European Centre
for Medium Range
Weather Forecasts

# An Improved Algorithm for the Direct Solution of Poisson's Equation over Irregular Regions.

Internal Report 5.
Research Dept.

February 77.

Centre Européen pour les Prévisions Météorologiques a Moyen Terme

Europäisches Zentrum Für Mittelfristige Wettervorhersagen

# AN IMPROVED ALGORITHM FOR THE DIRECT SOLUTION OF POISSON'S EQUATION OVER IRREGULAR REGIONS

By

C. Temperton

European Centre for Medium Range Weather Forecasts, Bracknell

## 1. Introduction

Discrete Poisson and Helmholtz equations arise frequently in connection with gridpoint numerical models of the atmosphere; for example in decomposing a wind field into its rotational and divergent components, in solving the balance equation, and in semi-implicit time integration schemes.

The use of direct methods for solving the discrete Poisson equation ( or simple Helmholtz equation ) over a rectangular region is well established; a comprehensive review of these methods was presented recently by Temperton (1977). With a little modification, they can be applied to a latitude-longitude grid on the sphere (Swarztrauber,1974). If we denote the number of operations required to obtain a solution on a rectangular N x M grid by $\theta($ N,M$)$, then for the simplest methods, based on either Fourier analysis or block-cyclic reduction, $\theta(N,M)$ is asymptotically proportional to NM $\log_2 N$. (In the most recently developed form of block-cyclic reduction (Schumann and Sweet,1976), this is true for arbitrary N, i.e., N need not be a power of 2 ). Swarztrauber (1976) and Temperton (1977) have shown that Fourier analysis and block-cyclic reduction can be combined to give an operation count $\theta(N,M)$ asymptotically proportional to $N^2 \log_2 (\log_2 N)$ (for M=N), and in fact for practicable grid sizes $\theta$ (N,M) is effectively proportional to NM.

In many cases, it would be convenient to extend direct methods to non-rectangular regions; for example, a number of operational numerical weather prediction models are based on a stereographic map projection, using an octagonal domain covering most of a hemisphere. Buzbee et al. (1971) showed that these direct methods could also be used to solve the discrete Poisson (or Helmholtz) equation over an irregular region, by embedding it in a rectangle; a typical example is shown in Fig. 1. Suppose that the dimensions of the rectangular grid are N x M, and that the discrete Poisson equation can be solved over the rectangle in $\theta(N,M)$ operations. Suppose also that there are p gridpoints within the interior of the rectangle which form part of the boundary of the embedded irregular region. The solution procedure falls into two parts : a preprocessing phase which is independent of the right-hand side of the equation (and thus need only be performed once when the equation has to be solved a number of times with different right-hand sides ), and the solution phase itself. The algorithm given by Buzbee et al. (1971) required $p\,\theta(N,M) + O(p^3)$ operations for the preprocessing phase and $2\,\theta\,(N,M) + 2p^2$ operations for the solution phase. The development of the algorithm was given without reference to any specific direct method for the rectangular region.

Later, Buzbee and Dorr (1974) showed that the operation count for the preprocessing phase could be reduced to O $(pNM + p^3)$ by careful application of the Fourier analysis method. The solution phase still required 2 $\theta(N,M)$ + $2p^2$ operations, i.e. more than twice as many as for the corresponding rectangular problem.

In this paper, it is shown that the strategy used by Buzbee and Dorr to reduce the operation count for the preprocessing phase can be extended to the solution phase. We derive an algorithm whose operation count for each solution is not much greater than $\theta$ (N,M), showing that direct methods can handle irregular regions almost as efficiently as rectangles. As a bonus, the storage requirement is also reduced by approximately MN in comparison with the algorithm of Buzbee et al.

## 2. Development of a fast embedding algorithm

Some detailed aspects of the embedding procedure are discussed by Buzbee et al. (1971); for convenience they are reviewed briefly here.

Let the dimensions of the rectangular grid be N x M. Denote the boundary of the rectangle by $\partial R$, and the discrete interior (i.e., the set of interior gridpoints) of the rectangle by $R_h$. $R_h$ contains $n = (N-1)(M-1)$ gridpoints.

Denote the boundary of the embedded irregular region Q by $\partial Q$, and its discrete interior by $Q_h$. Let p be the number of points in $\partial Q \cap R_h$. The embedded irregular region may be a simple domain with a non-rectangular shape, as in Fig. 1, or it may be the rectangle itself with specified boundary conditions at a number of interior points, as in Fig. 2; for the potential problem (Hockney,1970) the latter case corresponds to the presence of internal electrodes.

For simplicity we assume here that the boundary conditions on $\partial Q$ are Dirichlet, so that the problem we have to solve is

$$\left.\begin{aligned} \nabla^2\psi &= f \text{ in } Q_h \\ \psi &= g \text{ on } \partial Q \end{aligned}\right\} \quad \ldots\ldots\ldots \qquad (1)$$

The functions f and g can be extended arbitrarily to the whole rectangle, e.g. by setting $f = o$ in $R_h - Q_h - (\partial Q \cap R_h)$ and $g = o$ on $\partial R - (\partial Q \cap \partial R)$. The problem (1) then becomes

$$\left.\begin{aligned} \nabla^2\psi &= f \text{ in } R_h - (\partial Q \cap R_h) \\ \psi &= g \text{ on } \partial Q \cup \partial R \end{aligned}\right\} \quad \ldots\ldots\ldots \qquad (2)$$

The discrete Poisson equation over the rectangle can be written as $B\underset{\sim}{x} = \underset{\sim}{y}$ , where B is an n x n matrix. Without loss of generality, we can assume that the points of the rectangular mesh are ordered in such a way that the first p points are those which lie in $\partial Q \cap R_h$.

Let B be partitioned in the form

$$B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}$$

where $B_1$ and $B_2$ are p x n and (n-p)xn matrices respectively. The problem (2) can then be written as

$$A\underset{\sim}{x} = \underset{\sim}{y} \tag{3}$$

where
$$A = \begin{pmatrix} A_1 \\ B_2 \end{pmatrix}$$

and $A_1$ is a pxn matrix of the form

$$A_1 = (I_p \quad 0).$$

The first p components of $\underset{\sim}{y}$ in Eq. (3) correspond to the boundary values g on $\partial Q \cap R_h$; the remaining components correspond to the function f, extended to $R_h - (\partial Q \cap R_h)$, and modified as necessary at points adjacent to $\partial Q \cap \partial R$ to accommodate nonhomogeneous boundary conditions. After solving the system (3), the components of $\underset{\sim}{x}$ corresponding to points outside Q may be discarded.

The preprocessing phase consists of determining the pxp capacitance matrix $C = A_1 B^{-1} A_1^T$ , and computing either the explicit inverse or a triangular factorization of C to be stored for use during the solution phase. An efficient method for determining C was given by Buzbee and Dorr (1974), and this aspect will not be considered further here.

The solution phase can be divided into the following steps:

(1) Given the problem represented by Eq.(1), extend the functions f and g to the whole rectangle, as described above.
(2) Solve the discrete Poisson equation $B\underset{\sim}{w} = \underset{\sim}{y}$ over the rectangle.
(3) Solve $Cv = A_1(\underset{\sim}{y} - \underset{\sim}{w})$; this system only involves the p points of $\partial Q \cap R_h$.
(4) Set $\underset{\sim}{z} = \underset{\sim}{y} + A_1^T \underset{\sim}{v}$ (i.e. modify the original right-hand side at the points of $\partial Q \cap R_h$) .
(5) Solve the discrete Poisson equation $B\underset{\sim}{x} = \underset{\sim}{z}$. The vector x is the required solution in $Q_h$, and matches the specified boundary conditions on $\partial Q \cap R_h$.

Notice that in step (2) the solution vector w cannot overwrite the input vector y, since the latter is required again in step (4).

The above procedure is precisely that outlined by Buzbee et al. (1971) and Buzbee and Dorr (1974). We now consider the application in steps (2) and (5) of a specific simple method for the solution of the discrete Poisson equation over a rectangle, namely the "basic FFT" method. To make the discussion more concrete, let the embedded irregular region be an octagon of the form illustrated in Fig. 1. The octagon is obtained by removing four triangular corners of the rectangle; with the notation in Fig. 1, $p = 4\,(q-1)$.

For simplicity, we assume that the mesh length is 1. Let S be the matrix representation of a Fourier sine transform; thus S is a square matrix of order $(N-1)$ with elements given by $S = (s_{ij})$, where $s_{ij} = (2/N) \sin(ij\,\pi/N)$. With this scaling, $S^{-1} = (N/2)\,S$. Let $\lambda_k = 2 \cos(k\,\pi/N) - 4$, and denote by $T_k$ the tridiagonal matrix of order $(M-1)$ with constant diagonal term $\lambda_k$, and 1's on the sub- and super-diagonals.

In order to keep the notation compact in the following discussion, we adopt the following definitions. For a variable x defined at gridpoints, let

$$\underset{\sim}{x}_j = \begin{pmatrix} x_{1,j} \\ x_{2,j} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{N-1,j} \end{pmatrix} .$$

Denote the Fourier sine transform of $\underset{\sim}{x}_j$ by $\bar{\underset{\sim}{x}}_j$, i.e. $\bar{\underset{\sim}{x}}_j = S\underset{\sim}{x}_j$, with components defined by

$$\bar{\underset{\sim}{x}}_j = \begin{pmatrix} \bar{x}_{1,j} \\ \bar{x}_{2,j} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \bar{x}_{N-1,j} \end{pmatrix} ,$$

where the first subscript of each component refers to the wavenumber k. We will also wish to reorder the variables in Fourier space and define a vector of values with a common wavenumber:

$$\underset{\sim}{\bar{x}}_k = \begin{pmatrix} \bar{x}_{k,1} \\ \bar{x}_{k,2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \bar{x}_{k,M-1} \end{pmatrix}$$

where the second subscript refers to the row number j.

Step (2) can now be broken down into three smaller steps:
(2a) Perform a Fourier sine transform on each row j, i.e. calculate $\bar{\underset{\sim}{y}}_j = S\underset{\sim}{y}_j$, $1 \le j \le M-1$.
(2b) For each wavenumber k ( $1 \le k \le N-1$) , solve the tridiagonal system $T_k \underset{\sim}{\bar{w}}_k = \underset{\sim}{\bar{y}}_k$.
(2c) Perform an inverse Fourier sine transform on each row j, i.e. calculate $\underset{\sim}{w}_j = S^{-1}\, \underset{\sim}{\bar{w}}_j$, $1 \le j \le M-1$.

Step (5) can be broken down in exactly the same way, with $\underset{\sim}{x}$ replacing $\underset{\sim}{w}$ and $\underset{\sim}{z}$ replacing $\underset{\sim}{y}$.

For this basic method of solving the discrete Poisson equation over a rectangle, the operation count given by Temperton (1977) is $\theta(N,M) = (3 \log_2 N + 7)$ NM additions and $(2 \log_2 N + 1)$ NM multiplications, assuming that the tridiagonal systems are solved by Gaussian elimination using precalculated coefficients. Steps (2) and (5) each require $\theta(N,M)$ operations, while step (3) requires $p^2$ additions and $p^2$ multiplications, giving an operation count for the whole algorithm of $(6 \log_2 N + 14)\, NM + p^2$ additions and $(4 \log_2 N + 2)\, NM + p^2$ multiplications.

A cursory examination of this algorithm shows that some of the computation is redundant. For example, $\underset{\sim}{z}_j$ in step (5) differs from $\underset{\sim}{y}_j$ in step (2) only on those rows containing one or more of the p points in $\partial Q \cap R_h$. Thus, for the example in Fig. 1, $\underset{\sim}{z}_j = \underset{\sim}{y}_j$ ( and hence $\bar{\underset{\sim}{z}}_j = \bar{\underset{\sim}{y}}_j$ ) for $q \le j \le M-q$, and so, provided $\bar{\underset{\sim}{y}}_j$ can be stored, step (5a) need not be carried out for these values of j.

A more searching analysis shows that nearly half the computation is redundant. Consider first step (3); in order to perform this step we need the components of the vector $\underset{\sim}{w}$ only at the p points in $\partial Q \cap R_h$ . These can be obtained more efficiently as follows : perform steps (1), (2a) and (2b) above to obtain $\bar{w}_j$, $1 \le j \le M-1$. Then, for each point (i,j) in $\partial Q \cap R_h$, $w_{i,j}$ is given directly by

$$w_{i,j} = \sum_{k=1}^{N-1} \bar{w}_{k,j} \sin(ik\pi/N) \ldots \tag{4}$$

Exploiting the symmetry of the sine function, Eq. (4) becomes

$$w_{i,j} = \sum_{k=1}^{N/2-1} (\bar{w}_{k,j} - (-1)^i \bar{w}_{N-k,j}) \sin(ik\pi/N) + \bar{w}_{N/2,j} \sin(i\pi/2) \ldots\ldots \tag{5}$$

So, for each $(i,j)$ in $\partial Q \cap R_h$, $w_{i,j}$ can be calculated in approximately N additions and N/2 multiplications. Direct use of Eq. (4) was suggested by Buzbee and Dorr (1974) in their application of the capacitance matrix method to the discrete biharmonic equation.

The modified form of step (2c) thus requires pN additions and pN/2 multiplications, in comparison with the original $(1.5 \log_2 N + 2.5)$ NM additions and $(\log_2 N - 0.5)$ NM multiplications. Moreover, for the octagonal region of Fig. 1, if $(i,j)$ is in $\partial Q \cap R_h$, then so is $(N-i,j)$ and

$$w_{N-i,j} = -\sum_{k=1}^{N/2-1} (-1)^k (\bar{w}_{k,j} - (-1)^i \bar{w}_{N-k,j}) \sin(ik\pi/N) - (-1)^{N/2} \bar{w}_{N/2,j} \sin(i\pi/2) \ldots \tag{6}$$

Comparison of Eqs. (5) and (6) shows that $w_{i,j}$ and $w_{N-i,j}$ can be computed together in a total of N additions and N/2 multiplications, further reducing the operation count for the modified form of step (2c) to pN/2 additions and pN/4 multiplications.

Having obtained $w_{i,j}$ for all $(i,j)$ in $\partial Q \cap R_h$, steps (3) and (4) can proceed as before. We turn now to the modifications required in step (5). In step (5a) we calculate $\bar{z}_j = S z_j$, $1 \leq j \leq M-1$. It has already been pointed out that, if row j contains no points in $\partial Q \cap R_h$, then $z_j = y_j$ and $\bar{z}_j = \bar{y}_j$, which has already been determined in step (2a). Suppose now that row j contains one of the p points in $\partial Q \cap R_h$, namely $(i,j)$. Then $z_j$ differs from $y_j$ only at this point. Let $z_j = y_j + e_j$.

We have

$$\underset{\sim}{\bar{z}}_j = S\underset{\sim}{z}_j = S(\underset{\sim}{y}_j + \underset{\sim}{e}_j) = \underset{\sim}{\bar{y}}_j + \underset{\sim}{\bar{e}}_j ,$$

where $\underset{\sim}{\bar{e}}_j$ is the Fourier sine transform of a vector with only one nonzero component; hence

$$\bar{z}_{k,j} = \bar{y}_{k,j} + (2/N)e_{i,j} \sin (ik\,\pi/N).... \qquad (7)$$

Again, we can exploit the symmetries of the sine function to obtain

$$\bar{z}_{N-k,j} = \bar{y}_{N-k,j} - (2/N)\ (-1)^i\ e_{i,j} \sin(ik\pi/N).. \qquad (8)$$

If row j contains more than one point in $\partial Q \cap R_h$ then Eq. (7) is replaced by

$$\bar{z}_{k,j} = \bar{y}_{k,j} + (2/N)\sum_i e_{i,j} \sin (ik\,\pi/N).... \qquad (9)$$

where the sum is taken over values of i such that (i,j) is in $\partial Q \cap R_h$; and similarly for Eq. (8). This modified form of step (5a) requires pN additions and pN/2 multiplications, the same as for the modified form of step (2c). In the case of the octagon illustrated in Fig. 1, we can again exploit the fact that if (i,j) is in $\partial Q \cap R_h$ then so is (N-i,j). Eq. (9) becomes

$$\bar{z}_{k,j} = \bar{y}_{k,j} + (2/N)\ (e_{i,j} - (-1)^k\ e_{N-i,j}) \sin(ik\pi/N),$$

and similarly for $\bar{z}_{N-k,j}$ and the operation count is reduced to pN/2 additions and pN/4 multiplications.

Steps (5b) and (5c) then proceed as before.

Although we have now saved a considerable amount of computation in comparison with the original algorithm, we have not reduced the storage requirements; the vectors $\underset{\sim}{\bar{y}}_j$, $1 \le j \le M-1$, must be saved from step (2a) as they are required again in step (5a). A further modification reduces the storage requirements for the arrays from 2NM + p to NM + p + M.

Consider step (5b), in which we solve the tridiagonal systems

$$\bar{\underset{\sim}{x}}_k = T_k^{-1}\ \bar{\underset{\sim}{z}}_k \quad , \quad 1 \leq k \leq N-1 \qquad \ldots\ldots \tag{10}$$

Now $\bar{\underset{\sim}{z}}_k = \bar{\underset{\sim}{y}}_k + \bar{\underset{\sim}{e}}_k$, and we already have $T_k^{-1}\bar{\underset{\sim}{y}}_k = \bar{\underset{\sim}{w}}_k$ from step (2b). Eq. (10) can thus be written

$$\bar{\underset{\sim}{x}}_k = \bar{\underset{\sim}{w}}_k + T_k^{-1}\bar{\underset{\sim}{e}}_k.$$

Step (5a) can be modified to obtain $\bar{e}_{k,j}$ rather than $\bar{z}_{k,j}$ ; for example Eq. (9) is replaced by

$$\bar{e}_{k,j} = (2/N) \sum_i e_{i,j} \sin\,(ik\,\pi/N)\ldots\ldots \tag{11}$$

where again the sum is taken over values of i such that (i,j) is in $\partial Q \cap R_h$.

With this modification, the vectors $\bar{y}_j$ need no longer be retained, and by performing steps (5a) and (5b) for each wavenumber k before proceeding to wavenumber (k+1), the storage requirement is reduced.

The fast embedding algorithm thus consists of the following steps.

(1) Extend the functions f and g to the whole rectangle to define the vector $\underset{\sim}{y}$. Save the components of $\underset{\sim}{y}$ corresponding to the p points of $\partial Q \cap R_h$ in a work area.

(2a) Perform a Fourier sine transform on each row j, i.e. calculate $\bar{\underset{\sim}{y}}_j = S\underset{\sim}{y}_j$, $1 \leq j \leq M-1$.

(2b) For each wavenumber k, $1 \leq k \leq N-1$, solve the tridiagonal system $T_k \bar{\underset{\sim}{w}}_k = \bar{\underset{\sim}{y}}_k$ .

(2c) For each point (i,j) in $\partial Q \cap R_h$, calculate $w_{i,j}$ using Eq. (4), and overwrite the work area with values of $(y_{i,j} - w_{i,j})$ at these points, thus defining the vector $A_1(\underset{\sim}{y}-\underset{\sim}{w})$.

(3) Solve $C\underset{\sim}{v} = A_1\,(\underset{\sim}{y}-\underset{\sim}{w})$.

(4) Define $\underset{\sim}{e} = A_1^T\underset{\sim}{v}$ (i.e., the components of e are the same as those of $\underset{\sim}{v}$ for (i,j) in $\partial Q \cap R_h$, and zero elsewhere).

It is convenient to multiply the components by (2/N) at this point.

(5a,5b) For each wavenumber k, calculate $\bar{e}_k$ using Eq.(11) and then calculate $\bar{x}_k = \bar{w}_k + T_k^{-1}\bar{e}_k$. An additional work area of length M is required. For irregular regions with symmetry about i = N/2, it is more efficient to consider wavenumbers k and (N-k) together, since for example in the octagon of Fig. 1 $\bar{e}_{N-k,j} = -(-1)^i\,\bar{e}_{k,j}$, where (i,j) is in $\partial Q \cap R_h$.

(5c) Perform an inverse Fourier sine transform on each row j, i.e. calculate $x_j = S^{-1}\bar{x}_j$, $1 \leq j \leq M-1$. The vector $x$ is the required solution.

In comparison with the original algorithm, use of the fast embedding procedure reduces the operation count from $(6 \log_2 N + 14)\,NM + p^2$ additions and $(4 \log_2 N+2)\,NM + p^2$ multiplications to $(3 \log_2 N + 10)\,NM + pN + p^2$ additions and $(2 \log_2 N + 3)\,NM + pN+p^2$ multiplications, while the storage requirement for the arrays is reduced from 2NM + p to NM + M + p. Instead of having to solve the Poisson equation over the rectangle twice, the only O (NM) part of the computation which as to be repeated is the solution of (N-1) tridiagonal systems of order (M-1). For typical values p = N = M = 64, the total number of floating-point operations is reduced by over 40%.

The development of the fast embedding algorithm arose from a requirement to solve the discrete Poisson equation over an octagon with N = M = 60, q = 18 (p=68). Programs to solve the equation over 60 x 60 square (using the basic FFT method), and over the embedded octagon (using the algorithm above), were written in Assembler language and run on an IBM 360/195 at the U.K. Meteorological Office. The respective CPU times were : for the square, $3.63 \times 10^{-2}$ seconds; for the octagon, $4.41 \times 10^{-2}$ seconds. Experiments were carried out to compare the computed solutions with true solutions consisting of random numbers in the range $-1 \leq x_{ij} \leq 1$; the mean maximum error was found to be $9.8 \times 10^{-5}$ for both regions.

## 3. Combining fast embedding with a fast solver

At this juncture the following objection may be raised: the development of the fast embedding algorithm of Section 2 depends on the use of the "basic FFT" method for solving Poisson's equation over the rectangle. In this case, the use of fast embedding reduces the operation count by nearly half. However, a similar reduction could be obtained by using the basic embedding method of Buzbee et al. (1971) together with a faster Poisson-solver on the rectangle. Consider for example the unstabilized FACR(1) algorithm of Hockney (1965, 1970); the operation count given for this algorithm by

Temperton (1977) is (1.5 $\log_2 N$+ 7.5) NM additions and ($\log_2$ N + 2) NM multiplications for the rectangle, again assuming that the tridiagonal systems are solved by Gaussian elimination using precomputed coefficients. Combining basic embedding with two solutions over the rectangle using this algorithm would therefore require a total of ( 3 $\log_2$ N + 15) NM + $p^2$ additions and (2 $\log_2$ N + 4 ) NM + $p^2$ multiplications, which is similar to the operation count given in Section 2 for the combination of fast embedding and basic Poisson-solver. Similar operation counts would again be obtained by combining basic embedding with either one of two alternative variants of the FACR (1) algorithm introduced by Temperton (1977), in which the original set of equations for all (i,j) is reduced to a new set for i even (cyclic reduction on i) or for ( i + j) even ( "diagonal" cyclic reduction ).

Applying the same principles as in Section 2, the fast embedding procedure can in fact be combined with any of these fast Poisson-solvers. The details will not be given here, since they depend on which fast solver is used.

The choice of the most convenient algorithm depends on the position of the points (i,j) in $\partial Q \cap R_h$. If j is even for all such (i,j), as in Fig. 2, then Hockney's FACR (1) algorithm (with cyclic reduction on j) is the most suitable; if i is even for all such (i,j), as in Fig. 4, then the algorithm using cyclic reduction on i should be used; if ( i+j) is even for all such (i,j), as in Fig. 1 if N,M and q are even, then "diagonal" cyclic reduction is the most convenient.

The latter algorithm can also be used in the most general case, since if the solution $w_{i,j}$ in step (2c) above is required at a point (i,j) with (i+j) odd, it can easily be obtained from the solution at the neighbouring points ( i ± 1,j) and (i, j ± 1) since these points are all <u>even.</u> In Hockney's FACR (1) algorithm, on the other hand, the solution at a single point (i,j) with j odd can only be found by solving a tridiagonal system, the right-hand side of which depends on the solution at (i,j + 1) and (i, j - 1 ) for <u>all</u> i.

For the octagon problem with N = M = 60, q = 18, the approximate number of floating-point operations per point is given in Table 1, for basic and fast embedding combined with both basic and fast Poisson-solvers. The fast Poisson-solver in this case uses one preliminary step of "diagonal" cyclic reduction.

The amount of computation required to obtain the solution over the octagon, combining fast embedding with a fast Poisson-solver, is approximately equal to the amount required for just 2 complete sweeps of the iterative ADI method.

Table 1

Number of floating-point operations per point for octagon problem with $N = M = 60$, $q = 18$.

| Poisson-solver | embedding | additions | multiplications |
|---|---|---|---|
| basic | basic | 53 | 31 |
| basic | fast | 30.5 | 18.5 |
| fast | basic | 34.5 | 17.5 |
| fast | fast | 20.5 | 12 |

## 4. Further applications

The preceding sections have demonstrated the power of FFT-based Poisson-solvers for applications to irregular regions, especially when combined with the fast embedding algorithm. Buzbee et al. (1971) remark that "embedding the region in a rectangle may introduce an excessively large number of additional unknowns that are not necessary to the solution of the original problem"; an example might be the L-shaped region of Fig. 3. However, by noting that the sine transforms on the lines containing only a few points in $Q_h$ can rapidly be performed directly, using only elementary symmetries of the sine function rather than the FFT, an efficient algorithm may still be constructed. It is worth noting that block-cyclic reduction (Buneman's algorithm) appears to be much less flexible in this respect.

Another application of the embedding procedure which may be useful is in solving the discrete Poisson equation over simple rectangular regions whose dimensions are particularly inconvenient for the FFT. Suppose for example that we have to solve Poisson's equation on an $N \times M$ rectangle with $N = M = 62$. Hockney's FACR (1) algorithm would require sine transforms on 62 points; using the procedures of Cooley, Lewis and Welch (1970) to reduce these to periodic transforms on 31 complex points, and the method due to Singleton (1969) for performing these efficiently, the operation count for the Poisson-solver is approximately 27 additions and 19.5 multiplications per point, compared with 16.5 additions and 8 multiplications for a similar problem with $N = 64$.

Although these figures may be taken as an indication that the Fourier transform method is reasonably efficient even for awkward values of N, a faster algorithm can be obtained by embedding the N = 62 rectangle in a slightly larger one with N = 64 (see Fig.4), and combining the fast embedding procedure with the variant of the FACR (1) algorithm which uses cyclic reduction on i. The resulting operation count is approximately 20 additions and 11.5 multiplications per point.

Recent results (Schumann and Sweet, 1976) suggest that block-cyclic reduction can be applied just as efficiently for arbitrary N as for N a power of 2; however, the combination of fast embedding with fast FFT-based Poisson-solvers may be quite competitive.

## 5. Acknowledgement

Most of this work was carried out while the author was at the U.K. Meteorological Office.

References :


Buzbee, B.L., Dorr, F.W., George, J.A. and Golub, G.H. (1971) "The direct solution of the discrete Poisson equation on irregular regions", SIAM J. Numer. Anal. 8, 722-736

Buzbee, B.L. and Dorr, F.W. (1974) "The direct solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions", SIAM J. Numer.Anal. 11, 753-763

Cooley, J.W., Lewis, P.A.W. and Welch, P.D. (1970) "The Fast Fourier Transform algorithm: programming considerations in the calculation of sine, cosine and Laplace transforms", J. Sound Vib. 12,315-337

Hockney, R.W. (1965) "A fast direct solution of Poisson's equation using Fourier analysis", J. Assoc.Comp.Mach. 12 95-113.

Hockney, R.W. (1970) "The potential calculation and some applications", Methods in Computational Physics 9,135-211.

Schumann, U. and Sweet, R.A. (1976) "A direct method for the solution of Poisson's equation with Neumann boundary conditions on a staggered grid of arbitrary size", J. Comp. Phys. 20,171-182.

Singleton, R.C. (1969) "An algorithm for computing the mixed radix Fast Fourier Transform", IEEE Transactions on Audio and Electroacoustics, 17, 93-103.

References :


Swarztrauber, P.N. (1974) "The direct solution of the discrete Poisson equation on the surface of a sphere", J. Comp. Phys. 15, 46-54.

Swarztrauber, P.N. (1976) "The methods of cyclic reduction, Fourier analysis and cyclic reduction - Fourier analysis for the discrete solution of Poisson's equation on a rectangle", ( To appear )

Temperton, C. (1977) "Direct methods for the solution of Poisson's equation: some comparisons" ( In preparation ).

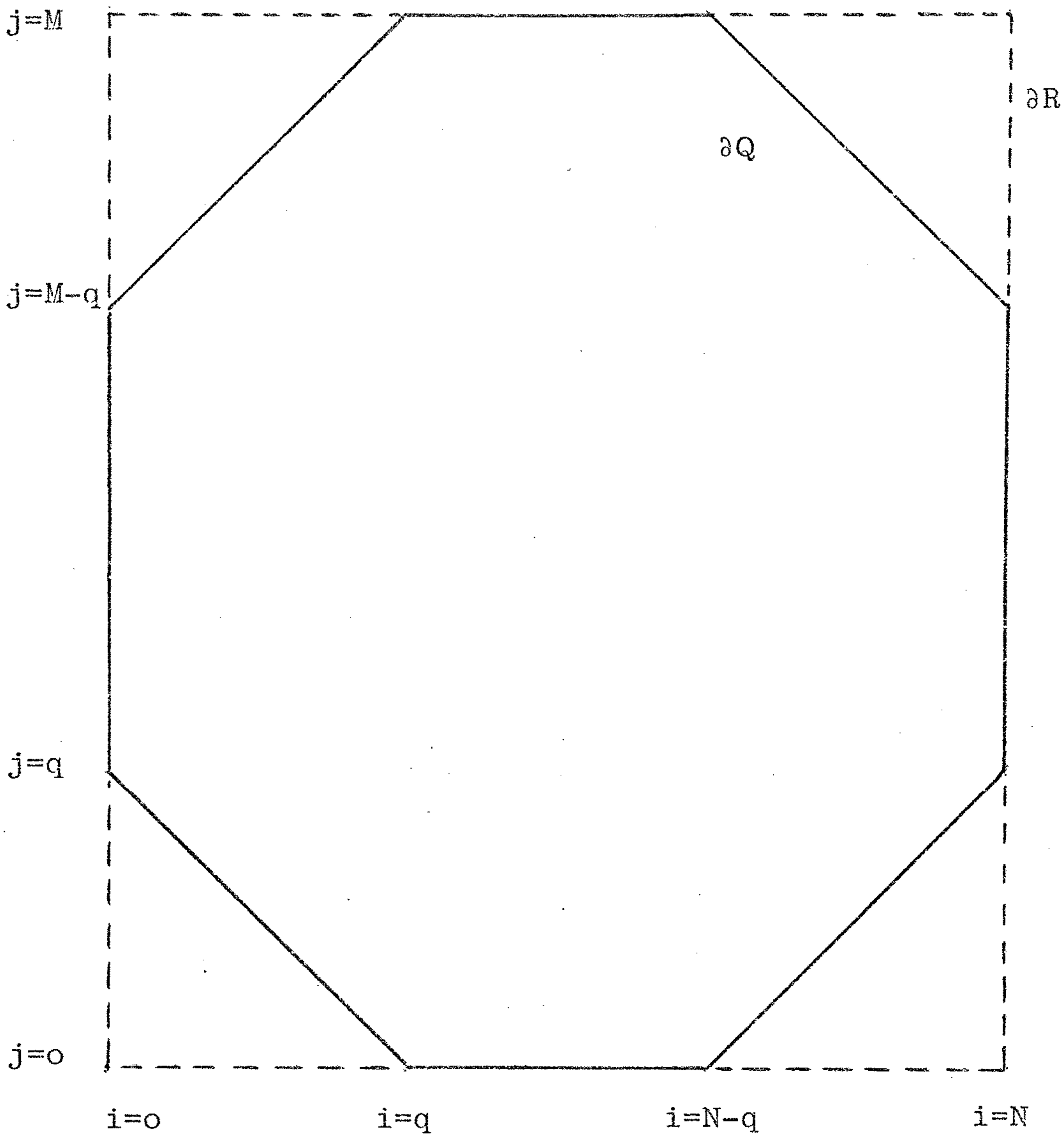


Fig. 1 Octagonal region embedded in rectangle.

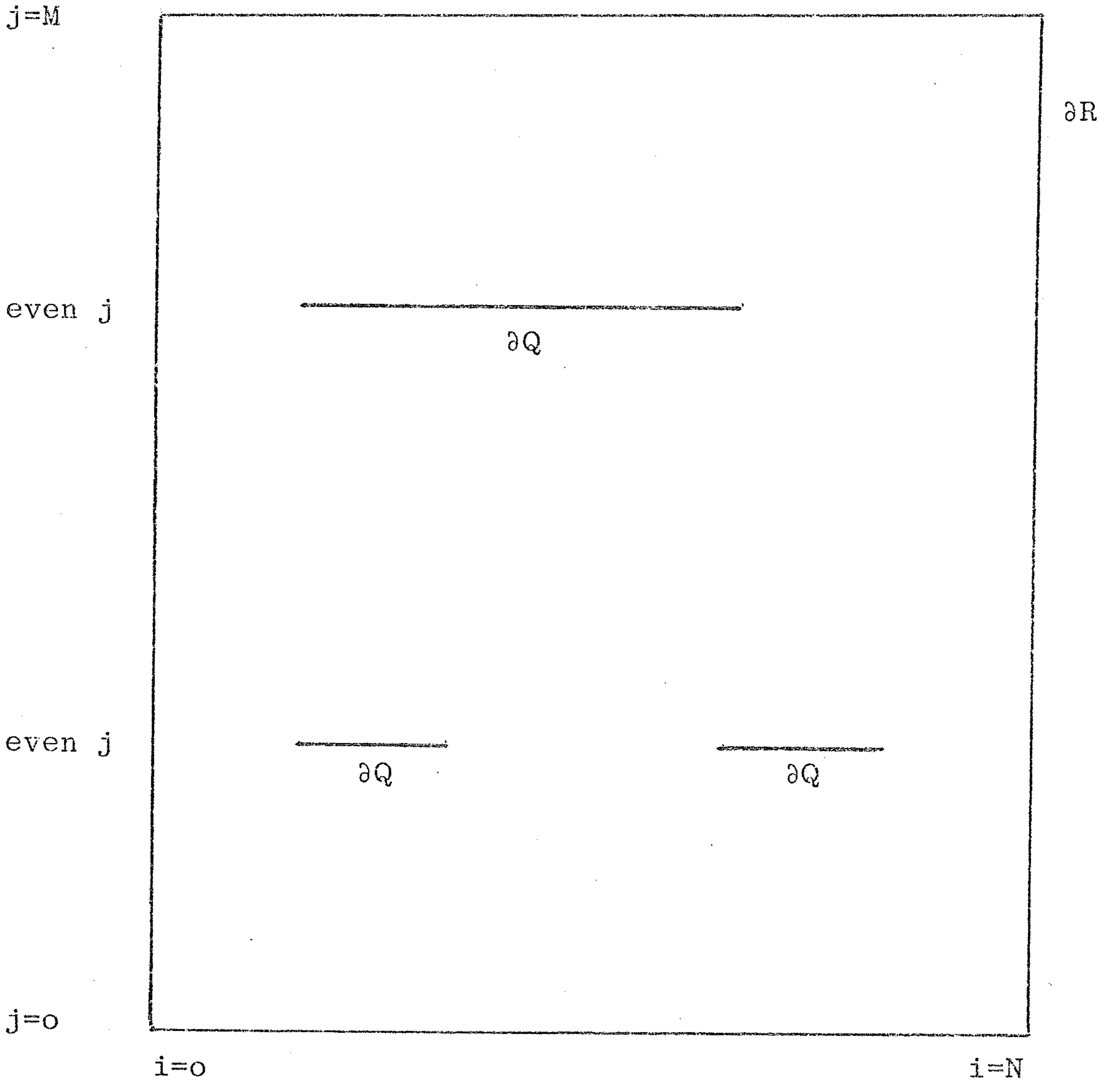


Fig. 2 Rectangle with internal boundaries.

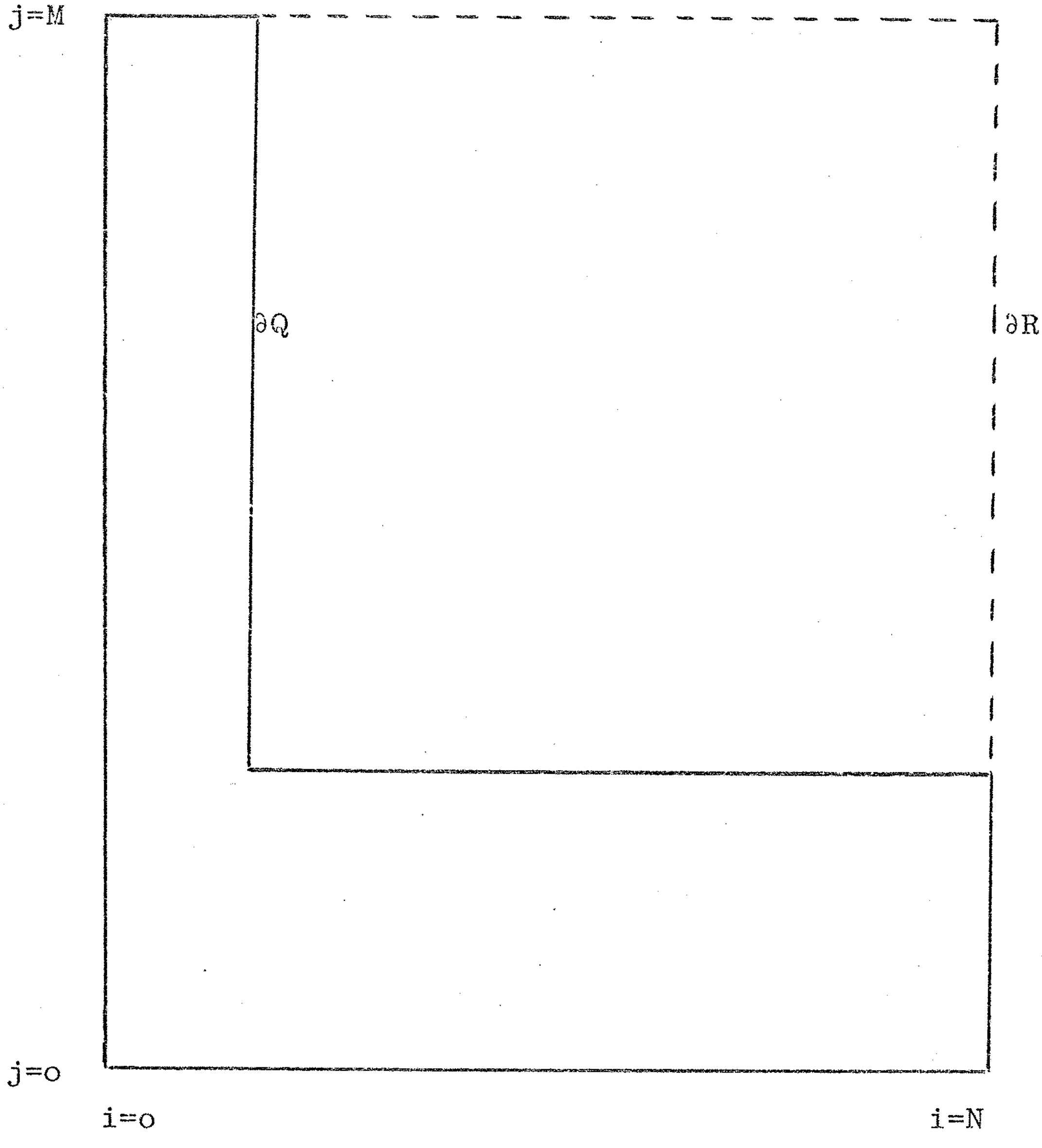


Fig. 3 L-shaped region embedded in rectangle

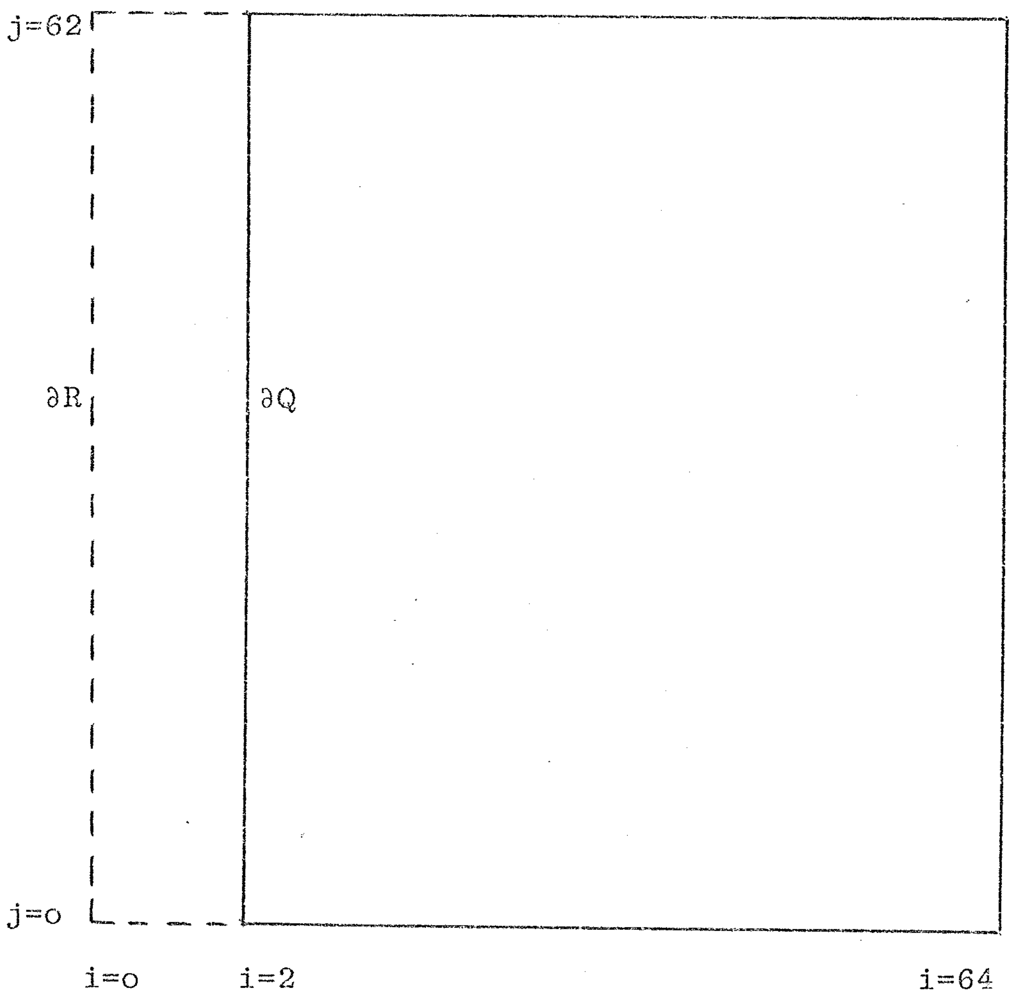


Fig. 4  62 x 62 rectangle embedded in 64x62 rectangle

EUROPEAN CENTRE FOR
MEDIUM RANGE WEATHER
FORECASTS

Research Department (RD)

Internal Report No. 5