# Finite Difference Method

# Finite Difference Method

Examples:

$\nabla^2 u = 0$ Laplace's equation

- Potential flow
- Electrical potential distribution
- Pressure distribution

$\nabla^2 u + \lambda^2 u = 0$ Helmholtz's equation

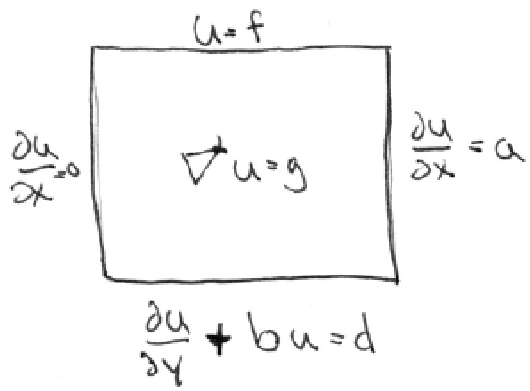
- Harmonic elasticity
- Harmonic acoustic waves

$\nabla^2 u = k$ Poisson's equation

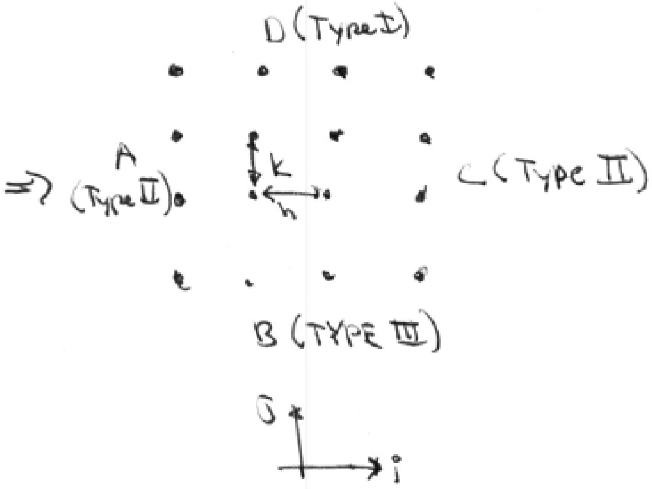
- Electrical Potential distribution in the presence of dipole
- Sink / source modeling

# Finite Difference Method

Ex. Consider:




⇒




PDE: $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = g$

⇒ want second-order, centered FD expressions:

# Finite Difference Method

$$\Rightarrow \frac{\delta_x^2 u_{i,j}}{h^2} + \frac{\delta_y^2 u_{i,j}}{k^2} = g_{i,j}$$

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = g_{i,j}$$

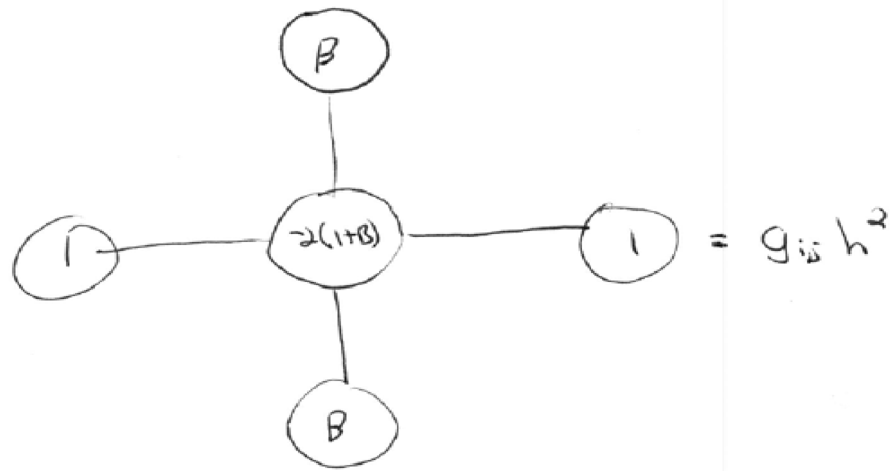
let $\beta = \frac{h^2}{k^2}$, $\Rightarrow$ $u_{i+1,j} - 2u_{i,j} + u_{i-1,j}$

$+$

$\beta(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = h^2 g_{i,j}$

# Finite Difference Method

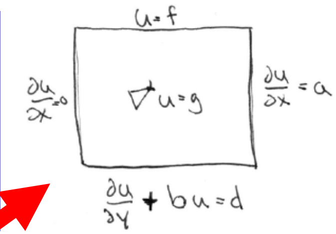
At each point on the grid:



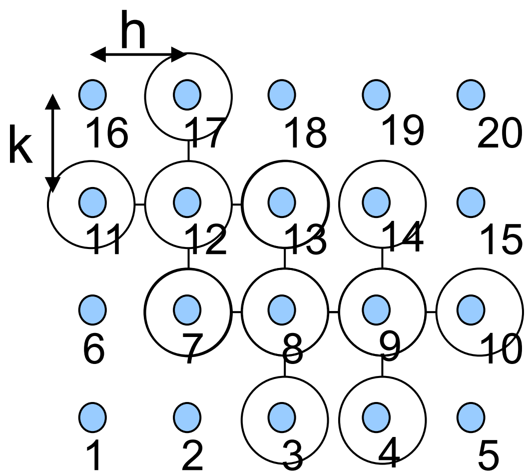

i.e. "computational molecule"

Valid at all interior nodes, but what about boundaries? ⇒ molecule "spills" over

# Let's be a bit more deliberate…




## FDM applied to… gives




Enforcing physics spatially…

$$\vdots$$

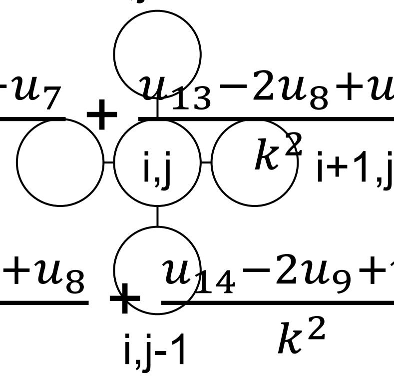
$$\frac{u_8-2u_7+u_6}{h^2}+\frac{u_{12}-2u_7+u_2}{k^2}=g_7$$

$$\frac{u_9-2u_8+u_7}{h^2}+\frac{u_{13}-2u_8+u_3}{k^2}=g_8$$

$$\frac{u_{10}-2u_9+u_8}{h^2}+\frac{u_{14}-2u_9+u_4}{k^2}=g_9$$

$$\frac{u_{13}-2u_{12}+u_{11}}{h^2}+\frac{u_{17}-2u_{12}+u_7}{k^2}=g_{12}$$

$$\vdots$$



# Finite Difference Method

- Boundary Conditions

  - Type I (Dirchlet): value of dependent variable 'u' directly specified on boundary point (i.e. $u = ??$)

  - Type II (Neumann): value of normal-to-the-boundary derivative of dependent variable specified (i.e. $\frac{\partial u}{\partial n} = ??$)

  - Type III (Robin): Linear combination of the dependent variable and its normal derivative specified (i.e. $\alpha u + \beta \frac{\partial u}{\partial n} = ??$)

# Finite Difference Method

- For elliptic PDE, one and only one of these boundary conditions must be specified at every point to get unique solution.

Note: For Laplace, specification of all Type II boundary conditions is not a unique solution.

e.g

$$\frac{\partial^2 u}{\partial x^2} = 0$$

$$\iint \frac{\partial^2 u}{\partial x^2} \Rightarrow u(x) = Ax + b$$

Type II info. Only gives information about 'A' – arbitrary to a constant

# Finite Difference Method

Type I BCs:

- specification of exact value: (strong influence)
- throwing out PDE equation in favor of actual value

e.g.

- $\nabla \cdot \sigma \nabla V = 0$ } Laplace's Eqn.

$V = \Phi$ } Type I BC specifying potential at a point equal to $\Phi$

$\nabla \cdot \sigma = 0$ } mechanical equilibrium

$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} d_x \\ d_y \\ d_z \end{Bmatrix}$ Type I BC specifying displacement as cartesian components $\{d_x, d_y, d_z\}$

# Finite Difference Method

Type II BCs

- specification of flux : (natural BC)

$$J \cdot \hat{n} = -\sigma \frac{\partial V}{\partial n} = a$$ } change in potential in the direction normal to the boundary is 'a'

$\sigma_t = 0$
$\sigma_n = m$ } allows slippage tangential to the boundary but a stress 'm' is applied normal to the surface

# Finite Difference Method

Type III BCs

- specification of linear combination to represent flux (radiation conditions)

e.g. $\frac{\partial T}{\partial t} = \nabla \cdot k \nabla T$ } Diffusion of heat

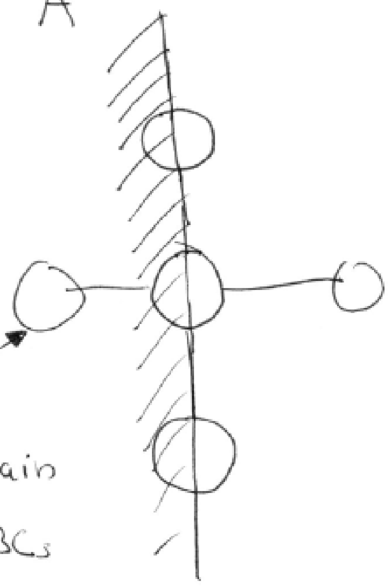
$-k \frac{\partial T}{\partial n} = h(T - T_o)$ } Type III specification where temperature outside ($T_o$) interacts with temperature flux into object (LHS) through coupling coefficient 'h'

$-\nabla \cdot k \nabla p = 0$ } ICP in brain

$-k \frac{\partial p}{\partial n} = h(P - P_o)$ } Type III BC specification allowing transfer of fluid across brain boundary

# Finite Difference Method

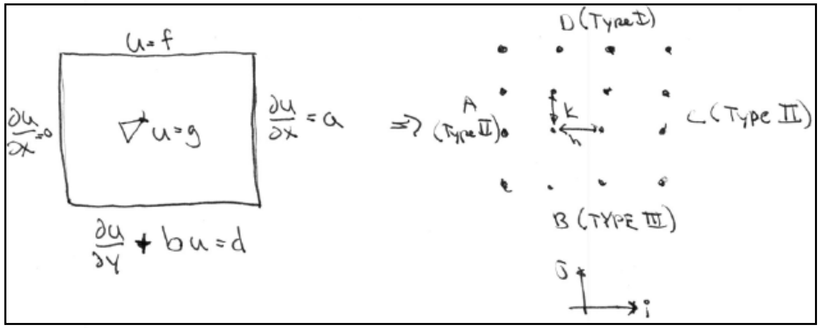
e.g. Boundary A



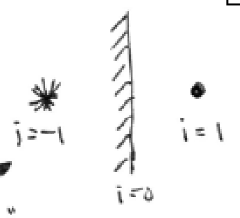
interior

outside the domain

Remove through BCs




$\left.\frac{\partial u}{\partial x}\right|_{i=0} = 0$ given $\Rightarrow$

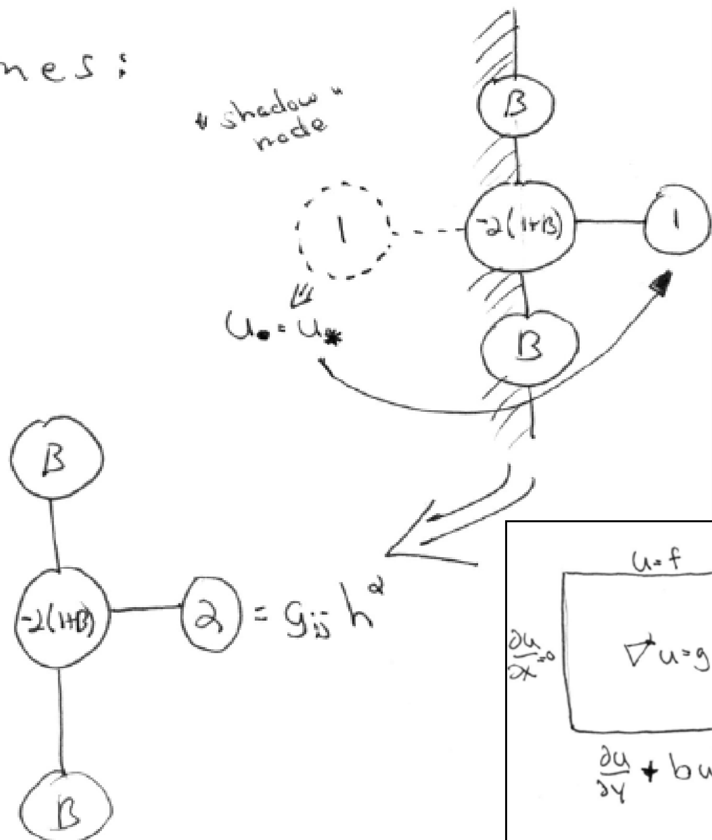


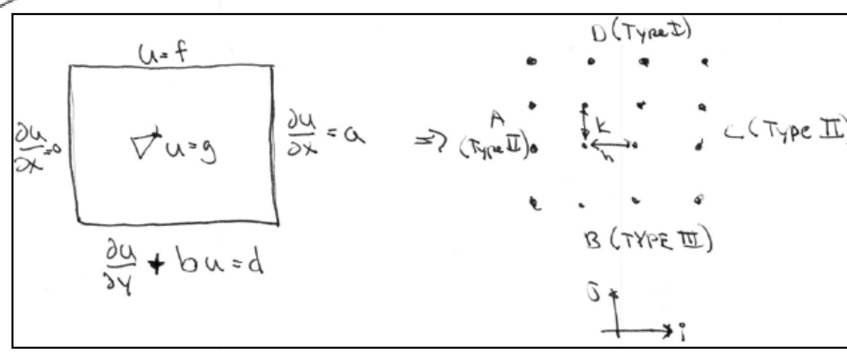

$\Rightarrow \frac{u_\bullet - u_*}{2h} = 0$

$\Rightarrow u_\bullet = u_*$

# Finite Difference Method

⇒ for nodes on Boundary A, the molecule becomes:

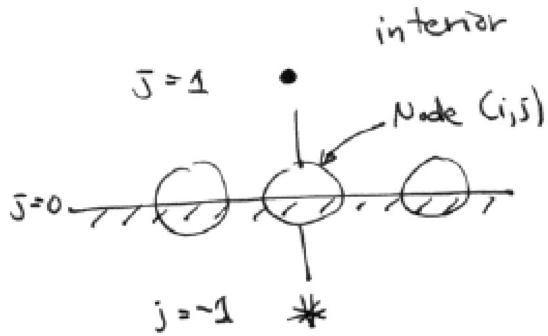


# Finite Difference Method

Boundary B:

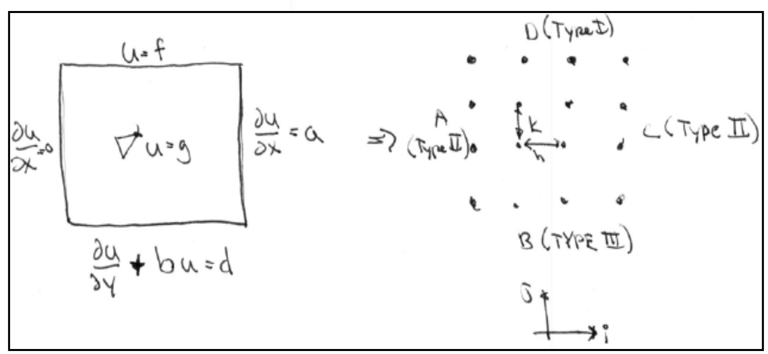
$$\left.\frac{\partial u}{\partial y}\right|_{j=0} + bu = d$$



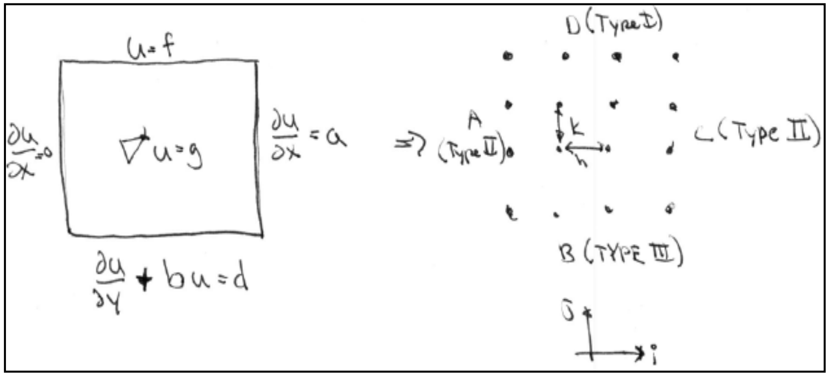

$$\Rightarrow \frac{u_\bullet - u_*}{2k} + b u_{i,j} = d$$

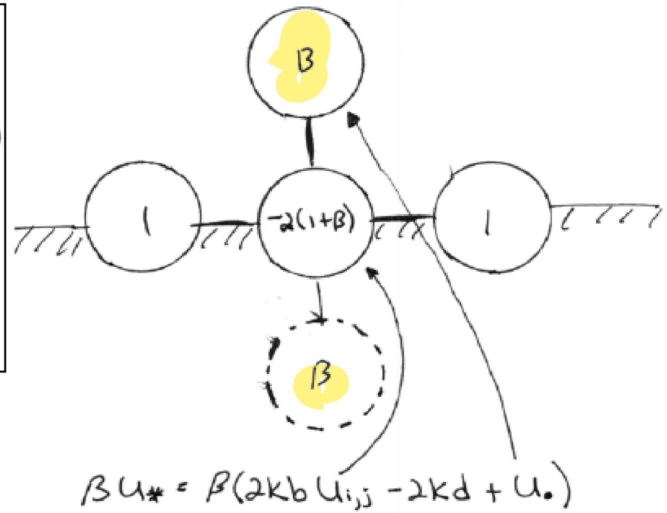
solve for $u_*$

$$\hookrightarrow u_* = 2kb\,u_{i,j} - 2kd + u_\bullet$$

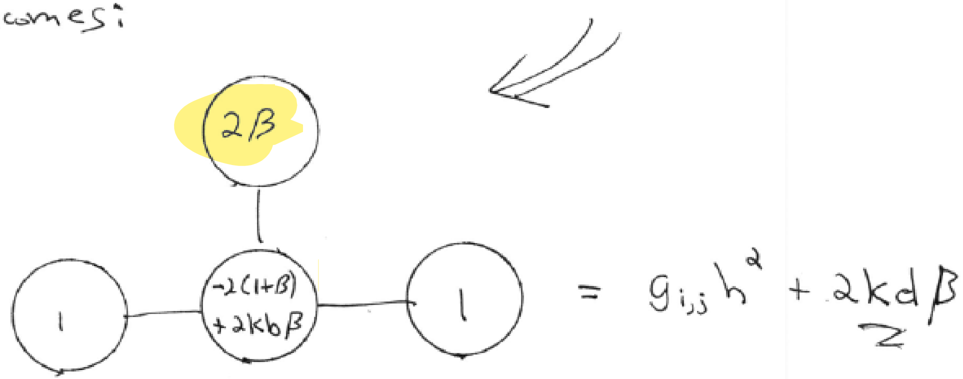


# Finite Difference Method



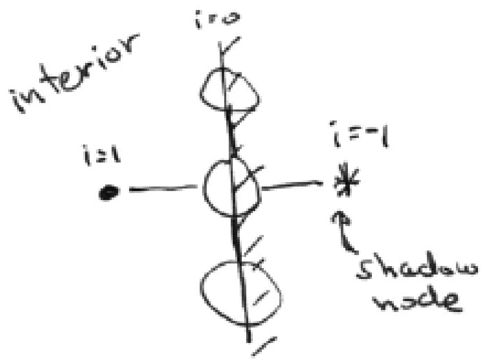

becomes:



# Finite Difference Method

Boundary C:

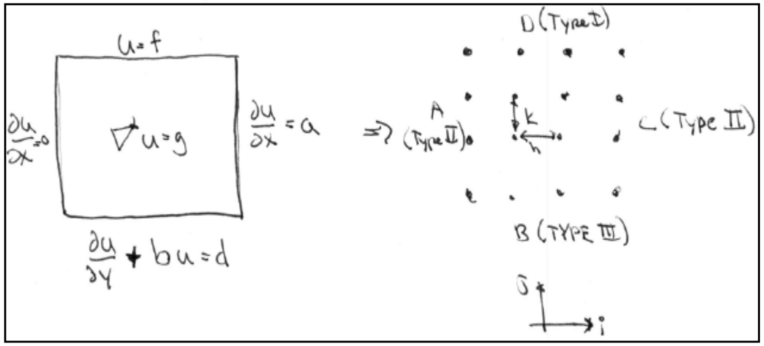
$$\left.\frac{\partial u}{\partial x}\right|_{i=0} = a$$



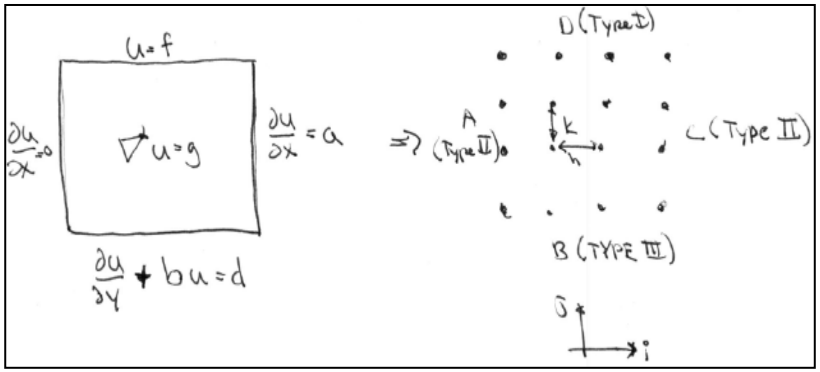

$$\Rightarrow \frac{u_* - u_0}{2h} = a$$

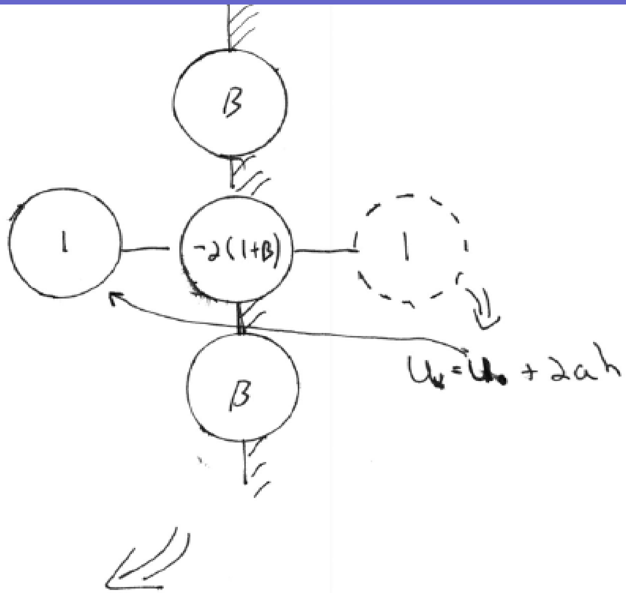
solve for $u_*$

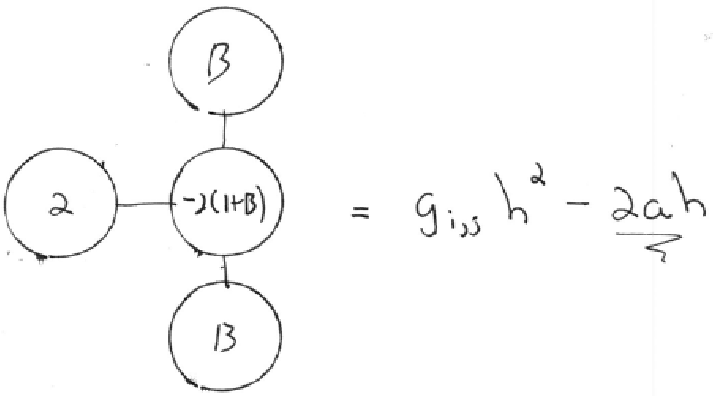
$$\hookrightarrow \quad u_* = u_0 + 2ah$$



# Finite Difference Method



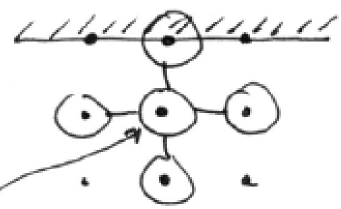



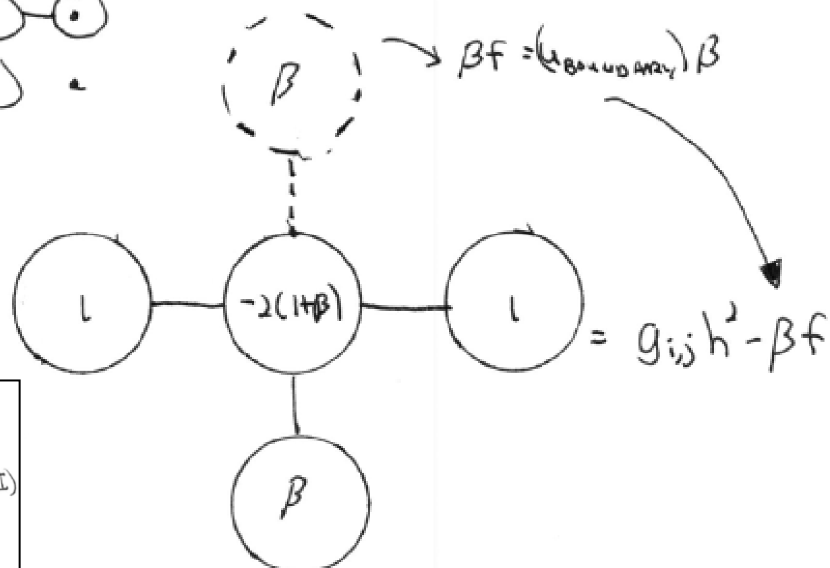
# Finite Difference Method

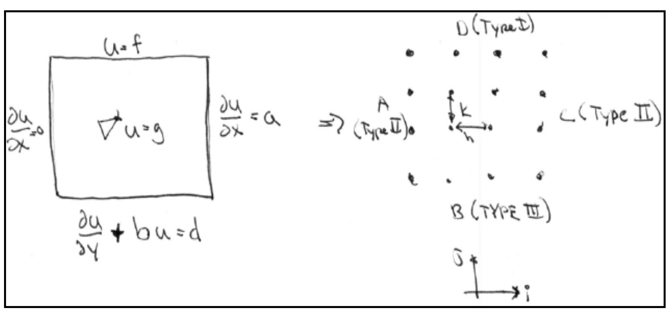
Boundary D: Type 1 condition ... don't use PDE

- Stop one node short of boundary



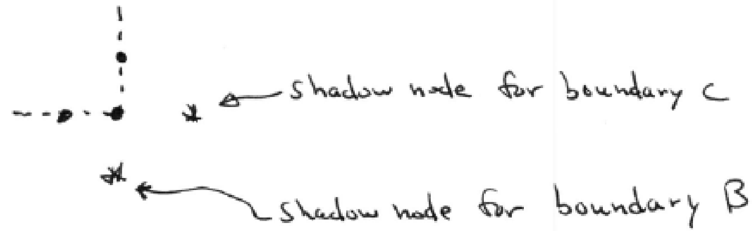
MOLECULE FOR THIS NODE :=>

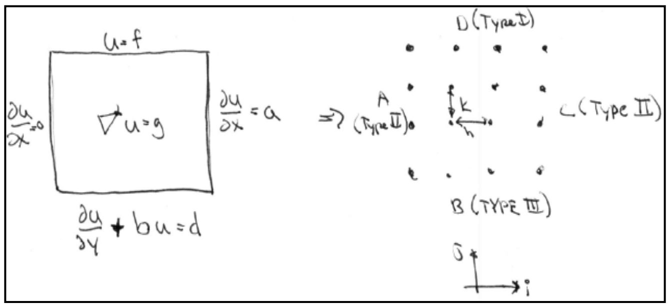


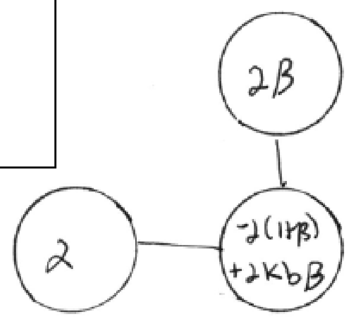



# Finite Difference Method

Corners: e.g




$$= g_{i,j} h^2 - 2ah + 2k\beta d$$

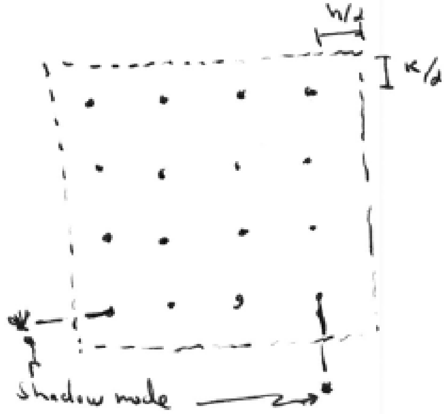
Basic Rule:

- Type I BC do not use PDE (i.e. let type 1 override)
- Type II, III, USE PDE PLUS BC together (i.e. use BC to eliminate shadow node)

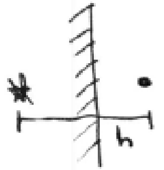
# Finite Difference Method

Alternate strategy ... place nodes $h/2$ from boundary

i.e. Think of nodes as center of "cells"



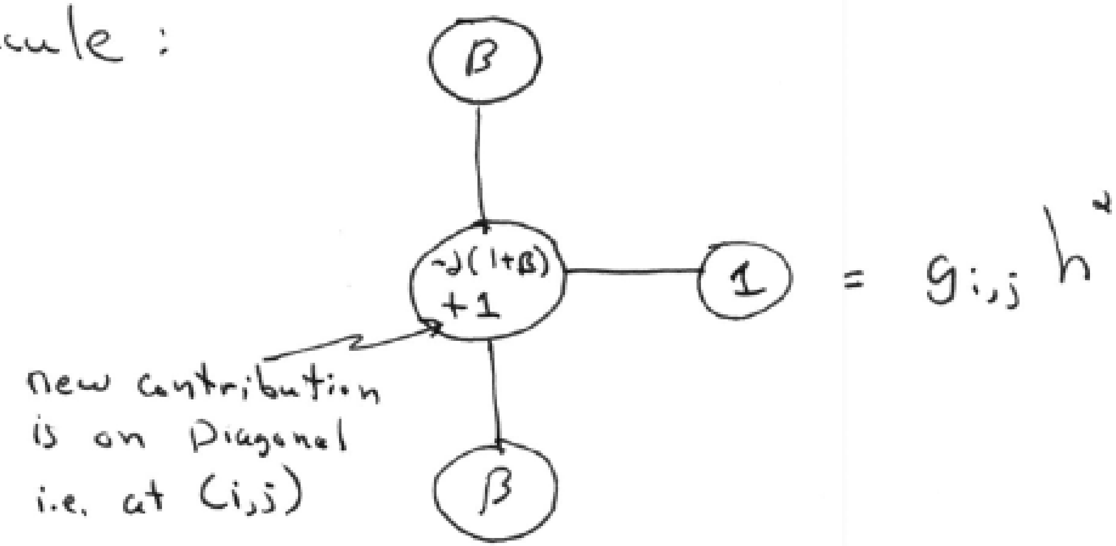

So TYPE II condition: $\frac{\partial u}{\partial x} = 0$



# Finite Difference Method

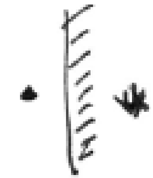
$$\frac{u_{\bullet} - u_{*}}{h} = 0 \Rightarrow u_{\bullet} = u_{*}$$

molecule :



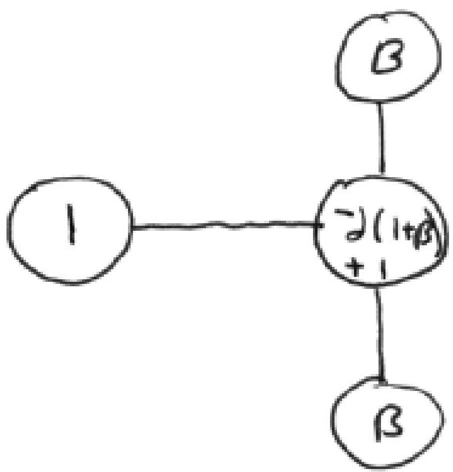
# Finite Difference Method

Type II condition (cont.): $\dfrac{\partial u}{\partial x} = a$



$$\frac{u_* - u_\bullet}{h} = a \Rightarrow u_* = ah + u_\bullet$$

molecule

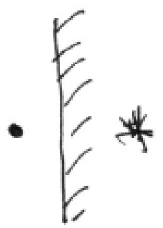



$= g_{ij} h^2 - ah$

# Finite Difference Method

"Cell" strategy not particularly convenient when $u$ involved on boundary $\Rightarrow$ Type I or Type III

e.g.



$$\frac{\partial u}{\partial x} + bu = a$$

$$\frac{u_* - u_0}{h} + b\, u_{bound} = a$$

Take as average of $u_0 + u_* \Rightarrow \dfrac{u_0 + u_*}{2}$

# Finite Difference Method

- Not just a simple average, but derived and have truncation error associated with estimation

$$U_* = U_b + \frac{h}{2}\frac{\partial U_b}{\partial x} + \frac{\left(\frac{h^2}{2}\right)}{2}\frac{\partial^2 U_b}{\partial x^2} + \cdots$$

$$U_\bullet = U_b - \frac{h}{2}\frac{\partial U_b}{\partial x} + \frac{\left(\frac{h^2}{2}\right)}{2}\frac{\partial^2 U_b}{\partial x^2} - \cdots$$

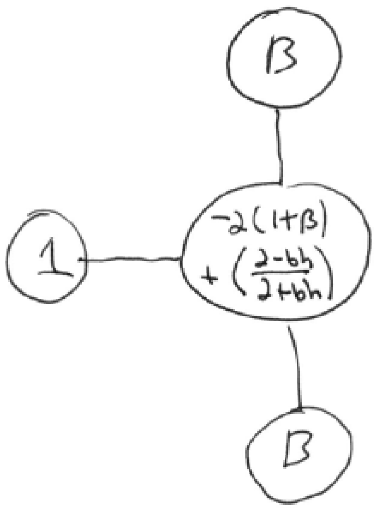
$$U_* + U_\bullet = 2U_b + \frac{h^2}{4}\frac{\partial^2 U_b}{\partial x^2}$$

solve for $U_b$ $\Rightarrow U_b = \frac{U_* + U_\bullet}{2} + O(h^2)$

# Finite Difference Method

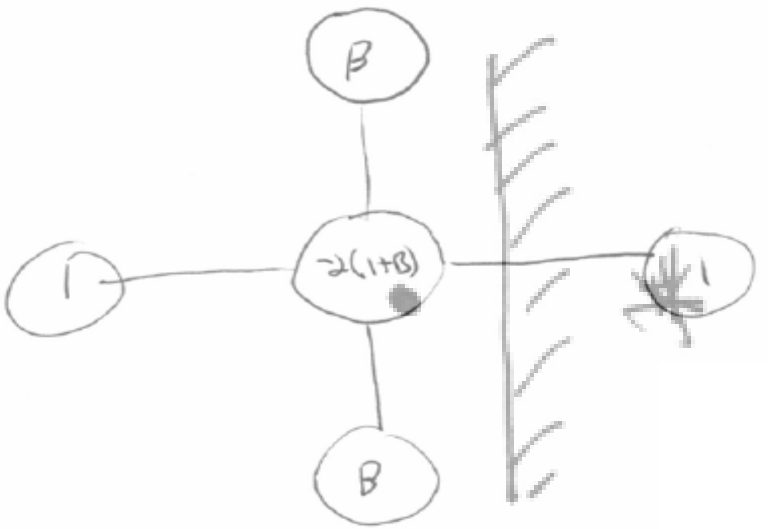
BC expression becomes: $\Rightarrow \dfrac{u_* - u_0}{h} + b\left(\dfrac{u_* + u_0}{2}\right) = a$

$$u_* = \frac{2ah}{2+bh} + \left(\frac{2-bh}{2+bh}\right) u_0$$




$$= g_{ij} h^2 - \frac{2ah}{2+bh}$$

# Finite Difference Method

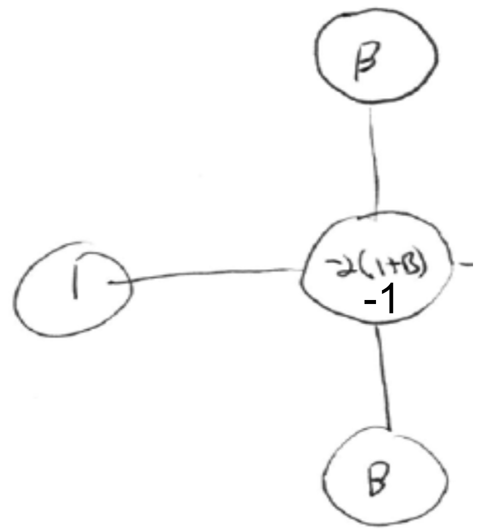



$u_b$=#

Type I

$\left(\frac{u_* + u_.}{2}\right) = u_b$

$$u_* = 2u_b - u.$$



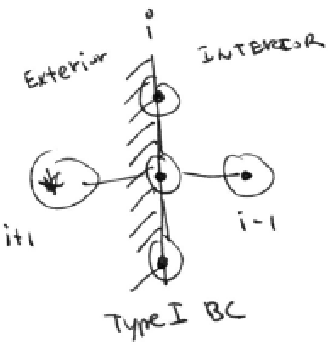

$= -2u_b$

# Finite Difference Method

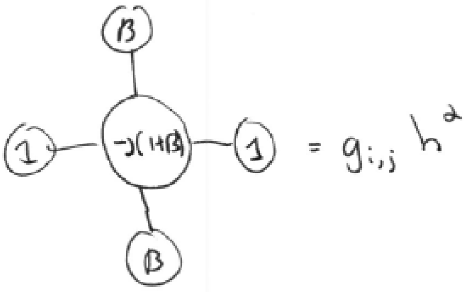
- Note that in case of Type I BC, we don't use the PDE at boundary NODE ... stop assembling PDE "one node in" from boundary

- We can use this equation to determine flux information ... once $u$ is determined, can reconstruct $\nabla u \cdot \hat{n}$ via the unused boundary molecule

# Finite Difference Method




use



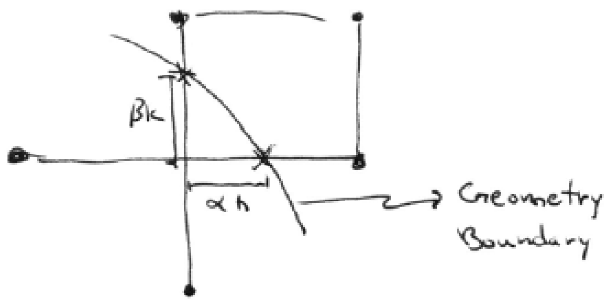

to solve for $u_*$ (shadow node value)

$$u_* = g_{i,j}h^2 \underbrace{- \beta u_{i,j+1} - \beta u_{i,j-1} + 2(1+\beta)u_{i,j}}_{\text{come from BC}} \underbrace{- u_{i-1,j}}_{\text{comes from solution}}$$

then $\frac{\partial u}{\partial x} = \nabla u \cdot \hat{n} = \frac{u_* - u_\circ}{2h} \Rightarrow$ Flux

# Finite Difference Method

What about a situation like



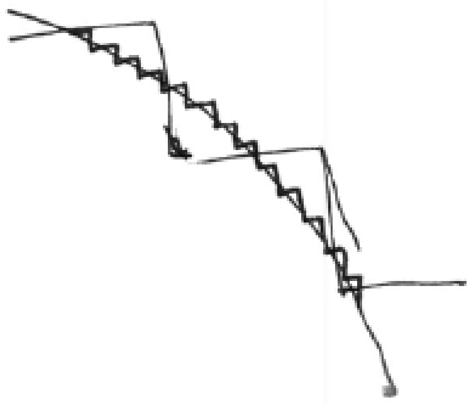

can write PDE molecule on uneven mesh (lose accuracy)
ok if Type I BC... simply replace $u_*$ with known value but if Type II BC... must approximate $\frac{\partial u_*}{\partial n}$ as difference expression involving internal points... can get very messy

# Finite Difference Method

Alternatively ... "stair step" the boundary through local grid refinement or uniform refinement

- then everything proceeds as usual

- necessary to translate BCs from geometry space to grid space but can be done



# Finite Difference Method

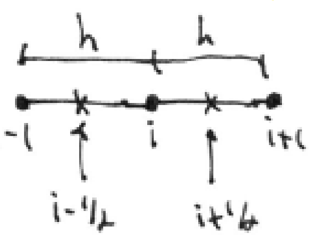
General Elliptic Eqn

$$\underbrace{\frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right)} + \frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = g$$

common to see this term differenced as . . .

$$\frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right) = \frac{1}{h}\left[a_{i+1/2}\left(\frac{u_{i+1}-u_i}{h}\right) - a_{i-1/2}\left(\frac{u_i-u_{i-1}}{h}\right)\right]$$

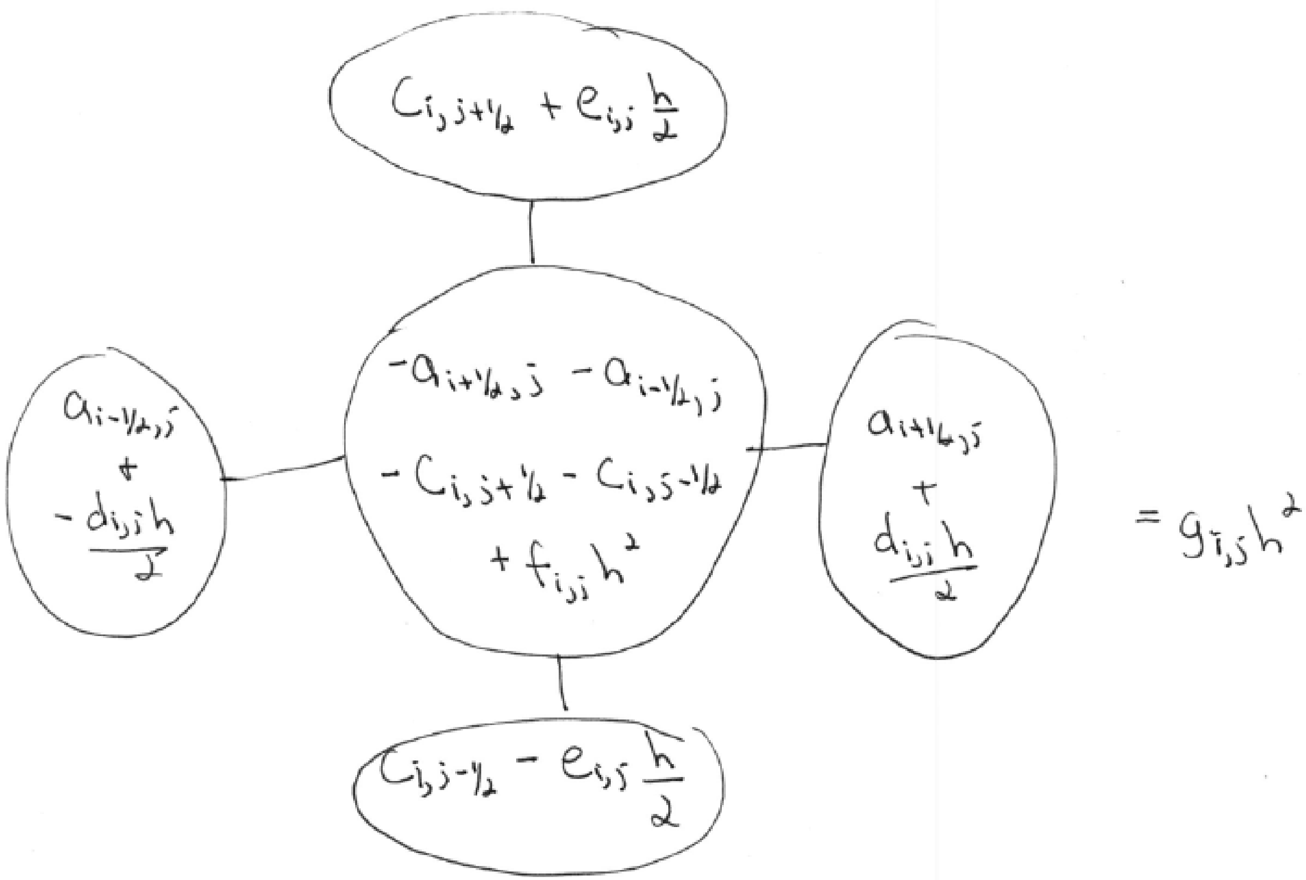
where:




coefficient evaluated "in between" nodes

# Finite Difference Method

Molecule:



# FDM Modeling Review

# FDM Modeling

## Modeling With PDEs

- Where do I start?
  - First Principles

e.g. Gauss's Law

integral form of Gauss's Law

$$\iint \vec{E}\cdot\hat{n}\, ds = \frac{1}{\varepsilon_0} q$$

enclosed charge

$$q = \iiint_V \rho \, d\tau$$

charge density

use divergence theorem

$$\iiint_V (\nabla\cdot\vec{E})\, d\tau = \iiint_V \left(\frac{\rho}{\varepsilon_0}\right) d\tau$$

# FDM Modeling

since this relationship holds for any volume the integrands must be equal ∴

$$\nabla \cdot \vec{E} = \frac{1}{\varepsilon_o} \rho$$

differential form of Gauss's Law

turns out that

$$\vec{E} = -\nabla V$$

potential

so

$$\nabla \cdot \vec{E} = \nabla \cdot (-\nabla V) = \rho / \varepsilon_o$$

$$-\nabla^2 V = \rho / \varepsilon_o$$

Poisson's Equation

is no internal charge

$$-\nabla^2 V = 0$$

$$\nabla^2 V = 0$$

Laplace's Equation

# FDM Modeling

or perhaps relate E through Ohm's Law

$$\vec{J} = \sigma(\vec{E} + \underbrace{\vec{V} \times \vec{B}}_{\text{neglect usually}})$$

$$\vec{J} = \sigma\vec{E} = -\sigma\nabla V$$

$\therefore$ if apply conservation with respect to current flow

$$\nabla \cdot \vec{J} = 0$$

$$\nabla \cdot (-\sigma \nabla V) = 0$$

no current source

# FDM Modeling

- Pick method of solution ... how about FDM?

- Start with PDE

$$\nabla \cdot (-\sigma \nabla V) = 0$$

expand

$$\nabla \cdot \left( -\sigma \begin{bmatrix} \frac{\partial V}{\partial x} \hat{i} \\ \frac{\partial V}{\partial y} \hat{j} \end{bmatrix} \right) = 0$$

$$\frac{\partial}{\partial x}\underbrace{\left(-\sigma \frac{\partial V}{\partial x}\right)}_{\text{let} = F} + \frac{\partial}{\partial y}\underbrace{\left(-\sigma \frac{\partial V}{\partial y}\right)}_{\text{let} = G} = 0$$

# FDM Modeling

$$\frac{\partial}{\partial x}\underbrace{\left(-\theta\frac{\partial V}{\partial x}\right)}_{\text{let} = F} + \frac{\partial}{\partial y}\underbrace{\left(-\theta\frac{\partial U}{\partial y}\right)}_{\text{let} = G} = 0$$

expand using FD technique @ half-grid point

$$\frac{F_{i+1/2} - F_{i-1/2}}{h} + \frac{G_{j+1/2} - G_{j-1/2}}{k} = 0$$

$$\frac{\left(-\theta_{i+1/2}\frac{\partial V}{\partial x}\Big|_{i+1/2}\right) - \left(-\theta\frac{\partial V}{\partial x}\Big|_{i-1/2}\right)}{h} + \frac{\left(-\theta_{j+1/2}\frac{\partial V}{\partial y}\Big|_{j+1/2}\right) - \left(-\theta_{j-1/2}\frac{\partial U}{\partial y}\Big|_{j-1/2}\right)}{k} = 0$$

⋮

expand again

# FDM Modeling

$$\frac{\left(-\theta_{i+1/2,j}\left(\frac{V_{i+1,j}-V_{i,j}}{h}\right)\right)-\left(-\theta_{i-1/2,j}\left(\frac{V_{i,j}-V_{i-1,j}}{h}\right)\right)}{h}$$

$$+$$

$$\frac{\left(-\theta_{i,j+1/2}\left(\frac{V_{i,j+1}-V_{i,j}}{k}\right)\right)-\left(-\theta_{i,j-1/2}\left(\frac{V_{i,j}-V_{i,j-1}}{k}\right)\right)}{k}=0$$

# FDM Modeling

simplify

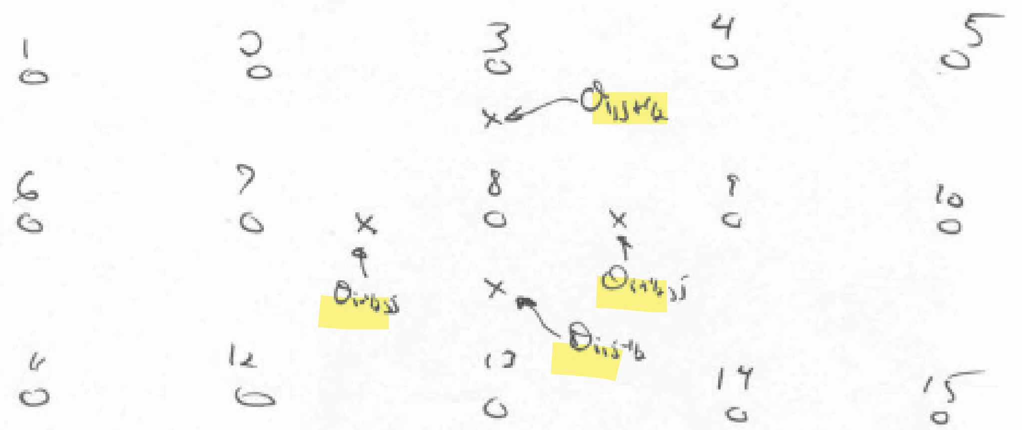
$$\frac{-\theta_{i+1/2,j}V_{i+1,j} + (\theta_{i+1/2,j}+\theta_{i-1/2,j})V_{i,j} - \theta_{i-1/2,j}V_{i-1,j}}{h^2} + \frac{-\theta_{i,j+1/2}V_{i,j+1} + (\theta_{i,j+1/2}+\theta_{i,j-1/2})V_{i,j} - \theta_{i,j-1/2}V_{i,j-1}}{k^2} = 0$$

on equal grid, $h=k$, multiply through by spatial step size..

$$-\theta_{i+1/2,j}V_{i+1,j} - \theta_{i-1/2,j}V_{i-1,j} - \theta_{i,j+1/2}V_{i,j+1} - \theta_{i,j-1/2}V_{i,j-1} + (\theta_{i+1/2,j}+\theta_{i-1/2,j}+\theta_{i,j+1/2}+\theta_{i,j-1/2})V_{i,j} = 0$$

# FDM Modeling

GRID




So for equation (8)

$$-\theta_{i+1/2,j} V_9 - \theta_{i-1/2,j} V_7 - \theta_{i,j+1/2} V_3 - \theta_{i,j-1/2} V_{13} + \left(\theta_{i+1/2,j} + \theta_{i-1/2,j} + \theta_{i,j+1/2} + \theta_{i,j-1/2}\right) V_8 = 0$$

# FDM Modeling

→ I could write similar equations for all terms.. All PDE gives you is the functional relationship between neighborhood of values that approximately satisfies PDE expression.

→ Boundary Conditions Give Specificity

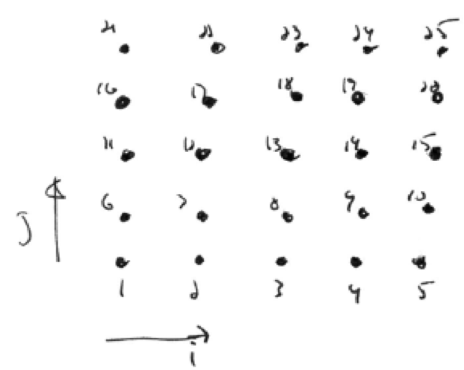
→ hence why these problems are called Boundary Valued Problems

# FDM Matrix Structure

# FDM Matrix Structure

- Need a mapping between $(i,j)$ template location and $(i,j)$ matrix entry in $A$ ... assign a unique number to each mesh point ... generates pentadiagonal structure provided some "natural" ordering is used

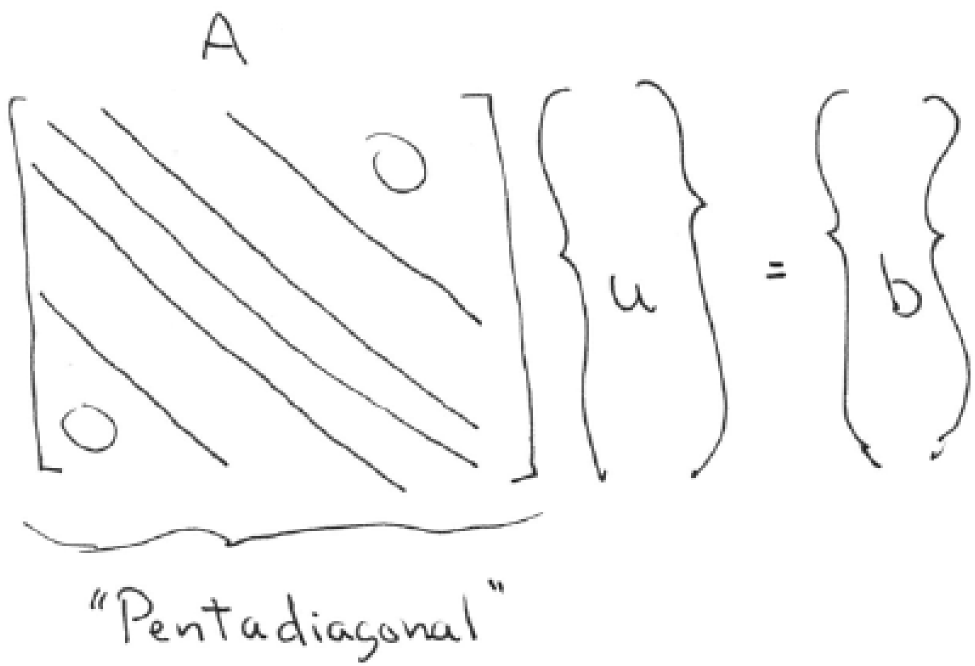
e.g




- each $u_{i,j}$ maps to a unique column in $A$
- each $(i,j)$ template "center" maps to unique row in $A$

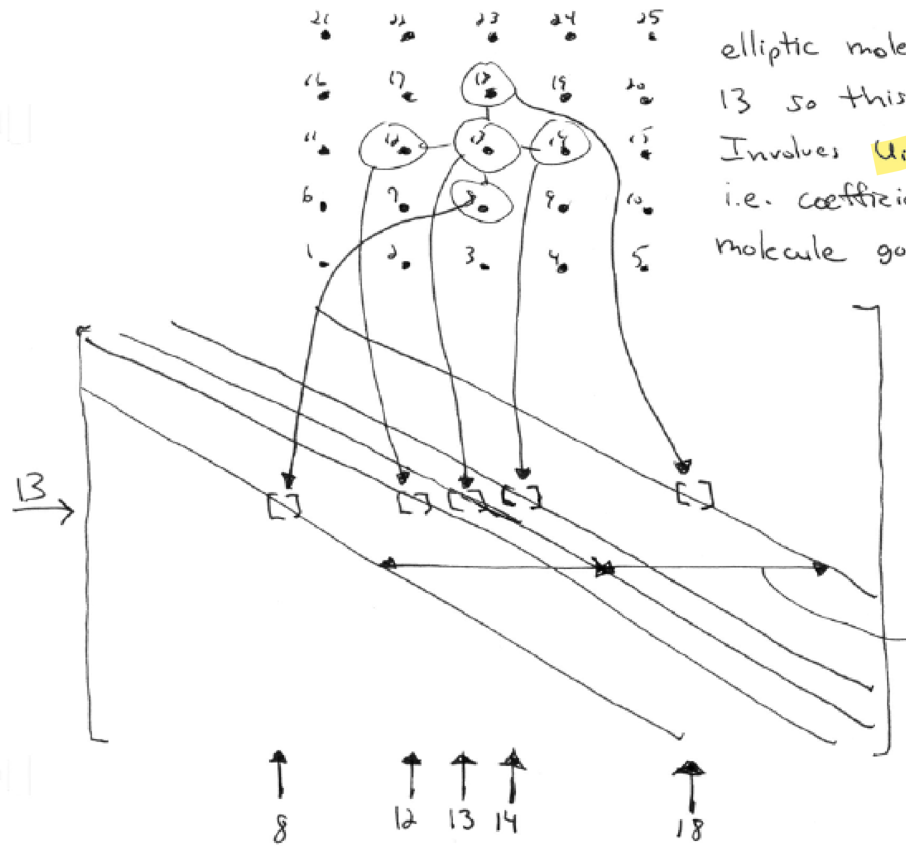
# Finite Difference Method

- Coupled set of equations can be put in matrix form:



# FDM Matrix Structure

elliptic molecule "centered" at 13 so this eqn for row 13.
Involves $u_{i,j}$ at 12, 13, 14, 8, 18
i.e. coefficients in template molecule go in these rows

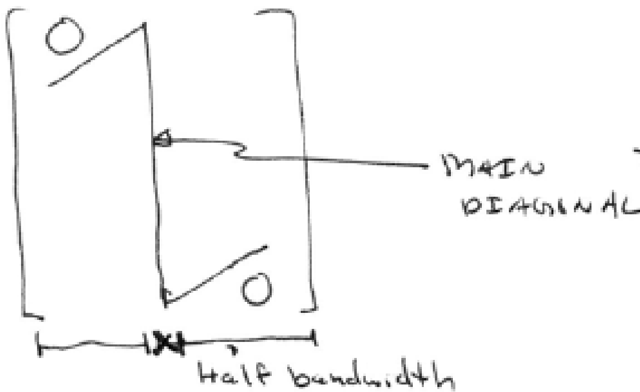



Half-bandwidth in this case = 5

Typical to store as banded matrix...

# FDM Matrix Structure

Typical to store as banded matrix...




e.g

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 4 & 5 & 0 \\ 0 & 6 & 7 & 8 \\ 0 & 0 & 9 & 10 \end{bmatrix} \Rightarrow A_{BW} = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 10 & 0 \end{bmatrix}$$

Half bandwidth = maximum difference between node numbers "connected" through the template

# FDM Matrix Understanding

# FDM Matrix Understanding

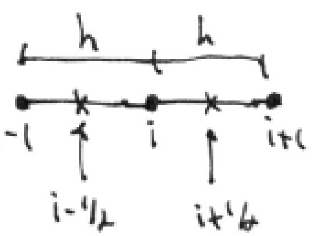
General Elliptic Eqn

$$\underbrace{\frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right)} + \frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = g$$

common to see this term differenced as . . .

$$\frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right) = \frac{1}{h}\left[a_{i+1/2}\left(\frac{u_{i+1}-u_i}{h}\right) - a_{i-1/2}\left(\frac{u_i - u_{i-1}}{h}\right)\right]$$

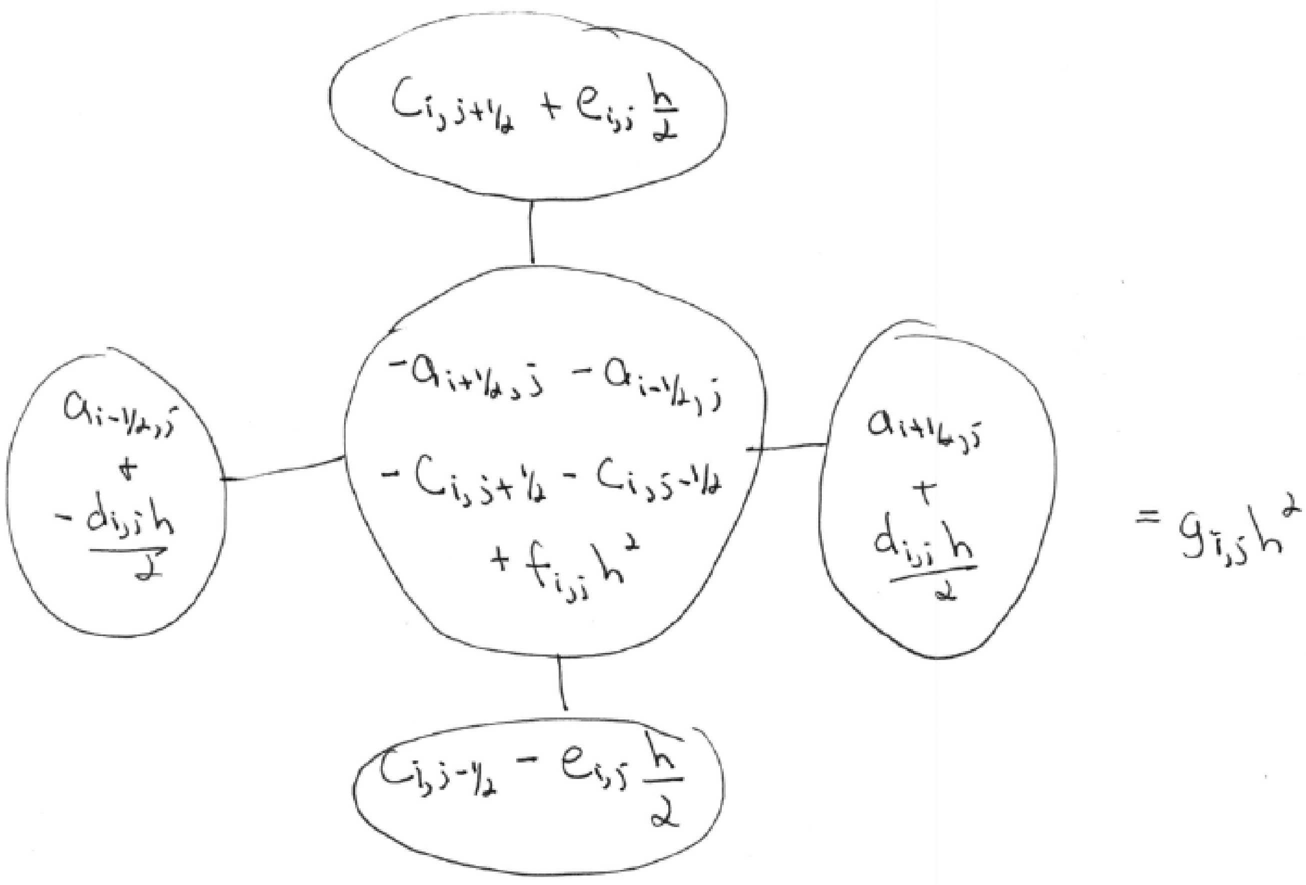
where:




coefficient evaluated "in between" nodes

# FDM Matrix Understanding

Molecule:



# FDM Matrix Understanding

- want to study properties of A ... are important for solving $Au=b$, especially iteratively

write out the molecule as ...

$$\beta_1 u_{i+1,j} + \beta_2 u_{i-1,j} + \beta_3 u_{i,j+1} + \beta_4 u_{i,j-1} - \beta_0 u_{i,j} = g_{i,j} h^2$$

where

$$\beta_1 = a_{i+1/2,j} + d_{i,j} h/2$$

$$\beta_2 = a_{i-1/2,j} - d_{i,j} h/2$$

$$\beta_3 = c_{i,j+1/2} + e_{i,j} \frac{h}{2}$$

$$\beta_4 = c_{i,j-1/2} - e_{i,j} h/2$$

$$\beta_0 = a_{i+1/2,j} + a_{i-1/2,j} + c_{i,j+1/2} + c_{i,j-1/2} - f_{i,j} h^2$$

# FDM Matrix Understanding

$$\beta_1 u_{i+1,j} + \beta_2 u_{i-1,j} + \beta_3 u_{i,j+1} + \beta_4 u_{i,j-1} - \beta_0 u_{i,j} = g_{i,j} h^2$$

$$\beta_0 = a_{i+1/2,j} + a_{i-1/2,j} + c_{i,j+1/2} + c_{i,j-1/2} - f_{i,j} h^2$$

$$\beta_1 = a_{i+1/2,j} + d_{i,j} h/2$$

$$\beta_2 = a_{i-1/2,j} - d_{i,j} h/2$$

$$\beta_3 = c_{i,j+1/2} + e_{i,j} \frac{h}{2}$$

$$\beta_4 = c_{i,j-1/2} - e_{i,j} h/2$$

- we note if $a>0$, $c>0$, $f \leq 0$

- then $\beta_0 > 0$ and $\beta_1 \to \beta_4$ can be made positive by choosing $h$ small enough

  i.e. $0 < h < \min \left\{ \frac{2a_{i\pm 1/2,j}}{|d_{i,j}|}, \frac{2c_{i,j\pm 1/2}}{|e_{i,j}|} \right\}$

- $\beta_0 \geq \sum_{i=1}^{4} \beta_i$

# FDM Matrix Understanding

Conclude: $Au=b$, $A$ with elements $\alpha_{i,j}$ has the properties

i) $\alpha_{ii} > 0$, $\alpha_{ij} \leq 0$ for $i \neq j$ (or the reverse)

ii) $|\alpha_{ii}| \geq \sum_{i \neq j} |\alpha_{ij}|$

In ii) get strict inequality if $f_{ij} < 0$

# FDM Matrix Understanding

Conclude: $Au = b$, $A$ with elements $\alpha_{i,j}$ has the properties

i) $\alpha_{ii} > 0$, $\alpha_{ij} \leq 0$ for $i \neq j$ (or the reverse)

ii) $|\alpha_{ii}| \geq \sum_{i \neq j} |\alpha_{ij}|$

diagonal dominance

(weak*) $$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad \text{for all } i,$$

In ii) get strict inequality if $f_{ij} < 0$

# FDM Matrix Understanding

Conclude: $Au = b$, $A$ with elements $\alpha_{i,j}$ has the properties

i) $\alpha_{ii} > 0$, $\alpha_{ij} \leq 0$ for $i \neq j$ (or the reverse)

ii) $|\alpha_{ii}| \geq \sum_{i \neq j} |\alpha_{ij}|$

In ii) get strict inequality if $f_{ij} < 0$

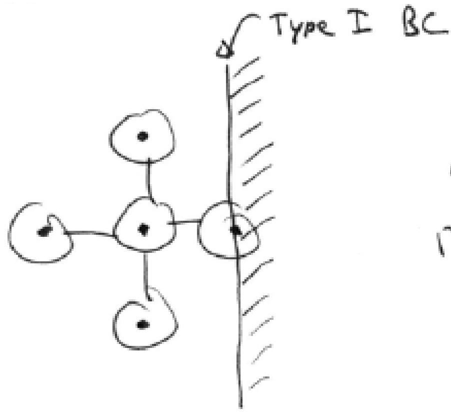
<u>or</u>

get strict inequality for some "i" if $f_{ij} = 0$ and we have Type I BCs

... what does this last item mean?

# FDM Matrix Understanding

e.g.




$B_1$ does not appear in matrix A so clearly

$$B_0 > \sum_{i=1}^{4} B_i$$

- A is diagonally dominant ... "not strict sense" though ... generally good news for iterative solvers

- can prove classical iterative methods converge in this case

# FDM Matrix Understanding

Aside: may seem restrictive requiring $a>0, c>0, f\leq 0$

- but must have a + c same sign and most physical problems have positive coefficients

- $f<0$ is more restrictive (clearly $f=0$ is common!)

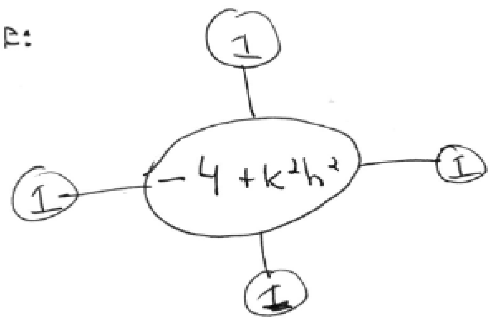
  ↳ clear cases exist with $f>0 \Rightarrow$ Helmholtz Eqn.
  $\nabla^2 u + k^2 u = 0$

F.D. $\Rightarrow \quad \delta_x^2 u_{i,j} + \delta_y^2 u_{i,j} + k^2 h^2 u_{i,j} = 0$

# FDM Matrix Understanding

F.D. $\Rightarrow \delta_x^2 u_{i,j} + \delta_y^2 u_{i,j} + k^2h^2 u_{i,j} = 0$

MOLECULE:




Electromagnetic waves traveling @ frequency $\omega$, 'c' is the wave speed.

$$k^2 = \frac{\omega^2}{c^2} \qquad c = \lambda f \qquad \omega = 2\pi f$$

$$k = 2\pi/\lambda$$

Diagonal Dominance lost as $h \to 0$ .... only have it when $k^2h^2 - 4 > 4 \Rightarrow k^2h^2 > 8$

$$k^2h^2 > 8$$
$$kh > \sqrt{8}$$
$$\left(2\pi/\lambda\right) h > \sqrt{8}$$
$$\lambda/h < 2\pi/\sqrt{8}$$

# FDM Solution Methods

# FDM Solution Methods

How do I solve?

→ Direct Methods: represent FD expressions as equations in matrix and solve using factorization or matrix inverse techniques

↳ LU Decomposition + Back Substitution

↳ Gauss Elimination

→ Point/Block Iterative Methods: take FD equations, begin with initial guess, update guess. Over the course of iterations, process converges to solution...

# FDM Solution Methods

Classical Point Iteration Methods
(For Solving System $Au=b$)

- start w/ $u^0_{i,j}$ ... "Initial Guess" for each $(i,j) \ni \underset{\sim}{u}^0$
update solution repeatedly point-by-point

- Need stopping criterion

Typical : $\| \underset{\sim}{u}^{\ell+1} - \underset{\sim}{u}^{\ell} \| < \epsilon$ Absolute

Better : $\dfrac{\| \underset{\sim}{u}^{\ell+1} - \underset{\sim}{u}^{\ell} \|}{\| \underset{\sim}{u}^{\ell+1} \|} < \epsilon$ Relative

Relative criterion accounts for solution size!

# FDM Solution Methods

- Jacobi

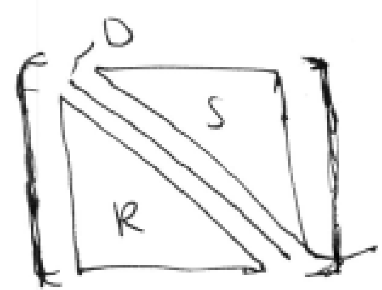
$$u_{i,j}^{\ell+1} = \frac{1}{\beta_0}\left[\beta_1 u_{i+1,j}^{\ell} + \beta_2 u_{i-1,j}^{\ell} + \beta_3 u_{i,j+1}^{\ell} + \beta_4 u_{i,j-1}^{\ell} - h^2 g_{i,j}\right]$$

$\ell \equiv$ iteration # ; $u_{i,j}^{0}$ "initial guess"

- simple
- 2 arrays : $u^{\ell}, u^{\ell+1}$ of length total # of nodes
- get one new value at a time
- order of calculation does not matter

common to analyze from matrix perspective (but don't build in practice)

$$[A] = \underbrace{[R]}_{\text{below}} + \underbrace{[D]}_{\text{diagonal}} + \underbrace{[S]}_{\text{above}} \Rightarrow$$



# FDM Solution Methods

then

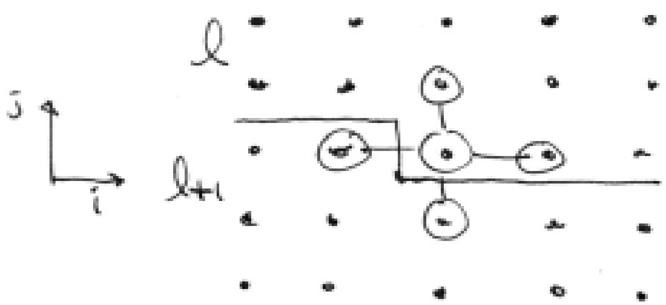
$$[D]\{u\}^{\ell+1} = -[R]\{u\}^{\ell} - [S]\{u\}^{\ell} + \{b\}$$

$$\{u\}^{\ell+1} = \underbrace{-[D]^{-1}[R+S]}_{G_J \,\equiv\, \text{Jacobi Iteration matrix}}\{u\}^{\ell} + [D]^{-1}\{b\}$$

# FDM Solution Methods

- Gauss - Seidel
  - use latest info
  - order does make a difference

e.g.

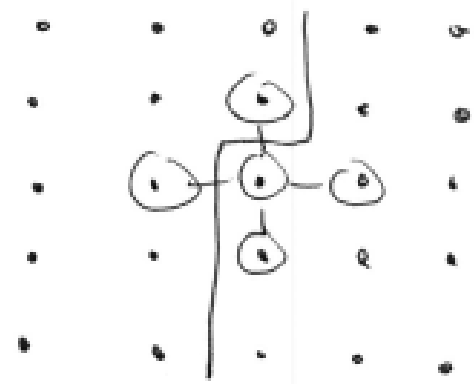



proceed row-wise

$$u_{i,j}^{l+1} = \frac{1}{\beta_0}\left[\beta_1 u_{i+1,j}^{l} + \beta_2 u_{i-1,j}^{l+1} + \beta_3 u_{i,j+1}^{l} + \beta_4 u_{i,j-1}^{l+1} - h^2 g_{i,j}\right]$$

# FDM Solution Methods

proceed column-wise
(downward



$l+1$ $\quad$ $l$

$$u_{i,j}^{l+1} = \frac{1}{\beta_0}\left[\beta_1 u_{i+1,j}^{l} + \beta_2 u_{i-1,j}^{l+1} + \beta_3 u_{i,j+1}^{l+1} + \beta_4 u_{i,j-1}^{l} - g_{i,j} h^2\right]$$

# FDM Solution Methods

$$\{u\}^{l+1} = \underbrace{-[D]^{-1}[R+S]}_{G_J}\{u\}^{l} + [D]^{-1}\{b\}$$

$G_J \equiv$ Jacobi Iteration matrix

In matrix form:

$$[R+D]\{u\}^{l+1} = -[S]\{u\}^{l} + \{b\}$$

$$\{u\}^{l+1} = \underbrace{-[R+D]^{-1}[S]}_{G_{GS}}\{u\}^{l} + [R+D]^{-1}\{b\}$$

$G_{GS} \equiv$ Gauss-Seidel Iteration matrix

# Point Iterative Solution Convergence

– Have general form

$$u^{\ell+1} = G u^{\ell} + r$$

Primary result: Spectral Radius $= \rho(G) < 1$ guarantees it

Max eigenvalue

Proof has two key observations:

$\rightarrow e^{\ell+1} = G e^{\ell}$ where $e^{\ell+1} = u^{\ell+1} - u$

Recall:

$$\begin{aligned} u^{\ell+1} &= G u^{\ell} + r \\ u &= G u + r \\ \hline u^{\ell+1} - u &= G(u^{\ell} - u) \end{aligned}$$

# Point Iterative Solution Convergence

→ $e^0$ can be expressed in terms of eigenvectors of $G$:

$$e^0 = \sum_{i=1}^{N} b_i v_i \qquad v_i = \text{eigenvector of } G$$

then $e^1 = Ge^0 = \sum_{i=1}^{N} b_i G v_i = \sum_{i=1}^{N} b_i \lambda_i v_i$

$\vdots$

iterate

$$e^\ell = \sum_{i=1}^{N} b_i \lambda_i^\ell v_i$$

exponent in this case

If we want $\lim_{\ell \to \infty} e^\ell = 0$, then need $|\lambda_i| < 1$

So eigenvalues of $G$ are critical!!

**e.g. [A]{v}=λ{v}**

$$e^2 = GGe^0 = \sum_{i=1}^{N} b_i GGv_i$$

$$e^2 = \sum_{i=1}^{N} b_i \lambda_i G v_i = \sum_{i=1}^{N} b_i \lambda_i^2 v_i$$

# Point Iterative Solution Convergence

Eigenvalues of Iteration Matrices

- Jacobi: $G_J = -D^{-1}[R+S]$

So $\det(\lambda I - G_J) = 0 \Rightarrow \det(\lambda I + D^{-1}(R+S)) = 0$

$\Rightarrow \det(D^{-1}\lambda D + D^{-1}(R+S)) = \det D^{-1}(\lambda D + R + S) = 0$

$= \det D^{-1} \det(\lambda D + R + S) = 0$

$\therefore \boxed{\det(\lambda D + R + S) = 0}$

- Gauss-Seidel: $G_{GS} = -(D+R)^{-1} S$

same form as above... replace: $D$ w/ $D+R$

$R+S$ w/ $S$

$\therefore \boxed{\det(\lambda D + \lambda R + S) = 0}$

# Point Iterative Solution Convergence

⇒ Recall that strict diagonal dominance guarantees convergence... but we don't have this for general elliptic FD molecules...

- have instead for A:

i) $\alpha_{ii} > 0$, $\alpha_{ij} \leq 0 \quad i \neq j$

ii) $\alpha_{ii} \geq \sum_{\substack{j=1 \\ i \neq j}}^{N} |\alpha_{ij}|$ w/ strict inequality for some "i"

($f_{ij} < 0$ or $f_{ij} = 0$ w/ Type I)

⇒ can still show that this will produce $\rho(G) < 1$

# Point Iterative Solution Convergence

- Determining $\rho(G)$ in practice
  - can compute w/ Power Method requires us to actually construct G → (see B+F, Num. Anal.)
  - estimate during iteration

$$\rho \approx \frac{\|\delta^{\ell}\|}{\|\delta^{\ell-1}\|} \quad \text{where } \delta^{\ell} = u^{\ell} - u^{\ell-1}$$

Theoretical basis:

$$\begin{aligned} u^{\ell+1} &= G u^{\ell} + r \\ u^{\ell} &= G u^{\ell} + r \\ \hline \delta^{\ell+1} &= G \delta^{\ell} \end{aligned}$$

same as before ... expand $\delta^{1} = \sum c_i v_i$

# Point Iterative Solution Convergence

- Rate of convergence
  - fundamentally governed by $\rho(G)$
  - for "large" $\ell$ $\Rightarrow e^{\ell+1} \simeq \rho(G)e^{\ell}$

$$\therefore \quad \frac{\|e^{\ell}\|}{\|e^{\ell+1}\|} \simeq 1/\rho(G)$$

- $\log \rho(G)$ indicates # digits by which each iteration reduces the error
  - so for error reduction by factor K

$$\|e\|^{\ell+m} = \rho^m \|e\|^{\ell}, \quad K = \rho^m \Rightarrow m > \frac{\log K}{\log(\rho)}$$

# Point Iterative Solution Convergence

- Rate of convergence
  - fundamentally governed by $\rho(G)$
  - for "large" $\ell$ $\Rightarrow$ $e^{\ell+1} \simeq \rho(G) e^{\ell}$

$$\therefore \frac{\|e^{\ell}\|}{\|e^{\ell+1}\|} \simeq 1/\rho(G)$$

- $\log \rho(G)$ indicates # digits by which each iteration reduces the error

↳ so for error reduction by factor $K$

$$\|e\|^{\ell+m} = \rho^{m}\|e\|^{\ell}, \; K = \rho^{m} \Rightarrow m > \frac{\log K}{\log(\rho)}$$

e.g. $\varepsilon_o^{l}=6$, $\rho_1=0.1$, $\rho_2=0.9$
$\varepsilon^{l+1}=\rho_1\varepsilon_o^{l}=0.6$ , $\varepsilon^{l+1}=\rho_2\varepsilon_o^{l}=5.4$

$\log_{10}(0.1)=-1$, $\log_{10}(0.9)=-0.0458$
*In example 1, $\log_{10}(\rho)$ says I move one digit, one decimal place
*In example 2, one iteration moves a fraction of a digit, in order to move 1 digit, need 1/0.0458=21.83 iterations at that spectral radius.