Annales Univ. Sci. Budapest., Sect. Comp. 19 (2000) 3-16

DOUBLE BLOCK-PULSE SERIES SOLUTION OF SECOND ORDER PARTIAL DIFFERENTIAL EQUATIONS

Rafat Riad (Cairo, Egypt)

Abstract. This paper presents a technique for solving a second order partial differential equation via block-pulse functions. A double block-pulse series approximation of functions of two independent variables is first introduced. The procedure reduces the computational effort to a minimum while retaining accuracy.

1. Introduction

Approximating a function as a linear combination of a set of orthogonal basis functions is a standard tool in numerical analysis. Recently, Corrington [1] proposed a method of solving nonlinear differential and integral equations using a set of Walsh functions as the basis. His method is aimed at obtaining piecewise constant solutions of dynamic equations and requires previously prepared tables of coefficients for integrating Walsh functions. To alleviate the need for such tables Chen and Hsiao [2-4] introduced an operational matrix to perform integration of Walsh functions. This operational matrix approach has been applied to various problems such as time-domain analysis [2] and synthesis of linear systems [3], piecewise constant feedback gain determination for optimal control of linear systems [4] and for inverting irrational Laplace transform [5].

Block-pulse functions (BPF) and Walsh functions are closely related. As basis functions in an approximation the two sets of functions lead to the same results. Block-pulse functions have received less attention than Walsh functions in applications to control problems [6]-[9]. In [10] Shih and Han discuss the use of double Walsh series to find the solution to first order partial differential equations. One disadvantage of this approach is the method requires computation of inverses of very large matrices. Here we replace Walsh functions with block-pulse functions and solve second order partial differential equations. Our procedure results in lower triangular matrices whose inverses can be computed explicitly. This makes our approach very practical.

2. Block-pulse functions (BPF)

A set of BPF on a unit interval $[0,1)$ is defined as follows: for each integer i $(0 \leq i < m$ and $m \in \{1,2,\ldots\})$ the function φ_i is given by

$$\varphi_i(t) = \begin{cases} 1 & i/m \leq t < (i+1)/t, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

This set of functions can be concisely described by an m vector $\Phi_m(t)$

$$\Phi_m(t) = (\varphi_0(t), \varphi_1(t), \ldots, \varphi_{m-1}(t))^T,$$

where "T" means transpose.

A function $f(t)$, which is absolutely integrable in the interval $[0,1)$, can be approximately represented by block-pulse series

$$f(t) \cong \sum_{i=0}^{m-1} f_i \varphi_i(t) = f^T \Phi_m(t), \tag{2}$$

where f is the block-pulse coefficient vector $f = (f_0, f_1, \ldots, f_{m-1})$, f_i's are obtained by the minimization of the integral square error

$$\int_0^1 \left(f(t) - \sum_{i=0}^{m-1} f_i \varphi_i(t) \right)^2 dt,$$

f_i's are the mean values of $f(t)$ in the interval $[i/m, (i+1)/m)$

$$f_i = m \int_{i/m}^{(i+1)/m} f(t)dt, \qquad 0 \leq i < m. \tag{3}$$

Block-pulse functions have the following useful properties:

- disjoint property

$$\varphi_i(t)\varphi_j(t) = \begin{cases} \varphi_i(t) & \text{for } j = i, \\ 0 & \text{for all } j \neq i; \end{cases} \tag{4}$$

- orthogonality

$$\int_0^1 \varphi_i(t)\varphi_j(t)dt = \begin{cases} 1/m & \text{for } j = i, \\ 0 & \text{for all } j \neq i. \end{cases} \tag{5}$$

The function t^k, $t \in [0,1)$, $k \in \{0,1,\ldots\}$ can be approximated as a BPF series of size m. Indeed, using (2) and (3), we get

$$t^k \cong \frac{m}{k+1} \sum_{i=0}^{m-1} \left[\left(\frac{i+1}{m}\right)^{k+1} - \left(\frac{i}{m}\right)^{k+1} \right] \varphi_i(t). \tag{6}$$

The first integration of BPF can be expressed by BPF. Indeed, from (1) we have

$$\int_0^t \varphi_i(x)dx = \begin{cases} 0, & 0 \leq t < \frac{i}{m}, \\ t - \frac{i}{m} = (t - \frac{i}{m})\varphi_i(t) + \frac{1}{m} \sum\limits_{j=i+1}^{m-1} \varphi_j(t), & \frac{i}{m} \leq t < \frac{i+1}{m}, \\ \frac{1}{m}, & \frac{i+1}{m} \leq t < 1. \end{cases}$$

From (6) and using the disjoint property in (4) we obtain

$$\int_0^t \varphi_i(x)dx \cong \frac{1}{2m}\varphi_i(t) + \frac{1}{m} \sum_{j=i+1}^{m-1} \varphi_j(t), \qquad 0 \leq i < m.$$

Therefore we can write the relationship between BPF and their integrals in the following matrix form

$$\begin{bmatrix} \int_0^t \varphi_0(x)dx \\ \int_0^t \varphi_1(x)dx \\ \\ \int_0^t \varphi_{m-1}(x)dx \end{bmatrix} \cong \frac{1}{2m} \begin{bmatrix} 1 & 2 & 2 & & 2 \\ 0 & 1 & 2 & & 2 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & & 1 \end{bmatrix} \begin{bmatrix} \varphi_0(t) \\ \varphi_1(t) \\ \vdots \\ \varphi_{m-1}(t) \end{bmatrix},$$

or in compact form

$$\int_0^t \Phi_m(x)dx \cong B_m \Phi_m(t). \tag{7}$$

B_m is called the operational matrix of dimension m which relates BPF and their integrals. The operational matrix B_m is triangular and it has some properties which reduce the calculations in solving a second order partial differential equation. An elementary calculation shows that the powers B_m^k, $k \geq 1$ has the form

$$B_m^k = \frac{1}{(2m)^k} \begin{bmatrix} b_1^{(k)} & b_2^{(k)} & b_3^{(k)} & & b_m^{(k)} \\ 0 & b_1^{(k)} & b_2^{(k)} & & b_{m-1}^{(k)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & & b_1^{(k)} \end{bmatrix}, \tag{8}$$

where the elements $b_i^{(k)}$ are determined by the recursive formulae

$$\begin{aligned} &b_1^{(1)} = 1, \quad b_i^{(1)} = 2 && (i = 2,3,\ldots,m), \\ &b_i^{(j)} = \sum_{s=1}^{i} b_s^{(j-1)} b_{i-s+1}^{(1)} && (j = 2,3,\ldots,k;\ \ i = 1,2,\ldots,m). \end{aligned} \tag{9}$$

Moreover, for the inverse of B_m we get

$$B_m^{-1} = 2m \begin{bmatrix} p_1 & p_2 & p_3 & & p_m \\ 0 & p_1 & p_2 & & p_{m-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & & p_1 \end{bmatrix}, \tag{10}$$

where p_i $(i = 1,2,\ldots,m)$ are obtained by the recursive formulae

$$p_1 = 1, \qquad p_i = -2\sum_{j=1}^{i-1} p_j \qquad (i = 2,3,\ldots,m). \tag{11}$$

3. Double block-pulse series

A function of two independent variables $f(x, y)$, which is integrable on the unit square $0 \leq x, y < 1$, can be approximately represented by a BPF series of size n respect to y as follows

$$f(x, y) \cong \sum_{i=0}^{n-1} f_i(x)\varphi_i(y). \tag{12}$$

Using the orthogonal property of BPF in (5) the coefficient functions $f_i(x)$ become

$$f_i(x) = n \int_{i/n}^{(i+1)/n} f(x, y)dy \qquad (i = 0, 1, \dots, n-1).$$

Similarly, a BPF series approximation of $f_i(x)$ gives

$$f_i(x) \cong \sum_{j=0}^{m-1} \varphi_j(x)c_{ji}, \tag{13}$$

where c_{ji} are coefficients obtained by

$$c_{ji} = m \int_{j/m}^{(j+1)/m} f_i(x)dx = mn \int_{j/m}^{(j+1)/m} \int_{i/n}^{(i+1)/n} f(x, y)dydx. \tag{14}$$

The combination of (12) and (13) yields

$$f(x, y) \cong \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \varphi_j(x)c_{ji}\varphi_i(y). \tag{15}$$

Letting the BPF vectors

$$\Phi_n(y) = (\ \varphi_0(y), \varphi_1(y), \dots, \varphi_{n-1}(y)\ )^T, \tag{16}$$

$$\Phi_m(x) = (\ \varphi_0(x), \varphi_1(x), \dots, \varphi_{m-1}(x)\ )^T \tag{17}$$

and coefficient matrix C of $m \times n$ dimension $C = [c_{ji}]$, the equation (15) is written in matrix form as

$$f(x, y) \cong \Phi_m^T(x)\ C\ \Phi_n(y). \tag{18}$$

This is the double BPF series approximation of $f(x, y)$. For the convenience of computation we choose $m = 2^s$, $n = 2^w$, where s, w are positive integers. The integration of BPF vectors of (16) and (17) gives respectively

$$\int_0^y \Phi_n(t)dt \cong B_n \Phi_n(y), \tag{19}$$

$$\int_0^x \Phi_m(t)dt \cong B_m \Phi_m(x). \tag{20}$$

4. Solution of second order partial differential equations

Consider a general second order partial differential equation PDE with constant coefficients

$$\frac{\partial^2 f}{\partial x^2} + a_1 \frac{\partial^2 f}{\partial x \partial y} + a_2 \frac{\partial^2 f}{\partial y^2} + a_3 \frac{\partial f}{\partial x} + a_4 \frac{\partial f}{\partial y} + a_5 f = g. \tag{21}$$

The boundary conditions are

$$f(x, 0) = u_1(x), \qquad f_y(x, 0) = u_2(x), \tag{22}$$

$$f(0, y) = v_1(y), \qquad f_x(0, y) = v_2(y), \tag{23}$$

where a_i are constants and $g(x, y)$ is the input function. We try to obtain an approximate solution by BPF for $x \in [0, 1)$ and $y \in [0, 1)$. From (18), the double block-pulse series approximation of $f(x, y)$ is given by

$$f(x, y) \cong \Phi_m^T(x)\ C\ \Phi_n^T(y). \tag{24}$$

Similarly

$$g(x, y) \cong \Phi_m^T\ G\ \Phi_n^T(y), \tag{25}$$

where G is an $m \times n$ matrix with components

$$g_{ji} = mn \int_{j/m}^{(j+1)/m} \int_{i/n}^{(i+1)/n} g(x, y) dy dx$$

$$(j = 0, 1, \ldots, m-1; \quad i = 0, 1, \ldots, n-1).$$

A block-pulse series approximation of $u_i(x)$ $(i = 1, 2)$ of (22) is

$$u_i(x) = \Phi_m^T(x)\, U_i, \tag{26}$$

where

$$U_i = (u_{i0}, u_{i1}, \ldots, u_{i,m-1})^T \quad \text{and} \quad u_{ij} = m \int_{j/m}^{(j+1)/m} u_i(x) dx.$$

Since $\sum_{k=0}^{n-1} \varphi_k(y) = 1$ for $y \in [0, 1)$, equation (26) can be written as

$$u_i(x) = \Phi_m^T\, E^{(i)}\, \Phi_n(y) \qquad (i = 1, 2), \tag{27}$$

where $E^{(i)}$ is an $m \times n$ matrix each column of which is U_i. Similarly, a block-pulse series approximation of $v_i(y)$ of (23) can be obtained

$$v_i(y) = V_i^T\, \Phi_n(y) = \Phi_m^T(x)\, F^{(i)}\, \Phi_n(y) \qquad (i = 1, 2), \tag{28}$$

where

$$V_i^T = (v_{i0}, v_{i1}, \ldots, v_{i,n-1}), \qquad v_{ij} = n \int_{j/n}^{(j+1)/n} v_i(y) dy$$

and $F^{(i)}$ is an $m \times n$ matrix each row of which is V_i^T. The elements of $E^{(i)}$, $F^{(i)}$ $(i = 1, 2)$ and G are known. Now we try to obtain the elements of C.

Integration of equation (21) twice with respect to x and y gives

$$\int_0^y \int_0^x \int_0^{y'} \int_0^{x'} \{f_{\bar{x}\bar{x}} + a_1 f_{\bar{x}\bar{y}} + a_2 f_{\bar{y}\bar{y}} + a_3 f_{\bar{x}} + a_4 f_{\bar{y}} + a_5 f\}\, d\bar{x} d\bar{y} dx' dy' =$$

$$(29) \qquad = \int_0^y \int_0^x \int_0^{y'} \int_0^{x'} g(x,y) d\bar{x} d\bar{y} dx' dy'.$$

Using the block-pulse series approximation of (19), (20), (24), (25), (27) and (28), the seven terms of (29) can be evaluated, respectively, as follows

$$Term\ 1 = \int_0^y \int_0^x \int_0^{y'} \int_0^{x'} f_{\bar{x}\bar{x}}(\bar{x},\bar{y}) d\bar{x} d\bar{y} dx' dy' =$$

$$= \int_0^y \int_0^x \int_0^{y'} [f_{x'}(x',\bar{y}) - f_{x'}(0,\bar{y})] d\bar{y} dx' dy' =$$

$$= \int_0^y \int_0^{y'} \int_0^x [f_{x'}(x',\bar{y}) - \Phi_m^T(x') F^{(2)} \Phi_n(\bar{y})] dx' d\bar{y} dy' =$$

$$= \int_0^y \int_0^{y'} [f(x,\bar{y}) - f(0,\bar{y}) - \Phi_m^T(x) B_m^T F^{(2)} \Phi_n(\bar{y})] d\bar{y} dy' =$$

$$= \int_0^y \int_0^{y'} [\Phi_m^T(x) C \Phi_n(\bar{y}) - \Phi_m^T(x) F^{(1)} \Phi_n(\bar{y}) -$$

$$- \Phi_m^T(x)\ B_m^T\ F^{(2)}\ \Phi_n(\bar{y})] d\bar{y} dy' =$$

$$= \Phi_m^T(x)[C - F^{(1)} - B_m^T\ F^{(2)}]\ B_n^2\ \Phi_n(y).$$

$$Term\ 2 = a_1 \int_0^y \int_0^x \int_0^{x'} [f_{\bar{x}}(\bar{x},y') - f_{\bar{x}}(\bar{x},0)] dx' dy' =$$

$$= a_1 \int_0^y \int_0^x [f(x',y') - f(0,y') - f(x',0) + f(0,0)] dx' dy'.$$

$\sum\limits_{k=0}^{m-1} \varphi_k(x) = 1$ for $x \in [0,1)$ and $\sum\limits_{k=0}^{n-1} \varphi_k(y) = 1$ for $y \in [0,1)$. Then $f(0,0)$ can be written as $f(0,0) = \Phi_m^T\ F^{(0)}\ \Phi_n(y)$, where $F^{(0)}$ is an $m \times n$ matrix each element of which is $f(0,0)$. Hence

$$Term\ 2 = a_1 \Phi_m^T(x) B_m^T [C - F^{(1)} - E^{(1)} + F^{(0)}] \Phi_n(y).$$

Similarly,

$$\begin{aligned}
Term\ 3 &= a_2\Phi_m^T(x)(B_m^T)^2[C - E^{(1)} - E^{(2)}B_n]\Phi_n(y),\\
Term\ 4 &= a_3\Phi_m^T(x)B_m^T[C - F^{(1)}]B_n^2\Phi_n(y),\\
Term\ 5 &= a_4\Phi_m^T(x)(B_m^T)^2[C - E^{(1)}]B_n\Phi_n(y),\\
Term\ 6 &= a_5\Phi_m^T(x)(B_m^T)^2CB_n^2\Phi_n(y),\\
Term\ 7 &= \quad \Phi_m(x)(B_m^T)^2GB_n^2\Phi_n(y).
\end{aligned}$$

Substitution of these terms into (29) gives an equation of the following form

$$\Phi_m^T(x)[\ \ldots\ ]\Phi_n(y) = 0. \tag{30}$$

Since (30) is valid for any x and y in the domain of consideration, the quantity in the brackets should be equal to zero. That is

(31)

$$\begin{aligned}
&CB_n^2 + a_1B_m^TCB_n + a_2(B_m^T)^2C + a_3B_m^TCB_n^2 + a_4(B_m^T)^2CB_n +\\
&+ a_5(B_m^T)^2CB_n^2 = [F^{(1)} + B_m^TF^{(2)}]B_n^2 + a_1B_m^T[F^{(1)} + E^{(1)} - F^{(0)}]B_n +\\
&+ a_2(B_m^T)^2[E^{(1)} + E^{(2)}B_n] + a_3B_m^TF^{(1)}B_n^2 + a_4(B_m^T)^2E^{(1)}B_n + (B_m^T)^2GB_n^2.
\end{aligned}$$

The components of C are c_{ij}, $i = 0, 1, \ldots, m-1$ and $j = 0, 1, \ldots, n-1$, which can be obtained from the set of $m \times n$ linear algebraic equations (31) in matrix form. The explicit form of the solution for C is of practical importance for computer computation. Derivation of the explicit form of the solution for C is as follows. First, multiply (31) by B_n^{-1} from the right:

$$CB_n + a_1B_m^TC + a_2(B_m^T)^2CB_n^{-1} + a_3B_m^TCB_n + a_4(B_m^T)^2C + a_5(B_m^T)^2CB_n = \\ = Q,$$

where

$$\begin{aligned}
Q =&[F^{(1)} + B_m^TF^{(2)}]B_n + a_1B_m^T[F^{(1)} + E^{(1)} - F^{(0)}] +\\
&+ a_2(B_m^T)^2[E^{(1)}B_n^{-1} + E^{(2)}] + a_3B_m^TF^{(1)}B_n +\\
&+ a_4(B_m^T)^2E^{(1)} + (B_m^T)^2GB_n.
\end{aligned}$$

Q is an $m \times n$ known matrix. The first column of C is defined as c_0, the second column c_1, etc. Similarly, the first column of Q is defined as q_0, the second column q_1, etc. Using the Kronecker product technique introduced by Chen and Hsiao [2], [10], we rearrange C as a vector Z with mn components and rearrange Q as a vector W with mn components

$$Z = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}, \qquad W = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-1} \end{bmatrix}.$$

Finally, we have

$$\begin{aligned} &[I_m \otimes B_n^T]Z + a_1[B_m^T \otimes I_n]Z + a_2[(B_m^T)^2 \otimes (B_n^{-1})^T]Z + \\ &+ a_3[B_m^T \otimes B_n^T]Z + a_4[(B_m^T)^2 \otimes I_n]Z + a_5[(B_m^T)^2 \otimes B_n]Z = W, \end{aligned} \tag{32}$$

where $A \otimes D$ is the Kronecker product, i.e. each element of D is multiplied by the whole of A. Therefore, the solution for C comes directly from (32)

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = D^{-1} \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-1} \end{bmatrix}, \tag{33}$$

where

$$\begin{aligned} D = &I_m \otimes B_n^T + a_1 B_m^T \otimes I_n + a_2(B_m^2)^T \otimes (B_n^{-1})^T + \\ &+ a_3 B_m^T \otimes B_n^T + a_4(B_m^2)^T \otimes a_5(B_m^2)^T \otimes B_n^T \end{aligned} \tag{34}$$

is $mn \times mn$ matrix and I is the identity matrix. If we use (33) directly to determine the solution C, some difficulties might occur in obtaining the inverse of a square matrix $mn \times mn$, especially, if m and n are large values. Depending on the special properties of the operational matrix B, an algorithm is established to reduce the calculations.

Using the definition of Kronecker product of matrices, the matrix D in (34) can be written as

$$D = \begin{bmatrix} D_1 & O_m & O_m & O_m \\ D_2 & D_1 & O_m & O_m \\ D_3 & D_2 & D_1 & O_m \\ \vdots & \vdots & \vdots & \vdots \\ D_n & D_{n-1} & D_{n-2} & D_1 \end{bmatrix},$$

where

$$(35)\quad \begin{aligned} D_1 &= \frac{1}{2n} I_m + \left(a_1 + \frac{a_3}{2n}\right) B_m^T + \left(2na_2p_1 + a_4 + \frac{a_5}{2n}\right)(B_m^2)^T, \\ D_i &= \frac{1}{n} I_m + \frac{a_3}{n} B_m^T + \left(2na_2p_i + \frac{a_5}{n}\right)(B_m^2)^T \end{aligned}$$

are $m \times m$ matrices and O_m is $m \times m$ zero matrix. The matrix D has a special form and its inverse is easily obtained by

$$D^{-1} = \begin{bmatrix} R_1 & O_m & O_m & & O_m \\ R_2 & R_1 & O_m & & O_m \\ R_3 & R_2 & R_1 & & O_m \\ \vdots & & \vdots & & \vdots \\ R_n & R_{n-1} & R_{n-2} & & R_1 \end{bmatrix},$$

where R_i, $i = 1, 2, \ldots, n$, are $m \times m$ matrices determined by the following recursive formulae

$$R_1 = D_1^{-1}, \qquad R_i = -R_1 \left(\sum_{j=1}^{i-1} D_{i-j+1} R_j \right), \qquad i = 2, 3, \ldots, n.$$

From (35) the matrix D_1 can be written as

$$D_1 = \begin{bmatrix} d_1 & 0 & 0 & & 0 \\ d_2 & d_1 & 0 & & 0 \\ d_3 & d_2 & d_1 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ d_m & d_{m-1} & d_{m-2} & & d_1 \end{bmatrix},$$

where

$$d_1 = \frac{1}{2n} + \frac{\alpha}{2m} + \frac{\beta}{(2m)^2}, \qquad d_i = \frac{\alpha}{m} + \frac{\beta}{m^2}(i-1), \qquad i = 2, 3, \ldots, m$$

and

$$\alpha = a_1 + \frac{a_3}{2n}, \qquad \beta = 2na_2 + a_4 + \frac{a_5}{2n}.$$

Now, the inverse matrix D_1^{-1} is obtained by

$$D_1^{-1} = \begin{bmatrix} z_1 & 0 & 0 & & 0 \\ z_2 & z_1 & 0 & & 0 \\ z_3 & z_2 & z_1 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ z_m & z_{m-1} & z_{m-2} & & z_1 \end{bmatrix},$$

where

$$z_1 = \frac{1}{d_1}, \qquad z_i = -z_1 \left(\sum_{j=1}^{i-1} d_{i-j+1} z_j \right), \qquad i = 2, 3, \ldots, m.$$

This completes the derivation of the inverse of the matrix D. The solution C is easily found by substituting D^{-1} into (33), namely

$$c_j = \sum_{s=0}^{j} R_{j-s+1} q_s, \qquad j = 0, 1, \ldots, n-1.$$

Obviously, in this algorithm the calculation of matrix inverse of $mn \times mn$ is avoided. Therefore, we have saved computing time, storage and, consequently, have reduced the roundoff errors significantly. This algorithm is easy to implement on computer. A program has been written in BASIC language for solving a second order PDE. Computer output of an example is presented.

Example. Consider the second order PDE

$$\frac{\partial^2 f}{\partial x^2} - 5\frac{\partial f}{\partial x} + 6f = 12x.$$

The boundary conditions are

$$f(x,0) = 2x + \frac{5}{3}, \qquad f(0,y) = y + y^2 + \frac{5}{3},$$
$$f_y(x,0) = e^{2x}, \qquad f_x(0,y) = 2y + 3y^2 + 2.$$

For the purpose of comparison the exact solution is

$$f(x,y) = ye^{2x} + y^2 e^{3x} + 2x + \frac{5}{3}.$$

The following table (see on the next page) shows a comparison of the eight-term ($m = n = 8$) approximation and the true value at each midinterval. The accuracy is very good considering that the BPF series was truncated after the eighth term.

References

[1] **Corrington M.S.**, Solution of differential and integral equations with Walsh functions, *IEEE Trans. Circuit Theory,* **20** (5) (1973), 470-476.

(x,y)	BPF SOL.	EXACT	ERROR	BPF SOL.	EXACT	ERROR
At x= .0625				At x= .5625		
y=.0625	1.869506	1.8672	-2.305269E-03	3.016102	3.005297	-1.080489E-02
y=.1875	2.050824	2.046539	-4.285813E-03	3.581972	3.55926	-2.271152E-02
y=.3125	2.270604	2.263572	-7.032633E-03	4.32334	4.28216	-4.118014E-02
y=.4375	2.528847	2.518299	-1.054764E-02	5.240198	5.173994	-6.620407E-02
y=.5625	2.825548	2.810722	-1.482678E-02	6.332554	6.234765	-9.778929E-02
y=.6875	3.160715	3.140839	-1.987648E-02	7.600411	7.464472	-.1359391
y=.8125	3.534342	3.50865	-2.569151E-02	9.043758	8.863114	-.180644
y=.9375	3.946427	3.914157	-3.226996E-02	10.66263	10.43069	-.2319355
At x= .1875				At x= .6875		
y=.0625	2.142872	2.139459	-3.413201E-03	3.33537	3.319584	-1.578641E-02
y=.1875	2.382759	2.376179	-6.579876E-03	4.093773	4.059769	-3.400374E-02
y=.3125	2.678859	2.667744	-1.111531E-02	5.108671	5.045754	-6.291723E-02
y=.4375	3.03117	3.014154	-1.701617E-02	6.380066	6.27754	-.1025267
y=.5625	3.439699	3.41541	-2.428961E-02	7.907929	7.755125	-.1528034
y=.6875	3.904431	3.871511	-3.291988E-02	9.692344	9.478512	-.2138329
y=.8125	4.425392	4.382458	-.0429349	11.73319	11.4477	-.2854948
y=.9375	5.002548	4.94825	-5.429888E-02	14.03059	13.66269	-.3679018
At x= .3125				At x= .8125		
y=.0625	2.423436	2.418407	-5.028725E-03	3.676803	3.653772	-2.303052E-02
y=.1875	2.741745	2.731738	-1.000762E-02	4.696995	4.646212	-5.078221E-02
y=.3125	3.14221	3.124868	-1.734257E-02	6.092049	5.996289	-9.575987E-02
y=.4375	3.624838	3.597797	-2.704072E-02	7.861994	7.704004	-.1579905
y=.5625	4.189615	4.150527	-3.908873E-02	10.0068	9.769355	-.2374468
y=.6875	4.83656	4.783056	-5.350352E-02	12.52652	12.19235	-.3341751
y=.8125	5.565647	5.495384	-7.026291E-02	15.42104	14.97297	-.4480658
y=.9375	6.376911	6.287513	-8.939838E-02	18.69057	18.11124	-.5793324
At x= .4375				At x= .9375		
y=.0625	2.713491	2.70611	-7.381201E-03	4.047829	4.014263	-.033566
y=.1875	3.137193	3.122077	-1.511574E-02	5.425414	5.349725	-.0756898
y=.3125	3.680966	3.654152	-2.681327E-02	7.350874	7.205545	-.145329
y=.4375	4.344822	4.302335	-4.248715E-02	9.824272	9.581724	-.242548
y=.5625	5.128746	5.066626	-6.212044E-02	12.84554	12.47826	-.36728
y=.6875	6.032761	5.947025	-8.573628E-02	16.41465	15.89516	-.5194884
y=.8125	7.056824	6.943531	-.1132932	20.53182	19.83242	-.6994038
y=.9375	8.201004	8.056146	-.1448584	25.19667	24.29003	-.9066372

[2] **Chen C.F. and Hsiao C.H.**, A state-space approach to Walsh series solution of linear systems, *Int. J. Syst. Sci.*, **6** (9) (1975), 833-858.

[3] **Chen C.F. and Hsiao C.H.**, Time domain synthesis via Walsh functions, *Proc. IEE*, **122** (5) (1975), 565-570.

[4] **Chen C.F. and Hsiao C.H.**, Design of piecewise constant gains for optimal control via Walsh functions, *IEEE Trans. Automat. Contr.*, **20**, (5) (1975), 596-603.

[5] **Chen C.F., Tsay Y.T. and Wu T.T.**, Walsh operational matrices for fractional calculus and their application to distributed systems, *J. Franklin Inst.*, **303** (1977), 267-284.

[6] **Gopalasami N. and Deekshatulu B.L.**, Comments on design of piecewise constant gains for optimal control via Walsh functions, *IEEE Trans. Automat. Contr.*, **21** (1976), 634-635.

[7] **Prasada R.G. and Srinivasan T.**, Analysis and synthesis of dynamic systems containing time delays via block-pulse functions, *Proc. IEE,* **125** (9) (1978), 1064-1068.

[8] **Rafat Riad,** Block-pulse functions series solution of time-varying linear systems of higher order, *Annales Univ. Sci. Bud. Sect. Comp.*, **10** (1989), 167-187.

[9] **Sannuti P.**, Analysis and synthesis of dynamic systems via block-pulse functions, *Proc. IEE,* **124** (6) (1977), 569-571.

[10] **Yen-Ping Shih and Joung-Yi Han,** Double Walsh series solution of first-order partial differential equations, *Int. J. Syst. Sci.,* **9** (5) (1978), 569-578.

(Received January 22, 1992)

Rafat Riad
Mathematics Department
Faculty of Education
Ain Shams University
Heliopolis, Cairo
11341 Egypt