_{11} h \right] h + O(h^2) \Rightarrow \\ x(t+h) &= x(t) + a_1 X(t, x) h + a_2 \left[ X(t, x) + \frac{\partial X}{\partial t} p_1 h + \frac{\partial X}{\partial x} X(t, x) q_{11} h \right] h + O(h^2). \end{aligned} \quad (4.115)$$

Comparando a expressão entre colchetes com a equação (4.113), observamos que podemos escrever:

$$x(t+h) = x(t) + [a_1 X(t, x) + a_2 X(t + p_1 h, x + q_{11} X(t, x) h)] h + O(h^2), \quad (4.116)$$

onde  $p_1$ ,  $q_{11}$ ,  $a_1$  e  $a_2$  são constantes a determinar. Procedendo analogamente ao desenvolvimento acima, mas usando termos de ordem mais alta da série de Taylor, obteremos o esquema (4.108).

Os valores das constantes  $p_1$ ,  $q_{11}$ ,  $a_1$  e  $a_2$  na expressão (4.116) devem ser coerentes com a série de Taylor de  $x$ , dada pela expressão (4.109). Portanto, estas constantes devem satisfazer as equações:

$$\begin{aligned} a_1 + a_2 &= 1, \\ a_2 p_1 &= \frac{1}{2}, \\ a_2 q_{11} &= \frac{1}{2}. \end{aligned} \quad (4.117)$$

Claramente, existem infinitas soluções para este sistema. Pode-se escolher uma variável e parametrizar a solução em função desta ( $a_2$ , por exemplo). Neste caso, cada escolha pra  $a_2$  fornece um esquema diferente. Em particular, o método de Heun, com um único corretor, e o método do polígono melhorado, são dados, respectivamente, pelas seguintes escolhas:

$$a_2 = \frac{1}{2} \Rightarrow a_2 = \frac{1}{2}, p_1 = q_{11} = 1, \quad (4.118)$$

$$a_2 = 1 \Rightarrow a_1 = 0, p_1 = q_{11} = \frac{1}{2}. \quad (4.119)$$

Outras possibilidades usualmente encontradas na literatura são listadas em [Chapra-Canale(1988)]. Os métodos de Runge-Kutta definidos por estes valores são

de segunda-ordem, ou seja, o erro de truncamento da série de Taylor em (4.109) é quadrático em  $h$ .

Métodos de ordem mais alta podem ser obtidos usando-se mais equações do sistema (4.108). Em particular, esquemas de quarta-ordem são comumente utilizados em função de sua precisão. Assim como no caso acima, existem infinitas possibilidades para escolher os valores das constantes. O chamado Método Clássico de Runge-Kutta de quarta-ordem é dado por:

$$x_{i+1} = x_i + \left[ \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \right] h, \quad (4.120)$$

onde

$$\begin{cases} k_1 = X(t, x), \\ k_2 = X\left(t + \frac{1}{2}h, x + \frac{1}{2}hk_1\right), \\ k_3 = X\left(t + \frac{1}{2}h, x + \frac{1}{2}hk_2\right), \\ k_4 = X(t + h, x + hk_3). \end{cases} \quad (4.121)$$

#### 4.7. Smoothed Particle Hydrodynamics

Os fundamentos do método *SPH* (*Smoothed Particle Hydrodynamics*) estão na teoria de interpolação. Neste método, parte-se de uma função suave  $W$ , denominada núcleo de suavização (*smoothing kernel*), a qual deve satisfazer as seguintes propriedades:

A função  $W$  deve ser normalizada:

$$\int W(x - x', h) dx' = 1$$

onde a constante  $h$  é usada para definir o suporte do núcleo de suavização.

Suporte compacto:

Em geral, o suporte de  $W$  é definido por:

$$W(x - x', h) = 0 \quad \text{se } |x - x'| \geq kh, \quad \text{onde } k \text{ é um fator de escala.}$$

$$W(x - x', h) \geq 0 \quad \text{se } |x - x'| \leq kh,$$

O núcleo de suavização deve ser monotonicamente decrescente

A função de suavização  $W$  deve se comportar como a função Delta de Dirac

para  $h \rightarrow 0$ :

$$\lim_{h \rightarrow 0} W(x - x', h) = \delta(x - x').$$

A função  $W$  deve ser uma função par e suficientemente suave.

Partido destas propriedades, dada uma função escalar  $f$ , o método SPH procura aproximar esta função em pontos de interesse, partindo da representação da mesma, obtida via o núcleo de suavização  $W$ :

$$\langle f(x) \rangle = \int f(x')W(x-x',h)dx', \quad (4.122)$$

Supondo agora que a função  $f(x)$  é conhecida apenas em alguns pontos  $x_1, x_2, \dots, x_N$  do seu domínio, teremos:

$$f(x') = \sum_{j=1}^N \delta(x' - x_j) f(x_j) (dx)_j, \quad (4.123)$$

Inserindo esta expressão na equação (4.122), obtemos:

$$\langle f(x) \rangle = \sum_{j=1}^N \int \delta(x' - x_j) f(x_j) (dx)_j W(x-x',h) dx' \quad (4.124),$$

resultando:

$$\langle f(x) \rangle = \sum_{j=1}^N f(x_j) (dx)_j W(x-x_j, h), \quad (4.125)$$

O passo final é substituir o elemento de volume por:

$$(dx)_j = \frac{m_j}{\rho(r_j)},$$

obtendo:

$$\langle f(x) \rangle = \sum_{j=1}^N \frac{m_j}{\rho(r_j)} f(x_j) W(x-x_j, h), \quad (4.126)$$

Expressões para as derivadas de quantidades escalares e vetoriais podem ser analogamente obtidas. Em particular, o gradiente de  $f$  pode ser estimado pela seguinte expressão:

$$\langle \nabla_x f(x) \rangle = \int \nabla_{x'} f(x') W(x-x', h) dx', \quad (4.127)$$

onde  $\nabla_{x'}$  indica o gradiente com relação as coordenadas  $x'$ . Integrando por partes, teremos:

$$u = W(x-x', h) \Rightarrow du = \nabla_{x'} W(x-x', h) dx', \quad (4.128)$$

$$dv = \nabla_{x'} f(x') dx' \Rightarrow v = f(x'), \quad (4.129)$$

Obtemos, portanto:

$$\langle \nabla_x f(x) \rangle = f(x') W(x-x', h) \Big|_{Fronteira} + \int f(x') \nabla_{x'} W(x-x', h) dx'. \quad (4.130)$$

Lembrando a condição (2) acima, observamos que a primeira parcela se anula, gerando:

$$\langle \nabla_x f(x) \rangle = \int f(x') \nabla_{x'} W(x - x', h) dx', \quad (4.131)$$

O próximo passo é substituir as derivadas  $\nabla_{x'}$  por  $\nabla_x$  no integrando. Isto pode ser feito observando-se que as restrições (1)-(6) acima são satisfeitas se:

$$W(x - x', h) = f(\|x - x'\|, h), \quad (4.132)$$

portanto:

$$\nabla_{x'} W(x - x', h) = \left[ \frac{-(x_1 - x'_1)}{\|x - x'\|} \frac{\partial f}{\partial x'_1}, \frac{-(x_2 - x'_2)}{\|x - x'\|} \frac{\partial f}{\partial x'_2}, \frac{-(x_3 - x'_3)}{\|x - x'\|} \frac{\partial f}{\partial x'_3} \right], \quad (4.133)$$

Comparando esta expressão com aquela análoga obtida tomando-se as derivadas com relação à  $x$ , é imediato que:

$$\nabla_{x'} W(x - x', h) = -\nabla_x W(x - x', h), \quad (4.134)$$

conseqüentemente:

$$\langle \nabla_x f(x) \rangle = \int f(x') \nabla_x W(x - x', h) dx', \quad (4.135)$$

onde  $\nabla_x$  indica o gradiente tomada com relação à  $x$ . Utilizando novamente a expressão (4.123) obteremos finalmente:

$$\langle \nabla_x f(x) \rangle = \sum_{j=1}^N \frac{m_j}{\rho(r_j)} f(x_j) \nabla_x W(x - x_j, h). \quad (4.136)$$

Esta expressão tem o inconveniente de não ser simétrica, o que pode ser verificado para o caso de apenas duas partículas interagindo ( $x_1$  e  $x_2$ ). Neste caso, uma vez que o gradiente  $\nabla_x W(r, h) = 0$  se  $r = 0$  (conseqüência das Propriedades 4 e 6 acima), vemos que gradiente  $\langle \nabla_x f(x = x_1) \rangle$  será dado por:

$$\langle \nabla_x f(x = x_1) \rangle = \sum_{j=1}^2 \frac{m_j}{\rho(r_j)} f(x_j) \nabla_x W(x - x_j, h) = \frac{m_2}{\rho(r_2)} f(x_2) \nabla_x W(x - x_2, h), \quad (4.137)$$

ou seja, depende apenas da posição da segunda partícula (idem para  $\langle \nabla_x f(x = x_2) \rangle$ ). Portanto, a quantidade  $\langle \nabla_x f(x) \rangle$  não será simétrica, de maneira geral. Para evitar possíveis instabilidades que possam ser geradas em função da assimetria da expressão (4.136), diferentes propostas foram feitas no sentido de obter uma versão simétrica da mesma. A proposta a seguir tem se mostrado eficiente para fins de animação de fluidos em computação gráfica:

$$\langle \nabla_x f(x_i) \rangle = \sum_{j=1}^N m_j \left( \frac{f(x_j) + f(x_i)}{\rho(r_j)} \right) \nabla_x W(x - x_j, h), \quad (4.138)$$

As expressões (4.136) e (4.138) ressaltam o fato já comentado no final da seção 4.2 de que o método *SPH* não faz uso de malhas para estimar as derivadas. Esta é uma de suas vantagens em relação a métodos de diferenças ou elementos finitos, e que vem sendo explorada em computação gráfica.

Um desenvolvimento análogo àquele utilizado para obter a expressão (4.136) pode ser usado para aproximar o Laplaciano  $\langle \nabla_x^2 f(x) \rangle$  da forma:

$$\langle \nabla_x^2 f(x) \rangle = \sum_{j=1}^N \frac{m_j}{\rho(r_j)} f(x_j) \nabla_x^2 W(x - x_j, h), \quad (4.139)$$

No caso particular de um campo de velocidades  $\bar{v}$ , o Laplaciano será utilizado para estimar as forças oriundas da viscosidade do fluido (equações (2.18) e (2.19)). Uma vez que estas forças dependem de diferenças entre velocidades, e não do valor absoluto da velocidade, uma forma correta do ponto de vista físico, e eficiente do ponto de vista computacional para tornar simétrica a equação (4.139), é dada por:

$$\langle \nabla_x^2 f(x) \rangle = \sum_{j=1}^N m_j \left[ \frac{f(x) - f(x_j)}{\rho(r_j)} \right] \nabla_x^2 W(x - x_j, h), \quad (4.140)$$

O método *SPH* tem seu campo mais forte de aplicações na área de simulação de fluidos compressíveis, o que pode trazer perspectivas interessantes na aplicação do mesmo como um método alternativo àquele usado em [Witting (1999)], o qual será descrito na seção 6.4. Para isto, as equações de fluidos são discretizadas, substituindo-se cada termo da equação por sua versão obtida via núcleo de suavização. Desta forma, obtém-se um sistema de equações onde as variáveis desconhecidas são os valores das quantidades de interesse em pontos amostrais de domínio. Em [Muller (2003)] encontramos um exemplo da aplicação do *SPH* na animação de fluidos para computação gráfica, o qual também será discutido no Capítulo 6.

## Capítulo 5

### Técnicas para Visualização de Fluidos

#### 5.1. Introdução

Neste capítulo apresentamos um resumo das principais técnicas utilizadas para visualização de dados em DFC.

Duas etapas se distinguem no desenvolvimento de softwares para visualização científica em DFC: o desenvolvimento de técnicas específicas de visualização e a integração destas técnicas em um sistema para visualização de dados [Rosemblum et al. (1994)]. Neste trabalho, estamos interessados no primeiro aspecto. No decorrer desse capítulo faremos uma análise geral das principais técnicas nesta área.

Podemos distinguir as varias técnicas de visualização de campos em DFC sob diversos aspectos [Hesselink et al. (1994)]. Um critério bem simples pode ser definido a partir da ordem dos dados a serem visualizados: escalares, vetoriais ou tensoriais. A análise da dimensão espacial dos objetos fornece outro critério, distinguindo dados 0D (pontos), 1D (linhas / curvas), 2D (superfícies) ou 3D (volumes). O conceito de “nível de informação”, que se refere à abrangência da informação associada a um ponto do espaço de interesse. Esses níveis podem ser: elementar, local ou global. O nível elementar indica que a informação é restrita a somente aquele ponto. O nível local pode indicar o comportamento de um gradiente em uma região do espaço, enquanto que o nível global representa informações que afetam o conjunto de dados globalmente. A tabela 5.1 classifica as principais técnicas de visualização em DFC de acordo com estes critérios.

As técnicas para a visualização de campos escalares (visualização direta (volume rendering) e isosuperfícies) são largamente utilizadas na visualização científica e serão discutidas na próxima seção. A visualização por isosuperfícies [Carneiro et al. (1996)] é um recurso valioso na visualização de campos escalares. No caso bidimensional, em particular, o traçado de isolinhas, em geral, fornece boa noção do comportamento do campo. No caso tridimensional a sobreposição visual das isosuperfícies impõe uma dificuldade adicional a visualização. Nesse caso pode ser necessária a aplicação de rotações aos dados ou mesmo uma mudança de ponto de vista. O uso de isosuperfícies pode ser enriquecido pela utilização conjunta de recursos para visualização de campos vetoriais ou mesmo pela utilização de traçado de raios para mapear o comportamento de um campo escalar sobre uma isosuperfície de um outro campo escalar.

*Volume rendering* [Max (1995)] é uma técnica que é largamente empregada para visualização de campos escalares estacionários, podendo ser estendida também para campos não estacionários [Rosemblum et al. (1994)]. O modelo usado

procura extrair características de um campo tridimensional usando “partículas virtuais” que o atravessam (espalhamento).

Técnica	Ordem dos Dados	Domínio	Nível
Rendering de Volume	Escalar	Volume	Elementar
Isosuperfície	Escalar	Superfície	Elementar
Setas	Vetor	Ponto	Elementar
Traçado de Linhas	Vetor	Linha	Elementar
Superfícies de Corrente	Vetor	Superfície	Elementar
Partículas	Vetor	Ponto	Elementar
LIC	Vetor	Linha	Elementar
Tensor Probe	Vetor	Ponto	Local
Topologia de Campos Vetoriais	Vetor	Volume	Global
Vórtices	Vetor/Escalar	Volume	Local
Hyperstreamlines	Tensor	Linha	Elementar

Tabela 1. Classificação das principais técnica para visualização em DFC.

Uma dificuldade básica com estas técnicas de *volume rendering* é a grande massa de dados a ser analisada. A utilização direta destas técnicas pode ter um elevado custo computacional, uma vez que muitos cálculos podem ser necessários para visualizar o comportamento do fluido em regiões de pouco interesse. Uma solução para esse problema é adicionar uma etapa de filtragem da massa de dados para a seleção prévia das regiões de interesse.

Para os campos vetoriais, as técnicas de nível elementar na tabela 1 (setas, linhas de corrente, superfícies de corrente e traçado de partículas) são muito convenientes em situações onde o campo não apresenta complexidade elevada.

A utilização de símbolos (setas) é uma técnica muito comum e tem a grande vantagem de compactar em um único símbolo a direção, sentido e magnitude do campo em um ponto. No entanto, sua limitação está na representação de campos vetoriais 3D devido a inevitável sobreposição de setas em regiões de maior complexidade do campo. Outro ponto sensível dessa técnica diz respeito a sua ambigüidade em relação as noções de direção e magnitude do campo decorrentes da projeção plana utilizada para a representação no espaço da tela (janela 2D).



Dentre as técnicas baseadas em traçado de linhas a mais comum é o traçado de linhas de corrente. Linhas de corrente são linhas tangentes ao campo de velocidades em um instante específico [Roseblum et al. (1994)]. No projeto de uma asa de avião, por exemplo, a utilização de um conjunto de linhas deste tipo, cada uma delas partindo de um ponto de partida convenientemente escolhido, pode fornecer uma boa noção do comportamento do ar nas proximidades de fuselagem. Outro exemplo pode ser identificado no projeto de turbinas, onde podemos utilizar duas linhas de corrente conectadas por um malha de polígonos (*streamribbons*) para representar a rotação do fluido em uma região [Ueng et al. (1996)].

A utilização de partículas está baseada na liberação de partículas de prova no interior do fluido. Cada partícula fica sujeita à dinâmica do fluido [Roseblum et al. (1994)]. O acompanhamento do movimento dessas partículas nos fornece uma boa noção do comportamento do campo. Uma extensão desta técnica, baseada em partículas de superfície, procura fazer uso também das vantagens das técnicas baseadas em superfícies (superfícies de corrente, etc). Neste caso as partículas possuem propriedades geométricas (vetor normal) e ópticas e a variação da posição e geometria das mesmas é calculada a cada instante, realçando comportamentos de interesse [Roseblum et al. (1994)]. Caso o número de partículas seja muito elevado, a distinção entre as partículas individualmente deixa de existir, dando a noção não mais de um conjunto de elementos mas sim de uma textura.

A utilização de texturas para visualização de dados é o conceito básico da técnica de *LIC* [Cabral-Leedom (1993)] da tabela 1. Neste caso, se faz uma convolução entre o campo e uma textura previamente estabelecida. A técnica de *LIC* é usada para visualização de resultados em aerodinâmica e apresenta bons resultados em casos estacionários podendo ser estendida para campos dependentes do tempo [Forsell-Cohen (1995)].

Para que técnicas baseadas em texturas apresentem bons resultados é necessário que as propriedades da textura possam ser parametrizadas pelos dados. Essa parametrização deve ser tal que a textura resultante seja capaz de representar as propriedades de interesse relativas ao fluido, tais como direção do fluxo e intensidade do campo.

A visualização da topologia de campos vetoriais [Hermann-Hesselink (1991)] é a única técnica de nível global da tabela 1, e está baseada na teoria de pontos críticos das equações diferenciais ordinárias. A topologia de um campo vetorial consiste de seus pontos críticos (pontos onde o campo se anula) e curvas e superfícies integrais conectando estes pontos. Os resultados obtidos por essa técnica são particularmente interessantes por realçar as características básicas do escoamento (sua estrutura topológica), descartando as informações redundantes, facilitando, portanto, a análise dos resultados. Esta técnica é particularmente interessante para os escoamentos estacionários 2D, onde a topologia do campo é totalmente descrita pelos pontos críticos e um conjunto conveniente de linhas integrais. Para os caso 3D deve-se calcular também superfícies integrais, o que traz complicações adicionais, principalmente quando o campo é não-estacionário.

Outras técnicas como as hiper linhas de corrente (*Hyperstreamlines*) [Hesselink et al. (1997)] e *Tensor Probe* [Roseblum et al. (1994)] podem ser usadas. No primeiro caso, a idéia é generalizar a noção de linha de corrente para campos tensoriais simétricos, via a utilização do campo de autovetores associado. No segundo caso, o objetivo é obter uma representação compacta das informações que um tensor pode carregar, tais como rotação e tensões internas (Tensor de Deformação). Por serem técnicas pouco expressivas na área e pela dificuldade de serem aplicadas para campos mais complexos (não-estacionários) não abordaremos estas técnicas neste trabalho.

A grande dificuldade na visualização de campos não-estacionários está na variação global da topologia do campo ao longo do tempo, além da grande massa de dados a ser analisada. Estas dificuldades assumem um caráter crítico no estudo de escoamentos turbulentos.

Escoamentos turbulentos ocorrem, em geral, quando a velocidade do escoamento, ou mais precisamente, o número de Reynolds, excede um valor crítico. O número de Reynolds é definido como o produto das escalas representativas da velocidade e comprimento dividido pela viscosidade cinemática [Landau-Lifshitz (1959)]. A forma particular de instabilidade gerada no regime de escoamento turbulento é caracterizada pela presença de flutuações estatísticas de todas as quantidades físicas do fluxo [Hirsch (1988)]. Tais flutuações criam uma riqueza de comportamento difícil de ser capturada pelas técnicas citadas até aqui.

Nestes casos pode ser mais interessante buscar estruturas ou padrões significativos no interior do fluido. Um exemplo de tais estruturas são os *vórtices*, caracterizados por regiões de rotação no interior do fluido. Outros exemplos são as ondas de choque [Roseblum et al. (1994)] e a própria visualização de topologias de campo, embora esta seja muito limitada para o caso da turbulência devido mudanças radicais da topologia, decorrentes das flutuações comentadas no parágrafo anterior.

A grandeza física fundamental no estudo dos vórtices é a *vorticidade* (rotacional do vetor velocidade, seção A.4 do Apêndice A). As leis relativas a sua conservação e dispersão também constituem objeto de interesse dos cientistas nesta área [Saffman (1995)].

As técnicas para identificação de vórtices no volume de dados gerado pela simulação são baseadas em heurísticas, fundamentadas em conceitos da dinâmica de fluidos [Banks-Singer(1995)]. Estas técnicas constituem um exemplo singular da aplicação conjunta de métodos para visualização de dados científicos. Por isso são descritas na última seção deste capítulo.

## **5.2. Visualização de Campos Escalares**

### **5.2.1 Isosuperfícies**

No caso específico da mecânica de fluidos temos basicamente dois campos escalares: o campo de pressão e a densidade volumétrica de massa  $\rho$ . Podemos juntar a estes os campos definidos pelo módulo da vorticidade e da velocidade.

Os métodos mais comumente utilizados para a visualização destes campos são baseados em isosuperfícies e *volume rendering*.

No caso das técnicas baseadas em isosuperfícies, a idéia é gerar uma aproximação poligonal da superfície (isosuperfície) definida por uma equação do tipo:

$$F(x_1, x_2, x_3) = \lambda \quad (5.1)$$

onde  $F$  é o campo escalar  $F: \Delta \rightarrow \mathfrak{R}$ , com  $\Delta \subset \mathfrak{R}^3$  e  $c$  é uma constante.

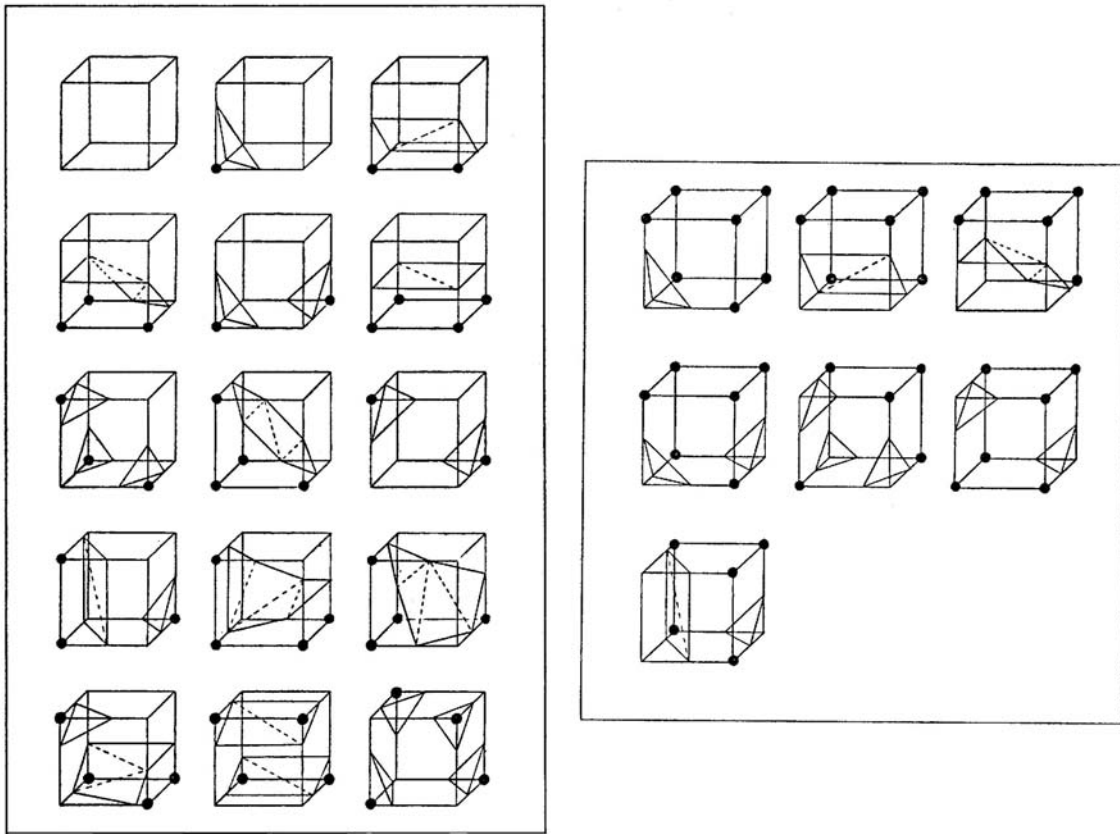
Nas aplicações de nosso interesse, os campos escalares estão discretizados e, portanto, a solução de (5.1) será obtida via uma busca dos pontos  $(x_1, x_2, x_3)$  da malha que aproximem suficientemente a superfície procurada.

As soluções encontradas na literatura para este problema estão vinculadas ao tipo de malha utilizado na discretização.

Para malhas regulares de hexaedros, um algoritmo comumente usado na geração de isosuperfícies é o algoritmo dos *cubeos marchantes* [Lorensen (1987)]. Vejamos uma descrição sucinta deste algoritmo.

Inicialmente, consideremos um cubo da malha e façamos uma enumeração binária dos seus vértices da seguinte forma: é dado o valor "1" para os vértices cujos valores do campo forem maiores ou iguais ao valor  $c$  da isosuperfície procurada e "0" para os demais. Assim, as arestas dos hexaedros que tiverem um vértice com valor "1" e outro com valor "0" são aquelas cortadas pela superfície procurada.

Uma vez que um cubo tem 8 vértices, temos um total de  $2^8 = 256$  configurações possíveis. Diremos que duas configurações são equivalentes quando uma for idêntica à outra a menos de rotações simples. Se considerarmos apenas as classes de equivalência correspondentes, obteremos um total de 22 configurações básicas, como mostra a Figura 5.1.



*Figura 5.1- Configurações básicas para a intersecção entre um cubo e uma superfície.*

Cada uma destas configurações será representada por um índice de 8 bits construído a partir dos estados dos vértices correspondentes.

Cada um destes índices será um ponteiro em uma tabela que fornece todas as arestas cortadas pela superfície para cada configuração básica. Tal procedimento pressupõe uma enumeração dos vértices e das arestas do cubo, tal como aquela apresentada na Figura 5.2.

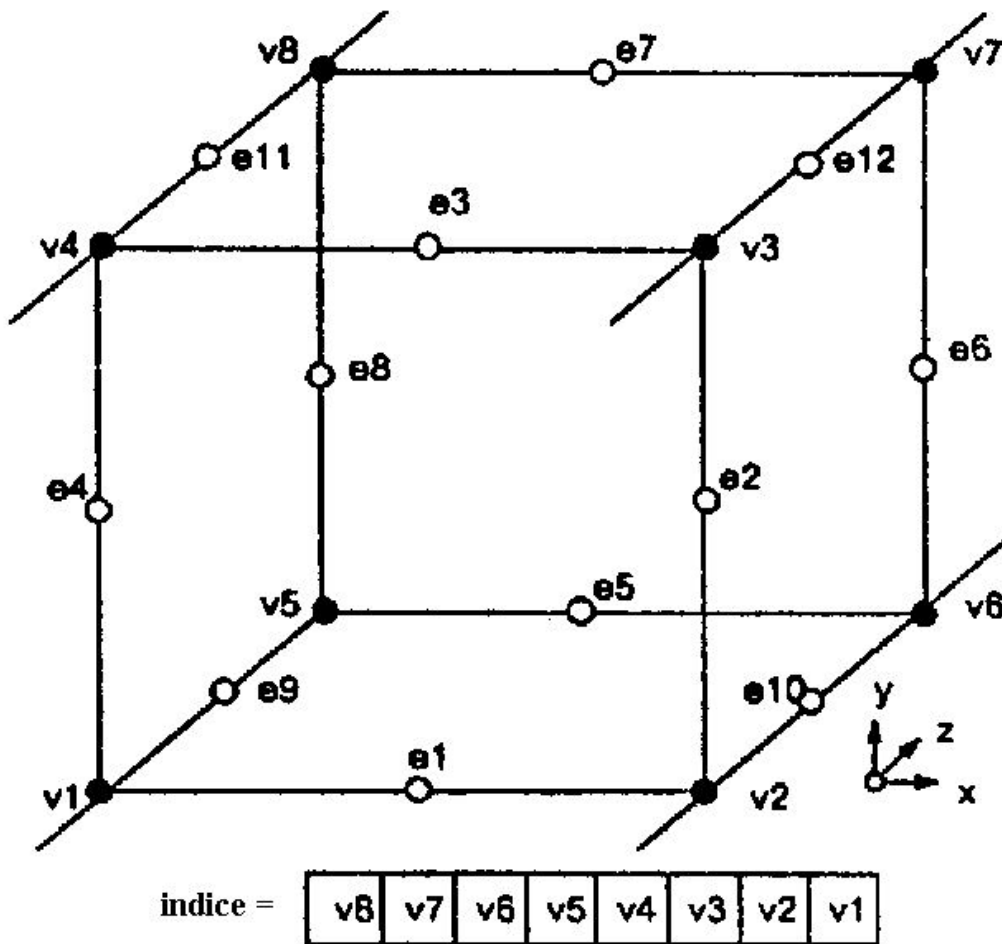


Figura 5.2 – Enumeração das arestas e lados do cubo juntamente com o índice da configuração correspondente.

Para os casos 0, 1 e 2 da Figura 5.1, por exemplo, as linhas da tabela têm a seguinte forma:

Índice								Arestas Cortadas
0	0	0	0	0	0	0	0	Nenhuma aresta é cortada
0	0	0	0	0	0	0	1	e1, e4, e9
0	0	0	0	0	0	1	1	e2, e4, e9, e10

Figura 5.3 – Índice da configuração básica e arestas para os casos 0, 1 e dois da Figura 5-1, seguindo a enumeração dos vértices e arestas da Figura 5-2.

Procedendo analogamente, podemos montar a tabela para os 22 casos básicos da Figura 5.1. Esta tabela será fundamental para o cálculo dos polígonos que concatenados definirão a superfície.

Tomemos um cubo qualquer da malha. Uma vez analisados os valores de campo em seus vértices, temos uma das 256 possíveis configurações para o mesmo e podemos identificar a configuração básica correspondente.

Feito isto, teremos o índice correspondente na tabela acima e conseqüentemente as arestas cortadas. Basta então tomar os valores de campo nos vértices destas arestas para, via interpolação, estimar os pontos de interseção sobre as mesmas. Estes pontos definirão um polígono que aproximará a superfície no interior do cubo. Uma vez executado este procedimento para todos os cubos da malha teremos uma aproximação poligonal da superfície em questão.

A estimativa do vetor normal em cada vértice da superfície poligonal gerada pelo algoritmo de cubos marchantes é necessária para uma visualização da superfície onde as transições entre faces não se tornem aparentes. Ou seja, para que a superfície gerada possa ser visualizada como uma superfície suave, é necessário calcular o vetor normal em cada vértice da malha poligonal, o que pode ser feito por diferenças finitas. Outra possibilidade é ajustar uma superfície paramétrica (*spline*, por exemplo) aos vértices da superfície poligonal. Essa abordagem pode ser interessante do ponto de vista da qualidade e controle da superfície gerada, porém seu custo computacional pode ser elevado para grandes massas de dados.

O método de geração de isosuperfícies apresentado acima deixa aparente uma limitação desta técnica para visualização em grandes massas de dados: o número de polígonos formando a superfície pode se tornar muito alto trazendo problemas de performance para a etapa de visualização.

Quando isto acontece, podemos lançar mão de técnicas para simplificação de superfícies poligonais [Cignoni et al.(1994)],[Cohen et al. (1996)] o que por sua vez implica em problemas referentes à preservação da topologia das isosuperfícies e possível perda de detalhes importantes.

Por outro lado, técnicas baseadas em isosuperfícies têm a vantagem de serem robustas e apresentam boa performance para massas de dados de pequeno e médio porte.

## 5.2.2 Volume Rendering

Uma outra possibilidade para a visualização de um campo escalar tridimensional é o *rendering* direto de volume; ou seja, produzir uma projeção plana diretamente do volume de dados.

Para isto, uma possibilidade é usar técnicas baseadas na idéia de raios de luz atravessando o volume e interagindo com este de acordo com algum modelo físico [Max (1995)]. Estes modelos simulam os fenômenos básicos de interação da luz com a matéria: reflexão, espalhamento, absorção e geração de luz pelo meio.

Em [Krueger (1991)] encontramos uma outra formulação baseada na teoria de transporte linear que inclui os modelos baseados em traçados de raios como casos particulares ou aproximações. Esta abordagem é fundamentada na idéia de “partículas virtuais” que atravessam o conjunto de dados, carregando consigo informações sobre o mesmo.

A informação base deste modelo é a intensidade  $I(x, \vec{s}, E)$  que descreve o número de partículas em um ponto  $x$  que se movem em uma direção  $\vec{s}$  com energia  $E$ . No espaço de cores, a intensidade de um *pixel* é dada por valores médios:

$$I_i(x, \vec{s}) = \int_E I_i(x, \vec{s}, E) dE,$$

com  $i$  correspondendo às intensidades de cada canal de cor do sistema RGB.

A equação básica para a teoria de transporte linear é a equação de Boltzmann, que descreve os ganhos e perdas da intensidade das partículas em um elemento de volume [Case-Zweifel (1967)]. Para o caso estacionário, uma forma conveniente desta equação para a visualização de campos escalares tridimensionais é dada por:

$$\begin{aligned} (\vec{s} \cdot \vec{\nabla}) \cdot I(x, \vec{s}; E) = & -\sigma_t(x; E) \cdot I(x, \vec{s}; E) + q(x, \vec{s}; E) + \\ & \sigma_s(x; E) \int d\omega' \cdot p(x, \vec{s}' \rightarrow \vec{s}) \cdot I(x, \vec{s}'; E) \end{aligned} \quad (5.2)$$

onde  $d\omega'$  é o elemento de ângulo sólido em torno da direção dada pelo vetor unitário  $\vec{s}'$ .

Os parâmetros relevantes para o processo de visualização são o coeficiente de extinção  $\sigma_t$  e o termo fonte  $q$ . O coeficiente de extinção é dado por:

$$\sigma_t = \sigma_a + \sigma_s,$$

onde  $\sigma_a$  e  $\sigma_s$  são os coeficientes de absorção e espalhamento, respectivamente.

A função  $p(x, \vec{s}' \rightarrow \vec{s})$  é a fase de espalhamento normalizada que modela a mudança na direção ( $\vec{s}' \rightarrow \vec{s}$ ) do raio de partículas.

### Mapeamento das Características dos Dados nos parâmetros de visualização

A equação (5.2) só tem sentido prático se estabelecermos regras eficientes para definir os parâmetros  $\sigma_t$ ,  $\sigma_a$  e  $\sigma_s$  em função de grandezas facilmente deriváveis do campo  $F(x)$  a ser visualizado. Nesta discussão vamos assumir que  $F$  é um campo normalizado:

$$F: \Delta \rightarrow [0,1].$$

Iniciemos com a análise do termo fonte  $q$ .

#### **Termos Fonte**

O termo  $q(x, \vec{s}; E) \geq 0$  em (5.2) atua como uma fonte interna para a intensidade de partículas  $I$ . De acordo com o seu suporte espacial  $q$  pode ser classificado como uma fonte pontual, superficial ou volumétrica.

Para o caso das fontes volumétricas, poderíamos usar a seguinte regra:

$$q_v(x, \vec{s}; E) = cF(x) \quad (5.3)$$

onde  $F$  é o campo (normalizado) e  $c > 0$  é uma constante de normalização da intensidade  $I$ .

Se considerarmos apenas este termo no lado direito da equação (5.2), esta tomará a forma:

$$(\vec{s} \cdot \vec{\nabla}) \cdot I(x, \vec{s}; E) = q(x, \vec{s}; E) \quad (5.4)$$

Uma vez que não há espalhamento, temos uma situação onde a velocidade das partículas não sofre mudança em direção. Durante seu trajeto, uma partícula apenas ganha ou perde energia, numa taxa caracterizada pela equação (5.4).

Assim, usando-se um conjunto conveniente de raios (paralelos, por exemplo) obteremos um campo de intensidades  $I$  que poderia ser mapeado num mapa de cores fornecendo assim uma visualização do campo  $F$ . Este modelo é interessante pela sua simplicidade, mas limita-se à visualização da distribuição espacial do campo e seu decaimento. A Figura 5.4 mostra o resultado para a visualização da densidade de elétrons de átomos altamente excitados.

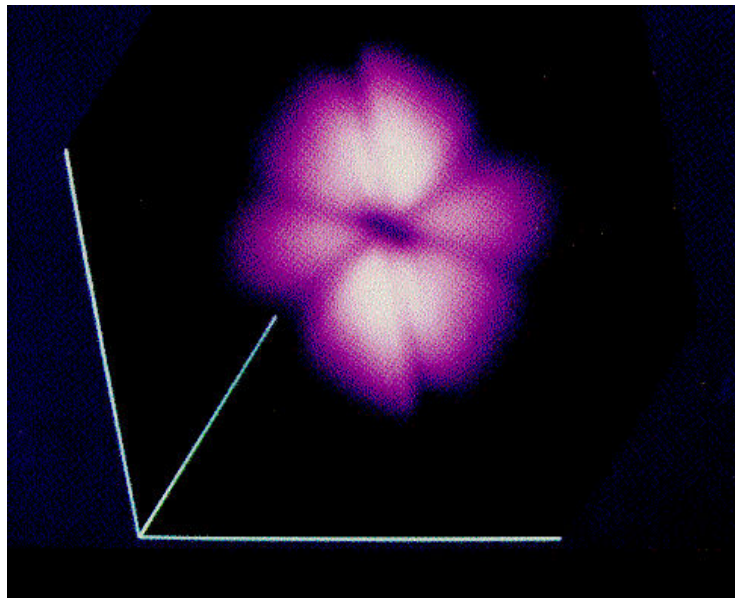


Figura 5.4 – Efeito usando apenas o termo fonte  $q_v$ .

Este modelo pode também ser utilizado para a visualização de isosuperfícies na forma  $F(x) = \lambda$ , via a definição de fontes superficiais  $q_s(x, \vec{s}, E)$ . Uma



possibilidade direta para a definição destas fontes pode ser obtida a partir da definição (5.3) modificada:

$$q_s(x, \vec{s}; E) = c(\vec{s} \cdot \vec{n}_{surf}) \cdot F(x_{surf}) \quad (5.5)$$

onde  $\vec{n}_{surf}$  é o vetor normal à isosuperfície em um ponto  $x_{surf}$  da mesma.

Outra possibilidade também empregada é usar o gradiente do campo F:

$$q_s(x, \vec{s}; E) = c(\vec{s} \cdot \vec{n}_{surf}) \cdot \|\vec{\nabla} F(x_{surf})\| \quad (5.6)$$

Esta última definição é particularmente interessante em situações onde a intensidade do campo  $F$  apresenta uma variação brusca em uma vizinhança da isosuperfície procurada, o que é o caso freqüentemente encontrado em imagens médicas quando temos a passagem do tecido ósseo para os tecidos adjacentes. A Figura 5.5 mostra o resultado para o mesmo campo da figura 5.4.

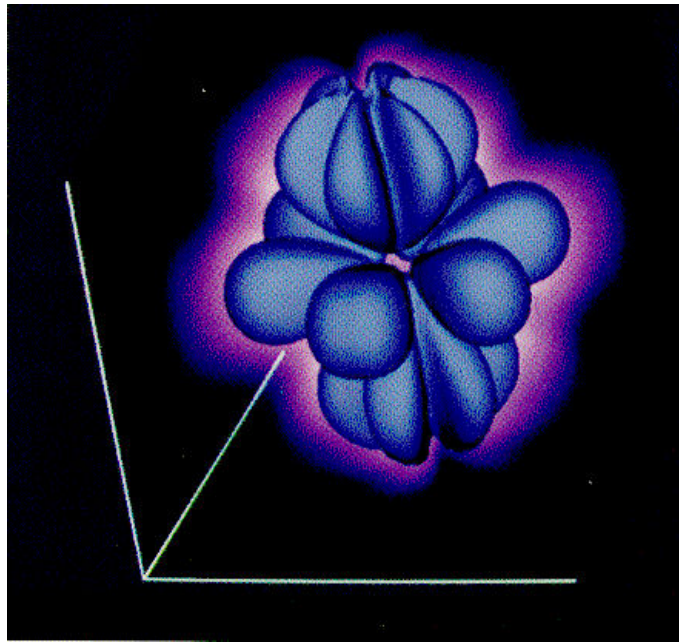


Figura 5.5 – Visualização de isosuperfícies usando o termo fonte  $q_s$  para os dados da figura 5.4.

### **Termo de absorção**

O termo de absorção na equação de transporte (4.2) causa uma atenuação exponencial na intensidade  $I$ . Considerando apenas este termo no lado direito de (4.2) obteremos o modelo básico da atenuação sofrida pelo raio-X ao atravessar um material. Analogamente ao item anterior, este termo não introduz mudanças na direção de movimento das partículas.

As duas escolhas mais comuns para a definição deste termo são dadas por:

$$\sigma_a(x; E) = cF(x) \quad (5.7)$$

OU:

$$\sigma_a = \|\vec{\nabla}F(x)\| \quad (5.8)$$

A Figura 5.6 mostra o efeito deste termo, juntamente com uma fonte superficial do tipo (5.5), para o campo da figura anterior.

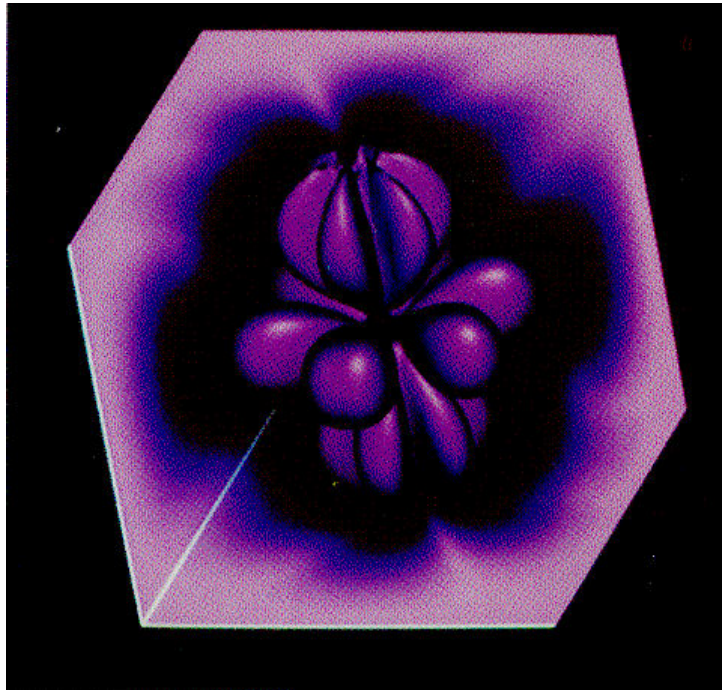


Figura 5.6 – Aparência dada pelo termo de absorção ao campo da Figura 5.5.

### **Termos de Espalhamento**

A exploração do termo de espalhamento  $\sigma_s$  leva a algoritmos computacionais mais elaborados, por exemplo, o Método de Monte Carlo [Hammersley (1964)]. Neste caso, temos mudanças na direção do movimento das partículas (de  $\vec{s}'$  para  $\vec{s}$ ).

Uma situação em que este termo pode ser incorporado é o caso de realce seletivo de flutuações locais da densidade de volume, o que pode ser modelado com o coeficiente de espalhamento volumétrico dado por:

$$\sigma_s^v(x) = cF(x) \quad (5.9)$$

e pela escolha de uma função para a fase de espalhamento apropriada:

$$P^v(x, \vec{s}' \rightarrow \vec{s}) = c_f \delta(\vec{s}' \cdot \vec{s} - 1) + (1 - c_f) P_s(x, \vec{s}' \cdot \vec{s}) \quad (5.10)$$

onde o primeiro termo modela o espalhamento avançado somente ( $\delta$  é a função Delta de Dirac) e  $P_s$  é uma função que modela a mudança de direção do raio de partículas [Krueger (1991)]. Este modelo é particularmente conveniente para a visualização de dados atmosféricos, e pode ser adaptado para incluir efeitos de superfície tais como reflexão especular e transparência. Por exemplo, podemos usar a seguinte expressão para  $P_s$  [Krueger (1991)]:

$$P_s(x, \vec{s}' \rightarrow \vec{s}) = \delta(x - x_{surf}) \cdot [c_f \cdot \delta(\vec{s}' \cdot \vec{s} - 1) + c_b \delta(\vec{s}' + \vec{s})], \quad c_f + c_b = 1,$$

onde  $x_{surf}$  é um ponto de uma superfície.

A Figura 5.7 mostra o resultado quando usamos a equação (5.2) com  $\sigma_s$  e  $p$  dados por (5.9) e (5.10), respectivamente, e  $P_s$  dado pela última expressão.

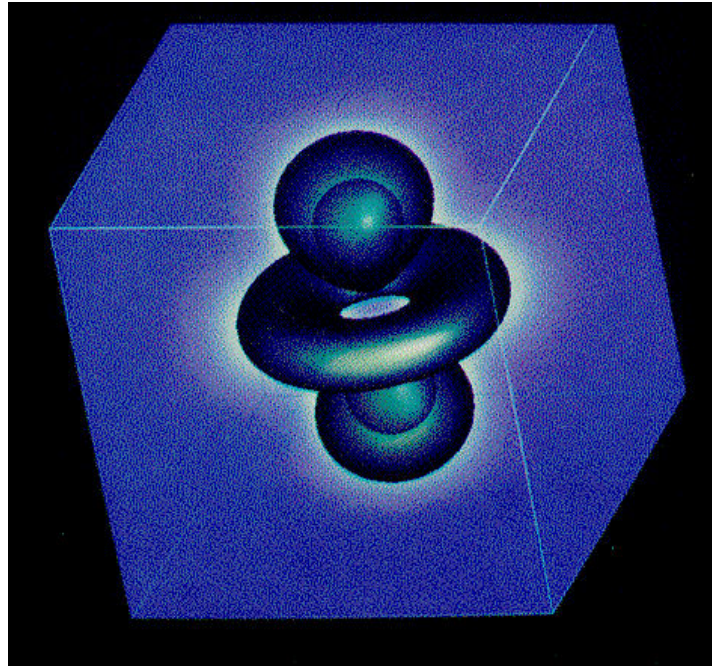


Figura 5.7 – Reflexão especular e transparência para realçar profundidade (densidade de elétrons do mesmo H-átomo, das figuras 5-4 à 5-6).

### 5.3. Splatting

A abordagem denominada *splatting* foi introduzida por Lee Westover [Westover (1989)] e tem sido usada como uma forma alternativa para visualizar, não só dados volumétricos, mas também modelos geométricos baseados em pontos [Zwicker (2001)].

### 5.3.1 Volume Splatting

Métodos para visualização volumétrica precisam produzir uma imagem bidimensional baseada na composição dos *voxels* que descrevem o volume. Em função de como o processo de composição é conduzido, podemos classificar os métodos de visualização volumétrica em duas categorias: composição “da-frente-para-trás” (*front-to-back compositing*) ou mapeamento “de-trás-para-frente” (*back-to-front compositing*). Como o nome já sugere, o que caracteriza cada classe é a ordem em que o processo de integração (ou convolução) dos *voxels* é feito. Podemos ver esquematicamente esses dois processos na Figura 5.8. No mapeamento *front-to-back* o processo de integração tem início no plano da imagem, onde para cada *pixel* um raio é disparado em direção ao espaço dos dados, como já descrito na seção 5.2.2. O mapeamento *back-to-front*, por sua vez, parte de cada voxel do espaço 3D, projetando-o no plano da imagem.

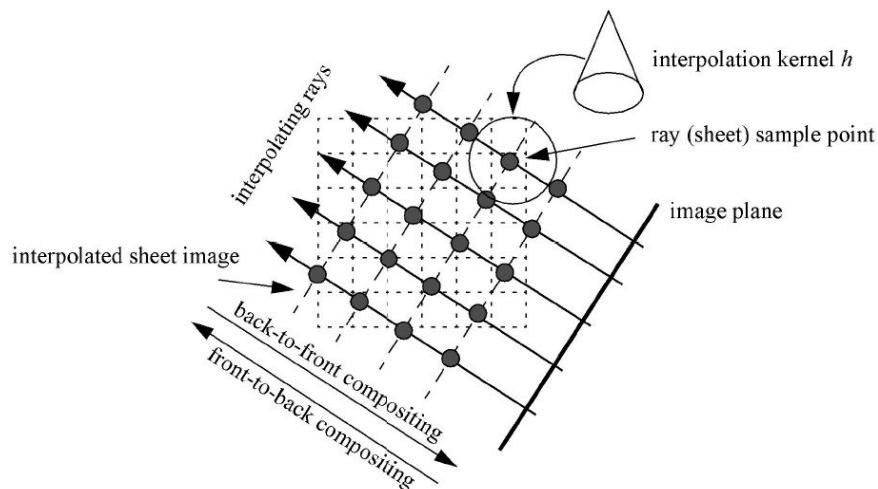


Figura 5.8 – Composição *front-to-back* e *back-to-front*.

Essa diferença deixa claro como e quando a reconstrução do sinal 3D é feita. No caso da abordagem *front-to-back* temos um processo envolvendo uma série de amostra de entrada (lista de todos os *voxels* interceptados pelo raio) gerando uma única amostra de saída (*pixel* da imagem). Já o processo *back-to-front* “espalha” cada a amostra de entrada (*voxel*) por várias amostras de saída (*pixels*). Esse processo de “espalhamento” da origem a técnica de *splatting*.

Um algoritmo típico de *splatting* para dados volumétricos envolve 4 passos básicos [Westover (1990)]: transformação, iluminação, reconstrução e visibilidade. Dado um *voxel*  $[i, j, k]$  (coordenadas inteiras da grade de amostras do volume) ele deve ser transformado para coordenadas  $(x, y, z)$  no espaço da tela. Uma vez no espaço da tela sua iluminação é determinada através de algum processo que associa cor e opacidade, tornando o ponto representado como  $(x, y, z, r, g, b, a)$ . A etapa seguinte é determinar que porção da imagem aquela amostra afeta e adicionar a contribuição do *voxel* a cada *pixel* da imagem final.

Associado a cada *voxel* um núcleo de reconstrução é definido. Esse núcleo é composto por uma função de suporte compacto e decrescente, tipicamente uma função Gaussiana, com variância vinculada ao raio de influência do núcleo.

É comum o processamento dos *voxels* ser feito por “folhas” (*sheets*) como podemos ver na Figura 5.8. Cada uma dessas folhas corresponde a um plano paralelo ao plano de projeção. Os *pixels* em cada folha são gerados pela composição da contribuição do *voxel* corrente com as contribuições dos *voxels* já processados. Dessa forma a ordenação dos *voxels* é conseguida de forma simples.

A projeção ou linha de integração contém a integral de todos os núcleos ao longo do raio que vem do infinito até o plano de visualização. Cada núcleo de reconstrução é centralizado no *voxel* e sua contribuição é acumulada na imagem a partir de sua projeção 2D chamada *splat*. Os *splats* são tipicamente pré-computados e re-amostrados em uma tabela de “pegadas” (*footprint table*), que é re-amostrada no *framebuffer* para cada *splat*. Para projeções ortogonais e núcleos de interpolação com simetria radial, uma única entrada nessa tabela pode ser calculada e aplicada para todos os *voxels*, a menos de um deslocamento. Podemos visualizar esse processo na Figura 5.9.

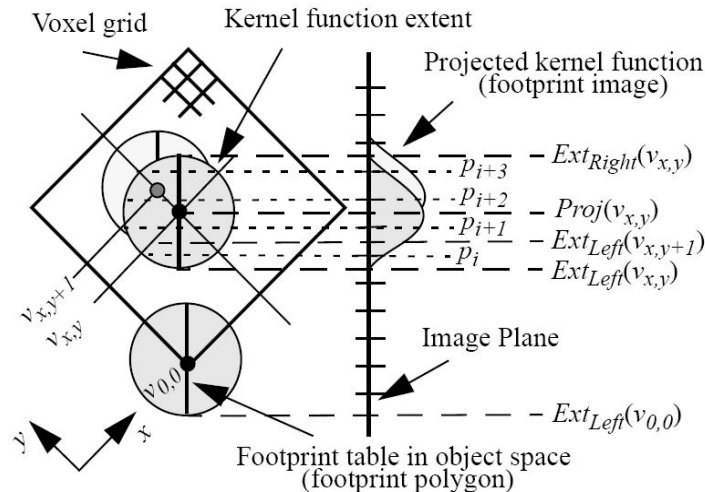


Figura 5.9 – Projeção Ortogonal dos núcleos associados a cada *voxel*.

Na Figura 5.10 podemos observar como vários *voxels* são responsáveis pelo valor final de um único *pixel*. Ao longo de um raio, com origem em um *pixel*, vários núcleos podem contribuir para a geração da imagem do *pixel*, mesmo sem que esses *voxels* sejam interceptados pelo raio.

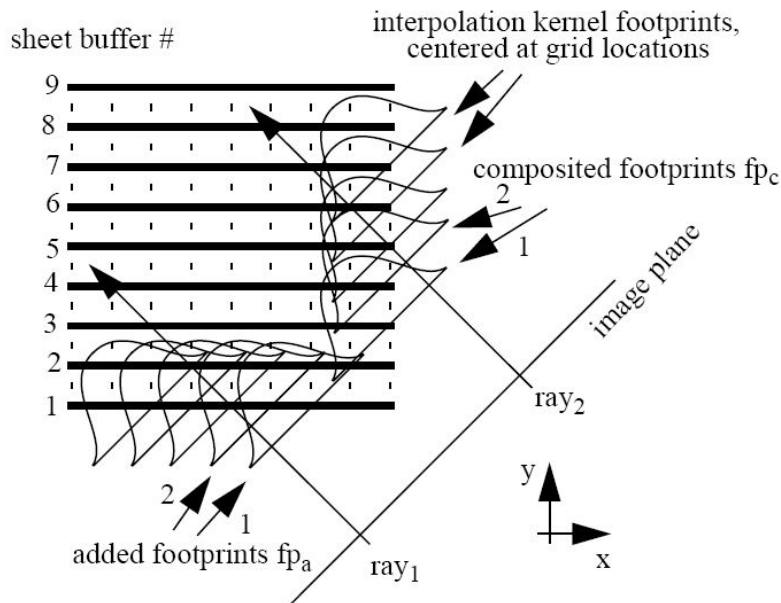


Figura 5.10 – Influência de vários núcleos associados aos voxels localizados ao longo de um mesmo raio (vinculado a um determinado pixel).

Vale ressaltar que em geral o processo de visualização usa como base a projeção paralela. O processo de *splatting* torna-se mais complexo no caso de um projeção em perspectiva. Nesse caso os núcleos de reconstrução sofrem um processo de deformação (*warping*) durante a projeção, o que torna mais difícil o uso de uma simples tabela de *footprints*, como mostra a Figura 5.11.

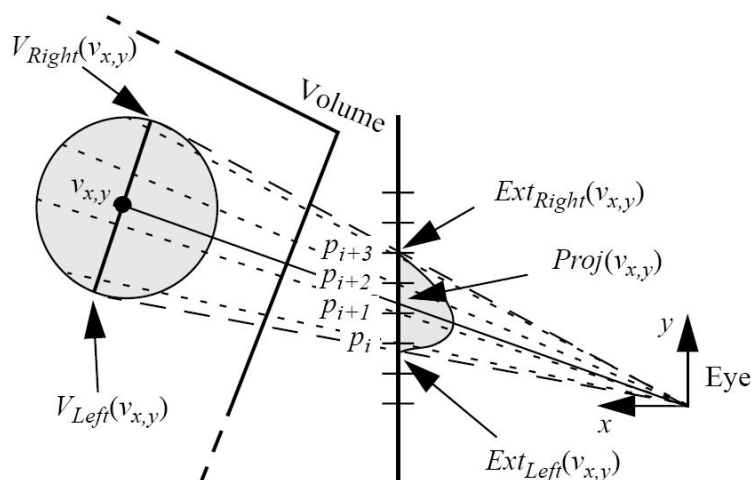


Figura 5.11 – Uso de projeção perspectiva deformando o núcleo de reconstrução.

### 5.3.2 Surface Splatting

O uso da técnica de *Splatting* aplicada à visualização de superfícies vem ganhando especial atenção nos últimos anos. Uma das razões para esse interesse é o crescente uso de pontos como primitiva básica para modelagem e visualização [Pauly (2001)] [Alexa (2003)] [Pauly (2003a)] [Pauly (2003b)] [Pfister (2003)] [Alexa (2004)]. A idéia de se utilizar pontos como primitiva básica para representar objetos tridimensionais não é recente [Levoy(1985)],[Szeliski(1992)],[Witkin(1994)]. Um fator que tem aumentado o interesse no uso de *splatting* para visualização de superfícies é o recente aprimoramento das técnicas de aquisição de dados tridimensionais, como scanners a laser e métodos baseados em imagens (*image-based methods*). Com essas ferramentas podemos gerar grandes volumes de amostras sobre a superfície de objetos tridimensionais complexos.

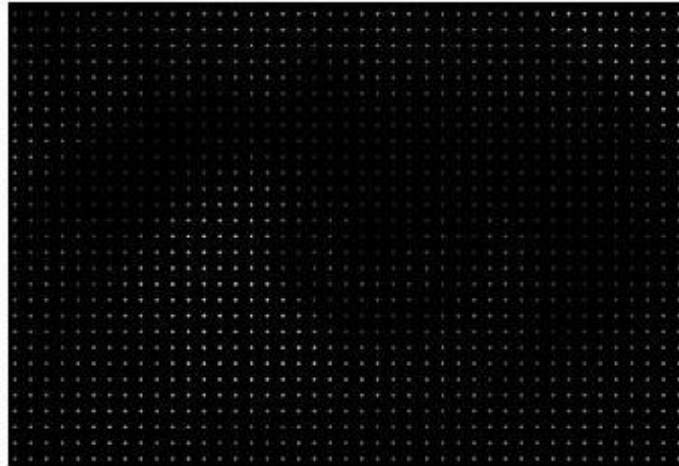
Tradicionalmente a visualização de superfícies é feita a partir da construção de uma superfície aproximada, linear por partes, composta por polígonos (em geral triângulos) formados a partir de conexão das amostras da superfície. A construção dessa malha poligonal envolve a determinação de relações de conectividade entre os pontos, bem como um controle da topologia da superfície final. Esse processo não é simples e envolve algoritmos não triviais, como por exemplo, o “*Power Crust*” [Amenta (2001)] e o “*Ball Pivot*” [Bernardini (1999)].

Uma superfície representada através de pontos é caracterizada por um conjunto de amostras  $\{P_k\}$ . Essas amostras estão definidas no espaço tridimensional, são irregularmente distribuídas, e não possuem informações topológicas associadas, ou seja, sua conectividade não é explicitada. Sobre esses pontos conhecemos apenas sua posição no espaço, sua normal e a sua aparência, ou seja, suas propriedades visuais como cor ou textura. Dessa forma podemos dizer que esses pontos representam uma amostragem da geometria e da aparência da superfície obtidas na mesma taxa de amostragem. De maneira similar aos pontos, um sistema de partículas não define relações de conectividade ou regularidade em sua distribuição espacial. No entanto, partículas possuem propriedades extras, tais como massa, velocidade e forças, por exemplo.

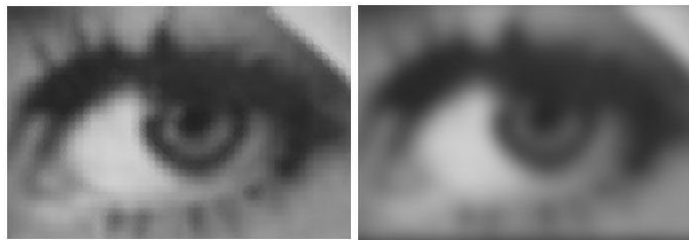
Uma das vantagens na ausência de informações de conectividade é um custo computacional mais baixo, não só em termos de armazenamento como também para certos tipos de processamento, como re-amostragem local ou re-estruturação dos pontos. Informações adicionais à cerca da topologia não precisam ser armazenadas nem mantidas e / ou validadas.

Um ponto crítico em uma representação baseada em pontos é a taxa de amostragem necessária para representar de forma adequada a superfície. É intuitivo que a densidade da amostragem está diretamente relacionada com a área da superfície que se quer representar. Esse ponto é interessante, já que implica em uma alta densidade de amostras, mesmo em regiões planas da superfície. Esse fato contrasta com os modelos poligonais, cuja densidade na amostragem é vinculada, em geral, a curvatura da superfície, requerendo, portanto, menos amostras em regiões planas.

No entanto, amostragens muito densas não se mostram interessantes do ponto de vista computacional, por seu alto custo. Na Figura 5.12(a) podemos ver um exemplo bidimensional simples, onde uma imagem é representada apenas por pontos. Claramente a densidade não é suficiente para reproduzir de forma acurada a imagem original, ou seja, não é capaz de “preencher” todo o espaço da imagem.



(a)



(b)

(c)

*Figura 5.12 – Exemplo de amostragem e reconstrução. (a) imagem com baixa densidade de amostras; (b) e (c) reconstrução utilizando um núcleo de reconstrução com diferentes raios de influência.*

Para resolver esse problema duas técnicas podem ser empregadas: reconstrução no espaço da imagem ou re-amostragem no espaço do objeto. Técnicas classificadas nessa última categoria procuram ajustar dinamicamente a taxa de amostragem, de tal sorte que a densidade dos pontos projetados seja adequada a resolução da imagem. Fica patente que essa re-amostragem é dependente dos parâmetros do sistema de visualização, e, portanto, deve ser refeita dinamicamente a cada frame. As técnicas de reconstrução procuram associar a cada amostra uma função ou núcleo de reconstrução. Essa função ocupa uma área ao redor do ponto e permite reconstruir a informação entre amostras. O efeito desse tipo de abordagem pode ser visto nas Figuras 5.12(b) e (c). A escolha da função de



reconstrução e seus parâmetros têm influência direta na visualização, como fica patente nas Figuras 5.12 (b) e (c). A técnica de *splatting* se enquadra nessa última categoria.

Sabemos que as representações poligonais reconstróem uma superfície a partir de um conjunto de amostras, construindo uma aproximação linear por partes, com continuidade  $C^0$ , como na Figura 5.13 à esquerda. Nesse caso presume-se que a conectividade das amostras é conhecida, o que não acontece no caso das representações baseadas em pontos. Para esse tipo de representação duas alternativas se colocam. A primeira é a construção de uma aproximação constante por partes, de continuidade  $C^{-1}$ , como na Figura 5.13 ao centro. Apesar de simples, fica evidente que esse tipo de função de reconstrução produz um erro superior a reconstrução linear por parte. Outra alternativa é utilizar uma aproximação linear por partes, de continuidade  $C^{-1}$  como na Figura 5.13 à direita. Nesse caso o erro gerado na aproximação é da mesma ordem da aproximação linear por partes tradicional, porém sem a exigência da conectividade entre as amostras.

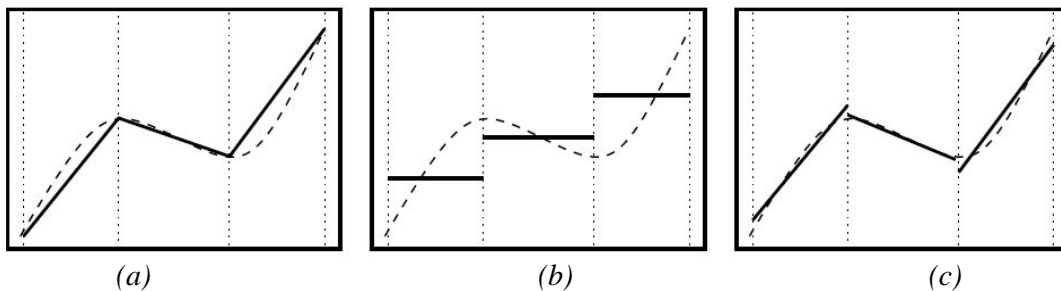


Figura 5.13 – Reconstrução de uma função unidimensional, em linhas pontilhadas. Reconstrução (a) linear por partes gerando uma função de continuidade  $C^0$ ; (b) constante por partes gerando uma função de continuidade  $C^{-1}$ ; (c) linear por partes com continuidade  $C^{-1}$ .

Esse último esquema de reconstrução é a base da técnica de *surface splatting* [Pfister (2003)]. Nesse contexto denominamos a função de reconstrução de núcleo de reconstrução. O núcleo de reconstrução tem por finalidade “estender” as dimensões dos pontos de modo que passem a preencher uma área ao seu redor.

Uma forma simples de representar esse núcleo de reconstrução é através de uma função linear, definida ao redor do ponto, formando um disco de raio  $r$  (*splat*), como vemos na Figura 5.14 à esquerda. No entanto, no caso de superfícies é interessante que esse núcleo de reconstrução possa se adaptar a curvatura da superfície subjacente. Essa característica pode ser obtida utilizando-se um *splat* elíptico. Um *splat* elíptico é capaz de gerar uma aproximação linear local ótima caso seus eixos principal e secundário estejam alinhados com as direções principais de curvatura da superfície, e seus raios sejam inversamente proporcionais as curvaturas máxima e mínima da superfície.

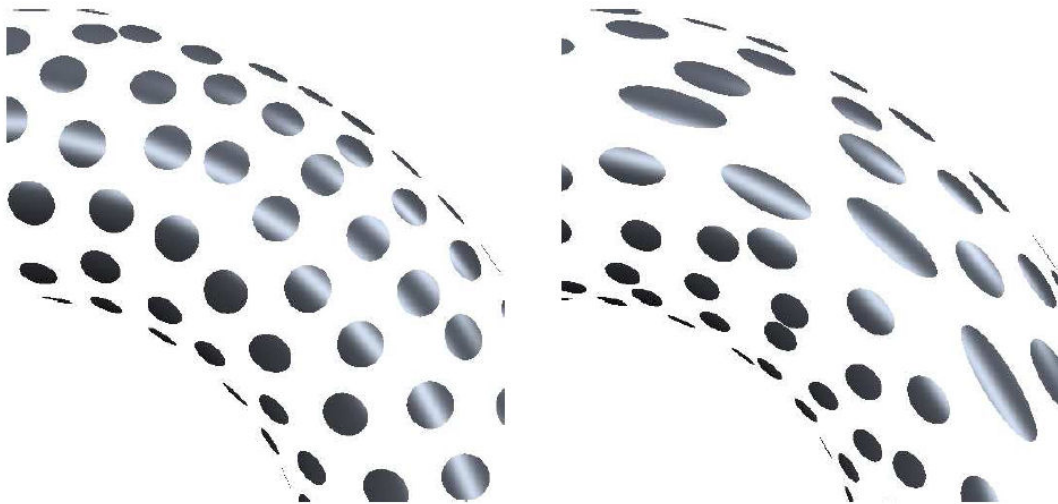


Figura 5.14 – Diferentes geometrias para o splats. Claramente observamos que o uso de elipses caracteriza de forma mais precisa a curvatura da superfície original.

O uso de uma aproximação linear, mesmo que de continuidade, nos permite adaptar a taxa de amostragem em função da curvatura da superfície, como nas aproximações poligonais lineares por partes. Dessa forma, regiões de baixa curvatura podem ter taxa de amostragem mais baixa, enquanto que regiões de alta curvatura necessitam de taxas de amostragem mais altas.

Uma vez definido o tipo de reconstrução que desejamos, ainda é preciso controlar de alguma forma o raio de influência de cada *splat*, de modo a conseguir uma imagem final contínua da superfície. Caso contrário teremos como resultado uma superfície composta de elipses desconexas, como na Figura 5.15(a).

Uma solução ingênua e simplista é definir raios suficientemente grandes que permitam que os *splats* se sobreponham. Essa solução permite que os espaços entre *splats* sejam preenchidos, porém as custas da introdução de discontinuidades na iluminação como podemos observar na Figura 5.15(b). O resultado torna-se muito similar à aplicação do algoritmo *Flat Shading* em uma superfície poligonal.

Como forma de minimizar esse problema, Zwicker [Zwicker (2001)] propõe o uso de um método anisotrópico de anti-aliasing, similar ao método *EWA – Elliptical Weight Average* – definido por Heckbert [Heckbert (1989)] para filtragem de texturas. Na abordagem de Zwicker cada ponto está associado com uma função de base simétrica e radial  $r_k$ , além de seu conjunto de atributos de cor  $w_k$ . As funções de base são núcleos de reconstrução definidos em domínios paramétricos locais. Dessa forma podemos definir uma função de textura contínua sobre a superfície representada pelo conjunto de pontos  $\{P_k\}$ .

Utilizando como funções de base Gaussianas podemos então suavizar a transição entre os *splats* associados a cada ponto, tornando assim a visualização da

superfície bem mais suave, como na figura 5.15 (c). A seguir descrevemos com mais detalhes essa formulação.



Figura 5.15 – Resultados produzidos por diferentes formas de integração dos splats: (a) Sem interferência entre cada splat, espaço vazios se formam; (b) núcleo de reconstrução constante, imagem semelhante a produzida pelo algoritmo Flat de iluminação; (c) núcleo de suavização Gaussiano, suaviza as transições entre splats, resultado semelhante ao algoritmo Gouraud de iluminação; (d) Phong Splatting, suaviza a transição entre as normais de cada splat.

Na Figura 5.16 podemos ver que a cor de um ponto qualquer  $Q$  da superfície, com coordenadas locais  $u$ , é avaliado pela combinação das funções de base  $r_k$ , gerando assim uma superfície contínua dada por:

$$f_c(u) = \sum_{k \in N} w_k r_k(u - u_k) \quad (5.11)$$

onde  $u_k$  é a coordenada local do ponto  $P_k$ .

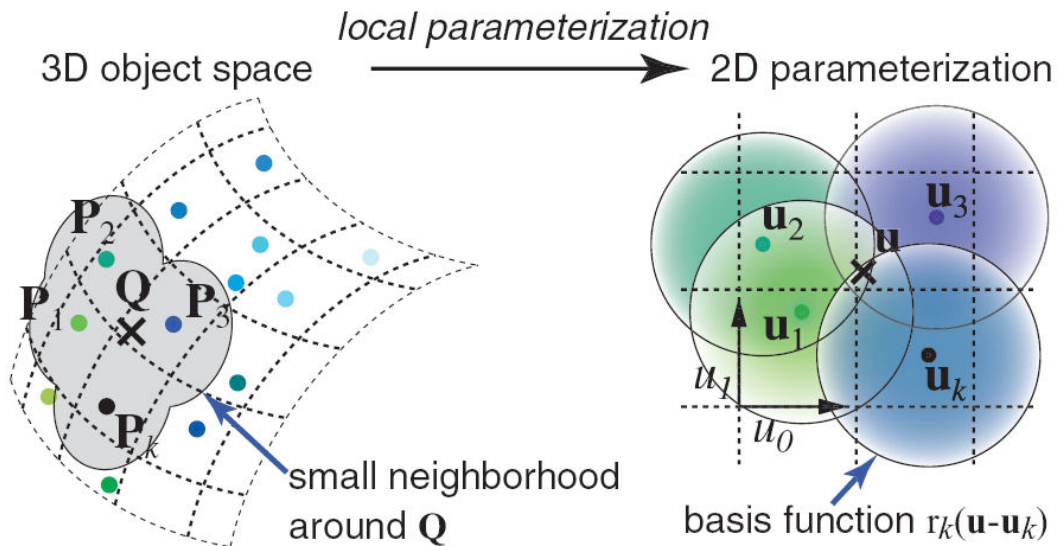


Figura 5.16 – Geração de uma parametrização local baseada no conjunto de amostras na proximidade de um ponto qualquer  $Q$ .

No trabalho de Heckbert os pontos  $P_k$  são distribuídos uniformemente, por se tratar de uma imagem (textura). Portanto a visualização da função (textura)  $f_c(u)$  produz uma função de saída  $g_c(x)$ , no espaço da tela que respeita o critério de Nyquist para a resolução da imagem final, evitando assim problemas de *alias*.

O processo de visualização envolve as seguintes etapas:

A função  $f_c(u)$  é deformada (*wrapped*) no espaço da tela, utilizando-se para isso um mapeamento afim local da projeção perspectiva de cada ponto, como mostra a Figura 5.17.

O sinal contínuo gerado no espaço da tela é então filtrado pela convolução com um filtro  $h$ , gerando uma nova função  $g_c(x)$ , onde  $x$  é a coordenada no espaço da tela.

A função  $g_c(x)$  pode ser re-escrita como um somatório ponderado de filtros de re-amostragem  $\rho_k(x)$  no espaço da tela [Zwicker (2003)], tal que:

$$g_c(x) = \sum_{k \in N} w_k \rho_k(x)$$

onde:

$$\rho_k(x) = (r'_k \otimes h)(x - m_k(u_k)) \quad (5.12)$$

As funções de base  $r'_k$  são definidas como funções elípticas Gaussianas. Essa escolha se deve principalmente as propriedades como invariância a transformações afins e convolução.

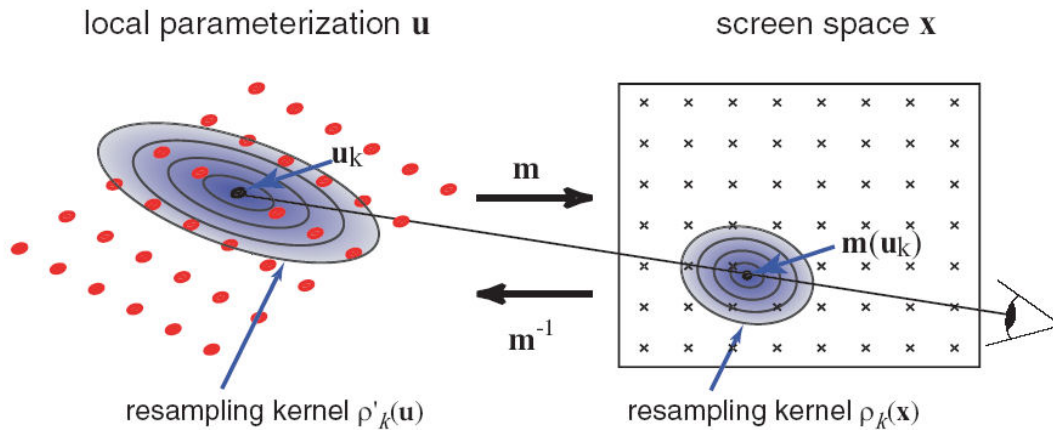


Figura 5.17 – Transformação do núcleo de suavização do espaço local de parametrização (definido sobre o objeto) para o espaço discreto da tela.

Szwicker chama a formulação dada pela equação (5.12) para o filtro  $\rho_k(x)$  de **Screen Space EWA**. No entanto, do ponto de vista computacional, é mais interessante que esse processo de re-amostragem seja definido como uma função na superfície paramétrica definida no espaço do objeto. Swicker propõe em [Swicker (2002)] o que chama de **Object Space EWA Splatting**, onde redefine o filtro de re-amostragem  $\rho_k(x)$  como:

$$\rho'_k(u) = (r_k \otimes h')(u - u_k)$$

Nessa nova formulação a função  $g_c(x)$  passa a ser escrita como:

$$g_c(x) = \sum_{k \in N} w_k \rho'_k(m^{-1}(x) - u_k)$$

Na Figura 5.18 podemos observar o efeito desse processo de re-amostragem baseado em EWA a partir de um padrão de textura quadriculado.

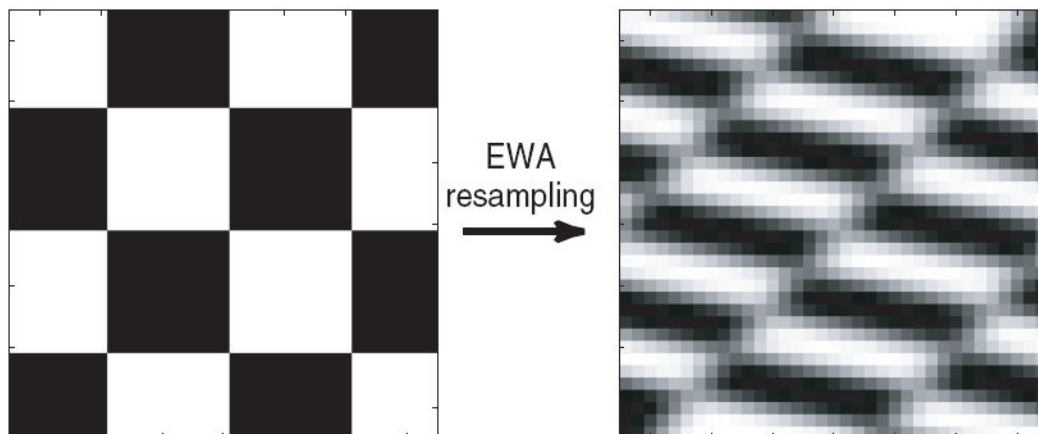


Figura 5.18 – Efeito da transformação mostrada na figura 5.17 utilizando-se a técnica de reamostragem EWA.

Em termos de implementação, o processo descrito acima é bastante custoso, uma vez que em geral é todo implementado em software. Segundo Botsch [Botsch (2004)] essa abordagem torna-se proibitiva para a visualização de modelos complexos em tempo real, uma vez que as taxas de *splat rate* alcançadas variavam na faixa de 1M *splats* por segundo (análise feita com o hardware da época).

Duas frentes de pesquisa aparecem com o objetivo de acelerar o *splat rate* desse tipo de visualização. A primeira procura atuar como as técnicas de multi-resolução em modelos poligonais, reduzindo o número de *splats* a serem processados em função de qualidade / ponto de vista da visualização. Dois trabalhos nessa linha são os de Botsch [Botsch 2002] e o *QSplat* [Rusinkiewicz 2000]. Ambos são ainda processados em software, mas conseguem produzir *splat rates* da ordem de 5M *splats* por segundo.

Outra linha de ação que vem sendo bastante explorada é a transferência de parte do processamento da CPU para a GPU (*Graphics Processing Unit*). Essa classe de processadores permite que programas sejam escritos para atuarem ao longo do *pipeline* gráfico em duas etapas: no processamento dos vértices e no processamento dos *pixels*. Os programas que são carregados na GPU são denominados *shaders*. A grande vantagem desse tipo de solução é a possibilidade de atuação tanto no espaço do objeto, pelo uso de *vertex shaders*, quanto no espaço da tela, através dos *fragment shaders* ou *pixel shaders*. Exemplos de trabalhos nessa linha podem ser encontrados em [Botsch 2002], [Botsch 2004] e [Zwicker 2004].

Os métodos baseados em *EWA* produzem resultados visuais satisfatórios (Figura 5.15(c)), mas ainda longe daqueles produzidos em malhas poligonais pelo algoritmo de *Phong*. Assim como o algoritmo de *Gouraud*, o método de *splatting* baseado em funções de base Gaussianas tende a produzir modelos com iluminação “borrada” (*blurry*). Botsch et al. [Botsch 2004] propõe uma solução para esse tipo de problema, mantendo a essência da proposta de *EWA splatting*. Para cada *splat* um mapa de normais linear é associado. A visualização desses *Phong-splats* pode ser feita de forma bastante simples e eficiente pelo uso dos recursos de programação em GPUs. O resultado final é visualmente bastante próximo ao obtido pelo próprio algoritmo de *Phong* aplicado a modelos poligonais, como podemos ver na figura 5.15(d).

## 5.4. Visualização de Campos Vetoriais

### 5.4.1 Integrais de Campo

Esta técnica envolve o cálculo das trajetórias de campo definidas na seção 3.3. Estaremos supondo os campos de velocidade e vorticidade com sendo de classe  $C^1$ , e portanto, estas trajetórias estão bem definidas de acordo com os resultados das seções 3.2 e 3.3.

Vejamos o exemplo de um método que será utilizado no capítulo 6 e que apresenta as principais características desta metodologia. A idéia básica deste método é construir linhas de trajetória em campos vetoriais 3D não-estacionários, definidos em uma malha estruturada de hexaedros não regulares. Nesta apresentação estaremos supondo o caso do campo de velocidades embora o método possa ser aplicado para outros campos.

A primeiro aspecto a definir é se vamos trabalhar no espaço das variáveis da simulação numérica ou se vamos efetuar os cálculos em outro espaço. Para o caso de malhas estruturadas não regulares de hexaedros ou tetraedros, pode ser interessante transformar as coordenadas iniciais (físicas) para outras coordenadas mais eficientes do ponto de vista computacional (coordenadas de computação).

Sejam então, o espaço físico aquele cujas coordenadas correspondem a um subconjunto aberto  $\Delta \subset \mathbb{R}^4$  de pontos  $(x_1, x_2, x_3, t)$ , onde  $x_1, x_2, x_3$  são coordenadas espaciais e  $t$  o tempo. Este espaço é suposto discretizado em um conjunto finito de quadras  $\{(x_1, x_2, x_3, t) \in \mathbb{R}^4; i, j, k, l \in \mathbb{N}\}$ , conectadas de acordo com uma topologia que gere uma malha estruturada de hexaedros  $M$ .

Vamos supor a existência de um difeomorfismo local (seção A.5)  $\bar{T}: \Delta \rightarrow \bar{\Delta} \subset \mathbb{R}^4$  que transforma a malha  $M$  em uma malha regular  $\bar{M} \subset \bar{\Delta}$ . Ao espaço correspondente às coordenadas  $(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{t}) \in \bar{\Delta}$  chamaremos espaço computacional.

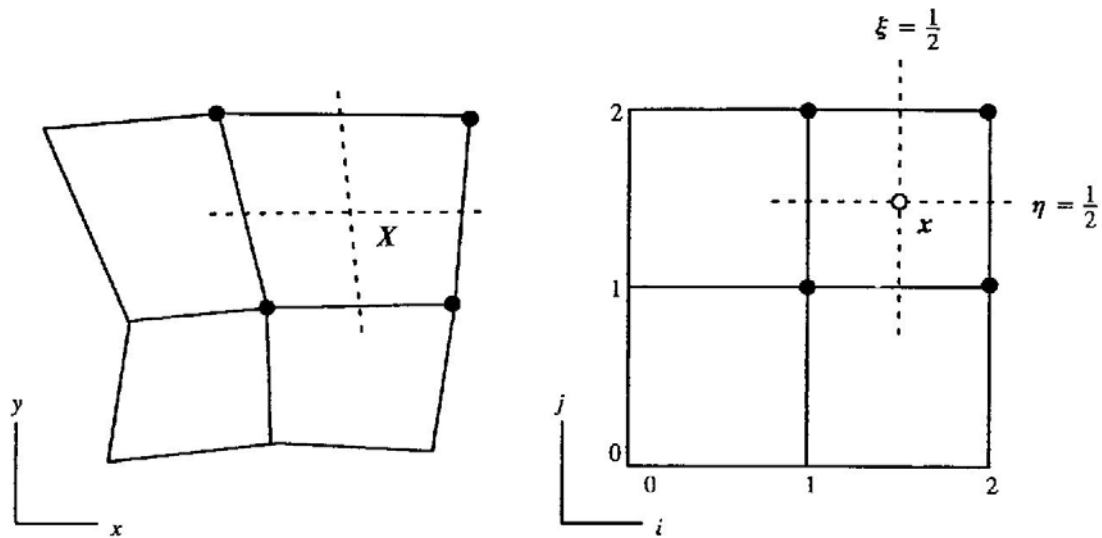


Figura 5.19 – Espaço físico:  $(x, y)$  espaço computacional:  $(\bar{x}, \bar{y})$ , e espaço de parâmetros:  $(\xi, \eta)$ .

Uma vez que  $\bar{T}$  é um difeomorfismo local, um campo vetorial  $\bar{V}$  definido em  $\Delta \subset \mathfrak{R}^4$  é transformado naturalmente em um campo vetorial  $\bar{v}$ , topologicamente conjugado a  $V$  (seção 3.4.2), pela relação:

$$\bar{v} = (D\bar{T})\bar{V}$$

onde  $D$  denota a diferencial de uma aplicação definida no Apêndice A, seção A.4.

Uma vez que o espaço físico está discretizado, o primeiro passo nesta metodologia é estabelecer uma regra de interpolação para os vértices das células da malha  $M$ . Neste trabalho, assumimos que cada célula do espaço físico possa ser representada por uma aplicação quadrilinear da forma:

$$X(\xi, \eta, \zeta, \tau) = (x_1(\xi, \eta, \zeta, \tau), x_2(\xi, \eta, \zeta, \tau), x_3(\xi, \eta, \zeta, \tau), t(\xi, \eta, \zeta, \tau)) = \\ \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 \sum_{l=0}^1 X_{i,j,k,l} B_i^1(\xi) B_j^1(\eta) B_k^1(\zeta) B_l^1(\tau), \xi, \eta, \zeta, \tau \in [0,1]$$

onde  $X_{i,j,k,l}$  são vértices de uma célula no espaço físico contendo  $X$  e  $B_i^1$  é o polinômio de Bernstein-Bézier de grau um (Apêndice A). Esta expressão pode ser interpretada como uma parametrização da célula da malha  $M$  com vértices  $X_{i,j,k,l}$ . O espaço correspondente às variáveis  $\xi, \eta, \zeta$  e  $\tau$  será denominado espaço de parâmetros locais (Figura 5.19).

Naturalmente, cada célula de  $\bar{M} \subset \bar{\Delta}$  ficará parametrizada pela transformação:

$$\bar{X}(\xi, \eta, \zeta, \tau) = (\bar{x}_1(\xi, \eta, \zeta, \tau), \bar{x}_2(\xi, \eta, \zeta, \tau), \bar{x}_3(\xi, \eta, \zeta, \tau), \bar{t}(\xi, \eta, \zeta, \tau)) = \\ \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 \sum_{l=0}^1 \bar{X}_{i,j,k,l} B_i^1(\xi) B_j^1(\eta) B_k^1(\zeta) B_l^1(\tau), \xi, \eta, \zeta, \tau \in [0,1]$$

onde:

$$\bar{X}_{i,j,k,l} = \bar{T}(X_{i,j,k,l})$$

são os pontos  $(X_{i,j,k,l})$  da malha original transformados por  $\bar{T}$ . O campo de velocidades no espaço físico e seu correspondente no espaço computacional ficam também parametrizados por:

$$\bar{V}(\xi, \eta, \zeta, \tau) = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 \sum_{l=0}^1 \bar{V}_{i,j,k,l} B_i^1(\xi) B_j^1(\eta) B_k^1(\zeta) B_l^1(\tau), \xi, \eta, \zeta, \tau \in [0,1] \quad (5.13)$$

$$\bar{v}(\xi, \eta, \zeta, \tau) = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 \sum_{l=0}^1 \bar{v}_{i,j,k,l} B_i^1(\xi) B_j^1(\eta) B_k^1(\zeta) B_l^1(\tau), \xi, \eta, \zeta, \tau \in [0,1]$$

onde:

$$\bar{v}_{i,j,k,l} = (D\bar{T})\bar{V}_{i,j,k,l}$$



Caso a transformação  $\bar{T}$  seja conhecida apenas nos pontos da malha então a diferencial na última expressão é determinada via diferenças finitas.

Nosso objetivo é traçar linhas de trajetória definidas pelo problema de Cauchy (3.8). A idéia central dos métodos para a geração de linhas de trajetória é aproximar a curva integral correspondente por uma seqüência de pontos  $(X_n)$ . Em [Hamann et al. (1995)], isto é feito calculando-se inicialmente a seqüência de pontos correspondente  $(\bar{X}_n)$ , no espaço computacional, para em seguida mapear esta seqüência no espaço físico via  $\bar{T}^{-1}$ .

Seja então o ponto inicial  $X_0$  no espaço físico. Efetuaremos toda a computação nas coordenadas locais  $\xi, \eta, \zeta$  e  $\tau$ . Neste sentido, o primeiro passo é encontrar os parâmetros locais  $\xi_l, \eta_l, \zeta_l$  e  $\tau_l$  deste ponto. Isto é feito resolvendo-se a equação:

$$X_0 = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 \sum_{l=0}^1 X_{i,j,k,l} B_i^1(\xi) B_j^1(\eta) B_k^1(\zeta) B_l^1(\tau), \xi, \eta, \zeta, \tau \in [0,1]$$

o que pode ser feito usando-se Newton-Raphson para gerar uma seqüência de pontos  $(d_n)$  no espaço de parâmetros tal que  $X(d_n) \rightarrow X_0$  se  $n \rightarrow +\infty$ .

O próximo passo é integrar (3.8) no espaço computacional. Neste ponto, a abordagem usual encontrada na literatura faz uso de Runge-Kutta (em geral de quarta ordem).

Nesta apresentação nós preferimos utilizar a metodologia alternativa apresentada em [Hamann et al. (1995)] que permitirá uma abordagem unificada e computacionalmente eficiente para os problemas de interpolação dos campos que encontraremos.

A idéia central deste método é usar uma curva polinomial de segundo grau expressa na base de Bezier-Bernstein para aproximar localmente a curva integral e em seguida integrar (3.8) sobre esta curva. Vejamos estes passos em detalhe.

A Figura 5.20 representa uma célula no espaço computacional e um ponto  $\bar{X}_l$  no interior da mesma o qual é sabido pertencer à linha de trajetória (no passo inicial este ponto seria o próprio  $\bar{X}_0$ ).

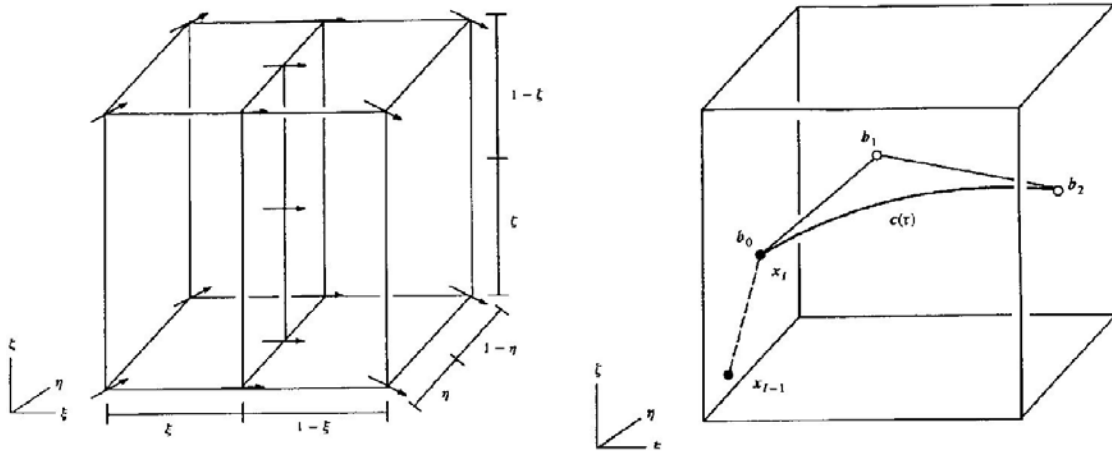


Figura 5.20 – Interpolação tri-linear de um campo 3D e aproximação local da linha de trajetória usando uma curva de Bézier de segunda ordem.

Queremos obter uma curva polinomial de segundo grau que aproxime (no espaço computacional) a linha de trajetória localmente no interior da célula representada. Os pontos  $b_1$  e  $b_2$  da Figura 5.20 são dois pontos de controle a serem determinados.

A curva em questão pode ser representada parametricamente por:

$$c(\tilde{\tau}) = \sum_{i=0}^2 b_i B_i^2(\tilde{\tau}), \quad \tilde{\tau} \in [0, T] \quad (5.14)$$

onde  $T$  é um passo de tempo previamente escolhido e  $\tilde{\tau}$  é um tempo local tal que  $\tilde{\tau} = 0$  está associado ao ponto  $\bar{X}_i$ . A expressão para os polinômios de Bezier-Bernstein no parâmetro  $\tilde{\tau}$  são obtidas diretamente da expressão (A-1) fazendo  $t = \frac{\tilde{\tau}}{T}$ ,  $n = 2$ ; e são dadas por:

$$B_i^2(\tilde{\tau}) = \frac{1}{T^2} \binom{2}{i} \tilde{\tau}^i (T - \tilde{\tau})^{2-i} \quad (5.15)$$

Uma vez que  $\bar{X}_i$  deve ser um ponto de  $c$ , podemos usar os resultados (A-5) e (A-7) do Apêndice A, juntamente com (5.15), para expressar os pontos de controle da curva polinomial procurada da seguinte forma:

$$b_0 = \bar{X}_i,$$

$$b_1 = \bar{X}_i + \frac{T}{2} \bar{v}_i,$$

$$b_2 = \bar{X}_i + T \bar{v}_i + \frac{T^2}{2} a_i,$$

onde estamos impondo  $\frac{dc}{d\tilde{\tau}}(0) = \bar{v}_I$ ,  $\frac{d^2c}{d\tilde{\tau}^2}(0) = \bar{a}_I$ . As expressões para  $\bar{v}_I$  e  $\bar{a}_I$  são obtidas diretamente de (5.13) e são dadas por:

$$\bar{v}_I = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 \sum_{l=0}^1 v_{i,j,k,l} B_i^1(\xi_I) B_i^1(\eta_I) B_i^1(\zeta_I) B_i^1(\tau_I)$$

$$\bar{a}_I = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 (v_{i,j,k,1} - v_{i,j,k,0}) B_i^1(\xi_I) B_i^1(\eta_I) B_i^1(\zeta_I).$$

onde  $(\xi_I, \eta_I, \zeta_I, \tau_I)$  são as coordenadas locais de  $\bar{X}_I$  (iguais às coordenadas locais de  $X_I$ ).

Desta forma, temos a representação local (5.14) para a linha de trajetória. O próximo ponto  $\bar{X}_{I+1}$  da linha de trajetória (no espaço computacional) pode ser obtido pela integral de linha:

$$\bar{X}_{I+1} = \bar{X}_I + \int_0^T \bar{v}(c(\tilde{\tau}), \tilde{\tau}) d\tilde{\tau}$$

expressa em função do tempo local  $\tilde{\tau}$ .

Procedendo desta forma vamos obter uma seqüência de pontos  $(\bar{X}_n)$  da linha de trajetória no espaço computacional. Em seguida, basta usar  $\bar{T}^{-1}$  para obter a seqüência correspondente  $(X_n)$  no espaço físico.

Procedimentos análogos poderiam ser utilizados para as demais linhas de campo definidas em (3.6)-(3.9), mais uma vez, podendo-se optar pela utilização do método aqui descrito ou Runge-Kutta para a solução numérica da equação diferencial correspondente.

As linhas de trajetória são de particular interesse para nosso trabalho para o acompanhamento das estruturas ao longo do tempo. Trataremos esta questão no capítulo 6.

A definição (3.6) gera uma superfície. A idéia de gerar um conjunto de curvas integrais para aproximar esta superfície deve ser tomada com cuidado, pois pode ser cara ou mesmo ineficiente o que obriga o desenvolvimento de técnicas específicas para obter aproximações da mesma usando-se um menor número possível de curvas integrais [Ueng (1996)].

#### 5.4.2 Line Integral Convolution

Em vez de desenharmos uma coleção de curvas integrais para estudar um campo bidimensional, a idéia deste método é fazer a convolução de uma função ruído  $T$  (textura) ao longo da linha de corrente.

Seja então um escoamento bidimensional e estacionário. Seja também a solução  $x(s)$  de (3.7) passando pelo ponto  $x = x_0$ . O presente método consiste em calcular a integral:

$$I(y) = \int_{y-L}^{y+L} K(s-y)T(x(s))ds \quad (5.16)$$

onde  $K$  é o *kernel* de um filtro unidimensional com suporte igual a  $2L$ .

O resultado deste método é uma alta correlação dos valores das intensidades na direção da linha de corrente, enquanto que perpendicular a esta direção praticamente não existe correlação. Assim, a estrutura direcional do campo emerge visualmente, o que pode ser ilustrado pela Figura 5.21.

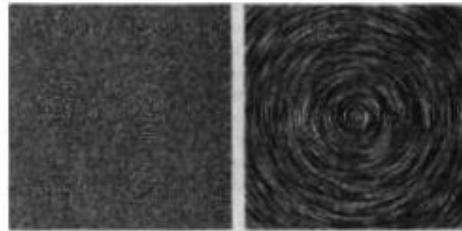


Figura 5.21 – Função ruído (esquerda) é filtrada usando LIC sobre um campo vetorial circular gerando a imagem à direita.

Resolver a integral de convolução acima para cada *pixel* é muito caro. Uma forma de contornar este problema é considerar a função  $T$  constante no interior de cada célula  $C_i$  cortada pela linha de corrente  $x(s)$ . Assim, a integral (5.16) seria aproximada por um somatório do tipo [Forsell-Cohen (1995)]:

$$I(y) = \sum_{i=1}^{N_y} T(p_i)h(p_i, y), \text{ onde}$$

$$h(p_i, y) = \int_{a_i}^{b_i} K(s-y)\bar{P}_i(x(s))ds$$

onde  $p_i$  é um ponto da linha de corrente no interior de  $C_i$ ,  $s = a_i$  e  $s = b_i$  correspondem aos pontos onde a linha de corrente  $x(s)$  corta a célula  $C_i$ ,  $N_y$  é o número de células cortadas e  $\bar{P}_i$  é tal que:

$$\bar{P}_i(x) = \begin{cases} 1, & \text{se } x \in C_i, \\ 0, & \text{cc.} \end{cases}$$

Para campos vetoriais não estacionários a idéia é usar filtros de movimento (*motion filters*) e integrar ao longo das linhas de emissão em vez das linhas de corrente [Forsell-Cohen (1995)]. Desta forma procura-se manter a coerência entre os diversos *frames* e obter a impressão de um fluxo no tempo.

Possíveis extensões deste método para campos 3D são discutidas em [Interrante-Grosch (1997)] e são baseadas na utilização de uma segunda textura para se obter noção de profundidade para as linhas de corrente.

Uma possibilidade que pode ser interessante para a visualização estruturas no interior do fluido é aplicar esta técnica sobre a fronteira da estrutura para visualizar o campo de velocidades projetado sobre a mesma [Battlke et al. (1996)]. A Figura 5.22 mostra o resultado deste método aplicado sobre a superfície de um vórtice do tipo daquele da figura 5.23. Os quatro entalhes observados correspondem a 4 linhas de corrente (do campo projetado) e ajudam a compreender o movimento helicoidal do fluido no interior do vórtice.

Esta técnica pode ser enriquecida para simular efeitos obtidos por técnicas experimentais tais como injeção de substâncias coloridas (*dye*) no fluido. Isto é feito acrescentando-se cores a regiões isoladas da textura *T*. Estas regiões são carregadas pelo fluxo, da mesma forma que a substância colorida é carregada pelo escoamento, e o resultado é uma mancha que se desloca [Shen et al. (1996)] realçando informações dos campos nas regiões por ela atravessadas.



Figura 5.22- Realce de vórtices por LIC.

#### 5.4.3 Visualização da Topologia de Campos Vetoriais

Para campos vetoriais 2D estacionários esta técnica consiste basicamente dos seguintes passos[Rosenblum et al (1994)]:

- a) Detecção dos pontos críticos.
- b) Classificação dos pontos críticos
- c) Construção de um esqueleto interligando convenientemente estes pontos.

No item c) acima, o que devemos entender por convenientemente?

Os conjuntos  $\omega$ -limite e  $\alpha$ -limite, estudados na seção 3.4.6, podem fornecer uma resposta a esta pergunta. Vejamos novamente a Figura 5.23.

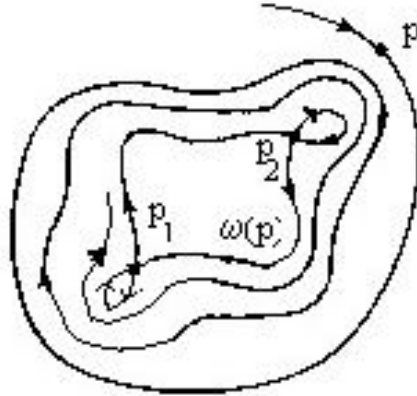


Figura 5.23 – Esqueleto formado pelo conjunto  $\omega(p)$ .

Neste caso, dispondo do esqueleto formado por  $\omega(p)$ , basta usar apenas a continuidade do fluxo para inferir o comportamento das trajetórias em vizinhanças do mesmo. Novamente, pela discussão da seção 3.4.6, na ausência de outros pontos singulares, não existem trajetórias fechadas e a única possibilidade para a topologia do campo fora da região representada na Figura 5.23 é dada pelo teorema do fluxo tubular.

Se tomamos um ponto  $p$  tal que  $\omega(p)$  é um pontos singular então podemos precisar do conjunto  $\alpha(p)$  também. Como exemplo, podemos retomar a Figura 3.6, mas com um foco estável na origem. Neste caso, dado um ponto  $p$  no interior de  $C$  teríamos  $\omega(p) = \{(0,0)\}$  e  $\alpha(p) = C$ , obtendo-se assim os elementos determinantes da topologia da região representada.

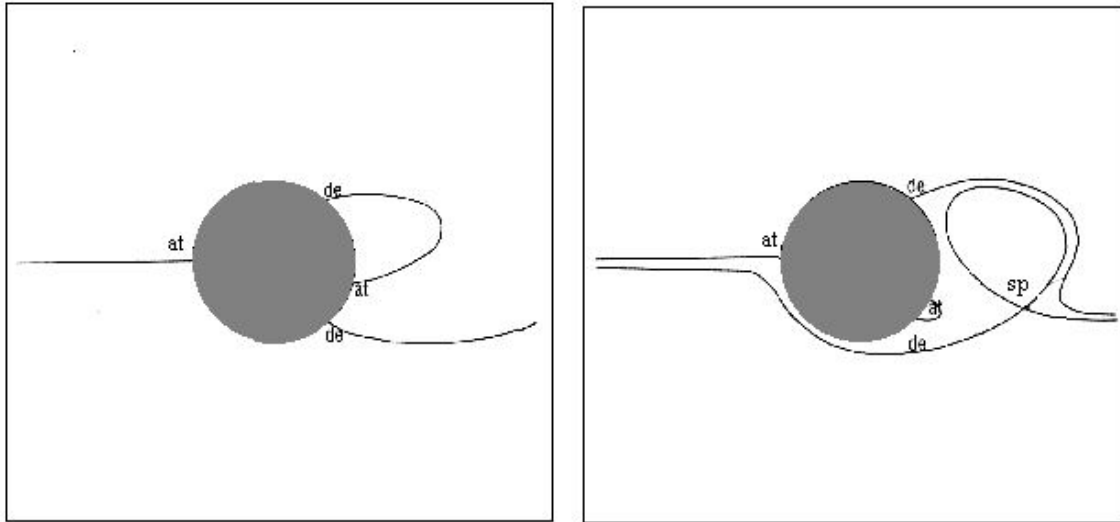
Se o campo for 2D não-estacionário então podemos tomar os três passos acima para cada instante de tempo e em seguida conectar os esqueletos obtidos convenientemente para representar a evolução temporal da topologia do fluxo.

No caso dos campos provenientes da solução das equações de Navier-Stokes (2.18), para condições iniciais e de contorno apropriadas, devemos fazer algumas considerações especiais.

O primeiro cuidado nestes casos refere-se às condições de fronteira. Para o caso de fronteiras sem escorregamento (*no-slip boundary*), coloca-se velocidade nula nas fronteiras do domínio, o que vai implicar num conjunto infinito e não discreto de singularidades para o campo de velocidades.

Um comportamento comum nestes casos é aquele representado na Figuras 5.24, onde vemos uma linha de corrente, inicialmente presa ao cilindro nos pontos *de* e *at* desprender-se do mesmo em uma das extremidades num instante posterior.

Observemos que esta linha de corrente, na figura à esquerda, inicia em *de* e termina em *at*, pois a circunferência correspondente ao cilindro não é uma linha de corrente (velocidade deve ser diferenciável também nos pontos da fronteira do cilindro, logo tem-se a unicidade das curvas integrais).



*Figura 5.24 – Linhas de corrente nas proximidades de um cilindro em dois instantes de tempo. O ponto **sp** é um ponto de cela e os pontos **at** e **de** são pontos de junção e separação, respectivamente.*

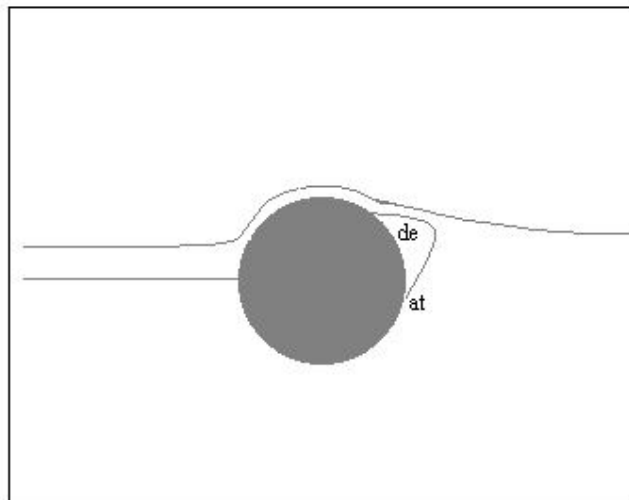
Os pontos *de* e *at* (ponto de separação e junção, respectivamente), ligados por uma mesma linha de corrente, são particularmente importantes, pois estes indicam uma região onde o movimento do fluido ficará confinado durante um certo intervalo de tempo (no caso da Figura 5.24, esta região é aquela limitada pela fronteira do cilindro e a linha de corrente que inicia em *de* e termina em *at*).

O método apresentado em [Helman-Hesselink (1991)] dá atenção especial a estes pontos e às linhas de corrente que os ligam. Quando isolados estes podem definir regiões de natureza topológica bem distintas, daí sua importância para a visualização do campo.

Para o caso 3D a idéia é usar a mesma metodologia, mas agora aplicá-la inicialmente ao estudo da topologia do escoamento sobre a superfície dos corpos imersos no fluido.

Para tal, considera-se o campo de velocidades definido no plano da malha que é um ponto acima da superfície do objeto (estamos supondo superfícies sem escorregamento) e procuram-se os pontos de separação e junção. Com isto, os pontos de separação serão aqueles com componente normal da velocidade positiva

e os pontos de junção com componente normal da velocidade negativa. Neste caso, estes pontos formarão linhas sobre a superfície do objeto, denominadas linhas de separação. Estas linhas indicam que superfícies de corrente que inicialmente estão próximas da fronteira do objeto, afastam-se abruptamente da mesma numa vizinhança destas curvas. A Figura 5.25 ajuda visualizar este fato aqui novamente representado para o caso 2D.



*Figura 5.25 – Linha de corrente em uma vizinhança da superfície afastando-se próximo ao ponto de separação **de**.*

Estas superfícies, da mesma forma que a linha de corrente ligando *at* com *de* na Figura 5-24, determinarão o comportamento do fluido numa vizinhança do objeto podendo indicar a presença de vórtices.

Para aplicações onde o interesse é acompanhar a evolução de uma estrutura (vórtices, por exemplo) no tempo, esta técnica torna-se limitada, pois a mesma objetiva acompanhar a evolução (global) da topologia não se atendo a evolução temporal de uma estrutura específica.

Apesar disto, é certamente uma técnica elegante e robusta por basear-se em elementos de fácil identificação e caracterização: pontos críticos isolados, pontos de separação e junção.

### **5.5. Técnicas para Visualização de Vórtices**

As técnicas encontradas na literatura para a identificação e visualização de vórtices no conjunto de dados podem ser classificadas em três categorias: técnicas baseadas em isosuperfícies, técnicas baseadas no core e técnicas baseadas em extração de características (segmentação). Do ponto de vista da classificação apresentada na Tabela 1, estas técnicas visam um resultado que traduza o comportamento do fluido uma região do mesmo. É neste sentido que estas técnicas são locais.



Por outro lado, a informação obtida pode se referir ao campo de vorticidade e de pressão na região tendo assim a natureza escalar / vetorial indicada naquela tabela. O domínio volumétrico indica mais uma característica destas técnicas: delimitar uma região do espaço onde o fluido esteja em “rotação”.

Vejamos as técnicas do primeiro tipo.

### 5.5.1 Técnicas Baseadas em Isosuperfícies

A metodologia básica neste caso é encontrar um campo escalar que possa ser facilmente obtido a partir dos campos das equações 2.18 e 2.23 e tal que, para certos valores, as isosuperfícies correspondentes envolvam os vórtices. Encontramos duas propostas neste sentido.

#### Isosuperfícies de Baixa Pressão

Em [Robinson (1989)] propõe-se que volumes alongados com baixa pressão em escoamentos turbulentos e incompressíveis indicam a presença de vórtices.

Assim, uma vez feita uma estatística inicial do campo de pressão que permita determinar um limiar para baixa pressão, uma técnica de geração de isosuperfícies do tipo discutida na seção 4.2.1 pode trazer bons resultados para a identificação de estruturas do tipo vórtice.

A dificuldade principal com esta técnica é a grande massa de dados em geral envolvida, o que levará a uma superfície poliédrica com muitas faces trazendo dificuldades para o processo de *rendering* de acordo com a discussão da seção 5.2.1.

A Figura 5-26 mostra um vórtice em forma de “grampo” obtido com esta técnica.

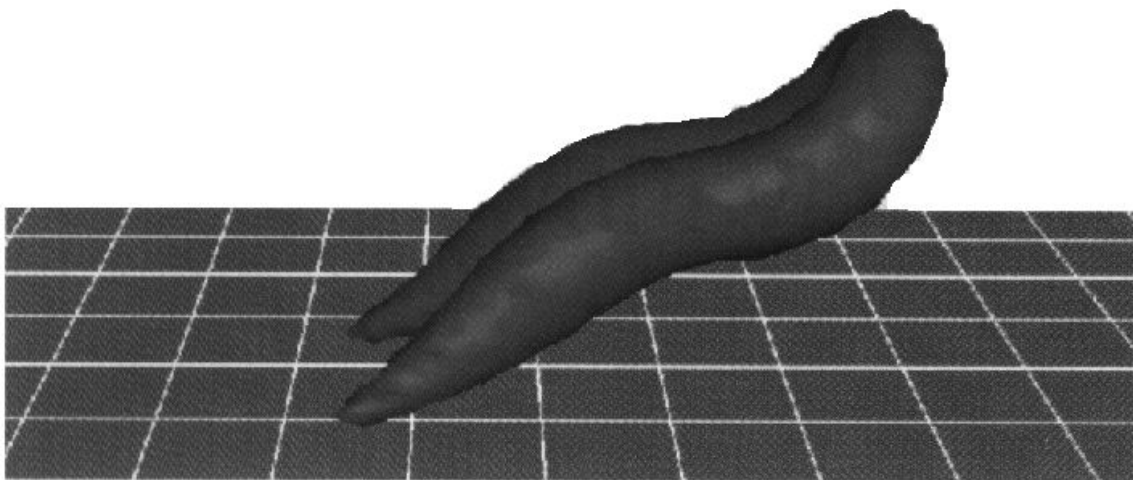


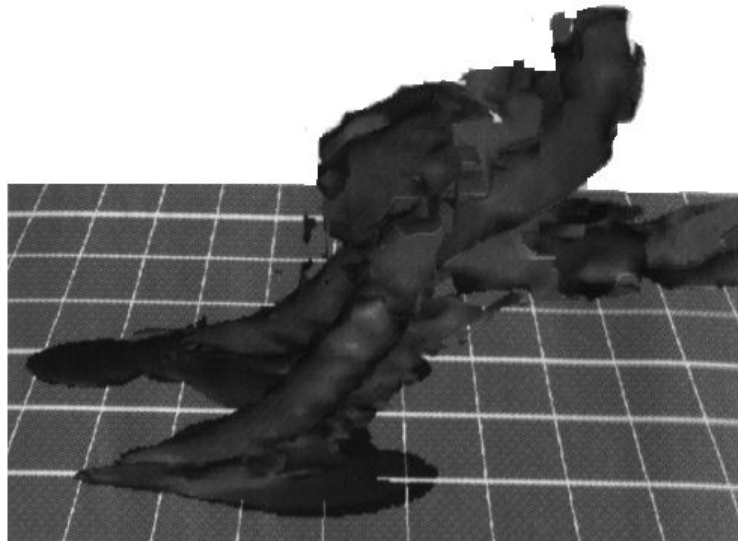
Figura 5.26 – Isosuperfície de pressão.

## Autovalores do Tensor de Deformação

Em [Chong-Perry (1990)] usam-se as idéias discutidas da seção 2.6 para propor uma caracterização dos vórtices via os autovalores do tensor de deformação (ver 2.30). Neste trabalho, os autovalores deste tensor são calculados em pontos de alta vorticidade. Autovalores com parte imaginária suficientemente alta vão indicar que o ponto pertence a um vórtice.

Este método foi aplicado no estudo de vórtices em fluidos *free-shear* e se mostrou eficiente para identificar vórtices de maior intensidade no interior do fluido. No entanto, esta técnica vai capturar também pequenas estruturas sem oferecer recursos para determinar como estas se conectam (figura 5.27). Além disso, certamente teremos superfícies poliédricas como muitas faces trazendo o mesmo problema das isosuperfícies de baixa pressão discutido no item anterior.

A figura 5.27 mostra um resultado obtido com esta técnica. A superfície mostrada envolve uma região do fluido com alta vorticidade onde o tensor de deformação apresenta autovalores imaginários.



*Figura 5.27 – Isosuperfície formada por pontos onde o tensor de deformação possui autovalores imaginários.*

## 5.5.2 Técnicas Baseadas na Identificação do Core

### Algoritmo para time tracking de tubos de vorticidade em Turbulência.

O fenômeno da turbulência em mecânica de fluidos é caracterizado por altas flutuações estatísticas das quantidades que caracterizam o escoamento. Estes fenômenos ocorrem (em geral) para valores altos do número de Reynolds (definido como o produto das escalas representativas da velocidade e comprimento dividido pela viscosidade cinemática [Landau-Lifshitz (1959)]) e são fundamentais para a compreensão do comportamento dos fluidos na natureza.

O objetivo desta técnica é estudar a morfologia e a evolução temporal de tubos de vorticidade em turbulência homogênea. Podemos entender estes tubos no sentido da definição de tubos vórtices da seção 2.6, mas com a superfície obtida envolvendo uma região de alta vorticidade. Assim estes objetos podem ser vistos como vórtices alongados e, portanto, não constituem uma estrutura nova.

A técnica em questão vai centrar a atenção não em um conjunto de tubos, mas apenas em um único tubo e acompanhar sua história no tempo. Vejamos alguns detalhes da mesma.

#### ***Descrição do Algoritmo.***

Um tubo de vorticidade é geometricamente uma estrutura alongada e conexa que pode ser caracterizada por um esqueleto [Villasenor-Vincent (1992)].

O algoritmo funciona em dois estágios. Primeiramente o esqueleto é progressivamente construído usando-se uma série de segmentos de linhas, de igual tamanho, conectados seguindo o eixo geométrico do tubo. A seguir os vetores que definem o tubo são identificados pelo método descrito abaixo.

#### ***Extraindo o Esqueleto do Tubo.***

Feita a determinação de um ponto semente (definido pelo usuário ou automaticamente) o próximo passo será encontrar a direção mais provável para o primeiro segmento do eixo da estrutura. Isto é feito gerando-se as coordenadas de varias centenas de pontos espalhados sobre a superfície de uma esfera centrada no ponto semente. Feito isto um segmento de reta é desenhado partindo do ponto semente até um destes pontos e o campo de vorticidade em uma região cilíndrica do espaço possuindo este segmento como eixo é representado pela média das normas dos vetores dentro do cilindro.

Este procedimento é repetido para todos os cilindros, e aquele com maior vorticidade média é utilizado para definir a direção do primeiro segmento do esqueleto.

O processo é repetido usando-se agora uma extremidade deste segmento como ponto semente. Atinge-se um extremo do core quando o processo acima só encontra regiões com baixa intensidade de campo.

Ao detectar-se o primeiro extremo, o algoritmo retorna ao primeiro ponto encontrado usando-se o ponto semente e inicia-se a busca novamente, mas desta vez em direção oposta. Repete-se o processo até se encontrar o novo externo.

A seleção dos vetores que formarão o tubo é feita escolhendo todos os vetores no interior de uma vizinhança tubular, previamente determinada, do esqueleto obtido.

### ***Tracking: Janela Fixa ou Janela Lagrangiana.***

O volume de dados pode ser bem reduzido considerando-se somente os dados em uma região cúbica do espaço (“janela”) contendo a estrutura de interesse.

Aqui temos duas possibilidades. Usar uma janela fixa e suficientemente grande para conter o tubo durante um período de tempo especificado. Aplicando limiares de corte para os dados podemos enfocar apenas o tubo durante a sua passagem pela janela.

Outra possibilidade é usar Janelas Lagrangeanas em lugar de janelas fixas, as quais se moveriam de acordo com a velocidade média do fluido no interior da janela.

### ***Busca em Espiral***

Uma vez que um tubo com seu esqueleto foi localizado no instante  $t = t_0$ , o problema é encontrar um segmento do esqueleto no instante seguinte ( $t = t_0 + \Delta t$ ) sob a suposição de que (1) ele será aproximadamente paralelo ao seu correspondente em  $t = t_0$  e (2) o deslocamento do tubo durante o intervalo  $\Delta t$  é pequeno. Estas suposições são corretas se o intervalo de tempo  $\Delta t$  for suficientemente pequeno.

Desta forma, o próximo passo assume características bidimensionais. Mais especificamente tomaremos o segmento central do esqueleto e faremos a partir deste uma busca em espiral para localizar sua posição no instante  $t = t_0 + \Delta t$ .

Seja então o segmento central no instante  $t = t_0$  copiado para o instante  $t = t_0 + \Delta t$ . Tomemos um plano perpendicular a este segmento e vamos definir um sistema de coordenadas polares com origem no ponto de interseção do plano com o segmento. Seja uma espiral de Arquimedes  $r = a\theta$ ,  $a > 0$ , neste plano sobre a qual definimos um conjunto de pontos da forma:  $\{(r_n, \theta_n), n = 1, \dots, N\}$ .

Para cada um destes pontos tomemos um cilindro perpendicular ao plano. A posição do segmento no instante  $t = t_0 + \Delta t$  é então determinada pelo cilindro onde a vorticidade média tiver valor máximo. Se a busca falhar para o segmento central, usa-se os segmentos adjacentes para encontrar a nova posição do esqueleto. A partir do segmento assim obtido, repete-se o processo descrito na subseção que descreve a extração de um esqueleto de um tubo, agora no instante  $t = t_0 + \Delta t$ , para a obtenção de toda a estrutura neste instante.

### ***Preditor-Corretor***

A técnica apresentada em [Banks-Singer (1995)] visa a detecção dos vórtices a partir dos campos de pressão e vorticidade.

Este método está baseado na heurística justificada na seção 2.7: um ponto do vórtice apresenta alta vorticidade e baixa pressão.

Partindo desta regra, o método descrito a seguir vai construir o core em duas etapas. A partir de um ponto previamente determinado, um novo ponto do core é predito via uma integração na direção do campo de vorticidade. Em seguida o ponto predito é corrigido para um mínimo local de pressão sobre um plano ortogonal ao campo de vorticidade local.

A idéia de usar um plano de corte é motivada pela idéia central comumente aceita para a estrutura de vórtice: existência de padrões circulares ou espirais para as linhas de corrente mapeadas sobre um plano perpendicular ao core do vórtice.

O método que vamos propor no capítulo 6 tem muito em comum com esta técnica e por isto faremos uma análise detalhada da mesma a seguir.

### ***Pontos Sementes***

Para determinar os pontos semente, o método faz uso de limiares de corte obtidos, por exemplo, a partir de uma análise estatística do conjunto de dados.

Inicialmente procuram-se pontos com alta vorticidade e baixa pressão, usando os limiares obtidos. Esta busca é feita em planos perpendiculares à direção principal do escoamento. Em cada plano, o valor da vorticidade e pressão são comparados com os limiares previamente estabelecidos.

Identificado um ponto, o passo seguinte é corrigi-lo para um mínimo local de pressão sobre um plano perpendicular ao vetor vorticidade no ponto obtido. Este novo ponto será o ponto semente a ser utilizado na construção do core do vórtice.

### ***Construção do Core***

Uma vez que um ponto semente foi detectado pelo método descrito no item anterior o *core* do vórtice é construído por um processo de continuação. Neste processo, o próximo ponto do vórtice é predito integrando-se ao longo do vetor vorticidade (Figura 5.28) no ponto semente. Se simplesmente repetimos esse processo a partir do novo ponto podemos nos afastar do *core*. Assim, uma etapa de correção se faz necessária.

A idéia é observar na expressão (2.53) que o oposto do gradiente da pressão tem um papel essencial para a aceleração centrípeta. Assim, se um ponto está no *core*, esperamos que a pressão neste ponto seja baixa. Mais ainda, postula-se em [Banks-Singer (1995)] que a pressão sobre um plano perpendicular ao *core* tenha um mínimo local de pressão no ponto de interseção do plano com o *core*.

Desse modo, seja  $p_i$  um ponto sobre o *core* (podendo ser o ponto semente) e  $\bar{p}_{i+1}$  o ponto predito integrando-se ao longo do vetor vorticidade  $\vec{\xi}_i$  calculado em  $p_i$  (figura 5.28(a)-(b)). Seja  $P$  o plano por  $\bar{p}_{i+1}$  que é normal à vorticidade ( $\vec{\xi}_{i+1}$ ) neste ponto.  $\bar{p}_{i+1}$  é então corrigido para o ponto  $p_{i+1}$  que corresponde a um mínimo local da pressão restrita ao plano  $P$  (figura 5.28(d)).

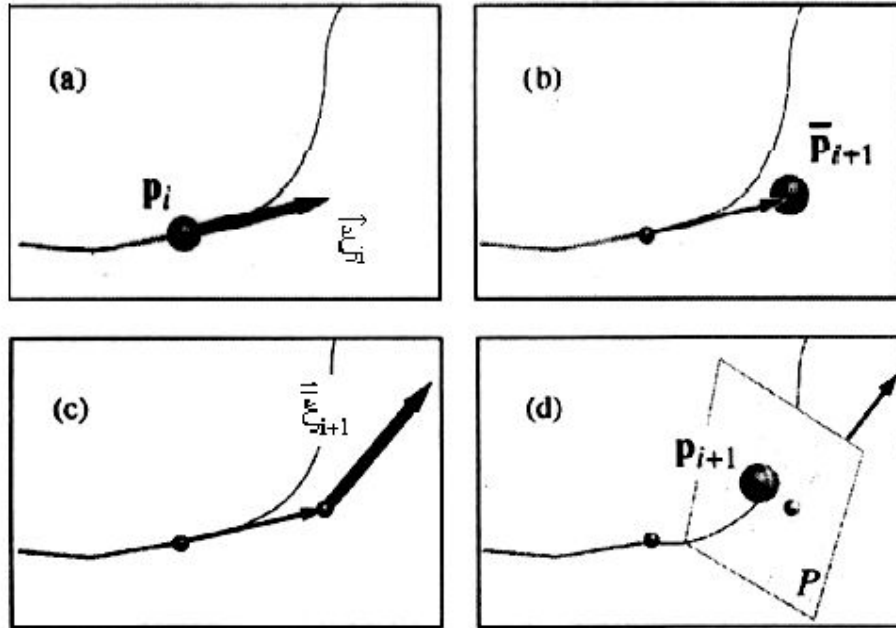


Figura 5.28 – Etapas da construção do core: predição (a)-(b) e correção (c)-(d).

Este procedimento tem duas características importantes. Suponha que tomemos um ponto  $p_j$  próximo do ponto  $p_i$  como ponto semente. Uma vez que estamos em uma região com alta vorticidade esperamos um certo alinhamento do vetor vorticidade (seção 2.6). Logo, o ponto predito  $\bar{p}_{j+1}$  estará próximo do ponto  $\bar{p}_{i+1}$  da Figura 5.28. Em seguida, a etapa de correção irá aproximar este ponto ainda mais do *core* procurado. Esta observação mostra a robustez do método no que diz respeito à construção do *core* e será útil a seguir.

### **Encontrando pontos Extremos do Core**

Uma forma eficiente de determinar os pontos extremos do *core* é através das seções transversais do vórtice que serão definidas na seção a seguir. Quando encontramos um ponto onde esta seção tem área zero (seção nula) estamos em um extremo do *core*.

Este critério, embora eficiente na maioria dos casos, não é suficiente. Podemos observar situações em que o *core* desenvolve uma trajetória muito longa o que pode dar origem a comportamentos duvidosos.

Para evitar problemas deste tipo, em [Banks-Singer (1995)] limita-se o comprimento de arco total ao longo do *core* ao dobro (no máximo) da maior dimensão do domínio do escoamento.

Este critério (área zero ou comprimento de arco máximo) tem mostrado bons resultados computacionais.

### **Filamentos Conectando Vórtices**

Existem situações em que é útil continuar o *core* para além dos extremos obtidos pelo método acima. Por exemplo, suponha que um vórtice possui uma região com baixa intensidade de vorticidade separando duas regiões de alta intensidade. Neste caso, o método de continuação do *core* pode atingir o critério de parada no momento em que é encontrada a região de baixa vorticidade o que vai conduzir a um resultado incorreto uma vez que as três regiões formam um único vórtice.

Assim, pode ser interessante continuar o *core* para além de um extremo, determinado pelo critério acima, para identificar um filamento que conecta regiões de alta intensidade. A questão é quais critérios usar para decidir se um tal filamento é ou não parte da estrutura de um vórtice.

Durante a continuação para além de um extremo, o *core* pode interceptar uma fronteira do domínio, entrar em um novo vórtice, ou parar em um ponto qualquer pelo critério de comprimento máximo. No primeiro e no terceiro casos, o segmento do *core* além do extremo deve ser descartado uma vez que temos um filamento com seção nula e que não conecta dois vórtices. No segundo caso a decisão dependerá de algum critério eficiente.

A idéia aqui é usar as conseqüências do alinhamento do campo de vorticidade numa vizinhança do *core*. Vejamos, por exemplo, a Figura 5.29(a) abaixo. Nesta figura o *core* deixa o primeiro vórtice no ponto  $p_1$  e após  $n$  passos (preditor + corretor) encontra outro vórtice no ponto  $p_2$ .

Se o filamento  $p_1 p_2$  fizer parte do *core* então ao executarmos o esquema preditor-corretor  $n$  vezes no sentido inverso (sentido oposto da vorticidade) a partir de  $p_2$ , o alinhamento do campo implicará em retornarmos a um ponto  $p_3$  próximo de  $p_1$ .

Assim, a aplicação do método no sentido oposto é um bom método para detectar se o campo de vorticidade ao longo do filamento é suficientemente intenso para que este seja mantido. Especificamente, mantém-se o filamento  $p_1 p_2$  se:

$$\|p_1 - p_3\| \leq \frac{1}{10} \|p_1 - p_2\|.$$

Na Figura 5.29(b) temos uma situação em que a aplicação do método no sentido oposto perde a curva inicial (curva  $p_4p_5$ ) caminhando numa direção qualquer dentro do volume e parando num ponto  $p_6$ . Uma vez que a distância entre os pontos  $p_4$  e  $p_6$  é uma fração maior que 10% da distância entre  $p_4$  e  $p_5$  os pontos  $p_4$  e  $p_5$  serão considerados pontos terminais de dois vórtices distintos e o filamento  $p_4p_5$  é descartado.

A figura 5.29(c) mostra a situação em que o processo de continuação avança em um outro vórtice lateralmente terminando finalmente num ponto  $p_8$  próximo ao *core* do mesmo. No caso representado, o resultado após o mesmo número de passos na direção oposta é um ponto  $p_9$  muito distante de  $p_7$  o que mostra que o filamento  $p_7p_8$  deve ser descartado também.

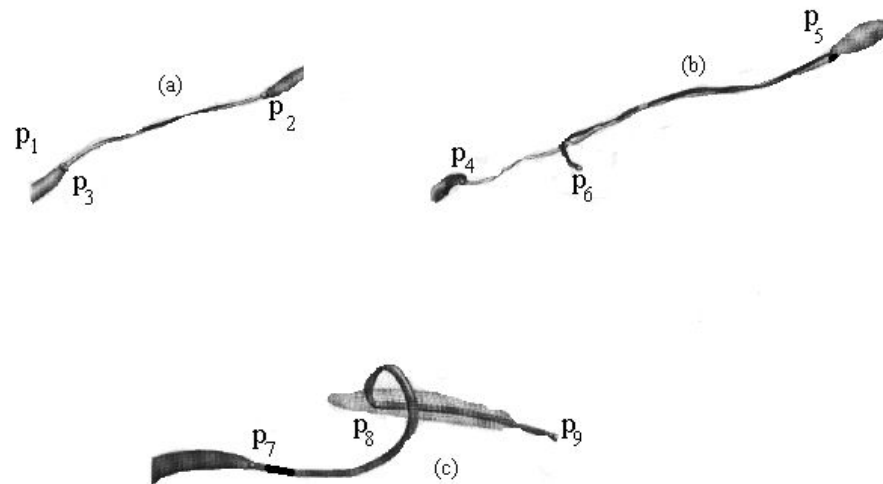


Figura 5.29 – Execução do método no sentido oposto: (a) Obtem-se aproximadamente o mesmo caminho, (b) o caminho inicial é perdido, (c) a trajetória inicial entra em outro vórtice lateralmente terminando no ponto  $P_8$  próximo ao segundo *core*. O ponto  $P_9$  é o resultado da integração no sentido oposto e mostra que o filamento  $P_7P_8$  deve ser descartado.

### **Detalhes da Implementação para os Pontos Semente**

O primeiro detalhe a ser considerado é o fato de vários pontos sementes gerarem cores muito semelhantes. Este fato também é consequência do alinhamento do vetor vorticidade em uma vizinhança do *core*.

Para evitar cálculos redundantes decorrentes da utilização de pontos sementes que geram cores muito próximas, uma possibilidade é estabelecer uma vizinhança em torno de cada ponto semente e de cada ponto do *core*, tal que qualquer ponto semente dentro desta vizinhança seja descartado. O termo constante da representação em série de Fourier para a seção transversal é tomado como o raio desta vizinhança.



Outra questão a ser considerada são pontos que ficarão fora desta vizinhança, mas que geram cores falsas que se aproximam do *core* verdadeiro à medida que vamos aplicando o preditor-corretor. Estes são casos típicos de pontos que estão próximos da borda do vórtice (Figura 5.30). Lembrando que o esquema preditor/corretor converge para o *core*, podemos resolver este problema usando novamente uma metodologia de integração para traz análoga àquela utilizada na seção anterior.

Assim, se partirmos de um suposto ponto semente  $p_0$  na fronteira do vórtice e aplicarmos o esquema estaremos sempre caminhando em direção ao *core* e portanto ao executarmos a integração para traz perderemos a trajetória inicial atingindo no final um ponto distante de  $p_0$ . A Figura 5.30 auxilia na compreensão deste fato.

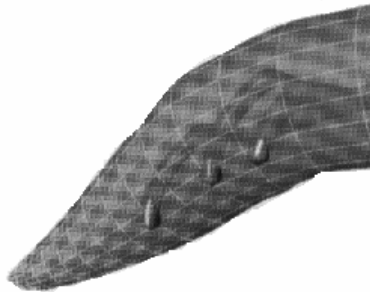


Figura 4.19 -Cores falsos gerados por pontos na fronteira do vortice.

Figura 5.30 – Cores falsas geradas por pontos na fronteira do vórtice.

### **Detalhes dos Métodos Numéricos**

Durante a execução do método será necessário a determinação da pressão e vorticidade em pontos não pertencentes à malha original. Assim tem-se que definir alguma estratégia de interpolação.

Em [Banks-Singer (1995)] usa-se interpolação polinomial de Lagrange de terceira ordem em cada uma das direções coordenadas como uma metodologia para eliminar vieses. Próximo à fronteira do domínio, onde podemos não dispor do número de pontos necessários usa-se interpolação linear.

Este esquema de interpolação é interessante do ponto de vista numérico por ser bem preciso, mas é caro do ponto de vista computacional.

A integração na etapa de predição pode ser efetuada pelo método de Runge-Kutta de quarta ordem. Esta é uma possibilidade que traz bons resultados, mas que apresenta dificuldades para a otimização deste passo. Em vez disso, em [Banks-Singer (1995)] o ponto predito é encontrado avançando a menor dimensão da célula contendo o ponto  $p_i$  da Figura 4.29(a) na direção do vetor vorticidade local.

Este procedimento assegura que pontos sucessivos nunca estarão mais de uma célula distantes um do outro, e assim, se a simulação numérica das equações do fluido for executada com a devida resolução, o core também será suficientemente preciso. Este procedimento também reduz a possibilidade de efetuar várias integrações dentro de uma mesma célula.

Para calcular o mínimo local de pressão sobre um plano de corte usa-se o método de steepest-descent [Chapra-Canale (1988)]. A menor dimensão da célula contendo o ponto  $\bar{p}_{i+1}$  é usada como escala local de comprimento na marcha ao longo da direção do gradiente.

Não há nenhuma garantia de que a fase de correção sempre convergirá para o core. Uma forma de resolver este problema é limitar o ângulo no qual a vorticidade pode mudar durante o método de correção. Assim impõe-se que o cosseno do ângulo entre a vorticidade no ponto corrigido e aquela no ponto predito seja pelo menos 0.9. No caso deste ângulo ultrapassar este limite, se descarta a fase de correção e se toma o ponto predito como o novo ponto do core.

### **Encontrando as Seções Transversais**

Uma vez detectado o core o passo seguinte é obter a forma do vórtice correspondente. A metodologia neste passo consiste em determinar seções transversais do vórtice. Um bom ponto de partida é aplicar nesta etapa a heurística até aqui utilizada. Para tal, devemos estabelecer limiares locais de vorticidade e pressão para podermos incorporar ao core uma região vizinha que envolva a estrutura procurada, o vórtice propriamente dito.

Feito isto, se faz uma varredura radial a partir do ponto  $p_{i+1}$  em direções paralelas ao plano  $P$  da Figura 5.29 até encontrar-se um ponto que viole o limiar da vorticidade ou da pressão previamente estabelecidos. Aos pontos extremos dos segmentos radiais obtidos nesta etapa será ajustada uma curva que representará uma secção transversal da fronteira do vórtice em  $p_{i+1}$ . A questão central agora é que tipo de ajuste utilizar.

Antes de apresentar a solução utilizada devemos fazer duas considerações. Primeiramente, um vórtice isolado geralmente assume uma forma alongada que pode ser localmente aproximada por um cilindro. No entanto, quando vórtices interagem, as seções transversais são deformadas e assim precisamos de um tipo de ajuste que possa admitir curvas mais gerais. Uma possibilidade que tem apresentado bons resultados é uma série de Fourier truncada para a distância radial  $r$  a partir de  $p_{i+1}$ :

$$r(t) = a_0 + a_1 \cos(t) + b_1 \sin(t) + a_2 \cos(2t) + b_2 \sin(2t), 0 \leq t \leq 2\pi.$$

A eficiência desta representação é um fato notável, embora deva ser tomada com cuidado, pois pode ser dependente das características do fluido e, portanto, variar de um conjunto de dados para outro.

A interação entre dois vórtices traz também a necessidade de um critério para definir a qual vórtice pertence um ponto da região de interação. Uma possibilidade neste sentido é retomar a interpretação da vorticidade como proporcional à velocidade angular dada por (2.34). Por esta interpretação se a vorticidade em um ponto encontrado fizer um ângulo maior que 90 graus com o *core* isto indicará que o fluido nesta posição radial gira em uma direção diferente daquela definida pelo *core*, não podendo fazer parte do vórtice em questão. Assim, o problema de decidir a que vórtice pertence um ponto é resolvido limitando-se o ângulo entre o vetor vorticidade no ponto do *core* e o vetor vorticidade em qualquer posição radial.

Ao calcularmos as seções transversais em cada um dos pontos  $p_{i+1}$  teremos uma seqüência de seções que serão conectadas por uma malha quadrilateral para em seguida efetuar-se o *rendering*.

### **5.6. Técnica Baseada em Extração de Características**

Esta técnica [Samtaney et al. (1994)] visa extrair regiões coerentes e amorfas de campos escalares ou vetoriais, bi e tridimensionais. Portanto, é uma técnica de aplicações mais gerais que as descritas acima, mas cuja aplicação para visualização de vórtices em CFD tem apresentado bons resultados. É uma metodologia híbrida, resultado da combinação de técnicas provenientes de áreas como visão computacional, processamento de imagem, computação gráfica e geometria computacional.

Se os campos em questão forem não estacionários então podemos aplicar essa metodologia para cada instante de tempo e então acompanhar a evolução temporal das estruturas detectadas. Nesta apresentação vamos nos ater ao caso de um único campo escalar (magnitude da vorticidade ou pressão).

Vejamos alguns detalhes desta técnica.

#### **5.6.1 Extrair Regiões de Interesse.**

Chamaremos objeto a uma estrutura consistindo de um conjunto de pontos vizinhos cujo campo escalar associado possui valores acima (ou abaixo) de um certo limiar, juntamente com uma fronteira associada. Portanto, um vórtice é um objeto no contexto desta técnica.

Neste caso também, o algoritmo parte de um ponto semente e de um limiar de corte previamente estabelecido para testar se os pontos 3D vizinhos pertencem ou não ao vórtice procurado. Desta forma se faz um crescimento da região pela inclusão de pontos que satisfazem o limiar previamente determinado. Quando é atingido um ponto abaixo do limiar escolhido este crescimento é interrompido na direção correspondente.

O conjunto de dados fica assim particionado em “objetos” e *background* e o conjunto de pontos que constituem os objetos é armazenado em uma estrutura de dados para posterior manipulação( por exemplo, em uma octree [Wilhelms-Gelder(1992)]).

### 5.6.2 Atributos dos Objetos

Atributos são úteis para quantificar e extrair regiões bem como para *tracking*. Alguns atributos comumente utilizados estão definidos abaixo.

Massa:

Dado um campo escalar  $\omega$  e um conjunto  $\Omega = \{x | \omega(x) \geq \text{limiar}_\omega\}$  então definimos massa como sendo:

$$W = \int_{\Omega} \omega(x) d\Omega.$$

Centróide:

$\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_d)$ , tal que:

$$\bar{x}_k = W^{-1} \int_{\Omega} \omega(x) x_k d\Omega, \quad (5.17)$$

onde  $k=1,2,\dots,d$ ; com  $d$  sendo a dimensão do espaço em questão e  $x_k$  é a coordenada espacial  $k$ .

Máximos:

Uma região extraída pode ter vários máximos locais. Estes podem ser detectados usando-se um algoritmo de crescimento de regiões [Silver-Zabusky (1993)].

Volume:

O número de pontos (nós da malha) contidos numa região.

Momentos:

De maneira geral, nós definimos o momento de ordem  $N = p_1 + p_2 + \dots + p_d$ ,  $0 \leq p_i \leq d$ , de  $\omega$  pela expressão:

$$I_{p_1 p_2 \dots p_d} = W^{-1} \int_{\Omega} \omega(x) (x_1 - \bar{x}_1)^{p_1} (x_2 - \bar{x}_2)^{p_2} \dots (x_d - \bar{x}_d)^{p_d} d\Omega,$$

onde  $\bar{x}_i, i = 1, \dots, d$ , são as coordenadas do centróide calculadas por (5.17).

Os momentos mais comumente usados são o momento de inércia ( $0 \leq p_i \leq 1, i = 1, \dots, d$ ) e os momentos de segunda ordem ( $0 \leq p_i \leq 2, i = 1, \dots, d$ ). Os momentos de inércia e de segunda ordem são boas aproximações para regiões correspondentes a esferas deformáveis, mas eles não capturam convenientemente estruturas tipo tubo ou folhas. Momentos de ordem mais alta podem oferecer uma melhor aproximação, bem como uma medida de erro para os momentos de segunda ordem.

Bordas:

Os pontos localizados nas fronteira das regiões extraídas correspondem em geral a isosuperfícies.

Esqueletos:

São úteis para representar regiões de forma compacta(o core pode ser pensado como um esqueleto do vórtice).

### 5.6.3 Tracking

*Tracking* pode ser efetuado na forma de um pré-processamento ou de um pós-processamento. No primeiro caso, as regiões extraídas de um conjunto de dados são usadas para buscar as regiões no conjunto de dados seguintes (ver [Villasenor-Vincent (1992)] para um exemplo).

No pós-processamento as regiões são inicialmente extraídas de todo o conjunto de dados para então serem correlacionadas.

No que segue nos concentraremos no pós-processamento. A suposição mais básica aqui é a de que o intervalo de tempo entre conjuntos de dados sucessivos ( $\Delta t = t_{i+1} - t_i$ ) é pequeno. Esta suposição tem sentido uma vez que nas simulações envolvendo vorticidade e turbulência é necessário uma discretização bastante fina do tempo para se obter bons resultados.

### 5.6.4 Interações

Uma forma de caracterizar a evolução temporal dos objetos é através dos possíveis fenômenos aos quais os mesmos podem estar sujeitos. Em geral, e particularmente no caso dos vórtices, temos as seguintes possibilidades para a evolução dos objetos:

#### **Continuação:**

Um objeto presente no instante  $t_i$  continua presente no instante  $t_{i+1}$ , com possíveis alterações de forma e posição no espaço.

Para determinar se um dado objeto em um conjunto de dados continua sendo um objeto no conjunto de dados subsequente usamos o seguinte critério:

O Objeto  $O_B^{i+1}$  (objeto B no instante  $t=i+1$ ) corresponde ao objeto  $O_A^i$  se:

$$\{O_B^{i+1} | \text{overlap}(O_A^i, O_B^{i+1}) > \text{overlap}(O_A^i, O_G^{i+1}), \forall O_G^{i+1} \in O_N^{i+1}, \text{com } G \neq B\},$$

onde  $O_N^{i+1}$  indica um conjunto de  $N$  objetos extraídos no instante  $t = i+1$ .

Além disso impõe-se que:  $\text{overlap}(O_A^i, O_B^{i+1}) > T_{\text{over}}$ , onde  $T_{\text{over}}$  é um limiar para o tamanho da interseção.

#### **Bifurcação:**

Um objeto se separa em duas ou mais subestruturas.

Usa-se a seguinte definição:

#### Definição:

Se um grupo de  $N > 1$  objetos,  $O_N^{i+1}$  adicionados são equivalentes ao objeto  $O_A^i$ , então o objeto  $O_A^i$  bifurcou-se no conjunto de objetos  $O_N^{i+1}$  no instante  $i+1$ .

Correspondentemente tem-se o seguinte critério:

O objeto  $O_A^i$  bifurcou-se no conjunto  $O_N^{i+1}$  se:

$$\sum_{j=1}^N \text{atributo}(O_j^{i+1}) - \text{atributo}(O_A^i) < T_{\text{atributo}},$$

onde  $T_{\text{atributo}}$  é o limiar escolhido para o atributo.

**Amalgamação:**

Dois ou mais objetos se fundem em um único.

É exatamente o inverso do processo acima.

**Criação:**

Novos objetos aparecem no instante  $t = i+1$ .

Este processo ocorre quando um objeto não pode ser correlacionado com qualquer objeto no instante anterior.

Definição:

Se um objeto  $O_A^{i+1}$  não é uma continuação de algum objeto no instante  $t_i$  e não foi classificado como sendo uma amalgamação ou parte de uma bifurcação então  $O_A^{i+1}$  é um novo objeto.

**Dissipação:**

Um objeto sofre dissipação quando não existe um objeto no instante seguinte que pode ser correlacionado de alguma forma com ele.

Definição:

Se um objeto  $O_A^i$  não sofreu bifurcação, amalgamação ou continuação, então  $O_A^i$  sofreu dissipação.

A Figura 5.31 exemplifica estes fenômenos e será retomada a seguir.

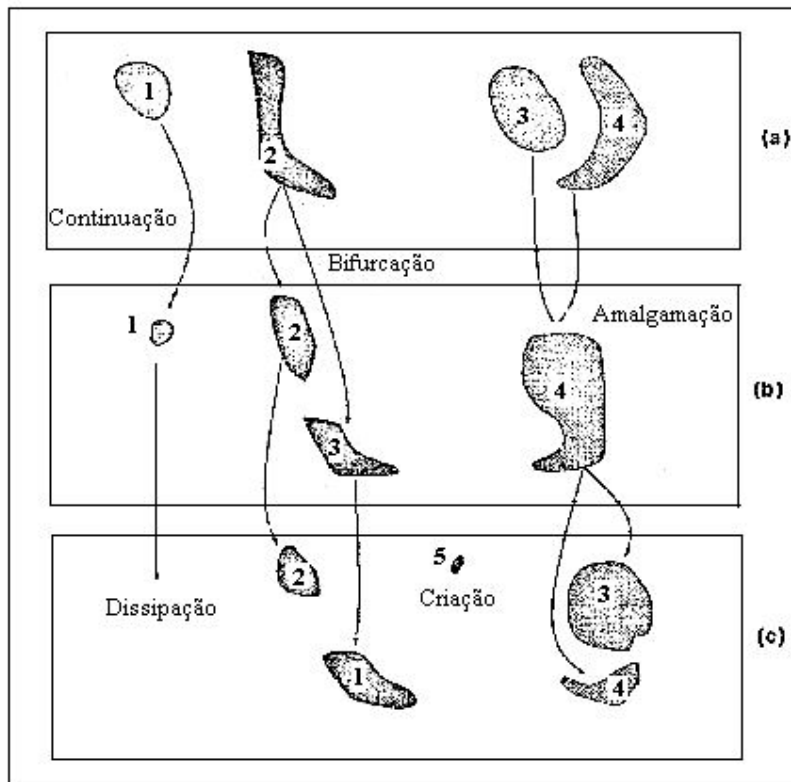


Figura 5.31 – Possibilidades para a evolução temporal de uma estrutura: continuação, bifurcação, amalgamação, dissipação e criação.

### 5.6.5 Visualização e Tracking

O método proposto em [Samtaney et al. (1994)] pode ser resumido nos seguintes passos.

Extrair os objetos (vórtices) de cada conjunto de dados, numerar cada objeto e criar uma lista de atributos,

Para cada objeto (centróide) no conjunto de dados  $i$ , encontrar os objetos (centróides) vizinhos no conjunto de dados  $i+1$  e efetuar o teste (4.31) para detectar continuação. Se um *matching* é encontrado, marcar os objetos correspondentes e removê-los da lista.

Após a remoção de todos os objetos que sofrem continuação devemos testar combinações de dois ou mais objetos nos instantes  $t_i$  e  $t_{i+1}$  para detectar amalgamações e bifurcações.

Os objetos nos instantes  $t_{i+1}$  e  $t_i$  que não foram retiradas da lista nos passos 2 e 3 corresponderão a criações e dissipações.

Uma vez que o número de comparações será de ordem exponencial devemos tomar certos cuidados para limitar o número de testes. No caso dos vórtices deve-se

observar que as interações entre vórtices só podem ocorrer entre vizinhos e, portanto, as distâncias entre os centróides devem ser tomadas como um parâmetro para evitar comparações desnecessárias.

O resultado do algoritmo resumido nos passos 1-4 acima pode ser representado em um grafo direcionado e acíclico (DAG). A Figura 5.32 mostra este grafo para a evolução das regiões na Figura 5.31 acima.

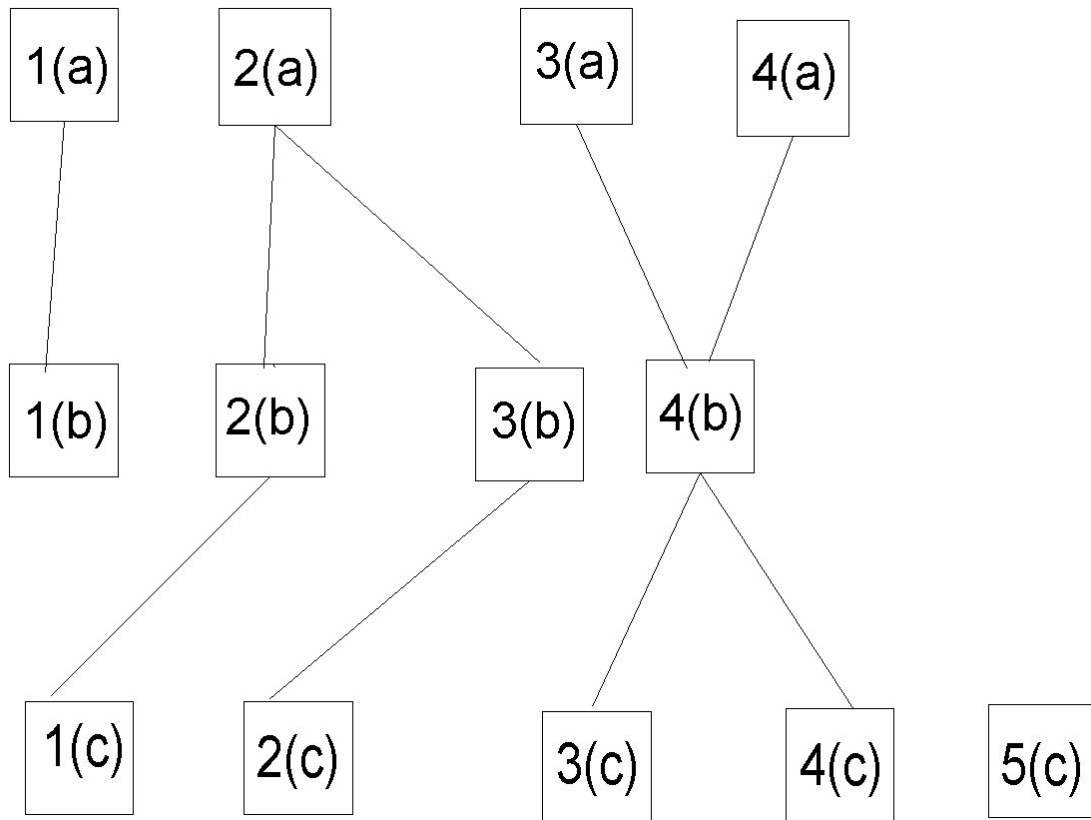


Figura 5.32 – Grafo acíclico direto da evolução dos objetos representados na Figura 5.31.

A partir do DAG e dos atributos correspondentes a cada objeto (isosuperfície por exemplo) pode-se escolher uma região específica e acompanhar sua evolução no tempo.



## Capítulo 6

### Animação Computacional de Fluidos

#### 6.1. Introdução

Animação computacional é uma subárea da computação gráfica que envolve técnicas de *rendering*, modelagem geométrica e modelagem de sistemas físicos. Independente da aplicação em foco, o objetivo final nesta área é reproduzir a dinâmica dos objetos de uma cena com o realismo necessário [Kerlow (1996)], [Taylor (1996)], [Watt (1991)].

No caso específico da animação de fluidos, o *pipeline* completo envolve:

- Modelagem geométrica da cena,
- O modelo matemático dos fluidos em questão,
- A simulação computacional deste modelo,
- Rendering.

As técnicas de modelagem geométrica podem envolver modelos poligonais da fronteira do domínio por onde o fluido irá escoar (paredes, objetos rígidos, etc), superfícies paramétricas e texturas para a representação da superfície visível do fluido e malhas regulares ou irregulares para discretização do domínio e solução numérica das equações.

O modelo matemático é geralmente dado pelas equações de Navier-Stokes, apresentadas no Capítulo 2. A simulação computacional pode ser feita por uma das técnicas descritas no Capítulo 4. Os campos gerados nestas simulações deverão receber um tratamento gráfico via métodos de visualização científica e computação gráfica descritos no Capítulo 5.

O texto abaixo não tem por objetivo fazer uma revisão exaustiva dos trabalhos em animação computacional de fluidos. Como já apontado no Capítulo 1, os métodos encontrados na literatura para animação de fluidos via modelos de DFC, são fundamentados nas equações de Navier-Stokes, com técnicas de discretização baseadas em diferença finitas implícitas [Stam (1999)] e explícitas [Foster-Metaxas (1997)], bem como em métodos Lagrangeanos tais como Método das Características [Stam (1999)], *Smoothed Particle Hydrodynamics (SPH)* [Liu et al. (2003)], [Müller et al. (2003)] e *Moving-Particle Semi-Implicit (MPS)* [Prempo et al. (2003)].

Dentre estes trabalhos, consideramos que as duas linhas principais são aquelas que utilizam métodos de diferenças finitas e aquelas baseadas no SPH (seção 4.6). Esta observação fundamenta-se na generalidade destas técnicas, uma vez que a base matemática e física envolvidas permitem, ao menos potencialmente, a simulação de qualquer cena envolvendo fluidos, com o realismo satisfatório.

A partir deste ponto de vista, descrevemos inicialmente o trabalho de Foster e Metaxas, um marco nesta linha de pesquisa [Foster-Metaxas (1997)]. Este trabalho pode ser considerado o precursor daquele descrito na seção 6.3, devido a Jos Stam [Stam (1999)], o qual propõe uma solução para o custo computacional elevado do método de Foster e Metaxas para malhas com alta resolução e/ou velocidades altas. Em seguida, descrevemos o trabalho de Patrick Witting [Witting (1999)], o qual considera um modelo matemático para fluidos (gases) bem mais completo que os dois anteriores. Neste caso, o fluido (gás) é compressível e o modelo matemático envolve também equações termodinâmicas. Este método usa diferenças finitas no espaço e Runge-Kutta para interação no tempo (ver capítulo 4).

Finalmente, apresentamos o trabalho [Müller et al. (2003)], onde as equações de Navier-Stokes são resolvidas via SPH. Uma vantagem deste método com relação aos anteriores é sua independência de malhas previamente definidas, o que já discutimos na seção 4.6 onde apresentamos as principais características desta técnica.

Na fase de *rendering*, os trabalhos descritos a seguir fazem uso *volume rendering* [Foster-Metaxas (1997), Witting (1999)], *LIC* [Witting (1999)], e *splatting e marching-cubes* [Müller et al. (2003)]. Estes métodos de visualização foram descritos no capítulo anterior.

## **6.2. Animação via Modelo para Gases Aquecidos**

Em [Foster-Metaxas (1997)] encontramos um modelo em DFC voltado à animação de gases aquecidos e turbulentos. Neste trabalho tratamos os gases como fluidos incompressíveis, o que é uma simplificação drástica destes sistemas físicos, se pensada do ponto de vista de aplicações nas ciências ou engenharias. Como já ressaltado anteriormente, isto faz sentido porque o objetivo final neste caso é a criação de um efeito visual, e não a descrição de um fenômeno físico. Assim, o grau de sofisticação de um modelo depende apenas do tipo de efeito visual que se deseja criar.

Tais simplificações têm também um outro aspecto. Os modelos em animação computacional devem capazes de gerar imagens em tempo real, pois o objetivo final de uma pesquisa nesta área é disponibilizar novos aplicativos gráficos para os animadores projetarem suas cenas. O custo computacional dos algoritmos desenvolvidos é um fator fundamental.

Algoritmos que demandam plataformas computacionais não-convencionais implicam em aplicativos de uso muito restrito. Assim, existe um compromisso entre realismo e custo computacional. A busca de soluções ótimas, ou seja, algoritmos que ofereçam o realismo necessário, com custo computacional aceitável, é um grande desafio do estado da arte da animação de fluidos.

A partir destas considerações, Foster e Metaxas propuseram em [Foster-Metaxas (1997)] o seguinte modelo para a evolução da velocidade  $\vec{u}$  de um gás aquecido:

$$\frac{\partial \bar{u}}{\partial t} = \nu \nabla \cdot (\nabla \bar{u}) - (\bar{u} \cdot \nabla) \bar{u} - \frac{1}{\rho} \nabla p + \bar{f}_T, \quad (6.1)$$

onde  $p$  é o campo de pressão no interior do fluido, e os operadores  $\nabla$  e  $(\nabla \cdot)$  representam os operadores gradiente e divergente, respectivamente, como já definidos anteriormente (ver Apêndice A). A grandeza  $\rho$ , densidade do fluido, é tomada como constante ( $\rho = 1$ ). O campo vetorial  $\bar{f}_T$  é um campo de forças decorrente da variação da temperatura ( $T$ ) no interior do fluido, a qual é modelado pela seguinte equação diferencial:

$$\frac{\partial T}{\partial t} = \lambda \nabla \cdot (\nabla T) - \nabla \cdot (T \bar{u}), \quad (6.2)$$

onde  $\lambda$  é um coeficiente de difusão. Esta expressão é um caso particular da Lei de Conservação dada pela equação (2.28), quando  $\bar{f}_e = 0$ ,  $q_H = 0$ ,  $E = T$  e o efeito das forças de atrito internas forem desprezíveis para a variação da temperatura, ou seja,  $\bar{\nabla} \cdot (\bar{\sigma} \cdot \bar{v})$  puder ser desconsiderado na equação de conservação da energia. A força  $\bar{f}_T$  é dada por:

$$\bar{f}_T = -\beta \bar{g}_v (T_0 - T), \quad (6.3)$$

onde  $\bar{g}_v$  é o campo gravitacional (vertical),  $\beta$  é o coeficiente de expansão térmica e  $T_0$  é a temperatura inicial de referência.

Deve-se acrescentar a estas equações a condição de incompressibilidade:

$$\nabla \cdot \bar{u} = 0, \quad (6.4)$$

uma vez que supõe-se fluidos incompressíveis neste modelo.

Vejamos alguns detalhes do método proposto em [Foster-Metaxas (1997)] para animação de fluidos via as equações (6.1)-(6.4).

### 6.2.1 Modelagem do Ambiente (Cena)

Em [Foster-Metaxas (1997)] tanto a fronteira dos objetos (rígidos) que compõe a cena, quanto o domínio de escoamento do fluido são modelados via *voxels*. Assim, faz-se uma decomposição regular de todo o ambiente da cena usando uma malha regular (seção 4.2), contendo *voxels* interiores aos objetos e *voxels* interiores ao domínio de escoamento do fluido. *Voxels* que possuírem faces interiores e faces exteriores ao domínio de escoamento serão denominados *voxels* de fronteira.

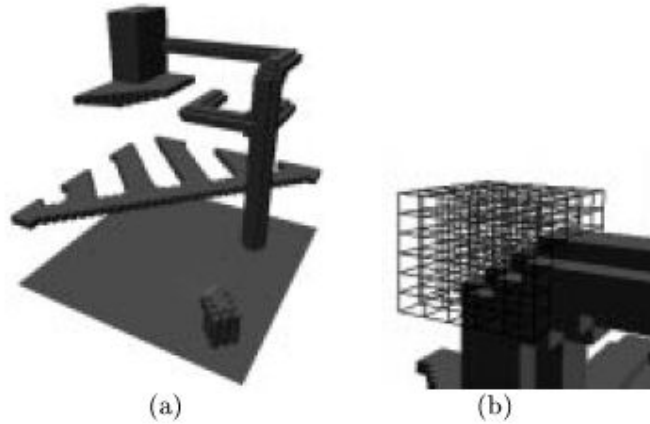


Figura 6.1 - (metaxas-fig1.bmp)

Os campos escalares e vetoriais de interesse serão conhecidos nos nós da malha  $(i, j, k)$  da malha. Se necessitarmos de valores destes campos em qualquer outro ponto do domínio de escoamento, então algum tipo de interpolação deverá ser usada.

### 6.2.2 Discretização das Equações

Os esquemas de diferenças finitas utilizados são descritos a seguir. Seguindo a notação usada no Capítulo 2, dada uma grandeza física, a temperatura  $T$ , por exemplo, indicaremos por  $T_{i,j,k}^n$ , o valor da temperatura, no nó  $(i, j, k)$  da malha, no instante  $n\Delta t$  da simulação (estaremos supondo o instante inicial  $t_0 = 0$ ). Em [Foster-Metaxas (1997)] são utilizados os seguintes esquemas para aproximar as derivadas parciais de primeira e segunda ordem que aparecem nas equações (6.1)-(6.4):

$$\frac{\partial T}{\partial t} = \frac{T(t + \Delta t, x, y, z) - T(t, x, y, z)}{\Delta t}, \quad (6.5)$$

$$\frac{\partial T}{\partial x} = \frac{T(x + h, y, z) - T(x - h, y, z)}{2h}, \quad (6.6)$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x + h, y, z) - 2T(x, y, z) + T(x - h, y, z)}{h^2}, \quad (6.7)$$

Estas expressões correspondem à primeira, terceira e oitava linhas da Tabela 4.1, seção 4.3. Usando a notação acima, teremos:

$$\frac{\partial T((n+1)\Delta t, i, j, k)}{\partial t} = \frac{T_{ijk}^{(n+1)\Delta t} - T_{ijk}^{n\Delta t}}{\Delta t}. \quad (6.8)$$

com equações análogas para as derivadas espaciais.

Desta forma, a equação pode ser discretizada na forma:

$$\begin{aligned}
T_{ijk}^{n+1} = T_{i,j,k}^n + \Delta t \left\{ \left( \frac{1}{h} \right) \left[ (Tu)_{i-1/2,j,k}^n - (Tu)_{i+1/2,j,k}^n + (Tv)_{i,j-1/2,k}^n - (Tv)_{i,j+1/2,k}^n + \right. \right. \\
+ (Tw)_{i,j,k-1/2}^n - (Tw)_{i,j,k+1/2}^n \left. \right] + \frac{\lambda}{h^2} \left[ (T_{i+1,j,k}^n - 2T_{i,j,k}^n + T_{i-1,j,k}^n) \right. \\
\left. + (T_{i,j+1,k}^n - 2T_{i,j,k}^n + T_{i,j-1,k}^n) + (T_{i,j,k+1}^n - 2T_{i,j,k}^n + T_{i,j,k-1}^n) \right] \left. \right\}, \quad (6.9)
\end{aligned}$$

onde o termo  $(Tu)_{i+1/2,j,k}^n$  é dado por:

$$(Tu)_{i+1/2,j,k}^n = \frac{1}{2} u_{i+1/2,j,k}^n (T_{i,j,k} + T_{i+1,j,k}), \quad (6.10)$$

e a velocidade  $u_{i+1/2,j,k}^n$  é obtida via interpolação.

### 6.2.3 Conservação da Massa

Pela Tabela 4.1, vemos que as expressões (6.6)-(6.7) acima envolvem um erro de ordem  $O(h^2)$ . Este erro implica que a solução numérica pode não preservar a conservação da massa, a qual é expressa por:

$$\vec{\nabla} \cdot \vec{v} = 0, \quad (6.11)$$

estando implícita na equação (6.1), de acordo com a exposição da seção 2.2.

Assim, é preciso utilizar um esquema de correção para garantir esta condição. Primeiramente, usa-se diferenças finitas centrais para aproximar a equação (6.11):

$$(\vec{\nabla} \cdot \vec{v})_{i,j,k} = \frac{1}{h} (v_{i+1/2,j,k}^1 - v_{i-1/2,j,k}^1 + v_{i,j+1/2,k}^2 - v_{i,j-1/2,k}^2 + v_{i,j,k+1/2}^3 - v_{i,j,k-1/2}^3), \quad (6.12)$$

onde  $(\vec{\nabla} \cdot \vec{v})_{i,j,k}$  é o valor de  $\vec{\nabla} \cdot \vec{v}$  na posição  $(i, j, k)$  e  $(v^1, v^2, v^3)$  representam as componentes do vetor velocidade. A conservação da massa implica que o campo definido pela expressão (6.12) seja nulo em todo o domínio, ou seja, precisamos resolver uma equação de Laplace em 3D. Contudo, podemos usar a teoria desenvolvida na seção 2.8 (Decomposição de Helmholtz) para resolver este problema. Para isto, seja então a equação (2.77) dada por:

$$\vec{\nabla} \cdot \vec{v} = \Delta \phi \Rightarrow \left( \frac{\partial v^1}{\partial x} - \frac{\partial^2 \phi}{\partial x^2} \right) + \left( \frac{\partial v^2}{\partial y} - \frac{\partial^2 \phi}{\partial y^2} \right) + \left( \frac{\partial v^3}{\partial z} - \frac{\partial^2 \phi}{\partial z^2} \right) = 0. \quad (6.13)$$

Desta equação, fica claro que uma forma de satisfazermos a condição (6.11) em todo o domínio será corrigir a velocidade  $\vec{v}$  segundo as expressões:

$$v^1_{new} = v^1 - \frac{\partial \phi}{\partial x}, \quad (6.14)$$

$$v^2_{new} = v^2 - \frac{\partial \phi}{\partial y}, \quad (6.15)$$

$$v^3_{new} = v^3 - \frac{\partial \phi}{\partial z}, \quad (6.16)$$

pois, podemos verificar que  $\vec{\nabla} \cdot \vec{v}_{new} = 0$ . Assim, temos que encontrar um potencial  $\phi$  que satisfaça a equação (6.13) e em seguida usar as expressões (6.14)-(6.16) para obter o campo de velocidades corrigido. O campo  $\phi$  desejado pode ser calculado por um esquema iterativo obtido como segue. Por (6.13), estamos procurando um campo  $\phi$  que satisfaça a expressão:

$$0 \approx -(\vec{\nabla} \cdot \vec{v})_{i,j,k} + \left( \frac{\phi_{i+1,j,k}^n - \phi_{i-1,j,k}^n + \phi_{i,j+1,k}^n - \phi_{i,j-1,k}^n + \phi_{i,j,k+1}^n - \phi_{i,j,k-1}^n - 6\phi_{i,j,k}^n}{h^2} \right).$$

onde  $(\vec{\nabla} \cdot \vec{v})_{i,j,k}$  é dado pela equação (6.12) e usamos uma expressão análoga a (6.7) para aproximar o Laplaciano de  $\phi$  via diferenças finitas centrais. Multiplicando ambos os membros desta expressão por  $h^2/3$  e adicionando  $\phi_{i,j,k}^n$  obtemos a expressão:

$$\phi_{i,j,k}^n \approx -\frac{h^2}{3}(\vec{\nabla} \cdot \vec{v})_{i,j,k} + \frac{1}{3}(\phi_{i+1,j,k}^n - \phi_{i-1,j,k}^n + \phi_{i,j+1,k}^n - \phi_{i,j-1,k}^n + \phi_{i,j,k+1}^n - \phi_{i,j,k-1}^n) - \phi_{i,j,k}^n.$$

Assim, se utilizarmos o esquema iterativo a seguir:

$$\phi_{i,j,k}^{n+1} = -\frac{h^2}{3}(\vec{\nabla} \cdot \vec{v})_{i,j,k} + \frac{1}{3}(\phi_{i+1,j,k}^n - \phi_{i-1,j,k}^n + \phi_{i,j+1,k}^n - \phi_{i,j-1,k}^n + \phi_{i,j,k+1}^n - \phi_{i,j,k-1}^n) - \phi_{i,j,k}^n, \quad (6.17)$$

o qual será interrompido quando  $|\phi_{i,j,k}^{n+1} - \phi_{i,j,k}^n|$  for suficientemente pequeno, teremos um campo  $\phi$  com as características desejadas. Finalmente, as seguintes versões discretas para as expressões (6.14)-(6.16) fornecem a velocidade final:

$$v^1_{i,j,k} \leftarrow v^1_{i,j,k} - \frac{\phi_{i+1,j,k} - \phi_{j,ki}}{h}, \quad (6.18)$$

$$v^2_{i,j,k} \leftarrow v^2_{i,j,k} - \frac{\phi_{i,j+1,k} - \phi_{j,ki}}{h}, \quad (6.19)$$

$$v^3_{i,j,k} \leftarrow v^3_{i,j,k} - \frac{\phi_{i,j,k+1} - \phi_{j,ki}}{h}. \quad (6.20)$$

#### 6.2.4 Estabilidade Numérica

O esquema numérico apresentado na seção anterior tem a vantagem de ser facilmente implementável, mas possuem a desvantagem de serem numericamente instáveis para grandes passos de tempo.

A origem desta limitação está no termo não-linear de advecção (ou convecção) dado pela expressão  $(\vec{u} \cdot \nabla)\vec{u}$ . Se este termo for tratado numericamente via diferenças finitas, como proposto por Foster e Metaxas, obtém-se um método estável apenas para passos de tempo suficientemente pequenos. Pode-se mostrar que o intervalo de tempo  $\Delta t$  deve satisfazer:

$$\Delta t < \frac{h}{\|\vec{u}\|}, \quad (6.21)$$

onde  $h$  é a resolução da malha usada na discretização.

Portanto, para malhas com alta resolução e/ou velocidades altas, o passo no tempo deve ser muito pequeno, o que aumenta o custo computacional do algoritmo. Além desta condição, a análise linear para as equações de Navier-Stokes do problema mostra que a seguinte condição também deve ser satisfeita:

$$\nu < \left(\frac{\Delta t}{2}\right) \max\left((v^1)^2, (v^2)^2, (v^3)^2\right), \quad (6.22)$$

para que a estabilidade numérica seja mantida. Esta condição limita o valor da viscosidade cinemática do fluido.

#### 6.2.5 Condições de Contorno e Iniciais

O comportamento do gás nas paredes dos objetos sólidos da cena deve ser definido a priori para que as equações de diferenças finitas possam ser avaliadas nos pontos próximos à fronteira do domínio de escoamento do fluido. Por exemplo, na parede de um radiador de calor teríamos a componente normal da velocidade nula, pois não existe fluxo nesta direção. Quanto à componente tangencial  $\vec{v}_T$  podemos ter condições tipo “*no-slip*”, ou seja,  $\vec{v}_T = 0$ , ou condições tipo “*slip*”, quando o fluido tem liberdade para escorregar na direção tangente à fronteira do domínio. A temperatura flui livremente do radiador para o gás, assim, a temperatura inicial das células da fronteira será a mesma temperatura  $T$  da parede do radiador. Em [Foster-Metaxas (1997)], de maneira geral a velocidade na superfície de objetos rígidos é do tipo “*no-slip*” e portanto  $\vec{v} = 0$  nas células da fronteira. A temperatura na fronteira é a temperatura ambiente. A pressão nas fronteiras é dada pela pressão externa ao domínio.

Devemos também definir condições iniciais para os campos de velocidade e temperatura, uma vez que as equações (6.1)-(6.2) envolvem derivadas de primeira ordem para estes campos. Estas condições são determinadas em função do efeito que se deseja obter. Não há uma maneira sistemática para obter tais condições ficando este ponto dependente da experiência do usuário.

### 6.2.6 *Rendering*

Uma vez determinados os campos de velocidade e temperatura, passamos à fase de *rendering* cujo resultado será a seqüência de cenas, ou a animação propriamente dita. Em [Foster-Metaxas (1997)] a etapa de *rendering* é baseado no traçado de partículas, descrito pela equação 3.3 (ver seção 4.3.1 também). A densidade de partículas utilizada, ou seja, número de partículas por unidade de volume, pode depender da velocidade ou temperatura do fluido, ou de critérios subjetivos do usuário. O mesmo pode ser dito com respeito à densidade de cada partícula individualmente. Assim, uma vez definida a distribuição de partículas no instante  $t$ , a nova distribuição no instante  $t + \Delta t$  será dada por uma expressão do tipo:

$$r_i^{t+\Delta t} = r_i^t + \delta t \bar{v}_i^t, \quad (6.23)$$

onde  $r_i^t$  é a posição da partícula  $i$  no instante  $t$ ,  $\Delta t$  é o passo no tempo utilizado e  $\bar{v}_i^t$  é a velocidade do fluido na posição ocupada pela partícula  $i$  no instante  $t$ .

Finalmente, a distribuição de partículas em cada instante é projetada via *volume rendering* para a geração da imagem correspondente.

### 6.2.7 Algoritmo para Animação de Gases

O algoritmo descrito em [Foster-Metaxas (1997)] pode ser dividido em tres etapas. Na primeira, devem ser definidos dos parâmetros físicos de viscosidade, expansão térmica e difusão molecular, condições iniciais e de contorno, assim como como parâmetros numéricos. Na segunda etapa, obtemos a solução numérica das equações de fluidos (6.1)-(6.2). Finalmente, na última etapa, executa-se o *rendering* via traçado de partículas e *volume rendering*. Mais detalhadamente, temos o seguinte:

Primeira Etapa:

- a) Geração da malha regular segundo o valor de  $h$ ,
- b) Definir condições de contorno para velocidade, temperatura e pressão,
- c) Definir condições iniciais para velocidade e temperatura,
- d) Definir os parâmetros  $\nu$ ,  $\beta$  e  $\lambda$  de acordo com o efeito desejado. Segundo [Foster-Metaxas (1997)], valores da ordem de  $1/10 h$  fornecem fluxos pouco turbulentos enquanto que para valores menores, da ordem de  $1/100 h$  ou menos, a turbulência é um efeito predominante.
- e) Usar  $\Delta t$  como sendo o menor valor entre aqueles obtidos pelas expressões (6.21)-(6.22)

Segunda Etapa:



- a) Usar as equações de diferenças finitas (6.5)-(6.7) para obter a evolução temporal dos campos de interesse, ou seja, obter a velocidade e temperatura no instante  $t + \Delta t$  a partir destes campos no instante  $t$ .
- b) Aplicar as expressões (6.18)-(6.20) para corrigir o campo de velocidade obtido no item anterior
- c) Voltar ao passo (a) desta etapa.

Terceira Etapa:

- a) Definir uma distribuição inicial de partículas
- b) Evoluir esta distribuição de acordo com a equação (6.23)
- c) Para cada instante de tempo  $t$  usar volume rendering para gerar a imagem correspondente

### 6.2.8 Resultados

A imagem mostrada na Figura 6.2 demonstra a interação de gás aquecido com objetos sólidos. A versão discreta deste cenário é mostrada na Figura 6.1. O gás é forçado para dentro do ambiente impondo condições de contorno para a temperatura e velocidade correspondentes a uma válvula de pressão. A velocidade inicial do fluido na válvula é  $0.3 \text{ m/s}$  e a temperatura do mesmo é  $80^\circ \text{C}$ . Estas condições são consistentes com aquelas de um gás sendo ventilado a partir de um boiler. O resultado é a ejeção de fluido no interior da cena. Neste caso, observa-se turbulência apenas na extremidade da válvula para velocidades mais altas do fluido.

O mesmo ambiente foi usado para animar a interação entre gases emitidos a partir de três válvulas separadas. A Figura 6.3 mostra três *frames* para esta animação. As rotações causadas pelo resfriamento e mistura dos gases podem ser observadas nesta seqüência. Em ambos experimentos, partículas de prova foram introduzidas à taxa de  $2000/s$  e as imagens mostradas são o resultado do *volume rendering* a partir desta distribuição de partículas.



Figura 6.2 -

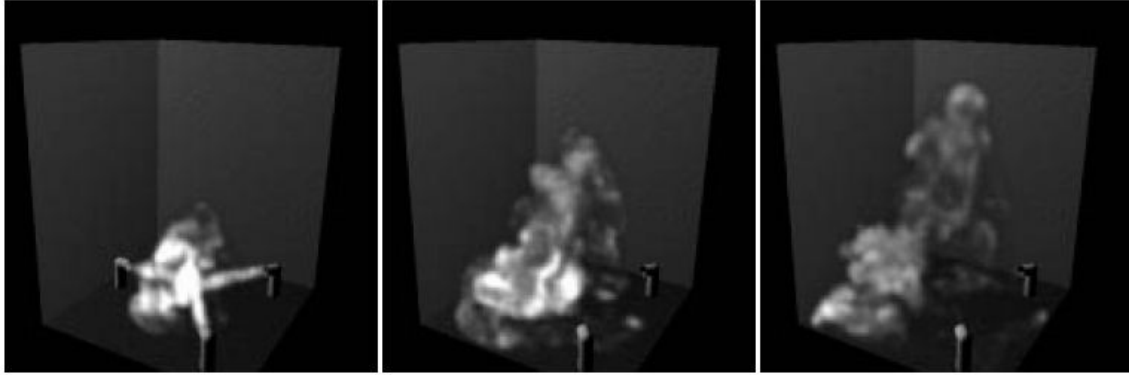


Figura 6.3 -

### 6.3. Fluidos Estáveis

Jos Stam propõe como solução para o custo computacional do método de Foster e Metaxas a utilização de métodos Lagrangeanos e esquemas implícitos para solução numérica das equações de Navier-Stokes [Stam (1999)].

O modelo matemático, para as equações de fluidos, considerado por Jos Stam é composto pelas equações (6.1) e (6.2) acima. Contudo, usa-se a Decomposição de Helmholtz-Hodge, apresentada na seção 2.8, segundo a qual, um campo vetorial  $\vec{w}$  diferenciável, pode ser decomposto, na forma:

$$\vec{w} = \vec{u} + \nabla q, \quad (6.24)$$

onde  $\vec{u}$  possui divergente nulo ( $\nabla \cdot \vec{u} = 0$ ) e  $q$  é um campo escalar. Aplicando o operador divergente em ambos os membros da equação anterior, obteremos:

$$\nabla \cdot \vec{w} = \nabla^2 q, \quad (6.25)$$

onde usamos o fato de  $\vec{u}$  possuir divergente nulo. Obtemos assim uma equação de Poisson para o campo escalar  $q$ . A solução desta equação permite definir um operador de projeção dado por:

$$\vec{u} = P\vec{w} = \vec{w} - \nabla q, \quad (6.26)$$

Seja então a equação de Navier-Stokes para o campo  $\vec{w}$ :

$$\frac{\partial \vec{w}}{\partial t} = \nu \nabla \cdot (\nabla \vec{w}) - (\vec{w} \cdot \nabla) \vec{w} - \frac{1}{\rho} \nabla p + \vec{f}_T, \quad (6.27)$$

Aplicando este operador a ambos os membros desta equação, supondo densidade  $\rho = 1$ , obtemos:

$$\frac{\partial \vec{u}}{\partial t} = P \left( \nu \nabla \cdot (\nabla \vec{u}) - (\vec{u} \cdot \nabla) \vec{u} + \vec{f}_T \right), \quad (6.28)$$

onde usamos as expressões:

$$P\left(\frac{\partial \bar{w}}{\partial t}\right) = \frac{\partial}{\partial t}(P\bar{w}) = \frac{\partial \bar{u}}{\partial t}, \quad (6.29),$$

$$P(\nabla p) = 0. \quad (6.30).$$

Pela equação (6.28) vemos que a contribuição da pressão é nula para o cálculo da componente  $\bar{u}$  da expressão (6.24), a qual satisfaz a condição de incompressibilidade:

$$\nabla \cdot \bar{u} = 0. \quad (6.31)$$

Desta forma, a solução da equação (6.28) é aquela desejada. O método apresentado para a integração desta equação é a principal contribuição do trabalho de Stam. Basicamente, resolve-se a equação (6.27), sem considerar a contribuição do termo envolvendo a pressão, para, em seguida, obter a componente  $\bar{u}$  via as equações (6.25)-(6.26). Vejamos alguns detalhes do mesmo.

### 6.3.1 Esquema Numérico

No esquema numérico proposto, seja  $\bar{w}(x,t)$  a solução da equação (6.27) para um instante  $t$  e seja  $\Delta t$  o passo de tempo considerado. Para obter  $\bar{w}(x,t + \Delta t)$ , resolve-se numericamente a equação (6.27), para o intervalo  $\Delta t$ , em quatro passos. Tomando-se  $\bar{w}_0(x)$  como condição inicial desta equação, resolve-se primeiramente a parcela correspondente à força externa:

$$\bar{w}_1 = \bar{w}_0 + \Delta t f_T, \quad (6.32)$$

Em seguida trata-se do termo de advecção (ou convecção) definido pela expressão  $(\bar{u} \cdot \nabla)\bar{u}$ , fonte principal de instabilidade numérica, segundo a discussão da seção (6.2.3). Em [Stam (1999)] este termo é tratado numericamente via o Método das Características, bem conhecido no contexto de equações diferenciais parciais. Assim, obtém-se um esquema numérico mais estável, mesmo para passos de tempo que não satisfazem a restrição dada pela expressão (6.21). Isto diminui o custo computacional dos algoritmos em questão, sem trazer prejuízos para a geração de efeitos visuais, sendo, portanto, uma contribuição importante do ponto de vista da computação gráfica [Stam (1999)].

O Método das Características consiste na seguinte formulação. Seja a equação de convecção:

$$\frac{\partial a(x,t)}{\partial t} = -\bar{v}(x) \cdot \nabla a(x,t), \quad (6.33),$$

$$a(x,0) = a_0(x),$$

onde  $a(x,t)$  é um campo escalar e  $\bar{v}(x)$  é um campo vetorial estacionário.

Seja  $c$  a solução do seguinte problema de valor inicial:

$$\begin{cases} \frac{dc(x_0, t)}{dt} = \vec{v}(x); \\ c(x_0, 0) = x_0. \end{cases} \quad (6.34)$$

A curva  $c(x_0, t)$ , solução do problema acima, é denominada a característica do campo vetorial  $\vec{v}(x)$ , passando pelo ponto  $x_0$  em  $t = 0$ . Seja agora  $\bar{a}(x_0, t) = a(c(x_0, t), t)$  o valor do campo escalar ao longo da característica considerada. Aplicando a regra da cadeia, verifica-se diretamente que:

$$\frac{d\bar{a}}{dt} = \frac{\partial \bar{a}(x, t)}{\partial t} + \vec{v}(x) \cdot \nabla \bar{a}(x, t) = 0. \quad (6.35)$$

Portanto, o valor do campo escalar  $a(x, t)$  não varia ao longo de uma linha de corrente. Em particular,  $\bar{a}(x_0, t) = \bar{a}(x_0, 0) = a_0(x_0)$ . Este é o princípio básico do Método das Características.

Este método é usado em [Stam (1999)] para resolver a equação de advecção. Consideremos um intervalo de tempo genérico  $[t_0, t_0 + \Delta t]$  e  $\vec{w}_1(x, t) = (w_1^1, w_1^2, w_1^3)$ , onde  $w_1^i(x)$  são as componentes do vetor velocidade. Substituindo  $\vec{v} = \vec{w}_1(x, t_0)$  e  $a_0(x) \equiv w_1^1(x, t_0)$  no problema (6.33) obtemos:

$$\begin{aligned} \frac{\partial w_1^1(x, \tau)}{\partial \tau} &= -\vec{w}_1(x) \cdot \nabla w_1^1(x, \tau), \quad (6.36) \\ w_1^1(x, 0) &= a_0(x), \end{aligned}$$

onde  $\tau = t - t_0$  é uma variável auxiliar.

Seja agora  $c(x_0, \tau)$  a característica correspondente, solução do problema (6.34). Então, fazendo  $\tau = \Delta t$  teremos:

$$w_1^1(c(x_0, \Delta t), \Delta t) = w_1^1(c(x_0, 0), 0) = w_1^1(x_0, 0). \quad (6.36)$$

O próximo passo é resolver o termo correspondente à viscosidade, o que é equivalente a resolver a seguinte equação de difusão:

$$\frac{\partial w_2(x, t)}{\partial t} = \nu \nabla^2 w_2(x, t) = 0, \quad (6.37)$$

O método mais direto para resolver numericamente esta equação é através de esquemas explícitos de diferenças finitas. Contudo, estes métodos são instáveis para valores altos de  $\nu$ . Assim, em [Stam (1999)] optou-se por um método numérico implícito dado por:

$$(I - \nu \Delta t \nabla^2) \vec{w}_3(x) = \vec{w}_2(x), \quad (6.38)$$

onde  $I$  é o operador identidade.

Este esquema conduz a matrizes esparsas podendo ser resolvido computacionalmente de maneira eficiente.

Finalmente, o último passo, vai fornecer a projeção. O resultado é o campo  $\bar{w}_4(x)$  obtido pela expressão:

$$\bar{w}_4 = \bar{w}_3 - \nabla q, \quad (6.39)$$

onde  $q$  é solução da equação:

$$\nabla \cdot \bar{w}_3 = \nabla^2 q, \quad (6.40)$$

### 6.3.2 Condições de Contorno Periódicas

Quando usamos condições de contorno periódicas, o método apresentado acima assume uma forma muito elegante, baseada na transformada (rápida) de Fourier (FFT). Seja então a FFT, com relação às variáveis espaciais, da velocidade  $\bar{v}(x,t)$ , denotada por  $\hat{v}(\vec{k},t)$ , onde  $\vec{k}$  é o vetor das variáveis livres no domínio de Fourier, tendo a mesma dimensão de  $x$ , e denominado número de onda. Lembrando a propriedade:

$$F\left(\frac{\partial^n f}{\partial x^n}\right) = (i)^n k^n F(f), \quad (6.41)$$

onde  $i^2 = -1$  e  $F$  é a transformada de Fourier de  $f$ .

Fica claro que, no domínio de Fourier, os operadores de difusão e projeção tomam a forma [Stam (1999)]:

$$I - \nu \Delta t \nabla^2 \rightarrow I + \nu \Delta t k^2, \quad (6.41)$$

$$Pw \rightarrow \hat{P}\hat{w}(\vec{k}) = \hat{w}(\vec{k}) - \frac{1}{k^2}(\vec{k} \cdot \hat{w}(\vec{k}))\vec{k}, \quad (6.42)$$

onde  $k = \|\vec{k}\|$ .

Observando a expressão (6.42) vemos que, no domínio de Fourier, o operador de  $P$  projeta um vetor o vetor  $\hat{w}(\vec{k})$  sobre o plano normal ao número de onda  $\vec{k}$ . Portanto, a transformada de Fourier de um campo com divergente nulo é sempre ortogonal ao número de onda  $\vec{k}$ . Por outro lado, a expressão (6.41) mostra que a difusão pode ser interpretada como um filtro “passa-baixa” cuja largura é proporcional ao intervalo de tempo  $\Delta t$  e à viscosidade  $\nu$ . Desta forma, a solução no domínio de Fourier pode ser obtida como segue:

$$\text{Força Externa: } \bar{w}_1 = \bar{w}_0 + \Delta t f_T, \quad (6.43)$$

$$\text{Advecção: } \bar{w}_2(x) = \bar{w}_1(c(x, -\Delta t)), \quad (6.44)$$

$$\text{Transformada: } \hat{w}_2(\vec{k}) = F(\bar{w}_2(x)), \quad (6.45)$$

$$\text{Difusão} : \hat{w}_3(x) = \frac{\hat{w}_2}{1 + \nu \Delta t k^2}, \quad (6.46)$$

$$\text{Projeção} : \hat{P} \hat{w}_3(\vec{k}) = \hat{w}_4(\vec{k}), \quad (6.47)$$

$$\text{Inversa} : F^{-1}(\hat{w}_4(\vec{k})) = \bar{w}_4(x), \quad (6.48)$$

Este método é de fácil implementação, mas tem a desvantagem de ter custo computacional  $O(N \log N)$  onde  $N$  é o número de nós da malha utilizada.

### 6.3.3 Advecção e Difusão de Substâncias

Um recurso interessante para criação de efeitos visuais a partir do campo de velocidades obtido é simular a dispersão de substâncias não-reativas no interior do fluido. Isto significa que a substância sofrerá advecção e difusão. A equação geral neste caso segue a lei de conservação para campos escalares apresentada na seção 2.2.1, equação (2.8), e tem a forma:

$$\frac{\partial S}{\partial t} = -\vec{v} \cdot \vec{\nabla} S + \nu_s \Delta S - \alpha_s S + Q_s, \quad (6.49)$$

onde  $S$  é a concentração da substância,  $\vec{v}$  a velocidade do fluido,  $Q_s$  representa o termo fonte e  $\nu_s, \alpha_s$  denotam os coeficientes de difusão e dissipação, respectivamente.

Comparando esta equação com equações de Navier-Stokes dadas pela expressão (6.27), observamos que, do ponto de vista matemático, o termo de dissipação é o único termo novo. Este termo pode ser resolvido numericamente pela seguinte expressão:

$$(1 + \Delta t \alpha_s) S(x, t + \Delta t) = S(x, t), \quad (6.50)$$

a qual representa a solução numérica (implícita), por diferenças finitas, da equação:

$$\frac{\partial S}{\partial t} = -\alpha_s S. \quad (6.51)$$

Os demais termos podem ser resolvidos numericamente usando a mesma metodologia apresentada na seção (6.3.1).

### 6.3.4 Renderização e Resultados

O método descrito acima foi implementado e adicionado a um sistema gráfico que permite a interação do usuário com o fluido em tempo-real. Em [Stam (1999)] são fornecidos detalhes deste sistema. A visualização do movimento do fluido é feita pela adição de substâncias ou texturas, as quais sofrerão advecção, difusão e dissipação, de acordo com a equação (6.49). Em seguida, usa-se *volume rendering* para a fase de *rendering*. Efeitos de sombreado podem ser obtidos adicionando-se uma fonte de luz direcional em algum ponto da cena.

A Figura 6.4(a) mostra uma seqüência 2D obtida com a utilização de texturas. As Figuras 6.4 (b)-(g) mostram *frames* obtidos via fluidos 3D. Nestes casos, utilizaram-se substâncias não-reativas, *volume rendering* e efeitos de sombreamento. Foram utilizadas malhas com resoluções entre  $16^3$  e  $30^3$ , na plataforma gráfica SGI Octane com processador R10K e 192 Mbytes de RAM.

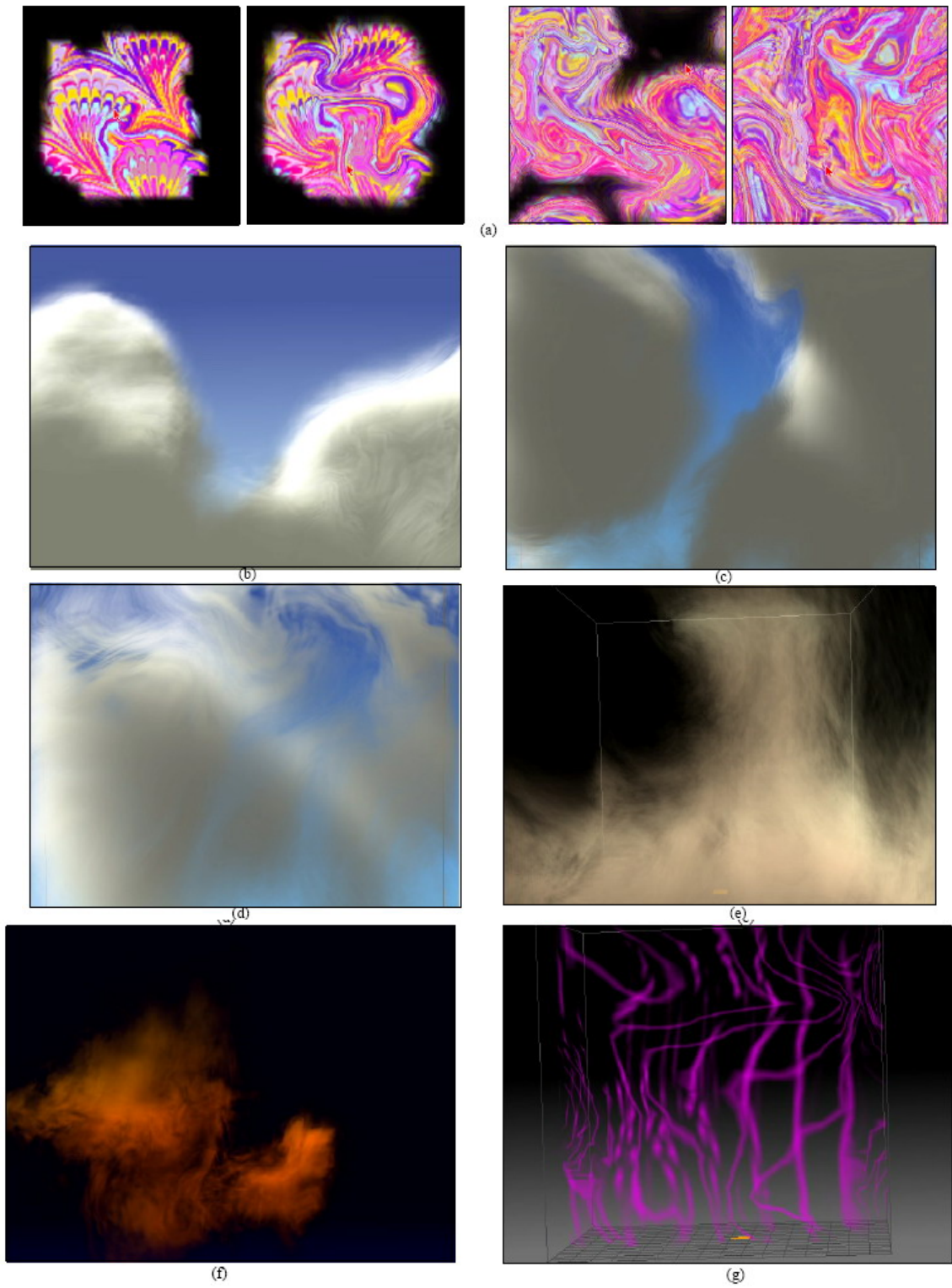


Figura 6.4 -



#### 6.4. Fluidos Compressíveis

Nos dois modelos descritos acima, os fluidos são considerados incompressíveis. Em [Witting (1999)], Patrick Witting considera um modelo matemático para fluidos (gases) bem mais completo que os anteriores. Este método foi usado para gerar alguns efeitos visuais do filme *The Prince of Egypt*.

Primeiramente, em [Witting (1999)] os gases são modelados como compressíveis e a densidade do fluido não é suposta constante. Assim como em [Stam (1999)], descrito acima, advecção e / ou difusão de campos escalares artificiais pode ser utilizada para criar efeitos visuais. O modelo para fluidos utilizado é uma versão simplificada daquele encontrado em [Klemp (1978)], usado para simulação de fenômenos atmosféricos.

Neste modelo, em lugar da pressão ( $p$ ) e temperatura usuais ( $T$ ), são utilizadas a pressão adimensional e a temperatura potencial, definidas, respectivamente, por:

$$\Pi = \left( \frac{p}{p_0} \right)^{\frac{R}{c_p}}, \quad (6.52)$$

$$\theta = T \left( \frac{p}{p_0} \right)^{-\frac{R}{c_p}}, \quad (6.53)$$

Em meteorologia usa-se temperatura potencial, dada pela expressão (6.53), em lugar da temperatura usual, como uma medida da estabilidade estática do fluido, uma vez que as grandezas usuais não são conservadas na atmosfera [Klemp (1978)].

As grandezas  $p$  e  $T$  estão relacionadas pela equação do gás ideal:

$$p = \rho RT, \quad (6.54)$$

Finalmente, supondo  $\vec{u} = (u_1, u_2, u_3)$  a velocidade do fluido, as equações do mesmo são dadas por:

$$\frac{\partial u_1}{\partial t} = -u_1 \frac{\partial u_1}{\partial x} - u_2 \frac{\partial u_1}{\partial x} - u_3 \frac{\partial u_1}{\partial x} - c_p \bar{\theta} \frac{\partial \pi}{\partial x} + \nu \nabla^2 u_1 + f_1, \quad (6.55a)$$

$$\frac{\partial u_2}{\partial t} = -u_1 \frac{\partial u_2}{\partial x} - u_2 \frac{\partial u_2}{\partial x} - u_3 \frac{\partial u_2}{\partial x} - c_p \bar{\theta} \frac{\partial \pi}{\partial y} + \nu \nabla^2 u_2 + f_2, \quad (6.55b)$$

$$\frac{\partial u_3}{\partial t} = -u_1 \frac{\partial u_3}{\partial x} - u_2 \frac{\partial u_3}{\partial x} - u_3 \frac{\partial u_3}{\partial x} - c_p \bar{\theta} \frac{\partial \pi}{\partial z} + \nu \nabla^2 u_3 + g \left( \frac{\theta - \bar{\theta}}{\bar{\theta}} \right) + f_3, \quad (6.55c)$$

$$\frac{\partial \pi}{\partial t} = \frac{\bar{c}^2}{c_p \bar{\rho} \bar{\theta}^2} \left( \frac{\partial}{\partial x} (\bar{\rho} \bar{\theta} u_1) + \frac{\partial}{\partial y} (\bar{\rho} \bar{\theta} u_2) + \frac{\partial}{\partial z} (\bar{\rho} \bar{\theta} u_3) \right), \quad (6.55d)$$

$$\frac{\partial \theta}{\partial t} = \nu_0 \nabla^2 \theta + f_\theta. \quad (6.55e)$$

onde a quantidade  $\pi$  é uma perturbação da pressão adimensional, ou seja:

$$\Pi = \bar{\Pi} + \pi, \quad (6.56)$$

onde:

$$\frac{\partial \bar{\Pi}}{\partial z} = \frac{-g}{c_p \bar{\theta}}. \quad (6.57)$$

Nestas equações,  $g$  é a aceleração da gravidade,  $f_1, f_2, f_3, f_\theta$  são componentes de forças,  $\nu$  e  $\nu_0$  são coeficientes de difusão,  $c_p$  é o calor específico a pressão constante  $\bar{c}$  é a velocidade do som e  $\bar{\theta}$  uma temperatura de referência.

Adicionalmente, a estas equações podem ser utilizadas equações descrevendo a advecção e / ou difusão de um campo escalar  $S$ , sob a ação do fluxo, de acordo com o modelo dado pela expressão (6.50). Em particular, pode-se utilizar um campo de texturas inicial, o qual será difundido pelo fluido, o que será explorado na seção 6.4.3.

#### 6.4.1 Fenômenos Incorporados

As equações (6.55a)-(6.55e) permitem a simulação de efeitos que não podem ser obtidos pelos modelos anteriores. Esta generalidade do modelo decorre das seguintes características:

Compressibilidade:

Este é o aspecto mais importante deste modelo, se comparado com os aqueles descritos nas seções anteriores. Além da possibilidade de simular efeitos que dependem da natureza compressível do fluido, ainda há vantagens computacionais. Segundo Witting [Witting (1999)], as equações correspondentes podem ser resolvidas mais rapidamente, uma vez que não é preciso resolver a equação de Poisson (6.13) neste caso. Assim, não existe o custo computacional extra decorrente do esquema iterativo dado pela expressão (6.17).

Conservação da Massa:

A equação (6.55d), que expressa a conservação da massa foi adicionada explicitamente. Isto é necessário, uma vez que a condição de incompressibilidade, dada pela expressão ( $\vec{\nabla} \cdot \vec{v} = 0$ ) não é satisfeita e a densidade pode ser variável no tempo e no espaço.

Densidade Variável:

Uma vez que o modelo em questão permite densidade variável, aumentamos o espectro de efeitos que podem ser gerados.

#### 6.4.2 Solução Numérica

Em [Witting (1999)] usa-se método de Runge-Kutta de Quarta ordem, dado pela equação (4.120)-(4.121), para integração no tempo e diferenças finitas de quarta ordem para discretização das derivadas espaciais. A malha utilizada é regular, com células cúbicas. Assim o sistema de equações (6.55a)-(6.55e) é escrito na forma:

$$\frac{d\Psi}{dt} = F(\Psi), \quad (6.57)$$

$$\Psi(t_0) = \Psi_0,$$

onde  $\Psi = (u_1, u_2, u_3, \pi, \theta)^T$  e  $d/dt$  representa a derivada material, dada por:

$$\frac{d\Psi}{dt} = \frac{\partial\Psi}{\partial t} + u_1 \frac{\partial\Psi}{\partial x} + u_2 \frac{\partial\Psi}{\partial y} + u_3 \frac{\partial\Psi}{\partial z}, \quad (6.58)$$

e  $\Psi_0$  é uma condição inicial conveniente, conhecida nos nós da malha regular.

A partir de então, a evolução temporal dos campos (componentes de  $\Psi$ ) será obtida via o esquema numérico descrito acima, aplicado ao sistema (6.57).

Embora seja possível utilizar esquemas numéricos com precisão menor, em [Witting (1999)] optou-se pela precisão de quarta ordem, pois o custo computacional não se mostrou restritivo e, além disso, com esta precisão pode-se utilizar malhas com menor resolução.

A condição de estabilidade neste caso é dada por:

$$\Delta t < \frac{2h\sqrt{2}}{c_* \sqrt{mn}}, \quad (6.58)$$

onde  $h$  é a resolução da malha,  $n$  a dimensão do espaço,  $m$  e  $c_*$  são fatores relacionados com propriedades numéricas [Witting (1999)].

#### 6.4.3 Resultados

A Figura 6.5 mostra um resultado obtido com esta técnica. Neste exemplo, a luminância da imagem fornecida pelo usuário (topo) é interpretada como sendo a distribuição inicial de temperatura do sistema. O fluido no interior das letras é mais frio e denso que o fluido no exterior. Com o avanço do tempo, os fluidos se

misturam. A visualização deste processo, obtida pela construção de um mapa de cores para cada instante de simulação, produz o efeito visual observado.



*Figura 6.5. Distribuição de Temperatura após 0, 100 e 400 instantes de simulação*

A Figura 6.6 mostra outro exemplo, neste caso foi explorando a evolução do mapa inicial de texturas representado a paisagem da Figura 6.6(a). Este mapa é dado como condição inicial para uma equação do tipo (6.49), com termo fonte e coeficiente de dissipação nulos. Resolveu-se assim, um problema do tipo:

$$\frac{\partial S}{\partial t} = -\vec{v} \cdot \vec{\nabla} S + \nu_S \Delta S, \quad (6.59)$$

$$S(0) = S_0$$

onde  $S_0 : \mathfrak{R}^2 \rightarrow \mathfrak{R}$  é a imagem da Figura 6.6 e  $\vec{v} = (u_1, u_2, u_3)$  é o campo de velocidades obtido pela solução das equações (6.55a)-(6.55d).

O resultado, para  $t = 400$  instantes de simulação, é a apresentado na Figura 6-6(b).



(a)



(b)

Figura 6.6 -

### 6.5. Animação via SPH

Em [Müller et al. (2003)] é proposto um método para animação de fluidos baseado na técnica de SPH. Seguindo a metodologia apresentada na seção 4.6, serão utilizadas expressões do tipo (4.138) e (4.140) para calcular as forças de pressão e viscosidade. Para permitir interatividade (tempo-real) são utilizados núcleos de suavização específicos, que procuram manter o realismo necessário sem comprometer a custo computacional.

O comportamento do fluido é descrito pela equação de continuidade desenvolvida na seção 2.2.3, com  $Q_v = Q_s = 0$ , e por uma versão simplificada da equação de Navier-Stokes, dadas, respectivamente, por:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) = 0, \quad (6.60)$$

$$\rho \frac{d\vec{v}}{dt} = \rho \left( \frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = -\nabla p + \rho \vec{g} + \Delta \vec{v}, \quad (6.61)$$

onde  $\vec{g}$  é um campo de forças externo.

A utilização de partículas em lugar de uma malha estacionária para resolver numericamente estas equações simplifica o problema. Primeiramente, mantendo o número de partículas constante durante a simulação e supondo todas as partículas com a mesma massa, temos como consequência a conservação da massa e portanto a equação (6.60) pode ser omitida. Além disto, vamos substituir a expressão  $\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v}$  pela derivada material  $\frac{d\vec{v}}{dt}$ , uma vez que não temos uma malha para estimar derivadas parciais. A densidade na posição de cada partícula é calculada via equação (4.126), com  $f(x_j) = \rho_j$ :

$$\langle \rho(x_i) \rangle = \sum_{j=1}^N m_j W(x - x_j, h), \quad (6.62)$$

Em seguida, a pressão pode ser computada por uma equação do tipo gás ideal dada por:

$$p = k(\rho - \rho_0), \quad (6.63)$$

onde  $k$ , a exemplo da equação (6.54) depende da temperatura e  $\rho_0$  é a densidade de inicial.

Passemos agora às versões discretas das forças de pressão ( $f^{\text{Pressão}}$ ) e viscosidade ( $f^{\text{Visc}}$ ), obtidas via a metodologia apresentada na seção 4.6. Usando as equações (4.138) e (4.140) para estimar o gradiente da pressão  $p$  e o Laplaciano da velocidade, respectivamente, obteremos as seguintes versões discretas:

$$f_i^{\text{Pressão}} = -\langle \nabla_x p(x_i, t) \rangle = -\sum_{j=1}^N m_j \left( \frac{p(x_j) + p(x_i)}{\rho(r_j)} \right) \nabla_x W(x_i - x_j, h). \quad (6.64)$$

$$f_i^{\text{Visc}} = \mu \langle \nabla_x^2 \vec{v}(x_i, t) \rangle = \mu \sum_{j=1}^N m_j \left[ \frac{\vec{v}(x_j) - \vec{v}(x_i)}{\rho(r_j)} \right] \nabla_x^2 W(x_i - x_j, h), \quad (6.65)$$

Além destas forças e da força externa, que aparecem explicitamente na equação (6.61), são considerada também em [Müller et al. (2003)] forças de superfície que atuam no fluido em superfícies livres. A superfície livre do fluido, ou seja, a fronteira do mesmo que não está sujeita às condições de contorno, pode ser obtida usando-se uma quantidade adicional que assume valor unitário nas posições das partículas. Este campo, denominado campo de cor na literatura, tem sua versão suavizada dada pela expressão:

$$\langle c_s(x) \rangle = \sum_{j=1}^N m_j \frac{1}{\rho(r_j)} W(x - x_j, h), \quad (6.66)$$

Portanto, a normal e curvaturas da superfície livre podem ser obtidas por:

$$\vec{n} = \vec{\nabla} \langle c_s(x) \rangle, \quad \kappa = -\frac{\Delta \langle c_s(x) \rangle}{\|\vec{n}\|}. \quad (6.67)$$

Finalmente, as forças na superfície livre são modeladas por:

$$t^{slivre} = \sigma \kappa \frac{\vec{n}}{\|\vec{n}\|}, \quad (6.68)$$

onde  $\sigma$  é um parâmetro que controla a influência desta força.

O objetivo final é reproduzir o efeito de minimização de curvatura, observado para este tipo de força. De fato, se considerarmos uma superfície inicial (frente)  $S$  que se movimenta de acordo com uma equação do tipo:

$$\frac{\partial S}{\partial t} = \sigma \kappa \frac{\vec{n}}{\|\vec{n}\|}, \quad (6.69)$$

Observamos exatamente uma evolução que suaviza pontos onde a curvatura é mais elevada [Sethian (1996)].

As condições de contorno são implementadas seguindo uma metodologia simples inspirada na interpretação do fluido como um sistema de partículas. Ou seja, quando uma partícula colide com a parede de um objeto sólido, o movimento da partícula (sua velocidade), na direção normal à superfície rígida, é refletida para o interior do fluido.

Os núcleos de suavização são dados pelas seguintes expressões:

$$W_{cor}(x, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2), & 0 \leq r \leq h, \\ 0, & cc. \end{cases} \quad (6.70)$$

$$W_{press}(x, h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3, & 0 \leq r \leq h, \\ 0, & cc. \end{cases} \quad (6.71)$$

$$W_{visc}(x, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1, & 0 \leq r \leq h, \\ 0, & cc. \end{cases} \quad (6.72)$$

Esta última foi escolhida por possuir Laplaciano positivo em todo domínio, dado por:

$$\Delta W_{visc}(r, h) = \begin{cases} \frac{45}{\pi h^6} (h - r), & 0 \leq r \leq h \\ 0, & cc. \end{cases} \quad (6.73)$$

Além disto, observamos também que  $\nabla W_{visc}(r = h, h) = 0$ . Segundo [Müller at al. (2003)], estas propriedades são importantes do ponto de vista numérico, pois asseguram que o algoritmo terá uma maior estabilidade.

### 6.5.1 Simulação

As equações de Navier-Stokes dadas pela expressão (6.61) são discretizadas, substituindo-se cada termo do segundo membro pela sua versão obtida via núcleo de suavização, ou seja, pelas expressões (6.64)-(6.65), com a densidade  $\rho$  dada pela equação (6.62), e, usando-se no primeiro membro a derivada material  $\frac{d\vec{v}}{dt}$ . Desta forma, obtém-se o seguinte esquema numérico:

$$\frac{d\vec{v}_i}{dt} = Q_i^t, \quad (6.74)$$

onde

$$Q_i^t = \frac{-\sum_j m_j \frac{p_j^t + p_j^i}{2\rho_j^t} \nabla W_1(r_i^t - r_j^t, h) + \rho_j^t \vec{g}_j + \sum_j m_j \left( \frac{\vec{v}_j^t - \vec{v}_i^t}{\rho_j^t} \right) \nabla^2 W_2(r_i^t - r_j^t, h)}{\rho_j^t} \quad (6.75)$$

Seguindo as equações (6.62)-(6.63), temos:

$$\rho_j^t = \sum_k m_k W_2(r_j^t - r_k^t) \quad (6.76)$$

$$P_j^t = k(\rho_j^t - \rho_j^0). \quad (6.77)$$

Uma vez resolvida a equação (6.74), o campo de acelerações  $\left( \frac{d\vec{v}_i}{dt} \right)$  obtido é finalmente utilizado para atualizar a velocidade e posição das partículas, de acordo com o seguinte esquema:

$$\vec{v}_i^{t+\Delta t} = \vec{v}_i^t + \frac{\delta t}{2} Q_i^t, \quad (6.78)$$

$$r_i^{t+\Delta t} = r_i^t + \delta t \vec{v}_i^{t+\Delta t}. \quad (6.79)$$

Este esquema, denominado Leap-Frog, é um método iterativo de segunda ordem. Em [Müller at al. (2003)], usou-se um intervalo de tempo constante de  $\delta t = 10^{-2}$  segundos.

Estruturas de dados adicionais podem ser utilizadas para reduzir a complexidade dos algoritmos. Primeiramente, uma vez que os núcleos de suavização possuem suporte limitado, dados por  $h$ , pode-se usar uma malha regular, com células (cubos) de lado  $h$ , e distribuir o conjunto de partículas de acordo com esta malha. Assim, dada uma partícula  $i$ , as candidatas a participarem com a mesma durante o cálculo das quantidades em questão ou estarão na mesma célula



da partícula  $i$ , ou estarão nas células vizinhas desta célula. Assim, o custo computacional em um laço do esquema acima fica reduzido de  $O(n^2)$  para  $O(nm)$ , onde  $n$  é o número de partículas e  $m$  é a quantidade média de partículas, por célula, da malha auxiliar (observe que  $n \gg m$  de maneira geral). Neste caso temos duas opções: guardar apenas referências às partículas nas células da malha ou duplicar a informação guardando cópias das partículas. Segundo [Müller et al. (2003)], esta última opção tem permitido aumentar a eficiência do algoritmo de um fator de 10, em função da proximidade em memória da informação necessária para a interpolação.

### 6.5.2 Visualização e Resultados

Em [Müller et al. (2003)] são utilizadas técnicas baseadas em superfícies e técnicas baseadas em nuvens de pontos para efetuar a visualização. Em ambos os casos, a idéia é fazer uso do campo de cor, dado pela expressão (6.66), e seu gradiente, calculado por uma expressão do tipo (4.138), para identificar partículas da superfície livre do fluido e encontrar sua normal. Primeiramente, consideraremos uma partícula sobre a superfície se:

$$\|\vec{n}\| > l, \quad (6.80)$$

onde  $l$  é limiar a ser escolhido previamente.

Assim, a normal a uma partícula de superfície na posição  $x_i$  é dada por:

$$-\vec{n}(x_i). \quad (6.81)$$

Desta forma, obtemos um conjunto de pontos, com suas normais, mas sem informações de conectividade. Estes são os elementos necessários para efetuar o *rendering* via *point splatting*, descrito na seção 4.3.2. Contudo, devemos ficar atentos com relação à quantidade de pontos de superfície. Usualmente, as técnicas de *splatting* são aplicadas para nuvens de pontos obtidas via scanners, com tipicamente 10.000 a 100.000 pontos. Em [Müller et al. (2003)] usa-se apenas algumas centenas de partículas, para evitar um custo computacional que impossibilite a interatividade em tempo-real. Considerando-se ainda que a quantidade de pontos de superfícies é apenas uma fração deste montante, fica a dúvida com relação à qualidade visual da cena.

Na Figura 6.7(a) temos um exemplo da utilização de *splatting* nesta aplicação. A água no copo foi amostrada com 2200 partículas. Um campo de forças rotacional externo faz o fluido se movimentar. A primeira imagem mostra as partículas individualmente. A Figura 6.7(b) mostra um resultado obtido pela utilização de *splatting* para a visualização da superfície livre da água. Embora a qualidade deste resultado possa ser considerada satisfatória, observamos um resultado bem mais realista quando utiliza-se *Marching-Cubes* (Figura 6.7(c)), descrito na seção 4.2.1. Neste caso, a superfície alvo seria uma isosuperfície do campo de cor suavizado, dado pela expressão (6.66), definida por  $\langle c_s(x) \rangle = 0$ . Escolheu-se uma malha regular, fixa, e usou-se *Marching-Cubes* para obter uma representação poligonal da

superfície em questão. A implementação usada em [Müller et al.] inicia a busca pelas células que contém partículas de superfície, varrendo recursivamente as células próximas a estas. Usa-se uma tabela *hash* para evitar que uma mesma célula seja visitada mais de uma vez. Contudo, o custo computacional torna-se mais elevado. Nestes exemplos, usou-se um Pentium IV com 1.8 GHz e placa gráfica GForce4. Para *splatting*, conseguiu-se uma taxa de 20 *frames* por segundo enquanto que usando *Marching-Cubes* a taxa diminuiu para 5 *frames* por segundo.

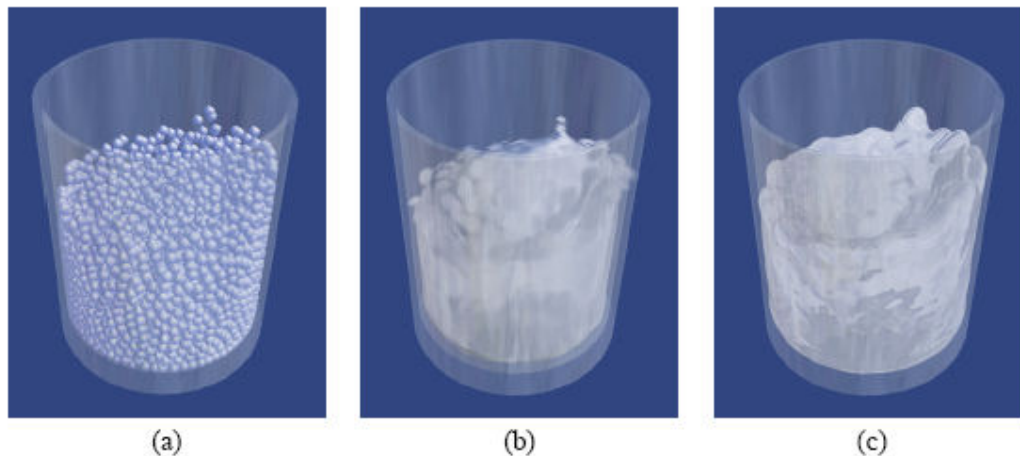
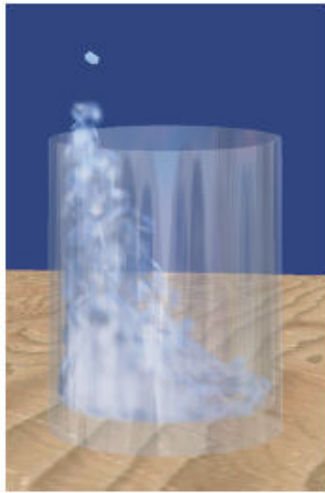
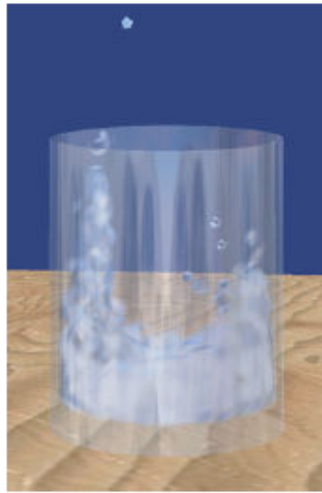


Figura 6.7 -

A Figura 6.8 mostra um outro exemplo, onde o usuário interage com o fluido, via mouse, gerando um campo externo que faz a água “espirrar”. Neste caso, a superfície livre foi visualizada via *splatting*. Usou-se 1300 partículas, conseguindo-se uma taxa de 25 *frames* por segundo com a arquitetura descrita acima.



(a)



(b)



(c)

*Figura 6.8 –*

## Capítulo 7      Perspectivas e Conclusões

Os trabalhos discutidos no Capítulo 6 fornecem um rápido panorama dos métodos utilizados em animação computacional de fluidos. Como já ressaltado, a proposta daquele capítulo era apenas apresentar uma revisão objetiva de alguns métodos, os quais vêm sendo aprimorados, assim como novas propostas vêm sendo desenvolvidas. Em particular, interação entre fluidos e sólidos deformáveis [Carlson (2002)][Müller (2004)][Teschner (2005)] ou não [Carlson (2004)], problemas inversos aplicados na geração de animações de fluidos a partir de imagens [Wijk(2002)] e novas estruturas de dados [Doug(2004)][Losasso (2004)] são alguns exemplos do estado da arte nesta área. Podemos notar um crescente interesse na comunidade científica, principalmente na área de Computação Gráfica, observando o número significativo de publicações nas últimas edições do *Siggraph*, onde sessões específicas são dedicadas a esse tema.

Neste cenário, podemos destacar algumas perspectivas que se mostram interessantes para novos trabalhos na área de animação de fluidos. Primeiramente, a independência de malhas que o método *SPH* proporciona, tem se apresentado como uma alternativa interessante para aplicações cujo foco principal não é a precisão do modelo de simulação. Este método permite evitar custos adicionais com a geração / adaptação da malha que discretiza o domínio do fluido quando a fronteira deste é modificada. Além disso, o método *SPH* pode ser também aplicado na solução das equações que descrevem o fluido, para modelar choques de objetos e explosões. Essa versatilidade coloca este método como uma alternativa interessante para a geração de cenas envolvendo fluidos e objetos em movimento.

Outro ponto que pode trazer ganhos efetivos, não só na simulação como também na visualização de fluidos é o uso de técnicas de programação em hardware gráfico. Essa é outra área que vem ganhando especial atenção não só da comunidade de computação gráfica, mas também de outras áreas científicas. O interesse vem da possibilidade de utilizar os novos processadores gráfico como ferramentas para programação de aplicações não gráficas [Krüger(2005)].

Com relação a potenciais aplicações para a animação computacional de fluidos, podemos citar simuladores [Fedkiw(2001)] [Premoze(2001)] [Rasmussen(2003)] [Wang(2005)], jogos eletrônicos [Stam(2003)b] e geração de efeitos especiais [Nguyen (2002)] [Geiger(2003)] [Feldman(2003)] [Treuille (2003)]. Nos dois primeiros casos, a necessidade de tempo-real é fundamental. No segundo caso, esta característica é desejável durante a fase de pré-produção, mas não deve ser considerada imprescindível.

Várias aplicações atualmente necessitam simular o comportamento de fluidos para aumentar o grau de realismo ao qual o usuário terá acesso. Aplicações em treinamento com imersão, como simuladores de navio ou submarinos, são bons exemplos. Neste tipo de aplicação um navegador em fase de instrução controla um barco virtual. Para que esse treinamento possa ser efetivo é interessante que o

simulador seja capaz não só de responder visualmente aos comandos do operador, mas também reproduzi-los com realismo físico tanto os movimentos do barco quanto o comportamento da água, segundo as escalas de tempo e espaço reais.

A forma mais eficiente e geral para adquirir realismo para fluidos em movimento é através dos modelos físicos discutidos no Capítulo 2. A redução do custo computacional depende fundamentalmente da utilização de métodos numéricos mais estáveis e eficientes, ou de soluções de hardware. Neste caso, aplicações envolvendo simuladores e geração de efeitos especiais para filmes podem gerar as imagens off-line ou mesmo utilizar supercomputadores ou clusters para atingir alta-performance. No caso dos jogos, tal solução é inviável, uma vez que neste caso os usuários seriam jogadores dispendendo de computadores pessoais, com bons recursos gráficos, mas sem alta performance para operações de cálculo intensivo.

Em aplicações na área de jogos a preocupação maior é o compromisso entre o realismo das imagens e a quantidade necessária de quadros por segundo (*fps*) que garanta uma visualização em tempo-real do fluido. Neste tipo de aplicação, no entanto, o compromisso com respeito às escalas de tempo do mundo real podem ser relaxado. Supondo, por exemplo, um jogo envolvendo uma batalha naval, o tempo de reação dos barcos / submarinos poderia ser menor que o real, com o objetivo de tornar o jogo mais dinâmico. Esta seria uma escolha do programador ou mesmo do jogador. Obviamente, este não é o caso para os simuladores.

Uma tecnologia que vem impulsionando o interesse de pesquisadores na área de animação de fluidos em computação gráfica é a possibilidade que as novas arquiteturas de hardware gráfico fornecem atualmente. Nos últimos anos o surgimento das *GPUs* – *Graphics Processing Units* – como um novo paradigma de arquitetura para aplicações gráficas ampliou de forma decisiva a pesquisa na área.

As mais recentes GPUs são vistas como processadores de fluxo (*stream processors*), operando em fluxos de dados [Comba (2003)]. Esses fluxos são formados por seqüências ordenadas de primitivas (e seus atributos). Essas primitivas são tipicamente vértices ou fragmentos.

De uma forma mais genérica, as *GPUs* podem ser vistas como computadores *SIMD* (*Single-Instruction Stream Multiple-Data Stream*), onde uma série de unidades de processamento executam simultaneamente o mesmo conjunto de instruções em um fluxo de primitivas. Em vários estágios do pipeline de renderização as *GPUs* possuem unidades de processamento programáveis e paralelas que podem atuar sobre os dados como uma arquitetura *SIMD*. Cada uma dessas unidades por sua vez é um processador vetorial, com capacidade de computar até 4 valores escalares em paralelo [Kilgariff(2005)].

Esse tipo de arquitetura se mostra bastante interessante para aplicações em simulação numérica [Krüger(2004)]. Em particular, aplicações de propósito geral que sejam intensivas em computação ou intensivas em acesso a memória podem obter ganhos efetivos no uso de GPUs, podendo superar em termos de tempo de

execução versões que rodam exclusivamente em CPU [Goodnight(2003)] [Bolz(2003)].

Atualmente um limitador para que esse tipo de estratégia seja mais utilizado por outras comunidades científicas fora da área de CG é a falta de uma ferramenta de desenvolvimento genérica para programação das aplicações. Ferramentas como com OGSF [Rost(2004)] e CG [Mark(2003)] são muito voltadas para aplicações gráficas, tornando necessário um conhecimento relativamente profundo do processo envolvido em um pipeline gráfico. Algumas ferramentas de alto nível, propondo o paradigma de programação “orientada” a fluxo (*stream programming*) estão surgindo, como *SH* [McCool(2002)], *Brook* [Buck(2004)], mas ainda encontram-se em versões preliminares.

Acreditamos que a combinação de métodos eficientes para modelar o fluido (como *SPH*, por exemplo) e o uso intensivo de programação em *GPUs* possa acelerar o processo de geração da animação do fluido, permitindo sua visualização em tempo real. Trabalhos nesse sentido já foram propostos [Botsch (2003)] [Kipfer(2004)] [Scheidegger (2004)].

A natureza multidisciplinar da animação de fluidos é outro aspecto a ser ressaltado. O desenvolvimento desta área depende da interação entre profissionais das áreas de computação e mecânica de fluidos. Este fato traz uma dimensão singular a esta área da animação computacional. Por um lado, os especialistas em animação procuram nas técnicas de modelagem de fluidos, ferramentas para atender suas necessidades em computação gráfica. A renderização final das cenas depende do desenvolvimento de recursos em visualização científica os quais, por sua vez, podem ser úteis para os engenheiros na análise dos resultados de suas simulações. Portanto podemos vislumbrar um campo de trabalho bastante fértil e com amplas possibilidades para pesquisa.

No LNCC, grupos de excelência vêm se associando com o intuito de aglutinar competência para tratar convenientemente problemas que envolvem conhecimentos em várias áreas das ciências e engenharias. A animação computacional de fluidos, na forma tratada nesta monografia, se insere neste contexto, podendo se constituir em uma linha de pesquisa dentro desta instituição, a qual vai agregar profissionais de duas áreas estratégicas para o LNCC: mecânica de fluidos e computação gráfica. Em particular, esta área poderá oferecer temas de tese para alunos do programa de pós-graduação do LNCC, uma vez que está em concordância com o perfil deste programa.

## Apêndice A

### A.1 Polinômios de Bernstein

Estes polinômios formam uma base muito usada na representação de curvas e superfícies, devido a suas propriedades das quais decorrem algoritmos simples e numericamente estáveis.

Os polinômios de Bernstein de grau  $n$  são os  $n+1$  polinômios da forma [Persiano (1996)]:

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n, \quad (A-1)$$

onde: 
$$\binom{n}{i} = \frac{n!}{i!(n-i)!},$$

Observemos que  $B_i^n(t) > 0$ ,  $i=0, \dots, n$ , se  $t \in [0,1]$  e que estes constituem uma partição da unidade pois:

$$\sum_{i=0}^n B_i^n(t) = (t + (1-t))^n = 1. \quad (A-2).$$

Assim, dados  $n+1$  pontos  $P_0, P_1, \dots, P_n$  de um espaço euclidiano  $\mathfrak{R}^m$ , a expressão:

$$c(t) = \sum_{i=0}^n P_i B_i^n(t), \quad (A-3)$$

é uma combinação convexa dos pontos  $P_0, P_1, \dots, P_n$ , fornece rigorosamente a representação paramétrica de uma curva no espaço  $\mathfrak{R}^m$ .

Os pontos  $P_i$  são denominados *pontos de controle* e o polígono formado por estes pontos é chamado *polígono de controle*.

Podemos observar diretamente de (A-3) que  $P_0$  e  $P_1$  são pontos sobre a curva. A expressão para a derivada de  $c$  mostrará que a curva é tangente ao polígono de controle nestes pontos. De fato, pode-se mostrar que [Persiano (1996)]:

$$\frac{dc}{dt} = n \sum_{i=0}^{n-1} (P_{i+1} - P_i) B_i^{n-1}(t), \quad (A-4)$$

portanto:

$$\frac{dc(0)}{dt} = n(P_1 - P_0) \text{ e } \frac{dc(1)}{dt} = n(P_n - P_{n-1}), \quad (A-5)$$

ficando assim demonstrado que a curva é tangente aos segmentos  $P_0P_1$  e  $P_nP_{n-1}$ .

Analogamente, mostra-se que[Persiano (1996)]:

$$\frac{d^2c}{dt^2} = n(n-1) \sum_{i=0}^{n-2} (P_{i+2} - 2P_{i+1} + P_i) B_i^{n-1}(t), \quad (A-6)$$

e portanto:

$$\frac{d^2c(0)}{dt^2} = n(n-1)(P_2 - 2P_1 + P_0), \quad (A-7)$$

$$\frac{d^2c(1)}{dt^2} = n(n-1)(P_2 - 2P_{n-1} + P_{n-2}). \quad (A-8)$$

## A.2 Splines

Uma *spline* polinomial é uma função polinomial por partes. *Splines* de grau  $m$  são formadas por polinômios de grau  $m$  e possuem derivadas contínuas até a ordem  $m-1$ .

Denotaremos por  $S^m(t_0, t_1, \dots, t_n)$  o espaço de todas as *splines* de grau  $m$  que apresentem derivadas contínuas até a ordem  $m-1$  em todos os seus  $n+1$  nós  $(t_0, t_1, \dots, t_n)$ ;  $t_0 < t_1 < \dots < t_n$ . Nestas condições, dada uma *spline*  $f(t) \in S^m(t_0, t_1, \dots, t_n)$  teremos que  $f$  restrita a um intervalo  $[t_i, t_{i+1}]$ ,  $0 \leq i \leq n-1$ , é uma função polinomial de grau  $m$ , que denotaremos por  $(f_i)$ , a qual pode ser escrita na base usual  $\{1, t, t^2, \dots, t^m\}$  na forma:

$$f_i(t) = a_{i0} + a_{i1}t + \dots + a_{im}t^m, \quad (A-9)$$



Para o caso de *splines* que se anulam fora do intervalo  $[t_0, t_n]$  podemos usar uma representação computacionalmente mais eficiente, obtida a partir da chamada base *B-spline* dada pela relação recursiva:

$$N_{i,0}(t) = \begin{cases} 1 & \text{se } t_i \leq t \leq t_{i+1}, \\ 0 & \text{cc.} \end{cases} \quad (A-10a)$$

$$N_{i,m}(t) = \frac{(t-t_i)N_{i,m-1}(t)}{t_{i+m+1}-t_i} + \frac{(t_{i+m}-t)N_{i+1,m-1}(t)}{t_{i+m}-t_{i+1}}, \quad (A-10b)$$

onde  $0 \leq i \leq n-m-1$ .

Dentre as propriedades das funções  $N_{i,m}$  destacamos as seguintes [Persiano (1996)]:

- Continuidade:  $N_{i,m}$  possui todas as derivadas contínuas até a ordem  $m-1$ .
- Suporte Limitado:  $N_{i,m}$  tem suporte dado pelo intervalo  $[t_i, t_{i+m+1})$ .
- Positividade: para todo  $t \in [t_0, t_n]$ ,  $1 \geq N_{i,m} \geq 0$ .
- Partição da Unidade:  $\sum_{i=0}^{n-m-1} N_{i,m}(t) = 1$ , se  $t \in [t_m, t_{n-m}]$ .

Pela propriedade (d) vemos que, dados  $n \times m$  pontos,  $P_0, P_1, \dots, P_{n-m-1}$  de um espaço euclidiano  $\mathfrak{R}^k$  então:

$$c(t) = \sum_{i=0}^{n-m-1} P_i N_{i,m}(t), \quad t \in [t_m, t_{n-m}] \quad (A-11)$$

define rigorosamente uma curva em  $\mathfrak{R}^k$  a qual denominamos *curva spline*.

Pela propriedade (b) vemos que a cada intervalo  $[t_{m+i-1}, t_{m+i}]$ ,  $1 \leq i \leq n-2m$ , corresponderá uma parte da curva, denominada *span*, a qual é dada por:

$$c_i(t) = \sum_{j=i-1}^{i+m-1} P_j N_{j,m}(t), \quad t \in [t_{m+i-1}, t_{m+i}]. \quad (A-12)$$

Esta expressão pode ser escrita na forma matricial a seguir:

$$c_i(t) = \left( N_{i-1,m}(t), N_{i,m}(t), \dots, N_{i+m-1,m}(t) \right) \begin{bmatrix} P_{i-1} \\ P_i \\ \dots \\ P_{i+m-1} \end{bmatrix}, \quad t \in [t_{m+i-1}, t_{m+i}] \quad (A-13)$$

Uma vez que em  $[t_{m+i-1}, t_{m+i}]$  cada função  $N_{i,m}$  é uma função polinomial de grau  $m$ , podemos usar (A-9) e escrever:

$$N_{j,m}(t) = a_{j0} + a_{j1}t + \dots + a_{jm}t^m, \quad i-1 \leq j \leq i+m-1. \quad (A-14)$$

Portanto, temos uma mudança de representação do *span*  $i$  dada por:

$$\begin{bmatrix} N_{i-1,m}(t) & N_{i,m}(t) & \dots & N_{i+m-1,m}(t) \end{bmatrix} = \begin{bmatrix} 1 & t & \dots & t^m \end{bmatrix} M_i, \quad (A-15)$$

onde  $M_i$  é uma matriz de ordem  $(m+1) \times (m+1)$ , dada pelos coeficientes de (A-14), a qual define uma mudança de base.

Logo, podemos escrever  $c_i$  na forma:

$$c_i(t) = (1, t, \dots, t^m)^T M_i Q_i, \quad (A-16),$$

onde  $Q_i = (P_{i-1}, P_i, \dots, P_{i+m-1})^T$  é o vetor de pontos de controle do *span*  $i$ .

Denotando por  $Q = (P_0, P_1, \dots, P_{n+m-1})^T$  o vetor de pontos de controle global podemos reescrever (A-16) na forma:

$$c_i(t) = (1, t, \dots, t^m)^T M_i G_i Q, \quad (A-17)$$

onde  $G_i$  é uma matriz  $(m+1) \times (n+m-1)$  que relaciona o vetor de pontos de controle (local)  $Q_i$  com o vetor de pontos de controle global  $Q$ .

Esta matriz tem a forma:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \dots 1 & 0 & 0 & \dots & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & \dots & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 1 & \dots & 0 & 0 & 0 \dots 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 0 & \dots & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 0 & \dots & 0 & 1 & 0 \dots 0 \end{bmatrix}. \quad (A-18)$$

### A.3 Superfícies Splines

Sejam dois espaços de *splines* polinomiais,  $S^m(u_0, u_1, \dots, u_n)$  e  $S^k(v_0, v_1, \dots, v_r)$ , como definido na seção anterior. Então, o produto tensor

$S = S^m(u_0, u_1, \dots, u_n) \otimes S^k(v_0, v_1, \dots, v_r)$  destes espaços gera um espaço de funções  $f: [u_0, u_n] \times [v_0, v_r] \rightarrow \mathfrak{R}$  que são polinomiais por partes. Se nos restringirmos apenas às *splines* que se anulam fora dos respectivos intervalos, obteremos um subespaço de  $S$  que pode ser gerado por uma base  $B^{m,k}$  obtida pelo produto tensor das base *B-splines* correspondentes, ou seja:

$$B^{m,k} = \left\{ N_{i,j}^{m,k} = N_{i,m}(u) \cdot N_{j,k}(v); \quad 0 \leq i \leq n-m-1, 0 \leq j \leq r-k-1 \right\}, \quad (A-19)$$

onde  $(\cdot)$  indica o produto usual de funções.

Esta base herda as propriedades das bases que a formam. Por exemplo, teremos partição da unidade no retângulo  $[u_m, u_{n-m}] \times [v_k, v_{r-k}]$ . Analogamente às curvas *splines*, podemos então definir uma superfície *spline*  $W$ , num espaço euclidiano  $\mathfrak{R}^l$ , pela expressão:

$$W(u, v) = \sum_{i=0}^{n-m-1} \sum_{j=0}^{r-k-1} P_{i,j} N_{i,m}(u) N_{j,k}(v), \quad (u, v) \in [u_m, u_{n-m}] \times [v_k, v_{r-k}] \quad (A-20)$$

onde  $\{P_{i,j}; 0 \leq i \leq n-m-1 \text{ e } 0 \leq j \leq r-k-1\} \subset \mathfrak{R}^l$ .

## A.4 Operadores

Sejam  $f: \Delta \subset \mathfrak{R}^n \rightarrow \mathfrak{R}$ ;  $f = f(x_1, x_2, \dots, x_n)$  uma função e  $F: \Delta \subset \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ ;  $F(x_1, x_2, \dots, x_n) = (F_1(x_1, x_2, \dots, x_n), F_2(x_1, x_2, \dots, x_n), \dots, F_n(x_1, x_2, \dots, x_n))$ , um campo vetorial. Se  $f$  (respectivamente  $F_i$ ,  $i=1, \dots, n$ ) possui derivadas parciais até a ordem  $r \geq 1$  contínuas em  $\Delta$  então dizemos que  $f$  é de classe  $C^r$  (respectivamente,  $F$  é de classe  $C^r$ ). Podemos então definir os seguintes operadores:

$$\text{Gradiente: } \vec{\nabla} f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right).$$

$$\text{Divergente: } \vec{\nabla} \cdot f = \frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial x_2} + \dots + \frac{\partial f}{\partial x_n}.$$

Laplaciano:

$$\Delta f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \dots + \frac{\partial^2 f}{\partial x_n^2}, \quad \Delta F = \left( \frac{\partial^2 F_1}{\partial x_1^2} + \frac{\partial^2 F_1}{\partial x_2^2} + \dots + \frac{\partial^2 F_1}{\partial x_n^2}, \dots, \frac{\partial^2 F_n}{\partial x_1^2} + \frac{\partial^2 F_n}{\partial x_2^2} + \dots + \frac{\partial^2 F_n}{\partial x_n^2} \right).$$

$$\text{Rotacional de } F (m=n=3): \vec{\nabla} \times F = \left( \frac{\partial F_3}{\partial x_2} - \frac{\partial F_2}{\partial x_3}, \frac{\partial F_1}{\partial x_3} - \frac{\partial F_3}{\partial x_1}, \frac{\partial F_2}{\partial x_1} - \frac{\partial F_1}{\partial x_2} \right)$$

Para o caso da vorticidade temos  $\vec{\xi} = \vec{\nabla} \times \vec{v}$ , onde  $\vec{v} = (v_1, v_2, v_3)$  é o campo de velocidades. Portanto:

$$\vec{\xi} = (\xi_1, \xi_2, \xi_3) = \left( \frac{\partial v_3}{\partial x_2} - \frac{\partial v_2}{\partial x_3}, \frac{\partial v_1}{\partial x_3} - \frac{\partial v_3}{\partial x_1}, \frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2} \right).$$

$$\text{Diferencial de } F: DF(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \dots & \frac{\partial F_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \dots & \frac{\partial F_n}{\partial x_n} \end{pmatrix},$$

Se tivermos um campo vetorial do tipo  $F: I \times \Delta \rightarrow \mathfrak{R}^n$ ,  $I \subset \mathfrak{R}$  e  $\Delta \subset \mathfrak{R}^n$ , então denotamos a diferencial acima por  $D_2F$  indicando que as derivadas são tomadas com relação à segunda variável  $x \in \mathfrak{R}^n$ .

## A.5 Definições

5.a - Homeomorfismo entre os conjuntos  $\Delta_1 \subset \mathfrak{R}^n$  e  $\Delta_2 \subset \mathfrak{R}^m$ :

É uma aplicação bijetora  $h: \Delta_1 \rightarrow \Delta_2$  e contínua, cuja inversa  $h^{-1}: \Delta_2 \rightarrow \Delta_1$  também é contínua.

5.b - Difeomorfismo de Classe  $C^r$ :

Sejam  $\Delta_1 \subset \mathfrak{R}^n$  e  $\Delta_2 \subset \mathfrak{R}^m$  conjuntos abertos. Então dizemos que uma aplicação bijetora  $h: \Delta_1 \rightarrow \Delta_2$ ;  $h(x) = (h_1(x), \dots, h_m(x))$ , é um difeomorfismo de  $\Delta_1 \subset \mathfrak{R}^n$  sobre  $\Delta_2 \subset \mathfrak{R}^m$  quando é diferenciável e sua inversa também o é; ou seja, quando existem as diferenciais  $Dh$  e  $Dh^{-1}$  para todo ponto de  $\Delta_1$  e  $\Delta_2$ , respectivamente. Neste caso dizemos que  $h$  é de classe  $C^1$ . Se  $h$  é  $r$  vezes diferenciável em  $\Delta_1$  ( $r > 1$ ), ou equivalentemente, se as funções coordenadas  $h_i$  são tais que suas derivadas parciais até a ordem  $r$  são contínuas em  $\Delta_1$ , ocorrendo o mesmo para  $h^{-1}$  em  $\Delta_2$ , então dizemos que  $h$  é um difeomorfismo de classe  $C^r$ .

### 5.c - Difeomorfismo Local:

Dizemos que uma aplicação diferenciável  $h: \Delta_1 \rightarrow \mathfrak{R}^m$ , definida no aberto  $\Delta_1 \subset \mathfrak{R}^n$ , é um difeomorfismo local quando para cada  $x \in \Delta_1$  existe um aberto  $V_x$ , com  $x \in V_x \subset \Delta_1$ , tal que a restrição de  $h$  a  $V_x$  é um difeomorfismo sobre um aberto  $W_{h(x)} \subset \mathfrak{R}^m$ .

## Referências

- [Alexa (2003)] Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C. T., 2003a. "Computing and rendering point set surfaces". IEEE Transactions on Visualization and Computer Graphics 9 (1), 3–15.
- [Alexa (2004)] Alexa, M., Gross, M., Pauly, M., Pfister, H., Stamminger, M., Zwicker, M., 2004. "Point-based computer graphics". In: SIGGRAPH 2004 Course Notes.
- [Amada(2004)] T. Amada, M. Imura, Y. Yasumuro, Y. Manabe and K. Chihara, "Particle-Based Fluid Simulation on GPU", ACM Workshop on General-Purpose Computing on Graphics Processors and SIGGRAPH 2004.
- [Amenta (2001)] Amenta, N., Choi, S., Kolluri, R. V., "The power crust", Proceedings of the sixth ACM symposium on Solid modeling and applications - 2001, p. 249-266, ISBN:1-58113-366-9, ACM Press, New York, NY, USA.
- [Banks-Singer (1995)] D.C. Banks, B.A. Singer, "A Predictor-Corrector Technique for Visualizing Unsteady Flow", IEEE Trans. on Visualization and Comp. Graphics Numerical 1, No. 2, June(1995).
- [Battlke at al.(1996)] H. Battlke, D. Stalling, H. Hege, "Fast Line Integral Convolution for Arbitrary Surfaces in 3D". Tech. Report (SC-96-59.ps) of Konrad-Zuse-Zentrum für Informationstechnik Berlin, December 1996.
- [Bernardini (1999)] Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G., "The Ball-Pivoting Algorithm for Surface Reconstruction", IEEE Transactions on Visualization and Computer Graphics, vol. 5, n. 4, p. 349-359, 1999.
- [Bolz(2003)] Jeff Bolz, Ian Farmer, Eitan Grinspun, Peter Schröder, "Sparse matrix solvers on the GPU: conjugate gradients and multigrid", ACM Trans. Graph., vol. 22, number 3, pp. 917-924, ACM Press, 2003.

- [Botsch (2002)] Botsch, M., Wiratanaya, A., Kobbelt, L., 2002. Efficient high quality rendering of point sampled geometry. In: Proc. of Eurographics Workshop on Rendering 02.
- [Botsch (2003)] Botsch, M., Kobbelt, L., 2003. "High-quality point-based rendering on modern GPUs". In: Proc. of Pacific Graphics 03.
- [Botsch (2004)] Botsch, M., Spornat, M., Kobbelt, L., 2004. Phong splatting. In: Proc. Of Symp. on Point-Based Graphics 04.
- [Buck(2004)] Ian Buck, Tim Foley, Daniel Horn, Jeremy Sugerman, Kayvon Fatahalian, Mike Houston, Pat Hanrahan, "Brook for GPUs: stream computing on graphics hardware", ACM Trans. Graph., vol. 23, n. 3, pp: 777-786, ACM Press, 2004.
- [Butkov (1968)] Butkov, Eugene, Mathematical Physics, Addison Wesley, 1968, ISBN: 0201007274.
- [Cabral-Leedom(1993)] B. Cabral, L. Leedom, "Imaging Vector Fields using Line Integral Convolution", S. Incerti, Proceedings SIGGRAPH '93, Anaheim, California, August(1993), 263--272.
- [Carlson (2002)] Mark Carlson, Peter J. Mucha, R. Brooks Van Horn III and Greg Turk , "Melting and Flowing", ACM SIGGRAPH Symposium on Computer Animation San Antonio, Texas, July 21-22, 2002
- [Carlson (2004)] Mark Carlson, Peter J. Mucha and Greg Turk , "Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid", ACM SIGGRAPH 2004
- [Carneiro at al.(1996)] B. P. Carneiro, C. Silva and A. E. Kaufman, "Tetra-Cubes: An algorithm to generate 3D isosurfaces based upon tetrahedra", Anais do IX SIBGRAPI, Outubro 1996.
- [Case-Zweifel (1965)] K. M. Case and P. F. Zweifel, "Linear Transport Theory", Addison-Wesley Publishing Company, 1967.
- [Chapra-Canale(1988)] C. Chapra and R.P. Canale. "Numerical Methods for Engineers." MacGraw-Hill International Editions, 1988.
- [Chong-Perry(1990)] M. S. Chong and A. E. Perry, "A general classification of three-dimensional flow fields", Phys. Fluids A 2 (5), May 1990, 765-783.

- [Chorin-Marsden (1990)] A. J. Chorin and J. E. Marsden. "A Mathematical Introduction to Fluid Mechanics", Springer-Verlag. Texts in Applied Mathematics 4. Second Edition., New York, 1990.
- [Cignoni et al.(1994)] P. Cignoni, L. De Floriani, C. Montani, E. Puppo and R. Scopigno, "Multiresolution Modeling and Visualization of Volume Data based on Simplicial Complexes", Tech. Report. ACM-VV94.ps.
- [Cohen et al. (1996)] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, F. Brooks, and W. Wright. "Simplification Envelopes". Tech. Report.
- [Comba (2003)] J. Comba, C. Dietrich, C. Pagot, C. Scheidegger, "Computation on GPUs: from a programmable pipeline to an efficient stream processor", Revista de Informática Teórica e Aplicada, v. 10, n.1, pages 41-70, 2003. ISSN 01034308.
- [Constantin-Procaccia(1995)] P. Constantin and I. Procaccia, "Creation and dynamics of vortex tubes in three-dimensional turbulence", Physical Review E 51 (4), 3207-3222, April 1995.
- [Desbrun-Gascuel (1998)] M. Desbrun and M. Cani-Gascuel. "Active implicit surface for animation". In Graphics Interface, 1998.
- [Doug(2004)] Doug L. James and Dinesh K. Pai. "BD-Tree: Output-Sensitive Collision Detection for Reduced Deformable Models". ACM Transactions on Graphics (SIGGRAPH 2004), 23(3), 2004.
- [Dx (1998)] Web Site:  
<http://www.tc.cornell.edu/Visualization/vis.html>
- [Eberly (2003)] D. Eberly. "Derivative Approximation by Finite Differences". Magic Software, Inc. web Site:  
<http://www.magic-software.com>, January 21, 2003
- [Enright (2002)] Enright, D., Marschner S., Fedkiw, R., "Animation and rendering of complex water surfaces". Proceedings of the 29th annual conference on Computer graphics and interactive techniques – SIGGRAPH 2002, pg. 736-744, ACM Press ISBN:1-58113-521-1.
- [Fedkiw(2001)] Ronald Fedkiw, Jos Stam and Henrik Wann Jensen, "Visual Simulation of Smoke", In



- SIGGRAPH 2001 Conference Proceedings, Annual Conference Series, August 2001, 15-22.
- [Feldman(2003)] Feldman, B. E., O'Brien, J. F., Arikan, O., "Animating Suspended Particle Explosions." The Proceedings of ACM SIGGRAPH 2003, San Diego, California, July, pp. 708-715.
- [Foster(1997)] Nick Foster, Dimitris Metaxas, Controlling Fluid Animation, CGI '97: Proceedings of the 1997 Conference on Computer Graphics International, ISBN: 0-8186-7825-9, pp. 178, IEEE Computer Society, 1997.
- [Foster-Metaxas (1997)] Nick Foster, Dimitris Metaxas, "Modeling the Motion of a Hot, Turbulent Gas". In Computer Graphics, pages 181–188, ACM SIGGRAPH, 1997.
- [Forsell-Cohen(1995)] L. Forsell, S. D. Cohen, "Using Line Integral Convolution for Flow Visualization: Curvilinear Grids, Variable-Speed Animation, and Unsteady Flows", IEEE Trans. on Visualization and Comp. Graphics Numerical 1, NO.2, June(1995).
- [Geiger(2003)] Geiger, W., Rasmussen, N., Hoon, S. and Fedkiw, R., "Big Bangs", SIGGRAPH 2003 Sketches and Applications, 2003.
- [Goodnight(2003)] Nolan Goodnight, Cliff Woolley, Gregory Lewin, David Luebke, Greg Humphreys, "A multigrid solver for boundary value problems using programmable graphics hardware", HWWS '03: Proceedings of the ACM SIGGRAPH / EUROGRAPHICS conference on Graphics hardware, 2003, ISBN: 1-58113-739-7, pp.102-111.
- [Goktekin (2004)] Goktekin, T. G., Bargteil, A. W., O'Brien, J. F., "A Method for Animating Viscoelastic Fluids." ACM SIGGRAPH 2004.
- [GPUGems(2004)] "GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics", editor: Randima Fernando, Addison-Wesley Professional, ISBN: 0321228324, 2004.
- [GPUGems2(2005)] "GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation", editores: Matt Pharr, Randima Fernando, Addison-Wesley Professional, ISBN: 0321335597, 2005.

- [Hamann et al.(1995)] B. Hamann, D. Wu and R. J. Moorhead II, "On Particle Path Generation Based on Quadrilinear Interpolation and Bernstein-Bézier Polynomials", IEEE Transactions on Visualization and Computer Graphics, 1 (3), September 1995.
- [Hammersley (1964)] J. M. Hammersley and D C Handscomb, "The Monte Carlo Method", Methuen, London (1964).
- [Heckbert (1989)] Heckbert, P. S., 1989. Fundamentals of texture mapping and image warping. Master's thesis, University of California at Berkeley.
- [Helman-Hesselink(1991)] James L. Helman and Lambertus Hesselink, "Visualizing Vector Field Topology in Fluid Flows", IEEE Computer Graphics & Applications, May 1990.
- [Hesselink et al. (1994)] L. Hesselink, F. H. Post and J. J. van Wijk, "Research Issues in Vector and Tensor Field Visualization", IEEE Computer Graphics & Applications, March 1994.
- [Hesselink et al. (1997)] L. Hesselink, Y. Levy and Y. Lavin, "The Topology of Symmetric, Second-Order 3D Tensor Fields", IEEE Transactions on Visualization and Computer Graphics, 3 (1), January-March 1997.
- [Hirsch(1988)] C. Hirsch, "Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization", Vol. 1, John Wiley & Sons, 1988.
- [Hughes(1987)] Thomas J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, 1987.
- [Interrante-Grosch(1997)] V. Interrante and C. Grosch, "Strategies for Effectively Visualizing a 3D Flow Using Volume Line Integral Convolution", Proceedings of Visualization 97.
- [Kerlow (1996)] Kerlow, Isaac Victor. "The Art of 3-D Computer Animation and Imaging". Van Nostrand Reinhold, New York, NY, 1996. ISBN: 047136004X
- [Kilgariff(2005)] Emmett Kilgariff and Randima Fernando, "The GeForce 6 Series GPU Architecture", GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, editors: Matt Pharr, Randima

- Fernando, Addison-Wesley Professional, ISBN: 0321335597, 2005.
- [Kipfer(2004)] Peter Kipfer, Mark Segal, Rüdiger Westermann, "ÜberFlow: A GPU-Based Particle Engine", Eurographics Symposium Proceedings Graphics Hardware 2004, pp. 115-122.
- [Klemp (1978)] J.B. Klemp and R.B. Wilhelmson, "The Simulation of Three-Dimensional Convective Storm Dynamics", J. Atmos. Sci., Vol. 35, p. 1070 – 1096, 1978.
- [Knight-Mallinson(1996)] D. Knight and Gordon Mallison, "Visualizing Unstructured Flow Data Using Dual Stream Functions", IEEE Transactions on Visualization and Computer Graphics, 2 (4), December 1996.
- [Krüger(2004)] J. Krüger, T. Schiwietz, P. Kipfer and R. Westermann, "Numerical Simulations on PC Graphics Hardware", ParSim 2004 (Special Session of EuroPVM/MPI 2004)
- [Krüger (2005)] J. Krüger, P. Kipfer, P. Kondratieva, R. Westermann, "A Particle System for Interactive Visualization of 3D Flows", pre-print, Accepted for Publication in IEEE Transactions on Visualization and Computer Graphics
- [Krueger(1991)] W. Krueger, "The application of transport theory to visualization of 3-D scalar data fields", Computers in Physics, 5 (4), July/August 1991.
- [Landau-Lifshitz(1959)] L. D. Landau and E. M. Lifshitz, "Fluid Mechanics", Pergamon Press, 1959.
- [Levoy (1985)] Levoy, M., Whitted, T., January 1985. "The use of points as display primitives". Tech. rep., CS Department, University of North Carolina at Chapel Hill.
- [Liboff (1990)] R. L. Liboff, "Kinetic Theory: Classical, Quantum, and Relativistic Descriptions", Prentice-Hall International Editions, 1990.
- [Liu at al. (2003)] M.B. Liu, G.R. Liu and K.Y. Lam. "Constructing smoothing functions in smoothed particle hydrodynamics with applications", Journal of Computational and Applied Mathematics, No. 155, pp. 263–284, 2003.

- [Lorensen (1987)] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface construction Algorithm", Computer Graphics, Vol 21, No. 4, July 1987.
- [Losasso (2004)] Losasso, F., Gibou, F. and Fedkiw, R., "Simulating Water and Smoke with an Octree Data Structure", SIGGRAPH 2004, ACM TOG 23, 457-462 (2004).
- [Mark(2003)] William R. Mark, R. Steven Glanville, Kurt Akeley, Mark J. Kilgard, "CG: a system for programming graphics hardware in a C-like language", ACM Trans. Graph., vol. 2, n. 3, pp-896-907, ACM Press – 2003.
- [Max (1995)] N. Max, "Optical Models for Direct Volume Rendering", IEEE Transactions on Visualization and Computer Graphics, 1 (2), 99-108, June 1995.
- [McCool(2002)] Michael D. McCool, Zheng Qin, and Tiberiu S. Popa, "Shader Metaprogramming", SIGGRAPH / Eurographics Graphics Hardware Workshop, pp. 57-68, 2002.
- [Metaxas (1996)] D. Metaxas. "Physics-Based Deformable Models: Applications to Computer Vision", Graphics and Medical Imaging. Kluwer, Boston, MA, 1996.
- [Müller (2004)] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, M. Gross, "Interaction of Fluids with Deformable Solids," Journal of Computer Animation and Virtual Worlds (CAVW), vol. 15, no. 3-4, pp. 159-171, July 2004.
- [Müller (2003)] M. Müller, D. Charypar and M. Gross. "Particle-Based Fluid Simulation for Interactive Applications", SIGGRAPH 2003.
- [Nguyen (2002)] Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen, "Physically Based Modeling and Animation of Fire". Proceedings of SIGGRAPH 2002.
- [Pauly (2001)] Pauly, M., Gross, M., 2001. "Spectral processing of point-sampled geometry". In: Proc. of ACM SIGGRAPH 01.

- [Pauly (2003a)] Pauly, M., "Point primitives for interactive modeling and processing of 3D geometry". Ph.D. thesis, ETH Zürich, 2003.
- [Pauly (2003b)] Pauly, M., Keiser, R., Kobbelt, L., Gross, M., "Shape modeling with point-sampled geometry". In: Proc. of ACG SIGGRAPH 03.
- [Persiano(1996)] R. M. Persiano, "Bases da Modelagem Geométrica", 10a Escola de Computação, Campinas, Julho de 1996.
- [Pfister (2003)] Pfister, H., Zwicker, M., van Baar, J., Gross, M., 2000. "Surfels: Surface elements as rendering primitives". In: Proc. of ACM SIGGRAPH 00. pp. 335–342.
- [Premoze(2001)] Simon Premoze, Michael Ashikhmin, "Rendering Natural Waters", Computer Graphics Forum, v. 20, no. 4 (2001), pp. 189-200.
- [Premoze at al. (2003)] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn and R. Whitaker, "Particle-Based Simulation of Fluids", EUROGRAPHICS, Volume 22, Number 3, 2003.
- [Rasmussen (2003)] Rasmussen, N., Nguyen, D., Geiger, W. and Fedkiw, R., "Smoke Simulation for Large Scale Phenomena", SIGGRAPH 2003, ACM TOG 22, p. 703-707.
- [Robinson (1989)] S. K. Robinson, "A review of vortex structures and associated coherent motions in turbulent boudary layers", Proc. Second IUTAM Symp. Structure of Turbulence and Drag Reduction, Fed. Inst. of Technology, Zurich, Switzerland, pp. 25-28, July 1989.
- [Roseblum et al.(1994)] "Scientific Visualization: Advances and Challenges", edited by L. Roseblum et al., Academic Press(1994).
- [Rost(2004)] Randi J. Rost, "OpenGL Shading Language", Addison-Wesley Professional, ISBN: 0321197895, 2004.
- [Rusinkiewicz (2000)] Rusinkiewicz, S., Levoy, M., 2000. "QSplat: a multiresolution point rendering system for large meshes". In: Proc. of ACM SIGGRAPH 00. pp. 343–352.

- [Saffman(1995)] P. G. Saffman, "Vortex Dynamics", Cambridge Monographs on Mechanics and Applied Mathematics, 1995.
- [Samtaney at al.(1994)] R. Samtaney, D. Silver, N. Zabusky, J. Cao, "Visualizing Features and Tracking their Evolution", IEEE Computer 27, NO. 7, pp. 20-27, July 1994.
- [Scheidegger (2004)] C. Scheidegger, J. Comba, R. Cunha, "Navier-Stokes on Programmable Graphics Hardware Using SMAC", Proceedings of XVII SIBGRAPI - II SIACG 2004. Pages 300-307. IEEE Press ISBN 0-7695-2227-0, 2004.
- [Sethian (1996)] J. A. Sethian, "Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences", Cambridge University Press, 1996.
- [Shen at al.(1996)] Han-Wei Shen, C. R. Johnson and Kwan-Liu Ma, "Visualizing Vector Fields Using Line Integral Convolution and Dye Advection", ICASE Tech. Report.
- [Silver-Zabusky (1996)] D. Silver and N. Zabusky, "Quantifying Visualizations for Reduced Modeling in Nonlinear Science: Extracting Structures from Data Sets", J. Visual Comm. and Image Representation, Vol. 4, No. 1, pp. 46-61, March 1993.
- [Sotomayor (1979)] J. Sotomayor, "Lições de Equações Diferenciais Ordinárias", Projeto Euclides, Gráfica Editora Hamburgo Ltda, São Paulo, 1979.
- [Spiegel (1992)] M. R. Spiegel, "Manual de Fórmulas, Métodos e Tabelas de Matemática", Segunda Edição Revisada e Ampliada, McGraw-Hill Ltda, Coleção Schaum, 1992.
- [Stam (1999)] Jos Stam, "Stable Fluids", In SIGGRAPH 99 Conference Proceedings, Annual Conference Series, pp. 121-128, August 1999.
- [Stam(2003)] Jos Stam, "Real-Time Fluid Dynamics for Games". Proceedings of the Game Developer Conference, March 2003.
- [Symon (1982)] K. R. Symon, "Mecânica", Editora Campos Ltda, Rio de Janeiro, 1982.

- [Szeliski (1992)] Szeliski, R., Tonnesen, D., 1992. "Surface modeling with oriented particle systems". In: Proc. of ACM SIGGRAPH 92.
- [Takens (1970)] F. Takens, "Partially Hyperbolic Fixed Points", Topology, Vol. 10, pp. 133-147, Pergamon Press, 1971.
- [Taylor (1996)] Taylor, R., "The Encyclopedia of Animation Techniques". Quarto Inc., London, England, 1996. ISBN: 156138531X
- [Teschner (2005)] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, P. Volino, "Collision Detection for Deformable Objects," Computer Graphics Forum, vol. 24, no. 1, pp. 61-81, March 2005.
- [Treuille (2003)] Adrien Treuille, Antoine McNamara, Zoran Popovic and Jos Stam, "Keyframe Control of Smoke Simulations", ACM Transactions On Graphics (TOG), Volume 22, Issue 3 (July 2003) : Proceedings of SIGGRAPH 2003, 716-723.
- [Ueng, et al. (1997)] Shyh-Kuang Ueng, K. Sikorski and Kwan Liu Ma, "Out-Of-Core Streamline Visualization on Large Unstructured Meshes", IEEE Transaction on Visualization and Computer Graphics, Vol. 3, No. 4, December 1997. ICASE Tech. Report No. 97-22, April 1997.
- [Ueng (1996)] S.Ueng, C. Sikorski, and Kwan-Liu Ma, "Efficient Streamline, Streamribbon, and Streamtube Constructions on Unstructured Grids", IEEE Trans. on Visualization and Comp. Graphics, Vol. 1, NO.2, June(1996).
- [Villasenor-Vincent (1992)] J. Villasenor and A. Vincent, "An Algorithm for Space recognition and Time Tracking of Vorticity Tubes in turbulence", CVGIP: Image Understanding", Vol. 55, No. 1, January(1992), pp. 27-35.
- [Wang(2005)] Huamin Wang, Peter J. Mucha and Greg Turk, "Water Drops on Surfaces", pre-print, to appear in SIGGRAPH 2005.

- [Watt (1991)] Watt, A., Watt, M., "Advanced Rendering and Animation Techniques: Theory and Practice. Addison-Wesley, 1992. ISBN: 0201544121
- [Westover (1989)] Westover, Lee, "Interactive volume rendering", VVS '89: Proceedings of Workshop on Volume Visualization - Chapel Hill, North Carolina, 1989, p. 9-16, ACM Press New York, NY, USA.
- [Westover (1990)] Westover, L., "Footprint Evaluation for Volume Rendering", Computer Graphics, Vol. 24, No. 4, pp. 367-376 (1990).
- [Wijk(2002)] Jarke J. van Wijk, "Image based flow visualization, SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, ISBN: 1-58113-521-1, pp. 745-754, ACM Press, 2002.
- [Wilhelms-Gelder (1992)] J. Wilhelms and A. Van Gelder, "Octrees for Faster Isosurface Generation," ACM Trans. Graphics, Vol. 11, No. 3, July 1992, pp. 201-227.
- [Witkin (1994)] Witkin, A., Heckbert, P., 1994. "Using particles to sample and control implicit surfaces". In: Proc. of ACM SIGGRAPH 94.
- [Witting (1999)] Patrick Witting, "Computational Fluid Dynamics in a Traditional Animation Environment", Siggraph, 1999.
- [Yeung et al.(1993)] R. W. Yeung, S. H. Sphaier, and M. Vaidhyanathan, "Unsteady Flow About Bluff Cylinders", International Journal of Offshore and Polar Engineering 3 (2), June 1993.
- [Zwicker (2001)] Zwicker, M., Pfister, H., van Baar, J., Gross, M., 2001. "Surface splatting". In: Proc. of ACM SIGGRAPH 01. pp. 371-378.
- [Zwicker (2003)] Zwicker, M., 2003. "Continuous reconstruction, rendering, and editing of point sampled surfaces". Ph.D. thesis, ETH Zürich.
- [Zwicker (2004)] Zwicker, M., Rasänen, J., Botsch, M., Dachsbacher, C., Pauly, M., 2004. Perspective accurate splatting. In: Proc. of Graphics Interface 2004.