L11: Jan 30, 2015 ME565, Winter 2015

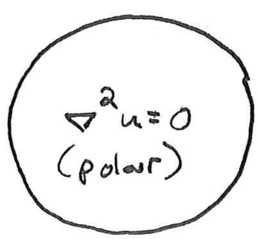
Overview of Topics

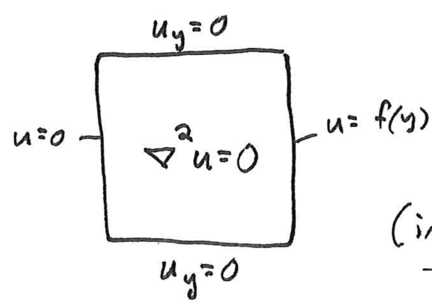
① Example of Laplace Solution in Matlab

Very Briefly!!!

② Orthogonal function spaces
(i.e. polynomial or sine bases)
(∞-dimensional vector space)

③ Other BCs:





(insulating).

Laplace's Equation (numerical):

① Use $u_t = \alpha \nabla^2 u$

and iterate forward ... i.e. finite difference in space & time!

crudest $\frac{\partial}{\partial t}$ possible! (but it works!)

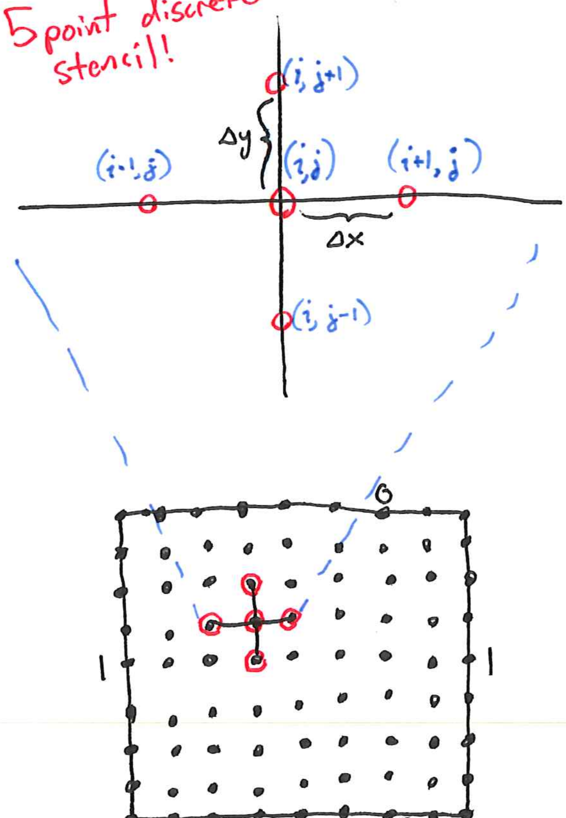
$$\frac{u(t+\Delta t) - u(t)}{\Delta t} = \alpha \nabla^2 u(t)$$

$$\Longrightarrow \quad u(t+\Delta t) = u(t) + (\alpha \Delta t)\, \nabla^2 u(t)$$

(A) $\nabla^2 u$ can be computed using 'del2' function

(B) $\nabla^2 u$ computed by hand using a stencil:

5 point discrete stencil!



$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\frac{u(i+1,j) - u(i,j)}{\Delta x} - \frac{u(i,j) - u(i-1,j)}{\Delta x}}{\Delta x}$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{\frac{u(i,j+1) - u(i,j)}{\Delta y} - \frac{u(i,j) - u(i,j-1)}{\Delta y}}{\Delta y}$$

Assume $\Delta x = \Delta y$ $(=1)$

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{1}{\Delta x^2}\Big(u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) - 4u(i,j) \Big)$$

$$= u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) - 4u(i,j)$$

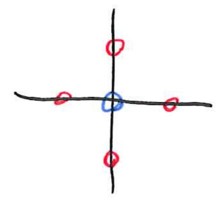
Laplace's Equation (numerical):

(2) Try to make $\nabla^2 u = 0$ at each time step

(A) 5-point stencil: (set $\nabla^2 u = 0$, solve for u_{ij})

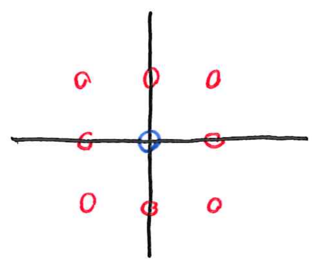
$$u(i,j) = \frac{1}{4}\left(u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) \right)$$

i.e. average neighbors.



(B) 8-point stencil. Similar, average neighboring 8 pts.

$u_{ij} = \frac{1}{8} \sum \text{neighbors}$.

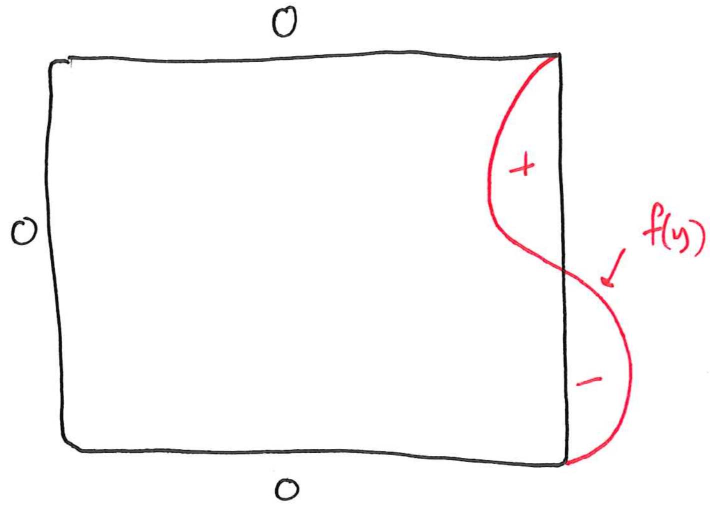


```
clear all, close all, clc
L = 100;
H = 100;
u = zeros(L,H);
load seahawks.mat
colormap(CC);

iteration = 0;
while (iteration<1000),
    iteration = iteration + 1

    % BCs
    u(1,:) = 0;  % top = 0
    u(L,:) = 0;  % bottom = 0
%     u(:,1) = 1;  % left = 1
    u(:,H) = 1;  % right = 1

    % Other BCs
%     u(:,H) = sin(2*pi*(1:L)/L);
    u(:,1) = 0;

    % Plot every 10 iterations
    if (mod(iteration,10)==0),
        imagesc(u);
        colorbar;
        drawnow;
    end

    %% OPTION 1, A:  Use 'del2' to compute Laplacian
    Lu = del2(u);   % Laplacian of u
    u(2:L-1,2:H-1) = u(2:L-1,2:H-1)+Lu(2:L-1,2:H-1);  % replace inner part
    % remember to vary alpha and dt... foreshadowing!


    %% OPTION 1, B:  Use a 5-point stencil
%     Lu = u;
%     for i=2:L-1
%         for j=2:H-1
%             Lu(i,j) = (1)*(-4*u(i,j)+u(i,j+1)+u(i-1,j)+u(i+1,j)+u(i,j-1));
%         end
%     end
%     u(2:L-1,2:H-1) = u(2:L-1,2:H-1)+.1*Lu(2:L-1,2:H-1);
%     % less numerically stable... need a smaller dt.
%

    %% OPTION 1, C:  Use an 8-point stencil
%     Lu = u;
%     for i=2:L-1
%         for j=2:H-1
%             Lu(i,j) = (-8*u(i,j)+u(i-1,j+1)+u(i,j+1)+u(i+1,j+1)+u(i-1,j)+u(i+1,j)+u(i-1,j-↙
1)+u(i,j-1)+u(i+1,j-1));
%         end
%     end
%     u(2:L-1,2:H-1) = u(2:L-1,2:H-1)+.1*Lu(2:L-1,2:H-1);
%     % less numerically stable... need a smaller dt.

%     %% OPTION 2, A:  Forget heat equation, just iteratively try to make Laplacian = 0
%     % 5-pt stencil
%     Lu = u;
%     for i=2:L-1
%         for j=2:H-1
%             Lu(i,j) = (1/4)*(u(i,j+1)+u(i-1,j)+u(i+1,j)+u(i,j-1));
%         end
%     end
%     u(2:L-1,2:H-1) = Lu(2:L-1,2:H-1);
%
```

```
%
    %% OPTION 2, B:  Forget heat equation, just iteratively try to make Laplacian = 0
%     % 8-pt stencil
%     Lu = u;
%     for i=2:L-1
%         for j=2:H-1
%             Lu(i,j) = (1/8)*(u(i-1,j+1)+u(i,j+1)+u(i+1,j+1)+u(i-1,j)+u(i+1,j)+u(i-1,j-1)+u↙
(i,j-1)+u(i+1,j-1));
%         end
%     end
%     u(2:L-1,2:H-1) = Lu(2:L-1,2:H-1);

end
```

$$f(y) = \sin\left(\frac{2\pi}{H} y\right), \quad \text{so} \quad A_2 = \frac{2}{H \sinh\left(\frac{2\pi L}{H}\right)} \int_0^H f(y) \sin\left(\frac{2\pi}{H} y\right) dy$$

all other $A_{\neq 2} = 0$!



So $\boxed{u(x,y) = A_2 \sin\left(\frac{2\pi y}{H}\right) \sinh\left(\frac{2\pi x}{H}\right)}$

Analytic Solution

Works too!!

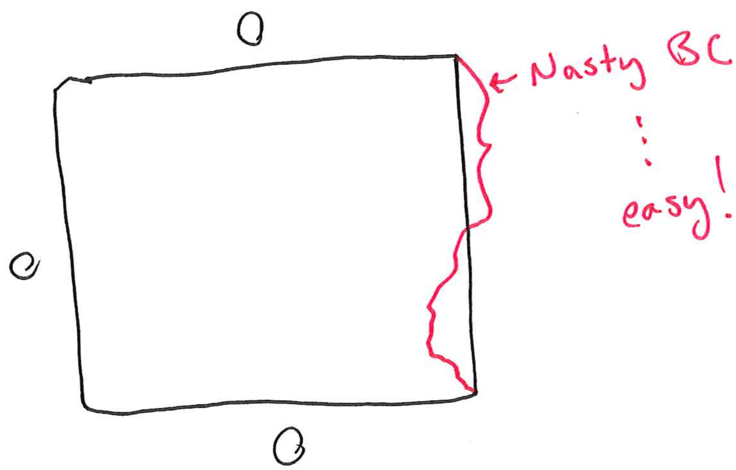
```
clear all, close all, clc
L = 100;
H = 100;
u = zeros(L,H);
[X,Y] = meshgrid(1:1:L,1:1:H);

load seahawks.mat
colormap(CC);

BC = sin(2*pi*(1:H)/H);
A2 = (2/(H*sinh(2*pi*L/H)))*sum(BC.^2);
u =A2*sin(2*pi*Y/H).*sinh(2*pi*X/H);
imagesc(u);
colorbar;
```

What about non-sinusoidal BC's?

We just write a <u>for</u> loop to compute A_K

and add $A_k \sin\left(\frac{k\pi}{\cdots}\right) \sinh\left(\frac{k\pi}{\cdots}\right)$

to solution!



```
clear all, close all, clc
L = 100;
H = 100;
u = zeros(L,H);
[X,Y] = meshgrid(1:1:L,1:1:H);

load seahawks.mat
colormap(CC);

BC = ones(size(H,1));

for k=1:100
    Ak = (2/(H*sinh(k*pi*L/H)))*sum(BC.*sin(k*pi*(1:H)/H));
    u = u + Ak*sin(k*pi*Y/H).*sinh(k*pi*X/H);
end
imagesc(u);
colorbar;
```