Elliptic equations

February 5, 2018

Consider the Laplace equation,

$$\Delta u = u_{xx} + u_{yy} = 0$$

on the square $(0,1)^2$. Use Dirichlet boundary conditions $u = g(x,y)$ on $\partial\Omega$, where

$$\begin{aligned}
g(0,y) &= 0, & 0 \le y \le 1,\\
g(x,0) &= 0, & 0 \le x \le 1,\\
g(1,y) &= 0, & 0 \le y \le 1,\\
g(x,1) &= \sin(\pi x), & 0 \le x \le 1.
\end{aligned}$$

Verify that

$$u_e(x,y) = \frac{1}{\sinh(\pi)} \sin(\pi x)\sinh(\pi y)$$

is the exact solution to this problem.

Task 1 Implement the five point formula for this problem, as described in Chapter 6.2 of the course notes. Use constant step-sizes h and k in each space direction (but allow for $h \neq k$). Verify order 2 convergence in both space directions. *Hand-in: Plot of the numerical solution for $h = k = 0.1$ and convergence plots that verify the correct order in both space directions.*

Notes: A major part of the challenge is understanding how to construct the solution as a vector, when the grid points are natural distributed more like a 2d array. For this purpose, matlab and python both have reshape commands: e.g. suppose that U is an n*m matrix, where `U(j,i)` - matlab, or `U[j-1][i-1]` - python corresponds to $U(x_i, y_j)$. Then

```
U=np.reshape(U,m*n)
U=np.reshape(U,(n,m))
```

Will turn this into a vector and back. It is a little more awkward in matlab as the default ordering is different. Here try

```
U=reshape(U',[m,1])
U=reshape(U,[n,m])'
```

Note that if the vector of grid points is constructed in the manner suggested above, the form of the matrix driving the system is simple

```
e=np.ones((m*n))
f=np.ones((m*n-1))
f[m-1:n*m-1:m]=0
A=sp.sparse.diags([e[0:-2]/k**2,f/h**2,-2*e*(1/h**2 + 1/k**2),f/h**2, e[0:-2]/k**2],
[-m,-1,0,1,m],shape=(m*n,m*n), format="csr")
```

constructs the matrix in python (note that the last two lines should be one line), or in matlab

```
e=ones(M*N,1);
f=ones(M*N,1);
f(M:M:N*M)=0;
g=[1;f(1:end-1)];
A = spdiags([1/k^2*e, 1/h^2*f, -2*(1/h^2+1/k^2)*e, 1/h^2*g, 1/k^2*e],
[-M,-1,0,1,M],M*N,M*N);
```

For the plots of the numerical solution, it is recommended that you reshape your solution U into a matrix as suggested above. Then construct a meshgrid:

```
[xx,yy] = np.meshgrid(np.linspace(h,1-h,m),np.linspace(k,1-k,n))
```

or

```
[xx,yy] = meshgrid(h:h:1-h,k:k:1-k);
```

Then construct the plot using either `plot_surface(xx,yy,U)` or `mesh(xx,yy,U)`. A typical plot: