Projeto 02: Equações de Onda - I

7600073 - Física Estatística Computacional - 2024/01
06/04/2024

Prof. Dr. Francisco Castilho Alcaraz
Guilherme Santana de Almeida (12694668)

Resumo

Nesta prática Nesta prática montaremos um algorítimo para calcular a solução da equação de onda. Aplicando condições de valores de contorno e propagando uma onda no tempo. Iremos analisar alguns detalhes do algorítimo/aproximação e os interpretar fisicamente.

Introdução

Nesse projeto resolveremos a equação de onda

$$\frac{\partial^2 Y}{\partial t^2} = c^2 \frac{\partial^2 Y}{\partial x^2} \tag{1}$$

sendo $Y(t, x)$ e c a função e velocidade da onda, respectivamente. Para a discretização, usaremos: $x = i\Delta x$ e $t = j\Delta t$ com $i, j = 0, 1, 2, ...$ E para discretizar a equação, realizamos o método de Verlet, ou seja, expandimos $Y(t, x)$ em $t \pm \Delta t$ e $x \pm \Delta x$:

$$Y(t \pm \Delta t, x) = Y(t, x) \pm \frac{\partial Y(t, x)}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 Y(t, x)}{\partial t^2}(\Delta t)^2 \pm ... \tag{2}$$

somamos ambas expressões e fazendo o mesmo para x

$$\frac{Y(t + \Delta t, x) + Y(t - \Delta t, x) - 2Y(t, x)}{(\Delta t)^2} = c^2 \frac{Y(t, x + \Delta t) + Y(t, x - \Delta x) - 2Y(t, x)}{(\Delta x)^2} \tag{3}$$

Como queremos propagar no tempo, isolamos $Y(t + \Delta t, x)$ e tomamos a discretização dada acima. Mas antes, definimos $r = c\Delta t/\Delta x$. Logo, o resultado final é

$$Y(j + 1, i) = 2(1 - r)^2 Y(j, i) + r^2[Y(j, i + 1) - Y(j, i - 1)] - Y(j - 1, i) \tag{4}$$

Tarefa 1

ENUNCIADO: Faça um programa que calcule as ondas perfeitas de velocidade c em um meio não dissipativo ou dispersivo de comprimento L. Especialize seu programa para a situação em que $Y(x,0) = Y_0(x)$ é dada e $\frac{d}{dt}Y(x,t)|_{t=0}$ (onda parte do repouso). Considere fronteiras fixas. Desta forma os parâmetros do programa serão L, c, Δx, r e Y_0 (que pode ser fornecida numa subrotina com a mesma discretização). Para testes escolha $L = 1m$ e $c = 300\,m/s$. Considere um pacote Gaussiano inicial

$$Y(x,0) = Y_0(x) = exp[-(x-x_0)^2/\sigma^2] \tag{5}$$

com $x_0 = L/3$ e $\sigma = L/30$.

(a) $r = 1$

Para esse item, utilizei $\Delta x = 0.01\,m$. Logo, cada iteração corresponde à $\Delta t = r\Delta x/c = 3.3 \times 10^{-5}\,s$, um valor muito pequeno. O valor de Δx escolhido é um valor padrão para discretizações em espaços deste tipo e geralmente é uma boa estimativa inicial do valor ideal. Aqui, iremos com a ideologia de que o valor ideal é aquele que funciona, dado esse fato verdadeiro, um valor menor se torna desnecessário, mas iremos falar um pouco da implicação de escolher um valor menor mais pra frente. E como demonstrado abaixo, neste item e nos demais, $\Delta x = 0.01\,m$ é de fato um valor ótimo e que nos permite visualizar bem os fenômenos de interesse.

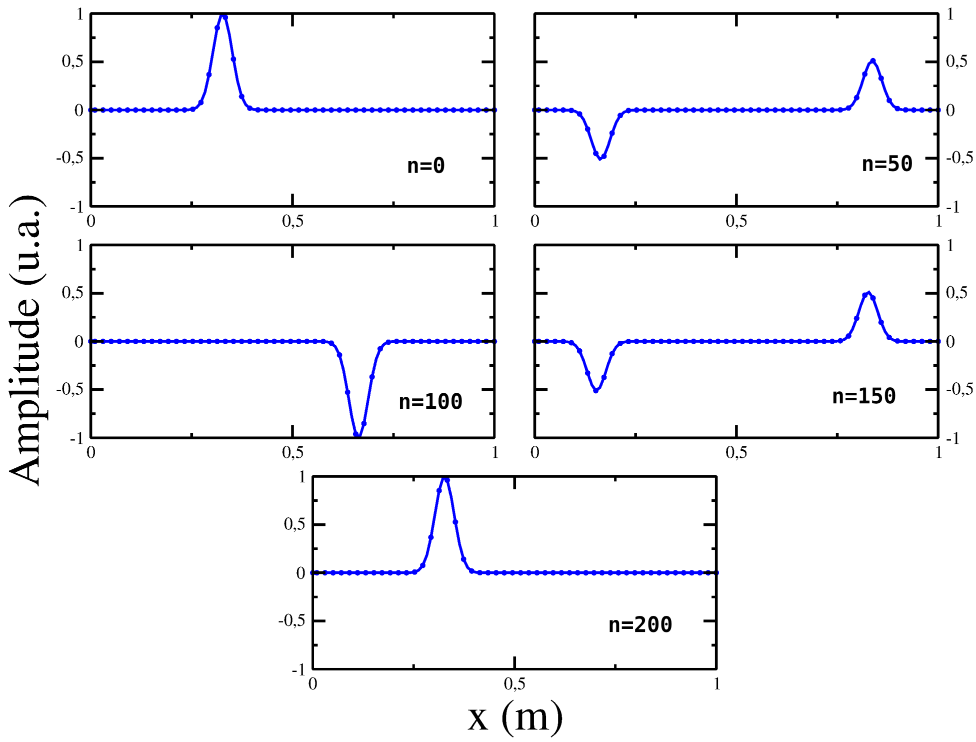

Figura 1: Propagação de uma onda no tempo com fator $r = 1$.

Como a velocidade inicial é nula, o pacote se divide igualmente em dois e mantém essa forma. Cada um dos dois pacotes resultantes são ondas semelhantes, de mesma velocidade, e que correspondem a uma função do tipo $f(x-ct)$ e $g(x+ct)$. Ou seja, ondas viajando em direções contrárias. E a reflexão que vemos nas extremidades é condizente com a solução para o cenário com extremidades fixas. O período, ou, em outras palavras, o tempo em que ela demora para voltar a sua condição incial é $T = \Delta t n$, em que $n = \frac{2L}{r\Delta x} = 200$ é o número de iterações. Logo, $T = 6.6 \times 10^{-3}\,s$. E na verdade, esse tempo é igual para todos os outros itens, $r = 2$ e $r = 0.25$, afinal ele se reduz à $\frac{2L}{c}$, oque muda é o número de iterações, já que junto com o r, o Δt também muda.

Ademais, podemos observar que não, o pacote não se deforma. Para esse algorítimo, estamos propagando a onda utilizando a expressão 4, cujo nossa onda tem velocidade c. Mas a nossa própria discretização possui um tipo de velocidade também, $\Delta x/\Delta t$, na qual e completamente independente da velocidade de onda c. No caso $r = 1$, não há nenhum tipo de incoerência entre nosso algorítimo e nosso objeto físico chamado onda, então podemos simular esse último com boa precisão. Para os próximos dois casos isso não ocorre.

(b) $r = 2$

Mantendo $\Delta x = 0.01\,m$, nossa iterações andarão $\Delta t = 2\Delta x/c = 6.66 \times 10^{-5}\,s$. Nesse caso a velocidade de deslocamento de um comprimento do nosso espaço $\frac{\Delta x}{\Delta t} = \frac{c}{2}$ é menor que a velocidade de propagação da própria onda. Ou seja, nossa onda se propaga mais rápido do que nosso algorítimo é capaz de calcular, isso faz nossa aproximação divergir violentamente. Outra forma de ver esse efeito, agora mais matematicamente, é perceber que os dois termos dependentes de r na equação 4 são > 1, então os valores de amplitude crescem exponencialmente ao contrário do item (a), onde esses termos eram < 1.

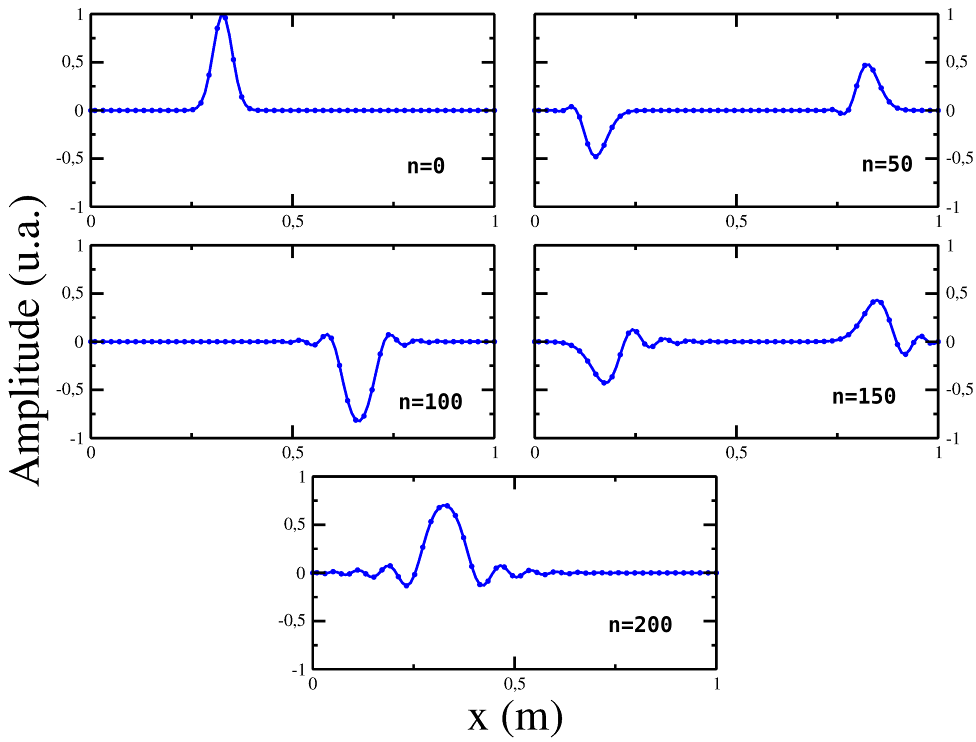

Figura 2: Propagação de uma onda no tempo com fator $r = 0.25$.

(c) $r = 0.25$

Neste último caso, temos o contrário do item anterior. Mantendo, mais uma vez, $\Delta x = 0.01\,m$, nosso intervalo de iteração é agora $\Delta t = \frac{\Delta x}{4c} = 8.3 \times 10^{-6}\,s$. E o número de iterações para retornar à configuração anterior é $n = 800$. Nosso $\frac{\Delta x}{\Delta t} = 4c$ é 4 vezes mais rápido que a velocidade da onda. Ou seja, nosso algorítimo está tentando ir mais rápido que o necessário e isso gera a deformação observada. Uma forma de pensar esse fenômeno é lembrando o que acontece quando um jato supersônico ultrapassa a velocidade do som. Se no nosso meio discreto uma onda não pode ultrapassar a velocidade c e quisermos propagá-la com $\frac{\Delta x}{\Delta t} > c$, então o meio tirará energia dessa onda supersônica. Podemos ver isso através do rastro que ela deixa por onde passa, e vendo que sua amplitude máxima vai diminuindo com o tempo, "perdendo energia". Um menor Δx, faria com que essas deformações demorassem mais para acontecer, afinal para manter $\frac{\Delta x}{\Delta t}$ igual é preciso iterar mais vezes para iterar por um mesmo tempo.

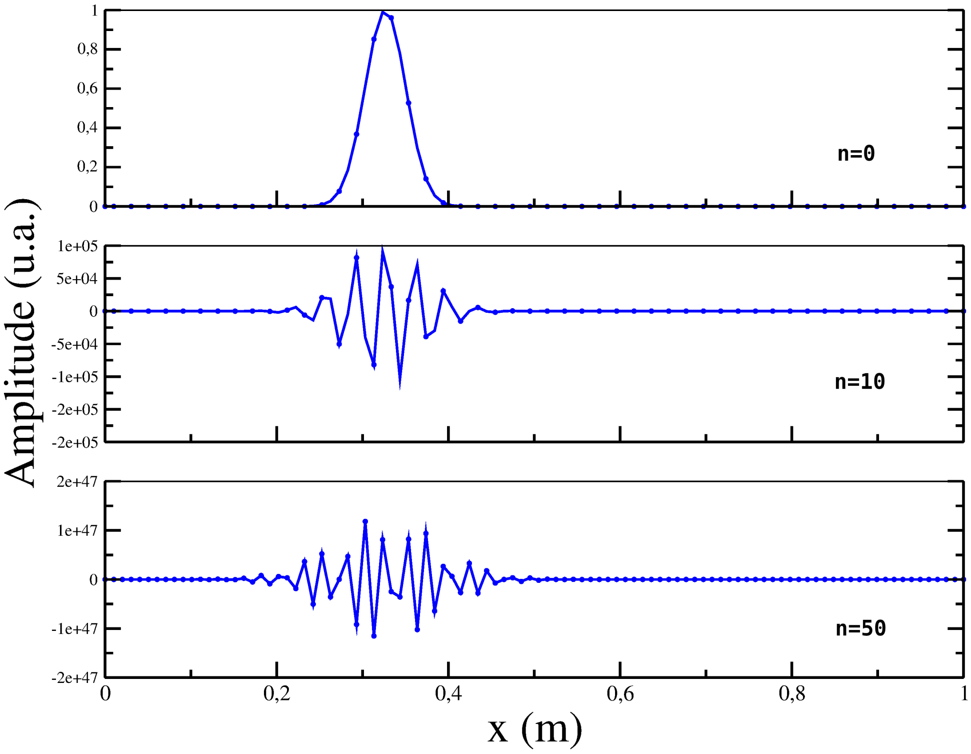

Figura 3: Propagação de uma onda no tempo com fator $r = 2$.

E só para esclarecer, no GIF anexado ela aparenta estar mais lenta que a do item (a), mas isso é porque os dois GIFs foram gerados com o mesmo *framerate*, mas no GIF do item (a) 1 *frame* equivale a 4 desse item. Então se o *framerate* escolhido fosse 4 vezes mais rápido, elas teriam a mesma velocidade no GIF, como deveria ser.

Tarefa 2

ENUNCIADO: Repetir o item (a) da tarefa I, mas agora, utilizando o perfil de onda da imagem taltaltal para $Y_0(x)$.

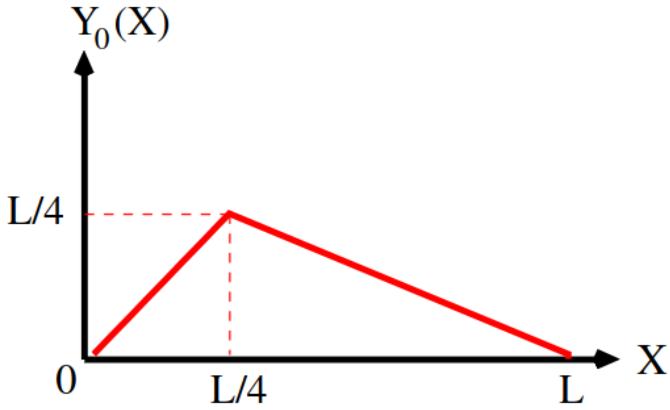

Figura 4: Função representando uma corda num violão.

Ou seja:

$$Y_0(x) = \begin{cases} x & \text{Se } 0 \leq x \leq L/4 \\ L/3(1-x) & \text{Se } L/4 < x \leq L \end{cases} \tag{6}$$

Bom, o comportamento da corda é quase idêntico, senão o mesmo que o item (a) da tarefa anterior. Utilizando $x = 0.01\,m$ e com $\Delta t = r\Delta x/c = 3.3 \times 10^{-5}\,s$, a onda demora também $T = 6.6 \times 10^{-3}\,s$ para voltar a condição inicial e ao longo do percurso ela não se deforma, mantendo a mesma tendência que no item (a). O que parecem interferências nos platôs são só artefatos da discretização, que talvez fique mais aparente por causa da natureza meio rígida da forma dessa onda.

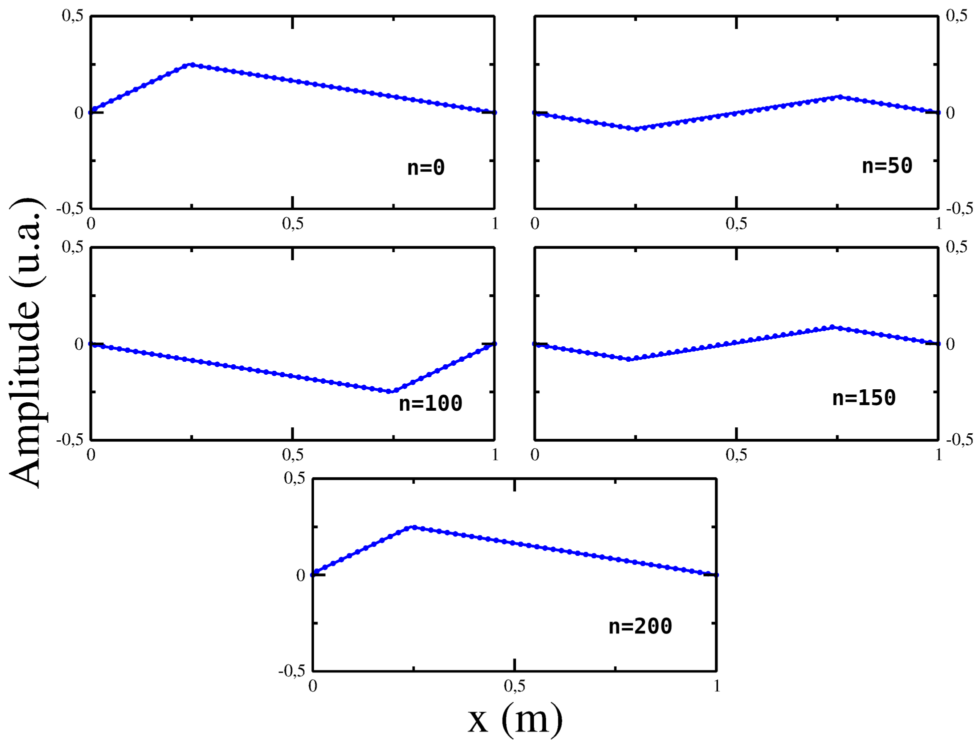

Figura 5: Propagação de uma onda numa corda de violão no tempo com fator $r = 1$.