CS601: Software Development for Scientific Computing

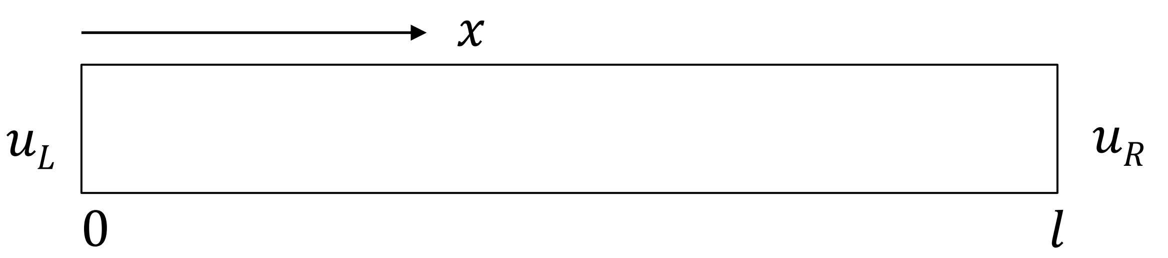
Autumn 2024

Week12: Structured Grids

Recap

Application: Heat Equation

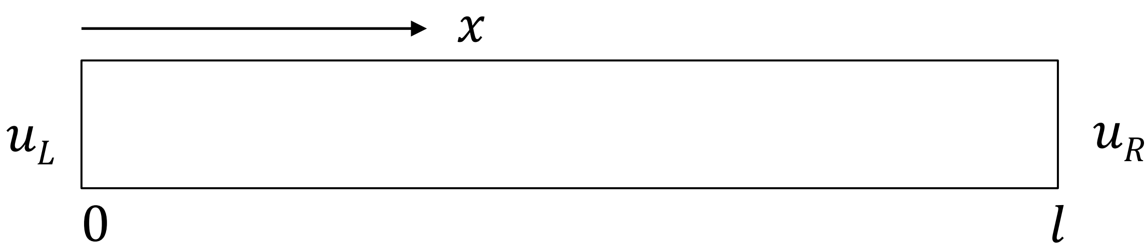
- Example: heat conduction through a rod



- $u = u(x, t)$ is the temperature of the metal bar at distance x from one end and at time t
- Goal: find u

Initial and Boundary Conditions

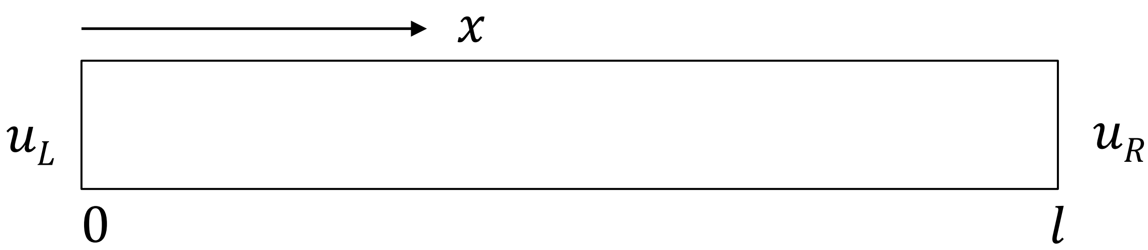
- Example: heat conduction through a rod



- Metal bar has length l and the ends are held at constant temperatures u_L at the left and u_R at the right
- Temperature distribution at the initial time is known $f(x)$, with $f(0) = u_L$ and $f(l) = u_R$

Equations

- Example: heat conduction through a rod



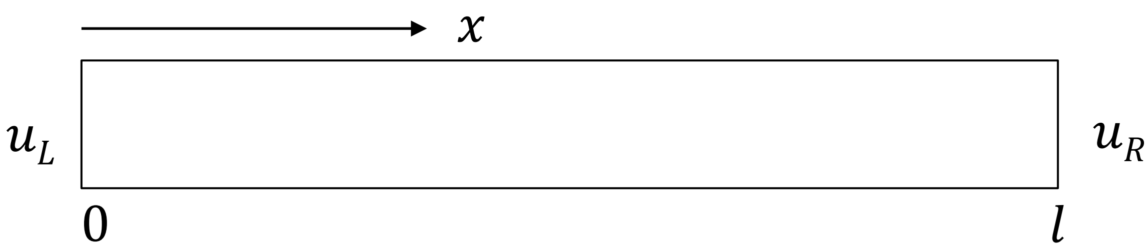
$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \qquad (0 < x < l, t > 0)$$

α is thermal diffusivity

(a constant if the material is homogeneous and isotropic.
copper = 1.14 cm^2 s^{-1}, aluminium = 0.86 cm^2 s^{-1})

Equations

- Example: heat conduction through a rod



$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \qquad (0 < x < l, t > 0)$$

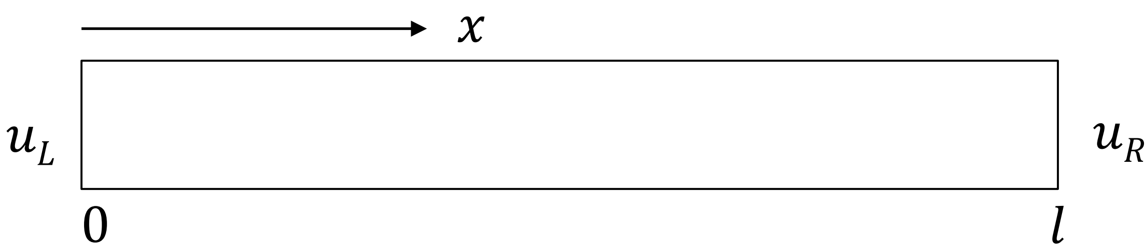
α is thermal diffusivity

(a constant if the material is homogeneous and isotropic.
copper = 1.14 cm^2 s^{-1}, aluminium = 0.86 cm^2 s^{-1})

- *Exercise: what kind of a PDE is this? (Poisson/Heat/Wave?)*

Equations

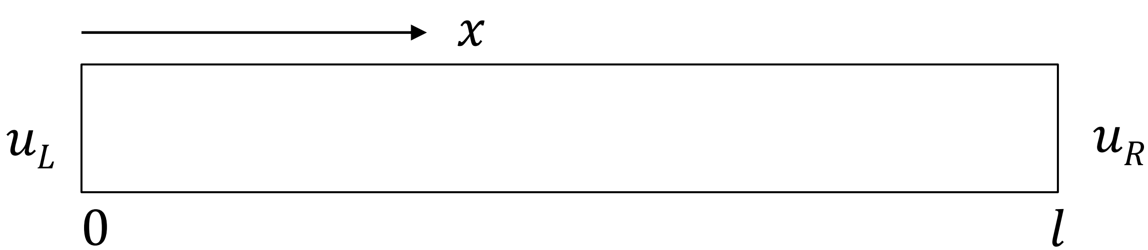
- Example: heat conduction through a rod



$\partial_t u = \alpha \Delta u$ as per the notation mentioned earlier

Equations

- Example: heat conduction through a rod



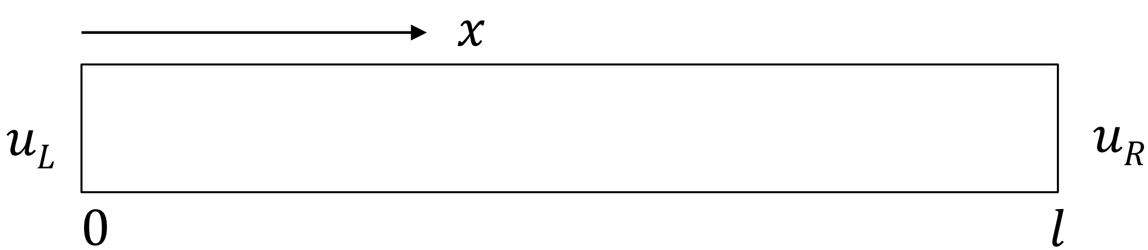
$$\partial_t u = \alpha \Delta u$$

Can also be written as:

$$\partial_t u - \alpha \Delta u = 0$$

Equations

- Example: heat conduction through a rod



$\partial_t u - \alpha \Delta u = 0$,

Based on initial and boundary conditions:

$u(0, t) = u_L$,
$u(l, t) = u_R$,
$u(x, 0) = f(x)$

Equations

- Summarizing:
  1. $\partial_t u - \alpha \Delta u = 0, 0<\mathrm{x}<\mathrm{l}, \mathrm{t}>0$
  2. $u(0,t) = u_L, t > 0$
  3. $u(l,t) = u_R\ , t > 0$
  4. $u(x,0) = \ f(x), 0 < x < l$

- Solution:

$$u(x,t) \quad = \sum_{m=1}^{\infty} B_m e^{-m^2 \alpha \pi^2 t / l^2} \sin(\frac{m\pi x}{l}) \quad ,$$

where, $B_m = 2/l \int_0^l f(s) \sin\left(\frac{m\pi s}{l}\right) ds$

Equations

- Summarizing:
  1. $\partial_t u - \alpha \Delta u = 0, 0<x<l, t>0$
  2. $u(0,t) = u_L, t > 0$
  3. $u(l,t) = u_R\,, t > 0$
  4. *But we are interested in a numerical solution*

- Solution:

$$u(x,t) \quad = \sum_{m=1}^{\infty} B_m e^{-m^2 \alpha \pi^2 t / l^2} \sin(\frac{m\pi x}{l}) \quad ,$$

where, $B_m = 2/l \int_0^l f(s) \sin\left(\frac{m\pi s}{l}\right) ds$

Approximating Partial Derivatives

- Suppose $y = f(x)$
  - Forward difference approximation to the first-order derivative of f w.r.t. x is:

$$\frac{df}{dx} \approx \frac{\left(f(x+\delta x)-f(x)\right)}{\delta x}$$

  - Central difference approximation to the first-order derivative of f w.r.t. x is:

$$\frac{df}{dx} \approx \frac{\left(f(x+\delta x)-f(x-\delta x)\right)}{2\delta x}$$

  - Central difference approximation to the second-order derivative of f w.r.t. x is:

$$\frac{d^2 f}{dx^2} \approx \frac{\left(f(x+\delta x)-2f(x)+f(x-\delta x)\right)}{(\delta x)^2}$$

Approximating Partial Derivatives

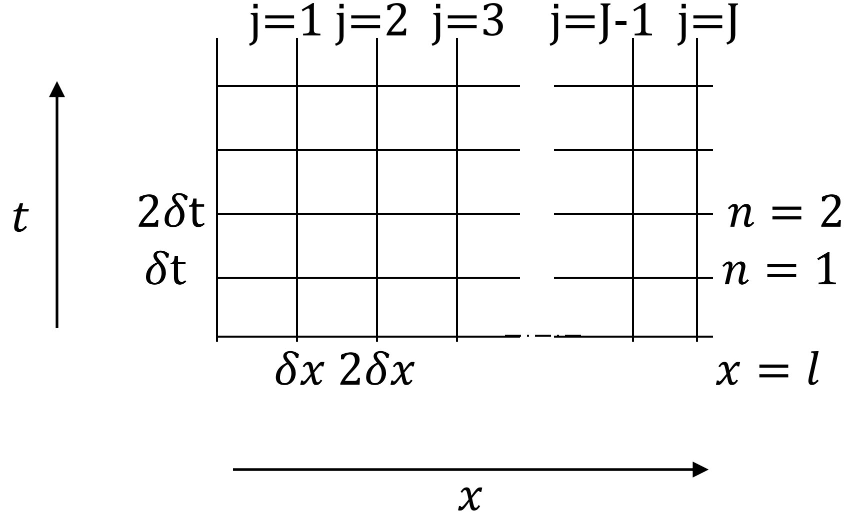
- In example heat application $f = u = u(x, t)$ and $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$
  - First, approximating $\frac{\partial u}{\partial t}$:

    $\frac{\partial u}{\partial t} \approx \frac{\left(u(x,t+\delta t)-u(x,t)\right)}{\delta t}$, where δt is a small increment in time
  - Next, approximating $\frac{\partial^2 u}{\partial x^2}$:

    $\frac{\partial^2 u}{\partial x^2} \approx \frac{\left(u(x+\delta x,t)-2u(x,t)+u(x-\delta x,t)\right)}{(\delta x)^2}$, where δx is a small increment in space (along the length of the rod)

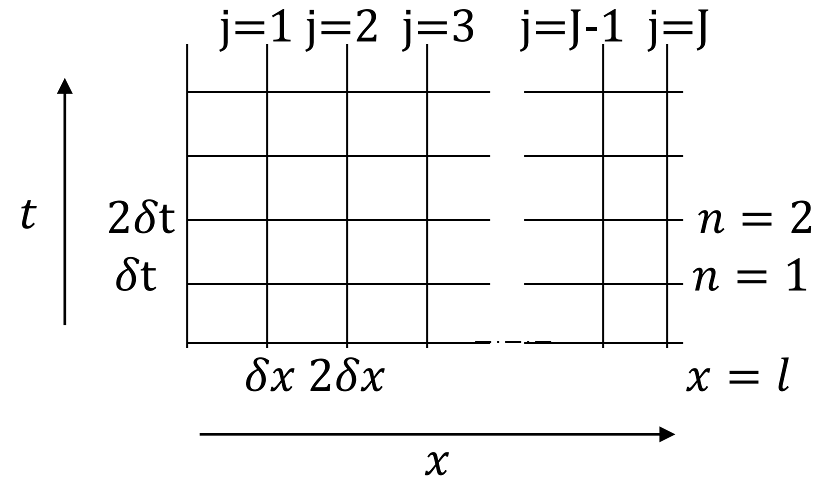
Approximating Partial Derivatives

- Divide length l into J equal divisions: $\delta x = l/J$ (space step)
- Choose an appropriate δt (time step)



Approximating Partial Derivatives

- Find sequence of numbers which approximate u at a sequence of (x, t) points (i.e. at the intersection of horizontal and vertical lines below)



- Approximate the exact solution $u(j \times \delta x, n \times \delta t)$ using the approximation for partial derivatives mentioned earlier

Approximating Partial Derivatives

$$\frac{\partial u}{\partial t} \approx \frac{\left(u(x, t+\delta t) - u(x,t)\right)}{\delta t}$$

$$= \frac{(u_j^{n+1} - u_j^n)}{\delta t}$$

where u_j^{n+1} denotes taking j steps along x direction and $n+1$ steps along t direction

Similarly, $\frac{\partial^2 u}{\partial x^2} \approx \frac{\left(u(x+\delta x, t) - 2u(x,t) + u(x-\delta x, t)\right)}{(\delta x)^2}$

$$= \frac{(u_{j+1}^n - 2\, u_j^n + u_{j-1}^n)}{(\delta x)^2}$$

Approximating Partial Derivatives

Plugging into $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$:

$$\frac{(u_j^{n+1} - u_j^n)}{\delta t} = \alpha \frac{(u_{j+1}^n - 2\, u_j^n + u_{j-1}^n)}{(\delta x)^2}$$

This is also called as difference equation because you are computing difference between successive values of a function involving discrete variables.

Approximating Partial Derivatives

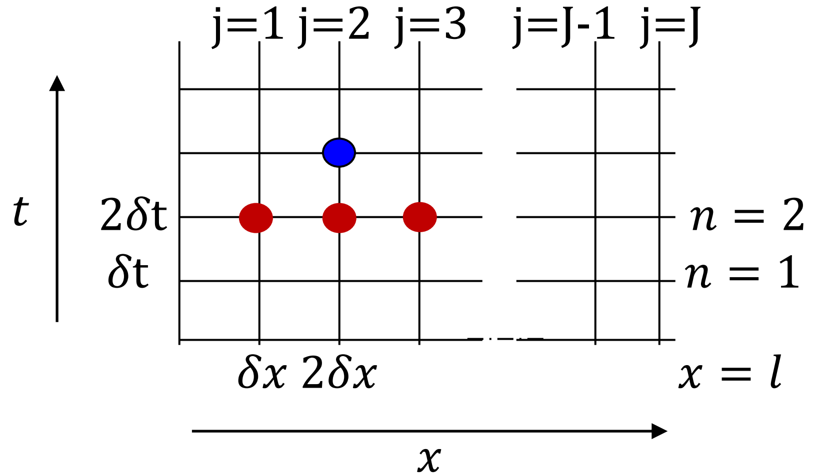
Simplifying:

$$u_j^{n+1} = u_j^n + r(u_{j+1}^n - 2\,u_j^n + u_{j-1}^n)$$
$$= ru_{j-1}^n + (1\ - 2r)u_j^n + ru_{j+1}^n,$$
$$where\ r = \alpha\,\frac{\delta t}{(\delta x)^2}$$

Approximating Partial Derivatives

visualizing,

$$u_j^{n+1} = ru_{j-1}^n + (1 - 2r)u_j^n + ru_{j+1}^n$$



To compute the value of function at blue dot, you need 3 values indicated by the red dots – 3-point stencil

Approximating Partial Derivatives

- Initial and boundary conditions tell us that:
  $u(0,t) = u_L$,
  $u(l,t) = u_R$,
  $u(x,0) = f(x)$

- $u_0^0, u_1^0\, u_2^0, \ldots.\, u_J^0$ are known (at time t=0, the temperature at all points along the distance is known as indicated by $f(x) = f_j$).
- u_0^1 is $u_{L,}\, u_J^1$ is u_R
- Now compute points on the grid from left-to-right:

Approximating Partial Derivatives

- Now compute points on the grid from left-to-right:

$$u_1^1 = u_1^0 + r(u_0^0 - 2u_1^0 + u_2^0)$$
$$u_2^1 = u_2^0 + r(u_1^0 - 2u_2^0 + u_3^0)$$

.

.

$$u_{J-1}^1 = u_{J-1}^0 + r\left(u_{J-2}^0 - 2u_{J-1}^0 + u_J^0\right)$$

- This constitutes the computation done in the first time step.
- Now do the second time step computation…and so on..

Explicit Difference Method: Stability

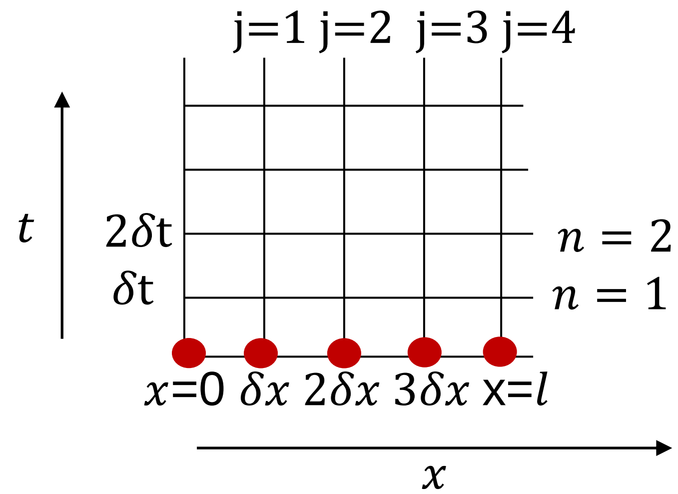
- Given: $l = 1,$
  $u(0, t) = u_L = 0,$
  $u(l, t) = u_R = 0,$
  $u(x, 0) = f(x) = x(l - x)$
  $\alpha = 1,$

- Choose: $\delta x = 0.25, \delta t = 0.075$
- Solve.

Explicit Difference Method: Stability

- Initialize u_j^0 values from initial and boundary conditions i.e. *get time-step 0 values*

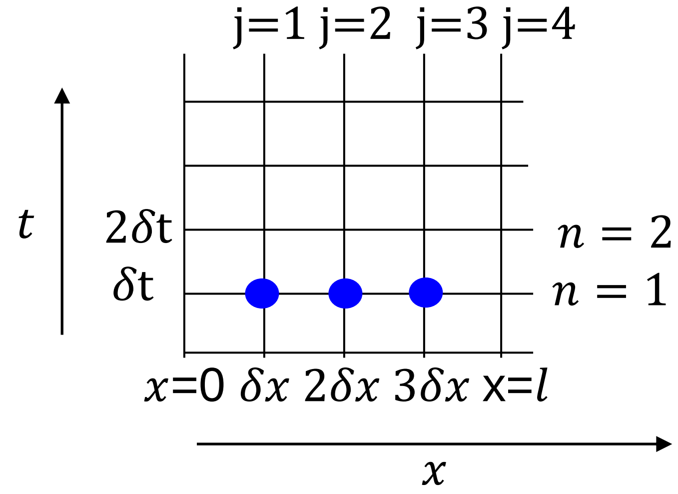
u_0^0 = 0
$u_1^0 = f(\delta x) = \delta x(l - \delta x)$ = .1875
$u_2^0 = f(2\delta x) = 2\delta x(l - 2\delta x)$ = .25
$u_3^0 = f(3\delta x) = 3\delta x(l - 3\delta x)$ = .1875
u_4^0 = 0



Explicit Difference Method: Stability

- Compute time-step 1 values

$$u_j^{n+1} = ru_{j-1}^n + (1 - 2r)u_j^n + ru_{j+1}^n$$

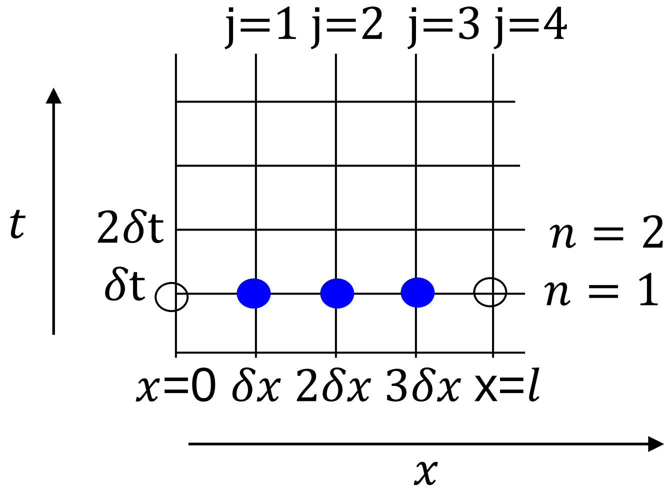


Explicit Difference Method: Stability

- Compute time-step 1 values

$$u_j^{n+1} = ru_{j-1}^n + (1 - 2r)u_j^n + ru_{j+1}^n$$

What about values of $u(x,t)$ at ○ ?



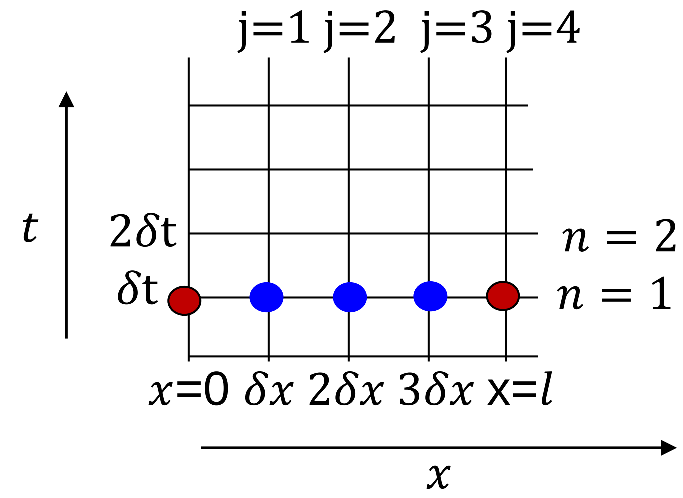
Explicit Difference Method: Stability

- Compute time-step 1 values

$$u_j^{n+1} = ru_{j-1}^n + (1 - 2r)u_j^n + ru_{j+1}^n$$

What about values of $u(x, t)$ at ○ ?

Get it from boundary conditions



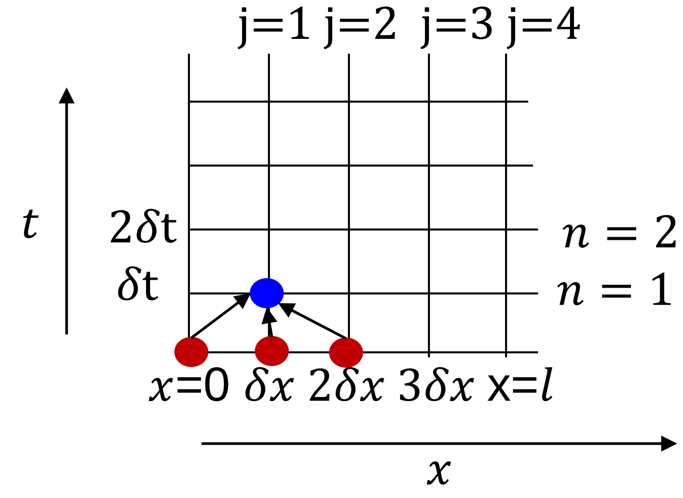
Explicit Difference Method: Stability

- Compute time-step 1 values

$$u_j^{n+1} = ru_{j-1}^n + (1 - 2r)u_j^n + ru_{j+1}^n$$

$r = \alpha\delta t/(\delta x)^2$ = 1.2

$u_1^1 = u_1^0 + r(u_0^0 - 2u_1^0 + u_2^0)$ = 0.03678



Explicit Difference Method: Stability

- Compute time-step 1 values

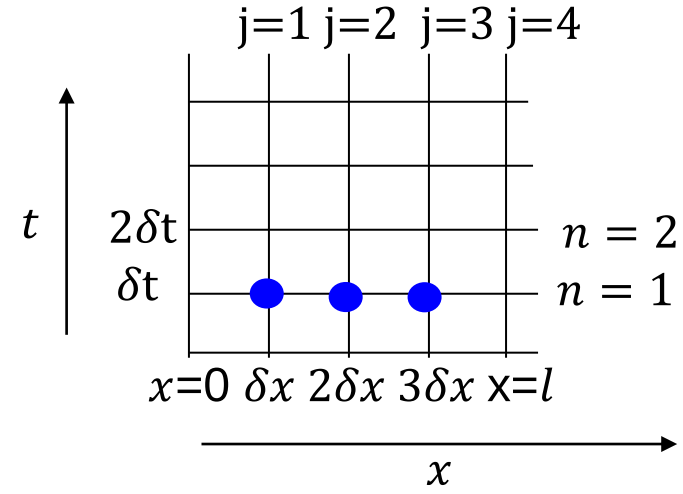
$$u_j^{n+1} = ru_{j-1}^n + (1 - 2r)u_j^n + ru_{j+1}^n$$

$r = \alpha\delta t/(\delta x)^2$ = 1.2

$u_1^1 = u_1^0 + r(u_0^0 - 2u_1^0 + u_2^0)$=0.03678

$u_2^1 = u_2^0 + r(u_1^0 - 2u_2^0 + u_3^0)$=0.1

$u_3^1 = u_3^0 + r(u_2^0 - 2u_3^0 + u_4^0)$=0.03678



Explicit Difference Method: Stability

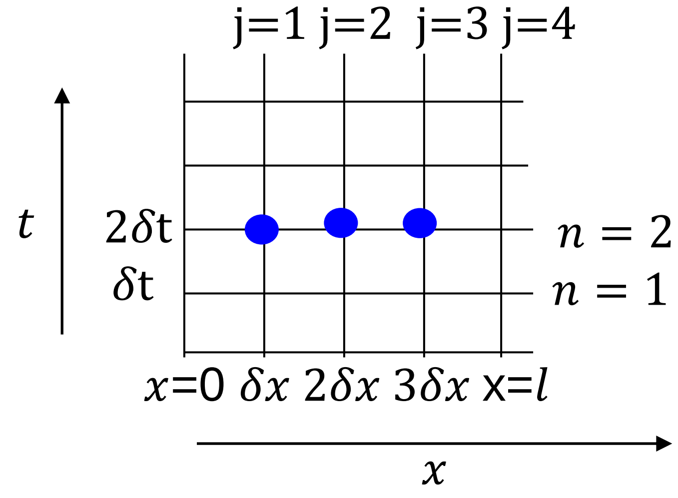
- Compute time-step 2 values

$$u_j^{n+1} = ru_{j-1}^n + (1 - 2r)u_j^n + ru_{j+1}^n$$

$u_1^2 = u_1^1 + r(u_0^1 - 2u_1^1 + u_2^1)$=0.06851

$u_2^2 = u_2^1 + r(u_1^1 - 2u_2^1 + u_3^1)$=**-0.05173**

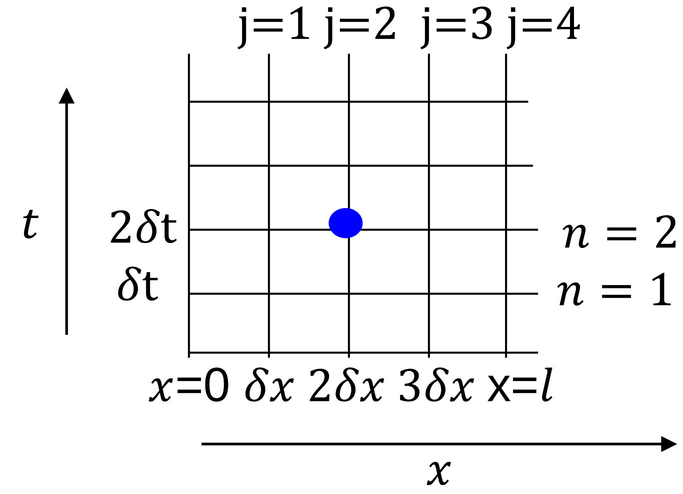
$u_3^2 = u_3^1 + r(u_2^1 - 2u_3^1 + u_4^1)$=0.06851



Explicit Difference Method: Stability

- Temperature at $2\delta x$ after $2\delta t$ time units went into negative! (when the boundaries were held constant at 0)
  - Example of *instability*

$$u_2^2 = u_2^1 + r(u_1^1 - 2u_2^1 + u_3^1) = \mathbf{-0.05173}$$



The solution is stable (for heat diffusion problem) only if the approximations for $u(x,t)$ do not get bigger in magnitude with time

Explicit Difference Method: Stability

- The solution for heat diffusion problem is stable only if:

$$r \leq \frac{1}{2}$$

Therefore, choose your time step in such a way that:

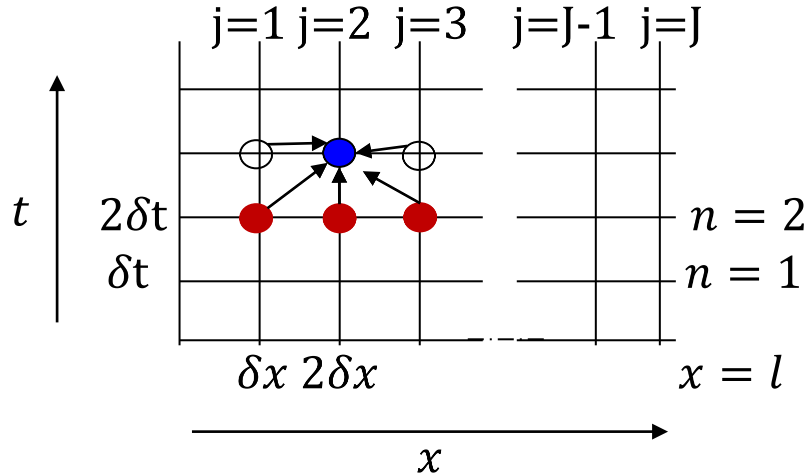
$$\delta t \leq \frac{\delta x^2}{2\alpha}$$

But this is a severe limitation!

Implicit Method: Stability

- Overcoming instability:

$$u_j^{n+1} = u_j^n \;+\; 1/2\, r(u_{j-1}^n - 2u_j^n + u_{j+1}^n + u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1} \;)$$



To compute the value of function at blue dot, you need 6 values indicated by the red dots (known) and 3 additional ones (unknown) above

Implicit Method: Stability

- Overcoming instability:

  $u_j^{n+1} = u_j^n \ + \ 1/2\ \mathrm{r}(\ u_{j-1}^n - 2u_j^n + u_{j+1}^n + u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}\ )$

- Extra work involved to determine the values of unknowns in a time step
  - Solve a system of simultaneous equations. Is it worth it?

Suggested Reading

- *J.W. Thomas. Numerical Partial Differential Equations: Finite Difference Methods*

- *Parabolic PDEs: https://learn.lboro.ac.uk/archive/olmp/olmp_resources/pages/workbooks_1_50_jan2008/Workbook32/32_4_prblc_pde.pdf*

Exercise

- Consider the *boundary-value* problem:

$u_{xx} + u_{yy} = 0$ in the square $0 < x < 1, 0 < y < 1$

$u = x^2 y$ on the boundary.

Is this Laplace equation or Poisson equation?

Elliptic Equation – Numerical Solution

1. Approximate the derivatives of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$ using central differences
2. Choose step sizes δx and δy for x and y axis resp.
   1. Both and x and y are independent variables here.
   2. Choose $\delta x = \delta y = h$
3. Write difference equation for approximating the PDE above

Elliptic Equation – Numerical Solution

1. Approximate the derivatives of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$ using central differences

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\left(u(x+\delta x, y) - 2u(x, y) + u(x-\delta x, y)\right)}{(\delta x)^2}$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{\left(u(x, y+\delta y) - 2u(x, y) + u(x, y-\delta y)\right)}{(\delta y)^2}$$

Where, δx and δy are step sizes along x and y direction resp.

Elliptic Equation – Numerical Solution

- Substituting in $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$:

$$\frac{\big(u(x+\delta x, y) - 2u(x, y) + u(x-\delta x, y)\big)}{(\delta x)^2}$$

$$\mathbf{+}$$

$$\frac{\big(u(x, y+\delta y) - 2u(x, y) + u(x, y-\delta y)\big)}{(\delta y)^2}$$

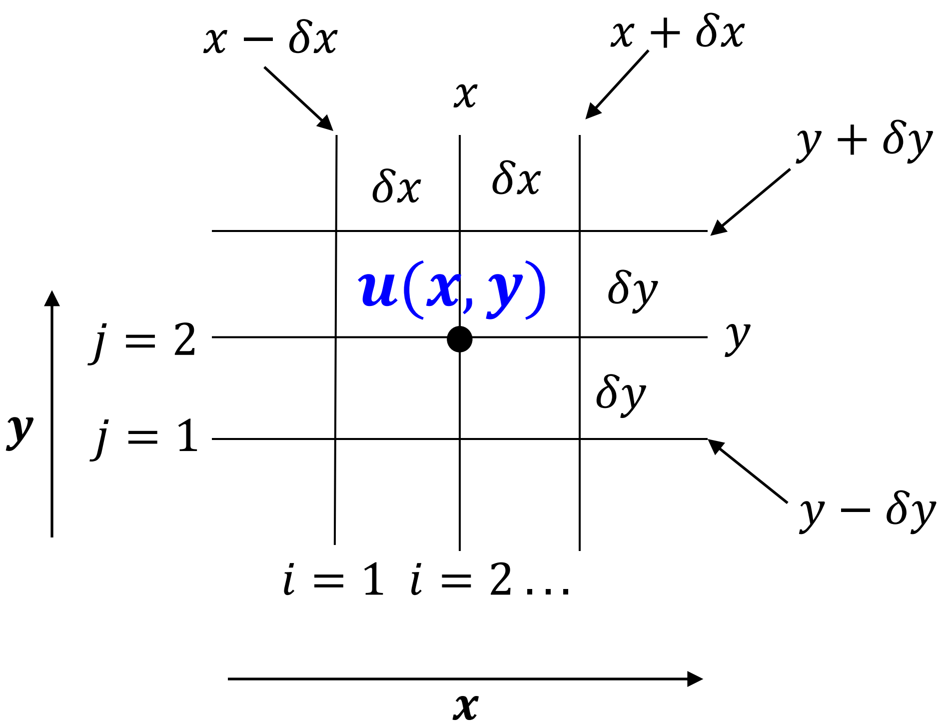
$$\mathbf{=}$$

$$\frac{\big(u(x+\delta x, y) + \ u(x, y+\delta y) - 4u(x, y) + u(x-\delta x, y) + \ u(x, y-\delta y)\big)}{(h)^2}$$

$$= f(x, y)$$

Elliptic Equation – Numerical Solution

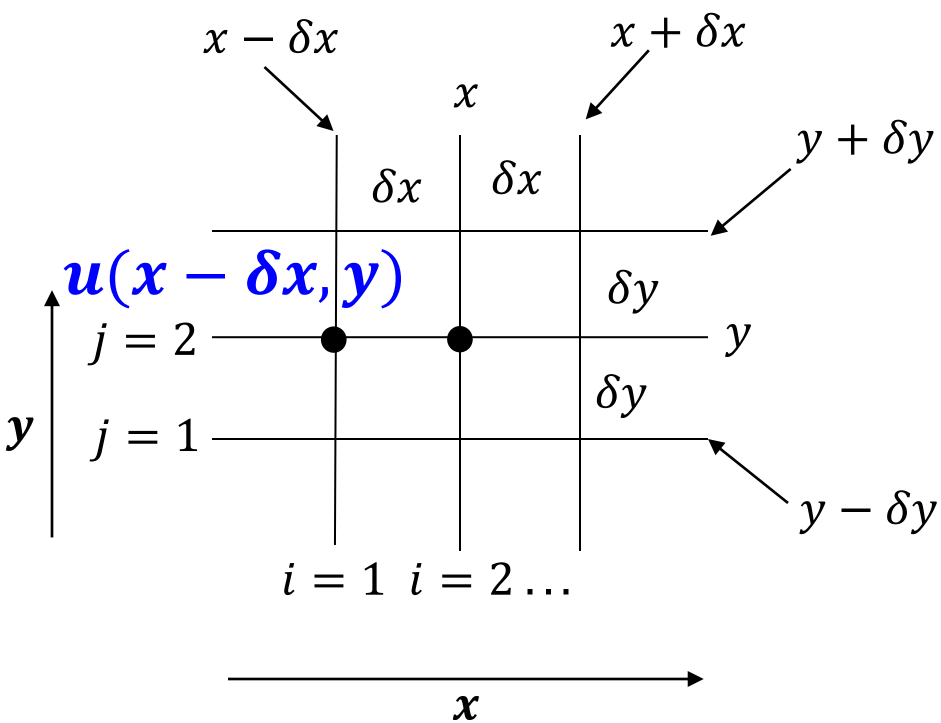
- Representing $u(x, y)$



Notation: $\mathbf{u_{i,j}}$

Elliptic Equation – Numerical Solution

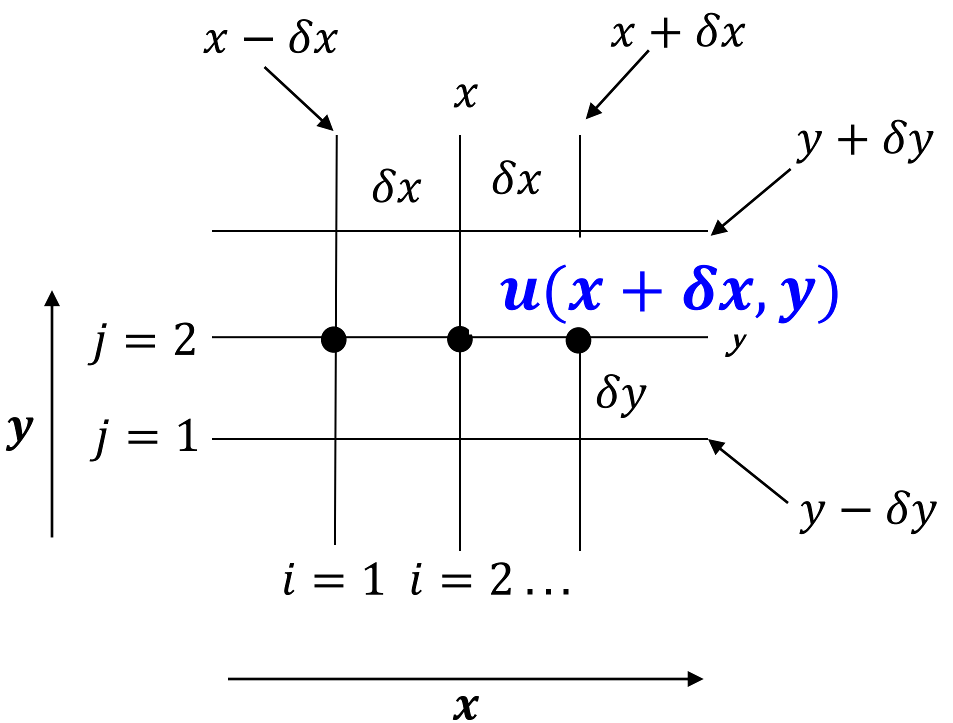
- Representing $u(x - \delta x, y)$



Notation: $\mathbf{u_{i-1,j}}$

Elliptic Equation – Numerical Solution

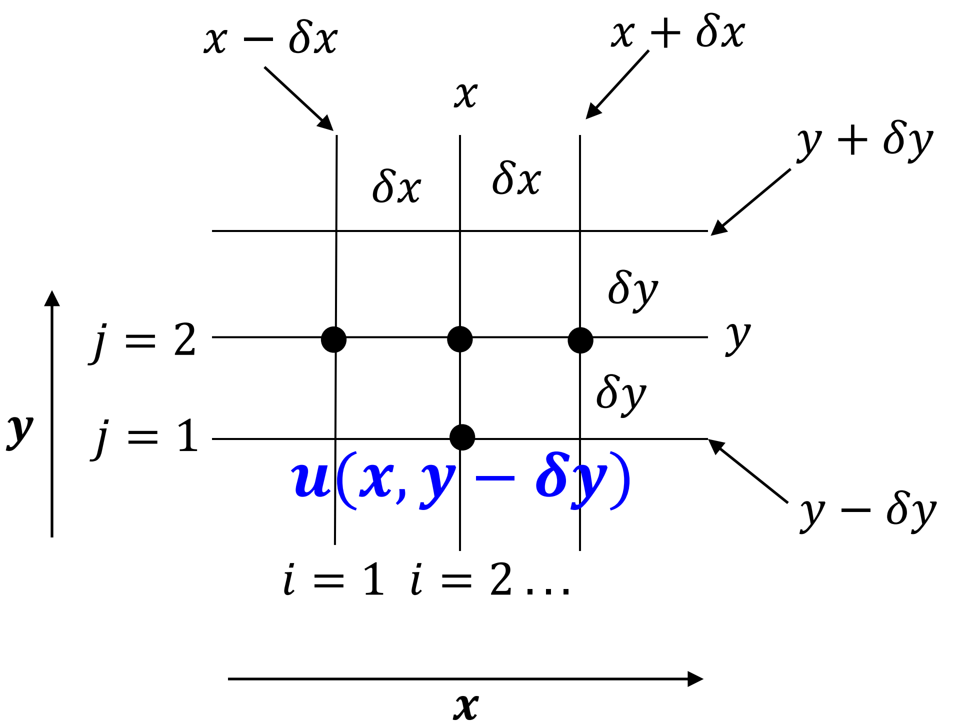
- Representing $u(x + \delta x, y)$



Notation: $\mathbf{u_{i+1,j}}$

Elliptic Equation – Numerical Solution

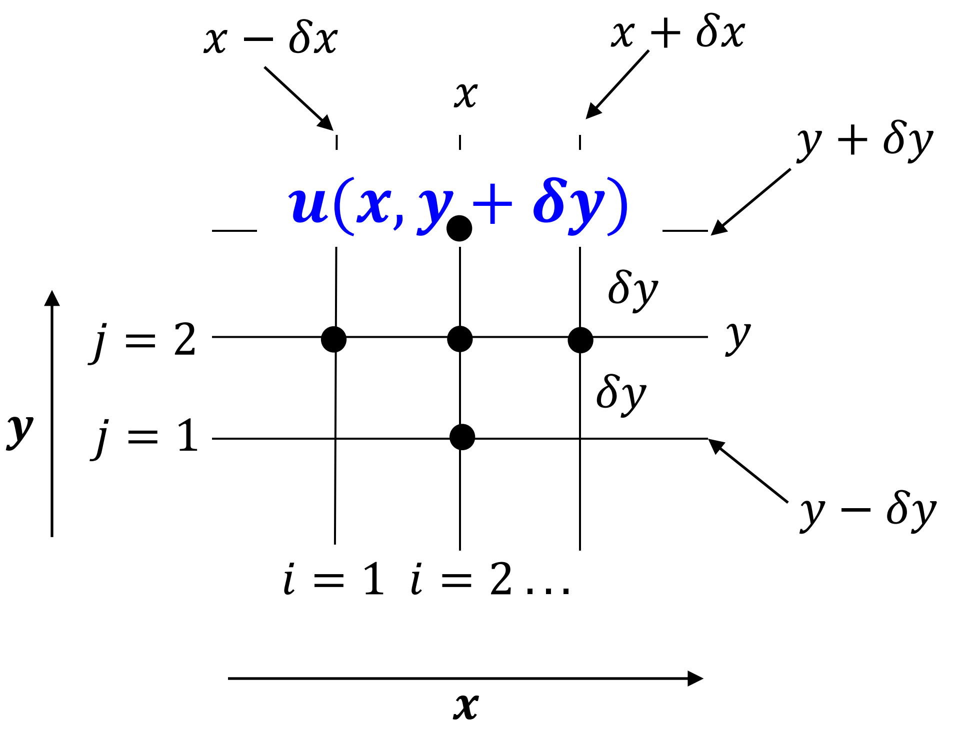
- Representing $u(x, y - \delta y)$



Notation: $\mathbf{u_{i,j-1}}$

Elliptic Equation – Numerical Solution

- Representing $u(x, y + \delta y)$



Notation: $\mathbf{u_{i,j+1}}$

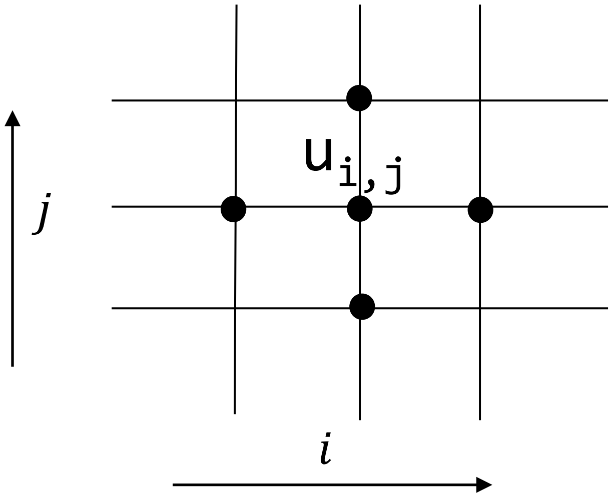
Elliptic Equation – Numerical Solution

- Rewriting:

$$\frac{\big(u(x+\delta x,y)+\ u(x,y+\delta y)-4u(x,y)+u(x-\delta x,y)+\ u(x,y-\delta y)\big)}{(h)^2}$$

$$= f(x,y)$$

$$\frac{\mathbf{u_{i+1,j} + u_{i,j+1}} - 4\mathbf{u_{i,j}} + \mathbf{u_{i-1,j} + u_{i,j-1}}}{\mathbf{h^2}} = \mathbf{f_{i,j}}$$



5-point stencil

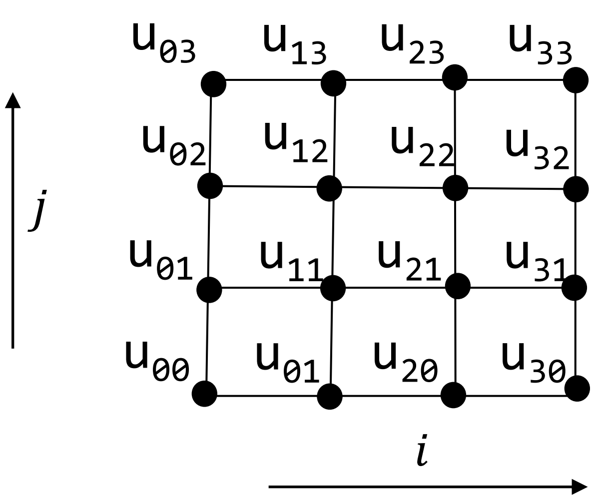
Elliptic Equation – Computing Stencil

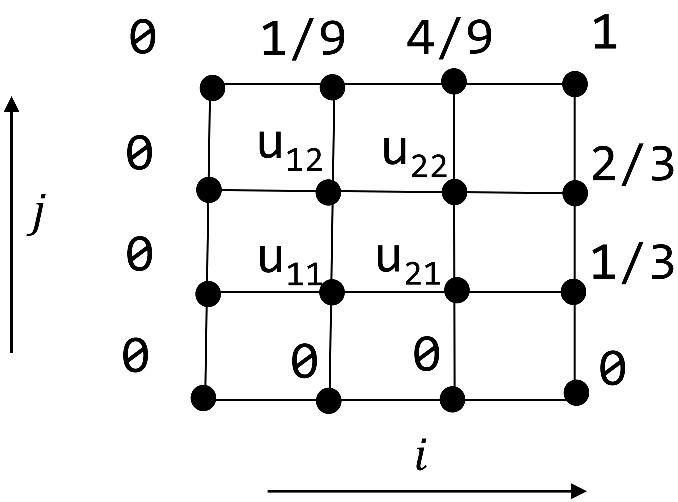
- Consider the *boundary-value* problem:

$u_{xx} + u_{yy} = 0$ in the square $0 < x < 1, 0 < y < 1$

$u = x^2 y$ on the boundary, $h = 1/3$

$$\frac{\mathbf{u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1}}}{\mathbf{h^2}} = \mathbf{0}$$





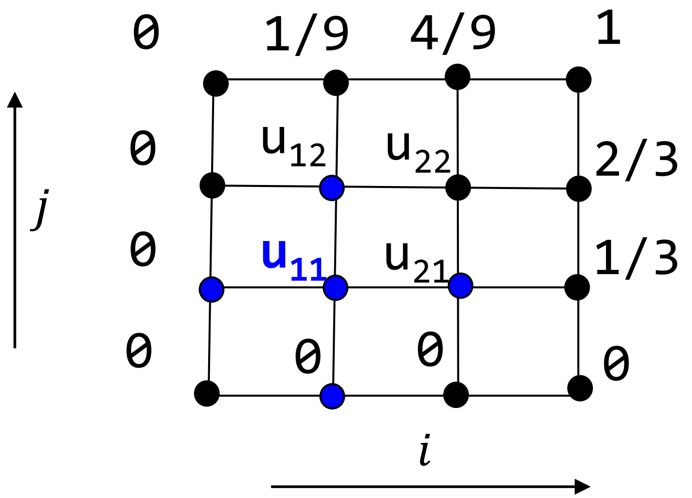
Elliptic Equation – Computing Stencil

- Computing u_{11}

$(u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1} = 0)$

$$u_{21} + u_{12} - 4u_{11} + u_{01} + u_{10} = 0$$

$$u_{21} + u_{12} - 4u_{11} + 0 + 0 = 0$$



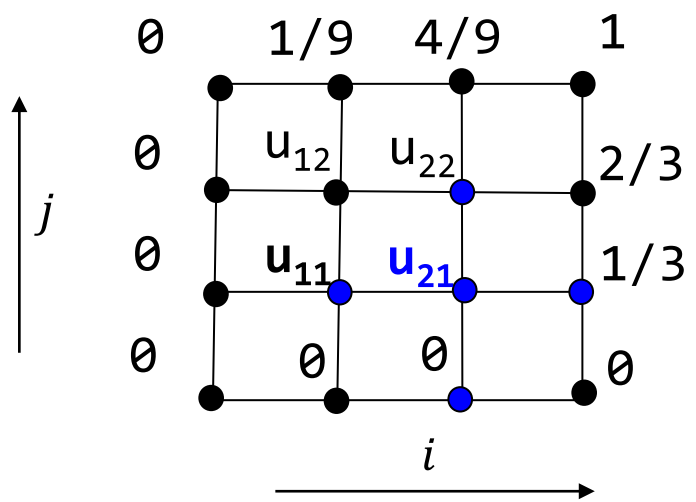
Elliptic Equation – Computing Stencil

- Computing u_{21}

$$(u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1} = 0)$$

$$u_{31} + u_{22} - 4u_{21} + u_{11} + u_{20} = 0$$

$$1/3 + u_{22} - 4u_{21} + U_{11} + 0 = 0$$



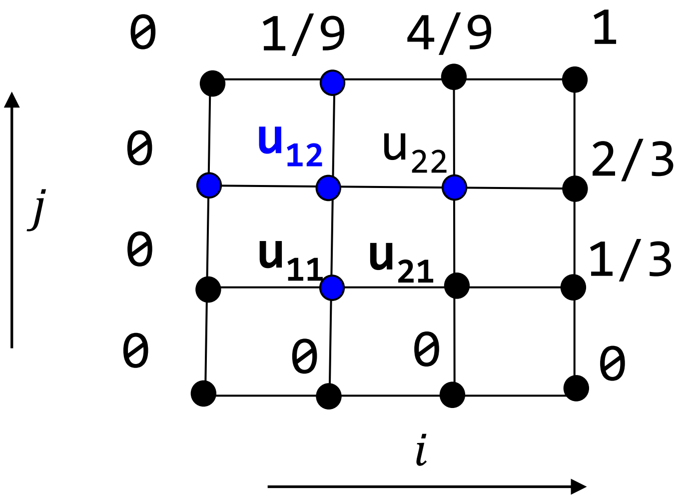
Elliptic Equation – Computing Stencil

- Computing u_{12}

$(u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1} = 0)$

$$u_{22} + u_{13} - 4u_{12} + u_{02} + u_{11} = 0$$

$$u_{22} + 1/9 - 4u_{12} + 0 + u_{11} = 0$$



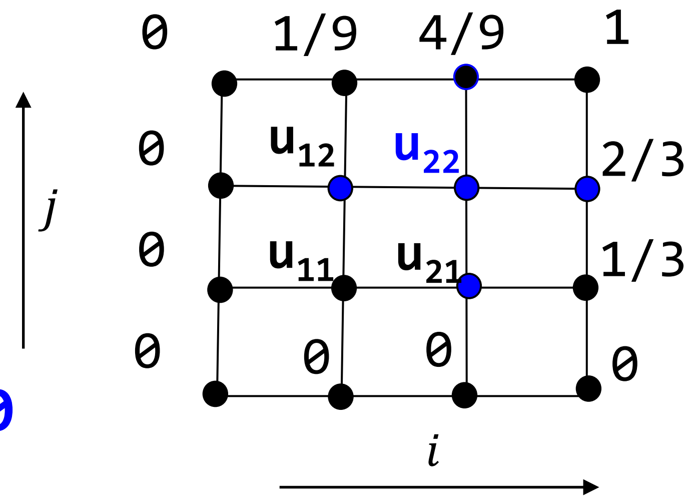
Elliptic Equation – Computing Stencil

- Computing u_{22}

$$(u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1} = 0)$$

$$u_{32} + u_{23} - 4u_{22} + u_{12} + u_{21} = 0$$

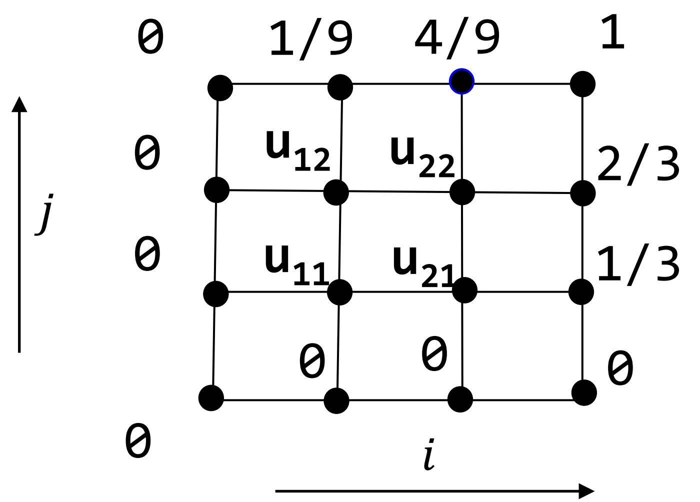
$$2/3 + 4/9 - 4u_{22} + u_{12} + u_{21} = 0$$



Elliptic Equation – Computing Stencil

- System of Equations

$$(u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1} = 0)$$



Right Top Center Left Bottom

$$u_{21} + u_{12} - 4u_{11} + 0 + 0 = 0$$

$$1/3 + u_{22} - 4u_{21} + u_{11} + 0 = 0$$

$$u_{22} + 1/9 - 4u_{12} + 0 + u_{11} = 0$$

$$2/3 + 4/9 - 4u_{22} + u_{12} + u_{21} = 0$$

Elliptic Equation – Computing Stencil

- Computing System of Equations:

$u_{21} + u_{12} - 4u_{11} + 0 + 0 = 0$

$1/3 + u_{22} - 4u_{21} + u_{11} + 0 = 0$

$u_{22} + 1/9 - 4u_{12} + 0 + u_{11} = 0$

$2/3 + 4/9 - 4u_{22} + u_{12} + u_{21} = 0$

$$\underset{A}{\begin{pmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{pmatrix}} \underset{x}{\begin{pmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{pmatrix}} \underset{=}{=} \underset{B}{\begin{pmatrix} 0 \\ -1/3 \\ -1/9 \\ -10/9 \end{pmatrix}}$$

Ax=B

Matrix A has only coefficients

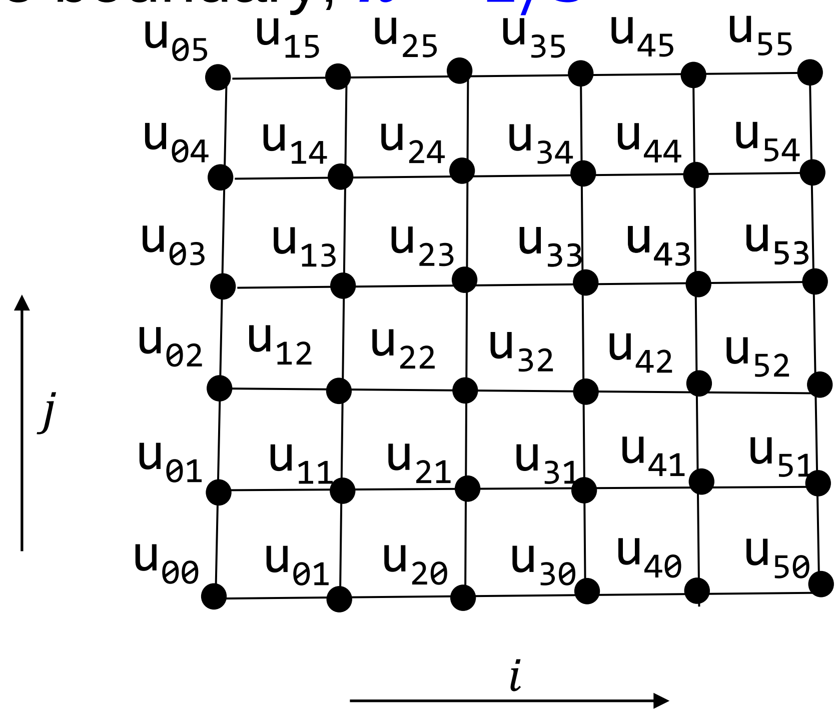
	1	
1	**-4**	**1**
	1	

Elliptic Equation – Computing Stencil

- Consider the *boundary-value* problem (here u_{xx} denotes $\partial^2 u/\partial x^2$):

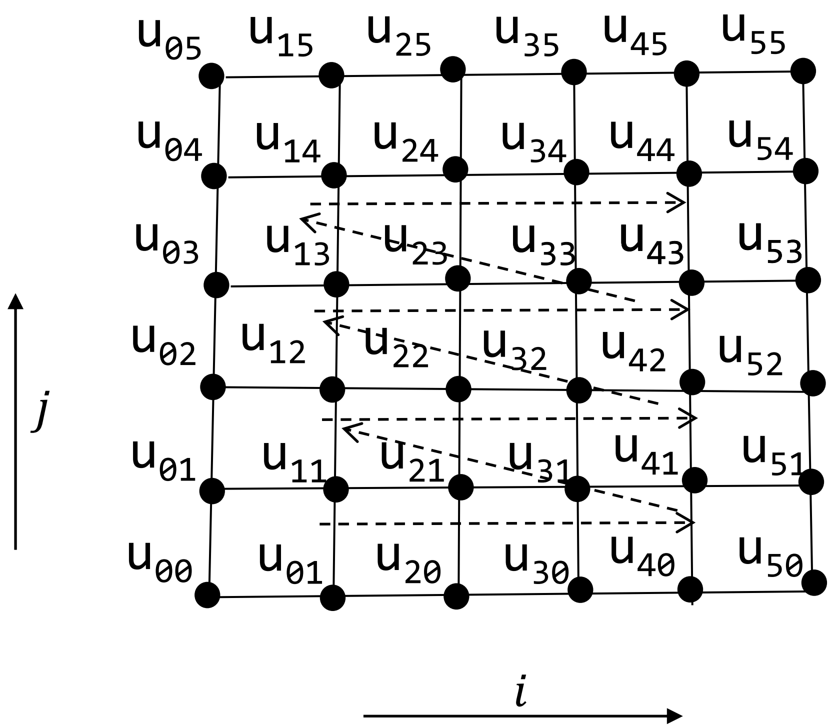
$u_{xx} + u_{yy} = 0$ in the square $0 < x < 1, 0 < y < 1$

$u = x^2 y$ on the boundary, $\boldsymbol{h = 1/5}$



Elliptic Equation – Computing Stencil

- Computing stencil (boundary values are all given): 16 unknowns (u_{11} to u_{44}), So, 16 equations.



Elliptic Equation – Computing Stencil

-4	1	0	0	1								
1	-4	1	0	0	1							
0	1	-4	1	0	0	1						
0	0	1	-4	1	0	0	1					
1	0	0	1	-4	1	0	0	1				
	1	0	0	1	-4	1	0	0	1			
		1	0	0	1	-4	1	0	0	1		
			1	0	0	1	-4	1	0	0	1	
				1	0	0	1	-4	1	0	0	1

- Lot of Zeros!
- Five non-zero bands
  - Top-left to bottom-right diagonals
- Main diagonal is all -4 (from center of the stencil)
- What about others?

Elliptic Equation – Computing Stencil

-4	1	0	0	1								
1	-4	1	0	0	1							
0	1	-4	1	0	0	1						
0	0	1	-4	1	0	0	1					
1	0	0	1	-4	1	0	0	1				
	1	0	0	1	-4	1	0	0	1			
		1	0	0	1	-4	1	0	0	1		
			1	0	0	1	-4	1	0	0	1	
				1	0	0	1	-4	1	0	0	1

Left

- Lot of Zeros!
- Five non-zero bands
  - Top-left to bottom-right diagonals
- Main diagonal is all -4 (from center of the stencil)
- What about others?

Elliptic Equation – Computing Stencil

-4	1	0	0	1								
1	-4	1	0	0	1							
0	1	-4	1	0	0	1						
0	0	1	-4	1	0	0	1					
1	0	0	1	-4	1	0	0	1				
	1	0	0	1	-4	1	0	0	1			
		1	0	0	1	-4	1	0	0	1		
			1	0	0	1	-4	1	0	0	1	
				1	0	0	1	-4	1	0	0	1

Right

- Lot of Zeros!
- Five non-zero bands
  - Top-left to bottom-right diagonals
- Main diagonal is all -4 (from center of the stencil)
- What about others?

Elliptic Equation – Computing Stencil

-4	1	0	0	1								
1	-4	1	0	0	1							
0	1	-4	1	0	0	1						
0	0	1	-4	1	0	0	1					
1	0	0	1	-4	1	0	0	1				
	1	0	0	1	-4	1	0	0	1			
		1	0	0	1	-4	1	0	0	1		
			1	0	0	1	-4	1	0	0	1	
				1	0	0	1	-4	1	0	0	1

Bottom

- Lot of Zeros!
- Five non-zero bands
  - Top-left to bottom-right diagonals
- Main diagonal is all -4 (from center of the stencil)
- What about others?

Elliptic Equation – Computing Stencil

-4	1	0	0	1								
1	-4	1	0	0	1							
0	1	-4	1	0	0	1						
0	0	1	-4	1	0	0	1					
1	0	0	1	-4	1	0	0	1				
	1	0	0	1	-4	1	0	0	1			
		1	0	0	1	-4	1	0	0	1		
			1	0	0	1	-4	1	0	0	1	
				1	0	0	1	-4	1	0	0	1

Top

- Lot of Zeros!
- Five non-zero bands
  - Top-left to bottom-right diagonals
- Main diagonal is all -4 (from center of the stencil)
- What about others?

Computing Stencil – Iterative Methods

- Jacobi and Gauss-Seidel
  - Start with an initial guess for the unknowns u^0_{ij}
  - Improve the guess u^1_{ij}
  - Iterate: derive the new guess, u^{n+1}_{ij} , from old guess u^n_{ij}
- Solution (Jacobi):
  - Approximate the *value of the center* with old values of (left, right, top, bottom)

Background – Jacobi Iteration

- **Goal:** find solution to system of equations represented by `AX=B`
- **Approach:** find sequence of approximations X^0 X^1 X^2 . . . X^n , which gradually approach `X.`
  - X^0 is called initial guess, X^i's called *iterates*
- **Method:**
  - Split `A` into `A=L+D+U` e.g.

$$\begin{pmatrix}-4 & 1 & 1 & 0\\ 1 & -4 & 0 & 1\\ 1 & 0 & -4 & 1\\ 0 & 1 & 1 & -4\end{pmatrix} = \underbrace{\begin{pmatrix}0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 1 & 0\end{pmatrix}}_{L} + \underbrace{\begin{pmatrix}-4 & 0 & 0 & 0\\ 0 & -4 & 0 & 0\\ 0 & 0 & -4 & 0\\ 0 & 0 & 0 & -4\end{pmatrix}}_{D} + \underbrace{\begin{pmatrix}0 & 1 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{pmatrix}}_{U}$$

Background – Jacobi Iteration

- **Compute:** $AX=B$ is $(L+D+U)X=B$

$\Rightarrow$ $DX = -(L+U)X+B$

$\Rightarrow$ $DX^{(k+1)} = -(L+U)X^{k}+B$ **(iterate step)**

$\Rightarrow$ $X^{(k+1)} = D^{-1}(-(L+U)X^{k}) + D^{-1}B$

(As long as D has no zeros in the diagonal $X^{(k+1)}$ is obtained)

- E.g. $$\begin{pmatrix} -4 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{pmatrix}^{1} = -\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{pmatrix}^{0} + \begin{pmatrix} 0 \\ -1/3 \\ -1/9 \\ -10/9 \end{pmatrix},$$

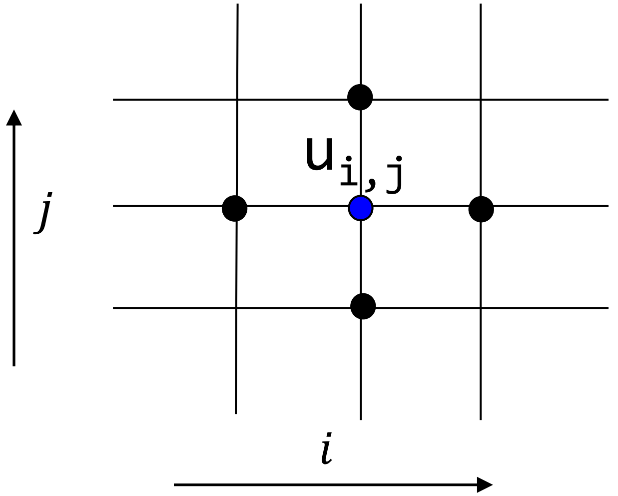
u_{ij} 's value in $(\mathbf{1})^{st}$ iteration is computed based on u_{ij} values computed in $(\mathbf{0})^{th}$ iteration

Background – Jacobi Iteration

- E.g. $\begin{pmatrix} -4 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{pmatrix}^{\mathbf{k+1}} = -\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{pmatrix}^{\mathbf{k}} + \begin{pmatrix} 0 \\ -1/3 \\ -1/9 \\ -10/9 \end{pmatrix}$,

u_{ij} 's value in $(\mathbf{k+1})^{st}$ iteration is computed based on u_{ij} values computed in $(\mathbf{k})^{th}$ iteration

- Center's value is updated. Why?



5-point stencil

Computing Stencil – Recap

- Jacobi and Gauss-Seidel (Solution approach)
  - Start with an initial guess for the unknowns u^0_{ij}
  - Improve the guess u^1_{ij}
  - Iterate: derive the new guess, u^{n+1}_{ij} , from old guess u^n_{ij}
- Solution (Jacobi):
  - Approximate the *value of the center* with *old values* of (left, right, top, bottom)

Computing Stencil – Recap

- $\boldsymbol{u_{right} + u_{top} - 4u_{center} + u_{left} + u_{bottom} = 0}$

  => $\boldsymbol{u_{center} = 1/4(u_{right} + u_{top} + u_{left} + u_{bottom})}$

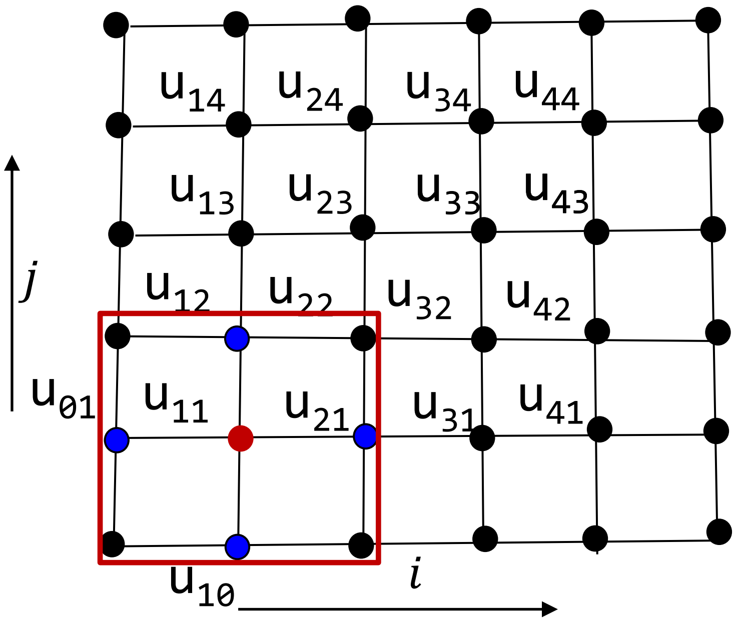
- Applying Jacobi Iteration:

  $$\boldsymbol{u_{center}^{(k+1)} = 1/4(u_{right}^{(k)} + u_{top}^{(k)} + u_{left}^{(k)} + u_{bottom}^{(k)})}$$

Computing Stencil – Recap

- Example: applying Jacobi Iteration:

$$u_{center}^{(k+1)} = 1/4(u_{right}^{(k)} + u_{top}^{(k)} + u_{left}^{(k)} + u_{bottom}^{(k)})$$



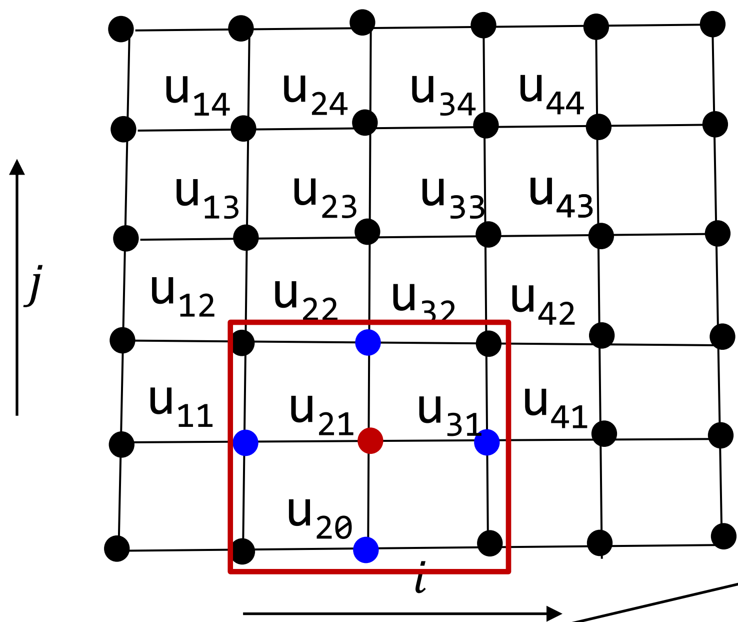
Iteration 1

1) Compute u_{11} using initial guess for u_{12} and u_{21}. u_{01} and u_{10} are known from boundary conditions

Computing Stencil – Recap

- Example: applying Jacobi Iteration:

$$u_{center}^{(k+1)} = 1/4(u_{right}^{(k)} + u_{top}^{(k)} + u_{left}^{(k)} + u_{bottom}^{(k)})$$



Iteration 1

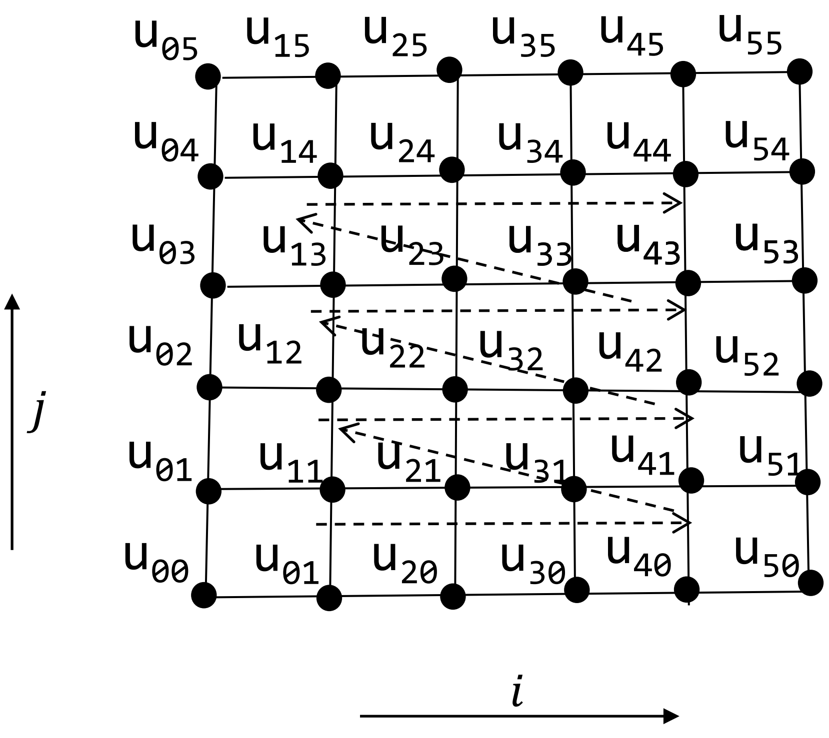
1) Compute u_{11} using initial guess for u_{12} and u_{21}. u_{01} and u_{10} are known from boundary conditions

2) Compute u_{21} using initial guess for u_{11}, u_{31}, and u_{22}. u_{20} are known from boundary conditions

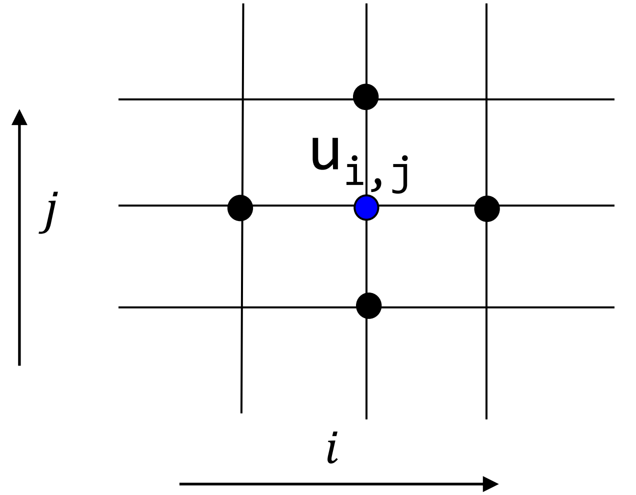
In 2), note that the initial guess for u_{11} is used even though u_{11} was updated just before in 1)

Elliptic Equation – Computing Stencil

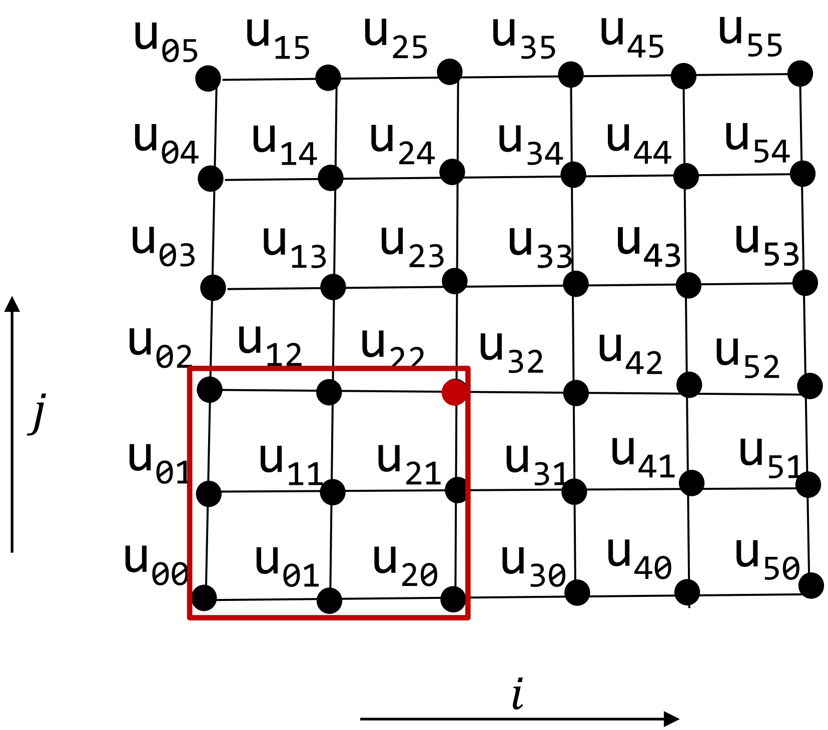
- In every iteration, suppose we follow the computing order as shown (dashed):



In any iteration, what are all the points of a 5-point stencil already updated while computing u_{ij} ?

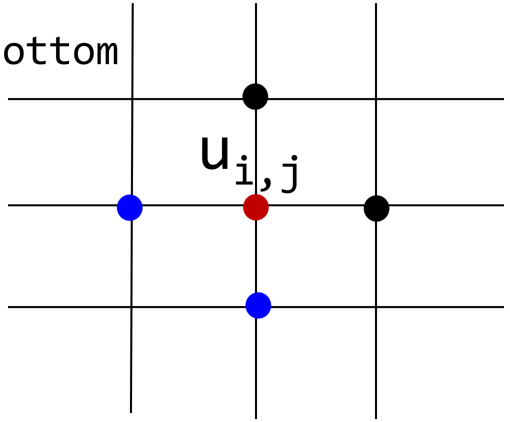


Elliptic Equation – Computing Stencil



What are the points that are already computed at $u_{i,j}$*?*

u_{left}, u_{bottom}



Background – Gauss-Seidel Iteration

- **Compute:** AX=B is (L+D+U)X=B

  $\Rightarrow$ (L+D)X = -UX+B

  $\Rightarrow$ $(L+D)X^{(k+1)} = -UX^{k}+B$ **(iterate step)**

  $\Rightarrow$ $X^{(k+1)} = (L+D)^{-1}\ (-UX^{k}) + (L+D)^{-1}B$

  (As long as L+D has no zeros in the diagonal $X^{(k+1)}$ is obtained)

- E.g. $$\begin{pmatrix} -4 & 0 & 0 & 0 \\ 1 & -4 & 0 & 0 \\ 1 & 0 & -4 & 0 \\ 0 & 1 & 1 & -4 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{pmatrix}^{1} = - \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \end{pmatrix}^{0} + \begin{pmatrix} 0 \\ -1/3 \\ -1/9 \\ -10/9 \end{pmatrix}$$

Computing Stencil – Gauss-Seidel

- Gauss-Seidel: Applying for 2D Laplace Equation

$$u_{center}^{(k+1)} = 1/4(u_{right}^{(k)} + u_{top}^{(k)} + u_{left}^{(k+1)} + u_{bottom}^{(k+1)})$$

- Gauss-Seidel: Observations
  - For a given problem and initial guess, Gauss-seidel *converges faster* than Jacobi
  - An iteration in Jacobi can be parallelized