Introdução a Métodos Computacionais em Física
Módulo 3

Leonardo Cabral

22 de agosto de 2019

Objetivo 1

Estudo do movimento de sistemas oscilatórios conservativos. Uso de algoritmos *simpléticos*.

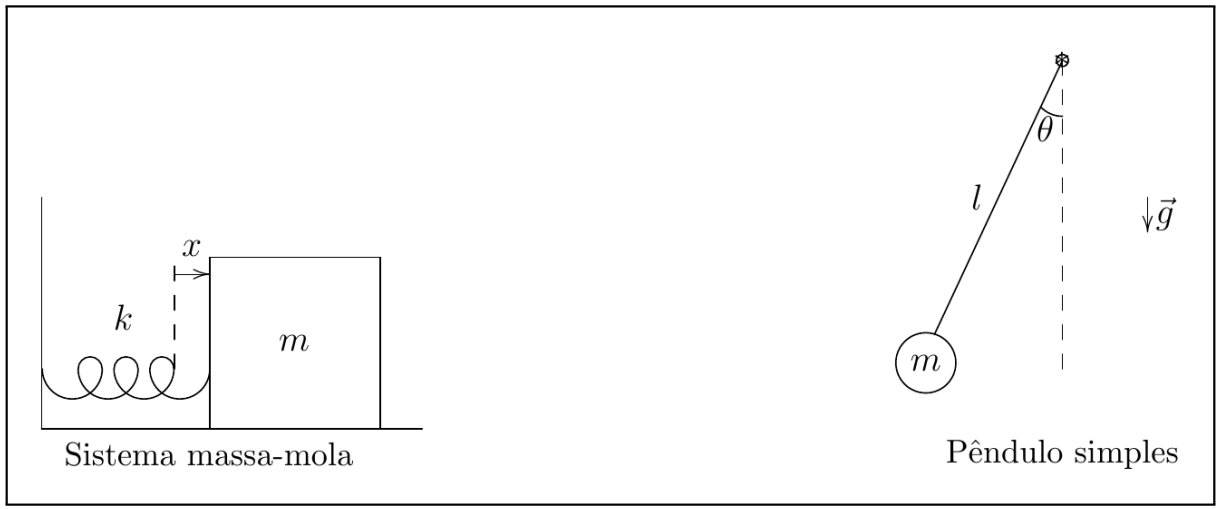
Objetivo 2

Estudo de sistemas oscilatórios não-conservativos e forçados.

Objetivo 3

Observação da evolução temporal de conjuntos de pontos no espaço de fase.

Oscilações: sistemas conservativos



Oscilador harmônico simples

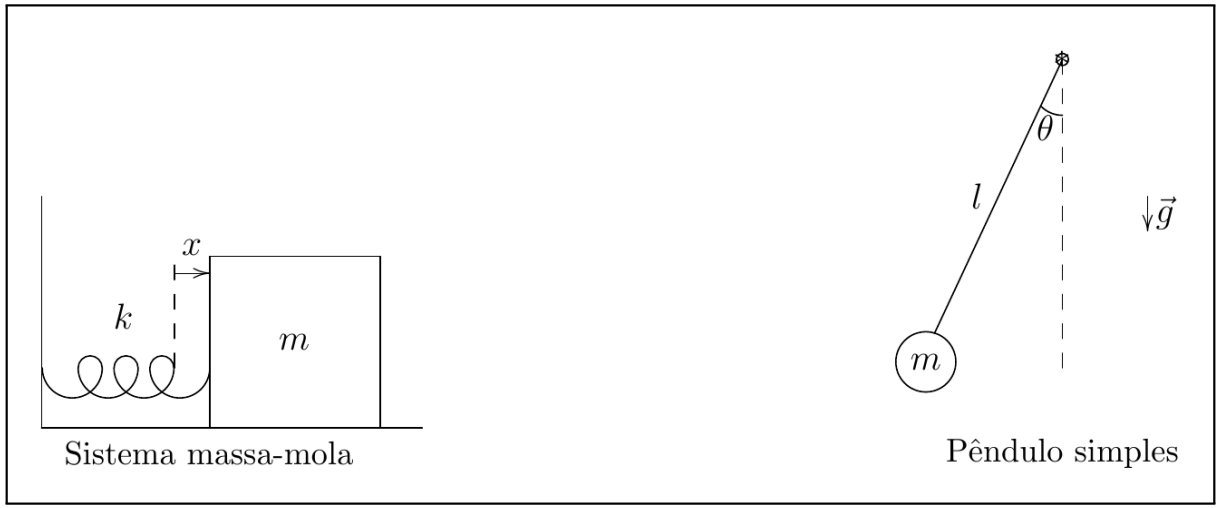
$$\mathcal{E} = \frac{1}{2}m\left(\frac{dx}{dt}\right)^2 + \frac{kx^2}{2}, \qquad \frac{d^2x}{dt^2} = -\omega_0^2 x, \qquad \omega_0 = \sqrt{\frac{k}{m}}$$

Solução analítica simples:

$$x(t) = A\cos\omega_0 t + B\,\mathrm{sen}\,\omega_0 t,$$

A e B dependem das condições iniciais.

Oscilações: sistemas conservativos



Pêndulo simples de comprimento l

$$\mathcal{E} = \frac{ml^2}{2}\left(\frac{d\theta}{dt}\right)^2 + mgl\,(1-\cos\theta)\,, \qquad \frac{d^2\theta}{dt^2} = -\frac{g}{l}\,\mathrm{sen}\,\theta.$$

Solução analítica simples para pequenas oscilações. Caso geral requer uso de funções especiais.

Oscilações: sistemas conservativos

Métodos simpléticos

- Em sistemas conservativos, além da anergia ser conservada, o volume ocupado por um conjunto de de estados no *espaço de fase* permanece constante!

Oscilações: sistemas conservativos

Métodos simpléticos

- Em sistemas conservativos, além da anergia ser conservada, o volume ocupado por um conjunto de de estados no *espaço de fase* permanece constante!
- Um método simplético é aquele em que o algoritmo satisfaz estas condições.

Oscilações: sistemas conservativos

Métodos simpléticos

- Em sistemas conservativos, além da anergia ser conservada, o volume ocupado por um conjunto de de estados no *espaço de fase* permanece constante!
- Um método simplético é aquele em que o algoritmo satisfaz estas condições.
- Utilizam, em geral, o formalismo hamiltoniano para construção do algoritmo.

Oscilações: sistemas conservativos

Métodos simpléticos

- Em sistemas conservativos, além da anergia ser conservada, o volume ocupado por um conjunto de de estados no *espaço de fase* permanece constante!
- Um método simplético é aquele em que o algoritmo satisfaz estas condições.
- Utilizam, em geral, o formalismo hamiltoniano para construção do algoritmo.
- Métodos de Euler-Cromer e de Verlet são simpléticos.

Atividade 1

Elabore rotinas que calculem:

1. a posição x e a velocidade $v = \dfrac{dx}{dt}$ do bloco no sistema massa-mola em função do tempo,
2. a posição angular θ e a velocidade angular $\dot{\theta}$ em função do tempo para o pêndulo simples.

Observações:

- Utilize os métodos de (i) Euler, (ii) Euler-Cromer ou Verlet, e (iii) algum método de ordem igual ou superior a 2.
- As equações de movimento para x e θ podem ser escritas em uma mesma 'roupagem' da seguinte forma

$$\frac{dV}{dt} = -\alpha f(X), \quad \frac{dX}{dt} = V, \quad \mathcal{E} = \frac{V^2}{2} + \alpha g(X)$$

Para o oscilador harmônico temos $X \equiv x$, $V \equiv v$, $\alpha = \omega_0^2$, $\mathcal{E} = E/m$, $f(X) = X$ e $g(X) = X^2/2$, enquanto que $X \equiv \theta$, $V \equiv \dot{\theta}$, $\alpha = g/l$, $\mathcal{E} = E/ml^2$, $f(X) = \operatorname{sen} X$ e $g(X) = 1 - \cos X$ para o pêndulo simples. Logo, se pode usar a mesma rotina de integração da EDO escolhendo-se o parâmetro α e as funções apropriadas $f(X)$ e $g(X)$.

Atividade 1

3. Construa gráficos da posição, x, em função de t (para OHS) e de θ em função de t (para o pêndulo simples).
4. Para força motriz nula e ausência de termos disipativos, obtenha as curvas de energia constante no espaço de fase para o OHS e para o pêndulo simples. Verifique que as curvas obtidas numericamente são iguais (do ponto de vista numérico) às curvas teóricas.

Observações:

- Quando for possível, compare seus resultados com os valores analíticos, i.e., calcule o módulo da diferença entre os valores numéricos e exatos. Verifique se a energia mecânica do sistema se mantêm constante no tempo (podem existir flutuações em torno de um valor médio constante). Esboce gráficos exemplicando seus resultados.

Oscilações forçadas

- Incluem-se força motriz externa $\mathbf{F}_{\text{ext}}(t)$ que oscila com frequência ω e termo dissipativo $\sim \mathbf{v}$ nas equações que descrevem o movimento do sistema. O sistema deixa de ser conservativo.

Oscilações forçadas

- Incluem-se força motriz externa $\mathbf{F}_{\rm ext}(t)$ que oscila com frequência ω e termo dissipativo $\sim \mathbf{v}$ nas equações que descrevem o movimento do sistema. O sistema deixa de ser conservativo.
- Nas equações de movimento para x e θ escritas genericamente em termos de V e X passam a ser

$$\frac{dV}{dt} = -\alpha f(X) - \gamma V + F_{\rm ext}(t), \qquad \frac{dX}{dt} = V,$$

onde $f(X)$ é a mesma função utilizada anteriormente e γ é uma constante.

Atividade 2

1. Escolha ou o OHS ou o pêndulo simples e inclua o termo dissipativo, assim como a força motriz (fique à vontade para definir essa força motriz, desde que tenha uma frequência ω_d).
2. Faça gráficos da amplitude da oscilação do sistema em função de t para diversos valores de ω_d.
3. Faça um gráfico da amplitude máxima de oscilação em função de ω_d. O que você observa para ω_d próxima à frequência natural do sistema. Utilize γ pequeno, mas diferente de zero.

Observações:

- Quando for possível, compare seus resultados com os valores analíticos, i.e., calcule o módulo da diferença entre os valores numéricos e exatos. Verifique se a energia mecânica do sistema se mantêm constante no tempo (podem existir flutuações em torno de um valor médio constante). Esboce gráficos exemplicando seus resultados.

Diagramas de fase

Objetivos:

- Estudar o diagrama de fase em sistemas oscilatórios.

Diagramas de fase

Objetivos:

- Estudar o diagrama de fase em sistemas oscilatórios.
- Elaborar programas computacionais que utilizem *arrays*.

Diagramas de fase

Objetivos:

- Estudar o diagrama de fase em sistemas oscilatórios.
- Elaborar programas computacionais que utilizem *arrays*.
- $*$ Tente fazer animações da evolução do área do espaço de fase no tempo. Em C/C++ pode-se utilizar a biblioteca `OpenGL` para observação da execução do programa em tempo real, assim como há bibliotecas em python para isso.

  Você pode encontrar tutoriais sobre `OpenGL` na internet, e.g., `http://www.glprogramming.com/red/`, `http://openme.gl/opengl-4-tutorial-code/`, ou `http://nehe.gamedev.net/`. Há ainda o *OpenGL RedBook* que pode ser encontrado em uma busca na internet.

UFPE

DF-UFPE

Diagramas de fase

Como construir o diagrama de fase?

- O diagrama de fase é um gráfico constituído por pontos referentes a grandezas que descrevem o estado de um sistema, por exemplo, coordenadas generalizadas $\{q_i\}$ e momentos conjugados $\{p_i\}$.

Diagramas de fase

Como construir o diagrama de fase?

- O diagrama de fase é um gráfico constituído por pontos referentes a grandezas que descrevem o estado de um sistema, por exemplo, coordenadas generalizadas $\{q_i\}$ e momentos conjugados $\{p_i\}$.
- Escolhe-se um conjunto de pontos que descrevem estados iniciais do sistema. Para isto, construa um *array* no seu programa.

Diagramas de fase

Como construir o diagrama de fase?

- O diagrama de fase é um gráfico constituído por pontos referentes a grandezas que descrevem o estado de um sistema, por exemplo, coordenadas generalizadas $\{q_i\}$ e momentos conjugados $\{p_i\}$.
- Escolhe-se um conjunto de pontos que descrevem estados iniciais do sistema. Para isto, construa um *array* no seu programa.
- Utilize um método computacional para integrar no tempo cada uma das condições iniciais dadas. Observe o resultado. Utilize métodos simpléticos e não-simpléticos.

Diagramas de fase

Por que utilizar *arrays*:

- A utilização de *arrays* permite descrever a evolução no tempo de diversos estados do sistema em uma mesma execução do programa.

Diagramas de fase

Por que utilizar *arrays*:

- A utilização de *arrays* permite descrever a evolução no tempo de diversos estados do sistema em uma mesma execução do programa.
- Ao invés de uma variável para cada grandeza (i.e., posição, velocidade, etc) associada a uma partícula, utilizam-se *arrays*. Cada elemento de um array descreve a respectiva grandeza para uma determinada partícula.

Diagramas de fase

Por que utilizar *arrays*:

- A utilização de *arrays* permite descrever a evolução no tempo de diversos estados do sistema em uma mesma execução do programa.
- Ao invés de uma variável para cada grandeza (i.e., posição, velocidade, etc) associada a uma partícula, utilizam-se *arrays*. Cada elemento de um array descreve a respectiva grandeza para uma determinada partícula.
- Em `C/C++` *arrays* são *ponteiros*, i.e., o início do *array* é um endereço na memória e os elementos são espaços contíguos (na maioria das vezes) à este endereço, separados pelo tamanho da variável escolhida (i.e., se for `int` ou `float` o espaço requerido para armazenar cada elemento é diferente e depende do tamanho da variável).

Diagramas de fase

Por que utilizar *arrays*:

- A utilização de *arrays* permite descrever a evolução no tempo de diversos estados do sistema em uma mesma execução do programa.
- Ao invés de uma variável para cada grandeza (i.e., posição, velocidade, etc) associada a uma partícula, utilizam-se *arrays*. Cada elemento de um array descreve a respectiva grandeza para uma determinada partícula.
- Em `C/C++` *arrays* são *ponteiros*, i.e., o início do *array* é um endereço na memória e os elementos são espaços contíguos (na maioria das vezes) à este endereço, separados pelo tamanho da variável escolhida (i.e., se for `int` ou `float` o espaço requerido para armazenar cada elemento é diferente e depende do tamanho da variável).
- Ainda, em `C/C++` *arrays* podem ser alocados estaticamente ou dinamicamente.

Diagramas de fase

* Visualização em tempo real

  - A visualização em tempo real permite diagnosticar alguns problemas na execução de programas rapidamente.

Diagramas de fase

∗Visualização em tempo real

- A visualização em tempo real permite diagnosticar alguns problemas na execução de programas rapidamente.
- Uma biblioteca que permite a visualização em tempo real é a biblioteca `OpenGL`. Esta dispõe de recursos para desenhar pontos, retas, curvas, áreas, etc, em uma janela.

Diagramas de fase

∗Visualização em tempo real

- A visualização em tempo real permite diagnosticar alguns problemas na execução de programas rapidamente.
- Uma biblioteca que permite a visualização em tempo real é a biblioteca `OpenGL`. Esta dispõe de recursos para desenhar pontos, retas, curvas, áreas, etc, em uma janela.
- É necessária outra biblioteca para gerenciar uma janela (i.e., requisitar ao sistema operacional uma janela, colocá-la na tela, (re)dimensioná-la, receber comandos do `OpenGL`, etc. Alternativas simples são as bibliotecas `glut` (`https://www.opengl.org/resources/libraries/glut/`) e `glfw` (`http://www.glfw.org/`).

Atividade 2

Elabore um arquivo `pdf` respondendo às questões seguintes:

1. Considere uma área no espaço de fase como um um conjunto de pontos iniciais $(\{q_i(0),\, p_i(0)\})$, onde $i = 1,\, 2 \cdots N$, referentes a N osciladores ou pêndulos independentes (você pode iniciar com um retângulo, por exemplo).
2. Para o OHS descreva a evolução no tempo da área inicial no diagrama de fase. Utilize métodos diferentes e compare os resultados daqueles de mesma ordem (tomando um mesmo valor para Δt) simpléticos e não simpléticos. A figura inicial muda de forma? A área da figura se modifica com o tempo?
3. O mesmo que no item anterior, mas considerando o pêndulo simples. Que diferenças você observa?

Observações:

- Considere os N osciladores ou pêndulos sendo representados por um *array*. Como são independentes a aceleração de cada elemento do array depende somente da posição e velocidade deste elemento.
- Escolha alguns *frames* da sua visualização com `OpenGL` para colocar no arquivo `pdf` (pode-se fazer um *printscreen* por exemplo).

UFPE

DF-UFPE