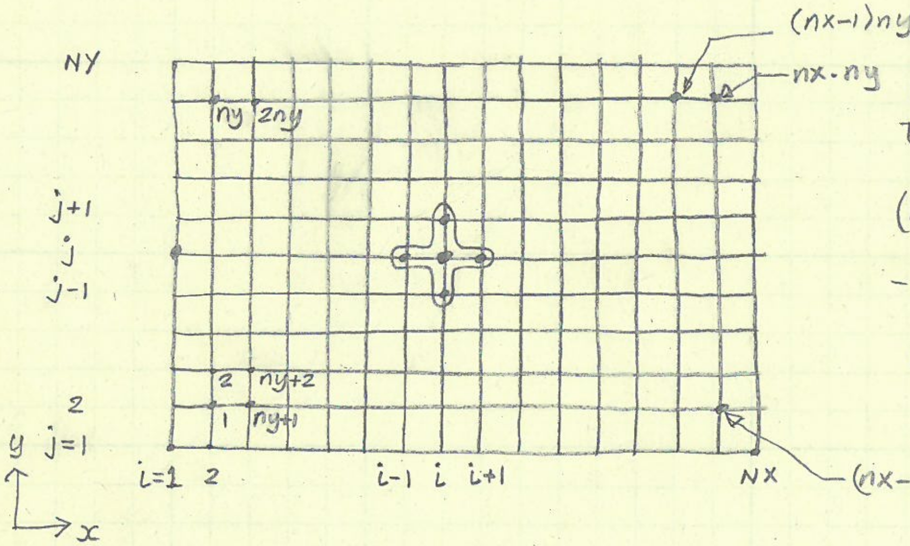
SOLUTION OF LAPLACE / POISSON EQUATION ON A RECTANGULAR DOMAIN



Nodal values $U_{i,j} = u((i-1)\Delta x, (j-1)\Delta y)$

$(NX \cdot NY)$ unknown $U_{i,j}$'s

– need $(NX \cdot NY)$ equations

Let $nx = NX-2$

$ny = NY-2$

The Domain is discretized into $(NX \cdot NY)$ grid points. Writing the finite difference approx. to $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)$ at $(x,y) = ((i-1)\Delta x, (j-1)\Delta y)$ or node (i,j) :

$$(A) \quad \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{\Delta x^2} + \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{\Delta y^2} = f_{i,j}$$

- writing (A) at each interior nodes generates $(NX-2)(NY-2)$ equations.
- The additional $2NX + 2NY - 4$ equations result from B.C. on the 4 edges. One B.C. must be specified at each boundary point. These may be of Type I (u specified), II ($\frac{\partial u}{\partial x}$ or $\frac{\partial u}{\partial y}$ specified) or III ($\alpha u + \beta \frac{\partial u}{\partial y}$ or $\beta \frac{\partial u}{\partial x}$ specified) – normal derivatives to the boundary are involved at any boundary point. At least part of the boundary should have type I B.C.
- Implementation of B.C. consider boundary point $(1,j)$ as an example, with

$\alpha = 1, \beta = 0$: TYPE I

$\alpha = 0, \beta \neq 0$: TYPE II

$\alpha \neq 0, \beta \neq 0$: TYPE III

$$(B) \quad \alpha U_{1,j} + \beta \underbrace{\frac{U_{2,j} - U_{1,j}}{\Delta x}}_{\frac{\partial u}{\partial x} \text{ (normal derivative)}} = \gamma$$

It is customary to assemble a system of equations involving only the interior nodal values of U. To do this, $U_{1,j}$ solved from (B) is used in (A) written at $(2,j)$ (or more generally the B.C. is substituted into the next interior node's (A)).

from (B), $\quad U_{1,j} = \dfrac{(\gamma\Delta x - \beta U_{2,j})}{(\alpha\Delta x - \beta)}$

Use in (A) to get :

$$\frac{U_{3,j}}{\Delta x^2} + \frac{U_{2,j-1} + U_{2,j+1}}{\Delta y^2} + \left(\overset{-\frac{2}{\Delta x^2} - \frac{2}{\Delta y^2}}{\downarrow} + \underset{\substack{\downarrow \\ 0 \text{ for } \beta = 0,\ \alpha = 1 \\ -1 \text{ for } \beta = 1,\ \alpha = 0}}{\frac{-\beta}{(\alpha\Delta x - \beta)\Delta x^2}} \right) U_{2,j} = f_{ij} - \frac{\gamma/\Delta x}{(\alpha\Delta x - \beta)} \qquad (C)$$

Once $U_{2,j}$ is obtained from the solution for all interior nodes, $U_{1,j}$ is obtained from (B).

Similar adjustments are needed at the other nodes adjacent to boundary nodes. Note some changes in sign also. e.g.

$$\alpha\, U_{NX,j} + \beta\, \frac{U_{NX,j} - U_{NX-1,j}}{\Delta x} = \gamma \quad \text{etc.}$$

Now, let $nx = NX - 2$, $ny = NY - 2$ → we have $nx \cdot ny$ equations in $nx \cdot ny$ unknowns. Let us number the nodes sequentially, along y.

$k = 1, 2, \ldots ny;\ ny+1, \ldots 2ny;\ \ldots\ldots;\ (nx-1)ny+1 \ldots\ldots nx \cdot ny.$

The node number for (i,j) in this scheme would be $(i-2) \cdot ny + (j-1)$

The equation at any node number k involves the U at nodes $k-1$ (down), $k+1$ (up), $(k-ny)$ left and $(k+ny)$ right ; with one missing neighbor on the first and last interior rows/columns.

→ MATLAB CODE FOR SOLVING IN MATRIX FORM – careful with indexing (first type or no-flux boundaries)

↳ (C) has $-\dfrac{2}{\Delta x^2} - \dfrac{2}{\Delta y^2}$, right side has $f_{i,j} - \dfrac{\overset{\text{known value}}{U_{1,j}}}{\Delta x^2}$

no flux: (C) has $-\dfrac{1}{\Delta x^2} - \dfrac{2}{\Delta y^2}$, right side has f_{ij} $(\gamma = 0)$ OR $f_{ij} + \gamma/\Delta x$

INTERIOR NODE EQN. AT NODE (2,j)

$$(A)\quad \left(\frac{1}{\Delta x^2}\right)U_{1,j} + \left(\frac{1}{\Delta y^2}\right)U_{2,j-1} + \left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)U_{2,j} + \left(\frac{1}{\Delta x^2}\right)U_{3,j} + \left(\frac{1}{\Delta y^2}\right)U_{2,j+1} = f_{2,j}$$

BDRY. CONDITION AT NODE (1,j) :

$$\alpha\, U_{1,j} + \beta\,\frac{U_{2,j} - U_{1,j}}{\Delta x} = \gamma$$

SOLVE FOR $U_{1,j}$ IN TERMS OF $U_{2,j}$: $U_{1,j} = \dfrac{\gamma\Delta x - \beta U_{2,j}}{(\alpha\Delta x - \beta)}$

a) TYPE I BDRY. COND.

$\alpha = 1,\ \beta = 0,\ \gamma =$ GIVEN VALUE OF $U_{1,j}$

subst. into (A) - eliminate $U_{1,j}$ as an unknown on the left side by taking it to the right side.

$$\text{----} + \left(\frac{1}{\Delta y^2}\right)U_{2,j-1} + \left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)U_{2,j} + \left(\frac{1}{\Delta x^2}\right)U_{3,j} + \left(\frac{1}{\Delta y^2}\right)U_{2,j+1} = f_{2,j} - \frac{\overset{\text{GIVEN}}{U_{1,j}}}{\Delta x^2}$$

MODIFIED FORM OF EQN. (A) AT NODE (2,j) ACCOUNTS FOR TYPE I B.C.

b) TYPE II BDRY. COND.

say $\alpha = 0,\ \beta = 1,\ \gamma =$ GIVEN VALUE OF GRADIENT $= \dfrac{U_{2,j} - U_{1,j}}{\Delta x}$

subst. into (A) - eliminate $\dfrac{-U_{1,j} + U_{2,j}}{\Delta x^2} = \dfrac{-\gamma}{\Delta x}$ from left side and take to right side.

$$\text{---} + \left(\frac{1}{\Delta y^2}\right)U_{2,j-1} + \left(\frac{-1}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)U_{2,j} + \left(\frac{1}{\Delta x^2}\right)U_{3,j} + \left(\frac{1}{\Delta y^2}\right)U_{2,j+1} = f_{2,j} + \frac{\gamma}{\Delta x}$$

MODIFIED FORM OF EQN. (A) AT NODE (2,j) ACCOUNTS FOR TYPE II B.C.

SIMILAR ADJUSTMENTS AT NODES $(NX-1, j)$ - RIGHT EDGE $\longrightarrow$ NOTE TYPE II B.C. WOULD BE $\dfrac{U_{NX,j} - U_{NX-1,j}}{\Delta x} = \gamma$

$(i,2)$ - BOTTOM EDGE, $(i, NY-1)$ - TOP EDGE.

$\downarrow$

(SEE TABLE FOR HOW EQN. (A) IS MODIFIED FOR EACH OF THESE CASES)

SO $\dfrac{U_{NX,j} - U_{NX-1,j}}{\Delta x} = \dfrac{+\gamma}{\Delta x}$

Becomes $\dfrac{-\gamma}{\Delta x}$ when taken to right side

SUMMARY OF MATRIX COEFFICIENTS FOR TYPE I & TYPE II B.C.

"MIDDLE - INTERIOR" NODE (STANDARD CASE)

$$\left(\frac{1}{\Delta x^2}\right) U_{i-1,j} + \left(\frac{1}{\Delta y^2}\right) U_{i,j-1} + \left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right) U_{i,j} + \left(\frac{1}{\Delta x^2}\right) U_{i+1,j} + \left(\frac{1}{\Delta y^2}\right) U_{i,j+1} = f_{i,j}$$

LEFT EDGE ($i=2$) INTERIOR NODES

TYPE	$U_{i-1,j}$	$U_{i,j-1}$	$U_{i,j}$	$U_{i+1,j}$	$U_{i,j+1}$	RHS
I	NO TERM					$f_{i,j} - \frac{U_{1,j}}{\Delta x^2}$
II	"		$\left(\frac{-1}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)$			$f_{i,j} + \frac{\gamma}{\Delta x}$

RIGHT EDGE ($i = NX-1$) INTERIOR NODES

TYPE	$U_{i-1,j}$	$U_{i,j-1}$	$U_{i,j}$	$U_{i+1,j}$	$U_{i,j+1}$	RHS
I				NO TERM		$f_{i,j} - \frac{U_{NX,j}}{\Delta x^2}$
II			$\left(\frac{-1}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)$	"		$f_{i,j} - \frac{\gamma}{\Delta x}$

BOTTOM EDGE ($j=2$) INTERIOR NODES

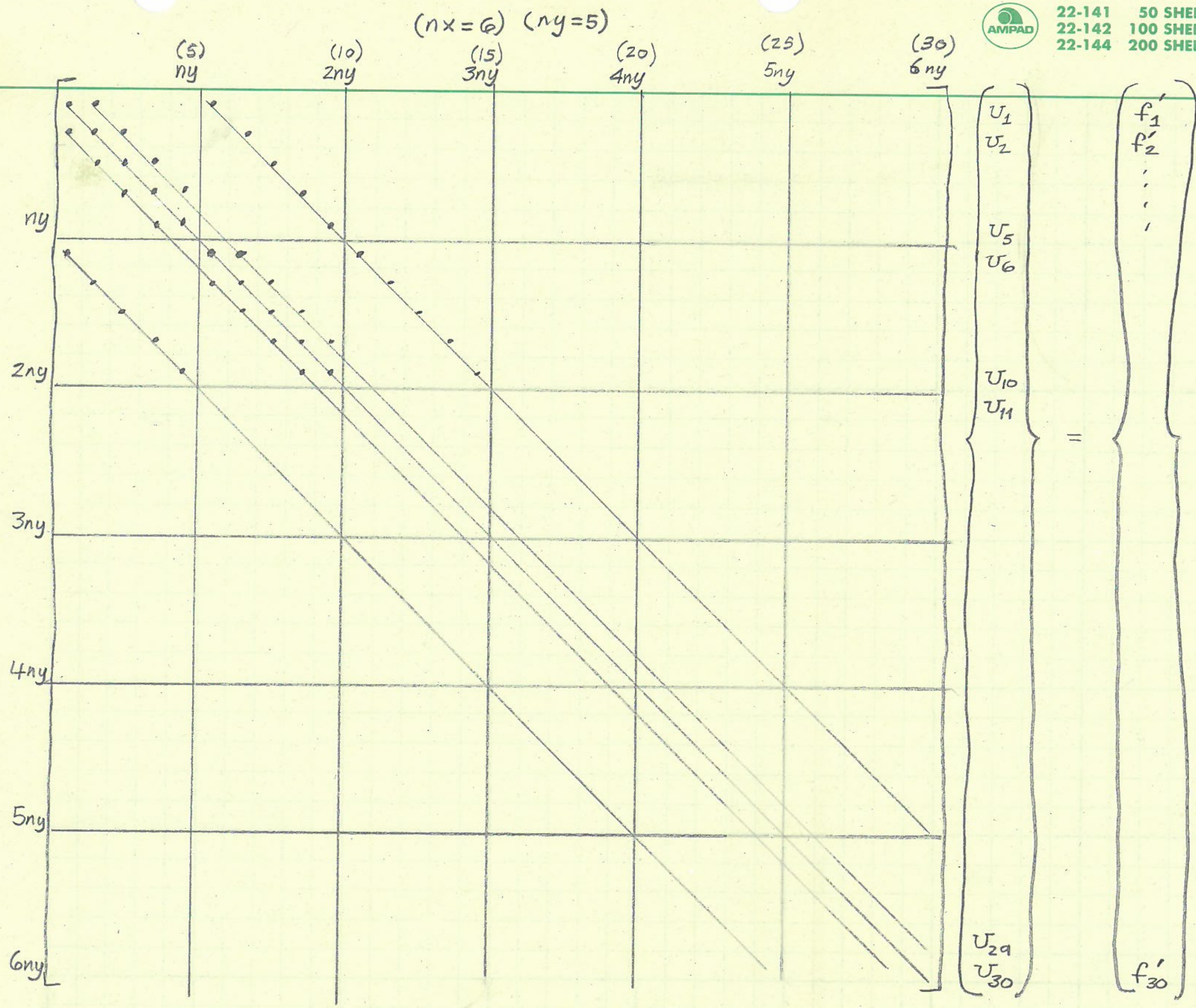
TYPE	$U_{i-1,j}$	$U_{i,j-1}$	$U_{i,j}$	$U_{i+1,j}$	$U_{i,j+1}$	RHS
I		NO TERM				$f_{i,j} - \frac{U_{i,1}}{\Delta y^2}$
II		"	$\left(\frac{-2}{\Delta x^2} + \frac{-1}{\Delta y^2}\right)$			$f_{i,j} + \frac{\gamma}{\Delta y}$

TOP EDGE ($j=NY-1$) INTERIOR NODES

TYPE	$U_{i-1,j}$	$U_{i,j-1}$	$U_{i,j}$	$U_{i+1,j}$	$U_{i,j+1}$	RHS
I					NO TERM	$f_{i,j} - \frac{U_{i,NY}}{\Delta y^2}$
II			$\left(\frac{-2}{\Delta x^2} + \frac{-1}{\Delta y^2}\right)$		NO TERM	$f_{i,j} - \frac{\gamma}{\Delta y}$

NOTE:

- FOR 4 CORNER INTERIOR NODES, NEED 2 ADJUSTMENTS EACH, ONLY 3 TERMS IN MATRIX ROW
- PROBABLY BETTER TO USE SUBSCRIPTS ON γ ALSO FOR VARIABLE FLUXES. WHEN $\gamma=0$ (NO FLUX BDRY.) RIGHT-HAND SIDE DOES NOT CHANGE.
- DONT FORGET $NX = nx+2$, $NY = ny+2$ (NX, NY refer to the real edges of the rectangles)
- FOR INSULATED/NO-FLUX BOUNDARIES, $\alpha = 0$, $\beta = 1$, $\gamma = 0$ → no change in RHS.



$f' = f$ adjusted for B.C.

5-banded matrix, block tridiagonal, SYMMETRIC, NEGATIVE DEFINITE, DIAGONALLY DOMINANT (Diagonal element $\geq$ sum of off-diagonal elements in each row)

```
%this script generates the SPARSE MATRIX needed for
%a matrix-based solution of Laplace equation USING spdiags
%and solves the system of equations
%the grid dimensions are (nx+2)*(ny+2),
%so that the number of interior nodes is nx*ny (more convenient notation)
%the boundary conditions are constant values on the left and right edges,
%with no flux boundaries along the top and bottom edges
%NOTE THAT THE TOTAL NUMBER OF UNKNOWNS IS n=nx*ny, so A is an nxn matrix
%
nx=101;
ny=101;
n=nx*ny;%total number of unknowns
%set first type boundary values on left and right boundary
uleft=zeros(ny,1);
uright=ones(ny,1);
%delx and dely calculated
delx=1/(nx+1);
dely=1/(ny+1);
%put -4 on the main diagonal
vecmain=-2*ones(n,1)/delx^2+-2*ones(n,1)/dely^2;
%put 1 on the first lower diagonal, later modify it
vecdown=ones(n-1,1)/dely^2;
%put 1 on the first upper diagonal, later modify it
vecup=ones(n-1,1)/dely^2;
%put 1 on the far lower diagonal, later modify it
vecleft=ones(n-ny,1)/delx^2;
%put 1 on the far upper diagonal, later modify it
vecright=ones(n-ny,1)/delx^2;
%set up right-hand side vector as zeros, later modify it
rhs=zeros(n,1);
rhs(fix(nx*ny/2))=10000;
%now modify the vectors that define the matrix for nodes adjacent to
boundaries
%left and right boundaries (no change in vectors, only in rhs
for k=1:ny
    rhs(k)=rhs(k)-uleft(k)/delx^2;
    rhs((nx-1)*ny+k)=rhs((nx-1)*ny+k)-uright(k)/delx^2;
end
%for bottom and top boundaries (no flux, so delete -1/dely^2 from main
%diagonal, and zero out vecup for top boundary, vecdown for bottom boundary
for l=0:nx-1
    %bottom boundary - ibot is node number, ibot-1 is the index in the
    %vector vecdown (one less than ibot)
    ibot=l*ny+1;
    if (ibot ~= 1)
        vecdown(ibot-1)=0;
```

```
    end
        vecmain(ibot)=vecmain(ibot)+1/dely^2;
    %top boundary - itop is the node number, the index in vectop is same
    itop=ibot+ny-1;
    if (itop ~= nx*ny)
        vecup(itop)=0;
    end
    vecmain(itop)=vecmain(itop)+1/dely^2;
end
%create sparse matrix S using the spdiags command in MATLAB
%spdiags works a bit non-intuitively. It takes the upper part of
%sub-diagonals and lower part of super-diagonals.
v1=[vecleft;zeros(ny,1)];
v2=[vecdown;0];
v3=vecmain;
v4=[0;vecup];
v5=[zeros(ny,1);vecright];
S=spdiags([v1,v2,v3,v4,v5],[-ny,-1,0,1,ny],n,n);
disp 'sparse matrix'
tic;u=S\rhs;toc
%now construct the uplot vector for contour plotting
uplot=zeros(nx+2,ny+2);
uplot(1,1)=0.0;
uplot(1,2:ny+1)=uleft';
uplot(1,ny+2)=0.0;
for i=1:nx
    for j=1:ny
        index=(i-1)*ny+j;
        iplot=i+1;
        jplot=j+1;
        [index iplot jplot];
        uplot(iplot,jplot)=u(index);
    end
end
for i=2:nx+1
    uplot(i,1)=uplot(i,2);
    uplot(i,ny+2)=uplot(i,ny+1);
end
uplot(nx+2,1)=1.0;
uplot(nx+2,2:ny+1)=uright';
uplot(nx+2,ny+2)=1.0;
x=0:delx:1;
y=0:dely:1;
contourf(x,y,uplot)
```

BASIC ITERATIVE (INDIRECT) METHODS FOR SOLVING $\underset{n\times n}{[A]}\ \underset{n\times 1}{\underline{u}} = \underset{n\times 1}{\underline{r}}$

$$A_{11}u_1 + A_{12}u_2 + \cdots + A_{1n}u_n = r_1$$

$$A_{21}u_1 + A_{22}u_2 + \cdots + A_{2n}u_n = r_2$$

$$\vdots$$

$$A_{n1}u_1 + A_{n2}u_2 + \cdots + A_{nn}u_n = r_n$$

STARTING FROM INITIAL GUESS $\underline{u}^0$, ITERATIVELY REFINE TO FIND $\underline{u}$

I. JACOBI ITERATION:

"old"

$$\text{"new"}\quad u_1^{(m+1)} = (r_1 - A_{12}u_2^{(m)} - A_{13}u_3^{(m)} - \cdots - A_{1n}u_n^{(m)})/A_{11}$$

$$u_2^{(m+1)} = (r_2 - A_{21}u_1^{(m)} - A_{23}u_3^{(m)} - \cdots - A_{2n}u_n^{(m)})/A_{22}$$

etc.

$$\boxed{u_k^{(m+1)} = \left(r_k - \sum_{\substack{\ell=1\\ \ell\neq k}}^{n} A_{k\ell}u_\ell^{(m)}\right)/A_{kk}}$$

for $k = 1$ to n in each iteration

<u>convergence criterion</u>: Let $\underline{e}^{(m+1)} = [A]\underline{u}^{(m+1)} - \underline{r}$, $e_k^{(m+1)} = \sum_{\ell=1}^{n} A_{k\ell}u_\ell^{(m+1)} - r_k$

i) $\max(|\underline{e}^{(m+1)}|) \leq$ tolerance or ii) $\frac{1}{n}\sum_{k=1}^{n}(e_k^{(m+1)})^2 \leq$ tolerance.

- Jacobi iteration converges if $|A_{kk}| \geq \sum_{\substack{\ell=1\\ \ell\neq k}}^{n} |A_{k\ell}|$ for all k

  with strict $>$ in atleast one k.

  "DIAGONALLY DOMINANT" MATRIX. FINITE-DIFFERENCE APPROX. TO LAPLACE EQN. SATISFY THIS CONDITION (AT LEAST ONE BDRY. POINT TYPE I)

II. GAUSS-SEIDEL ITERATION:

MODIFY JACOBI BY USING "new" estimates as they become available.

this "new" estimate is available already

e.g. $$u_2^{(m+1)} = (r_2 - A_{21}u_1^{(m+1)} - A_{23}u_3^{(m)} - \cdots A_{2n}u_n^{(m)})/A_{22}$$

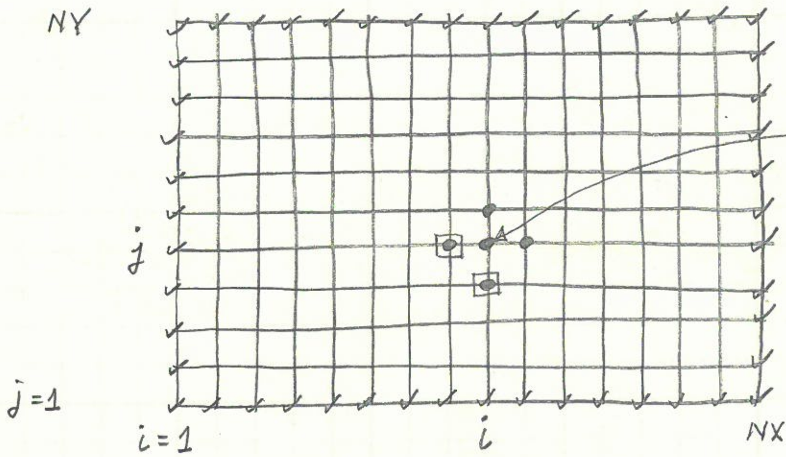
$$\boxed{u_k^{(m+1)} = \left(r_k - \sum_{\ell=1}^{k-1} A_{k\ell}u_\ell^{(m+1)} - \sum_{\ell=k+1}^{n} A_{k\ell}u_\ell^{(m)}\right)/A_{kk}}$$

- FOR FULL MATRICES, $O(n^2)$ OPERATIONS PER ITERATION, BUT FOR SPARSE MATRICES, $O(\underset{\uparrow}{\alpha} n)$ OPERATIONS PER ITERATION ... <u>COULD BE VERY EFFICIENT</u>

  # OF NON-ZERO DIAGONALS

GRID-BASED ITERATIVE METHODS FOR THE LAPLACE/POISSON EQUATION

- Let us use indexing compatible with MATLAB (no zero indices)



✓ = KNOWN FROM FIRST TYPE B.C. SAY

$U_{i,j}$

"GRID-BASED" ⇒ NOT ASSEMBLING AND STORING A MATRIX.

FINITE DIFF. APPROX. $$\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta x^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta y^2} = f_{i,j}$$

Let m = ITERATION #, m = "old", $m+1$ = "new" (superscript)

JACOBI: $$U_{i,j}^{m+1} = \frac{f_{i,j} - \frac{1}{\Delta x^2}\left(U_{i+1,j}^{m} + U_{i-1,j}^{m}\right) - \frac{1}{\Delta y^2}\left(U_{i,j+1}^{m} + U_{i,j-1}^{m}\right)}{\left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)};$$

consider the simple case, $\Delta x = \Delta y = \Delta$, Jacobi can be rewritten as:

$$U_{ij}^{m+1} = U_{i,j}^{m} + \frac{1}{4}\underbrace{\left(-f_{i,j}\Delta^2 + U_{i-1,j}^{m} + U_{i,j-1}^{m} + U_{i,j+1}^{m} + U_{i+1,j}^{m} - 4U_{i,j}^{m}\right)}_{\delta_{i,j}^{m}}$$

this is the error in the equation at (i,j) at iteration m! can use for convergence check

So,

```
for i = 2: NX-1
for j = 2: NY-1
   delta(i,j) =
   unew(i,j) =
end
end
  d = max(max(abs(delta)))
  u = unew
```

... this is basically what we need to do, while d > toler

IF WE FOLLOW THE JACOBI APPROACH CLOSELY, NOTE THAT WHEN WE GET TO (i,j), WE ALREADY HAVE $U_{i,j-1}^{m+1}$, $U_{i-1,j}^{m+1}$ - WHY NOT USE THESE HOPEFULLY BETTER ESTIMATES?

GAUSS-SEIDEL:

$$U_{i,j}^{m+1,G-S} = U_{i,j}^{m} + \frac{1}{4}\underbrace{\left(-f_{i,j}\Delta^2 + U_{i-1,j}^{m+1} + U_{i,j-1}^{m+1} + U_{i+1,j}^{m} + U_{i,j+1}^{m} - 4U_{i,j}^{m}\right)}_{\delta_{i,j}^{m,G-S}}$$

In this case there is no need to distinguish "old" and "new" - can overwrite the old u values as and when they are updated!

SUCCESSIVE OVER-RELAXATION: (SOR)

($\omega = 1/4$ is same as Gauss-Seidel)

$$U_{i,j}^{m+1,SOR} = U_{i,j}^{m} + \omega\,\delta_{i,j}^{m,G-S}, \quad \text{where } \omega > 1/4 \quad (<\tfrac{1}{4} \Rightarrow \text{"under relaxation"})$$

can rewrite as: (note: $\delta_{i,j}^{m,G-S} = 4\,(U_{i,j}^{m+1,G-S} - U_{i,j}^{m})$)

$$U_{i,j}^{m+1,SOR} = (1-4\omega)\,U_{i,j}^{m} + 4\omega\,U_{i,j}^{m+1,G-S}$$

for $\omega < 1/4$, this is like giving some weight to $U_{i,j}^{m+1}$ (Gauss-Seidel) and some weight to $U_{i,j}^{m}$ (old). For $\omega > 1/4$, we are assigning a negative weight to $U_{i,j}^{m}$ and excess weight to $U_{i,j}^{m+1}$ (Gauss-Seidel)! this turns out to accelerate the convergence!! (works for $\omega < 1/2$)

IN THE NEXT PROJECT WE WILL SEE HOW VARYING ω INFLUENCES THE NUMBER OF ITERATIONS NEEDED FOR CONVERGENCE.

CAN FOR-LOOPS IN JACOBI / GAUSS-SEIDEL / SOR BE REPLACED WITH A SINGLE LINE ACTING ON A WHOLE MATRIX? (VECTORIZING?)

WITH TYPE-II B.C., the nodal equations are changed as we discussed earlier while developing the matrix form.

e.g. with Type II on top edge, the eqns. for the uppermost interior nodes becomes: ($i, j = NY-1$)

specified flux

$$\frac{1}{\Delta x^2} U_{i-1,NY-1} + \frac{1}{\Delta y^2} U_{i,NY-2} + \left(\frac{-2}{\Delta x^2} + \frac{-1}{\Delta y^2}\right) U_{i,NY-1} + \frac{1}{\Delta x^2} U_{i+1,NY-1} = f_{i,NY-1} + \frac{\gamma}{\Delta y}$$

For $\Delta x = \Delta y = \Delta$, JACOBI ITERATION EQN. BECOMES:

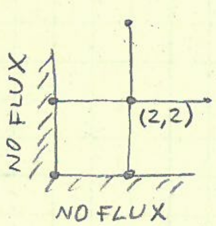
$$U^{m+1}_{i,NY-1} = U^{m}_{i,NY-1} + \underset{\substack{\uparrow \\ 1/3,\ \text{NOT } 1/4}}{\frac{1}{3}} \underbrace{\left(-f_{i,NY-1}\Delta^2 - \gamma\Delta + U^{m}_{i-1,NY-1} + U^{m}_{i,NY-2} + U^{m}_{i+1,NY-1} - 3U^{m}_{i,NY-1}\right)}_{\text{modified } \delta^{m}_{i,NY-1}}$$

$$U^{m+1,G\text{-}S}_{i,NY-1} = U^{m}_{i,NY-1} + \frac{1}{3} \underbrace{\left(\text{''} + \underline{\ \ }^{m+1} + \underline{\ \ }^{m+1} + \text{''} - \text{''} \right)}_{\text{modified } \delta^{m,G\text{-}S}_{i,NY-1}}$$

FOR SOR,

$$U^{m+1,SOR}_{i,NY-1} = U^{m}_{i,NY-1} + \frac{4}{3}\omega\, \delta^{m,G\text{-}S}_{i,NY-1}$$

- THESE MODIFICATIONS WILL NEED TO BE USED AT FIRST/LAST ROW/COLUMN INTERIOR NODES <u>ADJACENT TO A TYPE II BOUNDARY NODE</u>. THE CORRESPONDING BOUNDARY NODES $U_{i,NY}$ (OR EQUIVALENT) ARE NOT INVOLVED IN THE SYSTEM OF EQS. SOLVED. THEY ARE OBTAINED AFTER FULL SOLUTION, E.G. $U^{final}_{i,NY} = U^{final}_{i,NY-1} + \gamma\Delta y$ ETC.
- AT CORNER NODES, IF YOU HAVE TWO TYPE II BOUNDARIES, THE EQUATION WILL CHANGE FURTHER



SIMILARLY