

Métodos Computacionais da Física B

Aluno: Vítor Güz - Matrícula: 324610
IF-UFRGS

8 de agosto de 2022

Resumo

Nesta tarefa testaram-se os desempenhos dos métodos de integração numérica discutidos em aulas para um problema de mecânica onde uma partícula se movimenta com conservação de energia no intervalo de $\Delta t's = [0, 10]$, sendo estes métodos: Velocity Verlet, RK2, RK4. O desempenho é verificado através do erro médio das medidas estimadas de energia quando comparado com a energia inicial para vários $\Delta t's$. A conclusão é de que, como esperado, a exatidão é inversamente proporcional aos intervalos de tempo.

1 Introdução

Os algoritmos de integração numérica de EDOs tem erros de dois tipos: truncamento e arredondamento. Uma forma de entender o erro de truncamento é pela representação truncada da expansão de Taylor na extrapolação de um ponto t para $t + \Delta t$. Se usássemos uma série de Taylor infinita (todas as derivadas) não haveria erro. Mas isto não é possível pois conhecemos apenas alguma derivada da função incógnita a través da EDO. Por isso fazemos Δt o menor possível para diminuir esse erro. Um preço a pagar por isso é o aumento do tempo de cômputo; quanto menor o Δt maior o tempo de CPU. Por outro lado, o Δt não pode ser diminuindo infinitamente (como em cálculo, onde o erro de truncamento seria nulo), pois os números no computador devem ser finitos. Mas se for muito pequeno surge o erro de arredondamento. Uma forma simples de entender esse erro é na soma $1 + 10^{-n}$. O computador (na real o python ou compilador usado) reserva um número determinado de casas para representação dos reais (float). Se o número a ser somado a 1 tiver sua primeira casa diferente de zero em uma posição além dessa posição ocorre o erro de arredondamento. Em poucas palavras a soma desse número não tem efeito.

2 Método

As condições iniciais do sistema são: $m = 1; x(0) = -1; v(0) = 1$ Para cada t a dinâmica de partícula sera integrada com cada algoritmo e o erro médio ao longo da trajetória sera calculado como:

$$erro = \sqrt{\frac{1}{N} \sum_0^N (E(t_i) - E(0))^2} \quad (1)$$

onde a energia pode ser dada por:

$$E = \frac{1}{2}mv(t)^2 + (x(t)^4 - 2x(t)^2) \quad (2)$$

Note que tendo a Energia potencial podemos obter a aceleração:

$$\vec{F} = -\vec{\nabla}V(x) \quad (3)$$

$$F = -4x(t)^3 + 4x(t) \rightarrow a = \frac{F}{m} = F \quad (4)$$

O intervalo Δt começa em 0.1 e cai pela metade a cada iteração.

2.1 Método de Verlet

O Método de Verlet é um método numérico usado para estudar equações do movimento de Newton. É frequentemente usado no cálculo da trajetórias de partículas dinâmicas em simulações computacionais. [1], O algoritmo de Verlet reduz o nível de erros no cálculo da próxima posição de um corpo, a partir da posição anterior, sem uso da velocidade. Começando com a fórmula de Newton temos:

$$\ddot{x}(t) \approx \frac{x(t + \Delta t) + x(t - \Delta t) - 2x(t)}{\Delta t^2} = a(x, t) \quad (5)$$

A Eq. 5 acima corresponde a implementação da derivada segunda na equação de Newton.

Podemos escrever então:

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(x, t)\Delta t^2 \quad (6)$$

E a velocidade pode ser dada por:

$$v(t) = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t} \quad (7)$$

2.2 Método Runge-Kutta 2^a ordem

De maneira genérica uma EDO pode ser escrita como[2]:

$$\frac{df}{dt} = g(f(t), t) \quad (8)$$

$$f(t + \Delta t/2) = f(t) + g(f(t), t)\Delta t/2 \quad (9)$$

$$f(t + \Delta t) = f(t) + g(f(t + \Delta t/2), t + \Delta t/2)\Delta t \quad (10)$$

As equações de Newtown podem ser então descritas como:

$$x(t + \Delta t/2) = x(t) + v(t)\Delta t/2 \quad (11)$$

$$v(t + \Delta t/2) = v(t) + a(x(t))\Delta t/2 \quad (12)$$

$$x(t + \Delta t) = x(t) + v(t + \Delta t/2)\Delta t \quad (13)$$

$$v(t + \Delta t) = v(t) + a(x(t + \Delta t/2))\Delta t \quad (14)$$

2.3 Método de Runge-Kutta 4^a ordem

Usando o mesmo procedimento usado para calcular os Métodos de Runge-Kutta de Segunda Ordem, podemos definir os parâmetros c1, c2, c3, c4, a2, a3, a4, b21, b31, b32, b41, b42 e b43 do Método Runge-Kutta de 4-estágios para definir um Método de Runge-Kutta de Quarta Ordem. A demonstração pode ser encontrada nos slides de 14 á 17 da documentação base[2].

3 Resultados

3.1 8 intervalos Δt

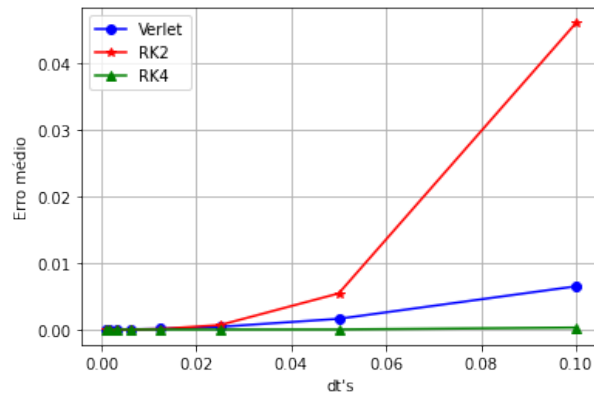


Figura 1: Normal Plot do erro medio variando com $8dt's$ diferentes

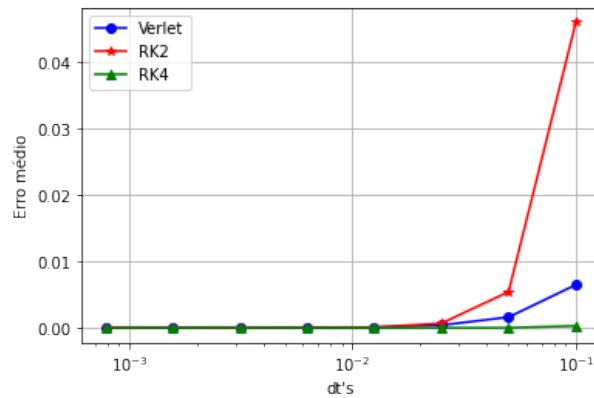


Figura 2: Semi Log X Plot do erro medio variando com $8dt's$ diferentes

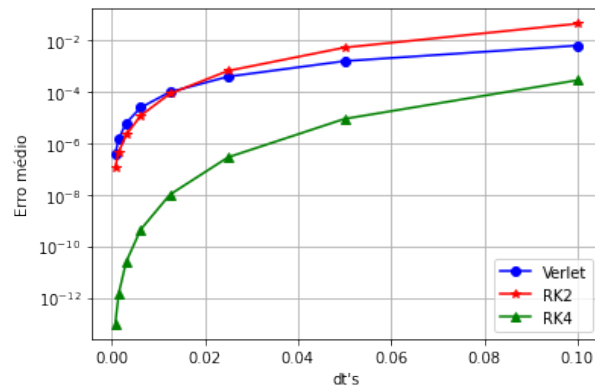


Figura 3: Semi Log Y Plot do erro medio variando com $8dt's$ diferentes

3.2 18 intervalos Δt

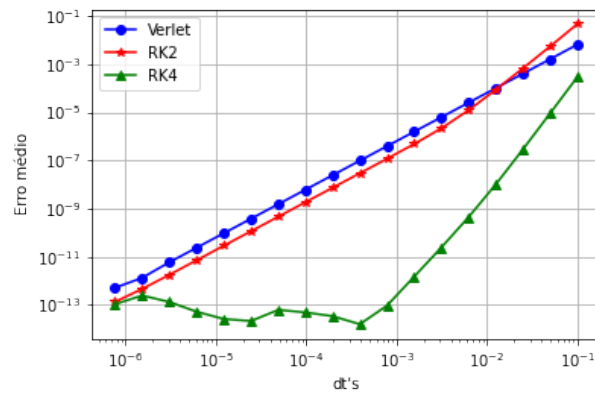


Figura 4: Log Log Plot do erro medio variando com $18dt's$ diferentes

4 Conclusões

De maneira geral, quanto menor for o intervalo de tempo que usamos mais precisa sera nossa estimativa. O método RK4 se demonstrou superior, com menores erros médios para todos os intervalos testados. Apesar de, como podemos ver na figura4, para intervalos muito pequenos os métodos tem erros médios parecidos. Vale ressaltar o quão rápido a precisão do método RK4 melhora em relação aos demais ,quando diminuimos o intervalo. O método RK2 se mostra até menos eficiente que o algoritmo de Verlet para intervalos de tempo consideravelmente grandes (na ordem de $10^{-1}s$), mas ultrapassa Verlet em desempenho conforme o intervalo diminui.

Referências

- [1] davinci.if.ufrgs.br, <https://archive.is/p1eG8>;
- [2] Marina Andretta/Franklina Toledo ICMC-USP 31 de outubro de 2013, <https://sites.icmc.usp.br/andretta/ensino/aulas/sme0300-2-13-fisica/aula20-edorungekutta.pdf>.