FFT-based 2D Poisson solvers

In this lecture, we discuss Fourier spectral methods for accurately solving multidimensional Poisson equations on rectangular domains subject to periodic, homogeneous Dirichlet or Neumann BCs. We also note how the DFT can be used to efficiently solve finite-difference approximations to such equations. The methods can be generalized to other elliptic equations with only even derivatives.

WIth $N = 2^p$ gridpoints along each of d dimensions, the methods require $\mathrm{O}(N^d \log N)$ flops, or $\mathrm{O}(\log N)$ flops per gridpoint, which is close to the best possible efficiency of $\mathrm{O}(1)$ flops per gridpoint. For large N and $d \geq 2$ this is better efficiency than a direct sparse LU solver applied to the FDA matrix problem.

The two-dimensional Poisson problem

Consider a two-dimensional Poisson equation:

$$u_{xx} + u_{yy} = f(x, y), \tag{1}$$

on a periodic domain $[0, L] \times [0, L]$. The Fourier gridpoints are (x_j, y_l), where $x_j = (j-1)h$, $y_l = (l-1)h$, and $h = L/N$. The spectral approximation to the solution is

$$u(x, y) = N^{-2} \sum_{m=1}^{N} \sum_{n=1}^{N} U_{mn} \exp[i(k_n x + l_p y)]$$

with $k_n = 2\pi M_n / L$ defined in terms of the harmonics M_n as in the 1D case, and l_p defined similarly. We define $u_{jl} = u(x_j, y_l)$. We now search for the approximate solution that exactly solves (1) at the Fourier gridpoints. This is

$$\begin{aligned}
\hat{\mathbf{f}} &= DFT_x(DFT_y(\mathbf{f})), \\
\hat{u}_{np} &= -\hat{f}_{np}/(k_n^2 + l_p^2), \\
\hat{u}_{11} &= 0, \\
\hat{\mathbf{u}} &= IDFT_x(IDFT_y(\hat{\mathbf{u}}))
\end{aligned}$$

We have assumed the solvability condition $\hat{f}_{11} = 0$ (the RHS has zero grid-mean over the 2D domain), and for uniqueness we have imposed the condition that the solution have zero mean ($\hat{u}_{11} = 0$).

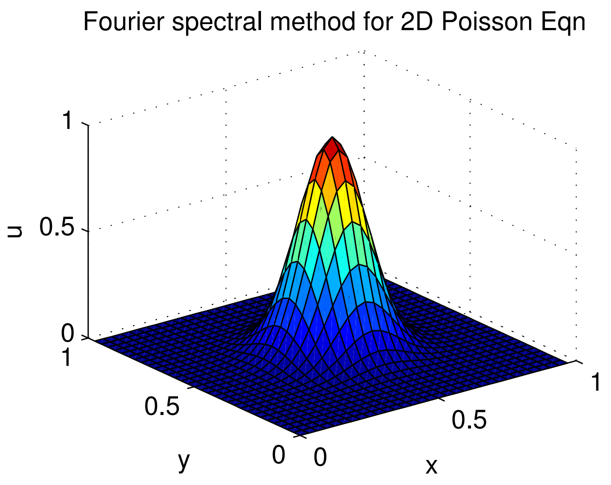

Figure 1: Fourier spectral solution of 2D Poisson problem on the unit square with doubly periodic BCs. Forcing is the Laplacian of a Gaussian hump. The hump is almost exactly recovered as the solution $u(x, y)$.

In Matlab, the function `fft2` and `ifft2` perform the operations DFTx(DFTy($\cdot$)) and the inverse. Using these, the script `pois2Dper.m` solves the Poisson equation in a square with a forcing in the form of the Laplacian of a Gaussian hump in the center of the square, producing Fig. 1. Just a few lines of Matlab code are needed. As expected, the solution accurately recovers the Gaussian hump.

Homogeneous Dirichlet or Neumann BCs can be handled by odd/even extension, as in the 1D case.

Fast FFT-based Finite Difference Poisson solvers

A DFT can be used to find efficient direct solutions to the centered finite difference approximation to Poisson's equation on rectangular domains with a uniform grid spacing in each direction. This method, though usually less accurate than a spectral method, is useful for a broader class of commonly arising problems, especially those with nonhomogeneous boundary conditions.

The key fact is that given a vector $\mathbf{v}$ with components v_i, the DFT of periodically shifted versions of this vector $v_{i\pm1}$ are $\exp(ik_n\Delta x)$ times the DFT of $\mathbf{v}$. This is easily proved by substituting into the DFT definition.

We describe the method for the two-dimensional Poisson equation (1) in a square of side L with $m \times m$ interior grid points with uniform spacing h in each direction and inhomogeneous Dirichlet BCs. We earlier derived the centered finite-difference approximation to this PDE,

$$(D_x^2 + D_y^2)u_{ij} = f_{ij} \qquad (i, j = 1, \ldots, m) \tag{2}$$

where

$$\begin{aligned} D_x^2 u_{ij} &= \frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} - 2u_{ij}) \\ D_y^2 u_{ij} &= \frac{1}{h^2}(u_{i,j+1} + u_{i,j-1} - 2u_{ij}) \end{aligned}$$

and f_{ij} is $f(x_i, y_j)$ suitably modified along the rows/columns neighboring a boundary to account for the inhomogeneous boundary conditions.

We make an odd extension in x of the solution and the right hand side, i. e. for each y_j we define the extended vector of length N_e=$2(m+1)$,

$$\mathbf{v}_j = [0, u_{1j}, u_{2j}, \ldots, u_{mj}, 0, -u_{mj}, \ldots, -u_{1j}]$$

$$\mathbf{g}_j = [0, f_{1j}, f_{2j}, \ldots, f_{mj}, 0, -f_{mj}, \ldots, -f_{1j}]$$

Now, let $\hat{g}_{nj}$ be the n'th element of the DFT of $\mathbf{g}_j$, and similarly for v. Applying the DFT to the $j'th$ column of v,

$$\text{DFT}[D_x^2 \mathbf{v}_j]_n = \frac{\exp(ik_n h) + \exp(-ik_n h) - 1}{h^2}\hat{v}_{nj} = -\mu_n \hat{v}_{nj}$$

where

$$\mu_n = \frac{4\sin^2(k_n h/2)}{h^2} = \frac{4\sin^2((n-1)\pi/N_e)}{h^2}$$

From here, there are two approaches. The first approach is to do an odd extension G, V of g, v and DFT in the y direction, leading to the equation

$$-(\mu_n + \mu_p)\hat{V}_{np} = \hat{G}_{np},\ 1 \le n, p \le N_e, \qquad \text{except } \hat{V}_{11} = 0$$

which can be solved for the 2D DFT $\hat{V}_{np}$ of the extended solution. The mode $n = p = 1$ corresponds to zero harmonics in both directions, i. e. the domain mean. For this case, the above equation would give an indeterminate $\hat{V}_{11}$, but since an odd extension must have zero mean over the extended domain, we can simply specify that this harmonic of the solution is zero. A 2D IDFT and subsetting to the non-extended domain recovers u_{ij}. A Matlab script `pois_FD_FFT_2D.m` implementing this method is on the class web page.

The second approach is to substitute into (2) to obtain

$$\frac{1}{h^2}\hat{v}_{n,j+1} - (\mu_n + \frac{2}{h^2})\hat{v}_{nj} + \frac{1}{h^2}\hat{v}_{n,j+1} = \hat{g}_{nj}, (j = 1, \ldots, m, n = 1, \ldots, N_e)$$

For each n, this is a tridiagonal system which can be solved in O(m) flops for the $\hat{v}_{nj}$. Taking the inverse DFT in x, we obtain the vectors $\mathbf{v}_j$, from which we can read off the FDA solution u_{ij}. This approach is a bit more efficient and can handle Poisson-like equations with coefficients varying in the y direction, but is also more complicated to implement than the first approach. Either approach requires $O(N^2 \log N)$ flops for a 2D Poisson equation, and is easily generalized to Poisson-like equations in rectangular boxes in three or dimensions.