

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

**Método baseado em física para visualização
3D de modelos complexos**

Andrei Alhadeff Monteiro

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Engenharia da Computação

Rio de Janeiro, Junho de 2009



Andrei Alhadeff Monteiro

**Método baseado em física para visualização 3D de
modelos complexos**

Relatório de Projeto Final, apresentado ao programa
Projeto Final da PUC-Rio como requisito parcial para a
obtenção do título de Engenheiro de Computação.

Orientador: Waldemar Celes Filho

Rio de Janeiro
Junho de 2009.

Agradecimentos

Ao meu orientador, Waldemar Celes, por me incentivar e propor este desafio que hoje pude realizar. A meus pais, Ivan de Castro Monteiro e Myriam Alhadeff Monteiro, e minha irmã, Camila Alhadeff Monteiro, por me incentivar a me tornar a pessoa que sou hoje. A meus amigos, que estiveram comigo durante toda esta fase de graduação até hoje no projeto final, sempre me incentivando. A Frederico Abraham por ajudar a gerar os modelos. Obrigado a todos.

Resumo

A. Monteiro, Andrei. Celes, Waldemar. Método baseado em física para visualização 3D de modelos complexos. Rio de Janeiro, 2009. 58 p. Relatório Final de Estágio Supervisionado II – Departamento de Administração. Pontifícia Universidade Católica do Rio de Janeiro.

Este projeto tem por objetivo a utilização de métodos de simulação física, como Dinâmica com Restrição, para explodir modelos 3D complexos e assim auxiliar especialistas a entender os mesmos. O método usado para que no final obtenha um modelo de reservatório de petróleo separado por iso-volumes é a técnica de barras rígidas e o cálculo de suas forças aplicadas para determinar os pontos de quebra frágeis do modelo. Esta pesquisa introduz uma nova técnica jamais utilizada na visualização científica, tendo resultados positivos, porém ainda prematuros. No final do projeto concluiu-se que estas técnicas podem ser usadas em modelos científicos desde que se saiba aplicar a força correta no modelo e aumente de algum modo a eficiência do programa, como desenvolvê-lo na GPU.

Palavras-chave

Física, Explosão, Força, Restrição e Modelo.

Abstract

A. Monteiro, Andrei. Celes, Waldemar. Physics-based methods for 3D visualization of complex models. Rio de Janeiro, 2009. 58 p. Relatório Final de Estágio Supervisionado II – Departamento de Administração. Pontifícia Universidade Católica do Rio de Janeiro.

The objective of this project is to use physics-based methods such as Constraint Dynamics in order to obtain an “exploded” view of 3D models, so it is possible to help specialists understand them. The method used so that in the end could obtain a Reservoir model exploded on its volumes was the rigid bars technique and the calculation of its applied forces to determine the weak points of the model. Furthermore, this research introduces a new technique never before used in Scientific Visualization, having positive but premature results. At the end of the research it was concluded that these techniques can be used in scientific models, but having in mind that the correct force must be applied, along with a greater program efficiency, such as implementing it on the GPU.

Keywords

Physics, Exploded View, Force, Constraint Dynamics, and Model.

Sumário

1. Introdução.....	1
2. Estado da Arte.....	3
3. Objetivos.....	7
4. Modelo Físico.....	8
4.1. Simulação.....	12
4.2. Centro de Massa.....	13
4.3. Propriedades.....	13
4.4. Modelo Massa-Mola.....	14
4.5. Restrições.....	16
4.6. O Método do Gradiente Conjugado.....	17
4.7. Barra Rígida.....	18
4.8. Determinando as forças nas barras.....	21
4.9. Restrição Radial.....	21
4.10. Juntando as restrições.....	23
4.11. Forças Externas.....	24
4.12. Avanço no tempo.....	29
4.13. Explosão do Modelo.....	30
5. Atividades Realizadas.....	32
5.1. Método.....	32
6. Projeto e Especificação do Sistema.....	35
6.1. Processo de Simulação.....	35
6.2. Estruturas de Dados.....	36
6.3. Módulos desenvolvidos.....	39
7. Implementação e Avaliação.....	42
7.1. Planejamento e execução de testes funcionais.....	42
7.1.1. Modelo massa-mola.....	42
7.1.2. Modelo de Barras Rígidas com força radial.....	44
7.1.3. Modelo de Barras Rígidas com agente radial seguindo a Caixa Orientada (OBB).....	47
7.1.4. Modelo de Barras Rígidas com Restrição radial.....	50
7.1.5. Modelo de Barras Rígidas com o algoritmo de “três passadas” seguindo o centro da OBB.....	52
7.1.6. Modelo real.....	52
7.2. Comentários sobre a implementação.....	54
8. Considerações Finais.....	60
9. Referências.....	62
10. ANEXO I – Formulação de Restrições.....	63

Lista de Figuras

- Figura 1.** Visão explodida de peça mecânica.
- Figura 2.** Explosão por seleção.
- Figura 3.** Modelo real de reservatório de petróleo.
- Figura 4.** Volume confinado.
- Figura 5.** Esquema de forças em locais frágeis do modelo.
- Figura 6.** Modelo massa-mola.
- Figura 7.** Algoritmo do Gradiente Conjugado.
- Figura 8.** Barra rígida.
- Figura 9.** Cisalhamento no modelo.
- Figura 10.** Restrição radial.
- Figura 11.** Agente externo radial.
- Figura 12.** Caixa orientada, ou *OBB*.
- Figura 13.** Contra-exemplo para agente seguindo a *OBB*.
- Figura 14.** Esquema algoritmo de “três passadas”.
- Figura 15.** Processo de simulação.
- Figura 16.** Estruturas de dados inicial.
- Figura 17.** Matriz Jacobiana.
- Figura 18.** Estruturas de dados final.
- Figura 19.** Modelo 1 antes da explosão (massa-mola).
- Figura 20.** Modelo 1 depois da explosão (massa-mola).
- Figura 21.** Modelo 1 explodido (barra-rígida) sem considerar a distância.
- Figura 22.** Modelo 1 explodido considerando a distância proporcional.
- Figura 23.** Modelo 1 explodido considerando a distância inversamente proporcional a distância.
- Figura 24.** Modelo 1 confinado a *OBB*.
- Figura 25.** Modelo 2 confinado a *OBB*.
- Figura 26.** Modelo 1 explodido seguindo a *OBB*.
- Figura 27.** Modelo 2 explodido seguindo a *OBB*.
- Figura 28.** Modelo 1 aplicado a restrição radial e de barras rígidas.
- Figura 29.** Modelo 1 explodido seguindo o algoritmo de “três passadas”.
- Figura 30.** Modelo 2 explodido seguindo o algoritmo de “três passadas”.
- Figura 31.** Modelo 3 explodido seguindo o algoritmo de “três passadas”.
- Figura 32.** Camada do modelo Namorado, no programa Geresim, do TecGraf.
- Figura 33.** Camada Namorado explodido por propriedades.
- Figura 34.** Camada Namorado explodido (1).
- Figura 35.** Camada Namorado explodido (2).
- Figura 36.** Modelo Namorado 3D no programa Geresim, do TecGraf.
- Figura 37.** Modelo Namorado 3D.
- Figura 38.** Modelo Namorado 3D explodido.

1. Introdução

Atualmente, pesquisadores necessitam de ferramentas poderosas para aumentar a qualidade e eficiência de trabalho. Quando se fala em visualização 3D, essas ferramentas são ainda mais relevantes, simplificando um modelo complexo e assim auxiliando o profissional em sua compreensão. Um dos ramos da Computação Gráfica é a Visualização Científica, que tem por objetivo a criação de imagens visuais que auxiliam na compreensão de modelos científicos complexos que contém vastas representações numéricas.

Para este trabalho, foi identificado o problema de visualização de modelos complexos, que podem ser simplificados e separados por propriedades utilizando métodos baseados em física, e assim ajudando na exibição por propriedades de objetos volumétricos, a fim de explorar sua estrutura e facilitar sua compreensão.

Modelos 3D complexos como ferramentas mecânicas e dispositivos eletrônicos são compostos de várias partes. Para compreender a estrutura interna desses objetos, especialistas criam visões explodidas cujas partes são separadas (ou explodidas) entre si a fim de revelar componentes que os interessam. A explosão bem ilustrada não só expõe os conceitos internos do objeto, mas também a estrutura global do modelo assim como a relação espacial entre partes. Por exemplo, um modelo de reservatório de petróleo, na qual contém vasta representação numérica, pode ser explodido por propriedades como aquíferos. Diferente de outras técnicas de ilustração que revelam o interior do objeto como cortes ou transparência, visões explodidas mostram a fundo e em detalhes as partes individuais.

Contudo, métodos tradicionais de explosão contêm várias limitações que tornam difícil para especialistas navegarem a estrutura e estudar os diferentes “sets” de partes dos modelos. Como a maioria das visões explodidas expõe todas as partes do objeto, um problema encontrado é o de “clutter”, ou desordem ou amontoamento. Como consequência disso, o usuário precisa inspecionar cuidadosamente todo o modelo para localizar as partes de interesse. Além disso, as partes que se encontravam próximas podem acabar distantes entre si após a explosão, tornando difícil para o usuário determinar como essas frações estavam posicionadas e orientadas em relação ao resto do modelo. Além disso, as explosões não permitem que os usuários analisem as relações espaciais em

diferentes níveis de detalhes. Por exemplo, um usuário pode querer ver como duas sub-partes se encaixam antes de examinar suas partes constituintes.

As explosões são muito utilizadas para facilitar a ilustração e estudo de um modelo, porém a simulação física é uma técnica pouca utilizada nas visões explodidas. Apesar de consumirem maior tempo de processamento, a simulação física pode garantir maior precisão e facilidade ao trabalhar com modelos complexos.

Neste trabalho serão apresentados métodos baseados em física, provendo uso de integradores numéricos, para explosão de modelos complexos a fim de melhor visualizá-los em função de suas propriedades. No caso de Visões Explodidas, estas auxiliam na compreensão da estrutura interna dos modelos. O projeto desenvolverá novas técnicas baseadas em métodos pré-existentes e será incorporado em um sistema. A demanda de evolução foi gerada a partir da necessidade de um melhor entendimento de modelos complexo pelos usuários-alvo, mais especificamente Engenheiros que trabalham na área de Petróleo, visualizando e gerenciando modelos de reservatório de petróleo.

Sobre o ambiente tecnológico de desenvolvimento, o projeto será desenvolvido provendo uso do sistema operacional Windows, da Microsoft, a linguagem de programação C++ e as bibliotecas de rotinas gráficas e de modelagem do OpenGL para visualização dos modelos, além das bibliotecas IUP para interface gráfica.

2. Estado da Arte

Para entender melhor a utilização das visões explodidas em modelos, é importante mencionar o ramo da Computação Gráfica da qual pertencem: a Visualização Científica. Como foi dito anteriormente, a Visualização Científica envolve visualizações de fenômenos em 3D como sistemas meteorológicos, de ciências biológicas, ou até de engenharia, com ênfase em *rendering* realística de volumes, superfícies, iluminação, entre outros. Este ramo foca principalmente no entendimento de conceitos científicos que em sua maioria são representados provendo uso de muitos dados numéricos. Outros exemplos em que essa vasta representação de dados pode ser usada para representar conceitos científicos incluem simulação de fluidos, colisão entre moléculas, estudos geológicos e visualização médica. Pode-se dizer que a Visualização Científica não está em seu fim, pelo contrário. Cientistas até hoje visualizam dados para encontrar padrões, traços, relações e anomalias, ou seja, para ajudar em seu entendimento.

Uma conferência sobre Visualização Científica promovido pela ACM SIGGRAPH (Association for Computing Machinery's Special Interest Group on Graphics and Interactive Techniques) em 2007 ressaltou os princípios e aplicações para o ramo [2]. Os conceitos apresentados foram visualização, percepção humana, métodos científicos e vários aspectos de dados, como classificação e armazenagem. As técnicas de visualização determinadas são 2D, 3D e multidimensionais, como visualizações de gases e fluidos, transformações de cores, renderização volumétrica, animação e técnicas para ambientes virtuais.

A seguir são apresentadas algumas áreas da Visualização Científica, com uma breve descrição sobre cada uma delas:

- a) **Animação por computador** (Computer animation): é técnica usada para criar imagens em movimento via meios de computação gráfica 3D ou 2D. Também pode ser usada para filmes. É também referida como CGI (ou Computer-generated imaging).
- b) **Simulação por computador** (Computer simulation): é um programa na qual simula um modelo abstrato de um sistema. É uma área de extrema relevância, ainda mais quando se fala de modelagem matemática de

sistemas físicos, sistemas humanos e no processo de engenharia e novas tecnologias.

- c) **Visualização de informação** (Information visualization): é o estudo da representação visual de coleções em grande escala e informações não-numéricas, como arquivos e linhas de código em softwares.
- d) **Renderização de superfícies** (Surface rendering): é o processo de geração de uma imagem a partir de um modelo através de programas. Algumas técnicas de renderização incluem “Ray casting”, “Radiosity” e “Raytracing”.
- e) **Visualização volumétrica** (Volume visualization): É a técnica utilizada na visualização de dados científicos associados a regiões de volume. Como exemplo pode-se citar a simulação de fluidos.

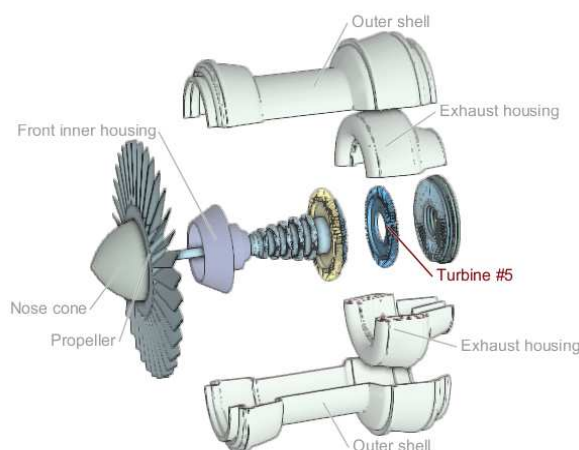


Figura 1. Esse exemplo mostra como a explosão é útil para ilustrar peças mecânicas, nesse caso um modelo de turbina.

A oclusão é um problema de extrema relevância quando se renderiza dados 3D na visualização científica. Por causa da oclusão, nem todos os dados podem ser mostrados simultaneamente. Frequentemente, o usuário deseja estudar o objeto de interesse que se encontra dentro do conjunto de dados volumétricos. Muitas vezes o usuário está interessado em explorar o objeto em questão levando em conta o conjunto inteiro. Para resolver este problema de oclusão, o contexto pode ser atribuído de uma maneira visual mais esparsa, como ter sua opacidade reduzida. Cortes e técnicas de transparência são usadas com este objetivo. Porém, o lado negativo dessas técnicas é que mesmo

assim alguma parte da informação relativa ao contexto é eliminada. Quando isso acontece, especialistas usam a técnica de explosão.

Em uma explosão, um objeto é decomposto em diversas partes que são posicionadas tal que detalhes internos estejam visíveis. Além de fornecer uma visão não obstruída, essa técnica revela outras informações interessantes, como seções transversais do modelo.

O conceito de “cortar” as partes do volume a fim de revelar a estrutura interna é comum na visualização científica. Quase todo volume renderizado impõe operações de “clipping”, ou recorte. Existem diversos trabalhos que apresentam diferentes formas de explodir um modelo. Algumas delas, que utilizam corte, são apresentadas a seguir:

- Operações de “clipping” interativo usando ‘geometria arbitrária’ para superar as limitações de planos de cortes.
- Planos de cortes deformáveis para ‘resseção’ virtual (“virtual resection”).
- Sistemas para modelagem e ilustração de volumes utilizando texturas de cortes baseado em modelos de superfícies.
- Conceito de funções espaciais de transferência para modelar deformações em volumes.
- Uso de descontinuidades, ou seja, partição de volume.

Algumas técnicas utilizam o conceito de “curve-skeleton”, que indica uma representação volumétrica reduzida que é gerada usando técnicas como “volume thinning”:

- Sistema interativo que permite edição manual de deformações de volumes baseados em um esqueleto (“skeleton”). Este trabalho foi estendido introduzindo uma “renderização seletiva” de componentes para aperfeiçoar a visualização.

Em um artigo publicado pela IEEE em 2006 [1], foi apresentado um algoritmo que utiliza forças de arranjos que automaticamente organizam as partes de uma explosão a fim de revelar o objeto em foco. Nesta técnica também é utilizado um algoritmo de *rendering* volumétrico baseado na GPU, na qual se distingue dois objetos derivados dos dados volumétricos: a seleção e *background*. A seleção é o objeto corrente em foco especificado pelo volume

selecionado. Tudo que não está selecionado faz parte do *background*, ou fundo, na qual representa o contexto. Segmentos do objeto *background* passam por transformações enquanto a seleção permanece estática. O espaço coberto pelo *background* é um número de partes que não se cruzam e são definidas por suas transformações e geometrias. Em resumo, pode-se gerar uma visão explodida do volume, onde as partes de *background* se afastam do objeto em foco, revelando a seleção. Um exemplo é mostrado na Figura abaixo.

Todos os métodos apresentados acima se diferem deste último publicado pela IEEE, que utiliza o conceito de força para explodir um modelo. No contexto atual, a simulação física é pouco utilizada quando se trata da visão explodida de um objeto. Neste projeto, a física será essencial para obter a visão explodida de um volume uma vez que o autor se proverá de métodos como restrição para separá-lo de acordo com suas propriedades.

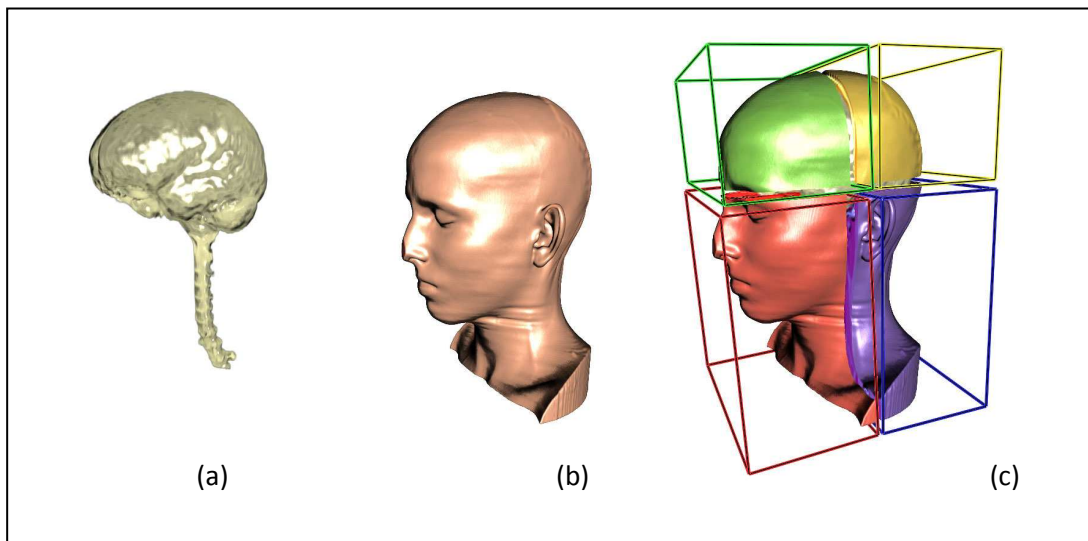


Figura 2. A figura mostra a explosão feita em cima de um modelo médico utilizando o conceito de seleção (a) e *background* (b). (c) mostra o objeto de *background* decomposto em quatro partes.

3. Objetivos

Este trabalho tem por objetivo a utilização de métodos baseados em física para a explosão por propriedades de modelos 3D complexos, para assim auxiliar usuários a melhor compreensão dos mesmos. Para isso, o autor se proverá de métodos baseados em simulações físicas utilizando Sistemas de Partículas. O escopo do sistema a ser alcançado é obter um modelo 3D de reservatório de petróleo explodido, aglomerando frações com propriedades em comum, tendo como meta auxiliar engenheiros em sua compreensão. No caso da peça mecânica explodida por partes como mostrado na Figura 1, ela se mostra uma explosão intuitiva buscada no projeto das partes. Porém, no caso do modelo de Reservatório de Petróleo tem-se uma explosão não intuitiva, o que leva a utilização de métodos físicos. O objetivo a ser alcançado é uma separação do modelo em iso-volumes, ou seja, volumes que distinguem em propriedades.

A solução proposta é utilização da Física Mecânica, representada por um Sistema de Partículas, na qual o modelo é representado por partículas que conectadas formam a superfície ou volume do modelo. Para conectar as partículas, será fundamental o uso de um sistema de barras rígidas. Além de métodos físicos, serão usados métodos numéricos para avaliar posições e velocidades no espaço, obtendo os efeitos visuais e físicos desejados. Também será considerado o problema de resolver um grande sistema de equações lineares devido ao método que será utilizado. As primeiras soluções a serem testadas serão em 2D, e caso tenha sucesso, será expandida para 3D. Após esses dois passos, o autor testará a explosão em um modelo real, tendo agora a preocupação com um número muito maior de dados. O modelo *Namorado*, que é considerado pequeno, contém aproximadamente 31 mil partículas e 80 mil barras. Portanto há uma preocupação de eficiência do programa quando esses modelos forem carregados, pois há muitas multiplicações de matrizes envolvidas, que são de ordem n^3 .

Para determinar os pontos de quebra entre células com propriedades distintas, basta verificar essa variação de propriedades, o que é trivial. O problema se dá quando existe um conjunto “fechado” de células com propriedades semelhantes na qual é necessário determinar seus pontos de quebra em locais frágeis. Esse é a principal motivação de se usar as forças nas barras rígidas, o que não é trivial.

4. Modelo Físico

Durante o desenvolvimento do projeto, o autor testou diversas técnicas novas para chegar ao resultado desejado. Para tal, foi preciso elaborar uma seqüência de etapas que no final compunham a simulação física para a explosão do modelo. Apesar de a meta principal ser em 3D, o autor passou a maior parte do tempo trabalhando em cima de testes com modelos 2D devido a menor complexidade (dois graus de liberdade), melhor visibilidade para enxergar os resultados e menos *visual clutter*, ou seja, menos desordem ou amontoamento.

O modelo de reservatório de petróleo que será explodido é dividido no que é chamado de células. Estas células contêm propriedades que serão usadas pelo autor para determinar a primeira etapa da explosão. Para realizar a segunda etapa da explosão, é preciso usar outro método além de verificar a variação de propriedades de células adjacentes. Esse método é o da restrição, e é usado para determinar pontos de quebra de um modelo fechado como, por exemplo, na Figura 4. Contudo, foi testado outro método para determinar a quebra em pontos frágeis do modelo. Essa técnica é o de massa-mola, que ao invés de ligar duas células por uma barra, liga-se por uma mola. Caso a mola estique muito, pode-se determinar uma quebra entre as duas partículas.

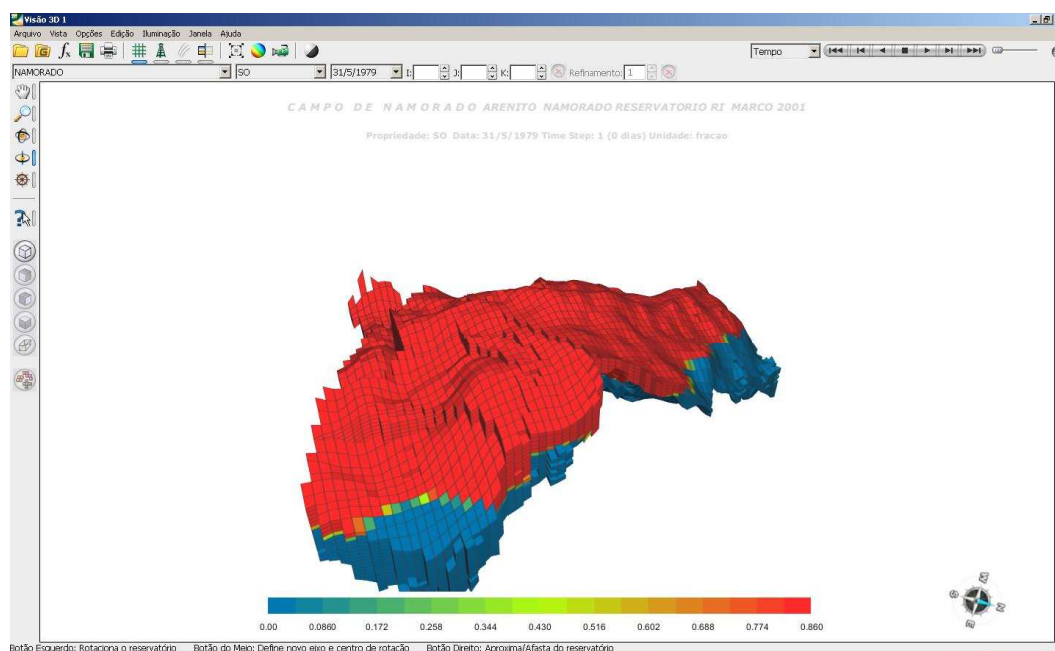


Figura 3. Modelo de Real de um reservatório de Petróleo.

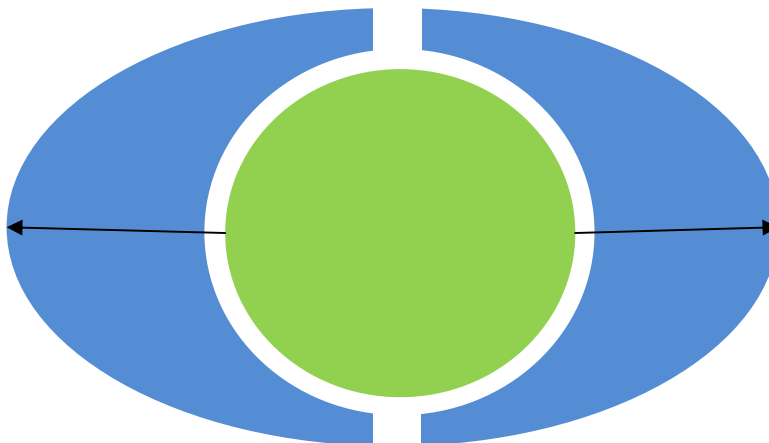


Figura 4. Para um volume confinado, é necessário quebrar o volume envolvente nos locais mais frágeis.

Caso seja necessário quebrar o modelo devido a casos como o da Figura 4, para determinar esses pontos de quebra será adotada uma restrição de barras rígidas no modelo. Calculando as forças nas barras, é intuitivo pensarmos que se houver mais massas em um dos lados do modelo, elas exercerão mais força, fazendo as áreas com menos massas estarem sujeitas a um maior módulo de forças. Essas áreas são mais frágeis que as áreas com maior densidade de massas conectadas por barras. Veja na figura abaixo. Será discutido mais tarde como encontrar essas forças de restrição.

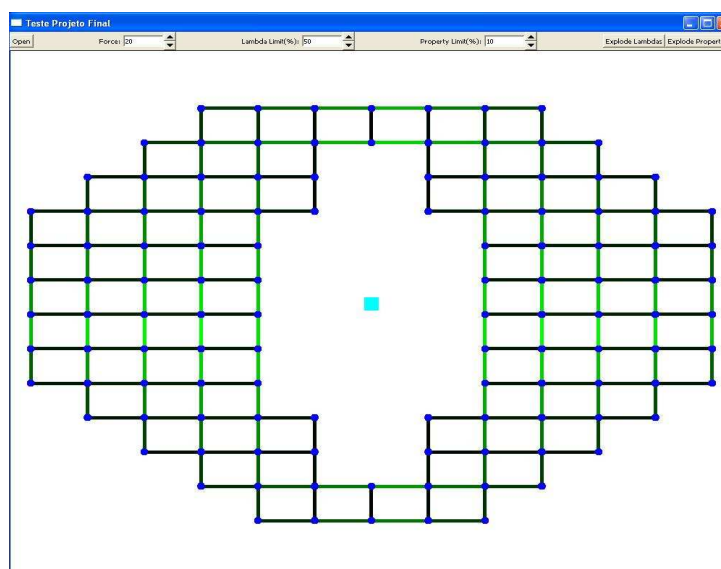


Figura 5. Esquema das maiores forças pintadas de verde claro e as mais fracas, escuras. Note que a conexão das maiores forças se encontram próximo aos eixos x e y do modelo.

Dado as células do reservatório de petróleo, é preciso uma maneira de representá-las no modelo físico. Como as barras rígidas serão relacionadas com as células e de que modo será exercida a explosão sobre elas? Para obter-se a saída desejada, precisaremos de algumas informações de entrada:

- Número total de células,
- O centro de cada célula,
- Informações de propriedades de cada célula,
- Índices de células adjacentes,
- E o número total de células adjacentes.

Usaremos um sistema de partículas para representar o modelo físico, de tal modo que o centro de cada célula é uma partícula representada por uma posição, velocidade e sujeita a uma força. Esta partícula também conterá propriedades de sua célula. Os índices de células adjacentes e o número total de células adjacentes serão usados para construirmos o modelo físico de barras rígidas, visto que uma barra conecta duas partículas de células adjacentes. Falta apenas a massa para completar os dados de entrada. Esta não faz diferença, pois vamos considerar que todas as partículas têm a mesma massa. O modelo original conterá um centro de massa antes de ser quebrado, mas após a quebra vale ressaltar que o modelo pode conter vários centros de massa.

O modelo todo estará sujeito a uma força de explosão que será discutido mais a frente. Mas é importante ressaltar que esta força será ajustada pelo usuário. Ele também poderá ajustar a força “limite” que será usada para controlar quais barras serão quebradas na etapa de quebra de propriedades e na quebra de barras.

Após a explosão do modelo físico, é preciso determinar o deslocamento dos componentes do modelo real, ou seja, em que direção eles se deslocarão para que não haja sobreposição de componentes. Para isso, trataremos o modelo (ainda ligado por barras, mas já quebrado) como um grafo. As partes quebradas que não contém ligação entre elas são os componentes conexos do grafo. Cada componente terá seu respectivo centro de massa que ajudará a descobrir a direção para qual ele se deslocará. Isso será discutido mais a frente.

Para desenvolvermos as equações de restrições é importante lembrar algumas notações matemáticas. Por exemplo, seja uma variável vetorial que descreve a posição de uma partícula dada por \vec{x} . Dizemos que $\dot{\vec{x}}$ descreve a sua derivada em relação ao tempo, nesse caso a velocidade \vec{v} . A mesma coisa vale para a aceleração, que é equivalente a $\ddot{\vec{x}}$. No capítulo de restrição trabalharemos

com vetores e matrizes. Chamaremos o vetor \mathbf{q} de vetor contendo todas as posições das partículas e o vetor $\dot{\mathbf{q}}$ contendo todas as velocidades das partículas. O mesmo vale para matrizes. Dizemos que a derivada de uma matriz \mathbf{J} é $\dot{\mathbf{J}}$, a matriz jacobiana.

$$\mathbf{q} = \begin{bmatrix} x_{ax} \\ x_{ay} \\ x_{az} \\ x_{ax} \\ x_{ay} \\ x_{az} \\ \vdots \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} v_{ax} \\ v_{ay} \\ v_{az} \\ v_{ax} \\ v_{ay} \\ v_{az} \\ \vdots \end{bmatrix}$$

Durante o desenvolvimento do sistema, o autor testou diversas maneiras de explodir o modelo. Foram elas:

- Modelo Massa-Mola, na qual ao invés de conectar as partículas por uma barra rígida, conecta-se por uma mola, “deixando” que as partículas se afastem uma das outras, mas não muito. Para tal, usou-se um coeficiente de rigidez de 100 e um coeficiente de dissipação de 120. O resultado de sua explosão é muito bom, porém, como veremos mais afrente, é necessário muitos passos de simulação, dependendo do tamanho do modelo.
- Modelo de Barras Rígidas aplicada a uma força radial, na qual a origem da força é o centro de massa de cada componente conexo do modelo e ela atua sobre todas as partículas com o mesmo módulo, independente da distância do ponto ao centro de massa. Essa força faz com que as barras imponham uma força de reação, sendo esta necessária para explodir o modelo. Também foram usadas nesse modelo a força radial que leva a distância em consideração, podendo ser proporcional ou inversamente proporcional a distância da partícula ao centro de massa de seu componente conexo.
- Modelo de Barras Rígidas com restrição radial, na qual além de impor uma restrição de barras no sistema, também é imposto uma restrição radial, fazendo com que as partículas só andem radialmente, evitando assim um problema que é muito comum na restrição de barras: o cisalhamento.
- Modelo de Barras Rígidas seguindo a Caixa Orientada, ou *Orienting Bounding Box*. Nela, a força aplicada que resultará nos cálculos das forças das barras será sempre aplicada na direção do maior eixo da Caixa Orientada, que é a direção de maior densidade de partículas,

“forçando” uma quebra nos locais mais frágeis do modelo. Mais tarde veremos que nem sempre essa direção é a mesma da direção dos locais com maior densidade.

- Algoritmo de “três passadas”, na qual é preciso calcular as forças nas barras três vezes, quebrá-las e depois aplicar a força externa no modelo para então avançar no tempo. Foi uma técnica intuitiva que se mostrou bastante eficiente quanto à explosão do modelo.

4.1. Simulação

Para explodirmos o modelo, é necessária uma seqüência de eventos que compõe um passo de simulação. Com esses eventos, geralmente estamos atrás do tempo real. O problema do tempo real é que o passo da simulação, ou seja, a diferença de tempo pode ficar muito grande. Por isso utiliza-se um passo fixo no integrador numérico. O passo fixo usado para a simulação deste projeto Foi de 0.005 para mola e 0.0005 para as restrições. A simulação contém basicamente os seguintes eventos: força aplicada, restrição, quebra, agente externo aplicado, restrição e movimento.

Para a simulação, segue-se uma seqüência de operações para atualizarmos as posições e velocidades finais das partículas. A simulação compõe os seguintes passos:

- 1.Exploda o modelo em função de suas propriedades, caso não o tenha feito;
- 2.Caso tenha explodido no passo anterior, recalcule os centros de massa de todos os componentes;
- 3.Adicione no sistema a força externa que fará com que o modelo se quebre;
- 4.Calcule as forças de restrição;
- 5.Quebre o modelo;
- 6.Adicionam-se os agentes externos no modelo que resultará em seu movimento, ou seja, para toda partícula adicione as forças de explosão;
- 7.Calculam-se as forças de restrição no sistema;
- 8.Avance no tempo, atualizando as posições e velocidades das partículas, gerando um movimento no modelo.
- 9.Caso a quebra não foi o suficiente para se ter uma boa visualização, repita os passos de simulação.

4.2. Centro de Massa

Durante várias etapas da simulação do modelo usaremos o conceito de centro de massa, seja para calcular o deslocamento relativo de componentes conexos do modelo ou para calcular a origem dos agentes externos, que será discutido mais a frente. A definição de centro de massa segue abaixo:

$$\vec{c} = \frac{\sum_{i=0}^n \vec{r}_i m_i}{M}$$

Onde \vec{r}_i é o vetor posição da partícula relativa à origem do eixo local, m_i é a massa da partícula, e M a massa total do modelo. Como é considerado que todas as partículas têm a mesma massa, o cálculo do centro de massa fica da seguinte forma:

$$\vec{c} = \frac{\sum_{i=0}^n \vec{r}_i}{n}$$

Onde n é o número total de partículas.

4.3. Propriedades

Foi mencionado anteriormente que cada célula do modelo de reservatório de petróleo contém uma série de propriedades, como aquíferos e vazão. Estaremos interessados nessas propriedades para executar a primeira fase de explosão. Nesse caso, a força limite com que a barra será quebrada é proporcional a uma variação de propriedade entre as duas células adjacentes. Para modelos não reais, as propriedades foram simuladas com um valor pertencente ao intervalo [0,1]. O usuário ajustará um valor que determina o valor limite para a barra quebrar considerando a variação de propriedade. Por exemplo, se o valor for 10, divide-se o intervalo [mínima propriedade, máxima propriedade] em 10, obtendo assim um conjunto de intervalos uniformemente divididos. Caso a propriedade da célula 1 estiver em um intervalo e a propriedade da célula adjacente 2 estiver em outro intervalo, a barra é quebrada. A equação segue abaixo:

$$Limit = \frac{|max - min|}{n}$$

Onde *max* e *min* são as propriedades máximas e mínimas do modelo respectivamente, e *n* o número de divisões que for desejado. Se a propriedade 1 e a propriedade 2 estiverem em intervalos diferentes, a barra será quebrada, onde o intervalo é calculado da seguinte forma:

$$interval = \left\lfloor \frac{property - property_{minimum}}{Limit} \right\rfloor$$

4.4. Modelo Massa-Mola

Além do método de restrição foi testado outro método de massa-mola, na qual duas partículas são conectadas por uma mola, ao invés de uma barra rígida. Essa é outra tentativa de encontrar pontos de quebra em locais frágeis do modelo. A diferença está não só na peça que conecta ambas as partículas, mas também ao considerar o comprimento da mola para quebrar o modelo. Ao invés de considerarmos força na mola para a quebra, considera-se o deslocamento relativo entre as duas partículas, ou seja, o comprimento atual sobre o comprimento de repouso da mola é a equação que será estudada para determinar uma quebra.

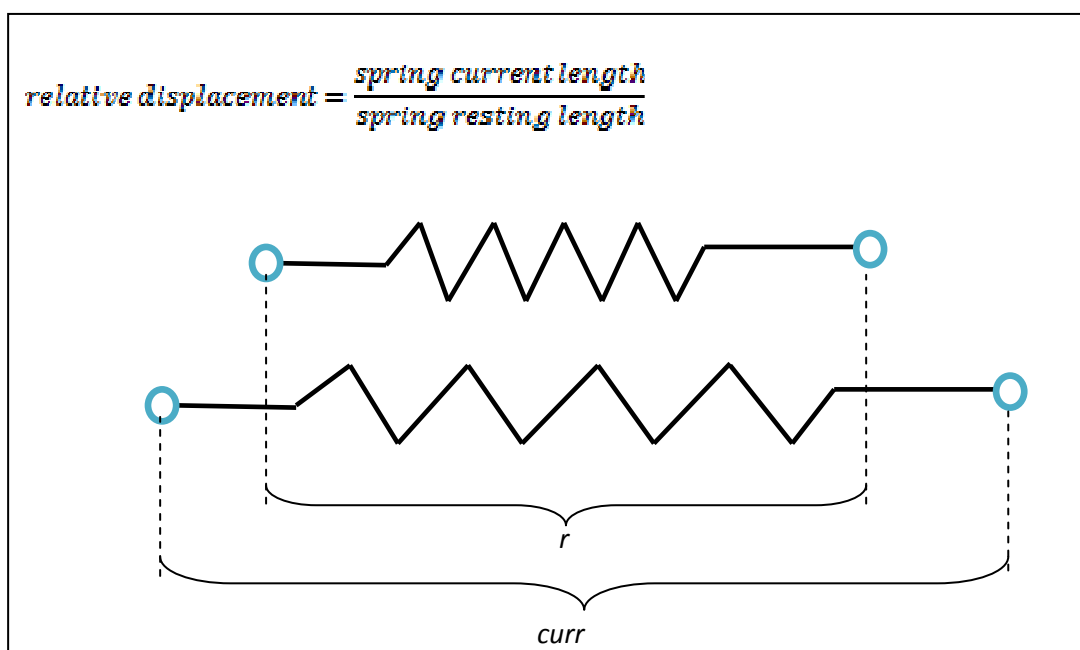


Figura 6. Mola e a relação comprimento corrente – comprimento de repouso.

O movimento de duas partículas conectadas por uma mola é descrito pela *Lei de Hooke*, e é dada por:

$$m\ddot{x} = -k_s \Delta x$$

onde k_s é o coeficiente de rigidez da mola. Mais especificamente, as forças em ambas as partículas se dá da seguinte forma:

$$\begin{aligned}\vec{F}_i &= k_s(x_j - x_i) \\ \vec{F}_j &= -k_s(x_j - x_i)\end{aligned}$$

O oscilador harmônico vai continuar o movimento infinitamente se não houver interferência de agentes externos. Isso é não realístico, já que no mundo as molas perdem energia e vão desacelerando. Essa dissipação de energia é usualmente modelada por uma força dissipativa, ou *damping viscosity force*:

$$\vec{F}_{damping} = -k_d \dot{x}$$

O que implica:

$$m\ddot{x} = -k_s x - k_d \dot{x}$$

A equação final para mola com amortecimento é usada na simulação do modelo:

$$\vec{F}_a = -\vec{F}_b = -\left[k_s(|\vec{l}| - r) + k_d \vec{l}\dot{l}\right] \hat{l}$$

Onde $\vec{l} = \vec{x}_a - \vec{x}_b$, $l = x_a - x_b$, r é o comprimento de repouso da mola, k_s, k_d são os coeficientes de rigidez e dissipação da mola, respectivamente.

Com essa técnica, é de esperar que as molas se estiquem mais em lugares mais frágeis do modelo, determinando os pontos de quebra.

4.5. Restrições

A principal etapa no projeto envolve o conceito de restrições, uma vez que as células do modelo são representadas por partículas ligadas por uma barra rígida. É preciso encontrar uma forma de calcular as forças nas barras, pois uma vez encontradas, permitem ajudar a achar os pontos de quebra do modelo. O método de restrições explica como encontrá-las.

Existem dois métodos de restrição: O método da “penalidade”, na qual se usa forças de restituição para restaurar as configurações invalidadas, e as forças de restrição, na qual será usada nesse projeto.

As forças de restrição impedem que o sistema adquira uma aceleração ilegal. O objetivo é manter uma função de comportamento igual a zero durante toda a simulação. Sendo \vec{f}_e as forças de restrição e \vec{f} as forças externas, tem-se a seguinte equação de movimento:

$$\vec{f} + \vec{f}_e = m\ddot{x}$$

Só precisamos achar as forças de restrição e somar com os agentes externos para assim obtermos o movimento total das partículas.

Para se formular a Restrição é necessário definir uma função Comportamento que impedirá que a partícula adquira uma aceleração ilegal.

As partículas tenderão a sair da restrição imposta sobre elas devido a imprecisões numéricas. Portanto, coloca-se uma “mola” para garantir estabilidade. A expressão final da restrição generalizada é:

$$JWJ^T\lambda = -\dot{J}\dot{q} - JWf - k_s C - k_a \dot{C}$$

Onde:

C é a função comportamento, tendo que ser igual a zero durante toda a simulação,

J é a matriz jacobiana,

$$\frac{\partial C}{\partial q} = \begin{bmatrix} \frac{\partial C_1}{\partial q_1} & \frac{\partial C_1}{\partial q_2} & \dots \\ \frac{\partial C_2}{\partial q_1} & \frac{\partial C_2}{\partial q_2} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

W é a matriz de inversa de massas, $W = \begin{bmatrix} \frac{1}{m_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{m_n} \end{bmatrix}$

λ são os multiplicadores de lagrange,

J é a derivada da matriz Jacobiana, $J = \frac{\partial \dot{C}}{\partial q}$

\dot{q} é o vetor de velocidades das partículas,

k_s é o coeficiente de rigidez da “mola”,

k_d é a constante de dissipação da “mola”,

\dot{C} é a derivada da função comportamento.

A formulação completa da Restrição encontra-se no Anexo I.

4.6. O Método do Gradiente Conjugado

No final, a única incógnita que precisa ser encontrada é λ , os Multiplicadores de Lagrange, para no final encontrar as forças de restrição utilizando a expressão $f_r = J^T \lambda$. Isso se resume a resolver um sistema linear $Ax = b$, onde x são os Multiplicadores de Lagrange. Denotaremos λ como Multiplicadores de Lagrange. Para resolver tal sistema, utilizaremos um resolvidor de sistema linear chamado de *Gradiente Conjugado*. Para entender porque usar esse resolvidor é preciso lembrar duas características importantes da matriz $A = JWJ^T$ do nosso sistema:

1. É uma matriz muito esparsa, ou seja, cheia de zeros,
2. É uma matriz simétrica, ou seja, $A = A^T$.

A primeira característica faz com que o método do *Gradiente Conjugado* possa resolver o sistema em $O(n)$ passos, que diferentemente de outros resolvidores, como *Decomposição LU*, que pode chegar a $O(n^3)$. Isso pode ficar muito caro para o programa. O *Gradiente Conjugado* só funciona para matrizes simétricas, que é o caso da matriz A .

```

Algorithm ConjugateGradient (A, b, r)
direction = b - A*r
residual = direction
for i = 1:length(b)
    alpha = (r'*r)/(direction'*A*direction)
    New_r = r+alpha*direction
    New_residual = residual - alpha*A*direction
    beta = (New_residual'*New_residual)/(residual'*residual)
    direction = New_residual + beta*direction
    r = New_r
end
end algorithm

```

Figura 7. Algoritmo do Gradiente Conjugado.

4.7. Barra Rígida

Agora queremos de fato ligar duas partículas por uma barra rígida, de forma a não permitir que as partículas se separem umas das outras de seus comprimentos de repouso. Primeiro, é necessário definir a função comportamento:

$$C = |\vec{l}| - r = 0$$

Onde $\vec{l} = \vec{x}_a - \vec{x}_b$ e r é o comprimento de repouso da barra. Vale lembrar que temos que garantir que $C(t_0) = 0$.

Dada duas partículas ligadas por uma barra rígida, quer-se achar as matrizes e vetores relacionados a elas. Caso há mais de uma barra no modelo, que é o caso real, teríamos um vetor para funções de comportamento:

$$\vec{C} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \end{bmatrix}$$

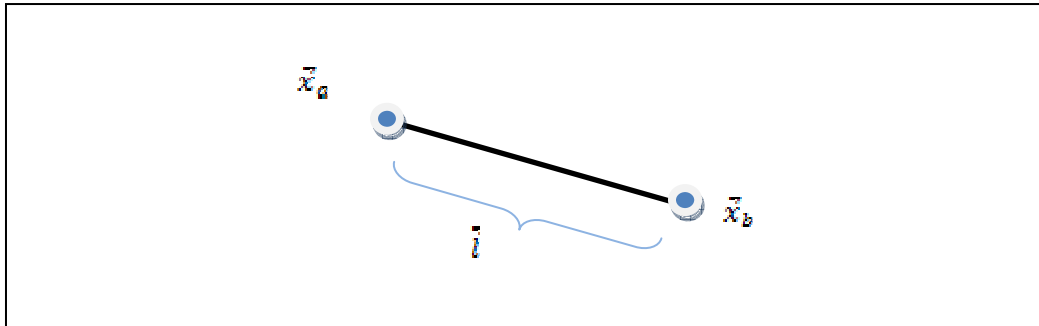


Figura 8. Barra rígida.

Também são dados os vetores de posição e velocidade das partículas:

$$\vec{q} = \begin{bmatrix} \vec{x}_a \\ \vec{x}_b \end{bmatrix}, \quad \dot{\vec{q}} = \begin{bmatrix} \vec{v}_a \\ \vec{v}_b \end{bmatrix}$$

Além da matriz da inversa da massa W .

A Matriz Jacobiana J é criada da seguinte forma:

$$J = \frac{\partial C}{\partial q} = \begin{bmatrix} \frac{\partial C}{\partial x_a} & \frac{\partial C}{\partial x_b} \end{bmatrix}$$

Onde $\frac{\partial C}{\partial x_a} = \left(\frac{\partial C}{\partial x} \quad \frac{\partial C}{\partial y} \quad \frac{\partial C}{\partial z} \right)$ e, pela regra da cadeia, $\frac{\partial C}{\partial x_a} = \frac{\partial C}{\partial l} \frac{\partial l}{\partial x_a}$.

$$\frac{\partial C}{\partial l} = \frac{\vec{l}}{|\vec{l}|}, \quad \frac{\partial l}{\partial x_a} = 1, \quad \frac{\partial l}{\partial x_b} = -1$$

Portanto:

$$J = \begin{bmatrix} \frac{\vec{l}}{|\vec{l}|} & -\frac{\vec{l}}{|\vec{l}|} \end{bmatrix}$$

Também temos a derivada da função comportamento dada por:

$$\dot{C} = \frac{\partial C}{\partial l} \dot{l} = \frac{\vec{l} \dot{l}}{|\vec{l}|}$$

Onde $\vec{l} = \vec{v}_a - \vec{v}_b$, a velocidade relativa entre as duas partículas.

Falta apenas a derivada da Matriz Jacobiana, dada da seguinte forma:

$$\dot{J} = \frac{\partial \dot{C}}{\partial l} \frac{\partial l}{\partial q}$$

Lembrando da regra de derivação de uma fração: $\frac{\partial}{\partial t} \left(\frac{f}{g} \right) = \frac{f'g - g'f}{g^2}$.

$$\frac{\partial \dot{C}}{\partial l} = \frac{(\dot{l} + \vec{l} \cdot 0)|\vec{l}| - \frac{\dot{l}}{|\vec{l}|} \vec{l} \cdot \vec{l}}{|\vec{l}|^2} = \frac{\dot{l}|\vec{l}| - \frac{\dot{l}}{|\vec{l}|} \vec{l} \cdot \vec{l}}{|\vec{l}|^2} = X$$

$$\frac{\partial l}{\partial x_a} = 1, \quad \frac{\partial l}{\partial x_b} = -1$$

Portanto, tem-se que:

$$\dot{J} = [X \quad -X]_{1 \times 2}$$

Basta resolver o sistema linear $Ax=b$,

$$JWJ^T \lambda = -\dot{J} \dot{q} - JWf - k_s C - k_d \dot{C}$$

E encontrando λ , aplica-se na equação

$$f_c = J^T \lambda$$

$$J^T = \begin{bmatrix} \frac{\vec{l}}{|\vec{l}|} \\ -\frac{\vec{l}}{|\vec{l}|} \end{bmatrix}$$

para achar as forças de restrição.

No caso de três ou mais partículas, pode-se escrever a matriz Jacobiana como:

$$J = \begin{bmatrix} \frac{\vec{l}_{ab}}{|\vec{l}_{ab}|} & -\frac{\vec{l}_{ab}}{|\vec{l}_{ab}|} & 0 & \dots \\ 0 & \frac{\vec{l}_{bc}}{|\vec{l}_{bc}|} & \frac{\vec{l}_{bc}}{|\vec{l}_{bc}|} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

4.8. Determinando as forças nas barras

O segundo cálculo de forças de restrição só será usado para avançar o sistema no tempo. Primeiro calcula-se as forças externas, depois se encontra as forças de restrição e adiciona-se no sistema para depois avançar no tempo, caso seja necessário mais passos para encontrar os pontos de quebra no modelo. As forças nas barras (i.e., o primeiro cálculo de forças de restrição no ciclo de simulação), porém, são determinadas pelos *Multiplicadores de Lagrange*, isso porque a força de restrição é proporcional a *lambda*, na direção da barra. Vale ressaltar que existirão *lambdas* positivos e negativos no sistema. Os negativos se dizem respeito de forças de compressão, que é o caso que estamos interessados, pois no momento em que uma barra pertencente a uma parte frágil do modelo é puxada, ela tenderá, em reação, a se comprimir para evitar que as partículas se afastem. Em contraposição, os *lambdas* positivos se dizem a respeito de forças de tração aplicadas sobre sua barra rígida. Contudo, nem sempre isso acontece. Por exemplo, no caso do algoritmo de “três passadas”, que será discutido mais afrente, os *Multiplicadores de Lagrange* são maiores que zero, isso porque somamos as os *lambdas*, tirando seu módulos.

4.9. Restrição Radial

Existe um problema da barra rígida: o cisalhamento. Contudo, ele pode ser contornado. Um triângulo de barras não cisalha; mas um quadrado ou mesmo um cubo podem cisalhar caso não haja barras em suas diagonais. Isso também depende da força aplicada sobre as partículas. Veja na figura abaixo.

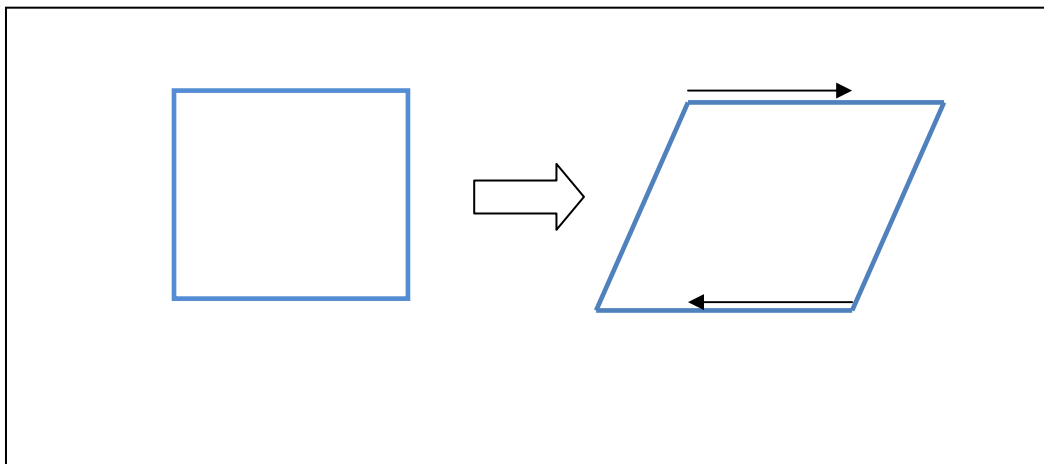


Figura 9. Cisalhamento no modelo.

Queremos que as forças nas barras em locais frágeis do modelo determinem o ponto de quebra com o mínimo de passos possíveis. Acreditamos isso possa acontecer quando o modelo não cisalhar. Para resolver esse problema de cisalhamento, introduz-se uma nova restrição, a radial. Dessa forma, além das partículas estarem ligadas por uma barra, elas só poderão andar radialmente, ou seja, na direção do vetor que vai do centro de massa de seu componente até a posição em que a partícula se encontra.

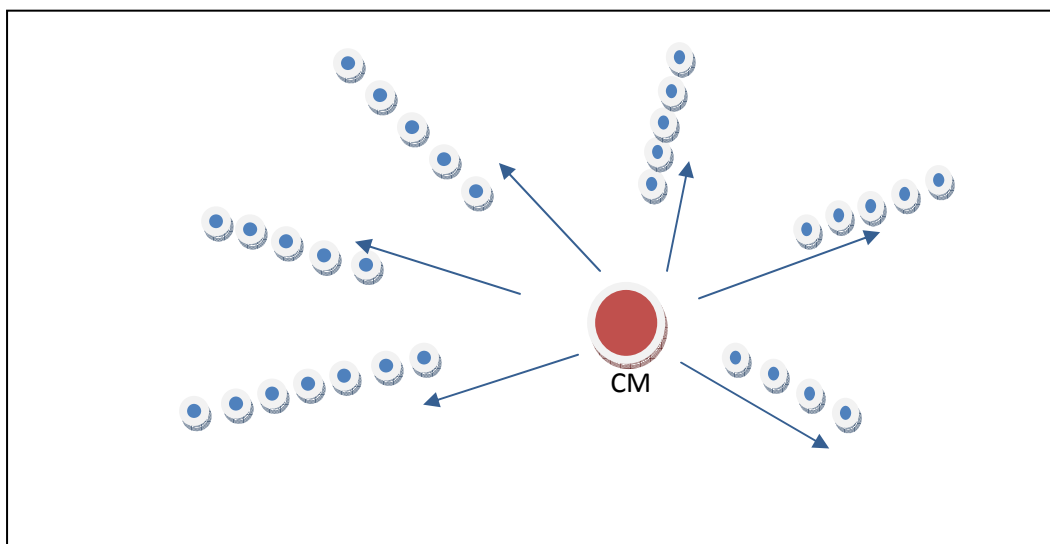


Figura 10. Partículas restritas a um movimento radial.

Define-se a função comportamento como:

$$c = (\vec{x}_a - \vec{c}) \cdot \hat{r}_{a\perp},$$

Onde \vec{x}_a é a posição da partícula, \vec{c} o centro de massa de seu componente, e $\hat{r}_{a\perp}$ é o vetor perpendicular ao vetor $\vec{x}_a - \vec{c}$, normalizado, antes de começar a simulação ou imediatamente depois de recalculer o centro de massa, caso o modelo tenha sido explodido anteriormente. Ou seja, se $\vec{x}_a - \vec{c} = (x, y, z)$ antes de começar a simulação, então $\hat{r}_{a\perp} = (y, -x, z)$. Esse produto escalar tem que ser zero, pois os vetores são perpendiculares.

Para o Jacobiano, derivada da função comportamento, e derivada do Jacobiano, têm-se, respectivamente:

$$J = \begin{bmatrix} \hat{r}_{a\perp} & 0 & 0 \\ 0 & \hat{r}_{b\perp} & 0 \\ 0 & 0 & \hat{r}_{c\perp} \end{bmatrix}$$

$$\dot{c} = \begin{bmatrix} \hat{r}_{a\perp} \vec{v}_a \\ \hat{r}_{b\perp} \vec{v}_b \\ \hat{r}_{c\perp} \vec{v}_c \\ \vdots \end{bmatrix}$$

$$\dot{J} = 0$$

O Jacobiano é zero pois a velocidade não depende da posição ao se calcular a derivada.

Apesar do problema de cisalhamento ter sido contornado, a restrição radial não ajudou em determinar as quebras nos locais mais frágeis, como será mostrado mais tarde na implementação de testes.

4.10. Juntando as Restrições

No momento em que se têm as restrições formuladas, pode-se juntá-las para satisfazer ambas ao mesmo tempo. Além de “prender” as partículas por barras rígidas, vamos restringir o movimento delas para que andem radialmente, assim evitando cisalhamento. Os vetores e matrizes ficam da seguinte forma:

$$\vec{C} = \begin{bmatrix} C_{11} \\ \vdots \\ C_{1n} \\ C_{21} \\ \vdots \\ C_{2m} \end{bmatrix}$$

Onde teremos n funções comportamento 1 (barra rígida), em que n é o número de barras no sistema, e m funções comportamento 2 (restrição radial), onde m é o número de partículas do sistema.

$$J = \begin{bmatrix} \frac{\partial C_1}{\partial x_a} & \frac{\partial C_1}{\partial x_b} & \dots \\ \vdots & \vdots & \dots \\ \frac{\partial C_2}{\partial x_a} & \frac{\partial C_2}{\partial x_b} & \ddots \end{bmatrix}$$

No final é obtido $n + m$ *Multiplicadores de Lagrange*, n lambdas para barras e m lambdas para movimento radial de partículas, calculando-se no final m forças de restrição, uma força para cada partícula. Vale lembrar que estamos apenas interessados nas forças da barra, portanto *não* usaremos todos os *Multiplicadores de Lagrange* para simular o modelo, mas apenas n lambdas para determinar as forças nas barras e consequentemente os pontos de quebra do modelo.

4.11. Forças externas

Na segunda fase de explosão, é necessário calcular as forças nas barras rígidas para determinar os pontos de quebra em locais frágeis do modelo. Para isso, é necessário aplicar antes uma força externa no modelo, que resultará numa reação das barras. O valor dessa força também será ajustado pelo usuário. Foram testados cinco tipos de forças externas: força radial, força radial proporcional à distância, força radial inversamente proporcional à distância, força radial seguindo o modelo de caixa orientada (*Orienting Bounding Box*) e força do algoritmo de “três passadas”. É importante não confundir essas forças com a força externa aplicada após a quebra que fará com que o modelo se movimente.

A primeira força é a mais simples. Esse agente externo atua como uma força radial, na qual seu módulo é ajustado pelo usuário e é o mesmo para cada partícula. Sua direção é um vetor do centro de massa do componente conexo

em que a partícula pertence até a partícula em si. Esse vetor é normalizado e multiplicado pelo módulo da força. O resultado se encontra na Figura abaixo.

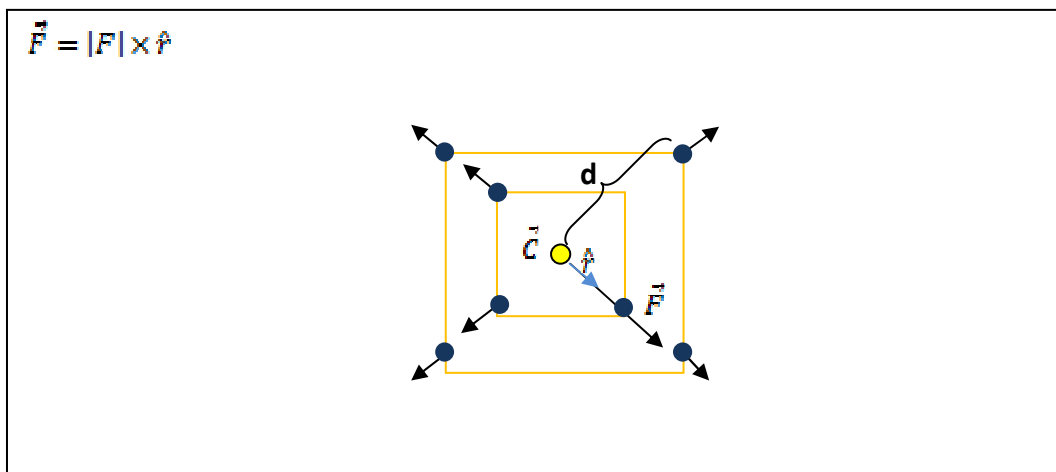


Figura 11. Agente externo aplicado radialmente.

Para evitar uniformidade no modelo, ou seja, um comportamento igual ou semelhante de cada partícula multiplica-se essa força radial pela distância de sua partícula ao centro de massa do seu respectivo componente conexo. Dessa forma, partículas mais longe do centro de massa estarão sujeitas a uma força maior. A equação ficaria da seguinte forma:

$$\vec{F} = |F| \times d \times \hat{r}$$

Também é possível usar o inverso da distância para calcular o vetor força. Desse modo, partículas mais perto do centro de massa estarão sujeitas a uma força maior:

$$\vec{F} = \frac{|F| \times \hat{r}}{d}$$

Talvez o resultado mais próximo ao desejado seja o da caixa orientada, ou *Orienting Bounding Box* (OBB). Essa caixa envolve todo o modelo, orientada de forma a acompanhar os lugares de maior densidade de partículas. Essa técnica é extremamente útil no nosso caso que queremos aplicar o agente externo na direção dos locais de maior densidade, à fim de concentrar maior força nos lugares mais frágeis. Primeiro calcula-se o centro geométrico da caixa:

$$\vec{c} = \frac{\sum_{j=0}^n v_j}{n}$$

A seguir, calcula-se o vetor do centro geométrico a partícula para cada partícula:

$$\vec{w}_j = \vec{v}_j - \vec{c}$$

Obtém-se uma matriz 3x3:

$$M_{3 \times 3} = \frac{1}{n} \times \sum_{j=0}^n w_j w_j^T$$

Os autovetores $\vec{u}_0, \vec{u}_1, \vec{u}_2$ dessa matriz dão as direções dos três eixos da *OBB*. Para encontrar o módulo desses vetores, calcula-se o produto escalar entre cada autovetor com todos os vetores w_j , tirando seu módulo e pegando o máximo e mínimo entre esses cálculos:

$$h_{i_{max}} = \max(u_i, w_j)$$

$$h_{i_{min}} = \min(u_i, w_j)$$

Encontrando os eixos, aplica-se em todas as partículas uma força na direção do maior dos eixos da *OBB*, pois essa direção contém a maior densidade de partículas.

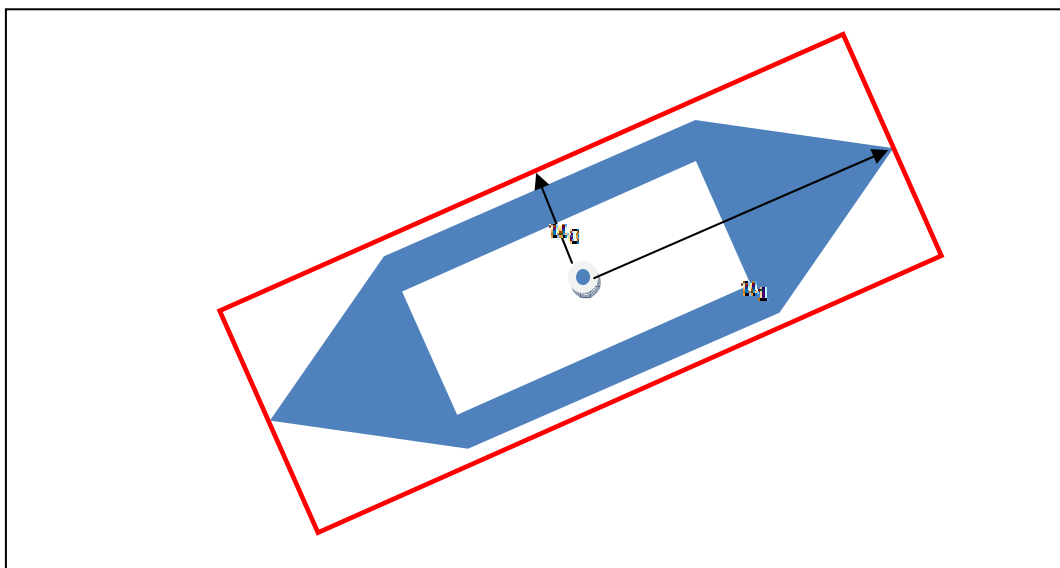


Figura 12. Modelo 2D confinado à sua caixa orientada, com seus respectivos eixos.

Existe um problema, no entanto, que faz com que os eixos de maior densidade nem sempre geram a força certa. A figura abaixo ilustra esse exemplo. Nela, pode-se observar que o eixo acompanha a direção de maior densidade de massas; porém, a força está sendo aplicada na direção que não se é desejada, ou seja, na direção dos pontos de quebra frágeis.

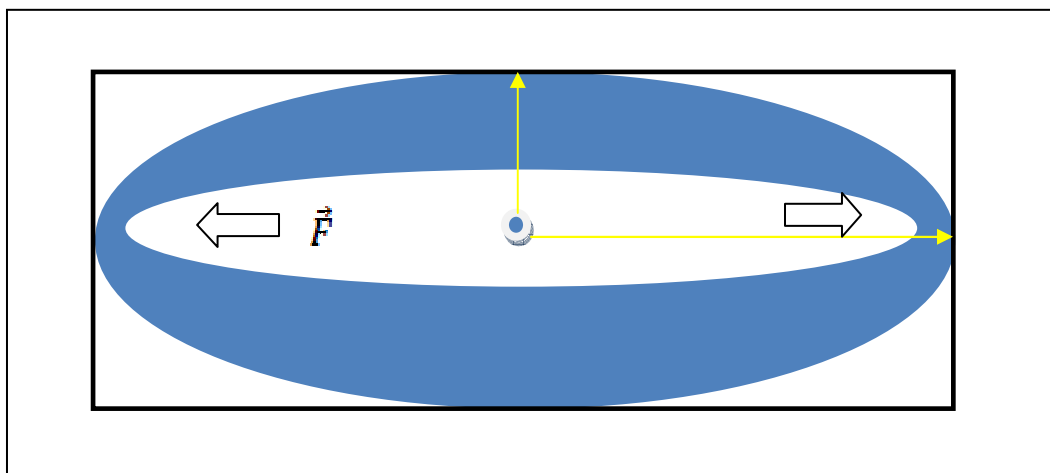


Figura 13. Contra-exemplo para a aplicação da força seguindo a caixa orientada.

Outro agente que foi testado foi seguindo um algoritmo de “três passadas”, isto porque é preciso calcular três vezes os Multiplicadores de Lagrange. Na primeira passada, aplica-se uma força restringida apenas no eixo x , onde:

- Partículas que têm valor na abscissa maior que o valor do centro de massa de seu componente em “x”, estarão sujeitos a uma força do tipo $(\vec{F}_x, 0, 0)$
- Partículas que têm o valor na abscissa menor que o valor do centro de massa de seu componente em “x”, estarão sujeitos a uma força do tipo $(-\vec{F}_x, 0, 0)$.

Após aplicar a força no eixo x, calcula-se os Multiplicadores de Lagrange. O passo é repetido para o eixo y, calculando-se os Multiplicadores de Lagrange, e para o eixo z.

Ao final dos três passos, teremos três conjuntos de Multiplicadores de Lagrange: ML_x , ML_y e ML_z . Para encontrar as forças na barra, substituímos na seguinte equação:

$$\vec{F}_i = |ML_{ix}\cos\theta_i + ML_{iy}\cos\varphi_i + ML_{iz}\cos\rho_i|$$

Onde θ, φ, ρ são, respectivamente, o ângulo que a barra faz com os vetores unitários i, j, k . A figura abaixo ilustra um exemplo 2D.

Ao invés de considerar o centro de massa como origem da força, podemos usar o centro da Caixa Orientada (OBB) como origem, verificando se os resultados obtidos serão mais razoáveis que os do centro de massa.

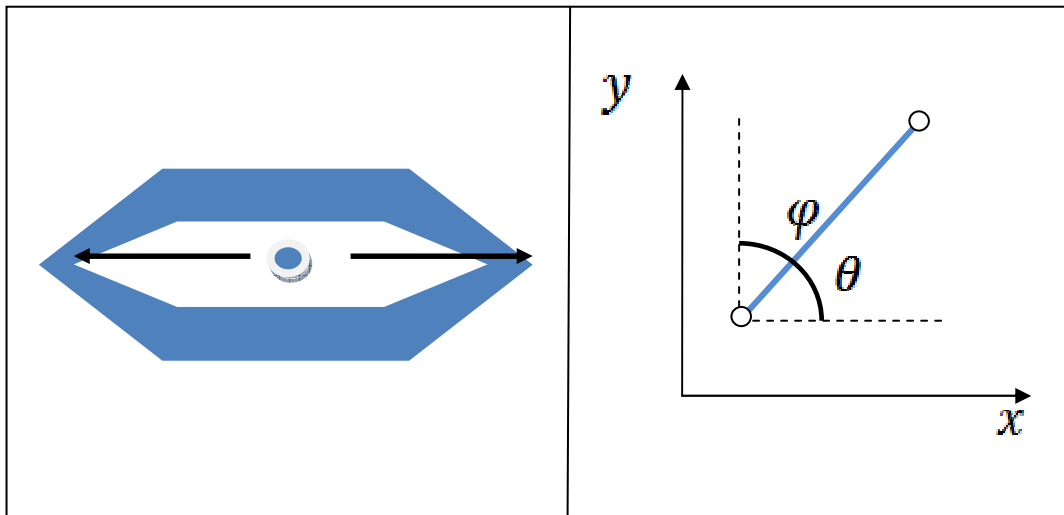


Figura 14. (a) Força aplicada comente no eixo x. (b) Projeção das força de restrição resultante do agente externo de x é feita seguindo o ângulo θ , enquanto a força de restrição resultante de y leva em consideração o ângulo φ .

4.12. Avanço no Tempo

Agora que as forças de restrição foram calculadas, é necessário atualizar as posições e velocidades das partículas após o cálculo das restrições do agente externo que fará com que o modelo se mova. Para isso, usa-se um integrador numérico. No nosso caso, foram testados dois: Método de Euler, e Euler Melhorado.

Para um sistema de partículas, assume-se que as posições e velocidades iniciais são conhecidas. O problema agora é um problema de valor inicial $\dot{p}(t) = F(p(t))$ imposta pela condição $p(t_0) = p_0 = [r_0, v_0]^T$, onde queremos resolver para um dado $t > 0$. A maioria dos problemas de valor inicial na animação de partículas não tem uma solução fechada; ao invés disso, precisam ser resolvidos por aproximações discretas, como é o caso do *Euler implícito* ou *explícito*. A maneira mais simples de resolver equações diferenciais de primeira ordem é usando a integração explícita de Euler.

O tempo é discretizado da seguinte forma:

$$t_{i+1} = h + t_i$$

Uma aproximação de segunda ordem da posição em um tempo t_{i+1} é dada pela Série de Taylor de primeira ordem:

$$p(t_{i+1}) = p(t_i) + h\dot{p}(t_i) + \mathcal{O}(h^2)$$

Onde se tem um erro de ordem dois.

Aplicando o método de Euler na equação de movimento temos que:

$$\dot{x} = v$$

$$\dot{v} = \frac{\vec{f}}{m}$$

$$x_{t+h} = x_t + hv_t$$

$$v_{t+h} = v_t + h \frac{f_t}{m}$$

Talvez um dos piores problemas de Euler é também um dos principais problemas nesse projeto: a imprecisão numérica e a necessidade de usar

passos pequenos. Caso o passo seja relativamente grande, Euler pode divergir, o que nos obriga a usar passos fixos ao invés de trabalhar com tempo real. O passo fixo usado nas simulações é de 0.005 para mola e de 0.0005 para as barras.

Testou-se também o Método de Euler Melhorado, que dá mais precisão ao sistema, apesar de ainda continuar relativamente instável:

$$x_{t+h} = x_t + \frac{h}{2} [v_t + v_{t+h}^*]$$

$$v_{t+h} = v_t + \frac{h}{2} \left[\frac{f_t}{m} + \frac{f_{t+h}^*}{m} \right]$$

Onde v_{t+1}^* e f_{t+1}^* são a velocidade e a força calculadas por Euler, respectivamente. Esse foi o método que garantiu um pouco mais de precisão no sistema.

Existem métodos mais precisos e estáveis, porém mais caros computacionalmente, como é o caso do Método Runge-Kutta de ordem quatro. O Método de Runge-Kutta é uma expansão da Série de Taylor e pode ser descrito pela seguinte equação:

$$y_{i+1} = y_i + \frac{1}{6} [K_1 + 2K_2 + 2K_3 + K_4]$$

Onde:

$$K_1 = hf(t_i, y_i)$$

$$K_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}K_1\right)$$

$$K_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}K_2\right)$$

$$K_4 = hf(t_i + h, y_i + K_3)$$

4.13. Explosão do Modelo

Agora que se sabe calcular as forças nas barras e avançar no tempo, atualizando as posições e velocidades das partículas, é preciso explodir o modelo em si. Como já foi dito, o usuário terá liberdade em escolher a “faixa” de valores de *lambdas* que vão ditar quais barras quebrarem. Essa “faixa” é controlada da seguinte forma: obtêm-se os *lambdas* mínimos e máximos do

sistema relacionados às restrições de barras apenas; a partir daí, calcula-se um “limite” para quais *lambdas* que tiverem valor maior terão suas barras quebradas:

$$\text{limit} = (\max - \min) * \text{perc} + \min$$

Onde *perc* é o valor no intervalo [0,1] ajustado pelo usuário. Se o *Multiplicador de Lagrange* correspondente àquela barra for menor que esse limite, a barra se quebra. Esse limite considera apenas os valores de *lambda* mais baixos e de compressão. Se quisermos obter um limite para que os *lambdas* com valores mais altos se quebrem (o que é o caso para o algoritmo de “três passadas”, que considera o valor absoluto dos *lambdas* e os cossenos de seus ângulos obtidos pelas forças nas direções dos três eixos), consideramos a seguinte expressão:

$$\text{limit} = \max - (\max - \min) * \text{perc}$$

Após a explosão, caso o número de componentes conexos não mude, nota-se que a faixa de valores ajustado pelo usuário não foi grande o suficiente para quebrar mais barras que deveriam ser quebradas. Portanto, pode ser que seja preciso avançar um ou mais passos no tempo e explodir novamente. Lembrando que o termo *perc* (percentual) é extremamente relevante para a explosão do modelo, uma vez que ele simboliza o “limite” que separa as barras que serão quebradas das não quebradas.

Se o número de componentes foi alterado, estamos prontos para vê-los deslocarem de si mesmos, como se houvesse uma força de repulsão entre eles. Para isso, faz-se um deslocamento relativo dos seus centros de massa ao centro de massa do modelo original, lembrando que cada componente conexo possui seu próprio centro de massa. Pra recalcular o centro de massa de cada componente toda vez que aumentar o número de componentes conexos, o modelo é tratado como um grafo, onde aplicamos um caminhamento em largura para garantir um percorrimento de ordem *n*.

5. Atividades Realizadas

5.1. Método

O processo de desenvolvimento deste projeto teve início na implementação de um sistema de barras rígidas. Neste pequeno programa desenvolvido, simulou-se um cubo sujeito à ação da gravidade e colisão com planos, cujas partículas (ou vértices) foram ligadas por uma barra. Esse programa foi essencial para o autor ter a base para desenvolver o sistema de restrições para a explosão dos modelos. Nesse caso, aproveitou-se a maior parte do sistema e sua estrutura de dados para a explosão, exceto a parte de colisão.

O processo do Projeto Final II iniciou-se na implementação e seguiu-se durante quase todo o semestre nos testes 2D. No início, usou-se um sistema massa-mola para tentar determinar os pontos de quebra do modelo. Porém, esse método, apesar de obter bons resultados, mostrou instabilidade numérica e a necessidade de dar muitos passos de integração até chegar ao resultado desejado. Depois, usou-se apenas a restrição de barras rígidas calculadas sobre o agente externo de forças radiais. Contudo, observou-se que os resultados desses métodos não satisfaziam o objetivo final, que era determinar os pontos de quebra do modelo, por dois motivos: era quebrado mais de um lugar (além dos pontos de quebra) e eram necessários vários passos de integração até chegar ao escopo a ser alcançado. A partir daí, foram testados no sistema outros agentes externos e outra força de restrição, como forças radiais, agentes radiais proporcionais e inversamente proporcionais à distância, além de agentes calculados pela caixa orientada e Multiplicadores de Lagrange calculados por “três passadas”. As restrições radiais, assim como os agentes externos anteriores não obtiveram bons resultados, visto que foram quebrados mais lugares além dos pontos de quebra e o número de componentes conexos aumentou significativamente. Já o algoritmo de “três passadas” e a caixa orientada foram os que mais se aproximaram do resultado esperado.

A próxima tarefa foi testar o sistema usando modelos reais e 3D. Porém, do jeito que as estruturas de dados foram desenhadas, era praticamente impossível fazer os testes devido ao imenso número de barras e partículas que o modelo continha. O próximo passo então era otimizar as estruturas de dados, reduzindo o número de alocações dinâmicas que seria usadas para guardar as

relações entre barras e os vetores auxiliares que guardam as matrizes. A partir daí, foi feito o teste em cima de um modelo 3D.

Abaixo segue um cronograma que inclui as atividades do Projeto Final I e II.

9/08	Pesquisa realizada a respeito de conceitos, técnicas e implementações de explosão de modelos na Visualização Científica;
10/08	Implementação de um sistema massa-mola utilizando um sistema de partículas;
11/08	Implementação 3D de um sistema de barras rígidas, sendo feito a seguir um cubo sujeito à ação da gravidade;
01/09	Uso do sistema massa-mola para determinar os pontos de quebra em modelos 2D;
02/09	Uso de restrições de barras rígidas sujeitas a forças radiais para explodir o modelo 2D;
03/09	Testes com restrição radial sujeitos a outros agentes externos;
04/09	Testes com apenas a restrição de barra rígida e resolução de forças nas barras usando o algoritmo de “três passadas”;
05/09	Testes em cima de uma modelo real 3D

Há algumas diferenças entre este cronograma e o apresentado no Projeto Final I. São elas:

- A tarefa de implementar um sistema de barras rígidas para explodir o modelo foi feito no Projeto Final II;
- A tarefa de explosão por propriedades também foi feita no Projeto Final II;
- A implementação 3D, que foi dita para ser feita no começo do Projeto Final II, foi deixada como último teste do mesmo;
- O estudo das estruturas de dados do projeto Geresim do TecGraf não precisou ser feita, no momento em que o sistema carrega um arquivo Mesh que foi gerado a partir das estruturas de dados do Geresim;
- Por falta de tempo e grande complexidade, não foi incorporado o sistema no projeto Geresim.

6. Projeto e Especificação do Sistema

Talvez um dos principais desafios deste projeto foi o desenvolvimento de suas estruturas de dados, que precisaram ser modificadas ao longo do período, principalmente devido ao grande número de dados do modelo real. Para o desenvolvimento do sistema, foi usado a linguagem C++, as bibliotecas do OpenGL, além das bibliotecas de interface do IUP.

6.1. Processo de Simulação

Para a arquitetura do projeto, foi usado como principal núcleo da simulação uma classe *System*, na qual é responsável por simular cada passo físico e entregar e atualizar para as demais classes informações relevante como posição e velocidade das partículas, além de ser responsável por chamar o integrador numérico, seja Euler ou Runge-Kutta, para atualizar as posições e velocidades das partículas. Ele calcula a diferença de tempo, usando-o como passo de integração. O sistema também é responsável por disparar o pioneiro da simulação: os agentes externos. Ele também contém referências para as classes de restrição e massa-mola, além de referência para uma classe responsável por calcular a caixa orientada dos componentes conexos do modelo.

Para a simulação, usou-se uma função *Idle* que calcula o próximo passo e desenha na tela durante todo o tempo de execução do programa. A função faz basicamente os seguintes passos:

Idle:

t = clock->GetTime()

system->Simulate(t)

Redraw()

Figura 15. Processo de simulação.

Repare que, no nosso caso, não é necessário fazer o primeiro passo de capturar o tempo, pois usamos um passo fixo e pequeno para evitar instabilidade no sistema. O Passo utilizado para a simulação com molas foi de 0.005, enquanto para a simulação de barras rígidas e restrições foi de 0.0005.

6.2. Estruturas de Dados

Como foi mencionado antes, foi preciso criar uma estrutura de dados compacta e objetiva, pois quando formos fazer testes em cima de modelos reais que contém facilmente milhares de partículas, evitamos assim o desperdício de memória, possibilitando maior eficiência. A primeira estrutura de dados que foi desenvolvida não levou em conta essas “otimizações”. Abaixo segue as estruturas de dados de alguns atributos.

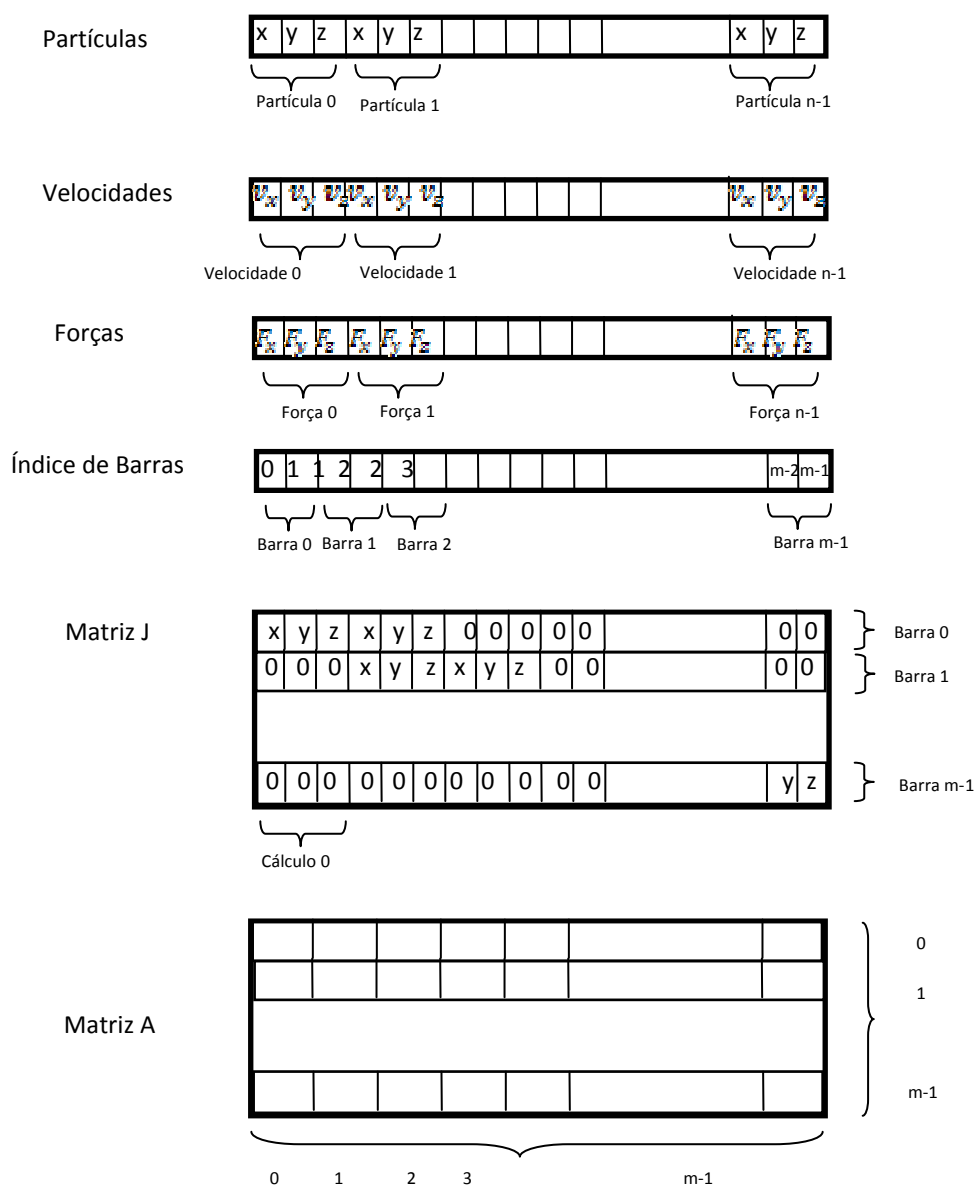


Figura 16. Estruturas de dados inicial.

O primeiro *array* representa as posições das partículas, o segundo as suas velocidades, e o terceiro as forças aplicadas a elas. O quarto *array* representa as barras em si, onde cada duas posições consecutivas simbolizam os índices das partículas na qual a barra liga. Já a Matriz Jacobiana é representada como uma matriz, e a matriz A representa a matriz do sistema linear $Ax=b$. Outros arrays não estão representados na figura pois suas estruturas são idênticas com algum representado acima, como por exemplo:

- Derivada da Matriz Jacobiana: idêntica a Matriz Jacobiana;
- Array da função comportamento: é um array de dimensão m , onde m é o número de barras do sistema;
- Array da derivada da função comportamento: idêntico ao array da função comportamento;
- Array b (do sistema $Ax=b$): um array de dimensão m , onde m é o número de barras;
- Array dos Multiplicadores de Lagrange: Dimensão m , onde m é o número de barras;

Existe, contudo, um problema na representação das matrizes Jacobianas e da matriz A (do sistema linear): Se houver n partículas e m barras, a dimensão da matriz Jacobiana será $3 * m * n$ e a dimensão de A será m^2 . Não é difícil imaginar que um modelo real possa conter por volta de 30 mil partículas e 80 mil barras, o que fica inviável ter uma dimensão da matriz Jacobiana de 7,2 bilhões e dimensão da matriz A de 6,4 bilhões. Esse problema fez com que o autor propusesse uma otimização para as estruturas de dados das matrizes.

Observando cuidadosamente as matrizes Jacobinas, pode-se notar que é uma matriz com apenas 6 valores em cada linha, tendo o resto como valor zero. A mesma observação vale para a sua derivada. Portanto, podemos denotar a matriz Jacobiana da seguinte forma:

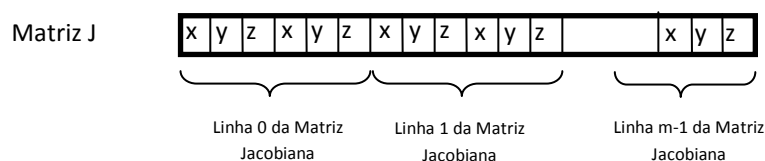


Figura 17. Matriz Jacobiana.

Para determinar as colunas na qual cada elemento se encontra na matriz, basta olhar para o vetor de índice de barras. Se a primeira barra tem índices 0 e 2, por exemplo, o primeiro elemento se encontra na coluna $3 \cdot 0$, o segundo, terceiro, quarto, quinto e sexto nas colunas $3 \cdot 0 + 1$, $3 \cdot 0 + 2$, $3 \cdot 1$, $3 \cdot 2 + 1$ e $3 \cdot 2 + 2$, respectivamente. Fazemos a mesma otimização para a derivada da matriz Jacobiana.

Como a matriz A é uma matriz esparsa, ou seja, cheio de elementos igual a zero, podemos fazer uma otimização semelhante. Porém, precisaremos guardar em dois vetores: no primeiro temos o número de elementos por coluna e em que coluna começa, e no segundo vetor temos quais colunas eles se encontram.

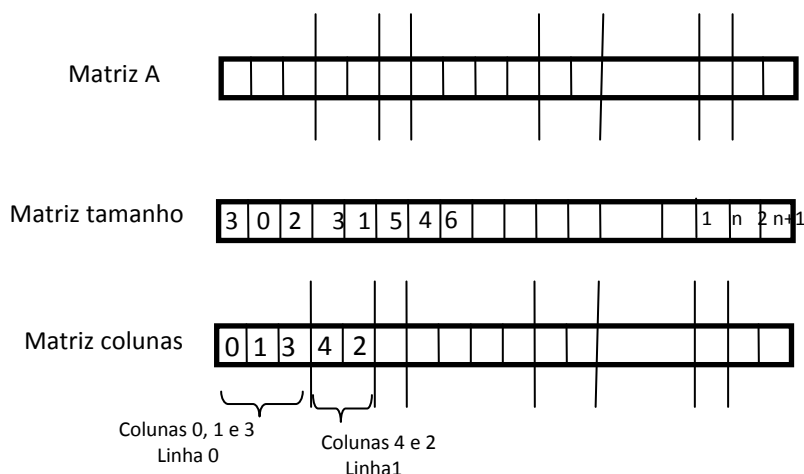


Figura 18. Estruturas de dados final.

Pode-se observar que o maior destes vetores são a matriz A e a matriz de colunas, que têm tamanho indefinido, dependendo do número de elementos em

cada linha. O vetor tamanho tem apenas dimensão m , o número de barras. Com isso, reduzimos muito o espaço em memória, que antes era por volta de m^2 .

Agora nos resta especificar a estrutura de dados para o grafo no modelo. Lembrando que esse grafo é usado para calcular os centros de massa dos componentes conexos e fazer seus deslocamentos relativos em ordem(n). Considerando que cada célula do modelo não contém mais que dez células vizinhas, cria-se uma matriz com dimensões $n \times 10$, onde n é o número de partículas. Ou seja, cada linha corresponde a uma partícula, e cada uma das dez colunas correspondem a seus vizinhos. Essa estrutura facilita executar um caminhamento em largura em um grafo sem a necessidade de criar uma lista de adjacências.

6.3. Módulos Desenvolvidos

Nesse Projeto foram desenvolvidos 16 módulos de implementação separados por uma funcionalidade específica. O usuário preocupou-se em tornar o sistema mais extensível possível, permitindo que adicione-se mais que uma restrição com facilidade, como é um dos casos de teste que contem as restrições de barra rígida e radial. Cada módulo determina uma classe, e são elas:

- Classe **Constraint**: é uma classe abstrata, definindo uma “interface” para uma determinada restrição. Junto com a classe **AllConstraints**, é possível adicionar mais que uma restrição no sistema com facilidade. O usuário que pretende escrever uma nova restrição precisa seguir rigorosamente essa “interface”;
- Classe **RigidBar**: A principal classe do sistema que estende a classe abstrata **Constraint**, é responsável por descrever as restrições de barra rígida e contém todas as estruturas de dados relacionadas a ela. Ela é responsável por construir as matrizes e o sistema da restrição de barras rígidas, além de resolver o sistema utilizando a classe **Cg**. Também armazena os Multiplicadores de Lagrange dessa restrição, além de conter as forças de restrição que evitam as partículas de seguirem uma trajetória indesejada.

- Classe **Radial**: Assim como a classe **RigidBar**, essa classe é outra restrição que estende a classe abstrata **Constraint** e fica responsável por todas as operações envolvendo restrições descritas na classe acima.
- Classe **AllConstraints**: Essa classe fica responsável por colher todas as informações (matrizes, forças, etc.) que as classes de restrições constroem, calculando as forças de restrições e os Multiplicadores de Lagrange envolvendo todas as restrições. Um exemplo de uso dessa classe é o momento em que usa-se as restrições de barra rígida e radial. Porém, para evitar o desperdício de memória alocando mais *arrays*, essa classe não é usada quando testado o modelo real.
- Classe **MassSpring**: Descreve o sistema massa-mola. Essa classe é usada para testar a explosão do modelo utilizando a técnica de mola ao invés das restrições. Contém índices para a relação de mola entre cada partícula, parecido com os da barra rígida.
- Classe **Model**: Descreve o modelo em si. Todas as posições e velocidades das partículas que foram alteradas pela simulação de barras rígidas, molas e outras restrições não fazem com que a velocidade e posição das partículas dessa classe se alterem. Aqui, os atributos das partículas só são alterados quando se faz um deslocamento relativo ao explodir o modelo. Aqui também contém as estruturas de dados relativas ao grafo, além de conter funções responsáveis por fazer um caminharmento em largura.
- Classe **Clock**: Responsável por capturar o tempo no sistema. Como não usamos tempo real e o passo é fixo, essa classe não é usada.
- Classe **Cg**: Concentra o método do Gradiente Conjugado, responsável por resolver um sistema linear e entregar para as classes de restrição. No início do desenvolvimento do projeto final, o usuário desenvolveu seu próprio código. Mas a medida em que se aumentava o sistema, este programa resultava em Multiplicadores de Lagrange menos precisos. Então, foi usado um código na internet.
- Classe **Explosion**: Responsável por receber o modelo e explodi-lo por propriedades ou por forças nas barras.

- Classe **OrientBoundingBox**: Somente usada para calcular a caixa orientada de cada componente conexo para que possa ser aplicada uma força na direção da maior densidade de partículas. Para encontrar a caixa orientada, essa classe calcula os autovetores e autovalores de uma matriz específica.
- Classe **Mesh**: Responsável por desenhar o modelo de restrições, ou o modelo de massa-mola, ou o modelo em si.
- Classe **Manipulator**: Responsável pelo manipulador do modelo, incluindo rotação e zoom.
- Classe **Solver**: Responsável por avançar um passo no tempo utilizando integradores numéricos como Euler, Euler Melhorado ou Runge-Kutta de ordem 4.
- Classe **Server**: É uma classe abstrata, definindo uma “interface” para a classe **System**.
- Classe **System**: Chama todas as classes que precisa para simular um passo, inicializando forças, adicionando os agentes externos, atualizando as matrizes para calcular os Multiplicadores de Lagrange e consequentemente as forças de restrição, e por fim chamando os integradores numéricos pela classe **Solver** para avançar no tempo.
- Módulo **main**: Concentra as funções de inicialização, desenho, e *idle*, chamando e inicializando as classes quando for necessário. Constrói todo o sistema.

7. Implementação e Avaliação

Para garantir a segurança e boa eficiência do sistema desenvolvido, foram feitos testes durante todo o desenvolvimento do programa. A maior parte deles foi realizado em cima de modelos 2D, pois como foi ressaltado antes, são menos complexos e seus resultados são melhores enxergados. Os modelos 2D foram criados especificamente para os testes, ou seja, não são reais. Para criar algum desses modelos foi desenvolvido um pequeno editor gráfico, na qual pega modelos do tipo 3DS e dando possibilidade do usuário conectar as partículas por uma linha, que será convertida numa relação de barra rígida entre ambas.

7.1. Planejamento e execução de testes funcionais

Para a realização dos testes, foram executados diversos casos para verificar a melhor maneira de explodir o modelo. São eles:

- Modelo massa-mola;
- Modelo de barras rígidas com agente externo radial;
- Modelo de barras rígidas com agente externo radial proporcional à distância;
- Modelo de barras rígidas com agente externo radial inversamente proporcional à distância;
- Modelo de barras rígidas com restrição radial;
- Modelo de barras rígidas com agente externo relacionado a caixa orientada (*OBB*);
- Modelo de barras rígidas com algoritmo das “três passadas”;

7.1.1. Modelo Massa-Mola

O primeiro caso testado foi o do modelo massa-mola, na qual, ao invés de conectar as partículas por uma barra, usa-se uma mola. Desse modo, ao aplicar a força externa radial, as molas permitem um deslocamento entre as duas partículas, diferente da barra que não permite deslocamento algum. Lembrando a equação usada para a explosão:

$$limit = max - (|max - min|) * perc$$

Onde *perc* é a percentagem ajustada pelo usuário, indicando a faixa de valores que determinam que molas se romperão. Vale lembrar que a fórmula que determina a quebra da mola se dá pela relação comprimento da mola sobre comprimento de repouso da mola:

$$relation = \frac{spring\ current\ length}{spring\ resting\ length}$$

Quanto maior e a relação, mais “esticada” a mola se encontra.

Executando o teste com um modelo mesh, obtivemos os seguintes resultados:

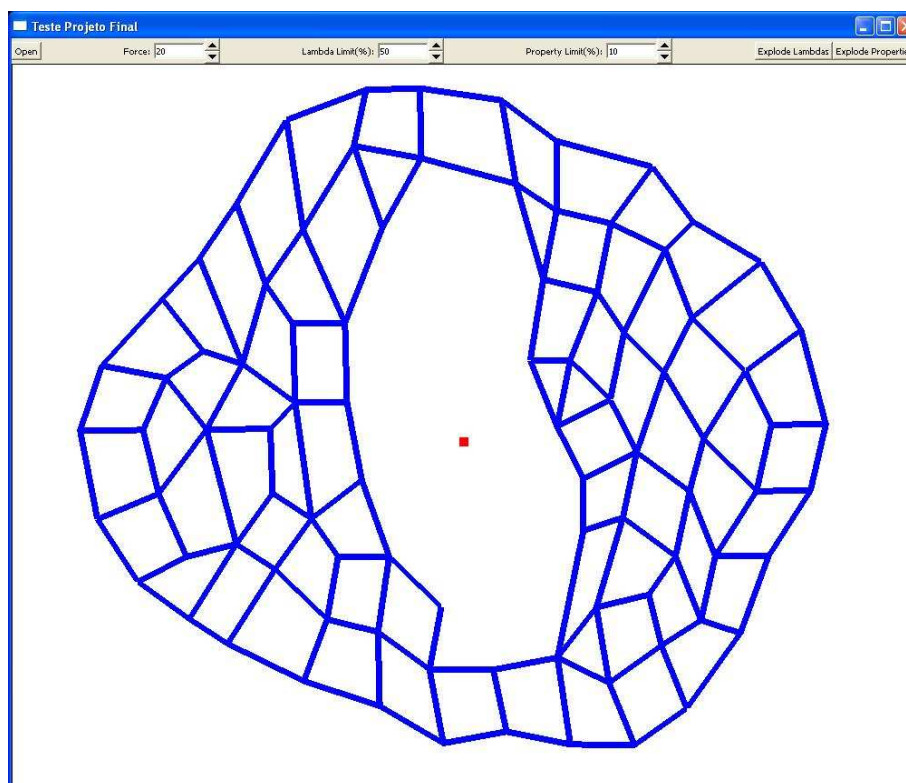


Figura 19. A figura mostra o modelo mesh em si antes da explosão. Note os pontos frágeis do modelo nas partes inferior e superior.

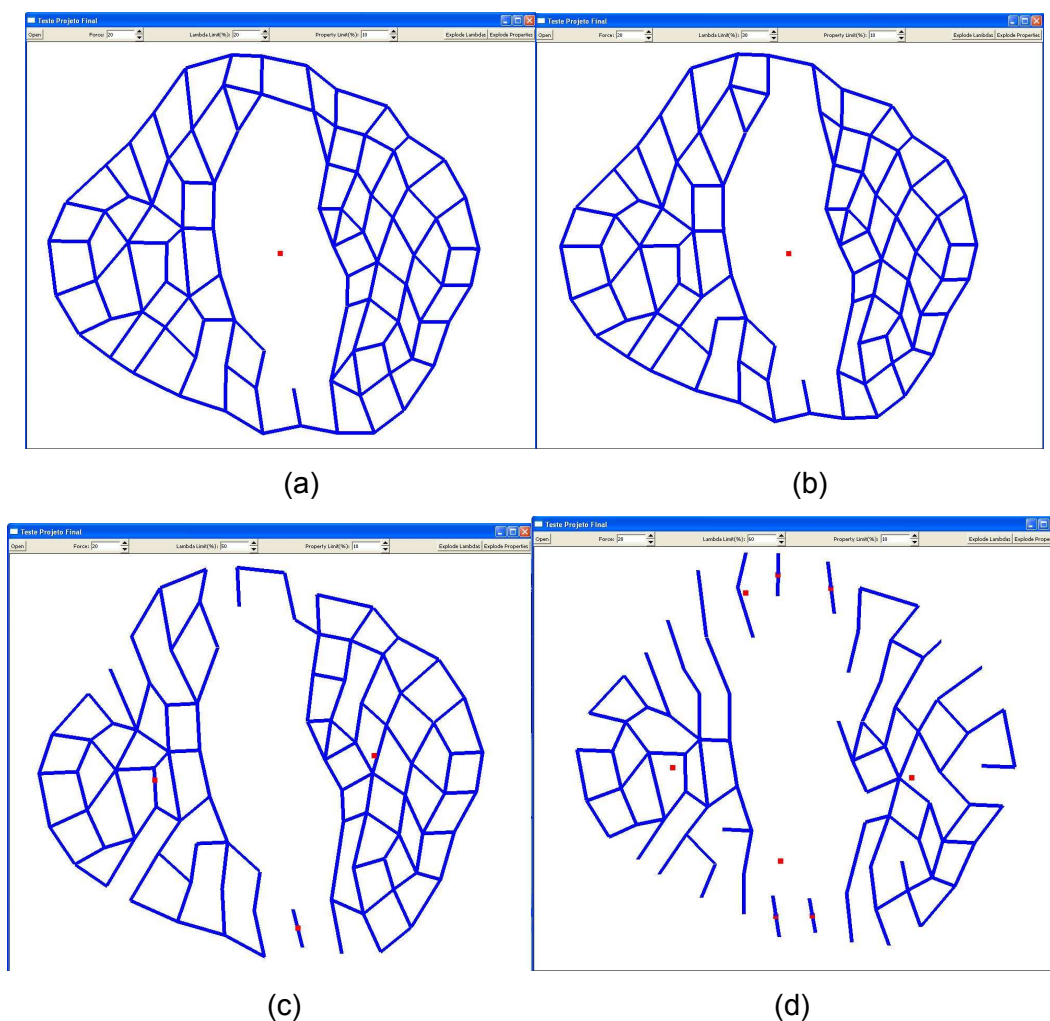


Figura 20. (a) Mostra a explosão com limite de percentagem 20%. (b) Explosão com percentagem 30%. (c) Explosão com percentagem 50%. (d) Explosão com percentagem 60%. Os pontos brancos indicam os centros de massa dos componentes. Essas simulações demoraram por volta de 650 passos, o que a torna mais lenta que as barras rígidas, além de apresentar instabilidades. O passo de integração fixo é 0.005.

7.1.2. Modelo de Barras Rígidas com força radial

O modelo seguinte a ser testado foi simulado com barras rígidas sujeito a um dos três agentes externos radiais: sem distância, proporcional à distância e inversamente proporcional à distância. Note que a equação de quebra é dada com o da mola:

$$limit = (|max - min|) * perc + min$$

É importante lembrar que nesse caso procuramos pelos Multiplicadores de Lagrange negativos, pois são as forças de compressão. Portanto, verifica-se antes se o λ é negativo para que depois veja se ele está abaixo do limite.

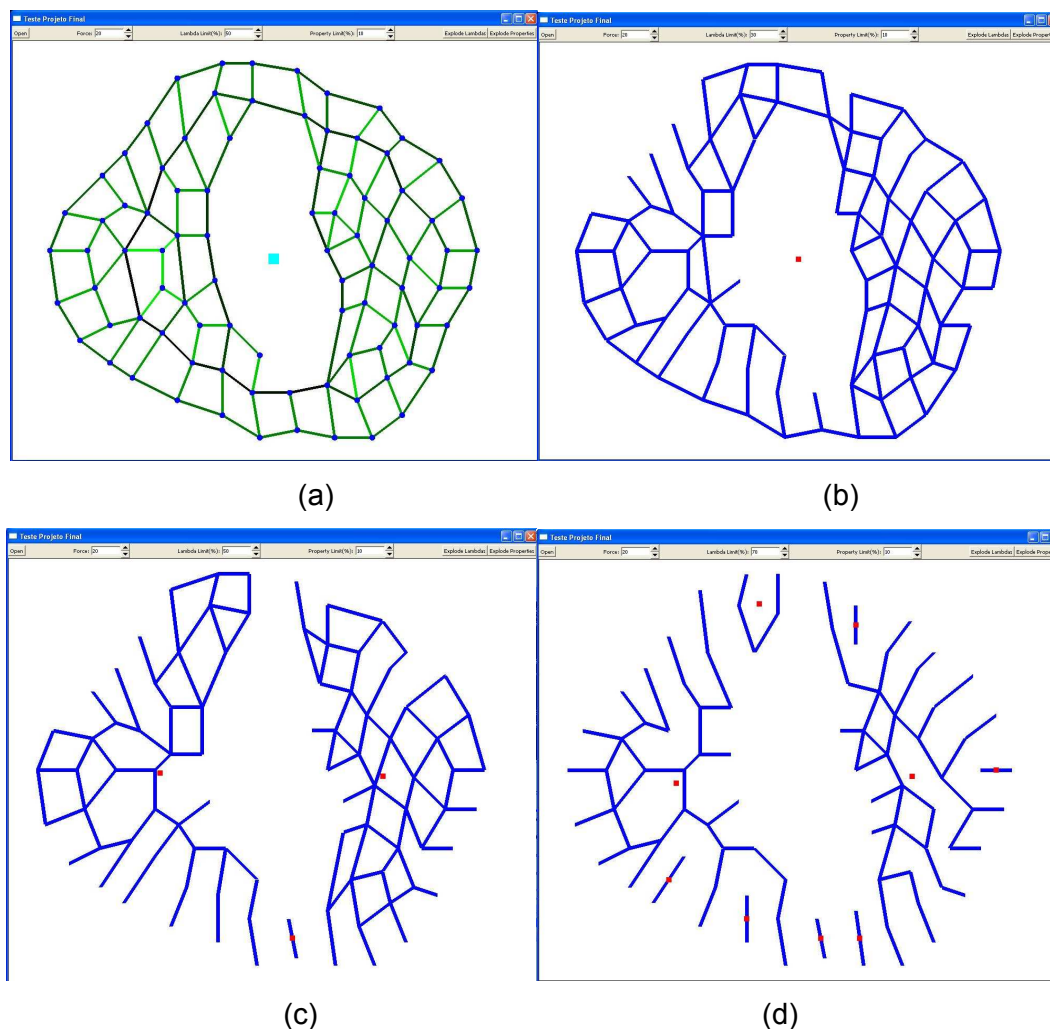


Figura 21. (a) Mostra a simulação com 52 passos. (b) Explosão com percentagem 30%, após 2 quebras e 53 passos de simulação. (c) Explosão com percentagem 50%, 1 quebra e 52 passos de simulação. (d) Explosão com percentagem 70%, 1 quebra e 52 passos.

Essa simulação foi feita sem considerar a distância do vértice ao centro de massa de seu componente. Na figura (a) pode-se observar que a parte inferior contém barras pretas, isso porque os Multiplicadores de Lagrange são negativos e contêm valores muito baixos, ou seja, existe uma força de compressão atuando na barra. As barras são pintadas da seguinte forma: quanto mais alto são seus valores, mais verdes são; quanto mais baixos seus valores, mais pretas são.

Observe que na figura (b) foi preciso quebrar 2 vezes o modelo, na primeira com 52 passos de simulação, e depois ao avançar mais um passo. Esse resultado mostra o contorno do modelo sendo pouco quebrado, diferente das figuras (c) e (d). Contudo, a figura (b) mostra que na parte inferior quebrou mais do que precisava, observando uma barra “solta” resultando da explosão. O problema desse método é o fato de ter que executar vários passos de simulação.

Agora será mostrado o método de barras rígidas usando agentes radiais e considerando a força como sendo proporcional à distância.

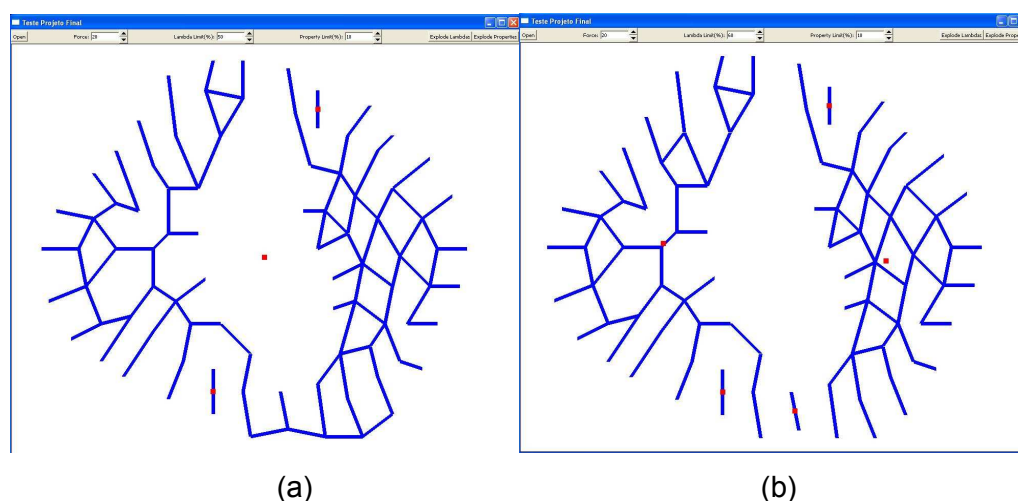


Figura 22. (a) Explosão com percentagem 50% , 1 quebra e 9 passos de simulação. (b) Explosão com percentagem 60%, após 1 quebra e 9 passos de simulação.

Como se pode observar na figura (a), explodindo uma vez apenas não é o suficiente para quebrar todos os pontos frágeis. Na figura (b), 60% são suficientes para quebrar os pontos frágeis, porém outros locais indesejados também são quebrados, como o contorno e algumas barras fazem com que aumente mais ainda o número de componentes conexos.

Resta mostrar o modelo aplicado a uma força radial inversamente proporcional à distância.

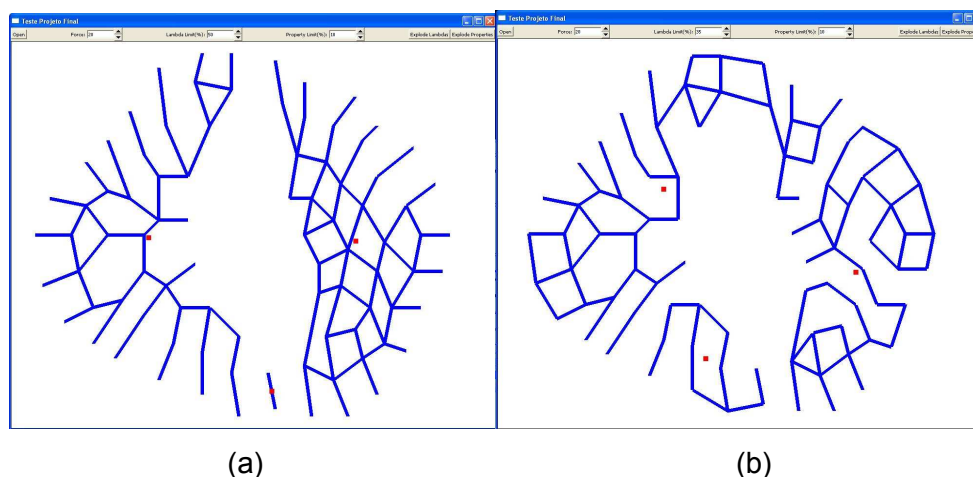


Figura 23. (a) Explosão com percentagem 50% , 2 quebras e 2 passos de simulação. (b) Explosão com percentagem 35%, após 5 quebras e 7 passos de simulação.

Nessa simulação, pode-se observar que não é uma boa opção usar o inverso da distância. Ele só atinge a explosão quase desejada por um preço alto: o de quebrar muitas barras além das desejadas com mais de uma quebra. Isso quer dizer que se avançou certo número de passos no tempo, quebraram-se as barras, avançou-se mais passos no tempo e quebraram-se as barras novamente. No caso da figura (b), mesmo após cinco quebras a explosão não foi bem-sucedida.

7.1.3. Modelo de barras rígidas com agente externo seguindo a Caixa Orientada (OBB)

Agora usaremos a Caixa Orientada, ou *Oriented Bounding Box*, para aplicar o agente externo. Lembrando que o maior eixo da Caixa define a direção e os dois sentidos (dependendo de que lado a partícula se encontra em relação ao centro de massa de seu componente) da força aplicada.

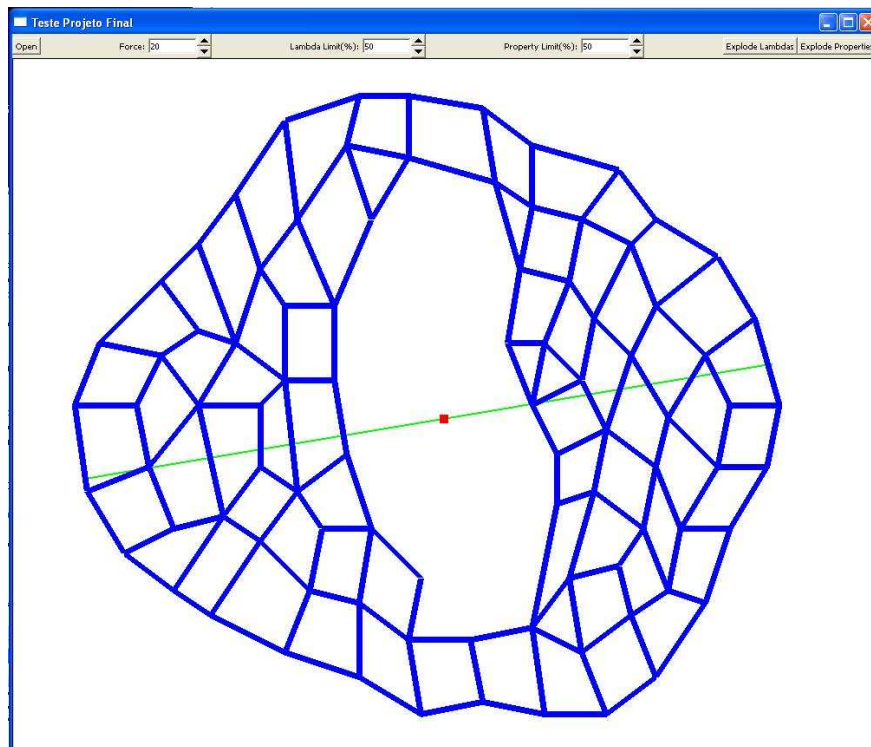


Figura 24. Modelo 1 a ser testado seguindo a caixa orientada.

Observa-se por essa figura que a linha em verde é a direção do maior eixo da *Bounding Box*, ou seja, o eixo que acompanha a densidade de partículas. É nessa direção que a força será aplicada.

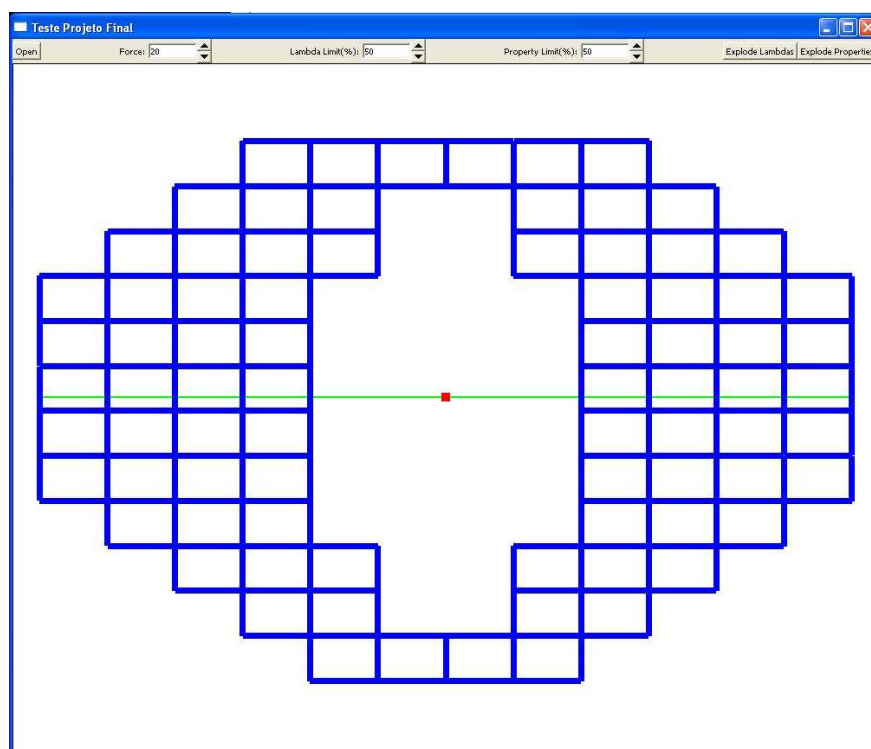


Figura 25. Modelo 2 a ser testado seguindo a caixa orientada.

A figura acima mostra outro modelo exemplificando o eixo da caixa orientada, que também será testado.

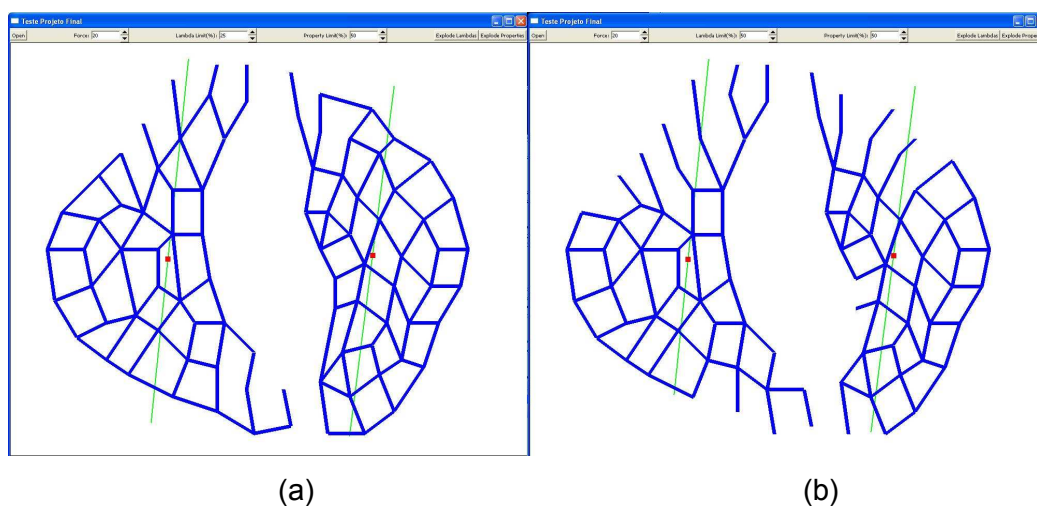


Figura 26. (a) Modelo explodido com percentual 25%, 3 quebras e 3 passos de simulação. (b) Explosão com percentual 50%, 2 quebras e 2 passos de simulação.

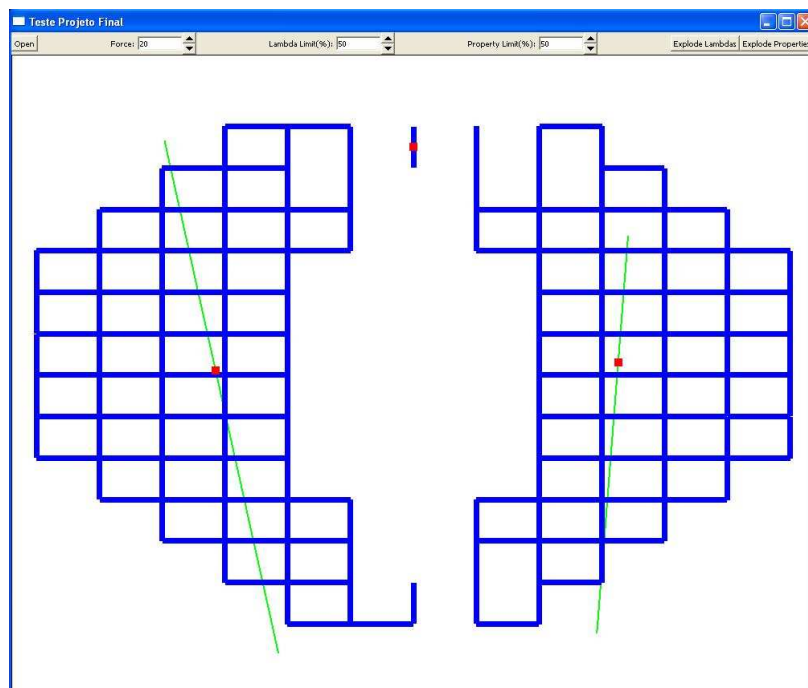


Figura 27. (a) Modelo explodido com 50% de percentual, 2 quebras e 2 passos.

Observando os dois modelos explodidos, pode-se afirmar que a Caixa Orientada é uma boa técnica, pois permite boas explosões por um preço barato: poucos passos de integração e poucas quebras.

7.1.4. Modelo de Barras Rígidas com Restrição Radial

Nesse caso será adicionada outra restrição, na qual as partículas só poderão se mover no eixo radial ao centro de massa de seu componente. A adição dessa restrição tem por objetivo evitar o cisalhamento entre células adjacentes. Foram testados diversos agentes externos, são eles: não considerar a distância da partícula ao centro de massa de seu componente nos cálculos, agente proporcional à distância, agente inversamente proporcional à distância e agente considerando a caixa orientada (*OBB*) para os três agentes anteriores.

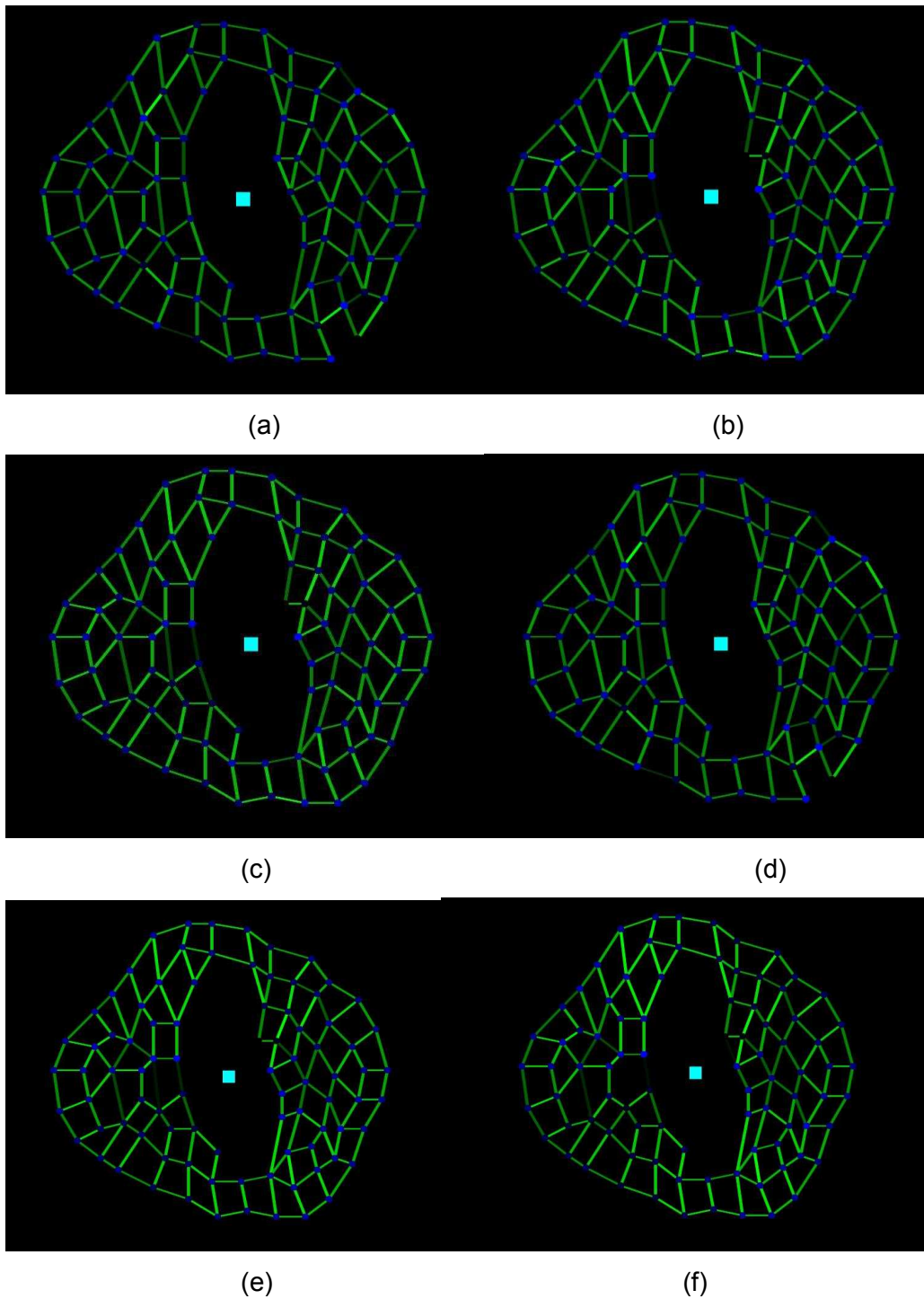


Figura 28. (a) Inversamente Proporcional à distância. (b) Proporcional à distância. (c) Sem considerar a distância. (d) Com a caixa orientada e força inversamente proporcional à distância. (e) Caixa orientada sem considerar a distância. (f) Caixa orientada com força proporcional à distância.

As figuras mostram a representação das forças nas barras, ou os Multiplicadores de Lagrange. Quanto mais verde claro, maior a força, e quanto mais escuro, menor a força, sendo que a mesma pode ser negativa (nesse caso

uma força de compressão). Note que, comparando os módulos das forças, observa-se que estas não são bem definidas. Não existe um padrão de maior fragilidade nas partes superior e inferior do desenho. Todas as barras parecem ter a mesma força aplicada a elas, o que não faz dessa uma boa técnica.

7.1.5. Modelo de Barras Rígidas com o algoritmo de “três passadas” seguindo o centro da *OBB*

O método usado a seguir faz uso do algoritmo de “três passadas”, isto é, calcula três vezes os *lambdas* referentes aos seguintes agentes externos: força apenas na direção do eixo *x*, na direção do eixo *y* e no eixo *z*, somando os resultados projetados na direção das barras. Note que aqui é preciso calcular três vezes os *Multiplicadores de Lagrange*, o que faz essa técnica ser mais lenta que as outras convencionais de uma passada. Observou-se que não é necessário recalcular a matriz *A* do sistema $Ax = b$ duas vezes nas três passadas, pois as matrizes Jacobianas que contém informações dos gradientes não são alteradas após calcular as forças. Porém, após um passo de simulação, isto é, após avançar no tempo, é necessário recalcular a matriz *A*. Aqui também se faz uso do agente externo multiplicado pela distância da partícula ao centro de massa, obtendo melhores resultados. Os testes também foram desenvolvidos levando em conta o centro da *OBB* como a origem da força externa aplicada ao modelo. Veja nas figuras abaixo. Lembrando da equação de quebra considerando que não há *lambdas* negativos:

$$limit = max - (max - min) * perc$$

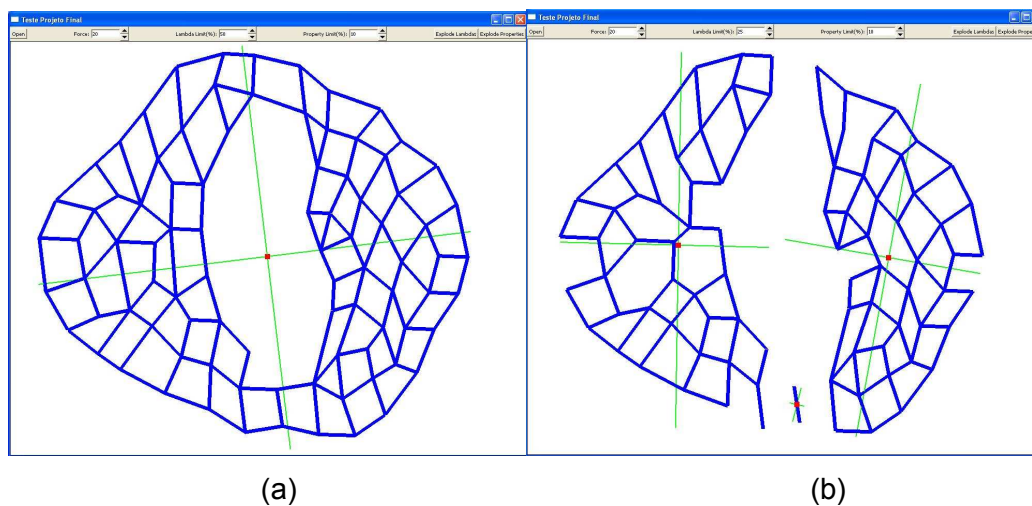


Figura 29. (a) O Modelo original, com os eixos da caixa orientada em verde. (b) Modelo explodido com percentual 25%, 2 passos de simulação e 2 quebras.

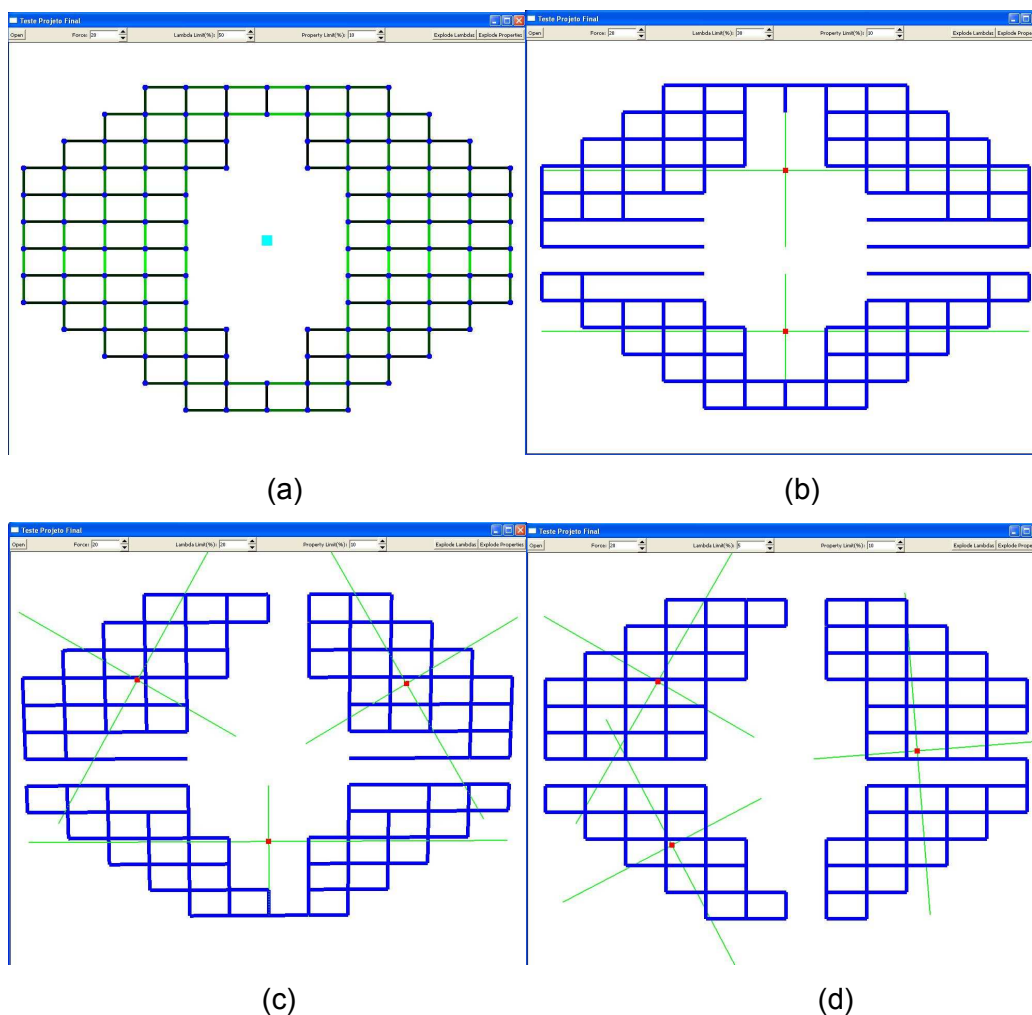


Figura 30. (a) O Modelo original, com *lambdas* mais claros com maior valor e *lambdas* mais escuros com menor valor. (Observação: não há *lambda* negativo.) (b) Modelo explodido com percentual 30%, 1 passo de simulação e 1 quebra. (c) Modelo explodido com 20% de percentual, 2 passos de simulação e 2 quebras. (d) Modelo explodido com 5%, 6 quebras e 6 passos de simulação.

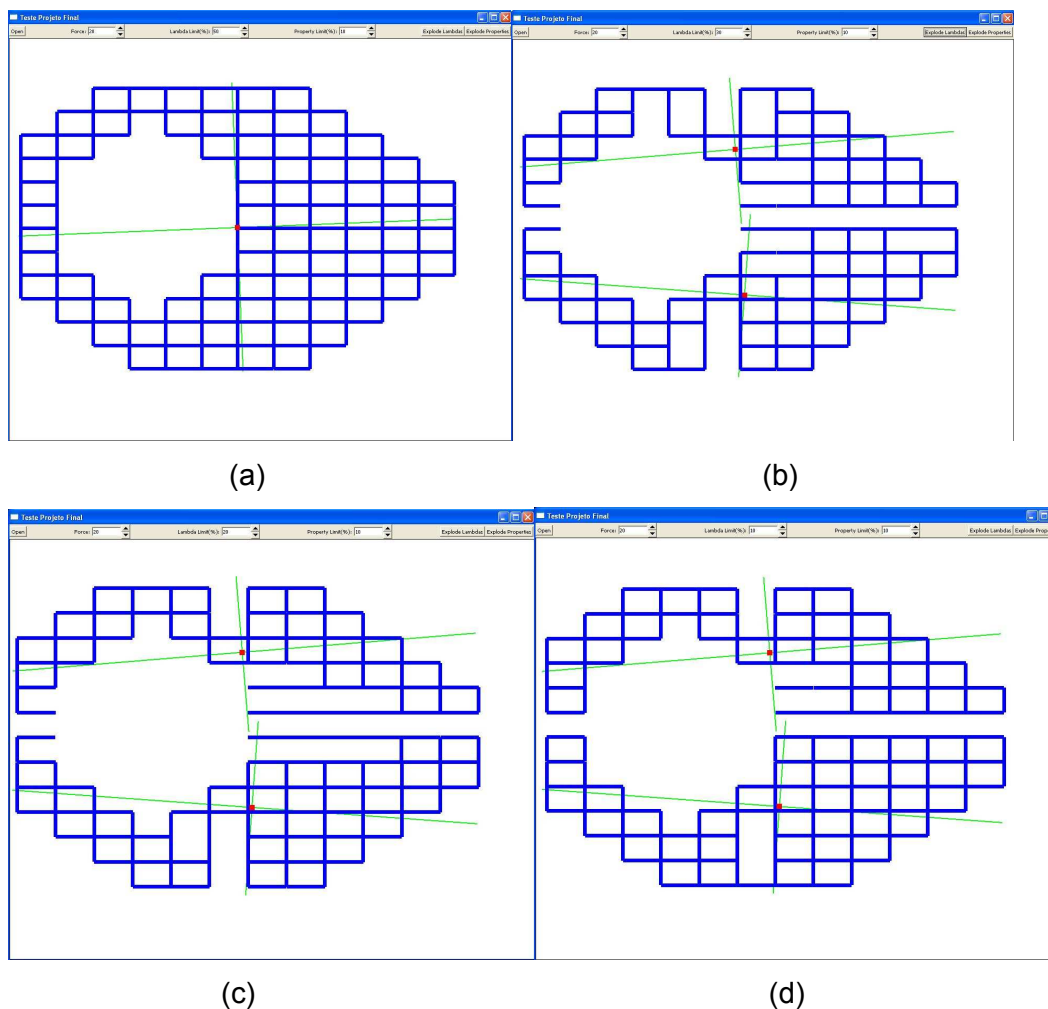


Figura 31. (a) O Modelo original. (b) Modelo explodido com percentual 30%, 2 passos de simulação e 2 quebras. (c) Modelo explodido com 20% de percentual, 3 passos de simulação e 3 quebras. (d) Modelo explodido com 10%, 5 quebras e 5 passos de simulação.

Nota-se que, com esse método, as explosões se dão mais naturalmente. Porém, no caso do segundo modelo, as quebras não ocorrem nos locais mais frágeis. Apenas quando aplicamos o algoritmo e quebramos várias vezes com um percentual mais baixo é que os locais frágeis se quebram. Mesmo assim esse algoritmo obteve melhores resultados que os métodos anteriores.

7.1.6. Modelo Real

O último passo foi testar o programa em um modelo real. Aqui será mostrado dois modelos: em uma camada do reservatório, e no modelo Namorado inteiro. O modelo Namorado contém 31550 células e 85521 barras, o

que torna o cálculo das forças nas barras demorado. Seguem os resultados abaixo.

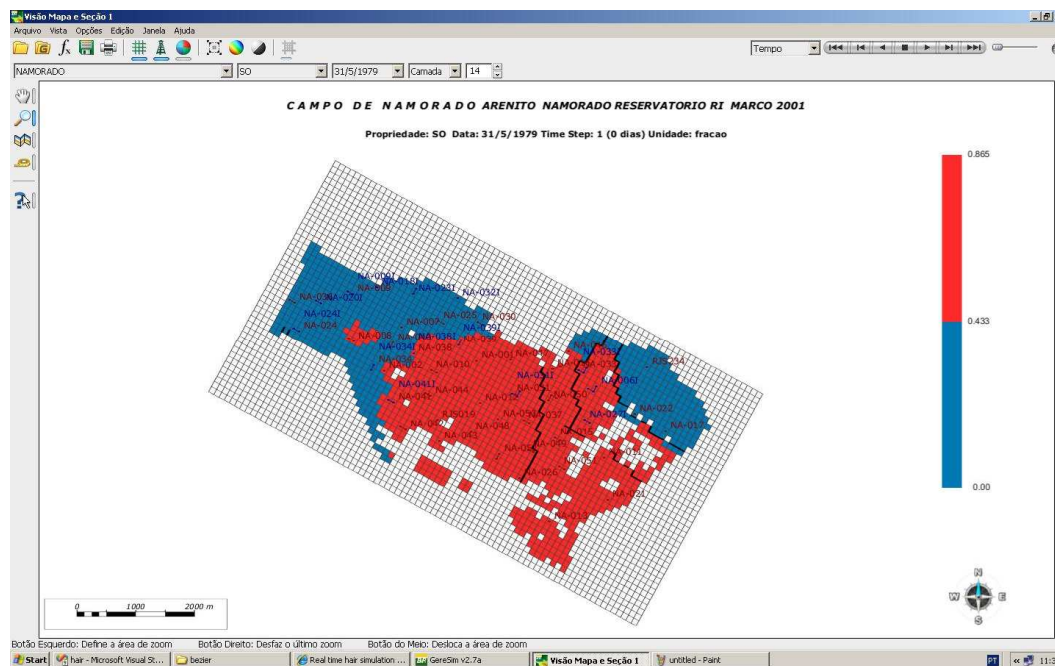


Figura 32. Camada do modelo Namorado no projeto Geresim, do TecGraf. Note que existe um volume confinado nas propriedades azuis da qual estaremos interessados em explodir.

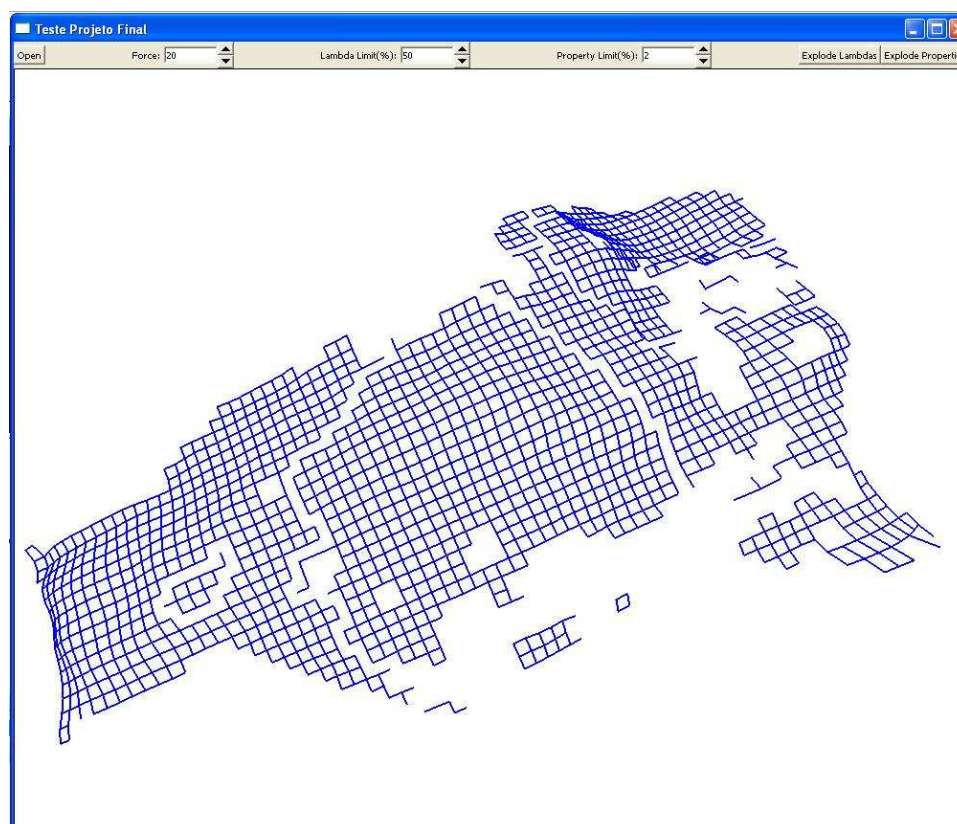


Figura 33. Uma camada do modelo Namorado já explodido por propriedades.

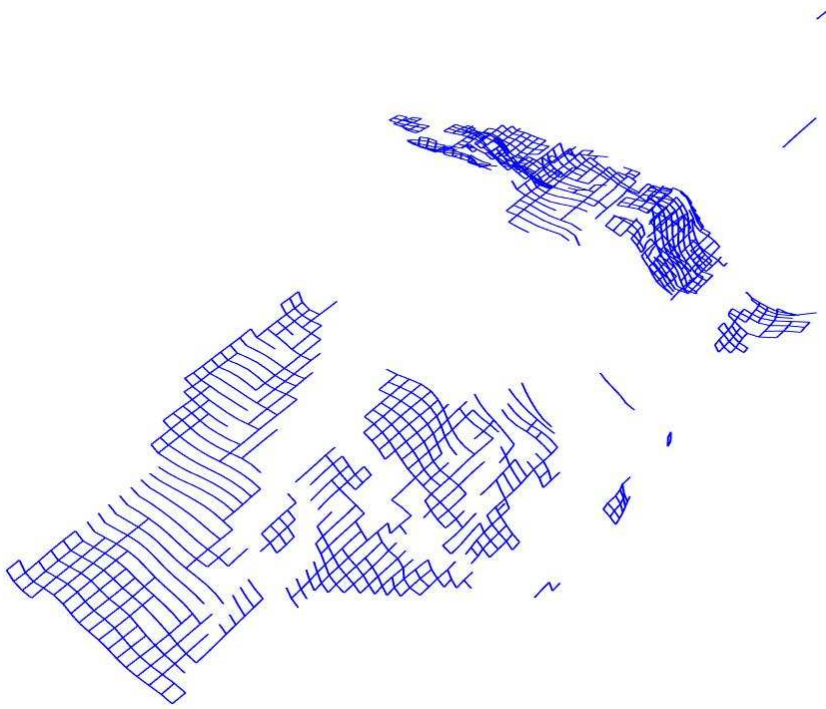


Figura 34. Camada do modelo Namorado já explodido. Repare que o volume envolvente da figura anterior agora se encontra livre.



Figura 35. Camada do modelo Namorado já explodido.

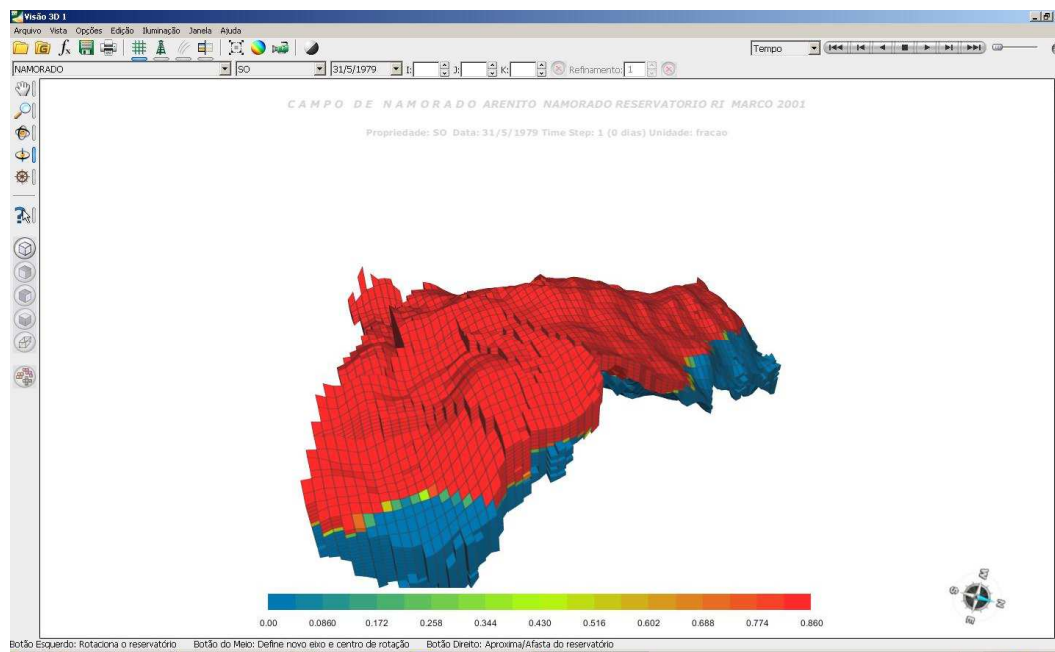


Figura 36. Modelo Namorado no projeto Geresim, do TecGraf.

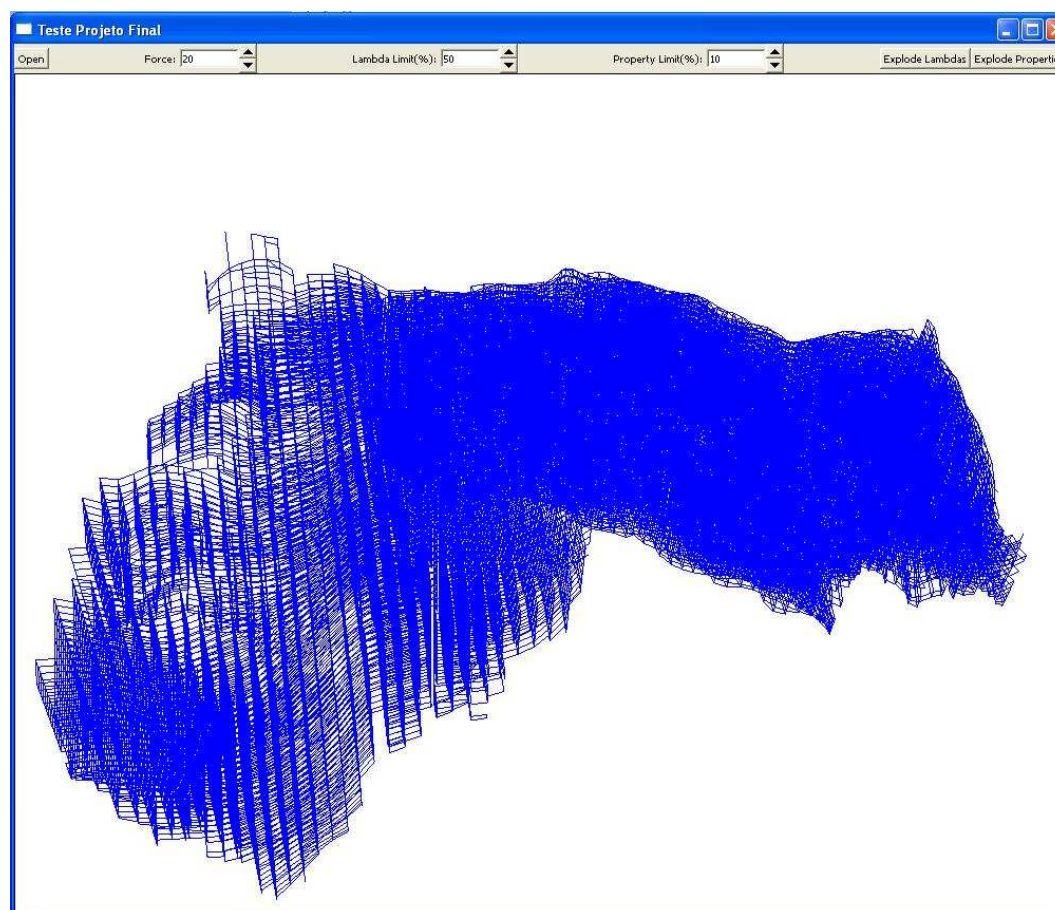


Figura 37. Modelo Namorado.

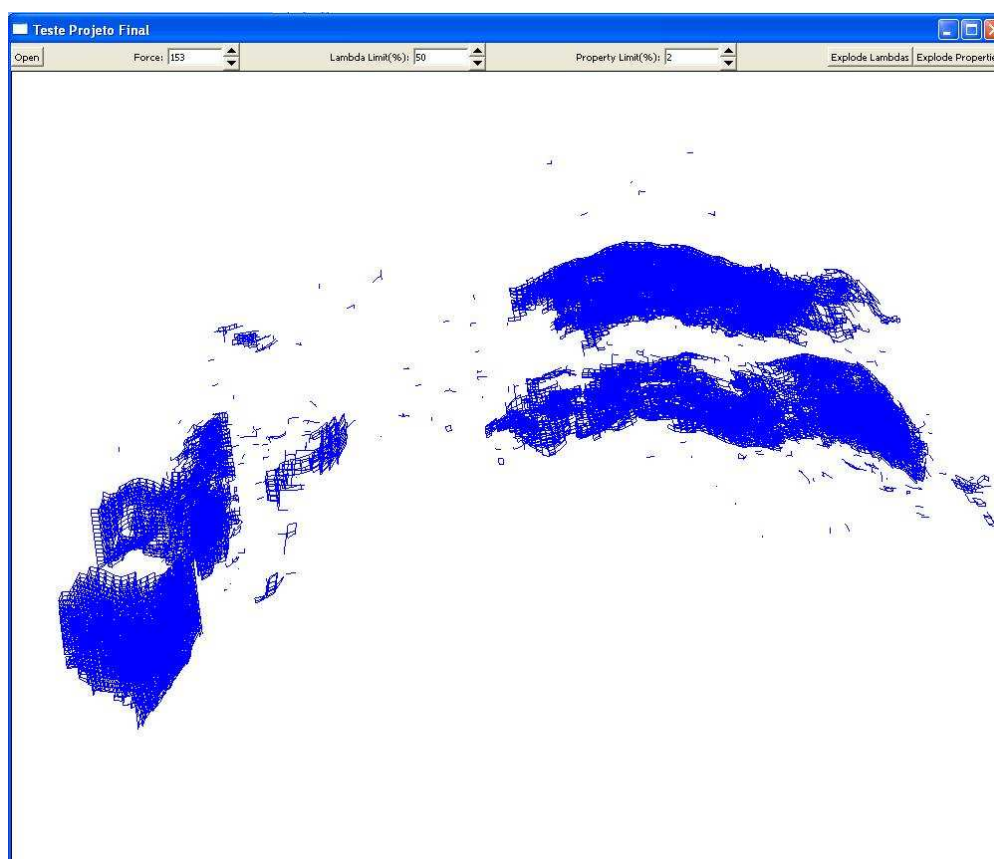


Figura 38. Modelo Namorado explodido.

Observando os resultados, pode-se concluir que após a explosão ainda há muitos pedaços isolados cuja separação não era desejada. Vendo a figura da explosão do mapa, também é possível observar que pôde se separar o pedaço com propriedade vermelha que se encontrava confinado na fatia com propriedade azul.

7.2. Comentários sobre a implementação

Durante o desenvolvimento do sistema foram encontrados inúmeros obstáculos, o que dificultou na sua implementação.

Houve uma extrema dificuldade no início do desenvolvimento do programa para calcular as restrições, pois como há muitas multiplicações de matrizes envolvidas, foi preciso testar cuidadosamente por partes. O primeiro teste consistiu em um cubo caindo sobre um plano, como mencionado anteriormente.

Outro problema encontrado, porém contornado, foi na implementação do caminhamento em grafo para calcular o centro de massa dos componentes e

sua caixa orientada. Durante toda a implementação usou-se um caminharmento em profundidade. Porém, como essa técnica é recursiva, incorporando num modelo real com cerca de 30 mil partículas fez com que a pilha de execução estourasse facilmente. Para contornar esse problema foi usado um caminharmento em largura.

Também com relação ao tamanho do modelo, houve a necessidade de reformular as estruturas de dados. Da maneira inicial que elas foram formuladas, ao testar um modelo real, houve extrema ineficiência do programa devido à multiplicação de matrizes gigantescas, uma extrema ineficiência da resolução dos sistemas lineares pelo gradiente conjugado (mais ainda que a multiplicação de matrizes), e impossibilidade de alocar memória de tal tamanho. Para contornar esse problema, as estruturas de dados foram reformuladas. Mesmo assim, a ineficiência do programa se manteve, pois mesmo reduzindo a complexidade de multiplicação de matrizes de ordem cúbica para quadrática, o tempo dessa operação continuou bastante lento.

Um resultado não esperado descrito anteriormente foi a quebra em locais não frágeis do modelo antes dos locais frágeis. Existem algumas possibilidades para justificar esse problema, como o modo em que aplicamos a força externa no modelo, isto é, sua origem ou seu módulo e direção. Outra possibilidade é o valor da percentagem que define o limite da qual os *lambdas* devem ser quebrados.

Como havia muito cisalhamento entre barras, uma das soluções propostas foi de restringir o movimento das partículas. Para tal havia duas soluções: ligar partículas por mais barras ou incorporar no sistema uma restrição radial, evitando assim que as partículas se movimentem radialmente. A primeira solução foi descartada, pois incorporar mais barras no sistema significa que seriam calculadas forças em barras que não nos interessam. A segunda hipótese foi aderida e testada, porém com resultados não esperados que não ajudasse no cálculo das forças nas barras, como mostrados anteriormente.

O integrador numérico usado em grande parte do desenvolvimento do sistema foi *Euler* Melhorado. Este método, porém, é perigoso devido a sua instabilidade e imprecisão, fazendo o modelo “explodir” após vários avanços no tempo.

8. Considerações finais

Este projeto envolve um conjunto de técnicas recém desenvolvidas para explosão de modelos científicos. Por se tratar de novas técnicas e nunca implementadas antes, os resultados obtidos foram, de uma maneira geral, imprevistos, isto é, poderiam ser tanto esperados quanto inesperados.

Apesar dos testes se concentrarem na visualização 2D, estes métodos físicos só são úteis na visualização científica em modelos 3D, pois o usuário não tem qualquer noção da representação volumétrica do modelo. Utilizando essas técnicas, é possível auxiliar engenheiros, técnicos e acadêmicos a entender a distribuição de propriedades ao longo do modelo através de sua visualização em iso-volumes.

Algumas considerações a respeito das técnicas utilizadas que foram aprendidas ao longo do desenvolvimento do projeto foram:

- Não se sabe ainda até quando quebramos o modelo. Quantos passos e quebras são necessários até obter a separação do modelo em iso-volumes ideal para o usuário visualizá-los?
- Qual o valor limite que indica a quebra das barras? Dependendo do modelo, o limite que separa as barras quebradas das não quebradas varia.
- Pelos testes realizados pôde-se observar que quanto menor é o percentual de quebra, maior é o número de quebras que precisa ser feito no modelo.
- Um dos principais motivos pelos resultados obtidos serem diferentes dos esperados (como os locais menos frágeis serem quebrados antes dos locais mais frágeis) é a aplicação da força externa no modelo. Este motivo nos leva a perguntar: Qual deve ser a força aplicada? Qual sua origem no modelo e nos componentes conexos? Qual seu módulo e direção?

Os testes 2D nos mostraram quais técnicas devem ser utilizadas para explodir o modelo. As técnicas que mostraram melhores resultados foram, sem dúvida, a aplicação de uma força radial proporcional à distância seguindo a Caixa Orientada e o algoritmo de três “passadas”. Porém, estudando a *OBB*, observou-se que essa técnica nem sempre funciona, pois como foi explicado anteriormente, nem todos os casos a Caixa define a direção da força como sendo na direção dos locais mais frágeis. O algoritmo de “três passadas”, apesar de serem necessários mais que um cálculo de forças, resulta em explosões tão

bem-definidas quanto o da Caixa Orientada. Este algoritmo, porém, resulta em explosões imprevistas, como foi o caso de se quebrar primeiro seus locais menos frágeis. Não podemos esquecer da simulação com molas, que obteve bons resultados, porém o número de simulações que precisam ser feitas até quebrar o modelo é grande.

Observando a explosão do modelo Namorado, pode-se concluir que esta resulta, além da separação dos principais iso-volumes, em vários componentes não desejados. Ou seja, existem mais componentes conexos de tamanho pequeno do que se era esperado. Uma possível explicação seria o modo em como se aplica a força externa ou também e a definição do valor limite que define quais *lambdas* serão quebrados.

Um dos maiores problemas observados foi, sem dúvida, a eficiência do programa. Ao testar o programa em um modelo real com 30 mil células e 80 mil barras, o tempo para se avançar um passo de simulação chega a quase 40 segundos. Isso acontece devido a construção da matriz A do sistema $Ax = b$, onde é preciso multiplicar as matrizes *Jacobianas*. Apesar de o autor conseguir reduzir essa conta de ordem cúbica para quadrática (usando as estruturas de dados otimizadas), o tempo para tal cálculo continua elevado. Isso nos leva a considerar a implementação de parte do programa, como a multiplicação de matrizes e o resolvidor de sistema linear, ser feito na placa gráfica, o que seria muito mais eficiente

Apesar de todos os problemas discutidos, pode-se afirmar que o método de restrição para explosão de modelos científicos é uma boa técnica, desde que se saiba aplicá-la corretamente. Há ainda muitas questões pendentes a serem resolvidas, tornando essa técnica ainda prematura. Em resumo, o autor prevê que, em um futuro próximo, o método baseado em física para visualização 3D de modelos complexos pode ser muito útil para auxiliar especialistas a melhor entendê-los.

9. Referências

1. BRUCKNER, S; GROLLER, M. Exploded Views for Volume Data, **IEEE Transactions on Visualization and Computer Graphics**, v. 12, n. 5, Set./Out. 2006. Disponível em: <http://www.cg.tuwien.ac.at/research/publications/2006/bruckner-2006-EVV/bruckner-2006-EVV-Paper.pdf>. Acesso em: 20 out. 2008.
2. LI, W.; AGRAWALA, M.; CURLESS, B.; SALESIM, D.; Automated Generation of Interactive 3D Exploded View Diagrams, **SIGGRAPH 2008**, Ago. 2008. Disponível em: <http://vis.berkeley.edu/papers/exview3D/>. Acesso em 18 out. 2008.
3. ERLEBEN, Kenny; SPORRING, Jon; HENRIKSEN, Knud; DOHLMANN, Henrik. **Physics-based Animation**. Massachusetts: Charles River Media, Inc, 2005. 817p.
4. WITKIN, A. An Introduction to Physically Based Modeling: Constraint Dynamics, **Robotics Institute**, 1997. Disponível em www.cs.cmu.edu/~baraff/pbm/constraints.pdf. Acesso em 20 mar. 2009.
5. AKENINE-MOLLER, Thomas; HAINES, Eric; HOFFMAN, Naty. **Real-Time Rendering**. Massachusetts: A K Peters, Ltd, 2008. 1027p.
6. SHREINER, Dave; WOO, Mason; NEIDER, Jackie, DAVIS, Tom. **OpenGL Programming Guide**. Addison-Wesley, 2008. 862 p.

10. Anexo I – Formulação de Restrições

Para a função de comportamento $C(\mathbf{x}) = 0, \forall t$, temos que garantir que, para um dado t_0 inicial:

$$\begin{aligned} C(t_0) &= 0, \\ \dot{C}(t_0) &= 0, \\ \ddot{C}(t) &= 0, \forall t \geq t_0 \end{aligned}$$

Pelo princípio do trabalho virtual, diz-se que as forças de restrição não podem realizar trabalho, pois não podem adicionar nem retirar energia do sistema. Como o trabalho W é dado por

$$W = \frac{\partial T}{\partial t} = \frac{1}{2} m \dot{\mathbf{x}} \dot{\mathbf{x}} + \frac{1}{2} m \dot{\mathbf{x}} \ddot{\mathbf{x}} = m \dot{\mathbf{x}} \ddot{\mathbf{x}}$$

Onde

$$T = \frac{1}{2} m \dot{\mathbf{x}} \dot{\mathbf{x}}$$

Como

$$\vec{f} + \vec{f}_c = m \ddot{\mathbf{x}},$$

Temos que

$$W = \vec{f} \cdot \dot{\mathbf{x}} + \vec{f}_c \cdot \dot{\mathbf{x}}$$

Como as forças de restrição não podem realizar trabalho, tem-se que

$$\begin{aligned} \vec{f}_c \cdot \dot{\mathbf{x}} &= 0, \\ \mathbf{x} \dot{\mathbf{x}} &= 0 \\ \vec{f}_c &= \lambda \vec{\mathbf{x}} \end{aligned}$$

Onde λ é uma incógnita.

Para evitar que imprecisões numéricas retirem as partículas da restrição, coloca-se uma “mola”. A expressão fica da seguinte forma:

$$\vec{f}_c = \lambda \vec{\mathbf{x}} - k_s C \frac{\partial C}{\partial \mathbf{x}} - k_d \dot{C} \frac{\partial C}{\partial \mathbf{x}}$$

Generalizando, dizemos que M é a matriz de massas e \vec{q} é o vetor de posições das partículas:

$$\vec{q} = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vdots \end{bmatrix}, M = \begin{bmatrix} m_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & m_n \end{bmatrix}$$

O objetivo é forçar que a função de comportamento $C(q, t) = 0$. Pela regra da cadeia, a derivada da função comportamento é:

$$\dot{C} = \frac{\partial C}{\partial q} \dot{q}$$

e a derivada da função comportamento em relação à posição é denotada pela matriz jacobiana:

$$\frac{\partial C}{\partial q} = \begin{bmatrix} \frac{\partial C_1}{\partial q_1} & \frac{\partial C_1}{\partial q_2} & \dots \\ \frac{\partial C_2}{\partial q_1} & \frac{\partial C_2}{\partial q_2} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Implicando em:

$$\begin{aligned} \dot{C} &= J \dot{q} \\ \ddot{C} &= \dot{J} \dot{q} + J \ddot{q} \\ J &= \frac{\partial \dot{C}}{\partial q} \end{aligned}$$

Sendo essa última expressão a derivada da matriz jacobiana.

Substituindo na equação de movimento $M \ddot{q} = f + f_c$ tem-se:

$$\ddot{q} = M^{-1}(f + f_c) = W(f + f_c),$$

Onde $W = \begin{bmatrix} \frac{1}{m_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{m_n} \end{bmatrix}$ é a matriz da inversa de massa de todas as partículas.

Como $\ddot{q} = J^{-1}(\ddot{C} - \dot{J} \dot{q})$, substituindo na equação de movimento, temos que:

$$J^{-1}(\ddot{C} - \dot{J}\dot{q}) = W(f + f_c)$$

Pelo princípio do trabalho virtual, diz-se que

$$f_c \dot{q} = 0, \quad \forall \dot{q} / J\dot{q} = 0 \\ f_c = J^T \lambda$$

Onde λ é um vetor de dimensão da função comportamento C chamado de *Multiplicadores de Lagrange*.

Substituindo essa expressão na anterior, e como temos que garantir que $\ddot{C} = 0$:

$$JWJ^T \lambda = -\dot{J}\dot{q} - JWf$$

Como já foi dito, as partículas tenderão a sair da restrição imposta sobre elas devido a imprecisões numéricas. Portanto, coloca-se uma “mola” para garantir estabilidade. Assim chega-se à expressão final da restrição generalizada:

$$JWJ^T \lambda = -\dot{J}\dot{q} - JWf - k_s C - k_d \dot{C}$$

Onde:

J é a matriz jacobiana,

W é a matriz de inversa de massas,

λ são os multiplicadores de lagrange,

\dot{J} é a derivada da matriz jacobiana,

\dot{q} é o vetor de velocidades das partículas,

k_s é a constante de rigidez da “mola”,

C é a função comportamento,

k_d é a constante de dissipação da “mola”,

\dot{C} é a derivada da função comportamento.