

Projeto 1 - Simulação de física 2D

Super Computação - 2018/2

Igor Montagner, Fábio {Hage, Pelicano, Orfali}, Luciano Soares

Data de entrega: 10/09

No primeiro projeto de SuperComputação iremos tratar o problema da simulação de física 2D. Faremos uma simulação simplificada em que todo corpo é representado por uma bola se movimentando em um plano. Neste contexto, técnicas de Super Computação são importantes pois permitem tanto aumentar o número de corpos na simulação, diminuir o intervalo de tempo usando nos cálculos e adotar modelos mais complexos de simulação.

Os objetivos deste projeto são:

1. Apresentar uma situação o mais realista possível de aplicação de Super Computação;
2. Aplicar conhecimentos de vetorização e programação multi-core para aumentar a acurácia da simulação;
3. Realizar um estudo detalhado do impacto que cada técnica aplicada tem no desempenho da simulação.

Um objetivo secundário, porém importante, é proporcionar uma primeira experiência de programação de um projeto minimamente complexo em C++. Alguns requisitos de qualidade e organização do projeto estão descritos na rubrica mais adiante.

Lembrete: esta atividade deve ser feita de maneira individual. Um documento contendo **Orientações sobre integridade intelectual em atividades de programação** (url: <https://tinyurl.com/comp-insper-atividades>) foi criado pela coordenação e responde algumas dúvidas comuns sobre colaboração em atividades que valem nota.

Modelo de simulação adotado no projeto

Adotaremos um modelo de simulação baseado em conceitos explorados em Física do Movimento. Cada corpo simulado possui massa e vetores representado sua aceleração, velocidade e posição.

Como visto em Física do Movimento, as seguintes equações diferenciais rejeitam a velocidade \vec{v} e a posição $\vec{r} = (x, y)$ de um objeto dada sua aceleração. Todas as equações abaixo estão escritas como vetores em duas dimensões e, portanto, podem ser decompostas em suas componentes verticais e horizontais.

$$\vec{v}(t) = \frac{\partial \vec{r}(t)}{\partial t}$$

$$\vec{a}(t) = \frac{\partial \vec{v}(t)}{\partial t} = \frac{\partial^2 \vec{r}(t)}{\partial t^2}$$

Vamos aproximar estas equações diferenciais aproximando a derivada por uma *diferença finita*.

$$\frac{\partial f}{\partial x}(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

Quanto menor o valor de h mais próxima será a aproximação. De fato, uma possível definição de derivada é

$$\frac{\partial f}{\partial x}(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

Aplicando nas equações acima obtemos as equações de movimento clássicas:

$$\vec{v}(t_0 + \Delta t) = \vec{v}(t_0) + \Delta t \vec{a}(t_0)$$

$$\vec{r}(t_0 + \Delta t) = \vec{r}(t_0) + \Delta t \vec{v}(t_0)$$

A aceleração \vec{a} é definida pelo *atrito dinâmico* entre uma bola e o chão. Sua direção é contrária a velocidade do objeto e seu comprimento é definido pela seguinte expressão

$$||a|| = \mu N = \mu mg$$

onde m é a massa da bola e $g = 9.8m/s^2$. Note que, se não houver atrito as bolas nunca parariam de se movimentar a simulação duraria para sempre!

Choque entre as bolas e a borda do campo

Ao se movimentarem é natural que as bolas se choquem com as bordas do campo de simulação e entre si. Vamos propor três modelos de colisão de dificuldade e realismo crescente.

Modelo 1: geométrico

O modelo de colisão geométrica foca somente na interpretação geométrica das colisões. Este modelo ignora o efeito da massa e das velocidades dos objetos na determinação dos vetores de velocidade após o choque. Os vetores velocidade resultante (azul na Figura abaixo) são apenas refletidos em relação a tangente do ponto de contato (em vermelho).

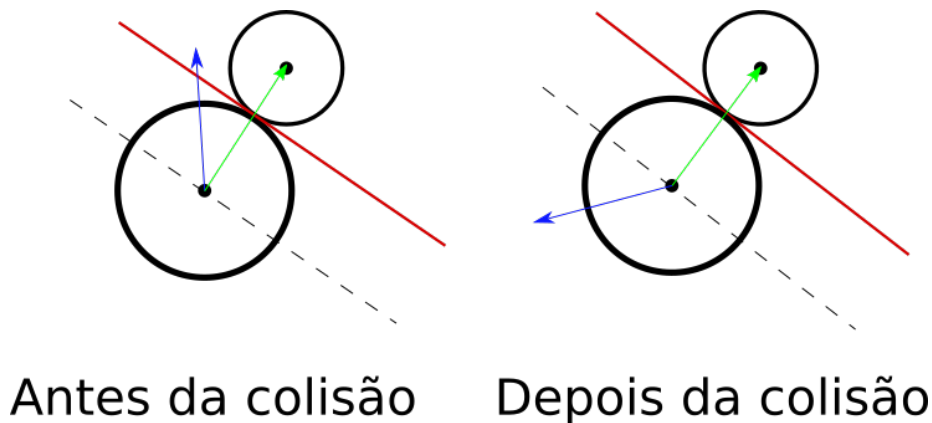


Figure 1: Modelo de colisão geométrico: direção é alterada independente da massa e velocidade dos objetos

Isto pode ser feito projetando o vetor velocidade (azul) no vetor que liga os centros de massa dos objetos (em verde) e inverter o sentido desta componente.

Modelo 2: colisão elástica

O modelo de colisão elástica considera que tanto a *Quantidade de movimento* quanto a *Energia cinética* se mantêm constantes após a colisão. Ou seja, dados dois objetos com massas m_1, m_2 e velocidades antes da colisão v_1, v_2 , suas velocidades após a colisão u_1, u_2 obedecem as seguintes regras. Como os objetos tem massas e velocidades diferentes, o choque entre os objetos altera o módulo dos vetores velocidade e não somente sua direção, como no exemplo anterior.

$$m_1 v_1 + m_2 v_2 = m_1 u_1 + m_2 u_2$$

$$\frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_1 u_1^2 + \frac{1}{2} m_2 u_2^2$$

Para tornar as contas mais simples vamos supor que $v_2 = 0$ (o que é equivalente a adotar como referencial o objeto 2). Então isolamos a velocidade u_2 , resultando em

$$u_2 = \frac{m_1(u_1 - v_1)}{m_2}$$

$$m_1 v_1^2 = m_1 u_1^2 + m_2 \left(\frac{m_1(u_1 - v_1)}{m_2} \right)^2$$

$$[m_1(m_1 + m_2)] u_1^2 - [2m_1^2 v_1] u_1 + [m_1^2 v_1^2 - m_2 m_1 v_1^2] = 0$$

Primeiramente, existe uma solução muito simples para este sistema de equações: $v_1 = u_1, u_2 = 0$. Claramente esta solução não é a que desejamos, pois segundo ela o objeto 1 pode continuar se movendo e ignorar o choque. A outra solução da equação acima é

$$u_1 = \frac{(m_1 - m_2)}{m_1 + m_2} v_1,$$

que resulta em

$$u_2 = \frac{2m_1}{m_1 + m_2} v_1.$$

Note que as equações acima foram dadas para o caso em uma dimensão. Porém, podemos decompor cada vetor velocidade em duas componentes: uma na direção da colisão (vetor rosa) e uma perpendicular a colisão (vetor amarelo). A velocidade amarela não tem nenhum efeito no choque, fazendo que o problema seja reduzido a 1D na direção da força rosa.

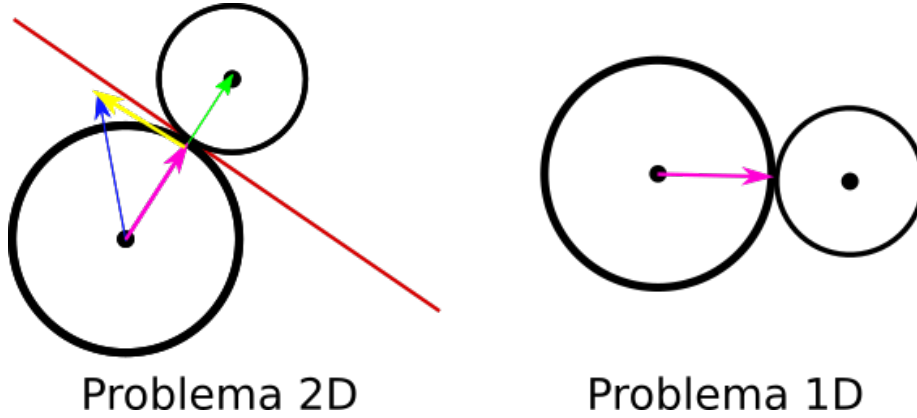


Figure 2: Redução do choque 2D para um problema 1D

Podemos então resolver o problema 1D e depois reconstruir o novo vetor velocidade mantendo a componente amarela e somando-a com a nova componente rosa

determinada usando as equações de conservação de momento e energia cinética. Assim como no caso 1D, mudar o referencial para um dos corpos ter velocidade 0 facilita muito as contas.

Modelo 3: colisão parcialmente inelástica

O modelo de colisão parcialmente inelástica modifica o modelo acima adicionando um fator α que representa a porção de energia que se mantém após o choque. Ou seja, quando ocorre um choque $(1 - \alpha)$ da energia cinética do sistema é perdida. Isto significa modificar a segunda equação do modelo acima da seguinte maneira.

$$\alpha\left(\frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2\right) = \frac{1}{2}m_1u_1^2 + \frac{1}{2}m_2u_2^2$$

Tratamos este problema em duas dimensões da mesma maneira que o anterior: projetando na direção da colisão, realizando o choque 1D e depois reconstruindo o novo vetor velocidade a partir da nova velocidade encontrada. Em geral α é um fator que depende dos materiais dos objetos envolvidos. Neste trabalho adotaremos um α_w para a colisão com as bordas da simulação e α_b para colisões entre bolas.

A adoção deste modelo de colisão envolve repetir você mesmo as contas do modelo anterior considerando agora a segunda equação para a energia cinética. As novas velocidades dependerão, agora, do valor α .

Atenção: este modelo NÃO é equivalente a simplesmente multiplicar as velocidades obtidas no modelo anterior por α !

Entrada e saída do programa

A entrada de seu programa estará no seguinte formato:

```
W H N
mu alpha_w alpha_b
<id> <raio> <massa> <x0> <y0> <vx0> <vy0>
```

Onde

- W e H são as dimensões do campo da simulação. Nenhum corpo pode sair deste retângulo e, se algum corpo tocar a borda da simulação deve ocorrer uma colisão.
- mu é o coeficiente de atrito dinâmico do campo.
- alpha_w é a proporção de energia cinética mantida após uma colisão entre um corpo e a parede. Se E_wall=1 a colisão é perfeitamente elástica.
- alpha_b é a proporção de energia cinética mantida após uma colisão entre dois corpos. Se E_ball=1 a colisão é perfeitamente elástica.

- N é o número de corpos a serem simulados.
- g é a aceleração da gravidade na simulação.

Cada linha a seguir define um novo corpo e suas propriedades (raio, massa, posição e velocidade inicial). Estamos supondo que todos os corpos são feitos do mesmo material, logo a colisão entre corpos segue a mesma regra de conservação (ou não) de energia cinética. Todos os valores devem ser escritos no sistema internacional (SI).

Para determinar o valor de Δt seu programa deve ler a variável de ambiente `DELTA_T`. Se ela não for passada você pode assumir $\Delta t = 0.01$. Se a variável `GUI=0` for passada seu programa deverá imprimir, a cada segundo da simulação, o tempo atual da simulação e uma lista contendo todos os objetos e seus estados atuais (no mesmo formato da entrada menos as primeiras duas linhas). Se houver um choque entre dois objetos seu programa deverá imprimir `CHOQUE ID_0 ID_1 DELTA_T`. Usaremos como convenção que $ID_0 < ID_1$.

Também deve ser possível selecionar a estratégia de paralelização desejada usando a variável de ambiente `STRATEGY`. Os valores possíveis são *seq* (sequencial), *simd* (para usar somente vetorização SIMD), *omp* (para usar somente OpenMP) ou *best* (a estratégia que você identificou como melhor).

Avaliação

Seu trabalho será avaliado em três critérios: complexidade da simulação C , qualidade do relatório R e uso de técnicas de SuperComputação S . A nota final será dada por

$$\begin{cases} C(0.4R + 0.6S) & \text{se } R, S \geq 5 \\ \min(R, S) & \text{c.c.} \end{cases}$$

Os modelos apresentados são incrementalmente mais complexos, o que significa também maior necessidade de poder computacional. Logo, implementar um modelo complexo oferece mais oportunidades de aplicação dos conceitos de Super Computação do que implementar um modelo simples. Em especial, fazer um modelo complexo executar em tempo real (tempo para calcular todos os movimentos é menor que o Δt adotado) é muito mais difícil que fazer o mesmo para um modelo simplificado. Por estas razões, a nota de modelagem C segue as regras abaixo. Note que é necessário implementar pelo menos um dos modelos corretamente, caso contrário sua nota será multiplicada por 0.49.

- 0,49: somente equações de movimento considerando atrito, sem choque
- 0,7 : equações de movimento corretamente implementadas mais choque geométrico
- 0.85 : choque elástico

- 1.0 : choque parcialmente inelástico

A rubrica usada para avaliar o relatório será a mesma disponibilizada na atividade de avaliação por pares.

O uso de técnicas de SuperComputação será avaliado segundo três critérios básicos: Qualidade (0.2), SIMD (0.2) e Multi-core (0.6). O critério SIMD é de implementação opcional. Nos itens multi-core e SIMD serão avaliadas as estratégias de implementação testadas e os ganhos de desempenho obtidos. Existe interseção entre alguns itens da rubrica do relatório e o que será avaliado neste item, mas neste caso o foco é em como os problemas foram analisados e decompostos em tarefas independentes. Existem estratégias não óbvias que podem ser muito vantajosas.

Será disponibilizado o código de um visualizador para que vocês possam verificar visualmente o resultado das suas simulações. Alunos que propuserem (e fizerem) modificações no visualizador podem receber bônus de até 1,0 na nota final desde que $R, S \geq 5$.

Requisitos de qualidade de software

Apesar da disciplina ser focada em alto desempenho, alguns aspectos básicos de qualidade de software serão exigidos neste projeto.

1. Seu projeto deve *CMake* para gerenciar a compilação.
2. Seu código deve ser organizado e comentado (quando necessário). Programas gigantes inteiros no `main` ou com prints e medição de tempo em funções da simulação são más práticas de programação.
3. Apesar do visualizador ser conveniente para visualizar o seu programa, ele não serve como medidor de desempenho. Você deve criar um segundo executável que mostre as diferenças de desempenho entre as técnicas testadas.
4. Você deve criar um pequeno conjunto de arquivos de teste que exemplifique a corretude de seu programa. Alternativamente, você pode fazer o mesmo usando testes de unidade.

Apêndice

Seja \vec{v} um vetor com direção θ e tamanho $||v||$, suas componentes podem ser obtidas usando as seguintes fórmulas:

$$v_x = ||v|| \cos \theta$$

$$v_y = ||v|| \sin \theta$$

Rubrica para SuperComputação – Relatórios

Critério	Conceito				
	I	D	C	B	A
Descrição do problema	O relatório não descreve o problema tratado.	O relatório descreve o problema de maneira errada ou que induz ao erro.	O problema tratado é descrito de maneira sucinta. Informações essenciais não são omitidas. O leitor que não está familiarizado com o problema pode ter dificuldades de entendimento.	O problema é descrito de maneira detalhada. O texto pode ser entendido por leitores "leigos", mas requer consultas a materiais externos.	O problema é descrito de maneira detalhada e sua essência pode ser entendida por leitores "leigos", mas não familiarizados. Aponha referências com mais detalhes para os interessados.
Descrição da implementação e da execução dos testes	O relatório não descreve a implementação usada.	O relatório descreve brevemente a implementação, mas não descreve que os testes foram realizados.	O relatório descreve brevemente a implementação e o hardware em que os testes foram realizados, relacionar a implementação com o desempenho obtido.	O relatório descreve em detalhes a implementação e o hardware em que os testes foram feitos. São apresentados os detalhes da implementação com seu desempenho esperado.	O relatório descreve em detalhes a implementação e o hardware em que os testes foram feitos. São apresentados os detalhes da implementação com seu desempenho esperado.
Reprodutibilidade	O relatório não está acompanhado de um programa que possa reproduzir seus resultados ou está acompanhado de um programa sem documentação.	Um programa com documentação reproduz o relatório, porém os testes é um processo que requer muita intervenção manual.	Um programa com documentação reproduz os testes, mas a documentação fornecida é confusa ou a saída do programa é de difícil interpretação.	Um programa com documentação é fornecido. É possível reproduzir os testes de maneira simples e a saída do programa é de fácil leitura.	Um programa com documentação acompanha o relatório. É possível gerar automaticamente todas as tabelas e gráficos usados usando um script.
Tamanho das instâncias de teste e quantidade de cenários testados	Não entregou os casos de teste.	Realiza testes apenas com um ou dois exemplos pequenos ou triviais.	Realiza um conjunto pequeno de testes mas consegue demonstrar alguma diferença significativa de desempenho.	Realiza um conjunto abrangente de testes, levando em conta o consumo de memória e tempo.	Leva a capacidade de sua máquina próxima de seu limite, realizando também testes intermediários de desempenho.
Medidas de tempo e memória.	Não mede nenhuma das três de maneira correta ou consistente.	São feitas medições usando o mesmo método medido consumo de memória.	É medido consumo de tempo usando std::chrono. E feita alguma medição de consumo de memória. São feitos comentários breves sobre os resultados no texto.	Ambas medidas são feitas e seus resultados são apresentados de maneira detalhada no relatório.	Ambos tempo e memória são medidos corretamente e é feita uma análise de como estas medidas afetam o tempo de trabalho de entrada e com a capacidade de memória de sua máquina.
Facilidade de leitura	O relatório não apresenta os dados brutos	Os resultados estão explicados diretamente no texto de maneira confusa. Não usa tabelas ou gráficos para mostrar as diferenças de desempenho.	Ilustra com gráficos ou tabelas, mas possui problemas de formatação que dificultam a leitura e os resultados desses recursos no texto.	Ilustra as diferenças de desempenho com tabelas e comentários brevemente seus resultados.	Ilustra as diferenças de desempenho com tabelas e gráficos e os interpreta de maneira extensiva no texto.

avaliado: _____ Trabalho _____ Nome: _____ Seu

Figure 3: Rubrica para avaliação do relatório

Da mesma maneira, dadas as componentes (v_x, v_y) , o tamanho e direção de um vetor podem ser obtidos pelas seguintes fórmulas:

$$\|v\| = \sqrt{v_x^2 + v_y^2}$$

$$\theta = \arccos \frac{v_x}{\|v\|}$$

Dados dois vetores \vec{a} e \vec{b} , a projeção de \vec{a} em \vec{b} é obtida pela expressão $a_1\vec{b}$, sendo que a_1 é obtido pela expressão

$$a_1 = \frac{a_x b_x + a_y b_y}{\|\vec{b}\|^2}$$

Veja abaixo uma ilustração deste conceito.

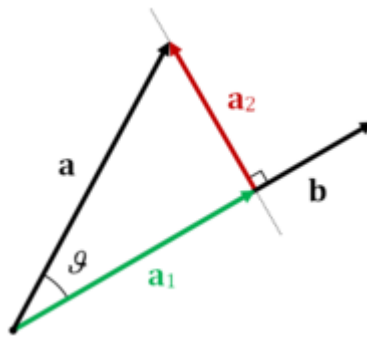


Figure 4: Projeção de vetores