# Boundary-value problems

*Phys 750 Lecture 8*

# Boundary-value problems

- Important class of problems in physics: differential equations with solutions having specified conditions at the boundaries
- For example,
  - Electrostatic potentials
  - Normal modes in wave problems
  - Heat flow

$$\forall\, x \in \mathcal{R}:$$

$$u''(x) = F(u(x), u'(x); x)$$

$$\forall\, x \in \partial\mathcal{R}:$$

$$u(x) = \alpha(x) \text{ or}$$

$$u'(x) = \beta(x)$$

Dirichlet

Neumann

# Boundary-value problems

- Electric potential produced by a distribution of static charges is described by the Poisson equation:

$$\nabla^2\phi = \left(\frac{\partial^2}{\partial x} + \frac{\partial^2}{\partial y} + \frac{\partial^2}{\partial z}\right)\phi(x, y, z) = \rho(x, y, z)$$

- Or, in free space, by the Laplace equation:

$$\nabla^2\phi = \left(\frac{\partial^2}{\partial x} + \frac{\partial^2}{\partial y} + \frac{\partial^2}{\partial z}\right)\phi(x, y, z) = 0$$

- Must be augmented by specific values of the potential and electric field ( $\phi$ and $\vec{E} = -\vec{\nabla}\phi$) at the boundaries

# Boundary-value problems

- Boundary-value ODEs also arise if we solve for the normal modes of time-dependent partial-differential equations (PDEs)

$$\frac{\partial u}{\partial t} = u_{xx}(x,t) - F(u(x,t), u_x(x,t); x)$$

- Connected by a Fourier transform in the time coordinate:

$$u(x,t) = \int d\omega\, u_\omega(x) e^{i\omega t}$$

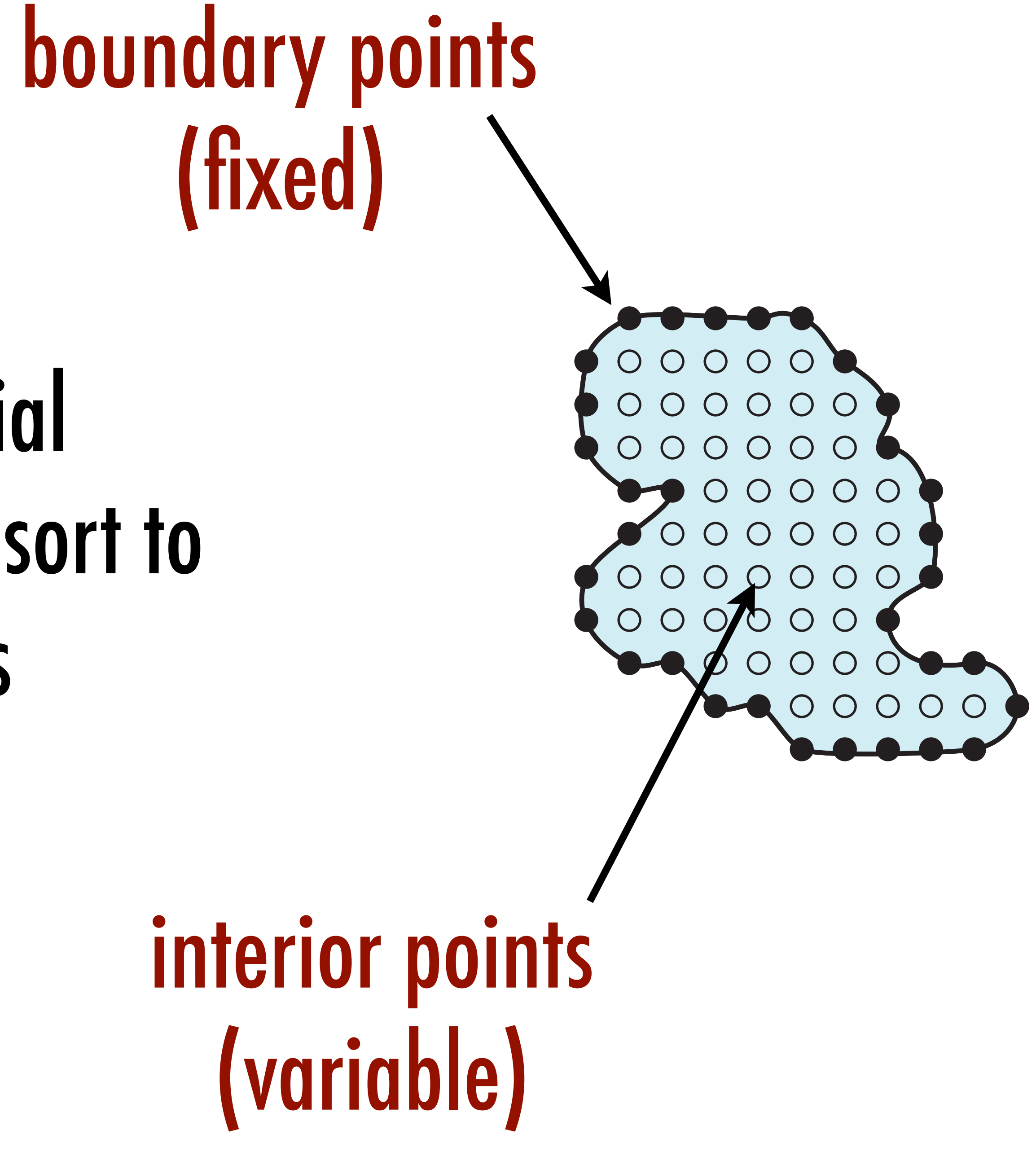
$$u''_\omega(x) = F(u_\omega(x), u'_\omega(x); x) - i\omega u_\omega$$

# Boundary-value problems

- Familiar analytical approach is to expand the solution using special functions: (sinusoidal or Bessel functions, cylindrical or spherical harmonics)
- The goal of such spectral methods is to decompose the solution in a complete set of functions that automatically satisfy the given boundary conditions
- Only convenient in situations with high symmetry (e.g., sphere, cylinder, or box)

# Discretization

- For regions with no special symmetry, we have to resort to finite-difference methods



# Discretization

- Generalize spatial derivatives to multiple dimensions

| | | |
|---|---|---|
| | | |
| 1 | -2 | 1 |
| | | |

+

| | | |
|---|---|---|
| | 1 | |
| | -2 | |
| | 1 | |

=

| | | |
|---|---|---|
| | 1 | |
| 1 | -4 | 1 |
| | 1 | |

$(\Delta x = \Delta y)$

$$u_{xx} \approx \frac{1}{(\Delta x)^2}\left[u(x_{i-1}) - 2u(x_i) + u(x_{i+1})\right]$$

$$u_{xx} + u_{yy} \approx \frac{1}{(\Delta x)^2}\left[u(x_{i-1}, y_j) + u(x_{i+1}, y_j) + u(x_i, y_{j-1}) + u(x_i, y_{j+1}) - 4u(x_i)\right]$$

$$\nabla^2 u \approx \frac{1}{(\Delta x)^2}\left(\sum_{\vec{\eta}} u(\vec{r} + \vec{\eta}) - N_{\mathrm{nn}} u(\vec{r})\right)$$

Nearest neighbour count

(orthogonal mesh)

# Discretization

- Discretized ODEs are linear; equivalent to a linear system of equations $M_{\alpha,\beta} U_\beta = A_\beta$
- Unified index: $U_\alpha = u(x_i, y_j, z_k) \equiv u_{i,j,k}$

$$\alpha(i,j,k) = i + Lj + L^2 k \quad (L \times L \times L \text{ box})$$

- Solution possible via matrix inversion $U = M^{-1} A$
- Method scales badly: vector size $\sim 1/(\Delta x)^3 \sim L^3$, matrix storage $\sim L^6$, matrix inversion complexity $\sim L^9$

# Relaxation methods

**Given**

$$u_{xx} = \rho(x)$$

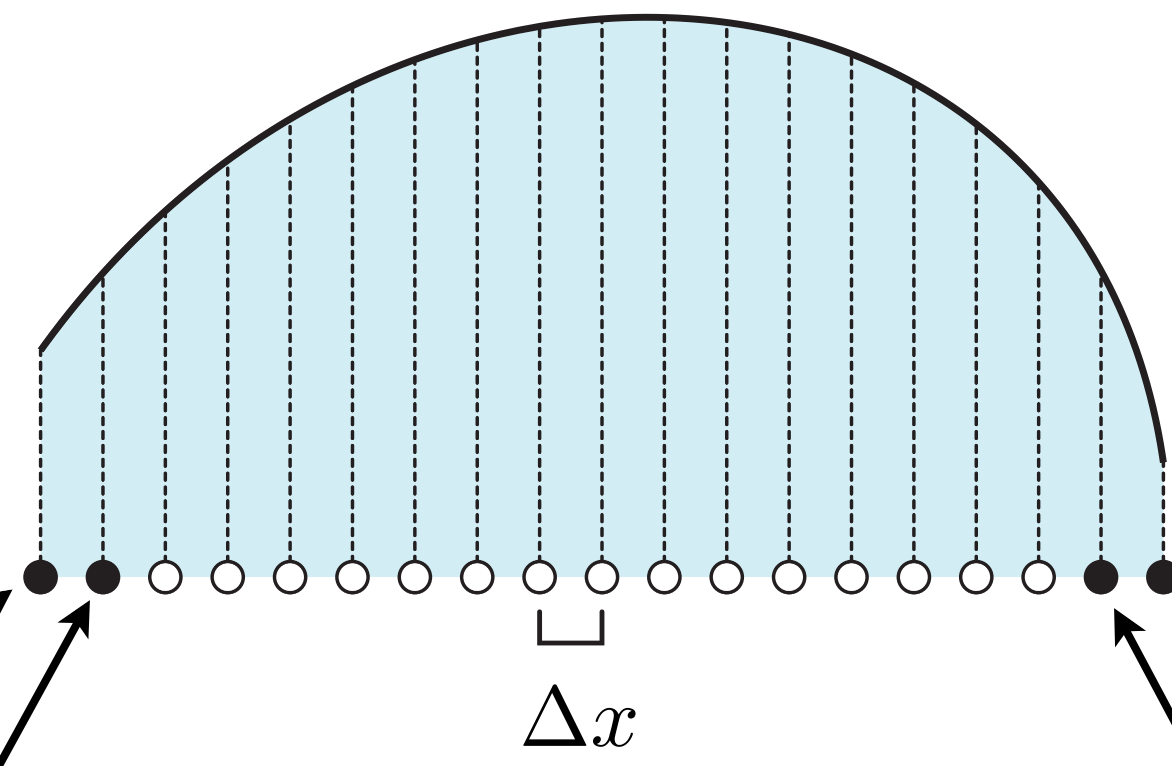
**Fix two of**

$$u(a), u'(a)$$

$$u(b), u'(b)$$

**Discrete mesh of points:**

$$u_i = u(x_i) = u(a + i(b-a)/L)$$




$$u_L = u(b)$$

$$u_0 = u(a)$$

$$u_1 = u_0 + u'(a)\Delta x$$

$$u_{L-1} = u_L - u'(b)\Delta x$$

**Converge via iteration?:**

$$u_i = \frac{1}{2}\left(u_{i+1} + u_{i-1} + (\Delta x)^2 \rho_i\right)$$

# Relaxation methods

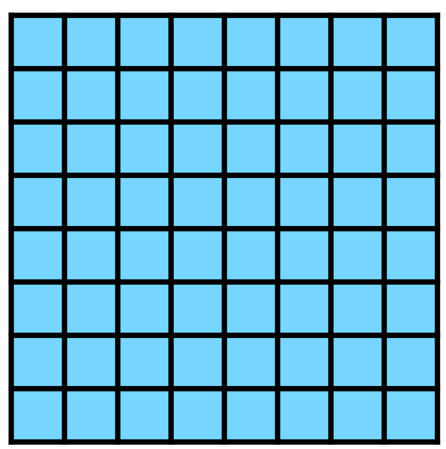
- Jacobi method algorithm:

- Set the fixed $u_i$ along the boundaries
- Loop though all interior points $x_i$
  - Set $u_i^{\text{new}} = \frac{1}{2}(u_{i+1}^{\text{old}} + u_{i-1}^{\text{old}} + (\Delta x)^2 \rho_i)$
  - Keep track of largest $\Delta u = |u_i^{\text{new}} - u_i^{\text{old}}|$
- Repeat until $\Delta u < \epsilon$

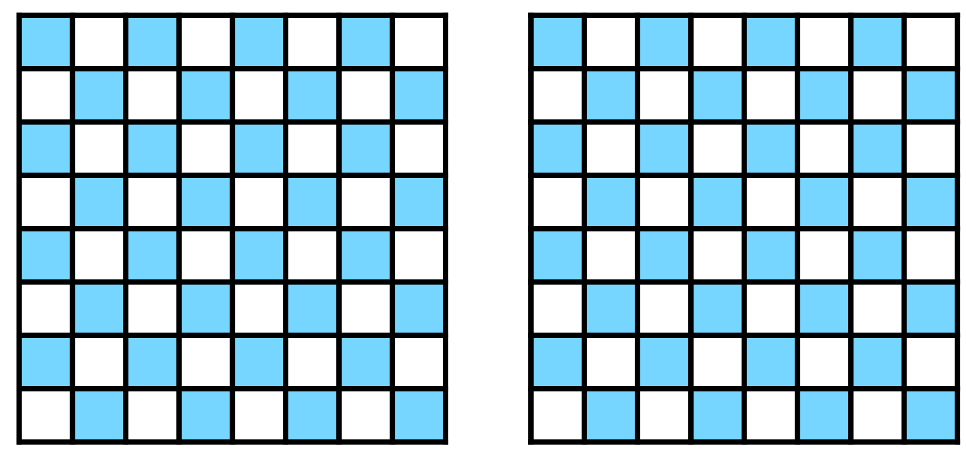
# Relaxation methods

- Various update orderings (with different convergence properties!) are possible

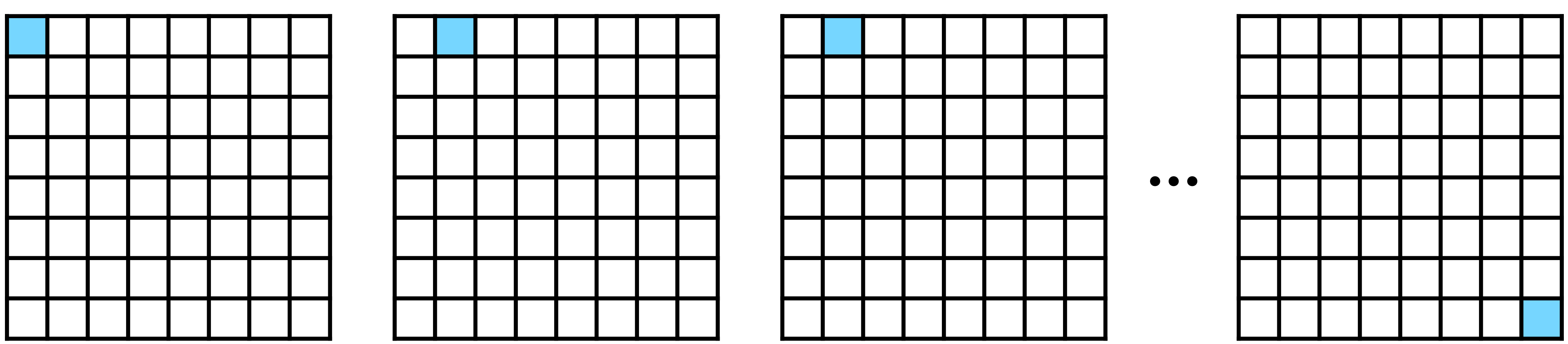
Jacobi



Checker-board



Gauss-Seidel



# Relaxation methods

- Slow convergence since it takes many steps for changes to propagate across the grid
- Better resolution ($\Delta x \to 0$) means that the length scale for propagation increases ($L = 1/\Delta x \to \infty$)
- Might try to reweight so that new values incorporate more of the changes from neighbouring points:

$$u^{\mathrm{new}} = \alpha \bar{u} + (1 - \alpha) u$$

$$\bar{u}(\vec{r}) = \frac{1}{N_{\mathrm{nn}}} \sum_{\eta} u(\vec{r} + \vec{\eta})$$

$\alpha = 1$ Jacobi

$0 < \alpha < 1$ Underrelaxation

$1 < \alpha < 2$ Overrelaxation

# Relaxation methods

- Overrelaxation can be connected to the corresponding time-dependent diffusion problem

$$u_t = u_{xx} + u_{yy} - F(u_x, u_y, u)$$

- Recover the original problem when $\lim_{t\to\infty} u_t = 0$

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} + (\Delta t)\left(\frac{u_{i+1,j}^{(n)} + u_{i,j+1}^{(n)} - 4u_{i,j}^{(n)} + u_{i-1,j}^{(n)} + u_{i,j1}^{(n)}}{(\Delta x)^2} - \rho_{i,j}\right)$$

- Introduce a fictitious time step; over-relaxation parameter connected to the choice of $\Delta t/(\Delta x)^2$

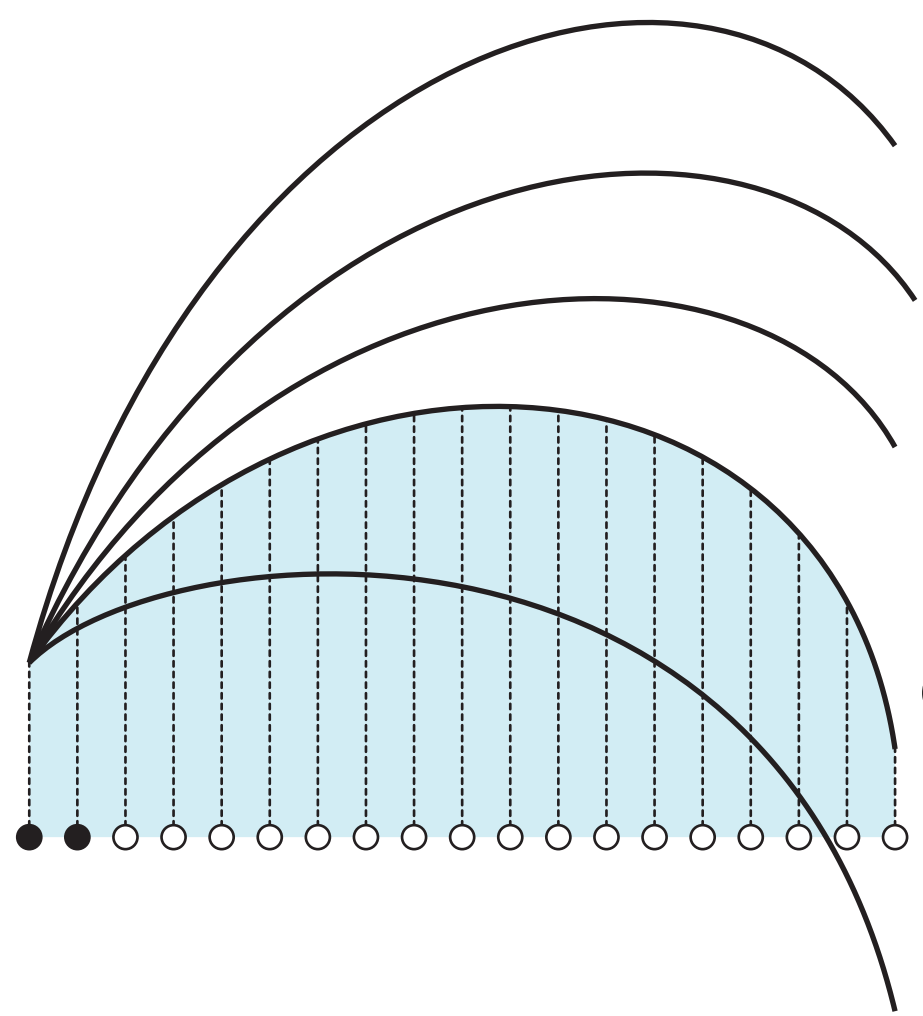
# Shooting method

- A shooting strategy involves converting the boundary-value problem to a related initial value problem

$$u'' = F(u, u', x) \quad u(a), u(b) \quad \longrightarrow \quad u'' = F(u, u', x) \quad u(a) \qquad u(b)$$

- Forward integrate assuming a derivative $g = u'(a)$
- Yields a 1-parameter family of solutions $u(x; g)$
- Unique solution to the boundary-value problem corresponds to the root of $G(g) = u(b) - u(b; g)$

# Shooting method




- Apply root finding algorithms to $G$

$$u_0 = u(a)$$

$$u_1 = u_0 + g\Delta x$$