Ann. Sc. Math. Québec, 1979, Vol. III, No 1, pp. 135-141

FAST TENSOR PRODUCT POISSON SOLVERS FOR THE DIRICHLET PROBLEM ON A RECTANGLE

Theodore Papatheodorou

RÉSUMÉ

On peut incorporer les caractéristiques principales de la transformée de Fourier rapide aux méthodes usuelles du produit tensoriel. Les méthodes du produit tensoriel rapide qui en résultent et que nous décrivons ont comme nombre d'opérations une quantité de l'ordre $0(cN^2\log_2 N)$. Pour le problème de Dirichlet et l'équation de Poisson, à l'aide des méthodes exposées, on obtient comme valeur de $c = 20$, 10 et 5. Elles se comparent bien en vitesse avec la transformée de Fourier rapide où $c = 5$, elles gardent aussi la simplicité des méthodes de produit tensoriel en analyse et en programmation, leur précision est enfin du $4^{ème}$ ordre lorsque l'on se sert d'une formule à 9 points.

ABSTRACT

The basic features of the Fast Fourier Transform (FFT) are incorporated into the standard Tensor Product (TP) methods. The resulting Fast Tensor Product methods that we describe, have an operation count of $0(cN^2\log_2 N)$. For the Dirichlet problem and Poisson's equation, the methods presented have $c = 20$, 10 and 5. Thus, they compete in speed with FFT (for which $c = 5$), at the same time they preserve the ease of the TP methods in analysis and programming and are 4th order accurate, as a 9-point formula is used.

1. INTRODUCTION

The Tensor Product (TP) method, introduced by Lynch, Rice and Thomas [5], presents a simplicity that allows easy analysis and implementation. However, within the framework of development and evaluation of special techniques for the numerical solution of PDE , the method has not been utilized to its full capacity. Main reason for this is its operation count. For example, this count for Poisson's equation is $0(8N^3)$ and, compared with the $0(5N^2\log_2 N)$ of the Fast Fourier Transform (FFT) , (Hockney [2]), is too large. In this note, we eliminate this disadvantage by incorporation of the basic features of FFT into the TP method. The resulting Fast TP method competes with FFT in speed.

As we are interested in high order methods, we deal directly with a 9-point approximation to Poisson's equation

$$u_{xx} + u_{yy} = f \ .$$

To fix the ideas, we consider Dirichlet boundary conditions and discretization $\Delta x = \Delta y = \frac{1}{N}$ of the unit square. For the Fast techniques we need to take $N = 2^m$.

Let $f_{ij} \equiv f(i\Delta x, j\Delta y)$ and

$$b_{ij} \equiv \frac{(\Delta x)^2}{2}\left[f_{i-1,j} + f_{i,j-1} + f_{i+1,j} + f_{i,j+1} + 8f_{ij}\right] \ , \ 1 \le i \ , \ j \le N - 1 \ .$$

Then define the $(N-1)$-vectors

$$b_i \equiv [b_{i1},\ldots,b_{i,N-1}]^T$$

and the $(N-1)^2$-vector

$$b \equiv [b_1,\ldots,b_{N-1}]^T \ .$$

Similarly, define u_i , u , from the unknowns u_{ij} .

The formula to be used may be taken directly from Rosser [6], or as a special case of a more general formula given in Houstis and Papatheodorou [3]. In TP

form, this is

$$(6I\otimes A + 6A\otimes I + A\otimes A)u = b \ , \tag{1.1}$$

where A is the $(N-1) \times (N-1)$ matrix

$$A = \begin{bmatrix} -2 & 1 & & & 0 \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & 1 & -2 \end{bmatrix} .$$

The form (1.1) is the customary one, but even in the case of the standard TP method, it is more economical to use (1.1) in the form

$$(\Delta\otimes\Delta - 36I\otimes I)u = b \ , \tag{1.2}$$

where Δ is the $(N-1) \times (N-1)$ matrix

$$\Delta = \begin{bmatrix} 4 & 1 & & & 0 \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & 1 & 4 \end{bmatrix} .$$

If we define the matrix S by $S_{ij} = \sqrt{\frac{2}{N}} \sin\left(\frac{ij\pi}{N}\right) \quad 1 \le i \ , \ j \le N - 1 \ ,$ then $S^{-1} = S^T = S$ and

$$S\Delta S = \Lambda(\Delta) \ , \tag{1.4}$$

where $\Lambda(\Delta)$ is diagonal with entries the eigenvalues of Δ

$$\lambda_j = 4 + 2 \cos\left(j\frac{\pi}{N}\right) \quad 1 \le j \le N - 1 \ . \tag{1.5}$$

Then the matrix

$$D = (S\otimes S)(\Delta\otimes\Delta - 36I\otimes I)(S\otimes S) = \Lambda(\Delta)\otimes\Lambda(\Delta) - 36I\otimes I \tag{1.6}$$

is diagonal, and

$$u = (S\otimes S)D^{-1}(S\otimes S)b \ . \tag{1.7}$$

Based on (1.7) the algorith for the TP method of [5] may be described in three steps: Starting with $U(I,J) \equiv b_{ij}$

(1.8) (i) $U \leftarrow SUS$, (ii) $U \leftarrow D^{-1}U$, (iii) $U \leftarrow SUS$,

where "$\alpha \leftarrow \beta$" means "obtain β and store it in α" .

We estimate only the leading term of the operation count. Hence, since D^{-1} is diagonal, step (ii) will be ignored in these estimates. Since each matrix multiplication requires $2N^3$ operations, the total count for the standard TP method of [5] is $8N^3$.

2. FAST TP METHODS

One basic feature of the FFT method of [2] is the use of a Fast summation algorithm, in the spirit of Cooley and Tukey [1], to obtain all the $M = 2^p$ sums

$$Y(\ell) = \sum_{k=1}^{M-1} Z(k) \sin \frac{k\ell\pi}{M} \quad , \quad \ell = 1,\ldots,M \tag{2.1}$$

in $5M\log_2 M$ operations. This is done with Hockney's FOUR67 package [2].

The first observation is that each row or column in each matrix product, in steps (i) and (iii) of (1.8), may be calculated by (2.1) with $M = N$. For one matrix product we thus need $5N^2\log_2 N$ operations and for all four of them we get a total of $20N^2\log_2 N$. For moderate values of N , say $N = 128$, this already represents savings of about 85 % .

Another basic feature of the FFT method is the process of odd-even reduction [2]. The second observation is that this process can be also applied to formula (1.2). The reduced equations for the even lines may be written in the form

$$(\Delta_2 \otimes \Delta^2 - I_2 \otimes (2\Delta^2 \otimes \Gamma^2))v = b^* \ , \tag{2.2}$$

where we use the subscript 2 in a matrix A_2 to denote the replacement of N (in the entries or dimension of A) by $\frac{N}{2}$. In (2.2), b^* is the modified vector of the even lines, $\Gamma = 4\Delta - 36$, and $v = [u_2, u_4, \ldots, u_{N-2}]^T$. The odd-line

unknowns are determined via the rule,

$$\Gamma u_{2i+1} = b_{2i+1} - \Delta(u_{2i} + u_{2i+2}) \ , \tag{2.3}$$

i.e., with no effect on the leading term of the count. Thus, we only need to estimate the number of operations from solving (2.2).

Replacing $S\otimes S$ by $S_2\otimes S$ and repeating the process of (1.6), (1.7) we now get

$$\hat{D} = \Lambda(\Delta_2)\otimes\Lambda(\Delta)^2 - I_2\otimes(2\Lambda(\Delta)^2 + \Lambda(\Delta)^2) \tag{2.4}$$

and

$$v = (S_2\otimes S)\hat{D}^{-1}(S_2\otimes S)b^* \ . \tag{2.5}$$

Therefore, starting with $V(I,J) = b_{i,j}$, i even, we have the three-step algorithm

$$\text{(i)}\ \ V \leftarrow S_2VS \ , \quad \text{(ii)}\ \ V \leftarrow \hat{D}^{-1}V \ , \quad \text{(iii)}\ \ V \leftarrow S_2VS \ . \tag{2.6}$$

Again, considering only steps (i) and (iii), and taking the reduced dimensions into account, we now have an improved total of $10N^2\log_2 N$ operations.

Furthermore, one could use $(I_2\otimes S)$ in place of $S_2\otimes S$. As this corresponds to Fourier analysis-synthesis in only one direction, the resulting method will be equivalent to a high order FFT . Setting

$$v = (I_2\otimes S)\hat{v} \ , \tag{2.7}$$

and applying $I_2\otimes S$ on (2.2), from the left, we get

$$[\Delta_2\otimes\Lambda(\Delta)^2 - I_2\otimes(2\Lambda(\Delta)^2 + \Lambda(\Gamma)^2)]\hat{v} = \hat{b} \ , \tag{2.8}$$

$$\hat{b} = (I_2\otimes S)b^* \ . \tag{2.9}$$

The coefficient matrix in (2.8) can be rearranged to become block diagonal. Thus, we have a total $\frac{N}{2} - 1$ systems of magnitude $N \times N$. If $\mu_j \equiv 2\lambda_j^2 + (4\lambda_j - 36)^2$, each of these systems has coefficient matrix

$$\begin{bmatrix} \mu_j & \lambda_j^2 & 0 \\ \lambda_j^2 & & \lambda_j^2 \\ 0 & \lambda_j^2 & \mu_j \end{bmatrix}$$

Hence, the solution of (2.8) does not contribute to the leading term of the operation count. The only contribution comes from (2.7) and (2.9). To form $(I_2 \otimes S)Z$, where $Z = b^*$ or $\hat{v}$, means to calculate the summations (2.1) $N - 1$ times, taking $M = \frac{N}{2} - 1$. Therefore, $(I_2 \otimes S)Z$ is calculated in $2.5N^2\log_2 N$ operations and the total estimate is $5N\log_2 N$. This count is the same as the one of Hockney's FFT, only that in our case we have established it for a method of high accuracy.

3. CONCLUSIONS

We have obtained high order Fast Tensor Product (FTP) methods whose arithmetic operation count is $0(cN^2\log_2 N)$. If Fourier analysis in both directions is used, then the minimum value of the coefficient is $c = 10$ and the corresponding FTP method appears to be about two times slower than FFT, for which $c = 5$. However, this is not the case, because, due to its programming simplicity, the FTP methods have smaller overhead than FFT. Numerical experiments in [3] indicate that the FTP method with $c = 20$, (which should be about four times slower than FFT), is only about 1.5 times slower. These experiments were carried out on the Purdue CDC6500 on a set of test problems taken from Houstis, Lynch, Papatheodorou and Rice [4]. If Fourier analysis in one only dimension is used, the resulting FTP method has coefficient $c = 5$, is as fast as FFT and has the advantage of being 4th order accurate.

REFERENCES

[1] COOLEY, J.W. and TUKEY, J.W., An algorithm for the machine calculation of complex Fourier series, Math. Comp., 19, (1965), 297-301.

[2] HOCKNEY, R.W., The potential calculation and some applications, Methods in Comp. Physics, 9, (1970), 135-211.

[3] HOUSTIS, E.N. and PAPATHEODOROU, T.S., Comparison of Fast Methods for Elliptic Problems, to appear, Trans. IMACS. Also, in Proceedings, 2nd Intern. Symposium on Computer Methods for PDE , June 1977.

[4] HOUSTIS, E.N., LYNCH, R.E., PAPATHEODOROU, T.S. and RICE, J.R., Evaluation of numerical methods for elliptic partial differential equations, to appear, J. Comp. Physics.

[5] LYNCH, R.E., RICE, J.R. and THOMAS, D.H., Direct solution of partial differential equations by tensor product methods, Num. Math., 6, (1964), 185-199.

[6] ROSSER, J. Barkley, Nine-point difference solutions for Poisson's equation, Comp. & Maths. with Appls., 1, (1975), 357-360.

Department of Mathematics and
Computer Science
Clarkson College of Technology
Potsdam, New York 13676

Department of Mathematics
University of Crete
Heraklion, Crete
Greece