Elliptic equations

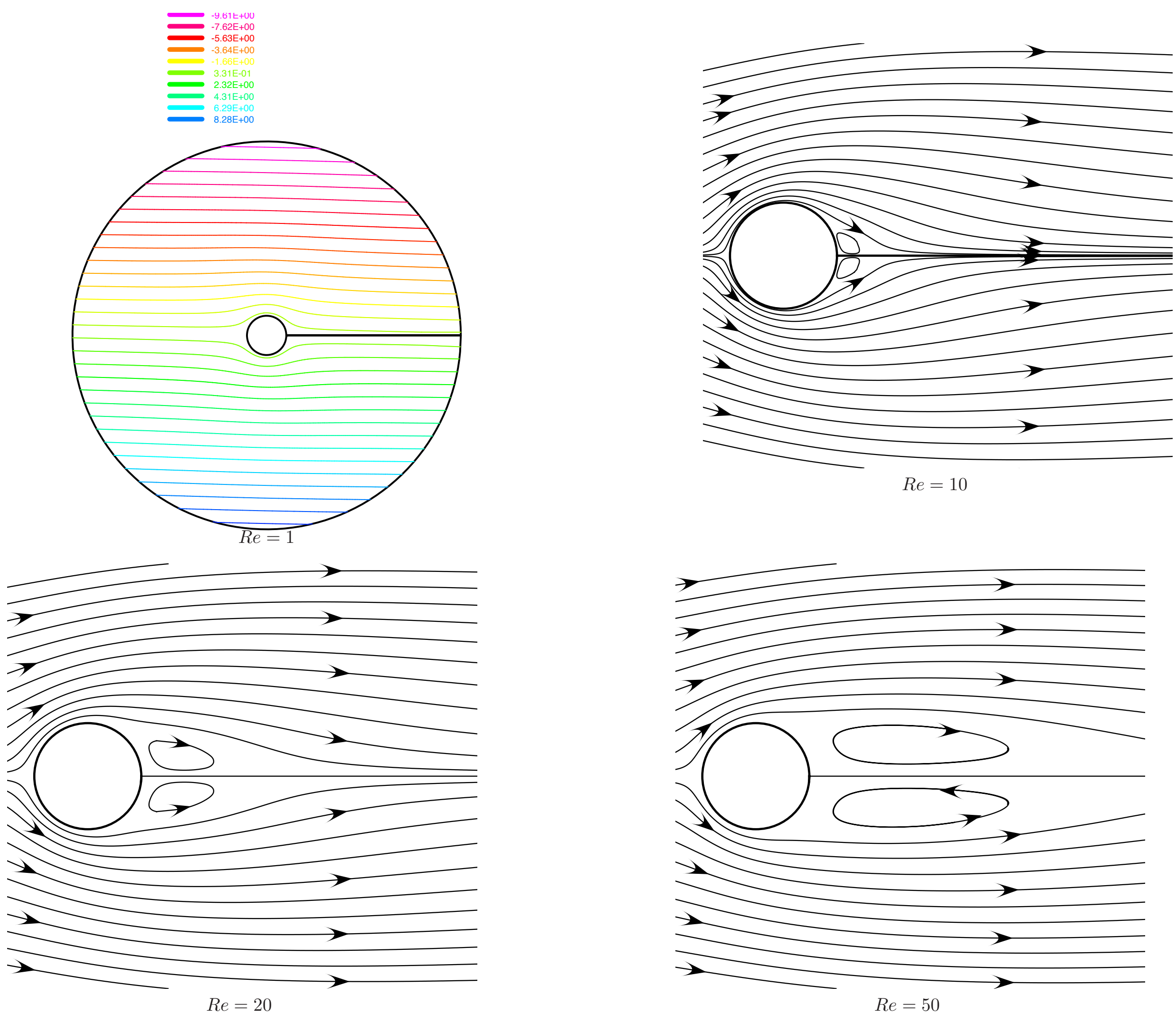
A. Phenomenology: action at a distance, e.g., potential flow (set up instantaneously; pressure forces)

$$\nabla^2\phi = 0 = \partial_x{}^2\phi + \partial_y{}^2\phi \; [+ \partial_z{}^2\phi \,] \quad \text{a (Laplace Equation; Poisson } \nabla^2\psi = f(x,y)\text{)}$$

Compressible slender body: $(1-M^2)\;\partial_x{}^2\phi + \partial_y{}^2\phi = 0$ [M<1 Elliptic, M>1 hyperbolic]

Incompressible Navier stokes: Poisson equation for pressure, $\nabla^2 P = -\rho\partial_i u_j \partial_j u_i$

Poisson equation for incompressible flow:

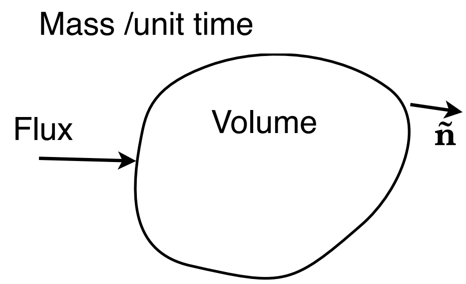


Irrotational and rotational streamline patterns

B. Derivation in fluid mechanics context

1. Mass conservation for incompressible flow

$$\int \rho \mathbf{u} \cdot \mathbf{n} \, dS = 0 \rightarrow \int \nabla \cdot \rho \mathbf{u} \, dV = 0$$



$$\nabla \cdot \mathbf{u} = 0 \quad \text{if } \rho = \text{constant}$$

If $u = -\partial\psi/\partial y$ and $v = \partial\psi/\partial x$ this is automatic

Note: for uniform flow $\psi = -U_\infty y$;

will be used as b.c. (far-field)

2. By definition of vorticity (as in vorticity-streamfunction method -- given ω find u)

$$\omega = \nabla \mathrm{x} \mathbf{u} = \partial v/\partial x - \partial u/\partial y = \nabla^2 \psi \qquad \text{Poisson equation}$$

Vorticity is twice angular rotation: for $u=-\Omega y$, $v = \Omega x$, $\partial v/\partial x - \partial u/\partial y = 2\Omega$.

ψ is streamfunction of flow induced by vorticity (at a distance).

Large r, $u_\Theta \rightarrow \int \omega \, dA / r = \Gamma/r$. etc. where $\int \omega \, dA = \Gamma$

Far-field $\psi \sim -U_\infty y + \Gamma/4\pi \, \mathrm{Ln}[(x^2+y^2) / a^2]$ -- Airfoil with circulation

3. On an impermeable surface, ψ=constant because $\mathbf{u} \cdot \hat{\mathbf{n}} = 0 \Rightarrow \hat{\mathbf{t}} \cdot \nabla\psi = 0$

Without circulation, the far-field boundary condition is $\psi \sim -U_\infty y_{boundary}$

So, given a grid, the boundary coordinates are (x_b, y_b) and the second is used.

Method 1: Elliptic equations by artificial time-stepping

Recall: $\partial_t T = \kappa \partial_x^2 T \rightarrow \partial_x^2 T = 0$ as $t \rightarrow \infty$

To solve $\nabla^2\phi = 0$ alter to $\nabla^2\phi = \partial\phi/\partial\tau$

So steady state solution is obtained by integrating in time. Same applies in 2-D:

$\partial_t T = \kappa \nabla^2 T$ + b.c.'s

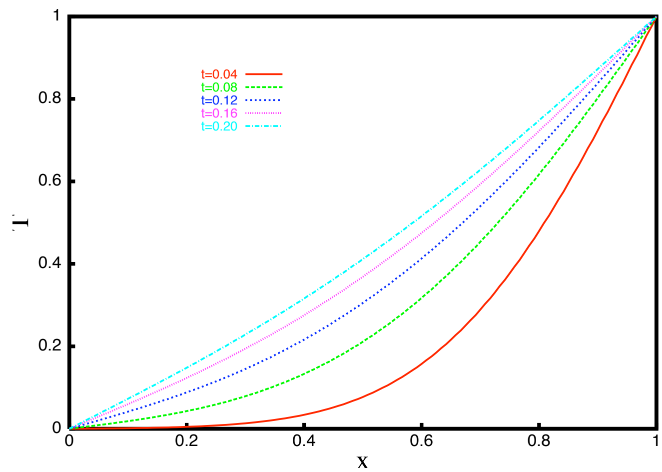
$\rightarrow$ $\nabla^2 T = 0$ + b.c.'s as $t \rightarrow \infty$

Parabolic leads to elliptic (steady state heat transfer via transient).

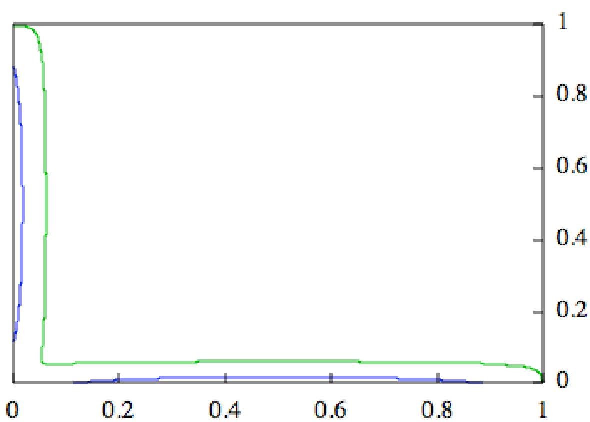
Or, artificial time-stepping

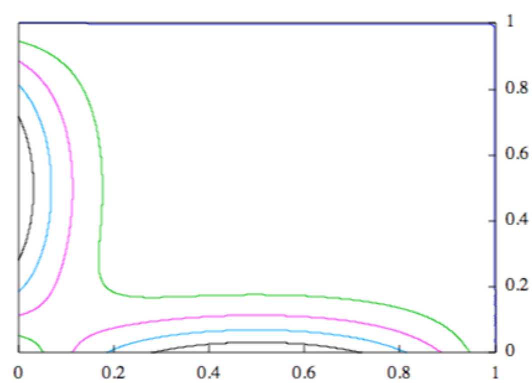
$\partial_\tau T = \nabla^2 T$

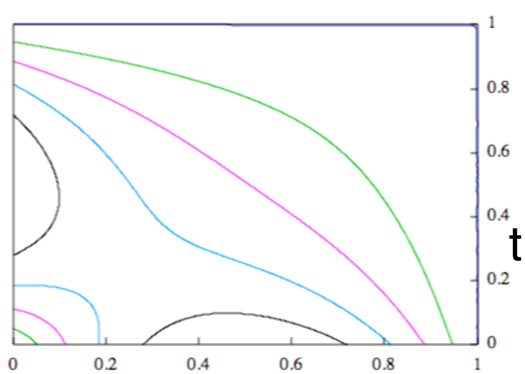
where τ is not treated accurately.



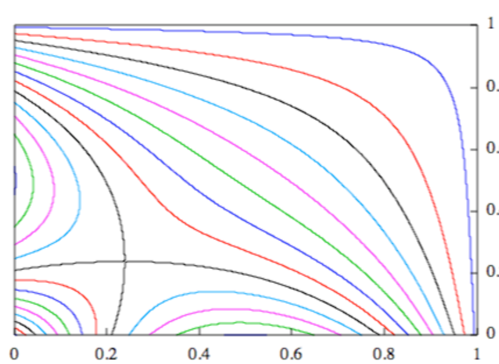
Bc's sin(x(1-x)), sin(y(1-y)),0,0







$t = \infty$



contour plot of the solution $\phi(x,y)$

E.g., want $\alpha = \Delta t / \Delta x^2 \sim \Delta t / \lambda^2$ to be large to get to steady state quickly. Long waves (large λ) damp slowly.

Argues for implicit method. Compare Euler to Crank-Nicholson. Recall VonNeuman stability, amplification factor. For Crank-Nicholson

$$G = \left| \frac{1 - 2\alpha \sin^2(\theta/2)}{1 + 2\alpha \sin^2(\theta/2)} \right|$$

When α is large $G \rightarrow 1$ (-1 inside absolute values, solution oscillates) so there is little damping. But we want damping, or transients to die out. For Euler implicit

$$G = \left| \frac{1}{1 + 4\alpha \sin^2(\theta/2)} \right|$$

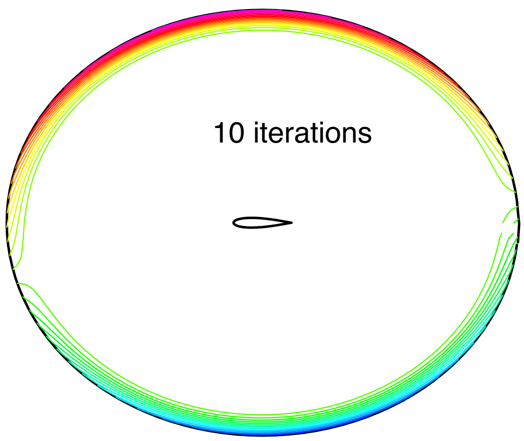
get faster damping, quicker approach to steady state.

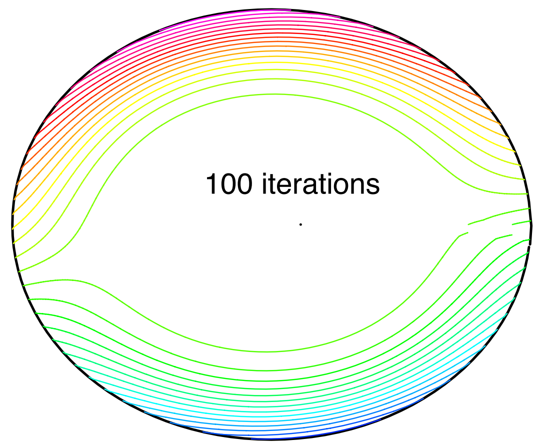
Next section will start section on genuinely Elliptic solvers.

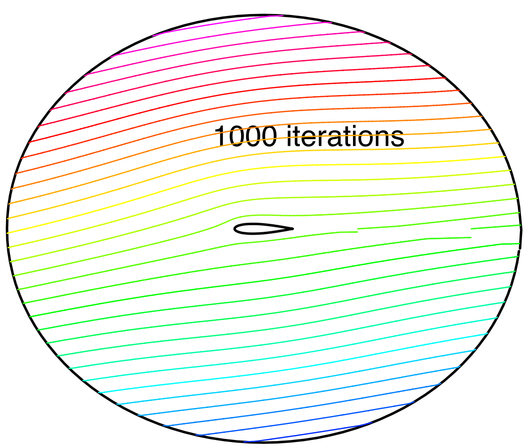
Iterative solution methods

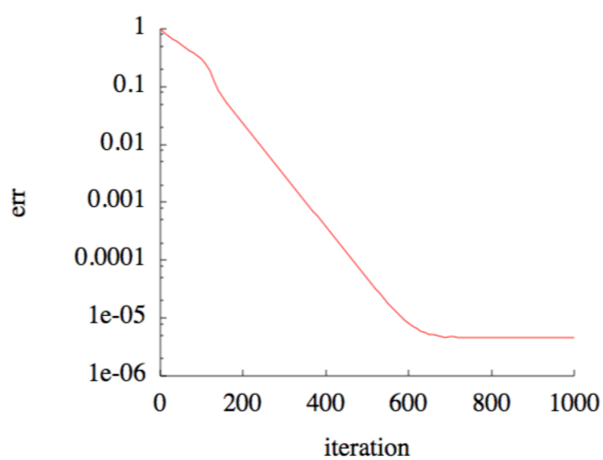
1) Start from guess + boundary conditions
2) Successive improvements -- sweeping through domain (like b.c.s propagating in)
3) Check size of ‘residual’ = ε; loop to step 2 until $|\varepsilon| <$ tolerance e.g. 10^{-4}

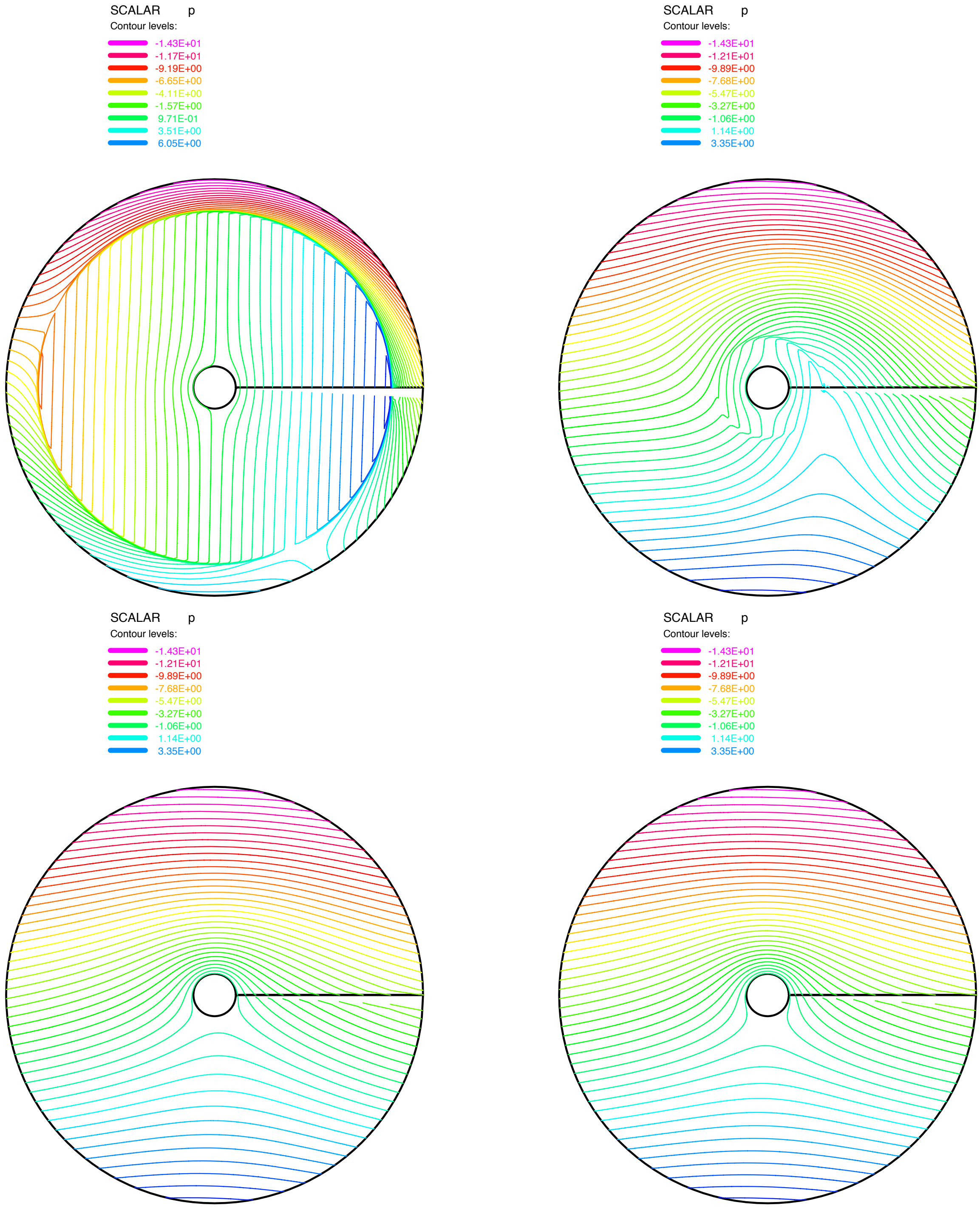
Can view as a search for the solution in high dimensional space (literally true for c.g.), or as diffusing the b.c.s into the domain, or as propagating the error out of the domain.









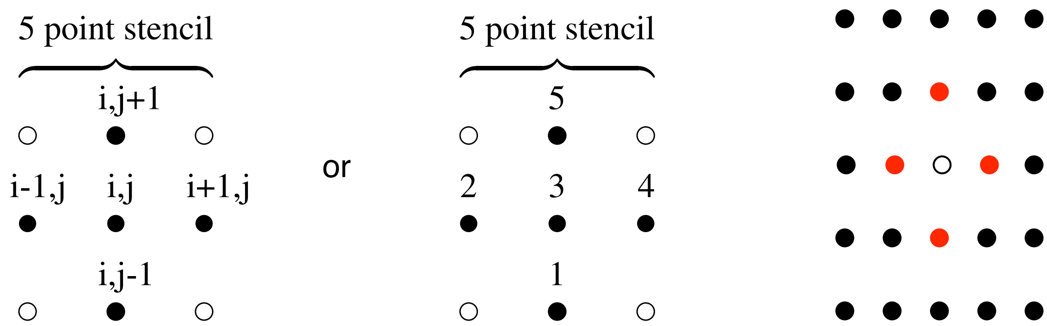


Cylinder with circulation 20, 100, 200, 1000 iterations

Boundary conditions, $\psi = -U_\infty yb + \Gamma/4\pi \, \mathrm{Ln}[\,(x^2+y^2)\,/a^2]$ on outer circle; $\psi=0$ on inner circle

Gauss-Seidel iterative method

(Euler explicit with overwriting)



A. Method

No ∂_t ; solve $\nabla^2\psi = 0 = \delta_x^2\psi + \delta_y^2\psi$ + B.C. Finite difference form

$$\frac{\psi^n_{i+1,j} - 2\psi^{n+1}_{i,j} + \psi^{n+1}_{i-1,j}}{\Delta x^2} + \frac{\psi^n_{i,j+1} - 2\psi^{n+1}_{i,j} + \psi^{n+1}_{i,j-1}}{\Delta y^2} = 0 \text{ (or } \omega(x,y)) + \ B.C.'s$$

Rearrange: †

$$2\frac{\psi_{i,j}}{\Delta x^2} + 2\frac{\psi_{i,j}}{\Delta y^2} = -\omega_{i,j} + \frac{1}{\Delta x^2}(\psi_{i+1,j} + \psi_{i-1,j})\frac{1}{\Delta y^2}(\psi_{i,j+1} + \psi_{i,j-1})$$

Solve $\mathbf{A}\cdot\boldsymbol{\psi} = \mathbf{b}$. Gauss elimination requires O(N BW^2). Reduce to N $\otimes$ # iterations.

$$A_3 \cdot \psi^{n+1}_{i,j} = \omega - (A_1 \cdot \psi^{n+1}_{i,j-1} + A_2 \cdot \psi^{n+1}_{i-1,j} + A_4 \cdot \psi^{n}_{i+1,j} + A_5 \cdot \psi^{n}_{i,j+1})$$

Use this method: ‘’Guess” ψ(*,*); say ψ=0 except for boundary values. Update iteratively: write as ψ_{ij} equals other side and treat as a formula

Prescribe ψ on boundaries e.g. -Uy; don’t change.

WHILE $\varepsilon > 10^{-4}$

  ε = 0

  DO k=2,K-1

    DO j=2,J-1

      ψ_S = ψ(j,k) ! *save for evaluating error*

⇒ ψ(j,k) = [ω(j,k)-($A_1 \cdot \psi^{n+1}$(i,j-1) + $A_2 \cdot \psi^{n+1}$(i-1,j)+$A_4 \cdot \psi^{n}$ (i+1,j)+$A_5 \cdot \psi^{n}$ (i,j+1)] / A_3

      ε = ε + (ψ_S-ψ(j,k))**2

    ENDDO

  ENDDO

  IF(ε_0=0) ε_0 = sqrt(ε) ! initialize r.m.s. error (L2)

  ε = sqrt(ε)/ε_0

END WHILE

† Comment: Linear algebra: Let **A** = **L** + **D** + **U**. Then **R** = ω - $\mathbf{L}\psi^{n+1}$ + $\mathbf{U}\psi^{n}$ and $\psi^{n+1} = \mathbf{D}^{-1}\,\mathbf{R}$

$$\frac{\psi^n_{i+1,j} - 2\psi^{n+1}_{i,j} + \psi^{n+1}_{i-1,j}}{\Delta x^2} + \frac{\psi^n_{i,j+1} - 2\psi^{n+1}_{i,j} + \psi^{n+1}_{i,j-1}}{\Delta y^2} = 0 \text{ (or } \omega(x,y)) + \ B.C.'s$$

NB: superscript *n's* were to show iteration level; they are not there in the code:

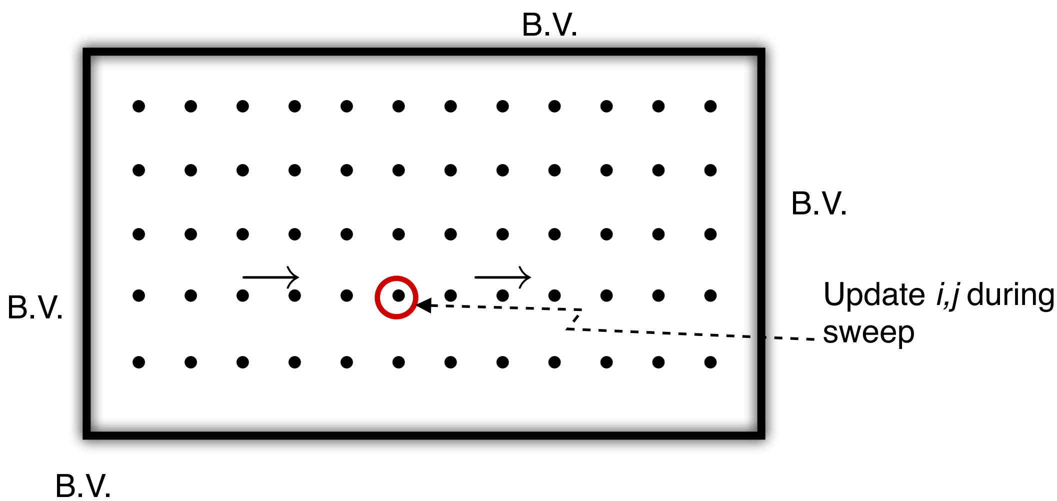
ψ(j,k) = [**-ω(j,k)** + (ψ(j-1,k)+ψ(j+1,k))/Δx^2 + (ψ(j,k-1)/dy^2-ψ(j,k+1))/Δy^2] / (2/Δx^2 +2/Δy^2)

A_1 **= 1/**Δy^2; A_2 **= 1/**Δx^2; A_2 **= 1/**Δx^2; A_4 **= 1/**Δy^2; A_3 **=** -(A_1 **+** A_2 **+** A_4 **+** A_5) (*why*?)

B. What are we doing? Sweep with overwrite. *n* shows iteration level

Residual: $\mathbf{R}^n = \mathbf{b} - \mathbf{A} \cdot \boldsymbol{\psi}^n$. Iterate to drive |**R**| to zero.

Each sweep (WHILE loop) diffuses solution until steady state is reached.
Sweep LL (2,2) to UR (J-1,K-1). First and last column, row are b.c.s.



Comment: Point Jacobi = sweep without overwrite: slower to converge, but can update in parallel

$$2\frac{\psi^{n+1}_{i,j}}{\Delta x^2} + 2\frac{\psi^{n+1}_{i,j}}{\Delta y^2} = -\omega_{ij} + \frac{1}{\Delta x^2}(\psi^n_{i+1,j} + \psi^n_{i-1,j}) + \frac{1}{\Delta y^2}(\psi^n_{i,j-1} + \psi^n_{i,j-1})$$

Comment: why not use ADI to solve $\mathbf{A} \cdot \boldsymbol{\psi} = \mathbf{b}$?

$\mathbf{A} = \mathbf{L}_x + \mathbf{L}_y$. Factoring as $\mathbf{A} = \mathbf{L}_x \cdot \mathbf{L}_y$ does not work. Need identity plus small terms:

$\mathbf{A} = \mathbf{I} + \Delta t\, (\mathbf{L}_x + \mathbf{L}_y) \approx (\mathbf{I} + \Delta t\, \mathbf{L}_x) \cdot (\mathbf{I} + \Delta t\, \mathbf{L}_y)$