problem. This approach has also been discussed by George [13] in connection with some iterative imbedding algorithms. We shall first describe Hockney's approach briefly, and then an efficient implementation of it will be presented using the symmetric marching technique.

Inverting the partitioned matrix in (4), we obtain

$$\begin{bmatrix} v_D \\ v_T \\ v_Q \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} W_D \\ W_T \\ 0 \end{bmatrix}, \tag{5}$$

and solving (5) for $B_{22}w_T$ gives

$$B_{22}w_T = v_T - B_{21}w_D. \tag{6}$$

Since the matrix B_{22} is positive definite, w_Q has been set equal to zero. A fast direct method can be used to solve the system

$$\begin{bmatrix} A_{11} & A_{12} & \\ A_{21} & A_{22} & A_{23} \\ & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} \bar{v}_D \\ \bar{v}_T \\ \bar{v}_Q \end{bmatrix} = \begin{bmatrix} w_D \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

to obtain $B_{21}w_D$ as $\bar{v}_T$. Having got $\bar{v}_T$, the w_T is then obtained by solving

$$B_{22}w_T = v_T - \bar{v}_T. \tag{8}$$

The method thus requires B_{22}, which means that we need p (the number of grid points on T) corresponding columns of the inverse of the coefficient matrix A. This method, therefore, requires solving $p+2$ systems of the form (3), and the solution of the p linear equations given by (8).

We now describe the computational procedure for the efficient implementation of the above technique using the SMT.

Step 1: SMT and the Fast Generation of the Capacitance Matrix B_{22} $(p \times p)$. In order to generate the p columns of B_{22}, corresponding to p grid points on T, p calls of the SMT applied to every grid point in S will be required. For the kth $(1 \leqslant k \leqslant p)$ call of the SMT, the following finite difference formulas will be used:

$$u_{i,j+1} = 4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1}. \tag{9}$$

This equation holds for all points belonging to $D \cup Q \cup T$-{kth grid point on T}. Also

$$u_{i,j+1} = 4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - 1 \tag{10}$$

for the kth grid point on T.

The error propagation equation corresponding to the formulas (9) and (10) will be [6]

$$e_{i,j+1} = 4e_{i,j} - e_{i-1,j} - e_{i+1,j} - e_{i,j-1} \tag{11}$$

for all points belonging to $D \cup Q \cup T$, where $e_{i,j} = u_{i,j} - u'_{i,j}$, and $u'_{i,j}$ denotes a provisional value for $u_{i,j}$. Using the formulas (9), (10), and (11), the SMT as described in [6] is applied on the rectangle S, and the components of the kth column of the matrix B_{22} are then the solution components corresponding to the p grid points on T.

It is to be noted that the error propagation equation (11) remains fixed for all the p calls of SMT, and thus the influence coefficient matrix [6] is to be computed only once and then employed for all subsequent calls of SMT. The major computational effort in SMT is the construction of the influence coefficient matrix; once this matrix is constructed and its LU decomposition stored, the subsequent calls of SMT will require very little extra computational effort.

Step 2: Construction of the Vector $\bar{v}_T$ $(p \times 1)$. The finite difference formulas for the SMT to construct $\bar{v}_T$ are given by

$$u_{i,j+1} = 4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} + h^2 g_{i,j} \qquad \text{for all points in } D \tag{12}$$

and

$$u_{i,j+1} = 4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} \qquad \text{for all points in } T \cup Q. \tag{13}$$

Again the error propagation equation for this case will be the same as Equation (11), and consequently the influence coefficient matrix of step 1 will be employed. Thus the construction of the vector $\bar{v}_T$, like the SMT solution components corresponding to the p grid points on T, requires little computational effort.

Step 3: Computation of the Vector w_T $(p \times 1)$. Having computed the vector $\bar{v}_T$ in step 2, the vector w_T is obtained by solving the system of p linear equations

$$B_{22} w_T = v_T - \bar{v}_T \tag{14}$$

using Gaussian elimination.

Step 4: Computation of the Final Solution in the Region D. This final call of SMT will require the following finite difference formulas:

$$u_{i,j+1} = 4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} + h^2 g_{i,j} \qquad \text{for all points in } D \tag{15}$$

and

$$u_{i,j+1} = 4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - w_T \qquad \text{for all points in } T. \tag{16}$$

Once again the error propagation equation for this case will be the same as (11), and the influence coefficient matrix of step 1 will work here too.

3. MILDLY NONLINEAR ELLIPTIC EQUATIONS

The application of fast direct methods to nonlinear elliptic equations has attracted little attention so far. A very special case of mildly nonlinear elliptic equations of the type $-\Delta u + cu = f$, where c is a constant (Helmholtz's equation) has however been considered by a few workers. For example, Proskurowski and Widlund [19] used a capacitance matrix approach to deal with such equations, and Hyman [18] has described a local inversion method (LIM).

In this section, we shall consider the entension of the SMT to mildly nonlinear elliptic equations of the type

$$u_{xx} + u_{yy} = F(x, y, u) \tag{17}$$

over a simply connected bounded region D whose boundary ∂D is piecewise regular, where the function F is continuous. We shall be concerned here with

the associated Dirichlet problem of finding $u(x, y)$ which, for a given $f(x, y)$ continuous on ∂D, satisfies:

(i) u is defined and continuous on $D \cup \partial D$,
(ii) u satisfies (17), and
(iii) $u = f$ on ∂D.

To ensure the existence of unique solution for the above problem, it is further assumed that $f(x, y) \geqslant 0$. For the sake of brevity, we shall assume the region D here to be a unit square. The main idea of the method is to employ the SMT to carry out the iterations in the iterative scheme [14], based on solving a sequence of linear problems. The resulting analytical iteration formula, based on Pohozaev's analytical method, for (17) gives

$$\Delta u^{(n+1)} - F_u\left(x, y, u^{(n)}\right) u^{(n+1)} = F\left(x, y, u^{(n)}\right) - F_u\left(x, y, u^{(n)}\right) u^{(n)},$$

$$n = 0, 1, 2 \ldots . \quad (18)$$

The above iteration formula is also popularly known as quasilinearization [5]. Using the usual five point analogue for the Laplacian operator Δ, the discretized form of (18) gives

$$\frac{-4u_{i,j}^{(n+1)} + u_{i+1,j}^{(n+1)} + u_{i-1,j}^{(n+1)} + u_{i,j+1}^{(n+1)} + u_{i,j-1}^{(n+1)}}{h^2} - F_u\left(ih, jh, u_{i,j}^{(n)}\right) u_{i,j}^{(n+1)}$$

$$= F\left(ih, jh, u_{i,j}^{(n)}\right) - F_u\left(ih, jh, u_{i,j}^{(n)}\right) u_{i,j}^{(n)},$$

$$2 \leqslant i, j \leqslant N-1, \quad n = 0, 1, 2, \quad (19)$$

The finite difference formula (19) can be rewritten as

$$u_{i,j+1}^{(n+1)} = \left[4 + h^2 F_u\left(ih, jh, u_{i,j}^{(n)}\right)\right] u_{i,j}^{(n+1)} - u_{i+1,j}^{(n+1)} - u_{i-1,j}^{(n+1)} - u_{i,j-1}^{(n+1)}$$

$$+ h^2\left[F\left(ih, jh, u_{i,j}^{(n)}\right) - F_u\left(ih, jh, u_{i,j}^{(n)}\right) u_{i,j}^{(n)}\right]$$

$$2 \leqslant i, j \leqslant N-1, \quad n = 0, 1, 2 \ldots . \quad (20)$$

The error propagation equation corresponding to the difference formula (20) takes the form

$$e_{i,j+1}^{(n+1)} = \left[4 + h^2 F_u\left(ih, jh, u_{i,j}^{(n)}\right)\right] e_{i,j}^{(n+1)} - e_{i+1,j}^{(n+1)} - e_{i-1,j}^{(n+1)} - e_{i,j-1}^{(n+1)},$$

$$2 \leqslant i, j \leqslant n-1. \quad (21)$$

Each iteration in the iterative scheme (20) is carried out using SMT.

Convergence Criterion

The approximate solution values inside the region at each stage of the iteration are used to compute the boundary values corresponding to the bottom and the top edge of the square. Then the computed boundary values are compared with the given boundary values to obtain the following stopping criterion: If

$$|u_{i,1} - \bar{u}_{i,1}| < \varepsilon \text{ and } |u_{i,N} - \bar{u}_{i,N}| < \varepsilon, \qquad i = 2,3,\ldots,N-1,$$

then stop the iteration; here $u_{i,1}, u_{i,N}$ are the given boundary values and $\bar{u}_{i,1}, \bar{u}_{i,N}$ are the computed boundary values on the bottom and the top edge of the square respectively.

4. TEST EXAMPLES AND NUMERICAL RESULTS

To demonstrate the efficiency of the methods discussed, the following test examples have been solved:

EXAMPLE 1. $\Delta u = 0$, $(x, y) \in R$ [R is the region given in Figure 1(a)]:

$$u(x, y) = x^2 - y^2, \qquad (x, y) \in \partial R.$$

EXAMPLE 2. $\Delta u = 2(x + y)$, $(x, y) \in R$:

$$u(x, y) = xy(x + y), \qquad (x, y) \in \partial R.$$

EXAMPLE 3. $\Delta u = 0$, $(x, y) \in S$ [S is the region given in Figure 1(b)]:

$$u(x, y) = x^2 - y^2, \qquad (x, y) \in \partial S.$$

EXAMPLE 4. $\Delta u = 2(x + y)$, $(x, y) \in S$ [Figure 1(b)]:

$$u(x, y) = xy(x + y), \qquad (x, y) \in \partial S.$$

EXAMPLE 5. $\Delta u = 0$, $(x, y) \in D$ (Figure 2):

$$u(x, y) = x^2 - y^2, \qquad (x, y) \in T \cup S.$$

EXAMPLE 6. $\Delta u = 2(x + y)$, $(x, y) \in D$ (Figure 2):

$$u(x, y) = xy(x + y), \qquad (x, y) \in T \cup S.$$

EXAMPLE 7. $\Delta u = e^u$, $(x, y) \in D$ (D is the unit square):

$$u(x, y) = 0, \qquad (x, y) \in \partial D.$$

EXAMPLE 8. $\Delta u = u^2$, $(x, y) \in D$ (unit square):

$$u(x, y) = 1, \qquad (x, y) \in \partial D.$$

The numerical results of these examples are presented in the Tables 1–8. We use the following abbreviations for convenience:

a.e.	Average error
m.e.	Maximum error
t	Run time (in seconds)
(a)	SMT
(b)	SMT solution refined to finer grids
(c)	Roache method

5. DISCUSSION AND CONCLUSION

The computational results in Tables 1–4 for Examples 1–4 using method I show that the SMT works on grids with mesh size $\frac{1}{32}$, whereas Roache method works on grids with mesh size $\frac{1}{16}$ and fails to converge on finer grids. Further, the solutions computed by SMT are of greater accuracy than with the Roache method. However, the computational time required in SMT is greater than in the Roache method.

The solutions on finer grids with $h = \frac{1}{32}$ and $\frac{1}{64}$ have been computed by refining the SMT solution on a coarser grid by a mesh refinement technique [6, 18]. This technique employs the SMT solutions on a crude mesh (say $h = \frac{1}{16}$) to obtain solutions on finer grids (say $h = \frac{1}{32}$ and $\frac{1}{64}$, etc.) by using an appropriate interpolation scheme. As is apparent from the results, the refinement process does not increase the error beyond the SMT solution on coarser grid.

TABLE 1

NUMERICAL RESULTS FOR EXAMPLE 1

		$h=\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
(a)	a.e.	4.6E-18	7.2E-14	2.9E-03	
	m.e.	3.2E-17	1.2E-13	6.0E-02	
	t	0.076	0.445	3.671	
(b)	a.e.			8.6E-14	2.1E-14
	m.e.			1.4E-13	1.4E-13
	t			0.546	0.825
(c)	a.e.	1.1E-17	4.9E-12		
	m.e.	3.1E-16	1.7E-11	Does not converge	
	t	0.025	0.149		

TABLE 2

NUMERICAL RESULTS FOR EXAMPLE 2

		$h=\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
(a)	a.e.	1.3E-18	3.8E-14	1.6E-03	
	m.e.	5.9E-17	5.6E-13	4.7E-02	
	t	0.085	0.454	3.685	
(b)	a.e.			7.4E-14	5.2E-14
	m.e.			5.8E-13	5.8E-13
	t			0.558	0.882
(c)	a.e.	6.4E-17	1.6E-12		
	m.e.	1.0E-16	6.6E-11	Does not converge	
	t	0.027	0.152		

TABLE 3

NUMERICAL RESULTS FOR EXAMPLE 3

		$h=\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
(a)	a.e.	2.3E-18	1.2E-15	7.9E-08	
	m.e.	1.7E-17	1.9E-14	1.1E-06	
	t	0.089	0.390	2.522	
(b)	a.e.			4.6E-15	4.8E-15
	m.e.			2.3E-14	2.3E-14
	t			0.501	0.782
(c)	a.e.	1.9E-16	2.2E-10		
	m.e.	5.6E-16	5.7E-10	Does not converge	
	t	0.030	0.151		

TABLE 4

NUMERICAL RESULTS FOR EXAMPLE 4

		$h=\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
(a)	a.e.	1.9E-18	1.4E-15	6.5E-08	$\frac{1}{64}$
	m.e.	1.7E-16	1.5E-14	9.1E-07	
	t	0.088	0.376	2.531	
(b)	a.e.			2.8E-15	3.2E-15
	m.e.			1.9E-14	1.9E-14
	t			0.512	0.796
(c)	a.e.	2.2E-17	1.4E-11		
	m.e.	2.4E-16	3.6E-11	Does not converge	
	t	0.033	0.153		

TABLE 5

NUMERICAL RESULTS FOR EXAMPLES 5 AND 6[a]

		SMT	SMT with grid refinement	
Example		$h=\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
5	a.e.	6.5E-15	8.2E-15	8.7E-15
	m.e.	3.2E-13	6.3E-13	6.3E-13
	t	3.428	3.564	3.835
6	a.e.	5.3E-10	8.9E-10	9.3E-10
	m.e.	2.6E-08	7.2E-08	7.2E-08
	t[b]	1.015	1.145	1.264

[a] Dimension of the capacitance matrix B_{22} is 24×24.

[b] The capacitance matrix for Example 6 is identical to that of Example 5, and the time given is the time for solving the additional problem.

TABLE 6

NUMERICAL RESULTS FOR EXAMPLE 7 USING SMT[a]

(x, y)	9×9 $t=0.277$	17×17 1.497	33×33 6.831
(0.25,0.25)	−0.042678	−0.0431431	−0.0432147
(0.25,0.50)	−0.0539161	−0.0544334	−0.0544946
(0.25,0.75)	−0.0426783	−0.0431431	−0.0432147
(0.50,0.25)	−0.0539161	−0.0544334	−0.0544946
(0.50,0.50)	−0.0690317	−0.0696881	−0.0697525
(0.50,0.75)	−0.0539161	−0.0544334	−0.0544946
(0.75,0.25)	−0.0426003	−0.0431431	−0.0532147
(0.75,0.75)	−0.0426783	−0.0431431	−0.0432147

[a] $\varepsilon = 1.0$E-09, $n=2$.

TABLE 7

NUMERICAL RESULTS FOR EXAMPLE 8[a]

(x, y)	9×9 $t = 0.241$	17×17 1.335	33×33 6.429
(0.25,0.25)	0.959132	0.958668	0.958633
(0.25,0.50)	0.948564	0.948050	0.948044
(0.25,0.75)	0.959132	0.958668	0.958633
(0.50,0.25)	0.948564	0.948050	0.948043
(0.50,0.50)	0.934377	0.933729	0.933747
(0.50,0.75)	0.948564	0.948050	0.948043
(0.75,0.25)	0.959132	0.958668	0.959633
(0.75,0.50)	0.948564	0.948050	0.948044
(0.75,0.75)	0.959132	0.958668	0.958633

[a] $\varepsilon = 1.0\text{E-}09$, $n = 2$.

TABLE 8

NUMERICAL RESULTS FOR EXAMPLES 7 AND 8[a]

	Example 7		Example 8	
(x, y)	SMT alone (33×33)[b] $t = 6.831$	Refining 17×17 SMT solution[c] 2.379	SMT alone (33×33)[b] 6.429	Refining 17×17 SMT solution[c] 2.203
(0.125,0.125)	− 0.0175591	− 0.0174726	0.983029	0.983093
(0.125,0.25)	− 0.0270531	− 0.0269682	0.973999	0.974041
(0.125,0.375)	− 0.0318986	− 0.0318194	0.969445	0.969466
(0.25,0.125)	− 0.0270531	− 0.0269682	0.973999	0.974041
(0.25,0.25)	− 0.0432626	− 0.0431431	0.958633	0.958668
(0.25,0.375)	− 0.0518685	− 0.0517392	0.950559	0.950574
(0.375,0.125)	− 0.0318986	− 0.0318194	0.963766	0.969466
(0.375,0.25)	− 0.0518685	− 0.0517392	0.950559	0.950574
(0.375,0.375)	− 0.0627321	− 0.0625793	0.940383	0.940381

[a] $\varepsilon = 1.0\text{E-}09$.
[b] $n = 2$.
[c] $n = 3$.

From the results given in Table 5 for Examples 5 and 6 using method II, it is evident that the method works well on grids with mesh size $h = \frac{1}{16}$, and the refinement technique to obtain solutions on finer grids ($h = \frac{1}{32}, \frac{1}{64}$) does not increase the error beyond the SMT solution on the coarser grid. The computational time required for the construction and factorization of the capacitance matrix was 2.464 seconds. Since the capacitance matrix depends only upon the geometry of the region and not on the boundary data, these matrices for Examples 5 and 6 turn out to be identical. Thus the time given for obtaining

the solution for Example 6 is the time required for solving an additional problem, excluding the time required for the construction and the LU decomposition of the capacitance matrix.

The numerical results for Examples 7 and 8 are given in Tables 6 and 7 respectively. These results show that the SMT works well on a 33×33 grid. The effectiveness of the mesh refinement technique has been demonstrated again through the numerical results presented in Table 8. The numerical experiments show that the computational time required in solving the problem by SMT alone is considerably greater than the time taken in first computing the solution on a coarser grid (17×17) and then refining it to a 33×33 grid. Moreover, the accuracy of the solutions by SMT alone and that of SMT and mesh refinement is comparable.

In conclusion, the adaptability of the SMT to irregular regions has been successfully shown through methods I and II. Method I has been shown to cover all irregular geometries, with the exception of a region with a hole. Method II, based on a capacitance matrix, has been shown to cover an arbitrary irregular bounded region. The dependence of the capacitance matrix only on the geometry of the region and not on the boundary data makes method II especially attractive and efficient when several problems with different boundary data in the same geometry are to be solved. The SMT in conjunction with quasilinearization has also been shown to work well for mildly nonlinear elliptic equations. The economical use of the SMT would be to compute the solution on coarser grid by SMT and obtain the solution on finer grids by mesh refinement.

REFERENCES

1 E. Angel, Discrete invariant imbedding and elliptic boundary value problems over irregular regions, *J. Math. Anal. Appl.* 23:471–484 (1968).

2 _____, Dynamic programming and linear partial differential equations, *J. Math. Anal. Appl.* 23:628–638 (1968).

3 _____, A building block technique for elliptic boundary value problems over irregular regions, *J. Math. Anal. Appl.* 26:75–81 (1969).

4 R. E. Bank and D. J. Rose, Marching algorithms for elliptic boundary value problems, I: The constant coefficient case, *SIAM J. Numer. Anal.* 14:792–829 (1977).

5 R. E. Bellman and R. E. Kalaba, *Quasilinearization and Nonlinear Boundary Value Problems*, American Elsevier, New York, 1965.

6 K. K. Bharadwaj, M. K. Kadalbajoo, and R. Sankar, Symmetric marching technique for the discretized Poisson equation, *Appl. Math. Comput.* 12:187–198 (1983).

7 O. Buneman, A compact non-iterative Poisson solver, Report 294, Institute for Plasma Research, Stanford Univ., 1969.

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

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

10 B. L. Buzbee, G. H. Golub, and C. W. Nielson, On direct methods for solving Poisson equation, *SIAM J. Numer. Anal.* 7:627–656 (1970).

11 F. W. Dorr, The direct solution of the discrete Poisson equation on a rectangle, *SIAM Rev.* 12:248–263 (1970).

12 D. Fischer, G. H. Golub, O. Mald, C. Leiver, and O. Widlund, On Fourier Toeplitz method for separable elliptic problems, *Math. Comp.* 28:349–368 (1974).

13 J. A. George, The use of direct methods for the solution of discrete Poisson equation on non-rectangular region, Report 159, Computer Science Dept., Stanford Univ., 1970.

14 D. Greenspan, *Lectures on the Numerical Solution of Linear, Singular, and Nonlinear Differential Equations*, Prentice Hall, Englewood Cliffs, N.J., 1968.

15 R. W. Hockney, The potential calculation and some applications, *Methods Comput. Phys.* 9:135–211 (1969).

16 _____, Formation and stability of virtual electrodes in a cylinder, *J. Appl. Phys.* 39:4166–4170 (1968).

17 _____, POT4—A fast direct Poisson solver for the rectangle allowing some mixed boundary conditions and internal electrodes, IBM Research, R.C. 2870, 1970.

18 J. M. Hyman, Mesh refinement and local inversion of elliptic partial differential equations, *J. Comput. Phys.* 23:124–134 (1977).

19 W. Proskurowski and O. Widlund, On the numerical solution of Helmholtz's equation by the capacitance matrix method, *Math. Comp.* 30:433–468 (1976).

20 P. J. Roache, *A New Direct Method for the Discretized Poisson Equation*, Lecture Notes in Physics, Vol. 8, Springer, 1971, pp. 48–53.

The Discrete W Transform*

Zhongde Wang† and B. R. Hunt

Department of Electrical Engineering
University of Arizona
Tucson, Arizona 85721

ABSTRACT

Four different versions of the discrete W transform (DWT) are introduced. The DWT may be decomposed into the discrete cosine transform (DCT) and the discrete sine transform (DST). Eight versions of both DCT and DST are introduced for the decomposition of the DWT. The relationship among different versions of DWT and their relation with the discrete Fourier transform (DFT) are given. Convolution theorems represented by different versions of the DWT are derived.

1. INTRODUCTION

The W transform is a real approach to harmonic analysis [1–3]. It is a conventional assumption in harmonic analysis that the harmonics should be the multiples of a fundamental frequency. However, this assumption is not a necessity. A fraction, or a fraction of a multiple, of the fundamental frequency may also be taken as a harmonic [2, 3]. Wang has shown that a data sequence may not only be composed of its integer harmonics, but also of its fractional harmonics [3]. However, integer harmonics, which are multiplies of the fundamental frequency, and half integer[1] harmonics, which are odd multiples of the fundamental frequency divided by 2, are preferred. In these two cases, the properties of symmetry and antisymmetry of a sequence may be used, and the W transform may always be decomposed into the discrete cosine transform (DCT) and the discrete sine transform (DST). Furthermore, fast al-

*This work was supported in part by the National Science Foundation, Grant No. ECS-8116192.
†Temporarily on leave; has now returned to Kunming Institute of Physics, China.

[1]A number is called a half integer if it is the sum of an integer and $\frac{1}{2}$.

52 Vanderbilt Ave., New York, NY 10017 0096-3003/85/$03.30

gorithms may be found more easily for these two types of harmonics than for any other type of harmonics. For these reasons, we shall be concerned only with integer harmonics and half-integer harmonics.

There are four versions of the DWT to be introduced in this paper. All of them are different approaches to harmonic analysis from the discrete Fourier transform (DFT); it will be shown that they relate closely to each other and to the DFT.

In the last decade, the DCT and DST have been of interest mainly in the area of image coding [11]. Since the orthogonal transform was introduced into image coding and bandwidth reduction by Andrews and Pratt [4–7], several versions of the DCT and DST have been introduced and their performance in bandwidth reduction has been examined [8–12]. However, no attempt has been made to use them as implementations of harmonic analysis. Therefore, another purpose of this paper is to give a common frame for all versions of the DCT and the DST. It will be shown that eight versions of both DCT and DST are naturally associated with the four versions of the DWT, and all versions of the DCT and DST, as well as the DWT, play important roles in harmonic analysis.

The version of the DWT depends on the symmetry type chosen for the sequences in the spatial (or temporal) domain and in the frequency domain. In Section 2, two symmetry types are described. Four versions of DWT are defined in Section 3. For convenience of discussion of the DWT, a special kind of function, the antiperiodic function, is defined and its properties are discussed in Section 4. Sections 5 and 6 contain some properties of the DWT. The relation among different versions of the DWT and between them and the DFT are discussed in Sections 7 and 8. In Section 9, eight versions of both DCT and DST are defined. The even-odd transform matrices introduced in Section 10 are just for the decomposition of the DWT matrices, which are given in Section 11. The convolution theorem represented by the DWT involves another type of convolution—antiperiodic convolution, which is defined in Section 12, and the convolution theorems represented by different versions of the DWT are given in Section 13.

2. SYMMETRY AND ANTISYMMETRY OF A SEQUENCE

A data sequence, whether it is finite or infinite, may be treated as samples taken from a continuous function. The symmetry or antisymmetry of a data sequence reflects the symmetry or antisymmetry of the function from which these samples are taken. However, the symmetry or antisymmetry of a data sequence depends on the choice of the symmetry center. If one of the sampling points is chosen to be the symmetry center, the symmetry or

antisymmetry of a data sequence will be referred to as odd symmetry or odd antisymmetry, and this symmetry type will be referred to as the odd symmetry type. For example, sequence $x(n)$ is said to be odd symmetric if

$$x(-n)=x(n), \tag{1}$$

and it is said to be odd antisymmetric if

$$x(-n)=-x(n). \tag{2}$$

In this symmetry type, all sampling points are integers. Discrete Fourier analysis is based on this symmetry type, and it has been investigated thoroughly. However, another symmetry type, which is called the even symmetry type, may sometimes be more convenient. If the midpoint between two adjacent sampling points is chosen to be the symmetry center, the symmetry or antisymmetry of a data sequence will be referred to as even symmetry or even antisymmetry, and this type of symmetry will be referred to as the even symmetry type. For example, $x(n)$ is said to be even symmetric if

$$x(-n-1)=x(n), \tag{3}$$

and to be even antisymmetric if

$$x(-n-1)=-x(n). \tag{4}$$

In this symmetry type, the sampling points are half integers.

There are two choices of symmetry type in both spatial (or temporal) and frequency domains. Therefore, there are four different combinations, which correspond to four different versions of the DWT.

3. FOUR VERSIONS OF THE DISCRETE *W* TRANSFORM

For a sequence of N data $x(n)$, $n=0,1,\ldots,N-1$, the following four sequences $X_j(m)$, $j=$ I, II, III, or IV, are called the discrete W transform

(DWT-j) of the $x(n)$;

$$X_{\mathrm{I}}(m)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}x(n)\sin\left[\frac{\pi}{4}+mn\cdot\frac{2\pi}{N}\right], \tag{5}$$

$$X_{\mathrm{II}}(m)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}x(n)\sin\left[\frac{\pi}{4}+m\left(n+\tfrac{1}{2}\right)\frac{2\pi}{N}\right], \tag{6}$$

$$X_{\mathrm{III}}(m)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}x(n)\sin\left[\frac{\pi}{4}+\left(m+\tfrac{1}{2}\right)n\cdot\frac{2\pi}{N}\right], \tag{7}$$

$$X_{\mathrm{IV}}(m)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}x(n)\sin\left[\frac{\pi}{4}+\left(m+\tfrac{1}{2}\right)\left(n+\tfrac{1}{2}\right)\frac{2\pi}{N}\right], \tag{8}$$

where $m=0,1,\ldots,N-1$.

Let $[W_N^j]$, $j=$ I, II, III or IV, be the DWT matrices, the elements of which at m row and n column are

$$\left[W_N^{\mathrm{I}}\right]_{m,n}=\sqrt{\frac{2}{N}}\sin\left[\frac{\pi}{4}+mn\cdot\frac{2\pi}{N}\right], \tag{9}$$

$$\left[W_N^{\mathrm{II}}\right]_{m,n}=\sqrt{\frac{2}{N}}\sin\left[\frac{\pi}{4}+m\left(n+\tfrac{1}{2}\right)\frac{2\pi}{N}\right], \tag{10}$$

$$\left[W_N^{\mathrm{III}}\right]_{m,n}=\sqrt{\frac{2}{N}}\sin\left[\frac{\pi}{4}+\left(m+\tfrac{1}{2}\right)n\cdot\frac{2\pi}{N}\right], \tag{11}$$

$$\left[W_N^{\mathrm{IV}}\right]_{m,n}=\sqrt{\frac{2}{N}}\sin\left[\frac{\pi}{4}+\left(m+\tfrac{1}{2}\right)\left(n+\tfrac{1}{2}\right)\frac{2\pi}{N}\right], \tag{12}$$

where $m,n=0,1,\ldots,N-1$. Roman numeral superscripts are used to denote the version number. The subscript N denotes the order of the matrix. Equations (5) through (8) then may be represented in terms of matrix multiplication:

$$\mathbf{X}_j=\left[W_N^j\right]\mathbf{x}, \qquad j=\text{I, II, III, or IV}, \tag{13}$$

where $\mathbf{x}$ and $\mathbf{X}_j$ are vector representations of sequences $x(n)$ and $X(m)$:

$$\mathbf{x}=\left[x(0)\quad x(1)\quad\cdots\quad x(N-1)\right]^T, \tag{14}$$

$$\mathbf{X}=\left[X(0)\quad X(1)\quad\cdots\quad X(N-1)\right]^T. \tag{15}$$

The original data vector x may be obtained by multiplying the inverse discrete W matrix $[W_N^j]^{-1}$ by the vector $\mathbf{X}_j$:

$$\mathbf{x} = \left[W_N^j\right]^{-1}\mathbf{X}_j, \qquad j = \text{I, II, III, or IV.} \tag{16}$$

It is not difficult to prove that inverse W matrices are related to W matrices by the following equations:

$$\left[W_N^{\text{I}}\right]^{-1} = \left[W_N^{\text{I}}\right], \tag{17}$$

$$\left[W_N^{\text{II}}\right]^{-1} = \left[W_N^{\text{III}}\right], \tag{18}$$

$$\left[W_N^{\text{III}}\right]^{-1} = \left[W_N^{\text{II}}\right], \tag{19}$$

$$\left[W_N^{\text{IV}}\right]^{-1} = \left[W_N^{\text{IV}}\right]. \tag{20}$$

From Equations (17) through (20), the original sequence may be represented by

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X_{\text{I}}(m) \sin\left[\frac{\pi}{4} + mn \cdot \frac{2\pi}{N}\right] \tag{21}$$

or

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X_{\text{II}}(m) \sin\left[\frac{\pi}{4} + m\left(n + \tfrac{1}{2}\right)\frac{2\pi}{N}\right] \tag{22}$$

or

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X_{\text{III}}(m) \sin\left[\frac{\pi}{4} + \left(m + \tfrac{1}{2}\right)n \cdot \frac{2\pi}{N}\right] \tag{23}$$

or

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X_{\text{IV}}(m) \sin\left[\frac{\pi}{4} + \left(m + \tfrac{1}{2}\right)\left(n + \tfrac{1}{2}\right)\frac{2\pi}{N}\right]. \tag{24}$$