

Métodos Computacionais – Ondas

Aula: Resolução numérica da Eq. de
Onda em 1D

Equação de onda e métodos espectrais

- Resolução numérica de equações de onda do tipo:

$$\frac{\partial^2}{\partial t^2} y(x, t) = c^2 \frac{\partial^2}{\partial x^2} y(x, t)$$

- Métodos espectrais (análise de Fourier)

Aula: Eq. de Onda - Objetivos

Com essa aula, você deverá ser capaz de:

- 1) Implementar a solução numérica para a propagação de uma onda transversal em 1D dadas as condições iniciais e as condições de contorno.
- 2) Fazer uma *animação* da onda em Octave/MatLab.

Tarefa: Escrever um script para calcular a propagação de um pacote de onda gaussiano em 1D $y(x,t)$ com o método de Euler discretizado em x e em t . Para cada tempo t , plotar pontos $y(x,t)$ vs x em um “*frame*” de modo que a sequência de todos os *frames* formará uma animação da dinâmica do pacote.

Tempo aproximado: 40 a 50 min.

Solução numérica da Eq. de Onda

Solução aproximada de:

$$\frac{\partial^2}{\partial t^2} y(x, t) = c^2 \frac{\partial^2}{\partial x^2} y(x, t)$$

Aproximação da 2a derivada.

$$\frac{\partial^2 y(x, t)}{\partial t^2} \approx \frac{y(x, t + \Delta t) - 2y(x, t) + y(x, t - \Delta t)}{(\Delta t)^2}$$

$$\frac{\partial^2 y(x, t)}{\partial x^2} \approx \frac{y(x + \Delta x, t) - 2y(x, t) + y(x - \Delta x, t)}{(\Delta x)^2}$$

Para determinar a evolução temporal, queremos uma eq do tipo:

$$y(x, t + \Delta t) = F[y(x, t' \leq t)]$$

Solução numérica da Eq. de Onda

Aproximação da 2a derivada.

$$y(x, t + \Delta t) - 2y(x, t) + y(x, t - \Delta t) = \frac{c^2(\Delta t)^2}{(\Delta x)^2} [y(x + \Delta x, t) - 2y(x, t) + y(x - \Delta x, t)]$$

Definimos o parâmetro adimensional: $r \equiv \frac{c\Delta t}{\Delta x}$

E temos uma expressão para $y(x, t + \Delta t)$:

$$y(x, t + \Delta t) = 2(1 - r^2)y(x, t) + r^2[y(x + \Delta x, t) + y(x - \Delta x, t)] - y(x, t - \Delta t)$$

que nos permite determinar a evolução temporal dados: r , $y(x, t)$, $y(x \pm \Delta x, t)$ e $y(x, t - \Delta t)$.

Solução numérica da Eq. de Onda

Expressão discretizada: $y(x, t) \rightarrow y(x_i, t_n) \equiv y(i, n)$

$$x_i = (i - 1) \Delta x \quad t_n = (n - 1) \Delta t \quad r \equiv \frac{c \Delta t}{\Delta x}$$
$$i, n = 1, 2, 3, \dots$$

A expressão para $y(i, n+1)$ será

$$y(i, n + 1) = 2(1 - r^2)y(i, n) + r^2[y(i + 1, n) + y(i - 1, n)] - y(i, n - 1)$$

que nos permite determinar a evolução temporal dados: r , $y(i, n)$, $y(i \pm 1, n)$ e $y(i, n-1)$.

Obs: No primeiro passo, “ $y(i, 0)$ ” não está definido de modo que usamos:

$$y(i, 2) = 2(1 - r^2)y(i, 1) + r^2[y(i + 1, 1) + y(i - 1, 1)] - y(i, 1)$$

Condições iniciais e de contorno

Condições iniciais:

$$\left\{ \begin{array}{l} y(x, t \leq 0) = f(x) \quad (\text{escolha}) \\ \left. \frac{\partial}{\partial t} y(x, t) \right|_{t \leq 0} = v(x) \quad (\text{tomaremos}=0) \end{array} \right.$$

Condições de contorno:

$$\left\{ \begin{array}{l} y(x_p, t) = y_p = \text{const} \quad \forall t \\ \text{e/ou} \quad (\text{p} \rightarrow \text{pontos específicos}) \\ \left. \frac{\partial}{\partial x} y(x_p, t) \right|_{x_p} = y'_p = \text{const} \quad \forall t \end{array} \right.$$

Exemplo de condição de contorno: $y(x_0=0, t) = y(x_1=L, t) = 0$ para qualquer t .

Na versão discretizada: $y(1, n) = y(N_x, n) = 0$ para qualquer n .

Eq. de Onda – Tarefa (Fazer upload!)

Uma onda (ideal) descrita pela função $y(x,t)$ se propaga com velocidade $c=50\text{m/s}$ em uma corda com extremos em $x_0=0$ e $x_1=1\text{m}$.

- *Considere que inicialmente a $y(x,0)$ tem o perfil de um pulso gaussiano centrado na posição $x_c=0,3\text{m}$ e $k=1000\text{ m}^{-1}$:*

$$y(x, t \leq 0) = e^{-k(x-x_c)^2}$$

- *Os extremos da corda estão presos. Ou seja, as condições de contorno são:*

$$y(x_0, t) = y(x_1, t) = 0 \quad \forall t$$

- *Faça um script que calcule $y(x,t)$ de $t=0$ a $t=0,1\text{s}$ com passo $\Delta t=0,0001$ e $r=c(\Delta t/\Delta x) = 1$*
- *Determine o Δx a ser usado a partir dos dados acima.*

Eq. de Onda – Tarefa - Dicas

- *Debug: seguem os valores de $y(i=60,61,62,63,n=1,2,3,4,5)$.*

```
n=1: tempo=0.0000  y(60,n)=0.9753  y(61,n)=1.0000  y(62,n)=0.9753  y(63,n)=0.9048
n=2: tempo=0.0001  y(60,n)=0.9295  y(61,n)=0.9506  y(62,n)=0.9295  y(63,n)=0.8690
n=3: tempo=0.0002  y(60,n)=0.8443  y(61,n)=0.8591  y(62,n)=0.8443  y(63,n)=0.8013
n=4: tempo=0.0003  y(60,n)=0.7309  y(61,n)=0.7380  y(62,n)=0.7309  y(63,n)=0.7092
n=5: tempo=0.0004  y(60,n)=0.6029  y(61,n)=0.6027  y(62,n)=0.6029  y(63,n)=0.6022
```

- *Para mostrar o resultado, você pode fazer um “filme” usando o comando `getframe(figure)`;*
- *A idéia é fazer um plot de $y(i,n)$ vs x_i para cada valor de n (“time frame”) e “juntar” tudo, fazendo um “filme” enquanto o script roda.*
- *Veja um exemplo de um “filme” de uma queda livre no arquivo*
[Aula10_ExMovie.m](#)
- *É possível salvar o “filme” em formatos como `.avi` (pequise na documentação).*