

A Computational Study with Finite Element Method and Finite Difference Method for 2D Elliptic Partial Differential Equations

George Papanikos, Maria Ch. Gousidou-Koutita

School of Mathematics, Aristotle University, Thessaloniki, Greece

Email: papaniksgeo@gmail.com, gousidou@math.auth.gr

Received 21 September 2015; accepted 27 November 2015; published 30 November 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we consider two methods, the Second order Central Difference Method (SCDM) and the Finite Element Method (FEM) with P_1 triangular elements, for solving two dimensional general linear Elliptic Partial Differential Equations (PDE) with mixed derivatives along with Dirichlet and Neumann boundary conditions. These two methods have almost the same accuracy from theoretical aspect with regular boundaries, but generally Finite Element Method produces better approximations when the boundaries are irregular. In order to investigate which method produces better results from numerical aspect, we apply these methods into specific examples with regular boundaries with constant step-size for both of them. The results which obtained confirm, in most of the cases, the theoretical results.

Keywords

Finite Element Method, Finite Difference Method, Gauss Numerical Quadrature, Dirichlet Boundary Conditions, Neumann Boundary Conditions

1. Introduction

Finite Difference schemes and Finite Element Methods are widely used for solving partial differential equations [1]. Finite Element Methods are time consuming compared to finite difference schemes and are used mostly in problems where the boundaries are irregular. In particular, it is difficult to approximate derivatives with finite difference methods when the boundaries are irregular. Moreover, Finite Element methods are more complicated than Finite Difference schemes because they use various numerical methods such as interpolation, numerical integration and numerical methods for solving large linear systems (see [2]-[6]). Also the mathematically deriva-

How to cite this paper: Papanikos, G. and Gousidou-Koutita, M.Ch. (2015) A Computational Study with Finite Element Method and Finite Difference Method for 2D Elliptic Partial Differential Equations. *Applied Mathematics*, 6, 2104-2124.

<http://dx.doi.org/10.4236/am.2015.612185>

tion of Finite Element Methods come from the theory of Hilbert Space, and Sobolev Spaces as well as from variational principles and the weighted residual method (see [5]-[12]).

In this paper, we will describe the Second order Central Difference Scheme and the Finite Element Method for solving general second order elliptic partial differential equations with regular boundary conditions on a rectangular domain. In addition, for both of these methods, we consider the Dirichlet and Neumann Boundary conditions, along the four sides of the rectangular area. Also, we make a brief error analysis for Finite element method. Moreover, for the finite element method, we site two other important numerical methods which are important in order that the algorithm can be performed.

These methods are the bilinear interpolation over a linear Lagrange element, Gauss quadrature and contour Gauss Quadrature on a triangular area. Furthermore, these two schemes lead to a linear system which we have to solve. For the purpose of this paper, we solve the outcome systems with Gauss-Seidel method which is briefly discussed. In the last section, we contacted a numerical study with Matlab R2015a. We apply these methods into specific elliptical problems, in order to test which of these methods produce better approximations when the Dirichlet and Neumann boundary conditions are imposed. Our results show us that the accuracy of these two methods depends on the kind of the elliptical problem and the type of boundary conditions. In Section 2, we study the Second order Central Difference Scheme. In Section 3, we give the Finite Element Method, bilinear interpolation in P_1 , Gauss Quadrature, Finite Element algorithm and error analysis. In Section 4, we give some numerical results, in Section 5, we give the conclusions and finally in Section 6 we give the relevant references.

2. Second Order Central Difference Scheme

The second order general linear elliptic PDE of two variables x and y given as follow:

$$p \frac{\partial^2 u}{\partial x^2} + s \frac{\partial^2 u}{\partial x \partial y} + q \frac{\partial^2 u}{\partial y^2} + b_1 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} + ru = f \quad (1)$$

with u defined on a rectangular domain $\Omega = [a, b] \times [c, d] \subset \mathbb{R}^2$, it holds $s^2 - 4pq < 0$. Also

$$p, q, s, b_1, b_2, r, f \in C^1(\Omega) \quad \text{and} \quad u \in C(\Omega) \cap C^2(\Omega)$$

Moreover in this paper two types of boundary conditions are considered:

$$u(x, y) = g(x, y) \text{ on } \Gamma_1 \quad (\text{Dirichlet Boundary Conditions}).$$

$$\frac{\partial u}{\partial n} = g_1(x) \text{ on } \Gamma_2 \quad (\text{Neumann Boundary Conditions}).$$

The boundary $\partial\Omega = \Gamma_1 \cup \Gamma_2$ and n is the normal vector along the boundaries.

We divide the rectangular domain Ω in a uniform Cartesian grid

$$(x_i, y_j) = ((i-1)h, (j-1)k) : i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M$$

where N, M are the numbers of grid points in x and y directions and

$$h = \frac{b}{N-1} \quad \text{and} \quad k = \frac{d}{M-1}$$

are the corresponding step sizes along the axes x and y . The discretize domain are shown in **Figure 1**.

Using now the central difference approximation we can approximate the partial derivatives of the relation (1) as follows:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2) \quad (2)$$

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i-1,j-1}}{4hk} + O(k^2 + h^2) \quad (3)$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} + O(k^2) \quad (4)$$

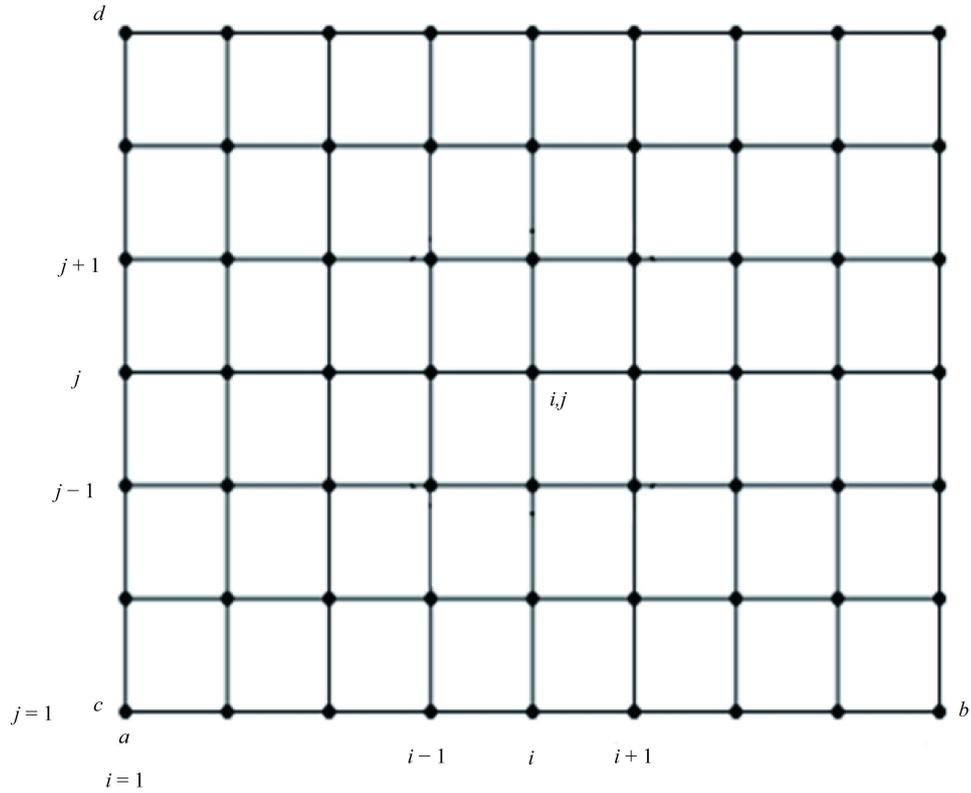


Figure 1. Discrete domain.

$$\frac{\partial u}{\partial x} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2) \tag{5}$$

$$\frac{\partial u}{\partial y} = \frac{u_{i,j+1} - u_{i,j-1}}{2k} + O(k^2) \tag{6}$$

where $O(h^2)$, $O(k^2)$ and $O(k^2 + h^2)$ are the truncation errors.

We now approximate the PDE (1) using the relations (2), (3), (4), (5), (6) and we obtain the second order central difference scheme:

$$4[h^2k^2r_{i,j} - 2k^2p_{i,j} - 2h^2q_{i,j}]u_{i,j} + 2k^2[2p_{i,j} + b_{1i,j}h]u_{i+1,j} + 2k^2[2p_{i,j} - b_{1i,j}h]u_{i-1,j} + 2h^2[2q_{i,j} + b_{2i,j}k]u_{i,j+1} + 2h^2[2q_{i,j} - b_{2i,j}k]u_{i,j-1} + s_{i,j}hk(u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i-1,j-1}) = 4h^2k^2f_{i,j} \tag{7}$$

With truncation error $O(k^2 + h^2)$.

The relation (7) can be written as a linear system:

$$Au = b \tag{8}$$

Dirichlet Boundary Conditions

The dimensions of the above linear system depends on the boundary conditions. More specific, if we have the Dirichlet Boundary Conditions:

$$u_{0,j} = g(a, y_j) \quad u_{N,j} = g(b, y_j) \quad \text{for each } j = 0, 1, \dots, M$$

$$u_{i,0} = g(x_i, c) \quad u_{i,M} = g(x_i, d) \quad \text{for each } i = 0, 1, \dots, N$$

then the dimensions of the matrix A , u and b are: $(N-1)(M-1) \times 1$ for the vectors u , b and

$$(N-1)(M-1) \times (N-1)(M-1)$$

for the matrix A . Moreover, the form of matrix A and the vector u are given by:

$$A = \begin{bmatrix} B_1 & D_1 & O & O & O & \cdots & O & O & O \\ G_2 & B_2 & D_2 & O & O & \cdots & O & O & O \\ O & G_3 & B_3 & D_3 & O & \cdots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ O & O & O & O & \cdots & C_{M-3} & B_{M-2} & D_{M-3} & O \\ O & O & O & O & \cdots & \cdots & C_{M-2} & B_{M-2} & D_{M-2} \\ O & O & O & O & \cdots & \cdots & O & G_{M-1} & B_{M-1} \end{bmatrix}$$

and

$$u = [u_{1,1}, u_{2,1}, u_{3,1}, \dots, u_{N-1,1}, u_{1,2}, u_{2,2}, u_{3,2}, \dots, u_{N-1,2}, \dots, u_{1,M-1}, \dots, u_{N-1,M-1}]$$

As we can see the matrix A is tri-diagonal block Matrix. These block matrices

$$B_k, k = 1, 2, \dots, M-1; G_l, l = 2, 3, \dots, M-1; D_m, m = 1, 2, \dots, M-2$$

are tri-diagonal as well of order $(N-1)(N-1)$

Neumann Boundary Conditions

$$\begin{aligned} \frac{\partial u}{\partial y}(x, d) &= g_1(x) & \frac{\partial u}{\partial y}(x, c) &= g_3(x) \\ \frac{\partial u}{\partial x}(b, y) &= g_2(y) & \frac{\partial u}{\partial x}(a, y) &= g_4(y) \end{aligned}$$

We approximate the Neumann boundary conditions in every side of the rectangular domain as follows

1st side (North side of the rectangular area)

$$\frac{\partial u}{\partial y}(x, d) = g_1(x) \Rightarrow \frac{u_{i,M+1} - u_{i,M-1}}{2k} = g_{1i} \Rightarrow u_{i,M+1} = u_{i,M-1} + 2kg_{1i} \quad \text{for } j = M, i = 1, 2, \dots, N-1 \quad (9)$$

2nd side (East side of the rectangular area)

$$\frac{\partial u}{\partial x}(b, y) = g_2(y) \Rightarrow \frac{u_{N+1,j} - u_{N-1,j}}{2h} = g_{2j} \Rightarrow u_{N+1,j} = u_{N-1,j} + 2hg_{2j} \quad \text{for } j = 1, 2, \dots, M-1, i = N \quad (10)$$

3rd side (South side of the rectangular area)

$$\frac{\partial u}{\partial y}(x, c) = g_3(x) \Rightarrow \frac{u_{i,1} - u_{i,-1}}{2k} = g_{3i} \Rightarrow u_{i,-1} = u_{i,1} - 2kg_{3i} \quad \text{for } j = 0, i = 1, 2, \dots, N-1 \quad (11)$$

4th side (West side of the rectangular area)

$$\frac{\partial u}{\partial x}(a, y) = g_4(y) \Rightarrow \frac{u_{1,j} - u_{-1,j}}{2h} = g_{4j} \Rightarrow u_{-1,j} = u_{1,j} - 2hg_{4j} \quad \text{for } j = 1, 2, \dots, M-1, i = 0 \quad (12)$$

Using the relations (9), (10), (11), (12) the values $u_{i,M+1}, u_{N+1,j}, u_{i,-1}$ and $u_{-1,j}$ which lies outside the rectangular domain can be eliminated when appeared in the linear system.

Thus the block tri-diagonal matrix A has dimensions $(N+1)(M+1) \times (N+1)(M+1)$ and the vectors u , b are of $(N+1)(M+1) \times 1$ order. The matrix A and the vector u are given below:

$$A = \begin{bmatrix} B_0 & L_0 & O & O & O & O & O & O & O \\ C_1 & K_1 & D_1 & O & O & O & O & O & O \\ O & C_2 & K_2 & D_2 & O & O & O & O & O \\ O & O & C_3 & K_3 & D_3 & O & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ O & O & O & O & O & C_{M-2} & K_{M-2} & D_{M-2} & O \\ O & O & O & O & O & O & C_{M-1} & K_{M-1} & D_{M-1} \\ O & O & O & O & O & O & O & L_M & B_M \end{bmatrix}$$

and

$$u = [u_{0,0}, u_{1,0}, u_{2,0}, \dots, u_{N,0}, u_{0,1}, u_{1,1}, u_{2,1}, \dots, u_{N,1}, \dots, u_{0,M}, \dots, u_{N-1,M}, u_{N,M}]$$

where $B_l, L_l, l = 0, M$ and $C_k, K_k, D_k, k = 1, 2, \dots, M - 1$ are tri-diagonal matrices with dimensions $(M + 1)^2$.

In order to solve the linear system (8), we use the Gauss-Seidel method (GSM) (see [2] [3]). An important property that the matrix A must have is to be strictly diagonally dominant in order the GCM to converge.

Theorem 1

If A is strictly diagonally dominant, then for any choice of $u^{(0)}$, Gauss-Seidel method give sequence $\{u^{(k)}\}_{k=0}^{\infty}$

that converge to the unique solution of $Au = b$.

The proof of theorem 1 can be found in [2] [3].

3. Finite Element Method

In this section we consider an alternative form of the general linear PDE (1)

$$\frac{\partial}{\partial x} \left(p \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial x} \left(\frac{s}{2} \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial y} \left(\frac{s}{2} \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(q \frac{\partial u}{\partial y} \right) + c \frac{\partial u}{\partial x} + d \frac{\partial u}{\partial y} + ru = f \tag{13}$$

where $p \in C^1(\bar{\Omega}), q \in C^1(\bar{\Omega}), s \in C^1(\bar{\Omega}), c \in C^1(\bar{\Omega}), d \in C^1(\bar{\Omega}), r \in C(\bar{\Omega}), f \in C(\bar{\Omega})$ and $u \in C^2(\Omega) \cap C(\bar{\Omega})$.

With boundary conditions

$u(x, y) = g(x, y)$ on Γ_1 (Dirichlet Boundary Conditions).

$\frac{\partial u}{\partial n} = g_1(x)$ on Γ_2 (Neumann Boundary Conditions).

And the boundary $\partial\Omega = \Gamma_1 \cup \Gamma_2$

In order to approximate the solution of (13) with FEM algorithm we must transform the PDE into its weak form and solve the following problem.

Find $u \in H_{\Gamma_1}^1(\Omega)$

$$a(u, v) = l(v) \quad \forall v \in H_{\Gamma_1}^1(\Omega) \tag{14}$$

where

$$a(u, v) = \iint_{\Omega} \left[p \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + q \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + \frac{s}{2} \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} + \frac{s}{2} \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} - c \frac{\partial u}{\partial x} v - d \frac{\partial u}{\partial y} v - ruv \right] dx dy$$

and

$$l(v) = -\iint_{\Omega} f v dx dy + \oint_{\Gamma_2} g_1 v ds$$

are bilinear and linear functionals as well.

It is sufficient now to consider that $u \in L_2(\Omega), \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \in L_2(\Omega)$. Also we assume that

$$p, q, s, c, d, r \in L_\infty(\Omega), f \in L_2(\Omega) \text{ and } g_1 \in L_2(\Gamma_2)$$

when the Neumann boundary Conditions are applied $\oint_{\Gamma_2} g_1 v ds \neq 0$, else if we have only Dirichlet Boundary

conditions then the line integral is equal to zero.

The finite element method approximates the solution of the partial differential Equation (13) by minimizing the functional:

$$J(v) = \iint_{\Omega} \left\{ \frac{1}{2} \left[p \left(\frac{\partial v}{\partial x} \right)^2 + q \left(\frac{\partial v}{\partial y} \right)^2 + s \frac{\partial v}{\partial y} \frac{\partial v}{\partial x} - c \frac{\partial v}{\partial x} v - d \frac{\partial v}{\partial y} v - r v^2 \right] + f v \right\} dx dy - \oint_{\Gamma_2} g_1 v ds \quad \forall v \in H_{\Gamma_1}^1(\Omega) \quad (15)$$

where $H_{\Gamma_1}^1(\Omega) = \{u \in H^1(\Omega) | u = g \text{ on } \Gamma_1\}$ and $H^1(\Omega) = \{u \in L_2(\Omega) : Du \in L_2(\Omega)\}$. Also with D we denote the weak derivatives of u . The spaces $H^1(\Omega)$, $H_{\Gamma_1}^1$ are Sobolev function spaces which also considered to be Hilbert spaces(see [7]-[9]).

The uniqueness of the solution of weak form (14) depends on **Lax-Milgram theorem** along with **trace theorem** (see [7]). In addition according to **Rayleigh-Ritz theorem** the solution of the problem (14) are reduced to minimization of the linear functional $J : H_{\Gamma_1}^1(\Omega) \rightarrow \mathbb{R}$, (see [7]).

The first step in order the FEM algorithm to be performed is the discretization of the rectangular domain $\Omega = [a, b] \times [c, d] \subset R^2$ by using Lagrange linear triangular elements.

We denote with P_k the set of all polynomials of degree $\leq k$ in two variables [7]. For $k = 1$ we have the linear Lagrange triangle and

$$\mathcal{P}_1 = \{\varphi \in C(\bar{\Omega}), \varphi(x, y) = a + bx + cy\}, \dim(\mathcal{P}_1) = 3$$

Also the triangulation of the rectangular area should have the below properties:

- We assume that the triangular elements $T_i, 1 \leq i \leq \kappa, \kappa = \kappa(h)$, are open and disjoint, where h is the maximum diameter of the triangle element.
- The vertices of the triangles all call nodes, we use the letter V for vertices and E for nodes.
- We also assume that there are no nodes in the interior sides of triangles.

3.1. Bilinear Interpolation in P_1

Let as consider now the triangulation of the rectangular domain $\Omega = [a, b] \times [c, d] \subset R^2$ as we describe to a previous section. In every triangle T_i of the domain we interpolate the function u with the below linear polynomial:

$$\varphi^{(i)}(x, y) = a + bx + cy$$

with interpolation conditions:

$$\varphi_j^{(i)}(x_j, y_j) = u(x_j, y_j), j = 1, 2, 3$$

in every vertex $V_j = (x_j^{(i)}, y_j^{(i)})$ of a triangular element.

Thus it creates the below linear system with unknown coefficients a, b, c .

$$\begin{bmatrix} \varphi_1^{(i)}(x_1, y_1) \\ \varphi_2^{(i)}(x_2, y_2) \\ \varphi_3^{(i)}(x_3, y_3) \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(i)} & y_1^{(i)} \\ 1 & x_2^{(i)} & y_2^{(i)} \\ 1 & x_3^{(i)} & y_3^{(i)} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Solving the system we find the approximate polynomial of u

$$\begin{aligned} \varphi^{(i)}(x, y) &= N_1^{(i)}(x, y)\varphi_1^{(i)}(x_1, y_1) + N_2^{(i)}(x, y)\varphi_2^{(i)}(x_2, y_2) + N_3^{(i)}(x, y)\varphi_3^{(i)}(x_3, y_3) \\ &= \sum_{j=1}^3 N_j^{(i)}(x, y)\varphi_j^{(i)}(x_j, y_j) \end{aligned}$$

where

$$\begin{cases} N_1^{(i)}(x, y) = a_1^{(i)} + b_1^{(i)}x + c_1^{(i)}y \\ N_2^{(i)}(x, y) = a_2^{(i)} + b_2^{(i)}x + c_2^{(i)}y \\ N_3^{(i)}(x, y) = a_3^{(i)} + b_3^{(i)}x + c_3^{(i)}y \end{cases}$$

and

$$\begin{aligned} a_1^{(i)} &= \frac{(x_2^{(i)}y_3^{(i)} - x_3^{(i)}y_2^{(i)})}{2A}, & b_1^{(i)} &= \frac{(y_2^{(i)} - y_3^{(i)})}{2A}, & c_1^{(i)} &= \frac{(x_3^{(i)} - x_2^{(i)})}{2A} \\ a_2^{(i)} &= \frac{(x_3^{(i)}y_1^{(i)} - x_1^{(i)}y_3^{(i)})}{2A}, & b_2^{(i)} &= \frac{(y_3^{(i)} - y_1^{(i)})}{2A}, & c_2^{(i)} &= \frac{(x_1^{(i)} - x_3^{(i)})}{2A} \\ a_3^{(i)} &= \frac{(x_1^{(i)}y_2^{(i)} - x_2^{(i)}y_1^{(i)})}{2A}, & b_3^{(i)} &= \frac{(y_1^{(i)} - y_2^{(i)})}{2A}, & c_3^{(i)} &= \frac{(x_2^{(i)} - x_1^{(i)})}{2A} \end{aligned}$$

The function $N_j^{(i)}(x, y) = a_j^{(i)} + b_j^{(i)}x + c_j^{(i)}y$ is the interpolation function or shape function and it has the following property:

$$N_j^{(i)}(x_k, y_k) = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}, \quad k = 1, 2, 3$$

3.2. Gauss Quadrature

An important step in order to implement the Finite Element algorithm is to compute numerically the double and line integrals which occurs in every triangular element (see [5] [12]).

Gauss Quadrature in Canonical Triangle

As a canonical triangle we consider the triangle with vertices (0, 0), (0, 1) and (1, 0) and we denote $T_\kappa = \{(x, y) : 0 \leq x, x + y \leq 1\}$. The approximation rule of the double integral in canonical triangle is given below:

$$\iint_{T_\kappa} f(x, y) dx dy \approx \frac{1}{2} \sum_{i=1}^{n_g} w_i f(x_i, y_i), \quad \forall f(x, y) \in P_\kappa \quad n_g(x_i, y_i)$$

Where n_g is the number of Gauss integration points, w_i are the weights and (x_i, y_i) are the Gauss integration points.

The linear space P_κ is the space of all linear polynomial of two variables of order k

The following **Table 1** gives the number of quadrature points for degrees 1 to 4 as given in Ref. [10]. It should be mentioned that for some N, the corresponding n_g is not necessarily unique. (see [10] and references therein).

Gauss quadrature in general triangular element

Initially we transform the general triangle T into a canonical triangle using the linear basis functions:

$$\begin{aligned} N_1(\xi, \eta) &= 1 - \xi - \eta \\ N_2(\xi, \eta) &= \xi \\ N_3(\xi, \eta) &= \eta \end{aligned}$$

Table 1. Quadrature points for degrees 1 to 4.

Quadrature points for degrees 1 to 4			
	N	$\dim(\mathcal{P}_N)$	n_g
	1	3	1
	2	6	3
	3	10	4
	4	15	5

The variables x, y for the random triangle can be written as affine map of basis functions:

$$x = r_1(\xi, \eta) = \sum_{i=1}^3 N_i(\xi, \eta) x_i = N_1(\xi, \eta) x_1 + N_2(\xi, \eta) x_2 + N_3(\xi, \eta) x_3$$

$$y = r_2(\xi, \eta) = \sum_{i=1}^3 N_i(\xi, \eta) y_i = N_1(\xi, \eta) y_1 + N_2(\xi, \eta) y_2 + N_3(\xi, \eta) y_3$$

Also we have the Jacobian determinant of the transformation

$$|J(\xi, \eta)| = \left| \frac{\partial(x, y)}{\partial(\xi, \eta)} \right| = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} = 2A_k$$

Using the above relations we obtain the Gauss quadrature rule for the general triangular element:

$$I = \iint_T F(x, y) dx dy \approx A_k \sum_{i=1}^{n_g} w_i F(r_1(\xi_i, \eta_i), r_2(\xi_i, \eta_i))$$

with

$$A_k = \frac{|x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|}{2}$$

is the area of the triangle.

Contour quadrature rule

In the Finite Element Method when the Neumann boundary conditions are imposed it is essential to compute numerically the below Contour integral in general triangular area.

$$I = \oint_{P_i P_j} g(x, y) ds = \int_{P_i}^{P_j} g(x, y) ds$$

The basic idea is to transform the straight contour $P_i P_j$ to an interval $l = [a, b]$, and then the Gaussian quadrature for single variable function.

Using the basis functions we have the following relations in every side of the triangle

Along side 1 ($P_1 P_2$):

$$\begin{aligned} \int_{P_1}^{P_2} g(x, y) ds &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \int_0^1 g\left(x_1 + (x_2 - x_1)\xi, y_1 + (y_2 - y_1)\xi\right) d\xi \\ &= \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{2} \int_{-1}^1 g\left(x_1 + \frac{(x_2 - x_1)(1 + \xi)}{2}, y_1 + \frac{(y_2 - y_1)(1 + \xi)}{2}\right) d\xi \\ &\approx \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{2} \sum_{i=1}^N c_i g\left(x_1 + \frac{(x_2 - x_1)(1 + \xi_i)}{2}, y_1 + \frac{(y_2 - y_1)(1 + \xi_i)}{2}\right) \end{aligned}$$

Along side 3: ($P_3 P_1$):

$$\begin{aligned} \int_{P_3}^{P_1} g(x, y) ds &= \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} \int_0^1 g\left(x_1 + (x_3 - x_1)\eta, y_1 + (y_3 - y_1)\eta\right) d\eta \\ &= \frac{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}}{2} \int_{-1}^1 g\left(x_1 + \frac{(x_3 - x_1)(1 + \eta)}{2}, y_1 + \frac{(y_3 - y_1)(1 + \eta)}{2}\right) d\eta \\ &\approx \frac{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}}{2} \sum_{i=1}^N c_i g\left(x_1 + \frac{(x_3 - x_1)(1 + \eta_i)}{2}, y_1 + \frac{(y_3 - y_1)(1 + \eta_i)}{2}\right) \end{aligned}$$

Along side 2: (P_2P_3) :

$$\begin{aligned} \int_{P_2}^{P_3} g(x, y) ds &= \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} \int_0^1 g\left(x_2 + (x_3 - x_2)\eta, y_2 + (y_3 - y_2)\eta\right) d\eta \\ &= \frac{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}{2} \int_{-1}^1 g\left(x_2 + \frac{(x_3 - x_2)(1+\eta)}{2}, y_2 + \frac{(y_3 - y_2)(1+\eta)}{2}\right) d\eta \\ &\approx \frac{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}}{2} \sum_{i=1}^N c_i g\left(x_2 + \frac{(x_3 - x_2)(1+\eta_i)}{2}, y_2 + \frac{(y_3 - y_2)(1+\eta_i)}{2}\right) \end{aligned}$$

The error of the bilinear interpolation Gauss quadrature depend on the dimension of the polynomial subspace (see [11]).

3.3. Finite Element Algorithm

The Finite element algorithm has the purpose to find the approximate solution of the problem (15) in a subspace of $H_{\Gamma_1}^1$. We consider as subspace the \mathcal{P}_1 of all piecewise linear polynomials with two variables polynomials of degree one. *i.e.*:

$$\varphi^{(i)}(x, y) = a + bx + cy$$

The index i represents the number of triangular elements which exist in the rectangular domain. The polynomials must be piecewise because the linear combination of them must form a continuous and integrable function with continuous first and second derivatives.

The existence and uniqueness of the approximate solution is ensured by the Lax-Milgram-Galerkin and Rayleigh-Ritz theorems (see [1] [7] [8]).

Firstly as we describe in a previous section, we have to triangulate the domain before the algorithm evaluated. After that the algorithm seeks approximation of the solution of the form:

$$u_h(x, y) = \sum_{i=1}^m \gamma_i \varphi_i(x, y)$$

where $\varphi_i, i = 1, 2, \dots, m$ is the linear combination of independent piecewise linear polynomials and $\gamma_i, i = 1, 2, \dots, m$ are constants with m is the number of nodes. Actually, the polynomials φ_i corresponds to shape functions $N_j, j = 1, 2, 3$ in every vertex of the triangles. Thus the approximate solution is the linear combination of all the independent interpolation functions multiplied with some constant γ_i . Some of these constants for example, $\gamma_{n+1}, \gamma_{n+2}, \dots, \gamma_m$ are used to ensure that the Dirichlet boundary conditions if there are exist

$$u_h(x, y) = g(x, y)$$

are satisfied on Γ_1 and the remaining constants $\gamma_1, \gamma_2, \dots, \gamma_n$ are used to minimize the functional $J(u_h)$.

Inserting the approximate solution $u_h(x, y) = \sum_{i=1}^m \gamma_i \varphi_i(x, y)$ for v into the functional $J(v)$ and we have:

$$\begin{aligned} J\left(\sum_{i=1}^m \gamma_i \varphi_i\right) &= \iint_{\Omega} \left\{ \frac{1}{2} \left[p \left(\sum_{i=1}^m \gamma_i \frac{\partial \varphi_i}{\partial x} \right)^2 + q \left(\sum_{i=1}^m \gamma_i \frac{\partial \varphi_i}{\partial y} \right)^2 + s \left(\sum_{i=1}^m \gamma_i \frac{\partial \varphi_i}{\partial y} \right) \left(\sum_{i=1}^m \gamma_i \frac{\partial \varphi_i}{\partial x} \right) - c \left(\sum_{i=1}^m \gamma_i \frac{\partial \varphi_i}{\partial x} \right) \left(\sum_{i=1}^m \gamma_i \varphi_i \right) \right. \right. \\ &\quad \left. \left. - d \left(\sum_{i=1}^m \gamma_i \frac{\partial \varphi_i}{\partial y} \right) \left(\sum_{i=1}^m \gamma_i \varphi_i \right) - r \left(\sum_{i=1}^m \gamma_i \varphi_i \right)^2 \right] + f \sum_{i=1}^m \gamma_i \varphi_i \right\} dx dy - \oint_{\Gamma_2} g_1 \sum_{i=1}^m \gamma_i \varphi_i ds \end{aligned} \tag{16}$$

Consider J as a function of $\gamma_1, \gamma_2, \dots, \gamma_n$. For minimum to occur we must have

$$\frac{\partial J}{\partial \gamma_j} = 0, \forall j = 1, 2, \dots, n$$

Differentiating (16) gives

$$\begin{aligned} & \sum_{i=1}^n \left[\iint_{\Omega} \left\{ p \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial y} + q \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial y} \right. \right. \\ & \left. \left. - \frac{c}{2} \left(\frac{\partial \varphi_i}{\partial x} \varphi_j + \frac{\partial \varphi_j}{\partial x} \varphi_i \right) - \frac{d}{2} \left(\frac{\partial \varphi_i}{\partial y} \varphi_j + \frac{\partial \varphi_j}{\partial y} \varphi_i \right) - r \varphi_i \varphi_j \right\} dx dy \right] \gamma_i \\ & = - \iint_{\Omega} f \varphi_j dx dy + \oint_{\Gamma_2} g_1 \varphi_j ds - \sum_{k=n+1}^m \left[\iint_{\Omega} \left\{ p \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_k}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_k}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_k}{\partial y} \right. \right. \\ & \left. \left. + q \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_k}{\partial y} - \frac{c}{2} \left(\frac{\partial \varphi_i}{\partial x} \varphi_k + \frac{\partial \varphi_k}{\partial x} \varphi_i \right) - \frac{d}{2} \left(\frac{\partial \varphi_i}{\partial y} \varphi_k + \frac{\partial \varphi_k}{\partial y} \varphi_i \right) - r \varphi_i \varphi_k \right\} dx dy \right] \gamma_k \end{aligned}$$

for each $j = 1, 2, \dots, n$. This set of equations can be written as a linear system:

$$A \mathbf{c} = \mathbf{b} \quad (17)$$

where $\mathbf{c} = (\gamma_1, \gamma_2, \dots, \gamma_n)^t$, $A = (a_{ij})$ and $\mathbf{b} = (\beta_1, \beta_2, \dots, \beta_n)^t$ are defined by

$$\begin{aligned} a_{ij} = \iint_{\Omega} \left\{ p \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial y} + q \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial y} \right. \\ \left. - \frac{c}{2} \left(\frac{\partial \varphi_i}{\partial x} \varphi_j + \frac{\partial \varphi_j}{\partial x} \varphi_i \right) - \frac{d}{2} \left(\frac{\partial \varphi_i}{\partial y} \varphi_j + \frac{\partial \varphi_j}{\partial y} \varphi_i \right) - r \varphi_i \varphi_j \right\} dx dy \end{aligned}$$

for each $i = 1, 2, \dots, n, j = 1, 2, \dots, m$.

$$\begin{aligned} \beta_i = - \iint_{\Omega} f \varphi_j dx dy + \oint_{\Gamma_2} g_1 \varphi_j ds - \sum_{k=n+1}^m \left[\iint_{\Omega} \left\{ p \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_k}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_k}{\partial x} + \frac{s}{2} \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_k}{\partial y} \right. \right. \\ \left. \left. + q \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_k}{\partial y} - \frac{c}{2} \left(\frac{\partial \varphi_i}{\partial x} \varphi_k + \frac{\partial \varphi_k}{\partial x} \varphi_i \right) - \frac{d}{2} \left(\frac{\partial \varphi_i}{\partial y} \varphi_k + \frac{\partial \varphi_k}{\partial y} \varphi_i \right) - r \varphi_i \varphi_k \right\} dx dy \right] \gamma_k \end{aligned}$$

The choice of subspace \mathcal{P}_1 for our approximation is important because can ensure us that the matrix A will be positive definite and band ([2]-[4]). According to the previous analysis leads again to a linear system, which can be solved as we described in a previous section with Gauss-Seidel Method (see [2]-[4]).

3.4. Error Analysis

Let us consider again the problem (14)

Find $u \in H_{\Gamma_1}^1(\Omega)$:

$$a(u, v) = l(v) \quad \forall v \in H_{\Gamma_1}^1(\Omega) \quad (18)$$

The approximation of finite element of the problem (18) is given below:

Find $u_h \in \mathcal{P}_1$:

$$u_h \in \mathcal{P}_1 \quad a(u_h, v_h) = l(v_h) \quad \forall v_h \in \mathcal{P}_1 \quad (19)$$

Cea's lemma

The finite element approximation $u_h \in \mathcal{P}_1$ of the weak solution $u \in H_{\Gamma_1}^1(\Omega)$ is the best fit to u in the norm $\|\bullet\|_{H_{\Gamma_1}^1(\Omega)}$ i.e:

$$\|u - u_h\|_{H_{\Gamma_1}^1(\Omega)} \leq \frac{c_1}{c_0} \min_{v_h \in \mathcal{P}_1} \|u - v_h\|_{H_{\Gamma_1}^1(\Omega)}$$

The error analysis of finite element method depend on the Cea's lemma for elliptic boundary value problems. The proof can be found in [1] [7].

Now we will present without proof the following statement [7].

$$\min_{v_h \in V_h} \|u - v_h\|_{H^1_1(\Omega)} \leq C(u)h^s \tag{20}$$

where $C(u)$ is positive constant dependent on the smoothness of the function u , h is the mesh size parameter and s is a positive real number, dependent on the smoothness of u and the degree of the piecewise polynomials comprising in \mathcal{P}_1 . In our case we have the Lagrange linear elements so the degree of the piecewise polynomials is one. Combining, relations Cea's lemma we shall be able to deduce that:

$$\|u - u_h\|_{H^1_1(\Omega)} \leq C(u) \frac{c_1}{c_0} h \tag{21}$$

The relation (21) gives a bound of the global error $e = u - u_h$ in terms of the size mesh parameter h . Such a bound on the global error is called **priori error bound**.

L₂-norm

For proving an error estimate in L_2 -norm the regularity of the solution of (13) plays an essential role. By the Aubin-Nitsche duality argument the error estimate in L_2 norm between u and its finite element approximation u_h is $O(h)$. However this bound can be improved to $O(h^2)$, (see [1] [7]).

4. Numerical Study

In this section we contact a numerical study using Matlab R2015a. For the purpose of this paper we cite representative examples of second order general elliptic partial differential equations in order to make comparisons between these two methods with various step-sizes and the mesh size parameters of finite element method. Thus in each example we present results for the absolute and relevant absolute errors in L_2 norm along with their graphs. Also we make graphical representations of the exact and approximate solution of the specific problem as well.

The problems of the examples can be found in [13] [14].

Example 1

Find the approximate solution of the partial differential equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad 0 \leq x \leq 4, \quad 0 \leq y \leq 4$$

with Dirichlet boundary conditions along the rectangular domain

$$u(x, y) = e^y \cos x - e^x \cos y, \quad (x, y) \in \partial\Omega$$

and exact solution

$$u(x, y) = e^x \cos x - e^y \cos y$$

Results (Table 2 and Table 3)

In **Figure 2** and **Figure 3** we have the graphs for SCDM and in **Figure 4** and **Figure 5** for FEM.

Example 2

Find the approximate solution of the partial differential equation

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + \frac{1}{10} \frac{\partial u}{\partial y} = f(x, y), \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1$$

with Dirichlet boundary conditions along the rectangular domain

$u = 0$ on three lower side of $\partial\Omega$ and Neumann boundary condition

$$\frac{\partial u}{\partial y}(x, 1) = 0$$

and exact solution

$$u(x, y) = \sin(\pi x) \sin\left(\frac{\pi y}{2}\right)$$

Table 2. Numerical results.

x	y	Second order central difference scheme with steps $h = 0.1, k = 0.1$		Finite element method with mesh size $h = 0.1$	
		<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$	<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$
0	1.1	0	0	4.440e-16	1.7411e-14
0.1	1.2	0.00029	0.01011	2.397e-05	0.0008257
0.2	1.3	0.00064	0.01969	0.001966	0.0601573
0.3	1.4	0.00104	0.02857	0.001066	0.0292564
0.4	1.5	0.00147	0.03664	0.000245	0.0060951
0.5	1.6	0.00192	0.04386	0.000144	0.0032965
0.6	1.7	0.00238	0.05022	0.000865	0.0182061
0.7	1.8	0.00283	0.05574	0.001287	0.0253187
0.8	1.9	0.00325	0.06045	0.000344	0.0064075

Table 3. Numerical results.

x	y	Second order central difference scheme with steps $h = 0.05, k = 0.05$		Finite element method with mesh size $h = 0.05$	
		<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$	<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$
0	1.1	0	0	3.996e-15	1.567e-13
0.1	1.2	1.6938e-05	0.00058345	0.0004007	0.0138044
0.2	1.3	5.3296e-05	0.00163013	3.983e-05	0.0012183
0.3	1.4	0.0001093	0.00300047	0.0007643	0.0209714
0.4	1.5	0.0001838	0.00457142	0.0001906	0.0047409
0.5	1.6	0.0002743	0.00624182	0.0002270	0.0051663
0.6	1.7	0.0003771	0.00793478	3.955e-05	0.0008322
0.7	1.8	0.0004880	0.00959848	0.0003905	0.0076820
0.8	1.9	0.0006026	0.01120612	0.0005635	0.0104793

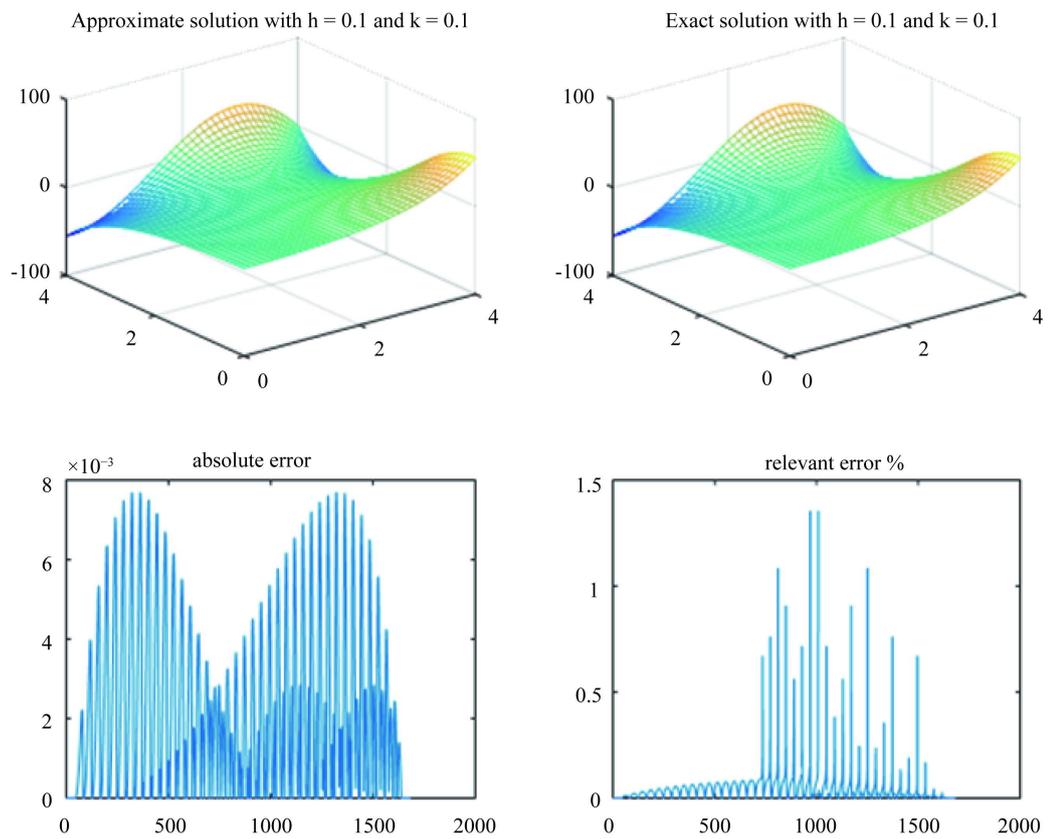


Figure 2. SCDM graphs.

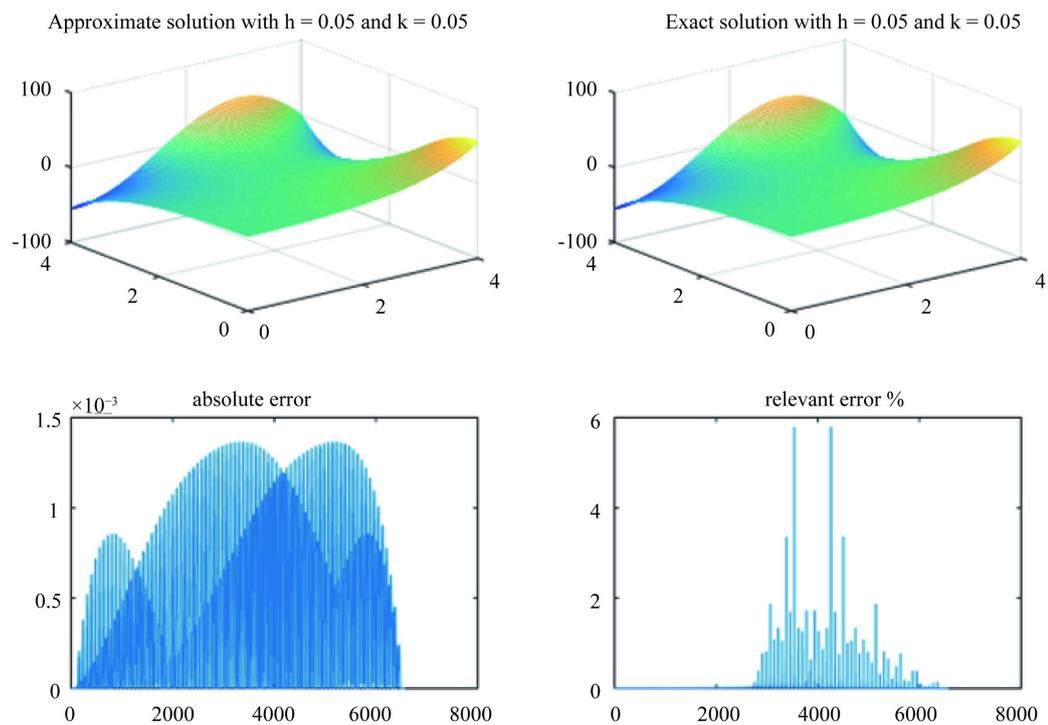


Figure 3. SCDM graphs.

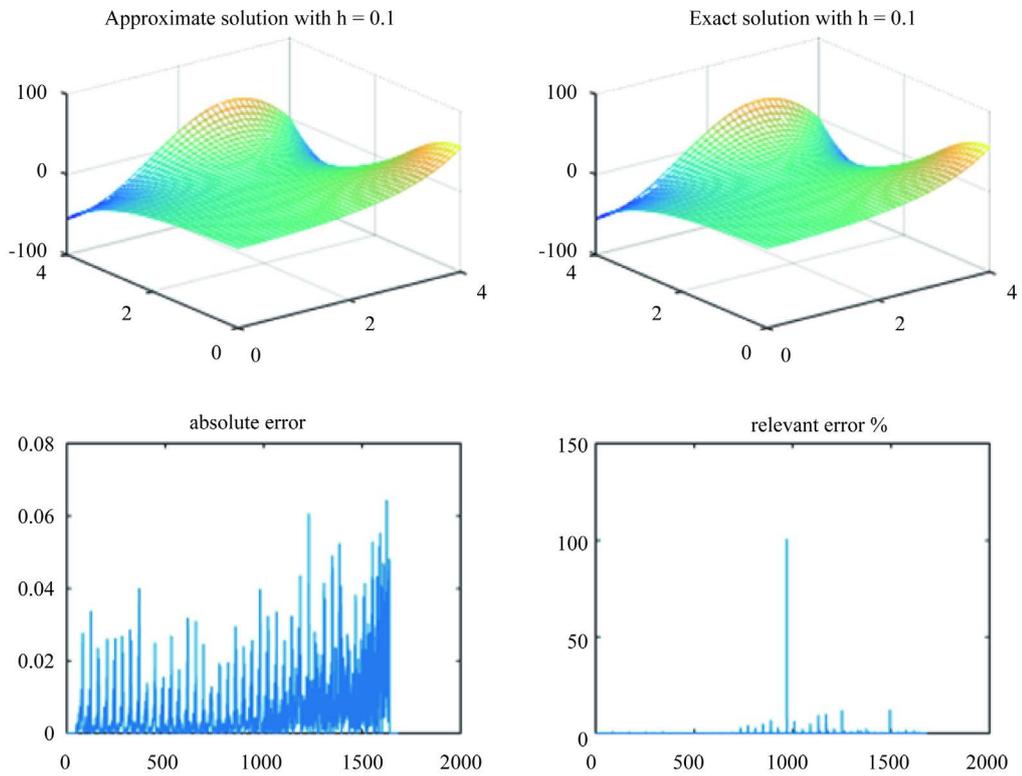


Figure 4. FEM graphs.

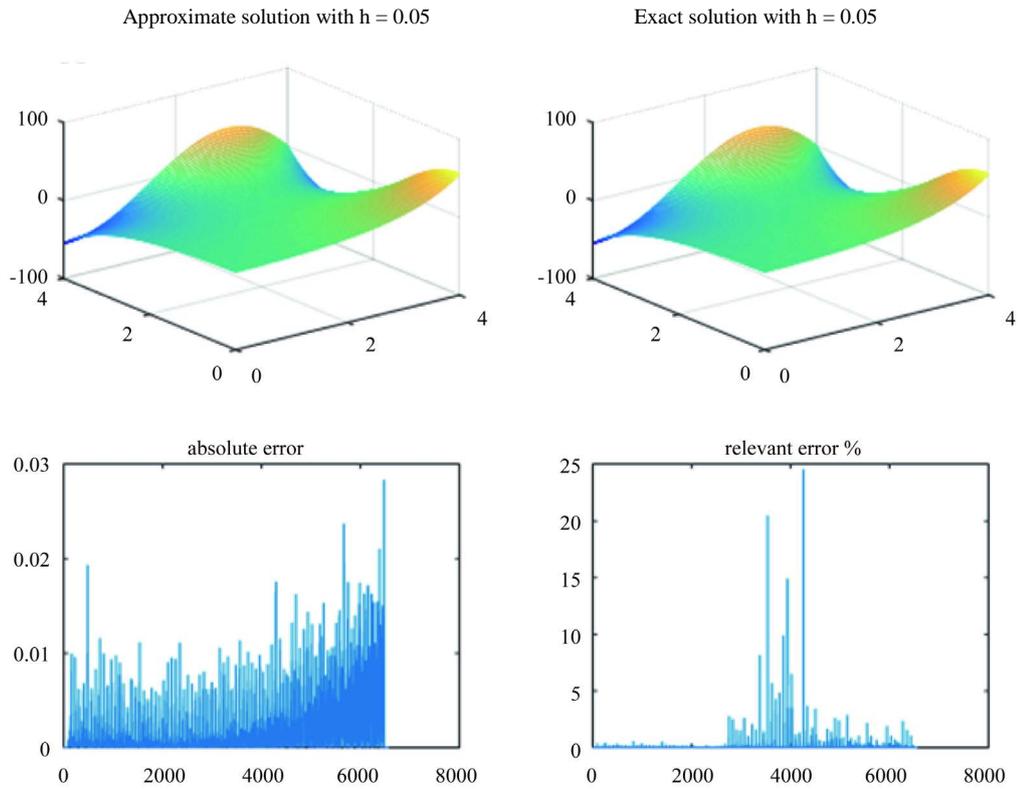


Figure 5. FEM graphs.

Results (Table 4 and Table 5)

In **Figure 6** and **Figure 7** we have the graphs for SCDM and in **Figure 8** and **Figure 9** for FEM.

Example 3

Find the approximate solution of the partial differential equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial}{\partial y} \left((1 + y^2) \frac{\partial u}{\partial y} \right) - \frac{\partial u}{\partial x} - (1 + 2y + y^2) \frac{\partial u}{\partial y} = f(x, y), \quad 0 \leq x \leq 1, 0 \leq y \leq 1$$

with Dirichlet boundary conditions along the rectangular domain

$$\begin{aligned} u(0, y) &= 0.1350e^y & u(1, y) &= 0.1350e^{y+1} \\ u(x, 0) &= 0.1350e^x & u(x, 1) &= 0.1350 \left(e^{x+1} + \log(2)(x - x^2)^2 \right) \end{aligned}$$

and exact solution

$$u(x, y) = 0.1350 \left(e^{x+y} + \log(y^2 + 1)(x - x^2)^2 \right)$$

Table 4. Numerical results.

x	y	Second order central difference scheme with steps $h = 0.1, k = 0.1$		Finite element method with mesh size $h = 0.1$	
		Absolute error $\ u - u_{approx}\ _{L_2}$	Relevant error % $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$	Absolute error $\ u - u_{approx}\ _{L_2}$	Relevant error % $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$
0	0.1	0	0	0	0
0.1	0.2	0.0006443	0.6748022	0.0027122	2.8403167
0.2	0.3	0.0018062	0.6768951	0.0020439	0.7659463
0.3	0.4	0.0032278	0.6787855	0.0047468	0.9982314
0.4	0.5	0.0045763	0.6805024	0.0065150	0.9687808
0.5	0.6	0.0055180	0.6820655	0.0059924	0.7407062
0.6	0.7	0.0057918	0.6834860	0.0026841	0.3167555

Table 5. Numerical results.

x	y	Second order central difference scheme with steps $h = 0.05, k = 0.05$		Finite element method with mesh size $h = 0.05$	
		Absolute error $\ u - u_{approx}\ _{L_2}$	Relevant error % $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$	Absolute error $\ u - u_{approx}\ _{L_2}$	Relevant error % $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$
0	0.1	0	0	0	0
0.1	0.2	9.4027e-05	0.0984664	0.0009176	0.9609699
0.2	0.3	0.0002686	0.1006831	0.0032602	1.2217428
0.3	0.4	0.0004890	0.1028373	0.0042003	0.8833012
0.4	0.5	0.0007056	0.1049364	0.0042126	0.6264231
0.5	0.6	0.0008655	0.1069866	0.0036606	0.4524755
0.6	0.7	0.0009235	0.1089924	0.0011581	0.1366761

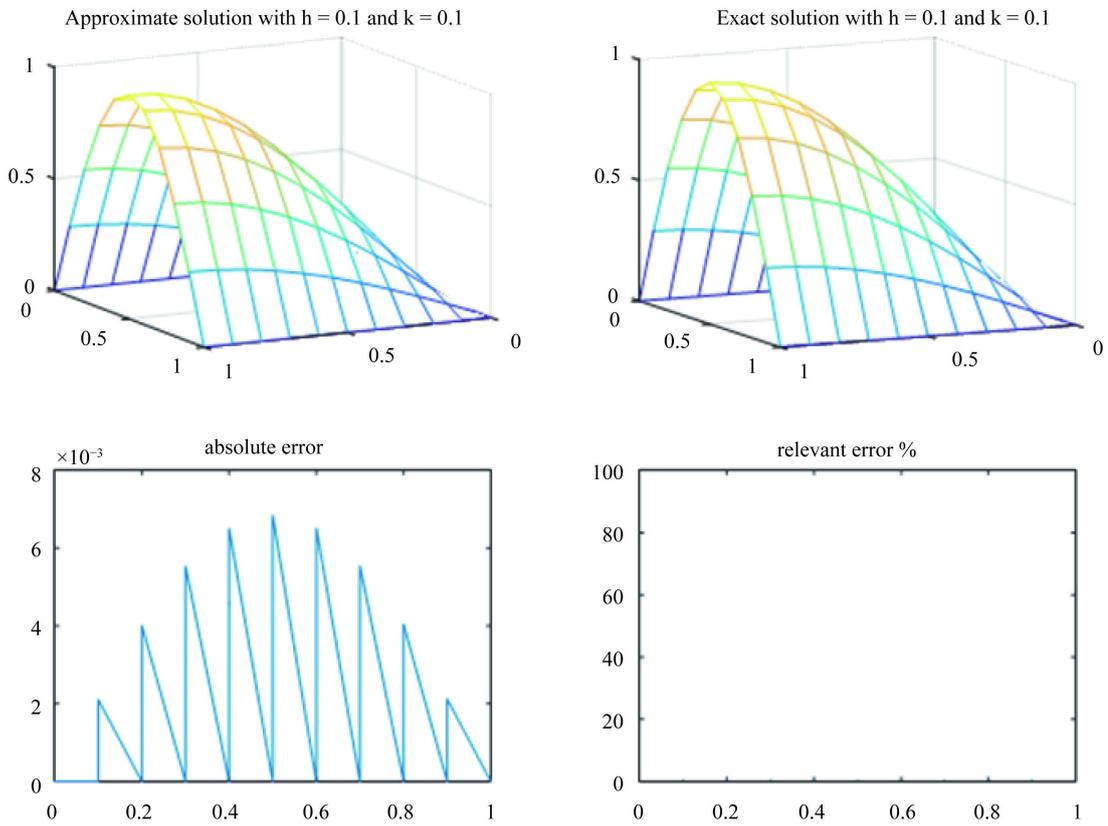


Figure 6. SCDM graphs.

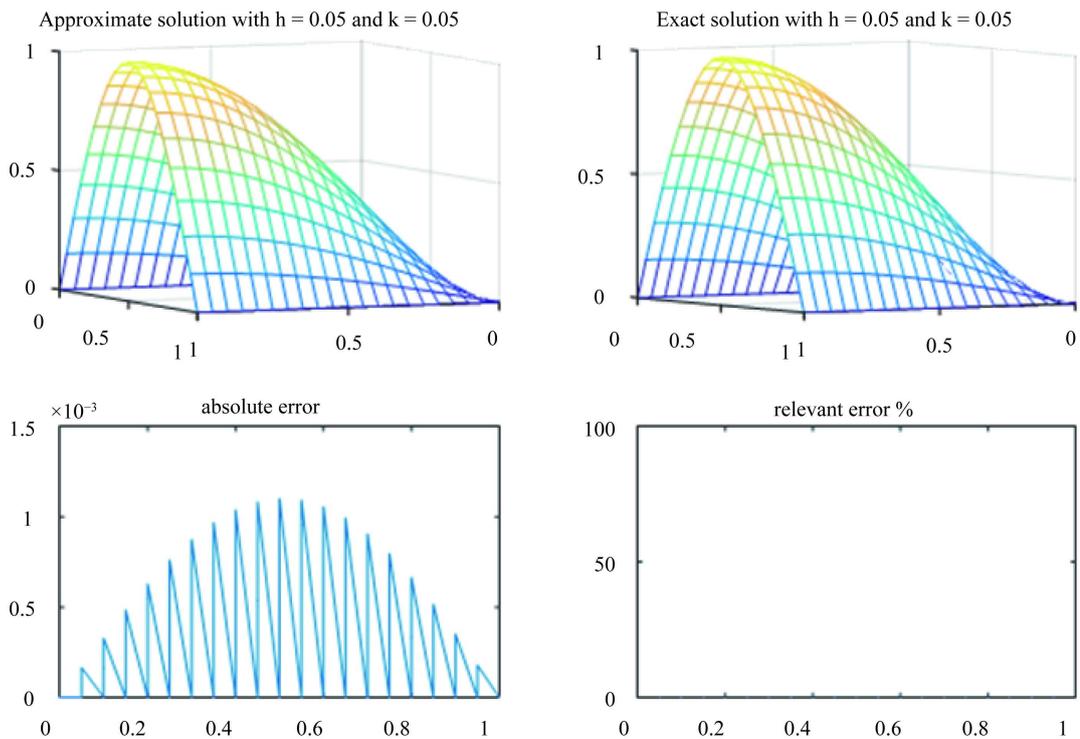


Figure 7. SCDM graphs.

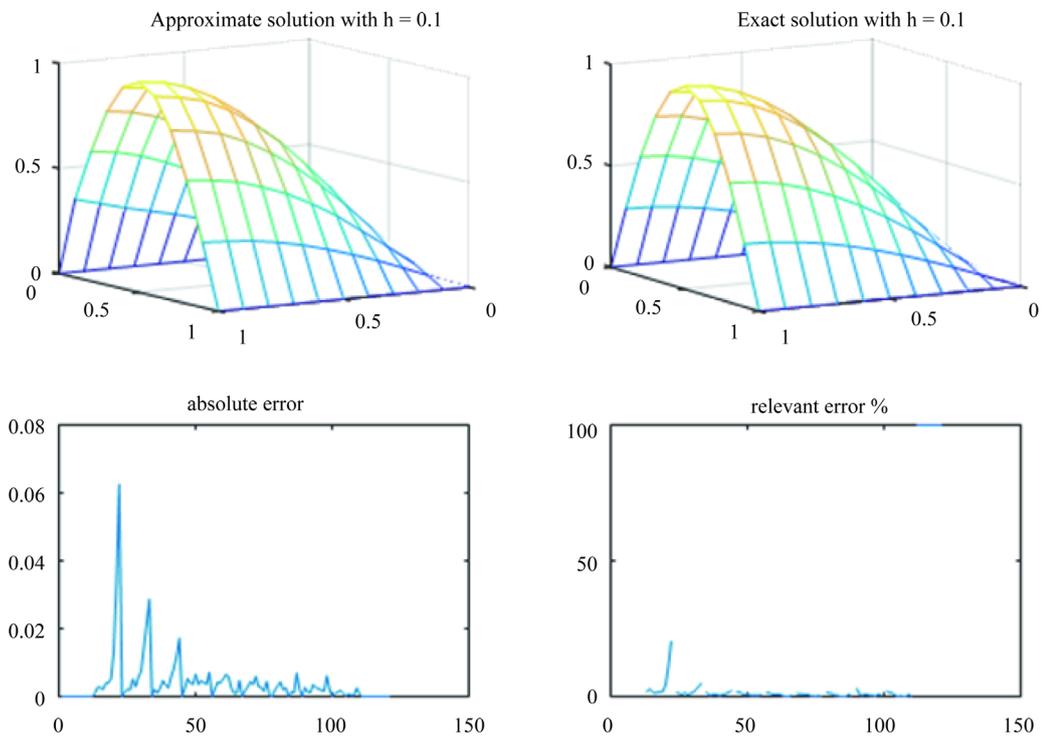


Figure 8. FEM graphs.

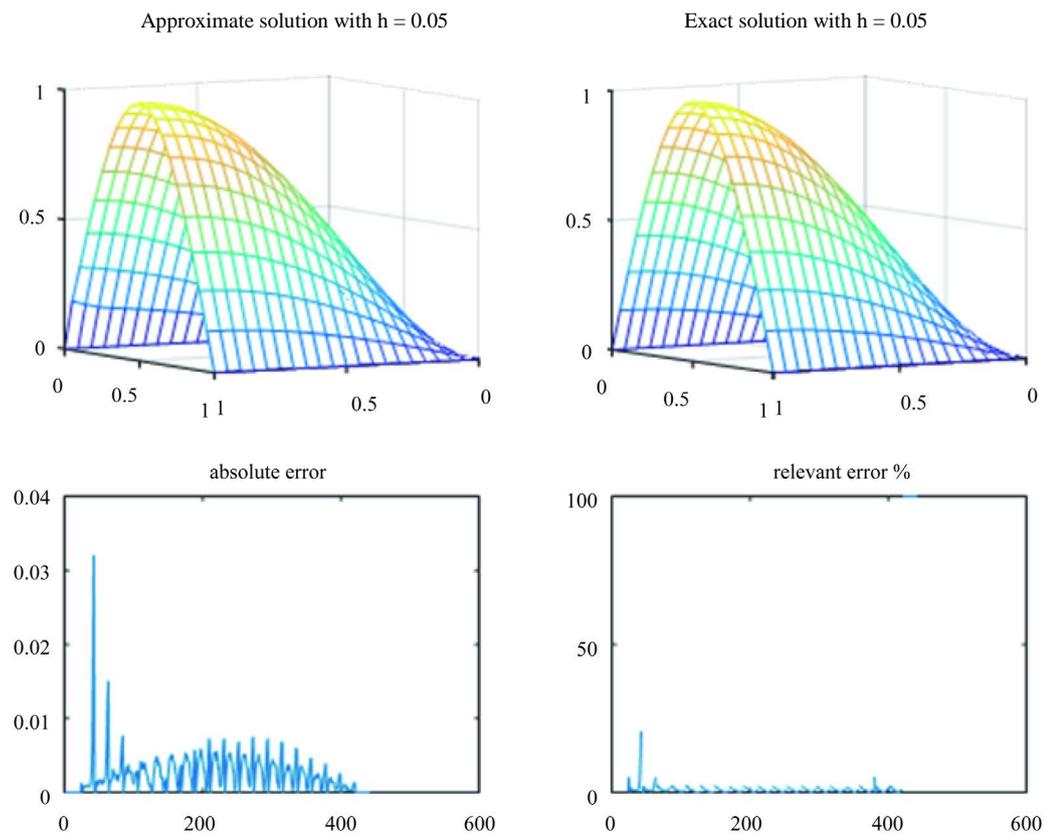


Figure 9. FEM graphs.

Results (Table 6 and Table 7)

In **Figure 10** and **Figure 11** we have the graphs for SCDM and in **Figure 12** and **Figure 13** for FEM.

Overall, what stands out from these examples is that the finite element method has better approximations for the first problem compared to finite difference method for all the step-sizes that we have. But in the second problem we have a small difference in the results with better accuracy for the finite difference method and in the third problem the finite element method has bigger relevant errors than the difference method. More specifically, in example 1 according to the tables we have better approximations for the finite element method in both of step sizes and the mesh size parameters to specific points but the graphs of the percentage of relevant errors suggest that the second order central finite difference scheme produce better approximations generally. On the other hand, in the other two examples according to the tables and the graphs of errors in the second problem we have a small difference between these methods and in the third problem we have better approximations of the second order central difference scheme almost in all points of the domain. Conclusively, we can notice that in the third example for both of these methods the results which we obtained are almost identical when different step sizes are applied in CFDM and mesh size parameters in FEM.

Table 6. Numerical results.

x	y	Second order central difference scheme with steps $h = 0.1, k = 0.1$		Finite element method with mesh size $h = 0.1$	
		<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$	<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$
0	0.1	0	0	0	0
0.2	0.2	0.0046708	2.3176755	0.0134396	6.6687403
0.4	0.3	0.0120560	4.4238110	0.0287555	10.551476
0.6	0.4	0.0189313	5.1426837	0.0383096	10.406785
0.8	0.5	0.0193264	3.8954666	0.0334284	6.7378980
1	0.6	0	0	2.886e-15	4.3169e-13

Table 7. Numerical results.

x	y	Second order central difference scheme with steps $h = 0.05, k = 0.05$		Finite element method with mesh size $h = 0.05$	
		<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$	<i>Absolute error</i> $\ u - u_{approx}\ _{L_2}$	<i>Relevant error %</i> $\frac{\ u - u_{approx}\ _{L_2}}{\ u\ _{L_2}} 100\%$
0	0.1	0	0	0	0
0.1	0.2	9.4027e-05	0.0984664	0.0009176	0.9609699
0.2	0.3	0.0002686	0.1006831	0.0032602	1.2217428
0.3	0.4	0.0004890	0.1028373	0.0042003	0.8833012
0.4	0.5	0.0007056	0.1049364	0.0042126	0.6264231
0.5	0.6	0.0008655	0.1069866	0.0036606	0.4524755
0.6	0.7	0.0009235	0.1089924	0.0011581	0.1366761

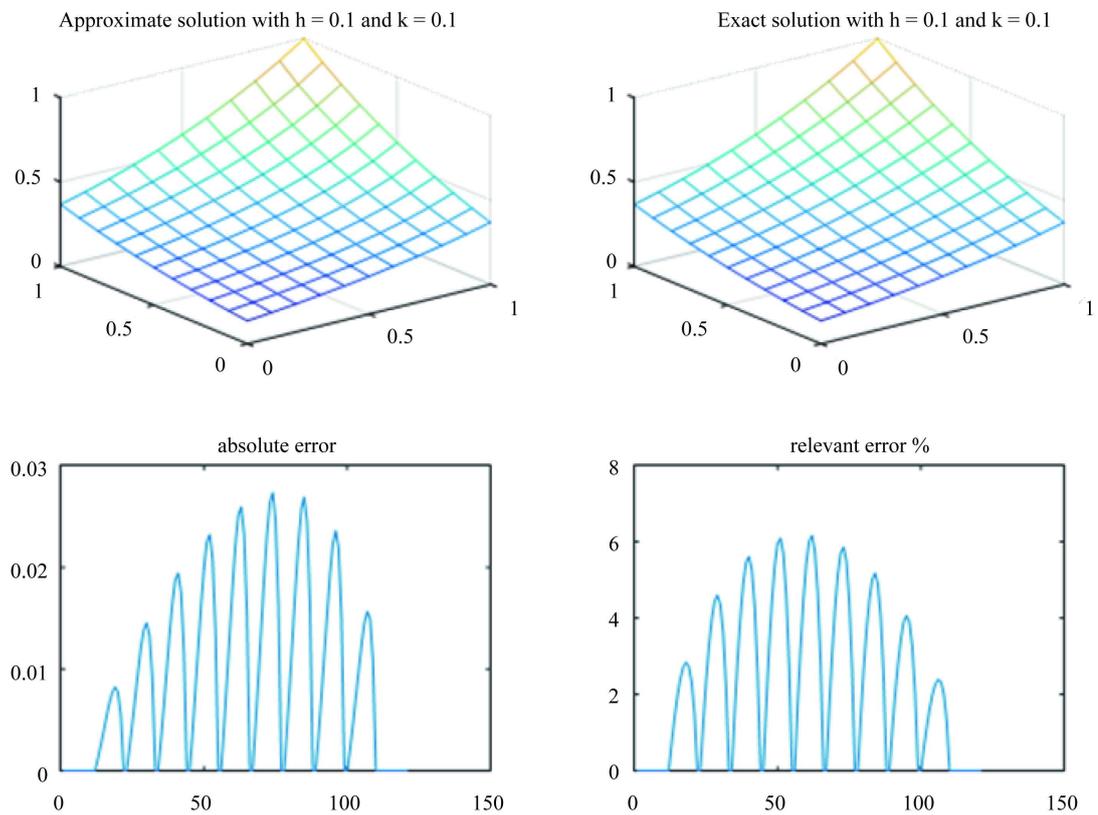


Figure 10. SCDM graphs.

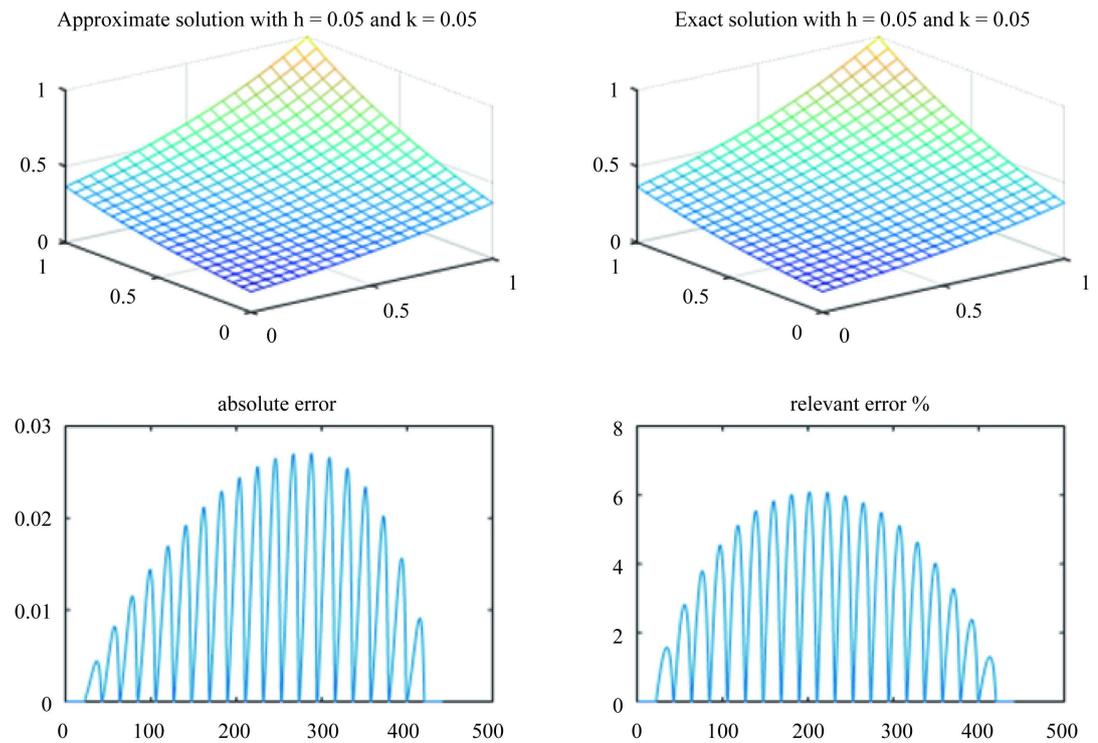


Figure 11. SCDM graphs.

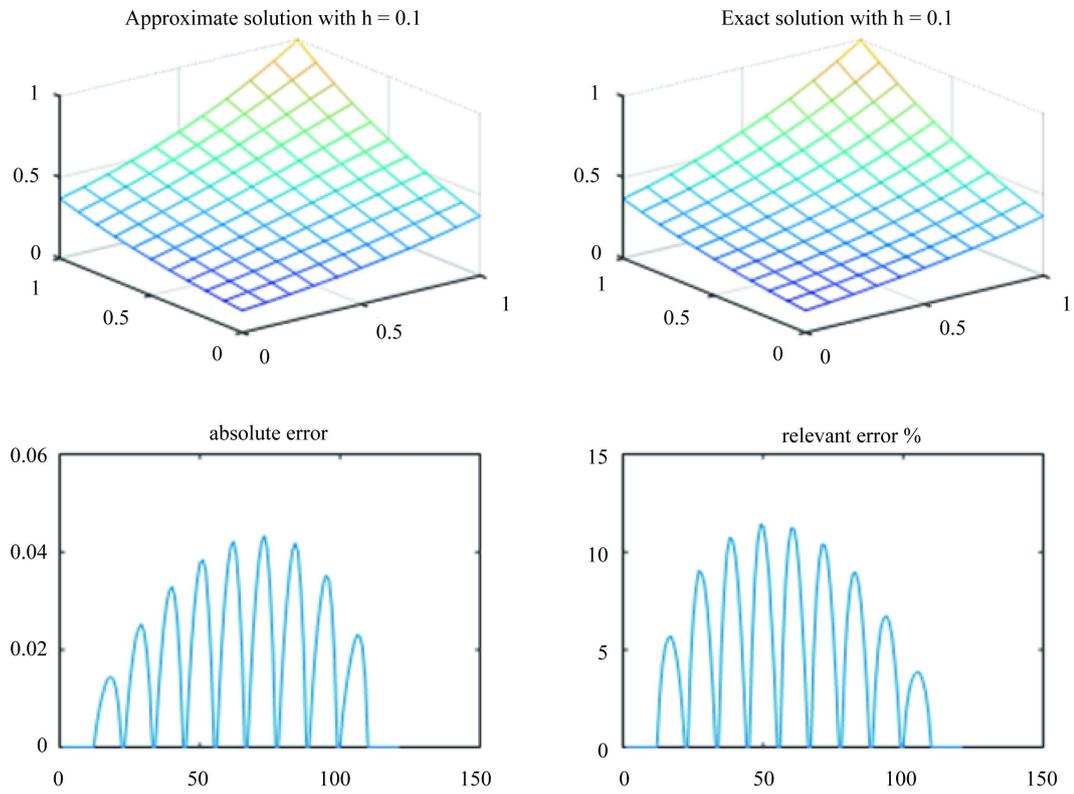


Figure 12. FEM graphs.

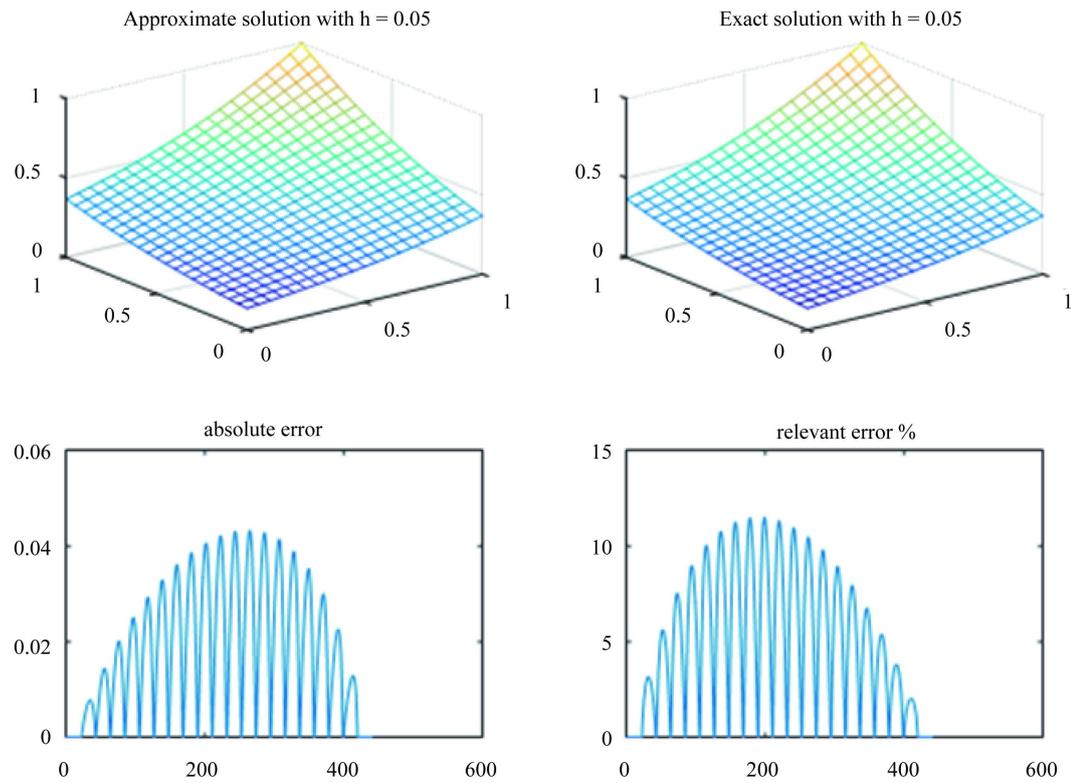


Figure 13. FEM graphs.

5. Conclusion

Finally, we can say that the data which we obtained from these examples show that both of these methods produce quite sufficient approximations for our problems. Also the results prove that the accuracy of them depends on the kind of the elliptical problem and the type of boundary conditions. For further research, the approximations of these methods can be improved. This improvement can be made if in the second order difference scheme we keep more Taylor series terms in order to approximate the derivatives and in finite element method if we use higher order elements such as quadratic Lagrange triangular elements or cubic Hermite triangular elements.

References

- [1] McDonough, J.M. (2008) Lectures on Computational Numerical Analysis of Partial Differential Equations. <http://www.engr.uky.edu/~acfd/me690-lctr-nts.pdf>
- [2] Gousidou-Koutita, M.C. (2014) Computational Mathematics. Tziolas, Ed.
- [3] Gousidou-Koutita, M.C. (2009) Numerical Methods with Applications to Ordinary and Partial Differential Equations. Notes, Department of Mathematics, Aristotle University of Thessaloniki, Thessaloniki.
- [4] Gousidou-Koutita, M.C. (2004) Numerical Analysis. Christodoulidis, K., Ed.
- [5] Dunavant, D.A. (1985) High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle. *International Journal for Numerical Methods in Engineering*, **21**, 1129-1148. <http://dx.doi.org/10.1002/nme.1620210612>
- [6] Reynolds, A.C., Burden, R.L. and Faires, J.D. (1981) Numerical Analysis. Second Edition. Youngstown State University, Youngstown.
- [7] Dougalis, V.A. (2013) Finite Element Method for the Numerical Solution of Partial Differential Equations. Revised Edition, Department of Mathematics, University of Athens, Zografou and Institute of Applied Mathematics and Computational Mathematics, FORTH, Heraklion.
- [8] Suli, E. (2012) Lecture Notes on Finite Element Method for Partial Differential Equations. Mathematical Institute, University of Oxford, Oxford.
- [9] Everstine, G.C. (2010) Numerical Solution of Partial Differential Equations. George Washington University, Washington DC.
- [10] Szabó, B. and Babuska, I. (1991) Finite Element Analysis. Wiley, New York.
- [11] Isaacson, E. and Keller, H.B. (1996) Analysis of Numerical Methods. John Wiley and Sons, New York.
- [12] Papanikos, G. (2015) Numerical Study with Methods of Characteristic Curves and Finite Element for Solving Partial Differential Equations with Matlab. Master's Thesis.
- [13] Yang, W.Y., Cao, W., Chung, T.S. and Morri, J. (2005) Applied Numerical Methods Using MATLAB. John Wiley & Sons, Hoboken.
- [14] Gropp, W.D. and Keyest, D.E. (1989) Domain Decomposition with Local Refinement. Research Report YALEU/DCS/RR-726.