

1 Introdução

Partimos das relações estudadas anteriormente, sendo elas a transformada de Fourier discreta (DFT)

$$\mathcal{Y}_k = \sum_{\ell=0}^{N/2-1} y_{\ell} e^{2\pi i(k\ell/N)} \quad (1)$$

onde N é o número de amostras utilizadas e y_{ℓ} são resultados gerados pela solução da equação de onda do projeto anterior, observada em apenas um ponto do espaço, fixo.

$$\mathcal{Y}(i, n+1) = 2(1-r^2)\mathcal{Y}(i, n) + r^2(\mathcal{Y}(i+1, n) + \mathcal{Y}(i-1, n)) - \mathcal{Y}(i, n-1) \quad (3)$$

Com esses resultados vamos fazer análise espectral de potências das componentes de Fourier de amplitudes de onda para um tempo total observado.

$$P(\mathcal{F}) = \mathbb{R}(\mathcal{F})^2 + \mathbb{I}(\mathcal{F})^2 \quad (4)$$

onde \mathcal{F} é referente à frequência das componentes real e imaginária.

Com isso temos as ferramentas que vão ser utilizadas para estudar oscilações em uma corda com condições de contorno fixas e com uma extremidade solta.

2 Simulação de ondas em cordas

Iniciamos o projeto e portanto a discussão com condição de contorno de bordas presas. Podemos mapear o sistema da corda com bordas presas para uma EDP com condições de contorno $Y(0, t) = 0 = Y(L, t)$. Por algum método como separação de variáveis ou utilizando funções $f(x \pm vt)$ quaisquer¹ podemos estudar o comportamento da onda no espaço. Partindo disso, temos que em x , a onda pode ser descrita por uma EDO da forma $\mathcal{X}(x) = A \cos(kx) + B \sin(kx)$. Pelas condições de contorno da $Y(x, t)$ temos que $\mathcal{X}(0) = 0 \Rightarrow A = 0$, logo nossa solução será apenas em termos de senos. Segue que $\sin(kx) = 0 = \sin(kL) \Rightarrow k_n = \frac{n\pi}{L}$, $n = 1, 2, 3, \dots$ é a solução fundamental de \mathcal{X} , ou seja, descreve o comportamento da onda em relação aos modos normais dela.

Utilizando a relação para os modos normais k_n podemos obter uma relação para os comprimentos de onda para cada um deles, pois $k_n = \frac{2\pi}{\lambda_n}$, então

$$\lambda_n = \frac{2L}{n} \quad (5)$$

Partindo da (5) podemos obter uma expressão para as frequências para os modos, isto é $f_n = c/\lambda_n$ nos leva à

$$f_n = \frac{nc}{2L} \quad (6)$$

Um detalhe importante para as análises que vamos fazer é o impacto das condições iniciais escolhidas. O perfil inicial, isto é, a posição x_0 escolhida para fixar a amplitude gaussiana no início da simulação determina a quantidade de modos fundamentais a oscilação vai ter.

Mais que isso, a posição inicial escolhida pode contribuir para formação dos nós já que de acordo com simetria de onde parte vai gerar interferências destrutivas. Isso também vale para as outras interferências, dependendo da fase de π entre as ondas propagantes à esquerda e direita partindo de x_0 podem haver componentes componentes que ao realizar transformada de Fourier⁽¹⁾ vão possuir contribuir com maiores amplitudes.

Com isso temos como qual deve ser o comportamento esperado da transformada de Fourier para pontos obtidos em uma posição de observação x , isto é, a frequência base será $f_1 = c/2L$ onde $c = 300$ (m/s) e $L = 1$ (m), então $f_1 = 150$ (Hz) escrevemos a frequência fundamental desse caso como

¹Que obedecem as condições de contorno.

$$f_n = n f_1 \quad (7)$$

Tratar o caso de uma borda solta, as relações para comprimento de onda e frequência passam a ser

$$\lambda_n = \frac{2L}{n - \frac{1}{2}} \quad (8)$$

$$f_n = \left(n - \frac{1}{2}\right) \frac{c}{2L} \quad (9)$$

Consequentemente a frequência fundamental f_1 nesse caso passa a ser $f_1 = \frac{c}{4L} = 75$ (Hz) e as próximas são

$$f_n = (2n - 1) f_1 \quad (10)$$

2.1 Implementação

Segue abaixo as rotinas utilizadas para simulações do projeto. Primeiramente temos a rotina `DFT(Y, N, DT)`, relativa ao primeiro projeto e apenas com modificação para armazenar o espectro de potências (4):

```

1      subroutine dft(Y, N, dt)
2      implicit real*8 (a-h,o-y)
3      implicit complex*8(z-z)
4
5      parameter(pi = acos(-1.0e0))
6      dimension Y(4000)
7
8      zi = (0.0, 1.0)
9      open(unit = 2, file = "saidas/dft-item-a.dat")
10     zeta = exp(2.d0*pi*zi/N)
11
12     do k = 0, N/2 - 1
13         zY = 0
14         do m = 2, N
15             zY = zY + Y(m)*zeta**(k*m)
16         end do
17
18         res = real(zY)**2 + aimag(zY)**2
19         ! res = res / (N/2)
20         freq = k/(N*dt)
21         print *, freq, res
22         write(2, '(3000F20.8)') freq, res
23     end do
24     close(2)
25 end subroutine dft

```

Os resultados da execução são armazenados no diretório `saidas/`.

Abaixo está o código das subrotinas referente ao projeto anterior, propagação de ondas em uma corda fixa, livre e uma rotina para escrever evolução temporal do perfil da onda `DRIVE_WAVE_FIXED(GRID, NX, R)`, `DRIVE_WAVE_FREE(GRID, NX, R)`, `SAVE_WAVE_STATE(GRID, NX)`, respectivamente.

```

1      subroutine drive_wave_fixed(grid, nx, r)
2      implicit real*8(a-h, o-y)
3      dimension grid(1000, 3)
4      ! y_next = 2(1-r^2)y_curr + r^2[y(t+1,n)+y(t-1,n)] - y_prev
5      grid(1, 3) = grid(1, 2)
6      grid(nx,3) = grid(nx, 2)
7      y = 2.e0*(1.e0-r*r)
8      do i = 2, nx-1
9          grid(i,3)=y*grid(i,2)+r*r*(grid(i+1,2)+grid(i-1,2))-grid(i,1)
10     end do

```

```

11  grid(:, 1) = grid(:, 2)
12  grid(:, 2) = grid(:, 3)
13  end subroutine drive_wave_fixed
14
15  subroutine drive_wave_free(grid, nx, r)
16  implicit real*8(a-h, o-y)
17  dimension grid(1000, 3)
18  ! y_next = 2(1-r^2)y_curr + r^2[y(t+1,n)+y(t-1,n)] - y_prev
19  grid(1, 3) = grid(1, 2)
20
21  y = 2.e0*(1.e0-r*r)
22  do i = 2, nx-1
23      grid(i,3)=y*grid(i,2)+r*r*(grid(i+1,2)+grid(i-1,2))-grid(i,1)
24  end do
25
26  grid(nx, 3) = grid(nx-1, 3)
27  grid(:, 1) = grid(:, 2)
28  grid(:, 2) = grid(:, 3)
29
30  end subroutine drive_wave_free
31
32
33  subroutine save_wave_state(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 save_wave_state
38

```

Por último temos a função utilizada para escrever os perfis iniciais de onda em forma de pacotes gaussianos dada por GAUSSIAN(X, X0, SIGMA):

```

1  function gaussian(x, x0, sigma)
2  implicit real*8(a-h, o-y)
3  gaussian = exp(-((x-x0)**2)/(sigma)**2)
4  end function gaussian

```

Partindo dessas rotinas e função podemos montar a simulação completa, presente no arquivo tarefa-1-12559016.f

```

1
2  implicit real*8(a-h,o-y)
3  implicit real*8(l-l)
4  implicit complex*8(z-z)
5
6  external gaussian
7
8  dimension grid(1000, 3)
9  dimension Y(4000)
10
11  nx = 1000
12  nt = 4000
13  r = 1.0d0
14  L = 2.0d0
15  c = 300.0d0
16  dx = L / (nx*1.d0)
17  dt = r * dx / c
18
19  print *, "Paramêtros:"
20  print *, "-----"

```

```

21     print *, "r = ", r
22     print *, " nx = ", nx
23     print *, " nt = ", nt
24     print *, " L = ", L
25     print *, " t = ", nt * dt
26     print *, " dx = ", dx
27     print *, " dt = ", dt
28     print *, "-----"
29
30     !   aplica as condicoes iniciais ao grid
31     !   t = 0
32     print *, "x0 = ", L/2.0d0
33     print *, " " = ", L/30.0d0
34
35     open(unit = 1, file = "saidas/evolucao-item-a.dat")
36
37     do i = 1, nx
38         grid(i, 2) = gaussian(i*dx, L/2.0d0, L/30.d0)
39     end do
40
41     grid(:, 1) = grid(:, 2)
42     grid(:, 3) = 0.d0
43     call save_wave_state(grid, nx)
44
45     !   posicao x = L/4 no vetor das ondas.
46     ni = nx / 4
47     print *, "Posicao L/4 = ", ni
48
49     Y(1) = grid(ni, 2)
50
51     do i = 3, nt
52         call drive_wave_fixed(grid, nx, r)
53         !   call drive_wave_free(grid, nx, r)
54         call save_wave_state(grid, nx)
55         Y(i) = grid(ni, 2)
56     end do
57     close(1)
58
59     call dft(Y, nt, dt)
60     end

```

Nesse arquivo principal foram feitas todas as simulações do projeto. Dessa forma, para executar rotina de ondas com bordas fixas ou solta basta comentar ou descomentar a linha relativa à chamada dessas rotinas.

Como foi dito, todas as rotinas estão espalhadas pelo diretório central, sendo os arquivos delas rotinas-ondas-12559016.f e rotinas-dft-12559016.f. Além desses arquivos também há um arquivo tipo Makefile que compila essas rotinas. Não há nada de especial nele, apenas automatização do comando

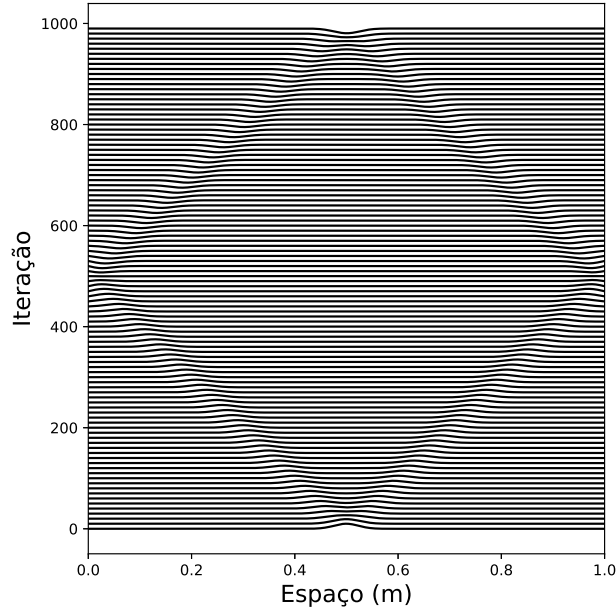
```
gfortran tarefa-1-12559016.f rotinas-ondas-12559016.f rotinas-dft-12559016.f -o tarefa-1-12559016.exe
```

Em todas as simulações os parâmetros utilizados foram mantidos fixos com $\sigma = L/30$ e a posição de observação sendo em $x = L/4$. Foi estudado a propagação de onda e feita análise do espectro de potências para diversos casos de posições iniciais x_0 para pulso do perfil gaussiano.

2.2 Simulação para $x_0 = L/2$

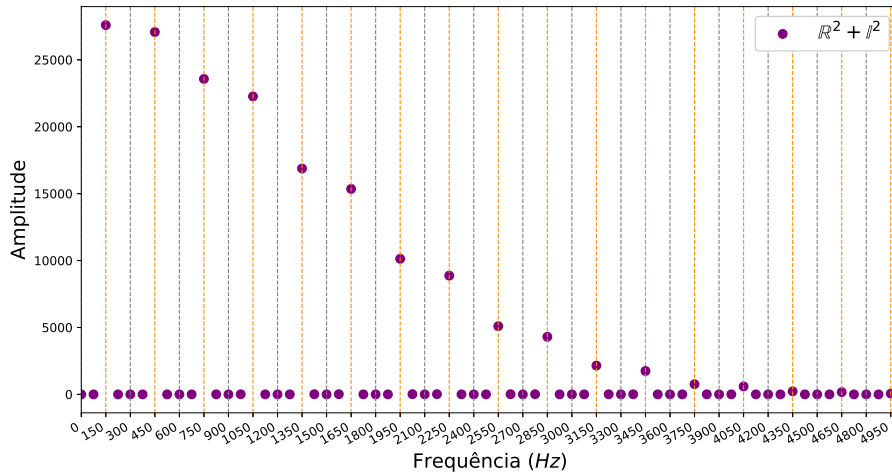
Como podemos ver a simulação está centrada na origem e ocorrem as reflexões em $x = 0$ e $x = L$.

Figura 1:



Com isso temos o resultado para o espectro de potências sendo (2). A simetria da escolha de x_0 implica em apenas modos ímpares contribuírem para perfil da oscilação.

Figura 2:

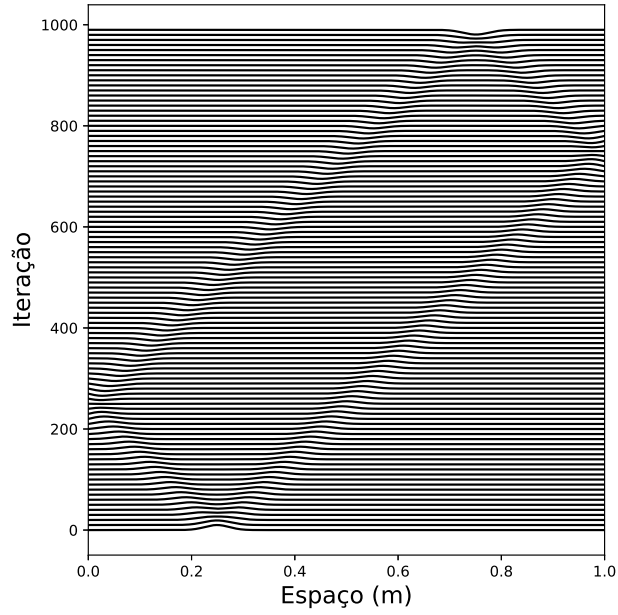


Nota-se que, como foi dito na discussão em (2) surgem modos fundamentais de frequência igual à $f_n = n \cdot 150$ (Hz) e esse é o resultado que foi obtido no projeto anterior para esses parâmetros, já que $\lambda = 2$ (m). Além disso, podemos ver na (2) que os tracejados em cinza são referentes às frequências múltiplas de 300 (Hz) que não fazem parte do conjunto de modos fundamentais.

2.3 Simulação para $x_0 = L/4$

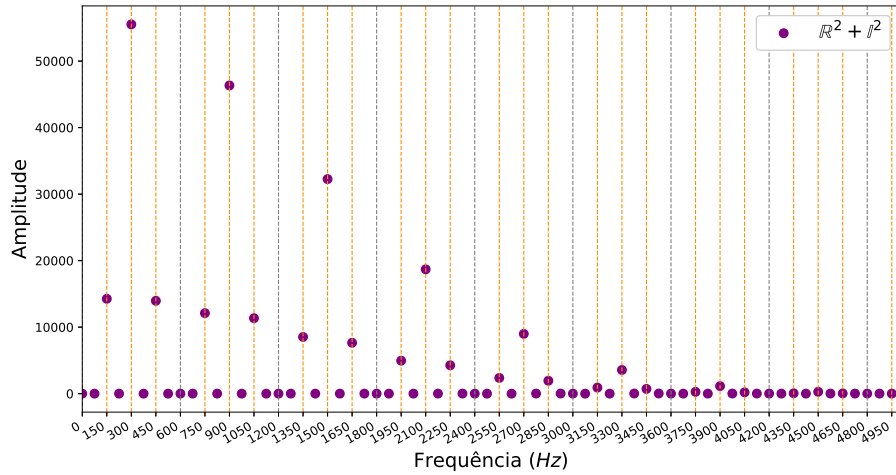
A posição x_0 coincide com a posição em que fazemos nossa observação. Como foi discutido em (2), não será possível observar os modos múltiplos 4, é como ter posicionado o “sensor” em um dos modos e amplitude é fixa igual à zero.

Figura 3:



Pelo gráfico (4) podemos ver que de fato frequências múltiplas de $4f_1$ não fazem parte do espectro. Além disso as frequências pares que compõe o espectro tem amplitude maiores que às impares.

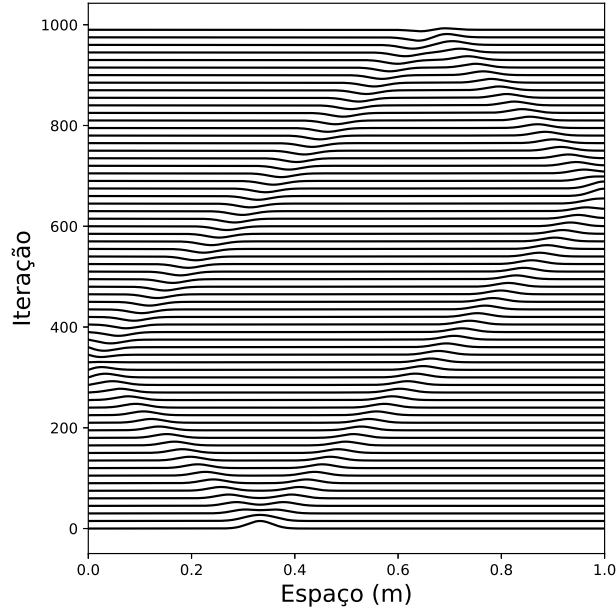
Figura 4:



2.4 Simulação para $x_0 = L/3$

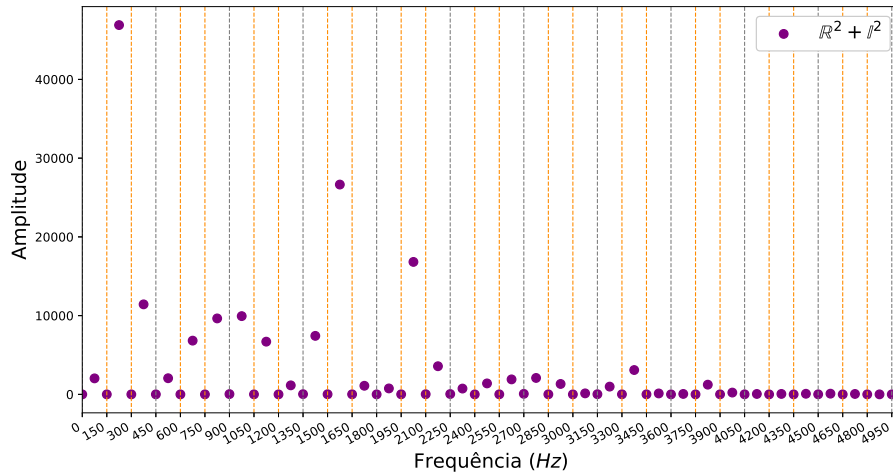
Mais uma vez os modos referentes à parte das frequências fundamentais não estão presente nas componentes espectrais em (6).

Figura 5:



Observa-se que os modos múltiplos de 3 e 4 não compõem o espectro de potências.

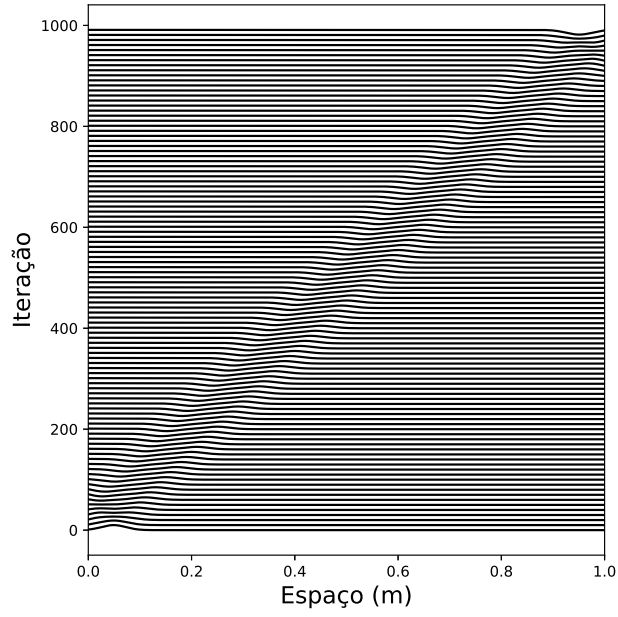
Figura 6:



As condições iniciais impostas nessa simulação são tais que os senos se anulam para frequências fundamentais múltiplas de 3. Isso também contribui para a amplitude do espectro observada para as componentes de termos pares.

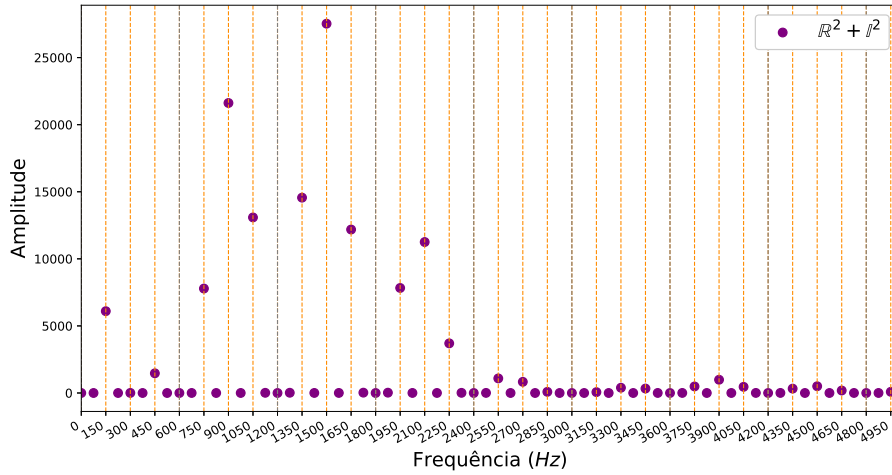
2.5 Simulação para $x_0 = L/20$

Figura 7:



Nessa condição as componentes espectrais tem aparência de gaussianas e não ficam tão localizadas em pontos de frequências baixas como os resultados anteriores. Isso pode ser visto pela diferença entre as componentes de frequência 300 (Hz) e 900 (Hz). Há um salto muito grande entre $2f_1$ e $6f_1$.

Figura 8:



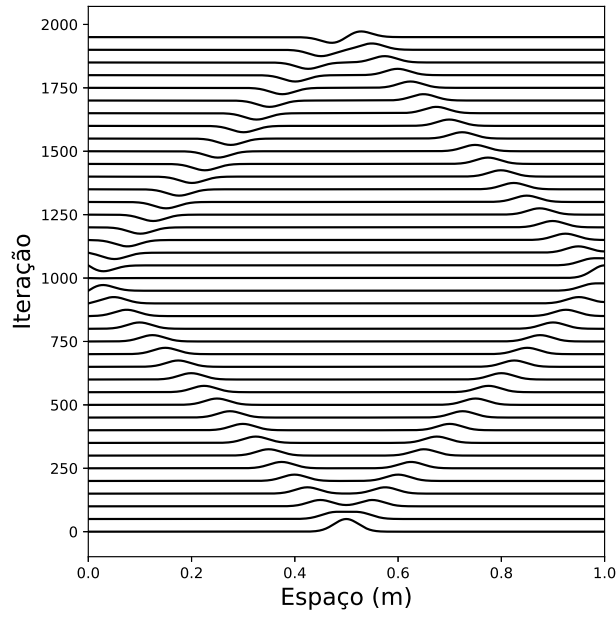
Novamente os modos faltantes são os de múltiplos de $4n$.

2.6 Simulação para $x_0 = L/2$ (Borda solta)

Para a implementação dessa simulação foi mudado apenas a rotina chamada no programa principal e executado.

Nota-se que na última iteração na figura (9) podemos ver a forma da onda se propagando.

Figura 9:



Da figura (10) podemos notar que as frequências fundamentais passam a ser $f_n = (75 + 150n)$ (Hz), ou seja, há uma fase adicionada à componente do seno, isso satisfaz a condição de ser máximo em $x = L$.

Figura 10:

