Lecture 12

Using this Kronecker-product machinery, constructed A for N_x=10 and N_y=15 for L_x=1 and L_y=1.5 in Julia. Visualized the pattern of nonzero entries with spy. Solved for the eigenfunctions, and plotted a few; to convert a column vector **u** back into a 2d matrix, used reshape(**u**,N_x,N_y), and plotted in 3d with the surf command. The first few eigenfunctions can be seen to roughly match the $\sin(n_x\pi x/L_x)\sin(n_y\pi x/L_y)$ functions we expect from separation of variables. However, N_x=10, N_y=15 is rather coarse, too coarse a discretization to have a really nice (or accurate) picture of the solutions.

In order to increase N_x and N_y, however, we have a problem. If the problem has $N=N_xN_y$ degrees of freedom, we need to store N^2 numbers ($8N^2$ bytes) just to store the matrix A, and even just solving Ax=b by Gaussian elimination takes about N^3 arithmetic operations. Worked through a few numbers to see that even $N_x=N_y=100$ would have us waiting for 20 minutes and needing a GB of storage, while 3d grids (e.g. 100×100×100) seem completely out of reach. The saving grace, however, is sparsity: the matrix is mostly zero (and in fact the 5-point stencil A has < 5N nonzero entries). This means that, first, you can store only the nonzero entries, greatly reducing storage. Second, it turns out there are ways to exploit the sparsity to solve Ax=b much more quickly, and there are also quick ways to find a *few* of the eigenvalues and eigenvectors.

In Julia, you exploit sparsity by using the `sparse` command and friends to create sparse matrice. Once you have a sparse matrix, Matlab automatically uses algorithms to exploit sparsity if you solve Ax=b by x=A\b and use the `eigs` function to find a few eigenvalues (instead of `eig`).

Starting with the ∇^2 operator on a square grid (from last lecture), showed how we can convert to any other Ω shape with Dirichlet boundaries just by taking a subset of the rows/cols. Looked at a couple of triangular domains, and recovered the Bessel solutions for a circular domain.