Study Highlights: 01/16/2015

Recall:

- Boundary condition types for the heat equation from the study notes of 01/14/2015.
    - Neumann Boundary condition: derivative specified at endpoints
    - Dirichlet Boundary condition: values of the unknown function specified at endpoints
    - Mixed Boundary Conditions: ONE ENDPOINT is Neumann and one endpoint is Dirichlet
    - Initial Value: When $u(x,0)$ is specified. Needed for time dependent problems.

- Steady State Heat Equation:
    - More detail for the steady state problem can be found in the study highlight notes of 01/14/2015.
    - The term "Steady State" means that $\frac{\partial}{\partial t} u(x,t) = 0$ i.e. that $u(x,t) = u(x)$ depends only on the space variable, x.
    - Steady state problems do not need an initial condition.

$$\left[\frac{\partial}{\partial x} \left(K(x) \frac{\partial}{\partial x} u(x) \right) = f \right]$$ ← the Steady State heat equation.

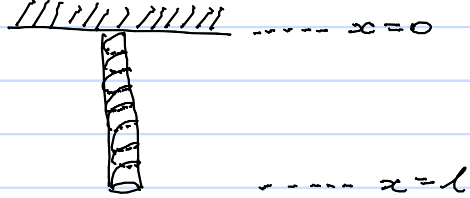
    - Boundary Conditions: Dirichlet, Neumann, or mixed
    - No initial value is needed!

Chapter 2.2: The wave equation for the Hanging Bar

Goal: Derive the Wave Equation by considering a hanging bar (or, equivalently a tightly coiled spring)

- Note: The wave equation can be derived by considering a hanging bar (Chap 2.2) or a vibrating string (Chap 2.3) the same differential equation represents both but the boundary conditions can be different. See Chap 2.2.1 and the end of Chap 2.3, respectively.

Picture: A long thin bar / tightly compressed spring hangs from a fixed wall. The length of the bar is l.

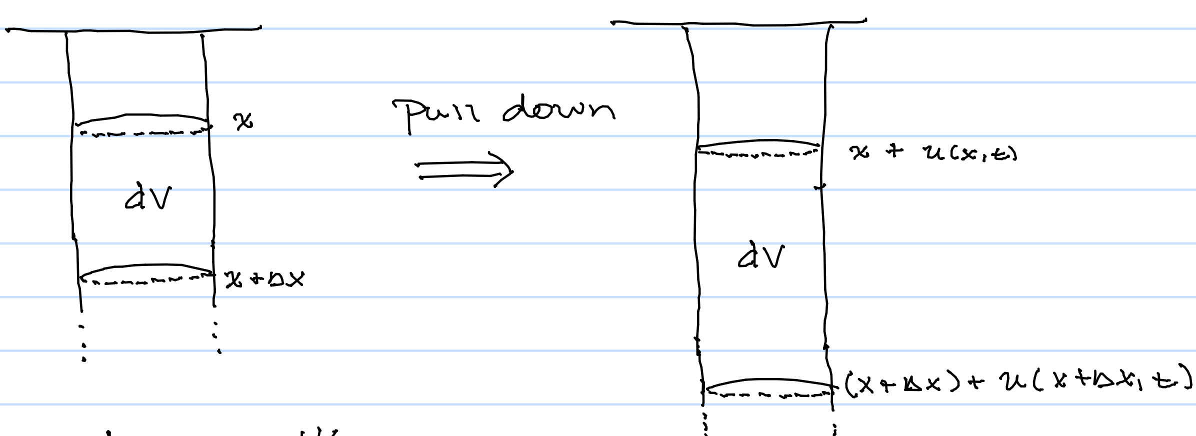


IDEA: Much of the same "volume element analysis" that was done for the heat equation can be repeated here. However, instead of considering <u>heat energy</u> we will be looking at the <u>forces</u> on a small volume element.

IDEA: Focus on a small volume element dV and use Newton's Second law: "$F = ma$" to derive the wave equation.

- Assume pulling the bar results only in vertical motions. Then the motion can be described fully by a displacement function $u(x,t)$.
  - → the part of the bar <u>originally</u> at "x" moves to $x + u(x,t)$

- Small volume element dV between "x" and "$x+\Delta x$"



- Change in length of dV is:

$$\underbrace{\{[(x+\Delta x) + u(x+\Delta x,t)] - [x+u(x,t)]\}}_{\text{Final length}} - \underbrace{\{(x+\Delta x) - x\}}_{\text{initial length}} = u(x+\Delta x,t) - u(x,t)$$

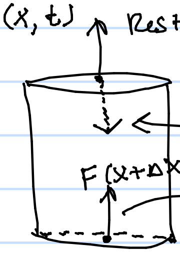
- Change in length of dV <u>relative</u> to original length of dV is given by:

$$\frac{u(x+\Delta x,t) - u(x,t)}{\Delta x}$$

- $\dfrac{u(x+\Delta x,t) - u(x,t)}{\Delta x} \approx \dfrac{\partial}{\partial x} u(x,t)$

- If you stretch a bar | string the forces acting on it, internally, seek to return it to its rest position. Called a <u>restoring</u> force.

- <u>Assumption</u> (the elasticity assumption): the internal <u>restoring forces</u> (per unit area) acting on dV are <u>proportional to the relative change</u> of length of dV.



$F(x,t)$ Restoring force at x acting away from dV so the <u>negative force</u> acts <u>on</u> dV.

$F(x+\Delta x,t)$ Restoring force at $x+\Delta x$ is acting <u>on</u> dV

$$\rightarrow \quad F(x+\Delta x,t) = A\,K(x+\Delta x,t) \frac{\partial}{\partial x} u(x+\Delta x,t)$$
$$F(x,t) = A\,K(x,t) \frac{\partial}{\partial x} u(x,t)$$

$\rightarrow$ Total Force Acting <u>on</u> dV is:

$$A\left(K(x+\Delta x,t) \frac{\partial}{\partial x} u(x+\Delta x,t) - K(x,t) \frac{\partial}{\partial x} u(x,t) \right)$$

Which from the fund. theorem of calculus is:

$$\int_x^{x+\Delta x} A \frac{\partial}{\partial x}\left(K(s,t) \frac{\partial}{\partial x} u(s,t) \right) ds \qquad [1]$$

- Additional "Body forces" (any force which is <u>not</u> an internal restoring force) acting on dV are lumped into a "body force term" per unit area, $f(x,t)$.

- Total body forces acting on dV are thus: $\int_x^{x+\Delta x} A f(s,t)\, ds$

$$\Rightarrow \text{Total force on } dV: \int_x^{x+\Delta x} A \left(\frac{\partial}{\partial x}\left(K(s,t) \frac{\partial}{\partial x} u(s,t) \right) + f(s,t) \right) ds$$

- The mass times the acceleration of the mass dV can be expressed by the term:

$$\int_x^{x+\Delta x} A\rho \frac{\partial^2}{\partial t^2} u(s,t)\, ds$$

- Newtons second law, $F = ma$, applied to dV is thus:

$$\int_x^{x+\Delta x} A \left(\frac{\partial}{\partial x} \left(K(s,t) \frac{\partial}{\partial x} u(s,t) \right) + f(s,t) \right) ds = \int_x^{x+\Delta x} A\rho \frac{\partial^2}{\partial t^2} u(s,t)\, ds$$

- This holds for all x, $x+\Delta x$ and all times t so that we can drop the integration and get:

$$\boxed{\frac{\partial^2}{\partial t^2} u - \frac{\partial}{\partial x} \left(K(x,t) \frac{\partial}{\partial x} u(x,t) \right) = f}$$ The wave equation.

if $K(x,t) = K$ is constant: $\boxed{\frac{\partial^2}{\partial t^2} u - K \frac{\partial^2}{\partial x^2} u = f}$

- For the steady state case: $\boxed{-K \frac{\partial^2}{\partial x^2} u = f}$

▷ Boundary Conditions:

Any physical system will need boundary conditions to be well posed/meaningful. However for the wave equation, since we have a second derivative in time, we need two initial conditions. ONE for the $u(x,0)$ and one for $\frac{\partial}{\partial t}(u(x,0))$

- The actual boundary and initial conditions depend on the physics of situation. Please read Chap 2.2.1 for the hanging bar and for the vibrating string read Chap 2.3.

Finite differences

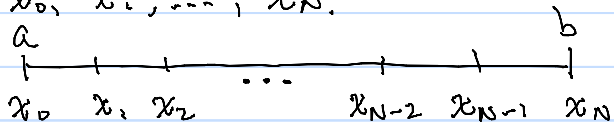
- The finite difference method refers to one of many techniques for transforming a continuous differential equation into a discrete linear algebra problem.

- This material is not covered in your book but can be found in numerous textbooks; the basic concepts are suitably discussed on wikipedia as well.

- General Setup:
  - you have a differential equation with boundary conditions
  - you have an interval $[a,b]$ where you would like to find the solution.

  - Partition the interval into N segments by introducing $N+1$ equally spaced points. Label these points by $x_0, x_1, \dots, x_N$.

    And let $h = 1/N$ denote the length of each segment.



  - For a function f let f_i denote $f(x_i)$ for $i = 0, 1, 2, \dots, N$

  - The idea is to replace the derivatives in your PDE with finite difference approximations. There are s eral ways to do this.

    - Forward Approximations: approximates at x_i usi values after (in front of) x_i

    - Backward Approximations: approx. at x_i using values before (in back of) x_i

    - Central Approximation: approx at x_i using values around (in front + behind) x_i.

Approximations for a first derivative:

Forward: $$\left[\frac{\partial}{\partial x} f\right](x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h} = \frac{f_{i+1} - f_i}{h}$$

Backwards: $$\left[\frac{\partial}{\partial x} f\right](x_i) \approx \frac{f(x_i) - f(x_{i-1})}{h} = \frac{f_i - f_{i-1}}{h}$$

Central: $$\left[\frac{\partial}{\partial x} f\right](x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h} = \frac{f_{i+1} - f_{i-1}}{2h}$$

$$\left[\frac{\partial}{\partial x} f\right](x_i) \approx \frac{f(x_{i+1/2}) - f(x_{i-1/2})}{h} = \frac{f_{i+1/2} - f_{i-1/2}}{h}$$

Approximations for a second derivative:

Forward: $$\left[\frac{\partial^2}{\partial x^2} f\right](x_i) \approx \frac{f_{i+2} - 2f_{i+1} + f_i}{h^2}$$

Backward: $$\left[\frac{\partial^2}{\partial x^2} f\right](x_i) \approx \frac{f_i - 2f_{i-1} + f_{i-2}}{h^2}$$

Central: $$\left[\frac{\partial^2}{\partial x^2} f\right](x_i) \approx \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}$$

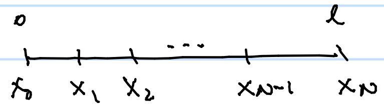
Key idea: Some approximations, for the same derivative, are better than others! The accuracy of an approximation is called its <u>order</u>; the higher the order, the better.

<u>Ex:</u> the forward difference for approximating $\frac{\partial}{\partial x} f$ is <u>first order</u> while the central difference is <u>second order</u>.

Central Differences for the Steady State Heat equation.

Suppose we want to solve: $\frac{\partial^2}{\partial x^2} u(x) = f(x)$ on a bar of length l with the boundary conditions $u(0) = u(l) = 0$ (e.g. *homogeneous* Dirichlet conditions)

Key idea: we want the solution only at a set of equally spaced discrete points $x_0 = 0$, $x_1, \dots, x_N = l$ of the interval $[0, l]$.



The true solution, $u(x)$, would satisfy $\left[\frac{\partial^2}{\partial x^2} u\right](x_i) = f(x_i)$ at each of the points $x_0, x_1, \ldots, x_N$.

Key idea: Approximate $\left[\frac{\partial^2}{\partial x^2} u\right](x_i)$ by central finite differences

$$\left[\frac{\partial^2}{\partial x^2} u\right](x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \qquad \text{where, recall, } h = 1/N.$$

- From the boundary conditions we know that $u_0 = 0$ and $u_N = u(x_N) = u(\ell) = 0$ so for $i = 1, 2, \ldots, N-1$ we have:

$$\frac{1}{h^2}\left(u_{i+1} - 2u_i + u_{i-1}\right) = f_i \qquad (*)$$

If we treat $\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}$ as a vector of unknowns

and $\vec{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}$ as a <u>known</u> vector (we presumably know the function f if we are solving the steady state equation)

then $(*)$ can be written as a discrete system of the form:

$$\frac{1}{h^2} A \vec{u} = \vec{f} \quad \text{where } A \text{ is a } (N-1)\times(N-1) \text{ matrix.}$$

What matrix is A? Lets inspect the rows.

If $i=2$ (row 2 of A) then $(*)$ is: $\frac{1}{h^2}\left(u_3 - 2u_2 + u_1\right) = f_2$

So that row 2 of A is $[1 \; -2 \; 1 \; 0 \; 0 \; 0 \cdots 0]$

The same will hold for $i = 3, 4, \ldots, N-2$.

What about $i=1$? $(*)$ becomes: $\frac{1}{h^2}\left(u_2 - 2u_1 + u_0\right) = f_1$

but we know that $u_0 = u(x_0) = u(0) = 0$! so this becomes $\frac{1}{h^2}\left(u_2 - 2u_1\right) = f_1$ so row one is $[1 \; -2 \; 0 \; 0 \cdots 0]$

What about when $i = N-1$? $(*)$ is: $\frac{1}{h^2}\left(u_N - 2u_{N-1} + u_{N-2}\right) = f_{N-1}$

but $u_N = u(x_N) = u(\ell) = 0$! so: $\frac{1}{h^2}\left(-2u_{N-1} + u_{N-2}\right) = f_{N-1}$

So row $N-1$ is: $[0 \; 0 \; 0 \; \cdots\cdots \; 0 \; 1 \; -2]$

- Hence we have the linear system:

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & 0 & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & 0 & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{N-2} \\ f_{N-1} \end{bmatrix} \qquad (▨)$$

Note: we already know that $0 = u(0) = u(x_0)$ and $0 = u(\ell) = u(x_N)$ but notice that the boundary conditions played an important role in deriving the linear system (▨).

Q: Can you write a program to solve (▨) in Matlab for your favorite function f, choice of length ℓ and a user input number of points, N?