Lec 20: Partial Differential Equations II

15-369/669/769: Numerical Computing

Instructor: Minchen Li

Table of Content

Table of Content

Representing Derivative Operators

Key Intuition

- **Central Principle**: Derivatives act on functions in the same way that sparse matrices act on vectors.
- This analogy motivates viewing differential operators as linear operators.
- It enables the use of linear–algebraic tools when constructing numerical PDE methods.

Representing Derivative Operators

Linearity

- Differentiation is linear:

$$\frac{d}{dx}(af(x) + bg(x)) = af'(x) + bg'(x).$$

- Therefore, derivatives behave like matrix–vector multiplication on an infinite-dimensional space.

- PDE operator:

$$(\nabla^T A \nabla + \nabla \cdot b + c)u = 0.$$

- Boundary operator:

$$R_{\partial\Omega} u = u_0.$$

- Can be combined in matrix-like notation: $Mu = w$.

Representing Derivative Operators

Discretization as Linear Systems

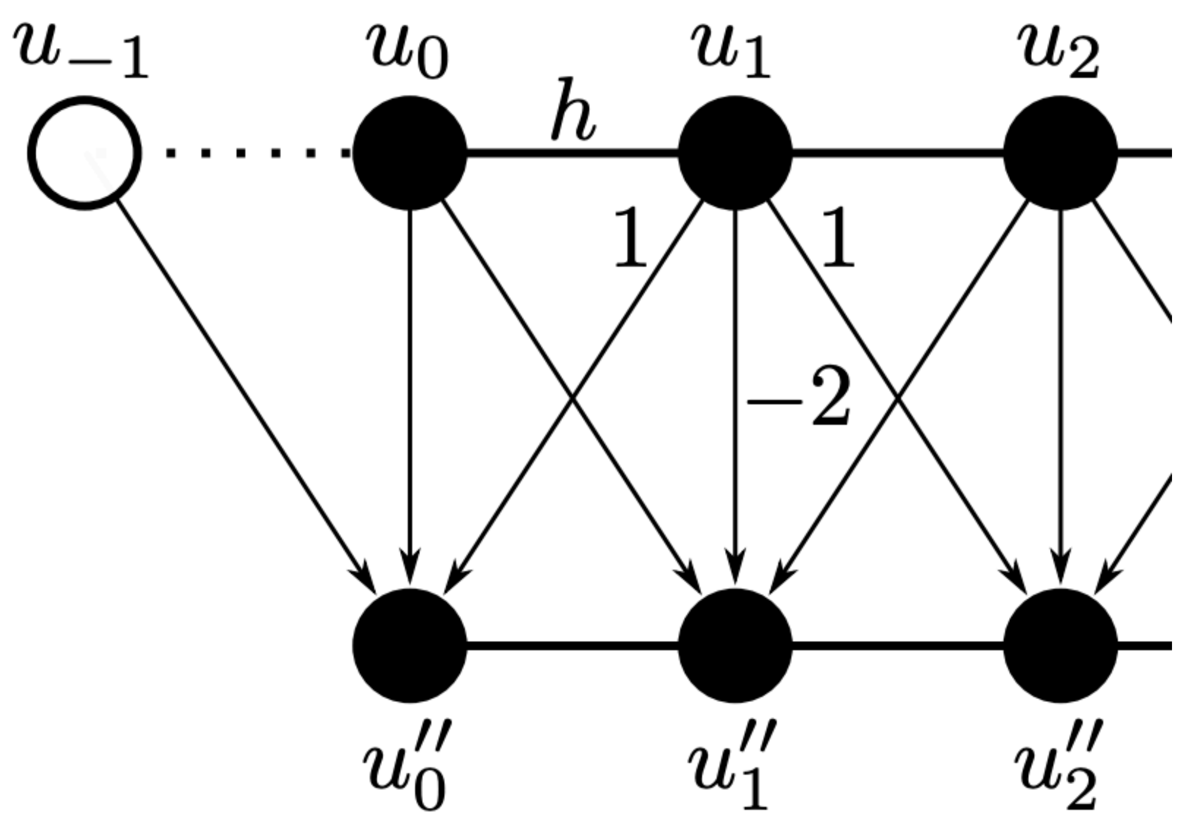
- If we discretize M as a matrix, then recovering the solution u of the original equation is as easy as writing

$$``u \approx M^{-1} w''.$$

- Different PDE solvers differ mainly in:
  - how derivatives are discretized,
  - stability and accuracy properties,
  - computational cost of solving $Mu = w$.

Representing Derivative Operators

Finite Differences (1D): Second-Order Derivatives



- Grid: $x_k = kh$ with $h = 1/n$.
- Centered second-difference approximation:

$$u''(x_k) \approx \frac{u(x_k + h) - 2u(x_k) + u(x_k - h)}{h^2}.$$

- Discrete form:

$$u''_k \approx \frac{u_{k+1} - 2u_k + u_{k-1}}{h^2}.$$

- Corresponding stencil:

$$(1,\ -2,\ 1).$$

Representing Derivative Operators

Finite Differences (1D): Visualization of the Laplacian Operator

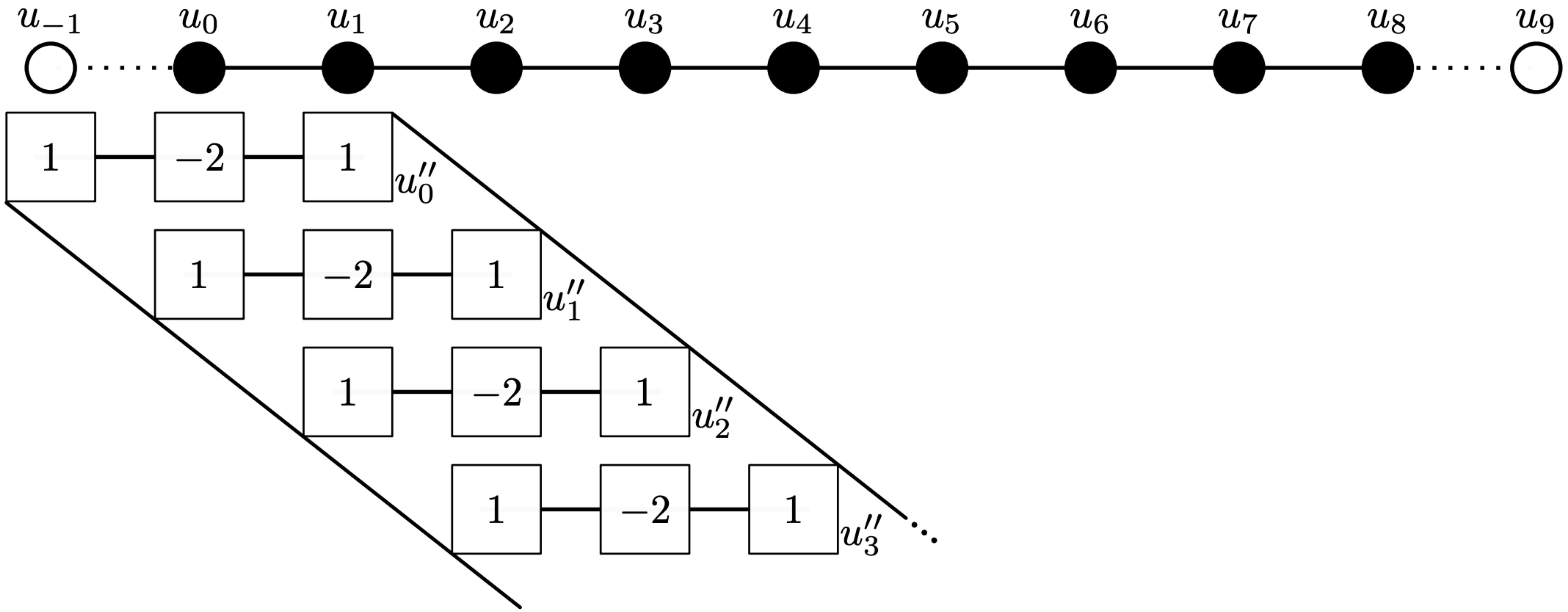


Figure 16.12 The one-dimensional finite difference Laplacian can be thought of as dragging a (1)—(−2)—(1) stencil across the domain.

Representing Derivative Operators

Finite Differences (1D): Boundary Conditions

- Second derivatives at boundaries require fictitious u_{-1} or u_{n+1}.
- Typical options:
  - **Dirichlet:** $u_{-1} = u_{n+1} = 0.$
  - **Neumann:** $u_{-1} = u_0, \; u_{n+1} = u_n.$
  - **Periodic:** $u_{-1} = u_n, \; u_{n+1} = u_0.$
- Stacked system:

$$h^2 w = Lu.$$

where we stack the samples u_k into a vector $u \in \mathbb{R}^{n+1}$ and the samples u_k'' into a second vector $w \in \mathbb{R}^{n+1}$, and L is one of the choices below.

Representing Derivative Operators

Discrete Laplacian with Different Boundary Conditions

Dirichlet

$$\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}$$

Neumann

$$-2u_0 + u_0 \downarrow$$

$$\begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}$$

$$\uparrow -2u_n + u_n$$

Periodic

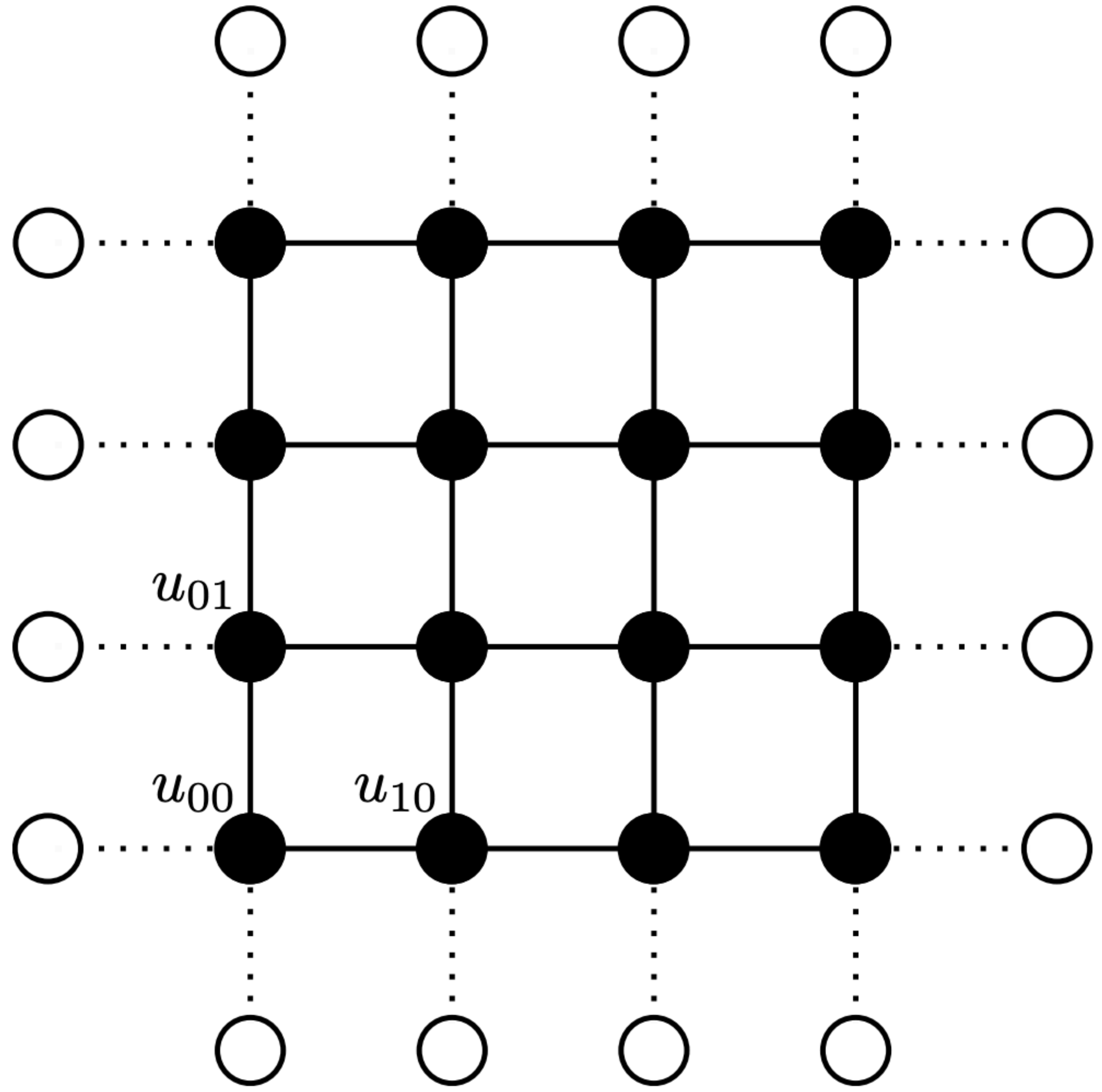
$$u_n \downarrow$$

$$\begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix}$$

$$\uparrow u_0$$

Representing Derivative Operators

Finite Differences (2D): Laplacian



- Grid samples:

$$u_{k,\ell} = u(kh, \ell h).$$

- 5-point Laplacian:

$$(\nabla^2 u)_{k,\ell} \approx \frac{u_{k-1,\ell} + u_{k+1,\ell} + u_{k,\ell-1} + u_{k,\ell+1} - 4u_{k,\ell}}{h^2}.$$

- Stencil:

$$\begin{matrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{matrix}$$

- Produces a large sparse matrix L_2.

Representing Derivative Operators

Finite Differences (2D): Solving Poisson's Equation

- PDE:

$$\nabla^2 u = w.$$

- Discrete system:

$$L_2 u = h^2 w.$$

- Boundary nodes fixed:

$$u_{k,\ell} = u_0(kh, \ell h).$$

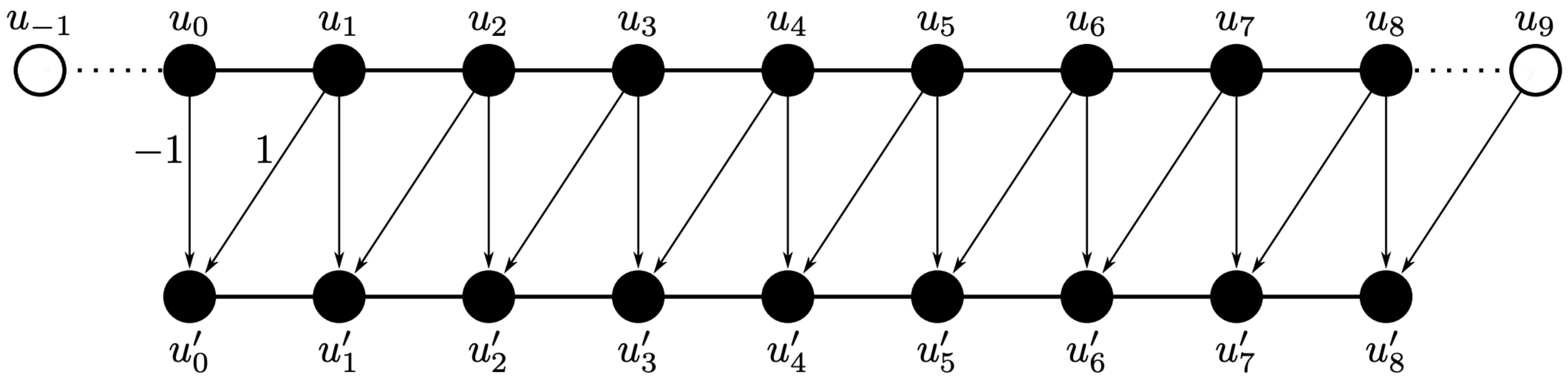
- Interior equation:

$$u_{k-1,\ell} + u_{k+1,\ell} + u_{k,\ell-1} + u_{k,\ell+1} - 4u_{k,\ell} = h^2 w_{k,\ell}.$$

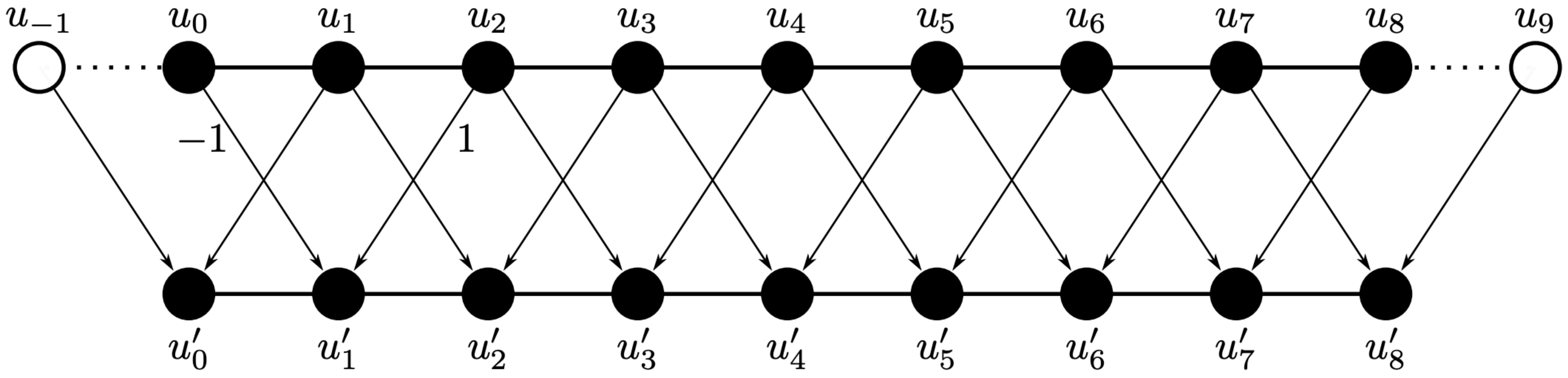
Representing Derivative Operators

Finite Differences: Failure of First-Order Discretizations

- Forward difference: $u'_k \approx \frac{u_{k+1} - u_k}{h}$ creates asymmetric boundary requirements.



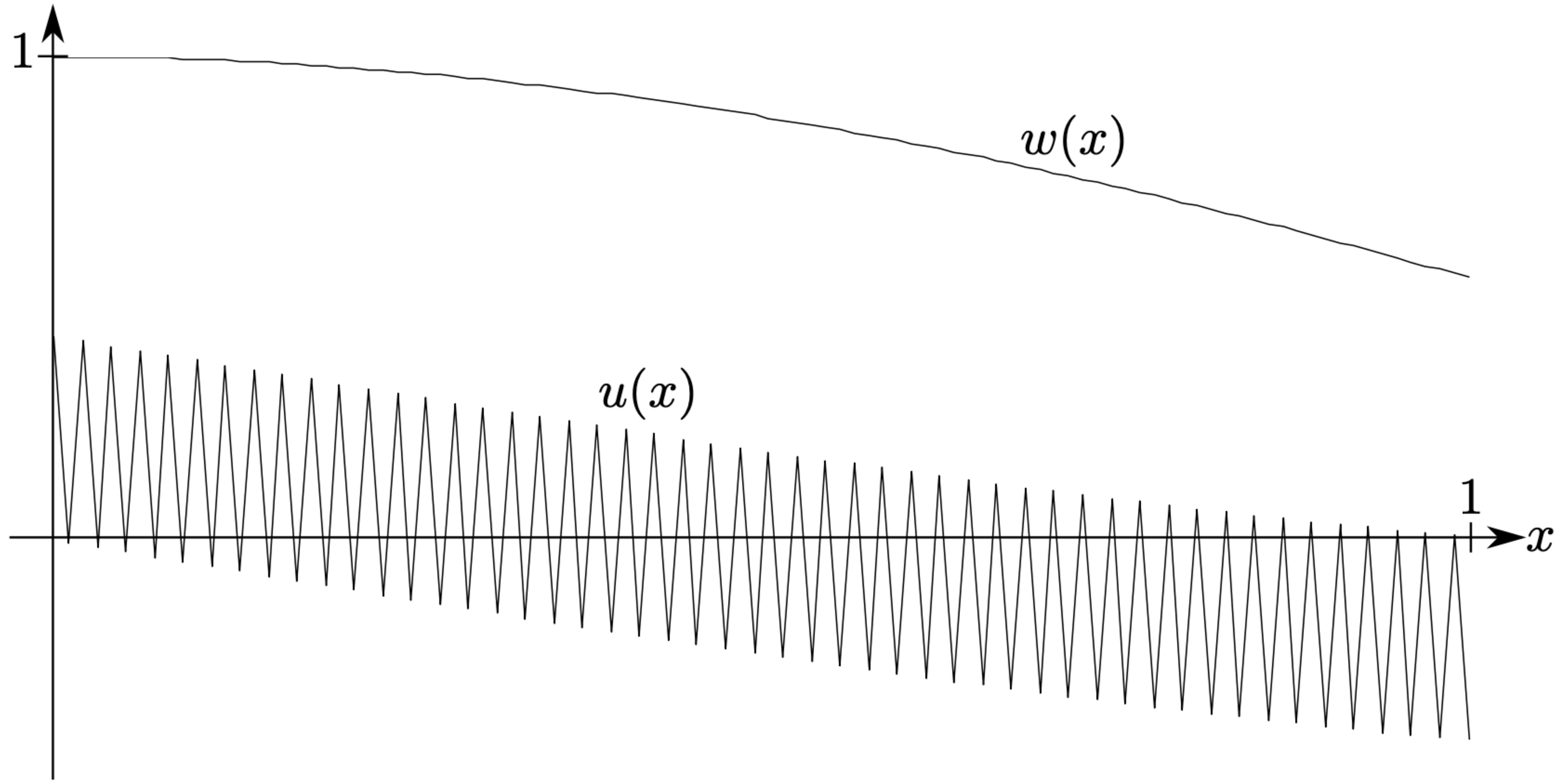
- Symmetric difference: $u'_k \approx \frac{u_{k+1} - u_{k-1}}{2h}$ suffers from the **fencepost problem**.



Representing Derivative Operators

Finite Differences: Failure of First-Order Discretizations

- Forward difference: $u'_k \approx \frac{u_{k+1} - u_k}{h}$ creates asymmetric boundary requirements.
- Symmetric difference: $u'_k \approx \frac{u_{k+1} - u_{k-1}}{2h}$ suffers from the **fencepost problem**.
- Consequences:
  - uneven treatment of grid points,
  - non-differentiable numerical solutions,
  - failure to converge despite smooth u.



u_k with odd or even k form smooth solutions, but not together!

Representing Derivative Operators

Finite Differences: Staggered Grids

- Derivatives placed at half-grid points:

$$u'_{k+1/2} \approx \frac{u_{k+1} - u_k}{h}.$$

- Eliminates asymmetry and fencepost issues.
- Widely used in physical simulations:
  - pressures at cell centers,
  - velocities on edges,
  - fluxes on faces.
- Improves stability and accuracy while preserving structure.

Representing Derivative Operators

Collocation: Linking Continuous and Discrete Models

One may approximate a continuous function by expanding it in a basis:

$$u(\mathbf{x}) \approx \sum_{i=1}^{k} a_i \, \phi_i(\mathbf{x}).$$

- Coefficients a_i provide the best approximation of u within the chosen basis.
- As more basis functions are added, the approximation may converge to the true solution.
- This viewpoint generalizes interpolation, quadrature, and numerical differentiation.

Representing Derivative Operators

Collocation: Enforcing the PDE at Sample Points

To solve a PDE such as $\nabla^2 u = w$:

- Choose collocation points $\mathbf{x}_1, \ldots, \mathbf{x}_k$ in the domain.
- Require the PDE to hold *exactly* at these points:

$$w(\mathbf{x}_i) = \nabla^2 u(\mathbf{x}_i) = \sum_{j=1}^{k} a_j \, \nabla^2 \phi_j(\mathbf{x}_i).$$

- Unknowns are the coefficient vector $a \in \mathbb{R}^k$.
- Leads to a square linear system; can be overdetermined for least-squares variants.

Representing Derivative Operators

Collocation: Basis Choices and Practical Considerations

- Basis functions may be global (polynomials, trigonometric functions) or local (compactly supported).
- Compactly supported bases yield sparse linear systems.
- No grid structure required; suitable for irregular domains.
- Limitation: the behavior between collocation points is not controlled. Poorly chosen points or bases can cause oscillations or degeneracy.
- Remedies:
  - optimize collocation points and bases,
  - use methods that incorporate integral information (e.g., finite elements).

Representing Derivative Operators

<u>Finite Elements</u>: Weak Formulation

- Finite element methods approximate $u(\mathbf{x})$ using basis functions but enforce the PDE in an *integrated* (weak) sense.

$$\langle v, \nabla^2 u \rangle := \int_\Omega v(\mathbf{x}) \nabla^2 u(\mathbf{x})\, dx.$$

- If u solves $\nabla^2 u = 0$ with Dirichlet data, then

$$\int_\Omega v\, \nabla^2 u\, dx = 0$$

  for all test functions v vanishing on $\partial\Omega$.

- Test functions probe the PDE without requiring second derivatives of u.

Representing Derivative Operators

Finite Elements: Integration by Parts

Using the divergence theorem and $v|_{\partial\Omega} = 0$: (along with Integration by parts)

$$\langle v, \nabla^2 u \rangle = -\int_\Omega \nabla v(\mathbf{x}) \cdot \nabla u(\mathbf{x})\, dx.$$

- Reduces the PDE to first-order derivatives.
- Defines the bilinear form

$$(u, v)_{\nabla^2} = -\int_\Omega \nabla u \cdot \nabla v\, dx.$$

- The weak solution satisfies $(u, v)_{\nabla^2} = 0$ for all test functions v.

Representing Derivative Operators

Finite Elements: Galerkin Approximation

- Approximate u and test functions using the same basis $\{\phi_1, \ldots, \phi_k\}$:

$$u(\mathbf{x}) \approx \sum_{j=1}^{k} a_j \phi_j(\mathbf{x}).$$

- Require $(u, \phi_i)_{\nabla^2} = 0$ for all i. *(choosing v to traverse all ϕ_i's will be sufficient)*

- Leads to the linear system:

$$\sum_{j=1}^{k} a_j (\phi_j, \phi_i)_{\nabla^2} = 0.$$

- Matrix form: $Ka = 0$, where K is the stiffness matrix.

- Boundary values imposed by replacing the appropriate rows.

Representing Derivative Operators

Finite Elements: Poisson Equation with Forcing

For $\nabla^2 u = w$, expand both:

$$u(\mathbf{x}) = \sum_i a_i \phi_i, \qquad w(\mathbf{x}) = \sum_i b_i \phi_i.$$

- Weak form becomes

$$(\phi_i, \phi_j)_{\nabla^2} a_j = \langle \phi_i, w \rangle.$$

- Linear system:

$$Ka = Mb,$$

  where K is the stiffness matrix and M is the mass matrix.

Representing Derivative Operators

Finite Elements: Common Basis Choices

- **Piecewise-linear hat functions** on a triangulation.
  - Local support; stiffness matrix is sparse.
- **Spectral bases** (sines, cosines).
  - Orthogonality simplifies mass and stiffness matrices.
  - FFT accelerates computation.
- **Adaptive FEM**.
  - Refines basis where the approximation error is large.

Representing Derivative Operators

Piecewise Linear FEM: Basis Construction

Consider Poisson's equation $u''(x) = w(x)$ on $[0, 1]$ with Dirichlet conditions.
Use the reference hat function

$$\phi(x) = \begin{cases} 1 + x, & x \in [-1, 0], \\ 1 - x, & x \in [0, 1], \\ 0, & \text{otherwise.} \end{cases}$$

- Define $\phi_i(x) = \phi(kx - i)$ for $i = 0, \ldots, k$.
- Compact support leads to sparse matrices.

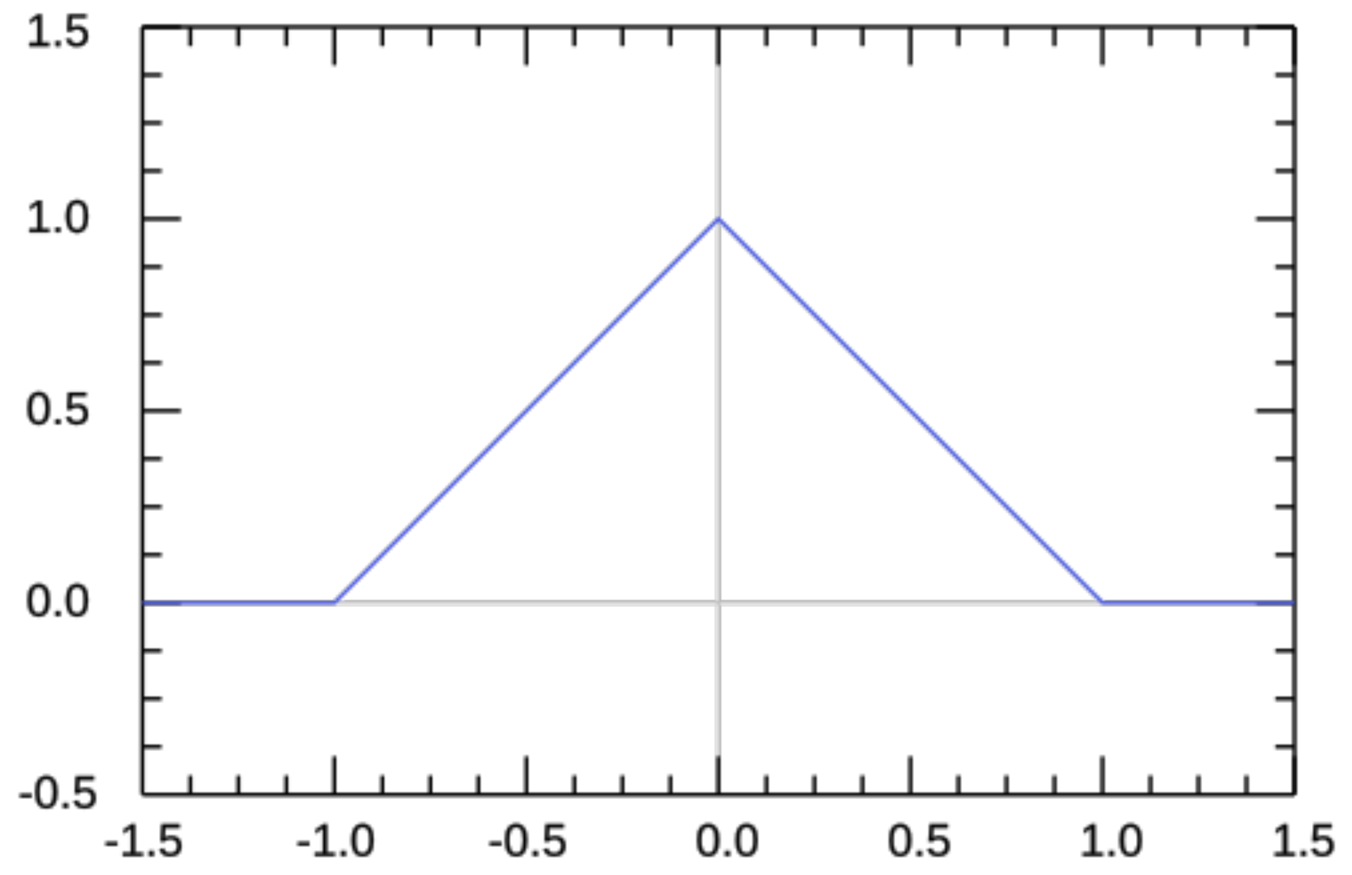


Image source: https://en.wikipedia.org/wiki/Triangular_function

Representing Derivative Operators

Piecewise Linear FEM: Mass and Stiffness Matrices

Precomputed integrals give:

$$\int_{-1}^{1} \phi(x)^2 \, dx = \int_{-1}^{0} (1+x)^2 \, dx + \int_{0}^{1} (1-x)^2 \, dx = \frac{2}{3}$$

$$\int_{-1}^{1} \phi(x)\phi(x-1) \, dx = \int_{0}^{1} x(1-x) \, dx = \frac{1}{6}.$$

$$\langle \phi_i, \phi_j \rangle = \frac{1}{6k} \begin{cases} 4, & i = j, \\ 1, & |i-j| = 1, \\ 0, & \text{otherwise}, \end{cases}$$

$$\langle \phi_i, \phi_j \rangle_{\nabla^2} = k \begin{cases} -2, & i = j, \\ 1, & |i-j| = 1, \\ 0, & \text{otherwise}. \end{cases}$$

- These coincide (up to scaling) with classical divided-difference matrices.

Representing Derivative Operators

Piecewise Linear FEM: Final Discretized System

Let $b_i = w(i/k)$.

$$k \begin{pmatrix} 1/k & & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & & 1/k \end{pmatrix} a = \frac{1}{6k} \begin{pmatrix} 6k & & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & & 6k \end{pmatrix} \begin{pmatrix} c \\ b_1 \\ \vdots \\ b_{k-1} \\ d \end{pmatrix}.$$

- End rows enforce boundary data. (1st row: $a_0 = c$, last row: $a_k = d$)
- Interior rows come from the FEM weak formulation.

Representing Derivative Operators

Piecewise Linear FEM: Visualization of Solution

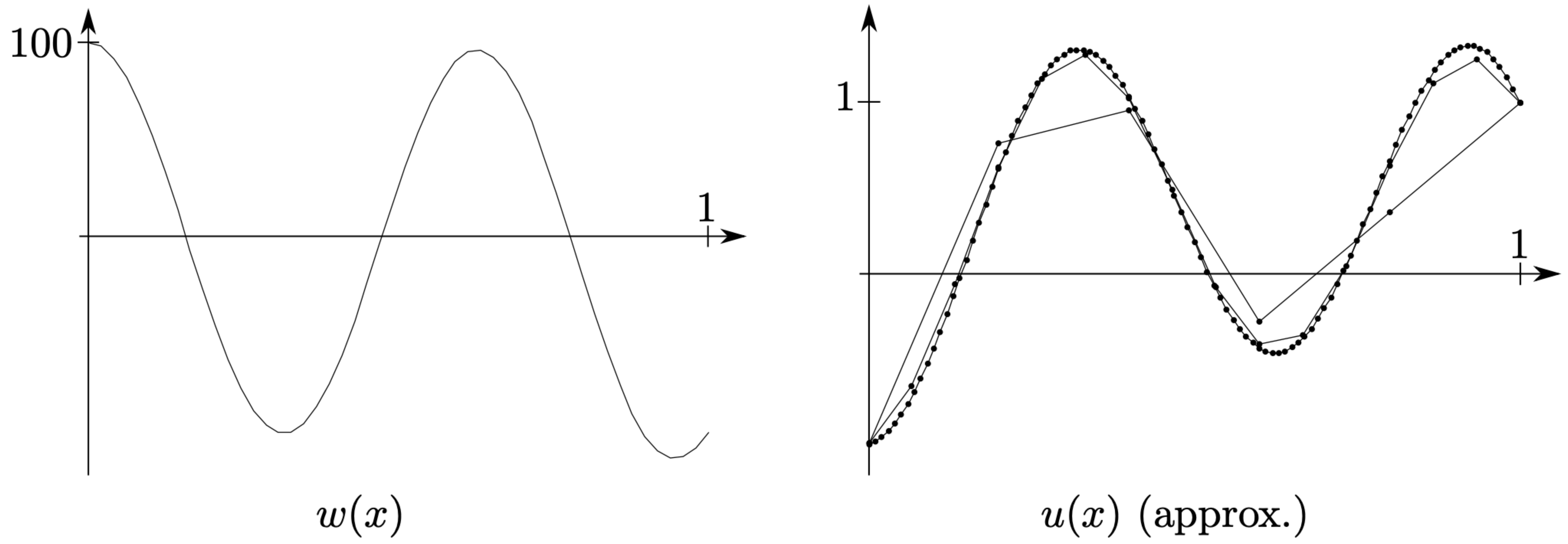


Figure 16.17 Approximated piecewise linear solutions of $u''(x) = w(x)$ computed using finite elements as derived in Example 16.9; in these examples, we take $c = -1$, $d = 1$, and $k \in \{5, 15, 100\}$.

Representing Derivative Operators

Finite Volumes: Core Idea

- The finite volume method sits between finite elements and collocation.
- Instead of enforcing the PDE at points, the PDE is enforced *on average* over cells.
- Used widely in fluid dynamics, conservation laws, and problems requiring strict flux balance.
- Key principle: integrate the PDE over each cell of a partition of Ω.

Representing Derivative Operators
Finite Volumes: Divergence Theorem

- Let $\Gamma \subseteq \Omega$ and consider the vector field $\mathbf{v}(x)$.
- The divergence theorem states:

$$\int_{\Gamma} \nabla \cdot \mathbf{v}(x)\, dx = \int_{\partial\Gamma} \mathbf{v}(x) \cdot \mathbf{n}(x)\, dx.$$

- $\mathbf{n}$ is the outward-facing unit normal on $\partial\Gamma$.
- Interpretation: total divergence inside Γ equals net flux through its boundary.

Representing Derivative Operators

Finite Volumes: Applying to Poisson's Equation

- Solve $\nabla^2 u = w$ in Ω.
- Integrate over Γ:

$$\int_\Gamma w(x)\,dx = \int_\Gamma \nabla^2 u(x)\,dx = \int_{\partial\Gamma} \nabla u(x) \cdot \mathbf{n}(x)\,dx.$$

- This characterizes Poisson solutions *in average form*.

Representing Derivative Operators

Finite Volumes: Discretizing the Domain

- Partition $\Omega = \bigcup_{i=1}^{k} \Omega_i$.
- Approximate $u(x)$ using a basis:

$$u(x) \approx \sum_{j=1}^{k} a_j \, \phi_j(x).$$

- Integrate over each cell:

$$\int_{\Omega_i} w(x)\, dx = \sum_{j=1}^{k} a_j \left[\int_{\partial\Omega_i} \nabla\phi_j(x) \cdot \mathbf{n}(x)\, dx \right].$$

- Produces a linear system for coefficients a_j.

Representing Derivative Operators

Finite Volumes: Practical Choices

- Common choice: piecewise-linear “hat” basis functions ϕ_j.
- Typical cells: Voronoi regions around mesh nodes.
- Flux terms across each face computed using appropriate numerical quadrature.
- Method naturally preserves conservation laws and flux balances.

Representing Derivative Operators

Summary

	Approximation	Enforcement of PDE
Finite Differences	Derivatives (w/ divided differences)	Sample points
Collocation	The function (w/ basis functions)	Sample points
Finite Elements	The function (w/ basis functions)	On average over test functions
Finite Volumes	The function (w/ basis functions)	On average over cells

Representing Derivative Operators

Examples of Other PDE Discretization Methods

- Many additional discretization techniques exist beyond finite differences, FEM, and FVM.
- **Domain decomposition:** Solve PDE on subregions of Ω iteratively.
  - Enables parallel computation.
  - Useful for preconditioning iterative solvers.
- **Mesh-free methods:** Represent physical fields using particles, not grids.
  - Example: smoothed particle hydrodynamics (SPH).
  - Useful when refinement is needed in localized regions.
- **Boundary element and analytic element methods:** Use basis functions on $\partial\Omega$ to reduce dimensionality of the problem.

Table of Content

Solving Parabolic and Hyperbolic Equations

Time as a Variable

- Parabolic and hyperbolic PDEs introduce a time variable t in addition to spatial variables.
- Time and space often play different physical roles, so we may discretize them differently.
- Two main strategies:
  - **Semidiscrete methods (method of lines):** discretize space, keep t continuous.
  - **Fully discrete methods:** discretize both space and time simultaneously.

Solving Parabolic and Hyperbolic Equations

Heat Equation Example

- Consider the 1D heat equation on $x \in [0, 1]$:

$$u_t = u_{xx}, \quad u(t; 0) = u(t; 1) = 0, \quad u(0; x) = u_0(x).$$

- Discretize x using grid points $x_i = ih$, $h = 1/n$.
  - Let $u_i(t)$ denote the temperature at position x_i.
  - Use the finite difference Laplacian L from earlier.
- Semidiscrete system:

$$h^2 \mathbf{u}'(t) = L\,\mathbf{u}(t), \quad \mathbf{u}(t) = (u_0(t), \ldots, u_n(t))^\top.$$

- This is an ODE in time that can be solved with time stepping methods.

Solving Parabolic and Hyperbolic Equations

Implicit Time Stepping

- Write the semidiscrete system as: $\mathbf{u}'(t) = A\,\mathbf{u}(t), \quad A = \frac{1}{h^2}L$.
- Backward Euler with time step Δt:

$$\mathbf{u}^{k+1} \approx (I - \Delta t\, A)^{-1}\,\mathbf{u}^k = \left(I - \frac{\Delta t}{h^2}L\right)^{-1}\mathbf{u}^k.$$

- For diffusive problems like heat flow:
  - A is typically negative definite.
  - Implicit Euler is unconditionally stable.
- Hyperbolic PDEs may need more careful integrators to avoid overdamping.

Solving Parabolic and Hyperbolic Equations

Eigendecomposition Viewpoint

- Suppose A is time-independent with eigenpairs $(\lambda_i, \mathbf{v}_i)$.
- Expand the initial condition:

$$\mathbf{u}(0) = \sum_i c_i \mathbf{v}_i.$$

- Solution of $\mathbf{u}' = A\mathbf{u}$:

$$\mathbf{u}(t) = \sum_i c_i e^{\lambda_i t} \mathbf{v}_i.$$

- For semidiscrete PDEs, eigenvalues and eigenvectors often have physical meaning.
  - For the Laplacian, λ_i correspond to resonant frequencies of the domain.
  - Low-frequency modes yield compact closed-form approximations after truncation.

Table of Content

Numerical Considerations

Consistency, Convergence, Stability

- Key properties when choosing a PDE discretization:
  - **Convergence:** discrete solutions approach the true PDE solution as mesh spacing $\to 0$.
  - **Consistency:** discrete derivatives approximate the PDE derivatives as spacing $\to 0$.
  - **Stability:** errors do not grow uncontrollably under the numerical scheme.
- For finite difference schemes:
  - Lax–Richtmyer theorem: for a well-posed linear problem, consistency + stability $\iff$ convergence.
  - Stability often checked using Taylor expansions or von Neumann analysis.
  - CFL condition relates Δt to spatial grid size for hyperbolic/advection-dominated PDEs.

Numerical Considerations

Convergence in Various Methods

- Advection-dominated problems can develop fronts and shocks.
    - Upwinding schemes modify stencils to transport features in the correct direction.
- Finite element convergence:
    - Depends on choice of basis functions.
    - Basis must be rich enough to approximate the theoretical solution as mesh is refined.
- Without convergence, numerical PDE solutions cannot be trusted; production codes often test for degenerate or unstable behavior automatically.

Numerical Considerations

Linear Solvers for PDE: Structure of Discrete Systems

- PDE discretizations yield large linear systems with special structure:
  - matrices are sparse and often symmetric,
  - many come from elliptic or parabolic operators and are (semi)definite.
- Elliptic PDEs (e.g., Poisson) $\Rightarrow$ positive definite systems after appropriate boundary conditions.
- Parabolic PDEs lead to well-posed semidiscrete systems involving such matrices.
- Efficient solution techniques:
  - Cholesky factorization for small/medium problems,
  - conjugate gradients and related Krylov methods for large sparse systems.

Table of Content