FINITE DIFFERENCE METHOD ON TRIANGULATION

FAISAL A. FAIRAG*

Abstract. In this paper, we present a new method for solving Partial Differential Equations (PDE). This method combines the use of features of both Finite Element Methods (FEM) and Finite Difference methods (FDM). Similar to the FEM, this method uses the triangulation technique and function approximation. Moreover, it uses direct discretization of the PDE, similar to the FDM. The basic idea starts by selecting a finite difference representation of the PDE and a triangular element. The selected representation involves nodal point and non-nodal points. Then, the selected triangular element is used to approximate the function values of the non-nodal points. This method can be seen as a finite difference method when an irregular nodal arrangement is appropriate for a given problem. Derivation of the method with remarks is presented as well as several computational examples with their graphs and tables.

Key words. Finite Element Method, Finite Difference Method, Numerical Solution for PDE, Meshless Methods.

AMS subject classifications. 65N06, 76M20

1. Introduction. Scientists often model physical systems with PDE's. However, analytic solutions exist only for few of them. The rest must be tackled with numerical methods. There are many methods to solve PDE's numerically. For example, Finite Element Methods (FEM), Finite Difference methods (FDM), Finite Volume Method (FVM), Boundary Element Method (BEM) and others. The main popular and commonly used methods are FDM and FEM. In this paper, we present a method that combines features from both FDM and FEM. There are many motivations behind the study of this method. Firstly, the use of the triangulation technique which is one of the main features of the FEM. The triangulation technique eases the handling of any complex geometry.

Secondly, FDM starts by deriving the finite difference equations directly from the PDE by replacing the derivatives with differences. There are many high order schemes for derivatives, which generate a large number of high order schemes discretizing the most important PDE's. This provides a selection of high order schemes to suite any one's needs. The method presented here gives flexibility for choosing from high order schemes for discretizing PDE.

Finally, FEM are widely used in engineering and applications which cause a generation of huge number of mesh information (node coordinates and element node locations) for many physical problems. The method presented here has an advantage, from the computational point of view, that if one owns mesh information, this method can be easily used after adding a small code.

This method can be seen as a finite difference method on irregular grid. The majority of FDM have been devoted to the rectangular grid. There are many difficulties in applying rectangular grids on irregular domain. The major difficulties encountered when using grids with fixed rectangular problem domain is that of covering a non-rectangular problem domain. A second difficulty arising from the use of general rectangular grid is the need for substantially more nodes in the grid than is actually necessary to achieve a particular accuracy in the solution. The method

* Department of Mathematical Sciences, King Fahd University of Petroleum & Minerals, KFUPM Box 1533, Dhahran 31261, Saudi Arabia. http://faculty.kfupm.edu.sa/math/ffairag, ffairag@kfupm.edu.sa

presented here will eliminate this inefficiency by dropping the requirement that the grid be rectangular. The first attempts to apply FDM on irregular grids were published in [6, 11, 16, 17]. The basic idea was considered as an approximation method and published in [8, 10]. The several applications of the method have been done in [7, 11, 12, 16]. Various researchers consider those previous works as a one method of big class of methods recently rediscovered. This big class of methods is called Meshless Methods. Meshless Methods are developed to remove the problems of traditional meshing. These methods start by sprinkling points through the domain without requiring any pre-specified connectivity of these points or any structure. The meshless idea has been rediscovered and several meshless methods of numerical PDE solution have been presented in [13, 1, 2, 4, 3, 15]. Most of these works are based on the Moving Least Square (MLS) Method. A detailed presentation for MLS can be found in [8]. Recently, the work of Duarte and Oden [4] and Babuska and Melenk [13] add more to the understanding of these methods. They recognized that the methods based on Moving Least Square are specific instance of partitions of unity.

This paper is divided into 7 sections. We derive the method in Section 2. The solvability of the linear system is presented in Section 3. The interpolation with piecewise polynomial functions is introduced in Section 4. Some remarks are displayed in Section 5. Two computational examples are presented in Section 6. The conclusion is presented in Section 7.

2. Derivation of the Method. In this section, we provide a description of a new method applied to a simple Poisson's equation in two dimensions. This serves as an introduction to the method presented here and illustrates some of the computational aspects in its implementation.

Our model boundary value problem is the following:

Find u such that

$$\triangle\, u = f \quad \text{in} \quad \Omega, \tag{2.1}$$

$$u = g \quad \text{on} \quad \partial\Omega, \tag{2.2}$$

where $\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ and Ω is a bounded, simply connected, polygonal domain in R^2 and $\partial\Omega$ is its boundary. Let f and g be functions defined on $\overline{\Omega} = \Omega \cup \partial\Omega$. We start by subdividing the domain $\overline{\Omega}$ into triangles Ω_j , $j = 1, 2, 3, \ldots, NT$ such that

$$\overline{\Omega} = \cup_{j=1}^{N} \overline{\Omega}_j,$$

where the triangles, $\overline{\Omega}_j$, have pairwise disjoint interiors Ω_j. This mesh yields a collection of elements $\{\Omega_j\}_{j=1}^{NT}$ and nodes $\{N_i = (x_i, y_i)\}_{i=1}^{M}$. The nodes are the vertices of the triangles. These nodes are numbered globally and their $x - y$ coordinates are stored. These nodes can be classified into interior nodes $\{N_1, N_2, \cdots, N_m\}$ and nodes on the boundary $\{N_{m+1}, N_{m+2}, \cdots, N_M\}$.

Now the values of $u(N_m), u(N_{m+1}), \cdots, u(N_M)$ are known but the values of $u(N_1)$, $u(N_2)$, $\cdots$, $u(N_m)$ are unknowns and the proposed method will give an approximation for these values. Those approximated values will be denoted by

$$U(N_1), U(N_2), \cdots, U(N_m).$$

This will be done by solving an m linear system of algebraic equations in m unknowns

$$U(N_1), U(N_2), \cdots, U(N_m). \tag{2.3}$$

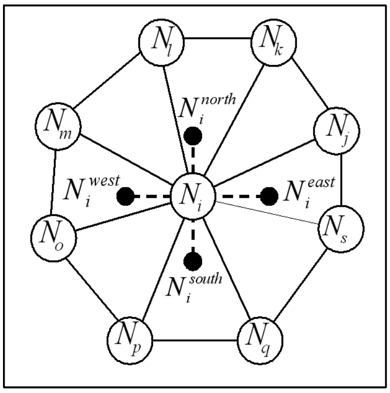


FIG. 2.1. *Node N_i and its associated four points*

For each node N_i, we generate one equation for this node in the following way. For the sake of simplicity, let us consider the node N_i in Figure (2.1).

We define G_i, direct neighbor triangles set associated with the node N_i, to be the set of all triangles whose one of their vertices is the node N_i. For each node N_i, we associate four points $N_i^{east} = (x_i + \delta, y_i)$, $N_i^{north} = (x_i + y_i, \delta)$, $N_i^{west} = (x_i - \delta, y_i)$ and $N_i^{south} = (x_i, y_i - \delta)$, where δ is chosen so that each point out of these four points belongs to at least one triangle from the set G_i. It is necessary to store the label of these four triangles. This is stored in an array $T(m \times 4)$. Specifically,

$$T(i,1) := \text{the triangle number where } N_i^{east} \text{ belongs,}$$
$$T(i,2) := \text{the triangle number where } N_i^{north} \text{ belongs,}$$
$$T(i,3) := \text{the triangle number where } N_i^{west} \text{ belongs,}$$
$$T(i,4) := \text{the triangle number where } N_i^{south} \text{ belongs.}$$

Now, the points N_i^{east}, N_i^{north}, N_i^{west}, N_i^{south} belong to triangles $\overline{\Omega}_{T(i,1)}$, $\overline{\Omega}_{T(i,2)}$, $\overline{\Omega}_{T(i,3)}$, $\overline{\Omega}_{T(i,4)}$ respectively.

To generate the i-th equation in the $m \times m$ linear system, we use the five-point scheme (Figure (5.1a)) which yields

$$\frac{4U(N_i) - [\, U(N_i^{east}) + U(N_i^{north}) + U(N_i^{west}) + U(N_i^{south})\,]}{\delta^2} = f(N_i). \tag{2.4}$$

In equation (2.4), $U(N_i)$ is one of the unknowns in equation (2.3) but $U(N_i^{east})$, $U(N_i^{north})$, $U(N_i^{west})$ and $U(N_i^{south})$ are not in the set of unknowns. Next, we need to express each of these values in terms of the unknowns $U(N_1), U(N_2), \cdots, U(N_m)$.

Since the node N_i^{east} belongs to the triangle $\overline{\Omega}_{T(i,1)}$, $U(N_i^{east})$ can be expressed in terms of $U(N_i)$, $U(N_s)$, and $U(N_j)$ as follows

$$U(N_i^{east}) = U(N_i)\ \phi_i(N_i^{east}) + U(N_s)\ \phi_s(N_i^{east}) + U(N_j)\ \phi_j(N_i^{east}), \tag{2.5}$$

where ϕ_i, ϕ_s, ϕ_j are the local basis functions for the triangle $\overline{\Omega}_{T(i,1)}$. These functions ϕ_i, ϕ_s, ϕ_j are of the form

$$\phi(x, y) = c_1 + c_2 x + c_3 y,$$

with the following conditions

$$\begin{array}{lll} \phi_i(N_i)=1\,, & \phi_i(N_s)=0\,, & \phi_i(N_j)=0\,,\\ \phi_s(N_i)=0\,, & \phi_s(N_s)=1\,, & \phi_s(N_j)=0\,,\\ \phi_j(N_i)=0\,, & \phi_j(N_s)=0\,, & \phi_j(N_j)=1\,. \end{array}$$

In the same way, we can write $U(N_i^{north})$, $U(N_i^{west})$ and $U(N_i^{south})$ as follows

$$U(N_i^{north}) = U(N_i)\ \widehat{\phi}_i(N_i^{north}) + U(N_k)\ \widehat{\phi}_k(N_i^{north}) + U(N_l)\ \widehat{\phi}_l(N_i^{north}), \tag{2.6}$$
$$U(N_i^{west}) = U(N_i)\ \widetilde{\phi}_i(N_i^{west}) + U(N_m)\ \widetilde{\phi}_m(N_i^{west}) + U(N_n)\ \widetilde{\phi}_n(N_i^{west}), \tag{2.7}$$
$$U(N_i^{south}) = U(N_i)\ \overline{\phi}_i(N_i^{south}) + U(N_p)\ \overline{\phi}_p(N_i^{south}) + U(N_q)\ \overline{\phi}_q(N_i^{south}), \tag{2.8}$$

where $\left\{\widehat{\phi}_i,\widehat{\phi}_k,\widehat{\phi}_l\right\}$, $\left\{\widetilde{\phi}_i,\widetilde{\phi}_m,\widetilde{\phi}_n\right\}$ and $\left\{\overline{\phi}_i,\overline{\phi}_p,\overline{\phi}_q\right\}$ are the local basis functions for the triangles $\overline{\Omega}_{T(i,2)}$, $\overline{\Omega}_{T(i,3)}$ and $\overline{\Omega}_{T(i,4)}$ respectively.

Using (2.5-2.8) in equation (2.4) give

$$\sum_{r\in V} c_r U(N_r) = \delta^2 f(N_i), \tag{2.9}$$

where

$$V = \{i,j,k,l,m,o,p,q,s\},$$
$$c_i = 4 - \{\phi_i(N_i^{east}) + \widehat{\phi}_i(N_i^{north}) + \widetilde{\phi}_i(N_i^{west}) + \overline{\phi}_i(N_i^{south})\},$$

$$c_j = -\phi_j(N_i^{east}),\ c_k = -\widehat{\phi}_k(N_i^{north}),\ c_l = -\widehat{\phi}_l(N_i^{north}),\ c_m = -\ \widetilde{\phi}_m(N_i^{west}),$$
$$c_o = -\widetilde{\phi}_o(N_i^{west}),\ c_p = -\overline{\phi}_p(N_i^{south}),\ c_q = -\overline{\phi}_q(N_i^{south}),\ c_s = -\ \phi_s(N_i^{east}).$$

Now, Equation (2.9) is the associated equation for the node N_i in Figure (2.1).

In general, for each node N_i where $i = 1, 2, \cdots, m$, we can generate an equation similar to (2.9). We obtain the following equation

$$\sum_{N_r\in V_i} a_{i,r} U(N_r) = \delta^2 f(N_i) \qquad i = 1, 2, \cdots, m \tag{2.10}$$

where $V_i = \{N_r : N_r$ is a vertex in one of the triangles $\overline{\Omega}_{T(i,1)}$, $\overline{\Omega}_{T(i,2)}$, $\overline{\Omega}_{T(i,3)}$, $\overline{\Omega}_{T(i,4)}\}$. Equation (2.10) is an $m \times m$ linear system of equations in the unknowns

$$U(N_1), U(N_2), \cdots, U(N_m). \tag{2.11}$$

This $m \times m$ linear system can be written as

$$A\ U \ = \ b, \tag{2.12}$$

where $U = [\ U(N_1), U(N_2), \cdots, U(N_m)\ ]^T$, $b = [\ \delta^2 f(N_1), \delta^2 f(N_2), \cdots, \delta^2 f(N_m)\ ]^T$ and $A_{i,r} = a_{i,r} \quad i, r = 1, 2, 3, \cdots, m$. The matrix A is sparse since number of non-zero entries in the i-th row is less than or equal to number of nodes in V_i. Note that the number of non-zero entries in the matrix A is less than or equal to $9m$ where m is the number of interior nodes.

3. The Solvability of the Linear System. In this section, we show that the linear system (2.12) has a unique solution. This can be done by proving that the matrix A is diagonally dominant with strict inequality for several rows, which implies that it has an inverse.

Let N^e, N^n, N^w, N^s are the four points associated with the node N where

$$N^e \in \Omega^1, \quad N^n \in \Omega^2, \quad N^w \in \Omega^3, \quad N^s \in \Omega^4,$$

and $\{\phi_1, \phi_2, \phi_3\}$, $\{\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3\}$, $\{\overline{\phi}_1, \overline{\phi}_2, \overline{\phi}_3\}$, $\{\tilde{\phi}_1, \tilde{\phi}_2, \tilde{\phi}_3\}$ are local basis functions for the triangles $\Omega^1, \Omega^2, \Omega^3, \Omega^4$ respectively.

Now from the proprieties of the local basis functions, we have

$$\begin{aligned} &\sum_{i=1}^{3} \phi_i(N^e) = 1\ , \quad \sum_{i=1}^{3} \hat{\phi}_i(N^n) = 1\ , \\ &\sum_{i=1}^{3} \overline{\phi}_i(N^w) = 1\ , \quad \sum_{i=1}^{3} \tilde{\phi}_i(N^s) = 1\ , \end{aligned} \tag{3.1}$$

and

$$\begin{aligned} &0 \le \phi_i(N^e) \le 1\ , \quad 0 \le \hat{\phi}_i(N^n) \le 1\ , \\ &0 \le \overline{\phi}_i(N^w) \le 1\ , \quad 0 \le \tilde{\phi}_i(N^s) \le 1\ , \end{aligned} \qquad i = 1, 2, 3\ . \tag{3.2}$$

Assume that the local labeling for the node N in the four triangles $\Omega^1, \Omega^2, \Omega^3, \Omega^4$ are j_1, j_2, j_3, j_4 respectively. This yields the following
Firstly, we have

$$\begin{aligned} 1 - \phi_{j_1}(N^e) &= \sum_{i=1, i\neq j_1}^{3} \phi_i(N^e), \quad 1 - \hat{\phi}_{j_2}(N^n) = \sum_{i=1, i\neq j_2}^{3} \hat{\phi}_i(N^n), \\ 1 - \overline{\phi}_{j_3}(N^w) &= \sum_{i=1, i\neq j_3}^{3} \overline{\phi}_i(N^w), \quad 1 - \tilde{\phi}_{j_4}(N^s) = \sum_{i=1, i\neq j_4}^{3} \tilde{\phi}_i(N^s), \end{aligned} \tag{3.3}$$

Secondly, using the five-point scheme (5.3a), we can show that

$$a_{ii} = 4 - [\phi_{j_1}(N^e) + \hat{\phi_{j_2}}(N^n) + \overline{\phi_{j_3}}(N^w) + \tilde{\phi_{j_4}}(N^s)]. \tag{3.4}$$

Equations (3.4) and (3.2) imply that a_{ii} is a positive quantity.
Thirdly, from the five-point scheme (2.4) and (3.2), all the nonzero entries in the row i excluding the diagonal entry (a_{ii}) are negatives.
Finally, by using the five-point scheme (2.4) we have

$$\sum_{i=1, i\neq j}^{m} (-a_{ij}) \le \sum_{i=1, i\neq j}^{3} \phi_i(N^e) + \sum_{i=1, i\neq j_2}^{3} \hat{\phi}_i(N^n) + \sum_{i=1, i\neq j_3}^{3} \overline{\phi}_i(N^w) + \sum_{i=1, i\neq j_4}^{3} \tilde{\phi}_i(N^s). \tag{3.5}$$

Equation (3.5) will have strict inequality if one of the nodes of the four triangles associated with the node N lies on the boundary. Equations (3.3) and (3.5) imply that

$$\sum_{i=1, i\neq j}^{m} -a_{ij} \le a_{ii},$$

which shows that the matrix A is strictly diagonally dominanent.

4. Interpolation with Piecewise Polynomial Functions. There are two major sources of error in using the studied method: approximation error due to approximating the function values at the four points (N^e, N^n, N^w, N^s); and discretization error due to replacing derivatives by differences. The aim of this section is to study the interpolation error which is related to the approximation of the function values. the Lemma and Remarks of this section will be later used in Section 5.

Let K be a triangle with vertices $a^{(i)},\ i = 1, 2, 3$. We define

$$\begin{aligned} h_K &= \text{the diameter of } K = \text{ the largest side of } K, \\ \sigma_K &= \text{the diameter of the circle inscribed in } K, \\ h &= \max_{K\in\Omega^h} h_K,\ \Omega^h \text{ be a given triangulation of } \Omega. \end{aligned}$$

Given $u \in C^\circ(\overline{K})$, we define the interpolant $I_K u \in P_1(K)$, where $P_1(K)$ is the space of linear functions defined on K, by

$$I_K u(a^{(i)}) = u(a^{(i)}) \quad i = 1, 2, 3.$$

Thus $I_K u$ is the linear function agreeing with u at the vertices $a^{(1)}, a^{(2)}, a^{(3)}$. The following Lemma estimates the interpolation error $u - I_K u$ on the triangle K.

LEMMA 4.1. *Let $K \in \Omega^h$ be a triangle with vertices $a^{(i)},\ i = 1, 2, 3$. Given $u \in C^2(\overline{K})$. Let the interpolant $I_K u \in P_1(K)$, where $P_1(K)$ is the space of linear functions defined on K, be defined by*

$$I_K u(a^{(i)}) = u(a^{(i)}) \quad i = 1, 2, 3.$$

Then there exist a positive constant C_1 such that

$$| \ u(z) - I_K u(z) \ | \le 2C_1 h_K^2 \quad \forall\ z \in K.$$

Proof. Let $\phi_i,\ i = 1, 2, 3$, be the basis functions for $P_1(K)$. The basis functions satisfy

$$\phi_i(a^{(i)}) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, 2, 3.$$

The interpolant $I_k u$ has the representation

$$I_K u(z) = \sum_{i=1}^{3} u(a^{(i)})\phi_i(z) \quad z \in K. \tag{4.1}$$

Taylor expansion of the function u at $\tilde{z} = (\tilde{x}, \tilde{y}) \in K$ is

$$u(z) = u(\tilde{z}) + u_x(\tilde{z})(x - \tilde{x}) + u_y(\tilde{z})(y - \tilde{y}) + R(x, y), \tag{4.2}$$

where

$$\begin{aligned} R(x, y) = \frac{1}{2}[u_{xx}(\overline{z})(x - \tilde{x})^2 + 2u_{xy}(\overline{z})(x - \tilde{x})(y - \tilde{y}) \\ + u_{yy}(\overline{z})(y - \tilde{y})^2], \end{aligned}$$

is the reminder term of order 2 and $\overline{z}$ is a point on the line segment between z and $\tilde{z}$. Setting $z = a^{(i)}$ in (4.2) gives

$$u(a^{(i)}) = u(\tilde{z}) + P_i(\tilde{z}) + R_i(\tilde{z}), \tag{4.3}$$

where

$$P_i(\tilde{z}) = u_x(\tilde{z})(x^{(i)} - \tilde{x}) + u_y(\tilde{z})(y^{(i)} - \tilde{y}),$$
$$R_i(x) = R(x, a^{(i)}).$$

Since

$$| x^{(i)} - \tilde{x} |, | y^{(i)} - \tilde{y} | \leq h_K \quad i = 1, 2, 3 \ ,$$

we have the following estimate of the reminder term

$$R_i(x) \leq 2h_K^2 C_1 \quad i = 1, 2, 3 \ , \tag{4.4}$$

where

$$C_1 = \max_{|\alpha|=2} \| D^\alpha u \|_{L_\infty(K)} \ .$$

Now (4.1) and (4.3) give

$$I_K u(\tilde{z}) = u(\tilde{z}) \sum_{i=1}^{3} \phi_i(\tilde{z}) + \sum_{i=1}^{3} P_i(\tilde{z})\phi_i(\tilde{z}) + \sum_{i=1}^{3} R_i(\tilde{z})\phi_i(\tilde{z}), \ \tilde{z} \in K. \tag{4.5}$$

From the properties of the basis functions, we have

$$\sum_{i=1}^{3} \phi_i(\tilde{z}) = 1, \tag{4.6}$$

$$\sum_{i=1}^{3} P_i(\tilde{z})\phi_i(\tilde{z}) = 0. \tag{4.7}$$

Now, (4.5), (4.6) and(4.7) give

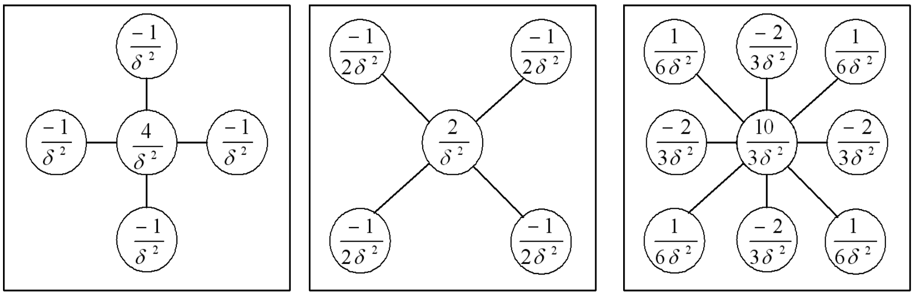
$$I_K u(\tilde{z}) = u(\tilde{z}) + \sum_{i=1}^{3} R_i(\tilde{z})\phi_i(\tilde{z}),$$

which gives us the following representation of the interpolation error

$$u(z) - I_K u(z) = -\sum_{i=1}^{3} R_i(z)\phi_i(z),$$

which implies that

$$| u(z) - I_K u(z) | \leq \sum_{i=1}^{3} | R_i(z) || \phi_i(z) | \ .$$



(a) 5-point scheme (b) × scheme (c) 9-point scheme

FIG. 5.1. *Three finite different schemes for Laplace operator*

Since $0 \le \phi_i(z) \le 1$, if $z \in K$, $i = 1, 2, 3$, we have

$$\begin{aligned} \mid u(z) - I_K u(z) \mid &\le \sum_{i=1}^{3} \mid R_i(z) \mid \phi_i(z), \\ &\le \max_i \mid R_i(z) \mid \sum_{i=1}^{3} \phi_i(z), \\ &\le \max_i \mid R_i(z) \mid . \end{aligned}$$

Using (4.4) gives

$$\mid u(z) - I_K u(z) \mid \le 2C_1 h_K^2 \quad z \in K.$$

□

REMARK 4.1. *If we work with polynomials of degree $r \ge 1$ on triangle K, we have the following estimate*

$$\mid u(z) - I_K u(z) \mid \le C h_K^{r+1} \quad z \in K.$$

5. Remarks. In this section, we will present some remarks and comments on the presented method. These remarks may lead to further research and investigation. It also give more details on this method where one may discover more features.

1. The selection of a specific finite difference scheme with a specific triangular element will generate a numerical algorithm to solve a PDE. In section 2, we consider the five-point scheme with C^0-piecewise linear in triangle. Instead of using the five-point scheme (2.4) , one may use the nine-point scheme or any higher order scheme for the Laplace operator (Figure (5.1)). Also, instead of using C^0-piecewise linear element one can use the C^0-quadratic element on triangle or any other element.
2. There are two major sources of error in using the studied method: approximation error due to approximating the function values at the four points (N^e, N^n, N^w, N^s); and discretization error due to replacing derivatives by differences. Using the standard five-point scheme, and Taylor Series expansions about the point N, we can develop the approximate relationship

$$\frac{4u(N) - [u(N^e) + u(N^n) + u(N^w) + u(N^s)]}{\delta^2} - \triangle u = C\ \delta^2. \qquad (5.1)$$

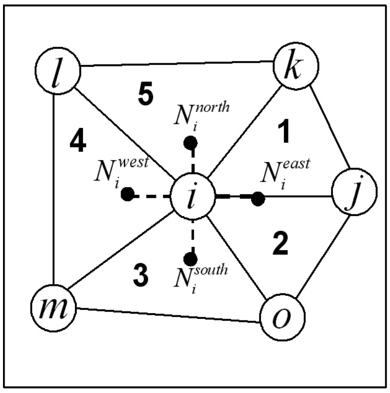


FIG. 5.2. *A case where the point* N_i^{north} *belongs to two triangles*

By using polynomial of degree r on the triangle K as an approximation method and Remark 4.1, we can write

$$U^e := U(N^e) = u(N^e) + C_e\ h_K^{r+1}, \tag{5.2}$$

$$U^n := U(N^n) = u(N^n) + C_n\ h_K^{r+1}, \tag{5.3}$$

$$U^w := U(N^w) = u(N^w) + C_w\ h_K^{r+1}, \tag{5.4}$$

$$U^s := U(N^s) = u(N^s) + C_s\ h_K^{r+1}. \tag{5.5}$$

By using (5.2-5.5), we can rewrite (5.1) as

$$\frac{4\ U - [U^e + U^n + U^w + U^s] + \tilde{C} h_K^{r+1}}{\delta^2} - \triangle u = C\ \delta^2.$$

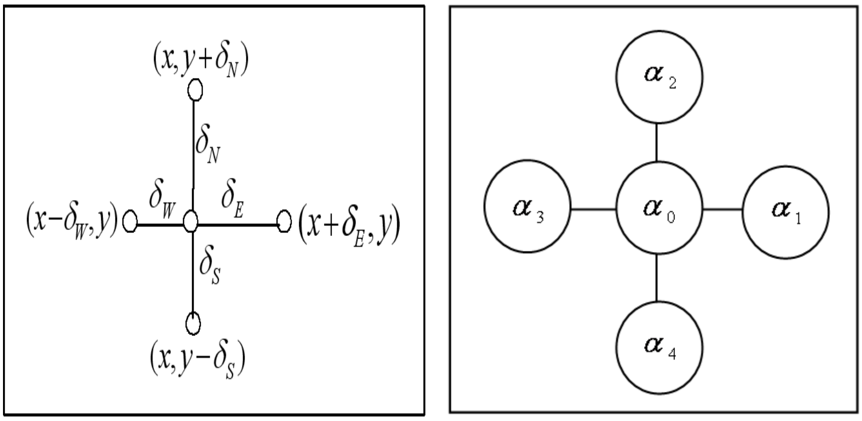
After some arrangement, we get

$$\frac{4\ U - [U^e + U^n + U^w + U^s]}{\delta^2} - \triangle u = C_1\ \delta^2 + C_2\ \frac{h_K^{r+1}}{\delta^2}. \tag{5.6}$$

The right hand side of equation (5.6) consists of two terms. The first term results from the discretization of the PDE. The second term results from the approximation method that has been used.

3. In section 2, we present a case where all the four points (N^{east}, N^{north}, N^{west}, N^{south}) are interior points of their associated triangles. It may happen that one of the four points is not an interior point of any triangle meaning that it lies on a common side of two triangles such as in Figure (5.2). The question that arises here is which of the two triangles we should associate. There are some suggested approaches to deal with this case. The first approach is by associating any one of the two triangles. The second approach is by approximating the function value of the point using the two triangles and then taking the average. The third approach is by generating two equations for this node which leads to an over determinant system of equations where number of equations is greater than number of unknowns.
4. In section 2, we present a case where δ is fixed. However, δ may vary for each node N_i. In the case of varying δ equation (5.6) can be rewritten as

$$\frac{4\ U - [U^e + U^n + U^w + U^s]}{\delta^2} - \triangle u = C_1\ \delta_K^2 + C_2\ \frac{h_K^{r+1}}{\delta_K^2}. \tag{5.7}$$



(a) (b)

FIG. 5.3. *(a) FD net with unequal spacing (b) FD for unequal five-point spacing where* $\alpha_0 = \frac{2}{\delta_e \delta_w} + \frac{2}{\delta_n \delta_s}$, $\alpha_1 = \frac{-2}{\delta_e(\delta_e+\delta_w)}$, $\alpha_2 = \frac{-2}{\delta_n(\delta_n+\delta_s)}$, $\alpha_3 = \frac{-2}{\delta_w(\delta_w+\delta_e)}$, $\alpha_4 = \frac{-2}{\delta_s(\delta_s+\delta_n)}$.

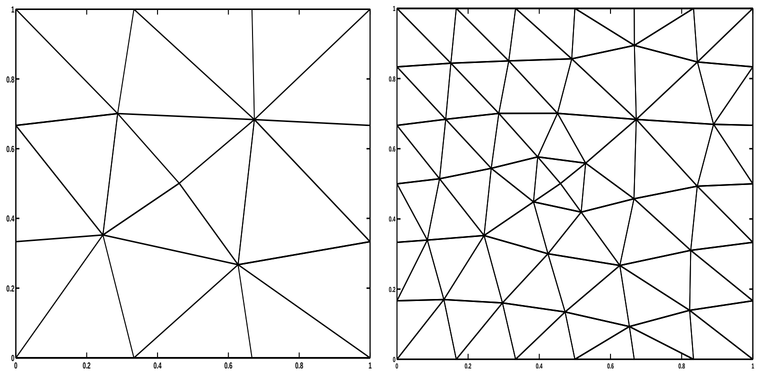
Equation (5.7) suggests a relationship between h_K and δ. This relationship is due to balancing the two terms in the right hand side. This relationship can be expressed as

$$h_K^{r+1} \approx \delta_K^4.$$

5. Also, the value of δ could be different in each direction, see Figure (5.3a). In this case, the suggested FD scheme shown in Figure (5.3b) which can be found in any FDM textbook such as [5, 9]
6. We mentioned in section 2 that the matrix A is sparse and the number of non-zero entries is less than or equal to $9m$ where m is the number of interior nodes. This is because we use the five-point scheme with C^0-piecewise linear in triangle. In general, if we use the five-point scheme with any other element, then the number of non-zero entries is less than or equal to $(4n-3)m$ where n is the number of degrees of freedom per triangle.
7. The extension of 2-dimensional derivation to 3-dimensional (or more if desired) is straightforward. Analogous to the five-point scheme, see Figure (5.1a) in two space variables, we will have a seven-point scheme. In the same manner, as the nine-point scheme, see Figure (5.1c) for two space variables was developed, we will have a 19-point scheme in three space variables. These schemes can be found in any standard FDM textbook, such as [14, 9].

6. Computational Examples. To demonstrate that the method described in this paper generates a good approximation, we apply the method to the solution of the Poisson equation in $\Omega = [0,1] \times [0,1]$ with exact solution given by

$$(a)\ u(x,y) = x(x-1)y(y-1),$$
$$(b)\ u(x,y) = e^{x+y}.$$



(a) (b)

FIG. 6.1. *Two different meshes for C^0-piecewise linear*

6.1. Example 1. We consider the following problem

$$\text{Find } u \text{ such that}$$
$$\triangle u = f \quad \text{in} \quad \Omega, \tag{6.1}$$
$$u = g \quad \text{on} \quad \partial\Omega, \tag{6.2}$$

where $f(x, y) = 2\ (x^2 + y^2 - x - y)$, $g(x, y) = 0$ and $\Omega = (0, 1) \times (0, 1)$. The exact solution for this problem is $u(x, y) = x(x-1)y(y-1)$. We use C^0- piecewise linear on triangles. We solve the problem for two different meshes, see Figures 6.1(a-b). The algorithm for this example is as follows

ALGORITHM 6.1.

1. Read the mesh, then calculate and store the data information needed.

2. Assemble the linear system $Ax = b$.

3. Solve the linear system.

The first step in Algorithm (6.1) is detailed as follows:
After the reading of the mesh data information, the code is to perform the following

1. For $i = 1, \cdots, m$, where $m =$ the number of interior nodes
   (a) identify the associated four triangles $\overline{\Omega}_{T(i,1)}$, $\overline{\Omega}_{T(i,2)}$, $\overline{\Omega}_{T(i,3)}$, $\overline{\Omega}_{T(i,4)}$ for the node N_i.
   (b) calculate $\delta_i^{(e)}$, $\delta_i^{(n)}$, $\delta_i^{(w)}$, $\delta_i^{(s)}$ where
      i. $\delta_i^{(e)} =$ the distance between the node N_i and the intersection point between the opposite side and the line $y = y_i$ as shown in Figure (6.2); and $\delta_i^{(w)}$ is calculated similarly.
      ii. $\delta_i^{(n)} =$ the distance between the node N_i and the intersection point between the opposite side and the line $x = x_i$; and $\delta_i^{(s)}$ is calculated similarly.
   (c) calculate $\delta_i = \min\{\delta_i^{(e)}, \ \delta_i^{(n)}, \ \delta_i^{(w)}, \ \delta_i^{(s)}\}$.
2. calculate $\delta_{mesh} = \min\limits_{1,\cdots,M} \delta_i$.

We solve the problem with the following options:

- FIXD: we use the standard five-point scheme with a fixed value of δ for all nodes (where $\delta = \delta_{mesh}$).

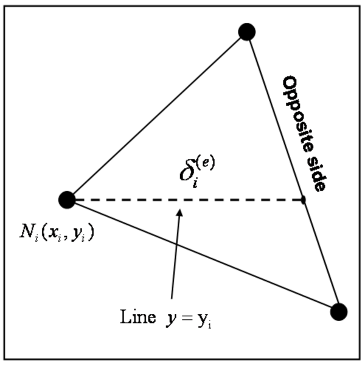


FIG. 6.2. *The triangle* $\Omega_i^{(1)}$.

TABLE 6.1
The exact values and the computed values using FIXD, VD, VDS and FEM for mesh (a)

x	y	FIXD	VD	VDS	Exact	FEM
0.4613	0.5008	0.0712	0.0444	0.0509	0.0621	0.0548
0.2875	0.5008	0.0509	0.0352	0.0401	0.0430	0.0390
0.2459	0.5008	0.0500	0.0316	0.0385	0.0423	0.0405
0.6274	0.5008	0.0513	0.0340	0.0420	0.0458	0.0437
0.6735	0.5008	0.0535	0.0393	0.0450	0.0476	0.0478
Errors		3.3e-3	5.3e-3	2.6e-3	0	1.8e-3

- VD: we use the standard five-point scheme with varying value of δ for each node (where $\delta = \delta_i,\ i = 1, 2, \cdots, M$).
- VDS: we use the unequal five-point spacing scheme in Figure (5.3b) with varying value of δ for each node and direction.
- FEM: FEM using C^0- piecewise linear on triangles.

For the four options listed above, Table (6.1) and Table (6.2) show the computed function values performed on the meshes shown in Figure (6.1a) and (6.1b), respectively. In addition, Figure (6.3a) and (6.3b) show a comparison of the u-values along the line $x = 0.5$ for the meshes shown in Figure (6.1a) and (6.1b), respectively. The results show that the VDS gives better approximation than VD, which in turns shows better approximation than FIXD. Among all of these methods, FEM shows the best approximation.

6.2. Example 2. We consider the following problem where $f(x, y) = -2e^{x+y}$, $g(x, y) = e^{x+y}$ and $\Omega = (0, 1) \times (0, 1)$. The exact solution for this problem is $u(x, y) = e^{x+y}$. We use the five-point scheme with C^0- piecewise quadratic on triangles. The mesh we use is shown in Figure (6.4a). This mesh consists of 5 interior nodes and 8 nodes on the boundary. We solve the problem with different values for δ. Table (6.3) displays the computed values and the exact values for $\delta = 0.25,\ 0.15$. This table shows good approximation of the function values with such a mesh of few nodes.

A numerical program is performed for a series of different choices of δ. Each time we calculate the difference between the computed solution and the exact solution. Then, we plot the result from these cases to obtain the graph in Figure (6.5). This Figure shows more clearly that the error decreases as the value of δ increases. The

Table 6.2
The exact values and the computed values using FIXD, VD, VDS and FEM for mesh (b)

x	y	FIXD	VD	VDS	Exact	FEM
0.6263	0.2672	0.0544	0.0354	0.0399	0.0458	0.0457
0.6725	0.6835	0.0551	0.0382	0.0425	0.0476	0.0486
0.1515	0.8438	0.0220	0.0140	0.0153	0.0169	0.0164
0.3142	0.8502	0.0349	0.0223	0.0245	0.0274	0.0269
0.1371	0.6838	0.0333	0.0205	0.0227	0.0256	0.0250
0.3955	0.5768	0.0743	0.0452	0.0501	0.0584	0.0570
0.1319	0.1702	0.0203	0.0123	0.0142	0.0162	0.0156
0.2964	0.1606	0.0342	0.0218	0.0243	0.0281	0.0278
0.0854	0.3396	0.0229	0.0129	0.0154	0.0175	0.0173
0.1197	0.5143	0.0342	0.0202	0.0230	0.0263	0.0260
0.3837	0.4485	0.0740	0.0446	0.0498	0.0585	0.0573
0.2649	0.5442	0.0618	0.0375	0.0420	0.0483	0.0474
0.8225	0.1395	0.0213	0.0136	0.0157	0.0175	0.0170
0.6532	0.0933	0.0238	0.0142	0.0168	0.0192	0.0189
0.4724	0.1347	0.0353	0.0220	0.0249	0.0291	0.0288
0.8444	0.8473	0.0196	0.0129	0.0147	0.0170	0.0165
Errors		1.7e-3	1.7e-3	1.0e-3	0	1.4e-4

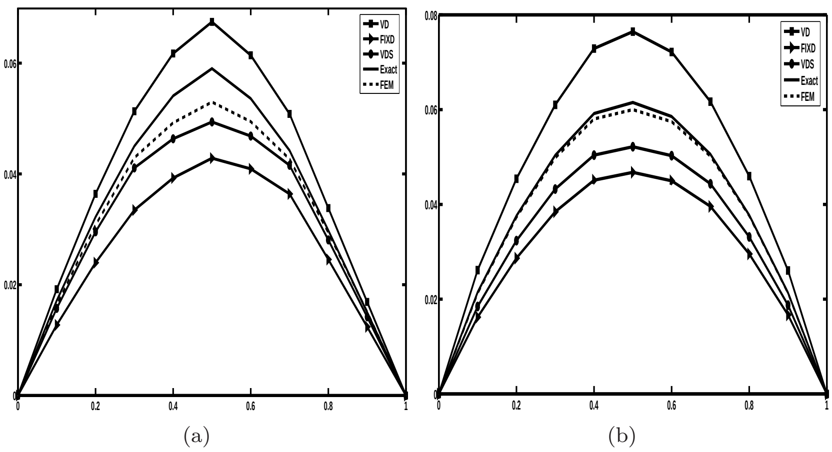
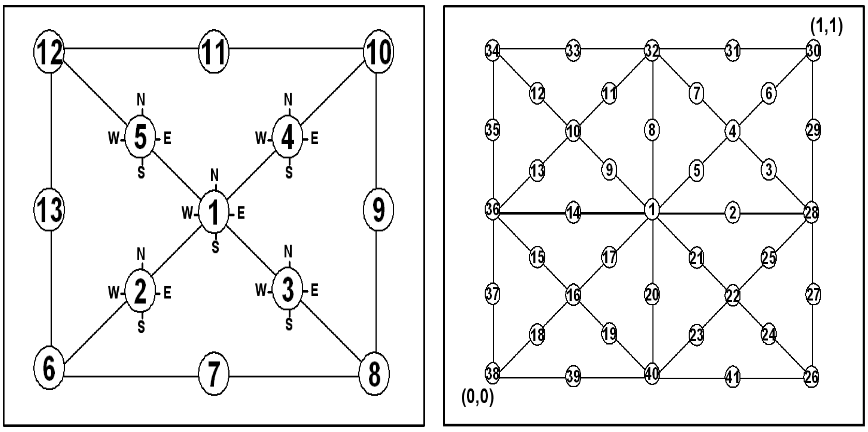


Fig. 6.3. *Comparisons of u-values along the line $x = 0.5$.*

Table 6.3
Computed and Exact values of u at (x, y) where $\delta = 0.15, 0.25$

x	y	Exact	Computed $\delta = 0.15$	Computed $\delta = 0.25$
0.5000	0.5000	2.7183	2.6911	2.7017
0.2500	0.2500	1.6487	1.6427	1.6459
0.7500	0.2500	2.7183	2.6975	2.7054
0.7500	0.7500	4.4817	4.4774	4.4813
0.2500	0.7500	2.7183	2.6975	2.7054



(a) A mesh with 5 interior nodes (b) A mesh with 25 interior nodes

FIG. 6.4. *Two meshes for C^0-piecewise quadratic*

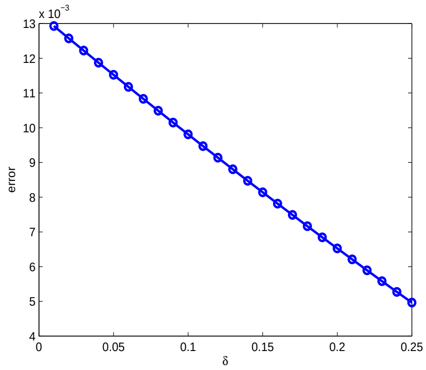


FIG. 6.5. *Error .vs. δ plot*

smallest error occurs when δ reaches its maximum value. The same problem has also been solved but for a varying δ for each node where $\delta = 0.25$ for nodes $2, 3, 4, 5$ and $\delta = 0.5$ for node 1. The error is equal to 0.005 when δ is fixed and is equal to 0.004 when δ varies.

The same problem has also been solved with more fine mesh (Figure (6.4b)). This mesh consists of 25 interior nodes and 41 nodes on the boundary. We solve the problem with $\delta = 0.125$. Table (6.4) displays the computed values and the exact values.

7. Conclusion. In this paper, we present a new method for solving PDE. This method can be seen as a Finite Difference Method when an irregular nodal arrangement is appropriate for a given problem. Several investigations on the remarks mentioned above in addition to error analysis will be considered in future research.

Acknowledgments. This research work was undertaken while the author was on sabbatical leave from King Fahd University of Petroleum and Minerals (KFUPM). The author wishes to express his gratitude to the University for the financial support.

TABLE 6.4
Computed and Exact values of u at (x, y) ($\delta = 0.125$)

x	y	Computed	Exact	x	y	Computed	Exact
0.5000	0.5000	2.7162	2.7183	0.2500	0.5000	2.1161	2.1170
0.7500	0.5000	3.4892	3.4903	0.1250	0.3750	1.6479	1.6487
0.8750	0.6250	4.4800	4.4817	0.2500	0.2500	1.6477	1.6487
0.7500	0.7500	4.4795	4.4817	0.3750	0.3750	2.1159	2.1170
0.6250	0.6250	3.4889	3.4903	0.1250	0.1250	1.2838	1.2840
0.8750	0.8750	5.7543	5.7546	0.3750	0.1250	1.6479	1.6487
0.6250	0.8750	4.4800	4.4817	0.5000	0.2500	2.1161	2.1170
0.5000	0.7500	3.4892	3.4903	0.6250	0.3750	2.7163	2.7183
0.3750	0.6250	2.7163	2.7183	0.7500	0.2500	2.7168	2.7183
0.2500	0.7500	2.7168	2.7183	0.6250	0.1250	2.1165	2.1170
0.3750	0.8750	3.4899	3.4903	0.8750	0.1250	2.7174	2.7183
0.1250	0.8750	2.7174	2.7183	0.8750	0.3750	3.4899	3.4903
0.1250	0.6250	2.1165	2.1170				

REFERENCES

[1] T. Belytschko, Y.Y. Lu, and L. Gu. Crack propagation by element free Galerkin methods. *Advanced Computational Methods for Material Modeling*, AMD-Vol. 180/PVP-Vol. 268:191–205, 1993.

[2] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *Int. J. Numer. Methods Eng.*, 37(2):229–256, 1994.

[3] C.A.M. Duarte and J.T. Oden. Hp clouds-a meshless method to solve boundary-value problems. Technical Report 95-05, TICCAM, The University of Texas at Austin, 1995.

[4] C.A.M. Duarte and J.T. Oden. *H-p* clouds – an *h-p* meshless method. *Numer. Methods Partial Differ. Equations*, pages 1–34, 1996.

[5] C. A. Hall and T. A. Porsching. *Numerical analysis of partial differential equations.* Englewood Cliffs, NJ: Prentice-Hall. xv, 319 p. , 1990.

[6] P. S. Jensen. Finite difference techniques for variable grids. *Computers and Structures*, 2:17–29, 1972.

[7] Z. Kaakowski and R. Tribillo. A generalization of the finite difference formulas. *International Journal for Numerical Methods in Engineering*, 3(9), 1975.

[8] P. Lancaster and K. Salkauskas. Surface generated by moving least square methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[9] L. Lapidus and G. F. Pinder. *Numerical solution of partial differential equations in science and engineering.* A Wiley-Interscience Publication. New York etc.: John Wiley & Sons, Inc. IV, 667 p. , 1982.

[10] Tadeusz Liszka. An interpolation method for an irregular net of nodes. *Int. J. Numer. Methods Eng.*, 20:1599–1612, 1984.

[11] T. Liszka and I. Orkisz. Finite difference method of arbitrary irregular meshes in nonlinear problems of applied mechanics. *Computers and Structures*, 11:83–95, 1980.

[12] T. Liszka and I. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *IV Structural Mechanics in Reactor Technology, San Francisco*, Aug 1977.

[13] J.M. Melenk and Ivo Babuska. The partition of unity finite element method: Basic theory and applications. *Comput. Methods Appl. Mech. Eng.*, 139(1-4):289–314, 1996.

[14] A.R. Mitchell and D.F. Griffiths. *The finite difference method in partial differential equations.* A Wiley-Interscience Publication. Chichester etc.: John Wiley & Sons. XII, 272 p. 8.95 , 1980.

[15] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.

[16] V. Pavlin and N. Perrone. Finite difference energy techniques for arbitrary meshes applied to linear plate problems. *Int. J. Numer. Methods Eng.*, 14:647–664, 1979.

[17] N. Perrone and R. Kao. A general finite difference method for arbitrary meshes. *Computers and Structures*, 5:45–58, 1975.