NumPDE course, Skoltech, Term 3, 2016

Problem Set 3

Spectral Methods

Consider a problem of solving

$$\begin{aligned} -\Delta u &= f, \\ u|_\Gamma &= 0. \end{aligned}$$

by a spectral method, where $\Omega = [0, 1]^2$ is a unit square and $\Gamma = \partial\Omega$.

In a spectral method you will be working with a representation of the solution in the form

$$u(x, y) = \sum_{i=1}^{N} \sum_{j=1}^{N} \hat{u}_{i,j} \sin(\pi i x) \sin(\pi j y),$$

where $\hat{u}^N$ is just an $N \times N$ array.

Knowing $u(x, y)$, it is often practically not possible to compute $\hat{u}_{i,j}$ exactly. Hence one needs to use the Discrete Sine Transform (http://docs.scipy.org/doc/scipy-dev/reference/tutorial/fftpack.html#discrete-sine-transforms) (DST). The python's `dst` computes a one-dimensional DST, but we need a two-dimensional (2D) DST, which we compute in the following way:

- Compute values $u(x, y)$ for N^2 points inside the domain $(x, y) = (hk, h\ell)$, $1 \leq k, \ell \leq N$, $h = 1/(N + 1)$.
- Apply `dst` to each line in this array. This way you will be computing a "partial" 2D DST

$$u(x, y) = \sum_{j=1}^{N} \tilde{u}_j(x) \sin(\pi j y),$$

- Apply `dst` to each column in the resulting array. This way you will be computing a full 2D DST.

Problem 1 (Spectral Methods) (80pt)

- Part (a)
  - Take $u^0(x, y) = \sin(\pi x) \sin(\pi y)$ and compute $f = -\Delta u^0$.
  - **(8pt)** Calculate, explicitly, the coefficients $\hat{f}_{i,j}$ in

    $$f(x, y) = \sum_{i=1}^{N} \sum_{j=1}^{N} \hat{f}_{i,j} \sin(\pi i x) \sin(\pi j y).$$

    Then, using values of f at N^2 points, calculate the coefficients $\hat{f}^N_{i,j}$ by the procedure outlined above. (Here $\hat{f}$ is an "infinite array" that you will compute explicitly, while $\hat{f}^N$ will be an NxN array that you will compute using python.) Compare $\hat{f}$ and $\hat{f}^N$.
  - In this case, obviously, a solution to the problem is $u = u_0$, but let us now pretend we do not know it.

- **(8pt)** Compute the coefficients $\hat{u}^N$ from $\hat{f}^N$ found in part (a).
- **(8pt)** We will estimate the error between the two solutions in the following way. We will take N^2 points in our domain, of the form $(hk, h\ell)$, same as before. We will then be interested in

$$\mathrm{err}_N = \max_{1\le k,\ell\le N} |u^N(hk,h\ell) - u^0(hk,h\ell)|$$

which we call an error of the solution. Calculate the error of your solution for a number of values of N. Explain your results

- Part (b)
  - Let us now take something a little more complicated,

$$u^0(x,y) = \sin(\pi x^2)\sin(\pi y^2), \qquad \text{and} \qquad f = -\Delta u_0.$$

  - You cannot compute the coefficients $\hat{f}$ explicitly, so you'll have to live with only $\hat{f}^N$.
  - **(8pt)** Compute $\hat{u}^N$, the error err_N, and report the error for different values of N. Would you say the error decays fast as N increases?
  - Suppose that the spectral method has an order of convergence, in other words the error behaves like $\mathrm{err}_N = CN^{-\mathrm{ord}}$ and you need to find ord.
    - **(8pt)** Derive the formula

$$\mathrm{ord}_N = \frac{\ln(\mathrm{err}_N) - \ln(\mathrm{err}_{2N})}{\ln(2)}$$

    - **(8pt)** Hence compute ord_N for $N = 1, \ldots, 20$ and comment on your results. (You should start with taking each value of N between 1 and 20 to understand the behavior, but you don't need to present all these numbers in your report, as long as you can illustrate the right behavior.)
- Part (c)
  - Let us now play a "fair game": take

$$f(x,y) = 1.$$

  We then do not know the solution.
  - Use your code from part (b) to compute $\hat{f}^N$, and $\hat{u}^N$.
  - **(8pt)** Instead of the exact error we have to use the error estimate

$$\mathrm{errest}_N = \max_{1\le k,\ell\le N} |u^N(hk,h\ell) - u^{2N+1}(hk,h\ell)|$$

  (Why did we take $2N+1$ instead $2N$?) Report the values of errest_N for a sequence of values of N.
  - **(8pt)** Hence compute errest_N for a sequence of N and comment on your results.
  - **(8pt)** Finally, using the same formula (but with errest),

$$\mathrm{ord}_N = \frac{\ln(\mathrm{errest}_N) - \ln(\mathrm{errest}_{2N})}{\ln(2)}$$

  compute ord_N for a squence of values of N. Comment on your results.
- **(8pt)** Compare the behavior for err, errest, and ord for parts (b) and (c). What is the main reason for the qualitative difference in the speed of convergence?

Problem 2 (Adaptivity) (20 pts)

Complete the adaptive FEM code that solves the Poisson equations form the lecture. Use this notebook (PS3_Pr2_code.ipynb)