

Diego Dias Bispo Carvalho

***Modelo Computacional para Representação Gráfica
do Comportamento Elástico de Superfícies
Deformáveis e Interação com Sólidos para
Aplicações em Tempo Real.***

Florianópolis - SC - Brasil

26 de março de 2007

Diego Dias Bispo Carvalho

***Modelo Computacional para Representação Gráfica
do Comportamento Elástico de Superfícies
Deformáveis e Interação com Sólidos para
Aplicações em Tempo Real.***

Dissertação apresentada para obtenção do Grau
de Mestre em Ciência da Computação pela Uni-
versidade Federal de Santa Catarina.

Orientador:

Aldo von Wangenheim, Prof. Dr. rer. nat.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Florianópolis - SC - Brasil

26 de março de 2007

Dissertação de Mestrado em Ciências da Computação sob o título “*Modelo Computacional para Representação Gráfica do Comportamento Elástico de Superfícies Deformáveis e Interação com Sólidos para Aplicações em Tempo Real.*”, defendida por Diego Dias Bispo Carvalho e aprovada em 26 de março de 2007, em Florianópolis, Estado de Santa Catarina:

Prof. Dr. Rogério Cid Bastos
Coordenador do PPGCC-INE-UFSC

Banca examinadora constituída pelos professores:

Prof. Dr. rer. nat. Aldo von Wangenheim
Orientador

Profa. Dra. Agma Juci Machado Traina
Universidade Estadual de São Paulo

Prof. Dr. Daniel Martins
Universidade Federal de Santa Catarina

Prof. Dr. rer. nat. Eros Comunelo
Universidade Federal de Santa Catarina

Agradecimentos

Agradeço aos meus pais, pela minha criação e incentivo

Aos meus colegas do Projeto Cyclops que me auxiliaram nestes últimos dois anos. Especialmente a todo pessoal que fez parte do grupo de Computação Gráfica e às meninas do setor administrativo, por ouvirem minhas reclamações e pelo cafézinho de todos os dias.

Aos meus amigos de universidade que me acompanharam em incontáveis festas, churrascos e roubadas em todos esses anos de UFSC.

Aos meus companheiros de banda.

Ao Prof. Aldo von Wangenheim pela orientação e por fornecer estrutura para que este trabalho se concretizasse no escopo do projeto Cyclops.

e à FEESC e à CAPES pelo auxílio financeiro.

Resumo

Este trabalho apresenta um modelo computacional utilizado para a representação gráfica de superfícies deformáveis. A interação destas superfícies com sólidos rígidos pode resultar em alterações morfológicas proporcionais à região de contato, à força aplicada e ao tempo em que o mesmo ocorreu. A validação se fez através de experimentos com malhas tridimensionais que representam a deformação de um tecido que sofre a ação de uma força externa. Durante a simulação o modelo busca atender características físicas e temporais. As malhas deformáveis e os sólidos são armazenadas em estruturas denominadas DLFLs (Double Linked Face List). Essas estruturas são grafos de rotação que possuem características topológicas que garantem uma grande flexibilidade de manipulação, evitam a redundância de informações e permitem percorrer todos os vértices da estrutura em um único ciclo. Essa última propriedade torna possível calcular a intersecção da DLFL com elementos geométricos e verificar quais elementos da mesma serão afetados por esta operação; respeitando sua ligação topológica de vizinhança. A representação física das forças que agem sobre os tecidos e suas implicações são feitas através de um sistema massa-mola. Este sistema consiste em pontos de massa conectados por molas que podem se deformar ao sofrer a ação de uma força externa e tender a oscilar no momento que a ação é interrompida voltando a seu estado original. O sistema foi adotado em conjunto com a DLFL, de modo que cada vértice representa uma massa e cada aresta representa uma mola, criando um sistema por onde as forças podem ser propagadas aos vértices vizinhos àquele que sofre uma ação.

palavras chave: Computação Gráfica, Comportamento Elástico de Tecidos, Tratamento de Colisões, DLFL.

Abstract

This work presents a computational model for deformable surfaces graphical representation. The interactions between these surfaces and rigid solid objects may lead to morphological changes proportional to the region that the contact has affected. This model validation was made through experiments with tridimensional meshes representing the deformation that an external force may cause. The model aims to achieve expected physical and temporal characteristics during the simulation in comparison with real world constraints. The deformable mesh and the solids are stored in structures known as DLFL (Double Linked Face List). This structure is a rotation graph with characteristics that guarantee manipulation flexibility, avoiding redundant information and allowing to visit all the structures' vertices in a single cycle. This last property allows to calculate the DLFL intersection with geometrical objects and to verify which elements will be affected in this operation, respecting their connectivity order. The forces that interact with the meshes are calculated by a mass-spring system. This system consists in mass points interconnected with springs that may deform on suffering external forces actions, when these forces influence ceases, the springs tend to return to their original length. This system was adapted to make use of the DLFL characteristics; every vertex represents a mass point and every edge represents a spring connecting two mass points. Leading to a system where the forces may be spread through direct neighbors mass points.

Key-words: Computer Graphics , Mesh Elastic Behavior, Collision Treatment DLFL.

Lista de Algoritmos

1	Característica de Euler. vértices(v), arestas(e) e faces(f).	p. 37
2	Cálculo da força de restauração do mola.	p. 46
3	Propagação da força em travessia por profundidade.	p. 47
4	Propagação da força em travessia por profundidade para a aplicação de forças externas em múltiplas partículas.	p. 51
5	Propagação de forças em profundidade atualizado	p. 52
6	Versão definitiva, cálculo da força de reação da mola	p. 60
7	Versão definitiva, Propagação de Forças	p. 61
8	Versão definitiva, deslocamento por Velocity Verlet	p. 62
9	Algoritmo de atualização de sistema de partículas	p. 63
10	Algoritmo para inserção de uma partícula em um ponto de contato de uma malha.	p. 77

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 15
1.1	Objetivo	p. 18
1.2	Objetivos Específicos	p. 19
1.2.1	Escolha de um modelo para o cálculo da deformação.	p. 19
1.2.1.1	Estabilidade e Fidelidade da Representação	p. 19
1.2.1.2	Aplicação de Forças Externas	p. 19
1.2.1.3	Aplicação de Forças Internas e Equilíbrio Estático	p. 19
1.2.1.4	Escolha do método de integração numérico	p. 20
1.2.2	Representar o contato de um elemento deformável com um sólido rígido	p. 20
1.2.2.1	Detectar os pontos de colisão	p. 20
1.2.2.2	Aplicação de forças	p. 20
2	Modelos matemáticos para representação de deformações em tecidos	p. 21
2.1	Modelos Geométricos	p. 21
2.2	Modelos Físicos	p. 24

2.2.1	Elementos Finitos	p. 25
2.2.2	Massa-Mola	p. 28
2.2.3	Chain Mail	p. 31
2.3	Definição do modelo	p. 33
3	Estrutura para armazenar informações geométricas	p. 35
3.1	Estruturas para a Representação de Objetos Tridimensionais	p. 35
3.2	DLFL	p. 39
3.3	Modificações na DLFL	p. 41
4	Modelo para Deformação de Tecidos	p. 43
4.1	Propagação de Forças Através da Estrutura	p. 44
4.2	Métodos de Integração	p. 53
4.3	Manutenção de Equilíbrio Estático	p. 57
5	Interação de Objetos Sólidos Rígidos com Objetos Deformáveis	p. 65
5.1	Detecção de Colisão	p. 66
5.2	Tratamento de colisão de tecidos com sólidos rígidos imóveis	p. 68
5.3	Tratamento de colisão de tecidos com sólidos rígidos	p. 69
5.4	Tratamento de Colisões: Subdivisão da malha	p. 73
6	Resultados	p. 78
6.1	Experimentos: seção 4.2	p. 78
6.1.1	Experimento 1	p. 78
6.1.2	Experimento 2	p. 80

6.1.3	Experimento 3	p. 81
6.2	Experimentos: seção 4.3	p. 82
6.2.1	Experimento 1	p. 83
6.2.2	Experimento 2	p. 85
6.2.3	Experimento 3	p. 87
6.2.4	Experimento 4	p. 88
6.3	Experimentos: seção 5.2	p. 89
6.3.1	Experimento 1	p. 90
6.3.2	Experimento 2	p. 91
6.4	Experimentos: seção 5.3	p. 91
6.4.1	Experimento 1	p. 92
6.4.2	Experimento 2	p. 93
6.4.3	Experimento 3	p. 94
6.4.4	Experimento 4	p. 96
7	Conclusões e Trabalhos Futuros	p. 98
8	Anexos	p. 105
	Referências	p. 110

Lista de Figuras

- 1 Superfície Bézier, quadriláteros marcam os dezesseis pontos de controle. . . . p. 22
- 2 Superfície Bézier, alterando apenas os quatro pontos de controle representados por quadriláteros maiores. p. 22
- 3 Deformação de tecidos utilizando o modelo de elementos finitos. p. 26
- 4 Grid de zhuang à esquerda de imagens capturadas de uma animação onde ocorre a deformação de um paralelepípedo. p. 26
- 5 Deformação gerada pelos movimentos da mãos baseada em coeficientes locais calculados como nas regiões em amarelo. p. 27
- 6 Simulação de neurocirurgia endoscópica, construída utilizando o framework proposto por Çavosuglu *et al.* p. 27
- 7 Malha no estado inicial e à direita após 200 iterações da simulação. p. 29
- 8 Trabalho de Debunne, demonstrando deformações em consequência de colisões com ferramentas. p. 30
- 9 Cilindro para representação de vasos, molas em branco em destaque. Ao lado direito imagens da simulação em tempo real. p. 30
- 10 Cabeça reconstruída a partir de série imagens de CT a esquerda, a direita a mesma dividida ao meio. p. 32

11	Representação de uma artéria proveniente de uma reconstrução tridimensional a partir de uma série de imagens de tomografia. A esquerda representada por uma Octree, a direita por uma representação por bordas (DLFL) (SANTOS, 2006).	p. 37
12	Conexões da aresta na estrutura <i>winged edge</i> (<i>adaptado de www.baumgart.org</i>)	p. 38
13	Organização dos elementos da DLFL na representação de um tetraedro, adaptado de (AKLEMAN; CHEN; SRINIVASAN, 2003 apud SANTOS, 2006).	p. 40
14	Reconstrução a partir uma série de CT previamente segmentada (CARVALHO; SANTOS; WANGENHEIM, 2006).	p. 42
15	Comportamento de um sistema massa-mola composto uma mola ligada a um ponto fixo e a uma partícula móvel.	p. 44
16	Propagação de forças em uma malha (setas tracejadas), resultantes da aplicação de uma força externa FE.	p. 45
17	Ordem de propagação das forças.	p. 48
18	Trabalho de Dochev e Vassilev (2003), ordem dos vértices organizada de acordo com a posição de pontos estáticos (sp)	p. 49
19	Sistema massa-mola sofrendo a ação de duas forças simultâneas	p. 49
21	Cálculo da Força de reação da mola em R^3	p. 50
20	Cálculo das forças em um sistema com duas partículas	p. 50
22	Vetores circulos por elipses demonstram forças aplicadas no vértice de onde partiu a força externa, porém são provenientes da propagação das forças de reação da mola.	p. 58
23	Colisão entre dois Polígonos planares e colisão entre um polígono planar e outro não planar. Triângulo em linhas pontilhadas representa a posição anterior à colisão.	p. 66

24	Detecção de colisão durante a trajetória da partícula.	p. 67
25	vetor \vec{v}_2 resultante da reflexão do vetor \vec{v}_1 no plano com vetor normal \vec{n} . . .	p. 68
26	Posição da partícula é corrigida (esquerda). Após correção da velocidade, a posição da partícula é novamente atualizada (direita).	p. 69
27	Colisão entre uma bola e uma malha à esquerda, colisão entre malha e lança à direita. Pontos de contato em destaque.	p. 70
28	Malha interceptada entre uma iteração e outra (esq). Posição da malha corri- gida (dir).	p. 71
29	Subdivisão da malha durante uma colisão. Dividiu a face em três com base no ponto de contato.	p. 75
30	Consecutivas subdivisões de faces.	p. 76
31	Gráfico Experimento 1	p. 79
32	Gráfico Experimento 1	p. 80
33	Gráfico Experimento 2	p. 81
34	Gráfico Experimento 3	p. 82
35	Atualizações da malha ao longo do tempo. Iteração 0 e iteração 6000.	p. 83
36	Atualizações da malha ao longo do tempo. Iteração 4600 e iteração 5000. . .	p. 84
37	Atualizações da malha ao longo do tempo. Iteração 5200 e iteração 7000. . .	p. 84
38	Atualizações da malha ao longo do tempo. Iteração 9000 e iteração 20000. . .	p. 84
39	Atualizações da malha ao longo do tempo. Iteração 5000 e iteração 10000. . .	p. 85
40	Experimento 1 Iteração 20000 e Experimento 2 Iteração 20000.	p. 86
41	Experimento 3, Iterações 5000 e 10000.	p. 87
42	Experimento 3, Iteração 20000.	p. 87

43	Experimento 4, Iterações 5000 e 7000.	p. 88
44	Experimento 4, Iterações 10000 e 20000.	p. 89
45	Imagens do três testes para verificação de colisão. Vetores (0 -10 0)N, (0 -100 0)N e (0 -200 0)N da esquerda para a direita respectivamente.	p. 90
46	Deformações sofridas pela malha, decorrentes da ação da gravidade e colisão com a superfície.	p. 91
47	Malha e sólido antes da colisão. Reações da malha à colisão.	p. 93
48	Reações da malha à colisão. Sólido na ascendente (esq) e descendente (dir).	p. 93
49	Malha apresentou uma deformação maior. Ocasionalmente o rompimento de uma mola.	p. 94
50	Deformações da malha.	p. 95
51	Malha retornando ao estado de equilíbrio estático.	p. 95
52	Partículas ligadas a um número diferente de molas(esq). Malha utilizada no experimento 4 (dir).	p. 96
53	Malha deformada em função do contato com o sólido.	p. 97
54	Deformação menor que ε	p. 99
55	Artéria e adaga. Ambos compostos por polígonos planares.	p. 102
56	Colisão lâmina-artéria. Artéria se rompe.	p. 103
57	Artéria aproximada em modelo de arame, após rompimento. Elipse indicando região afetada.	p. 104
58	Algoritmo 8 - Diagrama de Blocos	p. 105
59	Algoritmo 9 - Diagrama de Blocos	p. 105

Lista de Tabelas

1	Classificação de modelos para deformação de tecidos segundo Delingette. . .	p. 24
2	Comparação entre métodos.	p. 79
3	Comparação entre métodos.	p. 80
4	Comparação entre métodos.	p. 81
5	Comparação entre métodos.	p. 82
6	Dados experimento 1, valores em metros.	p. 85
7	Dados experimento 2, valores em metros.	p. 86
8	Dados experimento 3, valores em metros.	p. 88
9	Dados experimento 4, valores em metros.	p. 89
10	experimento 1a - seção 4.3	p. 106
11	experimento 1b - seção 4.3	p. 107
12	experimento 2 - seção 4.3	p. 108
13	experimento 3 - seção 4.3	p. 109

1 Introdução

Desde a década de 80 as técnicas de computação gráfica aliadas aos avanços de hardware permitem a visualização e manipulação de ambientes tridimensionais. Nos anos 90 tornou-se possível representar não apenas elementos como modelos de arame, mas diversos polígonos e efeitos de texturização, iluminação, variações atmosféricas, neblina entre outros (FOLEY et al., 1990).

A grande dificuldade na visualização de formas geométricas em três dimensões era a geração de imagens através de pixels. Essas operações eram efetuadas por um software executado paralelamente, ocupando um grande tempo de processamento da placa de vídeo e do processador central. Com o surgimento de placas de vídeo 3D, as operações de visualização de formas geométricas passaram a ser executadas pelos processadores destas placas (Graphical Processing Unit - GPU), tornando o processo menos custoso ao processador central, possibilitando aumentar a taxa de atualização das imagens acrescentando realismo (EBERLY; SHOEMAKE, 2004). Apesar deste tipo de dispositivo dedicado a gráficos tridimensionais estar disponível para estações de trabalho profissionais, foi apenas em 1996 com o lançamento da placa Voodoo 1 da 3Dfx que usuários de computadores pessoais tiveram acesso a esta tecnologia (ECCLES; ALLEN, 2000 apud MÖLLER; HAINES, 2002). Durante os anos 90 o desenvolvimento de bibliotecas para a renderização de cenas 3D como DirectX (BARGEN; DONELLY, 1998) e OpenGL (SEGAL; AKELEY, 1993, 2004) facilitou a implementação de aplicações gráficas tridimensionais, adicionando uma camada lógica extra que fazia o acesso aos recursos de hardware. Desta forma o fabricante da placa de vídeo precisava fornecer suporte às instruções da biblioteca, livrando o programador de conhecer detalhes do funcionamento do dispositivo.

O aumento do poder de processamento permitiu não apenas o acréscimo de efeitos garantindo uma representação visualmente mais realista, mas também adicionar características de interação entre diferentes elementos como cinética e colisão. Esse advento permitiu a construção de simuladores 3D e jogos com maiores possibilidades de imersão. Tornou-se factível a manipulação diferentes objetos calculando seu deslocamento e trajetória por aplicação de diversas forças, bem como colisões e deformações garantindo a simulação de efeitos mais realistas como um choque entre veículos, o impacto de um golpe e a deformação que uma ferramenta cirúrgica pode ocasionar em um órgão do corpo humano. Atualmente fenômenos como deslocamento e colisão são largamente implementados, entretanto deformações, comportamentos elásticos e simulações de fluidos têm sua utilização limitada devido ao seu elevado custo computacional. Para aplicações em tempo real deve-se manter uma faixa de atualização da cena em pelo menos 15 quadros por segundo para haver o mínimo de interação com o usuário centrado na ação e reação (MÖLLER; HAINES, 2002).

Este trabalho apresenta um modelo matemático para representação de um tecido em função da aplicação de forças e sua interação com elementos sólidos não deformáveis visando uma simulação a ser executada em tempo real. Segundo Melo e Wu (2002): “Os modelos deformáveis têm sido usados, por exemplo, na manufatura e projeto de vestuários, animação de vestuários sobre figuras humanas, no comportamento de caimento ou cobrimento de tecidos (ou panos) sobre objetos, panos suspensos, animações de expressões faciais de figuras humanas e animais, deformações de objetos 3D, etc. Também são usados na área de medicina para análises de imagens médicas (incluindo segmentação, representação, registro, na captura de movimento da forma), modelagem e animação de músculos, simulações cirúrgicas e treinamento de pessoal, ambos com a exigência de tempo-real e modelagem fisicamente realista, de tecidos humanos complexos, não-lineares e deformáveis (GIBSON, 1997a apud MELO; WU, 2002).” O modelo demonstrado neste trabalho visa atender condições temporais, físicas e visuais para que possa, inclusive, futuramente ser utilizado em um simulador cirúrgico.

Alguns simuladores com diferentes propósitos já foram desenvolvidos, como o ROBO-SIM (REDETZKY et al., 1998) para simulação de neuro-cirurgias. O simulador utiliza um conjunto de

dados padrão, a partir do qual foi feita uma reconstrução com características físicas já definidas. Brown et al. (2001) apresentam um simulador de cirurgias endovasculares, o sistema é capaz de realizar operações como suturas e incisões, porém utiliza um cilindro com pontos conhecidos para representar a artéria e não simula a corrente sanguínea. Heng et al. (2004) apresentam um simulador de cirurgias ortopédicas que modela órgãos deformáveis através de modelos físicos e órgãos rígidos com modelos geométricos a partir de séries de imagens. Moosegaard (2004) apresenta um modelo derivado do massa-mola para simular cirurgias arteriais com dados experimentais. Çavusoglu, Goktekin e Tendick (2006) propõem uma arquitetura com a implementação de algumas etapas intermediárias de um simulador cirúrgico de código livre; envolvendo desde segmentação, reconstrução de dados a partir de uma série de imagens de pacientes reais, até à simulação cirúrgica imersiva com simulação física e representação tridimensional.

A implantação de uma prótese endoluminal é um exemplo de procedimento que poderia ser inserido no escopo de um simulador cirúrgico. Esta atividade requer uma análise pré-operatória de características da artéria como: morfologia, diâmetro e comprimento Schumacher et al. (1997); do mesmo modo que a avaliação destes fatores na escolha de próteses pode reduzir problemas que ocorrem após a cirurgia, como *endoleaks* segundo R.A.White et al. (2000). Um simulador cirúrgico poderia auxiliar neste planejamento, permitindo ao usuário realizar operações como sutura e incisão e verificar em tempo real as conseqüências imediatas de ações executadas.

Os requisitos necessários especificamente para simulações cirúrgicas definidos por Robb e Cameron (1995) são de uma taxa de atualização de 30 quadros por segundo; diferentemente da definição para simulações gráficas quaisquer de (MÖLLER; HAINES, 2002); tempo de resposta de dados de entrada do usuário de no máximo 100ms e idealmente em torno de 10ms. Porém a qualidade visual das cenas deve ser suficientemente convincente, não bastando fornecer velocidade de renderização e captura de eventos gerados pelo usuário. Segundo Delingette (1998), em uma simulação cirúrgica as restrições base para a simulação de tecidos são a exatidão da deformação e o tempo de cálculo. Em uma análise científica para planejamento e definição de implantes ou procedimentos, a precisão dos dados prevalece sobre o tempo de resposta (DELIN-

GETTE, 1998); o mesmo não ocorre em uma simulação em tempo real, cujo maior interesse é garantir respostas em intervalos de tempo satisfatórios ao usuário. As escolhas a respeito do modelo adotado e adaptações realizadas foram feitas preocupando-se com tais fatores.

O comportamento elástico de tecidos pode ser classificado em dois tipos: modelos geométricos e modelos físicos. Os modelos puramente geométricos realizam alterações nos objetos através de transformações geométricas desconsiderando características físicas (MELO, 2004). Não são indicados para representações que possuam variações ao longo do tempo, pois apresentam modelagens estáticas. Os modelos físicos incluem nos cálculos características físicas dos objetos, podendo representar alterações morfológicas em tempo real.

As propriedades físicas dos tecidos foram implementadas estendendo uma estrutura conhecida como DLFL (AKLEMAN; CHEN, 2000, 1999; AKLEMAN; CHEN; SRINIVASAN, 2003) e fazendo uso das suas características topológicas para efetuar a propagação de forças pelos elementos que a compõe. Através de operadores implementados na DLFL (SRINIVASAN; AKLEMAN; KEYSER, 2004), novos elementos podem ser inseridos na mesma, mantendo suas características de rotação. A simulação física foi realizada através de um sistema massa-mola, sempre procurando atender a requisitos de estabilidade da simulação e conseqüente realismo tal qual os requisitos temporais.

O sistema implementado é capaz de atender a aplicação de diversas forças externas simultâneas e retornar ao seu equilíbrio estático instantes após estas forças deixarem de ser aplicadas. O massa-mola é caracterizado por sua velocidade de processamento, mas também por sua instabilidade, portanto medidas foram tomadas para atenuar esse problema; destacando a fórmula de atualização das partículas bem como o algoritmo responsável pela propagação de forças ao longo da malha.

1.1 Objetivo

Desenvolvimento de um modelo computacional para calcular a deformação de um tecido elástico. Aplicar as propriedades cinéticas deste modelo durante o contato de um tecido com

um sólido não deformável, representando graficamente em um ambiente tridimensional com um comportamento realista. Neste trabalho o termo 'realista' se aplica a alterações fisicamente plausíveis, através de restrições que impeçam deformações impossíveis em situações reais.

1.2 Objetivos Específicos

1.2.1 Escolha de um modelo para o cálculo da deformação.

O modelo deve atender aos requisitos especificados por Robb e Cameron (1995), ao mesmo tempo deve ser adaptável e os requisitos temporais devem ser relevados na busca de um grau de realismo convincente de maneira a atingir um equilíbrio.

1.2.1.1 Estabilidade e Fidelidade da Representação

O modelo escolhido deve apresentar deformações condizentes com as leis da físicas, ser tolerante a oscilações e propagar a força proporcionalmente entre as regiões do tecido envolvidas.

1.2.1.2 Aplicação de Forças Externas

O modelo deve permitir a aplicação de múltiplas forças externas sobre o tecido, tanto a ação da gravidade como o contato com um elemento qualquer. Essas forças devem ser independentes de orientação e intensidade.

1.2.1.3 Aplicação de Forças Internas e Equilíbrio Estático

A aplicação de uma força externa deve gerar forças internas de oposição, que tendem a fazer o tecido retornar ao seu estado inicial. Na ausência total de forças externas, após períodos em que o tecido sofreu uma deformação, o modelo deve garantir que o tecido atinja um estado de equilíbrio estático.

1.2.1.4 Escolha do método de integração numérico

Uma simulação de fenômenos físicos em tempo real envolve o cálculo de deslocamento de elementos envolvidos ao longo do tempo. O método deve garantir estabilidade e ser suscetível às variações de aceleração e intervalos de tempo. Deve-se permitir taxas de aceleração variáveis, satisfazendo o item 1.2.1.2. Devido às diversas operações que são executadas a cada iteração, o tempo de execução da simulação não pode ser previsto. Portanto o sistema deve ser capaz de garantir uma simulação com intervalos de tempo variáveis, não necessariamente sendo constantes entre cada iteração.

1.2.2 Representar o contato de um elemento deformável com um sólido rígido

O modelo deve representar a deformação ocasionada em um objeto deformável com um sólido rígido. Esta deve ser proporcional à força exercida pelo sólido no objeto deformável, bem como a força de reação gerada por este contato.

1.2.2.1 Detectar os pontos de colisão

Deve ocorrer checagem de colisão para saber se existem e quais seriam os pontos de contato entre as estruturas.

1.2.2.2 Aplicação de forças

As forças exercidas sobre os elementos, devem ser proporcionalmente aplicadas sobre cada ponto de contato, tal qual as forças de reação.

2 Modelos matemáticos para representação de deformações em tecidos

Neste capítulo serão apresentados os diferentes modelos existentes para representação gráfica do comportamento elástico de tecidos e suas características; tal qual suas limitações e vantagens relacionadas aos objetivos deste trabalho.

Como foi dito anteriormente os modelos para representação de deformações podem ser classificados em modelos geométricos e físicos. A diferença principal é que os modelos puramente geométricos só podem expressar deformações através de técnicas geométricas empregadas por um projetista com conhecimento prévio das alterações (GIBSON, 1997a). Além disso os modelos puramente geométricos não apresentam restrições a alterações de forma, podendo ocorrer transformações não realistas. Nos modelos físicos por sua vez as deformações são decorrentes de contato entre objetos ou da ação de algum agente externo, onde os objetos elásticos respondem naturalmente às forças aplicadas em função de suas características físicas e restrições (MCINERNEY; TERZOPOULOS, 1996). Neste capítulo há uma maior ênfase aos modelos físicos, sendo que a proposta é o desenvolvimento de um modelo computacional que comporte a interação entre objetos sólidos e deformáveis em tempo real.

2.1 Modelos Geométricos

Os primeiros trabalhos relacionados à representação gráfica do comportamento elástico ocorreram na área de CAD (*Computer Aided Design*), ligados diretamente à representação de

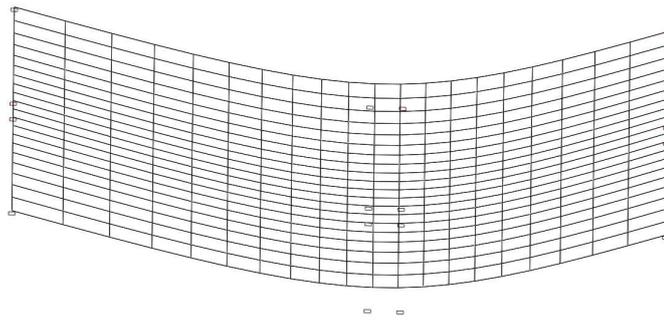


Figura 1: Superfície Bézier, quadriláteros marcam os dezesseis pontos de controle.

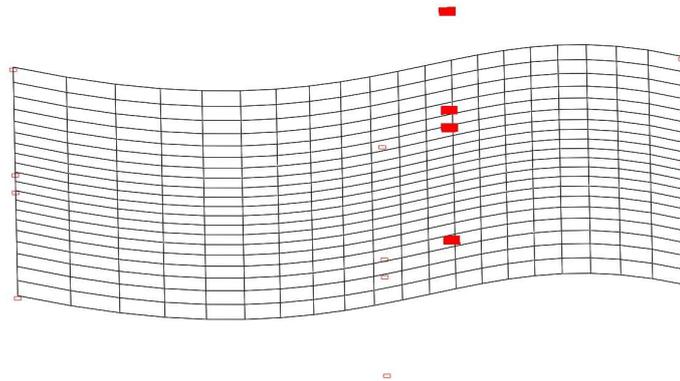


Figura 2: Superfície Bézier, alterando apenas os quatro pontos de controle representados por quadriláteros maiores.

curvas e superfícies. De acordo com a classificação contexto incluem-se representações implícitas como: curvas e superfícies de Bézier, B-splines, B-splines racional, não uniforme B-splines racional, β -splines e superquádricas(forma explícita); tal qual representações implícitas como: superquádricas (forma implícita) e hiperquádricas.

Nas representações explícitas a curva ou superfície é determinada através de parâmetros de forma, neste caso, pontos de controle. Alterações na forma podem ser feitas através de mudanças na coordenadas dos pontos de controle, na remoção destes, inserção de outros pontos de controle ou por meio de ajustes nos pesos destes pontos, como pode ser verificado nas figuras 1 e 2.

Nas representações implícitas a diferença está no número e natureza dos parâmetros que definem a forma da superfície. A inserção de um grande número de parâmetros acarreta um impacto na complexidade das deformações, bem como parâmetros podem ter um impacto exclusivamente local, possibilitando a obtenção de formas complexas. Isto tem como contrapartida

uma elevada complexidade computacional e uma maior instabilidade em relação à aplicação de parâmetros globais(MONTAGNAT; H.; AYACHE, 2001).

No caso de superquádricas os parâmetros que definem sua forma, são conjuntos de superelipsoides previamente definidas por equações paramétricas. Representações com este modelo estão limitadas a formas simétricas, impossibilitando sua aplicação para elementos com um maior número de detalhes(MONTAGNAT; H.; AYACHE, 2001). Hiperquádricas são uma extensão das superquádricas na qual a superfície está contida em um invólucro definido por um conjunto de 3 ou mais planos. A vantagem deste modelo em relação às superquádricas é a possibilidade de realizar modificações interativas na superfície, porém apresenta a característica de estar limitada a uma esfera representada pela subdivisão parametrizada do espaço em que se encontra(MONTAGNAT; H.; AYACHE, 2001).

As representações geométricas possuem em comum a dificuldade de estimar uma deformação precisa da superfície baseando-se nos parâmetros de entrada(GIBSON, 1997a). Este fator é limitante no que diz respeito a simulações de comportamento elástico em tempo real, pois o que se espera de um modelo não é apenas a possibilidade de sofrer modificações interativas, mas sim que essas modificações possam ser resultado direto do contato do material elástico com um agente externo, podendo este ser a ação de uma força como a gravidade ou de uma força que age através de uma colisão com outro elemento.

O simulador e sistema de planejamento cirúrgico Kismet (KUHNAPFEL et al., 1995) realiza operações de deformação em tecidos fazendo uso de representação por NURBS, porém sem que as animações geradas pelas deformações ocorressem em tempo real. Em 2000 desenvolvedores deste sistema adotaram uma abordagem híbrida (CAKMAK; KUHNAPFEL, 2000), incorporando modelagem física em tempo real com a inclusão de operações de deformação realizadas por um sistema massa-mola.

Bardinet, Cohen e Ayache (1996) apresentam uma abordagem de representação do movimento do ventrículo cardíaco esquerdo através de superquádricas. A fidelidade visual é garantida, pois uma reconstrução 3D de imagens de pacientes reais é efetuada, porém a animação

ocorre sem a ação de algum agente externo causador de alguma deformação em tempo real.

2.2 Modelos Físicos

A utilização de modelos físicos em representação de deformações de tecidos surgiu em virtude da carência das soluções baseadas em modelos geométricos para expressar deformações com maior precisão e que fossem decorrentes de contato com objetos, não apenas uma seqüência de alterações de parâmetros pré-estabelecida.

Nessa categoria estão presentes as representações de elementos finitos (NIELSEN; COTIN, 1996) apresentada por Terzopoulos et al. (1987); massa-mola (DOCHEV; VASSILEV, 2003; JI; LI; QIU, 2006) apresentado por Provot (1995), na representação de um modelo de voxels, conhecido como *chain mail* (GIBSON, 1997b; EGLI; STEWART, 2004). Os três são baseados em sistemas de partículas e as diferenças em suas implementações resultam em singulares níveis de flexibilidade, plasticidade, realismo e custo computacional. Delingette (1998) também define a precisão das deformações e o tempo de processamento como as principais restrições em uma simulação de um tecido deformável apresentando uma tabela classificativa entre os modelos de elementos finitos, massa-mola e os que utilizam superfície geométrica implícita, neste caso especificamente o modelo definido por Gascuel e Desbrun (1997).

Modelos	Precisão de Deformações	Tempo de Processamento	Corte	Visualização
Elementos Finitos	***	*	*	***
Massa-Mola	*	**	**	**
Implícitos	*	**	**	*

Tabela 1: Classificação de modelos para deformação de tecidos segundo Delingette.

(* indica má adequação do modelo ao requisito, *** indica boa adequação)

Os modelos de elementos finitos, massa-mola e voxels serão discutidos apontando seus pontos positivos e negativos e os motivos que levaram à escolha do modelo massa-mola.

2.2.1 Elementos Finitos

Modelo para a representação de superfícies e volume largamente utilizado na engenharia, geralmente para calcular os efeitos de deformações sobre resistência de materiais. Este modelo descreve a forma de um objeto como um conjunto de elementos básicos (triângulos, quadriláteros, tetraedros, etc.), chamados elementos finitos. Esses elementos estão interconectados a nós localizados em suas bordas ou interior, a posição destes pontos é um parâmetro a princípio desconhecido. Um conjunto de funções é escolhido para definir unicamente o posicionamento de cada elemento finito, bem como de seus respectivos nós. Essas funções definem as forças de equilíbrio estático (esforço) que existem entre os elementos do material no estado inicial, proporcionalmente a estes valores a tensão do material pode ser calculada ao longo do tempo dependente da aplicação de forças externas. Sempre há um esforço contrário dos elementos, tentando manter sua posição inicial; a tensão gerada é distribuída àqueles elementos aos quais estão conectados gerando uma alteração na forma originalmente representada (ZIENKIEWICZ; ZHU; TAYLOR, 2005).

O trabalho de Terzopoulos et al. (1987) foi pioneiro em utilizar esta modelagem para representar graficamente a deformação de tecidos, figura 3. O resultado foi alcançado empregando métodos numéricos de integração para resolver conjuntos de equações diferenciais que definiam a relação entre os elementos. A simulação foi garantida incluindo como parâmetro de entrada o tempo atual, que influía diretamente na forma final. As deformações ocorriam em função de aplicação de forças externas. Apesar destes resultados obtidos, o modelo não era capaz de gerar simulações com taxas de atualização razoáveis devido a dois fatores determinantes, o poder de processamento dos computadores da época e pela complexidade dos cálculos inerentes a este modelo.

Zhuang e Canny (2000) apresentaram adaptações a este modelo buscando torná-lo mais flexível em relação às variações nos intervalos de tempo entre as iterações, corrigir distorções em deformações globais e reduzir sua complexidade computacional aplicando uma superfície 2D ponderada, figura 4, que apresenta regiões com diferentes áreas de influência, sobre a objeto

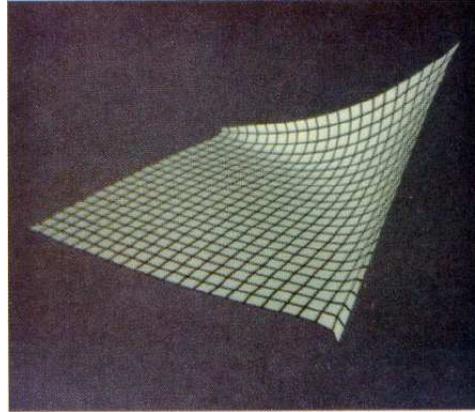


Figura 3: Deformação de tecidos utilizando o modelo de elementos finitos.

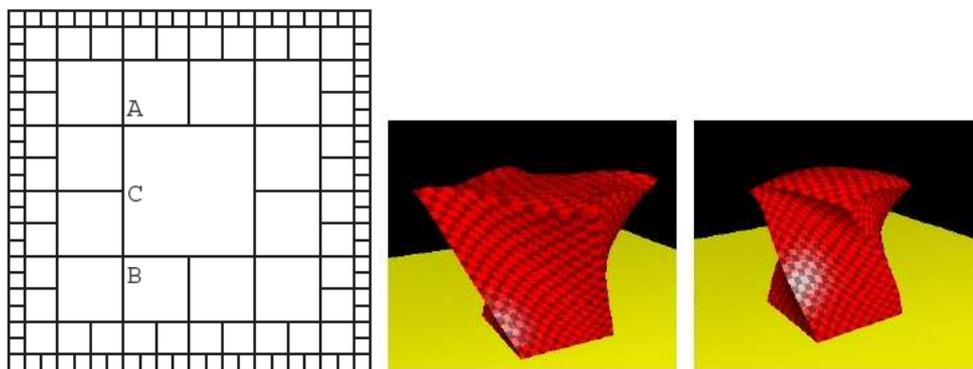


Figura 4: Grid de zhuang à esquerda de imagens capturadas de uma animação onde ocorre a deformação de um paralelepípedo.

3D representado. A aplicação desta superfície consistia em evitar que uma força de tensão fosse propagada para regiões do objeto que não iriam sofrer influência relevante, economizando cálculos computacionais desnecessários. Entretanto Zhuang e Canny (2000) afirmam que é difícil obter um objeto que se encaixe na geometria da superfície.

Uma solução para reduzir a complexidade computacional, foi o cálculo de coeficientes locais de deformação previamente a simulação em tempo real proposta por Kry, James e Pai (2002). Durante a simulação quando um elemento exerce uma influência sobre outro esse coeficiente é aplicado. Para representar graficamente este modelo foi utilizada uma mão em 3D com movimentos capturados por uma luva com sensores nas articulações. Esse sistema mostra-se eficaz para esta aplicação, mas em uma simulação mais complexa, nem todos os movimentos poderiam ser baseados em valores previamente calculados, partindo da imprevisibilidade das operações a serem realizadas. Na figura 5 as partes em amarelo representam algumas das

regiões onde coeficientes locais foram previamente calculados.

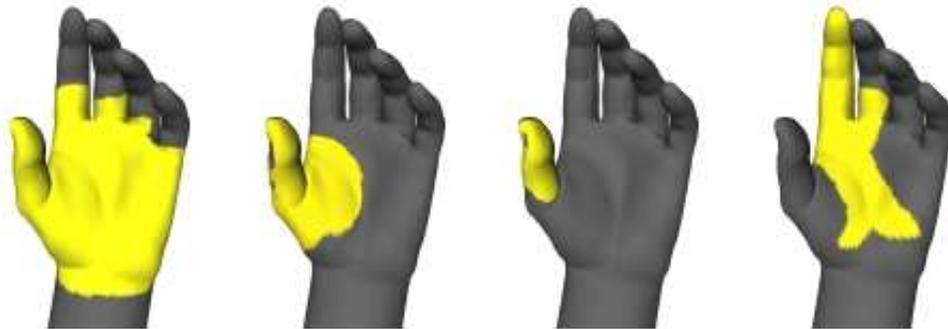


Figura 5: Deformação gerada pelos movimentos da mãos baseada em coeficientes locais calculados como nas regiões em amarelo.

O framework proposto por Çavusoglu, Goktekin e Tendick (2006), apresenta um modelo baseado em elementos finitos para simulações cirúrgicas que atende requisitos de precisão de deformação, colisão, aplicação de forças exercidas por elementos externos, além da simulação de fluidos. A qualidade dos gráficos apresentados pode ser acompanhada na figura 6. Entretanto mesmo aliando isto à capacidade dos computadores atuais; o programa foi executado em computador com processador Intel Xeon 2.8-Ghz com 1Gb de memória RAM e uma placa de vídeo NVidia Quadro FX 1300 com interface PCI-Express. Uma iteração no estudo de caso demonstrado; músculo cardíaco formado por 1415 nós e 4237 elementos em forma de tetraedro com simulação de fluidos representando o sangue, consumiu 246ms com desvio padrão de 1,3ms. Muito aquém dos requisitos estabelecidos por Roob & Cameron.



Figura 6: Simulação de neurocirurgia endoscópica, construída utilizando o framework proposto por Çavosuglu *et al.*

É consenso que simulações gráficas que empregam modelos de elementos finitos possuem maior apelo visual, que em contrapartida significa maior custo computacional. Provavelmente com o desenvolvimento de máquinas mais poderosas no futuro será possível utilizá-lo e atender os requisitos temporais.

2.2.2 Massa-Mola

O modelo de massa-mola tem como base a lei de Robert Hooke, físico contemporâneo a Isaac Newton. Essa lei define que um corpo ou mola que sofre uma deformação gera uma força contrária a este movimento que é igual ao deslocamento da massa a partir do seu ponto de equilíbrio vezes a característica constante elástica. Não estando a mola em sua posição original, ou seja comprimida ou esticada, essa força contrária sempre faz com que ela tenda a voltar a sua posição de equilíbrio. Quando maior for a constante elástica da mola, mais rígido tende a ser o corpo representado, pois gerará uma força maior de oposição ao deslocamento. Essa lei é caracterizada por uma equação linear, entretanto o modelo pode ser adaptado para suportar equações que apresentem outro comportamento. Posteriormente foram introduzidos coeficientes limitantes referentes aos comprimentos máximos que a mola pode alcançar, caso ela exceda esses valores ocorre um rompimento.

O trabalho de Provot (1995) utiliza um sistema massa mola para representar a deformação de tecidos referentes às forças externas à malha. A malha é composta por partículas (pontos de massa) conectados entre si por molas de três tipos diferentes. A diferenciação entre as molas está em quais partículas elas se conectam, se isso se dá ao vizinho imediato nas posições verticais e horizontais (*structural springs*), ao vizinho imediato nas posições diagonais (*shear springs*) ou aos vizinhos distantes duas posições nas direções verticais e horizontais (*flexion springs*). Apesar deste trabalho se mostrar eficaz para representação de tecidos de pano; existem problemas relacionados a comportamentos estranhos como a superelasticidade, que pode ocorrer devido a variações bruscas no intervalo de tempo da simulação. Além disso neste modelo somente as *structural springs* fazem realmente parte da representação gráfica, sendo difícil estimar como o modelo se comportaria em caso de rompimento, pois um ponto de massa não

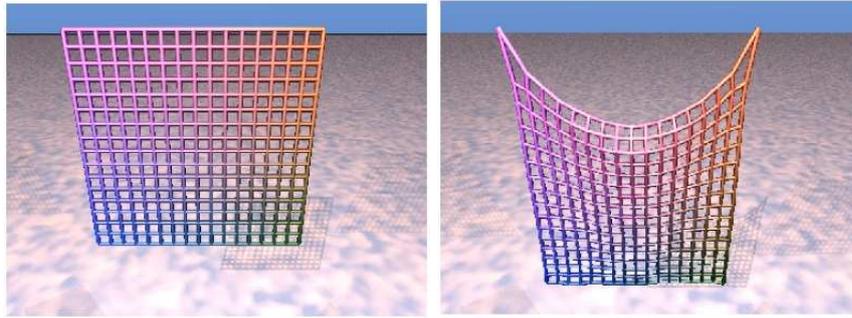


Figura 7: Malha no estado inicial e à direita após 200 iterações da simulação.

faria influência direta apenas aos pontos de massa pertencentes às faces vizinhas diretas. Na figura 7 pode-se observar a deformação da malha com duas extremidades fixas.

Hutchinson, Preston e Hewitt (1996) propõem um algoritmo iterativo para adaptação dos valores de parâmetros do sistema de maneira a evitar comportamentos inesperados e não-realistas com uma estrutura de molas como a de Provot. Desbrun, Schröder e Barr (1999) apresentaram um modelo com um tipo único de mola que trata situações de colisão de tecidos com sólidos, porém o passo de tempo utilizado é constante, característica que nem sempre pode ser garantida; um método para corrigir posições é aplicado. Debunne et al. (2001) estendem o trabalho anterior de Desbrun, Schröder e Barr (1999), introduzindo um método matemático para atenuação da instabilidade em deformações globais. O trabalho de Zhang, Prakash e Sung (2001) demonstra a utilização de superfícies massa-mola para representação de expressões humanas, neste trabalho não há colisão da face com outros elementos. Zhang, Payandeh e Dill (2003) apresentam uma simulação em tempo real com um sistema massa-mola que suporta uma interface háptica, transmitindo sensações tácteis ao usuário.

Brown et al. (2001) apresentam um protótipo de um simulador cirúrgico endovascular que permite operações de incisões e suturas em tempo real, sem simulação de fluidos. O problema da simulação reside na forma utilizada para representar a artéria, um cilindro com molas conectando lados opostos das parede internas, figura 9, não apresentando flexibilidade para definir diferentes objetos.

Dochev e Vassilev (2003) introduziram a velocidade da iteração anterior nos cálculos de posição das partículas para evitar o problema da super-elasticidade, suavizando o movimento,

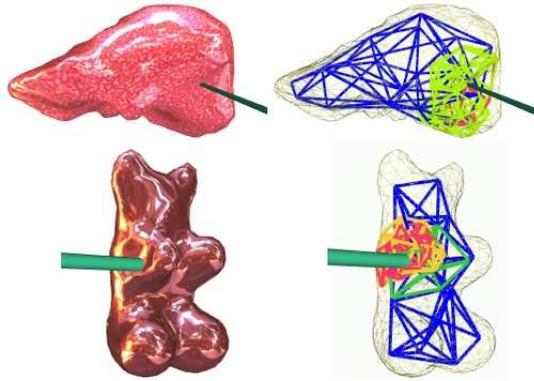


Figura 8: Trabalho de DeBunne, demonstrando deformações em consequência de colisões com ferramentas.

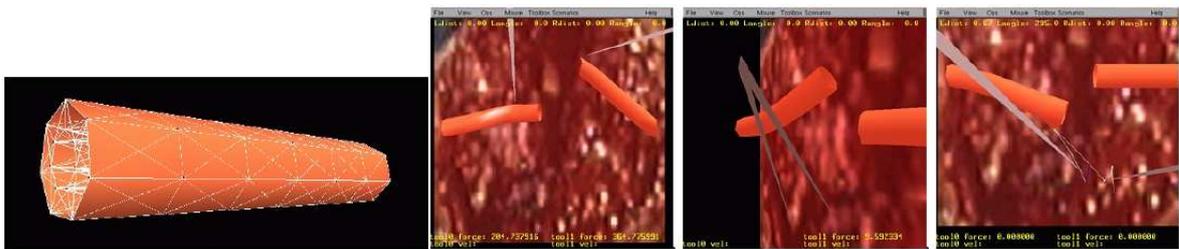


Figura 9: Cilindro para representação de vasos, molas em branco em destaque. Ao lado direito imagens da simulação em tempo real.

amenizando mudanças bruscas na trajetória e por consequência fazendo o sistema atingir o equilíbrio estático na ausência de uma força externa sobre a malha.

Volino e Thalmann (2005) abandonam o modelo de massa-mola proposto por Provot em virtude das dificuldades no que diz respeito a estabilidade da deformação que a topologia das molas gera em deformações globais e adota um modelo de partículas baseado em malhas de triângulos. Os triângulos possuem um equilíbrio estático nas suas posições iniciais e as deformações ocorrem quando as coordenadas dos vértices são alteradas gerando variações de tensão, propagando o esforço gerado aos triângulos vizinhos.

Em relação ao modelo de elementos finitos, os modelos baseados em sistemas massa-mola apresentam menor custo computacional, pois seu elemento básico é uma partícula com um valor de massa; diferente do modelo de elementos finitos onde os elementos básico são polígonos ou poliedros composto por partículas menores. Em um modelo-massa mola a propagação de força respeita uma regra de vizinhança; um movimento em uma partícula pode ser propagado através de partículas conectadas a esta por molas, enquanto a força for maior que um coeficiente

de relevância pré-estabelecido. No modelo de elementos finitos as forças são propagadas por equações que fazem uso de pontos pertencentes a diferentes elementos, não necessariamente vizinhos; logo sua complexidade é proporcional às equações que definem as relações entre os elementos. A vantagem deste modelo é poder representar deformações que não são necessariamente explícitas por conexões que apresentem vizinhança espacial direta. O preço pago pelo menor custo computacional do modelo massa-mola em relação ao modelo de elementos finitos é o menor grau de realismo relacionado as deformações e algumas características que geram instabilidade em uma simulação que serão comentadas ao final deste capítulo.

2.2.3 Chain Mail

O modelo de Chain Mail busca uma representação de sólidos através de pequenas partículas posicionadas lado a lado sem nenhuma ligação por molas ou interferência de vizinhos aos quais não existe um contato direto. Essa modelagem foi proposta por Gibson (1997b) como uma alternativa aos modelos de elementos finitos e massa-mola para representação de deformação em objetos volumétricos. A origem do nome vem da semelhança do modelo a malhas constituídas por conjuntos de elos de correntes. As vantagens imediatas dizem respeito ao menor custo computacional na atualização de posições das partículas, uma facilidade para aplicar operações de corte e união de volumes diferentes, as desvantagens estão em deformações com menor grau de realismo em comparação aos outros modelos.

Neste modelo cada partícula é armazenada em um vetor volumétrico e tem a relação entre os vizinhos diretos. Especificamente no modelo proposto por Gibson, as partículas são pequenos cubos, portanto podem possuir até seis vizinhas diretos, uma para cada face.

Neste modelo cada partícula movida é armazenada em uma lista, suas vizinhas diretas são armazenadas em outras listas de candidatas a movimento de acordo com suas localizações de vizinhança (lista de vizinhas no topo, lista de vizinhas da esquerda...). As listas de candidatas são checadas, comparando a posição anterior e atual das partículas, caso a diferença entre elas seja maior que uma tolerância pré-estabelecida elas são movidas e seus vizinhos armazenados em listas candidatas. Se a partícula já estiver na lista de movidas, ela não é adicionada a ne-

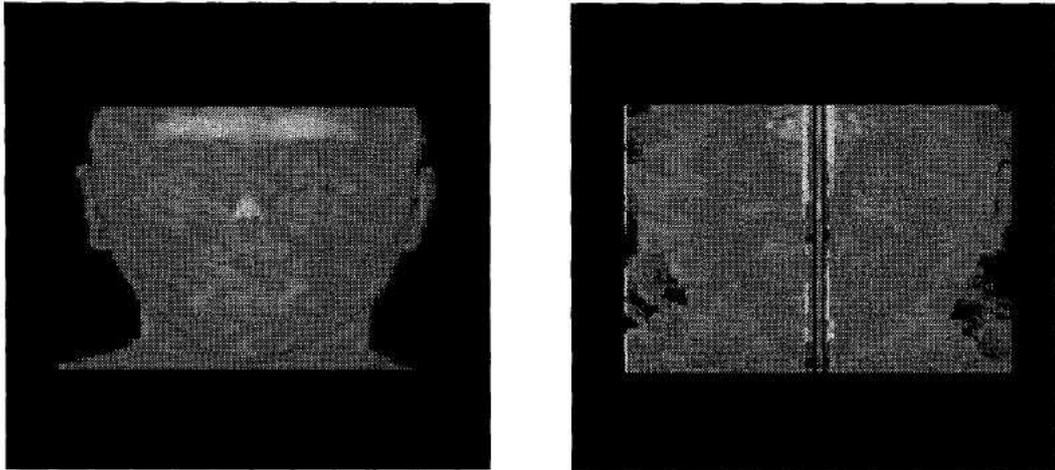


Figura 10: Cabeça reconstruída a partir de série imagens de CT a esquerda, a direita a mesma dividida ao meio.

nhuma lista de candidatas. Isto pode ser considerado uma vantagem sobre outros modelos no que diz respeito a custo computacional, porém implica diretamente na precisão da deformação. Um movimento muito brusco em uma partícula ao longo de várias iterações pode levar a uma propagação inesperada, pois esse movimento pode gerar uma deformação global no objeto representado, fazendo que o movimento se propague por todas as partículas retornando a partícula de origem. Neste caso o movimento não seria computado, pois ela já foi movida uma vez.

A separação e união de objetos formados por chain mail entretanto é relativamente simples, apenas removendo ou adicionando relações de vizinhança entre os elementos. Isto pode ser verificado na figura 10, onde Sakamoto, Tuchiya e Kato (1999) apresentam a divisão de uma cabeça reconstruída a partir de imagens de CT (tomografia computadorizada). Caso as partículas apresentem um volume muito grande, a simulação pode não representar com fidelidade uma operação de corte apenas separando vizinhos, deveria existir a possibilidade quebrar uma partícula. Contudo isto poderia gerar problemas, do tipo uma partícula ter mais de um vizinho direto em um mesmo lado.

Teschner et al. (2004) apresentam um modelo de chain mail no qual partículas volumétricas e tetraedros, formam objetos volumétricos maiores. Este modelo apresenta a vantagem de as partículas também sofrerem deformações, acrescentando realismo visual à simulação. Contudo não sofrem subdivisão em caso de corte. Fragmentar demais o modelo, com tetraedros menores

leva a um custo computacional mais elevado, pois aumentaria o número de propagações a cada iteração.

Apesar destes problemas, chain mail pode ser uma alternativa viável para simulações nas quais a precisão não se faz tão necessária; ou até mesmo ser utilizado para representar elementos não tão relevantes em uma simulação que exija precisão. Como por exemplo, em um simulador de cirurgia endo-vascular, chain mail podia ser utilizado para representar órgãos próximos à artéria.

2.3 Definição do modelo

Através desta pesquisa foi possível analisar as características dos diversos modelos existentes para a representação gráfica de deformações de tecidos, enumerando pontos fracos e fortes de cada um. Os modelos geométricos puramente aplicados servem apenas como ilustração, sendo que não permitem estimar de maneira precisa uma deformação em tempo real. A utilização destes em simuladores fica condicionada a uma solução híbrida, adotando também um modelo físico.

Dentre os modelos de deformação que envolvem operações físicas, destacam-se os de elementos finitos, massa-mola e chain mail. Os modelos de elementos finitos apresentam resultados mais realistas, porém demandam elevado custo computacional, podendo não ser aplicáveis em tempo real até para as máquinas padrão desktop recentes, vide Çavusoglu, Goktekin e Tendick (2006). Os modelos massa-mola são largamente utilizados para deformação de têxteis; em operações nas quais existe um maior grau de realismo, tanto do ponto de vista da ação de forças quanto visual eles são preteridos pelos elementos finitos. Entretanto, alguns trabalhos apresentaram alternativas para contornarem os problemas do modelo massa-mola, tornando possível a realização de simulações em tempo real conferindo um grau satisfatório de realismo. Os modelos de chain mail apresentam cálculos mais simples que os modelos de massa-mola, tal qual uma menor precisão de deformações em relação a estes. Estes modelos são adequados para simulações nas quais o grau de realismo tem um valor menor, em função da velocidade

de processamento devido a seu menor custo computacional. Este trabalho busca a adoção de um modelo que possa ser aplicado em uma simulação gráfica em tempo real, não podendo sacrificar tempo de processamento das operações em detrimento exclusivo da fidelidade física da simulação.

Os passos seguintes deste trabalho foram tomados a partir destes fatos. A decisão em torno do modelo massa-mola, deve-se não apenas pelo seu equilíbrio entre realismo de deformação versus velocidade de processamento; mas também por observar pontos em que o mesmo poderia ser melhorado, apresentando uma alternativa às propostas existentes.

Os principais problemas do massa-mola residem na sua instabilidade, gerando resultados imprevisíveis e na incapacidade de retornar ao equilíbrio estático, podendo apresentar deformações irreversíveis. Este trabalho busca atenuar os efeitos negativos do modelo massa-mola, introduzindo um método de integração numérica mais estável para os cálculos de deslocamentos das partículas bem como uma propagação baseada em uma estrutura 2-manifold que garante que o sistema tenda a retornar ao estado inicial após a ausência de propagação de forças sem apresentar o problema da superelasticidade.

3 Estrutura para armazenar informações geométricas

Neste capítulo será apresentada a estrutura utilizada para representar as superfícies e que posteriormente foi estendida para a inclusão de propriedades físicas. Esta estrutura possui características topológicas que foram essenciais para a construção do modelo no que se refere à propagação de forças ao longo de um tecido por ela representado, bem como a manutenção do equilíbrio estático na ausência da ação de forças externas. A princípio serão discutidas algumas estruturas com propostas diferentes para representação de superfícies e o porquê da escolha da DLFL. Posteriormente serão demonstradas suas principais características; algumas adaptações realizadas no trabalho de Santos (2006) e como estas contribuíram para a inclusão de propriedades físicas.

3.1 Estruturas para a Representação de Objetos Tridimensionais

Além das propriedades físicas a serem abordadas, a representação geométrica de um objeto foi o primeiro critério considerado na escolha de uma estrutura para a representação de um tecido. A busca foi por um modelo no qual pudessem ser apresentados tecidos de diferente dimensões de largura e comprimento, permitindo a representação de detalhes gráficos e alterações morfológicas que surgiriam em decorrência de deformações geradas durante a simulação. A estrutura deve apresentar a possibilidade de alcançar elementos vizinhos, de modo a propagar eventuais deformações, bem como fornecer a possibilidade de identificar colisões.

Modelos para representação de volume por interior como grids e Octrees foram descarta-

dos. Esses modelos dividem o espaço em elementos menores que, ao serem agrupados, dão forma a elementos mais complexos. Apesar de estruturas deste tipo permitirem a representação com alto nível de detalhes, isto acarretaria em uma maior utilização de memória em relação a modelos de representação por bordas. Geralmente essa utilização é justificável quando existe a necessidade de separar elementos particionando o espaço no qual estão contidos através de alguma heurística, como a variação dos componentes de cor que apresentam por exemplo. Este tipo de modelo não é adequado para a aplicação direta do modelo massa-mola sem alterações nas características de conectividade, pois não existe uma estrutura que conecte fisicamente os elementos que apresentem vizinhança espacial direta, neste caso, que pudesse ser representada geometricamente como uma 'mola'. Octrees apresentam conexões hierárquicas, mas essa característica não seria aplicável a este modelo, pois ele necessita de conexões que representem como as forças de equilíbrio de um elemento estão organizadas. Como o resultado final da divisão do espaço nessas estruturas é um conjunto de pequenos elementos volumétricos, a aplicação de modelos de deformação de elementos finitos ou chain mail seria mais indicada, também efetuando devidas adaptações.

Estruturas para a representação de bordas, também conhecida como *b-reps*, apresentam um objeto através de conjuntos de elementos como faces, vértices e arestas. Mäntylä (MANTYLA, 1988 apud SANTOS, 2006), classificou estas estruturas como baseadas em polígonos (faces), vértices ou arestas; de acordo com o tipo do elemento utilizado para acessar os demais. Por exemplo, face como elemento chave: ao percorrer cada face, encontram-se os vértices que a compõe e conseqüentemente as arestas que os conectam. A vantagem de armazenar objeto em faces ao invés de pequenos elementos volumétricos é que a forma não é limitada dimensionalmente pelo tamanho mínimo que os elementos poderiam assumir. A adaptação do modelo massa-mola a esse tipo de estrutura ocorre de uma maneira mais natural, sendo que existe o conceito de arestas que conectam vértices, assim como partículas que apresentam valores de massa estão conectadas a molas. O trabalho de Santos (2006) adota uma solução híbrida dos dois modelos, utilizando octrees para efetuar operações de segmentação, fornecendo informações como dado de entrada para a construção de uma *b-rep*. A figura 11 apresenta um mesmo

objeto em estruturas de representação de volume e de bordas.

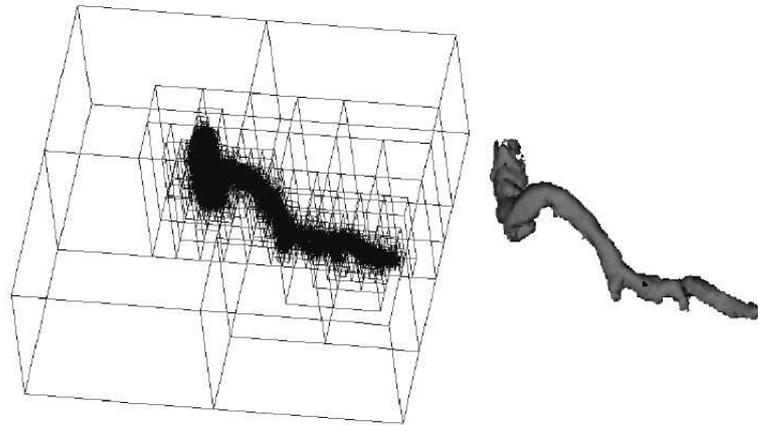


Figura 11: Representação de uma artéria proveniente de uma reconstrução tridimensional a partir de uma série de imagens de tomografia. A esquerda representada por uma Octree, a direita por uma representação por bordas (DLFL) (SANTOS, 2006).

A consistência entre os componentes da estrutura (faces, aresta e vértices) é mantida através de um critério topológico que garante que a malha representada por seus elementos apresenta a característica definida como 2-manifold. Esta característica define que a estrutura pode ser representada num espaço euclidiano. Todos os vértices e arestas de uma estrutura tridimensional podem ser dispostos em um único plano sem que haja cruzamento de arestas localmente; ou seja, uma estrutura tri-dimensional pode ser representada topologicamente por um conjunto de estruturas dispostas em um espaço bi-dimensional. A garantia de manutenção da topologia 2-manifold é conquistada através de operadores de inserção e exclusão de vértices e arestas. Essa topologia também garante a característica de Euler, expressa pela equação:

Algoritmo 1 Característica de Euler. vértices(v), arestas(e) e faces(f).

$$v - e + f = 2$$

Mantidas estas propriedades, todas as arestas da malha podem ser percorridas em um único ciclo; ou seja, não há repetição de arestas durante uma travessia completa pela estrutura. Essa

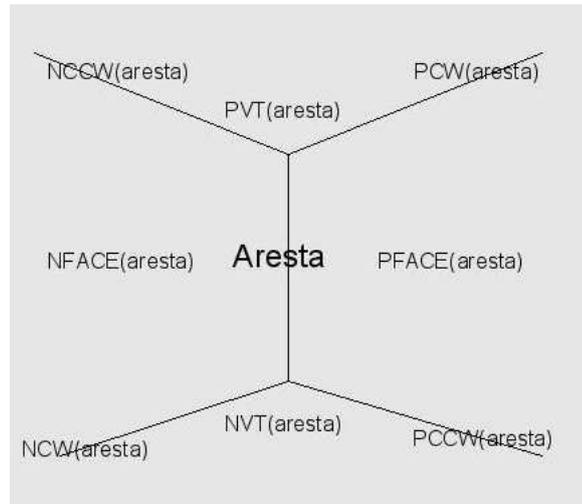


Figura 12: Conexões da aresta na estrutura *winged edge* (adaptado de www.baumgart.org)

característica é fundamental ao modelo de propagação de forças que será explicado no capítulo seguinte.

Baumgart (1972) introduziu a *winged edge*, estrutura *b-rep* baseada em arestas. Na definição de *winged edge*, um objeto é representado por quatro elementos básicos. O corpo é um destes elementos, contendo um apontador para três listas circulares, uma de arestas, outra de vértices e uma de faces. Cada face e cada vértice apontam diretamente a apenas uma face no seu perímetro. Cada aresta possui conhecimento das quatro arestas (PCW, PCCW, NCW e NCCW) que são suas vizinhas imediatas, bem como das duas faces (NFACE e PFACE) a que está ligada, além dos vértices aos quais se conecta (NVT e PVT) como se pode ver na figura 12. Essa topologia permite que partindo de uma aresta qualquer se possa encontrar qualquer outro elemento na estrutura. As características 2-manifold são garantidas através de operadores para inserção, exclusão e modificação na estrutura, os quais são chamados operadores de Euler.

Mäntylä (MANTYLA, 1988 apud SANTOS, 2006) desenvolveu uma estrutura chamada *half-edge* baseada na *winged edge*, que é capaz de representar 2-manifolds com bordas. Ela apresenta as mesmas características, com exceção nas bordas, em que as arestas não possuem a mesma identidade topológica (SANTOS, 2006). A *half-edge* tem a adição de um novo elemento, o anel no qual dentro pode existir ou não uma face, a ausência de uma face significa a presença de uma borda, na qual existe um novo elemento, chamado de meia-aresta. Mäntylä alterou os

operadores propostos por Baumgart, apresentando apenas dez operadores que efetuam modificações na estrutura.

Apesar de manterem as propriedades 2-manifold, estruturas baseadas em aresta podem apresentar inconsistências em modificações utilizando os operadores propostos (AKLEMAN; CHEN; SRINIVASAN, 2003 apud SANTOS, 2006). Akleman apresentou uma estrutura baseada em faces que garante sempre a característica 2-manifold das malhas apresentadas com apenas 4 operadores. Na próxima seção esta estrutura será demonstrada, na seção subsequente serão explicadas as mudanças apresentadas por dos Santos e quais vantagens elas trazem ao modelo de deformação proposto neste trabalho.

3.2 DLFL

Estrutura proposta inicialmente por Chen e implementada por Akleman e Chen, a DLFL (Double Linked Face List); implementa um sistema de rotação de grafo que garante sempre uma malha 2-manifold orientada, realizando alterações através dos quatro operadores básicos (SANTOS, 2006). Esta condição é importante para garantir uma simulação realista durante uma situação de colisão. Na simulação uma malha elástica pode entrar em contato com algum outro objeto, interceptando faces que a compõe, sendo necessária realizar uma subdivisão da face intersectada para destacar o ponto de contato. Como a manutenção da característica 2-manifold é garantida, não existem problemas do ponto de vista topológico em realizar esta operação durante a simulação.

A DLFL é composta por uma lista de faces, uma de arestas e uma lista de vértices. Na adaptação proposta por Santos (2006), cada face é composta por uma lista duplamente encadeada dos extremos de arestas. As arestas possuem um par destes extremos aos quais se conectam. Os vértices contém as coordenadas que os identificam no espaço tridimensional e uma lista de apontadores para todos os extremos aos quais pertencem. A figura 13 demonstra como os diferentes elementos que constituem a DLFL se organizam. Essa topologia permite alcançar em um algoritmo de busca qualquer elemento desejado partindo de qualquer outro, sendo ele face,

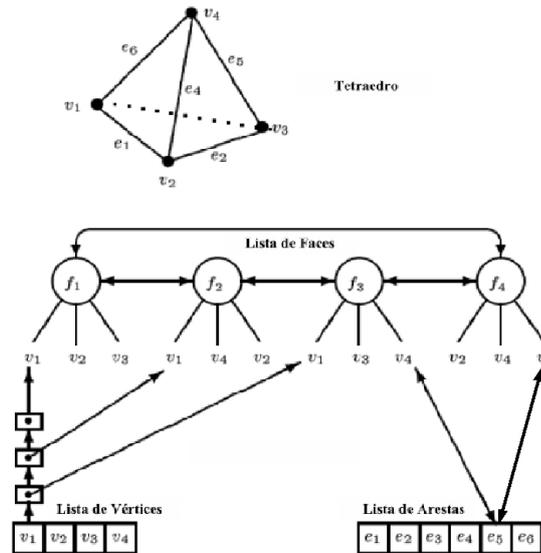


Figura 13: Organização dos elementos da DLFL na representação de um tetraedro, adaptado de (AKLEMAN; CHEN; SRINIVASAN, 2003 apud SANTOS, 2006).

vértice ou aresta; respeitando a ordem aos quais eles estão conectados sem que haja a repetição de uma aresta previamente visitada. No trabalho de Carvalho, Santos e Wangenheim (2006), está descrito o algoritmo que realiza esta tarefa fazendo uso dos operadores da DLFL. Essa travessia em uma ordem correta de conectividade é essencial para o algoritmo de propagação de forças, de modo a transmitir uma deformação aos vizinhos.

Os quatro operadores básicos da DLFL executam as operações de inserção e remoção de um vértice e o mesmo para uma aresta. As faces são organizadas durante a criação de uma aresta, de maneira a manter a característica 2-manifold. Um vértice ao ser criado é colocado em uma nova face, sem nenhuma conexão com uma aresta. A remoção de um vértice da estrutura só é possível, se todas as conexões que ele possui com arestas sejam previamente removidas (SANTOS, 2006). As operações mais complexas como subdivisão de faces, ou união de duas estruturas diferentes são efetuadas fazendo uso desses quatro operadores básicos.

A DLFL entretanto apresenta uma limitação no que diz respeito a representação de sólidos com bordas e de estruturas *non-manifold*. Estruturas apresentam esta condição quando uma única aresta faz parte da constituição de mais de duas faces. Santos (2006) introduziu alterações na DLFL que permitem a representação de estruturas com estas condições mantendo as características 2-manifold.

3.3 Modificações na DLFL

Durante uma simulação na qual as decisões são tomadas em tempo real, os eventos devem ser tratados em tempo de execução. A interação de um elemento deformável com um sólido pode levar não apenas a deformações imprevisíveis como também alterar a topologia e a constituição do tecido, através de abertura de fissuras ou a união deste tecido com outro. Sendo o modelo computacional de deformação dependente da manutenção da topologia 2-manifold, a estrutura escolhida deve ser capaz de tratar alterações morfológicas, que alterando a topologia, mantenham-se essas condições.

As modificações apresentadas por Santos (2006) são capazes de contornar estas situações. Apesar de o modelo de deformação ainda não fazer uso de todas as novas funcionalidades, devido a problemas que serão discutidos no capítulo cinco. Santos definiu uma heurística para a inserção de arestas em superfícies com bordas, sendo a borda representada por uma face com rótulo que diferencia seu estado de uma face convencional. A representação de estruturas non-manifolds pela DLFL é efetuada através da adição de dois novos operadores.

No trabalho de Santos (2006) o algoritmo marching cubes proposto por Lorensen e Cline (1987) foi adaptado para possibilitar o armazenamento de sólidos tridimensionais resultantes de reconstruções de séries de imagens médicas na DLFL. Através do pré-processamento que busca separar os órgãos internos é possível identificar visualmente estruturas diferentes, como pode ser visto na figura 14. Esta funcionalidade abre a possibilidade para futuras adaptações do modelo computacional de deformações de tecidos e interação com sólidos aqui proposto para ser utilizado no contexto de um simulador cirúrgico que trate informações provenientes de pacientes reais.

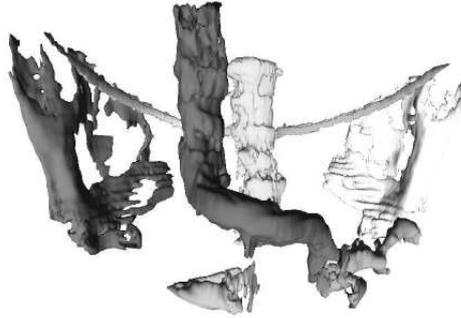


Figura 14: Reconstrução a partir uma série de CT previamente segmentada (CARVALHO; SANTOS; WANGENHEIM, 2006).

Além destas contribuições, o trabalho de Santos apresenta um operador no qual se busca o extremo de cada aresta. Este operador é utilizado no modelo de deformação aqui proposto; sua importância bem como as adaptações realizadas na estrutura proposta por Santos serão discutidas no capítulo seguinte.

4 Modelo para Deformação de Tecidos

Este capítulo demonstra como a DLFL modificada por Santos (2006) foi adaptada para a inserção de características físicas. A etapa de propagação de forças através da estrutura sofreu diversas modificações neste processo, no que diz respeito a ordem e método de busca de vizinhos e na extensão dos elementos que constituem a DLFL. Esta etapa foi desenvolvida baseada em execução de experimentos e observação dos fenômenos físicos.

As alterações na forma de um tecido representado são decorrentes das forças aplicadas a seus elementos. Em uma simulação em tempo real não apenas valores dos vetores que representam forças devem ser levados em conta para efetuar estas transformações, porém também como eles estão relacionados ao tempo. O cálculo de deslocamentos das partículas é realizado através de operações de integração em função da variação dos valores de tempo entre as iterações da simulação. Diferentes métodos de integração foram analisados; os pontos positivos e negativos de cada um serão aqui apresentados em sintonia com a adaptabilidade ao modelo de deformação. O comportamento elástico do tecido é dependente dos resultados destas equações que levam em conta também as características físicas assumidas por suas molas e partículas. Uma nova extensão foi integrada ao sistema de partículas, permitindo incluir em uma mesma estrutura partículas sensíveis à aplicação de forças e partículas fixas.

A aplicação de uma força externa sobre um tecido pode gerar uma deformação; dependendo dos efeitos ocasionados, essa transformação pode ou não ser reversível. O equilíbrio estático do tecido é alcançado através de novas adaptações no algoritmo de atualização de posições das partículas. Dependendo da disposição dos elementos da estrutura, a mesma pode retornar a sua posição inicial, ou a um estado no qual suas partículas praticamente cessem o movimento na

ausência de uma força externa.

4.1 Propagação de Forças Através da Estrutura

O sistema massa mola é composto essencialmente por partículas ligadas entre si por molas. Ao movimentar uma partícula, as molas conectadas a ela sofrem uma alteração no seu comprimento original, gerando uma força segundo a lei de Hooke, conforme foi demonstrado no capítulo dois. Esta força de reação da mola contrária a mudanças de seu tamanho movimentam seus extremos de modo a restaurar seu comprimento original, a figura 15 mostra variações de direção assumidas pelas molas em função da alteração do seu comprimento. A direção desta força será dependente do movimento ocasionado pela partícula, bem como se isto resultou numa compressão ou expansão da mola.

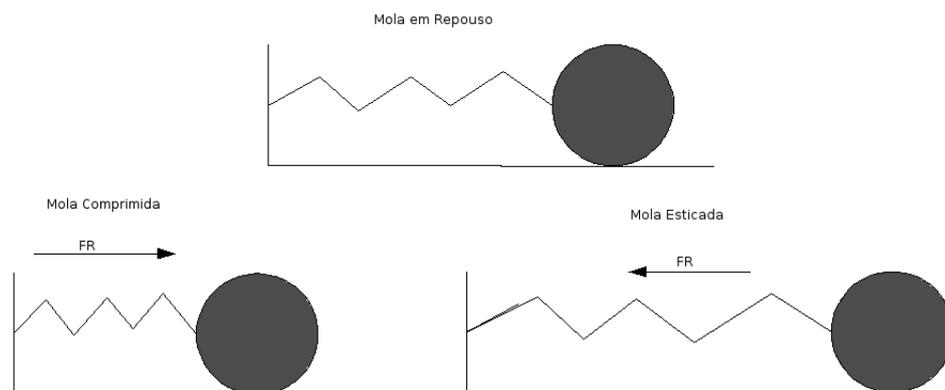


Figura 15: Comportamento de um sistema massa-mola composto uma mola ligada a um ponto fixo e a uma partícula móvel.

Calculada a força de restauração da mola; esta é aplicada em cada uma das partículas, gerando um novo movimento. Em um sistema que envolva diversas conexões de molas com partículas essa perturbação na mola pode se propagar a outras, como está demonstrado na figura 16. O movimento em uma partícula pode causar uma alteração no comprimento de outras molas às quais esse vértice está conectado. Esta alteração pode deslocar também partículas vizinhas e assim sucessivamente. Para evitar a propagação de movimentos que fossem gerar deslocamentos irrelevantes nas partículas é adotado um coeficiente de mínimo de deformação ε , caso a movimentação de uma partícula gere uma variação no tamanho original da mola superior

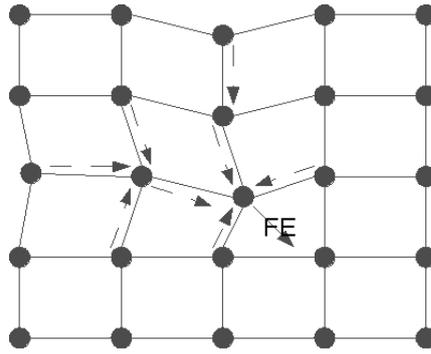


Figura 16: Propagação de forças em uma malha (setas tracejadas), resultantes da aplicação de uma força externa FE.

a ε ; somente então a força é propagada aos vizinhos.

Cada vértice da DLFL foi estendido para adquirir as características de uma partícula, atribuindo a ele um valor de massa μ e a possibilidade de calcular seu deslocamento de acordo com a força a qual sofreu ação. A aceleração de uma partícula é expressa de acordo com a segunda lei de Newton:

$$a = \frac{F}{\mu}$$

O deslocamento de uma partícula é calculado a partir da sua velocidade no instante anterior e de sua aceleração atual em função do tempo. Para obter a posição de uma partícula no espaço, soma-se a posição no instante anterior com o deslocamento calculado, como se verifica na equação de Euler para cálculo de deslocamento:

$$v_{i+1} = v_i + \Delta t \cdot a \quad (4.1a)$$

$$x_{i+1} = x_i + v_{i+1} \cdot \Delta t \quad (4.1b) \text{ onde:}$$

x_{i+1} posição da partícula no instante t_{i+1}

x_i posição da partícula no instante t_i

v_{i+1} velocidade da partícula no instante t_{i+1}

v_i velocidade da partícula no instante t_i

$\Delta t = t_{i+1} - t_i$ variação de tempo entre os instantes t_{i+1} e t_i

Como há a necessidade de conhecer a velocidade no passo anterior para a atualização, a partícula deve ter conhecimento da mesma tal qual de sua posição espacial. A velocidade é utilizada no cálculo da força de amortecimento da mola, uma força contrária à força de restauração. Este cálculo será ignorado nesta seção, tornando a ser discutido na última seção deste capítulo. Outra utilidade da velocidade que será discutida mais adiante é sua inserção em cálculos de correção de posição de partículas após detecção de colisões.

O deslocamento de uma partícula implica que as molas a ela conectadas sofrerão uma alteração no seu comprimento. A adaptação da DLFL ao sistema massa-mola se fez tratando as arestas como molas atribuindo a elas novas características; além do já mencionado coeficiente mínimo de deformação ε . Neste modelo as molas possuem uma constante elástica k_e , o comprimento da mola no estado inicial l_0 , e o valores máximo de comprimento que ela pode assumir sem que a haja rompimento l_{max} . O comprimento atual da mola l é calculado pela distância euclidiana dos dois vértices aos quais ela conecta. O cálculo da força de restauração F_r da mola segue a Lei de Hooke:

Algoritmo 2 Cálculo da força de restauração do mola.

$$\Delta l = l - l_0$$

se $(|\Delta l| > \varepsilon)$ então

$$F_r = \Delta l \cdot k_e$$

senão

$$F_r = 0$$

Como já foi dito anteriormente, outra equação que não especifique a lei de Hooke pode ser adaptada para cálculo da força de reação da mola. A adoção desta equação deve-se a sua simplicidade, causando menor impacto no custo computacional em relação a uma equação mais complexa.

Sempre que nos algoritmos os termos partícula e mola forem citados, são referentes às extensões feitas ao vértice e à aresta da DLFL respectivamente. Forças externas são aquelas aplicadas por um agente não pertencente à malha, internas são as forças de restauração das molas. A primeira tentativa de propagação de forças foi através de uma busca em profundidade

pelos vértices da estrutura. Ao ocorrer a ação de uma força externa sobre uma única partícula, o deslocamento desta era calculado e sua posição atualizada, através da seguinte lógica.

Algoritmo 3 Propagação da força em travessia por profundidade.

propagação(partícula p , Força F , Tempo Δt)

início

 calculada a aceleração sobre p de acordo com a segunda lei de newton

 atualizada a posição de p segundo a equação 4.1

 Para cada extremo de aresta e pertencente à lista de extremos de arestas de p faça

 início

$mola \leftarrow aresta(e_i)$ {operador da DLFL, retorna a aresta a qual e_i pertence}

 se($visitada(mola) = falso$)então {confere se a aresta já foi visitada}

 início

$visitada(mola, verdadeiro)$ {marca a aresta como visitada}

$F \leftarrow$ força calculada aplicando o algoritmo 2 sobre $aresta$

 se ($F \neq 0$)então

 início

$p_i \leftarrow vertexComplementar(e_i)$ {operador da DLFL, retorna o outro vértice da aresta a qual o extremo pertence}

 propagação(p_i, F)

 fim

 fim

 fim

fim

Este algoritmo usa uma lógica recursiva para atingir todos vértices passíveis de atualização de posição, propagando as forças por todas as arestas da DLFL. Ao marcar as arestas visitadas se evita com que ocorra repetição. Entretanto esse algoritmo apresenta problemas, pois atualiza

os vértices durante a busca em uma ordem de propagação de forças não esperada. O movimento gerado não apresentava um comportamento satisfatório. Em um sistema massa mola as forças se propagam radialmente e não em profundidade, sempre respeitando um grau de vizinhança, como demonstra a figura 17.

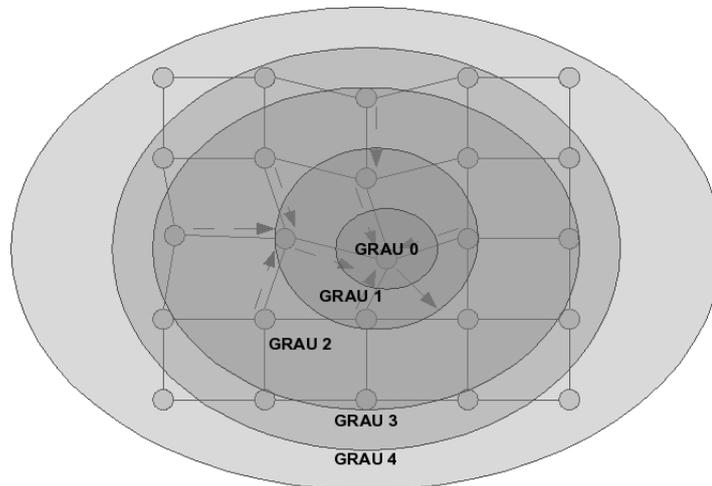


Figura 17: Ordem de propagação das forças.

Caso um vértice seja alcançado a partir de outro por apenas uma aresta, a vizinhança entre eles é dita de grau um; caso seja alcançado visitando duas arestas, a vizinhança é de grau dois. A propagação de forças de maneira radial se faz percorrendo inicialmente todas as arestas do menor grau possível, só assim passando a percorrer as arestas do grau seguinte. Dochev e Vassilev (2003) apresentam um modelo no qual a ordem de percorrimento dos vértices ocorre de maneira radial a partir de pontos estáticos na malha, como demonstrado na figura 18. O principal problema dessa técnica é que existe uma etapa intermediária que organiza a ordem na qual as posições dos vértices devem ser computadas, permitindo checar vértices que poderiam não sofrer alterações.

O algoritmo 3 foi modificado para percorrer a estrutura em largura, ao invés de profundidade, existindo a necessidade de checar o grau de vizinhança de cada vértice visitado em relação àquele de onde a força foi aplicada.

Durante a aplicação de uma única força pontual, com força externa agindo em apenas um único vértice, verificou-se que os vértices eram checados na ordem correta, porém a checagem

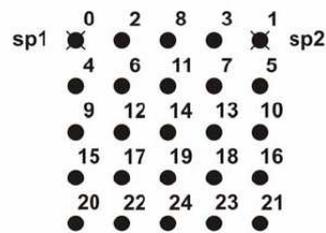


Figura 18: Trabalho de Dochev e Vassilev (2003), ordem dos vértices organizada de acordo com a posição de pontos estáticos (sp)

de grau do vértice a cada visita, bem como contínua aplicação da equação 4.1 para atualizar as posições tornaram a execução muito mais custosa. Pois a cada nova força aplicada vértices já percorridos poderiam ser visitados novamente .

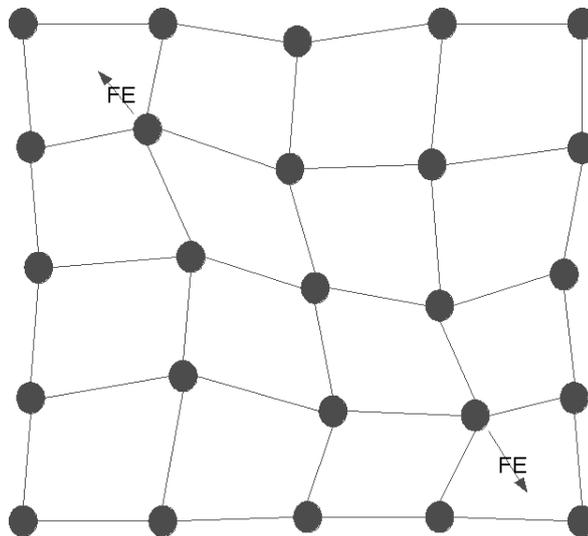


Figura 19: Sistema massa-mola sofrendo a ação de duas forças simultâneas

Contudo na aplicação de múltiplas forças simultâneas, como na figura 19 as partículas não eram atualizadas corretamente, pois uma segunda força poderia causar deformações em molas já modificadas por outra força. Ou seja esse processo é dependente não apenas da ordem a qual partícula é alcançada, mas também a ordem a qual elas são atualizadas em função da força propagada. Além deste fatores, nesta etapa foi encontrada uma falha, a força de reação da mola estava sendo aplicada integralmente a apenas um vértice, quando na verdade ela deve ser aplicada em cada um dos vértices em sentidos opostos, como demonstra a figura 20.

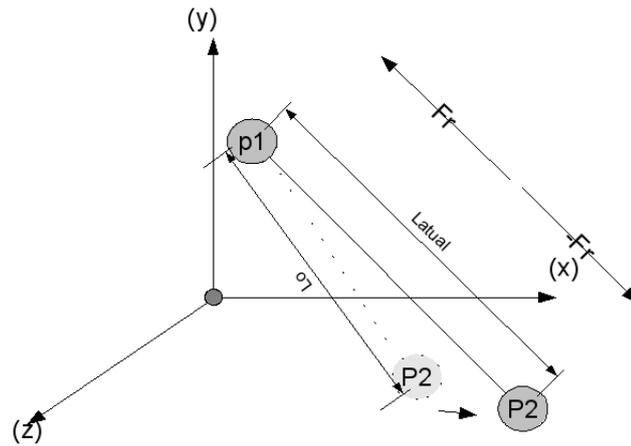


Figura 21: Cálculo da Força de reação da mola em R^3 .

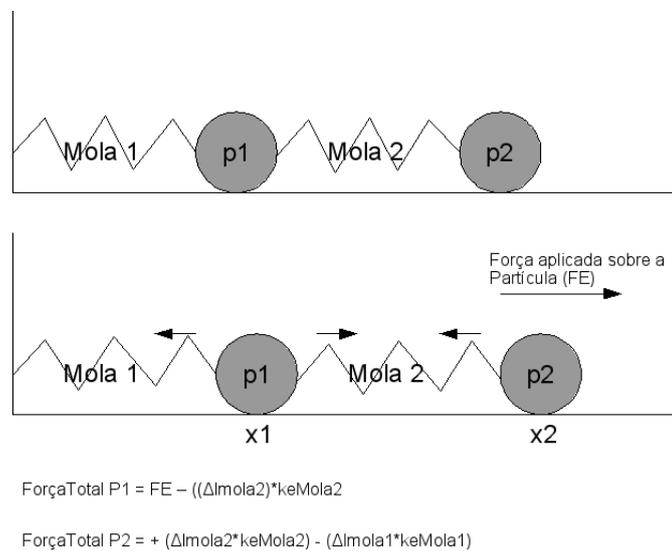


Figura 20: Cálculo das forças em um sistema com duas partículas

A aplicação deste algoritmo no espaço R^3 exigiu uma adaptação. A intensidade da força de reação da mola (Fr) é calculada como na figura anterior, porém a direção do vetor é calculada subtraindo a posição de uma partícula que a mola conecta pela outra, e normalizando este vetor diferença. A Fr aplicada em cada uma das molas é produto da intensidade pelo vetor direção.

$$\vec{F}_r = (L_{atual} - L_0) \cdot k_e \cdot |(p2_{posicao} - p1_{posicao})|$$

Estes problemas foram solucionados, chegando a uma nova versão do modelo algoritmo para propagação de forças. A grande diferença que pode ser notada é que as forças a serem aplicadas em cada vértice lhe são atribuídas, acumulando-as durante a travessia de propagação,

sendo atualizadas apenas no final de uma iteração completa. Desta forma a propagação pode ocorrer em qualquer ordem, pois os vértices serão atualizados apenas no final do processo, não havendo a necessidade de se preocupar com alterações de posição fora de ordem. Nesta travessia, porém ocorre a necessidade de remarcar a atribuição de visitada da aresta como falsa a cada vez que se deseja aplica um par de força externa à partícula.

Algoritmo 4 Propagação da força em travessia por profundidade para a aplicação de forças externas em múltiplas partículas.

propagação(vetorPartículas vPartículas, vetorForças vForças, Tempo Δt) {par de vetores com mesmo tamanho, o índice de cada um coincide com a força aplicada sobre cada partícula}

início

Para $i=1$ até $i \leq$ tamanho(vPartículas) faça

início

acumulaForça(vPartículas[i], vForças[i])

propagação(vPartículas[i], vForças[i]) {algoritmo 5}

Para cada mola da DLFL faça

visitada(listaDeArestas[i], falso)

fim

Para cada partícula da DLFL faça

início

calculada a aceleração sobre p de acordo com a equação 4.1

atualizada a posição de p_i segundo a equação 4.1 usando a força

acumulada em p_i

fim

fim

Algoritmo 5 Propagação de forças em profundidade atualizado

 propagação(partícula p , Força F , Tempo Δt)

início

 acumula(p, F) {acumula a força na partícula}

 Para cada extremo de aresta e pertencente à lista de extremos de arestas de p faça

início

 mola \Leftarrow aresta(e_i) {operador da DLFL, retorna a aresta a qual e_i pertence}

 se($visitada(mola) = falso$)então {confere se a aresta já foi visitada}

início

visitada(mola, verdadeiro) {marca a aresta como visitada}

 $Fr \Leftarrow$ força calculada aplicando o algoritmo 2 sobre mola

 se ($Fr \neq 0$)então

início

 acumula($p, -Fr$) {acumula a força de restauração na partícula no sentido inverso}

 $p_i \Leftarrow$ vertexComplementar(e_i) {operador da DLFL, retorna o outro vértice da aresta a qual o extremo pertence}

 propagação(p_i, Fr)

fim

fim

fim

 fim

Nesta etapa os problemas ao longo do capítulo descritos foram resolvidos. O sistema permite a aplicação de múltiplas forças em uma malha, a topologia da DLFL permite que todas as partículas passíveis de alteração de posição sofram atualização independente da ordem na qual são alcançadas. O desafio seguinte é a obtenção do equilíbrio estático do sistema na ausência de forças externas; até aqui, a propagação de forças foi disparada pela aplicação de uma força

externa. Contudo o sistema deve ser capaz de entrar em posição de equilíbrio na ausência destas forças, ou seja, as molas devem exercer forças sobre as partículas de maneira a restaurar seu comprimento original. A obtenção deste modelo ocorreu com a introdução de conceitos utilizados em um método de integração diferente do algoritmo 4.1, que será analisado na seção seguinte.

4.2 Métodos de Integração

A equação de Euler para cálculo de deslocamento espacial, permite calcular o deslocamento de uma partícula em um intervalo de tempo estabelecido. Se este intervalo de tempo se manter constante a cada iteração, sua utilização possibilita a manutenção da estabilidade em uma simulação envolvendo várias partículas, porém o mesmo não pode ser garantido se o intervalo de tempo for alterado com frequência entre as iterações da simulação. Nos ensaios realizados o tempo não se manteve constante, apresentando taxas de atualização da simulação variáveis.

Esta equação é construída a partir de equações de diferenças de primeira ordem, podendo ser resolvidas por um método de integração numérico; como o do Trapézio, Simpson ou Romberg, dividindo o intervalo de tempo em intervalos menores e os utilizando como atributos de entrada. A dificuldade reside em estimar esta quantidade de intervalos menores, visto que o intervalo de tempo entre iterações pode variar. Além do fato destes métodos serem iterativos, nos quais adota-se como critério de parada um número máximo de iterações ou uma precisão pré-estabelecida. Esta segunda forma de garantir a parada do algoritmo possui uma imprevisibilidade inerente, pois é custoso estimar a priori quantas iterações serão necessárias para alcançar a precisão desejada. A princípio foi empregada uma modificação a equação de Euler, integrando manualmente o passo de atualização da velocidade a ser incluído no cálculo do deslocamento, porém a velocidade precisa ser atualizada para a iteração seguinte por um método numérico. A equação (4.2a) é diferencial de segunda ordem, o que garante maior suavidade de variações entre iterações consecutivas.

$$x(t + \Delta t) = x(t) + \Delta t \cdot v \quad (4.1a)$$

$$v(t + \Delta t) = v(t) + \Delta t \cdot a \quad (4.1b)$$

integrando (4.1b) se obtém $v(t)\Delta t + \frac{\Delta t^2 \cdot a}{2}$

substituindo em (4.1a):

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{\Delta t^2 \cdot a}{2} \quad (4.2a)$$

Porém, a necessidade de calcular a velocidade através de um método de integração numérico, além de tornar a atualização das posições dependente da escolha do número de passos, introduziria um custo computacional extra.

Um método muito utilizado para cálculo de deslocamento de partículas é o método popularizado por Verlet (1967), contudo também é conhecido como Störmer-Verlet devido a experimentos realizados por C. Störmer em 1907 (HAIRER; LUBICH; WANNER, 2003). Entretanto em 1860, J.F. Encke já havia utilizado equações de diferenças de segunda ordem que apresentavam a mesma forma para cálculo de influência de perturbações em órbitas planetárias (HAIRER; LUBICH; WANNER, 2003). O algoritmo de Verlet pode ser deduzido a partir do algoritmo 4.2a:

$$x(t_{i+1}) = x(t_i) + v(t_i) \cdot (t_{i+1} - t_i) + \frac{(t_{i+1} - t_i)^2 \cdot a_i}{2} \quad \text{algoritmo (4.2a)}$$

$$\text{sendo } x(t_{i+1}) - x(t_i) = v(t_i) \cdot (t_{i+1} - t_i) + \frac{(t_{i+1} - t_i)^2 \cdot a_i}{2} \quad (4.3a)$$

$$x(t_i) - x(t_{i-1}) = v(t_{i-1}) \cdot (t_i - t_{i-1}) + \frac{(t_i - t_{i-1})^2 \cdot a_{i-1}}{2} \quad (4.3b)$$

O próximo passo é retirar a velocidade do cálculo da equação (4.3a):

$$v(t_i) = v(t_{i-1}) + a_i \cdot (t_i - t_{i-1}) \equiv v(t_{i-1}) = v(t_i) - a_i \cdot (t_i - t_{i-1}) \quad (4.3c).$$

substituindo em (4.3b) :

$$x(t_i) - x(t_{i-1}) = (v(t_i) - a_i \cdot (t_i - t_{i-1})) \cdot (t_i - t_{i-1}) + \frac{(t_i - t_{i-1})^2 \cdot a_{i-1}}{2} \equiv$$

$$x(t_i) - x(t_{i-1}) = v(t_i) \cdot (t_i - t_{i-1}) - a_i \cdot (t_i - t_{i-1})^2 + \frac{(t_i - t_{i-1})^2 \cdot a_{i-1}}{2} \equiv$$

$$x(t_i) - x(t_{i-1}) = v(t_i) \cdot (t_i - t_{i-1}) - \frac{(t_i - t_{i-1})^2 \cdot a_{i-1}}{2} \quad (4.3d)$$

Para o próximo passo será feita uma afirmação, a qual os efeitos serão discutidos posteriormente. Assumindo que não houve alteração na aceleração entre os dois instante de tempo,

sendo $a_i = a_{i-1} = a$, bem como $(t_{i+1} - t_i) = (t_i - t_{i-1}) = \Delta t$. Adicionando $a \cdot \Delta t^2$ em cada lado da equação tem-se:

$$x(t_i) - x(t_{i-1}) + a \cdot \Delta t^2 = v(t_i) \cdot \Delta t - \frac{(\Delta t)^2 \cdot a}{2} + a \cdot \Delta t^2 \equiv$$

$$x(t_i) - x(t_{i-1}) + a \cdot \Delta t^2 = v(t_i) \cdot \Delta t + \frac{(\Delta t)^2 \cdot a}{2} \quad (4.3e)$$

O lado direito da equação (4.3e) é exatamente igual ao lado direito da equação (4.2a) portanto se:

$$x(t_{i+1}) - x(t_i) = v(t_i) \cdot (\Delta t) + \frac{(\Delta t)^2 \cdot a}{2} \text{então:}$$

$$x(t_{i+1}) - x(t_i) = x(t_i) - x(t_{i-1}) + a \cdot \Delta t^2 \equiv$$

$$x(t_{i+1}) = (x(t_i) - x(t_{i-1})) + x(t_i) + a \cdot \Delta t^2 \quad (4.3f)$$

A este algoritmo dá-se a denominação de *Position Verlet*, pois exclui-se a velocidade dos cálculos de deslocamento. Trabalhos como o de Batcho e Schlick (2001) e Qian e Schlick (2002) apresentam uma análise detalhada desse modelo, no que diz respeito a sua melhor exatidão em comparação a outros.

Esta equação assume que a aceleração não sofreu variação entre os intervalos de tempo, e manteve-se constante nesses instantes. Na prática isto ocorre pois o sistema não atualiza a aceleração durante uma iteração apenas no começo desta, porém o mesmo não se pode dizer a respeito do tempo, que geralmente sofre variações. Devido a isto foi utilizada uma versão do algoritmo de Verlet (TCV-Time Corrected Verlet), como a demonstrada em Qian e Schlick (2002), que ameniza este problema.

$$x(t_i) - x(t_{i-1}) = v(t_i) \cdot (t_i - t_{i-1}) - \frac{(t_i - t_{i-1})^2 \cdot a_{i-1}}{2} \quad (4.3d)$$

multiplica-se cada lado da equação por $(t_{i+1} - t_i)/(t_i - t_{i-1})$

$$(x(t_i) - x(t_{i-1})) \cdot \frac{(t_{i+1} - t_i)}{(t_i - t_{i-1})} = (v(t_i) - \frac{(t_i - t_{i-1}) \cdot a}{2}) \cdot (t_i - t_{i-1}) \cdot \frac{(t_{i+1} - t_i)}{(t_i - t_{i-1})} \equiv$$

$$(x(t_i) - x(t_{i-1})) \cdot \frac{(t_{i+1} - t_i)}{(t_i - t_{i-1})} = (v(t_i) - \frac{(t_i - t_{i-1}) \cdot a}{2}) \cdot (t_{i+1} - t_i) \quad (4.4a)$$

Neste ponto assume-se que $(t_i - t_{i-1})$ que multiplica a é igual a $(t_{i+1} - t_i)$, chegando à:

$$(x(t_i) - x(t_{i-1})) \cdot \frac{(t_{i+1}-t_i)}{(t_i-t_{i-1})} = v(t_i) \cdot (t_{i+1} - t_i) - \frac{(t_{i+1}-t_i)^2 \cdot a}{2} \quad (4.4b)$$

Acrescentando $a \cdot (t_{i+1} - t_i)^2$ em cada lado da equação chega-se em:

$$\begin{aligned} a \cdot (t_{i+1} - t_i)^2 + (x(t_i) - x(t_{i-1})) \cdot \frac{(t_{i+1}-t_i)}{(t_i-t_{i-1})} &= v(t_i) \cdot (t_{i+1} - t_i) - \frac{(t_{i+1}-t_i)^2 \cdot a}{2} + a \cdot (t_{i+1} - t_i)^2 \equiv \\ a \cdot (t_{i+1} - t_i)^2 + (x(t_i) - x(t_{i-1})) \cdot \frac{(t_{i+1}-t_i)}{(t_i-t_{i-1})} &= v(t_i) \cdot (t_{i+1} - t_i) + \frac{(t_{i+1}-t_i)^2 \cdot a}{2} \quad (4.4c) \end{aligned}$$

O lado direito da equação pode ser substituído pelo esquerdo de (4.4a):

$$\begin{aligned} a \cdot (t_{i+1} - t_i)^2 + (x(t_i) - x(t_{i-1})) \cdot \frac{(t_{i+1}-t_i)}{(t_i-t_{i-1})} &= x(t_{i+1}) - x(t_i) \equiv \\ x(t_{i+1}) &= x(t_i) + a \cdot (t_{i+1} - t_i)^2 + (x(t_i) - x(t_{i-1})) \cdot \frac{(t_{i+1}-t_i)}{(t_i-t_{i-1})} \quad (4.4d) \end{aligned}$$

A clara vantagem deste modelo em relação aos demais, é não necessitar do auxílio de um método de integração numérica para apresentar resultados com maior exatidão. Entretanto, este algoritmo apresenta a característica de utilizar em seus cálculos posições e tempo das duas últimas iterações, não apenas da iteração imediatamente anterior. Portanto há necessidade de a partícula contar com a informação dos seus dois últimos valores de posição. Isto é um problema pois o modelo precisa de uma equação que inicie em repouso, com uma posição conhecida. Porém o maior problema deste método é o cálculo da velocidade no instante t_i , sendo necessário conhecer x_{i+1} e x_{i-1} . No modelo aqui proposto foi utilizada uma outra variação de Verlet, conhecida como Velocity Verlet. Essa equação é obtida a partir da equação de Verlet partindo do princípio $x(t_i)dt = v(t_i)$ e $v(t_i)dt = a(t_i)$; ela foi apresentada por Swope et al. (1982).

$$\text{partindo de } x(t_{i+1}) = (x(t_i) - x(t_{i-1})) + x(t_i) + a \cdot \Delta t^2 \quad (4.3f):$$

$$\text{chega-se a } a_i = \frac{(x(t_{i+1}) - 2x(t_i) + x(t_{i-1})))}{\Delta t^2} \quad (4.5a) \text{ e } v_i = \frac{(x(t_{i+1}) - x(t_{i-1}))}{2\Delta t} \quad (4.5b) :$$

atribui-se uma nova grandeza conhecida como z tal que $z(t_i) = \frac{(x(t_{i+1}) - x(t_i))}{\Delta t}$ (4.5c) pode-se afirmar que:

$$\begin{aligned} x(t_i) &= x(t_{i-1}) + \Delta t \cdot z(t_{i-1}) \text{ logo, sendo } a = x(t_i)dt^2 \text{ então } z(t_i) = z(t_{i-1}) + \Delta t \cdot \\ x(t_i)dt^2 & \quad (4.5d) \text{ pois} \end{aligned}$$

$$\frac{(x(t_{i+1}) - x(t_i))}{\Delta t} = \frac{(x(t_i) - x(t_{i-1}))}{\Delta t} + \frac{(x(t_{i+1}) - 2x(t_i) + x(t_{i-1})))}{\Delta t^2} \cdot \Delta t \quad (4.5e)$$

$$v(t_i) = \frac{(z(t_i) + z(t_{i-1}))}{2} \quad (4.5f) \text{ chegando a:}$$

$$x(t_{i+1}) = x(t_i) + v(t_i) \cdot (t_{i+1} - t_i) + \frac{(t_{i+1} - t_i)^2 \cdot a_i}{2} \quad (4.5g)$$

$$v(t_{i+1}) = \frac{v(t_i) + (a_i + a_{i+1}) \cdot \Delta t}{2} \quad (4.5h)$$

As equações aqui apresentadas foram testadas em um experimento que apresenta condições como aceleração e tempo variável; fenômenos que podem perfeitamente ocorrer na simulação. Independente das características que levaram à escolha pelo método Velocity Verlet. Este método foi comparado a outros no que diz respeito a sua exatidão de resultados, não levando em conta qualquer força calculada em função da velocidade.

O método de Velocity Verlet, as duas variações anteriores do Positional Verlet e o método de Euler (equação 4.2) foram comparados em experimentos que simulam situações que podem ocorrer em uma simulação. Os experimentos foram adaptados de (DUMMER, 2005) e podem ser conferidos no capítulo 6. As tabelas com todos os valores obtidos estão no capítulo 8 de anexos.

Na seção seguinte será demonstrada a versão final da equação adotada para a atualização das partículas, bem como novas características inerentes ao algoritmo de propagação de forças para que esta característica em especial seja mantida. A equação de Velocity Verlet foi adaptada para funcionar no espaço R^3 .

4.3 Manutenção de Equilíbrio Estático

Nesta seção serão demonstradas as adaptações realizadas nos algoritmos e equações das duas seções anteriores para a construção do modelo definitivo de deformação. Uma deformação em um tecido ocasionada por uma força externa dificilmente será reversível caso não existam partículas físicas fixas envolvidas, ou aplicação de outras forças externas. Por exemplo, ao amassar uma camiseta ela não retornará ao estado anterior sem que haja a aplicação de uma força externa. Para verificar a reversibilidade do modelo, foram inseridas partículas não sensíveis às forças externas ou de restauração das molas, chamadas partículas estáticas. Um experimento foi realizado com esse intuito e será discutido mais ao final da seção.

Até aqui o modelo deformação apresenta alguns problemas relacionados à manutenção do

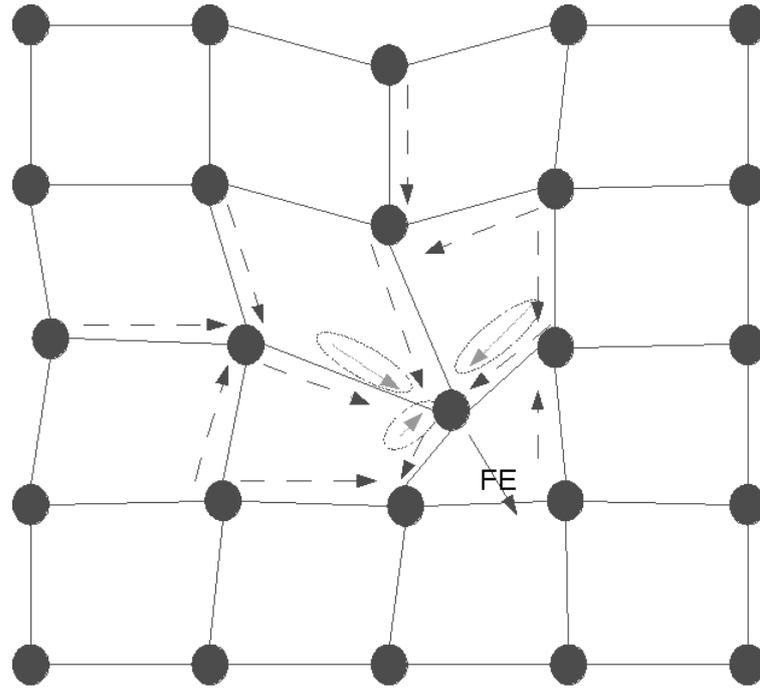


Figura 22: Vetores circulos por elipses demonstram forças aplicadas no vértice de onde partiu a força externa, porém são provenientes da propagação das forças de reação da mola.

equilíbrio estático:

- Todas as forças se propagam a partir dos vértices aos quais as forças externas são aplicadas, na ausência de uma força externa nada mais ocorria.
- Durante a travessia responsável pela propagação de uma única força externa, uma aresta marcada não é mais visitada. Mesmo que todas aquelas passíveis de modificação sejam visitadas, a força de restauração de uma mola pode ser significativa a ponto de ocasionar uma alteração na posição de uma partícula na mesma iteração. Como é demonstrado na figura 22.
- A força de amortecimento da mola ainda não está sendo levada em consideração.

O primeiro problema foi resolvido aplicando um rótulo no vértice, caracterizando se ele foi visitado ou não durante as travessias para propagação de forças externas. Após isto a lista de vértices é percorrida conferindo se o vértice foi visitado ou não, caso falso, sua velocidade no instante é checada, estando acima de um limiar pré-estabelecido, as molas a ele conectadas são verificadas e dependendo de heurísticas já comentadas, a força de restauração das molas é

propagada. Independente de ter ocorrido a ação de alguma força externa, as partículas que estavam em movimento na iteração anterior serão checadas, bem como nas iterações subseqüentes, até que sua velocidade seja desprezível, ou seja, que o equilíbrio entre as partículas tenha sido atingido. Esse caso também é válido para a situação na qual uma região com partículas com velocidade considerável não é afetada pela ação de alguma força externa vigente.

A resolução do segundo problema consistiu não em marcar as arestas, mas seus extremos. Desta forma uma aresta pela qual uma força já foi propagada, poderá propagar uma força de reação desde que esta seja relevante; em outro sentido.

A força de amortecimento da mola passou a ser incluída nos cálculos. Para tal ela foi somada com a força incidente em determinada partícula pela força aplicada na mesma pela propagação. Atenção para kd , geralmente um valor negativo, visto que é uma força contrária a velocidade.

$$F = +(-kd \cdot v)$$

Abaixo segue a versão definitiva de todos os algoritmos envolvidos com a etapa de propagação de forças com as adaptações em relação aos algoritmos anteriores.

Algoritmo 6 Versão definitiva, cálculo da força de reação da mola

 $\vec{F}r \text{forcaDeReacao}(Particula\ p, Mola\ mola)$

{A força de reação da mola precisa ser um Vetor em R^3 , os passos a seguir tem por intuito determinar a direção e intensidade desta força}

inicio

$$\Delta l = comprimento(mola) - mola.l_0$$

se $(|\Delta l| > mola.\varepsilon)$ então

$$intensidade = \Delta l \cdot ke$$

senão

inicio

$$\vec{F}r \Leftarrow 0$$

retorna

fim

Partícula pComplementar \Leftarrow *vertexComplementar*(e_i)

$$direcao = p_{posicao} - pComplementar_{posicao}$$

normalize($direcao$)

$$\vec{F}r \Leftarrow direcao \cdot intensidade$$

fim

Algoritmo 7 Versão definitiva, Propagação de Forças

propagação(Partícula p , Força \vec{F})

início

 acumula força \vec{F} na partícula p

 Para cada extremo de aresta e pertencente à lista de extremos de arestas de p faça

 início

 marca e_i como visitado

 marca p como visitada {posteriormente é interessante saber se p foi ou não percorrida}

$mola \leftarrow aresta(e_i)$ {operador da DLFL, retorna a aresta a qual e_i pertence}

 início

$extremo \leftarrow extremoComplementar(e_i)$ {operador da DLFL, retorna o outro extremo da aresta a qual o extremo pertence}

 se $extremo$ não foi visitado

 início

$\vec{F}r \leftarrow foradereacao(p, mola)$

 se($\vec{F}r \neq 0$) então

 início

 acumula $\vec{F}r$ em p

$p_i \leftarrow particula(extremo)$ {busca a partícula a qual o extremo referencia}

$propagacao(p_i, -F)$ {aplica a força da mola no sentido inverso em p_i }

 fim

 fim

 fim

 fim

 fim

Algoritmo 8 Versão definitiva, deslocamento por Velocity Verlet

{O algoritmo de Velocity Verlet necessita dos valores de $\frac{\Delta t^2}{2}$ e $\frac{\Delta t}{2}$ além de Δt . Como esses valores são aplicados a cada partícula que se deseja calcular o deslocamento, são passados como parâmetro ao invés de serem recalculados a cada chamada do procedimento. Durante o cálculo da nova posição da partícula, a posição anterior é armazenada; a finalidade desta operação será discutida no próximo capítulo}

atualizaPos(Particula *p*, Tempo Δt , Tempo $\frac{\Delta t^2}{2}$, Tempo $\frac{\Delta t}{2}$)

início

$$\vec{novaPos} \leftarrow p_{\vec{posicao}} + p_{\vec{velocidade}} \cdot \Delta t + p_{\vec{aceleracao}} \cdot \frac{\Delta t^2}{2}$$

$F_{\vec{amortecimento}} \leftarrow p_{\vec{velocidade}} \cdot p_{kd}$ {força de amortecimento, calculada pela velocidade vezes o coeficiente de amortecimento kd .

Por questões de modelagem esse coeficiente foi atribuído à partícula.}

$$p_{\vec{Facumulada}} \leftarrow p_{\vec{Facumulada}} + F_{\vec{amortecimento}}$$

$$p_{\vec{aceleracao}} \leftarrow \frac{p_{\vec{Facumulada}}}{p_{\mu}} + p_{\vec{aceleracao}}$$

$$p_{\vec{velocidade}} \leftarrow p_{\vec{aceleracao}} \cdot \frac{\Delta t}{2}$$

$$p_{\vec{posicaoAntiga}} \leftarrow p_{\vec{posicao}}$$

$$p_{\vec{posicao}} \leftarrow \vec{novaPos}$$

fim

Algoritmo 9 Algoritmo de atualização de sistema de partículas

```

atualizaSistemaParticula( VetorPartículas vPartículas, vetorForças
vForças, Tempo  $\Delta t$ ) {par de vetores com mesmo tamanho, o índice de cada
um coincide com a força aplicada sobre cada partícula}

início
     $\frac{\Delta t}{2} = \Delta t \cdot 0,5$ 
     $\frac{\Delta t^2}{2} = \Delta t \cdot \frac{\Delta t}{2}$ 
    Para i=1 até i<= tamanho(vPartículas) faça
        início
             $particula[i]_{forcaAcumulada} \leftarrow 0$ 
            propagação(vPartículas[i], vForças[i]) {algoritmo 5}
        fim
        Para cada partícula da DLFL faça
            início
                se ( $particula[i]_{visitada} = falso$ ) então
                    se ( $particula[i]_{velocidade} > \epsilon v$ ) então {PASSO A}
                        propagação(  $particula[i]$ , 0) {PASSO B}
                    fim
                Para cada partícula da DLFL faça
                    início
                        se ( $particula[i]_{visitada}$ ) então
                            início
                                 $atualizaPos(particula[i], \Delta t, \frac{\Delta t^2}{2}, \frac{\Delta t}{2})$ 
                                 $particula[i]_{visitada} \leftarrow falso$ 
                                Para cada extremo que referencia  $particula[i]$  DLFL faça
                                     $extremo[i]_{visitado} \leftarrow falso$ 
                                fim
                            fim
                        fim
                    fim
                fim
            fim
        fim
    fim

```

No passo marcado como A, são somados os valores das coordenadas que compõem a veloci-

dade da partícula, caso este valor seja maior que uma constante de irrelevância pré-estabelecida, a posição da partícula é atualizada. Este passo visa detectar partículas que estão retornando ao equilíbrio estático, ou seja estão em movimento, porém não sofrem interferência de alguma força externa.

No passo B; como não há interferência de força externa, é passado um vetor nulo. Porém isto não significa que nenhuma alteração de posição ocorrerá, pois as forças de reação da mola podem exercer influência.

No capítulo 6 o algoritmo com a versão definitiva da propagação de forças foi testado através de alguns experimentos.

O capítulo 4 apresentou como o modelo computacional abordou questões relacionadas a propagação de forças e a obtenção do equilíbrio estático em uma malha que incorpora características físicas de um sistema massa-mola. Até então as forças externas são aplicadas em pontos isolados e previamente definidos, não há nenhum agente que entre em contato com a malha ocasionando uma deformação. No capítulo seguinte será abordada a relação de um tecido deformável com sólidos rígidos.

5 Interação de Objetos Sólidos Rígidos com Objetos Deformáveis

Neste capítulo será discutido a respeito do contato de sólidos rígidos com tecidos, os fenômenos que esta operação pode causar e as diferentes abordagens empregadas na representação de ambos. Um tecido deformável sofre uma alteração morfológica ao entrar em contato com um objeto sólido rígido; o resultado desta modificação depende da forma, posição, movimento e velocidade aos quais os dois objetos apresentam no instante exato da colisão.

Os sólidos rígidos podem admitir tanto propriedades cinéticas, como serem permanentemente imóveis; invariantes a qualquer força que o tecido possa exercer sobre os mesmos. Esta última característica não se aplicaria em um cenário ideal, porém existem casos em que a força aplicada por um tecido sobre um sólido rígido, considerando as características espaciais e valores de massa, não será relevante para ocasionar uma alteração em sua posição. Por exemplo jogar uma toalha sobre uma mesa não causará grandes alterações na posição da mesa, não sendo necessário calcular o deslocamento que essa força desprezível causaria. Neste caso, o sólido considerado permanentemente imóvel apresenta um valor de massa muito superior ao de qualquer malha que venha a chocar-se com ele. Entretanto o choque de uma bola de vôlei contra uma rede pode causar alterações não apenas na rede, mas também na trajetória da bola.

No presente modelo, os algoritmos de detecção de colisão prezam pela precisão utilizando cálculos fundamentais da geometria analítica para identificar os pontos de contato. O objetivo é encontrar estes pontos e tratar as conseqüências; não focalizando em técnicas de colisão otimizadas do ponto de vista do custo computacional. Os algoritmos de colisão aqui apresentados são baseados em busca por força bruta, abrindo possibilidades para futuras otimizações desta

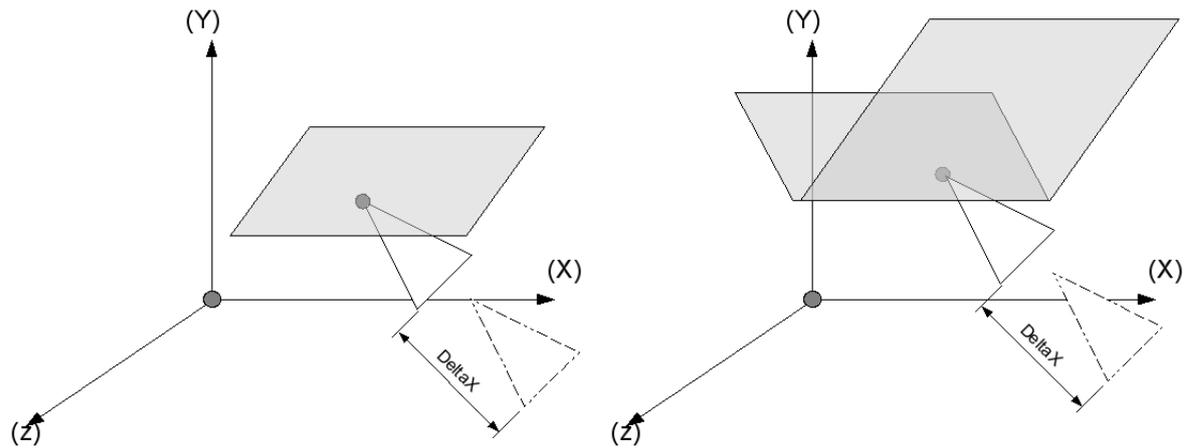


Figura 23: Colisão entre dois Polígonos planares e colisão entre um polígono planar e outro não planar. Triângulo em linhas pontilhadas representa a posição anterior à colisão.

etapa. Na seção seguinte será comentado como esta detecção é realizada, tanto para sólidos estáticos quanto em movimento.

5.1 Detecção de Colisão

O modelo computacional utiliza-se de malhas, poliedros e polígonos para a representação de objetos em um espaço tridimensional. A intersecção entre polígonos planares que não apresentem concavidade é possível de ser identificada com exatidão e relativa facilidade por cálculos fundamentais de geometria analítica. Basta verificar se os planos não são paralelos, garantida esta condição, checa-se os pontos de intersecção entre as arestas de um polígono e o plano representado pelo outro. Esses pontos de contato são checados para saber se estão dentro de ambos os polígonos. Caso o polígono não seja planar, as condições acima não são mais válidas, pois o polígono não pode ser inserido em um único plano. Na situação de o polígono apresentar uma concavidade, este precisa ser dividido em outros polígonos menores que não apresentem concavidade, a fim de tornar possível determinar se o ponto encontrado está contido no seu interior. As malhas tal como os poliedros que representam sólidos são compostos por um conjunto de polígonos. Portanto a detecção é feita conferindo cada um dos polígonos que os constituem. Os polígonos que compõem os objetos são todos planares de modo a facilitar a busca dos pontos de intersecção.

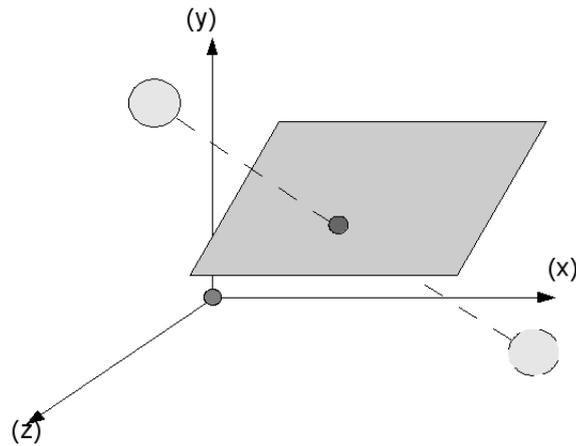


Figura 24: Detecção de colisão durante a trajetória da partícula.

Em sistemas compostos por partículas independentes, a trajetória da partícula é usada como critério de colisão. A intersecção entre um polígono planar e um segmento de reta, formado pela posição atual e a posição anterior da partícula, é calculada. Desta forma é possível encontrar o ponto no qual a partícula interceptou uma região descrita por um outro objeto, como está demonstrado na figura 24.

Em um sistema massa-mola as partículas não são independentes, sua posição não varia apenas em função da força externa aplicada diretamente sobre elas. As partículas são elementos básicos de um conjunto, causando uma influência na posição de seus vizinhos e possivelmente nos sólidos ao quais colidiram. Neste trabalho a forma da malha depende das faces; ou seja, da maneira na qual as partículas e arestas estão organizadas. As partículas são apenas pontos no espaço, não apresentando dimensões de largura, comprimento ou profundidade; sendo praticamente impossível que a posição espacial da partícula de um sólido esteja no interior de uma face de um outro sólido exatamente no instante de tempo no qual a colisão calculada. As partículas que compõem uma malha devem ser analisadas uma a uma, sempre levando em consideração os efeitos ocorridos entre uma iteração e outra e que alterações causam em outros elementos da malha.

5.2 Tratamento de colisão de tecidos com sólidos rígidos imóveis

No instante em que ocorre a colisão entre uma malha e um sólido imóvel, não há a necessidade de calcular a força que o tecido exerce sobre o sólido, pois este não sofrerá nenhuma alteração de posição referente a esta troca de energia. A colisão é calculada pela trajetória da partícula, traçando um segmento de reta com a posição atual e a anterior da partícula. Caso exista alguma colisão entre o segmento de reta e o sólido, significa que o ponto resultante representa as exatas coordenadas onde a partícula deveria ter colidido no intervalo de tempo entre o instante atual e a iteração anterior.

A correção da posição da partícula é feita através de dois procedimentos que visam atualizar sua posição a velocidade. Identificado o exato ponto no qual ocorreu a colisão, as coordenadas da partícula são substituídas pelas do ponto. O vetor velocidade é atualizado através de um procedimento para inverter sua direção tomando como base a normal do polígono.

O vetor velocidade da partícula \vec{v} pode ser decomposto em outros dois. No vetor \vec{v}_n paralelo ao vetor normal do plano e outro vetor \vec{v}_p perpendicular ao mesmo (MÖLLER; HAINES, 2002).

$$\vec{v}_n = \vec{n}(\vec{v} \cdot \vec{n}) \quad (5.1a)$$

$$\vec{v}_p = \vec{v} - \vec{v}_n \quad (5.1b)$$

Para o cálculo do vetor final inverte-se o valor de \vec{v}_n e soma-o novamente a \vec{v}_p :

$$\vec{v}_f = \vec{v}_p - \vec{v}_n \quad (5.1c)$$

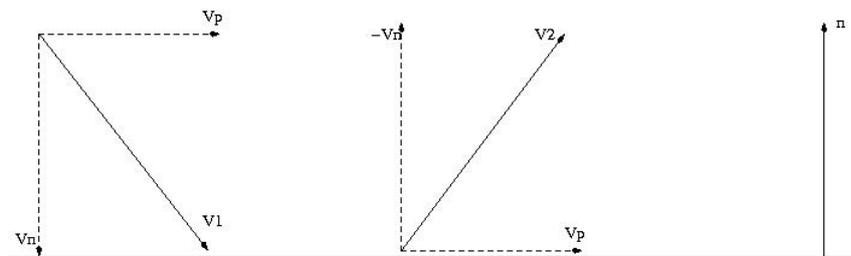


Figura 25: vetor \vec{v}_2 resultante da reflexão do vetor \vec{v}_1 no plano com vetor normal \vec{n}

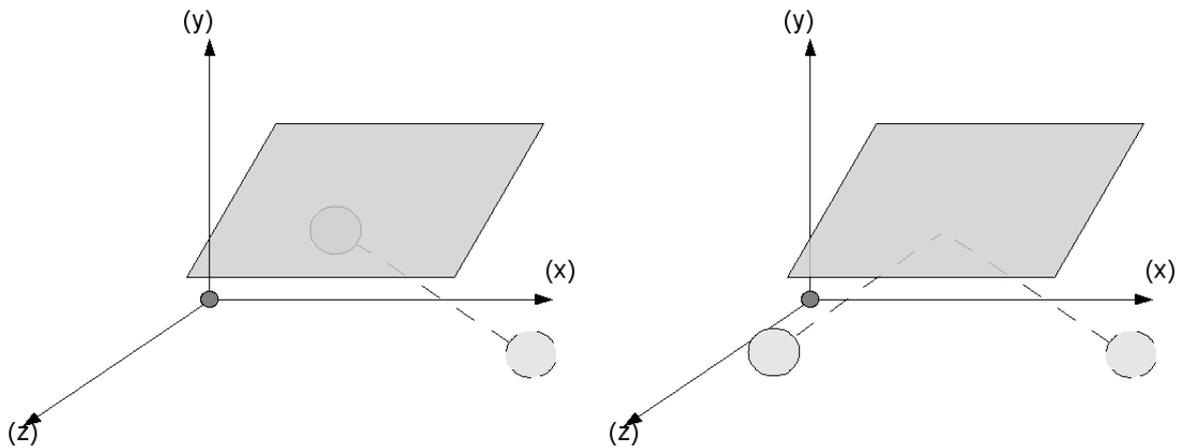


Figura 26: Posição da partícula é corrigida (esquerda). Após correção da velocidade, a posição da partícula é novamente atualizada (direita).

Como foi utilizado o algoritmo de Velocity Verlet; onde há o cálculo prévio da velocidade para a atualização da posição da partícula, já existe o conhecimento desta grandeza, sendo então refletida para a direção oposta ao polígono. A figura 26 ilustra as etapas de detecção de colisão e correção da posição da partícula.

O capítulo 6 contém dois experimentos realizados referentes a colisão de malhas com sólidos rígidos imóveis.

Em relação a colisão, nos experimentos realizados, pôde-se observar que algumas molas atravessam a superfície. Isto ocorre por que esta detecção de colisão é orientada à partícula e não à mola. Uma solução possível para este problema será descrita ao final do capítulo, subdividindo as faces no exato ponto da colisão.

5.3 Tratamento de colisão de tecidos com sólidos rígidos

Diferentemente de um sólido rígido imóvel, um sólido rígido ao entrar em contato com um tecido troca energia com ele de tal modo que possibilita a ocorrência de alterações na morfologia do tecido e na localização espacial de ambos. Esta troca de energia é resultante das características física que apresentavam no instante da colisão. Outro fator relevante é identificar de que forma a colisão se deu, quantos pontos estão em contato e em qual região de qual face os pontos de contato estão posicionados.

Por exemplo uma bola e uma lança com mesma massa são arremessadas contra uma malha em ocasiões diferentes. A área de contato da bola com a malha é bem maior do que a da lança. Isso significa que força que a bola aplica pressionando a malha no instante da colisão será dividida entre os pontos de contato existentes, seguindo a equação:

$$P = \frac{F}{A} \text{ onde: } P=\text{pressão e } A=\text{área}$$

Portanto para saber qual força deve aplicada sobre a malha deve-se ter conhecido de todas as regiões de contato, distribuindo a força de que ocasionou a colisão entre todos os pontos. Voltando ao exemplo da bola e da lança; a bola tendo mais pontos de contato que a lança causa uma deformação com uma maior distribuição de forças, enquanto o contato da lança tende a ser localizado, mesmo sendo aplicada uma força de igual valores em ambas no instante da colisão como pode ser visto na figura 27.

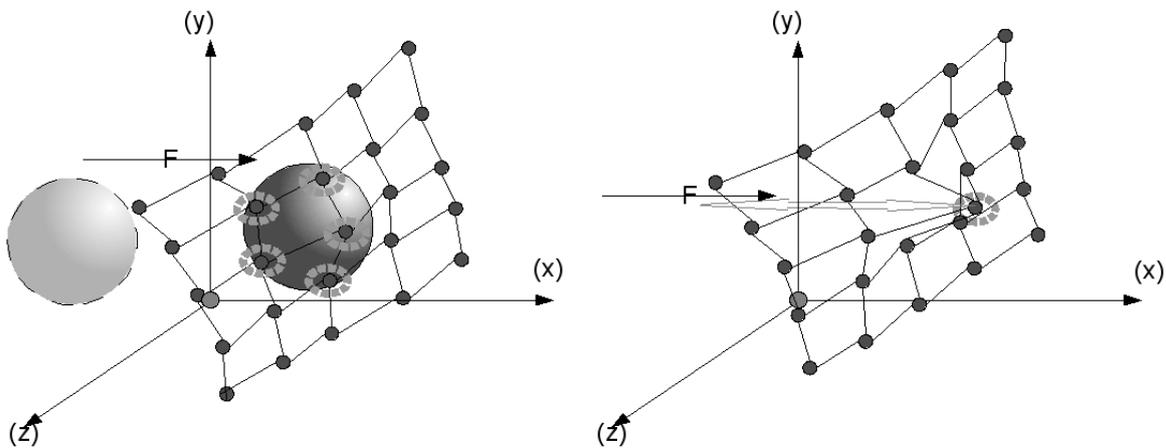


Figura 27: Colisão entre uma bola e uma malha à esquerda, colisão entre malha e lança à direita. Pontos de contato em destaque.

Aplicada a força em cada ponto de contato, deve-se calcular a força de reação que as molas geraram em cada um destes pontos, somadas em um único vetor força e aplicada no sólido que ocasionou a deformação. Caso o vetor desta força tenha a intensidade menor do que o vetor força que movimenta a partícula, o sólido terá seu movimento desacelerado; caso tenha intensidade maior, o sólido será repellido. O sólido é tratado como se fosse uma partícula única,

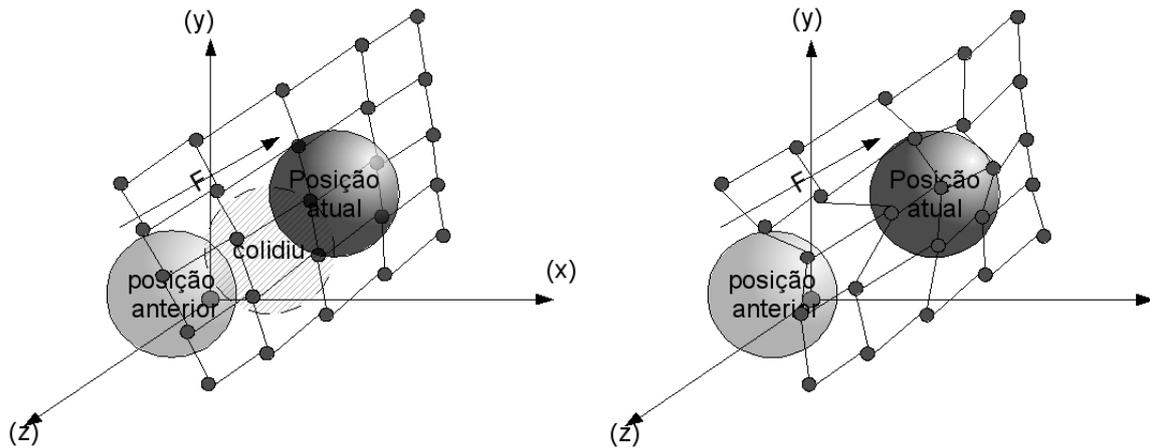


Figura 28: Malha interceptada entre uma iteração e outra (esq). Posição da malha corrigida (dir).

tendo sua posição atualizada da mesma forma que a partícula original da malha; obviamente sem a ação de molas conectadas.

Nestes exemplos acima admitem-se casos ótimos, os pontos de contato são as próprias partículas. É praticamente impossível que isto ocorra em uma simulação em tempo real a todo instante. Neste caso existem duas possibilidades, a primeira é adicionar uma partícula extra a estrutura, exatamente no ponto em que o contato ocorreu, outra alternativa é propagar a força por todas as partículas da face a qual ocorre a colisão. Por hora, será discutida apenas a segunda opção. Neste tipo de colisão além da colisão entre aresta e faces da malha e do sólido, também se utiliza a trajetória como critério, pois a malha pode ser atravessada entre uma iteração e outra. A malha precisa ser adaptada à posição do sólido, inclusive podendo extrapolar seu valor máximo de comprimento, o que ocasionaria seu rompimento. Este trabalho vislumbra esta possibilidade na seção final do presente capítulo, porém apenas levanta os problemas envolvidos nesta operação, não demonstrando como isto pode ser tratado.

A força que o sólido aplica na malha é calculada em função do deslocamento, proporcionado por uma colisão em um instante de tempo em cada face. Toda vez que o sólido atinge uma face da malha, ele a desloca. Naturalmente deve ocorrer uma atualização da posição da face, entretanto isso não pode ser feito de maneira instantânea, somente transladando as partículas para a posição desejada ignorando as alterações na velocidade e aceleração. Isto acabaria com

a toda a suavidade obtida na aplicação dos métodos de integração, tornando o sistema praticamente imprevisível pois os valores referentes a velocidade e a aceleração não correspondem ao movimento executado. A resolução deste problema consiste em encontrar o valor da força necessária para movimentar as partículas da face à posição desejada.

Em um choque elástico a energia cinética deve ser conservada. Energia cinética é a quantidade de trabalho que teve que ser realizado sobre um objeto de massa u para tirá-lo do repouso e colocá-lo a uma velocidade v , sendo definida por $E_c = \frac{u \cdot v^2}{2}$. No choque entre uma partícula em repouso pY e outra pZ em movimento a energia é calculada da seguinte forma segundo o princípio de conservação do momento linear:

$$pZ_u \cdot pZ_{velocidadeInicial} = pZ_u \cdot pZ_{velocidadeFinal} + pY_u \cdot pY_{velocidadeFinal} \quad (5.2a)$$

A velocidade de pA é calculada em função da energia que foi transmitida a ela pelo choque, sendo a velocidade final de pB calculada em função da energia que ela manteve.

Para calcular a energia transmitida à n partículas por um sólido B utiliza-se a fórmula abaixo:

$$B_u \cdot B_{v_{inicial}} = B_u \cdot B_{v_{final}} + \sum_{i=1}^n (p[i]_u \cdot p[i]_v) \quad (5.2b)$$

Foram necessários alguns ajustes para aplicação desta idéia no modelo. Essa adaptação foi realizada de maneira experimental, pois não considera o momento angular, não havendo como garantir sua validação. Como foi dito anteriormente a colisão se dá geralmente entre instantes, intervalos de entre iterações. Na figura 28 a bola sofre um deslocamento entre dois instantes de tempo, caso aplicada nas partículas a mesma quantidade de energia elas sofrerão a mesma variação de posição; quando o esperado é que atinjam apenas o deslocamento necessário para manterem-se em contato com a bola. Para tal deve-se calcular a força necessária para deslocar a partícula até a posição desejada. Partindo do princípio que:

$$\frac{B_u \cdot B_{v_i^2}}{2} = \frac{B_u \cdot B_{v_f^2}}{2} + \sum_{i=1}^n \left(\frac{p[i]_u \cdot p[i]_v^2}{2} \right) \quad (5.2c)$$

$$B_{E_{c_i}} = B_{E_{c_f}} + \sum_{i=1}^n (p[i]_{E_c}) \quad (5.2d)$$

$$B_{E_{c_i}} - B_{E_{c_f}} = \sum_{i=1}^n (p[i]_{E_c}) \quad (5.2e)$$

$$-B_{\Delta Ec} = \sum_{i=1}^n (p[i]_{Ec}) \quad (5.2f)$$

A variação de energia cinética foi o trabalho W realizado sobre o sólido em Δt , ou seja $W = \Delta Ec$. Para encontrar o trabalho aplicado em cada uma das n partículas colididas faz-se:

$$p_W = \frac{B_W}{n} \quad (5.2g)$$

Trabalho também pode ser definido em função da força aplicada e pelo deslocamento efetuado:

$$W = \vec{F} \cdot \Delta X \quad (5.2h)$$

Esta fórmula é utilizada para calcular a força a ser aplicada na partícula para que ela atinja a posição desejada, ou seja aquela a qual a colisão com o sólido a teria deslocado:

$$p_{\vec{F}} = \frac{p_W}{p_{X(t+\Delta T)} - p_{X(t)}} \quad (5.2i)$$

Em posse do valor da força, atualiza-se a aceleração e velocidade da partícula. Após aplicar este procedimento a cada partícula da malha afetada, todas as posições da malha são atualizadas pelo algoritmo 9; desta forma toda energia é dissipada pela malha. Durante a execução deste algoritmo os valores de velocidade de cada partícula afetada serão novamente atualizados, desta vez em função das forças de reação das molas da malha. Estas velocidades são somadas e aplicando a equação 5.2b descobre-se o velocidade final do sólido. Este modelo não inclui o movimento angular nos cálculos, desconsiderando sua relevância.

O capítulo 6 contém experimentos relacionados à colisão de malhas e sólidos rígidos.

5.4 Tratamento de Colisões: Subdivisão da malha

Nesta seção será discutida uma alternativa às técnicas que foram vistas até agora para o tratamento de colisão. Entretanto a operação aqui sugerida foi implementada apenas com o intuito de manter as características topológicas 2-manifold, as quais o algoritmo de deformação é dependente.

Na colisão do tecido com um sólido imóvel, checa-se a trajetória de cada um dos vértices

da malha. No caso de um sólido em movimento checa-se a trajetória dos vértices do sólido. Na primeira situação o vértice da malha é atualizado em função de sua velocidade, na segunda situação em função da troca de energia do contato. Em ambos os casos os objetos não são tratados como contínuos e sim como se fossem compostos por elementos geométricos menores.

Uma flexibilidade maior no tratamento dos contatos garantiria uma visualização mais realista do ponto de vista físico. Seria interessante que o ponto exato de contato sofresse a deformação nesse instante, ao invés das partículas que estão próximas ou daquelas que compõem a face onde o ponto se localiza. Para realizar esta operação é necessário subdividir a malha em tempo real, necessariamente que esta mudança garanta características físicas como equilíbrio estático e manutenção da massa e energia potencial.

Fazendo uso de operadores da DLFL foi possível inserir partículas novas na malha. Aplicando esta técnica exclusivamente em triângulos é possível manter malhas constituídas exclusivamente por polígonos planares e manter as características 2-manifold, como mostra a figura 29. Porém a resolução deste problema traz uma série de novos questionamentos. Partículas são pontos no espaço que possuem massa e em conjunto com outras descrevem uma face. A massa que essa nova partícula deveria ter para manter a massa do sistema é uma das incógnitas, que leva a outros problemas; como a definição do coeficiente elástico das molas que conectariam esta partícula. Isto tudo acarretaria num problema ainda maior, como a alteração que outras partículas e molas do sistema inteiro deveriam sofrer para garantir que esses novos coeficientes representassem uma malha com propriedades iguais às aquelas anteriores ao choque.

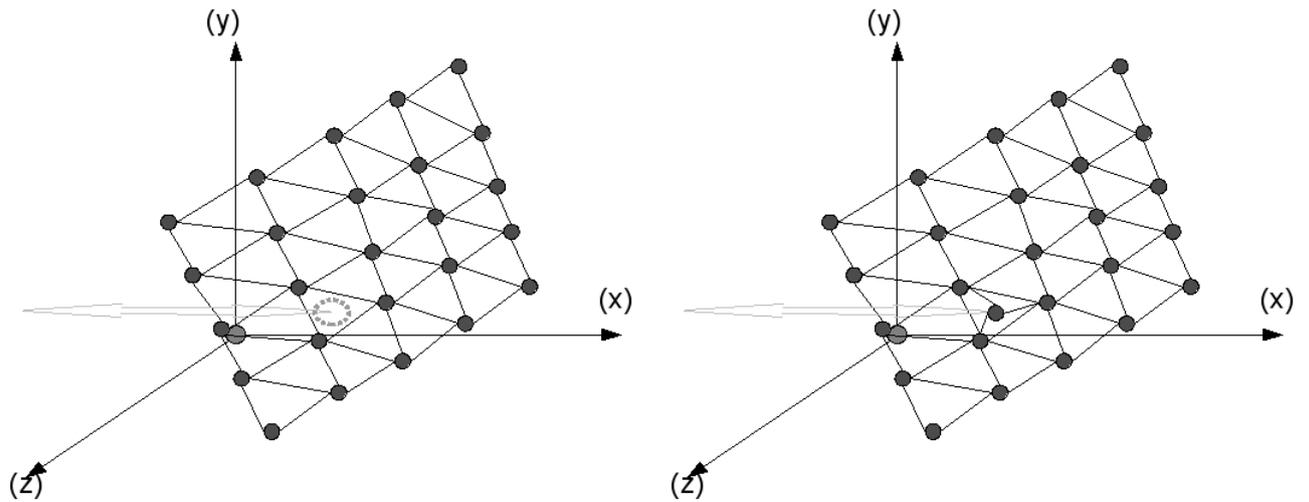


Figura 29: Subdivisão da malha durante uma colisão. Dividiu a face em três com base no ponto de contato.

Caso a colisão do objeto com a malha cause uma deformação muito pequena, o contato de ambos no instante seguinte pode ser em um ponto muito próximo, isto pode acabar gerando subdivisões com faces muito pequenas. O problema desta situação é que seriam inseridas molas de comprimento reduzido ocasionando um rompimento prematuro de uma aresta. Para evitar tal problema pode-se adotar um coeficiente de distância mínima; um novo vértice só será inserido caso ele se localize a uma distância euclidiana maior de todos os vértices pertencentes a face que seria subdividida.

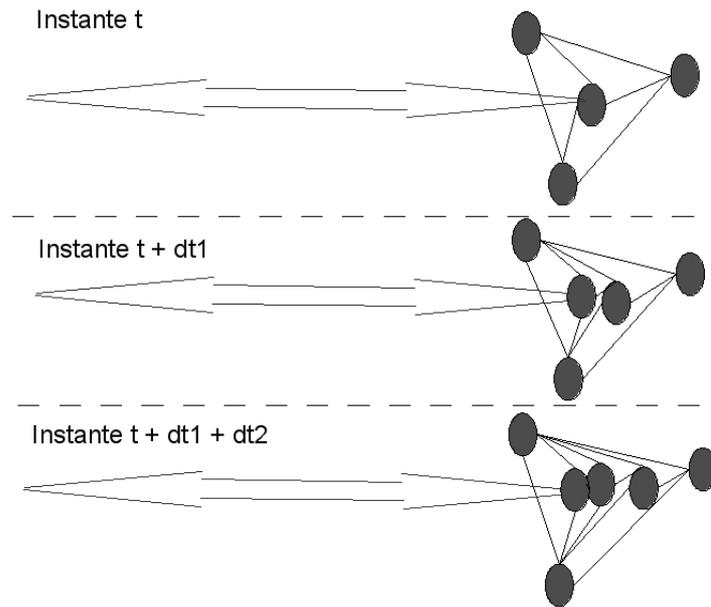


Figura 30: Consecutivas subdivisões de faces.

Ignorando os efeitos colaterais que a inserção de uma nova partícula causaria nas propriedades físicas da malha, a seguir está o algoritmo para a inserção de uma partícula em uma face planar respeitando as características topológicas da malha.

Algoritmo 10 Algoritmo para inserção de uma partícula em um ponto de contato de uma malha.

```

inserePonto(Face facePlanar, Ponto pontoColisao, float
minDistancia, DLFL dlfl)

/*nesta etapa supõe-se que a face é planar e que o ponto está em contido em seu interior*/

    Para vértice de facePlanar faça
        se( $distanciaEuclidea(vertice[i], pontoColisao) < minDistancia$ )
            fim
        listaExtremos  $\leftarrow$  facePlanar.listaExtremos

/*guarda-se uma lista com ponteiros para os extremos da face
pois durante a inserção de vértices entre extremos a face irá sofrer alterações para manter a
característica topológica 2-manifold*/

    extremo  $\leftarrow$  dlfl.insereVertice(ponto)

/*este método insere um vértice na dlfl e retorna o extremo da face para qual ele está referen-
ciado. Como ele não tem nenhuma aresta conectada a ele, pertence a uma face com um único
vértice.*/

    Para extremo de listaExtremos faça
        dlfl.insereAresta(extremo, listaExtremos[i])

/*método conecta os extremos dos vértices referenciados, criando uma nova face a cada ciclo*/

    fim

```

Neste ponto pode-se afirmar que em todas as etapas de trabalho as características matemáticas e físicas devem ser sempre avaliadas em conjunto, analisando os efeitos que essa combinação gera. Esse trabalho definiu diversos pontos, cujo desenvolvimento levou a outros questionamentos. No próximo capítulo serão demonstrados os resultados obtidos em experimentos realizados ao longo de todas as etapas. No capítulo subsequente será feita uma análise geral e discutido um roteiro para que os passos seguintes possam ser tomados com o que foi pesquisado até então.

6 *Resultados*

6.1 Experimentos: seção 4.2

Os experimentos desta seção visam avaliar a estabilidade dos algoritmos de integração especificados na seção 4.2 na página 53. Os algoritmos de integração serão avaliados sobre condições que podem ocorrer em uma simulação em tempo real.

6.1.1 Experimento 1

Uma partícula em um espaço unidimensional localizada na coordenada 0, com velocidade inicial de $61,29 \frac{m}{s}$ sofre uma aceleração de $-9,81 \frac{m}{s^2}$ constante com $\Delta t = 0,25s$. Para cada método foram calculadas as 25 primeiras iterações.

- X_verlet - Positional Verlet
- TCV - Time Corrected Verlet (Verlet com correção de Tempo)
- X_VVerlet - Velocity Verlet

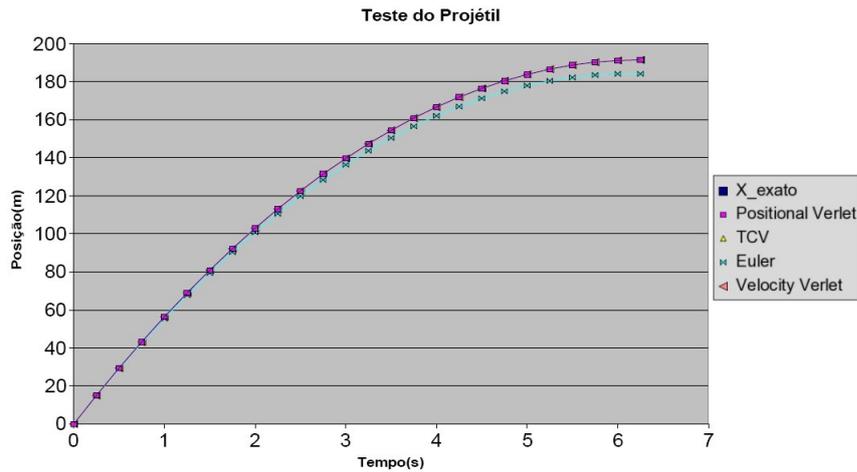


Figura 31: Gráfico Experimento 1

Com exceção do método de euler para cálculo de deslocamento os demais apresentaram valores exatos. Abaixo segue a tabela de desvio padrão:

Método	Desvio Padrão Posição(m)
Positional verlet	0
Verlet com Correção de Tempo	0
Euler	2,32
Velocity Verlet	0

Tabela 2: Comparação entre métodos.

Após este teste foi realizado outro com os mesmos valores, porém com intervalos de tempo variáveis e definidos de maneira aleatória:

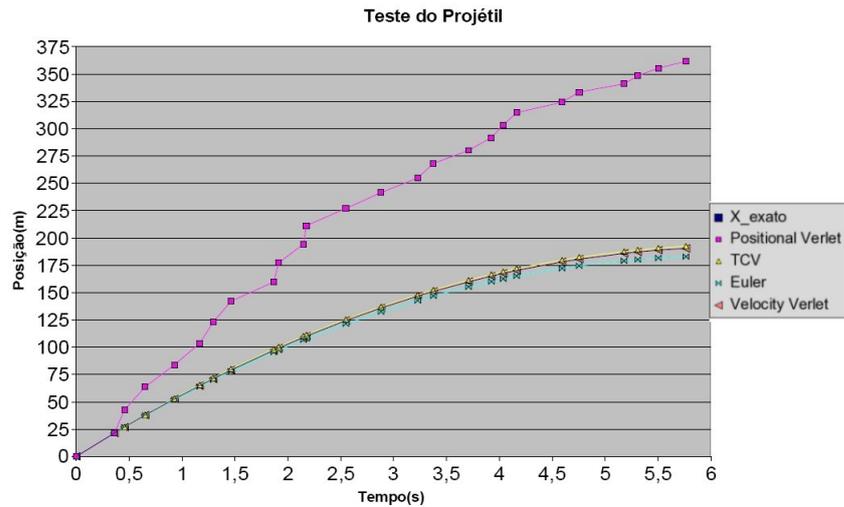


Figura 32: Gráfico Experimento 1

Método	Desvio Padrão Posição(m)
Positional Verlet	53,74
Verlet com Correção de Tempo	0,58
Euler	2,51
Velocity Verlet	0

Tabela 3: Comparação entre métodos.

Como dito anteriormente, o método Positional Verlet apresenta fragilidades quando o intervalo de tempo não é constante. Os valores apresentados por ele foram muito discrepantes, tornando impossível sua utilização em uma simulação que admita tempo variável.

6.1.2 Experimento 2

Neste experimento a seguinte função polinomial discretiza a posição de uma partícula em um espaço unidimensional em função do tempo:

$$x(t) = 2 \cdot \Delta t^3 - 7 \cdot \Delta t^2 + 5\Delta t$$

$$\text{como } v(t) = x(t)\Delta t \equiv 6 \cdot \Delta t^2 - 14 \cdot \Delta t + 5 \text{ e } a(t) = v(t)\Delta t \equiv 12 \cdot \Delta t - 14$$

A partícula inicia na coordenada espacial 0 com velocidade inicial de $5 \frac{m}{s}$. Os intervalos de

tempo são variáveis e definidos de maneira aleatória.

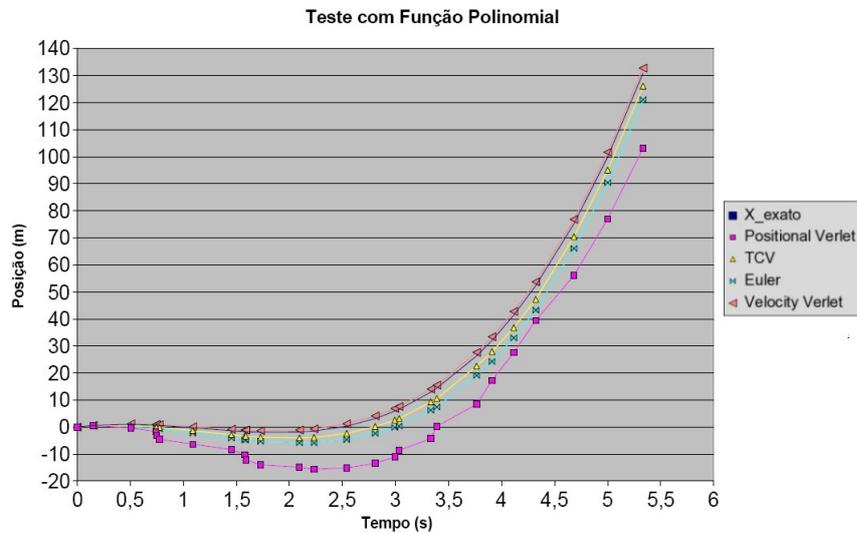


Figura 33: Gráfico Experimento 2

Método	Desvio Padrão Posição(m)
Positional verlet	7,09
Verlet com correção de tempo	1,65
Euler	3,13
Velocity Verlet	0,53

Tabela 4: Comparação entre métodos.

Novamente o método de Velocity Verlet apresentou melhores resultados, seguido pelo Verlet com correção de tempo.

6.1.3 Experimento 3

Neste experimento o deslocamento de uma partícula em função do tempo foi calculado por uma função senoidal, sendo:

$$x(t) = -\text{sen}(t); v(t) = -\text{cos}(t); a(t) = \text{sen}(t)$$

A partícula tem como posição inicial a coordenada 0 e velocidade inicial de $-1 \frac{m}{s}$. Os intervalos de tempo são variáveis e definidos de maneira aleatória.

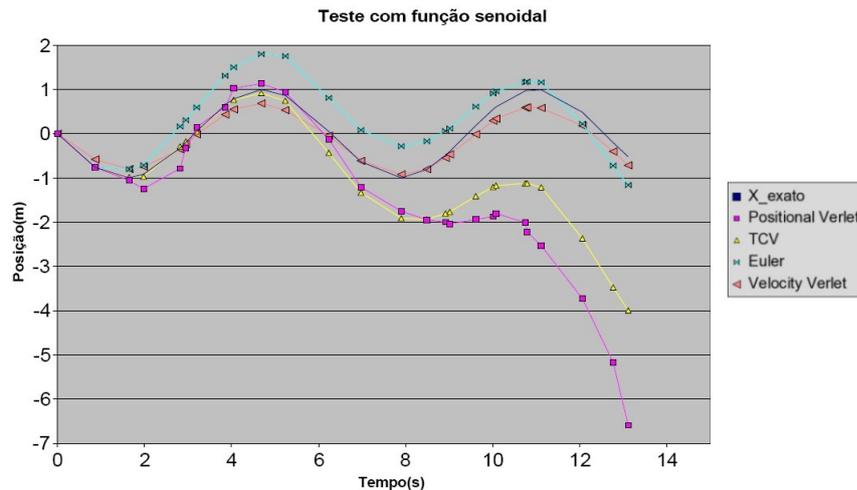


Figura 34: Gráfico Experimento 3

Método	Desvio Padrão Posição(m)
Positional Verlet	1,74
Verlet com Correção de Tempo	1,11
Euler	0,25
Velocity Verlet	0,13

Tabela 5: Comparação entre métodos.

Neste experimento 3 a trajetória apresentada pelo Velocity Verlet não chega a ser tão próxima a da posição correta, entretanto deve-se levar em conta que além do tempo a aceleração possui seu sentido invertido ao longo do tempo. Contudo, analisando os resultados pode-se constatar que este método apresentou o menor desvio padrão entre os demais, assim como em todos os experimentos anteriores. Como já foi dito a adoção de um método numérico que realize uma integração com maior exatidão acarretaria em um maior custo computacional, prejudicando a execução de uma simulação em tempo real.

6.2 Experimentos: seção 4.3

Os experimentos desta seção visam analisar se o modelo computacional demonstrado na 4.3 na página 57 para representação de deformações é capaz de restaurar o estado inicial de uma

malha na ausência de forças externas, após a malha sofrer uma deformação. Todos estes experimentos, bem como os das seções 6.3 e 6.4, foram realizados em um PC com processador AMD Athlon-64 bits - 2Ghz de frequência, memória RAM de 1 GB e placa de vídeo GeForce MX 4400 com 64MB. Todo o código foi feito em linguagem C++, o código binário foi gerado através do compilador GCC 3.4.6.

6.2.1 Experimento 1

Neste experimento uma malha de 21x21 partículas (4,5m x 4,5m) sofre uma ação de um vetor força externo ao sistema de (0 0 0,15)N por iteração na sua partícula localizada na região central. Na iteração 5000 a força externa deixa de ser aplicada, e o sistema tende a retornar a uma posição de equilíbrio, ou seja, até que a forças de reação das molas assumam valores irrelevantes. Não há a aplicação de forças relativas à gravidade neste sistema. As partículas nos extremos da malha não são sensíveis a movimentos, ou seja são imóveis. O tempo de cada iteração varia no intervalo de $\frac{1}{60}s$ a $\frac{1}{110}s$ As constantes do sistema são:

$$\mu = 10g; \varepsilon = 0,001 \cdot l_0; l_{min} = 0,1 \cdot l_0; l_{max} = 10 \cdot l_0; ke = 4N/m; kd = -0,8N/m/s$$

A seguir imagens capturadas de uma simulação em tempo real, os índices da iterações são aproximados em ± 100 iterações. Todos os experimentos contidos neste trabalho foram tratados com valores passos de tempo atualizados durante a execução, ou sejam o passo do tempo é variável.

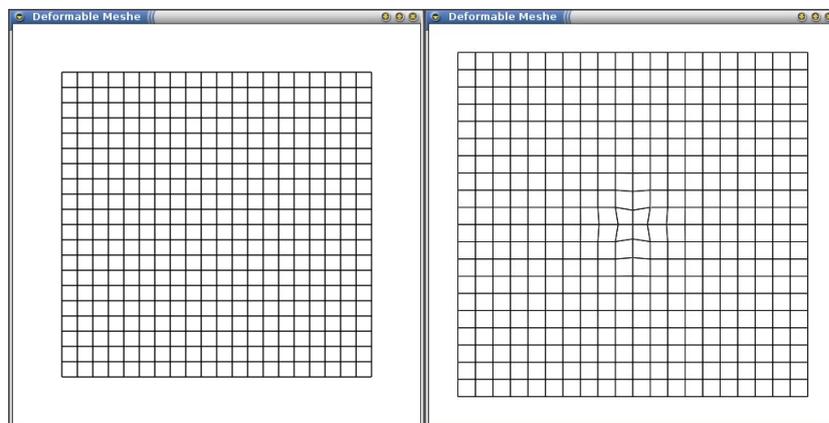


Figura 35: Atualizações da malha ao longo do tempo. Iteração 0 e iteração 6000.

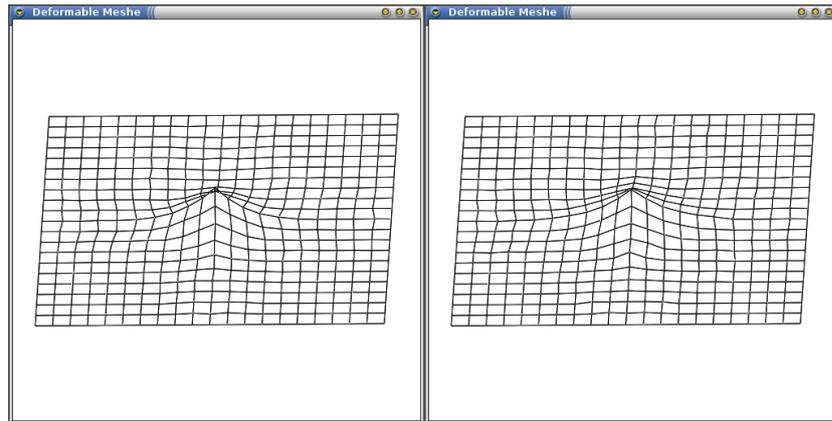


Figura 36: Atualizações da malha ao longo do tempo. Iteração 4600 e iteração 5000.

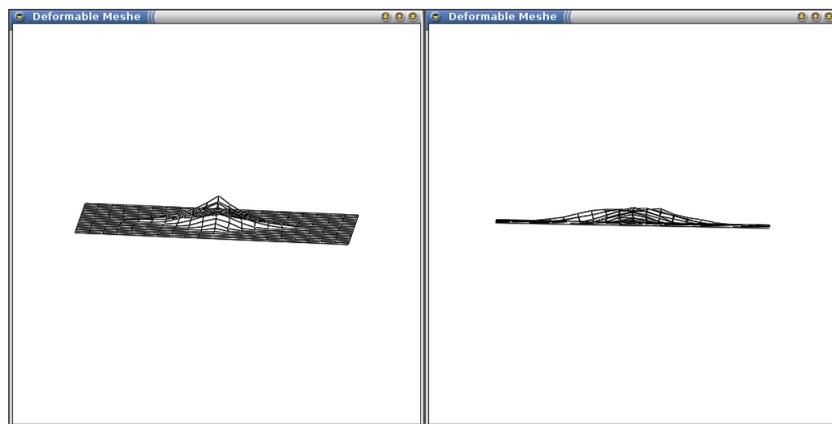


Figura 37: Atualizações da malha ao longo do tempo. Iteração 5200 e iteração 7000.

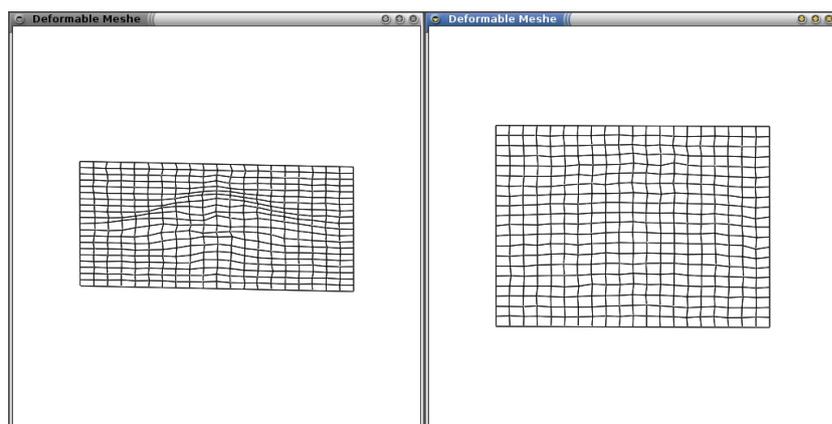


Figura 38: Atualizações da malha ao longo do tempo. Iteração 9000 e iteração 20000.

Após a iteração 20000 não foram observadas alterações significativas nas posições das partículas, isto ocorre devido as forças de reação da mola não serem mais capazes de gerar um

desequilíbrio no sistema. Para cada coordenada de cada partícula foi calculada a diferença entre o valor original e na posição na iteração 5000 e 20000, bem como a distância euclidiana em R^3 e o desvio padrão destas distâncias. A tabela abaixo apresenta alguns dados obtidos.

Dados	Iteração 5000	Iteração 20000
Maior diferença coord. X	0,10890	0,02867
Maior diferença coord. Y	0,11154	0,03484
Maior diferença coord. Z	0,45550	0,09833
Maior distância euclideana	0,45559	0,09834
Desvio Padrão (distâncias)	0,05044	0,02400

Tabela 6: Dados experimento 1, valores em metros.

6.2.2 Experimento 2

Neste experimento foram mantidas as mesmas características no experimento 1, adotando as seguintes constantes:

$$\mu = 5g; \varepsilon = 0,001 \cdot l_0; l_{min} = 0,1 \cdot l_0; l_{max} = 10 \cdot l_0; ke = 8N/m; kd = -1,6N/m/s$$

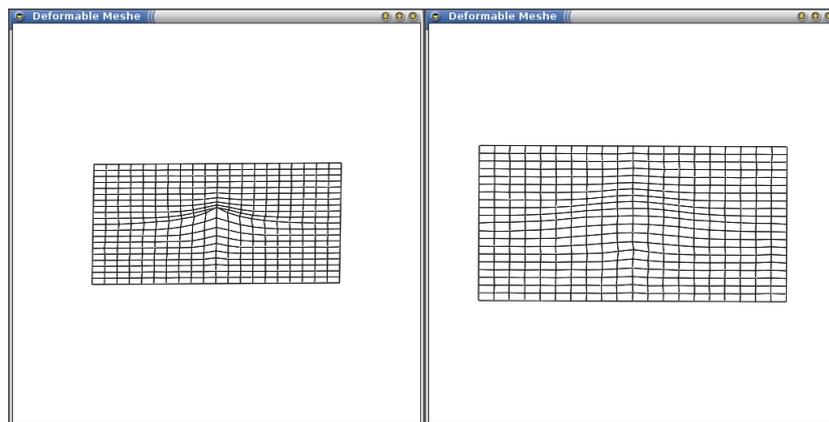


Figura 39: Atualizações da malha ao longo do tempo. Iteração 5000 e iteração 10000.

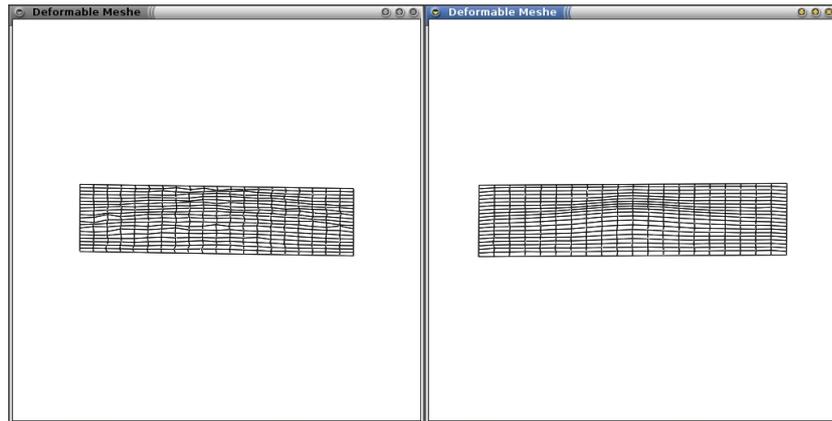


Figura 40: Experimento 1 Iteração 20000 e Experimento 2 Iteração 20000.

Para o experimento 2 foram coletados os mesmos dados do experimento 1, segue tabela.

Dados	Iteração 5000	Iteração 20000
Maior diferença coord. X	0,04965	0,01509
Maior diferença coord. Y	0,05194	0,01531
Maior diferença coord. Z	0,35186	0,10253
Maior distância euclideana	0,35226	0,10269
Desvio Padrão (distâncias)	0,03566	0,02267

Tabela 7: Dados experimento 2, valores em metros.

Com os coeficientes adotados neste experimento, a força externa propagou-se por menos molas do que no experimento 1, ocasionando menos alterações nas posições das partículas. Isto deve-se ao sistema apresentar vetores de reação com intensidades maiores do que no experimento 1. Na figura 40, pode-se observar a formação de picos e vales espalhados pela superfície na figura do experimento 1, isto ocorreu porque a força externa chegou a ser propagada às partículas fixas da extremidades, gerando pequenas ondas de propagação de forças que por sua vez geraram uma oscilação na malha.

Observando estes dois experimentos pode-se afirmar o sistema sempre tende a voltar a seu estado original, porém dependendo dos coeficientes estabelecidos, pode ser que as partículas não atinjam exatamente a posição original. Além da disposição da molas e das partículas na

malha, os coeficientes de ambas devem ser ajustados para a obtenção de um comportamento elástico desejado.

6.2.3 Experimento 3

Este experimento visa demonstrar as possibilidades do sistema em lidar com a aplicação de diferentes forças. Para tal foram tomados os mesmos coeficientes do experimento 1, porém com a aplicação de duas forças externas ao sistema, uma de $(0\ 0\ 0,15)N$ e outra de $(0\ 0\ -0,15)N$.

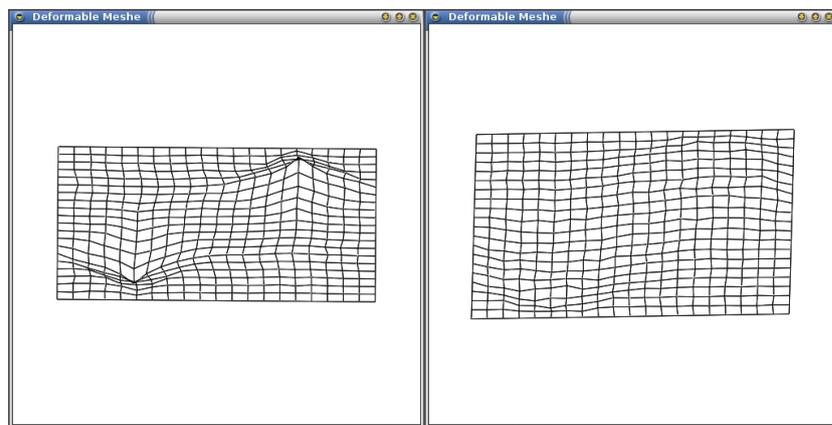


Figura 41: Experimento 3, Iterações 5000 e 10000.

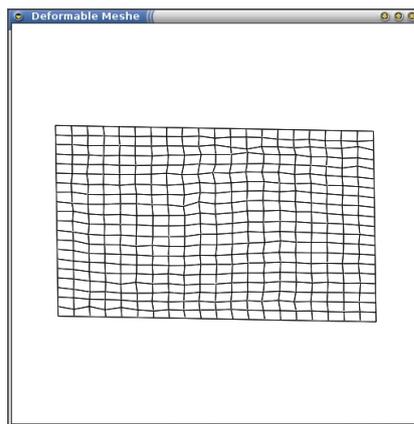


Figura 42: Experimento 3, Iteração 20000.

Novamente aplicou-se a mesma metodologia com os dados da simulação, segue adiante a tabela de valores:

Dados	Iteração 5000	Iteração 20000
Maior diferença coord. X	0,11200	0,03174
Maior diferença coord. Y	0,10257	0,03482
Maior diferença coord. Z	0,45104	0,09319
Maior distância euclideana	0,45546	0,09348
Desvio Padrão (distâncias)	0,06994	0,02226

Tabela 8: Dados experimento 3, valores em metros.

6.2.4 Experimento 4

Neste experimento ao invés de aplicar uma força de $(0\ 0\ 0,15)N$ por 5000 iterações, aplicou-se uma força de $(0\ 0\ 250)N$ por três iterações, da 4897 a 4900. Os demais valores aplicados foram os mesmos do experimento 1.

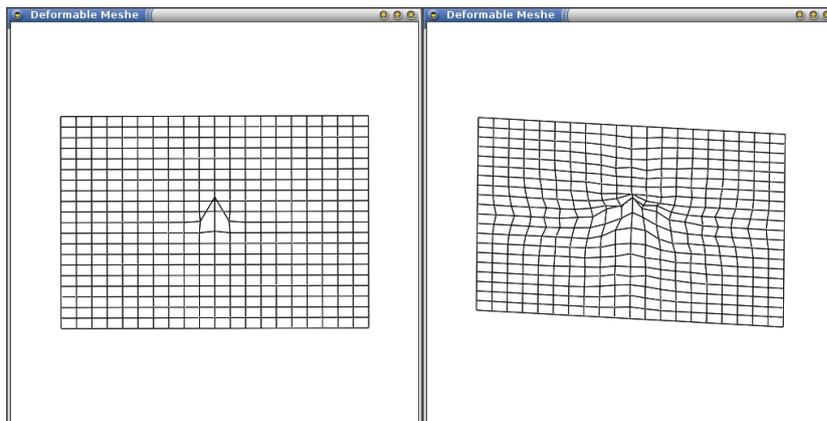


Figura 43: Experimento 4, Iterações 5000 e 7000.

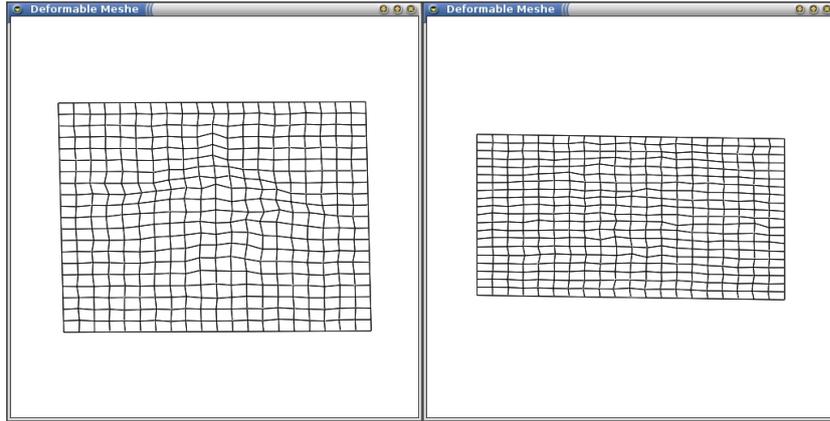


Figura 44: Experimento 4, Iterações 10000 e 20000.

Dados	Iteração 5000	Iteração 20000
Maior diferença coord. X	0,01473	0,02536
Maior diferença coord. Y	0,01465	0,02655
Maior diferença coord. Z	0,64872	0,08203
Maior distância euclideana	0,64873	0,08462
Desvio Padrão (distâncias)	0,03096	0,01996

Tabela 9: Dados experimento 4, valores em metros.

Este experimento apresentou valores menores para a maior distância e para o desvio padrão. Isto deve-se exatamente a força ter propagado-se a um maior número de molas, resultando em um maior número de forças em ação para a manutenção do equilíbrio.

6.3 Experimentos: seção 5.2

Os experimentos desta seção visam demonstrar como ocorre a colisão entre malhas e sólidos rígidos imóveis, operação esta especificada na seção 5.2 na página 68.

6.3.1 Experimento 1

Neste experimento foi utilizada uma malha com 20x20 partículas de 4,25mX4,25m. O intuito é verificar a validade da checagem de colisão. Para tal a malha foi suspensa a altura de 1m e cada partícula sofreu uma força com o intuito de serem pressionadas contra uma superfície. Foram feitos três testes, o primeiro com este vetor com os valores de (0 -10 0)N, posteriormente com (0 -100 0)N e finalmente com (0 -200 0)N. As constantes do sistema são:

$$\mu = 10g; \varepsilon = 0,001 \cdot l_0; l_{min} = 0,1 \cdot l_0; l_{max} = 10 \cdot l_0; k_e = 1000N/m; kd = -200N/m/s$$

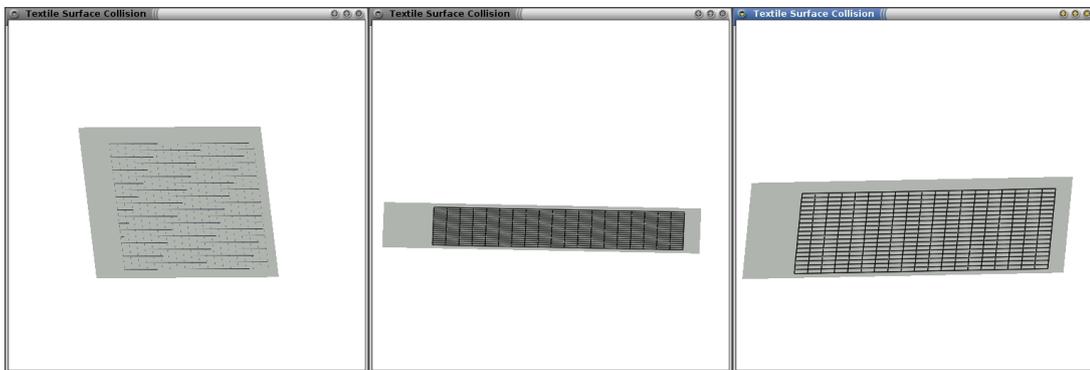


Figura 45: Imagens do três testes para verificação de colisão. Vetores (0 -10 0)N, (0 -100 0)N e (0 -200 0)N da esquerda para a direita respectivamente.

No primeiro teste a malha ficou sobre a superfície e em nenhum momento a atravessou. No segundo teste a malha chegou a atravessar a superfície, oscilando ao lado contrário ao qual foi pressionada, porém não chegou a se distanciar desta. O mesmo foi verificado no terceiro teste. Isto deve-se ao fato de que a reflexão da velocidade e o ajuste de posição são capazes corrigir a trajetória da partícula, porém quando o deslocamento é calculado em função da aceleração exercida em função de uma força externa à partícula, esta retorna ao lado oposto. A velocidade de reflexão e a correção espaciais são suficientes para garantir que a partícula tenha a posição corrigida, mesmo em caso que o deslocamento gerado pela aceleração atual seja suficientemente relevante a ponto de fazer com que a superfície seja traspassada novamente

6.3.2 Experimento 2

Utilizou-se a mesma malha do experimento 1, porém a superfície teve suas dimensões reduzidas, a fim de observar a deformação gerada na malha. Como cada partícula tem 10g de massa, para simular a ação da gravidade foi utilizado um vetor força em cada uma de (0 -98 0)N, causando uma aceleração constante em cada uma de $9,8m/s^2$, sendo modificada apenas quando as molas exercem uma força de reação.



Figura 46: Deformações sofridas pela malha, decorrentes da ação da gravidade e colisão com a superfície.

Os extremos da malha sofrem a ação da gravidade e seguem sua direção. No momento em que a malha entra em contato com a superfície; algumas partículas mesmo sofrendo a ação da gravidade, deixam de atualizar sua posição unicamente em função da força exercida sobre elas, mas também pelo colisão. As partículas que não colidem continuam sofrendo apenas ação da gravidade e de forças internas, esticando as molas a elas conectadas como efeito colateral. Chega-se a um instante de tempo em que a ação da gravidade é totalmente anulada pela força de reação das molas, não sendo observadas alterações morfológicas relevantes na malha.

6.4 Experimentos: seção 5.3

A seguir serão demonstrados alguns experimentos referentes a colisão de uma malha formada por triângulos menores e um cubo. Esses experimentos foram realizados em simulações

mantendo o tempo de atualização em 0,01s. O sistema foi dimensionado para aceitar taxas de tempo variáveis, esse valor foi fixado para maior facilidade de obtenção de dados para os experimentos; pois com tempo conhecido, uma simulação pode ser repetida várias vezes obtendo-se os mesmos valores.

6.4.1 Experimento 1

Um sólido no formato de um cubo de 0,5m de lado, com o centro localizado em (-0,75 0,625 -0,75)m, foi arremessado por uma força de (0 -25 0)N contra uma malha de 4,5x4,5m com o centro localizado em (-0,75 -1,25 -0,75)m formada por 85 partículas que compõem faces triangulares. Os coeficientes da malha são os seguintes:

$$\mu = 10g; \varepsilon = 0,001 \cdot l_0; l_{min} = 0,1 \cdot l_0; l_{max} = 10 \cdot l_0; k_e = 100N/m; k_v = -20N/m/s$$

Os coeficientes do sólido são os seguintes:

$$\mu = 50g; k_d = -0,05N/m/s$$

A fórmula para o cálculo de deslocamento do sólido é o Velocity Verlet, neste caso o sólido é uma partícula isolada e o k_d funciona como uma resistência do meio que ele se encontra a sua velocidade atual; por exemplo a resistência do ar ou água.

Com o impacto a malha se deformou e o objeto foi arremessado para cima. O ponto mais ascendente do objeto, ou seja quando seu vetor velocidade assumiu o menor valor possível de detectar nas iterações foi (3.87336 7.78133 -13.2428)m. Apesar de o objeto ter sido arremessado no centro da malha, e a deformação da mesma ter sido simétrica, a trajetória da partícula não foi. Provavelmente pelo movimento angular ter sido desconsiderado.

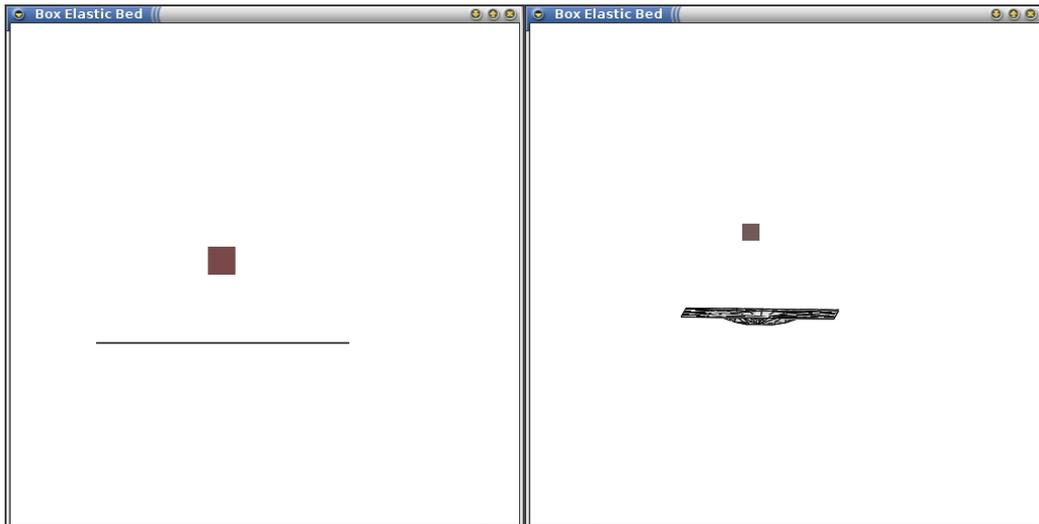


Figura 47: Malha e sólido antes da colisão. Reações da malha à colisão.

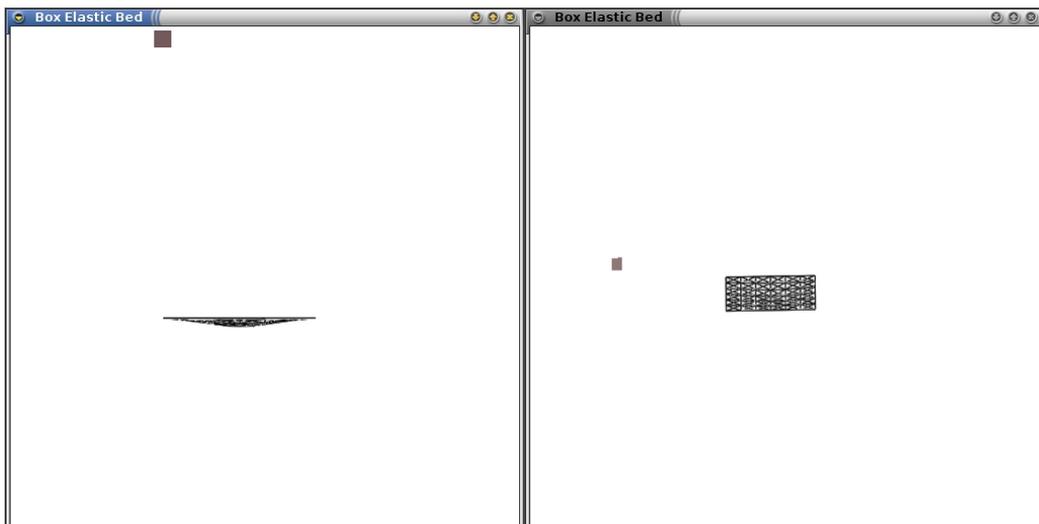


Figura 48: Reações da malha à colisão. Sólido na ascendente (esq) e descendente (dir).

6.4.2 Experimento 2

Os dados de entrada do experimento anterior foram mantidos, alterando apenas a força de ação sobre o sólido para $(0 \ -500 \ 0)N$.

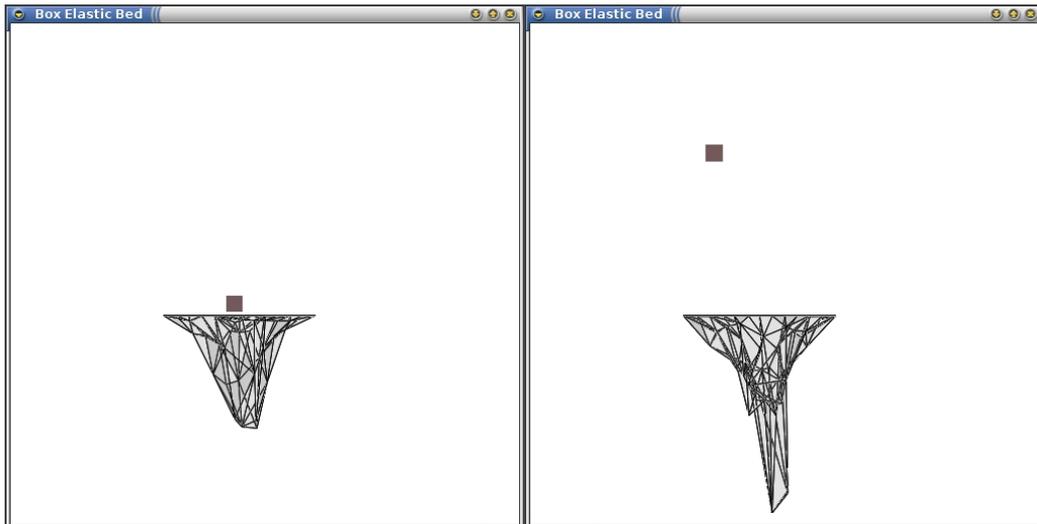


Figura 49: Malha apresentou uma deformação maior. Ocasionalmente o rompimento de uma mola.

Neste experimento a energia transmitida à malha durante o choque acabou gerando um trabalho na partícula suficiente para as molas se romperem, alcançando um comprimento maior do que o máximo possível. Como o modelo não trata rompimento no instante que o mesmo ocorre a malha pára.

6.4.3 Experimento 3

Assumindo os mesmos valores do experimento 2, alterando apenas para as molas:

$$k_e = 250N/m; k_d = -50N/m/s;$$

E para as partículas:

$$\mu = 15g$$

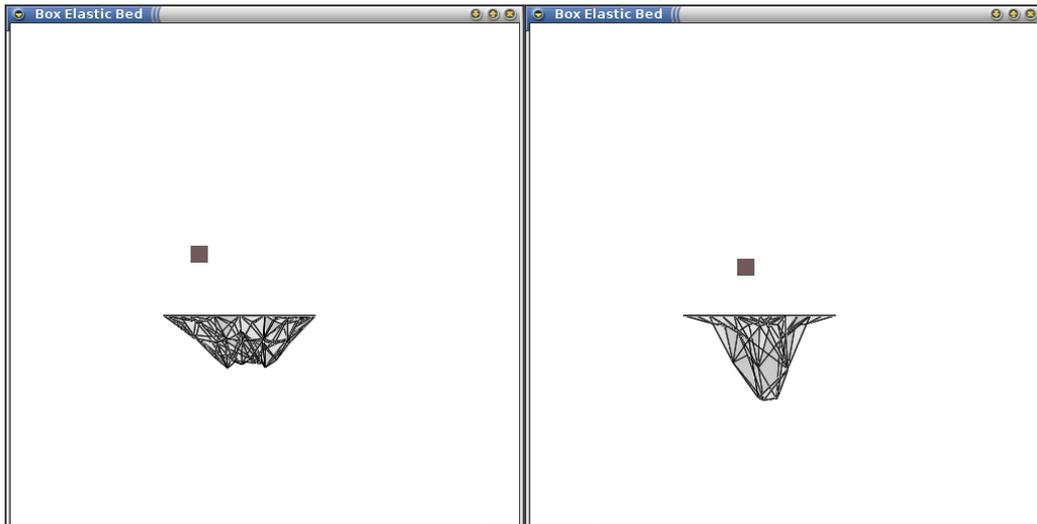


Figura 50: Deformações da malha.

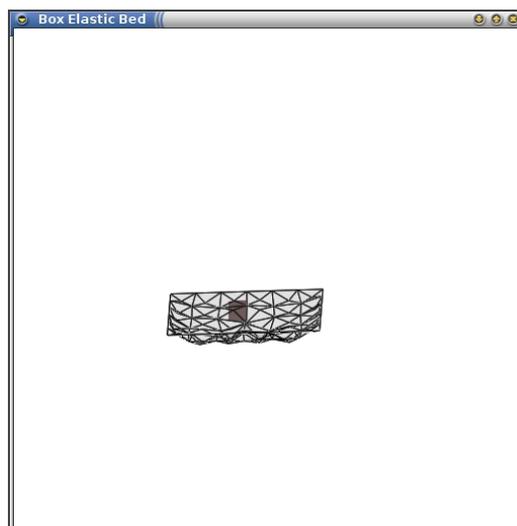


Figura 51: Malha retornando ao estado de equilíbrio estático.

Neste experimento o sólido foi expelido pela malha, porém seu centro atingiu a altura máxima em $(-1.6584 \ 0.765983 \ -4.38504)$ m. Isto ocorreu possivelmente devido às partículas apresentarem um valor de massa maior; logo, a velocidade final do sólido após o impacto tende a ser menor. Pelo fato de as molas terem coeficientes elásticos maiores elas acabaram por desacelerar as partículas as quais estão conectadas, influenciando também a velocidade destas.

6.4.4 Experimento 4

Neste experimento adotou-se uma configuração diferente para a disposição das partículas na malha. A figura 52 demonstra como as partículas estavam distribuídas nos experimentos anteriores estão circuladas duas partículas, uma conectas a 8 arestas e outra a 4. Na malha deste experimento com exceção das partículas da bordas, todos as demais estão ligadas a 6 molas. Mantendo para este experimento a mesma massa da malha anterior. Como as partículas da bordas são imóveis, para atribuir os valores de massa foi adotada a seguinte heurística, sendo np o número de partículas e npi o número de partículas imóveis.

$$(npMalha1 - npiMalha1) * Malha1_{\mu} = (npMalha2 - npiMalha2) * Malha2_{\mu}$$

Atribuindo para cada partícula:

$$\mu = 18,673469388g$$

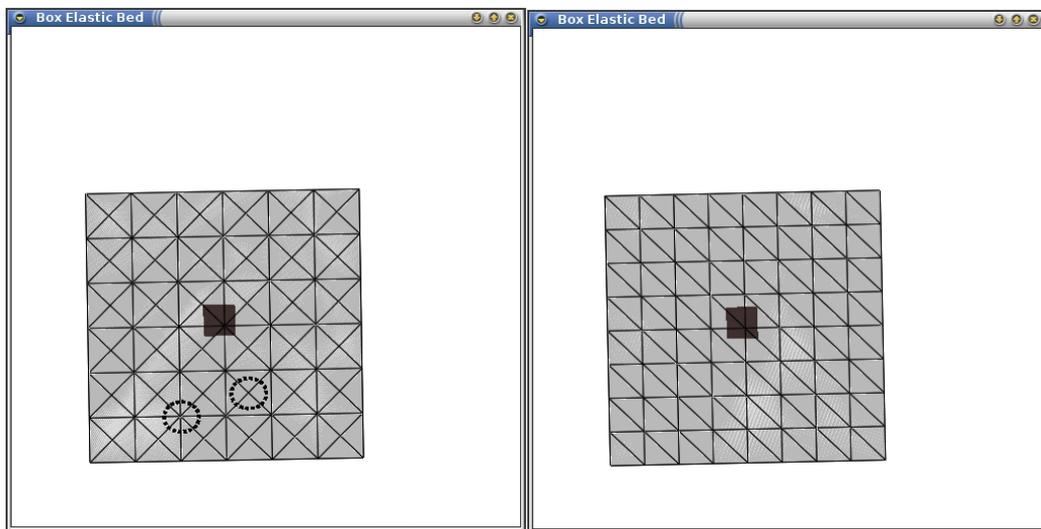


Figura 52: Partículas ligadas a um número diferente de molas(esq). Malha utilizada no experimento 4 (dir).

As molas com maior comprimento, no caso as que ligam duas partículas pela diagonal; exercem uma força maior. devido a isto a deformação da malha acaba sendo menos simétrica

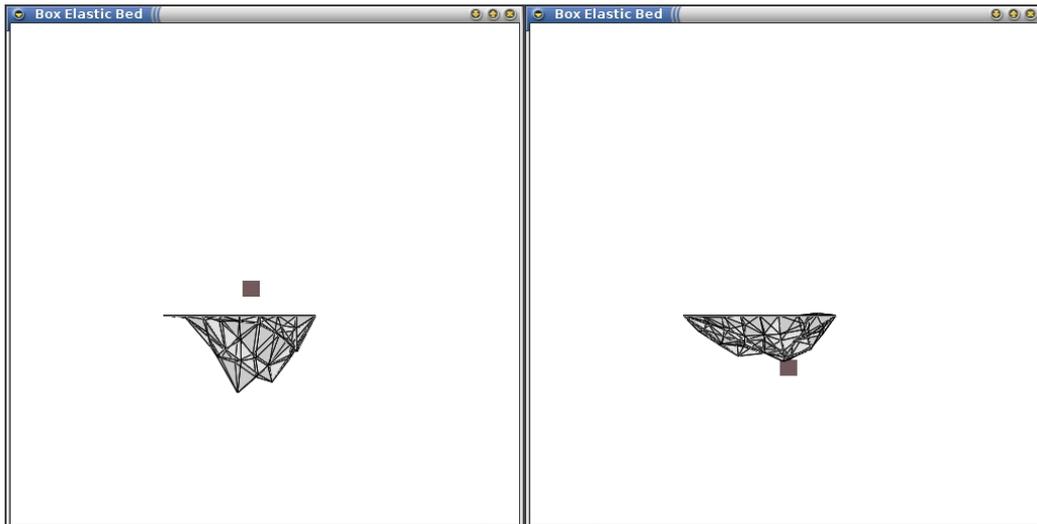


Figura 53: Malha deformada em função do contato com o sólido.

do que na malha anterior. O ponto mais alto do centro do sólido após o choque foi $(-0.439296$
 $-0.442468 -2.80337)$ m.

7 *Conclusões e Trabalhos Futuros*

Ao longo deste trabalho cada objetivo pré-estabelecido teve sua resolução condicionada a observação de efeitos colaterais gerados em protótipos desenvolvidos. A cada experimento pôde-se notar que a resolução de um problema incorporava algum outro efeito que podia gerar um comportamento ainda inesperado. Alcançar os objetivos passou a ser um exercício no qual as limitações precisavam ser minimizadas em conjunto, diminuindo a influência de todos os aspectos negativos, tornando o modelo mais estável e confiável.

As decisões tomadas nesse trabalho foram baseadas no equilíbrio de dois critérios principais geralmente antagônicos: realismo físico e tempo de processamento. Assim como os resultados apresentados, sempre em função de ambos. A seguir será discutido como o modelo se adaptou a estes dois critérios, quais foram as deficiências identificadas e que não puderam ser solucionadas no escopo temporal deste trabalho.

Como já explicado anteriormente, o realismo da simulação se refere à maneira que o modelo se insere em um contexto que necessita retornar respostas compatíveis com fenômenos físicos ocorridos. A adoção de um modelo híbrido que fazendo uso de técnicas geométricas em conjunto com propriedades físicas garantiu que a resposta fosse proporcional às grandezas espaço-temporais aplicadas. Isto pôde ser comprovado ao longo dos experimentos da seção final do capítulo 4 e de todo o capítulo 5.

A manutenção do equilíbrio estático de um tecido deformável foi crucial para a representação de deformações elásticas reversíveis. Entretanto, mesmo com a malha aproximando-se bastante da posição anterior à deformação, pode ser que nem todas as partículas retornem ao estado inicial. Esse problema pode ser explicado pela figura 54. O valor de ε é utilizado como

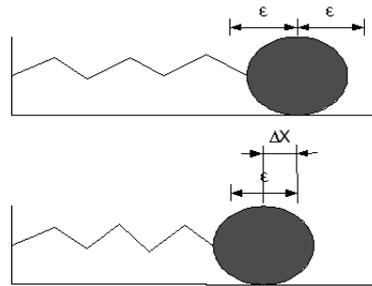


Figura 54: Deformação menor que ε

ponto limitante em relação à força de reação gerada pela mola, ou seja se ele for menor que Δx a força de reação é calculada. Portanto pode ser que a mola atinja uma variação de seu comprimento menor que ε , o que não garante que ela voltará ao estado inicial. Para corrigir isso deveria-se usar $\varepsilon = 0$, porém o efeito colateral disto seria que todas as molas do sistema seriam checadas a cada atualização, independente de sofrerem alteração ou não; ou forças desprezíveis seriam somadas às partículas durante uma propagação, acarretando que todas as partículas teriam que passar pelo processo de atualização de posição.

A atualização de posição de cada partícula depende das forças de atualização das molas conectadas a ela. Durante o procedimento responsável pelo cálculo do deslocamento a aceleração no instante em que ocorre a iteração é calculada em função da força e da massa da partícula. O deslocamento é calculado pelo método de Velocity Verlet que apesar de não ser totalmente exato, ele apresenta uma maior proximidade com resultados esperados em situações adversas, como aceleração e tempo variável; como pôde-se verificar em experimento realizados.

Em relação ao tempo uma explicação mais detalhada faz-se necessária. A simulação só apresenta resultados iguais em duas diferentes execuções se os intervalos de tempo forem iguais, não necessariamente constantes, em ambas. Este artifício foi utilizado na série de experimentos do capítulo 5 nos quais se verificavam os efeitos de uma colisão entre uma malha deformável e um objeto sólido. Nos demais experimentos o tempo não foi pré-estabelecido, ele era definido em tempo de execução, através do tempo que o processo que referencia o programa responsável pela simulação utilizava.

Estabelecidos os algoritmos responsáveis pela deformação de um tecido que garante a re-

versibilidade da deformação, passou-se a estudar o contato com elementos sólidos. Na falta de bibliografias específicas a respeito do tratamento de colisão em tempo real entre um sistema de partículas massa-mola semelhante ao do modelo com partículas ou planos, foram feitas adaptações de soluções já consolidadas para o tratamento de colisão entre partículas isoladas. A colisão de um sistema massa-mola com um sólido imóvel não leva em conta a troca de energia que ocorre no choque, pois não seria considerada nenhuma alteração em propriedades do sólido estático como velocidade ou posição. Entretanto o cálculo da velocidade da partícula é baseado no sentido que a normal que a face planar do sólido estático aponta, como pode se ver na figura 25 na página 68. Nos testes realizados a normal apontava para o sentido ao qual as partículas estavam localizadas, caso a normal estivesse apontando para outro sentido, a velocidade seria refletida para uma direção incorreta. Outro ponto importante é que não há troca de energia, logo supõe-se que a velocidade é a mesma antes e após o choque. Neste ponto seria possível incluir um coeficiente de dissipação, que diminuiria a velocidade da partícula da malha a cada choque.

A colisão de um objeto sólido rígido com um sistema de partículas massa-mola foi inspirado na colisão entre duas partículas isoladas no espaço. Neste caso existe a troca de energia entre ambas, podendo significar em alterações de velocidade e posição. No caso de partículas cujo o deslocamento ocasiona um choque que acarreta em um outro choque com uma terceira partícula, as duas colisões devem ser consideradas para o cálculo do momento linear. Neste modelo apresentado, os efeitos do segundo choque são calculados de maneira implícita. No primeiro choque a energia dissipada é transmitida às demais partículas conectadas e todo o efeito que isto gera na malha através do algoritmo 63. Depois de calculada toda a influência da energia recebida, a velocidade das partículas colididas estará atualizada de acordo com todos os elementos da malha passíveis de exercer alguma alteração nesta mesma grandeza. O cálculo do momento linear é então realizado sobre esta velocidade.

A inclusão do cálculo do momento angular neste choque estaria condicionada a encontrar o eixo de rotação de cada elemento envolvido e calcular o raio referente ao deslocamento aplicando sua rotação. Neste trabalho desconsiderou-se esta etapa, visto que exigiria uma adaptação

do modelo de deformação, além do choque em si. Contudo nos experimentos realizados pôde-se verificar modificações no comportamento apresentado pelos elementos envolvidos na colisão que possuíam uma relação direta às modificações aplicadas nos coeficientes.

Este trabalho deve ser visto como uma pesquisa de transição; um passo importante na modelagem de elementos deformáveis. Os resultados alcançados satisfazem os objetivos traçados previamente, alguns totalmente como no caso da manutenção do equilíbrio estático, outro parcialmente como no caso do choque entre sólidos e tecidos.

A maneira mais lógica de verificar se as deformações, colisões e grandezas físicas são condizentes com fenômenos possíveis no mundo real seria utilizar coeficientes provenientes de materiais os quais se deseja simular. A partir do momento que são atribuídos coeficientes hipotéticos pode-se ter uma idéia do comportamento através apenas da relevância que cada coeficiente representa nos algoritmos e equações definidas.

A heurística para a definição de coeficientes pode vir de um modelo como o de Choi, Kwak e Wang (2005) que utiliza lógica fuzzy para determinar os coeficientes de um sistema massa-mola simples. O trabalho de Hongping, Yuzhou e Jiaxiang (2002) faz uso de um sistema massa-mola com coeficientes definidos pela aplicação de uma variação do algoritmo back-propagation em redes neurais recorrentes para avaliar a resistência de materiais em estruturas metálicas de construção civil. O trabalho de Monserrat et al. (2001) definiu coeficientes elástico do fígado a partir de experiências ex-vivo realizadas com fígados de porcos.

A extensão deste modelo à realização de uma simulação que exija uma grande exatidão de resultados; como uma simulação que envolva tecidos humanos está condicionada não apenas à adoção de coeficientes físicos relacionados, mas também a uma representação poligonal que seja próxima à forma real a qual se deseja simular. No último experimento do capítulo 5 ficou claro que a maneira que os polígonos da malha são organizados também influi no comportamento elástico.

Em uma simulação deste tipo deve haver também a possibilidade de romper e unir tecidos. A adição destas possibilidades estaria condicionada a um maior estudo de como manter o equi-

líbrio estático na eliminação de molas, além dos efeitos que a inclusão ou remoção de vértices pode causar, como foi frisado na última seção do capítulo 5.

Com todos os conceitos demonstrados até aqui, com as limitações sendo demonstradas, segue adiante um último experimento. O intuito é apresentar novas idéias, demonstrando o que pode ser feito no estado atual de desenvolvimento, analisando os pontos negativos e propondo caminhos para novas adaptações e melhorias. Como foi demonstrado anteriormente, o trabalho de Santos (2006) apresentou modificações na DLFL que tornaram possível armazenar volumes de reconstruções de parte do corpo humano a partir de séries de imagens médicas. O experimento demonstra uma aplicação que fez uso desta funcionalidade, adaptando uma artéria aorta abdominal a uma malha deformável. Para interagir com esta malha, foi importada uma adaga tridimensional. Esta adaga foi criada no software de modelagem 3D Blender e salva no formato STL, que guarda informações de faces e vértices que as constituem. Com uma rotina desenvolvida por mim paralelamente a este trabalho, esses dados foram importados e adaptados a um sólido rígido. A figura 55 demonstra os dois objetos posicionados, antes da simulação.

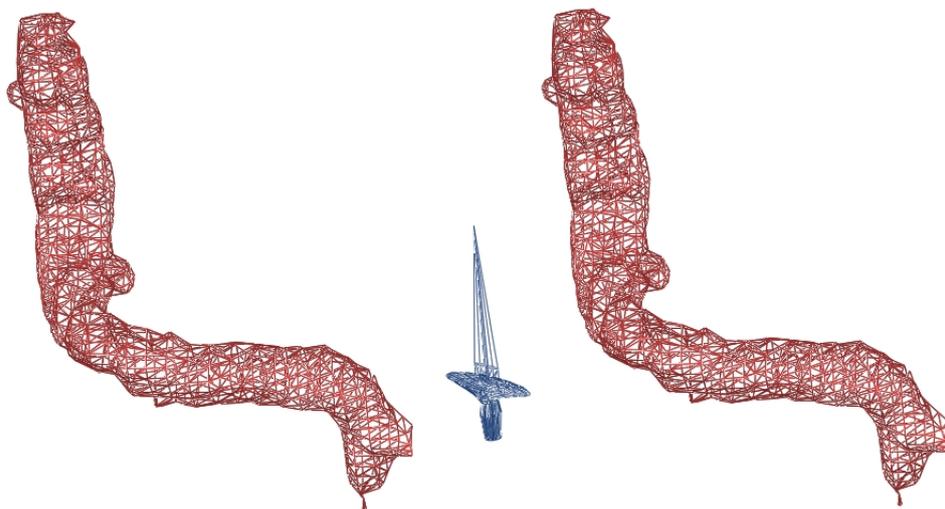


Figura 55: Artéria e adaga. Ambos compostos por polígonos planares.

Durante a primeira simulação a ser executada, observou-se que a taxa de atualização estava muito baixa, em torno de 1 quadro a cada 5 segundos, muito diferente dos requisitos estabelecidos anteriormente e taxas de atualização dos experimentos anteriores que variavam de 60 até

100 quadros por segundo. Um fator relevante era a quantidade de polígonos da adaga, formado por 467 faces. Então um teste seguinte foi executado só com a lâmina da adaga, formada por 25 faces, rodando a aproximadamente 3 frames por segundo. A conclusão disto é que reduzindo o número de faces do sólido deformável, foi reduzida a quantidade de vezes que a rotina de detecção de colisão foi acionada. Nesta etapa poderiam ser adotados algoritmos que otimizassem esta tarefa, particionando o espaço e checando apenas estruturas que apresentem proximidade espacial. Existe bibliografia suficiente nos dias atuais para efetuar isto, como se pode conferir em Bergen's (2004).

No experimento a lâmina da adaga foi posicionada abaixo da artéria e pressionada contra ela, os resultados são conferidos nas figuras a seguir.



Figura 56: Colisão lâmina-artéria. Artéria se rompe.

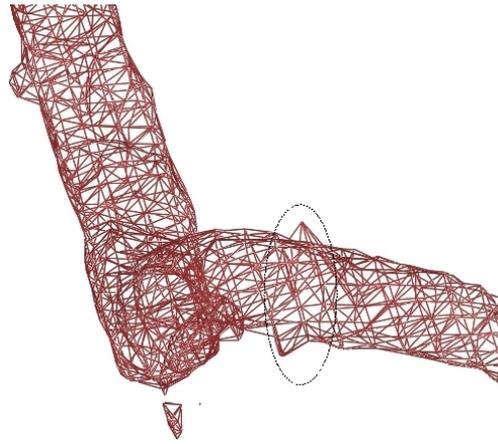


Figura 57: Artéria aproximada em modelo de arame, após rompimento. Elipse indicando região afetada.

Provavelmente pela complexidade da malha de polígonos que constitui a artéria o efeito não possa ser identificado visualmente nas figuras aqui demonstradas, porém a força a qual a adaga deslocou a região da artéria a qual tocou, foi suficiente para arremessar as partículas para outro lado cruzando outra parede da artéria. Como o modelo não trata rompimentos neste instante a simulação foi interrompida, mas esse efeito levanta outra questão, é necessário também haver uma checagem de colisão entre os membros que constituem uma malha. Uma partícula de uma malha não deve ser capaz de atravessar uma face desta mesma malha; ela deve gerar uma troca de energia, assim como qualquer outra partícula externa a malha.

Após essas análises finais pode-se afirmar que este trabalho cumpriu sua proposta inicial de definir um modelo computacional para representação gráfica do comportamento elástico de tecidos e sua interação com sólidos. Apesar de diversos pontos não terem sido implementados, as fragilidades encontradas são destacadas indicando as direções para as possíveis soluções. Desta forma esse trabalho funciona como uma ponte, unindo idéias antes desconexas e apresentando um caminho para uma futura implementação.

8 Anexos

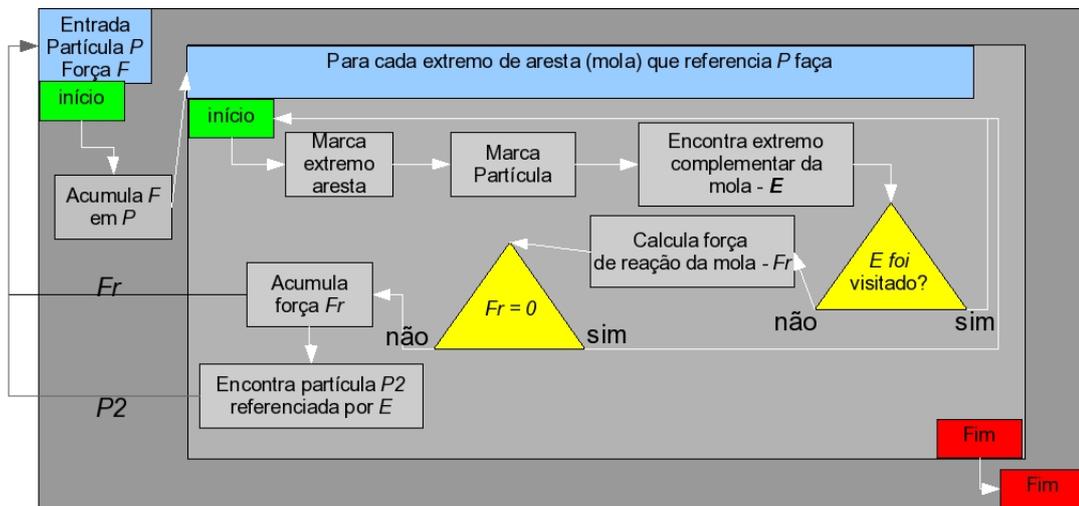


Figura 58: Algoritmo 8 - Diagrama de Blocos

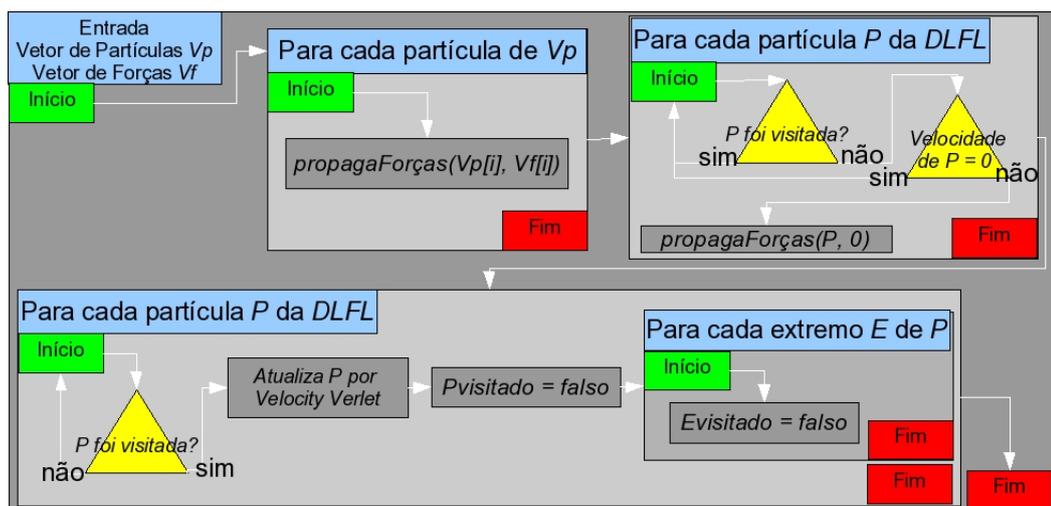


Figura 59: Algoritmo 9 - Diagrama de Blocos

dt	t	a	V_exato	X_exato	X_verlet	TCV	V_euler	X_euler	V_VVerlet	X_VVerlet
0,25	0	-9,81	61,29	0	0	0	61,29	0	61,29	0
0,25	0,25	-9,81	58,84	15,02	15,02	15,02	58,84	15,02	58,84	15,02
0,25	0,5	-9,81	56,38	29,42	29,42	29,42	56,38	29,11	56,38	29,42
0,25	0,75	-9,81	53,93	43,21	43,21	43,21	53,93	42,59	53,93	43,21
0,25	1	-9,81	51,48	56,38	56,38	56,38	51,48	55,47	51,48	56,38
0,25	1,25	-9,81	49,03	68,95	68,95	68,95	49,03	67,72	49,03	68,95
0,25	1,5	-9,81	46,58	80,9	80,9	80,9	46,58	79,37	46,58	80,9
0,25	1,75	-9,81	44,13	92,24	92,24	92,24	44,13	90,4	44,13	92,24
0,25	2	-9,81	41,68	102,96	102,96	102,96	41,68	100,82	41,68	102,96
0,25	2,25	-9,81	39,22	113,08	113,08	113,08	39,22	110,62	39,22	113,08
0,25	2,5	-9,81	36,77	122,58	122,58	122,58	36,77	119,82	36,77	122,58
0,25	2,75	-9,81	34,32	131,46	131,46	131,46	34,32	128,4	34,32	131,46
0,25	3	-9,81	31,87	139,74	139,74	139,74	31,87	136,36	31,87	139,74
0,25	3,25	-9,81	29,42	147,4	147,4	147,4	29,42	143,72	29,42	147,4
0,25	3,5	-9,81	26,97	154,44	154,44	154,44	26,97	150,46	26,97	154,44
0,25	3,75	-9,81	24,52	160,88	160,88	160,88	24,52	156,59	24,52	160,88
0,25	4	-9,81	22,06	166,7	166,7	166,7	22,06	162,11	22,06	166,7
0,25	4,25	-9,81	19,61	171,91	171,91	171,91	19,61	167,01	19,61	171,91
0,25	4,5	-9,81	17,16	176,51	176,51	176,51	17,16	171,3	17,16	176,51
0,25	4,75	-9,81	14,71	180,49	180,49	180,49	14,71	174,98	14,71	180,49
0,25	5	-9,81	12,26	183,86	183,86	183,86	12,26	178,04	12,26	183,86
0,25	5,25	-9,81	9,81	186,62	186,62	186,62	9,81	180,49	9,81	186,62
0,25	5,5	-9,81	7,35	188,77	188,77	188,77	7,35	182,33	7,35	188,77
0,25	5,75	-9,81	4,9	190,3	190,3	190,3	4,9	183,56	4,9	190,3
0,25	6	-9,81	2,45	191,22	191,22	191,22	2,45	184,17	2,45	191,22
0,25	6,25	-9,81	0	191,52	191,52	191,52	0	184,17	0	191,52

Tabela 10: experimento 1a - seção 4.3

dt	t	a	V_exato	X_exato	Pverlet	TCV	V_euler	Euler	V_VVerlet	Vverlet
0,36	0	-9,81	61,29	0	0	0	61,29	0	61,29	0
0,1	0,36	-9,81	57,76	21,39	21,39	21,39	57,76	21,39	57,76	21,39
0,2	0,46	-9,81	56,82	26,93	42,69	27,05	56,82	26,88	56,82	26,93
0,28	0,65	-9,81	54,9	37,83	63,62	38,11	54,9	37,6	54,9	37,83
0,24	0,93	-9,81	52,2	52,61	83,79	53,01	52,2	52	52,2	52,61
0,13	1,17	-9,81	49,85	64,83	103,41	65,36	49,85	63,94	49,85	64,83
0,17	1,29	-9,81	48,59	71,13	122,87	71,81	48,59	70,16	48,59	71,13
0,4	1,46	-9,81	46,95	79,12	142,05	79,96	46,95	78,02	46,95	79,12
0,05	1,87	-9,81	42,99	97,28	159,63	98,03	42,99	95,38	42,99	97,28
0,23	1,91	-9,81	42,53	99,28	177,19	100,1	42,53	97,36	42,53	99,28
0,03	2,15	-9,81	40,24	108,97	194,21	109,94	40,24	106,79	40,24	108,97
0,37	2,18	-9,81	39,95	110,15	211,23	111,16	39,95	107,96	39,95	110,15
0,33	2,55	-9,81	36,29	124,36	226,88	125,34	36,29	121,49	36,29	124,36
0,35	2,88	-9,81	33,03	135,9	241,45	136,93	33,03	132,49	33,03	135,9
0,15	3,23	-9,81	29,63	146,75	254,84	147,8	29,63	142,75	29,63	146,75
0,33	3,38	-9,81	28,18	151,03	268,01	152,24	28,18	146,92	28,18	151,03
0,22	3,71	-9,81	24,92	159,86	280,1	161,11	24,92	155,21	24,92	159,86
0,11	3,92	-9,81	22,81	165	291,73	166,4	22,81	160,12	22,81	165
0,13	4,04	-9,81	21,71	167,49	303,25	169,03	21,71	162,56	21,71	167,49
0,43	4,16	-9,81	20,47	170,16	314,6	171,84	20,47	165,14	20,47	170,16
0,16	4,59	-9,81	16,25	178,05	324,14	179,59	16,25	172,13	16,25	178,05
0,43	4,75	-9,81	14,67	180,55	333,43	182,24	14,67	174,5	14,67	180,55
0,13	5,18	-9,81	10,5	185,9	340,95	187,45	10,5	178,96	10,5	185,9
0,19	5,31	-9,81	9,23	187,18	348,3	188,88	9,23	180,16	9,23	187,18
0,26	5,5	-9,81	7,33	188,78	355,28	190,64	7,33	181,58	7,33	188,78
0,48	5,76	-9,81	4,77	190,36	361,59	192,35	4,77	182,82	4,77	190,36

Tabela 11: experimento 1b - seção 4.3

dt	t	a	V_exact	X_exact	X_verlet	TCV	V_euler	X_euler	V_VVerlet	X_VVerlet
0,15	0	-14	5	0	0	0	5	0	5	0
0,36	0,15	-12,2	3,03	0,6	0,6	0,6	3,03	0,6	3,03	0,61
0,23	0,51	-7,92	-0,55	1	-0,35	0,48	-1,31	0,13	-0,55	1,19
0	0,74	-5,11	-2,08	0,68	-1,73	-0,04	-3,17	-0,61	-2,08	0,92
0,03	0,75	-5,06	-2,1	0,67	-3,11	-0,05	-3,19	-0,62	-2,1	0,91
0,31	0,77	-4,7	-2,24	0,6	-4,5	-0,12	-3,34	-0,72	-2,24	0,85
0,36	1,09	-0,93	-3,13	-0,27	-6,35	-1,34	-4,82	-2,24	-3,13	0,1
0,13	1,45	3,43	-2,68	-1,38	-8,32	-2,86	-5,16	-4,11	-2,68	-0,82
0,01	1,58	4,95	-2,15	-1,68	-10,24	-3,34	-4,73	-4,71	-2,15	-1,11
0,14	1,59	5,07	-2,1	-1,7	-12,16	-3,37	-4,68	-4,76	-2,1	-1,13
0,37	1,73	6,74	-1,27	-1,94	-13,98	-3,79	-3,97	-5,31	-1,27	-1,36
0,14	2,09	11,13	1,99	-1,86	-14,9	-3,99	-1,5	-5,86	1,99	-1,08
0,31	2,23	12,78	3,64	-1,47	-15,61	-3,85	0,03	-5,86	3,64	-0,69
0,27	2,54	16,47	8,13	0,3	-15,11	-2,34	3,95	-4,64	8,13	1,21
0,18	2,81	19,7	13,01	3,13	-13,42	0,18	8,39	-2,38	13,01	4,11
0,04	2,99	21,91	16,84	5,88	-11,06	2,57	12,03	-0,16	16,84	6,89
0,3	3,03	22,41	17,76	6,59	-8,66	3,14	12,93	0,37	17,76	7,6
0,06	3,33	26	24,99	12,95	-4,26	9,3	19,63	6,24	24,99	14,07
0,37	3,39	26,68	26,49	14,42	0,23	10,56	21,11	7,44	26,49	15,54
0,14	3,76	31,17	37,32	26,31	8,45	22,56	31,1	19,08	37,32	27,64
0,21	3,91	32,91	41,97	32,06	17,33	27,86	35,62	24,25	41,97	33,4
0,21	4,11	35,38	48,98	41,39	27,6	36,76	42,38	32,95	48,98	42,76
0,36	4,32	37,86	56,56	52,3	39,38	47,25	49,7	43,23	56,56	53,71
0,32	4,68	42,18	70,97	75,22	56,08	70,4	63,33	66,04	70,97	76,82
0,33	5	45,99	84,98	99,95	77,03	95,08	76,73	90,42	84,98	101,68
0,29	5,33	49,99	100,98	130,91	103,09	126,09	92,06	121,11	100,98	132,78

Tabela 12: experimento 2 - seção 4.3

dt	t	a	V_exact	X_exact	X_verlet	TCV	V_euler	X_euler	V_VVerlet	X_VVerlet
0,87	0	0	-1	0	0	0	-1	0	-1	0
0,78	0,87	0,76	-0,65	-0,76	-0,76	-0,76	-0,65	-0,76	-0,67	-0,58
0,33	1,65	1	0,08	-1	-1,06	-0,98	-0,05	-0,8	0,02	-0,8
0,84	1,98	0,92	0,39	-0,92	-1,24	-0,97	0,28	-0,71	0,33	-0,74
0,13	2,81	0,32	0,95	-0,32	-0,79	-0,29	1,04	0,16	0,85	-0,35
0,25	2,95	0,19	0,98	-0,19	-0,32	-0,17	1,09	0,31	0,89	-0,24
0,65	3,2	-0,06	1	0,06	0,15	0,05	1,14	0,59	0,9	-0,02
0,2	3,85	-0,65	0,76	0,65	0,6	0,62	1,1	1,31	0,67	0,43
0,63	4,05	-0,79	0,62	0,79	1,03	0,76	0,97	1,5	0,53	0,55
0,55	4,68	-1	0,03	1	1,13	0,92	0,47	1,8	-0,04	0,69
1	5,23	-0,87	-0,5	0,87	0,94	0,74	-0,08	1,76	-0,55	0,53
0,74	6,23	-0,05	-1	0,05	-0,12	-0,43	-0,95	0,81	-1,01	-0,04
0,93	6,97	0,63	-0,77	-0,63	-1,21	-1,33	-0,98	0,08	-0,79	-0,61
0,59	7,9	1	0,04	-1	-1,75	-1,91	-0,4	-0,28	-0,04	-0,92
0,43	8,49	0,81	0,59	-0,81	-1,95	-1,93	0,19	-0,17	0,49	-0,8
0,1	8,92	0,49	0,87	-0,49	-2	-1,8	0,54	0,06	0,77	-0,55
0,6	9,01	0,4	0,92	-0,4	-2,04	-1,77	0,59	0,12	0,82	-0,47
0,4	9,61	-0,19	0,98	0,19	-1,94	-1,41	0,82	0,61	0,88	-0,01
0,07	10,01	-0,55	0,83	0,55	-1,87	-1,21	0,75	0,91	0,73	0,29
0,67	10,08	-0,61	0,8	0,61	-1,8	-1,17	0,71	0,96	0,69	0,34
0,04	10,75	-0,97	0,25	0,97	-2,01	-1,12	0,31	1,16	0,17	0,59
0,32	10,79	-0,98	0,2	0,98	-2,22	-1,12	0,27	1,18	0,12	0,59
0,95	11,11	-0,99	-0,12	0,99	-2,53	-1,21	-0,05	1,16	-0,19	0,58
0,71	12,06	-0,48	-0,87	0,48	-3,73	-2,37	-0,99	0,22	-0,89	0,18
0,35	12,77	0,2	-0,98	-0,2	-5,17	-3,48	-1,33	-0,72	-1	-0,4
0,16	13,11	0,52	-0,85	-0,52	-6,59	-4	-1,26	-1,16	-0,87	-0,72

Tabela 13: experimento 3 - seção 4.3

Referências

- AKLEMAN, E.; CHEN, J. Guaranteeing 2 manifold property for meshes. In: *Shape Modeling and Applications, 1999. Proceedings. Shape Modeling International '99. International Conference on*. [S.l.]: IEEE, 1999. p. 18 25. ISBN 0 7695 0065 X.
- AKLEMAN, Ergun; CHEN, Jianer. Guaranteeing the 2 manifold property for meshes with doubly linked face list. *International Journal of Shape Modeling*, v. 5, n. 2, p. 149 177, 2000.
- AKLEMAN, E.; CHEN, J.; SRINIVASAN, V. A minimal and complete set of operators for the development of robust manifold mesh modelers. *Graph. Models*, Academic Press Professional, Inc., San Diego, CA, USA, v. 65, n. 5, p. 286 304, 2003.
- BARDINET, E.; COHEN, L.D.; AYACHE, N. Tracking and motion analysis of the left ventricle with deformable superquadrics. *Medical Image Analysis*, v. 1, n. 2, p. 129 149, 1996.
- BARGEN, B.; DONELLY, P. *Inside DirectX*. [S.l.]: Microsoft corporation, 1998. ISBN 1572316969.
- BATCHO, P.F.; SCHLICK, T. Special stability advantages of position verlet over velocity verlet in multiple time step integration. *Journal of Chemical Physics*, v. 115, n. 9, p. 4019 4029, set. 2001.
- BAUMGART, B. G. *Winged Edge Polyhedron Representation*. San Francisco, CA, EUA, out. 1972.
- BERGEN'S, Gino van den. *Collision Detection in interactive 3D environments*. [S.l.]: Morgan Kaufmann Elsevier, 2004. (Morgan Kaufmann Series in Interactive 3D Technology).
- BROWN, J.; SORKIN, S.; BRUYNS, C.; LATOMBE, J. C.; MONTGOMERY, K.; STEPHANIDES, M. Real time simulation of deformable objects: tools and application. In: . [S.l.]: Computer Animation, 2001. The Fourteenth Conference on Computer Animation. Proceedings, 2001. p. 228 258. ISBN 0 7803 7237 9.
- CAKMAK, H.K.; KUHNAPFEL, U. Animation and simulation techniques for vr training systems in endoscopic surgery. In: *Eurographics Workshop on Animation and Simulation*. [S.l.: s.n.], 2000.
- CARVALHO, D.; SANTOS, T. dos; WANGENHEIM, A. von. Measuring arterial diameters for surgery assistance, patient customized endovascular prosthesis design and post surgery evaluation. In: *Proceedings 19th CBMS*. [S.l.]: IEEE, 2006. p. 225 230.
- ÇAVUSOGLU, M.C.; GOKTEKIN, T.G.; TENDICK, F. Gipsi: A framework for open source/open architecture software development for organ level surgical simulation. In: *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*,. [S.l.]: IEEE, 2006. v. 10, p. 312 322.

- CHOI, B J.; KWAK, S W.; WANG, H. Stability analysis of a simple structured fuzzy logic controller. In: *Journal of Intelligent and Robotic Systems*. [S.l.]: Springer Netherlands, 2005. v. 42, n. 2, p. 169 178.
- DEBUNNE, G.; DESBRUN, M.; CANI, M P.; BARR, A.H. Dynamic real time deformations using space & time adaptive sampling. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. [S.l.]: ACM Press, 2001. p. 31 36.
- DELINGETTE, H. Toward realistic soft tissue modeling in medical simulation. In: *IEEE : Special Issue on Surgery Simulation*. [S.l.]: IEEE, 1998. v. 86, n. 3, p. 512 523.
- DESBRUN, M.; SCHRÖDER, P.; BARR, A. Interactive animation of structured deformable objects. In: *Graphics Interface*. [S.l.: s.n.], 1999. p. 1 8.
- DOCHEV, V.; VASSILEV, T. Efficient super elasticity handling in mass spring systems. In: *CompSysTech '03: Proceedings of the 4th international conference conference on Computer systems and technologies*. New York, NY, USA: ACM Press, 2003. p. 483 490. ISBN 954 9641 33 3.
- DUMMER, J. *A Simple Time-Corrected Verlet Integration Method*. 2 2005. Web. Disponível em: <<http://www.gamedev.net/reference/articles/article2200.asp>>.
- EBERLY, David H.; SHOEMAKE, Ken. *Game Physics*. 500 Sansome Stree, Suite 400, San Francisco, CA 94111: Morgan Kaufmann Elsevier, 2004. ISBN 1 55860 740 4.
- ECCLES; ALLEN. *The Diamond Monster 3Dfx Voodoo 1*. 2000. Gamespy Hall of Fame. Disponível em: <<http://www.gamespy.com/halloffame/october00/voodoo1>>.
- EGLI, R.; STEWART, N. F. Chain models in computer simulation. In: *Mathematics and Computers in Simulation*. [S.l.]: Elsevier, 2004. v. 66, p. 449 468.
- FOLEY, James D.; DAM, Andries van; FEINER, Steven K.; HUGHES, John H. *Computer Graphics Principles and Practice*. 2. ed. Reading, Massachusetts, USA: Addison Wesley, 1990.
- GASCUEL, M.P. Cani; DESBRUN, M. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, v. 3, n. 1, p. 39 50, 1997.
- GIBSON, S.F.F. *A Survey of Deformable Modeling in Computer Graphics*. [S.l.], nov. 1997.
- GIBSON, S. F. F. *Linked Volumetric Objects for Physics based Modeling*. Cambridge, MA, nov. 1997.
- HAIRER, E.; LUBICH, C.; WANNER, G. Geometric numerical integration illustrated by the störmer/verlet method. *Acta Numerica*, Cambridge University Press, v. 12, p. 339 450, jul. 2003.
- HENG, P.A.; CHENG, C.Y.; WONG, T.T.; XU, Y.; CHUI, Y.P.; CHAN, K.M.; TSO, S.K. A virtual reality training system for knee arthroscopic surgery. In: *IEEE Transactions on Information Technology in Biomedicine*. [S.l.]: IEEE, 2004. v. 8, p. 217 227.

- HONGPING, Z.; YUZHOU, S.; JIAXIANG, T. Damage detection in structures using modified back propagations neural networks. In: *Acta Mechanica Solida Sinica*. [S.l.]: Springer Berlin, 2002. v. 15, n. 4, p. 358 370.
- HUTCHINSON, D; PRESTON, M; HEWITT, T. Adaptive refinement for mass/spring simulation. In: *7th Eurographics Workshop on Animation and Simulation*. [S.l.: s.n.], 1996.
- JI, F.; LI, R.; QIU, Y. Three dimensional garment simulation based on a mass spring system. *Textile Research Journal*, v. 76, p. 12 17, 2006.
- KRY, P.G.; JAMES, D.L.; PAI, D.K. Eigenskin: real time large deformation character skinning in hardware. In: *ACM SIGGRAPH/Eurographics symposium on Computer animation*. New York, NY, USA: ACM Press, 2002. p. 153 159. ISBN 1 58113 573 4.
- KUHNAPFEL, U.G.; KRUMM, H.; HÜBNER, M.; NEISIUS, B. Endosurgery simulations with kismet: A flexible tool for surgical instrument design, operation room planning and vr technology based abdominal surgery training. In: *Proc. Virtual Reality World*. [S.l.: s.n.], 1995.
- LORENSEN, William E.; CLINE, Harvey E. Marching cubes: A high resolution 3d surface construction algorithm. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. [S.l.]: ACM Press, 1987. p. 163 169.
- MANTYLA, Martti. *Introduction to Solid Modeling*. Rockville, Maryland, USA: Computer Science Press, 1988.
- MCINERNEY, T.; TERZOPOULOS, D. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, v. 1, n. 2, p. 91 108, 1996.
- MELO, V.F. de; WU, S T. *Estado da Arte de Modelos Deformáveis*. [S.l.], dez. 2002.
- MELO, Vanio Fragoso de. *Modelagem e Controle de Caimento e Dobras em Superfícies Deformáveis*. Tese (Doutorado) — Universidade Estadual de Campinas Faculdade de Engenharia Elétrica e Computação Departamento de Engenharia de Computação e Automação Industrial, Campinas SP, abr. 2004.
- MÖLLER, T. Akenine; HAINES, E. *Real Time Rendering*. Second. 888 Worcester Street, Suite 230 Wellesley, MA 02482: A K Peters, 2002. ISBN 1 56881 182 9.
- MONSERRAT, C.; MEIER, U.; ALCANIZ, M.; CHINESTA, F.; JUAN, M. C. A new approach for the real time simulation of tissue deformations in surgery simulation. *Computer Methods and Programs in Biomedicine*, v. 64, n. 2, p. 77 85, Feb 2001.
- MONTAGNAT, J.; H., Delingette.; AYACHE, N. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, v. 19, n. 14, p. 1023 1040, 2001.
- MOOSEGAARD, J. Lr spring mass model for cardiac surgical simulation. In: *Proceedings of Medicine Meets Virtual Reality*. [S.l.: s.n.], 2004.
- NIELSEN, Morten Bro; COTIN, Stephane. Real time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum*, Blackwell Publishers, v. 15, n. 3, p. 57 66, 1996. Disponível em: <citeseer.ist.psu.edu/bro nielsen96realtime.html>.

PROVOT, Xavier. Deformation constraints in a massspring model to describe rigid cloth behavior. In: . [S.l.]: Proc. Graphics Interface, 1995. p. 147 154.

QIAN, X.; SCHLICK, T. Efficient multiple time step integrators with distance based force splitting for particle mesh ewald molecular dynamics simulations. *Journal of Chemical Physics*, v. 116, n. 14, p. 5972 5983, abr. 2002.

R.A.WHITE; C.DONAYRE; WALOT, I.; STEWART, M. Abdominal aortic aneurysm rupture following endoluminal graft deployment: Report of a predictable event. *J Endovasc Ther*, v. 4, n. 2, p. 257 62, Aug 2000.

REDETZKY, A.; AUER, D.P.; C., Wimmer; PRETSCHNER, D.P.; AUER, L.M. Robo sim: a simulator for minimally invasive interventions (abstr). In: *Radiology*. [S.l.: s.n.], 1998. p. 209.

ROBB, R. A.; CAMERON, B. Computer aided surgery planning and rehearsal at mayo clinic. In: AL, R. Satara et (Ed.). *nteractive Technology and the New Paradigm for Healthcare*. Amsterdam, Netherlands: IOS Press, 1995. v. 18, p. 175 190.

SAKAMOTO, Y.; TUCHIYA, K.; KATO, M. Deformation method for surgery simulation using voxel spaceautomata. In: *IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999*. Tokyo, Japan: IEEE, 1999. v. 4, p. 1026 1031. ISBN 0 7803 5731 0.

SANTOS, T.R. dos. *Representação, Visualização e Manipulação de Dados Médicos Tridimensionais: Um Estudo Sobre as Bases da Simulação Cirúrgica Imersiva*. Dissertação (Mestrado) — UFSC, Florianópolis, SC, Brasil, 2006.

SCHUMACHER, H.; ECKSTEIN, H.H.; KALLINOWSKI, F.; ALLENBERG, J.R. Morphometry and classification in abdominal aortic aneurysms: Patient selection for endovascular and open surgery. *J Endovasc Surg*, v. 4, n. 1, p. 39 44, Feb 1997.

SEGAL, Mark; AKELEY, Kurt. *The OpenGL Graphics Interface*. Mountain View, CA,USA, 1993. Disponível em: <citeseer.ist.psu.edu/332034.html>.

SEGAL, M.; AKELEY, K. *The OpenGL Graphics System: A Specification*. [S.l.], 2004. Disponível em: <<http://www.opengl.org/documentation/specs/version2.0/glspec20.pdf>>.

SRINIVASAN, Vinod; AKLEMAN, Ergun; KEYSER, John. *Topological Construction of 2 Manifold Meshes from Arbitrary Polygonal Data*. Tamu College Station, Texas, USA, 2004.

SWOPE, W.C.; ANDERSEN, H.C.; BERENS, P.H.; WILSON, K.R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J.Chem. Phys.*, v. 76, n. 1, p. 637 649, jan. 1982.

TERZOPOULOS, D.; PLATT, J.; BARR, A.; FLEISCHER, K. Elastically deformable models. *Computer Graphics*, v. 21, 1987.

TESCHNER, M.; HEIDELBERG, B.; MULLER, M.; GROOS, M. A versatile and robust model for geometrically complex deformable solids. In: *Computer Graphics International, 2004. Proceedings*. [S.l.]: IEEE, 2004. p. 312 319. ISBN 0 7695 2171 1.

VERLET, L. Computer “experiments” on classical fluids. i. thermodynamical properties of lennard jones molecules. *Physical Review*, v. 159, p. 98 103, jun. 1967.

VOLINO, P.; THALMANN, N. Magnenat. Accurate garment prototyping and simulation. In: *Computer Aided Design & Applications*,. [S.l.: s.n.], 2005. v. 2, n. 1 4.

ZHANG, J.; PAYANDEH, S.; DILL, J. Levels of detail in reducing cost of haptic rendering: a preliminary user study. In: *Proc. 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS 2003)*. [S.l.: s.n.], 2003. p. 205 212.

ZHANG, Y.; PRAKASH, E.C.; SUNG, E. Real time physically based facial expression animation using mass spring system. In: *Computer Graphics International 2001*. Hong Kong, China: IEEE, 2001. p. 347 350.

ZHUANG, Yan; CANNY, John. *Real time simulation of physically realistic global deformations*. Tese (Doutorado) — Computer Science Department, University of California, Berkeley, 2000.

ZIENKIEWICZ, O.C.; ZHU, J.Z.; TAYLOR, R.L. *The Finite Element Method: Its Basis and Fundamentals*. 6. ed. [S.l.]: Elsevier, 2005.