

1 Discretização da Equação de Onda

Nesse trabalho buscamos estudar a equação do onda em 1D, para isso partimos da equação (1) e fazemos a sua discretização. Escolhemos utilizar a discretização para derivadas simétricas, e portanto obtemos a forma da equação de ondas com a qual trabalharemos (2)

$$\frac{\partial^2 \mathcal{Y}}{\partial t^2} = c^2 \frac{\partial^2 \mathcal{Y}}{\partial x^2} \quad (1)$$

Partindo das derivadas segundas, com $x = i\Delta x$ e $t = n\Delta t$ para $i, n = 1, 2, \dots$ temos

$$\frac{\mathcal{Y}(i, n+1) + \mathcal{Y}(i, n-1) - 2\mathcal{Y}(i, n)}{(\Delta t)^2} = c^2 \frac{\mathcal{Y}(i+1, n) + \mathcal{Y}(i-1, n) - 2\mathcal{Y}(i, n)}{(\Delta x)^2}$$

$$\mathcal{Y}(i, n+1) + \mathcal{Y}(i, n-1) - 2\mathcal{Y}(i, n) = \frac{c^2(\Delta t)^2}{(\Delta x)^2} [\mathcal{Y}(i+1, n) + \mathcal{Y}(i-1, n) - 2\mathcal{Y}(i, n)]$$

fazendo $r \equiv c\Delta t/\Delta x$ podemos escrever as iterações no tempo para cada posição da propagação na equação abaixo

$$\mathcal{Y}(i, n+1) = 2(1 - r^2)\mathcal{Y}(i, n) + r^2(\mathcal{Y}(i+1, n) + \mathcal{Y}(i-1, n)) - \mathcal{Y}(i, n-1) \quad (2)$$

Fica claro pela equação (2) que para nossa simulação podemos trabalhar com vetores que descrevem o estado de uma posição em cada posição, “anterior”, “atual” e “posterior”. No código das simulações foi implementado uma matrix “grid” que computa esses estados de acordo com a iteração.

Para finalizar a descrição do problema de propagação de ondas precisamos das condições de contorno. Nesse projeto inicial foram feitas simulações em condições para propagação ideal de ondas presas em uma caixa, ou seja, nas extremidades do nosso espaço vale que $\mathcal{Y}(0) = \mathcal{Y}(L) = 0$, para um comprimento L qualquer e $d\mathcal{Y}(x, t = 0)/dt = 0$.

1.1 Estabilidade numérica

A escolha dos menores Δx e Δt em geral levam a execuções mais estáveis. Porém, quando trabalhamos com dois passos diferentes surgem algumas possíveis instabilidades. Buscando simplificar essas análises escrevemos o fator r

$$r = \frac{\Delta t}{\Delta x} c \quad (3)$$

a partir dessa relação podemos tirar algumas conclusões: Δx e Δt devem variar juntos, pois se mantermos um deles fixo e variando o outro obtemos uma divergência, por exemplo se Δx são valores cada vez menores para um Δt fixo e o caso mais estável que podemos obter é no qual $r = 1$.

No projeto iniciamos as simulações no caso mais estável de uma onda ideal com $r = 1$ e depois as situações em que as instabilidades numéricas surgem. Nesses casos foi necessário trabalhar com menos iterações já que os valores de Y divergem muito rapidamente. Então basta algumas iterações para demonstrarmos esta instabilidade.

1.2 Erro associado à simulação

Pela relação (2), temos um erro associado à x que buscamos minimizar. No trabalho estamos usando números de precisão de 12 e portanto, o termo de erro da derivada segunda é a ordem que “escondemos” na equação de ondas discretizada

$$\mathcal{E}(\Delta x) = \Delta x^2 + \frac{10^{-12}}{(\Delta x)^2} \quad (4)$$

o ponto de mínimo desse erro é em torno de $\Delta x = 10^{-3}$, portanto esse é o nosso mínimo.

Partindo desse valor para Δx fixo trabalhamos variando os outros e observando qual a influência deles na instabilidade do algoritmo.

2 Simulações de ondas ideais

Como foi discutido anteriormente (1.2) o Δx escolhido foi fixado para todas simulações. Além disso a velocidade de nossas ondas também é mantida constante igual à $c = 300$ (m/s). Com isso podemos variar os parâmetros de r e Δt .

Foi escolhido um período total¹ de $t = 0,01$ (s) de forma que, para uma onda à velocidade da luz dentro de uma caixa de 1(m) hajam 3 reflexões.

Foram utilizadas as mesmas rotinas para implementação da simulação com uma única mudança, as condições iniciais, já que as condições de contorno, isto é, a corda presa nas extremidades, é válida para simulação das ondas na corda de um violão. Para a simulação de ondas Gaussianas usamos o modelo de condição inicial dado por

$$\mathcal{Y}(x, 0) = \mathcal{Y}_0(x) = \exp\left[-\frac{(x - x_0)^2}{\sigma^2}\right], x_0 = L/3, \sigma = L/30 \quad (5)$$

$d\mathcal{Y}/dt|_{t=0} = 0$ onde no código foi denotado por $Y0(x, s)$.

Segue abaixo a estrutura central do programa de simulação de ondas:

```
1  !      Simulação para ondas ideais
2  implicit real*8(a-h, o-y)
3  dimension grid(1000, 3)
4  ! L
5  s = 1.0d0
6  c = 300.0d0
7  r = 1.0d0
8  nx = 1000
9  dx = s / (nx*1.d0)
10 dt = r * dx / c
11 t = 0.01
12 nt = floor(t/dt)
13 open(unit = 1, file = "saida-tarefa1-a.dat")
14 grid(:, 3) = 0.e0
15 !      aplica as condicoes iniciais ao grid
16 !      t = 0
17 do i = 1, nx
18     grid(i, 2) = Y0(i*dx, s)
19 end do
20 !
21 grid(:,1) = grid(:, 2)
22 call write_to_file(grid, nx)
23 !      t = 1
24 grid(:, 2) = grid(:, 1)
25 !      simulação
26 do n = 3, nt
27     call drive_pulse(grid, nx, r)
28     call write_to_file(grid, nx)
29 end do
30 close(1)
31 end
32
33 subroutine write_to_file(grid, nx)
34 implicit real*8(a-h, o-y)
35 dimension grid(1000, 3)
36 write(1, '(3000F16.8)') (grid(i, 2), i=1, nx)
37 end subroutine write_to_file
38
39 subroutine drive_pulse(grid, nx, r)
40 implicit real*8(a-h, o-y)
41 dimension grid(1000, 3)
```

¹período?

```

42  !      y_next = 2(1-r^2)y_curr + r^2[y(t+1,n)+y(t-1,n)] - y_prev
43      grid(1, 3) = grid(1, 2)
44      grid(nx,3) = grid(nx, 2)
45
46      y = 2.e0*(1.e0-r*r)
47
48      do i = 2, nx-1
49          grid(i,3)=y*grid(i,2)+r*r*(grid(i+1,2)+grid(i-1,2))-grid(i,1)
50      end do
51  !      swap
52      grid(:, 1) = grid(:, 2)
53      grid(:, 2) = grid(:, 3)
54      end subroutine drive_pulse

```

Para o programa as simulações de ondas em uma corda temos a condição inicial dada por (5), implementada por

```

1      function Y0(x, s)
2      implicit real*8(a-h, o-y)
3      Y0 = exp(-((x-s/3)**2)/(s/30)**2)
4      end

```

condição aplicada à todo o grid na linhas linhas 17 à 24 do programa.

2.1 Caso de $r = 1$

Como podemos ver na figura (1) a oscilação não apresenta nenhuma deformação, e fazendo valores maiores de t e observando o comportamento da onda não surgem deformações para $r = 1$. Para condição de contorno de bordas fixas é esperado que a onda seja espelhada nas reflexões, e é o que observamos para os dois pacotes à esquerda e à direita. A primeira reflexão acontece em torno da iteração 30 e a segunda na iteração 70.

Ocorrem interferências de forma construtiva, pois pelas condições impostas ambos se encontram sempre do mesmo lado da reflexão e se somam formando a amplitude completa inicial. A configuração inicial do sistema vai se repetir na iteração número 200, isto é, no tempo $200 \cdot \Delta t$. Como nosso Δt utilizado é dado pela (3), então $\Delta t = 1 \cdot \Delta x / 300 \approx 3 \cdot 10^{-5}$ será a configuração inicial se repete em $t = 0,06$ (s). Esse é o tempo necessário para cada um dos pacotes menores percorrerem $2L$. Para um L qualquer, temos então $t = 2L/c$ será o tempo em que a situação inicial surge novamente.

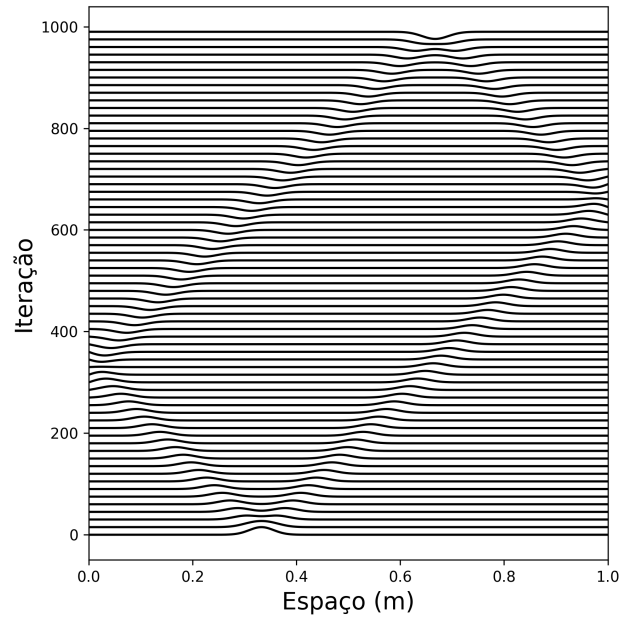


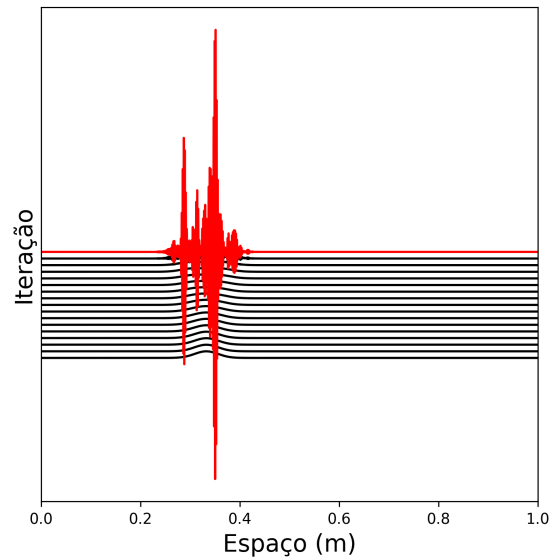
Figura 1: Evolução temporal da onda em 1000 iterações. Apenas metade dessas iterações estão nessa figura para melhor visualização das frentes de ondas.

2.2 Caso de $r = 2$

Nesse caso podemos observar a instabilidade desse que surge para valores de $r > 1$. A amplitude das ondas diverge muito rápido e na simulação bastaram pouquíssimas iterações para que visualizarmos o comportamento anômalo da onda.

O algoritmo diverge para $r > 1$ pois, como foi discutido (1.1), a melhor estabilidade que podemos ter é no caso em que $r = 1$. Podemos pensar em $r = 2$ como uma diminuição do tamanho do passo Δx e como isso ocorre para um Δt fixo (já que r mudou) as perturbações que percorrem esse espaço à cada passo tem velocidade c em $r = 1$ passam a ter velocidade menor que c e portanto cada iteração vai acumular valores cada vez maiores vindos dessas perturbações que se somam, fazendo a amplitude divergir rapidamente.

Figura 2: Divergência da equação da solução para $r = 2$.

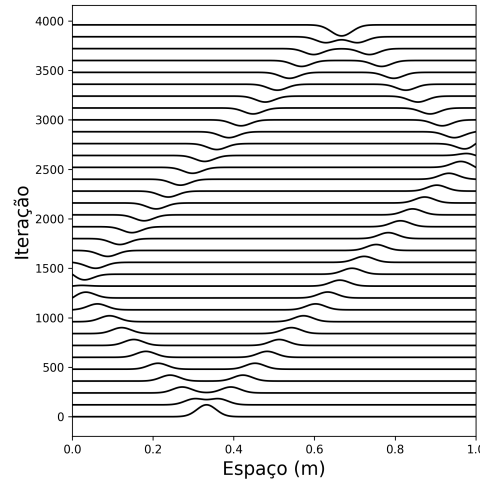


2.3 Caso de $r = 0.25$

A implementação com valores de $r < 1$ apresentam certa estabilidade, como pode ser observado na figura (3), porém para tempos grandes surgem deformações no pacote gaussiano. E além disso, para tempos pequenos apesar de simular bem, não é eficiente computacionalmente, já que executa muito mais iterações que no caso de $r = 1^2$. Isso porque, para Δx , c e r fixos, precisamos alterar o parâmetro de Δt , o que aumenta o número de iterações do nosso programa.

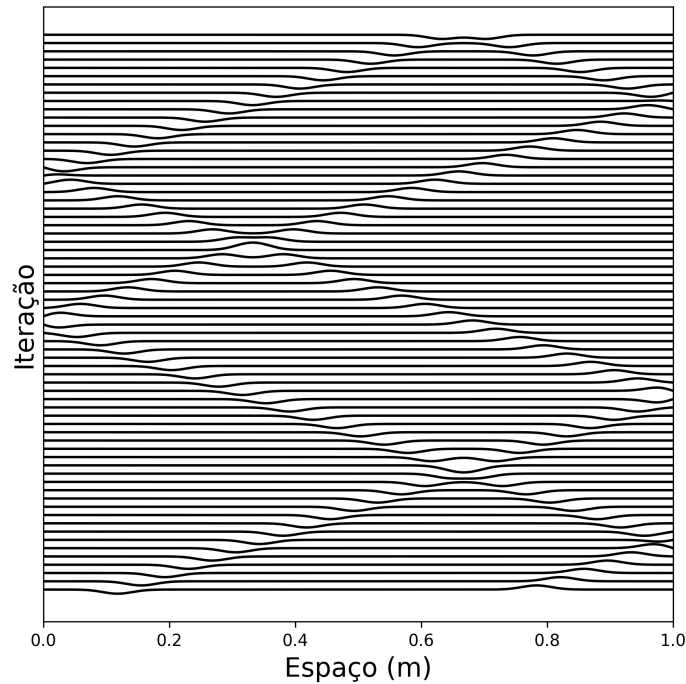
Para conseguir observar o surgimento das deformações foi necessário diminuir a precisão da grade para 10^{-2} , pois utilizando o Δx de ordem 10^{-3} não consegui observar as instabilidades. Foi testado para diversos tempos e não exibiu em nenhum momento deformações como as da (4). Como podemos ver, a escolha de Δx impactou bastante o desempenho dessa simulação.

Figura 3: Simulação para $r = 0,25$ nas primeiras iterações.



A figura (4) foi construída a partir da simulação com $r = 0.25$ e $t = 0,02$ e com $\Delta x = 10^{-2}$. No gráfico temos o comportamento da onda para um tempo grande, após muita iterações e podemos ver claramente as deformações do pacote gaussiano.

Figura 4: Comportamento da onda para um tempo grande.



²Nota-se, que assim como foi dito, o número de iterações aumenta. Para $r = 0.25$, ou seja $1/4$ do r anterior, o número de iterações quadruplica.

3 Simulação de ondas na corda de violão

Já o perfil inicial de pinça, na corda do violão, foi chamado de $Y0_pinca(x, s)$, e pela figura dada no projeto temos que o perfil inicial da onda deverá ser

$$\mathcal{Y}(x, 0) = \mathcal{Y}_0(x) = \begin{cases} x, & x < L/4 \\ \frac{1}{3}(L - x), & L/4 \leq x \leq L \end{cases} \quad (6)$$

O código da implementação dessa simulação é diferente em relação ao primeiro apenas nas condições iniciais. Portanto apenas compilarei no relatório esse trecho diferente dos programas:

```
1 function Y0_pinca(x, s)
2 implicit real*8(a-h, o-y)
3 if(x .le. s/4) then
4     Y0_pinca = x
5 else
6     Y0_pinca = (1.0e0/3.0e0)*(s - x)
7 end if
8 end
```

3.1 Caso de $r = 1$

Nesse caso o pacote não se deforma e como podemos ver, os pacotes menores sofrem reflexões nas próximo das iterações em torno de 250 e 780, comportamento igual ao do primeiro caso observado anteriormente. E assim como nele, também vai haver repetição da configuração inicial em um tempo igual à $t = 2L/c$, ou seja, ambos pacotes menores devem andar $2L$ antes de haver a repetição da situação inicial. E por fim, as interferências também são como antes: construtivas.

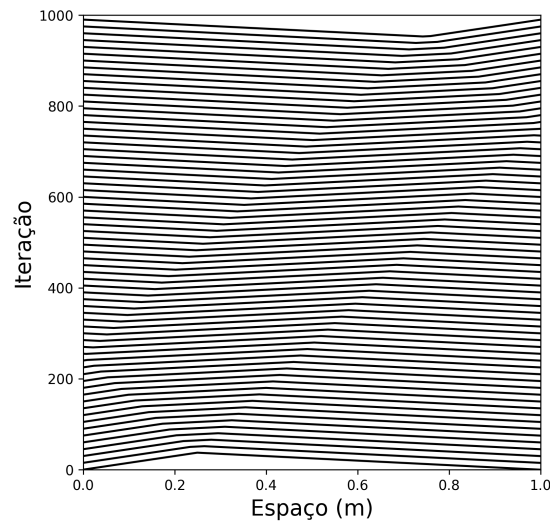


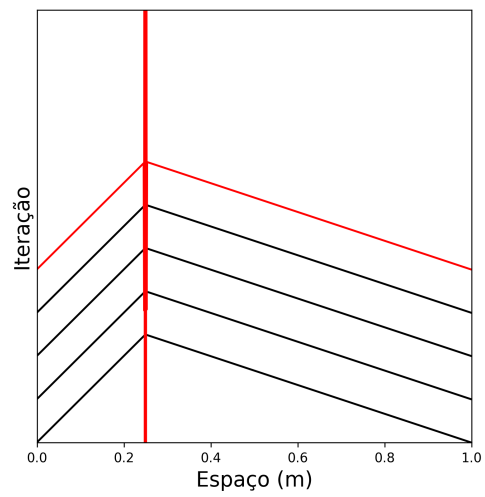
Figura 5:

3.2 Caso de $r = 2$

Esse caso é similar ao da tarefa anterior (2.2). Nesse caso temos a especificidade da divergência acontecer muito mais rápido que no anterior. Suponho que isso esteja relacionado ao perfil inicial da onda, já que nesse perfil temos muitos pontos diferentes de zero, ao contrário do perfil dos pacotes gaussianos. Esses valores se acumulam mais rapidamente que na (2.2). Além disso a divergência parece acontecer somente no ponto $x = L/3$.

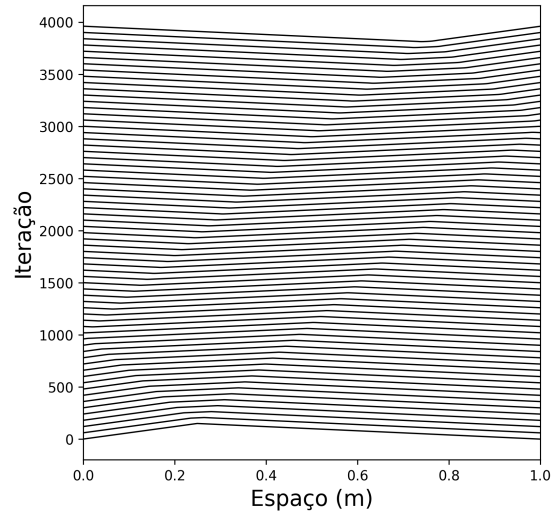
O motivo disso pode ser justamente as condições iniciais, já que na gaussiana boa parte dos valores iniciais da grade é nulo, e na condição de pinça (6) temos valores maiores que contribuem para as próximas iterações que como vimos são ampliadas muito rapidamente.

Figura 6: Simulação das ondas divergindo para $r = 2$ em poucas iterações.



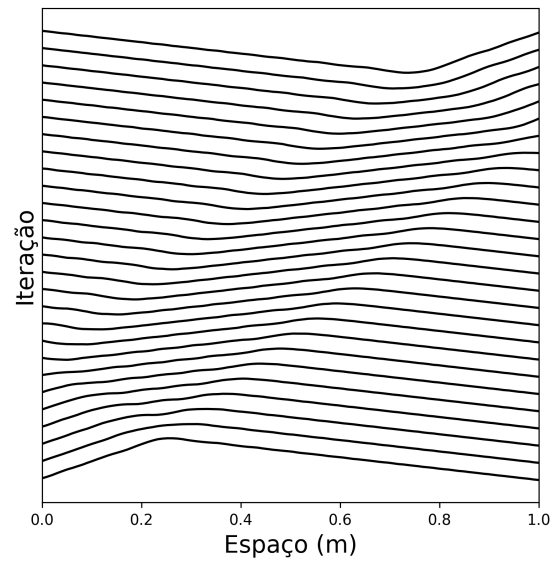
3.3 Caso de $r = 0.25$

Figura 7: Simulação das ondas no violão para $r = 0,25$.



Assim como na tarefa anterior também foi simulado o sistema para $\Delta x = 10^{-2}$ para tentar observar as instabilidades. Para $\Delta x = 10^{-3}$ não foi possível ver deformações para todos tempos testados.

Figura 8: Gráfico simplificado das deformações de ondas para condição inicial (6).



Apesar de difícil de perceber, podemos notar as deformações na propagação, sobretudo nas regiões de $x \approx 0,3$ e $x \approx 0,7$ que são pontos de interferência.