Notes on Fast Poisson Solvers

Dianne P. O'Leary

February 23, 2010

Update of May 2, 2005 version

There is a set of very fast algorithms for solving a finite difference approximation of Laplace's equation on a rectangle. (They also have application to some queueing network problems and some random walks.) They are based on the fact that the eigensystem of the resulting matrix is very simple. This set of notes discusses one such algorithm.

Let

$$A = \begin{bmatrix} T & -I & & & & \\ -I & T & -I & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & -I \\ & & & & -I & T \end{bmatrix}$$

be an $N \times N$ matrix with

$$T = \begin{bmatrix} 4 & -1 & & & & \\ -1 & 4 & -1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & -1 \\ & & & & -1 & 4 \end{bmatrix}.$$

The matrix T has dimensions $n \times n$, and let $m \equiv N/n$. The matrix A arises from 2nd order finite difference discretization of the partial differential equation

$$-\Delta u \equiv -u_{xx} - u_{yy} = f$$

in a rectangular region with Dirichlet boundary conditions (i.e., values of u are given on the boundary of the rectangle).

Eigenstructure of A

Let the matrix F_n be defined by

$$f_{ij} = \alpha_j \sin \frac{ij\pi}{n+1}\ ,\ \ i,j = 1,2,\ldots,n$$

where α_j is chosen so that each column of F_n has norm one. Then $F_n^T F_n = I$ and $F_n^T T F_n = \Gamma \equiv diag(\gamma_1,\ldots,\gamma_n)$, with $\gamma_j = 4 - 2\beta_j$ and $\beta_j = \cos\frac{j\pi}{n+1}$. Now let

$$\bar{A} = \begin{bmatrix} F_n^T & & & & \\ & F_n^T & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & F_n^T \end{bmatrix} A \begin{bmatrix} F_n & & & & \\ & F_n & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & F_n \end{bmatrix} = \begin{bmatrix} \Gamma & -I & & & \\ -I & \Gamma & -I & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & -I \\ & & & & -I & \Gamma \end{bmatrix}.$$

Let P be the permutation matrix that reorders the rows of the identity matrix as $1, n+1, 2n+1, \ldots, (m-1)n+1,\ 2, n+2, 2n+2, \ldots, (m-1)n+2$, etc. Then

$$P\bar{A}P^T = \begin{bmatrix} T_1 & & & & \\ & T_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & T_n \end{bmatrix}$$

where

$$T_j = \begin{bmatrix} \gamma_j & -1 & & & \\ -1 & \gamma_j & -1 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & -1 \\ & & & & -1 & \gamma_j \end{bmatrix}$$

has dimension $m \times m$. Now we can see that

$$\begin{bmatrix} F_m^T & & & & \\ & F_m^T & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & F_m^T \end{bmatrix} \begin{bmatrix} T_1 & & & & \\ & T_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & T_n \end{bmatrix} \begin{bmatrix} F_m & & & & \\ & F_m & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & F_m \end{bmatrix}$$

$$= \begin{bmatrix} \Gamma_1 & & & & \\ & \Gamma_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & \Gamma_n \end{bmatrix},$$

where $\Gamma_j = diag(\gamma_j - 2\beta_1, \ldots, \gamma_j - 2\beta_m)$.

Putting all this information together, we have the eigensystem of the matrix A:

$$UAU^T = \Lambda$$

with

$$U = \begin{bmatrix} F_m^T & & & & & \\ & F_m^T & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & F_m^T \end{bmatrix} P \begin{bmatrix} F_n^T & & & & & \\ & F_n^T & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & F_n^T \end{bmatrix}$$

and $\Lambda = diag(\lambda_{jk})$ with

$$\lambda_{jk} = 4 - 2\cos\frac{j\pi}{n+1} - 2\cos\frac{k\pi}{m+1}\,.$$

Solving Linear Systems $Az = b$

We can now give an efficient algorithm for solving the linear system $Az = b$ where $z = [z_{11}, \ldots, z_{1n}, \ldots z_{m1}, \ldots z_{mn}]^T$. It uses the representation $z = U^T\Lambda^{-1}Ub$. It takes advantage of the fact that multiplication by the matrix F_p or F_p^T is equivalent to a discrete Fourier transform of length p. This can be accomplished in $O(p\log_2 p)$ operations if p is a power of 2 and in some larger but still modest number of operations if p is a composite number with many factors. For convenience, we'll assume that m and n are powers of 2.

Step 1: Form

$$\hat{b} = \begin{bmatrix} F_n^T & & & & & \\ & F_n^T & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & F_n^T \end{bmatrix} b$$

by performing m discrete Fourier transforms

$$\hat{b}_k = F_n^T \begin{bmatrix} b_{k1} \\ \cdot \\ \cdot \\ \cdot \\ b_{kn} \end{bmatrix}, k = 1, \ldots, m\,.$$

Cost: $O(mn\log_2 n)$.

Step 2: Form

$$c = \begin{bmatrix} F_m^T & & & & & \\ & F_m^T & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & F_m^T \end{bmatrix} \hat{b}_{permuted}$$

by forming

$$c_j = F_m^T \begin{bmatrix} \hat{b}_{1j} \\ \cdot \\ \cdot \\ \cdot \\ \hat{b}_{mj} \end{bmatrix}, j = 1, \ldots, n\,.$$

Cost: $O(mn \log_2 m)$.

Step 3: Form $\bar{c} = \Lambda^{-1} U b$ by computing $\bar{c}_{kj} = c_{kj}/\lambda_{kj}$, $k = 1, \ldots, m$, $j = 1, \ldots, n$.
Cost: $O(mn)$

Step 4: Form

$$\hat{c} = \begin{bmatrix} F_m & & & & & \\ & F_m & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & F_m \end{bmatrix} \bar{c}$$

by forming

$$\hat{c}_j = F_m \begin{bmatrix} \bar{c}_{1j} \\ \cdot \\ \cdot \\ \cdot \\ \bar{c}_{mj} \end{bmatrix}, j = 1, \ldots, n\,.$$

Cost: $O(mn \log_2 m)$.

Step 5: Form

$$z = \begin{bmatrix} F_n & & & & & \\ & F_n & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & F_n \end{bmatrix} \hat{c}_{permuted}$$

by forming

$$z_k = F_n \begin{bmatrix} \hat{c}_{k1} \\ \cdot \\ \cdot \\ \cdot \\ \hat{c}_{kn} \end{bmatrix}, k = 1, \ldots, m\,.$$

Cost: $O(mn \log_2 n)$.

For further information on this method and other fast direct methods for solving this and similar problems, see

Paul N. Swarztrauber, "The Methods of Cyclic Reduction, Fourier Analysis, and the FACR Algorithm for the Discrete Solution of Poisson's Equation on a Rectangle," *SIAM Review* 19 (1977) 490-501.