Australian Journal of Basic and Applied Sciences, 3(3): 1701-1715, 2009
ISSN 1991-8178

June 25, 2009Solution of a System of Two-dimensional Linear Fredholm Integral Equation of the Second Kind by Quadrature Methods

Rostam K. Saeed and Mehdi H. Mahmud

Salahaddin University/Erbil – College of Science – Department of Mathematics, Erbil, Iraq

Abstract: In this paper two-dimensional quadrature methods are applied to find the approximate solution for a system of two-dimensional linear Fredholm integral equation of the second kind (SL2DFIE). A reliable MATLAB program for solving SL2DFIE was established. Some illustrative examples and comparison tables depending on the least square error are presented, to elucidate the accuracy of these methods.

Key words: Two-dimensional Fredholm integral equations, Quadrature methods in two-dimension

INTRODUCTION

Consider the following system of linear two-dimensional Fredholm integral equations of the second kind (SL2DFIE):

$$U_i(x,t) = G_i(x,t) + \sum_{l=1}^{M} \iint_D K_{il}(x,t,y,s) U_i(y,s) ds dy \quad ;(x,t),(y,s) \in D \tag{1}$$

i=1, 2,…,M,

where $U_i(x,t)$ is an unknown functions to be determined and $G_i(x,t)$, $K_{il}(x,t,y,s)$ are given continuous functions defined, respectively, on $D=[a,b][c,d]$, while a, b, c and d are real constants, Vasile (2001). This equation has a unique solution in the space $L^2(D)$. The considered integral equations in space-time play a very important role in mechanics and technology, some initial-boundary problems for a number of differential partial equations in physics can be reduced to consider integral equation, Hacia (2002).

Numerical solutions for solving single two-dimensional Fredholm integral equation have been treated using different methods by many authors; see (Hanson and Kauthen (1978), Brunner and Kauthen (1989), Carutasu (2001), Han and Wang (2001), Hacia (2002), Ismail (2006) and Ahmad (2006)). In this paper, two quadrature methods for multiple integrals (composite Simpson method and composite trapezoid method) are presented and used to solve SL2DFIE.

Numerical Integrations:

The problem of numerical integration (quadrature methods) arise when the integration can be carried out exactly or when the function is known only at a finite number of data. Furthermore, numerical integration rules are primary tool used by engineers and scientists to obtain approximate answers for definite integrals that can not be solved analytically. For this reasons numerical integration is used for finding the multiple integral:

$$I= \iint_D f(x,y) dx dy \tag{2}$$

where the integrand $f(x,y)$ may be known function or a set of discrete data. Some known functions have an exact integral, in which case equation (2) can be evaluated exactly in closed form. Many known functions, however, do not have exact integral, or it is known only at a set of discrete points, in which cases an approximate numerical procedure is required to evaluate (2).

Corresponding Author: Rostam K. Saeed, Salahaddin University/Erbil – College of Science – Department of Mathematics, Erbil, Iraq
E-mail: rostamkarim64@uni-sci.org

An important part of the analysis of any numerical integration method is to study the behavior of the approximation error as a function of the number of integrand evaluations. A method which yields a small error for a small number of evaluations is usually considered superior. Reducing the number of evaluations of the integrand reduces the number of arithmetic operations involved, and therefore reduces the total round-off error. Also, each evaluation takes time, and the integrand may be arbitrarily complicated (Mathews and Fink (2004)).

A simple rule for approximate integration (or quadrature) has the following form:

$$\int_a^b f(x) = \sum_{i=0}^{n} w_i f(x_i) + E(f)$$

where $E(f)$ is the error function, the points x_i, $i = 0,1,000$, n are called quadrature nods and w_i, $i = 0,1,000$, n are the quadrature weights. This mean that the integral is represented by a weighted sum of values of the integrand at a finite number of points, $x_0, x_1, \ldots x_n$, (Phillips and Taylor (1996).

For multiple integral (1) the quadrature rule has the following form:

$$\int_a^b \int_c^d f(x,y)\,dy\,dx = \sum_{i=0}^{N} \sum_{j=0}^{M} w_i' w_j f(x_i, y_j) + E(f) \qquad (3)$$

where $x_0, z_1, \ldots x_n$, $y_0, y_1, \ldots y_m$, are quadrature nodes, w_i, $i = 0,1,000$, n and w_j, $j=0, 1, \ldots, M$ are the quadrature weights and $E(f)$ is the truncation error, (Vasile (2002)).

In this chapter, two quadrature rules for multiple integrals (composite Simpson method and composite trapezoid method) are presented and used to treat SL2DFIE.

Numerical Integrations for multiple integral:

In this section, the numerical methods for one-dimensional integral evaluation can be modified in a straightforward manner for use in the approximation of two-dimensional integrals.

Consider the double integral (2) where $D = \{(x,y) : a \le x \le b, c \le y \le d\}$, for some constants a, b, c and d is a rectangular region in the plain, see Jain, et. al (2004).

We will employ composite Simpson method and the composite trapezoid method to illustrate the approximation technique as follows:

Composite Simpson method for multiple integral:

To apply Composite Simpson's rules (Burden and Faires (2001)), we divide the region D by partitioning both $[a,b]$ and $[c,d]$ into an even number of subintervals. To simplify the notation, we choose even integers n an m with the evenly spaced mesh points $x_0, z_1, \ldots x_n$, *and* $y_0, y_1, \ldots y_m$, respectively. These subdivision determine step sizes $h = \frac{b-a}{n}$ and $k = \frac{d-c}{m}$. Writing the double integral as the iterated integral

$$\iint_D f(x,y)\,dx\,dy = \int_a^b \left(\int_c^d f(x,y)\,dy \right) dx \approx S_{2D}(f,h,k),$$

we first use the composite Simpson method to approximate

$$\int_c^d f(x,y)\,dy,$$

treating x as constant. Let $y_j = c + jk$, for each $j=0,1,\ldots,m$. Then

$$\int_c^d f(x,y)dy = \frac{k}{3}\left[f(x,y_0) + 2\sum_{j=1}^{(m/2)-1} f(x,y_{2j}) + 4\sum_{j=1}^{m/2} f(x,y_{2j-1}) + f(x,y_m)\right]$$

$$-\frac{(d-c)k^4}{180}\frac{\partial^4 f(x,\mu)}{\partial y^4},$$

for some μ in (c,d). Thus,

$$\iint_D f(x,y)dxdy =$$

$$\frac{k}{3}\left[\int_a^b f(x,y_0)dx + 2\sum_{j=1}^{(m/2)-1}\int_a^b f(x,y_{2j})dx + 4\sum_{j=1}^{m/2}\int_a^b f(x,y_{2j-1})dx + \int_a^b f(x,y_m)dx\right]$$

$$-\frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f(x,\mu)}{\partial y^4}dx \tag{4}$$

The composite Simpson method is now employed on the integrals in (4). Let $x_i = a + ih$, for each $i=0,1,\ldots,n$. Then for each $j=0,1,\ldots,m$, we have

$$\int_c^d f(x,y_j)dy = \frac{h}{3}\left[f(x_0,y_j) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_j) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_j) + f(x_n,y_j)\right]$$

$$-\frac{(b-a)h^4}{180}\frac{\partial^4 f(\xi_j,y_j)}{\partial y^4}$$

for some ξ_j in (a,b). The resulting approximation has the form

$$S_{2D}(f,h,k) = \frac{hk}{9}\Bigg\{f(x_0,y_0) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_0) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_0) + f(x_n,y_0)$$

$$+2\left(\sum_{j=1}^{(m/2)-1} f(x_0,y_{2j}) + 2\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j}) + 4\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j}) + \sum_{j=1}^{(m/2)-1} f(x_n,y_{2j})\right)$$

$$+4\left[\sum_{j=1}^{m/2} f(x_0,y_{2j-1}) + 2\sum_{j=1}^{m/2}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j-1}) + 4\sum_{j=1}^{m/2}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j-1}) + \sum_{j=1}^{m/2} f(x_n,y_{2j-1})\right]$$

$$+\left(f(x_0,y_m) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_m) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_m) + f(x_n,y_m)\right)\Bigg\} \tag{5}$$

The error term *E(f)* is given by

$$E(f) = \frac{-k(b-a)h^4}{540}\left[\frac{\partial^4 f(\xi_0, y_0)}{\partial x^4} + 2\sum_{j=1}^{(m/2)-1}\frac{\partial^4 f(\xi_{2j}, y_{2j})}{\partial x^4} + 4\sum_{j=1}^{m/2}\frac{\partial^4 f(\xi_{2j-1}, y_{2j-1})}{\partial x^4}\right.$$
$$\left. + \frac{\partial^4 f(\xi_m, y_m)}{\partial x^4}\right] - \frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f(x,\mu)}{\partial y^4}dx.$$

If $\partial^4 f / \partial x^4$ is continuous, the intermediate value theorem (Burden and Faires (2001)), can be repeatedly applied to show that the evaluation of the partial derivatives with respect to x can be replaced by a common value an that

$$E(f) = \frac{-k(b-a)h^4}{540}\left[3m\frac{\partial^4 f(\overline{\eta}, \overline{\mu})}{\partial x^4}\right] - \frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f(x,\mu)}{\partial y^4}dx,$$

for some $(\overline{\eta}, \overline{\mu})$ in *D*. If $\partial^4 f / \partial y^4$ is also continuous, the Weighted Mean Value Theorem for integrals (Burden and Faires (2001)), implies that

$$\int \frac{\partial^4 f(x,\mu)}{\partial y^4}dx = (b-a)\left[\frac{\partial^4 f(\hat{\eta}, \hat{\mu})}{\partial y^4}\right],$$

for some $(\hat{\eta}, \hat{\mu})$ in *D*. Since $(d-c)/k$ the error term has the form

$$E(f) = -\frac{(d-c)(b-a)}{180}\left[h^4\left(\frac{\partial^4 f(\overline{\eta}, \overline{\mu})}{\partial x^4}\right) + k^4\left(\frac{\partial^4 f(\hat{\eta}, \hat{\mu})}{\partial y^4}\right)\right] \quad (6)$$

for some $(\overline{\eta}, \overline{\mu})$ and $(\hat{\eta}, \hat{\mu})$ in *D*.

Composite trapezoid method for multiple integral:

To derive this method, the internal [*a,b*] is subdivided into *m* sub intervals $[x_{i-1}, x_i], i = 0,1,...,n$ of equal width $h = \frac{b-a}{n}$ by using the equally spaced sample points $x_i = a + ih$, *i*=0,1,…,*n*. Also assume that the interval [*c,d*] is subdivided into *m* subintervals $[y_{j-1}, x_j], j = 0,1,...,m$ of equal width $k = \frac{d-c}{m}$ by using the equally spaced sample points $x_j = c + jk$ *j*=0,1,…,*m*.

In a similar manner for deriving composite Simpson method for multiple integrals, we get

$$T_{2D}(f,h,k) = \frac{hk}{4}\left[f(x_0,y_0) + f(x_n,y_0) + f(x_0,y_m) + f(x_n,y_m)\right.$$
$$+2\sum_{i=1}^{n-1} f(x_i,y_0) + 2\sum_{i=1}^{n-1} f(x_i,y_m) + \sum_{j=1}^{m-1} f(x_0,y_j)$$

$$+2\sum_{j=1}^{m-1} f(x_n, y_j) + 4\sum_{i=1}^{n-1}\sum_{j=1}^{m-1} f(x_i, y_j)\Bigg] + E(f), \tag{7}$$

where

$$E(f) = -\frac{(d-c)(b-a)}{12}\left[h^2\left(\frac{\partial^2 f(\overline{\eta}, \overline{\mu})}{\partial x^2}\right) + k^2\left(\frac{\partial^2 f(\hat{\eta}, \hat{\mu})}{\partial y^2}\right)\right] \tag{8}$$

for some $(\overline{\eta}, \overline{\mu})$ and $(\hat{\eta}, \hat{\mu})$ n *D*.

Approximate solution of S2DLFIE Using Two-Dimensional Quadrature Methods:

In this section we use composite Simpson method and composite trapezoid method for multiple integrals to find an approximate solution of equation (1) as follows:

Using Composite Simpson method:

Applying equation (5) to evaluate the integrals in (1) yields:

$$U_i(x,t) = G_i(x,t) + \frac{hk}{9}\sum_{l=1}^{M}\Bigg[k_{il}(x,t,y_0,s_0)U_l(y_0,s_0) + 2\sum_{i=1}^{(n/2)-1} k_{il}(x,t,y_{2i},s_0)U_l(y_{2i},s_0)$$

$$+4\sum_{i=1}^{(n/2)} k_{il}(x,t,y_{2i-1},s_0)U_l(y_{2i-1},s_0) + K_{il}(x,t,y_n,S_0)U_l(y_n,S_0)$$

$$+2\Bigg(\sum_{j=1}^{(m/2)-1} k_{il}(x,t,y_0,s_{2j})U_l(y_0,s_{2j}) + 2\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{(n/2)-1} k_{il}(x,t,y_{2i},s_{2j})U_l(y_{2i},s_{2j})$$

$$+4\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{n/2} k_{il}(x,t,y_{2i-1},s_{2j})U_l(y_{2i-1},s_{2j}) + \sum_{j=1}^{(m/2)-1} k_{il}(x,t,y_n,s_{2j})U_l(y_n,s_{2j})\Bigg)$$

$$+4\Bigg(\sum_{j=1}^{m/2} k_{il}(x,t,y_0,s_{2j-1})U_l(y_0,s_{2j-1}) + 2\sum_{j=1}^{m/2}\sum_{i=1}^{(n/2)-1} k_{il}(x,t,y_{2i},s_{2j-1})U_l(y_{2i},s_{2j-1})$$

$$+4\sum_{j=1}^{m/2}\sum_{i=1}^{n/2} k_{il}(x,t,y_{2i-1},s_{2j-1})U_l(y_{2i-1},s_{2j-1}) + \sum_{j=1}^{m/2} k_{il}(x,t,y_n,s_{2j-1})U_l(y_n,s_{2j-1})\Bigg)$$

$$+k_{il}(x,t,y_0,s_m)U_l(y_0,s_m) + 2\sum_{i=1}^{(n/2)-1} k_{il}(x,t,y_{2i},s_m)U_l(y_{2i},s_m)$$

$$+4\sum_{i=1}^{n/2} k_{il}(x,t,y_{2i-1},s_m)U_l(y_{2i-1},s_m) + k_{il}(x,t,y_n,s_m)U_l(y_n,s_m)\Bigg] \tag{9}$$

$i=1, 2, \ldots, M\ ;\ n,m \geq 2$.

In the equation (9), replace x by x_r and t by t_q where $x_r = y_r = a + rh$ for $r=0, 1, \ldots, n$ and $t_q = s_q = c + qk$ or $q=0, 1, \ldots, m$; we get:

$$U_{irq} = G_i(x_r,t_q) + \frac{hk}{9}\sum_{l=1}^{M}\Bigg[k_{il}(x_r,t_q,y_0,s_0)U_{l00} + 2\sum_{i=1}^{(n/2)-1}k_{il}(x_r,t_q,y_{2i},s_0)U_{l(2i)0}$$

$$+4\sum_{i=1}^{(n/2)}k_{il}(x_r,t_q,y_{2i-1},s_0)U_{l(2i-1)0} + K_{il}(x_r,t_q,y_n,S_0)U_{lno}$$

$$+2\left(\sum_{j=1}^{(m/2)-1}k_{il}(x_r,t_q,y_0,s_{2j})U_{l0(2j)} + 2\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{(n/2)-1}k_{il}(x_r,t_q,y_{2i},s_{2j})U_{l(2i)(2j)}\right.$$

$$\left.+4\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{n/2}k_{il}(x_r,t_q,y_{2i-1},s_{2j})U_{l(2i-1)(2j)} + \sum_{j=1}^{(m/2)-1}k_{il}(x_r,t_q,y_n,s_{2j})U_{ln(2j)}\right)$$

$$+4\left(\sum_{j=1}^{m/2}k_{il}(x_r,t_q,y_0,s_{2j-1})U_{l0(2j-1)} + 2\sum_{j=1}^{m/2}\sum_{i=1}^{(n/2)-1}k_{il}(x_r,t_q,y_{2i},s_{2j-1})U_{l(2i)(2j-1)}\right.$$

$$\left.+4\sum_{j=1}^{m/2}\sum_{i=1}^{n/2}k_{il}(x_r,t_q,y_{2i-1},s_{2j-1})U_{l(2i-1)(2j-1)} + \sum_{j=1}^{m/2}k_{il}(x_r,t_q,y_n,s_{2j-1})U_{ln(2j-1)}\right)$$

$$+k_{il}(x_r,t_q,y_0,s_m)U_{l0m} + 2\sum_{i=1}^{(n/2)-1}k_{il}(x_r,t_q,y_{2i},s_m)U_{l(2i)m}$$

$$+4\sum_{i=1}^{n/2}k_{il}(x,t,y_{2i-1},s_m)U_{l(2i-1)m} + k_{il}(x_r,t_q,y_n,s_m)U_{lnm}\Bigg] \quad (10)$$

were $U_{irq} = U_i(y_r,s_q)$ for i=1, 2,…,M ; r=0,1,…,n and q=0, 1,…,m.

From equation (10) we get a linear system of L equations in L unknown U_{irq}, where $L=M(n+1)(m+1)$. Solve the resulting system by using **LU** factorization method (or Gauss elimination method) to find the unknowns U_{irq} for i=1, 2,…,M; r=0, 1,…,n and q=0,1,…,m. Substitute the values of U_{irq} for i=1, 2,…,M in the equation (9), to get the approximation solution of (1).

Step-by-step procedure for composite Simpson method:

Step (1): Set the values of M, n, m, $[a, b]$ and $[c, d]$.

Step (2): Let $y_j=a+jh$; $j=0,1,..,n$, $t_v=c+vk$; v=0, 1,...,m, $h=\frac{b-a}{n}$ and $k=\frac{d-c}{m}$

Step (3): In the equation (9), replace x by x_r and t by t_q where $x_r = y_r = a + rh$ for r=0, 1,…,n and $t_q = s_q = c + qk$ for q=0, 1,…,m, we get a linear system of L equations in L unknown U_{irq} for i=1, 2, …, M; r=0, 1,…,n and q=0, 1,…,m.

Step (4): Solve the resulting linear system in step (3) by **LU** factorization; to find the unknowns U_{irq} for i=1, 2, …, M; r=0, 1,…,n and q=0, 1,…,m..

Step (5): Substitute the values of U_{irq} in the equation (9), to get the approximation solution of (1) for i=1, 2, …, M.

Using composite trapezoid method:

Applying equation (7) to evaluate the integrals of equation (1) yields:

$$U_i(x,t) = G_i(x,t) + \frac{hk}{4}\sum_{l=1}^{M}\Big\{ K_{il}(x,t,y_0,s_0)U_l(y_0,s_0) + K_{il}(x,t,y_n,s_0)U_l(y_n,s_0)$$
$$+ K_{il}(x,t,y_0,s_m)U_l(y_0,s_m) + K_{il}(x,t,y_n,s_m)U_l(y_n,s_m)$$
$$+2\sum_{j=1}^{n-1}\Big(K_{il}(x,t,y_j,s_0)U_l(y_j,s_0) + K_{il}(x,t,y_j,s_m)U_l(y_j,s_m)\Big)$$
$$+2\sum_{j=1}^{m-1}\Big(K_{il}(x,t,y_0,s_j)U_l(y_0,s_j) + K_{il}(x,t,y_n,s_j)U_l(y_n,s_j)\Big)$$
$$+4\sum_{j=1}^{n-1}\sum_{v=1}^{m-1} K_{il}(x,t,y_j,s_v)U_l(y_j,s_v) \Big\} \quad i=1,2,\ldots,M;\ n,m \geq 1. \tag{11}$$

In the equation (11), replace x by x_r and t by t_q where $x_r = y_r = a + rh$ $r=0,1,\ldots,n$ and $t_q = s_q = c + qk$ $q=0,1,\ldots,m$, we get:

$$U_{irq} = G_i(x_r,t_q) + \frac{hk}{4}\sum_{l=1}^{M}\Big\{ K_{il}(x_r,t_q,y_0,s_0)U_{l00} + K_{il}(x_r,t_q,y_n,s_0)U_{ln0}$$
$$+ K_{il}(x_r,t_q,y_0,s_m)U_{l0m} + K_{il}(x_r,t_q,y_n,s_m)U_{lnm}$$
$$+2\sum_{j=1}^{n-1}\Big(K_{il}(x_r,t_q,y_j,s_0)U_{lj0} + K_{il}(x_r,t_q,y_j,s_m)U_{ljm}$$
$$+2\sum_{j=1}^{m-1}\Big(K_{il}(x_r,t_q,y_0,s_j)U_{l0j} + K_{il}(x_r,t_q,y_n,s_j)U_{lnj} + 4\sum_{j=1}^{n-1}\sum_{v=1}^{m-1} K_{il}(x_r,t_q,y_j,s_v)U_{ljv} \Big\} \tag{12}$$

were $U_{irq} = U_i(y_r,s_q)$ or $i=1, 2,\ldots,M$; $r=0,1,\ldots,n$ and $q=0, 1,\ldots,m$.

Equation (12) produce a linear system of L equations in L unknown U_{irq} for $i=1, 2,\ldots,M$; $r=0,1,\ldots,n$ and $q=0,1,\ldots,m$. Solve the resulting system by using **LU** factorization method (or Gauss elimination method) to find the unknowns U_{irq} for $i=1, 2,\ldots,M$; $r=0, 1,\ldots,n$ and $q=0,1,\ldots,m$. Substitute the values of U_{irq} for $i=1, 2,\ldots,M$ in the equation (11), to get the approximation solution of (1).

Step-by-step procedure for composite trapezoid method:

Step (1): Set the values of M, n, m, $[a, b]$ and $[c, d]$.

Step (2): Let $y_j=a+jh$; $j=0,1,\ldots,n$, $t_v=c+vk$; $v=0, 1,\ldots,m$, $h = \frac{b-a}{n}$ and $k = \frac{d-c}{m}$

Step (3): In the equation (11), replace x by x_r and t by t_q where $x_r = y_r = a + rh$ $r=0,1,\ldots,n$ and $t_q = s_q = c + qk$; $q=0,1,\ldots,m$, we get a linear system of $L = M(n+1)(m+1)$ equations in L unknown U_{irq} for $i=1, 2, \ldots, M$; $r=0,1,\ldots,n$ and $q=0, 1,\ldots,m$,

Step (4): Solve the resulting linear system in step (3) by **LU** factorization; to find the unknowns U_{irq} for i=1, 2, …, M; r=0, 1,…,n and q=0, 1,…,m..
Step (5): Substitute the values of U_{irq} in the equation (11), to get the approximation solution of (1) for i=1, 2, …, M.

Algorithms for solving S2DLFIE:

Easy to see that, the complexity of solving system (1), can be very high in practice. In this section, we will give the basic algorithm of mechanization for solving (1). Formulae (9)–(10) and (11)–(12) can be well adapted to calculate the ***S2DLFIE*** using two-dimensional quadrature methods and also by applying MATLAB7.4 program. Now if we want to solve ***S2DLFIE*** based on two-dimensional quadrature methods, everything we have to do is just to input information about the equations, then the program will give out the approximation solution of the problem. Based on this, we can get the numerical solution of the equation if we input the value of parameters {a, b, c, d, kernels and G}.

In the program, the parameters are set as follows:

a: lower limit of the first integrals,
b: upper limit of the first integrals,
c: lower limit of the second integrals,
d: upper limit of the second integrals,
M size of the system,
n: number of subintervals, we choose even positive number for composite Simpson method, and any positive number for trapezoid method.
k: represents the kernels of the system,
G: represents the given functions of the system.

Note: To input the values of k and G in the program, see the solution of the numerical examples 1-4.
Hereunder, the main programs for solving S2DLFIE are given:

MATLAB program for Composite Simpson's method for multiple integral:

```
function appu=CSRmxm(M,n,a,b,c,d,ker,G)
%This program is for finding an approximate solution for the MxM system of linear two-
%dimensional Fredholm integral equations of the second kind by using composite Simpson
%method for multiple integral
%The function S2D is used to calculate the integrals in the system using composite
%Simpson method for multiple integrals
tic
syms x y s t
format long e
hx=(b-a)/(2*n);hy=(d-c)/(2*n);
 for i=1:M
     clear eqq
     eqq=0;
     for l=1:M
            eqq=eqq+S2D(n,a,b,c,d,ker(i,l),l);
     end
     EN(i)=eqq;
 end
     count=1;
   for i=0:2*n
    xx=a+i*hx;
    for j=0:2*n
        tt=c+j*hy;
        for l=1:M
        r=(['u' int2str(l) int2str(i) int2str(j)]); w1=subs(EN(l),{x,t},{xx,tt});
        w2=subs(G(l),{x,t},{xx,tt});   ww(count)=r-w1-w2;count=count+1;
        end
    end
```

```
end
for pp=1:count-1
    ii=1;
 for i= 0:2*N
            for j =0:2*N
                for l=1:M
               coef(pp,ii)=diff(ww(pp),['u' int2str(l) int2str(i) int2str(j)]);ii=ii+1;
                end
            end
 end
end
coef=vpa(coef,100);
for l=1:M
   for i=0:2*n
      for j=0:2*n
        eval(['u' int2str(l) int2str(i) int2str(j),'=0' ';'] );
      end
   end
end
                bb=-1*eval(ww).'; [L,U]=lu(eval(coef));
umat=U\(L\bb);
m=1;
for i=0:2*n
      for j=0:2*n
          for l=1:M
        eval(['u'int2str(l)int2str(i)int2str(j),'=umat(m)'';']); m=m+1;
      end
end
end
for l=1:M
appu(l)=G(l)+eval(EN(l))
end
toc
function me=S2D(n,a,b,c,d,eq,l)
syms x y s t r
hy=(b-a)/(2*n);hs=(d-c)/(2*n); j1=0;j2=0;j3=0;
for i=0:2*n
    ss=c+i*hs;
    yy=a;
    k1=sym(subs(eq,{y,s},{yy,ss}))*['u' int2str(l) int2str(0) int2str(i)];
    yy=b; k1=k1+sym(subs(eq,{y,s},{yy,ss}))*['u' int2str(l) int2str(2*n) int2str(i)];
    k2=0; k3=0;
    for jj=1:2*n-1
        yy=a+jj*hy; z=sym(subs(eq,{y,s},{yy,ss}))*['u' int2str(l) int2str(jj) int2str(i)];
        if mod(jj,2)==0
            k2=k2+z;
           else
            k3=k3+z;
        end
        W=(k1+2*k2+4*k3)*hy/3;
    end
        if i==0 | i==2*n
            j1=j1+W;
       else if mod(i,2)==0
            j2=j2+W;
        else
```

```
            j3=j3+W;
               end
               end
    me=(j1+2*j2+4*j3)*hs/3;
    end
```

MATLAB program for Composite Trapezoid method for multiple integrals:

```
function u=CTRmxm(M,n,a,b,c,d,ker,G,exactu)
%This program is for finding an approximate solution for the MxM system of linear two-
%dimensional Fredholm integral equations of the second kind by using composite
%Trapezoid method for multiple integral
%The function T2D is used to calculate the integrals in the system using composite Trapezoid method for
multiple integrals
tic
syms x y s t
format long e
hx=(b-a)/n;hy=(d-c)/n;
 for i=1:M
     clear eqq
     eqq=0;
     for l=1:M
            eqq=eqq+T2D(n,a,b,c,d,ker(i,l),l);
     end
     EN(i)=eqq;
 end
     count=1;
   for i=0:n
    xx=a+i*hx;
    for j=0:n
        tt=c+j*hy;
        for l=1:M
        r=(['u' int2str(l) int2str(i) int2str(j)]); w1=subs(EN(l),{x,t},{xx,tt});
        w2=subs(G(l),{x,t},{xx,tt}); ww(count)=r-w1-w2;count=count+1;
        end
    end
end
for pp=1:count-1
    ii=1;
 for i= 0:n
            for j =0:n
                for l=1:M
               coef(pp,ii)=diff(ww(pp),['u' int2str(l) int2str(i) int2str(j)]);ii=ii+1;
                end
            end
 end
end
coef=vpa(coef,100);
for l=1:M
   for i=0:n
      for j=0:n
        eval(['u' int2str(l) int2str(i) int2str(j),'=0'] ';');
      end
   end
end
             bb=-1*eval(ww).';   [L,U]=lu(eval(coef));
umat=U\(L\bb);
```

```
m=1;
for i=0:n
       for j=0:n
            for l=1:M
         eval(['u' int2str(l) int2str(i) int2str(j),'=umat(m)' ';']); m=m+1;
       end
end
end
for l=1:M
u(l)=G(l)+eval(EN(l))
end
toc
function me=T2D(n,a,b,c,d,eq,l)
syms x y s t r
hy=(b-a)/n;hs=(d-c)/n; j1=0;j2=0;
for i=0:n
     ss=c+i*hs; yy=a;
     k1=sym(subs(eq,{y,s},{yy,ss}))*['u' int2str(l) int2str(0) int2str(i)];
     yy=b;  k1=k1+sym(subs(eq,{y,s},{yy,ss}))*['u' int2str(l) int2str(n) int2str(i)];
     k2=0;
    for jj=1:n-1
          yy=a+jj*hy;
        k2=k2+sym(subs(eq,{y,s},{yy,ss}))*['u' int2str(l) int2str(jj) int2str(i)];
          end
          T=(k1+2*k2)*hy/2;
          if i==0 |i==n
               j1=j1+T;
                         else
                    j2=j2+T;
                 end
           me=(j1+2*j2)*hs/2;
      end
```

Use the following function to calculate the least square errors:

```
function f=finderrors(exactu,u,M)
syms x t xx tt
format long e
for i=1:M
     err(i)=0;
end
x1=0.1;t1=0.1;
for i=0:10
     xx=i*x1;tt=i*t1;
     for l=1:M
     err(l)=err(l)+(subs(u(l),{x,t},{xx,tt})-subs(exactu(l),{x,t},{xx,tt}))^2;
     end
end
disp('errors are')
err
```

Numerical Examples:

In this section we solve numerically four examples, to show the validity of the prescribed methods by depending on the least square errors.

Note (1): To solve the examples by using a computer program which is given in previous section, save all programs as an M-file under the name **CSRmxm**, **S2D**, **CTRmxm**, **T2D** and **finderrors** respectively.

(2): In this paper, we choose $n=m=2$.

Example 1:

Consider the following ***S2DLFIE*** with the exact solution $u_1(x,t) = x^2 + t^2 + 9$ and $u_2(x,t) = x^2 + t - 6$

$$u_1(x,t) = G_1(x,t) + \int_0^1\int_0^1 (xt + y)\, u_2(y,s)\, ds dy$$

$$u_2(x,t) = G_2(x,t) + \int_0^1\int_0^1 (x^2 - t ys)\, u_1(y,s)\, ds dy, w$$

where

$$G_1(x,t) = x^2 + t^2 + \frac{23}{2} + \frac{31}{6}xt \quad \text{and} \quad G_2(x,t) = -\frac{26}{3}x^2 + \frac{7}{2}t - 6$$

Solution: In MATLAB7.4, write the following commands:

```
clc;clear all
syms x t y s ker G
M=2;n=2;a=0;b=1;c=0;d=1;
exactu(1)=x^2+t^2+9;exactu(2)=x^2+t-6;
G(1)=x^2+t^2+23/2+31/6*x*t ; G(2)=-26/3*x^2+7/2*t-6;
ker(1,1)=0 ; ker(1,2)=x*t+y ; ker(2,1)=x^2-t*y*s ; ker(2,2)=0;
%Using Composite Simpson method
[u]=CSRmxm(M,n/2,a,b,c,d,ker,G)
finderrors(exactu,u,M)
%Using Composite Trapezoid method
[u]=CTRmxm(M,n,a,b,c,d,ker,G)
finderrors(exactu,u,M)
```

After running the program, we get the numerical solution. The least square errors obtained for Example 1 are shown in Table 1.

Example 2:

Consider the following ***S2DLFIE*** with the exact solution $u_1(x,t) = \sin(x) + \mathrm{Cos}(t)$ and $u_2(x,t) = \sin(t) + \mathrm{Cos}(x)$

$$u_1(x,t) = G_1(x,t) + \int_0^1\int_0^1 (xt - y\,s)\, u_2(y,s)\, ds dy$$

$$u_2(x,t) = G_2(x,t) + \int_0^1\int_0^1 (x^2 + t^2)(u_1(y,s) + u_2(y,s))\, ds dy$$

Where

$$G_1(x,t) = \sin(x) + \cos(t) + 0.341 - 1.3xt \quad \text{and} \quad G_2(x,t) = \sin(x) + \cos(t) - 2.6x^2 - 2.6t^2$$

Solution: In MATLAB7.4, write the following commands:

```
clc;clear all
syms x t y s ker G
```

```
M=2;n=2;a=0;b=1;c=0;d=1;
exactu(1)=sin(x)+cos(t); exactu(2)=cos(x)+sin(t);
ker(1,1)=0;ker(1,2)=x*t-y*s ; ker(2,1)=x^2+t^2;ker(2,2)=x^2+t^2;
G(1)=sin(x)+cos(t)+.341-1.30*x*t ; G(2)=cos(x)+sin(t)-2.60*x^2-2.60*t^2;
%Using Composite Simpson method
[u]=CSRmxm(M,n/2,a,b,c,d,ker,G)
finderrors(exactu,u,M)
%Using Composite Trapezoid method
[u]=CTRmxm(M,n,a,b,c,d,ker,G)
finderrors(exactu,u,M)
```

After running the program, we get the numerical solution. The least square errors obtained for Example 2 are shown in Table 2.

Example 3:

Consider the following ***S2DLFIE*** with the exact solution $u_1(x,t) = x^2 + t^4$ nd $u_2(x,t) = xt^6 + t^3$

$$u_1(x,t) = G_1(x,t) + \int_0^1\int_0^1 (e^x t - y)\, u_2(y,s)\, ds\, dy$$

$$u_2(x,t) = G_2(x,t) + \int_0^1\int_0^1 (e^x t - y)\, u_1(y,s)\, ds\, dy$$

where

$$G_1(x,t) = x^2 + t^4 + \frac{29}{168} - \frac{9}{28}e^x t \quad \text{and} \quad G_2(x,t) = xt^6 + t^3 + \frac{7}{20} - \frac{8}{15}e^x t$$

Solution: In MATLAB7.4, write the following commands:

```
clc;clear all
syms x t y s ker G
M=2;n=2;a=0;b=1;c=0;d=1;
exactu(1)=exp(x+t) ; exactu(2)=sin(t)*cos(x);
ker(1,1)=0;ker(1,2)=x*t*s ; ker(2,1)=x*t-y^2;ker(2,2)=0;
G(1)=exp(x+t)-0.2534*x*t ; G(2)=sin(t)*cos(x)-2953/1000*x*t+247/200;
%Using Composite Simpson method
[u]=CSRmxm(M,n/2,a,b,c,d,ker,G)
finderrors(exactu,u,M)
%Using Composite Trapezoid method
[u]=CTRmxm(M,n,a,b,c,d,ker,G)
finderrors(exactu,u,M)
```

After running the program, we get the numerical solution. The least square errors obtained for Example 3 are shown in Table 3.

Example 4:

Consider the following ***S2DLFIE*** with the exact solution $u_1(x,t) = \sin(t)$, $u_2(x,t) = xt + t^3$ and $u_3(x,t) = t + \cos(x)$

$$u_1(x,t) = G_1(x,t) + \int_0^1\int_0^1 (xt + y)(u_1(y,s) + u_2(y,s) + u_3(y,s))\,ds\,dy$$

$$u_2(x,t) = G_2(x,t) + \int_0^1\int_0^1 (x^2 - ts)(u_2(y,s) + u_3(y,s))\,ds\,dy$$

$$u_3(x,t) = G_3(x,t) + \int_0^1\int_0^1 (x - ty)(u_1(y,s) + u_3(y,s))\,ds\,dy$$

where

$G1(x,t) = \sin(t) - 1.38xt - 0.712$, $G2(x,t) = xt + t^3 + 0.647t - 0.921x^2$ and

$G3(x,t) = t\cos(x) - 0.421x + 0.421t$

Solution: In MATLAB7.4, write the following commands:

```
clc;clear all
syms x t y s ker G
M=3;n=2;a=0;b=1;c=0;d=1;
exactu(1)=sin(t) ; exactu(2)=x*t+t^3 ; exactu(3)=t*cos(x);
ker(1,1)=x*t+y ; ker(1,2)=x*t+y ; ker(1,3)=x*t+y;
ker(2,1)=0 ; ker(2,2)=x^2-t*s ; ker(2,3)=x^2-t*s;
ker(3,1)=(x-t)*y ; ker(3,2)=0 ; ker(3,3)=(x-t)*y;
G(1)=sin(t)-1.38*x*t-.712 ;G(2)=x*t+t^3+.647*t-.921*x^2 ;G(3)=t*cos(x)-.421*x+.421*t
%Using Composite Simpson method
[u]=CSRmxm(M,n/2,a,b,c,d,ker,G)
finderrors(exactu,u,M)
%Using Composite Trapezoid method
[u]=CTRmxm(M,n,a,b,c,d,ker,G)
finderrors(exactu,u,M)
```

After running the program, we get the numerical solution. The least square errors obtained for Example 4 are shown in Table 4.

Table 1: Show a comparison between the least square errors for different value of *n* and *m* for Example 1

The methods	Approximate solution	Least square error if:		
		$n=m=2$	$n=m=4$	$n=m=6$
CSM	U_1	0	0	0
	U_2	0	0	0
CTM	U_1	2.91933885e-2	8.658823e-3	1.63325396e-3
	U_2	3.59594474e-3	1.088165e-3	2.08212620e-4

Table 2: shows a comparison between the least square errors for different value of *n* and *m* for Example 2

The methods	Approximate solution	Least square error if:		
		$n=m=2$	$n=m=4$	$n=m=6$
CSM	U_1	7.57604325e-6	2.811096067e-8	1.08657067e-9
	U_2	8.444163521e-5	3.136020828e-7	1.21235923e-8
CTM	U_1	2.72556804e-2	1.318714914e-3	2.493199629e-4
	U_2	2.781068438e-1	1.470576127e-2	2.820126052e-3

Table 3: shows a comparison between the least square errors for different value of *n* and *m* of test Example 3

The methods	Approximate solution	Least square error if:		
		$N=m=2$	$n=m=4$	$n=m=6$
CSM	U_1	3.75567224e-3	1.90952951481e-5	7.792993551e-7
	U_2	1.392397924e-3	5.87034355223e-6	2.321294214e-7
CTM	U_1	1.1675023969e-1	8.67635262410e-3	1.772045788e-3
	U_2	1.4434064760e-1	1.04216316773e-2	2.116631533e-3

Table 4: shows a comparison between the least square errors for different value of *n* and *m* for Example 4

The methods	Approximate solution	Least square error if:		
		$n=m=2$	$n=m=4$	$n=m=6$
CSM	U_1	8.0379727612e-3	3.177334445e-5	1.242388134e-6
	U_2	4.0772896037e-4	1.595670288e-6	6.228149731e-8
	U_3	0	0	0
CTM	U_1	2.202765595e-1	2.6670448844e-3	4.139873873e-4
	U_2	2.550993756e-2	2.1028149985e-3	4.288628588 e-4
	U_3	0	0	0

Conclusion:

The numerical results of the proposed methods in this paper for solving SL2DFIE indicate that the presented methods are appropriate for solving such problem. A comparison is made between these methods depending on least-square error, which is calculated from the approximate solution against the exact solution. We note that, the composite Simpson method for multiple integrals better than the composite trapezoid method, also we note that, when the partitioning number of the intervals [a,b] and [c,d] are increasing, this leads to decreasing the least-square error. For this reason to obtain a good approximation to the exact solution, chose partitioning number of the intervals [a,b] and [c,d] sufficiently large. Also, the results of the examples indicate that the given programs CSRmxm and CTRmxm of quadrature method for multiple integrals is simple and effective, and can provide accuracy approximate solution or exact solution of SLSDFIE.

REFERENCE

Ahmad, S.S., 2006. Expansion method for solving Fredholm Integral equations in space-time, Journal of Kirkuk University-Scientific Studies, 1(1): 15-25.

Brunner, H. and J.P. Kauthen, 1989. The Numerical Solution of Two-Dimensional Volterra Integral equation by collocation and iterated collocation, IMA Journal of Numerical Analysis, 9: 47-59.

Burden, R.L. and J.D. Faires, 2001. Numerical Analysis, seventh edition, Prindle Weber & Schmidt.

Carutasu, V., 2001. Numerical Solution of Two-Dimensional nonlinear Fredholm Integral equation of the second kind by Spline functions, General Mathematics, 9(1-2): 31-48.

Hacia, L., 2002. Computational Results for Integral equation in space time, Computational Methods in Science and Technology, 8(1): 7-15.

Han, G. and R. Wang, 2002. Richardson extrapolation of iterated discrete Galerkin solution for Two-Dimensional Fredholm Integral equations, Journal of Computational and Applied mathematics, 139: 49-63.

Hanson, R.L. and J.L. Philips, 1978. Numerical Solution of Two-Dimensional Integral Equation Using Linear elements, 15(1): 113-121.

Ismail, A.S., 2006. On the numerical solution of two dimensional singular integral equation, Journal of applied mathematics and computation, 173: 389-393.

Jain, M.K., S.R.K. Iyengar, and R.K. Jain, 2004. Numerical Methods, problems and solutions, Revised second edition, New Age International (P) Limited, Publisher.

Mathews, J.H. and K.D. Fink, 2004. Numerical methods Using matlab" Fourth edition; Pearson Education, Inc.

Phillips, G.M.M. and P.J. Taylor, 1996. Theory and applications of Numerical Analysis, Elsevier Science & Technology Books.