Moving Least Squares and Gauss Legendre for Solving the Integral Equations of the Second Kind

R. El Jid

***Abstract*—This article investigates the numerical solution of the integral equations of Fredholm and Voltera in dimension one and two. The numerical scheme developed is based on the moving least squares method. The moving least squares methodology is an effective technique for the approximation of an unknown function by using a set of disordered data. It consists of a local weighted least square fitting, valid on a small neighborhood of a point and only based on the information provided by its N closet points. Hence the method is a meshless method and does not need any background mesh or cell structures. The error analysis of the proposed method is provided. The validity and efficiency of the method are demonstrated through several tests, and through a comparison with a finite elements method.**

***Index Terms*—Meshless Method, Moving Least Squares, Integral Equations, Gauss Legendre, Numerical examples.**

I. Introduction

MANY problems of mathematical physics, engineering and mechanics can be stated in the form of two-dimensional integral equations. For example, it is usually required to solve two-dimensional integral equations in the calculation of plasma physics [1], the image deblurring problem and its regularization [2,3], axisymmetric contact problems for bodies with complex rheology [4], diffraction theory [5] and the electrochemical behavior of an inlaid microband electrode (or an array of parallel microbands) for the case of equal diffusion coefficients [6]. Therefore, the study of these types of integral equations and methods for solving them are very useful in application. There are several numerical methods for solving linear integral equations [7,8] . Kauthen in [9] used a collocation method to solve numerically the VolterraFredholm integral equations. Borzabadi and Fard in [10] obtained a numerical solution of nonlinear Fredholm integral equations of the second kind. In recent years, meshless methods have been developed as alternative numerical approaches in order to eliminate known shortcomings of the mesh-based methods [11].The main advantage of these methods is to approximate the unknowns by a linear combination of shape functions. Shape functions are based on a set of nodes and a certain weight function with a local support associated with each of these nodes. The moving least square (MLS) [12] is an approximation method of constructing continuous functions from a set of unorganized sampled point values based on the calculation of a weighted least squares approximation. Since the numerical approximations starts from scattered nodes instead of interpolation on elements, there have been many meshless methods based on the MLS scheme for the numerical solution of differential equations. Typical of them are the element-free Galerkin method (EFGM) [13,14], the HP meshless method [15], the MLS reproducing kernel method [16] and the meshless local PetrovGalerkin method [17,18]. In recent years, many researchers have worked on this subject using different technics. For examples, Chen and Wong [19] used splines functions to resolve Fredholm integral equations, M. Asgari [20] and Hosseini Shekarabi et al. [21] considered triangular functions, Mosleh and Otadi [22] used a least squares approximation, and Assari and Dehghan [23] developed a MLS method based on radial functions.

In this paper, we propose a numerical approach based on the MLS method, using distributed nodal points to approach the unknown function, in order to solve Fredholm and Voltera integral equations of the second type.The error analysis of the proposed method is also provided.

This approach is of great use for solving mechanical problems. In this sense, we applied the MLS method on a unilateral contact problem with the Signorini conditions [24]. The results obtained showed the effectiveness of the method to solve this kind of physical problems.

The remaining of this paper is organized as follows. Section 2 introduces the MLS method. In Section 3, numerical integration methods are presented. In section 4, we study the convergence of the proposed method. The use of the MLS method in Fredholm and Voltera equations are presented in sections 5 and 6. In section 7, several examples are selected to prove the effectiveness of our approach. Section 8 is devoted to the conclusion.

II. Moving least squares approximation

The MLS as an approximation method has been introduced by Shepard [25] and Lancaster and Salkauskas [26]. It consists of three components: a basis function, a weight function associated with each node, and a set of coefficients that depends on node position. Using MLS approximation, a data values $\mathbf{u} = \{u_i\}_{i=1}^{N}$ at nodes $\mathbf{x}_i$ is approximated by a function $u^h \in C^s\left(R^d\right)$ in a weighted square sense, namely

$$u^h(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x})\, a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\, \mathbf{a}(\mathbf{x}) \tag{1}$$

where $p_i(\mathbf{x}), i = 1, 2, \ldots, m$ are monomial basis functions, $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})]$, m is the number of terms in the basis, and $a_i(\mathbf{x})$ are the coefficients of the basis functions.

In general, the basis functions are as follows: for example, for a 1-D problem, the linear basis is: $\{1, x\}$, and the quadratic basis is $\{1, x, x^2\}$. For a 2-D problem, the linear basis is: $\{1, x, y\}$, and the quadratic basis is $\{1, x, y, x^2, xy, y^2\}$ where $\mathbf{x}$ = (x, y).

Manuscript received July 01, 2018, revised January 03, 2019.

R. El Jid is with the Department of Mathematics and Computing Sciences, Faculty of Sciences and Technology, University Hassan I, Settat, Morocco. e-mail: rachideljid@gmail.com.

The coefficient vector $\mathbf{a}(\mathbf{x})$ is determined by minimizing a weight discrete square norm, which is defined by:

$$J(\mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}) \left(\mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) - u_i\right)^2 \tag{2}$$

where $w_i(\mathbf{x})$ is the weight function associated with the node i, N is the number of nodes in Ω for which the weight function $w_i(\mathbf{x}) > 0$ and u_i are the fictitious nodal values, but not the nodal values of the unknown trial function $u^h(\mathbf{x})$ i.e. $u^h(x_i) = u_i$.

Equation (2) can be written as:

$$J = [P.\mathbf{a}(\mathbf{x}) - \mathbf{u}]^T .W. [P.\mathbf{a}(\mathbf{x}) - \mathbf{u}], \tag{3}$$

where

$$P = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_n) & p_2(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n) \end{bmatrix} \tag{4}$$

$$and \quad W = \begin{bmatrix} w_1(\mathbf{x}) & 0 & \cdots & 0 \\ 0 & w_2(\mathbf{x}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n(\mathbf{x}) \end{bmatrix} \tag{5}$$

The stationary point of J, in equation (2), with respect to $\mathbf{a}(\mathbf{x})$ leads to the following linear relation between $\mathbf{a}(\mathbf{x})$ and $\mathbf{u}$:

$$A(\mathbf{x})\mathbf{a}(\mathbf{x}) = B(\mathbf{x})\mathbf{u}, \tag{6}$$

where the matrices $A(\mathbf{x})$ and $B(\mathbf{x})$ are defined by

$$A(\mathbf{x}) = P^T W P = B(\mathbf{x}) P = \sum_{i=1}^{N} w_i(\mathbf{x})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i), \tag{7}$$

$$B(\mathbf{x}) = P^T W = \left[w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2), \ldots, w_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)\right]. \tag{8}$$

The matrix A is often called the moment matrix, it is of size $m \times m$. Computing $a(x)$ using equation (6) and substituting it into equation (1), give:

$$u^h(\mathbf{x}) = \Phi^T(\mathbf{x}).\mathbf{u} = \sum_{i=1}^{N} \Phi_i(\mathbf{x}).u_i, \quad \mathbf{x} \in \mathbf{\Omega} \tag{9}$$

where

$$\Phi^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})A^{-1}(\mathbf{x})B(\mathbf{x}), \tag{10}$$

and

$$\phi_i(\mathbf{x}) = \sum_{k=0}^{m} p_k(\mathbf{x})\left[A^{-1}(\mathbf{x})B(\mathbf{x})\right]_{ki}. \tag{11}$$

$\phi_i(\mathbf{x})$ are called the shape functions of the MLS approximation, corresponding to nodal point $\mathbf{x}_i$. If $w_i(\mathbf{x}) \in C^r(\Omega)$ and $p_k(\mathbf{x}) \in C^s(\Omega)$, $i = 1, 2, \ldots, n$, $k = 0, 1, \ldots, m$, then $\phi_i(\mathbf{x}) \in C^{\min(r,s)}(\Omega)$.

These form functions ϕ_i reproduce the basic functions:

$$\sum_{i=1}^{N} \phi_i(x) x_i^p = x^p, 0 \le p \le m \tag{12}$$

for $p_1(x) = 1$ we obtain the property of linear consistency also verified by the method of the finite elements:

$$\sum_{i=1}^{N} \phi_i(x) = 1 \tag{13}$$

The cubic spline or quartic spline weight function is applied in the present work as:

the cubic spline weight function:

$$w_i(\mathbf{x}) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & r \le \frac{1}{2} \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3, & \frac{1}{2} < r \le 1 \\ 0, & r > 1 \end{cases} \tag{14}$$

and the quartic spline weight function:

$$w_i(\mathbf{x}) = \begin{cases} 1 - 6r^2 + 8r^3 - 3r^4, & r \le 1 \\ 0, & r > 1 \end{cases} \tag{15}$$

with $r = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{d_i}$ where d_i is the support size of node $\mathbf{x}_i$.

In two dimensions, circular and rectangular supports are usual:

circular support:

$$w_i(x) = w_i\left(\frac{\|x_i - x\|}{d_i}\right) \tag{16}$$

rectangular support:

$$w_i(x) = w_i\left(\frac{|x_i - x|}{d_i^x}\right) w_i\left(\frac{|y_i - y|}{d_i^y}\right) \tag{17}$$

In two dimensions and a square domain $[0.4] \times [0.4]$ with a regular distribution of 5×5 knots by choosing circular influence domain of support $d = 1.5$ one obtains the folowing figures that show the support (circlar support); the weight function and the function of form.

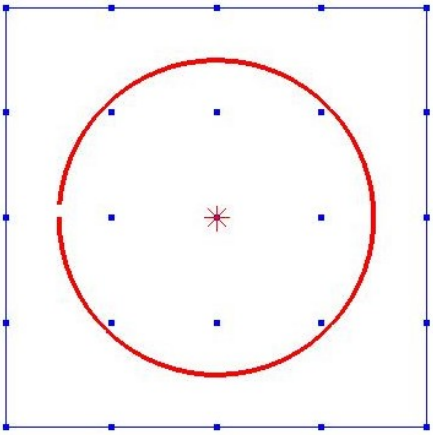

Fig. 1. The support of the central node (red circle).

III. Numerical intgration

In the MLS method, the concept of element does not exist, and the shape functions are not polynomial, so we can not evaluate the integrals as for the finite element method. However, we can use a direct integration nodes or an underlying grid that serves only in the numerical integration, and does not interfere in the approximation scheme. In this work, we use the second method [27] with a Gaussian quadrature.

IV. convergence in the MLS context

If we consider a space of functions for example, or other spaces, we know that different types of standard can be considered to measure different errors or quantities. In the context of MLS, we will also have a number of different standards depending on the context to be considered. Recall that the MLS approximation is written:

$$u^{app}(x) = \sum_{x_i \in V(x)} u_i \phi_i(x) \tag{18}$$

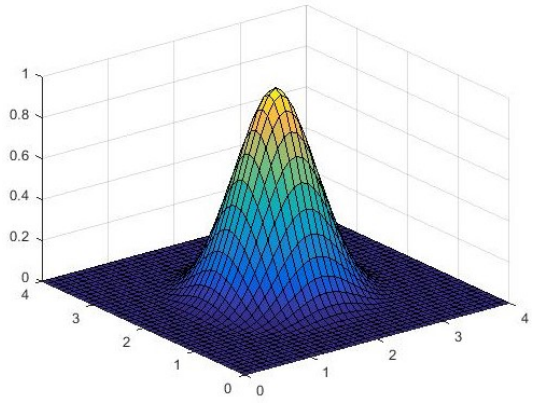

Fig. 2. weight function at the central node

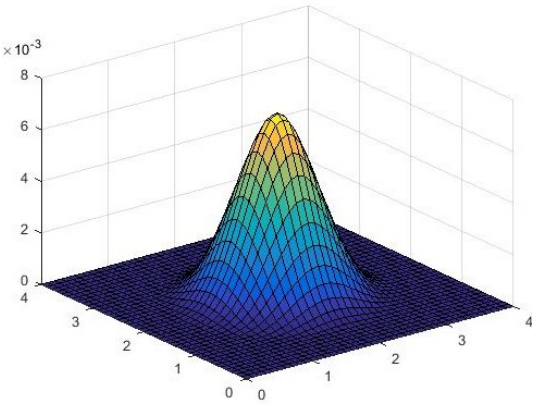

Fig. 3. function of form at the central node

where u_i means either numerically obtained nodal values $u^{num}(x_i)$, or the exact analytical values $u^{ex}(x_i)$ at the considered MLS node.

A. The numerical nodal error

1) L^∞-norm: The infinite nodal norm is defined by

$$\|U^{num} - U^{ex}\|_\infty = \sup_{1\leq i\leq N} |u^{num}(x_i) - u^{ex}(x_i)| \tag{19}$$

where $u^{num}(x_i)$ denotes the value obtained numerically and $u^{ex}(x_i)$ denotes the analytical value.

2) L^2-norm: The norm L^2 is defined by

$$\|U^{num} - U^{ex}\|_2 = \sum_{i=1}^{N} (u^{num}(x_i) - u^{ex}(x_i))^2 \tag{20}$$

B. Convergence in the case of a class solution C^{p+1}

The MLS u^{app} approximation can be compared to the exact solution u^{ex}. We consider the case 1D to simlify the proof, we also assume that the degree of reproduction is p and that the size of the supports is r

Theorem 1.

$$\|u^{app} - u^{ex}\|_{L^\infty([0,L])} \leq \frac{M}{(p+1)!} r^{p+1} \tag{21}$$

and

$$\|u^{app} - u^{ex}\|_{L^2([0,L])} \leq \frac{M}{(p+1)!} r^{p+1} \tag{22}$$

Deriving $u^{ex} - u^{app}$, we obtain a remainder of order p in norm H^1.

$$\|u^{app} - u^{ex}\|_{H^1([0,L])} \leq \frac{M}{p!} r^{p} \tag{23}$$

Before starting the demonstration, we recall a definition:

Definition 1. *We will talk about consistency or degree of reproduction in the order p when*

$$\sum_{i=1}^{N} \Phi_i(x) x_i^k = x^k, \forall k \in [0,p] \tag{24}$$

Proof 1. *First we implement the partition hypothesis of unity*

$$\sum_{i=1}^{N} \Phi_i(x) = 1 \tag{25}$$

then

$$u^{ex}(x) - u^{app}(x) = u^{ex} - \sum_{i=1}^{N} u^{ex}(x_i)\Phi_i(x) \tag{26}$$

$$u^{ex}(x) - u^{app}(x) = u^{ex} - \sum_{i=1}^{N} (u^{ex}(x) - u^{ex}(x_i))\Phi_i(x) \tag{27}$$

The development of Taylor Lagrange with integral remainder of u^{ex} around x gives :

$$\begin{aligned} u^{ex}(x) - u^{app}(x) = \sum_{i=1}^{N} (- \sum_{j=1}^{p} u^{ex(j)}(x) \frac{(x_i - x)^j}{j!} \\ - \int_x^{x_i} u^{ex(p+1)} \frac{(x_i - t)^p}{p!} dt) \Phi_i(x) \end{aligned} \tag{28}$$

or

$$\begin{aligned} u^{ex}(x) - u^{app}(x) = - \sum_{i=1}^{N} (\sum_{j=1}^{p} u^{ex(j)}(x) \frac{(x_i - x)^j}{j!}) \Phi_i(x) + \\ \sum_{i=1}^{N} \int_x^{x_i} u^{ex(p+1)} \frac{(x_i - t)^p}{p!} dt \Phi_i(x) \end{aligned} \tag{29}$$

on the other hand

$$\begin{aligned} \sum_{i=1}^{N} (\sum_{j=1}^{p} u^{ex(j)}(x) \frac{(x_i - x)^j}{j!}) \Phi_i(x) = \\ \sum_{j=1}^{p} \frac{u^{ex(j)}(x)}{j!} (\sum_{i=1}^{N} (x_i - x)^j \Phi_i(x)) \end{aligned} \tag{30}$$

Using the reproduction conditions in the order p (24) we obtain

$$\sum_{i=1}^{N} \Phi_i(x) (\frac{x_i - x}{r})^j = 0, \qquad \forall j \in [0,p] \tag{31}$$

So we can say that

$$\sum_{j=1}^{p} \frac{u^{ex(j)}(x)}{j!} (\sum_{i=1}^{N} (x_i - x)^j \Phi_i(x)) = 0 \tag{32}$$

and we reduce the expression of the error to

$$u^{app}(x) - u^{ex}(x) = \sum_{i=1}^{N} (\int_x^{x_i} u^{ex(p+1)} \frac{(x_i - t)^p}{p!} dt) \Phi_i(x) \tag{33}$$

Using the fact that $u^{ex} \in C^{p+1}([0,L])$, and thus its $(p+1)th$ derivative is continuous bounded on any compact we obtain

$$|u^{app}(x) - u^{ex}(x)| \leq \frac{M}{p!} \sum_{i=1}^{N} (\int_x^{x_i} |x_i - t|^p dt) \Phi_i(x) \tag{34}$$

or

$$|u^{app}(x) - u^{ex}(x)| \leq \frac{M}{(p+1)!} \sum_{i=1}^{N} |x - x_i|^{p+1} \Phi_i(x) \tag{35}$$

as $|x_i - x| \leq r$ *(because if not* $\Phi_i(x) = 0$*) we have*

$$|u^{app}(x) - u^{ex}(x)| \leq \frac{M}{(p+1)!} r^{p+1}, \qquad \forall x \in [0, L] \tag{36}$$

so

$$\|u^{app} - u^{ex}\|_{L^\infty([0,L])} \leq \frac{M}{(p+1)!} r^{p+1} \tag{37}$$

and

$$\|u^{app} - u^{ex}\|_{L^2([0,L])} \leq \frac{M}{(p+1)!} r^{p+1} \tag{38}$$

by deriving $u^{ex} - u^{app}$ *we get*

$$\|u^{app} - u^{ex}\|_{H^1([0,L])} \leq \frac{M}{p!} r^{p} \tag{39}$$

Theorem 2. *If we assume only* $u^{ex} \in H^{p+1}$ *and no more than class* C^{p+1} *then we get that*

$$\|u^{app} - u^{ex}\|_{L^2([0,L])} \leq C r^{p+\frac{1}{2}} \|u^{ex}\|_{H^{p+1}([0,L])} \tag{40}$$

Proof 2.

$$u^{app}(x) - u^{ex}(x) = \sum_{i=1}^{N} \left(\int_x^{x_i} u^{ex(p+1)}(t) \frac{(x_i - t)^p}{p!} dt\right) \Phi_i(x) \tag{41}$$

Using the fact that $u^{ex} \in H^{p+1}([0, L])$*, we can use the Cauchy-Schwartz inequality to increase the* $(p+1)th$ *derivative from* u*,*

$$u^{app}(x) - u^{ex}(x) = \sum_{i=1}^{N} \left(\int_x^{x_i} |u^{ex(p+1)}(t) \frac{(x_i - t)^p}{p!} dt\right) \Phi_i(x)| \tag{42}$$

or

$$|u^{app}(x) - u^{ex}(x)| \leq \sum_{i=1}^{N} \left(\int_x^{x_i} |u^{ex(p+1)}(t)|^2 dt\right)^{\frac{1}{2}} \left(\int_x^{x_i} |x_i - t|^{2p} dt\right)^{\frac{1}{2}} |\Phi_i(x)| \tag{43}$$

As the form functions are compact support, we have :

$$\Phi_i(x) = 0 \quad for \quad |x - x_i| > r \tag{44}$$

consequently,

$$\begin{aligned} |u^{app}(x) - u^{ex}(x)| &\leq C r^{p+\frac{1}{2}} \|u^{ex(p+1)}\|_{L^2} \\ &\leq C r^{p+\frac{1}{2}} \|u^{ex}\|_{H^{p+1}} \end{aligned} \tag{45}$$

V. Application of the MLS method to Fredholm integral equations

A. 1-D Fredholm integral equation

We consider the 1-D Fredholm's equation defined by:

$$\lambda u(x) + \int_a^b k(x, \xi) u(\xi)\, \mathrm{d}\xi = g(x), \qquad x \in [a, b] \tag{46}$$

where u is unknown function, λ is a real parameter, a and b are finite real numbers and $k(x, \xi)$ is called kernel function. Assume that u, λ, k and g have necessary conditions for solvability of the equation. To apply the method, at first N nodal points xi are selected on interval $[a, b]$ where $a \leqslant x_1 < x_2 < \ldots < x_N \leqslant b$. The distribution of nodes could be selected regularly or randomly. Then, instead of $u(x)$, we can replace $u^h(x)$ from (9), then the equation (46) becomes :

$$\lambda u^h(x) + \int_a^b k(x, \xi) u^h(\xi)\, \mathrm{d}\xi = g(x), \qquad x \in [a, b] \tag{47}$$

or equivalently

$$\sum_{j=1}^{N} [\lambda \phi_j(x) + \int_a^b k(x, \xi) \phi_j(\xi)\, \mathrm{d}\xi] u_j = g(x), \qquad x \in [a, b] \tag{48}$$

or

$$\sum_{j=1}^{N} [\lambda \phi_j(x_i) + \int_a^b k(x_i, \xi) \phi_j(\xi)\, \mathrm{d}\xi] u_j = g(x_i), \quad i = 1, 2, \ldots, N \tag{49}$$

Using the quadrature method of gauss with the weights w_k and the coefficients ξ_k in $[a, b]$ to approximate the numerical value of the integral in (49) we obtain:

$$\sum_{j=1}^{N} [\lambda \phi_j(x_i) + \sum_{k=1}^{m_1} k(x_i, \xi_k) \phi_j(\xi_k) \omega_k] u_j = g(x_i), \ i = 1, 2, \ldots, N \tag{50}$$

if we define F, u, and g by

$$F_{i,j} = \lambda \phi_j(x_i) + \sum_{k=1}^{m_1} k(x_i, \xi_k) \phi_j(\xi_k) \omega_k \tag{51}$$

$$u = [u_1, u_2, \ldots, u_N]^T \tag{52}$$

and

$$g = [g_1, g_2, \ldots, g_N]^T \quad with \quad g_i = g(x_i) \tag{53}$$

we get the system of the following linear equations :

$$Fu = g \tag{54}$$

the resolution of (54) gives the approximate value of u at each point of $x \in [a, b]$ by :

$$u(x) \simeq u^h(x) = \sum_{j=1}^{N} \phi_j(x) u_j \tag{55}$$

B. 2-D Fredholm integral equation

We consider the 2-D Fredholm's equation defined by:

$$\lambda u(x, y) + \int_\Omega k(x, y, \xi, \eta) u(\xi, \eta) \mathrm{d}\xi d\eta = g(x, y), \ (x, y) \in \Omega \tag{56}$$

where u is an unknown function, k and g are known functions, λ a real parameter and Ω is a 2-dimensional bounded domain. We consider Ω in the form

$$\Omega = \{(\xi, \eta) \in \mathbb{R}^2 - 1 \leq \eta \leq 1, \varphi_1(\eta) \leq \xi \leq \varphi_2(\eta)\}$$

To apply the method, we take N nodes X_i arbitrarily selected in Ω, we replace $u(x, y)$ with $u^h(x, y)$ in (56) then the equation is written.

$$\begin{aligned} \sum_{j=1}^{N} [\lambda \phi_j(x, y) + \int_{-1}^{1} \int_{\varphi_1(\eta)}^{\varphi_2(\eta)} k(x, y, \xi, \eta) \phi_j(\xi, \eta) \mathrm{d}\xi \mathrm{d}\eta] u_j \\ = g(x, y), \ (x, y) \in \Omega \end{aligned} \tag{57}$$

We transform the interval $[\varphi_1(\eta), \varphi_2(\eta)]$ to the fixed interval $[-1, 1]$ by the following linear transformation:

$$\xi(\eta, \theta) = \frac{\varphi_2(\eta) - \varphi_1(\eta)}{2} \theta + \frac{\varphi_2(\eta) + \varphi_1(\eta)}{2}$$

then the equation (57) is written in the form:

$$\begin{aligned} \sum_{j=1}^{N} [\lambda \phi_j(x, y) + \int_{-1}^{1} \int_{-1}^{1} K(x, y, \xi(\eta, \theta), \eta) \phi_j(\xi(\eta, \theta), \eta) \mathrm{d}\theta \mathrm{d}\eta] u_j \\ = g(x, y), \ (x, y) \in \Omega \end{aligned} \tag{58}$$

where

$$K(x, y, \xi(\eta, \theta), \eta) = \frac{\varphi_2(\eta) - \varphi_1(\eta)}{2} k(x, y, \xi(\eta, \theta), \eta)$$

If (58) checks at each point (x_i, y_i), then we will have

$$\sum_{i=j}^{N}[\lambda\phi_j(x_i,y_i)+\int_{-1}^{1}\int_{-1}^{1}K(x_i,y_i,\xi(\eta,\theta),\eta)\phi_j(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta]u_j = g(x_i,y_i), \tag{59}$$

Using the Gauss quadrature method with the weights w_k, w_p and the coefficients ξ_k, η_p in the interval $[-1,1]$ to approximate the numerical value of the double integral in (59) we obtain:

$$\sum_{j=1}^{N}[\lambda\phi_j(x_i,y_i)+ \sum_{p=1}^{m_1}\sum_{k=1}^{m_1}K(x_i,y_i,\xi(\eta_p,\theta_k),\eta_p)\phi_j(\xi_p,\eta_k)\omega_k\omega_p]u_j = g(x_i,y_i), \tag{60}$$

If we set F as a $N \times N$ matrix defined by :

$$F_{i,j} = \lambda\phi_j(x_i,y_i)+ \sum_{p=1}^{m_1}\sum_{k=1}^{m_1}K(x_i,y_i,\xi(\eta_p,\theta_k),\eta_p)\phi_j(\xi(\eta_p,\theta_k),\eta_p)\omega_k\omega_p \tag{61}$$

and vectors :

$$u = [u_1, u_2, \ldots, u_N]^T$$

$$g = [g_1, g_2, \ldots, g_N]^T$$

Then we get the system of linear equations:

$$Fu = g \tag{62}$$

Solving (62). the value of $u(x,y)$ at each point $(x,y) \in \Omega$ is approximated by :

$$u(x,y) \approx u^h(x,y) = \sum_{j=1}^{N}\phi_j(x,y)u_j,\ (x,y) \in \Omega \tag{63}$$

VI. Application to the integral equations of Volterra

A. 1-D Volterra integral equations

We consider the Volterra equation defined by:

$$\lambda u(x) + \int_a^x k(x,\xi)u(\xi)\,\mathrm{d}\xi = g(x),\ x \in [a,b] \tag{64}$$

We take N nodes x_i chosen in the interval $[a,b]$ such that: $a \le x_1 < x_2 < \ldots < x_N \le b$, if we replace $u(x)$ bay $u^h(x)$, then the equation (64) becomes :

$$\lambda u^h(x) + \int_a^x k(x,\xi)u^h(\xi)\,\mathrm{d}\xi = g(x),\ x \in [a,b] \tag{65}$$

or

$$\sum_{j=1}^{N}[\lambda\phi_j(x) + \int_a^x k(x,\xi)\phi_j(\xi)\,\mathrm{d}\xi]u_j = g(x),\ x \in [a,b] \tag{66}$$

we convert the interval $[a,x]$ to the fixed interval $[a,b]$ by the linear transformation :

$$\xi(x,\theta) = \frac{x-a}{b-a}\theta + \frac{b-x}{b-a}a, \tag{67}$$

the equation (66) is written:

$$\sum_{j=1}^{N}[\lambda\phi_j(x)+\int_a^b K(x,\xi(x,\theta))\phi_j(\xi(x,\theta))\,\mathrm{d}\theta]u_j = g(x),\ x \in [a,b] \tag{68}$$

with :

$$K(x,\xi(x,\theta) = \frac{x-a}{b-a}k(x,\xi(x,\theta)). \tag{69}$$

If the equation (68) is checked at each x_i we will have:

$$\sum_{j=1}^{N}[\lambda\phi_j(x_i) + \int_a^b K(x,\xi(x_i,\theta))\phi_j(\xi(x_i,\theta)),\mathrm{d}\theta]u_j = g(x_i), \tag{70}$$

Using the gauss quadrature method with the weights θ_k and the coefficients ω_k in $[a,b]$ to approximate the numerical value of the integral in (70), gives :

$$\sum_{j=1}^{N}[\lambda\phi_j(x_i) + \sum_{k=1}^{m_1}k(x,\xi(x_i,\theta_k))\phi_j(\xi(x_i,\theta_k))\omega_k]u_j = g(x_i), \tag{71}$$

We define $F, u\ and\ g$ by:

$$F_{i,j} = \lambda\phi_j(x_i) + \sum_{k=1}^{m_1}k(x,\xi(x_i,\theta_k))\phi_j(\xi(x_i,\theta_k))\omega_k \tag{72}$$

$$u = [u_1, u_2, \ldots, u_N]^T \tag{73}$$

and

$$g = [g_1, g_2, \ldots, g_N]^T \quad with \quad g_i = g(x_i) \tag{74}$$

we obtain the following linear system:

$$Fu = g \tag{75}$$

Solving (75) the value of $u(x)$ at each point $x \in [a,b]$ is approximated by

$$u(x) \approx u^h(x) = \sum_{j=1}^{N}\phi_j(x)\hat{u}_j. \tag{76}$$

B. 2-D Volterra integral equations

We consider the Volterra equation defined by

$$u(x,y) + \int_c^y\int_a^x k(x,y,\xi,\eta)u(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta = g(x,y), (x,y) \in [a,b]\times[c,d]. \tag{77}$$

Using the linear transformations and techniques employed in the 1-D case, we obtain

$$F_{i,j} = \lambda\phi_j(x_i,y_i)+ \sum_{p=1}^{m_1}\sum_{k=1}^{m_1}K(x_i,y_i,\xi(x_i,\theta_k),\eta(y_i,\zeta_p))\phi_j(\xi(x_i,\theta_k),\eta(y_i,\zeta_p))\omega_k\omega_p, \tag{78}$$

with

$$\xi(x,\theta) = \frac{x-a}{b-a}\theta + \frac{b-x}{b-a}a,$$

$$\eta(y,\zeta) = \frac{y-c}{d-c}\zeta + \frac{d-y}{d-c}c,$$

$$K(x_i,y_i,\xi(x_i,\theta_k),\eta(y_i,\zeta_p)) = \frac{x_i-a}{b-a}\frac{y_i-c}{d-c}k(x_i,y_i,\xi(x_i,\theta_k),\eta(y_i,\zeta_p)) \tag{79}$$

u , g are defined by :

$$u = [u_1, u_2, \ldots, u_N]^T$$

$$g = [g_1, g_2, \ldots, g_N]^T$$

Then we obtain the following linear equations system :

$$F\hat{u} = g \tag{80}$$

the solution of (80). gives the approximate value of $u(x, y)$ at each point $(x, y) \in \Omega$ by

$$u(x, y) \approx u^h(x, y) = \sum_{j=1}^{N} \phi_j(x, y) u_j. \tag{81}$$

VII. Numerical examples and results interpretation

In this section, we will implement the proposed method in numerical examples. For this, we propose to use Fredholm equations and Voltera equations in dimension one and two, and we compare the obtained results with the exact solutions. In addition, to justify the effectiveness of our approach, we compare its results with those obtained by the finite elements method.

A. 1-D Fredholm examples

Example 1:
We consider the Fredholm equation defined by :

$$u(x) + \int_0^1 x\xi u(\xi)\, \mathrm{d}\xi = \frac{4}{3}x, \ x \in [0, 1] \tag{82}$$

The exact solution is $u(x) = x$. We use the Gauss quadrature method with $m_1 = 7$ in $[0; 1]$ to approximate the numerical value of the integral in (82).
The table I shows the results for $N = 5$, $d = 0.5$, and using a linear basis.

TABLE I
RESULTS OF EXAMPLE 1 FOR $N = 5$, $d = 0.5$, AND A LINEAR BASIS.

x	u_{ex}	u_{app}	error
0.00	0.00000000	0.00000000	0,00
0.25	0.25000000	0.25000786	7.86×10^{-6}
0.50	0.50000000	0.50001572	1.57×10^{-5}
0.75	0.75000000	0.75002358	2.36×10^{-5}
1.00	1.00000000	1.00003144	3.14×10^{-5}

Variation of the number of nodes:
The table II shows the infinite error of our method for different values of the number of nodes N used in the discretization of the domain:

TABLE II
INFINITE ERROR FOR DIFFERENT VALUES OF N.

N	$d(support)$	error $\| e \|_\infty$
5	0.5	3.14×10^{-5}
9	$\frac{0.5}{2}$	3.93×10^{-6}
17	$\frac{0.5}{4}$	$1,96 \times 10^{-6}$
33	$\frac{0.5}{8}$	9.82×10^{-7}
66	$\frac{0.5}{16}$	4.91×10^{-7}

To show the effectiveness of our approach, we evaluate the infinite error obtained using the finite elements method, and we compare it with our results. The Fig.4 shows the results for different values of N.

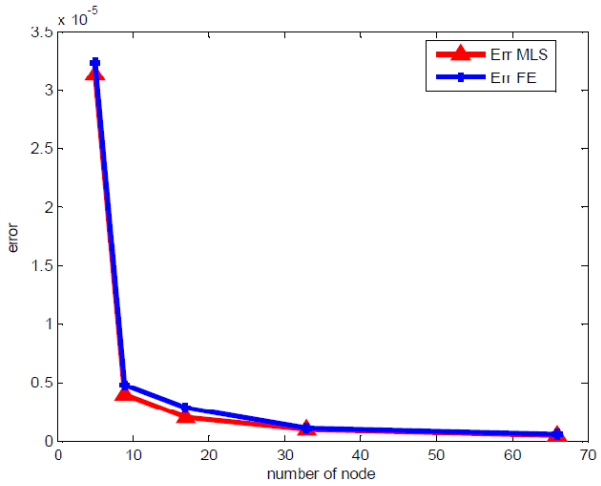

Fig. 4. Comparison of MLS and FE errors.

Example 2:
In this example we consider an equation with a symmetric domain.

$$u(x) + \int_{-1}^{1} x\xi u(\xi)\, \mathrm{d}\xi = \frac{5}{3}x, \quad x \in [-1, 1] \tag{83}$$

The exact solution is $u(x) = x$.
We use the Gauss quadrature method with $m_1 = 7$ in $[-1, 1]$ to approximate the numerical value of the integral in (83).
The tables III and IV show the results for $N = 5$ and $d = 0.51$, and when N is varying :

TABLE III
RESULTS FOR $N = 5$ AND $d = 0.51$, WITH A SYMMETRIC DOMAIN.

x	u_{ex}	u_{app}	error
-1.00	-1.00000000	-1.00525088	5.2×10^{-3}
0.50	-0.50000000	-0.50762544	7.6×10^{-3}
0.00	0.00000000	0.00000000	0.00
0.50	0.50000000	0.50762544	7.6×10^{-3}
1.00	1.00000000	1.07525088	5.2×10^{-3}

TABLE IV
VARIATION OF THE NUMBER OF NODES N WITH A SYMMETRIC DOMAIN.

N	$d(support)$	error $\| e \|_\infty$
5	0.51	5.2×10^{-3}
9	$\frac{0.51}{2}$	1.8×10^{-4}
17	$\frac{0.51}{4}$	9.40×10^{-4}
33	$\frac{0.51}{8}$	4.70×10^{-4}
66	$\frac{0.5}{16}$	1.45×10^{-4}

Example 3:
We consider the following equation :

$$u(x) + \frac{1}{3}\int_0^1 \exp(2x - \frac{5\xi}{3}) u(\xi)\, \mathrm{d}\xi = \exp(2x + \frac{1}{3}), \quad x \in [0, 1] \tag{84}$$

The exact solution is $u(x) = \exp(2x)$.
The table V shows the results for $N = 10$ and $d = 0.25$:

B. 1-D Voltera example

We consider the following equation :

$$u(x) - \frac{1}{8}\int_0^x x \times \xi u(\xi)\, \mathrm{d}\xi = 4x - \frac{1}{6}x^4, \quad x \in [0, 1] \tag{85}$$

TABLE V
RESULTS OF EXAMPLE 3 FOR $N = 10$ AND $d = 0.25$.

x	u_{ex}	u_{app}	error
0.00000000	1.00000000	1.00003463	3.46×10^{-5}
0.11111111	1.24884886	1.24889200	4.31×10^{-5}
0.22222222	1.55962349	1.55967736	5.38×10^{-5}
0.33333333	1.94773404	1.94780131	6.72×10^{-5}
0.44444444	2.43242545	2.43250947	8.40×10^{-5}
0.55555555	3.03773177	3.03783670	1.04×10^{-4}
0.66666666	3.79366789	3.79379893	1.31×10^{-4}
0.77777777	4.73771785	4.73788150	1.63×10^{-4}
0.88888888	5.91669359	5.91689795	2.04×10^{-4}
1.00000000	7.38905609	7.38931204	2.55×10^{-4}

The exact solution is $u(x) = 4x$.
The numerical results are presented in the table VI below for $N = 10$ and $d = 0.25$:

TABLE VI
RESULTS OF 1-D VOLTERA EXAMPLE, $N = 10$ AND $d = 0.25$.

x	u_{ex}	u_{app}	error
0.00000000	0.00000000	0.00000000	0,00
0.11111111	0.444444444	0.44444444	0.00
0.22222222	0.88888888	0.88888887	1.00×10^{-8}
0.33333333	1.33333332	1.33333314	1.80×10^{-7}
0.44444444	1.77777776	1.77777711	6.66×10^{-7}
0.55555555	2.22222222	2.22222040	1.82×10^{-6}
0.66666666	2.66666664	2.66666290	3.74×10^{-6}
0.77777777	3.11111108	3.11110363	7.45×10^{-6}
0.88888888	3.55555552	3.55554323	1.22×10^{-5}
1.00000000	4.00000000	3.99997857	2.14×10^{-5}

C. 2-D Fredholm example

We consider the Fredholm equation defined by:

$$u(x,y) + \int_0^1 \int_0^1 xy\xi\eta u(\xi,\eta)\,\mathrm{d}\xi\mathrm{d}\eta = \frac{10}{9}xy, \qquad 0 \le x, y \le 1 \tag{86}$$

The exact solution is $u(x, y) = xy$.
The table VII shows the results for $N = 3\times3$ and $d = 0.101$

TABLE VII
RESULTS OF 2-D FREDHOLM EXAMPLE, $N = 3 \times 3$ AND $d = 0.101$.

$(x; y)$	u_{ex}	u_{app}	error
(0;0)	0.00000000	0.00000000	0,00
(0.5;0)	0.00000000	0.00000771	7.71×10^{-6}
(1;0)	0.00000000	0.00001543	1.54×10^{-5}
(0;0.5)	0.00000000	0.00000000	0.00
(0.5;0.5)	0.25000000	0.25969089	9.69×10^{-3}
(1;0.5)	0.50000000	0.51938179	1.94×10^{-2}
(0;1)	0.00000000	0.00000000	0.00
(0.5;1)	0.5000000	0.51938950	1.94×10^{-2}
(1;1)	1.0000000	1.03877901	3.88×10^{-2}

Variation of the number of nodes:
the table VIII below shows the evaluation of the error when N is varing

TABLE VIII
VARIATION OF THE NUMBER OF NODES FOR 2-D FREDHOLM .

N	$d(support)$	error $\parallel e \parallel_\infty$
3×3	0.55	3.88×10^{-2}
5×5	0.25	2.02×10^{-2}
6×6	0.21	$1,10\times10^{-2}$
11×11	0.11	4.59×10^{-3}

D. 2-D Voltera example

We consider the Voltera equation defined by:

$$u(x,y) + \int_0^x \int_0^y xy\xi\eta u(\xi,\eta)\,\mathrm{d}\xi\mathrm{d}\eta = xy + \frac{1}{9}x^4y^4, \qquad 0 \le x, y \le 1 \tag{87}$$

The exact solution is $u(x, y) = xy$.
The results obtained are given in the table IX for $N = 3\times3$ and $d = 0.5021$:

TABLE IX
RESULTS OF 2-D VOLTERA EXAMPLE,$N = 3 \times 3$ AND $d = 0.5021$.

$(x; y)$	u_{ex}	u_{app}	error
(0 ;0)	0.00000000	0.00000000	0,00
(0.5;0)	0.00000000	0.00000007	7×10^{-8}
(1;0)	0.00000000	0.00000014	1.4×10^{-7}
(0;0.5)	0.00000000	0.00000000	0.00
(0.5;0.5)	0.25000000	0.25042044	4.20×10^{-4}
(1;0.5)	0.50000000	0.50655079	6.55×10^{-4}
(0;1)	0.00000000	0.00000000	0.00
(0.5;1)	0.50000000	0.50691734	6.91×10^{-3}
(1;1)	1.00000000	1.11032404	1.10×10^{-1}

Results interpretation:
The numerical examples presented above show that the approximate solutions are in accord with the exact solutions. The error obtained numerically is in accord with the theoretical estimation and is slightly better compared with the results obtained by finite elements method.

VIII. CONCLUSION

In this work the MLS method is presented and applied to solve Fredholm and Volterra integral equations in dimension one and two. The numerical tests presented are in perfect concordance with the exact solutions. The examples discussed show that the numerical error of our approach is comparable to that expected theoretically and is better than that obtained by the finite elements method.This makes it possible to evaluate the effectiveness of the method and to validate our code. This method can be used successfully in a very large number of applications where the use of mesh methods is limited by remeshing problems such as the cases of large deformation problems. One could also use this to solve the mixed integral equations and the class of nonlinear integral and integro-differential equations.

REFERENCES

[1] R. Farengo, Y.C. Lee, P.N. Guzdar, An electromagnetic integral equation: application to microtearing modes, Phys. Fluids 26 (1983) 3515-3523.
[2] P.C. Hansen, T.K. Jensen, Large-scale methods in image deblurring, Lect. Notes. Comput. Sci. 4699 (2007) 24-35.

[3] D. Rajan, S. Chaudhuri, Simultaneous estimation of super-resolved scene and depth map from low resolution defocused observations, IEEE Trans.Pattern Anal. Mach. Intell. 25 (2003) 1102-1117.
[4] A.V. Manzhirov, On a method of solving two-dimensional integral equations of axisymmetric contact problems for bodies with complex rheology, J. Appl. Math. Mech. 49 (1985) 777-782.
[5] J. Radlow, A two-dimensional singular integral equation of diffraction theory, Bull. Am. Math. Soc. (1964) 596-599.
[6] M.V. Mirkin, A.J. Bard, Multidimensional integral equations: a new approach to solving microelectrode diffusion problems, J. Electroad. Chem. 323(1992) 29-51.
[7] Y. Chen, T. Tang, Spectral methods for weakly singular Volterra integral equations with smooth solutions, J. Comput. Appl. Math. 233 (2009) 938-950.
[8] H.H. Sorkun, S. Yalcinbas, Approximate solutions of linear Volterra integral equation systems with variable coefficients, Appl. Math. Modell. 34 (2010) 3451-3464.
[9] J.P. Kauthen, Continuous time collocation method for VolterraFredholm integral equations, Numer. Math. 56 (1989) 409-424.
[10] A.H. Borzabadi, O.S. Fard, A numerical scheme for a class of nonlinear Fredholm integral equations of the second kind, J. Comput. Appl. Math. 232 (2009) 449-454.
[11] T. Belytschko, Y. Krongauz, D. Organ, Meshless methods: an overview and recent developments, Comput. Methods Appl. Mech. Eng. 139 (1996) 3-47.
[12] P. Lancaster, K. Salkauskas, Surface generated by moving least squares methods, Math. Comput. 37 (1981) 141-158.
[13] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, Comput. Methods Appl. Mech. Engrg. 139 (1996) 3-47.
[14] L. Zhang, J. Ouyang, X. Zhang, On a two-level element-free Galerkin method for incompressible fluid flow, Appl. Numer. Math. 59 (2009) 1894-1904.
[15] C.A. Duarte, J.T. Oden, Hp clouds An hp meshless method, Numer. Methods Part. Differ. Equat. 12 (1996) 675-705.
[16] W.K. Liu, S. Li, T. Belytschko, Moving least-square reproducing kernel methods (I) methodology and convergence, Comput. Methods Appl. Mech. Engrg. 143 (1997) 113-154.
[17] S.N. Atluri, S.P. Shen, The meshless local PetrovGalerkin (MLPG) method: A simple less-costly alternative to the finite element and boundary element methods, CMES: Comput. Modeling Eng. Sci. 3 (2002) 11-51.
[18] M. Dehghan, D. Mirzaei, Meshless Local PetrovGalerkin (MLPG) method for the unsteady magnetohydrodynamic (MHD) flow through pipe with arbitrary wall conductivity, Appl. Numer. Math. 59 (2009) 1043-1058.
[19] F. Chen, P.J.Y. Wong, Solutions of Fredholm integral equations via discrete biquintic splines, Mathematical and Computer Modelling 57 (2013) 551-563
[20] M. Asgari, Numerical Solution for Solving a System of Fractional Integro-differential Equations, IAENG International Journal of Applied Mathematics, vol. 45, no. 2, pp 85-91, 2015
[21] F. Hosseini Shekarabi, M. Khodabin and K. Maleknejad, The Petrov-Galerkin Method for Numerical Solution of Stochastic Volterra Integral Equations, IAENG International Journal of Applied Mathematics, vol. 44, no. 4, pp 170-176, 2014
[22] M. Mosleh ; M. Otadi, Least squares approximation method for the solution of HammersteinVolterra delay integral equations, Applied Mathematics and Computation 258 (2015) 105-110
[23] P. Assari ; M. Dehghan, A meshless method for the numerical solution of nonlinear weakly singular integral equations using radial basis functions, The European Physical Journal Plus (2017) 132-199
[24] R. EL JID, MLS method: numerical error and solution of a unilateral contact problem using penalty and Lagrange multiplier, submitted for publication in Advances in Applied Mathematics and Mechanics.
[25] D. Shepard, A two-dimensional interpolation function for irregularly spaced points, Proc. 23rd Nat. Conf. ACM, ACM Press, New York, (1968), 517-524.
[26] P. Lancaster and K. Salkauskas, Surface generated by moving least squares method, Math. Comput. 37 (1981), 141-158.
[27] T. Belytschko ; Lu, Y.Y.; Gu, L.; Tabbara, M. (1995). Element-free Galerkin methods for static and dynamic fracture, Int. J. Solids Struct. 32, pp. 2547-2570