More PDEs: Realistic Waves on Strings

Include Friction & Gravity

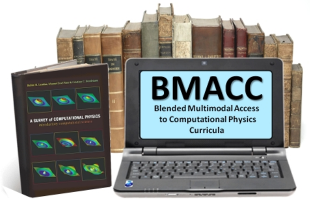
Rubin H Landau

Sally Haerer, Producer-Director

Based on *A Survey of Computational Physics* by Landau, Páez, & Bordeianu

with Support from the National Science Foundation

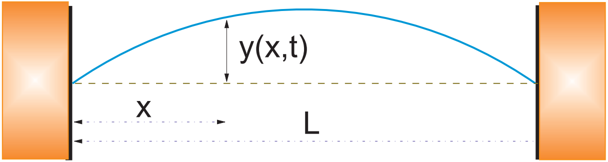
Course: **Computational Physics II**



Numerical Solution of Wave Equations

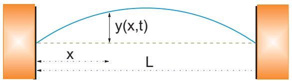
- Many PDE Wave Equations $y(x, t)$
- First standard “wave equation”, then beyond texts
- Again t-stepping, leapfrog algorithm
- Also quantum wave packets (complex), E&M vector
- Also CFD: dispersion, shocks, solitons

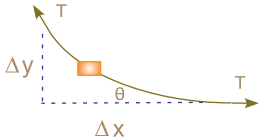
Theory: Hyperbolic Wave Equation



- Recall standing & travel wave demo (do!)
- L = length, fastened at ends
- ρ = density = mass/length = constant
- T = tension = constant = high, no g sag
- No friction
- $y(x, t)$ = small vertical displacement (1D)

Derive Hyperbolic (Linear) Wave Equation





- Small $\frac{y}{L}$
- Small slope $\frac{\partial y}{\partial x}$
- $\sin\theta \simeq \tan\theta = \frac{\partial y}{\partial x}$
- Isolate section Δx
- Restoring force = ΔT_y
- $c = \sqrt{T/\rho} \neq$ string velocity $= \partial y/\partial t$

$$\sum F_y = \rho\,\Delta x\,\frac{\partial^2 y}{\partial t^2} \qquad (F = ma) \qquad (1)$$

$$\sum F_y = T\sin\theta_{x+\Delta x} - T\sin\theta_x = T\left.\frac{\partial y}{\partial x}\right|_{x+\Delta x} - T\left.\frac{\partial y}{\partial x}\right|_{x} \simeq T\frac{\partial^2 y}{\partial x^2}\Delta x \qquad (2)$$

$$\Rightarrow \quad \frac{\partial^2 y(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 y(x,t)}{\partial t^2} \qquad (3)$$

Boundary & Initial Conditions on Solution

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

- PDE: two independent variables x and t
- Initial condition = triangular "pluck":

$$y(x, t=0) = \begin{cases} 1.25x/L, & x \le 0.8L, \\ (5 - 5x/L), & x > 0.8L, \end{cases} \quad (2)$$

- 2^{nd} $\mathcal{O}(t)$ $\Rightarrow$ need 2^{nd} IC
- Released from rest:

$$\frac{\partial y}{\partial t}(x, t=0) = 0, \quad \text{(initial condition 2)} \quad (3)$$

- Boundary conditions for all times

$$y(0,t) \equiv 0, \quad y(L,t) \equiv 0 \quad (4)$$

Normal-Mode Solution (Analytic But ∞)

Applet

1. Assume $y(x,t) = X(x)T(t)$
2. i) Substitute, ii) $\div y$, iii) iff:

$$\frac{d^2T(t)}{dt^2} + \omega^2 T(t) = 0, \quad \frac{d^2X(x)}{dt^2} + k^2 X(x) = 0, \quad k \stackrel{\text{def}}{=} \frac{\omega}{c} \tag{1}$$

3. Determine ω & k via BC

$$\Rightarrow \quad X_n(x) = A_n \sin k_n x, \quad k_n = \frac{\pi(n+1)}{L}, \quad n = 0, 1, \ldots \tag{2}$$

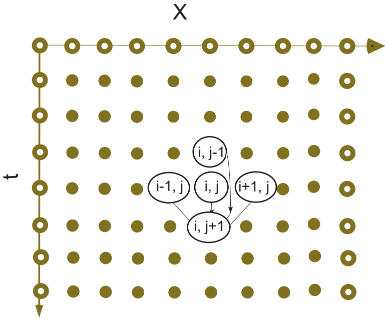
$$T_n(t) = C_n \sin \omega_n t + D_n \cos \omega_n t \tag{3}$$

4. Zero velocity IC2 $\Rightarrow C_n = 0$; linear superposition

$$y(x,t) = \sum_n^{\infty} B_n \sin nk_0 x \cos \omega_n t \tag{4}$$

$$B_m = 6.25 \ \sin(0.8m\pi)/m^2\pi^2 \tag{5}$$

Algorithm: Discretized Wave Equation



- Solve on space-time grid:
- $(x, \ t) = (i\Delta x, \ j\Delta t)$
- BC: vertical white dots
- IC: top row white dots
- Can't relax
- Central-difference derivatives

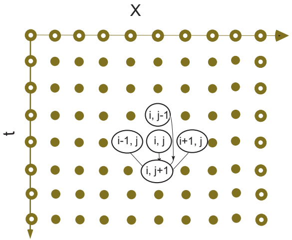
$$\frac{\partial^2 y}{\partial t^2} \simeq \frac{y_{i,j+1} + y_{i,j-1} - 2y_{i,j}}{(\Delta t)^2}, \qquad \frac{\partial^2 y}{\partial x^2} \simeq \frac{y_{i+1,j} + y_{i-1,j} - 2y_{i,j}}{(\Delta x)^2} \tag{1}$$

- Discretized (difference) wave equation:

$$\frac{y_{i,j+1} + y_{i,j-1} - 2y_{i,j}}{c^2(\Delta t)^2} = \frac{y_{i+1,j} + y_{i-1,j} - 2y_{i,j}}{(\Delta x)^2} \tag{2}$$

Wave Equation Algorithm: Time-Stepping

$$\frac{y_{i,j+1} + y_{i,j-1} - 2y_{i,j}}{c^2(\Delta t)^2} = \frac{y_{i+1,j} + y_{i-1,j} - 2y_{i,j}}{(\Delta x)^2} \tag{1}$$

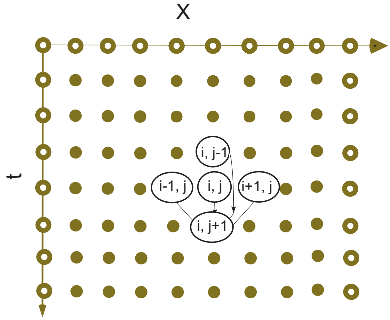


- NB: only 3 times enter
- (j+1, j, j-1)= (future, present, past)
- Predict future:

$$y_{i,j+1} = 2y_{i,j} - y_{i,j-1} + \frac{c^2}{c'^2}\left[y_{i+1,j} + y_{i-1,j} - 2y_{i,j}\right] \tag{2}$$

- $c' \stackrel{\text{def}}{=} \Delta x/\Delta t$
- $\frac{c'}{c}$ determines stability

Discussion: Time Stepping Algorithm



Generalities

- Leapfrog *vs relaxation*
- *Store only 3 time values*
- *Very small* Δt *for high precision*
- *Starting requires* $t < 0$
- *“At rest” IC + CD:*

$$\frac{\partial y}{\partial t}(x,0) \simeq \frac{y(x,\Delta t) - y(x,-\Delta t)}{2\Delta t} = 0$$

$$\Rightarrow \quad y_{i,0} = y_{i,2}$$

von Neumann (Courant) Stability Condition

$$\frac{y_{i,j+1} + y_{i,j-1} - 2y_{i,j}}{c^2(\Delta t)^2} = \frac{y_{i+1,j} + y_{i-1,j} - 2y_{i,j}}{(\Delta x)^2} \qquad (1)$$

General Truth: Can't pick arbitrary $\Delta x, \ \Delta t$

- Substitute into (1) $y_{m,j} = \xi^j \ \exp(ikm\,\Delta x)$
- Avoid exponential growth in time $|\xi| > 1$ (unstable)
- True generally for transport equations (Press):

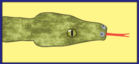
$$c \leq c' = \frac{\Delta x}{\Delta t} \qquad \text{(Courant condition)} \qquad (2)$$

- Better: smaller Δt; worse smaller Δx
- (1) = symmetric, yet IC, BC $\neq$ symmetric

Non Computational Exercises

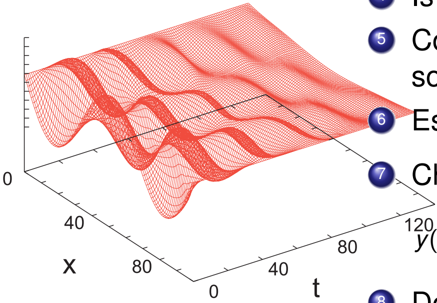
1. Suggest an algorithm to solve wave equation in 1 step.
    1. How much memory is required?
    2. How does this compare with the memory required for the leapfrog method?
2. Suggest an algorithm to solve the wave equation via relaxation (like Laplace's equation).
    1. What would you take as the initial guess?
    2. How would you know when the procedure has converged?
    3. How would you know if the solution is correct?

Wave Equation Implementation

Applet  CODE

- Study `EqString.py`, outlining the structure
- You will need to modify this code to add new physics.
- NB: $L = 1 \;\Rightarrow y/L \ll 1$ not OK ($L = 1000$ better)
- $\rho = 0.01$ kg/m, $T = 40$ N, $\Delta = 0.01$ cm

Assessment



1. Solve wave equation
2. Make surface or animation $y(x, t)$
3. Explore Δx & Δt combos
4. Is stability condition obeyed?
5. Compare "analytic" vs numeric solutions
6. Estimate c via graphs, compare $\sqrt{\frac{T}{\rho}}$
7. Choose IC for single normal mode:

$$y(x, t = 0) = 0.001 \sin 2\pi x, \quad \frac{\partial y}{\partial t}(x, t = 0) = 0$$

8. Do 2 near modes beat?
9. Interference if plucked in middle?