MATLAB CODING AND VISUALIZATION FOR 2-D PROBLEMS:

... From the 2-D Poisson equation.

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \omega(x, y)$$

Have a list of frequencies in the x direction:

```
kxlist = (2*pi/L) * [0:n/2-1 -n/2:-1]
```

Notation: $k_{x,j}$ is a number – the j^{th} entry of this list.

Have a list of frequencies in the y direction:

```
kylist = (2*pi/L) * [0:n/2-1 -n/2:-1]
```

Notation: $k_{y,l}$ is a number – the l^{th} entry of this list.

FFT coefficient of $\omega(x, y)$ with x-freqency $k_{x,j}$ and y-frequency $k_{y,l}$: $FFT\omega_{j,l}$.

FFT coefficient of solution $f(x, y)$ with x-freqency $k_{x,j}$ and y-frequency $k_{y,l}$:

$$FFT_{j,l} = -\frac{FFT\omega_{j,l}}{k_{x,j}^2 + k_{y,l}^2}$$

MATLAB CODING TECHNIQUES:

```
L=20;
N=100;
x=linspace(-L/2,L/2,N);
y=x;
```

Make a grid X of x values and a grid Y of y values

```
[X,Y]=meshgrid(x,y);
```

```
>> X(1:4,1:4)
  -10.0000   -9.7980   -9.5960   -9.3939
  -10.0000   -9.7980   -9.5960   -9.3939
  -10.0000   -9.7980   -9.5960   -9.3939
  -10.0000   -9.7980   -9.5960   -9.3939


>> Y(1:4,1:4)
  -10.0000  -10.0000  -10.0000  -10.0000
   -9.7980   -9.7980   -9.7980   -9.7980
   -9.5960   -9.5960   -9.5960   -9.5960
   -9.3939   -9.3939   -9.3939   -9.3939
```

Define a function $\omega(x, y)$ based on these X and Y values. Use elementwise operations!

```
omega = exp(-X.^2 - Y.^2);
>> omega(1:4,1:4)
   1.0e-82 *
    0.0000    0.0001    0.0003    0.0014
    0.0001    0.0003    0.0015    0.0067
    0.0003    0.0015    0.0068    0.0316
    0.0014    0.0067    0.0316    0.1462
```

Define a grid of FFT values: `FFTmatrix_omega=fft2(omega)`

Make a grid KX of k_x values and a grid KY of k_y values:

```
[KX,KY] = meshgrid(kxlist,kylist)
```

Do element-wise division on our grid ... and invert:

```
solution = ( ifft2( -FFTmatrix_omega./(KX.^2 + KY.^2) ) );
```

2-D VISUALIZATION: `two_d_visualization.m`

Create a function on a grid of x and y values:

```
[X,Y]=meshgrid(x,y);
u=cos(X/5).*cos(Y/2) ;
```

Surface plot:

```
surf(x,y,u);
xlabel('x'),ylabel('y'),zlabel('u')
```

More 2D plots

```
figure(2), surfc(x,y,u)  % with contour
figure(3), surfl(x,y,u,[0 1 0])  % with lighting from direction [0 1
figure(4), mesh(x,y,u)
figure(5), pcolor(x,y,u)  % top view
colorbar  %add colorbar
```

colormaps/shading

```
figure(7), surfl(x,y,u), shading interp, colormap(hot)
figure(8), surfl(x,y,u), shading interp, colormap(gray)
figure(9), surfl(x,y,u), shading interp, colormap(copper)
```

```
print -djpeg test.jpg
```

3-D VISUALIZATION: three_d_visualization.m

Say have $u(x, y, z)$.

```
[X,Y,Z]=meshgrid(x,y,z);
u=cos(X).*cos(Y).*cos(Z);
```

Plot isosurface (level set) – values of x, y, z such that $u(x, y, z) = 0.5$

```
isosurface(x,y,z,u,0.5), grid on
```

Add other isosurfaces

```
hold on
isosurface(x,y,z,u,-0.5)
isosurface(x,y,z,u,0.25)
isosurface(x,y,z,u,-0.25)
```