Lecture 20: Initial Boundary Value Problems

Today:

- ***Finite difference methods*** applied to ***initial boundary value problems (IBVPs)***
  - ***Punchline***: approximate solution to an ***IBVP*** by combining our FD methods for BVPs and IVPs

Where are we up to now?

Last several weeks.

(A) We developed ***finite difference methods*** (***one-step*** and ***multi-step***) methods for solving ***initial value problems.***

(B) We developed a suite of methods (finite difference, global spectral, and finite element) for ***boundary value problems.***

(C) We learned how to characterize the error for these methods and how to implement them.

This week. Finite difference methods for IBVPs — PDEs that depend on time and space

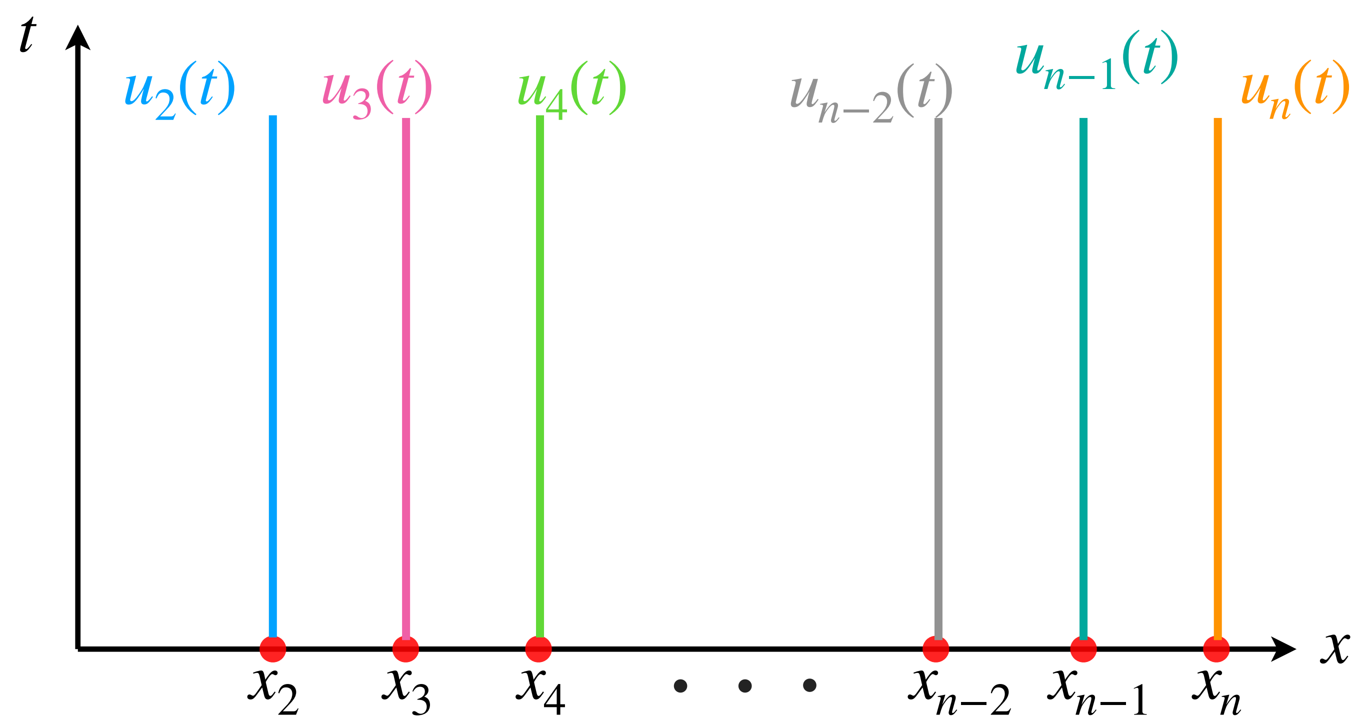
- Use an FD method to ***discretize*** the PDE in space → gives an IVP
- Solve the resulting IBVP with an FD method

This approach is called the ***method of lines***

This discretization approach is called the ***method of lines***

Discretizing the IBVP in space to get an IVP, and then solving that using your favorite time stepping method is called the ***method of lines***

Why? Schematically:



After discretizing in space, the solution at each spatial point evolves "along its own line" in time

Reminder: what is an IBVP?

Before talking about solving IBVPs, let's remind ourselves what an IBVP is.

An IBVP is a ***partial differential equation*** that depends on ***space and time.*** Most of the key engineering phenomena we care about are governed by an IBVP:

- The Navier-Stokes equations
- Governing equations for structural deformation (Euler Bernoulli beams, plates & shells, etc.)
- …

Solving these equations numerically is an active area of research, so we will focus on a canonical equation. The building blocks we develop here form the heart of methods developed for these more complex systems!

The canonical IBVP we will consider is the ***heat equation***

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + g(x,t), \quad 0 \leq t \leq T, \; a \leq x \leq b \tag{1}$$

This is the 1D version. Variants exist for 2D & 3D

where κ is the heat diffusivity and $g(x,t)$ is a prescribed forcing.

We need both ***boundary conditions*** and ***initial conditions*** to specify the solution

$$u(x, t=0) = \eta(x)$$

Prescribed initial condition; $\eta(x)$ is given

$$u(x=a, t) = g_a(t)$$
$$u(x=b, t) = g_b(t)$$

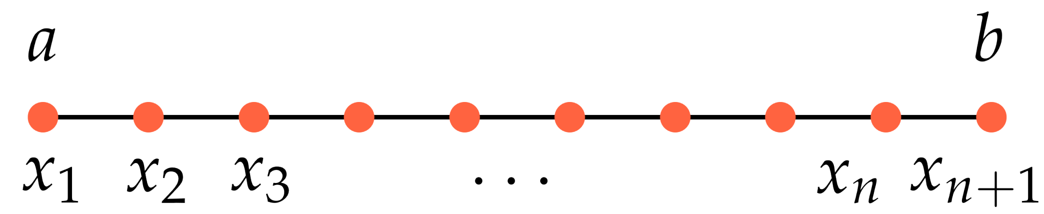
Prescribed boundary conditions; $g_a(t)$ and $g_b(t)$ are given

These are Dirichlet boundary conditions. Other options (Neumann, mixed) are possible!

Discretizing the IBVP in *space*

Just as for BVPs, we will use a uniformly distributed set of points:

$$x_j = a + \frac{(b-a)(j-1)}{n}, \quad j = 1, \ldots, n+1$$



Approximate the solution as a linear combination of locally defined basis functions

Remember, this is the i^{th} Lagrange basis function about x_j (Lecture 14, slides 6-7)

$$u(x,t) \approx \sum_{i=j-p/2}^{j+p/2} u_i(t) L_i^{(j)}(x) \qquad (2)$$

$$L_k^{(j)}(x) = \prod_{\substack{m=j-p/2 \\ (m \neq k)}}^{j+p/2} \frac{x - x_m}{x_k - x_m} \qquad k = j-p/2, \ldots, j, \ldots, j+p/2.$$

What's the difference between (2) and the analogous expression for BVPs?

Plug the approximation (2) into the PDE (1), and evaluate at x_j to get

$$\sum_{i=j-p/2}^{j+p/2} \dot{u}_i(t) L_i^{(j)}(x_j) = \kappa \sum_{i=j-p/2}^{j+p/2} u_i(t) \left.\frac{d^2 L_i^{(j)}}{dx^2}\right|_{x=x_j} + g(x_j, t) \qquad (j = 2, \ldots, n)$$

Remember that $L_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

We can say more about this when $p = 2$…

Simplifying the ***spatially discrete*** PDE for $p = 2$

$$\implies \dot{u}_j(t) = \kappa \sum_{i=\cancelto{1}{j-p/2}}^{\cancelto{1}{j+p/2}} u_i(t) \left.\frac{d^2 L_i^{(j)}}{dx^2}\right|_{x=x_j} + g(x_j, t) \qquad (j = 2, \ldots, n)$$

We showed in Lecture 14, slide 10 that this term is $\frac{1}{\Delta x^2}(u_{j-1}(t) - 2u_j(t) + u_{j+1}(t))$

This gives an ODE in time for each $j = 2, \ldots, n$. Can aggregate into a linear system of IVPs:

$$\dot{\begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}} = \frac{\kappa}{\Delta x^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} + \begin{bmatrix} g(x_2, t) + \frac{\kappa g_a(t)}{\Delta x^2} \\ g(x_3, t) \\ \vdots \\ g(x_{n-1}, t) \\ g(x_n, t) + \frac{\kappa g_b(t)}{\Delta x^2} \end{bmatrix} \qquad (3)$$

These two rows are different because of the BCs. Take the first row:

$$\dot{u}_2 = \frac{\kappa}{\Delta x^2}(\cancel{u_1} - 2u_2 + u_3) + g(x_2, t)$$

Using the BC, $u_1 = u(x_1, t) = g_a(t)$

Similar arguments apply to the last row.

Spatially discretizing leads to an IVP

Can write equation (3) more succinctly as $\dot{\boldsymbol{u}} = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{g}$

But this is just an IVP in the desired first order form, with $\boldsymbol{f}(\boldsymbol{u}, t) = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{g}(t)$

Solve using your favorite time stepping method (backward Euler, trapezoid method, ...)

To implement, we need to construct an initial condition for $\boldsymbol{u}$

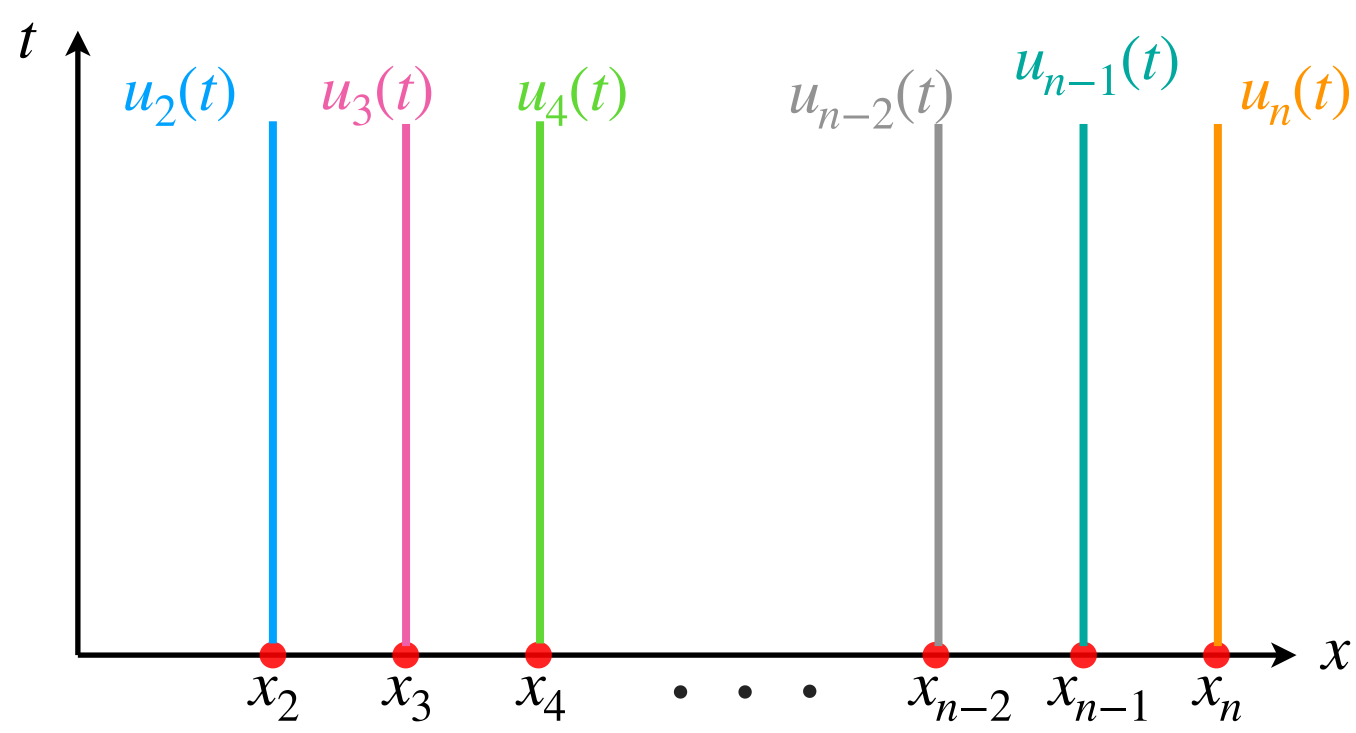
> Remember from slide 3 that the IC for the IBVP is $u(x, t = 0) = \eta(x)$, where η is a function given to us.
>
> Then our IC for the spatially discrete $\boldsymbol{u}$ can be constructed by evaluating at the discretization points $x_2, \ldots, x_n$:

$$\boldsymbol{u}(t = 0) = \begin{bmatrix} \eta(x_2) \\ \eta(x_3) \\ \vdots \\ \eta(x_{n-1}) \\ \eta(x_n) \end{bmatrix}$$

This discretization approach is called the ***method of lines***

Discretizing the IBVP in space to get an IVP, and then solving that using your favorite time stepping method is called the ***method of lines***

Why? Schematically:



After discretizing in space, the solution at each spatial point evolves "along its own line" in time