

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Leandro de Pinho Monteiro

Soluções Numéricas em um Modelo de Tecidos Baseado na Superfície de Cosserat

Orientadora: Profa. Dra. Wu, Shin-Ting

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Campinas, SP
Janeiro de 2008

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Leandro de Pinho Monteiro

Soluções Numéricas em um Modelo de Tecidos Baseado na Superfície de Cosserat

Orientadora: Profa. Dra. Wu, Shin-Ting

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Banca Examinadora

Profa. Dra. Rosângela Barros Zanoni Lopes Moreno .. DEP/FEM/UNICAMP
Prof. Dr. Léo Pini Magalhães DCA/FEEC/UNICAMP
Prof. Dr. Clésio Luis Tozzi DCA/FEEC/UNICAMP

Campinas, SP

Janeiro de 2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

M764s Monteiro, Leandro de Pinho
Soluções numéricas em um modelo de tecidos baseado na superfície de Cosserat / Leandro de Pinho Monteiro. – Campinas, SP: [s.n.], 2008.

Orientador: Wu, Shin-Ting
Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Computação gráfica. 2. Simulação e modelagem.
3. Animação por computador. 4. Superfícies - Modelos.
5. Simulação por computador. I. WU, Shin-Ting.
II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Numerical solutions to a cloth model based on the Cosserat surface
Palavras-chave em Inglês: Computer graphics, Physics simulation, Cloth simulation, Cloth modeling and animation, Continuum mechanics, Cosserat surface
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca Examinadora: Rosângela Barros Zanoni Lopes Moreno, Léo Pini Magalhães, Clésio Luis Tozzi
Data da defesa: 18/01/2008
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Leandro de Pinho Monteiro

Data da Defesa: 18 de janeiro de 2008

Título da Tese: "Soluções Numéricas em um Modelo de Tecidos Baseado na Superfície de Cosserat"

Profa. Dra. Wu Shin-Ting (Presidente): _____

Dra. Rosângela Barros Zanoni Lopes Moreno: _____

Prof. Dr. Léo Pini Magalhães: _____

Prof. Dr. Clésio Luis Tozzi: _____

Resumo

Atualmente, os modelos físicos são amplamente utilizados em simulações computacionais de vestuários e tecidos em geral. Eles podem ser divididos em duas abordagens: mecânica de partículas e mecânica de contínuos. A mecânica de partículas é hoje a abordagem mais utilizada, possuindo vantagens como a simplicidade da formulação e um bom desempenho computacional. Já a mecânica de contínuos é reconhecida por ser a abordagem mais precisa sob o ponto de vista físico, porém tem um alto custo computacional. Muitas vezes, o baixo desempenho da mecânica de contínuos está relacionado à solução numérica utilizada para obter diferentes estados do tecido ao longo do tempo, em decorrência da sua solução demandar resolução de sistemas de equações de um grande número de variáveis. As soluções encontradas na literatura são técnicas de elementos finitos ou de diferenças finitas, com integração semi-implícita ou implícita, todas com uma complexidade computacional não linear. Desta forma, este trabalho visa analisar a adequabilidade de utilizar métodos explícitos, que tem um comportamento linear, em um modelo físico fundamentado na teoria de superfície de Cosserat, cujos autores provam que ele consegue produzir dobras sob forças de compressão. Discretizando espacialmente este modelo com a técnica das diferenças finitas, métodos explícitos são avaliados para solucionar as equações diferenciais ordinárias resultantes. Com base nestas avaliações, foi selecionado o método de Verlet e implementado um simulador da dinâmica de malhas retangulares provido de uma interface gráfica interativa. Isso viabilizou uma validação prática do modelo, demonstrando a sua superioridade na produção de dobras em relação aos outros modelos existentes.

Palavras-chave: Computação Gráfica, Simulações Físicas, Simulação de Tecidos, Modelagem e Animação de Tecidos, Mecânica de Contínuos, Superfície de Cosserat.

Abstract

Currently, physics-based models are frequently used for cloth simulations. There are basically two approaches: particle mechanics and continuum mechanics. The particle mechanics is at the present time the most widely used method, having some advantages like an easy formulation and a high performance. The continuum mechanics is regarded as being a more precise approach, but it has a high computational cost. This low performance is mostly owing to the numerical solution that obtains cloth states along the time, because its formulation is based involves a large number of variables. The solutions found in the literature are the finite element and the finite differences techniques, with the semi-implicit and implicit integration. They possess a non linear computational complexity. So, this work aims to analyze the suitability of application of an explicit integration technique, that has a linear behavior, to a cloth model founded on the theory of Cosserat surface, whose authors claim to have proved it can produce natural folds under pure compression forces. Based on this evaluation, the Verlet method was selected and a rectangular mesh dynamics simulator with an interactive graphics interface has been implemented. This permits a practice validation of the cloth model, showing its superiority on production of folds compared with the other models.

Keywords: Computer Graphics, Physics Simulation, Cloth Simulation, Cloth Modeling and Animation, Continuum Mechanics, Cosserat Surface.

À minha mãe.

Agradecimentos

À minha mãe, Leda, pelo amor, suporte, carinho e paciência em todos os momentos.

À minha orientadora, Ting, pela ajuda e apoio intelectual e pela amizade conquistada nestes anos de pesquisa.

À professora Rosângela Barros Zanoni Lopes Moreno e aos professores Léo Pini Magalhães e Clésio Luis Tozzi, por aceitarem participar da banca examinadora desta dissertação. Em especial, agradeço ao professor Clésio pela atenção dispensada a mim na disciplina de Visão Robótica.

À minha namorada, Mariana, pelo amor, companheirismo e paciência durante a longa etapa dedicada a este mestrado.

Aos meus amigos do LCA, em especial ao Adler, Harlen, Jeremias e Alex.

Às pessoas que de algum modo contribuíram, por existirem.

À CAPES, pelo suporte financeiro.

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xv
Lista de Símbolos	xv
Trabalhos Publicados Pelo Autor	xix
1 Introdução	1
1.1 Simulação de Tecidos	2
1.1.1 Modelo de Tecido	2
1.1.2 Integração Temporal	5
1.1.3 Problemas	7
1.2 Conteúdo do Trabalho	8
1.2.1 Motivação	8
1.2.2 Hipótese	9
1.2.3 Objetivos	9
1.2.4 Escopo	9
1.2.5 Contribuições	10
1.3 Organização da Dissertação	10
2 Simulações de Tecidos	13
2.1 Modelos Físicos baseados na Mecânica de Partículas	14
2.1.1 Integrações Explícitas	14
2.1.2 Integrações Implícitas	15
2.1.3 Integrações Semi-implícitas	16
2.2 Modelos Físicos baseados na Mecânica de Contínuos	17
2.2.1 Integrações Implícitas	17
2.2.2 Integrações Semi-implícitas	18
2.3 Classificação dos Modelos	21
2.4 Uma modelagem baseada em uma Superfície de Cosserat	22
2.4.1 Superfície de Cosserat	22
2.4.2 Integração Semi-Implícita	29
2.5 Considerações Finais	30

3	Soluções Numéricas	33
3.1	Discretizações Espaciais	34
3.2	Métodos Explícitos de Integração	41
3.2.1	Método de Euler	42
3.2.2	Método de Runge-Kutta de Segunda Ordem	44
3.2.3	Método de Runge-Kutta de Quarta Ordem	46
3.2.4	Método de Verlet	49
3.2.5	Método <i>Leapfrog</i>	50
3.3	Trabalhos Relacionados	52
3.4	Resultados	57
3.4.1	Simulações de Teste	57
3.4.2	Método de Euler	60
3.4.3	Método de Runge-Kutta de Segunda Ordem	61
3.4.4	Método de Runge-Kutta de Quarta Ordem	62
3.4.5	Método de Verlet	64
3.4.6	Método <i>Leapfrog</i>	65
3.5	Análises Comparativas	66
3.5.1	Estabilidade Numérica	66
3.5.2	Desempenho Computacional	68
3.5.3	Qualidade das Imagens	70
3.5.4	Escolha do Método Numérico	70
3.6	Considerações Finais	71
4	Implementação	73
4.1	Modelagem de dados	74
4.1.1	Fluxo de Dados	74
4.1.2	Algoritmos	75
4.1.3	Análise de Requisitos	77
4.1.4	Classe <i>malha</i>	79
4.1.5	Classe <i>render</i>	80
4.2	Interface Gráfica	82
4.2.1	Plataforma Linux	82
4.2.2	Plataforma Windows	84
4.2.3	Exemplos de Simulação	86
4.3	Considerações Finais	89
5	Resultados	91
5.1	Metodologia	92
5.1.1	Especificação de Forças Externas	92
5.1.2	Calibração de Parâmetros do Tecido	93
5.1.3	Resolução da Malha	97
5.2	Simulações	98
5.2.1	<i>Bucklings</i>	99
5.2.2	<i>Drapings</i>	101

5.2.3	<i>Hangings</i>	104
5.3	Validação do Modelo	105
5.3.1	Dinâmica	106
5.3.2	Intuitividade	110
5.4	Considerações Finais	112
6	Conclusões e Trabalhos Futuros	113
	Referências bibliográficas	117

Lista de Figuras

2.1	Modelos massa-mola: (a) Haumann em 1987 [1]; e (b) Provot em 1995 [2].	15
2.2	Resultados do trabalho de Wu e de Melo em 2004 [3]: (a) dobras; e (b) caimento. . .	20
2.3	Referenciais fixo e arbitrário em instantes diferentes.	23
3.1	Malha discreta.	34
3.2	Problemas nas bordas.	36
3.3	Derivadas aproximadas com diferenças finitas: (a) para pontos internos; (b) para pontos nas bordas B1 e B2; e (c) para pontos nas bordas B3 e B4.	40
3.4	(a) Método de Euler explícito e (b) usado por Provot em 1995 [2].	44
3.5	Método de Runge-Kutta de segunda ordem.	46
3.6	Método de Runge-Kutta de quarta ordem.	48
3.7	Método de Verlet.	50
3.8	Método <i>Leapfrog</i>	52
3.9	Imagens reais de um tecido: (a) suspenso por 2 pontos; (b) suspenso por 1 ponto; (c) cobrindo um banco; (d) hasteado; (e) cobrindo uma poltrona e (f) comprimido.	58
3.10	Superfície inicial das simulações: (a) S_1, S_2, S_3, S_5 e S_6 ; e (b) S_4	58
3.11	Resultados visuais do método de Euler.	61
3.12	Resultados visuais do método de Runge-Kutta de segunda ordem.	62
3.13	Resultados visuais do método de Runge-Kutta de quarta ordem.	63
3.14	Resultados visuais do método de Verlet.	64
3.15	Resultados visuais do método <i>Leapfrog</i>	65
3.16	Instabilidade numérica causada por altos valores de Δt	67
3.17	Tempos totais de processamento para executar 10000 iterações.	69
4.1	Fluxograma da dinâmica de tecidos.	74
4.2	Classe <i>malha</i> e suas principais classes.	79
4.3	Modelagem da classe <i>objeto</i>	80
4.4	Modelagem da classe <i>render</i>	81
4.5	Exemplo de um arquivo de entrada com todos os dados de uma simulação.	82
4.6	Interface gráfica para a plataforma Linux.	83
4.7	Janela das restrições da simulação.	83
4.8	Janela de visualização da animação gerada.	84
4.9	Janelas de (a) simulação; e (b) visualização da animação gerada.	85
4.10	Janela de restrições.	86
4.11	Imagens de simulação.	86

4.12	Definição e ilustração das forças pontuais.	88
4.13	Efeitos do coeficiente ρ_D : (a) com ele e (b) sem ele.	89
5.1	Atuação de forças externas pontuais (a) no centro e (b) nos cantos.	93
5.2	Influência dos coeficientes elásticos.	95
5.3	Influência de ϕ	96
5.4	<i>Bucklings</i> em grades 60×60	98
5.5	<i>Bucklings</i> de diferentes materiais de tecido.	99
5.6	Formação de uma gravata borboleta (a) flexível e (b) rígida.	100
5.7	Pontos fixos na malha discreta para formar uma mesa (a) quadrada e (b) redonda. . .	101
5.8	Caimentos de tecidos sobre mesas redondas: (a) real cetim, (b) cetim, (c) real linho e (d) linho.	102
5.9	Caimento de tecido sobre uma mesa (a) redonda e (b) quadrada.	103
5.10	Caimento de um tecido sobre uma poltrona.	103
5.11	Cortinas suspensas de diferentes materiais: (a) seda; (b) algodão; e (c) plastificada. .	104
5.12	Bandeira no estado (a) inicial e (b) deformado.	105
5.13	Dinâmica de <i>bucklings</i>	107
5.14	Dinâmica de <i>drappings</i>	108
5.15	Dinâmica de <i>hangings</i>	109

Lista de Tabelas

2.1	Classificação dos trabalhos apresentados nas seções 2.1 e 2.2.	21
3.1	Passos de tempo máximos e iterações necessárias do método de Euler.	60
3.2	Tempos de processamento do método de Euler.	61
3.3	Passos de tempo máximos e iterações necessária de Runge-Kutta de segunda ordem.	62
3.4	Tempos de processamento do método de Runge-Kutta de segunda ordem.	62
3.5	Passos de tempo máximos e iterações necessárias de Runge-Kutta de quarta ordem.	63
3.6	Tempos de processamento do método de Runge-Kutta de quarta ordem.	63
3.7	Passos de tempo máximos e iterações necessárias do método de Verlet.	64
3.8	Tempos de processamento do método de Verlet.	64
3.9	Passos de tempo máximos e iterações necessárias do método <i>Leapfrog</i>	65
3.10	Tempos de processamento do método <i>Leapfrog</i>	65
5.1	Dados das simulações.	110

Lista de Símbolos

t	- Instante de tempo
t_0	- Instante de tempo inicial
Δt	- Passo de tempo
$s(t)$	- Superfície no instante t
$S = s(t_0)$	- Superfície não deformada correspondente ao instante t_0
x^1 e x^2	- Curvas coordenadas da superfície $s(t)$
$p = (x^1, x^2)$	- Ponto localizado nas coordenadas (x^1, x^2) de $s(t)$
$\mathbf{r} = \mathbf{r}(p)$	- Vetor-posição de um ponto p da superfície $s(t)$
$\mathbf{R} = \mathbf{R}(p)$	- Vetor-posição de um ponto p da superfície S
μ	- Densidade de massa por unidade de área da superfície $s(t)$
ϱ	- Coeficiente de amortecimento do meio em que se encontra a superfície $s(t)$
$\mathbf{f} = \mathbf{f}(p, t)$	- Forças internas de um ponto p da superfície $s(t)$
$\mathbf{F} = \mathbf{F}(p, t)$	- Forças externas de um ponto p da superfície $s(t)$
$\text{vel}(p, t)$	- Velocidade de um ponto p da superfície $s(t)$
$\text{acel}(p, t)$	- Aceleração de um ponto p da superfície $s(t)$
$\mathbf{E}_{\langle \text{tipo} \rangle}$	- Energia associada ao tipo descrito no subíndice
$\mathbf{a}_1 = \mathbf{a}_1(p) = \frac{\partial \mathbf{r}}{\partial x^1}(p)$	- Vetor de base de um ponto p em relação à coordenada x^1
$\mathbf{a}_2 = \mathbf{a}_2(p) = \frac{\partial \mathbf{r}}{\partial x^2}(p)$	- Vetor de base de um ponto p em relação à coordenada x^2
$\mathbf{a}^1 = \mathbf{a}^1(p)$ e $\mathbf{a}^2 = \mathbf{a}^2(p)$	- Vetores de base recíprocos de um ponto p
$\mathbf{n} = \mathbf{n}(p)$	- Vetor normal de um ponto p
$a_{\alpha\beta} = a_{\alpha\beta}(p)$	- Tensor de métrica de um ponto p de $s(t)$
$A_{\alpha\beta} = A_{\alpha\beta}(p)$	- Tensor de métrica de um ponto p de S
$\varepsilon_{\alpha\beta}$	- Deformação da métrica de s em relação à S
$a = \det(a_{\alpha\beta})$	- Determinante da matriz composta pelos elementos do tensor de métrica
$b_{\alpha\beta} = b_{\alpha\beta}(p)$	- Tensor de curvatura de um ponto p de $s(t)$
$B_{\alpha\beta} = B_{\alpha\beta}(p)$	- Tensor de curvatura de um ponto p de S
b_{β}^{α}	- Símbolos de Weingarten
$\kappa_{\alpha\beta}$	- Deformação da curvatura de $s(t)$ em relação à S
$\Gamma_{\alpha\beta}^{\lambda}$	- Símbolos de Christoffel
\mathbf{N}	- Vetor de força de curva medido por unidade de comprimento
\mathbf{M}	- vetor de força diretora medido por unidade de comprimento
$\mathbf{N}^{\alpha} _{\alpha}$	- Força interna de uma superfície de Cosserat
$\zeta_{\alpha\beta}$	- Coeficiente elástico do tecido associado a sua métrica
$\xi_{\alpha\beta}$	- Coeficiente elástico do tecido associado a sua curvatura

- $\phi_{\alpha\beta}$ - Coeficiente elástico do tecido associado à formação de dobras
- $\Phi_{\alpha\beta}$ - Parâmetro do material da superfície $s(t)$ associado a sua métrica
- $\Psi_{\alpha\beta}$ - Parâmetro do material da superfície $s(t)$ associado a sua curvatura
- $\Theta_{\alpha\beta}$ - Parâmetro do material da superfície $s(t)$ associado ao coeficiente de *buckling*
- m e n - Medidas da malha discreta dimensional
- ϱ_D - Coeficiente de amortecimento do meio
- ϱ_F - Coeficiente de densidade de fluido

Trabalhos Publicados Pelo Autor

1. L.P. Monteiro, S.T. Wu. “Simulação Realista de Tecidos”. In *Digital Prodeedings of the V Brazilian Symposium on Computer Games and Digital Entertainment (SBGames 2006)*, p. 22.FP.C., Recife, Pernambuco, Brasil, Setembro 2006. URL <http://www.cin.ufpe.br/sbgames/proceedings/papers.htm>.

Capítulo 1

Introdução

A simulação de tecidos é do interesse de diversas áreas de trabalho, como na propaganda e entretenimento, usando animações computadorizadas, e na indústria têxtil, auxiliando os profissionais da área de moda no projeto de vestuários e malhas em geral. Apesar de várias estratégias para modelagem e animação de tecidos por computador terem sido rapidamente desenvolvidas nas duas últimas décadas, o comportamento realista de sua dinâmica continua sendo um problema desafiador tanto para comunidade de computação gráfica quanto para a de engenharia têxtil. Enquanto os pesquisadores de computação gráfica estão mais interessados em um modelo estável e robusto numericamente para animar vestuários sobre personagens virtuais, os cientistas têxteis priorizam um modelo mecânico que relacione o comportamento de tecidos aos parâmetros de seu material.

Um dos grandes problemas relacionados à animação de tecidos é formular um modelo que agregue realismo e eficiência para atingir um comportamento natural de dobras sob forças de compressão. Fundamentado em superfícies elásticas de Cosserat [4], Wu e de Melo propuseram um modelo de superfície deformável que consegue representar várias características peculiares de tecidos, incluindo a formação natural de dobras sob compressão, através de um número relativamente pequeno de parâmetros [3]. Algumas simplificações foram necessárias para tornar o modelo computacionalmente factível. Embora tais simplificações não devam influenciar a dinâmica da simulação sob o ponto de vista teórico, as implementações disponíveis do modelo não permitem uma validação mais extensa. Esta dissertação apresenta uma alternativa de implementação, tornando os códigos mais modulares, como também viabilizando a validação prática das principais características do modelo.

Este capítulo está organizado da seguinte maneira: a Seção 1.1 apresenta uma breve introdução sobre etapas e problemas envolvidos na simulação de tecidos; a Seção 1.2 expõe o conteúdo deste trabalho, discutindo sua motivação, hipótese, foco, escopo e principais contribuições; por fim, a Seção 1.3 descreve resumidamente como está organizado o restante desta dissertação.

1.1 Simulação de Tecidos

Os tecidos, em geral, possuem um comportamento bem característico, que é de se curvar com facilidade e se esticar com dificuldade. Esta característica faz com que eles sofram muitas deformações, as quais são governadas por leis físicas complexas do mundo real que devem ser levadas em consideração, em uma simulação computacional, para se obter resultados realistas. O paradigma de modelagem de tecidos é o que diferencia duas abordagens físicas: a mecânica de contínuos e a mecânica de partículas, as quais são resumidamente explicadas na Subseção 1.1.1. Para simular a dinâmica de tecidos é preciso ainda obter as variações de sua geometria ao longo do tempo, o que pode ser feito basicamente de três formas: com uma integração implícita, semi-implícita ou explícita, conforme brevemente apresentado na Subseção 1.1.2. A combinação de modelagem apropriada do comportamento de um tecido sob forças e da solução numérica das equações envolvidas resulta em uma seqüência de formas que o tecido pode assumir. Há, porém, alguns problemas relacionados à eficiência e à estabilidade numérica, que são discutidos na Subseção 1.1.3.

1.1.1 Modelo de Tecido

Tecidos são estruturas compostas de fios, entrelaçados em um determinado padrão. Os fios são feitos de fibras, as quais determinam o material do tecido¹. A composição, o padrão de entrelaçamento e o tratamento dado aos fios de um tecido determinam sua reação quando sujeito à ação de forças externas. Devido à diversidade de parâmetros, que combinados resultam em um comportamento peculiar, a modelagem de um tecido pode ser feita de diferentes formas. Pode-se classificar estas formas basicamente em dois tipos: os modelos físicos, por exemplo, nos trabalhos [1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]; e os modelos geométricos, por exemplo [21, 22].

Os modelos geométricos não levam em consideração as propriedades físicas dos tecidos, utilizando entidades e transformações geométricas para descrever estados estacionários dos tecidos. Apesar da simplicidade, eles não são adequados para reproduzir o movimento de tecidos, pois se limitam em descrever suas formas usando um certo padrão pré-definido [22], desprezando os fenômenos naturais que ocorrem na realidade. Já os modelos físicos visam representar com maior fidelidade a complexidade envolvida no comportamento de tecidos, fazendo uso de um mecanismo que satisfaça leis físicas que vigoram na vida real e, portanto, atingem um realismo maior quando comparados aos modelos geométricos.

Os modelos físicos são reconhecidos por serem os mais adequados para produzir uma aparência natural da dinâmica de tecidos. Atualmente, os principais *softwares* gráficos comerciais, como o 3D Studio Max e Maya, utilizam modelos físicos dinâmicos para simular o movimento de tecidos.

¹Estas fibras podem ser naturais como a lã e o algodão ou sintéticas como o elastano e o couro.

Terzopoulos et al. [19] propuseram modelar um tecido como uma superfície deformável cuja dinâmica é governada pela equação

$$\mu \frac{\partial^2 \mathbf{r}(p)}{\partial t^2} + \varrho \frac{\partial \mathbf{r}(p)}{\partial t} + \mathbf{f}(p, t) = \mathbf{F}(p, t), \quad (1.1)$$

na qual \mathbf{r} corresponde ao vetor-posição de um ponto p de uma superfície s em um instante de tempo t e as forças externas $\mathbf{F}(p, t)$ sobre cada ponto $p \in s(t)$ precisam ser balanceadas pela somatória de três elementos, da esquerda para a direita: força correspondente à energia cinemática, forças dissipativas e forças internas. É neste último termo, referente às forças internas de fricção que são geradas pela variação da energia potencial acumulada na superfície deformável, que se distinguem dois tipos de modelos físicos: os que utilizam a mecânica de partículas e os que utilizam a mecânica de contínuos.

Os modelos que utilizam a mecânica de partículas consideram uma superfície deformável como um conjunto de pontos discretos, que são as partículas, dispostos em uma grade na qual são conectados uns aos outros por molas lineares ou não-lineares, responsáveis pelo comportamento elástico de alguns tipos de tecido, como os feitos de elastano. Estas molas podem assumir diferentes estruturas de ligação entre as partículas, como, por exemplo, nas direções horizontal, vertical e diagonal, e se deformam dentro e fora do plano da superfície criando tensões, distensões, cisalhamentos e curvamentos que são calculados de acordo com a mecânica clássica, sendo o material do tecido aproximado pela rigidez das molas. As deformações das molas são então usadas para estimar a energia interna acumulada em cada partícula da superfície. Normalmente, a energia resultante de uma partícula é calculada pelas deformações das molas adjacentes a ela ou dentro de uma vizinhança pré-estabelecida.

Já os modelos baseados na mecânica de contínuos consideram o tecido como uma superfície deformável contínua sobre a qual são aplicadas teorias das cascas ou das placas para analisar os comportamentos de tensão, distensão, cisalhamento e curvamento nas deformações de um tecido. Estas deformações podem ser descritas como a variação da energia interna em relação aos elementos da geometria diferencial, como os tensores de métrica e de curvatura. A variação da energia interna de um ponto da superfície do tecido pode ser então obtida em função da variação dos coeficientes da primeira e da segunda forma fundamental em relação ao estado não deformado do tecido.

Os modelos baseados na mecânica de contínuos precisam ser discretizados no domínio espacial para serem computacionalmente implementáveis, de forma que ambas as abordagens de modelagem física de partículas e contínuos podem ter formulações similares, diferindo essencialmente nas equações constitutivas, as quais são responsáveis por descrever as forças internas $\mathbf{f}(p, t)$ causadas por deformações no tecido. As equações constitutivas são relações entre duas quantidades físicas, como a lei de Hooke, que relaciona a força elástica de forma diretamente proporcional ao produto do coeficiente de rigidez de uma mola pelo deslocamento sofrido por ela.

Existem três metas relacionadas à simulação de tecidos que são colocadas em primeiro plano dependendo da finalidade da aplicação: o **realismo**, em animações, como filmes, desenhos e propagandas; a **eficiência**, relevante nos jogos em geral, pois de nada adianta um jogo realista se sua interatividade ficar prejudicada por algum método custoso que, na prática, serve apenas para lhe dar um maior adereço; e a **precisão física**, importante para análise e simulação do comportamento de tecidos nas aplicações da Engenharia Têxtil. Neste sentido, não existe uma única forma correta de simular tecidos, pois a escolha de qual modelo de tecido usar depende das necessidades do projetista.

O realismo na dinâmica de tecidos está relacionado com o uso de modelos físicos. A mecânica de contínuos consegue atingir uma precisão física superior à mecânica de partículas, pois se baseia, em geral, em um modelo geometricamente exato considerando o tecido da forma mais real possível, ao invés de imaginá-lo com uma grade conectada por molas. Entretanto, esta maior precisão física resulta em uma formulação mais complexa, o que torna o modelo de tecido mais difícil de ser implementado e também menos eficiente computacionalmente em relação à mecânica de partículas. Atualmente, os modelos baseados na mecânica de contínuos são reconhecidos por melhor representarem as deformações dos tecidos, porém na prática os modelos baseados na mecânica de partículas são os mais utilizados pela simplicidade na formulação das forças internas e pela eficiência.

Em 2000, Breen e House editaram um livro que compara em um de seus capítulos duas implementações com abordagens físicas distintas [23]. Nesta comparação foi utilizado o modelo de partículas de Breen et al. [17] e o modelo baseado na mecânica de contínuos de Eischen et al. [16]. A conclusão foi de que quando uma alta precisão é requerida, um modelo baseado na mecânica de contínuos é a melhor opção, enquanto em termos de eficiência computacional é recomendado um modelo baseado na mecânica de partículas.

Dessa forma, esta dissertação considera avaliar a utilização de um modelo baseado na mecânica de contínuos com uma implementação alternativa para simular tecidos que agregue realismo e precisão física junto com um comportamento linear na complexidade computacional em termos de desempenho. Para tanto, é necessário discretizar espacialmente o modelo baseado na mecânica de contínuos para que esta implementação fique clara.

A discretização espacial em modelos baseados na mecânica de partículas tem o objetivo de reduzir as equações diferenciais parciais, presentes na formulação do modelo, para equações diferenciais ordinárias em função do tempo. Estas equações são então solucionadas com um método numérico para obter diferentes estados do tecido ao longo do tempo, produzindo sua dinâmica. O processo numérico de encontrar diferentes formas seqüenciais do tecido ao longo do tempo é muitas vezes referenciado de integração temporal. A discretização espacial pode ser feita utilizando os métodos de diferenças finitas ou elementos finitos. O método de diferenças finitas possui uma formulação simples e com bom desempenho computacional, porém necessita do tratamento de problemas de

contorno, que ocorrem quando não existem pontos adjacentes em determinadas direções nas fronteiras do tecido. Já o método dos elementos finitos não possui problemas de contorno, porém apresenta uma formulação mais complexa e demanda maior esforço computacional.

O método das diferenças finitas foi o escolhido para discretizar espacialmente o modelo elástico proposto por Terzopoulos et al., o qual era baseado na mecânica de contínuos e visava simular o comportamento de modelos elásticos deformáveis, incluindo os tecidos [19]. Inspirados nesta discretização, Wu e de Melo aplicaram um esquema similar para discretizar um modelo fundamentado na teoria de superfície de Cosserat, também baseado na mecânica de contínuos e formulado para simular a dinâmica de tecidos [3]. Já os elementos finitos foi a opção de Eischen et al., que utilizaram um modelo de casca [18] para o mesmo propósito [16]. Além deles, Etmuss et al. usaram os elementos finitos mostrando que é possível obter eficiência computacional com esta técnica na simulação de tecidos [9]. Em um outro trabalho, Etmuss et al. demonstraram que é possível derivar uma formulação baseada na mecânica de partículas a partir de uma baseada na mecânica de contínuos utilizando um esquema de diferenças finitas [24]. Com isso, mesmo utilizando um modelo baseado na mecânica de contínuos para representar um tecido, pode-se obter uma formulação similar a um modelo baseado na mecânica de partículas para obter-se maior eficiência computacional.

Portanto, visando explorar a alternativa de usar um modelo baseado na mecânica de contínuos para produzir a simulação de tecidos de forma eficiente, é interessante utilizar um modelo discretizado com as diferenças finitas para obter uma formulação similar àquela usada na mecânica de partículas. Neste sentido, foi selecionado o modelo proposto por Wu e de Melo, o qual, fundamentado na teoria de superfície de Cosserat, consegue reproduzir o comportamento natural de dobras que se formam nos tecidos sob apenas forças de compressão [3]. Este comportamento ocorre porque, diferentemente dos outros modelos, os autores incluíram um termo que relaciona as variações de métrica com as de curvatura, preservando a compatibilidade entre estas duas grandezas geométricas e assegurando a regularidade da superfície. Além disso, este modelo tem um controle intuitivo dos parâmetros do material do tecido.

1.1.2 Integração Temporal

Apesar das diferentes abordagens de modelagem física de tecidos, elas procuram solucionar o mesmo problema, que é encontrar de forma precisa e estável as novas formas do tecido, ao longo do tempo, quando sujeito a forças. Este processo, chamado de integração temporal, recai na solução da Equação 1.1, que a partir de um instante t é usada para obter os pontos da superfície em $t + \Delta t$, sendo Δt chamado de passo de tempo. Existem muitos esquemas para solucionar esta equação, diferindo em características como estabilidade numérica, eficiência e precisão. Basicamente, estes esquemas podem ser divididos em três tipos de integração temporal: implícita, semi-implícita e explícita.

De forma similar à modelagem de tecidos, a escolha do método de integração temporal também depende da finalidade da aplicação. Para uma aplicação que priorize o realismo e a precisão física, os métodos implícitos combinados com modelos contínuos são os mais adequados. Já para uma aplicação em tempo real, como nos jogos modernos, que precisam de um melhor desempenho, uma integração explícita junto com um modelo de partículas é o mais indicado. Naturalmente, à medida que o passo de tempo Δt se aproxima de 0, as soluções aproximadas dos novos pontos da superfície do tecido em cada instante de tempo $t + \Delta t$ tendem a ficar mais precisas, para qualquer método de integração temporal, devido à proximidade entre os valores dos elementos do modelo de tecido entre passos subsequentes. Contudo, conforme o passo de tempo cresce, é possível notar a diferença de precisão e estabilidade numérica que cada tipo de integração consegue atingir.

A integração temporal implícita é a que obtém a solução aproximada com maior precisão numérica em relação à solução exata, pois nela os novos pontos da superfície do tecido são encontrados no instante $t + \Delta t$ com base nas forças do mesmo instante $t + \Delta t$. Além disso, este tipo de integração tem a vantagem de conseguir encontrar as soluções de forma estável com um passo de tempo maior do que os utilizados por esquemas semi-implícitos ou explícitos, embora tenha a desvantagem de precisar linearizar o modelo de tecido para torná-las computacionalmente solúveis.

Já na integração temporal explícita os novos pontos da superfície são encontrados em $t + \Delta t$ com base na energia interna armazenada no instante t , gerando um erro de aproximação proporcional ao tamanho do passo de tempo Δt , de forma que quanto menor for este valor, menor será o erro cometido. Por utilizar a energia interna do instante anterior t para encontrar as novas geometrias do tecido em $t + \Delta t$, esta técnica consiste essencialmente em linearizar o problema. Ela é menos precisa e também apresenta mais problemas de estabilidade numérica, principalmente para valores grandes de passo de tempo, pois depende muito da suavidade com que as energias variam entre dois instantes seguidos. Entretanto, este é o tipo de integração mais fácil de ser implementado e o que apresenta a melhor eficiência computacional.

Por último, a integração temporal semi-implícita é uma mistura dos dois esquemas, equacionando os novos pontos da superfície no instante $t + \Delta t$ com base em uma parcela de energia interna deste mesmo instante, que precisa ser formulada em função dos novos pontos, e também usando uma parcela de energia do instante anterior t . Desta forma, o esquema herda vantagens e desvantagens dos dois lados, mas é uma boa opção, por balancear as características relacionadas a uma integração temporal.

Na literatura muito já foi pesquisado sobre as integrações temporais para a simulação de tecidos. Os métodos de integração explícita dominaram os trabalhos da década de 90 [2], até que, em 1998, Baraff e Witkin demonstraram a superioridade do esquema numérico de integração implícita em comparação à explícita, atingindo uma dinâmica de tecido estável com o passo de tempo mais largo [14].

Para aliviar o alto custo computacional deste esquema, os autores propuseram um método do gradiente conjugado modificado. Os resultados foram tão promissores que desde então uma série de trabalhos tem se dedicado a aprimorar este sistema [25, 11]. Contudo, estava claro que a prioridade deste trabalho era a eficiência computacional e a estabilidade, pois as imagens obtidas, principalmente em relação à formação de dobras sob forças de compressão, não possuíam grande realismo. Segundo Choi e Ko, o realismo deste modelo era prejudicado pela adição de forças artificiais fictícias para solucionar colisões, o que evitava ou mesmo eliminava a formação de dobras [12]. No modelo proposto por Wu e de Melo não há este tipo de problema, pois a compatibilidade geométrica que assegura um comportamento natural das dobras em um tecido está implícita no modelo [3, 5]. Entretanto, o método semi-implícito de integração temporal usado por estes autores é pouco eficiente devido à sua complexidade computacional. Este baixo desempenho está relacionado com a dificuldade em linearizar a Equação 1.1 para solucioná-la como um sistema de equações lineares.

Assim, esta dissertação propõe avaliar as técnicas de integração explícita para solucionar a Equação 1.1 com o objetivo de contornar o problema de linearização das equações constitutivas sem sacrificar o realismo dos resultados visuais.

1.1.3 Problemas

Um dos problemas relacionados ao uso de uma integração temporal explícita refere-se à imposição em usar um passo de tempo pequeno. Esta imposição ocorre devido ao fato de se estimar as posições dos novos pontos da superfície do tecido em um instante $t + \Delta t$ usando as energias armazenadas do tempo t . Quanto menor for este passo de tempo, mais suave é a variação das energias acumuladas na superfície entre tempos subseqüentes e, conseqüentemente, menor é o erro de aproximação cometido. Normalmente, o passo de tempo de uma integração explícita gira em torno de $0.001s$. Esta limitação implica em outro problema, que diz respeito ao número de quadros que precisam ser gerados para uma animação com o intuito de atingir um determinado estado final. Assim, para se simular 10 segundos de uma simulação de tecidos que utiliza uma integração explícita, usando um passo de tempo de $0.001s$, são necessárias 10000 iterações, sendo cada uma delas correspondente a um novo estado do tecido, ao passo que uma integração implícita precisaria de apenas 1000 iterações a um passo de tempo de $0.01s$ para simular o mesmo tempo total. Contudo, cada iteração de uma integração explícita, em geral, costuma ser computada mais rapidamente do que uma implícita.

Devido à existência de diversos métodos de integração explícita, os quais diferem em alguns aspectos como a precisão numérica e o número de passos usados na solução de cada iteração, é possível que exista diferença nos valores máximos do passo de tempo, bem como nos resultados visuais, quando aplicados para simular tecidos. Os métodos numéricos de integração explícita são usados, normalmente, para determinar o valor de um ponto aplicado a uma função sem saber sua

formulação algébrica, porém conhecendo a expressão de sua derivada e ainda um valor inicial de um ponto da função. A maioria destes métodos baseia-se na expansão de série de Taylor, desprezando termos conforme a precisão que se deseja obter. No caso da simulação de tecidos, particularmente no modelo de tecido proposto por Wu e de Melo, não se conhece a definição das derivadas, que em relação à posição dos pontos correspondem a sua velocidade e aceleração, dadas pelo cálculo de forças em uma iteração do modelo [5]. Desta forma, os métodos explícitos precisam ser ajustados ao modelo antes de serem aplicados ao longo do tempo para obter diferentes geometrias do tecido.

Um outro problema relacionado à simulação de tecidos é a definição das condições de contorno, quando se utiliza a técnica de diferenças finitas para a discretização espacial do modelo de tecido, como em [19, 5]. Este problema consiste no cálculo de alguns elementos do modelo, como tensores de métrica e curvatura, nas fronteiras da malha discreta, nas quais pontos adjacentes necessários para os cálculos podem não existir. Na implementação do modelo de tecido selecionado é utilizada a mesma estratégia usada por Terzopoulos et al., que corresponde em zerar os elementos nos pontos em que não existirem os pontos adjacentes necessários [19]. Tal consideração pode provocar, em alguns tipos de simulação, um desequilíbrio das forças internas nos pontos zerados, causando problemas de instabilidade. Esta dissertação apresenta uma solução para contornar este problema.

1.2 Conteúdo do Trabalho

Este trabalho tem como principal conteúdo explorar o potencial da linearização do modelo da dinâmica de tecidos, proposto por Wu e de Melo [3], com o intuito de aplicar técnicas de integração explícita para obter as variações na sua geometria ao longo do tempo.

1.2.1 Motivação

A motivação deste trabalho é desenvolver uma simulação de tecidos utilizando uma abordagem alternativa, em relação aos trabalhos atuais, que consiste em combinar um modelo de tecido baseado na mecânica de contínuos com uma integração temporal explícita. Para tanto, é usado um modelo de superfícies deformáveis geometricamente exato, proposto por Wu e de Melo [3], que, derivado de uma teoria de cascas [4], se destaca dos demais modelos baseados na mecânica de contínuos por possuir um tratamento apropriado na formação de dobras sob forças de compressão. Neste modelo de tecido, o realismo dos resultados é obtido com precisão física, o que proporciona um controle intuitivo dos parâmetros do material do tecido. Contudo, em sua implementação original o modelo é integrado ao longo do tempo com um esquema semi-implícito, que possui um alto custo computacional. Assim, existe o interesse em verificar se este modelo pode ser implementado com um método de integração

de melhor desempenho, porém sem perder realismo nos resultados visuais.

1.2.2 Hipótese

A natureza do modelo de tecido escolhido impõe o processo de linearização das equações envolvidas para poderem ser resolvidas por técnicas implícitas. No entanto, este processo de linearização não é trivial em vista das grandezas envolvidas. Aparentemente, isso poderia ser contornado adotando métodos semi-implícitos, em que tais grandezas sejam aproximadas pelos valores obtidos na iteração anterior. Adotando métodos semi-implícitos, é necessário o uso de pequenos passos de tempo para garantir que a variação desses elementos geométricos ao longo do tempo seja suave. Este trabalho parte da hipótese de que, diferentemente da tendência dos trabalhos em curso, uma integração temporal explícita pode solucionar eficientemente a Equação 1.1, gerando resultados realistas, desde que a variação entre as formas geométricas de dois passos de tempos subseqüentes seja suave. É esperado que o realismo esteja inerente ao modelo proposto por de Wu e de Melo [3], enquanto o uso de um método de integração explícita proporcione um ganho de simplicidade e eficiência em relação à integração semi-implícita feita na implementação original do modelo.

1.2.3 Objetivos

O principal objetivo deste trabalho é melhorar a eficiência computacional da solução da Equação 1.1 com a força interna $f(p, t)$ sendo estimado pelo algoritmo proposto no modelo de Wu e de Melo, investigando qual a melhor técnica de integração explícita que pode ser aplicada para integrá-lo ao longo do tempo. Como objetivo secundário, pretende-se apresentar uma validação prática mais aprofundada das principais características do modelo, mostrando através de resultados visuais de simulações de tecidos, geradas com o melhor método explícito avaliado, sua eficiência na criação de dobras com diversidade e em situações em que só existem forças de compressão.

Além disso, esta dissertação tem o objetivo de apresentar um simulador de tecidos com uma nova reestruturação da implementação feita por de Melo [5], com o intuito de tornar os códigos mais modulares para facilitar a integração de novos algoritmos, como os referentes ao tratamento de colisões. Neste *software* também é desenvolvida uma nova interface gráfica visando aumentar a facilidade do usuário para manipular os parâmetros do modelo e gerar animações.

1.2.4 Escopo

Faz parte do escopo deste trabalho a análise da adequabilidade de técnicas de integração explícita para solucionar a Equação 1.1 no modelo proposto por Wu e de Melo [3] em termos de: expectativa

visual, estabilidade numérica e eficiência computacional. Para tanto, assume-se que a topologia da superfície que representa o tecido é uma malha retangular, pois se adota o mesmo esquema de discretização espacial proposta por Terzopoulos et al. [19] para determinar as variações das equações constitutivas do modelo deformável. Está fora do escopo desta dissertação incorporar algoritmos de tratamento de colisões à simulação de tecidos, embora eles sejam considerados na modularização do código-fonte para que possam ser agregados facilmente em trabalhos futuros.

1.2.5 Contribuições

A principal contribuição desta dissertação é desenvolver uma implementação, eficiente computacionalmente, do modelo de tecido proposto por Wu e de Melo [3]. Para este propósito, as técnicas de integração explícita mais populares são implementadas para serem usadas como solução numérica do modelo de tecido ao longo do tempo, o qual é baseado na mecânica de contínuos. Assim, apesar do uso de métodos explícitos para simular tecidos não ser uma novidade, bem como utilizar o modelo de tecido escolhido para este mesmo propósito, a combinação destas técnicas de integração com este tipo de modelo é inédita na literatura.

Esta dissertação também contribui fornecendo um procedimento para especificar os parâmetros do material de um tecido, bem como os parâmetros de simulação, que juntos, determinam uma grande diversidade de dobras que podem ser formadas. Para tanto, um projeto de *software* é apresentado visando desenvolver uma nova arquitetura para o modelo de tecido usado, criando uma interface lógica, com o intuito de facilitar a integração de novos algoritmos, e também uma interface gráfica, objetivando aumentar a usabilidade do usuário para produzir simulações de tecidos. Além disso, esta dissertação contribui com uma validação prática mais extensa do modelo, destacando as suas principais características através dos resultados visuais.

1.3 Organização da Dissertação

Esta dissertação está organizada em seis capítulos, os quais, com exceção desta introdução, são brevemente descritos nos parágrafos abaixo.

Capítulo 2 discute brevemente os trabalhos relacionados com a simulação de tecidos, destacando suas abordagens de modelagem e suas integrações temporais. Ao final deste capítulo, o modelo de superfície deformável usado neste trabalho é explicado de forma detalhada.

Capítulo 3 apresenta a discretização espacial do modelo de tecido, selecionado neste trabalho, para descrever de que forma os métodos explícitos mais conhecidos podem ser aplicados sobre ele.

Após mostrar as formulações, vantagens e desvantagens, estes métodos são implementados sobre o modelo e seus resultados analisados. A escolha do método de Verlet perante os demais é justificada em termos de eficiência computacional, estabilidade numérica e qualidade das imagens produzidas.

Capítulo 4 descreve o projeto de *software* de um simulador de tecidos, provido de uma interface lógica e gráfica. Ao final, alguns exemplos de simulações de tecidos são apresentados para mostrar sua usabilidade.

Capítulo 5 mostra os resultados visuais obtidos com o simulador desenvolvido, focando a qualidade e a diversidade das dobras produzidas, bem como a precisão física do modelo de tecidos. Este capítulo mostra que a abordagem proposta de simular tecidos é viável, além de permitir uma validação prática mais aprofundada do modelo, constatando sua superioridade na produção de dobras em relação aos outros modelos existentes.

Capítulo 6 resume as principais conclusões desta dissertação, destacando as suas contribuições e limitações. Por fim, as possibilidades de trabalhos futuros são comentadas.

Capítulo 2

Simulações de Tecidos

Este capítulo apresenta um panorama dos trabalhos atuais relacionados à simulação de tecidos, destacando neles a combinação da modelagem de tecidos com a solução numérica adotada, com o objetivo de contextualizar a proposta desta dissertação em uma lacuna existente na literatura. Como este trabalho é voltado para uma nova abordagem de simular a dinâmica de tecidos, neste capítulo são discutidos apenas modelos físicos, tanto os baseados na mecânica de contínuos quanto na de partículas, deixando de fora os modelos puramente geométricos, como [21, 22], por eles serem voltados para a simulação de tecidos em um estado estático. Para melhor organizar a discussão dos modelos físicos, a Seção 2.1, que descreve os modelos baseados na mecânica de partículas, e a Seção 2.2, que descreve os baseados na mecânica de contínuos, estão divididas de acordo com a solução numérica usada para integrar temporalmente os modelos, ou seja, com subseções referentes às integrações implícitas, semi-implícitas e explícitas. Após os principais modelos serem comentados, eles são sintetizados e classificados de acordo com a abordagem física e o método numérico usado na Seção 2.3.

Em seguida, este capítulo justifica a escolha e detalha o modelo físico usado neste trabalho, o qual é proposto por Wu e de Melo, baseado na mecânica de contínuos e discretizado no espaço com diferenças finitas [3]. Este modelo de superfície deformável é fundamentado em uma teoria de casca geometricamente exata, conhecida como superfície de Cosserat elástica [4]. O resultado é um comportamento apropriado do tecido quando submetido às forças de compressão e um controle intuitivo dos parâmetros do material por possuírem interpretações geométricas, que são aspectos diferenciais deste modelo em relação aos demais baseados na mecânica de contínuos. Desta forma, uma explicação aprofundada deste modelo é feita na Seção 2.4, dividindo-a em subseções que discutem a teoria geral da superfície de Cosserat e as restrições feitas sobre ela no modelo de tecido usado nesta dissertação. Por fim, a Seção 2.5 traz as considerações finais.

2.1 Modelos Físicos baseados na Mecânica de Partículas

Os modelos físicos que utilizam a mecânica de partículas representam os tecidos como uma grade de pontos interligados entre si com diferentes tipos de ligação. Conforme estes pontos ou partículas vão se deslocando devido a deformações, as forças exercidas sobre cada partícula são calculadas com base nas tensões existentes entre a ligação com as partículas adjacentes. O modelo mais conhecido é o modelo massa-mola, no qual cada partícula está ligada à outra por molas e as tensões são dadas pela lei de Hooke, que diz que a força elástica é igual ao produto entre o coeficiente de rigidez da mola e seu deslocamento a partir do ponto de repouso.

2.1.1 Integrações Explícitas

O trabalho precursor da mecânica de partículas é o trabalho de Haumann, que em 1987 desenvolveu o primeiro modelo massa-mola para modelar a dinâmica de uma superfície deformável, utilizando uma integração temporal explícita [1]. Neste modelo cada vértice da superfície possui uma massa e é conectado ao seu vizinho por uma mola, de forma que a cada duas faces triangulares, formadas por quatros pontos-massa, existe um outro tipo de ligação de mola, conforme a Figura 2.1a. Incorporando forças aerodinâmicas como o vento e a resistência ao ar e, um algoritmo de detecção de colisões, o autor utilizou como solução numérica o esquema explícito de Euler para frente.

Em 1995, Provot incrementou o sistema de Haumann propondo o uso de três tipos de molas que conectam os pontos-massa da grade bidimensional: molas de estrutura, as quais são usadas entre todos os vizinhos imediatos nas direções horizontais e verticais; molas de flexão, usadas entre vizinhos espaçados de um ponto de distância também nas direções horizontais e verticais; e molas de cisalhamento, usadas entre vizinhos imediatos nas diagonais [2]. Assim, cada ponto-massa da grade de partículas pode ter sua tensão resultante estimada com pontos em uma vizinhança pré-estabelecida, como mostra a Figura 2.1(b). Este trabalho conseguiu solucionar alguns dos principais problemas da época, que eram os efeitos de superelasticidade no tecido e instabilidade na simulação. O esquema explícito de Euler para frente foi aplicado para obter as soluções numéricas ao longo do tempo, partindo da segunda lei de Newton, que diz que a força resultante em um corpo é igual ao produto de sua massa pela aceleração adquirida, e utilizando as forças elásticas provocadas pelos deslocamentos das molas para encontrar a força resultante \mathbf{F} e, com ela, encontrar a aceleração (\mathbf{acel}) e atualizar a velocidade (\mathbf{vel}) e posição (\mathbf{r}) de cada partícula (k, l) , conforme as equações

$$\begin{aligned}\mathbf{acel}_{k,l}(t + \Delta t) &= \mathbf{F}_{k,l}(t), \\ \mathbf{vel}_{k,l}(t + \Delta t) &= \mathbf{vel}_{k,l}(t) + \Delta t \mathbf{acel}_{k,l}(t + \Delta t), \\ \mathbf{r}_{k,l}(t + \Delta t) &= \mathbf{r}_{k,l}(t) + \Delta t \mathbf{vel}_{k,l}(t + \Delta t).\end{aligned}$$

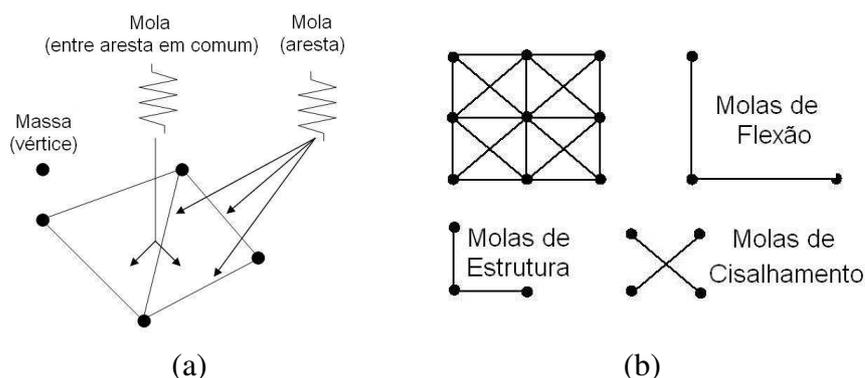


Figura 2.1: Modelos massa-mola: (a) Haumann em 1987 [1]; e (b) Provot em 1995 [2].

2.1.2 Integrações Implícitas

Também usando a mecânica de partículas, mas sem considerar a abordagem massa-mola, Breen et al. (1992) consideraram que o comportamento macroscópico do tecido é obtido modelando a sua estrutura microscópica e desenvolveram um modelo para o caimento de tecidos, representando as interações de seus fios em um reticulado de pontos tridimensionais [17]. Esta abordagem considera que um tecido seja melhor representado por um mecanismo de interação entre as partes mecânicas do que por uma substância contínua [23]. As conexões mecânicas entre as partículas são representadas como funções de energia gerada por distintas deformações, mais especificamente a energia de distensão $E_{stretch}$, de cisalhamento E_{shear} e de curvamento $E_{bending}$, de forma que a energia acumulada resultante E_R em cada partícula da grade é obtida por

$$E_R = E_{stretch} + E_{shear} + E_{bending}.$$

Cada uma destas parcelas de energia é calculada com funções que medem o deslocamento de um ponto com base em expressões angulares. Como os autores observaram que um simples ponto pode curvar-se para fora do plano da superfície do tecido, eles propuseram descrever este fenômeno pelo ângulo formado entre cada conjunto de três partículas cruzadas adjacentes [17]. Este modelo focava a configuração estável de tecidos, após o caimento sobre alguns objetos, que era alcançada com uma técnica de gradiente estocástico descendente, usada para relaxar as partículas até atingirem o estado de equilíbrio. Para o tratamento de colisões, eles consideraram uma parcela adicional E_{repel} , correspondente à força de repulsão, na composição da energia E_R para assegurar uma distância mínima entre as partículas do tecido. Em 1994, Breen et al. mostraram que o modelo pode ser usado para reproduzir de maneira precisa o caimento de diferentes materiais, porém sem produzir respostas dinâmicas necessárias para uma animação [26]. Em um trabalho de Breen e House em 2000, eles recomendaram

usar uma integração temporal implícita para estender o modelo para este propósito [23].

2.1.3 Integrações Semi-implícitas

Os principais trabalhos que unem a mecânica de partículas com uma integração semi-implícita foram publicados nesta última década. Em 2002, Choi e Ko [12] propuseram usar uma solução com um esquema semi-implícito, em conjunto com um modelo físico baseado na mecânica de partículas, inspirado no modelo de tecido de Breen et al. (1992) [17]. Eles usaram uma fórmula de diferenças para trás de segunda ordem, que define as novas posições \mathbf{r} dos pontos da superfície do tecido como

$$\mathbf{r}^{t+\Delta t} = \frac{1}{\Delta t} \left(\frac{3}{2}\mathbf{r}^{t+\Delta t} - 2\mathbf{r}^t + \frac{1}{2}\mathbf{r}^{t-\Delta t} \right).$$

Para solucionar esta equação os autores usaram um método de gradiente conjugado pré-condicionado. Contudo, Choi e Ko constataram que somente o emprego de métodos semi-implícitos não garante a estabilidade numérica do modelo em situações que ocorrem deformações de *buckling*, que corresponde ao ato de enrugar, amassar ou criar dobras irregulares em um objeto deformável.

A instabilidade gerada após situações de *buckling* é muito comum em modelos massa-mola, pois quando acontecem grandes deformações para fora do plano da superfície do tecido, as molas que formam sua estrutura têm que ser suficientemente rígidas para não haver grandes deformações de métrica, mas flexíveis para a mudança de curvatura. Baraff e Witkin introduziram forças fictícias para contornar este problema [14], as quais muitas vezes acabavam eliminando qualquer tipo de dobra mais acentuada que pudesse ser formada, diminuindo o realismo da simulação [12]. A principal contribuição do modelo físico de Choi e Ko foi propor uma técnica usada no pós-processamento de *bucklings* sem recorrer a nenhuma força dissipadora fictícia, aumentando a estabilidade e o realismo da simulação [12]. Eles mostraram seus resultados com vestuários sobre atores virtuais.

Também em 2002, Bridson et al. propuseram algumas técnicas robustas para tratar colisões, contato e fricção em simulações de tecidos, de forma a garantir a formação de dobras nos vesturários [25]. Estes autores utilizaram o modelo massa-mola de Provot [2], enfatizando que o foco deste trabalho era o controle robusto do tratamento de colisões, e não a modelagem do tecido. Para a discretização do tempo, eles destacaram a separação da evolução do tempo na dinâmica interna do tecido (e do ambiente ao seu redor) em relação ao algoritmo que processa as colisões. Pode-se classificar a solução numérica adotada neste trabalho como semi-explícita, pois o cálculo das novas posições é feito com um método explícito, enquanto a velocidade é calculada com uma equação implícita, similar àquela usada por Baraff e Witkin (1998) em [14].

No ano seguinte, Bridson et al. publicaram outro trabalho, incrementando o modelo de tecido anterior com várias considerações da mecânica de contínuos e expressões angulares para o cálculo

das forças internas, procurando preservar as dobras para deixar o modelo mais realista [11]. Eles também incrementaram o tratamento de colisões, solucionando problemas na formação de dobras e rugas que interpenetram a superfície do tecido com o objeto de colisão, como um personagem virtual. Neste último trabalho, os autores destacam a solução numérica adotada, que classificaram como semi-implícita e detalham o tipo de cada passo do procedimento, descrito a seguir.

- Explícito: $\text{vel}^{t+\frac{\Delta t}{2}} = \text{vel}^t + \frac{\Delta t}{2} \text{acel}^t$,
- Modificar $\text{vel}^{t+\frac{\Delta t}{2}}$ respeitando o algoritmo que processa as colisões,
- Explícito: $\mathbf{r}^{t+\Delta t} = \mathbf{r}^t + \Delta t \text{vel}^{t+\frac{\Delta t}{2}}$ (método *leapfrog*),
- Implícito: $\text{vel}^{t+\Delta t} = \text{vel}^{t+\frac{\Delta t}{2}} + \frac{\Delta t}{2} \text{acel}^{t+\Delta t}$ (regra trapezoidal),
- Modificar $\text{vel}^{t+\Delta t}$ respeitando os limites que o algoritmo que processa as colisões impõe.

Assim, os autores agregam um esquema explícito chamado de *leapfrog*, que possui precisão de segunda ordem, para calcular as novas posições dos pontos da superfície, junto com um método implícito de regra trapezoidal, também de precisão de segunda ordem, para atualizar as velocidades.

2.2 Modelos Físicos baseados na Mecânica de Contínuos

Os modelos físicos baseados na mecânica de partículas não conseguem realmente refletir a forma da superfície na vizinhança de um ponto, comprometendo a precisão das forças internas. Breen et al. afirmaram, após apresentarem resultados comparativos entre implementações usando uma abordagem de partículas em relação a uma de contínuos, que quando o realismo é colocado em primeiro plano os modelos contínuos são mais apropriados, enquanto que, em termos de eficiência, os modelos de partículas são mais adequados [23]. Esta afirmação é enfatizada no trabalho de Bridson et al., em que os autores descreveram seu trabalho focado no tratamento de colisões com um modelo massa-mola porque o foco não era o realismo, mas indicaram que imagens da indústria cinematográfica, mais precisamente o quimono do personagem Yoda do filme Guerra nas Estrelas, foi criado com suas técnicas de colisões em adição a um modelo físico contínuo [25].

2.2.1 Integrações Implícitas

Um dos primeiros trabalhos da mecânica de contínuos pertence a Feynman, no qual ele simulava propriedades mecânicas dos tecidos definindo um conjunto de funções de energia sobre um reticulado de pontos tridimensionais [20]. A energia total de seu modelo considerava deformações de métrica, de curvatura e termos relativos à gravidade. Então, as forças internas da superfície deformável eram

calculadas minimizando as energias de tensão e de curvamento produzidas pelas deformações. Este autor utilizava a integração temporal implícita com um método de relaxação.

Em 1998, Baraff e Witkin apresentaram um trabalho demonstrando a superioridade do esquema numérico implícito em relação ao explícito para integrar temporalmente um modelo de tecido com passos de tempo largos [14]. Estes autores descreveram as forças internas que atuam sobre a superfície deformável com expressões derivadas da mecânica de contínuos junto com expressões angulares. O resultado estava focado na eficiência e na estabilidade numérica em utilizar um método de integração implícita com o passo de tempo em torno de uma ordem maior que o usado pelos métodos explícitos. A solução era obtida montando um sistema simétrico de equações que era resolvido por um método modificado dos gradientes conjugados. Segundo Baraff e Witkin, a complexidade computacional do seu esquema implícito é de $O(n^{1.5})$. Choi e Ko constataram que o realismo deste modelo era prejudicado pela adição de forças artificiais fictícias usadas para tratar colisões, as quais evitavam ou mesmo eliminavam a formação de dobras [12]. Mais tarde, em 2003, Baraff et al. usou o mesmo modelo de tecido e integração temporal implícita para contribuir com técnicas novas de tratamento de colisões e auto-colisões que ocorrem quando o objeto virtual em contato com o tecido tem pontos que se interpenetram [10]. Posteriormente, neste mesmo ano, Bridson et al. apresentaram outra solução para o mesmo problema [11].

2.2.2 Integrações Semi-implícitas

Um ano depois do trabalho de Feynman [20], Terzopoulos et al. propuseram um modelo de superfície deformável baseado na teoria da elasticidade e fundamentado na mecânica de contínuos [19]. Os autores descreveram uma formulação sem o objetivo de simular precisamente um determinado tipo de objeto deformável, propondo um sistema físico capaz de produzir com qualidade alguns tipos de comportamento sem possuir uma formulação muito complexa. Eles estimavam a força elástica do modelo através das variações da energia potencial, que eram calculadas pela variação dos tensores métricos e de curvatura do estado do objeto deformado em relação ao estado não deformado. Neste trabalho foi utilizada uma integração temporal semi-implícita, em que a solução da posição $\mathbf{r}_{t+\Delta t}$ dos pontos da superfície no próximo instante $t + \Delta t$ era formulada como

$$\mathbf{A}_t \mathbf{r}_{t+\Delta t} = \mathbf{g}_t, \quad (2.1)$$

com a matriz \mathbf{A}_t definida por

$$\mathbf{A}_t = \mathbf{K}_{t+\Delta t} + \left(\frac{1}{\Delta t^2} \mathbf{M} + \frac{1}{2\Delta t} \mathbf{C} \right),$$

em que $\mathbf{K}_{t+\Delta t}$ é a matriz de rigidez do material, \mathbf{M} é a matriz contendo as densidades de massa de cada ponto, \mathbf{C} a matriz contendo os coeficientes de amortecimento do meio e o vetor \mathbf{g} definido por

$$\mathbf{g}_t = \mathbf{F}_t + \left(\frac{1}{\Delta t^2} \mathbf{M} + \frac{1}{2\Delta t} \mathbf{C} \right) \mathbf{r}_t + \left(\frac{1}{\Delta t} \mathbf{M} + \frac{1}{2\Delta t} \mathbf{C} \right) \mathbf{vel}_t,$$

sendo \mathbf{F}_t o vetor de forças externas no instante t e o vetor de velocidades

$$\mathbf{vel}_t = \frac{\mathbf{r}_t - \mathbf{r}_{t-\Delta t}}{\Delta t}.$$

Dessa forma, o esquema adotado é classificado como semi-implícito porque os termos da matriz $\mathbf{K}_{t+\Delta t}$, usados para obter \mathbf{A}_t , são formulados em função das novas posições $\mathbf{r}_{t+\Delta t}$ do instante $t + \Delta t$, e o vetor \mathbf{g}_t , do lado direito da Equação 2.1, é obtido com termos do instante t . Tanto a matriz \mathbf{A} , quanto a matriz \mathbf{K} são de ordem $mn \times mn$, em que m e n são os números de colunas e linhas, respectivamente, da malha discreta que representa a superfície do tecido. Terzopoulos et al. resolveram o sistema expresso na Equação 2.1 com métodos diretos, como a decomposição de Choleski e métodos de relaxação, como o de Gauss-Seidel [19]. O custo computacional deste procedimento de integração semi-implícita é aproximadamente da ordem de $O((mn)^2)$, devido às dimensões das matrizes que formam o sistema linear.

Um dos grandes problemas dos modelos baseados em mecânica de contínuos para modelar o comportamento de tecidos deve-se à dificuldade em definir a diversidade de dobras que podem ser formadas. Enquanto as deformações de métrica dentro do plano da superfície do tecido são bem estimadas com a variação do tensor métrico, as deformações de curvatura na direção para fora do plano da superfície ainda não atingiram o mesmo consenso. Isto ocorre porque existem deformações de curvatura nos tecidos que podem ser provocadas somente por forças tangenciais, como em dobras que se formam ao esticar ou comprimir um tecido somente em uma direção. Assim, a ligação entre as medidas de métrica e curvatura é essencial na modelagem de tecidos. A teoria da superfície de Cosserat, a qual é fundamentada na mecânica de contínuos, possui esta ligação representada pelo produto entre as variações de métrica e de curvatura [4].

Utilizar uma teoria de casca como a superfície de Cosserat para modelagem de tecidos não é uma novidade. Depois do trabalho de Green et al. ser publicado em 1965, descrevendo de forma completa a teoria geral de uma superfície de Cosserat [4], alguns autores começaram a usá-la para modelagem de objetos deformáveis. Eischen et al. apresentaram em 1996 um modelo fundamentado em uma superfície de Cosserat [16], após a publicação de três artigos por Simo e Fox em 1989, os quais mostraram que, apesar da complicada formulação, ela pode ser implementada de uma maneira numérica eficiente [18]. A principal contribuição deles foi descobrir uma parametrização que evita termos como os símbolos de Christoffel e os coeficientes da segunda forma fundamental.

Em 2004, Wu e de Melo apresentaram um modelo de superfície deformável visando o controle de caimentos e dobras, baseado em uma superfície de Cosserat [3]. Neste trabalho os autores propõem simplificações e restrições, específicas para tecidos, sobre a teoria geral de superfície de Cosserat [4], conseguindo obter um tratamento apropriado das dobras de tecidos quando eles são submetidos às forças de compressão. Este comportamento adequado, que aumenta o realismo do modelo, ocorre em virtude da consideração do termo referente ao acoplamento entre as variações de métrica com as de curvatura para estimar a energia interna, ao contrário de Terzoupoulos et al., que desprezam este termo [19]. Esta consideração é equivalente a dizer que o modelo incorpora as variações nas curvaturas médias, as quais governam as deformações dentro e fora do plano da superfície ao longo do tempo.

Além disso, esse modelo deformável apresenta um controle mais intuitivo dos parâmetros do material do tecido, principalmente quando comparado aos modelos massa-mola, que consideram a definição do material atribuindo-se valores aos coeficientes de rigidez das molas que formam o tecido. No modelo proposto por Wu e de Melo um material é definido com um conjunto de três parâmetros que possuem interpretações geométricas, como resistência à mudança de métrica, resistência à mudança de curvatura e grau de acoplamento entre a métrica e curvatura [3]. Os autores utilizam como solução numérica um esquema muito similar ao proposto por Terzoupoulos et al. [19], porém resolvendo o sistema com uma decomposição LU . O custo computacional desta solução numérica é elevado e, assim como o de Terzoupoulos et al., possui complexidade de ordem $O((mn)^2)$, sendo mn o número de pontos que formam a superfície do tecido. Devido a este baixo desempenho computacional, os resultados deste trabalho foram produzidos com malhas de pequena resolução, conforme as imagens de simulações de tecidos nas Figuras 2.2(a), que mostra dobras em um pedaço de pano, e 2.2(b), que mostra um caimento de uma toalha sobre uma mesa redonda.

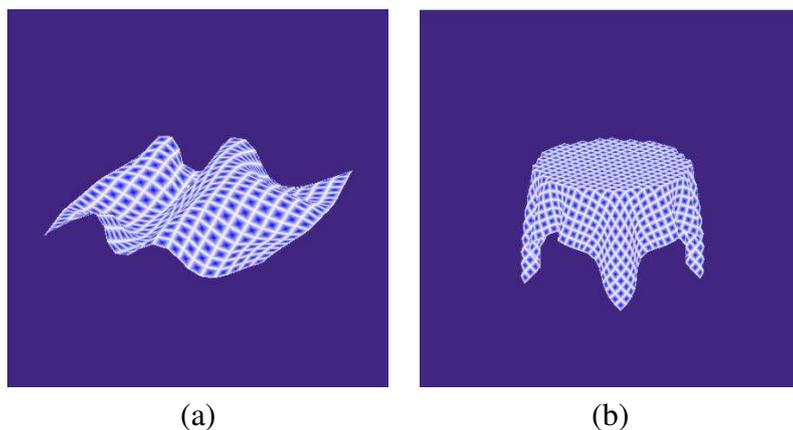


Figura 2.2: Resultados do trabalho de Wu e de Melo em 2004 [3]: (a) dobras; e (b) caimento.

2.3 Classificação dos Modelos

Como visto na Seção 2.2, ainda não existe na literatura um trabalho que aplique métodos de integração explícita sobre um modelo de tecido baseado na mecânica de contínuos. Isto se deve principalmente à ascensão dos modelos baseados na mecânica de partículas na metade da década de 90, como nos trabalhos de Breen et al. [17] e Provot [2], que possuem grande facilidade de implementação e eficiência computacional. Além disso, alguns anos depois, Baraff e Witkin [14] apresentaram um modelo contínuo em conjunto com uma integração implícita capaz de realizar a dinâmica de tecidos de maneira estável para valores de passo de tempo maiores do que os usados por integrações explícitas. Contudo, este último trabalho estava mais focado em atingir uma grande eficiência computacional do que o realismo nas simulações, tendo como ponto forte a solução numérica adotada, ao invés da modelagem de tecido. A partir daí, surgiram trabalhos que agregavam as vantagens da modelagem baseada na mecânica de partículas com as de se usar a integração implícita.

O processamento de colisões em uma simulação de tecidos é de fundamental importância para se alcançar um alto realismo nos resultados visuais, inclusive na formação mais natural de dobras, porém este processo envolve algoritmos de alto custo computacional. Assim, os pesquisadores começaram a investir mais esforços em aprimorar estes algoritmos sobre um *framework* de modelagem da dinâmica interna de tecidos bem aceito, que é o de utilizar integrações implícitas e semi-implícitas para solucionar numericamente modelos baseados na mecânica de partículas. Contudo, esta dissertação propõe um *framework* alternativo para simular tecidos, considerando que para agregar realismo e eficiência computacional é possível utilizar um modelo de tecido baseado na mecânica de contínuos com uma solução explícita ao longo do tempo. Conjectura-se que o realismo está inerente ao uso de um modelo que leva em consideração a compatibilidade de Weingarten, enquanto o desempenho computacional é proporcionado pelo uso de uma integração explícita.

Os trabalhos comentados nas Seções 2.1 e 2.2 estão sintetizados na Tabela 2.1, classificados de acordo com a abordagem de modelagem física e a solução numérica usada. Nela se nota uma lacuna exatamente onde se encaixa a proposta deste trabalho. Assim, com o objetivo de melhor preencher esta lacuna, avaliou-se as diferentes técnicas de integração explícita no modelo proposto por Wu e de Melo [3, 5]. Ele é explicado detalhadamente na Seção 2.4, enquanto os possíveis métodos explícitos que podem solucioná-lo ao longo do tempo é o tema do próximo capítulo desta dissertação.

Tabela 2.1: Classificação dos trabalhos apresentados nas seções 2.1 e 2.2.

Modelos Físicos / Integrações	Explícita	Semi-Implícita	Implícita
Partículas	Haumann [1] Provot [2]	Choi e Ko [12] Bridson et al. [25, 11]	Breen et al. [17, 26, 23]
Contínuos		Terozpoulos et al. [19] Wu e de Melo [3, 5]	Feynman [20] Baraff et al. [14, 10]

2.4 Uma modelagem baseada em uma Superfície de Cosserat

Devido ao comportamento característico de um tecido, de ser muito flexível para curvar e pouco para esticar, o ponto crítico de sua modelagem acontece no momento de descrever a diversidade de dobras que nele se formam com facilidade. Neste sentido, o modelo de superfície deformável escolhido, proposto por Wu e de Melo, é apropriado para descrever uma grande variedade de dobras formadas sob distintas condições, incluindo compressões e distensões [5]. Além disso, neste modelo o material de um tecido é definido com parâmetros que possuem significados geométricos intuitivos: resistência à mudança de métrica, resistência à mudança de curvatura e grau de formação de dobras.

O modelo de tecido selecionado baseia-se em um modelo de casca geometricamente exato, conhecido como superfície de Cosserat. A teoria de superfície de Cosserat foi proposta pelos irmãos Cosserat em 1909, redescoberta por Ericksen e Truesdell em 1958 para a modelagem de corpos orientados e, em 1965, usada para modelagem de cascas por Green et al. [4]. Estes últimos autores integraram os princípios dinâmicos e termodinâmicos da mecânica de contínuos à análise da dinâmica de uma superfície de Cosserat e a descreveram de forma geral e completa, derivando também alguns casos especiais, como o caso da superfície de Cosserat ter propriedades elásticas. É sobre este caso especial que Wu e de Melo fizeram restrições específicas para a modelagem de tecidos [3].

A formulação da superfície de Cosserat apresentada por Green et al. é complexa, pois ela, além de integrar princípios termodinâmicos, utiliza muitos elementos da geometria diferencial para descrever a dinâmica de uma superfície deformável sob a ação de forças externas. Nesta dissertação esta teoria é exposta de forma simplificada e resumida, destacando as restrições e simplificações feitas sobre ela por Wu e de Melo, como apresentado na Subseção 2.4.1. A integração semi-implícita do modelo ao longo do tempo feita por estes últimos autores é explicada na Subseção 2.4.2, usando como principal referência a tese de doutorado apresentada por de Melo [5].

2.4.1 Superfície de Cosserat

Nesta subseção são apresentados, em primeiro lugar, alguns conceitos de geometria diferencial para então definir uma superfície de Cosserat, suas medidas físicas e geométricas e a as forças que atuam sobre ela. Consideram-se um sistema fixo de coordenadas cartesianas z^i ($i = 1, 2, 3$) e um sistema arbitrário de coordenadas curvilíneas x^i ($i = 1, 2, 3$), de tal forma que a transformação entre estes referenciais não seja singular e tenha uma única inversa, ou seja, $\det \left[\frac{\partial z^i}{\partial x^i} \right] > 0$. Considera-se também uma superfície s , mergulhada no espaço euclidiano tridimensional z^i , definida pela equação $x^3 = 0$ e assumem-se as coordenadas convencionadas x^α ($\alpha = 1, 2$) em s e x^3 na direção normal à s . As coordenadas x^i são chamadas de convencionadas porque qualquer ponto em s tem as mesmas coordenadas em todos os estados ao longo da simulação.

Adicionalmente, cada ponto $p = (x^1, x^2) \in s$, em um instante t , é representado em relação à origem do sistema cartesiano fixo pelo vetor-posição $\mathbf{r}(x^1, x^2, t)$. No instante inicial $t = 0$, chamado t_0 , a superfície s é representada por S e cada vetor posição \mathbf{r} por \mathbf{R} , de forma que $\mathbf{r}(x^1, x^2, t_0) = \mathbf{R}(x^1, x^2)$. Esta mesma convenção é usada para diferenciar os valores iniciais e correntes de outros elementos da superfície, como também se convencionou $f_{,\alpha}$ representando a derivada parcial de uma função f em relação à coordenada x^α e a omissão do símbolo \sum para somatórias. A Figura 2.3 ilustra todas as considerações feitas até aqui exibindo configurações da superfície s em dois momentos diferentes: em um estado sem deformações, no instante inicial t_0 , e em um estado deformado, no instante $t = t_0 + \Delta t$, chamado t_1 . O símbolo Δt é conhecido como passo de tempo.

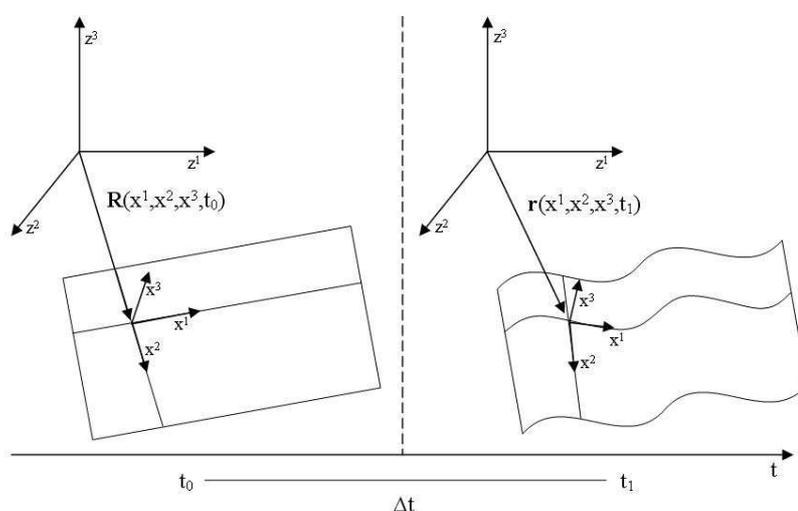


Figura 2.3: Referenciais fixo e arbitrário em instantes diferentes.

Em qualquer ponto p sobre a superfície s , em um instante t , é possível definir dois vetores de base \mathbf{a}_1 e \mathbf{a}_2 , contidos no plano tangente ao ponto p , que são vetores linearmente independentes e podem ser expressos pela derivada parcial do vetor-posição \mathbf{r} em relação às coordenadas x^1 e x^2

$$\mathbf{a}_\alpha(x^1, x^2, t) = \frac{\partial \mathbf{r}}{\partial x^\alpha}(x^1, x^2, t), \quad (2.2)$$

com $\alpha = 1, 2$. O vetor normal \mathbf{n} é o vetor que aponta na direção perpendicular ao plano tangente à p e normalmente é definido como o produto vetorial entre os vetores de base

$$\mathbf{a}_3 = \mathbf{n} = \frac{\frac{\partial \mathbf{r}}{\partial x^1}(x^1, x^2, t) \times \frac{\partial \mathbf{r}}{\partial x^2}(x^1, x^2, t)}{\left| \frac{\partial \mathbf{r}}{\partial x^1}(x^1, x^2, t) \times \frac{\partial \mathbf{r}}{\partial x^2}(x^1, x^2, t) \right|}. \quad (2.3)$$

Os vetores de base \mathbf{a}_1 e \mathbf{a}_2 , juntamente com o vetor normal \mathbf{n} , formam um triedro que representa um sistema de referência local em cada ponto $p \in s$, conforme ilustra a Figura 2.3.

Além dos elementos desse triedro, existem também os tensores métrico, métrico recíproco e de curvatura, associados a cada ponto $p \in s(t)$ e fundamentais para a descrição do modelo. Estes três tensores, juntos com os vetores que formam o referencial local, são elementos da geometria diferencial usados para expressar as deformações, forças e energias que atuam sobre uma superfície de Cosserat. Para tornar as equações mais genéricas, até o final deste capítulo é usada a convenção de que índices latinos (i, j, k) podem assumir os valores 1, 2 e 3, enquanto índices gregos (α, β, γ) apenas os valores 1 e 2, bem como omitir o argumento (x^1, x^2, t) das próximas definições por considerar que elas são válidas para todos os pontos da superfície em qualquer instante de tempo.

O tensor métrico $a_{\alpha\beta}$ é um tensor simétrico de segunda ordem definido como positivo e pode ser representado pela matriz $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, na qual seus elementos são conhecidos na geometria diferencial como os coeficientes da primeira forma fundamental. Este tensor pode ser usado para determinar o comprimento de um pequeno arco ou ângulo sobre as curvas coordenadas de $s(t)$ e pode ser expresso pelo produto escalar entre os vetores de base \mathbf{a}_1 e \mathbf{a}_2

$$a_{\alpha\beta} = \frac{\partial \mathbf{r}}{\partial x^\alpha} \cdot \frac{\partial \mathbf{r}}{\partial x^\beta}. \quad (2.4)$$

O tensor métrico recíproco $a^{\alpha\beta}$ é um tensor simétrico de segunda ordem expresso pelo produto escalar entre os vetores de base recíprocos \mathbf{a}^α . Os vetores \mathbf{a}^α e \mathbf{a}_β são chamados de recíprocos se

$$\mathbf{a}_\alpha \cdot \mathbf{a}^\beta = \delta_\alpha^\beta = \begin{cases} 1 & \text{se } \alpha = \beta \\ 0 & \text{se } \alpha \neq \beta \end{cases}.$$

O determinante a da matriz formada pelos coeficientes do tensor métrico resulta em número real que muitas vezes é usado para estimar a área delimitada pelos vetores de base no plano tangente em um ponto qualquer da superfície $s(t)$. Os coeficientes do tensor métrico recíproco podem ser expressos em função dos termos do tensor métrico $a_{\alpha\beta}$

$$a^{11} = \frac{a_{22}}{a}, \quad a^{12} = a^{21} = -\frac{a_{12}}{a}, \quad a^{22} = \frac{a_{11}}{a}. \quad (2.5)$$

O tensor de curvatura $b_{\alpha\beta}$, assim como tensor métrico, também é um tensor simétrico de segunda ordem e pode ser expresso pela matriz $\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$, em que seus elementos são conhecidos como coeficientes da segunda forma fundamental. Este tensor pode ser usado para estimar a curvatura gaussiana e curvatura média de um ponto em relação à superfície, sendo descrito por

$$b_{\alpha\beta} = \mathbf{n} \cdot \frac{\partial^2 \mathbf{r}}{\partial x^\alpha \partial x^\beta}. \quad (2.6)$$

Uma superfície de Cosserat é uma superfície mergulhada no espaço euclidiano tridimensional em que para cada um de seus pontos é assinalado um vetor deformável \mathbf{d} , não tangente à superfície s [4]. Este vetor, chamado diretor, não necessariamente é unitário ou está na direção normal à superfície s , mas possui a propriedade de permanecer invariável em comprimento sobre transformações de corpo rígido. O mergulho corresponde à maneira como a superfície é inserida no espaço circundante a ela e como se relacionam seus elementos de medição.

Dessa forma, considerando um vetor diretor \mathbf{d} associado a cada posição \mathbf{r} de um ponto $p \in s(t)$, pode-se caracterizar o movimento de uma superfície de Cosserat por diferentes configurações dos vetores \mathbf{r} e \mathbf{d} em instantes de tempo t diferentes

$$\mathbf{r}(t) \equiv \mathbf{r}(x^1, x^2, t), \quad \mathbf{d}(t) \equiv \mathbf{d}(x^1, x^2, t),$$

sobre a restrição de que o vetor $\mathbf{d}(t)$ precisa satisfazer

$$\mathbf{d}(t) = d_i \mathbf{a}^i(t) = d_\alpha \mathbf{a}^\alpha + d_3 \mathbf{n}.$$

A variação ao longo do tempo das medidas do tensor métrico, tensor de curvatura e vetor diretor define três medidas físicas, ou variáveis cinemáticas, que caracterizam as deformações em uma superfície de Cosserat. Estas variações representam as tensões e distensões que ocorrem em uma superfície $s(t)$ em relação ao seu estado não deformado S , expressas por:

1. **Deformações de membrana** ($\varepsilon_{\alpha\beta}$)

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta}). \quad (2.7)$$

2. **Deformações de curvatura** ($\kappa_{\beta i}$)

$$\begin{aligned} \kappa_{\alpha\beta} &= -(b_{\alpha\beta} - B_{\alpha\beta}), \\ \kappa_{\alpha 3} &= 0. \end{aligned} \quad (2.8)$$

3. **Deformações de vetor diretor** (γ_i)

$$\gamma_i = d_i - D_i. \quad (2.9)$$

Seja σ a área de $s(t)$, limitada por uma curva fechada \mathcal{C} e seja $\nu = \nu_\alpha \mathbf{a}^\alpha$ o vetor normal unitário exterior à \mathcal{C} e posicionado sobre s . Se \mathbf{N} é um campo vetorial tridimensional e se, para todos os

campos arbitrários de velocidade \mathbf{v} , o escalar $\mathbf{N} \cdot \mathbf{v}$ representar a taxa de trabalho por unidade de comprimento de \mathcal{C} , então \mathbf{N} é um vetor de força de curva medido por unidade de comprimento. Similarmente, se \mathbf{M} é um campo vetorial tridimensional e se, para todo os campos arbitrários de velocidade diretora \mathbf{w} , o escalar $\mathbf{M} \cdot \mathbf{w}$ é a taxa de trabalho por unidade de comprimento, então \mathbf{M} é chamado de vetor de força diretora, por unidade de comprimento de \mathcal{C} . Estes campos vetoriais podem ser expressos em termos de \mathbf{a}_i

$$\mathbf{N} = N^i \mathbf{a}_i = \nu_\alpha \mathbf{N}^\alpha = \nu_\alpha (N^{i\alpha} \mathbf{a}_i) = (\nu_\alpha N^{i\alpha}) \mathbf{a}_i, \quad (2.10)$$

$$\mathbf{M} = M^i \mathbf{a}_i = \nu_\alpha \mathbf{M}^\alpha = \nu_\alpha (M^{i\alpha} \mathbf{a}_i) = (\nu_\alpha M^{i\alpha}) \mathbf{a}_i, \quad (2.11)$$

onde $N^{\beta\alpha}$, $N^{3\alpha}$, $M^{\beta\alpha}$ e $M^{3\alpha}$ são tensores.

Seja $\mathbf{F} = F^i \mathbf{a}_i$ e $\mathbf{L} = L^i \mathbf{a}_i$, respectivamente, os campos tridimensionais de força atribuída e força diretora, por unidade de massa de $s(t)$, tais que $\mathbf{F} \cdot \mathbf{v}$ e $\mathbf{L} \cdot \mathbf{w}$, para todos \mathbf{v} e \mathbf{w} arbitrários, representem a taxa de trabalho por unidade de área de $s(t)$. Se μ é a densidade de massa por unidade de área de $s(t)$, Green et al. derivaram em [4] equações de equilíbrio de energia com a consideração de que $s(t)$ permanece inalterada sobre transformações de corpo rígido, obtendo a equação de conservação para o momento linear

$$\mu \text{ acel} - \mathbf{N}^\alpha|_\alpha = \mu \mathbf{F}(\mathbf{r}, t), \quad (2.12)$$

onde $\text{acel} = \frac{\partial^2 \text{vel}}{\partial t^2}$. A barra vertical $|_\alpha$ representa a derivada covariante¹ em relação à coordenada x^α , ou seja, a projeção da derivada de \mathbf{N}^α em relação à coordenada x^α sobre o plano tangente à superfície no ponto considerado.

O termo $\mathbf{N}^\alpha|_\alpha$ corresponde à força interna $\mathbf{f}(t)$ de reação da superfície, quando submetida às ações de forças externas $\mathbf{F}(t)$. Green et al. também descreveram a energia armazenada em uma superfície de Cosserat Elástica e a formularam para o caso especial do vetor diretor coincidir com o vetor normal unitário em $s(t)$, ou seja, $\mathbf{d} = \mathbf{n}$, para qualquer $p \in s$ em um instante t . O modelo de tecido proposto por Wu e de Melo considera este caso específico para a modelagem de tecidos [3, 5] e também considera que a espessura da superfície é suficientemente pequena a ponto de ser considerada nula. Neste caso, na teoria de cascas, a superfície é chamada de membrana, sendo $\gamma_i = 0$ e

$$N^{3\alpha} = M^{\alpha\beta}|_\beta, \quad (2.13)$$

onde $M^{\alpha\beta}|_\alpha$ corresponde à derivada covariante de um tensor de superfície de ordem 2 em relação à x^α .

¹No estudo da geometria intrínseca da superfície é definida uma derivada de um vetor do plano tangente à mesma, denominada derivada covariante, com a finalidade de se ter o vetor resultante da operação derivada pertencente ao plano tangente da superfície [5].

Considerando a Equação 2.13 pode-se escrever a derivada covariante do componente \mathbf{N}^α como

$$\begin{aligned} \mathbf{N}^\alpha|_\alpha &\equiv \left[(N^{\beta\alpha} \mathbf{a}_\beta)_{,\alpha} + \Gamma_{\alpha\lambda}^\lambda N^{\beta\alpha} \mathbf{a}_\beta \right] + \left[(N^{i\alpha} \mathbf{a}_i)_{,\alpha} + \Gamma_{\alpha\lambda}^\lambda N^{i\alpha} \mathbf{a}_i \right] \\ &\equiv \left[(N^{\beta\alpha} \mathbf{a}_\beta)_{,\alpha} + \Gamma_{\alpha\lambda}^\lambda N^{\beta\alpha} \mathbf{a}_\beta \right] + \left[(M^{\alpha\beta}|_\beta \mathbf{a}_3)_{,\alpha} + \Gamma_{\alpha\lambda}^\lambda M^{\alpha\beta}|_\beta \mathbf{a}_3 \right], \end{aligned} \quad (2.14)$$

na qual a derivada covariante do tensor de superfície $M^{\alpha\beta}$ pode ser expressa por

$$M^{\alpha\beta}|_\beta = (M^{\alpha\beta})_{,\beta} + (\Gamma_{\beta\lambda}^\alpha M^{\lambda\beta} + \Gamma_{\beta\lambda}^\beta M^{\alpha\lambda}), \quad (2.15)$$

e $\Gamma_{\beta\alpha}^\alpha$ são os símbolos de Christoffel, que são elementos de conexão entre distintos sistemas de referência. Estes elementos são associados ao sistema de coordenadas da superfície x^a e definidos por

$$\Gamma_{\alpha\beta}^\lambda = \mathbf{r}_{,\alpha\beta} \cdot \mathbf{a}^\lambda. \quad (2.16)$$

Desta forma, o termo $N^\alpha|_\alpha$ pode ser obtido encontrando os componentes $\mathbf{N}^{\alpha\beta}$ e $\mathbf{M}^{\alpha\beta}$ do vetor de força de curva \mathbf{N} e do vetor de força diretora \mathbf{M} , respectivamente. Estes componentes podem ser expressos em termos da energia interna \mathbf{E} por unidade de massa acumulada na superfície s e das medidas de deformação $\varepsilon_{\alpha\beta}$ e $\kappa_{\alpha\beta}$ de forma que

$$N^{*\alpha\beta} = \mu \frac{\partial \mathbf{E}}{\partial \varepsilon_{\alpha\beta}} \quad (2.17)$$

$$M^{\alpha\beta} = \mu \frac{\partial \mathbf{E}}{\partial \kappa_{\alpha\beta}}, \quad (2.18)$$

com

$$N^{\alpha\beta} = N^{*\alpha\beta} - b_\beta^\alpha M^{\alpha\beta}, \quad (2.19)$$

onde

$$b_\beta^\alpha = a^{\alpha\beta} b_{\alpha\beta} \quad (2.20)$$

são as relações de compatibilidade de Weingarten.

Para estimar a energia interna \mathbf{E} na superfície s , Wu e de Melo consideraram o caso em que ela é inicialmente homogênea, livre de forças diretoras e que suas deformações ocorrem em temperatura e entropia constantes. Assim, a aproximação para a energia interna pode ser dada em termos dos parâmetros que caracterizam as propriedades do material, $\Phi_{\alpha\beta}$, $\Psi_{\alpha\beta}$ e $\Theta_{\alpha\beta}$, de forma que

$$\mu \mathbf{E} = \left[\Phi^{\alpha\beta} \varepsilon_{\alpha\beta} \varepsilon_{\alpha\beta} + \Psi^{\alpha\beta} \kappa_{\alpha\beta} \kappa_{\alpha\beta} + \Theta^{\alpha\beta} \varepsilon_{\alpha\beta} \kappa_{\alpha\beta} \right]. \quad (2.21)$$

O primeiro e o segundo termo do lado direito da Equação 2.21 são as formas quadráticas das medidas relativas às deformações de métrica e de curvatura da superfície, respectivamente, enquanto o terceiro termo possui um produto entre elas, representando um acoplamento de ambos os efeitos. É este acoplamento na formulação da energia interna do modelo que produz de forma natural as dobras que se formam no tecido sob a ação de forças de compressão e distensão. Considerando que a superfície de Cosserat elástica é isotrópica e, portanto, que os parâmetros do material do tecido satisfazem certas condições de simetria, de Melo [5] simplificou a formulação destes parâmetros usando três coeficientes elásticos, $\zeta_{\alpha\beta}$, $\xi_{\alpha\beta}$ e $\phi_{\alpha\beta}$, para definir a elasticidade do material de um tecido, estabelecendo as equações

$$\begin{aligned}\Phi^{\alpha\beta} &= \Phi^{\beta\alpha} = \zeta_{\alpha\beta}(A^{\alpha\alpha}A^{\beta\beta} + 2(A^{\alpha\beta})^2) \\ \Psi^{\alpha\beta} &= \Psi^{\beta\alpha} = \xi_{\alpha\beta}(A^{\alpha\alpha}A^{\beta\beta} + 2(A^{\alpha\beta})^2) \\ \Theta^{\alpha\beta} &= \Theta^{\beta\alpha} = \phi_{\alpha\beta}(A^{\alpha\alpha}A^{\beta\beta} + 2(A^{\alpha\beta})^2).\end{aligned}\quad (2.22)$$

Os coeficientes $\zeta^{\alpha\beta}$ e $\xi^{\alpha\beta}$ estão associados as resistências da superfície às mudanças de métrica e de curvatura, respectivamente. Por exemplo, uma calça *jeans* possui um valor maior de resistência de curvatura do que uma calça de algodão, que por sua vez possui maior resistência métrica do que uma feita de elastano. Já o coeficiente $\phi^{\alpha\beta}$ é chamado de fator de *buckling* e está relacionado com o número de ondulações que ocorrem na superfície deformada quando sujeita às forças tangenciais.

Aplicando os princípios de conservação de massa, sendo μ e μ_0 as densidades de massa por unidade de área da superfície de Cosserat no estado inicial (\mathcal{S}) e deformado (s) e, substituindo a Equação 2.21 em 2.17 e 2.18, obtêm-se $N^{*\alpha\beta}$ e $M^{\alpha\beta}$ em termos da variação da área da superfície $\frac{\mathcal{S}_0}{\mathcal{S}}$ e dos parâmetros de material $\Phi^{\alpha\beta}$, $\Psi^{\alpha\beta}$ e $\Theta^{\alpha\beta}$

$$\begin{aligned}N^{*\alpha\beta} &= \mu \frac{\partial \mathbf{E}}{\partial \varepsilon_{\alpha\beta}} = \frac{\mu}{\mu_0} (2\Phi^{\alpha\beta} \varepsilon_{\alpha\beta} + \Theta^{\alpha\beta} \kappa_{\alpha\beta}) \\ &= \frac{\mathcal{S}_0}{\mathcal{S}} (2\Phi^{\alpha\beta} \varepsilon_{\alpha\beta} + \Theta^{\alpha\beta} \kappa_{\alpha\beta})\end{aligned}\quad (2.23)$$

$$\begin{aligned}M^{\alpha\beta} &= \mu \frac{\partial \mathbf{E}}{\partial \kappa_{\alpha\beta}} = \frac{\mu}{\mu_0} (2\Psi^{\alpha\beta} \kappa_{\alpha\beta} + \Theta^{\alpha\beta} \varepsilon_{\alpha\beta}) \\ &= \frac{\mathcal{S}_0}{\mathcal{S}} (2\Psi^{\alpha\beta} \kappa_{\alpha\beta} + \Theta^{\alpha\beta} \varepsilon_{\alpha\beta}),\end{aligned}\quad (2.24)$$

com

$$N^{\alpha\beta} = \frac{\mathcal{S}_0}{\mathcal{S}} [(2\Phi^{\alpha\beta} \varepsilon_{\alpha\beta} + \Theta^{\alpha\beta} \kappa_{\alpha\beta}) - b_{\alpha}^{\alpha} (2\Psi^{\alpha\beta} \kappa_{\alpha\beta} + \Theta^{\alpha\beta} \varepsilon_{\alpha\beta})].\quad (2.25)$$

Finalmente, substituindo as equações 2.24 e 2.25 em 2.14 pode-se encontrar $\mathbf{N}^{\alpha}|_{\alpha}$, que corresponde à força interna \mathbf{f} da superfície (Equação 1.1).

2.4.2 Integração Semi-Implícita

Nesta subseção é explicada a forma que de Melo integrou temporalmente seu modelo de superfície deformável, explicado na Subseção 2.4.1, para obter a dinâmica de tecidos. Em essência, o problema consiste em, dadas as medidas de deformação da métrica e curvatura da superfície, determinar os deslocamentos de seus pontos ao longo do tempo de forma coerente. Neste modelo isto é feito com uma integração semi-implícita, utilizando o método de decomposição \mathcal{LU} para solucionar um sistema de equações lineares que tem como incógnitas as posições dos pontos da superfície [5].

A grande dificuldade de utilizar um esquema semi-implícito para integrar o modelo de tecido ao longo do tempo diz respeito a linearização da força interna \mathbf{f} , que é equivalente ao termo $\mathbf{N}^\alpha|_\alpha$ expresso na Equação 2.14. A grosso modo, pode-se dizer que esta força interna de reação da superfície, quando submetida às ações de forças externas, depende de todos os elementos do modelo, inclusive de grandezas geométricas diferenciáveis que não estão prontas para serem calculadas. Desta forma, de Melo discretiza seu modelo no espaço com o método das diferenças finitas para que estas grandezas sejam encontradas a cada novo instante de tempo. Contudo, como o esquema adotado pelo autor é semi-implícito, ele requer, a partir de um instante de tempo t , a formulação de uma parcela de dados do modelo que são encontrados no próximo $t + \Delta t$, junto com as novas posições da superfície que se deseja obter. Em decorrência desta parcela surge a necessidade de linearizar estes dados para que sejam formulados em função das mesmas incógnitas do sistema de equações.

Para linearizar a força interna \mathbf{f} do modelo, de Melo utiliza o mesmo esquema proposto por Terzopoulos et al., explicado na Subseção 2.2.2, que usa duas matrizes esparsas quadradas, \mathbf{K} e \mathbf{A} , de dimensões $nm \times nm$, em que n e m são as medidas do reticulado bidimensional que forma a malha discreta do tecido. A matriz \mathbf{K} é chamada de matriz de rigidez e representa as tensões que cada ponto da superfície possui em relação aos seus pontos imediatamente adjacentes². Esta matriz contém a estimativa de uma parcela da força interna \mathbf{f} no instante $t + \Delta t$, sendo seus elementos expressos em função das posições incógnitas do sistema de equações lineares. Então, a matriz $\mathbf{K}_{t+\Delta t}$ é utilizada como base para adicionar termos do instante anterior t e, com isso, montar a matriz de sistema \mathbf{A}_t , a qual multiplica o vetor incógnita $\mathbf{r}_{t+\Delta t}$. Esta multiplicação é igualada ao vetor de forças externas \mathbf{g}_t , ou seja, $\mathbf{A}_t \mathbf{r}_{t+\Delta t} = \mathbf{g}_t$, sendo os vetores \mathbf{r}_t e \mathbf{g}_t de ordem mn . Assim, pode-se considerar que o esquema é semi-implícito, pois a solução das posições da superfície em cada instante de tempo $t + \Delta t$ utiliza dados do instante anterior t , como os do vetor \mathbf{g}_t e parte dos termos da matriz \mathbf{A}_t , e também dados estimados do instante $t + \Delta t$, como os termos da matriz $\mathbf{K}_{t+\Delta t}$ embutida em \mathbf{A}_t .

Maiores detalhes deste procedimento de integração semi-implícita podem ser encontrados em [5]. No momento, para esta dissertação, é interessante salientar que este procedimento numérico resulta

²Pontos imediatamente adjacentes a um ponto correspondem aos pontos que formam a vizinhança chamada de 1-anel (1-ring), sendo ela correspondente, neste modelo, para um ponto fora das bordas do tecido, a oito pontos vizinhos.

em uma complexidade computacional da ordem $O((mn)^2)$, devido à necessidade da multiplicação de cada uma das linhas da matriz \mathbf{A}_t pelo vetor $\mathbf{r}_{t+\Delta t}$. Para a solução numérica do sistema linear montado, de Melo utilizou a biblioteca matemática *Meschach* [27] e aplicou o método de decomposição \mathcal{LU} , obtendo grandes problemas de desempenho computacional e também uma limitação dos números de pontos que formam a superfície [5]. A maior superfície utilizada por este autor tem dimensões 37×37 , sendo que seu simulador desenvolvido consome, para esta superfície, mais de uma hora para simular 10 segundos de animação. Além disso, para utilizar este esquema de integração sobre seu modelo deformável, de Melo investiu um grande esforço para escrever os termos da matriz $\mathbf{K}_{t+\Delta t}$ de forma a linearizar todos os elementos usados, como o vetor normal de cada ponto, que é estimado com uma expressão linear envolvendo as derivadas de primeira e segunda ordem e os símbolos de Christoffel, com o intuito de contornar a formulação descrita na Equação 2.3.

2.5 Considerações Finais

Neste capítulo foi apresentado um panorama dos trabalhos relacionados com a simulação de tecidos, destacando neles sua abordagem de modelagem física e solução numérica. Na Seção 2.1 os modelos baseados na mecânica de partículas foram discutidos, destacando suas vantagens, como a facilidade de implementação e o desempenho computacional para calcular as forças internas, e também suas desvantagens, como não conseguir representar de forma apropriada o comportamento peculiar dos tecidos, diminuindo características como realismo e precisão física. Um dos representantes deste tipo de abordagem física é o modelo massa-mola, amplamente usado na simulação de tecidos. As soluções explícitas foram utilizadas junto com os modelos baseados na mecânica de partículas principalmente na década de 90, perdendo espaço desde então para os métodos implícitos e semi-implícitos.

Já na Seção 2.2 os modelos baseados na mecânica de contínuos foram apresentados, destacando sua formulação mais precisa, sob o ponto de vista físico, que consegue modelar o comportamento de tecidos de forma mais realista que os modelos baseados em partículas. Entretanto, como em geral as forças internas são obtidas pelas variações de elementos da geometria diferencial na superfície deformável, os modelos desta abordagem física necessitam encontrar mais parâmetros e, conseqüentemente, fazer mais cálculos, apresentando um baixo desempenho computacional. Além disso, como eles possuem uma formulação mais complexa, apresentam uma menor facilidade de implementação quando comparados aos modelos baseados na mecânica de partículas. As soluções implícitas e semi-implícitas foram muito utilizadas em conjunto com estes modelos há mais de duas décadas, sendo esta combinação também o foco de alguns trabalhos recentes.

Na Seção 2.3 os trabalhos apresentados nas seções 2.1 e 2.2 foram classificados com relação às

duas abordagens de modelagem física de tecidos: mecânica de partículas e mecânica de contínuos; e também em relação aos três tipos de integrações temporais: explícita, semi-implícita e implícita. Esta classificação de todos os trabalhos apresentados serviu para destacar que ainda não existe na literatura a abordagem proposta nesta dissertação, que consiste em simular a dinâmica de tecidos unindo um modelo de tecido baseado na mecânica de contínuos com uma solução explícita ao longo do tempo.

Por fim, a Seção 2.4 apresentou o modelo físico, baseado na mecânica de contínuos, usado neste trabalho. Este modelo deformável foi proposto por de Melo e, fundamentado na teoria de superfície de Cosserat, se destaca dos demais, principalmente, por conseguir reproduzir naturalmente dobras que são formadas nos tecidos por forças de compressão [5]. Com o intuito de sintetizar as demais qualidades que justificam sua escolha, as seguintes propriedades foram decisivas:

1. Ele é adequado para analisar grandes deformações, como as que acontecem nos tecidos.
2. As condições de compatibilidade de Weingarten são satisfeitas, assegurando a regularidade da superfície em qualquer estado que o tecido possa assumir.
3. A analogia entre as quantidades mecânicas (forças de contato e acoplamentos) e geométricas (esticamentos e curvamentos) pode ser facilmente estabelecida.
4. Os poucos parâmetros usados para definir um material possuem significados geométricos claros, tornando o controle do modelo mais fácil.

Ao final da Seção 2.4 a integração temporal semi-implícita do modelo selecionado, usada por de Melo para produzir a dinâmica de tecidos, foi explicada para que fique clara a mudança proposta nesta dissertação. A idéia é juntar as vantagens deste modelo de superfície deformável e apresentar uma solução alternativa, usando métodos de integração explícita, para que se agregue realismo e precisão física, inerentes ao modelo, com o desempenho computacional de comportamento linear dos esquemas explícitos. O próximo capítulo apresenta a discretização espacial do modelo em cada iteração, utilizando a técnica de diferenças finitas, para que então diferentes soluções explícitas, que linearizem o mesmo problema de solução dinâmica, sejam apresentadas e implementadas sobre ele. Com isso, espera-se descobrir qual é a mais adequada para preencher a lacuna destacada na Seção 2.3.

Capítulo 3

Soluções Numéricas

Este capítulo apresenta diferentes soluções numéricas explícitas que podem integrar ao longo do tempo o modelo de tecido proposto por de Melo [5]. Entretanto, antes de apresentar os métodos numéricos é necessário discretizar no espaço o modelo de tecido para transformar as suas equações diferenciais parciais em equações diferenciais ordinárias em função do tempo. Problemas de indefinição de pontos necessários para o cálculo de elementos do modelo nas bordas da malha discreta são solucionados com um esquema de diferenças finitas condicionado diferentemente para cada fronteira, apresentado na Seção 3.1. Este passo é essencial para o modelo baseado na mecânica de contínuos possuir uma formulação similar àquela usada na mecânica de partículas, aproveitando algumas de suas vantagens de eficiência computacional.

Os principais métodos de integração são explicados na Seção 3.2, destacando suas formulações, vantagens e desvantagens. No modelo de tecido usado neste trabalho não existe a definição de uma função que descreva o lugar geométrico de todos os pontos sobre a superfície, sendo o modelo responsável por, dada uma amostra discreta da superfície, calcular as forças totais aplicadas em cada ponto, determinar as acelerações resultantes e assegurar que as novas posições dos pontos da amostra produzam a percepção de uma superfície suave. Desta forma, os métodos apresentados possuem o propósito de produzir o movimento de tecidos, encontrando diferentes superfícies ao longo do tempo. A Seção 3.3 apresenta os trabalhos relacionados à simulação de tecidos que utilizam os métodos explícitos discutidos, com o intuito de destacar a eficácia desta abordagem em conjunto com um modelo contínuo, em relação às demais existentes atualmente. Para melhor preencher a lacuna destacada no Capítulo 2, a Seção 3.4 mostra resultados comparativos entre as implementações dos diferentes métodos numéricos com o objetivo de investigar qual é a melhor opção em termos de precisão e custo computacional. Na Seção 3.5 é feita uma análise dos resultados obtidos para justificar a escolha do método numérico que agrega de forma balanceada fatores como o realismo e eficiência aplicados ao modelo proposto por de Melo [5]. Por fim, a Seção 3.6 apresenta considerações finais.

3.1 Discretizações Espaciais

A primeira preocupação na discretização espacial do modelo proposto por de Melo [5] é tornar o domínio contínuo em um conjunto discreto de nós bidimensionais. Dessa forma, a superfície é discretizada por uma malha retangular de $m \times n$ pontos, dispostos em duas direções, x^1 e x^2 , cujos espaçamentos entre nós consecutivos são Δx^1 e Δx^2 , respectivamente. A segunda preocupação é aproximar as derivadas de primeira e de segunda ordem, que é feito aqui com o método das diferenças finitas. Neste método, cada nó $[k, l]$, com $k = 0..m - 1$ e $l = 0..n - 1$, refere-se a um ponto discreto no domínio de acordo com a convenção da Figura 3.1. Qualquer nó $[k, l]$ da malha bidimensional corresponde a um ponto p cuja posição tridimensional é dada pelo vetor-posição \mathbf{r} , de forma que $\mathbf{r}[k, l] = (p_x, p_y, p_z)$, sendo p_i o valor da coordenada do ponto p no eixo i .

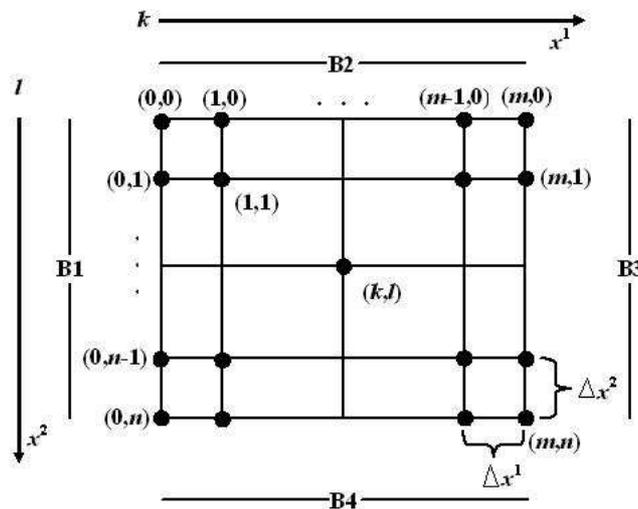


Figura 3.1: Malha discreta.

Considerando uma função qualquer f , são definidos, em uma determinada ordem, operadores avançados, atrasados e centrais de acordo com a direção em que os pontos vizinhos são utilizados para aproximar as derivadas desejadas [28]. Os operadores de diferença posterior (ou de diferença avançada) de primeira ordem podem ser expressos por

$$D_1^+(f[k, l]) = \frac{f[k + 1, l] - f[k, l]}{\Delta x^1}, \quad (3.1)$$

$$D_2^+(f[k, l]) = \frac{f[k, l + 1] - f[k, l]}{\Delta x^2}, \quad (3.2)$$

e os operadores de diferença anterior (ou diferença atrasada) de primeira ordem por

$$D_1^-(f[k, l]) = \frac{f[k, l] - f[k-1, l]}{\Delta x^1}, \quad (3.3)$$

$$D_2^-(f[k, l]) = \frac{f[k, l] - f[k, l-1]}{\Delta x^2}. \quad (3.4)$$

Para as derivadas parciais de segunda ordem utilizam-se operadores definidos pela combinação das equações anteriores, sendo os operadores de diferença cruzada posterior e anterior definidos por

$$D_{12}^+(f[k, l]) = D_{21}^+(f[k, l]) = \frac{f[k+1, l+1] - f[k+1, l] - f[k, l+1] + f[k, l]}{\Delta x^1 \Delta x^2}, \quad (3.5)$$

$$D_{12}^-(f[k, l]) = D_{21}^-(f[k, l]) = \frac{f[k-1, l-1] - f[k-1, l] - f[k, l-1] + f[k, l]}{\Delta x^1 \Delta x^2}, \quad (3.6)$$

e os operadores de diferença central de segunda ordem por

$$D_{11}(f[k, l]) = \frac{f[k+1, l] - 2 * f[k, l] + f[k-1, l]}{\Delta x^1 \Delta x^1}, \quad (3.7)$$

$$D_{22}(f[k, l]) = \frac{f[k, l+1] - 2 * f[k, l] + f[k, l-1]}{\Delta x^2 \Delta x^2}. \quad (3.8)$$

Considerando a convenção de bordas ($B1, B2, B3$ e $B4$) e a orientação dos índices ilustrada na Figura 3.1, de Melo considerou os operadores de primeira ordem avançados para definir os vetores de base \mathbf{a}_α de um ponto $[k, l]$ usando a função \mathbf{r} da seguinte forma [5]

$$\mathbf{a}_\alpha[k, l] = \mathbf{r}_{,\alpha}[k, l] = D_\alpha^+(\mathbf{r}[k, l]).$$

Desta forma, para os pontos da malha discreta que estão nas bordas $B3$ e $B4$ as derivadas direcionais para frente na direção x^1 e x^2 , respectivamente, não são definidas, pois não há pontos sucessores. Neste caso, as derivadas são consideradas nulas, ou seja,

$$\mathbf{a}_\alpha[k, l] = 0, \quad \text{para } (k, l) \in \{B3, B4\}. \quad (3.9)$$

A condição expressa pela Equação 3.9 implica em anular outros elementos do modelo deformável, como o tensor métrico, que tem seus coeficientes calculados por produtos escalares entre os vetores de base. A área associada a cada ponto, calculada pelo determinante da matriz formada com os coeficientes do tensor métrico, também é zerada nas mesmas bordas por consequência. Estas áreas anuladas afetam a área total da superfície, usada no cálculo da força interna do modelo. Além disso,

a variação do tensor métrico ao longo do tempo fornece medidas de deformação que também são usadas no cálculo das forças internas. Assim, as bordas $B1$ e $B2$ possuem valores corretos desses elementos enquanto $B3$ e $B4$ não, causando um desequilíbrio na superfície que pode resultar em resultados irrealistas, em que pontos apresentam grande deslocamentos, como ilustrado por uma imagem de simulação na Figura 3.2(a), ou na má formação de dobras, conforme a Figura 3.2(b). Uma solução alternativa seria resolver até a coluna e a linha para as quais as derivadas direcionais são bem definidas. No entanto, percebeu-se que sem estas derivadas direcionais não se pode atualizar corretamente os pontos nas colunas e nas linhas desconsideradas, podendo também produzir efeitos visuais indesejados.

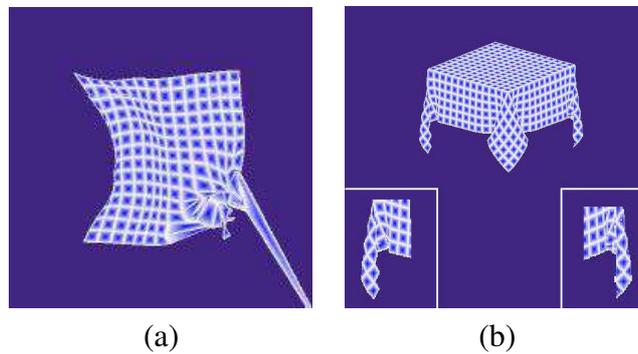


Figura 3.2: Problemas nas bordas.

A solução adotada visa não anular nenhuma derivada de primeira ou segunda ordem, empregando sempre pontos vizinhos na direção em que eles existem. Assim, para os vetores de base que utilizam o operador de primeira ordem para frente sobre a função $\mathbf{r}[k, l]$, os pontos das bordas $B3$ e $B4$ não possuem vizinhos nas direções positivas, mas possuem vizinhos em ambas as direções no sentido negativo, de forma que o operador de primeira ordem para trás, expressos pelas equações 3.3 e 3.4, podem ser usados nestas bordas. Esta nova consideração é formulada como

$$\mathbf{a}_\alpha[k, l] = D_\alpha^-(\mathbf{r}[k, l]), \text{ para } (k, l) \in \{B3, B4\}.$$

Essa solução não está restrita somente às derivadas da função-posição \mathbf{r} , sendo também aplicada as demais derivadas de primeira ordem do modelo, como nos termos $(M^{\alpha\beta})_{,\alpha}$ e $(N^{\beta\alpha})_{,\alpha}$. Assim, para o termo $(M^{\alpha\beta})_{,\alpha}$, que é formulado de forma discretizada como

$$(M^{\alpha\beta})_{,\alpha} = M^{\alpha\beta}|_\beta[k, l] = D_\alpha^-(m_{\alpha\beta}[k, l]) + (\Gamma_{\alpha\beta}^\lambda M^{\alpha\beta} + \Gamma_{\alpha\beta}^\lambda M^{\alpha\beta}),$$

a mesma solução, de usar a direção existente quando a do sentido do operador de diferença finita não

existir, é considerada. E da mesma forma para o termo $(N^{\beta\alpha})_{,\alpha}$, discretizado como

$$(N^{\beta 1})_{,1} = D_1^-(N^{\beta 1} \mathbf{a}_\beta)[k, l] = \frac{N^{\beta 1}[k, l]}{\Delta x^1} \mathbf{a}_\beta[k, l] - \frac{N^{\beta 1}[k-1, l]}{\Delta x^1} \mathbf{a}_\beta[k-1, l],$$

$$(N^{\beta 1})_{,2} = D_2^-(N^{\beta 2} \mathbf{a}_\beta)[k, l] = \frac{N^{\beta 2}[k, l]}{\Delta x^2} \mathbf{a}_\beta[k, l] - \frac{N^{\beta 2}[k, l-1]}{\Delta x^2} \mathbf{a}_\beta[k, l-1].$$

Prosseguindo com a discretização espacial dos demais elementos do modelo, o vetor normal, que em conjunto com os vetores de base formam o referencial local a cada ponto da superfície do tecido, são calculados utilizando todas as faces adjacentes existentes em cada ponto. Assim, para cada ponto da malha discreta o vetor normal é aproximado pela média normalizada dos vetores normais, estes também normalizados, das quatro (no máximo) faces adjacentes em cada ponto

$$\begin{aligned} \mathbf{n}^0[k, l] &= \frac{(\mathbf{r}[k+1, l] - \mathbf{r}[k, l]) \times (\mathbf{r}[k, l+1] - \mathbf{r}[k, l])}{|(\mathbf{r}[k+1, l] - \mathbf{r}[k, l]) \times (\mathbf{r}[k, l+1] - \mathbf{r}[k, l])|}, \\ \mathbf{n}^1[k, l] &= \frac{(\mathbf{r}[k, l+1] - \mathbf{r}[k, l]) \times (\mathbf{r}[k-1, l] - \mathbf{r}[k, l])}{|(\mathbf{r}[k, l+1] - \mathbf{r}[k, l]) \times (\mathbf{r}[k-1, l] - \mathbf{r}[k, l])|}, \\ \mathbf{n}^2[k, l] &= \frac{(\mathbf{r}[k-1, l] - \mathbf{r}[k, l]) \times (\mathbf{r}[k, l-1] - \mathbf{r}[k, l])}{|(\mathbf{r}[k-1, l] - \mathbf{r}[k, l]) \times (\mathbf{r}[k, l-1] - \mathbf{r}[k, l])|}, \quad e \\ \mathbf{n}^3[k, l] &= \frac{(\mathbf{r}[k, l-1] - \mathbf{r}[k, l]) \times (\mathbf{r}[k+1, l] - \mathbf{r}[k, l])}{|(\mathbf{r}[k, l-1] - \mathbf{r}[k, l]) \times (\mathbf{r}[k+1, l] - \mathbf{r}[k, l])|}, \end{aligned}$$

de forma que a média do vetor normal resultante, em cada ponto, seja computada por

$$\mathbf{n}[k, l] = \frac{\sum_{i=0}^3 \mathbf{n}^i}{|\sum_{i=0}^3 \mathbf{n}^i|}. \quad (3.10)$$

Assim, para os pontos de todas as bordas em que não existem as quatro faces adjacentes, a Equação 3.10 é utilizada apenas com as faces existentes, de forma que os quatro cantos da malha retangular possuem apenas uma face adjacente e os demais pontos das bordas, duas.

Partindo para a discretização espacial dos elementos de segunda ordem, que dependem do cálculo de derivadas parciais de segunda ordem, de Melo utilizou os operadores avançados de segunda ordem para $\alpha \neq \beta$, e os operadores centrais para $\alpha = \beta$ [5], de forma que

$$\begin{aligned} \mathbf{r}_{,\alpha\beta}[k, l] &= D_{\alpha\beta}^+(\mathbf{r}[k, l]), \quad se \quad \alpha \neq \beta, \quad e \\ \mathbf{r}_{,\alpha\beta}[k, l] &= D_{\alpha\beta}(\mathbf{r}[k, l]), \quad se \quad \alpha = \beta. \end{aligned}$$

Novamente, para as derivadas de segunda ordem existirá o problema de indefinição de pontos vizinhos nas bordas, mais especificamente nas bordas $B3$ e $B4$ para os operadores cruzados avançados e nas quatro bordas para o operador central. Nestes casos, o resultado da derivação é

$$D_{\alpha\beta}^+(\mathbf{r}[k, l]) = 0, \quad \text{para } (k, l) \in \{B3, B4\}, \quad e \quad (3.11)$$

$$D_{\alpha\beta}(\mathbf{r}[k, l]) = 0, \quad \text{para } (k, l) \in \{B1, B2, B3, B4\}. \quad (3.12)$$

Convencionando-se C_i , com $i = 1..4$, como os cantos entre as bordas da malha discreta, sendo o canto C_4 o que fica entre as bordas $B4$ e $B1$, o canto C_1 entre as bordas $B1$ e $B2$ e assim sucessivamente, a solução para a condição expressa pela Equação 3.11, para índices cruzados $\alpha \neq \beta$, é dividida em quatro situações:

- Para o ponto $p = (k, l) = C3$, a derivada é expressa pela Equação 3.6, ou seja,

$$\mathbf{r}_{,12}(k, l) = \mathbf{r}_{,21}(k, l) = D_{12}^-(\mathbf{r}[k, l]) = D_{21}^-(\mathbf{r}[k, l]).$$

- Para o ponto $p = (k, l) = C4$, a derivada é expressa por

$$\mathbf{r}_{,12}(k, l) = \mathbf{r}_{,21}(k, l) = \frac{\mathbf{r}[k+1, l-1] + \mathbf{r}[k+1, l] + \mathbf{r}[k, l-1] + \mathbf{r}[k, l]}{\Delta x^1 \Delta x^2}.$$

- Para os pontos $p = (k, l) \in \{B3 - C3\}$

$$\mathbf{r}_{,12}(k, l) = \mathbf{r}_{,21}(k, l) = \frac{\mathbf{r}[k-1, l+1] + \mathbf{r}[k-1, l] + \mathbf{r}[k, l+1] + \mathbf{r}[k, l]}{\Delta x^1 \Delta x^2}.$$

- Para pontos $p = (k, l) \in \{B4 - C3 - C4\}$

$$\mathbf{r}_{,12}(k, l) = \mathbf{r}_{,21}(k, l) = \frac{\mathbf{r}[k+1, l-1] + \mathbf{r}[k+1, l] + \mathbf{r}[k, l-1] + \mathbf{r}[k, l]}{\Delta x^1 \Delta x^2}.$$

O objetivo é sempre usar os pontos adjacentes existentes ao invés de anular as derivadas, evitando o desequilíbrio das forças internas. Quanto à solução para a condição expressa na Equação 3.12, para os quais de Melo [5] utilizou os operadores das Equações 3.7 e 3.8, o problema ocorre em todas as quatro bordas, pois os operadores centrais usam dois pontos vizinhos em cada uma das duas direções, de forma que sempre faltará um ponto em uma direção. O problema é solucionado inicialmente para os quatro cantos da malha discreta da seguinte forma:

- Para o ponto $p = (k, l) = C1$, a derivada é expressa pela Equação 3.5,

$$\mathbf{r}_{,11}(k, l) = \mathbf{r}_{,22}(k, l) = D_{12}^+(\mathbf{r}[k, l]) = D_{21}^+(\mathbf{r}[k, l]).$$

- Para o ponto $p = (k, l) = C2$, a derivada é expressa por

$$\mathbf{r}_{,11}(k, l) = \mathbf{r}_{,22}(k, l) = \frac{\mathbf{r}[k-1, l+1] + \mathbf{r}[k-1, l] + \mathbf{r}[k, l+1] + \mathbf{r}[k, l]}{\Delta x^1 \Delta x^2}.$$

- Para o ponto $p = (k, l) = C3$, a derivada é expressa pela Equação 3.6,

$$\mathbf{r}_{,11}(k, l) = \mathbf{r}_{,22}(k, l) = D_{12}^-(\mathbf{r}[k, l]) = D_{21}^-(\mathbf{r}[k, l]).$$

- Para o ponto $p = (k, l) = C4$, a derivada é expressa por

$$\mathbf{r}_{,11}(k, l) = \mathbf{r}_{,22}(k, l) = \frac{\mathbf{r}[k+1, l-1] + \mathbf{r}[k+1, l] + \mathbf{r}[k, l-1] + \mathbf{r}[k, l]}{\Delta x^1 \Delta x^2}.$$

Nesses pontos dos cantos, os índices centrais utilizam as mesmas soluções dos índices cruzados, pois considerando uma vizinhança de no máximo oito pontos imediatamente vizinhos (1-anel), eles possuem apenas três pontos adjacentes. Quanto às derivadas de segunda ordem de índices iguais para o restante (excluindo-se os cantos) dos pontos das bordas, pode-se dividir a solução conforme as situações:

- Para os pontos $p = (k, l) \in \{B1 - C1 - C4\}$, a derivada em relação a x^1

$$\mathbf{r}_{,11}(k, l) = D_{12}^+(\mathbf{r}[k, l]).$$

- Para os pontos $p = (k, l) \in \{B1 - C1 - C4\}$, a derivada em relação a x^2

$$\mathbf{r}_{,22}(k, l) = D_{22}(\mathbf{r}[k, l]).$$

- Para os pontos $p = (k, l) \in \{B2 - C2 - C1\}$, a derivada em relação a x^1

$$\mathbf{r}_{,11}(k, l) = D_{11}(\mathbf{r}[k, l]).$$

- Para os pontos $p = (k, l) \in \{B2 - C2 - C1\}$, a derivada em relação a x^2

$$\mathbf{r}_{,22}(k, l) = D_{12}^+(\mathbf{r}[k, l]).$$

- Para os pontos $p = (k, l) \in \{B3 - C3 - C2\}$, a derivada em relação a x^1

$$r_{,11}(k, l) = \frac{\mathbf{r}[k-1, l+1] + \mathbf{r}[k-1, l] + \mathbf{r}[k, l+1] + \mathbf{r}[k, l]}{\Delta x^1 \Delta x^2}.$$

- Para os pontos $p = (k, l) \in \{B3 - C3 - C2\}$, a derivada em relação a x^2

$$\mathbf{r}_{,22}(k, l) = D_{22}(\mathbf{r}[k, l]).$$

- Para os pontos $p = (k, l) \in \{B4 - C4 - C3\}$, a derivada em relação a x^1

$$\mathbf{r}_{,11}(k, l) = D_{11}(\mathbf{r}[k, l]).$$

- Para os pontos $p = (k, l) \in \{B4 - C4 - C3\}$, a derivada em relação a x^2

$$r_{,22}(k, l) = \frac{\mathbf{r}[k+1, l-1] + \mathbf{r}[k+1, l] + \mathbf{r}[k, l-1] + \mathbf{r}[k, l]}{\Delta x^1 \Delta x^2}.$$

Esta discretização espacial se baseia em uma vizinhança 1-anel, por isso se aproxima a derivada segunda de um canto, como $C1$, com operadores cruzados ao invés de usar operadores avançados para índices centrais, utilizando os pontos $[k, l]$, $[k+1, l]$ e $[k+2, l]$, sendo este último ponto pertencente à vizinhança 2-anel. Apesar dos exemplos desta subseção estarem focados nas derivações da função $\mathbf{r}[k, l]$, todas as demais são calculadas da mesma forma. A Figura 3.3 resume as diferentes considerações para as bordas.

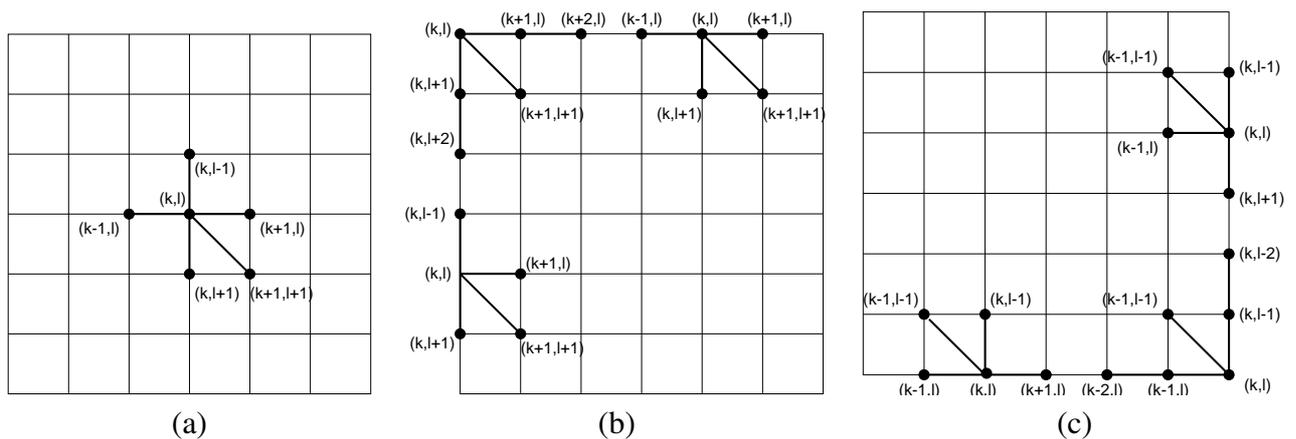


Figura 3.3: Derivadas aproximadas com diferenças finitas: (a) para pontos internos; (b) para pontos nas bordas B1 e B2; e (c) para pontos nas bordas B3 e B4.

3.2 Métodos Explícitos de Integração

Existem diversos métodos explícitos de integração numérica que podem ser de passos e ordens diferentes, variando características como estabilidade e precisão numérica. Estes métodos são muito utilizados para encontrar uma solução aproximada de equações diferenciais ordinárias. Geralmente, a solução visa encontrar o valor de uma função em um ponto sem saber a definição da função, porém conhecendo uma ou mais de suas funções derivadas e também valores da função em alguns pontos. Uma equação diferencial ordinária de ordem superior pode ser sempre reduzida em um conjunto de equações diferenciais ordinárias de segunda ordem à custa de variáveis auxiliares. Para solucionar a Equação 1.1, de segunda ordem, pode-se reescrevê-la em duas equações de primeira ordem acopladas

$$\frac{\partial \mathbf{r}(p)}{\partial t} = \mathbf{vel}(p, t), \quad (3.13)$$

$$\mathbf{acel}(p, t) = \frac{\partial^2 \mathbf{r}(p)}{\partial t^2} = \frac{\partial \mathbf{vel}(p)}{\partial t} = \frac{\mathbf{F}(p, t) - \varrho \mathbf{vel}(p, t) - \mathbf{f}(p, t)}{\mu}, \quad (3.14)$$

onde \mathbf{vel} é uma variável auxiliar correspondente à grandeza física velocidade.

Assim, o objetivo dos métodos numéricos pode ser solucionar funções na forma $\frac{dr(t_i)}{dt} = f_i(t, x(t_i))$, onde os valores de $f_i(t, x(t_i))$ são conhecidos. Associado à Equação 1.1 existe o estado inicial da malha do tecido $\mathbf{r}(t_0)$, a partir do qual se pretende conseguir novas posições da malha $\mathbf{r}(t + \Delta t)$, no instante de tempo $t + \Delta t$, com base nas variações dr e dt do instante anterior t . Desta forma, é gerado um erro por aproximar estas variações infinitesimais em passos finitos Δr e Δt , sendo este erro relacionado ao tamanho do incremento Δt , chamado neste contexto de passo de tempo. A principal limitação dos métodos explícitos é a imposição de se usar um valor pequeno de passo de tempo para garantir a boa aproximação da solução. Entretanto, eles são simples de serem implementados e possuem bom desempenho computacional.

As subseções a seguir apresentam os principais métodos explícitos que podem ser usados para integrar ao longo do tempo o modelo de tecido proposto por de Melo [5]. Em cada subseção, primeiro é apresentada a teoria do método, como sua formulação e precisão numérica, para então mostrar como se encontra, na prática, as novas posições da superfície do tecido em cada instante de tempo. Neste modelo deformável é necessário computar em cada iteração a Equação 3.14 para obter a aceleração em um instante t . O valor da força interna $\mathbf{f}(p, t)$ é obtido com o uso de um esquema de diferenças finitas aplicado na Equação 2.14, conforme explicado na Seção 3.1. A aceleração de cada ponto é então usada para encontrar sua nova velocidade e posição. Desta forma, o cômputo da Equação 3.14 é requerido para obter a variação das velocidades em um passo de tempo Δt e, com ela, aplica-se um método explícito para encontrar a nova velocidade $\mathbf{vel}(t + \Delta t)$ e a nova geometria $\mathbf{r}(t + \Delta t)$ em um processo iterativo até atingir determinado instante de tempo pré-especificado.

3.2.1 Método de Euler

No campo da matemática, física e computação, o método explícito de Euler, também chamado de método de Euler para frente, é o procedimento numérico mais simples e básico usado para encontrar a solução de equações diferenciais ordinárias com um dado valor inicial. Este método consiste em aproximar a função $x'(t) = f(t, x(t))$ por

$$\begin{aligned} f(t, x(t)) &= \frac{x(t + \Delta t) - x(t)}{\Delta t}, \\ x(t + \Delta t) &= x(t) + \Delta t f(t, x(t)). \end{aligned}$$

Comparando com a expansão de Taylor de $x(t)$

$$x(t_0 + \Delta t) = x(t_0) + \Delta t x'(t_0) + \frac{\Delta t^2}{2!} x''(t_0) + \dots + \frac{\Delta t^n}{n!} \frac{\partial^n x}{\partial t^n} + \dots, \quad (3.15)$$

que define a aproximação de $x(t)$ em termos das suas derivadas, observa-se que no método de Euler o erro é menor que uma potência de Δt

$$x(t_0 + \Delta t) = x(t_0) + \Delta t x'(t_0) + O(\Delta t^2). \quad (3.16)$$

Como ele possui um erro de truncamento de segunda ordem, diz-se que é um método de precisão numérica de primeira ordem, sendo um método pouco preciso, embora seja simples de ser implementado e um dos que possui maior desempenho computacional. Com ele, em qualquer instante de tempo t , primeiro a aceleração de cada ponto da superfície é encontrada com a Equação 3.14, para a nova posição ser atualizada em $t + \Delta t$ com a velocidade do instante t e, então, atualizar a velocidade do próximo instante $t + \Delta t$ com a aceleração de t . Este procedimento obtém, em cada iteração, uma nova geometria da superfície determinando as posições e velocidades em $t + \Delta t$ com base na velocidade e aceleração de t , respectivamente. Assim, considerando que as equações apresentadas são aplicadas a cada ponto p da superfície $s(t)$, o procedimento pode ser descrito pelos passos:

1. Cálculo da aceleração $\text{acel}(p, t)$ com a Equação 3.14, supondo $\mathbf{r}(t)$ e $\text{vel}(t)$ de qualquer p como conhecidos

$$\text{acel}(p, t) = \frac{\mathbf{F}(p, t) - \rho \text{vel}(p, t) - \mathbf{f}(p, t)}{\mu},$$

2. Atualização das posições

$$\mathbf{r}(p, t + \Delta t) = \mathbf{r}(p, t) + \Delta t \text{vel}(p, t),$$

3. Atualização das velocidades

$$\mathbf{vel}(p, t + \Delta t) = \mathbf{vel}(p, t) + \Delta t \mathbf{acel}(p, t).$$

Nessa formulação do método explícito de Euler as posições dos pontos são computadas antes das velocidades serem atualizadas com a aceleração encontrada no primeiro passo do procedimento. A aceleração da Equação 3.14 é considerada referente ao instante t devido ao fato das forças do lado direito da equação serem relativas ao mesmo instante. Entretanto, Provot considerou esta aceleração como relativa ao instante $t + \Delta t$ e usou uma fórmula física mais simples para encontrar a aceleração de cada ponto, utilizando o método explícito de Euler com uma pequena variação em relação ao procedimento apresentado: ao invés de aproximar por diferença finita atrasada, ele aproxima por diferença finita avançada. Assim, pode-se atualizar as posições $\mathbf{r}(t + \Delta t)$ já usando a nova velocidade $\mathbf{vel}(t + \Delta t)$ conforme o procedimento [2]:

1. Cálculo da aceleração $\mathbf{acel}(t + \Delta t)$ com a Equação 3.14

$$\mathbf{acel}(p, t + \Delta t) = \frac{\mathbf{F}(p, t) - \rho \mathbf{vel}(p, t) - \mathbf{f}(p, t)}{\mu},$$

2. Atualização das velocidades

$$\mathbf{vel}(p, t + \Delta t) = \mathbf{vel}(p, t) + \Delta t \mathbf{acel}(p, t + \Delta t).$$

3. Atualização das posições

$$\mathbf{r}(p, t + \Delta t) = \mathbf{r}(p, t) + \Delta t \mathbf{vel}(p, t + \Delta t),$$

Existe quem considere esse esquema usado por Provot como semi-implícito [13], pois nele primeiro estimam-se as velocidades do próximo instante $t + \Delta t$ com base na aceleração $\mathbf{acel}(t + \Delta t)$, para depois calcular as novas posições dos pontos em $t + \Delta t$ já usando as novas velocidades estimadas, ou seja, a atualização das posições é feita com dados do instante anterior (termo $\mathbf{r}(t)$) e do atual (termo $\mathbf{vel}(t + \Delta t)$). A Figura 3.4 ilustra graficamente a diferença entre o método explícito de Euler e o usado por Provot [2], considerando uma curva $\mathbf{r}(t)$ correspondente a posição de um ponto qualquer da superfície ao longo do tempo.

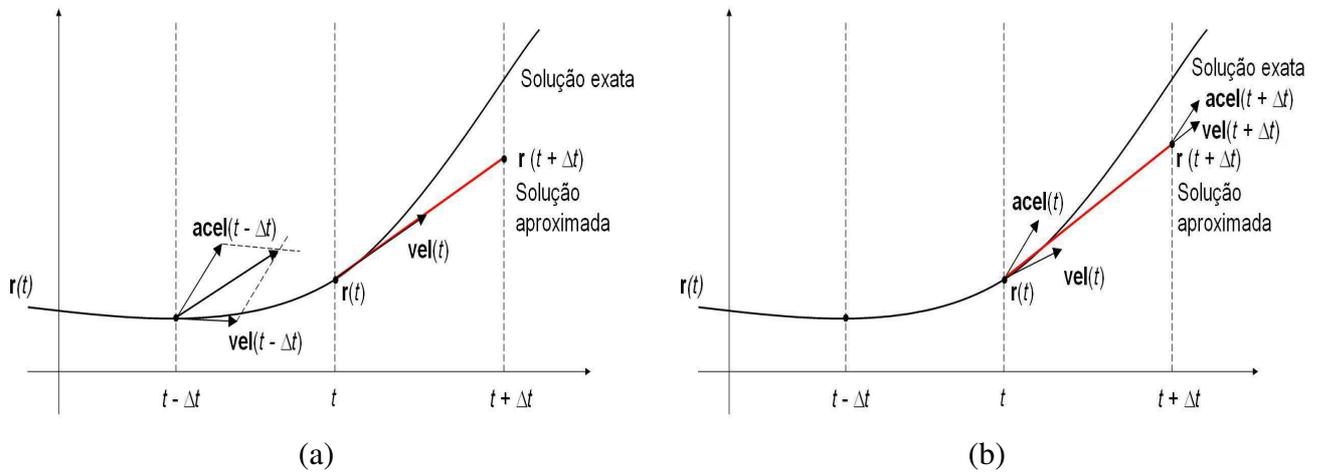


Figura 3.4: (a) Método de Euler explícito e (b) usado por Provot em 1995 [2].

3.2.2 Método de Runge-Kutta de Segunda Ordem

O princípio básico do método de Runge-Kutta é propagar a solução no intervalo $[t, t + \Delta t]$, combinando as estimativas intermediárias feitas nele para atingir uma melhor aproximação da solução entre o passo atual t e o seguinte $t + \Delta t$. Existem diferentes versões do método de Runge-Kutta que são divididas de acordo com a ordem de precisão numérica, sendo as versões de segunda e quarta ordem as mais utilizadas. O método de Runge-Kutta de segunda ordem é também conhecido como método do ponto intermediário, pois ele utiliza o método explícito de Euler para obter uma solução no ponto intermediário $t + \frac{\Delta t}{2}$ e a usa para melhorar a solução do ponto final $t + \Delta t$, em cada iteração.

O método de Runge-Kutta de segunda ordem pode ser descrito com dois passos, conforme o procedimento [29]:

$$\begin{aligned} k_1 &= f(t, x(t)), \\ k_2 &= f\left(t + \frac{\Delta t}{2}, x(t) + \frac{\Delta t}{2} k_1\right), \\ x(t + \Delta t) &= x(t) + \Delta t k_2 + O(\Delta t^3). \end{aligned} \quad (3.17)$$

Em termos de expansão em série de Taylor, pode-se mostrar que o erro é menor do que duas potências de Δt , ou seja, o método tem um erro de truncamento de terceira ordem

$$x(t_0 + \Delta t) = x(t_0) + \Delta t f\left(t + \frac{\Delta t}{2}, x(t_0) + \frac{\Delta t}{2} f(t_0, x(t_0))\right) + O(\Delta t^3).$$

Dessa forma, esse método tem precisão numérica de segunda ordem. Embora seja uma ordem mais preciso que o método explícito de Euler, ele demanda mais cálculos por iteração.

A adaptação do método de Runge-Kutta de segunda ordem para integrar o modelo de tecido ao longo do tempo é feita de forma similar ao método explícito de Euler, entretanto, neste método o primeiro passo não consiste em calcular a aceleração com a Equação 3.14, mas estimar a geometria da superfície no ponto intermediário $t + \frac{\Delta t}{2}$ a cada dois passos subsequentes t e $t + \Delta t$. Então, a geometria da superfície $\mathbf{r}(t + \frac{\Delta t}{2})$ serve como auxiliar para obter a aceleração correspondente a este instante usando a Equação 3.14. Esta aceleração é utilizada para estimar a velocidade do ponto intermediário $t + \frac{\Delta t}{2}$, que é usada para encontrar a posições final de cada ponto da superfície. Por fim, a velocidade final é atualizada com a última aceleração encontrada. O procedimento é descrito pelos seguintes passos:

1. Armazenar $\mathbf{r}(p, t)$ e $\mathbf{vel}(p, t)$ e encontrar $\mathbf{r}(p, t + \frac{\Delta t}{2})$ usando o método explícito de Euler

$$\begin{aligned} k_1 &= \mathbf{vel}(p, t), \\ \mathbf{r}(p, t + \frac{\Delta t}{2}) &= \mathbf{r}(p, t) + \frac{\Delta t}{2} k_1, \end{aligned}$$

2. Calcular uma iteração do modelo de tecido para determinar as acelerações de $\mathbf{r}(t + \frac{\Delta t}{2})$ com a Equação 3.14

$$\mathbf{acel}(p, t + \frac{\Delta t}{2}) = \frac{\mathbf{F}(p, t + \frac{\Delta t}{2}) - \rho \mathbf{vel}(p, t) - \mathbf{f}(p, t + \frac{\Delta t}{2})}{\mu},$$

3. Re-estimar a velocidade do ponto intermediário $t + \frac{\Delta t}{2}$ usando a aceleração $\mathbf{acel}(p, t + \frac{\Delta t}{2})$ do passo 2

$$k_2 = \mathbf{vel}(p, t + \frac{\Delta t}{2}) = \mathbf{vel}(p, t) + \frac{\Delta t}{2} \mathbf{acel}(p, t + \frac{\Delta t}{2}),$$

4. Atualizar a posição final usando a nova velocidade do ponto intermediário obtida no passo 3

$$\mathbf{r}(p, t + \Delta t) = \mathbf{r}(p, t) + \Delta t k_2,$$

5. Atualizar a velocidade final com a aceleração obtida no passo 2

$$\mathbf{vel}(p, t + \Delta t) = \mathbf{vel}(p, t) + \Delta t \mathbf{acel}(p, t + \frac{\Delta t}{2}).$$

Assim, esse procedimento requer a execução de uma iteração adicional (passo 2) da dinâmica do modelo de tecido a cada iteração da solução numérica. Isto é feito porque neste método é preciso

estimar a velocidade em um ponto intermediário. Esta velocidade é estimada com base na aceleração do instante $t + \frac{\Delta t}{2}$ que, por sua vez, é estimada com a Equação 3.14. O gráfico da Figura 3.5 ilustra a aproximação do método de Runge-Kutta de segunda ordem em comparação com a do método explícito de Euler, sobre a curva $r(t)$, que representa a geometria de um ponto da superfície ao longo do tempo. Nele, a linha pontilhada que liga t até $t + \Delta t$ refere-se à solução do método de Euler (passo 1). No ponto intermediário $t + \frac{\Delta t}{2}$, a aceleração é obtida com uma nova iteração do modelo de tecido (passo 2) e usada para estimar a velocidade k_2 (passo 3). Então, a inclinação do vetor que representa esta última velocidade é usada para avançar a solução do instante t até $t + \Delta t$, ilustrada com uma linha vermelha (passo 4).

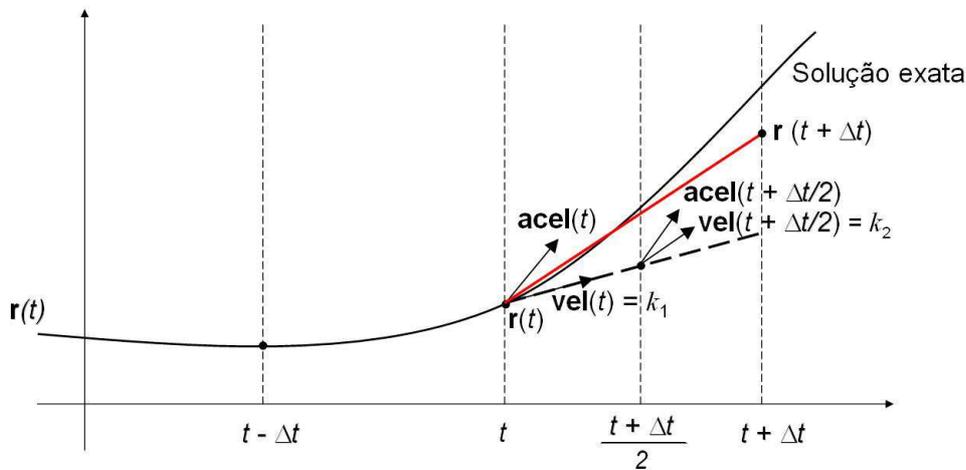


Figura 3.5: Método de Runge-Kutta de segunda ordem.

3.2.3 Método de Runge-Kutta de Quarta Ordem

A outra versão mais conhecida do método de Runge-Kutta é a de quarta ordem, que embora envolva mais cálculos do que a versão de segunda ordem, possui o dobro da precisão numérica (quarta ordem). Este método pode ser descrito pelo seguinte procedimento:

$$\begin{aligned}
 k_1 &= f(t, x(t)), \\
 k_2 &= f\left(t + \frac{\Delta t}{2}, x(t) + \frac{\Delta t}{2} k_1\right), \\
 k_3 &= f\left(t + \frac{\Delta t}{2}, x(t) + \frac{\Delta t}{2} k_2\right), \\
 k_4 &= f\left(t + \Delta t, x(t) + \Delta t k_3\right), \\
 x(t + \Delta t) &= x(t) + \frac{\Delta t}{6} k_1 + \frac{\Delta t}{3} k_2 + \frac{\Delta t}{3} k_3 + \frac{\Delta t}{6} k_4 + O(\Delta t^5).
 \end{aligned} \tag{3.18}$$

De forma similar ao que foi feito na adaptação do método de Runge-Kutta de segunda ordem para integrar o modelo de tecido ao longo do tempo, a versão do método de quarta ordem também precisa gerar novas iterações do modelo para obter a geometria auxiliares e, com elas, as novas estimativas de velocidade nos pontos intermediários do intervalo $[t, t + \Delta t]$. Entretanto, neste método são necessárias três iterações do modelo em uma única iteração da solução numérica, que correspondem às três transições entre os fatores k_1 e k_2 , k_2 e k_3 , e k_3 e k_4 . Os seguintes passos representam este método:

1. Armazenar $\mathbf{r}(p, t)$ e $\mathbf{vel}(p, t)$ e encontrar $\mathbf{r}(p, t + \frac{\Delta t}{2})$ usando o método explícito de Euler

$$\begin{aligned} k_1 &= \mathbf{vel}(p, t), \\ \mathbf{r}(p, t + \frac{\Delta t}{2}) &= \mathbf{r}(p, t) + \frac{\Delta t}{2} k_1, \end{aligned}$$

2. Calcular uma iteração do modelo de tecido para determinar as acelerações de $\mathbf{r}(t + \frac{\Delta t}{2})$ com a Equação 3.14

$$\mathbf{acel}(p, t + \frac{\Delta t}{2}) = \frac{\mathbf{F}(p, t + \frac{\Delta t}{2}) - \rho \mathbf{vel}(p, t) - \mathbf{f}(p, t + \frac{\Delta t}{2})}{\mu},$$

3. Re-estimar a velocidade no ponto intermediário $t + \frac{\Delta t}{2}$ usando a aceleração $\mathbf{acel}(p, t + \frac{\Delta t}{2})$ do passo 2

$$k_2 = \mathbf{vel}(p, t + \frac{\Delta t}{2}) = \mathbf{vel}(p, t) + \frac{\Delta t}{2} \mathbf{acel}(p, t + \frac{\Delta t}{2}),$$

4. Re-estimar a nova posição dos pontos da superfície no ponto intermediário

$$\mathbf{r}(p, t + \frac{\Delta t}{2}) = \mathbf{r}(p, t) + \frac{\Delta t}{2} k_2,$$

5. Re-estimar as acelerações de $\mathbf{r}(t + \frac{\Delta t}{2})$

$$\mathbf{acel}(p, t + \frac{\Delta t}{2}) = \frac{\mathbf{F}(p, t + \frac{\Delta t}{2}) - \rho \mathbf{vel}(p, t + \frac{\Delta t}{2}) - \mathbf{f}(p, t + \frac{\Delta t}{2})}{\mu},$$

6. Re-estimar a velocidade no ponto intermediário $t + \frac{\Delta t}{2}$ usando a aceleração $\mathbf{acel}(p, t + \frac{\Delta t}{2})$ encontrada no passo 5

$$k_3 = \mathbf{vel}(p, t + \frac{\Delta t}{2}) = \mathbf{vel}(p, t) + \frac{\Delta t}{2} \mathbf{acel}(p, t + \frac{\Delta t}{2}),$$

7. Estimar a posição dos pontos em $t + \Delta t$ usando a velocidade obtida no passo 6

$$\mathbf{r}(p, t + \Delta t) = \mathbf{r}(p, t) + \Delta t k_3,$$

8. Estimar as acelerações de $\mathbf{r}(t + \Delta t)$

$$\text{acel}(p, t + \Delta t) = \frac{\mathbf{F}(p, t + \Delta t) - \varrho \text{vel}(p, t + \Delta t) - \mathbf{f}(p, t + \Delta t)}{\mu},$$

9. Estimar a velocidade no ponto $t + \Delta t$ usando a aceleração $\text{acel}(p, t + \Delta t)$ obtida no passo 8

$$k_4 = \text{vel}(p, t + \Delta t) = \text{vel}(p, t) + \Delta t \text{acel}(p, t + \Delta t),$$

10. Estimar a posição final no instante $t + \Delta t$ usando $\mathbf{r}(p, t)$ e os quatro fatores k_1, k_2, k_3 e k_4

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \frac{\Delta t}{6} k_1 + \frac{\Delta t}{3} k_2 + \frac{\Delta t}{3} k_3 + \frac{\Delta t}{6} k_4,$$

11. Atualizar a velocidade final usando $\text{vel}(p, t)$ e a aceleração $\text{acel}(t + \Delta t)$ calculada no passo 8

$$\text{vel}(t + \Delta t) = \text{vel}(t) + \Delta t \text{acel}(t + \Delta t).$$

A Figura 3.6 ilustra a aproximação feita nesse método. Nela, pode-se notar que a linha que avança em direção da solução aproximada está dividida em quatro cores, sendo cada uma delas associada a uma das quatro velocidades k_1, k_2, k_3 e k_4 , que são ponderadas de acordo com a Equação 3.18.

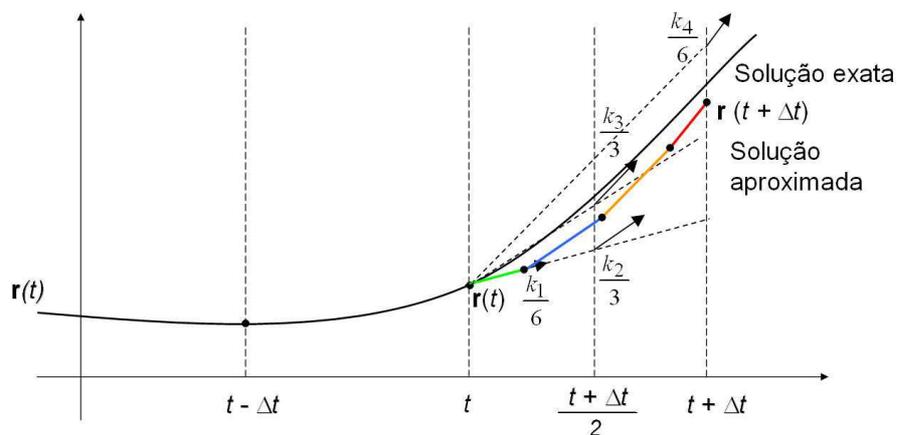


Figura 3.6: Método de Runge-Kutta de quarta ordem.

3.2.4 Método de Verlet

Ao invés de utilizar as informações do intervalo $[t, t + \Delta t]$, com um único passo a partir de t , para aprimorar a aproximação da solução em cada iteração, há técnicas que fazem uso de valores anteriores ao instante t para aumentar a precisão da estimativa. Estas técnicas são conhecidas como métodos de passo múltiplo. Um exemplo desta abordagem é o método de Verlet, bastante difundido na comunidade de Computação Gráfica. A idéia básica deste método consiste em escrever duas expansões em série de Taylor truncadas no elemento de quarta ordem, sendo uma para frente

$$x(t_0 + \Delta t) = x(t_0) + \Delta t x'(t_0) + \frac{\Delta t^2}{2!} x''(t_0) + \frac{\Delta t^3}{3!} x'''(t_0) + O(\Delta t^4), \quad (3.19)$$

e outra para trás

$$x(t_0 - \Delta t) = x(t_0) - \Delta t x'(t_0) + \frac{\Delta t^2}{2!} x''(t_0) - \frac{\Delta t^3}{3!} x'''(t_0) + O(\Delta t^4). \quad (3.20)$$

Somando as equações 3.19 e 3.20, chega-se a expressão

$$x(t_0 + \Delta t) - x(t_0 - \Delta t) = 2x(t_0) + \Delta t^2 x''(t_0) + O(\Delta t^4),$$

que reescrita com o termo $x(t_0 - \Delta t)$ do lado direito torna-se a expressão convencional da técnica de Verlet

$$x(t_0 + \Delta t) = 2x(t_0) - x(t_0 - \Delta t) + \Delta t^2 x''(t_0) + O(\Delta t^4). \quad (3.21)$$

Dessa forma, o método de Verlet requer o armazenamento de dois estados da superfície e atinge uma precisão numérica de terceira ordem por possuir um termo de erro de truncamento de quarta ordem $O(\Delta t^4)$ em sua formulação. Uma das vantagens em usar este método é atingir uma maior precisão numérica em relação ao método de Euler sem aumentar significativamente o número de cálculos, como acontece nos métodos de Runge-Kutta de segunda e quarta ordem. Na verdade, como em sua formulação são utilizadas duas geometrias da superfície, uma relativa ao instante anterior $\mathbf{r}(t - \Delta t)$ e outra ao atual $\mathbf{r}(t)$, o método permite atingir uma solução numérica sem o cálculo direto da velocidade, sendo ela calculada indiretamente pela variação das posições da superfície em instantes diferentes, o que resultaria em menos cálculos que o método de Euler. Contudo, como no modelo de tecido a aceleração de cada ponto é obtida com a Equação 3.14, a qual utiliza a velocidade, esta vantagem do método é descartada, sendo preciso calcular explicitamente a velocidade em cada iteração. Assim, considerando como condição inicial que $\mathbf{r}(t_0) = \mathbf{r}(t_1)$, o procedimento pode ser descrito pelo seguintes passos:

1. Cálculo da aceleração

$$\text{acel}(p, t) = \frac{\mathbf{F}(p, t) - \varrho \text{vel}(p, t) - \mathbf{f}(p, t)}{\mu},$$

2. Atualização das posições

$$\mathbf{r}(p, t + \Delta t) = 2 * \mathbf{r}(p, t) - \mathbf{r}(p, t - \Delta t) + \Delta t^2 \text{acel}(t),$$

3. Atualização das velocidades

$$\text{vel}(p, t) = \frac{\mathbf{r}(p, t + \Delta t) - \mathbf{r}(p, t - \Delta t)}{2\Delta t}. \quad (3.22)$$

A Figura 3.7 ilustra a solução aproximada do método de Verlet. Apesar da velocidade ser calculada em cada iteração da solução numérica (passo 3), ela é usada somente no cálculo da aceleração (passo 1). Esta última é usada então no passo 2 para encontrar a nova posição $\mathbf{r}(t + \Delta t)$ de cada ponto da superfície juntamente com $\mathbf{r}(t)$ e $\mathbf{r}(t - \Delta t)$.

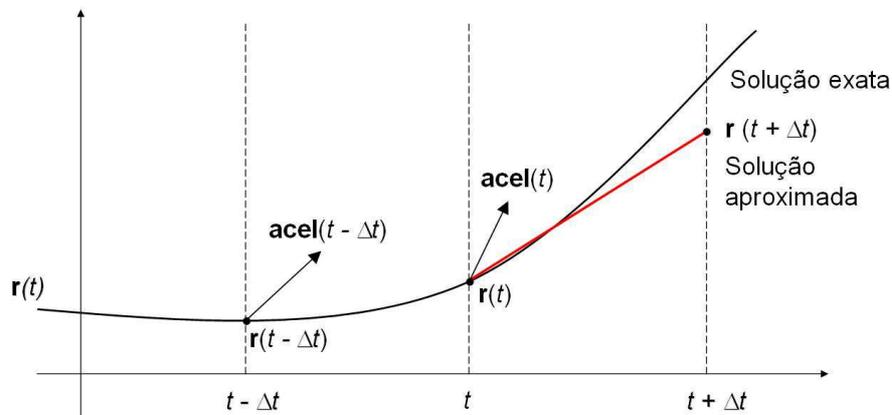


Figura 3.7: Método de Verlet.

3.2.5 Método *Leapfrog*

Quando se trata de equações diferenciais ordinárias de segunda ordem, como a Equação 1.1, pode ser vantajoso o uso do esquema de integração conhecido como *Leapfrog*, no qual as velocidades são estimadas para o ponto intermediário $t + \frac{\Delta t}{2}$ com o uso da aceleração obtida no instante t pela

Equação 3.14, de forma que

$$\mathbf{vel}(p, t + \frac{\Delta t}{2}) = \mathbf{vel}(p, t - \frac{\Delta t}{2}) + \Delta t \mathbf{acel}(t). \quad (3.23)$$

A posição dos pontos no instante $t + \Delta t$ é então calculada com base nesta velocidade

$$\mathbf{r}(p, t + \Delta t) = \mathbf{r}(p, t) + \Delta t \mathbf{vel}(p, t + \frac{\Delta t}{2}). \quad (3.24)$$

Dessa forma, a atualização da posição e da velocidade dos pontos da superfície é feita com mesma formulação do método explícito de Euler, porém com cada uma delas sendo atualizada em instantes diferentes, separados de $\frac{\Delta t}{2}$, de forma que os cálculos das equações 3.23 e 3.24 vão se saltando ao longo do tempo. Como as velocidades são calculadas sempre no meio de cada passo de tempo, de maneira similar ao método de Runge-Kutta de segunda ordem, o método *Leapfrog* possui uma precisão numérica de segunda ordem, com a vantagem de ser inversível, o que garante a conservação de energia ao longo do tempo. Ao adaptar este método para integrar o modelo de tecido ao longo do tempo, diferentemente dos demais vistos até aqui, é preciso, no instante inicial t_0 , da velocidade em $t_0 + \frac{\Delta t}{2}$ para que a velocidade $\mathbf{vel}(p, t - \frac{\Delta t}{2})$ sempre exista a partir de t_1 . Para estimar a velocidade $\mathbf{vel}(p, t_0 + \frac{\Delta t}{2})$ optou-se por usar o método do ponto intermediário, somente na primeira iteração a partir do instante t_0 , de forma que as demais iterações podem ser descritas pelos passos:

1. Cálculo da aceleração

$$\mathbf{acel}(p, t) = \frac{\mathbf{F}(p, t) - \varrho \mathbf{vel}(p, t - \frac{\Delta t}{2}) - \mathbf{f}(p, t)}{\mu},$$

2. Atualização das velocidades

$$\mathbf{vel}(p, t + \frac{\Delta t}{2}) = \mathbf{vel}(p, t - \frac{\Delta t}{2}) + \Delta t \mathbf{acel}(p, t).$$

3. Atualização das posições

$$\mathbf{r}(p, t + \Delta t) = \mathbf{r}(p, t) + \Delta t \mathbf{vel}(p, t + \frac{\Delta t}{2}),$$

Esse procedimento é muito parecido com o método de Euler usado por Provot [2], pois após a aceleração $\mathbf{acel}(t)$ ser encontrada pela Equação 3.14, a velocidade e a posição são atualizadas usando intervalos de Δt . A Figura 3.8 ilustra o gráfico das aproximações feitas com este procedimento. Pode-se notar que as posições e acelerações são calculadas em instantes espaçados de $\frac{\Delta t}{2}$ em relação

o modelo apresentava muitos problemas de instabilidade numérica relacionados ao tamanho do passo de tempo utilizado na integração temporal.

Também em 1995, Volino et al. publicaram um modelo baseado na mecânica de partículas que utiliza o método explícito de Runge-Kutta de segunda ordem para simular tecidos [30]. Neste trabalho os autores descreveram um modelo de superfície deformável que aborda a colisão com outros objetos e auto-colisões, que além de ter um bom desempenho computacional, apresenta menores problemas de instabilidade em relação ao modelo de Provot. Eles utilizaram a segunda lei de Newton para encontrar a aceleração e a lei de Hooke em adição às expressões angulares para encontrar a força interna de cada partícula. O trabalho de Volino et al. ainda possuía problemas de instabilidade, que eram resolvidos de uma forma artificial, diminuindo a precisão física do modelo. Os autores monitoram constantemente as variações da energia mecânica local para medir a instabilidade do sistema e, uma vez detectado seu aumento na região vizinha de alguma partícula, uma distribuição artificial de energia cinemática é feita nos elementos envolvidos, acelerando a propagação da perturbação. Outra dificuldade deste trabalho é o controle dos parâmetros mecânicos do material do tecido, que são definidos com a atribuição de valores empíricos para os coeficientes de rigidez das molas.

Realmente, a utilização de métodos explícitos para integrar ao longo do tempo um modelo de tecido limita o tamanho do passo de tempo para garantir estabilidade numérica, entretanto, os problemas de instabilidade não decorrem exclusivamente do método empregado. O modelo massa-mola, utilizado por Provot [2] e Volino et al. [30], não consegue representar adequadamente alguns comportamentos peculiares dos tecidos. Por exemplo, para simular um tecido que cai no chão e rapidamente se estabiliza, de forma a não apresentar mais nenhum movimento, é preciso inserir fatores de amortecimento nas molas que formam o tecido para que a energia interna armazenada nelas não gere um comportamento oscilatório, que produz um efeito artificial não desejado.

Em 1996, Eberhardt et al. utilizaram o método de Runge-Kutta de quarta ordem sobre um modelo de partículas, baseado no modelo de Breen et al. [26], para encontrar a nova geometria da superfície do tecido em cada instante de tempo [15]. Neste trabalho foi implementado o método de Runge-Kutta com passos de tempo adaptativos e também o método de Bulirsch-Stoer, chegando à conclusão que o método de Runge-Kutta é muito mais eficiente computacionalmente. Para o modelo de tecido, os autores estenderam o modelo de Breen et al. introduzindo técnicas para medir as forças internas com maior precisão e considerando propriedades específicas de tecidos, como comportamento anisotrópico e histerese, além de incrementá-lo para produzir a dinâmica de tecidos considerando a resistência do ar, vento e fricção de superfície.

Com o destaque das vantagens do método de Runge-Kutta no trabalho de Eberhardt [15], Volino e Thalmann [31], em 1997, apresentaram um trabalho utilizando o método de Runge-Kutta descrito em [29]. Este método é uma adaptação do método de Runge-Kutta de quarta ordem que atinge um

estado cinemático de todos os vértices do objeto simulado com uma estimativa de erro cometido neste estado. Esta estimativa é então usada como um fator de amortecimento que coloca o estado do modelo na menor deformação possível dentro do intervalo de erro considerado. Além disso, os autores realizaram uma correção da posição e velocidade dos vértices, após o processo de cálculo da dinâmica, que é feita para cada aresta comparando e corrigindo a posição de seus vértices extremos. Como resultado, os autores obtiveram um sistema robusto de simulação de tecidos, que previne o aumento da energia de deformação para tratar imprecisões numéricas, fazendo com o que modelo raramente resulte em uma situação de instabilidade numérica, na qual os pontos do tecido apresentam grandes deslocamentos e deixam de formar uma superfície regular. O modelo de tecido utilizado nesta dissertação, proposto por de Melo, considera a condição de compatibilidade geométrica de Weingarten [5]. Com isso, a superfície tende a preservar naturalmente a sua regularidade, evitando etapas de pós-processamento como as feitas por Volino e Thalmann.

Após o trabalho de Baraff e Witkin [14] em 1998, o uso de métodos implícitos para simular tecidos ganhou grande destaque, fazendo com que o emprego dos métodos explícitos para este propósito diminuísse. Contudo, em 2001, Oshita e Makinouchi apresentaram um trabalho utilizando o método explícito de Runge-Kutta de quarta ordem juntamente com um modelo massa-mola para produzir simulações de tecidos em tempo real [32]. Neste trabalho os autores utilizavam uma malha esparsa de partículas, que tinha no máximo cem pontos, e dividiram a dinâmica de tecidos em duas etapas: a parte de geração da superfície, que é feita com o método de Runge-Kutta, e a parte de suavização, a qual é processada pela combinação da modelagem das forças internas com técnicas geométricas de subdivisão em triângulos, fazendo com que a superfície final tenha uma resolução em torno de cinco vezes maior e apresente algumas dobras criadas nesta segunda etapa.

Também em 2001, Volino e Thalmann publicaram um trabalho que comparou a eficiência de diferentes métodos numéricos para realizar a simulação de tecidos [33]. Eles consideraram os métodos de Runge-Kutta de segunda e quarta ordem, o de Euler para trás e o de Rosenbrook. Estes métodos foram implementados sobre um sistema que permitia usar duas representações mecânicas para os tecidos: um modelo massa-mola simplificado e um modelo de superfície elástica baseado na mecânica de partículas. Segundo os autores, esta última representação do tecido tratava-se de uma abordagem completa de superfície elástica que permitia a simulação de elasticidade. Contudo, neste trabalho os autores não apresentam as formulações dos modelos usados nem indicam outros equivalentes na literatura, descrevendo apenas as duas representações sucintamente e de forma literal. Adicionalmente, os autores afirmam que o curvamento do modelo, embora tenha sido implementado, não é levado em consideração, como também consideram que as leis de elasticidade sejam computadas como interações entre cada três vértices de um triângulo que forma a superfície, refletindo um comportamento mecânico assumido como linear [33]. Assim, o modelo elástico considerado faz muitas simplifi-

cações, não possuindo a mesma precisão física dos modelos baseados na mecânica de contínuos, como o proposto por de Melo [5], utilizado nesta dissertação.

Ainda em 2001, Jakobsen publicou na rede um *site* que descreve os algoritmos desenvolvidos para o jogo *Hitman Codename 47*, da empresa *IO Interactive's Game* [34]. Neste trabalho o autor utiliza o esquema de integração de Verlet em conjunto com um modelo de partículas para processar o movimento de tecidos durante o jogo. O método modela corpos rígidos como partículas com restrições e conta com algoritmos que tratam colisões e penetrações por projeções, usando um método de relaxação para equilibrar o sistema. O resultado é um simulador de tecidos muito eficiente computacionalmente, pelo uso do método numérico de Verlet, e estável numericamente, em função da relaxação das forças internas para garantir um estado de equilíbrio. Estes dois pontos, bom desempenho e estabilidade, são cruciais em uma simulação de tecidos dentro de um jogo que, na prática, serve apenas para proporcionar-lhe um maior realismo. Também voltado aos jogos, uma outra implementação do esquema de Verlet que considera passos de tempo adaptativos foi descrita em [35]. Entretanto, em nenhum destes trabalhos o enrugamento de tecidos foi tratado.

Em 2003, Bridson et al. [11] propuseram um esquema de integração temporal que mistura formulações implícitas com explícitas em conjunto com um modelo de tecido sofisticado, baseado na mecânica de partículas, para simular a dinâmica de tecidos. Eles utilizaram a integração explícita no cálculo das forças elásticas, as quais independem da velocidade, e a integração implícita na obtenção das forças de amortecimento, que dependem da velocidade. Desta forma, os autores combinaram o método explícito de *Leapfrog*, cuja precisão numérica é de segunda ordem, para atualizar as posições dos pontos da superfície do tecido, enquanto um método implícito de regra trapezoidal, também de precisão de segunda ordem, era usado para a atualização das velocidades dos pontos. Ao contrário deste trabalho, no modelo proposto por de Melo, as forças de amortecimento não são necessárias para amortecer forças residuais que podem levar o modelo a um estado instável [5].

Também em 2003, Hauth publicou um trabalho voltado para a investigação de métodos numéricos aplicados especificamente para a simulação de tecidos [36]. Neste trabalho o autor descreve inicialmente diferentes tipos de discretização espacial e assume um modelo de tecido do tipo massa-mola, para então apresentar diversos métodos numéricos comparando os resultados de cada método com a solução analítica para funções paramétricas. Entre os métodos investigados, estão os métodos explícitos de Euler, do ponto intermediário, de Runge-Kutta de quarta ordem e o método chamado de Verlet, que segundo Hauth é também chamado de *Leapfrog*, embora este método que o autor se referiu corresponda ao método de *Leapfrog* apresentado na Subseção 3.2.5, e não ao de Verlet apresentado na Subseção 3.2.4.

Apesar de focado para o uso de métodos numéricos aplicados em simulação de tecidos, Hauth não chegou a exemplificar ou comparar esses métodos com imagens de simulações que utilizam diferentes

implementações de soluções numéricas sobre o modelo massa-mola considerado. Após a explicação e a comparação dos métodos numéricos com as soluções analíticas, Hauth analisa a estabilidade, eficiência e precisão de cada método, chegando a conclusão de que é praticamente impossível afirmar qual é o melhor método para uma dada aplicação sem que todos esses métodos sejam testados para o objetivo desejado [36]. O autor estende a mesma conclusão para a eficiência dos métodos investigados, pois enquanto geralmente os métodos implícitos requerem um maior tempo computacional por iteração, com eles é possível utilizar um passo de tempo mais largo em relação aos explícitos. Por fim, ele apresenta um digrama para auxiliar a tomada de decisão sobre qual método usar em determinadas aplicações, distinguindo-as em sistemas lineares e não lineares. Neste sentido, avaliou-se nesta dissertação a adequabilidade dos principais métodos explícitos para integrar ao longo do tempo o modelo proposto por de Melo [5].

Um ano mais tarde, em 2004, Natsupakpong divulgou através de um *site* um estudo sobre métodos de integração numérica aplicados a simulação de objetos deformáveis, utilizando a modelagem massa-mola [37]. Neste trabalho a autora compara os métodos explícitos de Euler, do ponto intermediário e o de Runge-Kutta de quarta ordem, considerando também o método semi-implícito de Euler (usado por Provot [2]) e o método implícito de Euler. A comparação feita consistia em encontrar os maiores passos de tempo para cada método que resultava em uma simulação estável, bem como comparar os tempos totais gastos para processar determinadas simulações. O método semi-implícito de Euler foi o que atingiu os melhores resultados neste estudo.

Finalizado este resumo sobre os principais trabalhos relacionados à simulação de tecidos que utilizam os métodos explícitos apresentados na Seção 3.2, é possível mais uma vez destacar a lacuna existente neste ramo de pesquisa, que consiste em unir métodos explícitos com um modelo baseado na mecânica de contínuos. Um dos principais motivos da existência desta lacuna deve-se ao sucesso da combinação oposta: a de usar integrações implícitas ou semi-implícitas com modelos baseados na mecânica de partículas. Este sucesso foi impulsionado ao longo da última década por trabalhos que salientaram a vantagem em unir as duas abordagens e, especialmente nos trabalhos mais recentes como [10, 11, 6], os modelos de tecido baseados na mecânica de partículas são agregados a sofisticados esquemas numéricos que misturam formulações explícitas e implícitas e ainda algoritmos avançados de tratamento de colisões e auto-colisões. Contudo, alguns dos problemas presentes nos trabalhos comentados nesta seção não estão necessariamente ligados ao método numérico empregado, mas ao modelo de tecido considerado. Desta forma, esta dissertação investiga estes mesmos métodos explícitos aplicados ao modelo de tecido proposto por de Melo, que consegue contornar tais problemas [5].

3.4 Resultados

Nesta seção são mostrados os resultados da implementação dos métodos apresentados na Seção 3.2 sobre o modelo de tecido proposto por de Melo [5]. Os resultados da implementação dos métodos numéricos são apresentados em termos de estabilidade, eficiência e precisão. Para medir estes quesitos é usado um conjunto de seis simulações de tecidos, que possuem parâmetros de material e restrições diferentes e atingem determinadas formas em um estado de equilíbrio considerado, conforme explicado na Subseção 3.4.1. As subseções seguintes apresentam os resultados de cada método separadamente, na mesma ordem em que eles foram apresentados na Seção 3.2. Nestes resultados, primeiro o valor do passo de tempo é testado até que se atinja o valor máximo capaz de alcançar de forma estável o estado de equilíbrio desejado e em cada simulação, sendo encontrado também o número de iterações necessárias para isso. Em seguida, este passo de tempo máximo é usado para medir a eficiência computacional: de uma forma relativa, que consiste no tempo gasto para processar as iterações necessárias para atingir o estado considerado; e de uma forma absoluta, tomando o tempo que o método realiza uma única iteração. Por fim, as imagens do estado de equilíbrio das simulações são mostradas para que na próxima seção a qualidade visual obtida seja discutida.

3.4.1 Simulações de Teste

As seis simulações de tecidos usadas para obter os resultados da implementação de cada método numérico serão chamadas de S_1 até S_6 , estando elas em ordem crescente de número de pontos que formam a superfície. Em todas as simulações é considerado um estado de equilíbrio, seja ele quando não ocorrem mais alterações na forma do tecido entre dois instantes de tempo consecutivos, ou quando o tecido apresenta um comportamento que repete as mesmas deformações, como as dobras que se formam e desaparecem em uma bandeira hasteada ao vento. Para a definição destes estados são consideradas situações reais, que foram fotografadas e estão ilustradas na Figura 3.9.

A simulação S_1 é uma malha quadrada 10×10 presa pelos seus dois vértices extremos superiores, posicionada na horizontal no instante t_0 e caindo de acordo com a força da gravidade ao longo do tempo, estabilizando na posição da Figura 3.9(a). A simulação S_2 é similar à S_1 , diferindo por ser uma malha 20×20 e por ter um ponto extremo fixo ao invés de dois, sendo considerado como estado de equilíbrio a Figura 3.9(b). S_3 é uma malha 30×30 sem pontos presos que cai horizontalmente de acordo com a gravidade, simulando o caimento de um tecido sobre um banco de praça, assumindo o formato apresentado na Figura 3.9(c). A simulação S_4 é uma malha 40×40 , suspensa pelos seus dois pontos extremos laterais, sendo deformada pela força do vento como uma bandeira, conforme a Figura 3.9(d), que foi obtida com o auxílio de dois ventiladores em direções opostas. Como nesta situação o tecido não pára de se movimentar, é considerado como estado de equilíbrio quando o

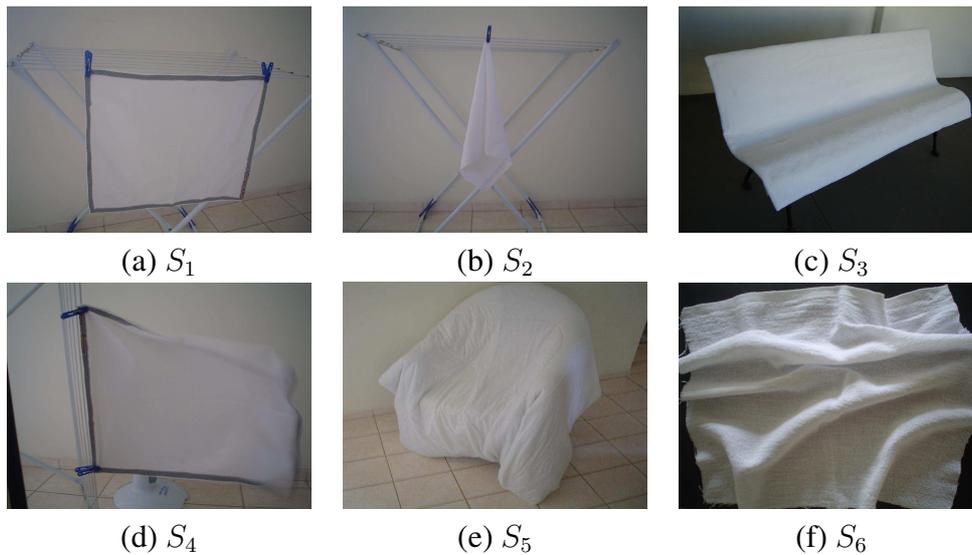


Figura 3.9: Imagens reais de um tecido: (a) suspenso por 2 pontos; (b) suspenso por 1 ponto; (c) cobrindo um banco; (d) hasteado; (e) cobrindo uma poltrona e (f) comprimido.

tecido, após cair com a gravidade e sofrer a força do vento, consegue se manter na mesma linha horizontal, perpendicular a haste em que ele está preso. S_5 é uma malha 50×50 aberta que, sob a força da gravidade, cai sobre uma poltrona, como mostra a Figura 3.9(e). Por último, S_6 é uma malha 60×60 deformada somente por forças de compressão tangenciais ao plano da superfície inicial do tecido, cujo estado de equilíbrio considerado é ilustrado na Figura 3.9(f).

Assim, a partir de um plano que representa o tecido, os parâmetros e as restrições das simulações de teste determinaram o estado final de equilíbrio do tecido. As superfícies iniciais de todas as simulações, com exceção de S_4 , estão ilustradas na Figura 3.10(a), possuindo apenas áreas e resoluções espaciais diferentes. A simulação S_4 parte de uma superfície inicial posicionada conforme a Figura 3.10(b).

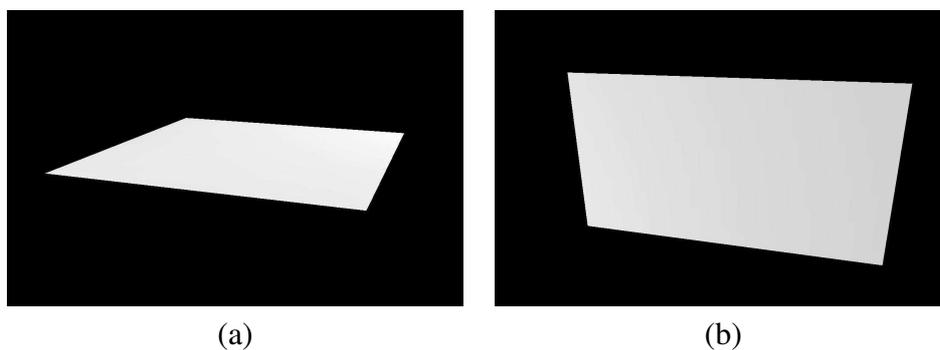


Figura 3.10: Superfície inicial das simulações: (a) S_1 , S_2 , S_3 , S_5 e S_6 ; e (b) S_4 .

Esses seis estados de equilíbrio representados pelas fotografias são usados como referência para

as simulações de teste objetivando medir a estabilidade numérica, a eficiência computacional e a qualidade das imagens produzidas pelos diferentes métodos numéricos. A palavra estabilidade é usada no contexto de simulações computacionais com diferentes interpretações. Neste trabalho, ela refere-se à estabilidade numérica associada à capacidade do método em produzir uma simulação de forma estável, evitando o problema de alguns pontos da superfície sofrerem deslocamentos muito grandes e deixarem de formar uma superfície regular. Este problema está diretamente ligado ao tamanho do passo de tempo usado na integração explícita ao longo do tempo. Valores de Δt maiores do que um certo limiar, resultam neste tipo de problema. Entretanto, devido às diferentes formulações dos métodos explícitos apresentados na Seção 3.2, é possível que haja diferença no limiar de cada método para manter as simulações de teste estáveis.

Diante dessa possível diferença no valor máximo do passo de tempo entre os métodos implementados, a eficiência computacional deve ser medida de diferentes maneiras. O primeiro modo é avaliando qual dos métodos processa o mesmo tempo de simulação da forma mais rápida, comparando os tempos obtidos. A segunda forma é medindo quanto tempo cada método demora para atingir os estados de equilíbrio citados, verificando quantas iterações são necessárias para isso. Neste teste o valor máximo do passo de tempo influencia diretamente o número de iterações necessárias para atingir o equilíbrio, permitindo, por exemplo, que se chegue em um estado de equilíbrio em menos tempo e iterações com o método de Runge-Kutta do que com o método de Euler, mesmo com Euler processando cada iteração da dinâmica de tecidos de forma muito mais rápida. Para avaliar a precisão do método são utilizados os resultados visuais obtidos, comparando-os com as fotos da Figura 3.9.

Desta forma, para todos os métodos, o primeiro teste consiste em verificar qual é o maior passo de tempo que ele consegue alcançar esses estados de equilíbrio de maneira estável para cada simulação separadamente. Este primeiro resultado é usado para discutir a estabilidade numérica do método e também sua eficiência de uma forma relativa, analisando quantas iterações são necessárias e qual é o tempo gasto para atingir esses estados. Em seguida, a eficiência é medida de forma absoluta, que consiste em coletar os tempos em que cada método processa uma única iteração¹. Esta etapa pode ser feita com qualquer passo de tempo que mantenha a simulação estável, pois o intuito é fixar um valor de referência para que os tempos de cada método possam ser comparados. Por último, as imagens obtidas dos estados de equilíbrio são mostradas para que se verifique quais são as diferenças no aspecto visual. Computacionalmente, um tecido é considerado que atingiu o seu estado de equilíbrio quando o deslocamento de todos os seus pontos entre duas iterações subsequentes for desprezível.

¹Uma iteração do método não corresponde necessariamente a um quadro da animação.

3.4.2 Método de Euler

Como a Subseção 3.2.1 apresentou duas versões do método de Euler, testes preliminares foram feitos para decidir qual delas é a mais apropriada na comparação com os outros métodos explícitos. Para tanto, o teste do maior passo de tempo possível foi testado nas duas versões e a usada por Provot, a qual atualiza os pontos da superfície em $t + \Delta t$ já com a velocidade atualizada em $t + \Delta t$, consegue usar um valor de Δt em torno de três ordens maior em relação a outra versão. Assim, os resultados desta subseção, bem como os comentários do método de Euler feitos no restante deste capítulo, referem-se à versão de Provot [2]. É interessante iniciar a exposição dos resultados pelo método de Euler, pois ele, na teoria, é o que possui maior desempenho computacional e menor precisão numérica. Todos os tempos de processamento desta seção foram obtidos em um processador AMD 64 de 3.2 GHz equipado com 2 GB de memória RAM e com uma placa de vídeo 7800GS.

O primeiro teste executado é referente à verificação do maior passo de tempo que este método suporta para alcançar de maneira estável os estados de equilíbrio de cada simulação, mostrados na Seção 3.4.1. Começando com um passo de tempo pequeno, $\Delta t = 0.001s$, e o incrementando sempre com este mesmo valor, é possível encontrar o limiar da estabilidade numérica em cada simulação. Uma vez encontrado este valor, ele é usado para atingir os estados de equilíbrio e verificar quantas iterações foram necessárias para isto. A Tabela 3.1 sintetiza os resultados.

Tabela 3.1: Passos de tempo máximos e iterações necessárias do método de Euler.

Simulação	Δt máximo (s)	Número de iterações
S_1	0.012	1150
S_2	0.004	2050
S_3	0.003	3650
S_4	0.002	1850
S_5	0.002	10000
S_6	0.002	2500

Com a Tabela 3.1 pode-se notar que os valores de Δt obtidos são próximos, com exceção de S_1 , que consegue utilizar um valor aproximadamente uma ordem maior que os demais. Isto ocorre porque as simulações de testes foram criadas de acordo com a variação de curvatura que o tecido sofre. Como para calcular a força interna são necessárias as quantidades geométricas diferenciais da métrica e curvatura da superfície e, como a métrica dos tecidos varia pouco, considera-se principalmente as mudanças de curvatura, ou seja, as variações dos vetores normais são as principais quantidade geométricas que governam o comportamento dinâmico do tecido. Assim, como a simulação S_1 é a que possui menor suavidade na variação de sua curvatura, consegue-se utilizar um valor de Δt maior. O número de iterações necessárias na terceira coluna desta tabela mostra quantas vezes a solução numérica precisa ser repetida, com o maior Δt encontrado, para que a superfície inicial do tecido

avance até o estado de equilíbrio o mais rápido possível e de forma estável. Este valor é então usado para medir o tempo que o método demora para atingir os estados considerados, conforme a coluna T_e da Tabela 3.2. A coluna desta tabela rotulada como T_i , corresponde ao tempo gasto para executar uma iteração de cada simulação. Desta forma, têm-se medidas de tempo absolutas, que correspondem aos tempos da terceira coluna, tomados sempre durante o intervalo de uma única iteração; e medidas de tempo relativas, que correspondem aos valores da quarta coluna, gerados com um número variável de iterações para se atingir o estado de equilíbrio em cada simulação. Em outras palavras, o tempo T_e equivale ao tempo T_i multiplicado pelo número de iterações necessárias da Tabela 3.1. As imagens obtidas pelo método de Euler são ilustradas na Figura 3.11.

Tabela 3.2: Tempos de processamento do método de Euler.

Simulação	Número de Vértices	T_i (s)	T_e (s)
S_1	100	0.0018	2
S_2	400	0.0072	15
S_3	900	0.0165	59
S_4	1600	0.0301	59
S_5	2500	0.0493	8 : 13
S_6	3600	0.071	2 : 58

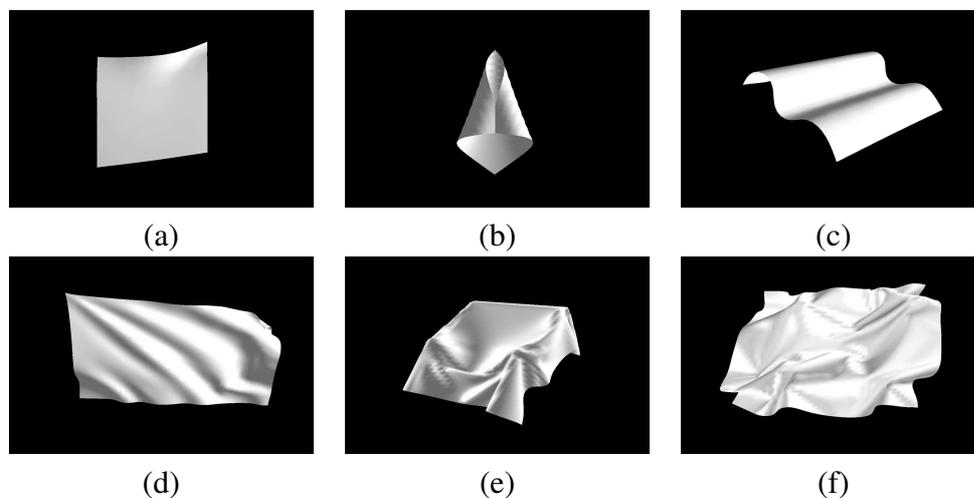


Figura 3.11: Resultados visuais do método de Euler.

3.4.3 Método de Runge-Kutta de Segunda Ordem

De forma similar a Euler, o primeiro teste feito com a implementação do método de Runge-Kutta de segunda ordem é verificar os maiores valores de Δt que atingem os estados de equilíbrio de maneira estável e quantas iterações são necessárias para isto. A Tabela 3.3 mostra os resultados

obtidos. O segundo teste consiste em medir os tempos de processamento gastos nas simulações de uma única iteração, que é o tempo total T_i , e o gastos até atingir o equilíbrio, tempo T_e , conforme a Tabela 3.4. A Figura 3.12 ilustra os resultados visuais.

Tabela 3.3: Passos de tempo máximos e iterações necessária de Runge-Kutta de segunda ordem.

Simulação	Δt máximo (s)	Número de iterações
S_1	0.012	1500
S_2	0.004	2350
S_3	0.003	3800
S_4	0.002	2000
S_5	0.002	11000
S_6	0.002	4550

Tabela 3.4: Tempos de processamento do método de Runge-Kutta de segunda ordem.

Simulação	Número de Vértices	T_i (s)	T_e (s)
S_1	100	0.0019	2
S_2	400	0.0077	18
S_3	900	0.0181	1 : 08
S_4	1600	0.0340	1 : 07
S_5	2500	0.0559	10 : 15
S_6	3600	0.0813	6 : 12

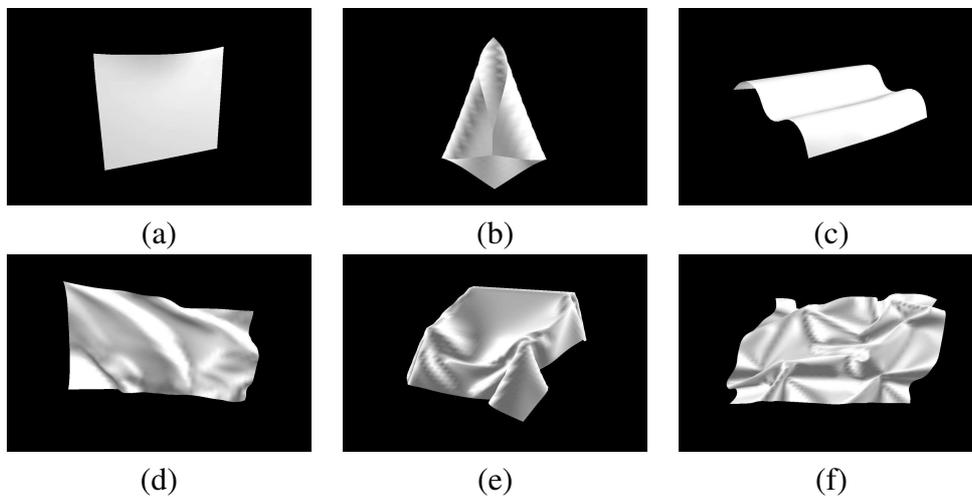


Figura 3.12: Resultados visuais do método de Runge-Kutta de segunda ordem.

3.4.4 Método de Runge-Kutta de Quarta Ordem

A implementação do método de Runge-Kutta de quarta ordem é a que possui o maior número de cálculos, sendo executadas três iterações da dinâmica interna do tecido em cada iteração da solução

numérica. Desta forma, é esperado que os tempos totais sejam superiores aos demais métodos. A Tabela 3.4.4 apresenta os valores máximos de Δt e o número de iterações necessárias para atingir o equilíbrio em cada simulação. A Tabela 3.4.4 mostra os resultados do teste de desempenho computacional e a Figura 3.13 ilustra os resultados visuais obtidos.

Tabela 3.5: Passos de tempo máximos e iterações necessárias de Runge-Kutta de quarta ordem.

Simulação	Δt máximo (s)	Número de iterações
S_1	0.012	1050
S_2	0.005	1750
S_3	0.003	3750
S_4	0.002	1750
S_5	0.002	10000
S_6	0.002	2450

Tabela 3.6: Tempos de processamento do método de Runge-Kutta de quarta ordem.

Simulação	Número de Vértices	T_i (s)	T_e (s)
S_1	100	0.0021	2
S_2	400	0.0087	16
S_3	900	0.0209	1 : 18
S_4	1600	0.0361	1 : 05
S_5	2500	0.0613	10 : 13
S_6	3600	0.0846	3 : 38

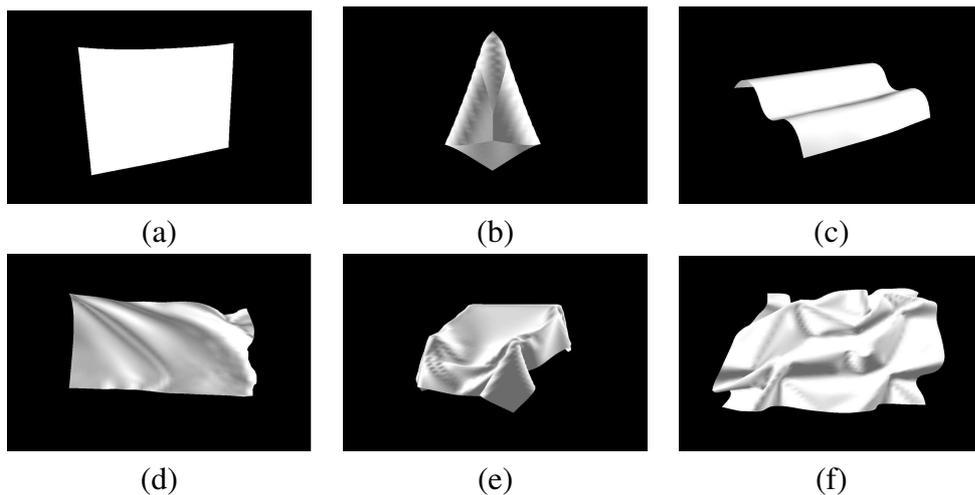


Figura 3.13: Resultados visuais do método de Runge-Kutta de quarta ordem.

3.4.5 Método de Verlet

A implementação do método de Verlet difere das demais feitas até aqui porque precisa de duas geometrias $\mathbf{r}(t_0)$ e $\mathbf{r}(t_1)$ para iniciar a dinâmica de tecidos. Assim, nesta implementação usou-se a mesma superfície inicial para os instantes t_0 e t_1 , pois como são as forças externas que deformam o tecido, esta consideração não afeta sua dinâmica. A Tabela 3.7 apresenta os resultados de Δt , a Tabela 3.8 os resultados de desempenho computacional e a Figura 3.14 os resultados visuais.

Tabela 3.7: Passos de tempo máximos e iterações necessárias do método de Verlet.

Simulação	Δt máximo (s)	Número de iterações
S_1	0.012	1150
S_2	0.004	1250
S_3	0.003	3200
S_4	0.002	1600
S_5	0.002	10000
S_6	0.002	2500

Tabela 3.8: Tempos de processamento do método de Verlet.

Simulação	Número de Vértices	T_i (s)	T_e (s)
S_1	100	0.0018	2s
S_2	400	0.0076	9s
S_3	900	0.0173	55s
S_4	1600	0.0326	59s
S_5	2500	0.0508	8 : 38s
S_6	3600	0.0727	3 : 03s

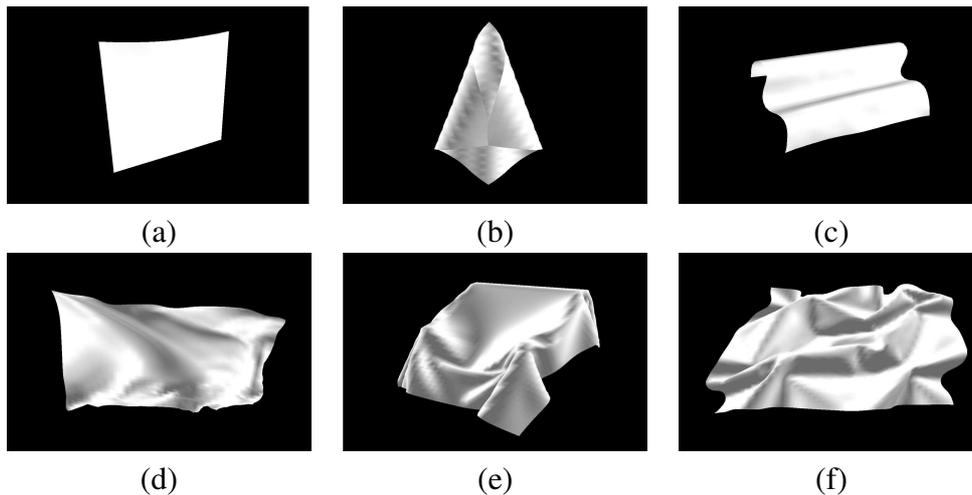


Figura 3.14: Resultados visuais do método de Verlet.

3.4.6 Método *Leapfrog*

A implementação do método *Leapfrog* é muito similar a de Euler, diferindo por uma estimativa da velocidade em $t_0 + \frac{\Delta t}{2}$ de cada ponto da superfície. Esta estimativa é feita pelo método de Runge-Kutta de segunda ordem e, partir dela, o método *Leapfrog* atualiza posições e velocidades da mesma forma explícita usada em Euler, porém com as posições e velocidades sendo atualizadas em instantes espaçados de $\frac{\Delta t}{2}$. A Tabela 3.9 exibe os testes de Δt e número de iterações, enquanto a Tabela 3.10 mostra os resultados de desempenho e a Figura 3.15 as imagens das simulações.

Tabela 3.9: Passos de tempo máximos e iterações necessárias do método *Leapfrog*.

Simulação	Δt máximo (s)	Número de iterações
S_1	0.012	1200
S_2	0.004	2100
S_3	0.003	3650
S_4	0.002	1850
S_5	0.002	10000
S_6	0.002	5000

Tabela 3.10: Tempos de processamento do método *Leapfrog*.

Simulação	Número de Vértices	T_i (s)	T_e (s)
S_1	100	0.0019	2s
S_2	400	0.0078	16s
S_3	900	0.0184	1 : 04s
S_4	1600	0.0331	59s
S_5	2500	0.0523	8 : 43s
S_6	3600	0.0776	6 : 46s

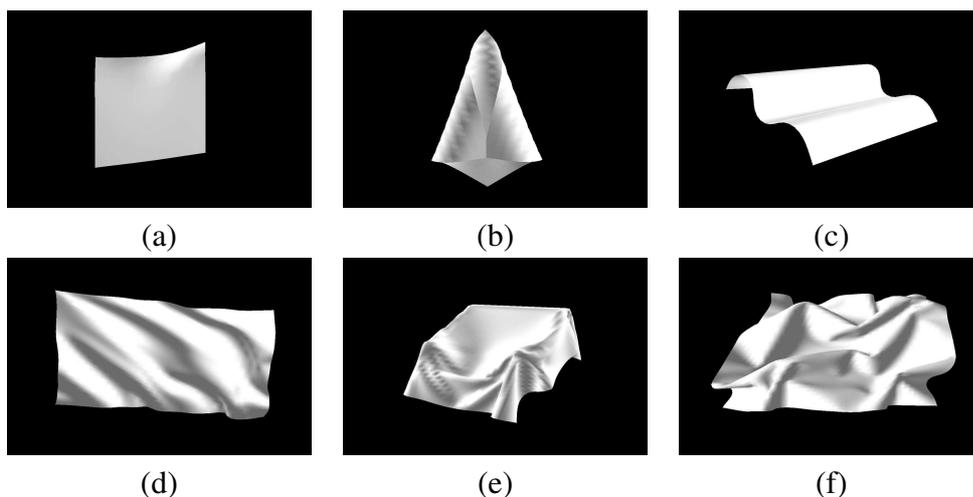


Figura 3.15: Resultados visuais do método *Leapfrog*.

3.5 Análises Comparativas

Visando organizar esta análise de acordo com os objetivos relacionados às simulações de tecidos, ela é dividida em: estabilidade numérica, desempenho computacional e qualidade das imagens. A análise da estabilidade numérica utiliza os resultados obtidos nos testes dos valores máximos de Δt e é feita na Subseção 3.5.1. A análise do desempenho computacional usa os resultados dos tempos de processamento, tanto o gasto em uma iteração quanto o gasto para atingir o equilíbrio, e é feita na Subseção 3.5.2. Para analisar a qualidade das imagens obtidas são utilizadas as imagens das simulações de teste geradas por cada método, como apresentado na Subseção 3.5.3. A escolha do método que melhor se adaptou ao modelo proposto por de Melo [5] é justificada na Subseção 3.5.4.

3.5.1 Estabilidade Numérica

Como pôde-se ver ao longo da seção 3.4, os valores máximos de Δt encontrados para cada simulação são iguais para todos os métodos, com exceção de S_2 para o método de Runge-Kutta de quarta ordem. Neste caso, o valor máximo de $\Delta t = 0.005s$ é de apenas $0.001s$ maior que os demais ($\Delta t = 0.004s$). Este resultados similares dos valores de Δt para os cinco métodos implementados pode ser considerado um resultado não previsto, pois, pela teoria, a expectativa era que pelo menos os métodos de Runge-Kutta de segunda e quarta ordem suportassem um Δt maior que os demais, afinal eles possuem novas iterações do modelo de tecido durante cada solução numérica, as quais aumentam a precisão da estimativa das velocidades dentro de um intervalo de tempo.

Investigando este comportamento similar dos métodos implementados em relação ao tamanho máximo de Δt , foi possível perceber que existe uma imposição de suavidade na variação dos elementos geométricos do modelo de tecido que limita o método explícito utilizado. Esta limitação ocorre quando existe uma grande variação do vetor normal dos pontos da superfície do tecido enquanto ele é deformado. Como a força interna do modelo é estimada pela Equação 2.14, a qual necessita da definição expressa na Equação 2.19 que, por sua vez, utiliza as quantidades geométricas diferenciais referentes a métrica e a curvatura da superfície, qualquer grande mudança de curvatura tem um grande impacto na força interna do modelo. Além disso, como a definição do material de um tecido é normalmente feita com uma alta resistência à mudança de métrica e baixa resistência à mudança curvatura, a maior parte das forças externas aplicadas ao tecido é convertida em energia que muda apenas sua curvatura, mantendo a mesma área.

Desta forma, a consideração inicial do modelo feita para integrá-lo ao longo do tempo com os métodos explícitos, de que as forças internas de cada ponto da superfície variam suavemente entre um instante de tempo e o seguinte, é violada quando ocorre uma resposta brusca da força interna do modelo a uma grande variação de curvatura. Esta resposta desproporcional normalmente faz a

posição de um ponto sofrer um grande deslocamento que rapidamente se propaga por toda a malha do tecido, resultando em uma superfície irregular e abortando a simulação por problemas numéricos. Assim, os métodos explícitos implementados apresentam a mesma estabilidade numérica enquanto as variações da força interna são pequenas dentro de um intervalo Δt .

Para melhor explicar essa situação em que a simulação mostra-se instável com determinado valor de Δt é dado um exemplo de um teste de valor máximo de Δt em que a simulação torna-se instável numericamente. As seqüências de imagens a seguir da simulação de teste S_1 está em modo de visualização da malha sem iluminação para que suas faces fiquem mais nítidas. As Figuras 3.16(a)-(d) mostram quatro imagens da simulação S_1 associadas a iteração i em que foram geradas pelo método de Euler com o passo de tempo 0.012. Já as Figuras 3.16(e)-(h) mostram as quatro respectivas imagens geradas pelo mesmo método, porém com o passo de tempo 0.013s.

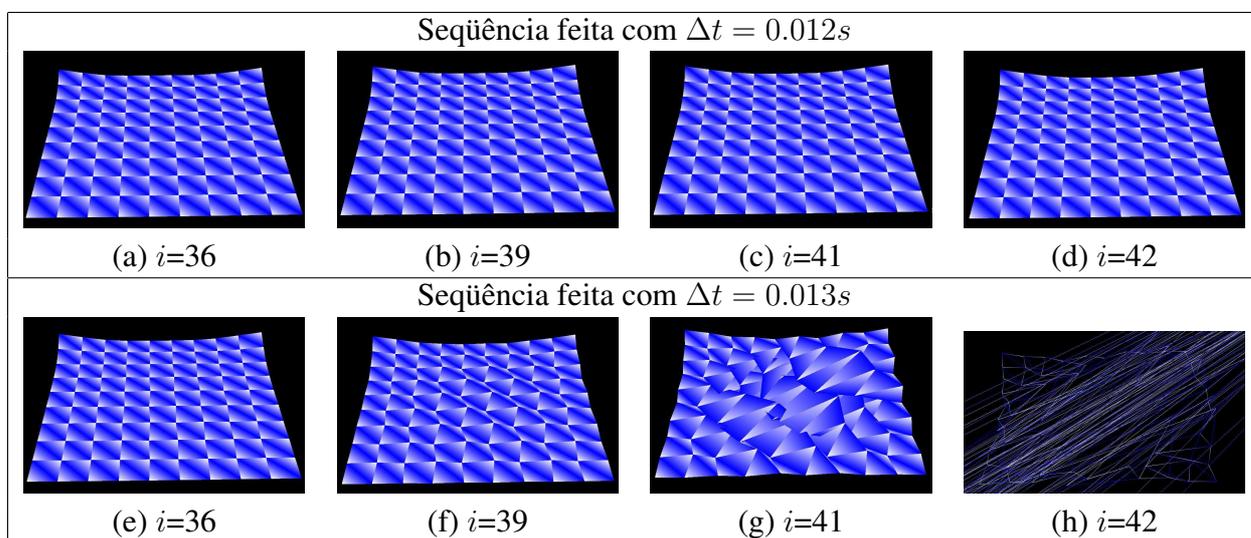


Figura 3.16: Instabilidade numérica causada por altos valores de Δt .

Na Figura 3.16(a) é possível notar que os pontos de cada coluna e linha da superfície estão bem alinhados. Já na Figura 3.16(e) os pontos apresentam deslocamentos que formam um serrilhado bem sutil ao longo das direções horizontais e verticais, quase imperceptíveis visualmente. Contudo, estes pequenos deslocamentos se propagam rapidamente e ficam mais evidentes, como mostra a imagem da Figura 3.16(f). Apenas duas iterações depois as faces da superfície do tecido já estão todas comprometidas, embora os deslocamentos não desejados permaneçam contidos no mesmo plano. Uma iteração depois a superfície se torna caótica, resultando, duas iterações mais tarde, em uma situação de instabilidade numérica, na qual é retornado ao usuário, normalmente, um erro de estouro na memória destinada a um número de ponto flutuante. A imagem da Figura 3.16(h) mostra a superfície no modo de visualização *wireframe* para salientar que os deslocamentos, a partir desta iteração, ocorrem fora

do plano também. Enquanto isso, as imagens da Figura 3.16(b)-(d) continuam mantendo seus pontos alinhados nas duas direções.

Apesar do exemplo fornecido ser somente da simulação S_1 com o método de Euler, as demais situações de instabilidade neste teste do valor máximo de Δt apresentam o mesmo padrão, ou seja, iniciam com pequenos deslocamentos, depois eles são propagados com maior intensidade de forma planar e, por fim, saem do plano com grandes deslocamentos, resultando em uma superfície não regular. Desta forma, aumentando o valor de Δt em cada simulação chega-se ao momento em que a alta mudança da curvatura da superfície do tecido em um intervalo Δt impede que forças internas adequadas sejam retornadas e a simulação se torna instável.

3.5.2 Desempenho Computacional

Na prática, viu-se que os passos de tempo máximos de cada método são iguais para cada simulação de teste, conforme explicado na Subseção 3.5.1. Isto faz com que os tempos T_e obtidos sirvam apenas como uma informação adicional sobre o desempenho computacional, embora o número de iterações necessárias para o equilíbrio ainda seja um dado interessante para analisar qual método avança com mais suavidade nos movimentos. Desta forma, a análise da eficiência computacional dos métodos implementados é feita primeiro com o tempo de processamento de uma iteração T_i , para em seguida avaliar o número de iterações necessárias para o equilíbrio. Durante os teste feitos na Seção 3.4 foram executados cinco vezes, para todos os métodos, 10000 iterações de cada simulação de teste com o valor de $\Delta t = 0.001s$, para encontrar a média do tempo gasto por iteração T_i . Contudo, aqui, com o intuito de agrupar os resultados de tempos T_i de todos os métodos e torná-los mais perceptíveis em termos de segundos, estes valores encontrados para simular 10000 iterações são usados para montar o gráfico da Figura 3.17.

Com o gráfico da Figura 3.17 nota-se que todos os tempos de processamento de cada método possuem um comportamento aproximadamente linear. Como em cada simulação de teste o número de restrições muda, existe uma pequena diferença de tempo para ser totalmente linear. Por exemplo, a simulação S_1 apresenta somente duas restrições de ponto fixo, enquanto a poltrona da simulação S_5 possui 360 restrições, incluindo forças pontuais e pontos fixos. Considerando como retas o resultado de cada método no gráfico, pode-se notar que a diferença entre eles é a inclinação da reta. O método de Runge-Kutta de quarta ordem, conforme previsto, é o que demanda maior tempo de processamento, afinal, ele tem três iterações da dinâmica interna do tecido dentro de cada iteração numérica. Em seguida, vem o método de Runge-Kutta de segunda ordem, que possui uma iteração adicional do modelo em cada solução das posições no tempo. Depois destes dois métodos, os demais são muito próximos em termos de tempo de processamento, com leve vantagem para o método de Euler.

Em relação ao número de iterações necessárias para atingir o estado de equilíbrio, nota-se um re-

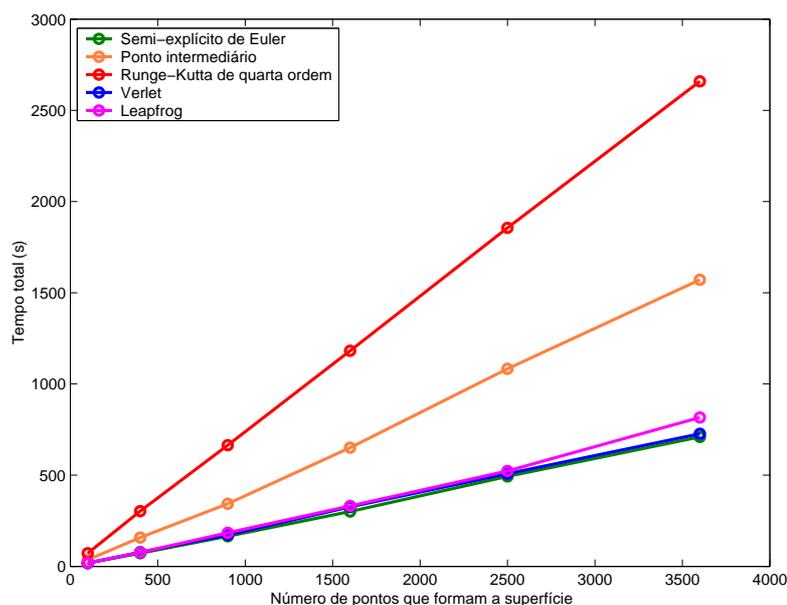


Figura 3.17: Tempos totais de processamento para executar 10000 iterações.

sultado mais diversificado. O método de Runge-Kutta de segunda ordem, por utilizar as velocidades dos pontos intermediários $t + \frac{\Delta t}{2}$ em cada intervalo Δt para avançar as posições de t até $t + \Delta t$, acaba aumentando a suavidade de movimentos na dinâmica dos tecidos e, conseqüentemente, resultando nos maiores valores do número de iterações necessárias para os equilíbrios em relação aos encontrados nos demais métodos. O método de Runge-Kutta de quarta ordem, ao contrário do de segunda, acelera mais rapidamente os movimentos da superfície, chegando antes que os demais no estado de equilíbrio. Isto acontece porque as velocidades são melhor estimadas conforme as três novas iterações da dinâmica do modelo. Contudo, apesar dos resultados dos métodos de segunda e quarta ordem de Runge-Kutta estarem em situações opostas em relação ao número máximo de iterações, eles possuem os dois maiores tempos de processamento T_i , que faz com que os resultados de T_e para ambos sejam inferiores em termos de desempenho computacional, pois eles são superiores aos obtidos nos outros três métodos. Dentre estes últimos, o método de Verlet praticamente acompanha o de Euler no número de iterações necessárias, enquanto em *Leapfrog* este número é um pouco maior.

Desta forma, pode-se concluir que em termos de desempenho computacional os métodos de Runge-Kutta de segunda e quarta ordem são inferiores aos demais por necessitarem de novas iterações do modelo de tecido. Já entre os outros três métodos: Euler, *Leapfrog* e Verlet, é possível notar que eles são muito próximos em termos de tempo de processamento e também no número de iterações necessárias para chegar ao equilíbrio. Contudo, em uma análise mais rigorosa, o método *Leapfrog* fica um pouco abaixo em relação ao método de Euler e Verlet, os quais foram os mais eficientes computacionalmente neste estudo.

3.5.3 Qualidade das Imagens

A análise dos resultados visuais deve ser feita em termos da qualidade das imagens geradas nas simulações de teste para cada método. Contudo, como se viu na Seção 3.4, todas as imagens obtidas nas simulações são muito similares visualmente em relação à proximidade com as fotografias da Figura 3.9. Assim, as diferentes precisões numéricas dos métodos implementados não refletem grandes diferenças na geometria das superfícies no estado de equilíbrio. Dentro de cada grupo de imagens associadas a uma simulação de teste, porém gerados com diferentes métodos, pode-se notar mais diferenças visuais na simulação S_4 , da bandeira ao vento, quando feita a comparação, por exemplo, da bandeira gerada pelo método semi-implícito de Euler com a gerada pelo método do ponto intermediário. Neste caso é interessante lembrar que a bandeira é a única das simulações de teste em que o estado final de equilíbrio considerado não corresponde a um estado estacionário. Assim, é considerado como estado de equilíbrio o momento que a base inferior da bandeira fica perpendicular ao poste que ela está fixada, o que não necessariamente implica na mesma geometria da superfície apresentada para esta simulação, embora as duas geometrias possam ser equivalentes em alguns instantes posteriores, dado o comportamento oscilatório da bandeira.

As geometrias das superfícies mostradas nas imagens das outras simulações de teste são muito parecidas, sendo preciso reparar a formação de pequenas dobras para perceber as diferenças visuais. Analisar estas pequenas diferenças e decidir qual apresenta melhor qualidade visual é uma tarefa subjetiva, pois enquanto uma pessoa pode achar que os resultados de um determinado método possuem mais qualidade por formarem mais dobras irregulares, outra pode achar melhor os resultados em que o tecido se deforma de maneira mais suave durante sua dinâmica. A dinâmica de tecidos é um fator importante para a análise da qualidade dos resultados visuais, porém ela não pode ser visualizada com imagens, sendo necessária a visualização das animações geradas em cada simulação de teste. Estas animações estão disponíveis no disco em anexo a esta dissertação e com elas pode-se concluir que os movimentos gerados pelos métodos de Runge-Kutta de quarta ordem e pelo método do ponto intermediário são mais suaves em relação aos outros três, embora isso não afete a qualidade das deformações produzidas na dinâmica. Desta forma, pode-se concluir que em termos de resultados visuais os métodos são todos equivalentes.

3.5.4 Escolha do Método Numérico

Para apresentar os resultados da simulação de tecidos com a abordagem proposta, este trabalho escolheu um dos métodos numéricos implementados para ser usado como padrão nos resultados do Capítulo 5. Como nesta dissertação é proposta uma alternativa que visa agregar realismo e eficiência computacional para simular tecidos, os métodos de Runge-Kutta de segunda e quarta ordem são

descartados pelo alto custo computacional que eles possuem. Com relação aos três métodos restantes, que são os métodos de Euler, *Leapfrog* e Verlet, viu-se que existe uma pequena diferença negativa do método *Leapfrog* tanto em desempenho quanto no número de iterações necessárias para atingir o equilíbrio nas simulações de teste em relação aos outros dois. Assim, o método de Euler, da forma usada por Provot [2], e o método de Verlet, são os melhores métodos deste estudo que podem integrar ao longo do tempo o modelo proposto por de Melo [5].

Devido a sua boa eficiência e precisão numérica, o método de Verlet é muito usado atualmente para o desenvolvimento de aplicações que necessitam de bom desempenho, como em *softwares* em tempo real e jogos. O software gráfico 3D Studio Max utiliza este método de integração juntamente com um modelo de tecido massa-mola, visando gastar o menor tempo possível na dinâmica de tecidos, para que com o tratamento de colisões o custo computacional de uma simulação não fique inviável. Este método também é estável em relação ao uso de passos de tempo adaptativos, fundamental para as técnicas recentes de tratamento de colisões [25, 11] e muito usado em jogos para controlar dinamicamente a relação entre velocidade e precisão, de forma que quando a precisão não for uma prioridade, o passo de tempo pode ser incrementado para um movimento mais rápido, e caso contrário, ele pode ser reduzido para gerar maior precisão. Existe também uma versão simples do método de Verlet que o adapta para o uso de passos de tempo variáveis, disponível em [35].

Além disso, embora não se tenham grandes diferenças visuais entre os métodos de Euler e Verlet, suas formulações teóricas apresentam uma diferença na precisão numérica, sendo Euler de precisão de primeira ordem enquanto Verlet é de terceira. Levando-se em conta as diferenças, optou-se por utilizar o método de Verlet como o integrador padrão do simulador desenvolvido.

3.6 Considerações Finais

Neste capítulo foram apresentados os principais métodos explícitos que podem integrar ao longo do tempo o modelo de tecido proposto por de Melo [5]. Os métodos de Euler, de Runge-Kutta de segunda e quarta ordem, Verlet e *Leapfrog* tiveram suas formulações explicadas na Seção 3.2 e foram adaptados para encontrar diferentes superfícies do tecido ao longo do tempo. Um resumo dos trabalhos relacionados com a simulação de tecidos e que utilizam os métodos explícitos discutidos foi apresentado na Seção 3.3. Para investigar qual o melhor método que se adapta ao modelo de tecido [5], a Seção 3.4 apresentou os resultados numéricos e visuais de cada método sobre um conjunto pré-determinado de simulações de testes. Com estes resultados foi possível analisar as diferenças entre os métodos em termos de estabilidade numérica, desempenho computacional e qualidade visual, conforme feito na Seção 3.5.

Com a análise feita sobre os resultados visuais e numéricos foi possível constatar que as dife-

rentes ordens de precisão numérica dos métodos explícitos não necessariamente implicam em efeitos visuais distintos. No caso da utilização destes métodos junto com um modelo baseado na mecânica de contínuos, como a formulação em diferenças finitas do modelo proposto por de Melo [5], não foi possível notar grandes diferenças visuais, contudo, os tempos de processamento mostraram uma grande diferença de desempenho computacional entre eles. Os métodos de Runge-Kutta de segunda e quarta ordem precisam de novas iterações da dinâmica do modelo em cada iteração de solução numérica, fazendo com que os tempos gastos sejam muito superiores em relação aos outros métodos. Com relação aos métodos de Euler, *Leapfrog* e Verlet, todos eles apresentam desempenhos parecidos, porém em termos do número de iterações necessárias para atingir o estado de equilíbrio das simulações de teste, o método de *Leapfrog* é um pouco inferior ao de Euler e ao de Verlet.

Analisando as diferenças entre os métodos de Verlet e de Euler, optou-se por utilizar o método de Verlet no restante dos resultados deste trabalho por ele possuir na teoria uma precisão numérica maior e por ser bastante usado atualmente em programas comerciais. Além disso, pensando na futura incorporação de algoritmos que façam o tratamento de colisões e auto-colisões, o método de Verlet já possui algumas versões que atendem bem a este propósito.

Com as soluções numéricas apresentadas, analisadas e, dentre elas, escolhida a mais apropriada para integrar o modelo de tecido proposto por de Melo [5], o Capítulo 4 aborda uma implementação de um simulador. No Capítulo 5 são apresentados os resultados das simulações de tecidos usando este simulador.

Capítulo 4

Implementação

Este capítulo apresenta uma implementação do simulador da dinâmica de tecidos. Como este trabalho visa analisar a dinâmica de tecidos produzida com uma nova implementação que torne o modelo de tecidos selecionado mais eficiente computacionalmente, optou-se por reestruturar os códigos já existentes para se adequarem às necessidades dos métodos explícitos. Além disso, o desenvolvimento de uma interface lógica de programação robusta e modular permitirão que futuros incrementos, como algoritmos de tratamento de colisões, possam ser facilmente adicionados sem grandes modificações. Para tanto, usou-se um projeto orientado a objetos, separando o processo referente à dinâmica de tecidos do de visualização da simulação, conforme explicado na Seção 4.1.

O simulador desenvolvido neste trabalho tem o objetivo de construir uma interface gráfica de fácil controle ao usuário, mas que permita criar animações complexas definindo apenas um pequeno conjunto de parâmetros de superfície e simulação. Os principais *softwares* gráficos comerciais, como o 3D Studio e Maya, possuem *plugins* com simuladores de tecidos acoplados, que são muitas vezes usados para produzir animações de excelente qualidade, entretanto, o manuseio destes *softwares* requer uma certa experiência e familiaridade com vários conceitos de simulações físicas complicados até para um usuário acostumado a usá-lo para outras tarefas. Assim, a interface gráfica proposta prioriza facilitar o manuseio de simulações de tecidos usando uma das grandes vantagens do modelo proposto por de Melo, que é a intuitividade dos parâmetros para definir as propriedades físicas do material do tecido [5]. Uma interface gráfica é desenvolvida para a plataforma Linux e outra para a Windows, as quais são detalhadas na Seção 4.2, juntamente com exemplos de simulações. Por fim, a Seção 4.3 apresenta as considerações finais.

4.1 Modelagem de dados

O primeiro passo da implementação computacional do simulador composto do modelo de tecido proposto por de Melo [5] com os métodos explícitos apresentados neste trabalho é a definição do fluxo de dados que produz a dinâmica de tecidos, como mostrado na Subseção 4.1.1. Em seguida, este fluxo de dados é apresentado de maneira mais detalhada em forma de algoritmo na Subseção 4.1.2, para então na Subseção 4.1.3 mostrar a análise de requisitos. O último passo consiste em modelar os dados envolvidos no simulador para criar uma interface lógica, usando um projeto orientado a objetos. Nesta modelagem, o simulador proposto é separado em dois módulos principais: o módulo da dinâmica de tecido, que estabelece as novas posições dos pontos da superfície em cada iteração, e o módulo de visualização, que mostrará ao usuário a animação criada com os parâmetros entrados através de uma interface gráfica. Desta forma, definem-se duas classes, a classe *malha*, responsável pela dinâmica de tecidos, e a classe *render*, responsável pela visualização da simulação, que são explicadas nas subseções 4.1.4 e 4.1.5, respectivamente.

4.1.1 Fluxo de Dados

É possível descrever a dinâmica de tecidos em termos do fluxo de dados que acontece durante toda a simulação. Considerando que o usuário informa os dados iniciais ao simulador e obtém uma animação de saída, cada iteração feita pode servir como um quadro da animação. Convencionando i como um contador crescente que representa a iteração atual em que o sistema se encontra e T como o número total de iterações que o usuário especifica, o fluxograma da Figura 4.1 ilustra a entrada, a saída, os dados, o teste de parada e os processos envolvidos no simulador.

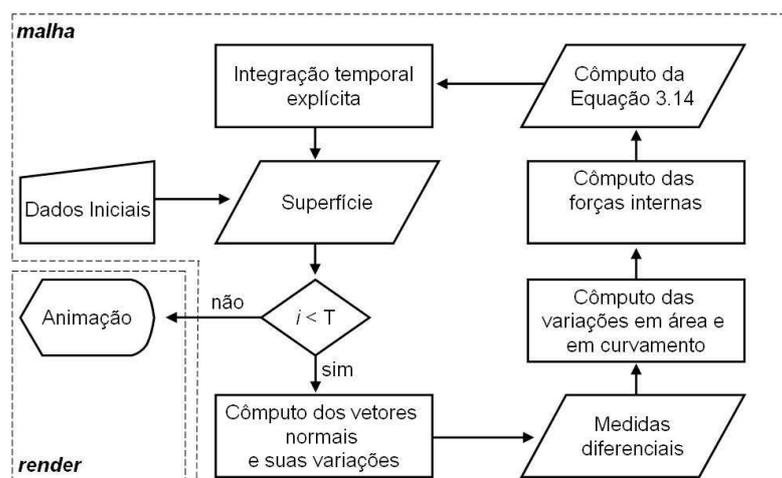


Figura 4.1: Fluxograma da dinâmica de tecidos.

Desta forma, o usuário informa um conjunto de dados para que o simulador construa a superfície inicial do tecido, que neste simulador é sempre um plano. Em seguida, os elementos da geometria diferencial são calculados com base nos pontos que formam a superfície, obtendo os vetores que formam o referencial local a cada ponto, como também seus tensores e sua área. Com as medidas geométricas calculadas, o processo seguinte consiste em calcular as variações da forma da superfície atual em relação à inicial, obtendo as medidas de deformação. Estas são usadas no cálculo das forças internas que, junto com as forças externas definidas pelo usuário, são aplicadas na Equação 3.14, a qual é solucionada por um dos métodos de integração explícita apresentados neste trabalho, para obter uma nova geometria da superfície. Este ciclo é feito até que o número de iterações feitas pelo sistema atinja o valor especificado pelo o usuário. A saída do sistema é uma animação formada pelas geometrias da superfície encontradas em cada iteração. Na prática, a construção da animação de saída é feita tomando-se iterações devidamente espaçadas, em uma taxa aceitável para animação (aproximadamente 30 quadros por segundo).

4.1.2 Algoritmos

A simulação de tecidos pode ser feita com um modelo de tecido em conjunto com uma integração temporal, que encontra diferentes geometrias da superfície ao longo do tempo. Assim, pode-se descrever o sistema dividindo-o em dois algoritmos: um para encontrar todos os elementos do modelo de tecido e obter as acelerações resultantes; e outro para obter a geometria da superfície em cada iteração a partir destas acelerações. O Algoritmo 2 apresenta as instruções da função *Deforma*, responsável por receber os dados iniciais da simulação, calcular na ordem correta os elementos do modelo deformável e chamar a função que realiza a integração temporal explícita ao seu final [3]. Já o Algoritmo 1 apresenta as instruções da função *Integra*, que corresponde ao método de Verlet.

Função Integra (var r_{i-1} : Ponto[]; var $v_{i-1}, f_{i-1}, F_{i-1}$: Vetor[]; var $\rho, \mu, \Delta t$: Real)

```

01 Var
02    $a_i, v_i, r_i, aux$  : Vetor[];
03 Início
04   Para cada ponto  $(k, l)$  da malha discreta faça
05      $a_i(k, l) \leftarrow \frac{F_{i-1}(k, l) - \rho v_{i-1}(k, l) - f_{i-1}(k, l)}{\mu}$ ;
06      $aux(k, l) \leftarrow r_i(k, l)$ ;
07      $r_i(k, l) \leftarrow 2 * r_{i-1}(k, l) - r_{i-1}(k, l) + \Delta t^2 a_i(k, l)$ ;
08      $v_i(k, l) \leftarrow \frac{r_i(k, l) - r_{i-1}(k, l)}{2 * \Delta t}$ ;
09      $r_{i-1}(k, l) \leftarrow aux$ ;
10   Fim para
11    $Integra \leftarrow r_i$ 
12 Fim

```

Algoritmo 1: Algoritmo de integração explícita (método de Verlet).

Função Deforma (**var** $\mathbf{v}_{inicial}, \mathbf{a}_{inicial}, \mathbf{f}_{externas} : \mathbf{Vetor}[]$; **var** $\zeta_{\alpha\beta}, \xi_{\alpha\beta}, \phi_{\alpha\beta}, massa, \rho, \Delta t : \mathbf{Real}$;
var $comp_x, comp_y, comp_z : \mathbf{Real}$; **var** $m, n, passos : \mathbf{Inteiro}$)

```

01 Var
02    $\mathbf{r}, \mathbf{r}_{antigo}, \mathbf{v}, \mathbf{F}, \mathbf{EstadoDeformado} : \mathbf{Vetor}[]$ ;
03    $S_0, S, \mu, fator : \mathbf{Real}$ ;
04    $i : \mathbf{Inteiro}$ ;
05 Início
06   Calcule a área inicial  $S_0$  com base em  $comp_x, comp_y$  e  $comp_z$ ;
07   Discretize  $S_0$  usando  $m$  e  $n$  de forma a criar espaçamentos iguais  $\Delta x^1$  e  $\Delta x^2$  e preencher o vetor  $\mathbf{r}$ ;
08   Para cada ponto  $(k, l)$  da malha discreta faça
09      $\mathbf{v} \leftarrow \mathbf{v}_{inicial}$ ;
10      $\mathbf{F} \leftarrow \mathbf{f}_{externas}$ ;
11     Calcule os vetores de base  $\mathbf{a}_\alpha$  usando a Equação 2.2 e o vetor normal  $\mathbf{n}$  usando a Equação 2.3;
12     Calcule os valores iniciais dos tensores métrico  $A_{\alpha\beta}$ , métrico recíproco  $A^{\alpha\beta}$  e de curvatura  $B_{\alpha\beta}$ , usando as
13     Equações 2.4, 2.5 e 2.6, respectivamente;
14     Calcule os parâmetros do material  $\Phi^{\alpha\beta}, \Psi^{\alpha\beta}, \Theta^{\alpha\beta}$ , usando  $\zeta_{\alpha\beta}, \xi_{\alpha\beta}, \phi_{\alpha\beta}$  e  $A^{\alpha\beta}$  (Equação 2.22);
15   Fim para
16    $i \leftarrow 0$ ;
17   Enquanto  $i$  for menor que  $passos$  faça
18     Determinar a área atual  $S$  com base nos pontos  $\mathbf{r}$ ;
19      $\mu \leftarrow \frac{massa}{S}$ ;
20      $fator \leftarrow \frac{S_0}{S}$ ;
21     Para cada ponto  $(k, l)$  da malha discreta faça
22       Calcule os vetores de base  $\mathbf{a}_\alpha$  e vetor normal  $\mathbf{n}$ ;
23       Calcule os valores atuais dos tensores métrico  $a_{\alpha\beta}$ , métrico recíproco  $a^{\alpha\beta}$  e de curvatura  $b_{\alpha\beta}$ ;
24       Calcule a medida de deformação  $\varepsilon_{\alpha\beta}$  usando a Equação 2.7;
25       Calcule a medida de deformação  $\kappa_{\alpha\beta}$  usando a Equação 2.8;
26       Calcule os símbolos de Christoffel  $\Gamma_{\alpha\beta}^\gamma$  usando a Equação 2.16;
27       Calcule os termos de Weingarten  $b_{\alpha\beta}^\beta$  usando a Equação 2.20;
28       Calcule os componentes da força de curva  $\mathbf{N}^{\alpha\beta}$  usando a Equação 2.19;
29       Calcule os componentes da força diretora  $\mathbf{M}^{\alpha\beta}$  usando a Equação 2.18;
30       Calcule a força interna  $\mathbf{N}^\alpha|_\alpha$  usando a Equação 2.14;
31       Calcule a força externa resultante usando  $\mathbf{F}$  e  $\mathbf{a}_{inicial}$ ;
32     Fim para
33      $\mathbf{r}_{antigo} \leftarrow \mathbf{r}$ ;
34      $\mathbf{r} \leftarrow \mathbf{Integra}(\mathbf{r}, \mathbf{v}, \mathbf{N}^\alpha|_\alpha, \mathbf{F}, \rho, \mu, \Delta t)$ ;
35      $\mathbf{EstadoDeformado}[i] \leftarrow \mathbf{r}$ ;
36     Para cada ponto  $(k, l)$  da malha discreta faça
37        $\mathbf{v} \leftarrow \frac{\mathbf{r} - \mathbf{r}_{antigo}}{\Delta t}$ ;
38     Fim para
39      $i \leftarrow i + 1$ ;
40 Fim enquanto
41  $\mathbf{Deforma} \leftarrow \mathbf{EstadoDeformado}$ ;
42 Fim

```

Algoritmo 2: Algoritmo de deformação.

Assim, o Algoritmo 1 é chamado quantas vezes for preciso até que o número de iterações especificado pelo usuário seja atingido na função **Deforma** do Algoritmo 2. Esta última função é responsável por calcular todos os elementos do modelo na ordem correta e passar os referentes à integração explícita para a função **Integra**. Elementos como os parâmetros de material são calculados para cada ponto da malha discreta somente na primeira iteração, de forma que o mesmo valor é usado nas demais até o fim. A cada iteração da dinâmica do simulador de tecidos é gerada uma geometria da superfície, representada pelo vetor de pontos \mathbf{r} , o qual contém as posições tridimensionais de cada ponto que são guardadas no vetor chamado **EstadoDeformado**. Ao final do laço principal da função, este vetor de estados é o retorno da função, sendo usado para criar a animação.

4.1.3 Análise de Requisitos

Observando os Algoritmos 1 e 2, pode-se avaliar quais são os dados constantes, os dados variáveis, as estruturas de dados e as funções necessárias para a implementação do simulador de tecidos. Um primeiro conjunto de dados que se pode destacar são os dados informados pelo usuário, de forma que todos eles precisam ser armazenados durante a simulação. Estes dados são listados a seguir por uma simples descrição, relacionando-os com os respectivos tipos de dados.

- Para definir o plano que forma a superfície inicial do tecido são necessários três valores reais, que correspondem aos comprimentos nos eixos x , y e z dado em m (metro).
- Para armazenar os pontos tridimensionais da superfície é preciso um vetor cuja dimensão depende da resolução espacial, que é definida por dois valores inteiros e correspondem às dimensões da grade discreta bidimensional.
- A velocidade inicial é informada em m/s (metro por segundo) em um único vetor, de forma que cada ponto possui esta velocidade na superfície inicial. Todos os pontos possuem uma velocidade atual que precisa ser armazenada em um vetor de dimensão igual ao vetor de posições.
- A aceleração inicial é dada em m/s^2 (metro por segundo quadrado) e funciona da mesma forma que a velocidade, também possuindo um único vetor inicial de entrada e outro de dimensão igual ao de posições para cada iteração.
- As forças externas que agem sobre o tecido podem ser especificadas em N (newton) sobre cada ponto da grade bidimensional, de forma que é preciso também de um vetor de dimensão igual aos demais.
- Para definir o material do tecido é preciso informar os coeficientes elásticos da superfície, especificados em N (newton) que correspondem aos quatro valores reais de cada um dos três coeficientes.

- Também relacionado ao material do tecido, sua massa é um valor real especificado em kg (quilograma).
- O coeficiente de amortecimento é um valor real especificado em Ns/m (newton segundo por metro).
- O passo de tempo que separa as iterações é um valor real especificado em s (segundo), mas o número total de iterações é um número inteiro.

Além destes dados informados pelo usuário é preciso também modelar os dados variáveis que são usados para produzir a dinâmica no simulador. Estes dados são listados a seguir [3].

- Cada ponto da grade discreta possui dois vetores de base e um vetor normal, além de uma área da vizinhança local ao ponto, a qual é representada por um número real.
- Para cada ponto também estão definidos os tensores métrico, métrico recíproco e o de curvatura, de forma que cada um deles possui quatro valores reais.
- Com operações sobre estes tensores e os elementos que formam o referencial local a cada ponto, são obtidas as medidas de deformação, que são dois valores reais, os símbolos de Christoffel, que são seis valores reais, e os termos de Weingarten, que são quatro valores reais, por ponto.
- A força interna é calculada com base nos componentes da força de curva e força diretora, que são representadas em um vetor de dimensão igual ao de posições.

É importante salientar que todos os vetores citados na modelagem desses dados são estruturas de dados com posições tridimensionais, como o da posição, velocidade e aceleração. Desta forma, os elementos utilizados na modelagem do simulador foram descritos e associados aos seus tipos, de maneira a incorporá-los na implementação. Feito isto, o próximo passo é verificar as estruturas e funções que podem ser definidas com base nas necessidades da dinâmica de tecidos. Observando a descrição dos requisitos do sistema, três estruturas podem ser destacadas: vetores, tensores e dados pertencentes a um único ponto da grade bidimensional. Quanto às funções, além das descritas nos Algoritmos 1 e 2, é preciso de duas que calculem os valores aproximados das derivadas de primeira e segunda ordem do modelo, além de outras auxiliares ao cálculo dos elementos.

Utilizando a programação orientada a objetos para modelar os requisitos exibidos nesta subseção, é possível apontar duas classes que gerenciaram a dinâmica de tecidos. A classe *malha* é responsável pelo fluxo de dados da Figura 4.1, com exceção da saída, de forma a incorporar a implementação dos Algoritmos 1 e 2. A classe *render* é responsável por coletar as diferentes configurações das posições dos pontos do tecido e transformar estes dados em imagens e animações, que devem ser vistas através de uma interface gráfica desenvolvida para o usuário. É considerado de fundamental importância que estas duas classes sejam projetadas como módulos independentes, de forma que a parte de visualização seja separada da dinâmica para tornar o módulo referente à classe *malha* portátil, bem como

adequar-se à proposta de Bridson et al. [25] no sentido de separar apropriadamente a dinâmica interna dos tecidos antes de integrá-la a um algoritmo de tratamento de colisões.

4.1.4 Classe *malha*

A classe *malha* é responsável por realizar a dinâmica de tecidos, de forma independente da classe *render*, de modo que seja possível portá-la entre diferentes plataformas e só mudar a forma de visualização da animação. Além disto, esta classe provê uma interface lógica robusta para que futuras incorporações sobre ela, como um sistema de tratamento de colisões, sejam facilmente implementadas. Analisando os Algoritmos 1 e 2, pode-se dizer que ambas as funções descritas neles formam a dinâmica de tecidos e, portanto, devem ser modeladas na classe *malha*, de maneira que seus atributos são os elementos do modelo de tecido e, seus métodos, as rotinas que calculam estes elementos. A classe *malha* e as principais classes usadas em sua definição são ilustradas na Figura 4.2.

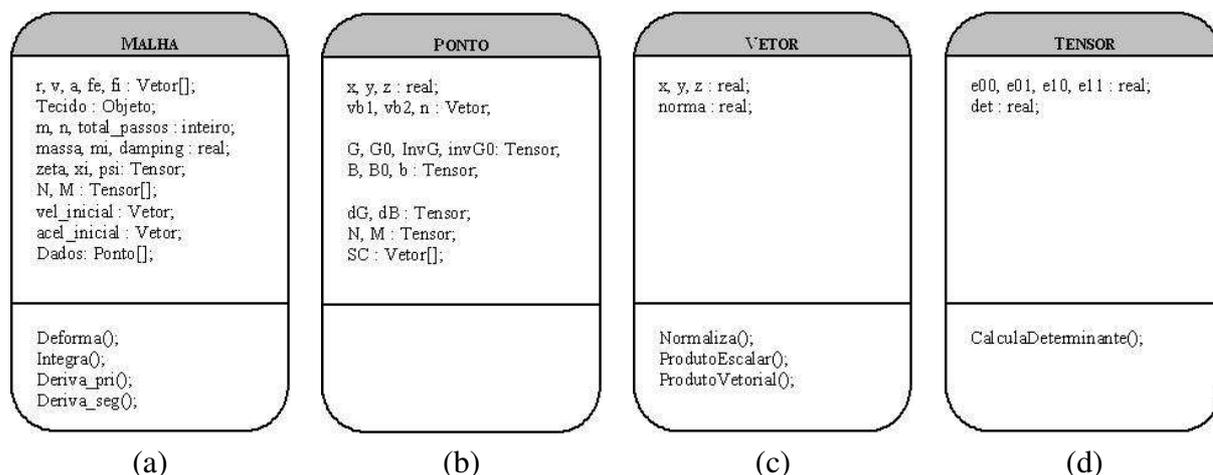


Figura 4.2: Classe *malha* e suas principais classes.

O ponto de partida para os atributos da classe *malha* são os parâmetros de superfície e simulação fornecidos pelo usuário, que definem as características geométricas e físicas do objeto deformável. Os parâmetros geométricos consistem nas dimensões do objeto de acordo com o seu tipo, como por exemplo, três comprimentos nos planos x , y e z para definir um objeto do tipo plano, um raio e um comprimento para definir um cilindro, um raio maior e um menor para um toro, e assim por diante. Apesar da implementação deste trabalho só considerar a deformação de planos em sua discretização espacial, o simulador proposto é planejado para dar suporte à criação de vários tipos de objetos, modelando-os através da classe *objeto* dentro da classe *malha*, como mostra a Figura 4.2(a). Nesta figura considera-se os nomes de tipos minúsculos como os básicos da linguagem de programação C++ e os maiúsculos como outras classes, como *ponto*, *vetor* e *tensor*. A classe *ponto*, ilustrada na

Figura 4.2(b), encapsula todos os dados do modelo de tecido que estão associados a cada nó da malha discreta, por isso o vetor chamado *Dados* faz parte da classe *malha*. As classes *vetor* e *tensor* são mostradas nas Figuras 4.2(c) e 4.2(d), respectivamente.

Por fim, a classe *objeto* é considerada uma classe virtual, de forma que ela impõe que seus métodos sejam implementados por suas sub-classes, como a classe *plano* usada no simulador, ou como a classe *cilindro*, ilustrada na Figura 4.3 como um exemplo de outros objetos que podem ser incorporados ao simulador.

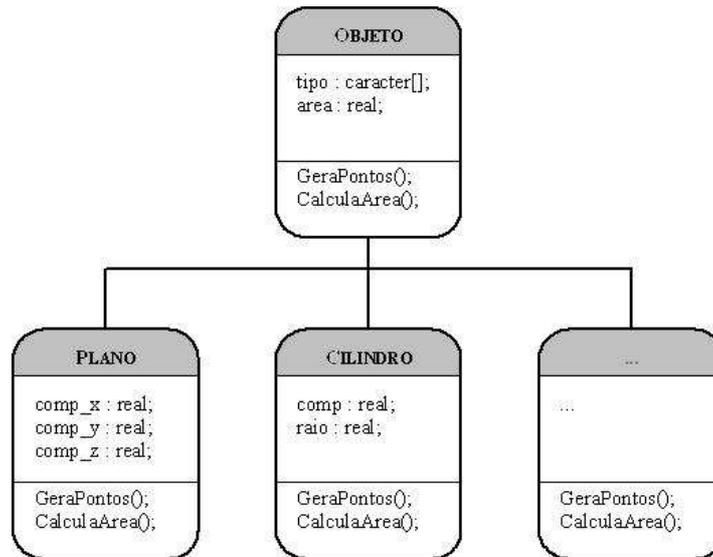


Figura 4.3: Modelagem da classe *objeto*.

4.1.5 Classe *render*

A classe *render* é responsável pela visualização de imagens e animações produzidas pela dinâmica de tecidos. Como ela deve ser feita de forma independente da classe *malha*, a única comunicação entre elas é feita com a passagem dos pontos de cada estado do tecido, que corresponde as linhas 34 e 40 do Algoritmo 2. Quando o usuário informa o número de iterações que serão feitas em uma simulação, junto com o atributo intervalo, é possível estabelecer quantos quadros a animação possuirá, que corresponde a divisão $quadros = \frac{número\ de\ iterações}{intervalo}$. Com o tempo total da animação, informada pelo usuário em segundos, e sabendo o número de quadros, é utilizado um temporizador que atualiza cada quadro após instantes iguais de tempo. Este tempo da animação não está vinculado ao montante de tempo simulado. O tempo simulado é obtido multiplicando o número de iterações pelo valor de Δt , contudo, este tempo pode ser visualizado em uma duração diferente, que corresponde ao tempo de animação. Desta forma, a classe *render* possui como atributos uma lista de posições tridimensionais

referentes a cada nova geometria da superfície do tecido e um valor inteiro que corresponde ao intervalo no qual as iterações produzidas pela dinâmica de tecidos tornam-se quadros da animação gerada para a saída da simulação.

Além desses atributos, que são especificados pelo usuário, a classe *render* possui métodos que geram e gerenciam as animações produzidas nas simulações. A geração consiste em armazenar as geometrias da superfície encontradas nos intervalos determinados e desenhá-las em uma cena tridimensional. Para este desenho escolheu-se usar a biblioteca gráfica OpenGL. Assim, cada geometria é desenhada em uma lista de visualização que é chamada durante a execução da animação. Contudo, para aumentar a usabilidade do simulador, é interessante gerar estas animações de dois modos: de maneira dinâmica, em que os quadros são exibidos à medida que são gerados, conhecido como modo *on-the-fly*; e de maneira estática, em que todos os quadros são gerados primeiro para depois a animação ser visualizada. Nestes dois modos foram construídas rotinas para carregar os quadros em *display lists* para que o processo de visualização da animação seja eficiente. Além disso, foram desenvolvidas rotinas para salvar e carregar os quadros para que não seja preciso executar novamente uma simulação somente para assistir a animação gerada.

Existem também recursos auxiliares ao processo de animação que são interessantes de serem incorporados na classe *render*. A visualização da grade da superfície do tecido, conhecida como modo de visualização *wireframe*, é muito importante principalmente quando o objetivo é depuração e calibração de parâmetros. A aplicação de texturas e fontes de luz na cena também contribuem para o aumento da qualidade da animação, incrementando também o realismo na visualização do tecido. Rotinas de movimentação da cena, como as de translação e rotação, bem como as de escala, também são consideradas. A Figura 4.4 ilustra os principais atributos e métodos da classe *render*.

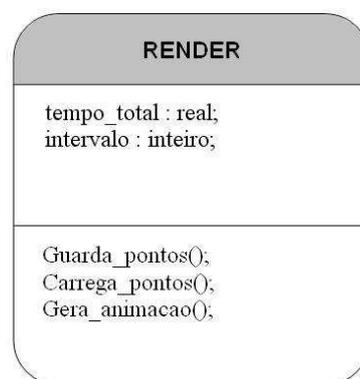


Figura 4.4: Modelagem da classe *render*.

4.2 Interface Gráfica

A interface gráfica do simulador de tecidos foi desenvolvida com a linguagem de programação C++ em conjunto com a biblioteca gráfica OpenGL. A implementação do sistema foi feita para as plataformas Linux, detalhada na Subseção 4.2.1, e também para plataforma Windows, descrita na Subseção 4.2.2. Em ambas as plataformas seguiram-se as exigências analisadas na Seção 4.1, de maneira a separar a dinâmica de tecidos da visualização de imagens e animações. Por fim, a Subseção 4.2.3 dá exemplos de utilização do simulador.

4.2.1 Plataforma Linux

A interface gráfica na plataforma Linux foi desenvolvida com o uso do Qt Designer 3.3. A janela principal contém um componente com quatro abas, sendo elas divididas em grupos de diferentes informações, como mostrado na Figura 4.6(a). É nesta janela que todas as rotinas de entrada e saída das simulações de tecidos são feitas. O único botão do menu principal tem como tarefa gerenciar os arquivos que contém os dados de entrada e salvar novas configurações em arquivo. O arquivo de entrada é um arquivo em ASC II com os valores de definição dos parâmetros da simulação que, após carregado, preenche a maioria dos campos dentro das quatro abas da janela principal. A sintaxe completa deste arquivo está no disco em anexo a esta dissertação. A Figura 4.5 mostra um exemplo deste arquivo, que corresponde à simulação S_1 do Capítulo 3. Pode-se também preencher todos os campos na interface e, ao salvar os valores preenchidos, é criado um arquivo texto com o mesmo formato dos dados. Na aba referente aos dados de simulação encontram-se as informações sobre o passo de tempo, tempo de animação, número de iterações e intervalo considerado entre as iterações para armazenar os quadros da animação. Nesta aba o usuário deve selecionar o tipo de saída das imagens, o método de integração utilizado e qual tipo de visualização da animação ele deseja.

```
% Este script foi criado automaticamente pelo Desmo 1.0
delta_t = 0.001
n_passos = 10000
pula = 50
tempo_animacao = 10
grid : 10 10
aceleracao : x = 0.0 y = -9.8 z = 0.0
velocidade : x = 0.0 y = 0.0 z = 0.0
eta : 100.0 100.0 100.0 100.0
qs1 : 0.3 0.3 0.3 0.3
psi : 0.001 0.001 0.001 0.001
massa = 10.0
gama = 0.2
obj_deformavel : plano 10.0 0.0 10.0
n_restricoes = 2
restricoes = {
0 0 ponto_fixo inicio=0 fim=20001 x=0 y=0 z=0
0 9 ponto_fixo inicio=0 fim=20001 x=0 y=0 z=0
}
```

Figura 4.5: Exemplo de um arquivo de entrada com todos os dados de uma simulação.

A aba da Figura 4.6(b) é referente aos valores do ambiente em que a superfície se encontra. Nesta aba ocorre a definição da *aceleração* do ambiente, da *velocidade inicial* dos pontos da superfície, do *coeficiente de amortecimento*, expresso pelo rótulo *Gama*, e também do módulo e direção da força *fluido*. É interessante lembrar que a força fluido é um parâmetro opcional. A aba seguinte é referente aos dados da malha discreta que forma o tecido, em que se definem as resoluções espaciais da *grade* e os parâmetros do material do tecido, como sua *massa* e seus coeficientes elásticos ζ (rótulo *ZETA*), ξ (*XI*) e ϕ (*PSI*), como mostra a Figura 4.6(c). Nesta figura, existe uma caixa de opção sobre o tipo de material do tecido, porém sua função é apenas definir a textura utilizada sobre o tecido na visualização. A última aba é referente aos dados do objeto deformável, que corresponde à superfície do tecido. Nela pode-se especificar as dimensões x , y e z do plano inicial a ser deformado, em termos das coordenadas dos eixos, conforme ilustra a Figura 4.6(d). Há também o botão que abre a janela das restrições, mostrada na Figura 4.7. O botão com o rótulo *fronteiras* é usado atualmente somente para depurações, assim como o botão de rótulo *Gera Dados*.

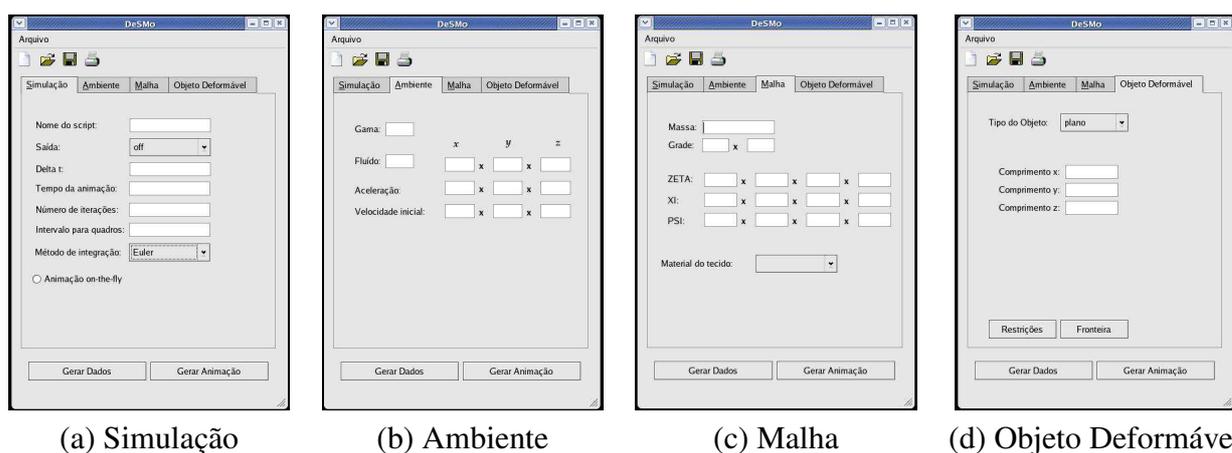


Figura 4.6: Interface gráfica para a plataforma Linux.



Figura 4.7: Janela das restrições da simulação.

Com todos os dados da janela principal preenchidos ou carregados, a simulação pode ser processada acionando o botão com o rótulo *Gerar Animação*. Ao apertar este botão a dinâmica de tecidos começa a ser calculada e a animação gerada é exibida em uma janela de desenho com as funcionalidades de OpenGL. Este contexto é criado somente com a superfície inicial do tecido, porém apertando teclas de atalho é possível acessar informações sobre o quadro que está sendo visualizado, conforme ilustrado na Figura 4.8(a). O movimento do tecido nesta janela é gerenciado pelos botões do *mouse*. O botão esquerdo alterna entre os modos de reprodução ou pausa da animação, enquanto o direito reinicia a partir do primeiro quadro. À medida que a animação é reproduzida pode-se visualizar mais informações, como feito na Figura 4.8(b), ou usar os recursos auxiliares, como adicionar e retirar iluminação, texturas, preenchimento das faces, ou realizar operações de escala e movimentação na superfície do tecido.

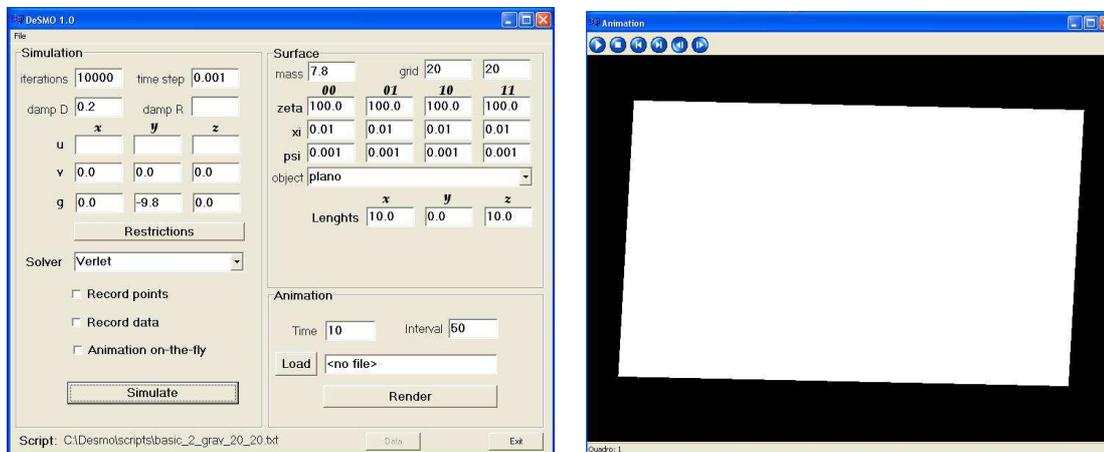


Figura 4.8: Janela de visualização da animação gerada.

4.2.2 Plataforma Windows

A interface gráfica desenvolvida para a plataforma Windows possui um formato diferente, dispendo todas as informações da simulação na janela principal, sem o uso de abas, como mostra a Figura 4.9(a). Esta interface foi feita no compilador C++ Borland Builder 6, usando a biblioteca gráfica VCL do Windows. No começo deste trabalho apenas a interface para a plataforma Linux estava disponível, porém para testar a portabilidade do módulo malha e também para tornar o simulador mais acessível, a interface gráfica foi feita para o sistema Windows com os rótulos em inglês. De forma similar à desenvolvida para o Linux, o botão *File* na barra do menu gerencia os arquivos de entrada e saída que preenchem a maioria dos parâmetros da simulação. Dentro desta janela principal os dados são divididos entre os grupos de simulação, de superfície e de animação. Os parâmetros de animação são o tempo de animação e o intervalo entre interações usado para armazenar os quadros. Os parâmetros de superfície são todos os parâmetros relacionados com a superfície do tecido, que são equivalentes

aos das abas de malha e objeto deformável na interface gráfica do Linux. Todos os demais parâmetros são considerados parâmetros de simulação na interface do Windows.



(a)

(b)

Figura 4.9: Janelas de (a) simulação; e (b) visualização da animação gerada.

Um procedimento recomendado para simular tecidos que possuam muitos pontos é utilizar o modo de animação *on-the-fly* até calibrar os parâmetros da simulação de maneira estável para garantir que não ocorra uma situação de instabilidade numérica. Quando a simulação estiver corretamente definida, pode-se armazenar arquivos contendo todos os seus pontos e dados para assistir a animação sempre que se desejar sem necessitar gerar os quadros novamente. Contudo, no modo *on-the-fly*, no qual os quadros da animação são calculados e armazenados em listas de visualização na memória volátil do computador, pode-se salvar individualmente em arquivo cada quadro separadamente, escolhendo-se sua extensão. Nesta implementação também estão disponíveis os mesmos recursos auxiliares acionados por teclas de atalho, que controlam as rotinas de iluminação, preenchimento, texturização, rotação, escala e translação.

A janela de visualização da animação ilustrada na Figura 4.9(b) difere um pouco da janela feita para a plataforma Linux, sendo ela de usabilidade mais convencional em relação à reprodução de animações. Utilizando botões de reprodução, pausa, retorno ao início, avanço ao final e navegação quadro a quadro, os quais correspondem, respectivamente, aos seis botões no alto da janela da Figura 4.9(b), o fluxo da animação pode ser facilmente controlado. O quadro atual da animação é mostrado no rodapé da janela, enquanto os valores de qualquer elemento do modelo, como a área do tecido, podem ser obtidos ligando a opção rotulada como *Record data*. Por fim, o tempo de processamento gasto em uma simulação é exibido em uma mensagem de alerta ao usuário no final da execução das iterações solicitadas.

4.2.3 Exemplos de Simulação

Nesta subseção são dados alguns exemplos de simulações de tecidos para mostrar a usabilidade do simulador implementado. Para tanto, são explicadas como foram feitas algumas das simulações de teste do Capítulo 3, evidenciando os parâmetros de simulação. A simulação S_1 , por exemplo, cujos dados estão ilustrados na Figura 4.5, é um tecido pequeno que cai de acordo com a força da gravidade preso por suas duas extremidades superiores. Esta simulação é feita com uma superfície de resolução espacial 10×10 , com massa $10kg$ e coeficientes $\zeta_{\alpha\beta} = 100N$, $\xi_{\alpha\beta} = 0.3N$ e $\theta_{\alpha\beta} = 0.001N$. A força da gravidade $\mathbf{g} = (0, -9.8, 0)m/s^2$ age sobre todos os pontos da superfície. Suas únicas restrições são dois pontos fixos que atuam desde a primeira iteração da dinâmica até a iteração 20000. O estado final da simulação S_1 está ilustrado na Figura 4.11(a).

Fazendo agora uma pequena alteração para mostrar como funciona a definição das restrições no simulador, deseja-se que o tecido caia de forma semelhante durante a iteração 1 até a 10000 e, a partir da iteração 10001 até 15000, só o ponto da direita seja fixo, enquanto o da esquerda possa se movimentar, mas somente ao longo dos eixo xz , ou seja, continuando fixo em y . Da iteração 15001 até 20000, o ponto da esquerda, que até chegar neste intervalo já sofreu um deslocamento, permanecerá fixo em todos os eixos, liberando o ponto fixo da direita, que estava fixo desde a iteração 1 até a 15000. A definição das novas restrições é mostrada na Figura 4.10(b). Já os resultados da simulação estão ilustrados na Figura 4.11, associando as imagens com as iterações.

Restrictions								
	U	V	Tipo	Inicio	Fim	X	Y	Z
0	0	0	ponto_fixo	0	20001	0	0	0
1	0	9	ponto_fixo	0	20001	0	0	0

(a)

Restrictions								
	U	V	Tipo	Inicio	Fim	X	Y	Z
0	0	0	ponto_fixo	0	10000	0	0	0
1	0	9	ponto_fixo	0	15000	0	0	0
2	0	0	ponto_fixo	10001	15000	1	0	1
3	0	0	ponto_fixo	15001	20000	0	0	0

(b)

Figura 4.10: Janela de restrições.

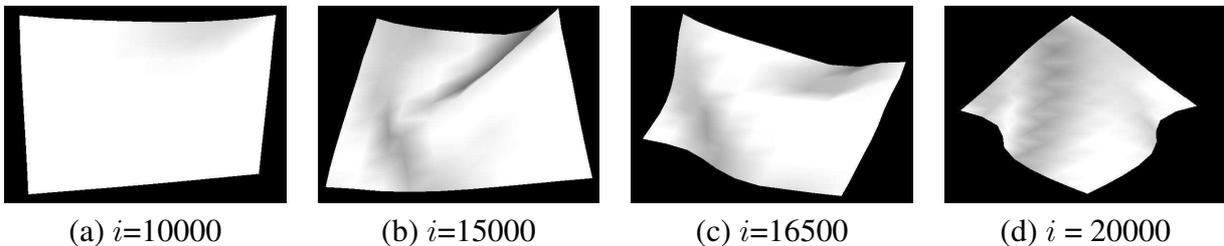


Figura 4.11: Imagens de simulação.

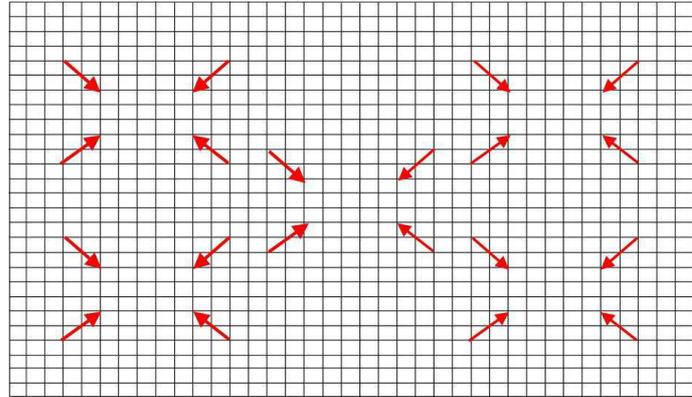
Desta forma, o controle das restrições no simulador implementado é feito com a definição de poucos parâmetros, obtendo efeitos interessantes por simplesmente deixar a força da gravidade agir com a alternância de pontos fixos em diferentes instantes e eixos. A simulação S_2 do Capítulo 3 possui as restrições definidas de forma parecida com as de S_1 , com a única diferença de se ter um ponto fixo ao invés de dois, fazendo com que a força da gravidade sobre o tecido suspenso forme a geometria de uma capa. Como esta implementação não possui algoritmos que tratem colisões, os caimentos de tecidos sobre objetos não podem ser reproduzidos de forma correta sob o ponto de vista físico. Entretanto, estes efeitos podem ser aproximados com a definição de pontos fixos que conforme seus parâmetros, como em quais eixos são fixos e sua duração, formam a geometria do objeto de colisão. Isto foi feito nas simulações S_3 , com o caimento de um tecido sobre um banco de praça, e em S_5 , com a forma de um tecido sobre uma poltrona, mostradas nas Figuras 3.10(c) e 3.10(e).

É possível também definir como restrições da simulação forças pontuais, as quais são aplicadas separadamente a cada ponto da malha discreta com um módulo e sentido determinado. Estas forças são muito úteis para dar um acabamento ao tecido simulando pequenos ajustes que ocorrem com a mão na realidade, ou até contornando a falta de tratamento de colisões na implementação corrente. A simulação S_6 , por exemplo, consiste de uma superfície em um ambiente sem gravidade sobre a qual são aplicadas forças de compressão tangenciais ao seu plano inicial (Figura 3.10(f)). Ao todo, vinte pontos da malha sofrem forças pontuais, aplicadas até a iteração 10000 com um módulo apropriado para evitar auto-colisões. Em seguida, estes mesmos pontos sofrem uma força menor somente para manter as dobras já formadas. A Figura 4.12(a) mostra algumas das quarentas restrições e 4.12(b) ilustra a localização e o sentido das forças na malha discreta.

Naturalmente, os dois tipos de restrições, que são os pontos fixos e as forças pontuais, podem ser utilizados em uma simulação ao mesmo tempo. Em relação às suas definições, a diferença ocorre somente na atribuição dos valores para os parâmetros X , Y e Z . Para pontos fixos, o valor 1 em uma destas coordenadas quer dizer que o ponto pode andar naquele eixo, ou seja, se um ponto tiver $X=Y=Z=1$ a restrição não terá validade, pois o ponto se deslocará nos três eixos, enquanto $X=Y=1$ e $Z=0$ indica que o ponto só pode andar no plano xy , como foi feito na imagem da Figura 4.11(b). Já quanto às forças pontuais, os valores são definidos com o módulo das forças em cada eixo, com o sinal determinando o sentido da força. Por exemplo, as forças da Figura 4.12(b) que estão no sentido da esquerda para direita têm os módulos no eixo x positivos, conforme as definições da Figura 4.12(a).

Contudo, além das forças pontuais e da gravidade atuando sobre os pontos fixos, é possível especificar a força fluido do ambiente em que o tecido está imerso. Utiliza-se a expressão proposta por Terzopoulos et al. para especificar esta força, que é definida com uma constante de viscosidade do fluido e com seu módulo e direção [19]. A bandeira da simulação S_4 foi feita usando este tipo de força,

U	V	Tipo	Inicio	Fim	X	Y	Z	
0	26	21	forca	0	10000	20.0	0.0	20.0
1	34	37	forca	0	10000	-20.0	0.0	-20.0
2	26	37	forca	0	10000	20.0	0.0	-20.0
3	34	21	forca	0	10000	-20.0	0.0	20.0
4	7	7	forca	0	10000	20.0	0.0	20.0
5	21	21	forca	0	10000	-20.0	0.0	-20.0
6	7	21	forca	0	10000	20.0	0.0	-20.0
7	21	7	forca	0	10000	-20.0	0.0	20.0
8	7	37	forca	0	10000	20.0	0.0	20.0
9	21	51	forca	0	10000	-20.0	0.0	-20.0
10	7	51	forca	0	10000	20.0	0.0	-20.0
11	21	37	forca	0	10000	-20.0	0.0	20.0
12	37	7	forca	0	10000	20.0	0.0	20.0
13	51	21	forca	0	10000	-20.0	0.0	-20.0
14	37	21	forca	0	10000	20.0	0.0	-20.0
15	51	7	forca	0	10000	-20.0	0.0	20.0
16	37	37	forca	0	10000	20.0	0.0	20.0
17	51	51	forca	0	10000	-20.0	0.0	-20.0
18	37	51	forca	0	10000	20.0	0.0	-20.0
19	51	37	forca	0	10000	-20.0	0.0	20.0
20	26	21	forca	0	10000	5.0	0.0	5.0
21	34	37	forca	0	10000	-5.0	0.0	-5.0



(a)

(b)

Figura 4.12: Definição e ilustração das forças pontuais.

que provoca um comportamento oscilatório, como se o tecido mergulhasse em um líquido como a água (Figura 3.10(d)). Neste caso, seria usado o coeficiente de viscosidade da água e o módulo e força do fluido seria equivalente a uma correnteza dentro da água. Para a simulação S_4 foi utilizado um coeficiente de viscosidade de $\varrho_R = 0.032 Ns/m$ e a direção de fluido $\mathbf{u} = (100, 0, 0)m/s$, com a superfície inicial posicionada no plano yz . Estes valores são então calculados com a velocidade \mathbf{vel} e o vetor normal \mathbf{n} de cada ponto da superfície, resultando em uma força \mathbf{F} tal que

$$\mathbf{F} = \varrho_R(\mathbf{n} \cdot (\mathbf{u} - \mathbf{vel})\mathbf{n}).$$

O coeficiente de viscosidade do fluido ϱ_R está rotulado como *damp R* na interface para Windows, ilustrada na Figura 4.9(a), e como *Fluido* para Linux, como mostra a Figura 4.6(b). Este campo só é preenchido quando se desejar este efeito oscilatório no tecido, que no caso da bandeira foi feito para aproximar o vento. O coeficiente ϱ_R , rotulado de *damp D* em Windows e de *Gama* em Linux, é o coeficiente de amortecimento ϱ que aparece na Equação 1.1. Ele é multiplicado pela velocidade atual de cada ponto da superfície, amortecendo sua energia potencial interna. Esta parcela corresponde ao montante de força acumulada na superfície que é dissipada pelo meio, sendo fundamental para produzir efeitos apropriados. O efeito visual do coeficiente ϱ_D é ilustrado na Figura 4.13, comparando uma simulação de um tecido preso por dois pontos caindo sob a força da gravidade com ele (Figura 4.13(a)) e sem ele (Figura 4.13(b)). Para todas as simulações mostradas nesta dissertação, tanto as de teste quanto as dos resultados do Capítulo 5, foi considerado $\varrho_D = 0.2 Ns/m$.

Por fim, os demais parâmetros referentes ao ambiente no qual a superfície do tecido se encontra,

como a aceleração e a velocidade inicial dos pontos, são especificados com a definição de vetores na janela principal do simulador e estes valores são atribuídos para cada ponto da superfície, com a diferença que a aceleração é somada para cada ponto em cada iteração, enquanto os valores da velocidade inicial são usados somente na primeira iteração, sendo ela atualizada depois em função da aceleração resultante obtida pela equação de equilíbrio.

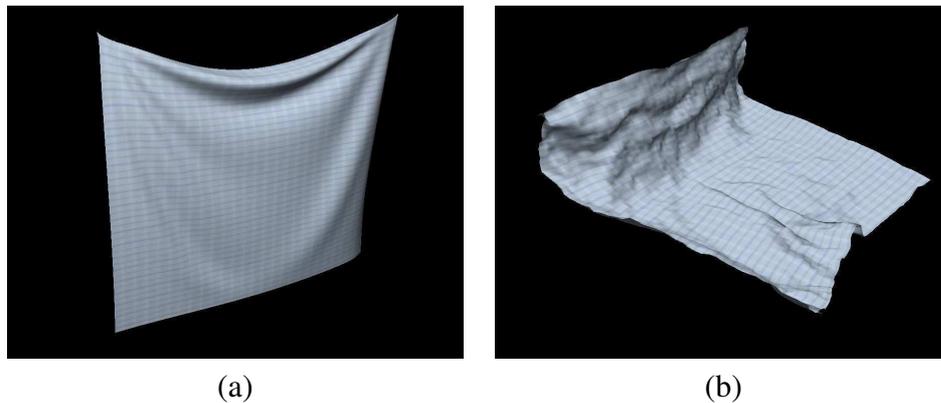


Figura 4.13: Efeitos do coeficiente ρ_D : (a) com ele e (b) sem ele.

4.3 Considerações Finais

Neste capítulo foi apresentada a implementação da proposta deste trabalho, que consiste em integrar o modelo de tecido proposto por de Melo [5] ao longo do tempo utilizando métodos explícitos. Para tanto, primeiro a modelagem dos dados envolvidos foi discutida na Seção 4.1, destacando o fluxo de dados existente na dinâmica de tecidos. Esta parte inicial da implementação diz respeito à construção de uma interface lógica que visa separar a simulação de tecidos em dois módulos independentes, um para produzir a dinâmica do tecido, chamado de *malha*, e outro para gerenciar a visualização das animações produzidas, chamado de *render*. Estes dois módulos também foram explicados nesta mesma seção. Em seguida, a Seção 4.2 apresentou a interface gráfica do simulador desenvolvido para as plataformas Linux e Windows, fornecendo alguns exemplos de simulações de tecidos nesta mesma seção para mostrar a usabilidade do *software*.

Portanto, este capítulo apresentou a parte implementacional deste trabalho usando alguns recursos de projeto orientado a objetos, deixando as demais informações disponíveis em [38], como os códigos fonte e relatórios técnicos. Além disso, a apresentação da interface gráfica permitiu a identificação dos parâmetros de simulação que são manipulados para obter distintos comportamentos de tecidos, como nos exemplos de simulação apresentados neste capítulo, que são muito importantes para entender como os resultados do Capítulo 5 foram gerados. O caimento de tecidos sobre mesas,

por exemplo, pode ser feito com o simulador desenvolvido mesmo sem que ele possua algoritmos que façam o tratamento de colisões. Este tipo de simulação pode ser feita com a definição de vários pontos fixos, nos três eixos, da parte central da malha discreta, formando um quadrado ou um círculo. No Capítulo 5 são mostradas imagens de animações de caimento e de outros tipos de simulação.

Capítulo 5

Resultados

Este capítulo apresenta os resultados obtidos no simulador desenvolvido, destacando o realismo nas imagens de animações e a precisão física. O objetivo é validar praticamente o modelo proposto por Wu e de Melo [5], mostrando os resultados visuais das principais características do modelo. Com as imagens de animações é possível constatar o realismo alcançado, enquanto a precisão física é evidenciada com seqüências de imagens que tentam mostrar a dinâmica de tecidos e também de acordo com as respostas dos diferentes materiais de tecidos às forças externas. Além disso, o controle intuitivo dos parâmetros do material do tecido neste modelo faz com que se obtenha com facilidade resultados de diferentes materiais simulados com as mesmas restrições.

Contudo, antes de mostrar efetivamente os melhores resultados visuais, a Seção 5.1 aborda algumas considerações importantes na obtenção de valores apropriados para produzir simulações de tecidos. A primeira delas refere-se à especificação das forças externas, que são definidas pelo usuário, de forma que elas podem alterar a forma do tecido. Como o simulador desenvolvido não possui o tratamento de colisões, a correta manipulação destas forças é fundamental para a obtenção dos resultados desejados. A segunda consideração é em relação à calibração dos parâmetros do material de um tecido para diferenciar determinados comportamentos, como, por exemplo, encontrar valores apropriados que diferem um tecido de *jeans* de um de algodão. A terceira consideração diz respeito à associação da resolução da malha do tecido com a qualidade visual das dobras formadas.

Os resultados são apresentados na Seção 5.2 e divididos em três subseções. Elas correspondem aos resultados visuais obtidos com simulações que utilizam deformações de: *bucklings*, destacando a formação de dobras sob ação de forças de compressão; *drappings*, mostrando caimentos de tecidos sobre objetos; e *hangings*, ilustrando a suspensão de tecidos por alguns pontos sob a ação da gravidade. A Seção 5.3 faz uma validação prática do modelo destacando o realismo obtido e também a precisão física, analisando a dinâmica dos tecidos com seqüências de imagens. A Seção 5.4 tece comentários finais.

5.1 Metodologia

Antes de exibir os resultados do simulador de tecidos é preciso salientar a forma que um determinado comportamento de tecido é atingido. As forças externas são os elementos que causam a deformação da superfície do tecido, sendo elas definidas pelo usuário na interface gráfica do simulador. Estas forças são especificadas com base no tipo de deformação que se deseja obter, conforme explicado na Subseção 5.1.1, e dependem muito do material do tecido. Para simular um material particular que compõe um tecido é necessário informar sua massa e seus coeficientes elásticos, sendo estes valores manipulados para produzir diferentes tipos de reação às forças externas. Estas reações estão relacionadas com a resistência ao curvamento e esticamento e a facilidade de formação de dobras, que são os parâmetros de material do tecido. A calibração destes parâmetros para se obter diferentes materiais é explicada na Subseção 5.1.2. Por fim, a Subseção 5.1.3 aborda os impactos causados pelas diferentes resoluções da malha do tecido.

5.1.1 Especificação de Forças Externas

As deformações que acontecem nos tecidos ocorrem devido à ação de forças externas, que são especificadas pelo usuário no simulador desenvolvido em cada simulação. Na implementação feita considera-se que as forças externas podem ser definidas de três maneiras. A primeira é definindo o módulo e a direção de uma força pontual como uma restrição da simulação. Esta força é então aplicada individualmente ao ponto da malha discreta durante o intervalo determinado, informado em termos das iterações da dinâmica. Uma outra maneira é definindo a aceleração do ambiente, que resulta na mesma força para todos os pontos da superfície. Esta força é muito usada para simular a gravidade de um ambiente. A última maneira é definindo uma força de fluido, informando a constante de densidade do fluido e sua direção. Esta força é usada frequentemente para simular o efeito causado por um vento.

As forças externas pontuais desempenham um papel fundamental na versão corrente do simulador de tecidos, que ainda não consegue tratar apropriadamente as colisões. Com elas é possível manipular a malha livremente, adicionando forças somente em um ponto ou em um conjunto de pontos. Por exemplo, para simular um pano de prato sobre uma pia sendo amassado por uma pessoa é necessário adicionar forças pontuais precisas, que simulam as colisões e fricções entre o tecido ao ser amassado. Este tipo de deformação não pode ser produzido pelas forças externas da aceleração do ambiente ou da força fluido, pois cada ponto sofre forças externas bem específicas que definem a forma da superfície do tecido. Enquanto a direção da força é mais intuitiva de ser especificada, seu módulo é mais complicado, pois depende da massa do objeto que recebe a força.

As imagens da Figura 5.1 ilustram duas aplicações de forças pontuais no simulador. A Figura 5.1(a)

corresponde a um tecido que originalmente possui a forma da Figura 5.2(a) e, após receber uma força $\mathbf{F} = (0, -9.8, 0)$ exatamente no seu ponto central, em um ambiente sem a existência da gravidade, cai levando o restante do tecido. Já a Figura 5.1(b) mostra um tecido esticado por forças posicionadas nos seus quatro cantos. Forças de módulos altos têm grandes chances de levar a simulação a uma situação de instabilidade numérica. Um bom valor de referência para as forças externas é o valor da gravidade -9.8 . Forças com módulos em torno de 10 até 100 normalmente são adequadas para a maioria das simulações de tecidos. Estes módulos também dependem da duração das forças, pois pode-se provocar um mesmo deslocamento de forma brusca ou suave. Uma cortina, por exemplo, pode ser aberta com um rápido movimento ou de maneira lenta, porém resultando no mesmo estado final do tecido.

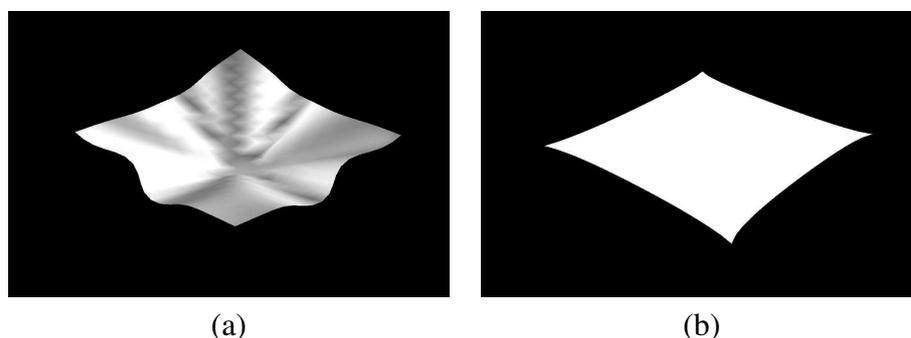


Figura 5.1: Atuação de forças externas pontuais (a) no centro e (b) nos cantos.

Com relação às forças externas da aceleração do ambiente, a definição é feita com a atribuição de um único vetor na interface gráfica do simulador. Assim, a aceleração preenchida na interface é adicionada para cada ponto da superfície, que provoca um comportamento uniforme na mesma direção dos pontos da superfície em resposta a estas forças. Nas simulações de teste do Capítulo 3 a maioria das simulações utiliza a aceleração do ambiente para definir a força de gravidade, como na formação do banco de praça e da poltrona, definindo pontos fixos e deixando a gravidade agir. Somente na simulação de teste da bandeira, S_4 , foi utilizada a força fluido.

5.1.2 Calibração de Parâmetros do Tecido

No modelo proposto por de Melo, o tipo de material que forma um tecido é definido por sua massa e por seus três coeficientes elásticos [5]. Os coeficientes elásticos são valores constantes que associam o material do tecido a determinadas características geométricas, como a resistência à mudança de métrica ($\zeta_{\alpha\beta}$), resistência à mudança de curvatura ($\xi_{\alpha\beta}$) e facilidade de formação de dobras ($\phi_{\alpha\beta}$). Estes valores são usados para obter, respectivamente, os parâmetros de material $\Phi_{\alpha\beta}$, $\Psi_{\alpha\beta}$ e $\Theta_{\alpha\beta}$, os quais são calculados uma única vez na primeira iteração da dinâmica para todos os pontos da

malha discreta, enquanto a massa é usada no cálculo da densidade de massa μ com base na área da superfície. Enquanto a massa do tecido é um valor constante, a densidade de massa é um valor variável em cada iteração, pois a superfície pode sofrer mudanças de área ao longo do tempo, ocasionadas por compressões ou distensões.

Uma consideração importante para calibrar o material de um tecido e obter o comportamento desejado é utilizar medidas apropriadas para a sua massa. Independentemente da resolução da grade da malha discreta, cada superfície tem uma área, medida pelo seu comprimento e sua largura, que está diretamente relacionada ao quanto o tecido deve pesar. Estas associações de área e peso são feitas observando informações dos fabricantes de tecidos para saber quanto pesa um metro quadrado de algodão, *jeans*, linho, veludo etc. As medidas são então usadas para aproximar de uma maneira um pouco mais real a massa dos tecidos simulados e, com isso, conseguir restringir os valores usados para calibrar as forças externas.

O coeficiente elástico ζ afeta predominantemente a métrica da superfície, de forma que baixos valores indicam uma facilidade em se esticar, enquanto os altos representam uma grande resistência a distensões, que é o comportamento mais adequado para tecidos. Já o coeficiente elástico ξ influi na curvatura da superfície, de modo que valores baixos determinam facilidade para se curvar e os altos, dificuldade. O parâmetro ϕ , chamado de fator de *buckling*, estabelece a facilidade do material em criar dobras, de forma que valores pequenos apontam pequeno acoplamento entre variações de área e de curvatura e valores altos grande acoplamento. Para tornar mais claro o significado geométrico de cada coeficiente elástico, as combinações de valores baixos e altos para cada um deles são associadas aos efeitos visuais resultantes, como mostra a Figura 5.2, com um plano fixado em todas as suas bordas sob a ação da força gravitacional $g = 9.8m/s^2$.

Considerando a superfície inicial, ilustrada na Figura 5.2(a), com massa igual a $80kg$ e orientada no plano xz , de forma que a gravidade atue na direção $(0, -9.8, 0)$, as simulações foram obtidas variando-se isoladamente os valores de cada coeficiente elástico da superfície, os quais seguem a convenção de que $\zeta = \zeta_{\alpha\beta}$, da mesma forma que $\xi = \xi_{\alpha\beta}$ e $\phi = \phi_{\alpha\beta}$. Todos os valores dos coeficientes elásticos apresentados nesta subseção estão especificados em N (newton).

O coeficiente ζ assume um valor de resistência métrica igual a 10 na Figura 5.2(b) e igual a 250 na Figura 5.2(c), ambas com ξ e ϕ iguais a 0. Pode-se notar que o coeficiente ζ está associado à resistência métrica da superfície, causando um esticamento quando ela, presa em suas quatro bordas, é deformada pela ação da gravidade. Considerando possível que um tecido varie 10% de sua área [11], os valores adequados para ζ variam entre 80, como em tecidos de elastano, até 500, como em lonas bem rígidas. Como em geral os tecidos são compostos de muitos materiais, qualquer combinação dentro deste alcance resulta em uma resistência métrica similar a de um tecido.

Já o coeficiente elástico ξ está associado à suavidade de curvamento da superfície. A Figura 5.2(d)

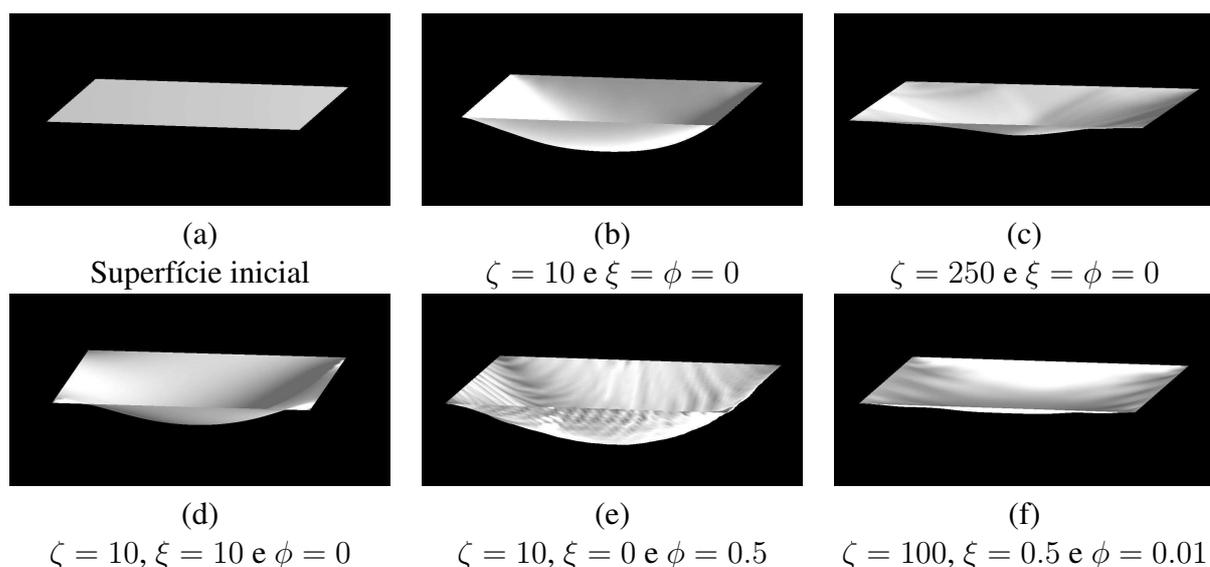


Figura 5.2: Influência dos coeficientes elásticos.

possui o mesmo valor de ζ que a Figura 5.2(b), só diferindo o valor de ξ , que ao invés de 0 é alterado para 10. Pode-se perceber que a curva formada pela região esticada na imagem 5.2(d) não está tão acentuada quanto em 5.2(b), de forma que com o mesmo valor de $\zeta = 10$, a elevada resistência à mudança de curvatura faz com que haja menor deformação da métrica da superfície. O coeficiente ξ possui um alcance diferente de ζ para simular tecidos, sendo mais sensível numericamente devido ao fato da resistência da métrica ser muito maior do que a da curvatura. Os valores devem variar entre 0 e 10, pois a partir deste último valor as superfícies possuem o comportamento de lâminas plásticas.

Por fim, o parâmetro ϕ está relacionado com a frequência de formação de dobras que acontecem nos tecidos. Este parâmetro difere de ξ porque não descreve a suavidade ou a resistência das dobras, mas a capacidade do material em formar as dobras. Esta relação entre a métrica e a curvatura é descrita com ϕ , como mostrado na Figura 5.2(e), que usa os mesmos valores de 5.2(b), mas com $\phi = 0.5$. Pode-se notar que a mudança do valor de ϕ fez com que várias dobras se formassem ao longo de todo o tecido. Devido ao coeficiente ϕ associar duas grandezas de comportamentos opostos, seu alcance é ainda mais sensível numericamente que ξ , variando de 0 a 1 para simular tecidos. Valores acima de 1 costumam parecer com oscilações dentro do plano da superfície do tecido, muito comum nos modelos do tipo massa-mola.

Considerando que os tecidos em geral possuem o tipo de deformação relacionado ao coeficiente elástico ϕ , é sempre interessante simular um tecido com um valor não nulo para ϕ , mesmo que da ordem de 0.01, para se obter um comportamento físico mais real. A Figura 5.2(f) ilustra a composição de todos os coeficientes elásticos usados em conjunto dentro dos alcances citados, sendo $\zeta = 100$, $\xi = 0.5$ e $\phi = 0.01$. Pode-se perceber na Figura 5.2(f) que, embora a simulação use um valor de ζ

menor do que o usado na Figura 5.2(c), a curva formada na região inferior às bordas fixadas apresenta uma suavidade maior. Além disso, como $\phi = 0.01$ na Figura 5.2(f), a imagem apresenta mais dobras na lateral do que a Figura 5.2(c), que tem $\xi = 0$ e $\phi = 0$.

Uma forma de mostrar a relação entre as grandezas referentes à métrica e à curvatura é através da simulação da Figura 5.2(b), na qual uma superfície de baixa resistência métrica, fixada em toda sua fronteira, sofre a ação da gravidade nos seus pontos internos e se estica, criando área e mudando a curvatura original da superfície. Outra maneira de se ressaltar esta dependência é no modo com que as Figuras 5.2(c) e 5.2(d) diminuem a inclinação da curva formada em 5.2(b). Na Figura 5.2(c) a resistência à métrica é maior, logo a distensão da área é menor e o curvamento é menor. Na Figura 5.2(d) a resistência à métrica é a mesma de 5.2(b), mas a resistência à curvatura é maior, de forma que mesmo que a área aumente, a resistência ao curvamento limita as dobras, resultando também em um curvamento final menor que 5.2(b). A Figura 5.3 ilustra a influência do coeficiente elástico ϕ , na qual uma superfície inicial recebe a ação de forças puramente compressivas e opostamente orientadas, sendo (a) com $\phi = 0$ e (b) com $\phi = 0.01$.



Figura 5.3: Influência de ϕ .

Assim, com base na análise feita sobre estes parâmetros, é possível propor um procedimento composto por cinco passos para calibrá-los de maneira apropriada, conforme descrito a seguir:

1. A densidade de massa do tecido é estimada com base nas especificações técnicas fornecidas pelos fabricantes.
2. Um conjunto de forças é definido em analogia a situações físicas reais.
3. Considerando $\xi_{\alpha\beta} = \phi_{\alpha\beta} = 0$, varia-se o valor de $\zeta_{\alpha\beta}$ até que se obtenha uma superfície visualmente sem esticamentos. Outra forma de fazer este passo é monitorando a mudança da área que é informada na visualização da animação, tomando como limite superior uma variação de 10% em relação à área inicial. Na maioria dos casos usou-se o valor $\zeta_{\alpha\beta} = 100.0$, entretanto,

notou-se que este valor é dependente da massa da superfície. Quanto maior é a massa, maior tem que ser o valor de $\zeta_{\alpha\beta}$.

4. Fixando $\zeta_{\alpha\beta} = 100.0$ e mantendo $\phi_{\alpha\beta} = 0$, testa-se empiricamente alguns valores para dois tipos de material, até que os comportamentos visuais possam ser comparados com situações reais. Estes dois tipos de material de tecido correspondem à seda, que é um material muito leve e macio, e à lona, que é um material pesado e rígido. A partir destes dois extremos, é possível estimar intuitivamente outros materiais com base no grau de rigidez que algum material apresenta dentro deste intervalo.
5. Estabelecendo os valores de $\zeta_{\alpha\beta}$ e $\xi_{\alpha\beta}$, determina-se o valor de $\phi_{\alpha\beta}$ aplicando forças de compressão e distensão e comparando os efeitos visuais com os reais. Quanto maior é este valor, maior o número de dobras formadas.

Este procedimento consegue encontrar empiricamente uma configuração apropriada para um material de tecido. A ordem dos passos é muito importante, pois ela parte primeiro da massa do tecido, a qual é definida da forma mais precisa possível considerando valores reais, para então estabelecer forças externas que são valores dependentes desta massa. A seguir, os coeficientes elásticos do tecido são calibrados um a um de forma empírica, sendo primeiro ζ referente à resistência à mudança de métrica, depois ξ referente à resistência à mudança de curvatura e, por último, ϕ correspondente à facilidade de formação de dobras. Embora seja um procedimento empírico, a realimentação existente com resultados visuais facilita a calibração dos valores para obter um comportamento desejado. Para a execução das simulações auxiliares até a obtenção da simulação final recomenda-se usar um passo de tempo em torno de $0.002s$ e, somente após calibrar corretamente os parâmetros para uma simulação estável, aumentar este valor, atualizando as restrições se necessário.

5.1.3 Resolução da Malha

Outro fator importante relacionado à simulação de tecidos é a resolução espacial da malha, que corresponde ao número de pontos que formam a superfície do tecido. Quanto maior for o número de pontos que formam a geometria do tecido, maior será a qualidade visual na representação de suas dobras. As imagens mostradas no Capítulo 3, por exemplo, utilizavam malhas de resolução 60×60 no máximo, resultando em 3600 pontos. Já os *bucklings* apresentados na Figura 5.5 foram todos feitos com malhas de resolução 120×120 , resultando em 14400 vértices. Nota-se que a diferença na qualidade das imagens é muito grande. Para salientar o impacto do número de pontos que formam a superfície na qualidade da imagem obtida, a Figura 5.4 mostra simulações de *bucklings* dos mesmos quatro panos da Figura 5.5, porém utilizando malhas de resolução 60×60 . Nesta comparação é

importante manter o valor dos espaçamentos Δx^1 e Δx^2 para que a discretização espacial não fique nem muito grosseira, estimando os elementos do modelo com pouca precisão, e nem muito pequena, a ponto de se tornar instável numericamente.

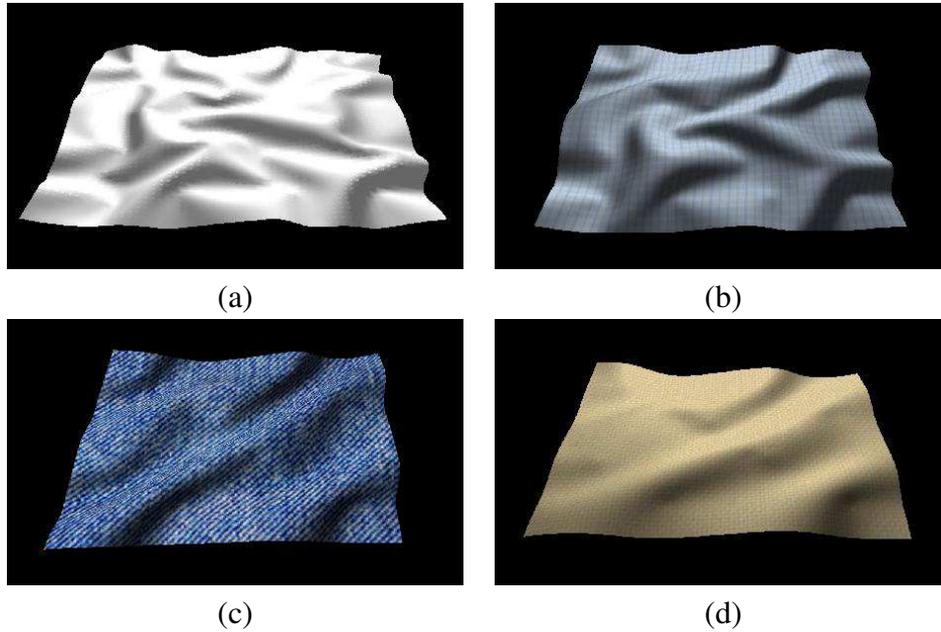


Figura 5.4: *Bucklings* em grades 60×60 .

Embora a qualidade das dobras caia bastante ao usar os mesmos parâmetros para os materiais e modificar apenas a resolução espacial da malha discreta, os resultados continuam condizentes com a física, ou seja, o tecido de seda é o que apresenta mais dobras, depois o algodão, o *jeans* e, por último, a lona. Este comportamento é reflexo dos valores usados na definição dos materiais.

5.2 Simulações

Nesta seção são apresentadas as simulações geradas com o modelo de tecido proposto por de Melo [5] integrado ao longo do tempo com o método de Verlet. Estas simulações estão divididas de acordo com o tipo de deformação que o tecido sofre. A ação de amassar, enrugar ou criar dobras em um tecido é conhecida como *buckling*, sendo esta deformação muito pesquisada na literatura [13, 12, 22]. As simulações de *bucklings* são mostradas com deformações causadas por forças pontuais sem a atuação da força de gravidade, conforme detalhado na Subseção 5.2.1. Outro tipo de deformação comum são os caimentos de tecidos sobre objetos, conhecidos como *drappings* e apresentados na Subseção 5.2.2. Como neste trabalho não existe o tratamento de colisões, as simulações deste tipo são feitas com a definição de pontos fixos ao longo do tempo que simulam a geometria de objetos de

colisão. Este tipo de deformação é muito comum para simular toalhas caindo sobre mesas com a ação da gravidade, como fez Breen et al. [26]. Algumas vezes, a suspensão de tecidos também é conhecida como *drappings*, mas neste trabalho elas serão chamadas de *hangings* e ilustradas na Subseção 5.2.3. Elas correspondem a simulações de cortinas e bandeiras, por exemplo.

5.2.1 *Bucklings*

As simulações de *bucklings* podem ser dividida de duas formas: as que utilizam somente forças de compressão e distensão tangentes ao plano da superfície, e as demais, que utilizam forças pontuais em qualquer direção. Uma das contribuições do modelo proposto por de Melo [5] é exatamente a de conseguir produzir efeitos de *buckling* somente com forças de compressão. As imagens da Figura 5.5 ilustram quatro tecidos submetidos às mesmas forças de compressão, porém variando o material do tecido. A Figura 5.5(a) ilustra um tecido de seda, o qual é muito permissivo à formação de dobras, como pode-se notar. O material do tecido da Figura 5.5(b) é o algodão, sendo possível notar que existem menos dobras em relação ao de seda. De forma semelhante, o tecido de jeans na Figura 5.5(c) e o de lona na Figura 5.5(d) também possuem quantidades de dobras diferentes.

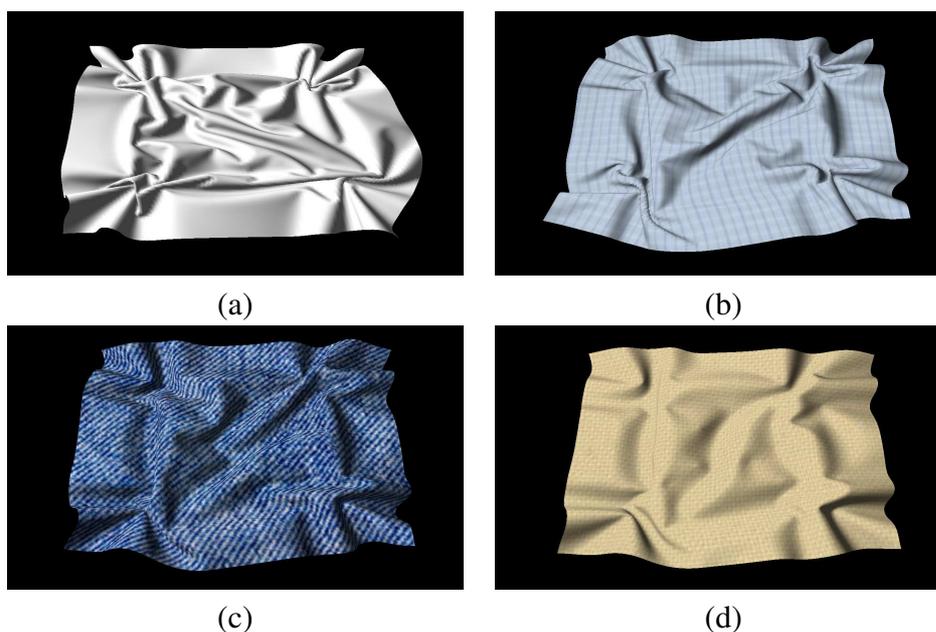


Figura 5.5: *Bucklings* de diferentes materiais de tecido.

O diferencial deste tipo de resultado é que as respostas da superfície na direção para fora do plano acontecem naturalmente, sem a imposição de forças artificiais nestas direções quando a superfície é submetida apenas a forças de compressão. Estas situações acontecem quando se aperta algum tipo de tecido no ar. Naturalmente, ao se comprimir um tecido, ele se curva na direção perpendicular à

superfície, como o comportamento das imagens da Figura 5.5. As diferenças entre os parâmetros usados para obter estas simulações são os valores de massa e principalmente os coeficientes elásticos $\xi_{\alpha\beta}$, o qual está relacionado com a dificuldade do material se curvar. Além disso, os coeficientes $\phi_{\alpha\beta}$ não são nulos, pois quando ocorrem somente forças de compressão o coeficiente de *buckling* é fundamental para obter resultados corretos.

Um segundo tipo de simulação de *buckling* que pode ser feito é usando forças pontuais que não sejam estritamente tangenciais ao plano da superfície. Desta forma, uma maneira de produzir uma simulação interessante deste modo é aplicando algumas forças de compressão e depois ir incrementando forças para fazer pequenos ajustes na superfície do tecido. As imagens da Figura 5.6 apresentam duas gravatas borboletas, sendo 5.6(a) uma gravata de material mais flexível (seda) e 5.6(b) mais rígido (veludo). Estas imagens foram obtidas adicionando forças de compressão em alguns pontos centrais da superfície do tecido na direção vertical, dividindo-as em sentidos opostos de cada lado para evitar auto-colisões. Depois da formação das dobras no meio do tecido, que correspondem a parte achatada da gravata borboleta, forças para fora do plano são adicionadas para fazer pequenos ajustes na gravata. Estes ajustes correspondem ao que ocorre na realidade, quando ao vestir uma gravata deste tipo puxa-se com as mãos suas extremidades para que seus lados fiquem iguais. Isto acontece porque como o tecido é muito flexível, ao fazer forças de compressão ele pode tanto se curvar para um lado quanto para o outro e, portanto, o ajuste ocorre no sentido de deixar as dobras para o mesmo lado.

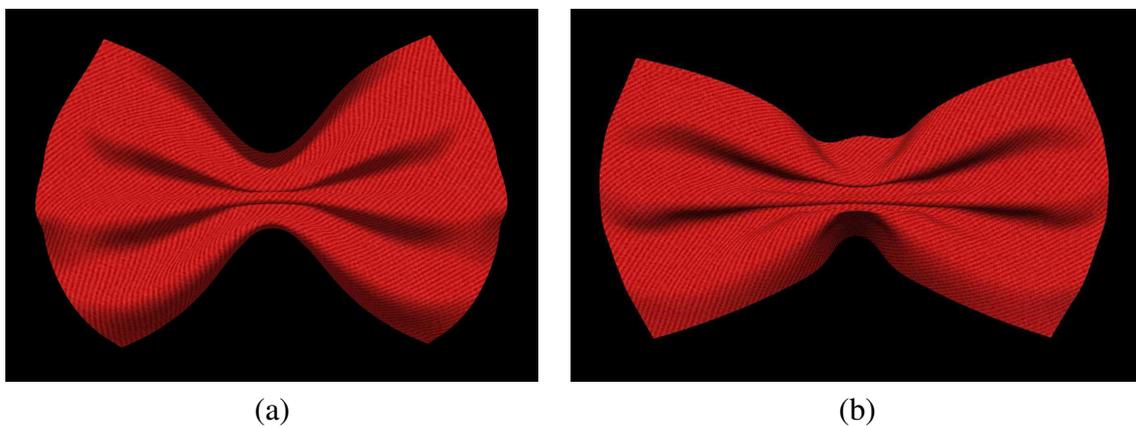


Figura 5.6: Formação de uma gravata borboleta (a) flexível e (b) rígida.

Pode-se notar que a Figura 5.6(a) apresenta os contornos mais arredondados do que a Figura 5.6(b). Esta diferença, ao contrário das imagens da Figura 5.5, ocorre simplesmente por valores dos coeficientes elásticos $\xi_{\alpha\beta}$ distintos, pois as duas gravatas borboletas estão definidas com a mesma massa e com os mesmos coeficientes elásticos $\zeta_{\alpha\beta}$ e $\phi_{\alpha\beta}$.

5.2.2 *Drapings*

As simulações de tecidos que consistem de caimentos da superfície do tecido sobre objetos são conhecidas como *drappings*. Como este trabalho não utiliza algoritmos que façam o tratamento de colisões, os resultados mostrados nesta subseção utilizam a definição de pontos fixos como restrições da simulação para simular a geometria dos objetos de colisão. Assim, um caimento de uma toalha sobre uma mesa redonda ou quadrada é feita definindo o formato da mesa com pontos fixos sobre a malha discreta. A Figura 5.7 ilustra os pontos fixos como bolas vermelhas sobre um reticulado de pontos que representa a malha discreta. A duração dos pontos fixos é definida em termos das iterações da simulação, fazendo com que cada ponto seja fixo nos três eixos a partir de uma certa iteração até a iteração final, deixando o caimento acontecer com a força da gravidade. Este caimento termina quando a superfície entra em um estado de equilíbrio, que no caso de um caimento sobre uma mesa corresponde ao momento em que a área do tecido que não está sobre a mesa não apresenta mais movimentos entre um instante de tempo e o seguinte.

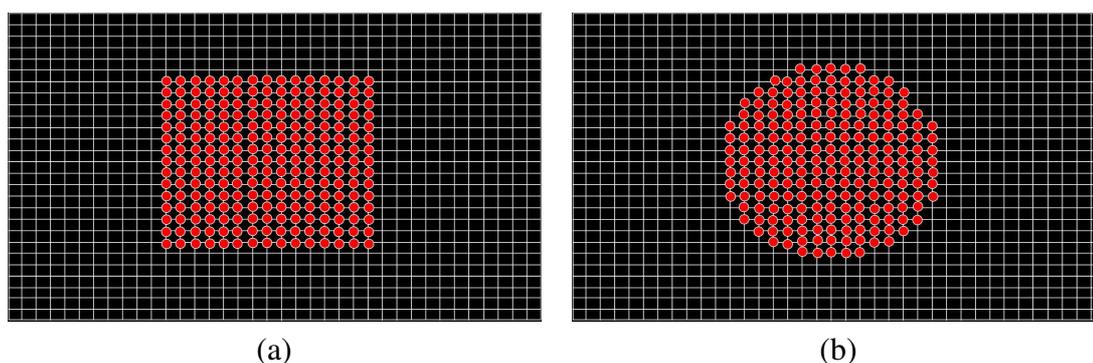


Figura 5.7: Pontos fixos na malha discreta para formar uma mesa (a) quadrada e (b) redonda.

As resoluções espaciais das malhas discretas da Figura 5.7 são apenas ilustrativas, pois pode-se definir a mesma simulação de *draping* com resoluções diferentes, ou até outras simulações similares deixando mais ou menos pontos para a definição do plano da mesa. Uma vez definidas as restrições de pontos fixos sobre uma malha discreta pode-se usá-las variando apenas o material do tecido e obter diferentes efeitos. Por exemplo, a Figura 5.8(b) ilustra o caimento de um tecido de cetim sobre uma mesa redonda, já a Figura 5.8(d) apresenta a imagem de uma simulação com as mesmas restrições porém com um tecido de linho. Ambas as imagens são comparadas com uma foto real da situação, conforme as fotos da Figura 5.8(a) e 5.8(c). Desta forma, é possível ter uma referência do comportamento do tecido durante o procedimento de calibração do material explicado na Seção 5.1, para que a cada tentativa de configuração dos parâmetros a imagem final do estado de equilíbrio atingido possa ser comparada com o caimento da foto real. Esta é uma forma interessante de avaliar o realismo obtido nas imagens, embora na prática nem sempre é possível ter uma foto de referência.

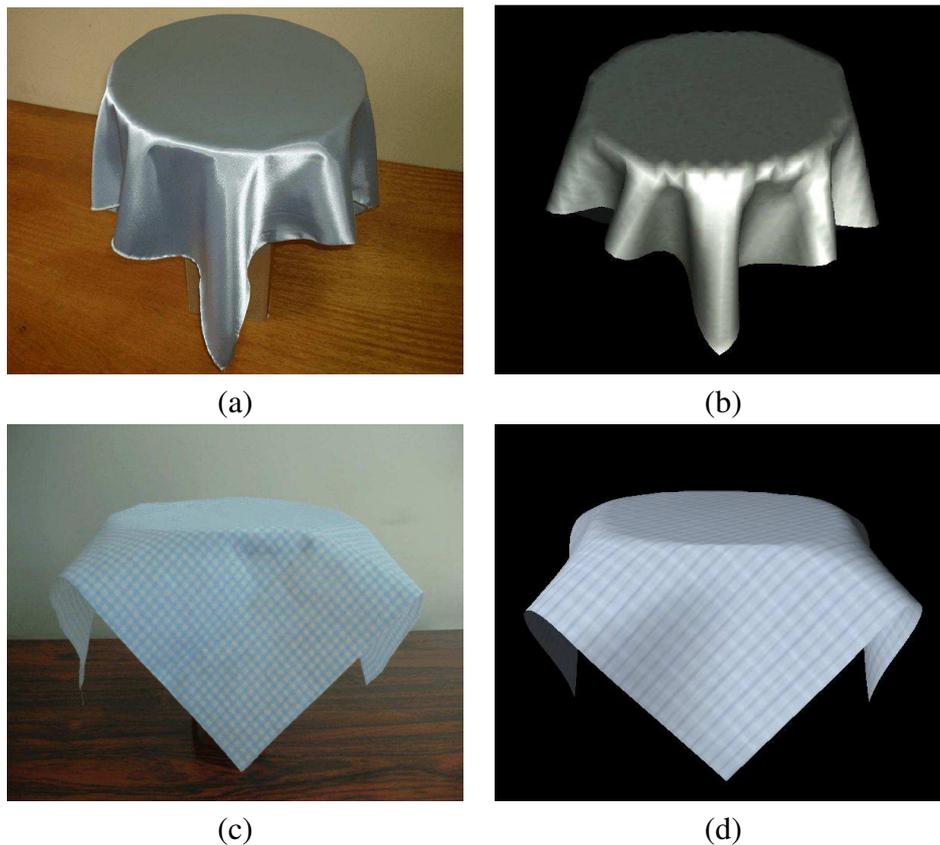


Figura 5.8: Caimentos de tecidos sobre mesas redondas: (a) real cetim, (b) cetim, (c) real linho e (d) linho.

Nestas imagens os parâmetros do material do tecido usados para o cetim são muito próximos dos valores usados para o linho, variando apenas a massa, sendo o linho mais pesado que o cetim, e também o valor do coeficiente elástico $\xi_{\alpha\beta}$, pois o linho tem maior resistência à mudança de curvatura do que o cetim. Mais um exemplo da facilidade em aproveitar as restrições e definições dos parâmetros da simulação e mudar apenas um valor para obter resultados bem diferentes é ilustrado na Figura 5.9(a), na qual usou-se a mesma massa do tecido de cetim mas com uma resistência à curvatura bem menor, como se fosse um algodão.

Com relação ao caimento de um tecido sobre uma mesa quadrada a simulação é feita de forma muito semelhante em relação a uma mesa redonda. Usando uma malha com os pontos fixos configurados de acordo com a Figura 5.7(a), a Figura 5.9(b) ilustra o resultado de um tecido usado para cobrir mesas de cozinha ou sala, que normalmente são de algodão mas contém uma camada de plástico na superfície, sendo mais rígido que o algodão porém tão leve quanto. Desta forma, as dobras formadas nos cantos do tecido são bem armadas e com dobras bem definidas. Este tipo de dobra formada na região da fronteira do tecido era um problema nas imagens obtidas no trabalho proposto por de Melo [5], pois como vários elementos do modelo eram zerados nestes pontos as dobras eram mal

formadas. Com a discretização espacial apresentada na Seção 3.1 deste trabalho, foi possível obter melhores resultados neste aspecto e assim aumentar o realismo das imagens obtidas.

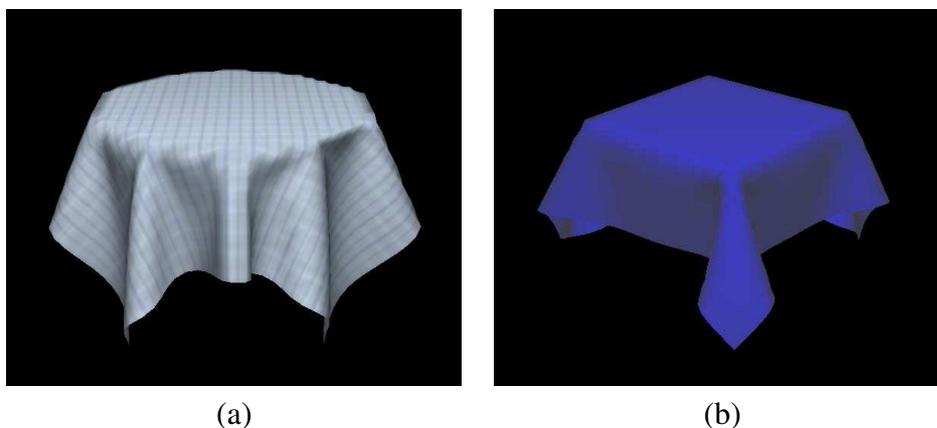


Figura 5.9: Caimento de tecido sobre uma mesa (a) redonda e (b) quadrada.

As simulações de *draping* também podem ser feitas com objetos mais complexos que mesas, demandando para isso um maior esforço na definição das restrições por falta de tratamento de colisões e auto-colisões. Para definir um objeto mais detalhado, como uma poltrona, é preciso definir pontos fixos em diferentes instantes de tempo que formarão as partes da poltrona de forma isolada. Desta forma, à medida que a superfície inicial do tecido cai de acordo com a gravidade, primeiro os pontos do encosto da poltrona são fixados para depois serem fixados os braços da poltrona e por último o assento. A Figura 5.10 ilustra o estado final de equilíbrio no caimento de um tecido de veludo sobre uma poltrona.

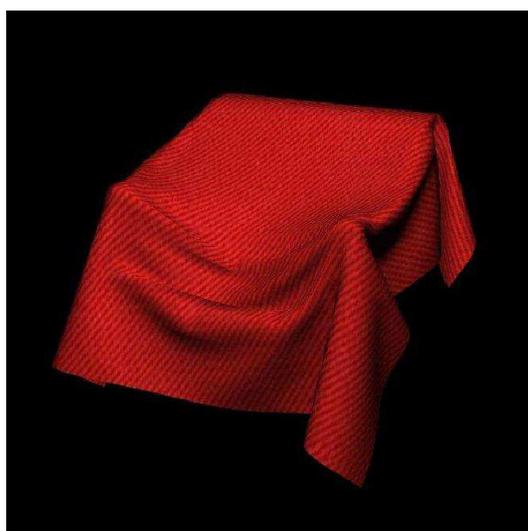


Figura 5.10: Caimento de um tecido sobre uma poltrona.

5.2.3 Hangings

Nas simulações de *hangings* o tecido é suspenso por alguns pontos fixos e a ação da gravidade provoca as deformações nele. Os exemplos mais comuns deste tipo de simulação são cortinas presas em trilhos e bandeiras hasteadas em um poste. Em relação às cortinas, as simulações podem ser feitas definindo pontos fixos em uma linha de pontos da malha discreta próxima da fronteira superior do tecido, de forma a permitir que estes pontos se desloquem somente no eixo horizontal para que a cortina possa ser aberta e fechada. As imagens da Figura 5.11 ilustram quatro simulações de cortinas com materiais de tecidos diferentes. A cortina da Figura 5.11(a) ilustra um tecido bem flexível, como uma cortina de algodão. A Figura 5.11(b) mostra uma cortina mais rígida, como uma cortina de sala relativamente pesada, em que as dobras são armadas. Já a Figura 5.11(c) exibe uma cortina de banheiro, que possui um material leve e plastificado, permitindo a formação de dobras armadas também. Todas estas simulações são feitas partindo de uma superfície lisa no plano xy e, após a definição dos pontos fixos que formam o trilho da cortina, forças pontuais são inseridas nas duas extremidades da cortina para simular o movimento de abri-la até certo ponto. Após algumas iterações, os pontos fixos do trilho são restritos em todos os eixos, inclusive no eixo horizontal, e o restante do tecido demanda ainda um número considerável de iterações até que suas áreas nas extremidades laterais caiam de forma a ficar o mais perpendicular possível ao trilho da cortina.

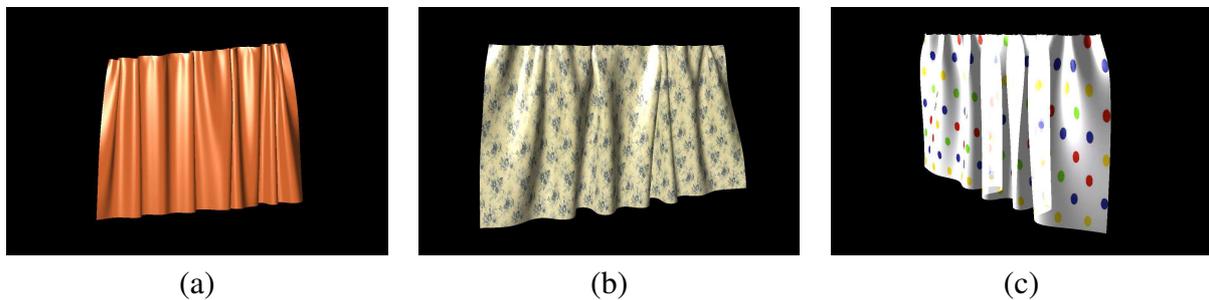


Figura 5.11: Cortinas suspensas de diferentes materiais: (a) seda; (b) algodão; e (c) plastificada.

Outro exemplo de uma simulação de *hanging* é uma bandeira balançando de acordo com a força do vento. Neste caso, dois pontos extremos da lateral da superfície do tecido são definidos como fixos em um eixo, simulando os pontos em que a bandeira está amarrada a um poste. Para obter o efeito da força do vento pode-se usar a força fluido definindo uma direção e uma densidade de fluido apropriada. Assim, a partir da superfície inicial definida no plano yz , conforme mostra a Figura 5.12(a), a força de fluido é especificada com a direção $\mathbf{F} = (100, 0, 0)N$ e com uma densidade $0.032Ns/m$, fazendo com que a bandeira inicialmente caia de acordo com a gravidade ao mesmo tempo que se desloca em direção da força fluido. Após algumas iterações, a bandeira que estava no plano yz agora encontra-se no plano xy e, uma vez atingido este plano, a superfície começa a

sofrer deformações que, embora não sejam iguais, repetem um certo padrão após algumas iterações em relação à formação de dobras. A imagem da Figura 5.12(b) ilustra a bandeira já deformada pela ação da força fluido. Este tipo de deformação pode ser interpretado como um mergulho da superfície em algum líquido ou meio, os quais podem ter uma força associada.

Em relação ao estado de equilíbrio final desta simulação, diferentemente das outras apresentadas até agora, a bandeira não atinge um estado em que não existam mais movimentos entre dois passos subseqüentes, a não ser que a força fluido seja interrompida. No capítulo 3, durante a execução das simulações de teste para descobrir qual o melhor método explícito que poderia ser aplicado ao modelo proposto por de Melo [5], foi considerado como estado de equilíbrio da bandeira o momento em que a superfície consegue pela primeira vez se manter em uma linha horizontal, mesmo com a ação da gravidade. A imagem da Figura 5.12(b) foi escolhida pensando neste equilíbrio, mas também visando mostrar o maior número de dobras produzidas por este tipo de deformação.

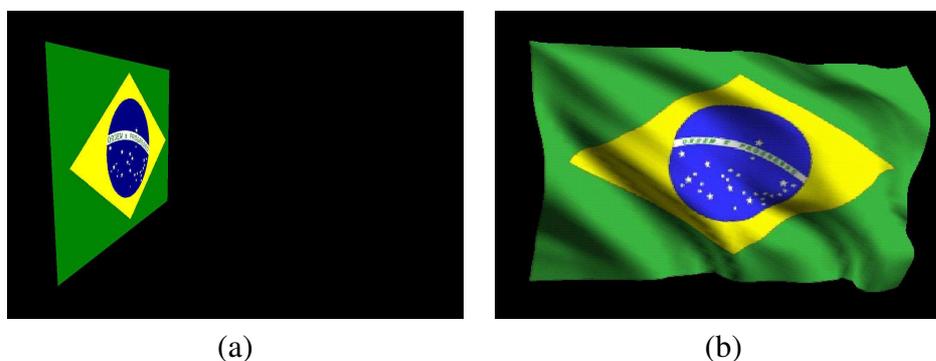


Figura 5.12: Bandeira no estado (a) inicial e (b) deformado.

5.3 Validação do Modelo

Almejando a validação prática do modelo proposto por de Melo, nesta seção são analisados os resultados mostrados na seção anterior, abordando o realismo nas imagens geradas e o movimento de tecidos, com também a intuitividade na definição dos parâmetros de uma simulação. A dinâmica dos tecidos é analisada com seqüências de imagens na Subseção 5.3.1, mostrando as variações da superfície inicial até a superfície deformada em estado de equilíbrio. A intuitividade dos parâmetros de simulação é discutida na Subseção 5.3.2. Nesta subseção apresenta-se uma tabela com os valores dos principais parâmetros do tecido e da simulação utilizados nas simulações apresentadas neste capítulo. A análise do tempo de processamento não é feita devido ao fato de se utilizar o método de Verlet, o qual apresenta complexidade computacional linear, como mostrado no capítulo 3. Contudo, os tempos por iteração de cada simulação são mostrados na Tabela 5.1 da Subseção 5.3.2.

5.3.1 Dinâmica

O objetivo desta subseção é mostrar como é a dinâmica de tecidos gerada no simulador desenvolvido para destacar a qualidade do modelo de convergir para um estado de equilíbrio condizente com a expectativa. A melhor forma de mostrar a dinâmica de tecidos é com a visualização de animações, para que se perceba efetivamente a continuidade nos movimentos e avalie-se o realismo atingido. Assim, todas as imagens mostradas ao longo desta seção são quadros de animações, que podem ser encontradas no disco em anexo a esta dissertação. Contudo, na tentativa de transmitir esta dinâmica sem o recurso da animação, esta subseção apresenta seqüências de imagens das diferentes geometrias da superfície ao longo do tempo. Para os principais resultados, são usados quatro quadros diferentes associados ao tempo em que eles foram obtidos. O primeiro quadro corresponde à superfície inicial e o último ao estado de equilíbrio ou a um determinado comportamento esperado. A Figura 5.13 apresenta as seqüências de imagens de *buckling* dos quatro panos exibidos na Figura 5.5. Ela ilustra uma das propriedades inéditas do modelo proposto por de Melo, que é a formação natural de dobras sob forças de compressão [5].

Na imagem 5.13 pode-se notar que as geometrias das quatro superfícies iniciais são iguais, embora sejam compostas de materiais diferentes e resultem em diferentes deformações. Como os materiais são diferentes, as massas dos tecidos são diferentes, portanto, é preciso atualizar os módulos das forças externas que atuam sobre o tecido para cada simulação. Isto é feito para que as deformações aconteçam de forma proporcional, destacando a formação de dobras de diferentes materiais após um determinado tempo de referência. Ao se aplicar uma força muito pequena em um tecido de seda, por exemplo, ele se deformará mesmo que lentamente, contudo, um tecido de lona pode permanecer imóvel com a aplicação da mesma força. Ao todo são aplicadas vinte forças externas de mesmo módulo de forma distribuída, fazendo o tecido se amassar em torno de seu centro (Figura 4.12(b)). Estas forças são exercidas até o tecido apresentar a maior deformação possível sem que ocorram auto-colisões, quando então o módulo das forças é diminuído para a superfície entrar em equilíbrio com as dobras formadas. Esta simulação é equivalente a situação em que uma pessoa faz certa força para deformar um tecido e, após deformado, o segura com menor força para manter as dobras formadas. É interessante ressaltar que, diferentemente dos modelos existentes, nenhuma força de fricção artificial [14, 25, 30] foi necessária para produzir dobras.

A imagem da Figura 5.13(b) mostra as deformações da superfície após 2s sob a ação das forças externas. Pelo fato da seda ser muito flexível, rapidamente o tecido se deforma formando diversas dobras. Nas imagens da coluna (b) pode-se notar que mesmo nos instantes iniciais de deformação já é possível verificar qual tecido é mais permissivo para se curvar em relação ao outro, que no caso, após a seda é o algodão, *jeans* e por último, a lona. Na Figura 5.13(c) e nas outras três desta coluna são mostradas as geometrias da superfície alguns instantes de tempo à frente. Percebe-se que como

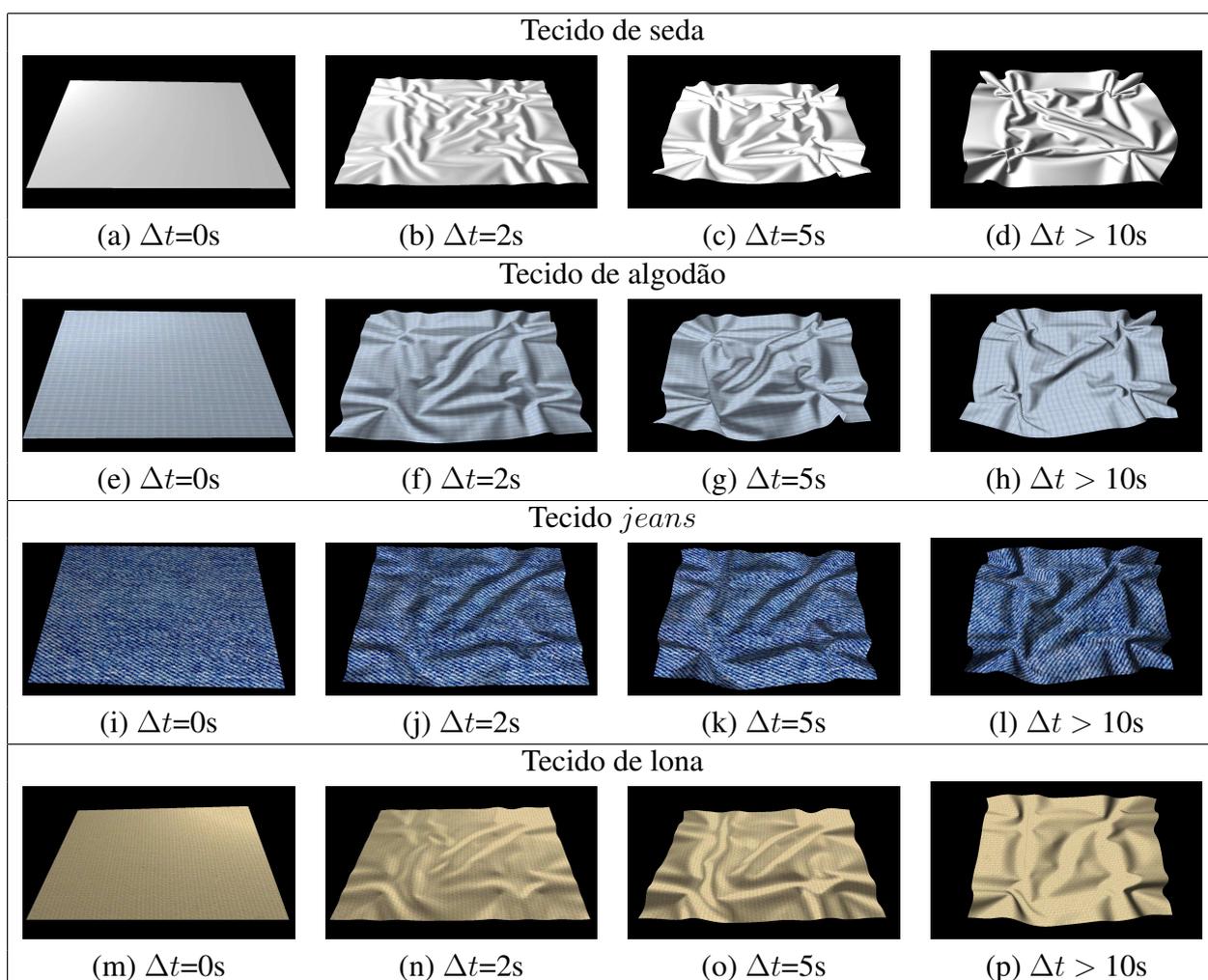


Figura 5.13: Dinâmica de *bucklings*.

as forças externas são aplicadas no sentido para o centro do tecido as dobras começam a ficar mais altas, diminuindo a área projetada no plano da superfície inicial. Após mais algum tempo, as forças são reduzidas para apenas manter as dobras criadas e o tecido entra em estado de equilíbrio, que corresponde às imagens da coluna (d). Nesta última coluna pode-se notar que algumas dobras antes criadas já não existem mais e, no caso da seda e do algodão, as dobras remanescentes são bem finas e acentuadas, enquanto as do *jeans* e da lona são grossas e suaves.

Com relação aos resultados de *drapings*, foram selecionadas as simulações do caimento de tecidos sobre a mesa redonda de algodão e sobre a poltrona. Nestas simulações o tecido cai verticalmente e, à medida que avança encontra restrições de pontos fixos que simulam o seu contato com a poltrona. A imagem da Figura 5.14(a) mostra a superfície inicial que cai sobre uma mesa redonda. Na Figura 5.14(b) já é possível ver que os pontos associados à mesa já foram fixados, restando cair

as áreas do tecido que ficarão penduradas. Na Figura 5.14(c) pode-se ver a criação de dobras na região do tecido que não está sobre a mesa e, na Figura 5.14(d) o tecido já se encontra em estado de equilíbrio. A dinâmica do tecido que cai sobre a poltrona também é feita da mesma forma, diferindo apenas no material do tecido e na definição dos pontos fixos. Partindo de uma superfície inicial como na Figura 5.14(e), o tecido cai de acordo com a gravidade e o primeiro grupo de pontos fixos que são executados são os associados à parte de cima do encosto da poltrona, conforme a Figura 5.14(f). Em seguida, a Figura 5.14(g) mostra a acomodação do tecido sobre os braços da poltrona e 5.14(h) o estado de equilíbrio.

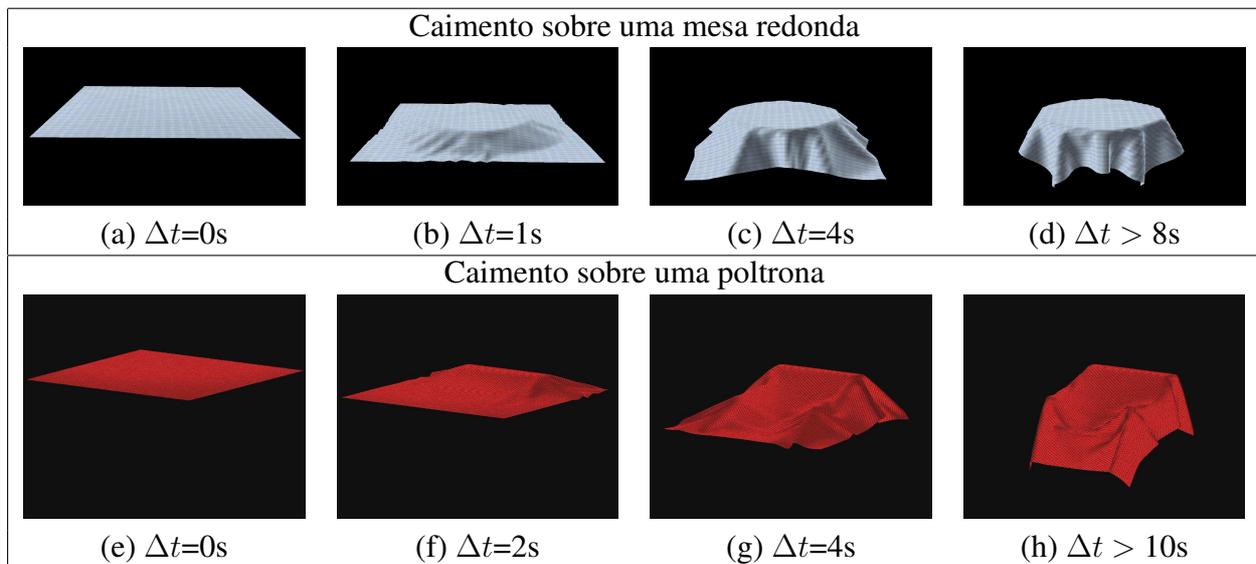


Figura 5.14: Dinâmica de *drappings*.

É interessante notar que a qualidade das animações pode ser comprometida pela falta de um tratamento apropriado de colisões e auto-colisões. Embora definir os pontos de colisão como pontos fixos seja uma solução visualmente aceitável, a desconsideração de movimentos sutis, como escorregamento do tecido sobre objetos de colisão devido a fricções, pode gerar efeitos um tanto artificiais.

Por fim, para mostrar a dinâmica dos resultados de *hanging*, foram selecionadas as simulações da bandeira balançando com a força do vento e dois tipos de cortina, a mais flexível e a mais rígida. A Figura 5.15(a) mostra a superfície inicial da cortina laranja da Figura 5.11(a). Possuindo um material bastante maleável, esta cortina sofre forças externas em cada uma de suas extremidades no sentido de sua parte central, como se a cortina estivesse sendo aberta. A imagem da Figura 5.15(b) mostra as primeiras deformações causadas na superfície e 5.15(c) corresponde ao exato momento em que a superfície atingiu seu limite de dobras sem auto-colisões. Neste instante, os pontos do trilho são fixados nos três eixos e então o tecido balança lentamente até parar de se mexer, que corresponde

ao estado de equilíbrio ilustrado na imagem da Figura 5.15(d). Esta cortina flexível foi simulada considerando um material leve, como a seda, visando ser uma cortina de sala.

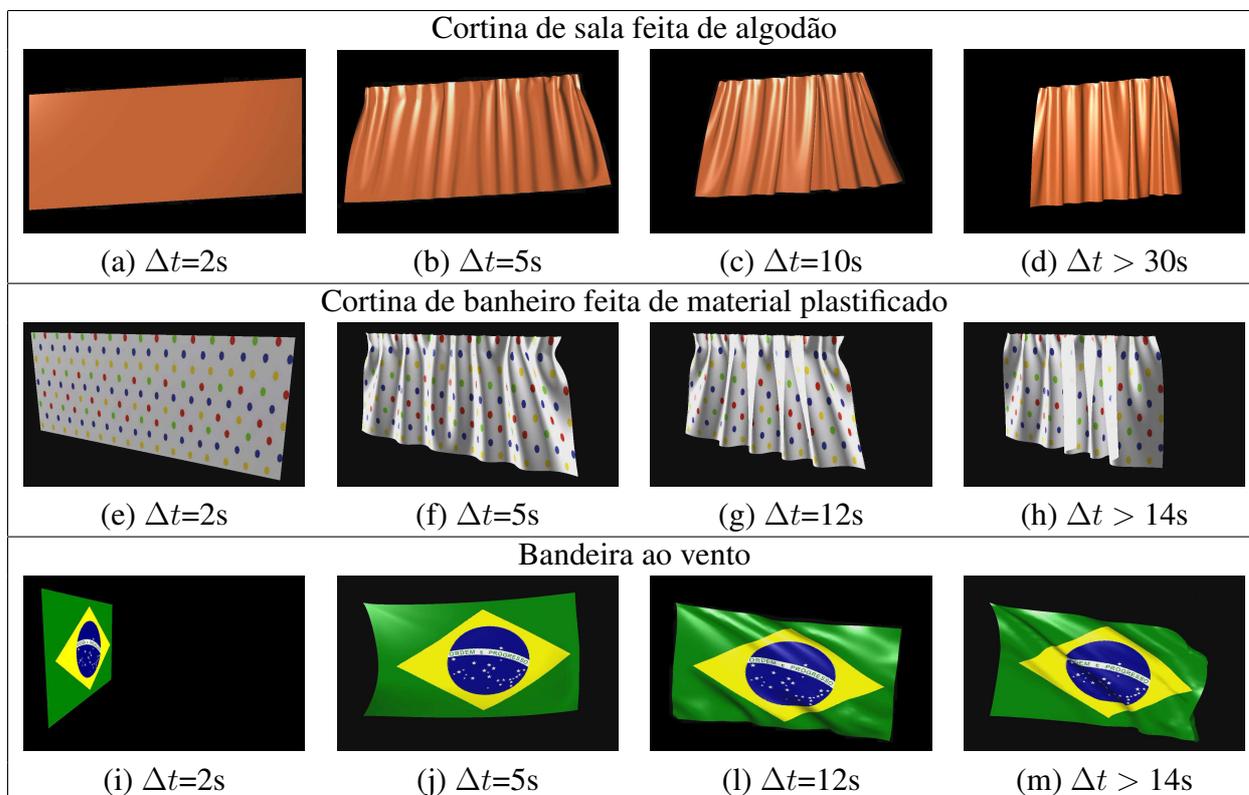


Figura 5.15: Dinâmica de *hangings*.

A cortina estampada corresponde a uma cortina plastificada de banheiro mostrada na Figura 5.11(c). Ela é uma cortina leve e que se dobra com facilidade, porém de forma mais armada. A Figura 5.15(e) mostra a superfície inicial desta cortina, que é de mesma resolução espacial e tem a mesma área da cortina flexível. Em seguida, pode-se ver a cortina rígida já bastante deformada em pouco tempo, como mostra a Figura 5.15(f). Nesta figura é possível reparar que o número de dobras formadas ao longo da direção horizontal do tecido é menor em relação à cortina flexível. Na Figura 5.15(g) esta diferença fica maior, com as dobras da cortina de banheiro tomando formas mais armadas. Por fim, a Figura 5.15(h) mostra o estado de equilíbrio, com as partes de baixo e cima da cortina alinhadas, assim como ocorre na flexível.

A dinâmica da bandeira da Figura 5.12 é simulada com aplicação da força fluido. Partindo da superfície inicial ilustrada na Figura 5.15(i), a qual está posicionada no plano yz , a bandeira sofre ação da gravidade no sentido para baixo e da força fluido no sentido positivo do eixo x , trazendo a superfície para o plano xy de forma já deformada, como mostra a Figura 5.15(j). A partir deste momento, a superfície começa a apresentar um comportamento oscilatório provocando dobras que

se formam no canto inferior esquerdo e se propagam até o canto superior direito. As Figuras 5.15(l) e 5.15(m) ilustram duas geometrias da superfície com este comportamento. Nesta simulação não existe um estado de equilíbrio em que não existam mais movimentos do tecido, a não ser que a força fluido seja interrompida.

5.3.2 Intuitividade

Como os tecidos são muito flexíveis para se curvar e pouco para esticar, eles possuem uma grande facilidade de formar dobras à medida que se deformam. As dobras formadas normalmente são de grande diversidade em termos de quantidade e suavidade, pois dependem muito do material de que é composto o tecido. Por esta diversidade, as dobras formadas nos tecidos são problemas que demandam mais atenção nas simulações de tecidos, pois conseguir associar cada tipo de deformação com um material é uma tarefa complexa. Com o intuito de demonstrar a facilidade na definição dos parâmetros do modelo proposto por de Melo, são analisadas nesta subseção as dobras formadas nos tecidos das imagens apresentadas, associando-as com os parâmetros do material da superfície.

Para melhor apresentar esta análise, os valores dos materiais do tecido utilizados nas simulações deste capítulo estão sintetizados na Tabela 5.1. Ela mostra o número de pontos (representado #pontos) que formam a superfície do tecido, sua massa, os coeficientes elásticos e o tempo de processamento. Este tempo de processamento corresponde ao tempo gasto por cada iteração da dinâmica de tecidos, com o intuito de mostrar o valor do tempo gasto para resoluções espaciais da malha diferentes daquelas usadas no Capítulo 3.

Tabela 5.1: Dados das simulações.

Imagem	#pontos	μ (g/m^2)	$\zeta_{\alpha\beta}$ (N)	$\xi_{\alpha\beta}$ (N)	$\phi_{\alpha\beta}$ (N)	Tempo (s)
Figure 5.5(a)	14400	78	100	0.01	0.1	0.2998
Figure 5.5(b)	14400	88	100	0.02	0.1	0.2941
Figure 5.5(c)	14400	111	100	0.03	0.1	0.2807
Figure 5.5(d)	14400	133	100	0.05	0.1	0.2816
Figure 5.6(a)	3600	80	100	0.015	0.1	0.0827
Figure 5.6(b)	3600	100	100	0.03	0.1	0.0755
Figure 5.8(b)	900	100	100	0.01	0.01	0.0174
Figure 5.8(d)	900	120	100	0.04	0.01	0.0163
Figure 5.9(a)	900	85	100	0.005	0.01	0.0187
Figure 5.9(b)	900	125	200	0.05	0.01	0.0151
Figure 5.10	2500	110	100	0.025	0.1	0.0514
Figure 5.11(a)	19200	70	100	0.005	0.1	0.2973
Figure 5.11(b)	19200	120	100	0.2	0.1	0.2874
Figure 5.11(c)	19200	90	100	0.05	0.1	0.2803
Figure 5.11(d)	19200	100	100	0.03	0.1	0.2876
Figure 5.12(b)	1600	90	100	0.01	0.1	0.0322

Analisando as imagens dos panos amassados mostradas na Figura 5.5, pode-se notar que os tecidos de seda e de algodão apresentam dobras com geometrias mais irregulares dos que as formadas nos tecidos de *jeans* e lona. A definição destas dobras irregulares proporciona um aspecto diferenciado para a qualidade da imagem, resultando em uma imagem mais realista na simulação de tecidos com essa flexibilidade. Pode-se destacar entre estes valores a influência do parâmetro $\xi_{\alpha\beta}$ na formação das dobras, determinando a suavidade com que elas se formam. Assim, valores altos de $\xi_{\alpha\beta}$ na definição do material tendem a formar dobras bem suaves no sentido de serem dobras armadas, como as dobras do *jeans* e principalmente as da lona, enquanto valores baixos de $\xi_{\alpha\beta}$ produzem dobras acentuadas, como as produzidas pela seda e algodão. Estes comportamentos distintos são manipulados de forma bem sensível através do valor de $\xi_{\alpha\beta}$, pois a seda, que é um extremo, tem o valor de $0.01N$, enquanto a lona, o outro extremo, $0.1N$. Já o coeficiente $\theta_{\alpha\beta}$, assim como $\zeta_{\alpha\beta}$, é mantido com os mesmos valores nas quatro simulações da Figura 5.5. Contudo, $\theta_{\alpha\beta}$ é de fundamental importância para quantidade de dobras, principalmente neste caso em que só há forças tangenciais de compressão. O valor deste coeficiente elástico determina a frequência com que as dobras são formadas com deslocamentos da métrica. O importante é que o valor do coeficiente $\theta_{\alpha\beta}$ não seja nulo, pois neste caso nenhum dos quatro panos sofreria deformações nestas simulações.

De forma similar às imagens dos panos amassados, as gravatas em formato borboleta da Figura 5.6 foram geradas apenas com valores distintos da massa do material e do coeficiente elástico $\theta_{\alpha\beta}$. Sem a presença da gravidade e utilizando forças pontuais nos mesmos pontos da malha discreta, nota-se que a imagem da Figura 5.6(a) se curva em relação ao seu centro com dobras mais arredondadas nas bordas do que a Figura 5.6(b), que possui uma resistência maior à curvatura.

As dobras criadas no caimento de tecidos sobre mesas, como as apresentadas na Figura 5.8, são de resolução pequena em relação aos panos amassados, mas mesmo assim é possível encontrar uma configuração parecida com as situações das fotos reais. Os valores dos parâmetros de material do linho quanto os de cetim são valores intermediários entre a seda e o *jeans*. O que acontece muitas vezes com os tecidos é que os fios que os formam são compostos de mais de um tipo de fibra. Atualmente, é fácil encontrar uma camiseta feita 50% de algodão e 50% de poliéster, que é feita deste modo para diminuir custos e também permitir que ela amasse menos do que uma feita 100% de algodão. A Figura 5.9(a) ilustra outra possível variação de material alterando apenas o coeficiente $\xi_{\alpha\beta}$, enquanto na Figura 5.9(b) o material do tecido é definido de forma bem mais rígida e suas dobras são bem aramadas. O coeficiente $\theta_{\alpha\beta}$ é mantido com o valor 0.01 nestas simulações de caimento pois não tem tanta influência visual como nas simulações de *buckling*.

Com relação à poltrona da Figura 5.10, foi considerado como material um tecido de veludo, caindo sob a força de gravidade. O veludo é configurado com valores próximos aos do algodão, mas com uma massa um pouco maior. Nesta simulação pode-se ver que as dobras formadas, principalmente na

região próxima do assento da poltrona, são dobras mais armadas.

As cortinas da Figura 5.11 foram geradas com malhas bastante refinadas. A superfície delas possui 19200 pontos, utilizando uma grade de 240×80 para representar o formato de uma cortina de maneira mais apropriada. Novamente, mudando somente a resistência da curvatura do tecido e as massas dos materiais pode-se obter comportamentos bem distintos, desde a cortina da Figura 5.11(a) que é bem flexível até a cortina de banheiro da Figura 5.11(c), sendo mais rígida e apresentando dobras suaves.

A bandeira feita na Figura 5.12 também é uma malha pequena, de material leve e maleável, que com o uso da força fluido consegue atingir dobras bem irregulares ao longo de todo o tecido, as quais se formam e se desfazem periodicamente.

5.4 Considerações Finais

Este capítulo apresentou os principais resultados obtidos com o simulador de tecidos desenvolvido com o modelo proposto por de Melo utilizando o método de Verlet para integrá-lo em relação ao tempo. A Seção 5.1 apresentou as metodologias usadas no desenvolvimento das simulações, mostrando de que forma são especificadas as forças externas que são definidas pelo usuário, bem como são calibrados os parâmetros do material e as influências da resolução da malha na qualidade visual. Nesta seção foi fornecido um procedimento para o auxílio no ajuste de valores do modelo para obter um tipo de material específico. A Seção 5.2 mostrou os resultados visuais deste trabalho, dividindo-os em categorias relativas ao tipo de deformação que os tecidos sofrem, como *bucklings*, *drappings* e *hangings*, que podem ser traduzidos como enrugamentos, caimentos e suspensões de tecidos. Por último, a Seção 5.3 validou os resultados obtidos, procurando destacar algumas características observadas que demonstram, na prática, a qualidade do modelo proposto por de Melo [5].

Com o término deste capítulo pode-se concluir que a abordagem de simular tecidos proposta, a qual reduz o modelo baseado na mecânica de contínuos em sistemas de partículas e solucioná-lo com o método de Verlet, é adequada para produzir comportamentos que apresentem realismo com uma complexidade computacional linear. Este paradigma de modelagem e animação de tecidos permitiu validar praticamente as principais características do modelo de tecidos proposto por Wu e Melo [3], como produzir dobras sob forças de compressão naturalmente. A precisão física do modelo também foi validada com a dinâmica de tecidos, mostrando um comportamento adequado do tecido quando ele atinge um estado de equilíbrio. Além disso, o controle intuitivo dos parâmetros do modelo permitiram formar uma grande variedade de dobras, que foram geradas em malhas de grande resolução espacial para avaliar o realismo atingido. A análise das dobras formadas mostrou que quanto mais densas forem as malhas, melhor é a qualidade dos resultados.

Capítulo 6

Conclusões e Trabalhos Futuros

Este capítulo finaliza o trabalho apresentando as principais conclusões sobre ele e indicando os possíveis trabalhos futuros. A principal conclusão foi que é possível aplicar em um modelo de tecidos baseado na superfície de Cosserat, na qual aparecem grandezas diferenciáveis, um método de integração explícita. Usando esta abordagem, além de se obter animações realistas e condizentes com a física, pode-se agregar também desempenho computacional. Com os resultados visuais do simulador implementado pôde-se notar que a qualidade das dobras formadas contribui de maneira significativa para o aumento da realidade no movimento de tecidos. Foi possível ver também a resposta apropriada da superfície do tecido quando ela é submetida à forças de compressão. Com o gráfico do tempo de processamento relacionado com o número de pontos da superfície, viu-se na Figura 3.17 que o comportamento do sistema é linear, implicando em maior eficiência computacional. Desta forma, pode-se dizer que o objetivo deste trabalho foi atingindo, pois a abordagem alternativa proposta para simular a dinâmica de tecidos, após ser estudada, implementada e analisada, mostrou-se promissora. A seguir são retomados os principais pontos apresentados em cada capítulo para tecer pequenas conclusões e finalmente apresentar as opções de desdobramento deste trabalho.

Conforme foi afirmado no Capítulo 1, a simulação de tecidos pode ser feita com um modelo de tecido integrado ao longo do tempo. A modelagem do tecido corresponde a uma formulação física, que pode ser baseada na mecânica de partículas ou na mecânica de contínuos. Algebricamente, ambas as técnicas podem ser descritas com o uso de equações diferenciais ordinárias em função do tempo. Já os métodos numéricos de integração podem ser explícitos, implícitos ou semi-implícitos. Atualmente, os modelos baseados em partículas junto com métodos semi-implícitos são os mais utilizados para simular tecidos, porém, este trabalho usou uma abordagem alternativa a esta combinação convencional objetivando um melhor desempenho. Utilizando um modelo de tecido preciso, fisicamente embasado e teoricamente comprovado para produção de dobras sob forças de compressão [5], este trabalho foi dedicado a investigar qual a melhor técnica de integração explícita que pode ser adaptada

a ele para produzir o movimento de tecidos. A idéia é manter o realismo e a precisão física do modelo e aumentar seu desempenho computacional.

Como pôde-se ver no Capítulo 2, essa abordagem alternativa proposta preenche uma lacuna existente na literatura em relação aos trabalhos voltados à simulação de tecidos. Devido ao sucesso da abordagem que une a mecânica de partículas com métodos semi-implícitos de integração, o paradigma proposto ainda não tinha sido testado de forma profunda. Com este trabalho foi possível verificar que esta é uma interessante abordagem para quem precisa simular a dinâmica peculiar de tecidos com um custo computacional linear. Neste capítulo também foi apresentado o modelo de tecido proposto por de Melo [5], que baseado na teoria geral da superfície de Cosserat [4] faz simplificações específicas para tecidos que conseguem atingir uma resposta natural da superfície quando submetida às forças de compressão. A escolha deste modelo deve-se principalmente ao fato de considerar o acoplamento entre as deformações referentes à métrica e à curvatura da superfície que, segundo seus autores, é fundamental para a formação de dobras sob compressão.

No Capítulo 3 foram apresentados cinco métodos numéricos de integração explícita que podem ser aplicados no modelo de tecido escolhido. A formulação de cada método permitiu analisar de forma teórica o número de cálculos e passos demandados, bem como a ordem de precisão numérica relacionada ao termo do erro de truncamento. Contudo, todos estes métodos precisavam ser testados com o modelo de tecido para verificar na prática qual era a melhor opção para se utilizar. Os trabalhos relacionados com a simulação de tecidos e que usam os métodos numéricos apresentados também foram discutidos neste capítulo, destacando a importância da proposta deste trabalho.

Ainda no Capítulo 3, foram mostrados os resultados das simulações de teste aplicadas a todos os métodos, sendo possível concluir que os mais eficientes computacionalmente são os métodos de Euler, *Leapfrog* e Verlet, enquanto os métodos de Runge-Kutta de segunda e quarta ordem demandam um tempo de processamento superior. Quanto à estabilidade numérica, no sentido de qual método suporta um passo de tempo maior de forma estável, todos os métodos se mantiveram no mesmo patamar para cada simulação de teste. Investigando o motivo deste resultado, concluiu-se que ao linearizar o problema, o uso de vetores normais, estimados no instante $t - \Delta t$, para determinar as forças elásticas internas no instante t , conforme mostra o algoritmo 2 da seção 4.1, faz com que a suavidade na variação destes vetores tenha maior influência no passo de tempo do que o método numérico. Já para o realismo obtido com cada método concluiu-se que como a qualidade das simulações eram muito próximas, a análise de qual é a melhor passa a ser subjetiva e, portanto, deu-se prioridade para o desempenho na escolha do método que gerou os resultados deste trabalho. Neste sentido, foi escolhido o método de Verlet, por agregar de forma balanceada eficiência computacional com precisão numérica, além de na teoria ser também o melhor método em termos do compromisso entre o custo e o benefício.

No Capítulo 4 foi apresentada uma implementação do modelo de tecidos com uso de técnicas de integração explícita, construindo uma interface lógica de programação e também uma interface gráfica que permitisse ao usuário manipular poucos parâmetros para obter as simulações. Adotando o paradigma orientado a objetos, este capítulo descreveu a modelagem de dados do sistema que o separou em dois módulos principais, sendo um responsável pela dinâmica da superfície e outro pela visualização das superfícies geradas. Quanto à interface gráfica, ela foi desenvolvida tanto para a plataforma Linux quanto para a Windows, em ambas usando a biblioteca gráfica OpenGL e a linguagem de programação C++. O processo de gerar algumas simulações foi detalhado para mostrar a usabilidade do simulador para obter resultados expressivos.

A facilidade de uso do simulador desenvolvido permitiu ainda validar na prática as propriedades do modelo de tecidos selecionado, que até então foram somente demonstradas teoricamente. Dentre as quais, destaca-se a sua facilidade em formação de dobras sob forças de compressão. No Capítulo 5 foram detalhados os testes de validação e os resultados obtidos. Antes dos resultados, foi dada uma explicação sobre como especificar as forças externas que agem sobre sua superfície e calibrar os coeficientes elásticos do material do tecido, fornecendo um procedimento de cinco passos para este propósito. As influências da resolução das malhas nos resultados visuais também foi comentado neste ponto. Em seguida, os resultados foram apresentados em três categorias, que correspondem aos tipos de deformação causada nos tecidos. *Buckling* é o termo associado ao ato de amassar, enrugar, ou simplesmente criar dobras em tecidos, já *draping* é o caimento de tecidos sobre objetos e, *hanging*, a suspensão de tecidos. Com os resultados deste capítulo foi possível concluir que a qualidade e diversidade das dobras formadas está diretamente associada ao número de pontos da superfície, como também que a dinâmica mostrada pela seqüência de imagens atinge estados de equilíbrio de forma apropriada. Além disso, observamos que o comportamento linear do algoritmo com a entrada do número de pontos da malha do tecido é preservado. Portanto, é possível simular malhas pequenas em tempo real, propiciando o seu emprego em jogos.

Em relação aos desdobramentos deste trabalho, pode-se indicar alguns trabalhos futuros. O primeiro deles é a inclusão de algoritmos que façam o tratamento de colisões e auto-colisões que acontecem na dinâmica dos tecidos. Este é um incremento fundamental para se atingir maior realismo como também validar a aplicação destes algoritmos com a proposta deste trabalho. Para tanto, a interface lógica criada neste trabalho visa facilitar a integração do processo referente às colisões, separando de forma apropriada a dinâmica interna de tecidos como sugere Bridson et al. [25]. Além disso, os diferentes métodos numéricos implementados podem ser de grande utilidade para investigar qual a melhor técnica de colisão que pode ser adaptada a este modelo. É interessante lembrar que alguns métodos numéricos apresentados, como o método de Verlet, possui variações bem específicas para contemplar os algoritmos de colisões que contam com passos de tempo adaptativos [25, 11].

Além da incorporação das colisões, uma das prioridades na continuidade deste trabalho é permitir o uso de qualquer tipo de malha triangular. Com isto, uma grande diversidade de objetos poderiam ser importados no simulador implementado e deformados como tecidos, retirando a limitação deste trabalho em só deformar malhas retangulares. Para desenvolver esta generalização na geometria da superfície do tecido é necessário modificar a maneira de calcular as forças internas e a discretização espacial feita com as diferenças finitas. Neste caso, pode-se aplicar o algoritmo proposto por Rusinkiewicz [39], para estimar os tensores de métrica e de curvatura em cada ponto da malha. Apesar de demandar maior esforço computacional, esta modificação pode ampliar o domínio das geometrias processáveis. A discretização no espaço do modelo de tecido proposto por Wu e de Melo [5] pode ser feita também com a técnica dos elementos finitos para comparar com a qualidade obtida em relação a das diferenças finitas, usada neste trabalho. Esta técnica foi usada recentemente por Bergou et al. para discretizar um modelo de superfícies inextensíveis deformáveis, apresentando uma implementação eficiente computacionalmente [8].

Embora o modelo de tecidos usado neste trabalho possua parâmetros de material com significados geométricos intuitivos que auxiliam a definição do seu tipo, os valores dos coeficientes elásticos dependem de medidas como a massa e área do tecido. Estas relações de dependência podem ser investigadas para conseguir uma simulação mais precisa dos parâmetros mecânicos do tecido. Isto permitiria o mapeamento automático de materiais, de forma que o usuário pudesse produzir simulações apenas selecionando um tipo de tecido entre uma lista e o simulador determinasse os coeficientes corretos para produzir o comportamento do material.

Outra possibilidade de trabalho futuro seria integrar algoritmos de corte e costura sobre os tecidos, permitindo que eles possam formar estruturas mais complexas e que possam até ser usados por projetistas relacionados à moda. Um último trabalho futuro desafiador seria estender este simulador para realizar animações em tempo real, pois com isso pode-se pensar em aplicar as simulações até para jogos. Com o recurso feito neste trabalho de visualizar os quadros à medida que eles forem calculados, chamado de modo *on-the-fly*, viu-se que para malhas pequenas, compostas de 400 pontos aproximadamente, isso já é possível; contudo, para malhas maiores, as quais apresentam mais qualidade visual, o simulador teria que ser bem modificado para otimizar as animações. Um passo considerável neste sentido seria a implementação da alternativa proposta diretamente na GPU ¹.

¹Graphics Processing Unit

Referências Bibliográficas

- [1] Haumann, D. R. Modeling the physical behavior of flexible objects. In *ACM SIGGRAPH 87 Course Notes 17 - Topics in Physically-Based Modeling*, Anaheim, California, Julho 1987.
- [2] Provot, X. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proceedings of Graphics Interface*, pages 147–154, Maio 1995.
- [3] Wu, S.-T. and Melo, V. F. A deformable surface model on the basis of the theory of a cosserat surface. In *SIBGRAPI 2004 conference proceedings*, 2004.
- [4] Green, A. E. and Naghdi, P. and Wainwright, W. A general theory of a cosserat surface. *Archive for Rational Mechanics and Analysis*, 20:287–308, 1965.
- [5] Melo, V. F. Modelagem e controle de caimento e dobras em superfícies deformáveis. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 2004.
- [6] Goldenthal, R. and Harmon, D. and Fattal, R. and Bercovier, M. and Grinspun, E. Efficient Simulation of Inextensible Cloth. In *SIGGRAPH (ACM Transactions on Graphics)*, 2007.
- [7] Thomaszewski, B. and Wacker, M. and Straßer, W. A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer Animation*, pages 107–116, 2006.
- [8] Bergou, M. and Wardetzky, M. and Harmon, D. and Zorin, Z. and Grinspun, E. A quadratic bending model for inextensible surfaces. In *Eurographics Symposium on Geometry Processing*, 2006.
- [9] Etmuss, O. and Keckeisen, M. and Straßer, W. A fast finite element solution for cloth modeling. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*, pages 244–252, 2003.
- [10] Baraff, D. and Witkin, A. and Kass, M. Untangling cloth. In *ACM Transactions on Graphics*, volume 22(3), pages 862–870, Julho 2003.

- [11] Bridson, R. and Marino, S. and Fedkiw, R. Simulation of clothing with folds and wrinkles. In *Eurographics / SIGGRAPH Symposium on Computer Animation*, 2003.
- [12] Choi, K.-J. and Ko, H.-S. Stable but responsive cloth. In *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, volume 21, pages 604–611, 2002.
- [13] Volino, P. and Thalmann, N. M. Fast geometric wrinkles on animated surfaces. In *WSCG 99*, pages 175–183, Plzen, Fevereiro 1999.
- [14] Baraff, D. and Witkin, A. Large steps in cloth simulation. In *Proceedings of SIGGRAPH 98*, pages 43–54, 1998.
- [15] Eberdardt, B. and Weber, A. and Strasser, W. A Fast, flexible, particle-system model for cloth draping. In *Computer Graphics in Textiles and Apparel (IEEE Computer Graphics and Applications)*, pages 52–59, Setembro 1996.
- [16] Eischen, J.W and Deng, S. and Clapp, T.G. Finite-element modeling and control of flexible fabric parts. In *IEEE Computer Graphics and Applications*, volume 16(5), pages 71–80, 1996.
- [17] Breen, David E. and House, Donald H. and Wozny, Michael J. A physically based particle model of woven cloth. In *The Visual Computer*, volume 8(5-6), pages 264–277, 1992.
- [18] Simo, J.C. and Fox, D.D. On a stress resultant geometrically exact shell model, part I: formulation and optimal parameterization. In *Computer Methods in Applied Mechanics and Engineering*, volume 72, pages 267–304, 1989.
- [19] Terzopoulos, D. and Platt, J. and Barr, A. and Fleischer, K. Elastically Deformable Models. In *Computer Graphics (SIGGRAPH proceedings)*, volume 21(4), pages 205–214, July 1987.
- [20] Feynman, C. Modeling the appearance of cloth. M.s. thesis, Massachusetts Institute of Technology, Massachusetts, 1986.
- [21] Weil, J. The synthesis of cloth objects. In *Computer Graphics (SIGGRAPH proceedings)*, pages 49–54, 1986.
- [22] Decaudin, P. and Julius, D. and Wither, J. and Boissieux, L. and Sheffer, A. and Cani, M.P. Virtual garments: a fully geometric approach for clothing design. In *EUROGRAPHICS 2006*, volume 25(3), 2006.
- [23] Breen, D. and House, D. *Cloth modeling and animation*. AK Peters, 2000.

- [24] Etmuss, O. and Gross, J. and Straßer, W. Deriving a particle system from continuum mechanics for the animation of deformable objects. In *IEEE Transactions on Visualization and Computer Graphics*, volume 9(4), pages 538–550, 2003.
- [25] Bridson, R. and Fedkiw, R. and Anderson, J. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 594–603, 2002.
- [26] Breen, D. E. and House, D. H. and Wozny, M. J. Predicting the drape of woven cloth using interacting particles. In *Computer Graphics (SIGGRAPH proceedings)*, pages 365–372. Addison-Wesley, July 1994.
- [27] David E. Stewart and Zbigniew Leyk. Meschach Library. Página na internet, School of Mathematical Sciences Australian National University, Último acesso em 11/06/2008 1994. <http://www.netlib.org/c/meschach/>.
- [28] Smith, G. D. *Numerical solution of partial differential equations: finite difference methods*. Oxford University Press, 3rd edition, 1985.
- [29] Press, W. H. and Teukolsy, S. A. and Vetterling, W. T. and Flannery, B. P. *Numerical recipes in c*. Cambridge University Press, 2nd edition, 1992.
- [30] Volino, P. and Courchesne, M. and Magnenat-Thalmann, N. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Computer Graphics (SIGGRAPH proceedings)*, pages 137–144. Addison-Wesley, 1995.
- [31] Volino, P. and Magnenat-Thalmann, N. Developing simulation techniques for an interactive clothing system. In *Virtual Systems and Multimedia (VSMM proceedings)*, pages 109–118, Geneva, Switzerland, 1997.
- [32] Oshita, M. and Makinouchi, A. Real-time cloth simulation with sparse particles and curved faces. In *Proceedings of Computer Animation*, Seoul, Korea, November 2001.
- [33] Volino, P. and Magnenat-Thalmann, N. Comparing efficiency of integration methods for cloth simulation. In *Computer Graphics International Proceedings*, pages 265–274, Plzen, Feb 2001.
- [34] Thomas Jakobsen. Advanced Character Physics. Página na internet, IO Interactive, Farvergade 2, Último acesso em 11/06/2008 2001. <http://www.teknikus.dk/tj/gdc2001.htm>.

- [35] Jonathan Dummer. A simple time-corrected verlet integration method. Página na internet, GameDev.net, Último acesso em 11/06/2008 2005. <http://www.gamedev.net/reference/articles/article2200.asp>.
- [36] Michael Hauth. Numerical Techniques for Cloth Simulation. In *SIGGRAPH 2003 Course 29: Clothing Simulation and Animation*, October 2003.
- [37] Suriya Natsupakpong. Numerical integration method in deformable object simulation. Página na internet, Electrical Engineering and Computer Science, Case Western Reserve University, Último acesso em 11/06/2008 2004. <http://filer.case.edu/~sxn66/spring2004/eecs466/Report.htm>.
- [38] Monteiro, L.P. Deformable Surface Modeler. Página na internet, Electrical Engineering and Computer Science Faculty, University of Campinas, Último acesso em 11/06/2008 2004. <http://www.dca.fee.unicamp.br/projects/desmo/pinho>.
- [39] Rusinkiewicz, S. Estimating curvatures and their derivatives on triangle meshes. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, 2006.