MAC2166 – Introdução à Ciência da Computação

Escola Politécnica - Computação / Elétrica - Primeiro Semestre de 2022

Exercício-Programa 2 (EP2)
Data de Entrega: **4 de junho de 2022**

Para se preparar bem para o desenvolvimento de seu EP2, cuja descrição se inicia na próxima página, leia com atenção as instruções abaixo.

- Utilize **somente** os recursos da linguagem que aprendeu nas aulas.
- Veja em `https://www.ime.usp.br/~mac2166/infoepsC/` instruções de entrega dos exercícios-programa e atente para as instruções de preenchimento do cabeçalho do seu programa.
- Leia um FAQ sobre compilação em `https://www.ime.usp.br/~mac2166/compilacao/`.
- Sempre compile seus programas com as opções **-Wall -ansi -pedantic -O2**.
- **Importante**: seu programa deve
  - funcionar para qualquer entrada que está de acordo com o enunciado;
  - estar em conformidade com o enunciado;
  - estar bem estruturado;
  - ser de fácil compreensão, com uso idiomático da linguagem C.

EP2: Corpos Celestes

Neste exercício-programa você implementará um simulador simples capaz de prever bastante bem a trajetória de três corpos celestes que interagem entre si através de atração gravitacional.

1 Simulação de um corpo

1.1 Atração gravitacional

Sejam B_1 e B_2 dois corpos de massas m_1 e m_2. Supomos que B_1 e B_2 têm forma esférica. Seja d a distância entre os centros de B_1 e B_2. A lei de gravitação de Newton diz que há entre B_1 e B_2 uma força de atração gravitacional de magnitude

$$G\,\frac{m_1 m_2}{d^2}, \tag{1}$$

onde G é uma constante universal, cujo valor é $6.67 \times 10^{-11}\,\mathrm{Nm}^2/\mathrm{kg}^2$.

Suponha agora que há um terceiro corpo no sistema, digamos B_0, de massa m_0 e também de forma esférica. Há atração gravitacional entre B_0 e B_1 e entre B_0 e B_2, cujas magnitudes podem ser calculadas pela lei de Newton (1). Para saber a força total que age sobre, digamos, o corpo B_0, devemos calcular a soma (vetorial) das forças F_1 e F_2 que os corpos B_1 e B_2 exercem sobre B_0.

Observações. Neste exercícios, trataremos corpos com dimensões desprezíveis em relação às distâncias envolvidas, de forma que podemos supor que os corpos são pontos.

1.2 Trajetória de um corpo

Seja r a posição de um corpo B em um instante t_0. Neste exercício, trabalharemos em duas dimensões, de forma que r pode ser escrito como um par $(r_x, r_y) \in \mathbb{R}^2$. Seja $v = (v_x, v_y)$ a velocidade de B e que essa velocidade seja constante ao longo do tempo. Sabemos então que, no instante $t_0 + \Delta t$, o corpo B estará na posição $r + (\Delta t)v$. Se B sofre uma aceleração $a = a(t)$ no instante t, então a velocidade $v(t)$ de B no instante t varia ao longo do tempo e $v(t) = v(t_0) + \int_{t_0}^{t} a(t)\,dt$. Assim, a posição de B no instante $t_0 + \Delta t$ será $r + \int_{t_0}^{t_0+\Delta T} v(t)\,dt$.

Neste exercício (felizmente :-)), você deve ignorar as integrais que ocorrem acima, e deve adotar uma forma aproximada para determinar a posição do corpo B no instante $t_0 + \Delta t$. Vamos supor que conhecemos a aceleração $a(t_0)$ no instante t_0. O que fazemos é atualizar o valor atual v da velocidade para o valor $v' = v + (\Delta t)a(t_0)$ e declaramos que B estará na posição $r + (\Delta t)v'$ no instante $t_0 + \Delta t$.

Observação. Quando Δt não é muito grande em relação às outras grandezas envolvidas, a aproximação acima é boa.

1.3 Trajetória sob uma força gravitacional

Seja m a massa do corpo B da Seção 1.2. Suponha que B sofre uma atração gravitacional dada por uma força $F = (F_x, F_y)$, devido a outros corpos presentes no sistema. A segunda lei de Newton diz que B sofre então uma aceleração a dada por $a = (1/m)F$. Note que, conforme B muda de posição, a força gravitacional F que ele sofre pode mudar, dado que F depende da posição relativa de B em relação aos outros corpos do sistema. De qualquer forma, conhecendo a posição r, a velocidade v de B e a força F no instante t_0, podemos usar o método descrito na Seção 1.2 para determinar (aproximadamente) a posição de B no instante $t_0 + \Delta t$ (nesse método, ignoramos que F pode mudar entre t_0 e $t_0 + \Delta t$). Fazemos o seguinte:

$$a \leftarrow (1/m)F \tag{2}$$

$$v \leftarrow v + (\Delta t)a \tag{3}$$

$$r \leftarrow r + (\Delta t)v \tag{4}$$

O valor de r calculado em (4) acima é a posição de B no instante $t_0 + \Delta t$ (aproximadamente).

Observação. Note que, em (2), (3) e (4), as quantidades a, F, v e r são vetores. Assim, por exemplo, para executar (2), você precisa fazer $a_x \leftarrow F_x/m$ e $a_y \leftarrow F_y/m$, onde $a = (a_x, a_y)$ e $F = (F_x, F_y)$. Algo análogo vale para (3) e (4).

1.4 Exemplo

Suponha que a Terra está na origem $(0, 0) \in \mathbb{R}^2$, e que ela não se move. Suponha que, no instante $t_0 = 0$, a Lua está na posição

$$(r_x^{(0)}, r_y^{(0)}) = (3.63 \times 10^8, 0) \tag{5}$$

e tem velocidade

$$(v_x^{(0)}, v_y^{(0)}) = (0.0, 1072). \tag{6}$$

Note que, em (5) e (6), omitimos as unidades. Quando fazemos isso, supomos que a unidade adotada é a do sistema internacional (assim, a unidade em (5) é o metro e em (6) é m/s). Vamos adotar 7.342×10^{22} kg como a massa da Lua e 5.972×10^{24} kg como a massa da Terra.

Com as informações acima, usando (1), vemos que

$$F_0 = (-2.21946 \times 10^{20}, 0) \tag{7}$$

é a força que age sobre a Lua no instante $t_0 = 0$ (os números são dados com no máximo 6 dígitos significativos). Executando os passos (2), (3) e (4) com $\Delta t = 3600\,\mathrm{s}$ (uma hora), vemos que a posição da Lua no instante $t_1 = 3600$ é

$$r(t_1) = r(3600) = (3.62961 \times 10^8, 3.8592 \times 10^6). \tag{8}$$

A força que age na Lua no instante $t_1 = 3600$ é

$$F_1 = (-2.21956 \times 10^{20}, -2.35996 \times 10^{18}), \tag{9}$$

e executando os (2), (3) e (4) com $\Delta t = 3600\,\mathrm{s}$ novamente, vemos que a posição da Lua no instante $t_2 = 7200$ é

$$r(t_2) = r(7200) = (3.62882 \times 10^8, 7.71798 \times 10^6). \tag{10}$$

Podemos repetir o processo acima para obter a posição da Lua a cada hora. Repetindo o processo até o instante $T = 864000 = 10 \times 24 \times 3600$ (dez dias), obtemos as posições $r(3600)$, $r(2 \times 3600)$, ..., $r(240 \times 3600)$. O arquivo `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/moon_10.out` contém a posição original $(3.63\times10^8, 0)$ da Lua e as 240 posições acima. O mar de números no arquivo `moon_10.out` não é muito fácil de se "entender". Plotando os 241 pontos em `moon_10.out`, obtemos a imagem no arquivo `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/moon_10.pdf`. Vemos assim que nossa simulação faz sentido.

Executando o processo acima para $T = 2332800$ (27 dias) e $T = 86400000$ (1000 dias), obtemos os arquivos `moon_27.*` e `moon_1000.*` em `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/`.

Se pomos como velocidade inicial da Lua

$$(v_x^{(0)}, v_y^{(0)}) = (0.0, 1400), \tag{11}$$

e simulamos sua trajetória por 250 dias e por 300 dias, obtemos os arquivos `moon14_250.*`

e `moon14_300.*` em `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/`. Note como a órbita da Lua é diferente com (12). Compare os valores em (6) e (12) para entender de onde vem a diferença, pelo menos intuitivamente.

Veja como seria a trajetória da Lua (300 dias) com velocidade inicial

$$(v_x^{(0)}, v_y^{(0)}) = (0.0, 1500) \tag{12}$$

nos arquivos `moon15_300.*` em `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/`.

Observação. Veremos mais à frente como usar o software gnuplot para produzir imagens como a imagem em `moon_10.pdf` a partir de arquivos como `moon_10.out`.

2 Sistema de vários corpos

O material discutido até agora é suficiente para simularmos sistemas celestes com vários corpos. Vamos agora considerar o caso em que temos três corpos B_0, B_1 e B_2. Suponha que temos as posições r_0, r_1 e r_2 e velocidades v_0, v_1, e v_2 desses corpos no instante t_0. Ademais, suponha que esses corpos têm massa m_0, m_1 e m_2. Suponha também que queremos descobrir a trajetória desses corpos até um dado instante T, determinando suas posições nos instantes t_0, $t_1 = t_0 + \Delta t$, $t_2 = t_0 + 2\Delta t$, $\ldots$, onde Δt é dado.

Para determinar a posição do corpo B_0 no instante t_1, determinamos primeiro a força gravitacional F_0 que age sobre B_0 por conta dos corpos B_1 e B_2 (veja a Seção 1.1). Agora executamos os passos (2), (3) e (4) para obter a posição $r_0(t_1)$ de B_0 no instante t_1. Note que devemos executar o procedimento análogo para os corpos B_1 e B_2, para obter suas posições $r_2(t_1)$ e $r_3(t_1)$ no instante t_1. Repetimos esse processo para obter a posições de B_0, B_1 e B_2 nos instantes t_2, t_3, etc.

Exemplo. No que segue, para especificarmos os dados de um corpo no instante $t_0 = 0$, vamos fornecer uma quíntupla, a saber,

$$(r_x, r_y, v_x, v_y, m). \tag{13}$$

Isto é, vamos fornecer as coordenadas da posição do corpo no instante t_0, as duas componentes do vetor velocidade no instante t_0, e sua massa. Neste exemplo, temos três corpos B_0, B_1 e B_2, caracterizados pelas seguintes três quíntuplas:

$$(0.0, 0.0, 0.0, 0.0, 1.9885 \times 10^{30}), \tag{14}$$

$$(1.47095 \times 10^{11}, 0.0, 0.0, 30290, 5.972 \times 10^{27}) \tag{15}$$

e

$$(1.36732 \times 10^{11}, 0.0, 0.0, 35290, 7.342 \times 22). \tag{16}$$

Simulando a evolução de B_0, B_1 e B_2 com o método acima, usando $\Delta t = 3600$ até $T = 31104000$ (360 dias), obtemos o arquivo em `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/fun.out`. Note que a primeira linha desse arquivo especifica as coordenadas de B_0, B_1 e B_2 no instante $t_0 = 0$. A segunda linha especifica as coordenadas desses corpos no instante t_1 e assim por diante. Plotando esses pontos, obtemos a imagem em `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/fun.pdf`.

3 Seu programa

Neste EP, você deve escrever um programa, digamos `ep2.c`, que recebe na entrada padrão três quíntuplas especificando três corpos como acima e os valores de T e Δt, e que envia para a saída padrão as coordenadas dos três corpos nos instantes $t_0 = 0$, $t_1 = t_0 + \Delta t$, $t_2 = t_0 + 2\Delta t$, ..., $t_k = t_0 + k\Delta t$, onde k é o maior inteiro tal que $t_k \leq T$.

Exemplo. Com a entrada `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/fun.in`, seu programa deve produzir a saída `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/fun.out`. De fato, o arquivo `fun.out` foi obtido executando-se

```
$ ep2 < fun.in > fun.out
```

Se você usa o PowerShell, você precisará fazer

```
PS C:\Users\... >  cat fun.in | .\ep2.exe > fun.out
```

Seu programa será executado como acima, para verificar se ele está de acordo com as especificações do enunciado.

Observação. O cômputo de trajetórias como descrevemos acima envolve aritmética de números em ponto flutuante. Assim sendo, pode ocorrer de seu programa produzir, com entrada `fun.in`, uma saída que é levemente diferente da saída `fun.out` fornecida acima, por conta de erros de arredondamento.

Alguns detalhes de implementação. Sua implementação deve conter, obrigatoriamente, as funções cujos protótipos são dados a seguir.

1. Função `dist()`:

```
double dist(double p1x, double p1y, double p2x, double p2y);
```

Esta função devolve a distância entre os pontos $(p1x, p1y)$ e $(p2x, p2y)$.

2. Função `forca()`:

```
double forca(char c, int i,
             double x0, double y0, double m0,
             double x1, double y1, double m1,
             double x2, double y2, double m2);
```

Esta função recebe um caractere $c \in \{$`'x'`, `'y'`$\}$, um inteiro $i \in \{0, 1, 2\}$, e dados que especificam três corpos: corpo B_0 na posição (x_0, y_0) com massa m_0, corpo B_1 na posição (x_1, y_1) com massa m_1, e corpo B_2 na posição (x_2, y_2) com massa m_2. Esta função devolve a componente c da força gravitacional total que os corpos B_j com $j \neq i$ exercem sobre o corpo i.

3. Função `atualize()`:

```
void atualize(double *x, double *y, double *vx, double *vy,
              double ax, double ay, double dt);
```

Esta função recebe a posição $(*x, *y)$, a velocidade $(*vx, *vy)$ e a aceleração (ax, ay) de um corpo em um dado instante, digamos t. Esta função atualiza os valores de $(*vx, *vy)$ e $(*x, *y)$ usando (3) e (4) acima, onde $\Delta t =$ `dt`, de forma que $(*x, *y)$ torna-se a posição do corpo no instante $t + \Delta t$.

Você poderá usar outras funções, se julgar necessário.

4 gnuplot

Para plotar as saídas produzidas pelo seu programa, você deve usar o programa gnuplot. Para instalar o gnuplot em sua máquina, visite `http://www.gnuplot.info`. Este software é muito útil e você poderá usá-lo em seus projetos futuros. Para ver umas imagens produzidas pelo gnuplot, veja `http://gnuplot.sourceforge.net/demo/`.

Você precisará conhecer uns poucos comandos do gnuplot para este EP. Uma vez instalado o gnuplot, execute o gnuplot para obter seu *prompt*:

```
gnuplot>
```

Neste *prompt*, execute `set size ratio -1`:

```
gnuplot> set size ratio -1
```

O comando acima faz com que as "escalas" do eixo x e do eixo y sejam a mesma. O gnuplot tem a noção de *diretório corrente*. Quando dizemos para o gnuplot plotar os pontos em um arquivo (algo que faremos mais à frente), ele supõe que o arquivo está neste diretório corrente (a menos que especifiquemos o nome do arquivo com seu *path name*).

Para produzir a imagem no arquivo `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/moon_10.pdf`, você pode fazer

```
gnuplot> plot "moon_10.out" w d
```

Omitindo `w d` (que é uma abreviação de `with dots`), um símbolo diferente é usado para indicar os pontos. Experimente executar o comando acima sem o `w d`. *É importante observar que, nos exemplos acima, supomos que o arquivo* `moon_10.out` *está no diretório corrente do gnuplot.*

Para produzir a imagem no arquivo `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/fun.pdf`, você pode fazer

```
gnuplot> plot "fun.out" u 1:2 w d t "B0", "" u 3:4 w d t "B1", "" u 5:6 w d t "B2"
```

Acima, `u` é uma abreviação de `using`. Quando dizemos `u 1:2`, estamos pedindo que gnuplot use a primeira e a segunda coluna de números em `fun.out` como as coordenas x e y dos pontos a serem plotados (com "título" `B0`).

5 Entrega

Neste EP, você deve entregar seu programa `ep2.c`. Você pode entregar também até 3 arquivos de entrada (como o arquivo `fun.in`) e as imagens correspondentes. Se você conseguir produzir entradas que produzem órbitas interessantes, você poderá receber um prêmio (a definir).

6 Mais um exemplo

Damos aqui um sistema interessante de três corpos. Simulamos o sistema durante intervalos de tempo variados, a saber, simulamos com valores de T iguais a 1×10^7, 2×10^7, 6.3×10^7, 6.35×10^7 e 6.5×10^7. Os arquivos de entrada que usamos são os arquivos `3body.in`, `3body_200.in`, `3body_630.in`, `3body_635.in` e `3body_650.in`, todos disponíveis em `https://www.ime.usp.br/~yoshi/2022i/mac2166/sandbox/EPs/EP2/`. As imagens correspondentes aparecem em `3body.pdf`, `3body_200.pdf`, `3body_630.pdf`, `3body_635.pdf` e `3body_650.pdf`.