Finite Element Solution to Integral Equations

Dieudonne Ndong Ovono

***Abstract*—Some complex natural phenomena in physics, economics and engineering can be represented by Integral Equations (IE). But one may be short of general mathematical tricks to solve them analytically, which could quickly turn into a life time enterprise. In the contrary, the use of computers or numerical methods introduced some decades ago has brought a different approach that is innovative and effective.**

In addition, the theory of measure, the study of polynomials and normed spaces, namely Banach, Hilbert, Lebesgue, Hölder, Lipschitz and Sobolev spaces led to advanced modern numerical methods to solve these complex IEs. When using computer, a generally continuous domain is divided into smaller pieces, or elements, so the calculation on any point of the continuous domain could be done using results from the elements; it is similar to approximate a circle by a regular polygon; as the number of side increases, the polygon become more and more close to a circle.

This method of solving IEs using subdivisions of the domain, along with discrete form of the equations, is referred to as the Finite Element Method (FEM). In this paper, we implement the advanced core theory of the finite element method into adaptive meshes for generic complex problems represented by IEs.

***Index Terms*—Finite Element, Integral Equation, Interpolation Operator, Adaptive Mesh.**

I. INTRODUCTION

THIS paper covers the implementation of the FEM that we have developed from scratch, which helped us fully extend our understanding of the topic and program the actual computer codes in solving IEs and be able to compare various approximation methods. This paper is organized into three main parts.

The first part summarizes the theoretical foundation of the FEM as well as an overview of the Finite Element interpolation technique. It restates the results on normed vector spaces ([1], p.463-492; [3], p.13-17; [4]), Lebesgue integration, distributional derivatives and Sobolev spaces ([1], p.463-492; [3], p.13-17), essential to derive "error estimate", convergence and well posedness.

Next, we review Integral Equations ([2], p.222-337; [3], p.31-42) and their approximation methods and results.

The last part, Implementation, covers practical implementation considerations such as meshing techniques, "a posterior error estimate" for mesh refinement or adaptive mesh ([1], p.337-457; [3], p.68-112), examples illustrating our simulator results.

Manuscript received September 29, 2014; revised Jan 02, 2015.
D.N.Ovono is an oil & gas engineer and researcher, he works for Halliburton, phone: 1-832-706-4036 (e-mail:arthur_ovono@yahoo.com).

II. THEORETICAL FOUNDATIONS OF THE FINITE ELEMENT METHOD

A. Theoretical Framework and Fundamental Analysis

Please refer to ([1], p.463-492; [3], p.13-17) for a complete overview.

Lebesgue Spaces: $L^1(\Omega)$ is the space of the scalar-valued functions that are *Lebesgue-integrable*. The space of locally integrable functions is denoted by $L^1_{loc}(\Omega)$ and is defined as $L^1_{loc}(\Omega) = \{f \in M(\Omega);\ \forall\ compact\ K \subset \Omega, f \in L^1(K)\}$

For $1 \le p \le +\infty$, let $L^p(\Omega) = \{f \in M(\Omega);\ \|f\|_{0,p,\Omega} < +\infty\}$ where

$$\|f\|_{0,p,\Omega} = \left(\int_\Omega |f(x)|^p\right)^{\frac{1}{p}} dx, for\ 1 \le p \le +\infty$$

$$\|f\|_{0,\infty,\Omega} = ess\ \sup|f(x)| = \inf\{M \ge 0; |f(x)| \le M, f \in \Omega$$

Sobolev Spaces: Let s and p be two integers with $s \ge 0$ and $1 \le p \le +\infty$, the *Sobolev space* is defined as

$$W^{s,p}(\Omega) = \{u \in D'(\Omega);\ \partial^\alpha u \in L^p(\Omega), |\alpha| \le s\}$$

with distributional derivatives ([1], p.463-492). There follows that $W^{s,p}(\Omega)$ is a Banach space when equipped with the norm

$$\|u\|_{W^{s,p}(\Omega)} = \sum_{|\alpha| \le s} \|\partial^\alpha u\|_{L^p(\Omega)}$$

For $p = 2$, $W^{s,2}(\Omega)$ has an Hilbert structure and is denoted by $H^s(\Omega)$.

B. Overview of the Finite Element Interpolation

Please refer to ([1], p.3-58; [3], p.18-25) for a complete literature.

Polynomials Used for One-Dimensional Interpolation:

The mesh: a *mesh* of $\Omega =]a, b[$ is an indexed collection of intervals with non-zero measure $\{I_i = [x_i, x_{i+1}]\}_{0 \le i \le N}$ forming a partition of Ω, ie, $\overline{\Omega} = \bigcup_{i=0}^{N} I_i$ and $I_i \cap I_j \ne \emptyset$ for $i \ne j$. It is constructed by taking $N + 2$ points of $\overline{\Omega}$ such that $a = x_0 < x_1 < \cdots < x_N < x_{N+1} = b$. The mesh is denoted by $T_h = \{I_i\}_{0 \le i \le N}$, where $h_i = x_{i+1} - x_i$ may be a variable step size and the subscript $h = \max_{0 \le x \le N} h_i$ refers to a refinement. The points in the set $\{x_0, \dots, x_{N+1}\}$ are called the *vertices* of the mesh and the intervals I_i are referred to as *elements* or *cells*.

The $\mathbb{P}_k$ Lagrange finite element: Let $k \ge 1$ and let $\{s_0, \dots, s_k\}$ be $k + 1$ distinct numbers. The *Lagrange polynomials* $\{\mathcal{L}_0^k, \dots, \mathcal{L}_k^k\}$ associated with the nodes $\{s_0, \dots, s_k\}$ are defined to be $\mathcal{L}_m^k(t) = \frac{\prod_{l \ne m}(t - s_l)}{\prod_{l \ne m}(s_m - s_l)}$, $0 \le m \le k$ and satisfy the important propriety $\mathcal{L}_m^k(s_l) = \delta_{ml}$, $0 \le m, l \le k$. For $j \in \{0, \dots, k(N+1)\}$ with $j = ki + m$ and $0 \le m \le k - 1$, define the functions

$$\varphi_{ki+m}(x) = \begin{cases} \mathcal{L}_{i,m}^k(x) & if\ x \in I_i \\ 0 & otherwise \end{cases}$$

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and for $m = 0$ $\varphi_{ki}(x) = \begin{cases} \mathcal{L}^k_{i-1,k}(x) & if\ x \in I_{i-1} \\ \mathcal{L}^k_{i,0}(x) & if\ x \in I_i \\ 0 & therwise \end{cases}$

On $I_i = [x_i, x_{i+1}] \in \mathrm{T}_h$ let choose the local degrees of freedom to be the (k+1) linear forms $\Sigma_i = \{\sigma_{i,0}, \dots, \sigma_{i,k}\}$ defined by $\sigma_{i,m}: \mathbb{P}_k \ni p \mapsto \sigma_{i,m}(p) = p(\xi_{i,m}),\ 0 \le m \le k$ for all $p \in \mathbb{P}_k$. Σ_i is a basis of $\mathcal{L}(\mathbb{P}_k, \mathbb{R})$ and the triplet $\{I_i, \mathbb{P}_k, \Sigma_i\}$ is called a one-dimensional $\mathbb{P}_k$ *Lagrange finite element*. the nodes are $\{\xi_{i,0}, \dots, \xi_{i,k}\}$; the local shape functions at nodes are the (k+1) Lagrange polynomials $\mathcal{L}^k_{i,m}$, $0 \le m \le k$.

Types of Finite Elements: Definitions and Examples

Main definition: following Ciarlet, a finite element is defined as a triplet $\{K, P, \Sigma\}$ where:

(i) K is a *compact*, *connected*, *Lipschitz* subset of $\mathbb{R}^d$with non-empty *interior*.

(ii) P is a vector space of functions $p: K \mapsto \mathbb{R}^m$ for some positive integer m, typically $m = 1$ or d.

(iii) Σ is a set of n_{sh} linear forms $= \{\sigma_1, \dots, \sigma_{n_{sh}}\}$, called *local degrees of freedom*, acting on the elements of P, and such that the linear mapping $P \ni p \mapsto \big(\sigma_1(p), \dots, \sigma_{n_{sh}}(p)\big), \sigma_{i,m}(p) \in \mathbb{R}^{n_{sh}}$is bijective, i.e., $\Sigma$is a basis for $\mathcal{L}(P; \mathbb{R})$.

Then, there exists a basis $\{\theta_1, \dots, \theta_{n_{sh}}\}$ in P, called the *local shape functions*, such that $\sigma_i(\theta_j) = p(\theta_j), 1 \le i, j \le n_{sh}$; $\dim P = card\ \Sigma = n_{sh}$ and $\forall p \in P, (\sigma_i(p) = 0, 1 \le i \le n_{sh}) \Rightarrow (p = 0)$, this propriety is known as *unisolvence*.

Lagrange finite element: For all $p \in P, \sigma_i(p) = p(a_i), 1 \le i \le n_{sh}$, $\{K, P, \Sigma\}$ is called a Lagrange finite element. The local shape functions $\{\theta_1, \dots, \theta_{n_{sh}}\}$, with $\theta_i(a_j) = \delta_{ij}, 1 \le i, j \le n_{sh}$, are called the *nodal basis* of P.

Simplicial Lagrange finite element: Let $\{a_0, \dots, a_d\}$ be a family of points in $\mathbb{R}^d$, $d \ge 1$. Assume that the vector $\{a_1 - a_0, \dots, a_d - a_0\}$ are linearly independent. Then, the convex hull of $\{a_0, \dots, a_d\}$ is called a *simplex*, and the points $\{a_0, \dots, a_d\}$ are called the *vertices* of the *simplex*. Let $\mathbb{P}_k$ be the space of polynomials in the variables $x_1, \dots, x_d$, with real coefficient and global degree at most k:

$$\mathbb{P}_k = \left\{ p(x) = \sum_{\substack{0 \le i_1, \dots, i_d \le k \\ i_1 + \dots + i_d \le k}} \alpha_{i_1 \dots i_d} x_1^{i_1} \dots x_d^{i_d};\ \alpha_{i_1 \dots i_d} \in \mathbb{R} \right\},$$

$\mathbb{P}_k$ is a vector space of $\dim \mathbb{P}_k = \binom{d+k}{k}$

There is also "Tensor Product" and "Prismatic" finite elements where the shape of cell is a cuboid, respectively a prism.

The Crouzeix-Raviart finite element:

$P = \mathbb{P}_1$ and take for the local degrees of freedom the *mean-value* over the $(d+1)$ faces of K, i.e., for $0 \le i \le d$, $\sigma_i(p) = \frac{1}{meas(F_i)} \int_{F_i} p$. Let $\Sigma = \{\sigma_i\}_{0 \le i \le d}$, then $\{K, \mathbb{P}_1, \Sigma\}$ is a finite element. The local shape functions are $\theta_i(x) = d(\frac{1}{d} - \lambda_i(x))$, $0 \le i \le d$. The local Crouzeix-Raviart interpolation operator is then defined as follows $\mathfrak{T}_k^{CR}: V(K) \ni v \mapsto \mathfrak{T}_k^{CR} v = \sum_{i=0}^d \left(\frac{1}{meas(F_i)} \int_{F_i} v \right) \theta_i \in \mathbb{P}_1$

Other finite element type are Raviart-Thomas, Nedelec, Hermite, etc.

Meshes: Basic Concepts

Mesh: Let Ω be a domain in $\mathbb{R}^d$. A mesh is a union of a finite number N_{el} of compact, connected, Lipschitz sets K_m, called mesh cells or elements, with non-empty interior such that $\{K_m\}_{1 \le m \le N_{el}}$ form a partition of Ω, i.e., $\bar{\Omega} = \bigcup_{m=1}^{N_{el}} K_m$ and $K_m \cap K_n \ne \emptyset$ for $m \ne n$. A mesh $\{K_m\}_{1 \le m \le N_{el}}$ is denoted by $\mathfrak{T}_h$, h referring to the refinement level so that $\forall K \in \mathfrak{T}_h,\ h_{\mathrm{K}} = diam(K) = \max_{x_1, x_2 \in K} \|x_1 - x_2\|_d$, where $\|.\|_d$ is the Euclidian norm in $\mathbb{R}^d$ and $h = \max_{K \in \mathfrak{T}_h} h_{\mathrm{K}}$

A sequence of successive refinement meshes is denoted by $\{\mathfrak{T}_h\}_{h>0}$.

Mesh generation: In practice, the mesh is generated from a *reference* cell $\hat{K}$, and a set of geometric transformations mapping $\hat{K}$ to the actual mesh cells $K \in \mathfrak{T}_h$, denote by $T_h: \hat{K} \to K$. We shall henceforth assume that the geometric transformations are C^1*-diffeomorphisms*. Usually $T_h$is specified using the Lagrange finite element $\{\hat{K}, \hat{P}_{geo}, \hat{\Sigma}_{geo}\}$, called the *geometric reference finite element*. Let $n_{geo} = card(\hat{\Sigma}_{geo})$, $\{\hat{g}_1, \dots, \hat{g}_{n_{geo}}\}$ referred to as the *geometric reference nodes* of $\hat{K}$ associated with $\hat{\Sigma}_{geo}$ and $\{\hat{\psi}_1, \dots, \hat{\psi}_{n_{geo}}\}$ are the *geometric reference shape functions*.

When $\hat{K}$ is a simplex, $\mathfrak{T}_h$ is called a *simplicial mesh*. A mesh generator usually provides a list of n_{geo}-uplets $\{\hat{g}_1^m, \dots, \hat{g}_{n_{geo}}^m\}_{1 \le m \le N_{el}}$, called *geometric nodes of the* m^{th} *element*, where $g_i^m \in \mathbb{R}^d$ and N_{el} is the number of mesh elements. The *geometric transformations* are $T_m: \hat{K} \ni \hat{x} \mapsto T_m(\hat{x}) = \sum_{i=1}^{n_{geo}} g_i^m\, \hat{\psi}_i(\hat{x}) \in \mathbb{R}^d$ so that $T_m(\hat{g}_i) = g_i^m$ for $1 \le i \le n_{geo}$ and $K_m = T_m(\hat{K})$. The numbering of the nodes had to be compatible, henceforth the convention that the Jacobian of T_m be positive so that T_m is a C^1*-diffeomorphisms*. When the transformation is affine, the mesh is called an *affine mesh* and if $\hat{K}$ is a simplex, the affine mesh is called a *triangulation*. For domain with a curved boundary, the mesh is generated using geometric transformation of degree $k_{geo} > 1$.

Approximation Spaces and Interpolation Operators

Global interpolation operator: A global interpolation operator can be constructed by first choosing its domain to be

$D(\mathfrak{T}_h) = \{v \in [L^1(\Omega_h)]^m; \forall K \in \mathfrak{T}_h, v_{|K} \in V(K)\}$, then the global interpolation operator $\mathfrak{T}_h v$ is defined elementwise, using the local interpolation operator, by $\mathfrak{T}_h: D(\mathfrak{T}_h) \ni v \mapsto \sum_{K \in \mathfrak{T}_h} \sum_{i=1}^{n_{sh}} \sigma_{K,i}(v_{|K})\, \theta_{K,i} \in W_h$, where the so called *approximation space* $W_h = \{v_h \in [L^1(\Omega_h)]^m; \forall K \in \mathfrak{T}_h, v_{|K} \in P_K\}$ is the codomain of $\mathfrak{T}_h$. If $W_h \subset V$, where V is a Banach space, $W_h$is said to be *V-conformal*.

III. APPROXIMATION OF INTEGRAL EQUATIONS

Please, refer to ([2], p.222-337) for an in depth overview on integral equations or ([3], p.31-42) for summary.

A. *Introduction*

An *integral equation* is an equation in which a function to be determined appears under an integral sign. It is said to be *linear* when no nonlinear functions of the unknown function

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

are involved. The most frequent form is the *Fredholm equation*:

$$\alpha(x)y(x) = F(x) + \lambda\int_a^b K(x,\varepsilon)y(\varepsilon)d\varepsilon$$

If $b = x$ is identified with the current variable, the equation is known as the *Volterra equation*. λ, a *and* b are constants and α, F and $K$are given functions, continuous on (a,b) as well as $y(x)$. $K(x,\varepsilon)$ is known as the *kernel*. The integral equation is said to be of the *first kind* if $\alpha \equiv 0$, *second kind* if $\alpha \equiv 1$ and of the *third kind* if α is a function. If α is positive, the Fredholm equation can take the form

$$\sqrt{\alpha(x)}y(x) = \frac{F(x)}{\sqrt{\alpha(x)}} + \lambda\int_a^b \frac{K(x,\varepsilon)}{\sqrt{\alpha(x)\alpha(\varepsilon)}}\sqrt{\alpha(\varepsilon)}y(\varepsilon)d\varepsilon$$

and be considered of the second kind in the unknown function $\sqrt{\alpha(x)}y(x)$ with a modified kernel. For a two-dimensional variable $w(x,y)$, the Fredholm equation is of the form

$$\alpha(x,y)w(x,y) = F(x,y) + \lambda\iint_{\mathcal{H}} K(x,y;\varepsilon,\eta)w(\varepsilon,\eta)d\varepsilon d\eta$$

B. *Numerical Solution of Integral Equations*

Approximation of Fredholm Equations by Sets of Algebraic Equations:

A definite integral of the form $J = \int_a^b f(\varepsilon)d\varepsilon$ can be defined as a limit of the form $J = \lim_{n\to\infty}\sum_{k=1}^n f(x_k)\,(\Delta x)_k$where the interval (a,b) is divided in subintervals of length $(\Delta x)_k$ and x_k is the point of the k^{th} interval. The integral "J" can be obtained by not proceeding to the limit but using a weighting coefficient D_k at x_k so that $J = \sum_{k=1}^n D_k f(x_k)$. D_k can be $(\Delta x)_k$ or in accordance with some rule such as the *trapezoidal rule* $\{D_1, D_2, \dots, D_{n-1}, D_n\} = h\{\frac{1}{2}, 1,1, \dots, 1,1, \frac{1}{2}\}$ or the *Simpson's rule* $\{D_1, D_2, D_3, D_4 \dots, D_{n-3}, D_{n-2}, D_{n-1}, D_n\} = \frac{h}{3}\{1,4,2,4 \dots, 4,2,4,1\}$ where $h = (b-a)/(n-1)$. There follows that the solution of

$y(x) = F(x) + \lambda\int_a^b K(x,\varepsilon)y(\varepsilon)d\varepsilon$ is approximated by $y(x_i) = F(x_i) + \lambda\sum_{k=1}^n D_k K(x_i,x_k)y(x_k)$ or $y_i = f_i + \lambda\sum_{k=1}^n K_{ik}D_k y_k$. If $y = \{y_i\}$, $K = [K_{ij}]$, $D = [D_i\delta_{ij}]$ and $f = \{f_i\}$, thus $(I - \lambda KD)y = f$ where I is the unit matrix of order n. This method is particularly useful when the kernel K is available as a table of values of an empirical influence function. In the other hand, the characteristic numbers of $y(x) = \lambda\int_a^b K(x,\varepsilon)y(\varepsilon)d\varepsilon$ are afforded by *reciprocals* of the characteristic numbers of the corresponding matrix formulation $y = \lambda\sum_{k=1}^n K_{ik}D_k y_k = \lambda KDy$.

Approximation Methods of Undetermined Coefficients

In this method the solution of

$y(x) = F(x) + \lambda\int_a^b K(x,\varepsilon)y(\varepsilon)d\varepsilon$ is approximated by

$y(x) \approx \sum_{k=1}^n c_k\phi_k(x)$ leading to the form

$\sum_{k=1}^n c_k[\phi_k(x) - \lambda\Phi_k(x)] \approx F(x)$ where

$\Phi_k(x) = \int_a^b K(x,\varepsilon)\phi_k(\varepsilon)d\varepsilon \quad (a \le x \le b)$ and the $\phi_k$are suitably chosen functions such as x^k and the c's are determined by a set of n algebraic equations. If end values are known in advance, it may be desirable to use

$y(x) \approx \phi_0(x) + \sum_{k=1}^n c_k\phi_k(x)$, with $\phi_0(x)$ verifying the known end conditions and the remaining ϕ's vanishing at end points.

The Method of Collocation

Following the preceding section by setting $s_k(x) = \phi_k(x) - \lambda\Phi_k(x) \equiv \phi_k(x) - \lambda\int_a^b K(x,\varepsilon)\phi_k(\varepsilon)d\varepsilon \quad (a \le x \le b)$ and by requiring that $\sum_{k=1}^n c_k s_k(x) \approx F(x)$ be an equality at n distinct points in (a,b), we obtain the n conditions $\sum_{k=1}^n c_k s_k(x_i) = F(x_i)$ or $Sc = f$ where

$c = \{c_i\}$, $S = [s_{ij}]$ with $s_{ij} = s_i(x_j)$ and $f = \{f_i\}$ *(i=1, 2, …,n)*, leading to the n algebraic equations in c_k, hence $y(x)$.

The Method of Weighting Functions or Galerkin

This method uses the n *orthogonality* conditions of the form $\sum_{k=1}^n c_k\int_a^b \psi_i s_k dx = \int_a^b \psi_i F dx \quad (i = 1,2,\dots,n)$ or $Mc = b$ where $M = [m_{ij}]$ with $m_{ij} = \int_a^b \psi_i s_j dx$ and $b_i = \int_a^b \psi_i F dx \quad (i = 1,2,\dots,n)$; a convenient choice of ψ_i is $1, x, x^2, \dots, x^n$; it is desirable to choose the ψ_i as a *complete* set of functions. The system $Mc = b$ leads to n algebraic equations in c_k, hence $y(x)$.

The Method of Least Square

This method requires that the integral of the square of the difference $\sum_{k=1}^n c_k s_k(x) - F(x)$ be as small as possible or $\int_a^b[\sum_{k=1}^n c_k s_k(x) - F(x)]^2 dx$=minimum and avoids the dependence on the choice of the collocation points or weighting functions. There then follows that $\int_a^b s_i(x)[\sum_{k=1}^n c_k s_k(x) - F(x)]dx = 0$ or $\sum_{k=1}^n c_k\int_a^b s_i s_k dx = \int_a^b s_i F dx \quad (i = 1,2,\dots,n)$ leading to $\sum_{k=1}^n c_k[\sum_{r=1}^N D_r s_i(x_r)s_k(x_r)] = \sum_{r=1}^N D_r s_i(x_r)F(x_r)$ $(i = 1,2,\dots,n)$ or $\sum_{k=1}^n c_k p_{ik} = q_i$, with change in indices, we obtain

with $P = [p_{ij}] = \sum_{r=1}^N D_r s_i(x_r)s_k(x_r) = \sum_{k=1}^N s_{ki}D_k s_{kj} = S^T DS$ and $q = \{q_i\} = \sum_{k=1}^N s_{ki}D_k f_i = S^T Df$ where $S = [s_{ij}] \equiv [s_i(x_j)]$, $D = [D_i\delta_{ij}]$, and $f = \{f_i\}$.

Since D is diagonal, $S^T D = (DS)^T$, $P = (DS)^T S$ and $q = (DS)^T f$, this result leads to the following procedure for determining the n linear equations represented by $\sum_{k=1}^n c_k s_k(x_i) = F(x_i)$ 1) Choose N points in (a,b) and write down the N equations. 2) Denote by S the N *x* n matrix of coefficients in the previous set of equations and form the "weighting matrix" S^*=DS

by multiplying the i[th] row of S by the weighting coefficient D_i associated with the point x_i in an appropriate integration scheme involving the N points. 3) Pre-multiply the *augmented matrix* of $\sum_{k=1}^n c_k s_k(x_i) = F(x_i) \quad (i = 1,2,\dots,N)$ by the *transpose* of the weighting matrix S^*. The resultant matrix is the augmented matrix of the required set of n linear equations which determines the constants $c_1, c_2, \dots, c_n$.

Approximation of the Kernel

It is sometimes convenient to approximate a kernel by a polynomial in x and ε or by a separable kernel and solve the resultant equation according to the method of separable kernel. For instance, a kernel can be approximated by $A_1 + A_2 x + A_3 x^2$ or $x(1-x)(B_1 + B_2 x + B_3 x^2)$ which should verify the end conditions and where the A's and $B's$ are determined as functions of ε by three-point collocation, the use of appropriate weighting functions or the use of least-square techniques.

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C. *Application: Comparison of Collocation, Galerkin, Least Square and Kernel Approximation*

Compare the exact solution of the following integral equation, if it exists, with approximate solutions from the methods of collocation, Galerkin, least square and kernel approximation:

$y(x) = x + \int_0^1 K(x,\varepsilon)y(\varepsilon)d\varepsilon$ where $K(x,\varepsilon) = \begin{cases} x, & \text{when } x < \varepsilon \\ \varepsilon, & \text{when } x > \varepsilon \end{cases}$

a) Exact solution:

By writing $\int_0^1 K(x,\varepsilon)y(\varepsilon)d\varepsilon = \int_0^x K(x,\varepsilon)y(\varepsilon)d\varepsilon + \int_x^1 K(x,\varepsilon)y(\varepsilon)d\varepsilon = \int_0^x \varepsilon y(\varepsilon)d\varepsilon + \int_x^1 xy(\varepsilon)d\varepsilon$ and by using the relation $\frac{d}{dx}\int_{A(x)}^{B(x)} G(x,\varepsilon)d\varepsilon = \int_{A(x)}^{B(x)} \frac{\partial G(x,\varepsilon)}{\partial x}d\varepsilon + G[x,B(x)]\frac{dB(x)}{dx} - G[x,A(x)]\frac{dA(x)}{dx}$ we obtain

$y'(x) = 1 + \int_0^x \frac{\partial[\varepsilon.y(\varepsilon)]}{\partial x}d\varepsilon + x.y(x)\frac{dx}{dx} - 0.y(0)\frac{d0}{dx} + \int_x^1 \frac{\partial[x.y(\varepsilon)]}{\partial x}d\varepsilon + x.y(1)\frac{d1}{dx} - x.y(x)\frac{dx}{dx} = 1 + x.y(x) + \int_x^1 y(\varepsilon)\,d\varepsilon - x.y(x)$

$y'(x) = 1 + \int_x^0 y(\varepsilon)\,d\varepsilon + \int_0^1 y(\varepsilon)\,d\varepsilon = 1 - \int_0^x y(\varepsilon)\,d\varepsilon + Constant \Rightarrow y''(x) = -y(x) \Rightarrow y''(x) + y(x) = 0$ which solution is of the form $y(x) = Asin(x) + Bcos(x)$. By evaluating $y(x)$ and $y'(x)$ at 0 and 1 we get end conditions $y(0) = F(0) = 0\ and\ y'(1) = F'(1) = 1$, we obtain $y(x) = \frac{sin(x)}{cos\,(1)}$.

b) Method of Collocation:

Assuming the approximation $y(x) = c_1 + c_2x + c_3x^2$, we have $\phi_1 = 1, \phi_2 = x\ and\ \phi_2 = x^2$; for $\Phi_k(x) \equiv \int_a^b K(x,\varepsilon)\phi_k(\varepsilon)d\varepsilon$ we obtain

$\Phi_1 = \int_0^x \varepsilon d\varepsilon + \int_x^1 xd\varepsilon = x - \frac{x^2}{2}$; $\Phi_2 = \int_0^x \varepsilon^2 d\varepsilon + \int_x^1 x\varepsilon d\varepsilon = \frac{x}{2} - \frac{x^3}{6}$; $\Phi_3 = \int_0^x \varepsilon^3 d\varepsilon + \int_x^1 x\varepsilon^2 d\varepsilon = \frac{x}{3} - \frac{x^4}{12}$

$s_1(x) = \phi_1(x) - \Phi_1(x) = 1 - x + \frac{x^2}{2}$; $s_2(x) = \phi_2(x) - \Phi_2(x) = \frac{x}{2} + \frac{x^3}{6}$; $s_3(x) = \phi_3(x) - \Phi_3(x) = -\frac{x}{3} + x^2 + \frac{x^4}{12}$

Hence, the approximate equality $\sum_{k=1}^n c_k s_k(x) \approx F(x)$ produces $c_1\left(1 - x + \frac{x^2}{2}\right) + c_2\left(\frac{x}{2} + \frac{x^3}{6}\right) + c_3\left(-\frac{x}{3} + x^2 + \frac{x^4}{12}\right) \approx x$ which evaluation at collocation points 0, 0.5 and 1 leads to the three equations in three unknown c's:

$$\left.\begin{array}{c} c_1 = 0 \\ \frac{5}{8}c_1 + \frac{13}{48}c_2 + \frac{17}{192}c_3 = \frac{1}{2} \\ \frac{1}{2}c_1 + \frac{2}{3}c_2 + \frac{3}{4}c_3 = 1 \end{array}\right\} \Rightarrow \left.\begin{array}{c} c_1 = 0 \\ c_2 = 1.98795 \\ c_3 = -0.43373 \end{array}\right\} \ or\ y(x) \approx 1.98795x - 0.43373x^2$$

c) Method of Weighted Coefficient or Galerkin:

Assuming $y(x) = c_1 + c_2x + c_3x^2$ and by choosing the weighting functions to be $\psi_1 = \phi_1 = 1, \psi_2 = \phi_2 = x$ and $\psi_3 = \phi_3 = x^2$, the Galerkin system of equations $\sum_{k=1}^n c_k \int_a^b \psi_i s_k dx = \int_a^b \psi_i F dx \quad (i = 1,2,3;\ k = 1,2,3)$ becomes:

$$\left.\begin{array}{c} c_1\int_0^1 \left(1 - x + \frac{x^2}{2}\right)dx + c_2\int_0^1 \left(\frac{x}{2} + \frac{x^3}{6}\right)dx \\ +c_3\int_0^1 \left(-\frac{x}{3} + x^2 + \frac{x^4}{12}\right)dx = \int_0^1 xdx \\ c_1\int_0^1 x\left(1 - x + \frac{x^2}{2}\right)dx + c_2\int_0^1 x\left(\frac{x}{2} + \frac{x^3}{6}\right)dx \\ +c_3\int_0^1 x\left(-\frac{x}{3} + x^2 + \frac{x^4}{12}\right)dx = \int_0^1 x^2dx \\ c_1\int_0^1 x^2\left(1 - x + \frac{x^2}{2}\right)dx + c_2\int_0^1 x^2\left(\frac{x}{2} + \frac{x^3}{6}\right)dx \\ +c_3\int_0^1 x^2\left(-\frac{x}{3} + x^2 + \frac{x^4}{12}\right)dx = \int_0^1 x^3dx \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{c} c_1\left[x - \frac{x^2}{2} + \frac{x^3}{6}\right]_0^1 + c_2\left[\frac{x^2}{4} + \frac{x^4}{24}\right]_0^1 + c_3\left[-\frac{x^2}{6} + \frac{x^3}{3} + \frac{x^5}{60}\right]_0^1 = \left[\frac{x^2}{2}\right]_0^1 \\ c_1\left[\frac{x^2}{2} - \frac{x^3}{3} + \frac{x^4}{8}\right]_0^1 + c_2\left[\frac{x^3}{6} + \frac{x^5}{30}\right]_0^1 + c_3\left[-\frac{x^3}{9} + \frac{x^4}{4} + \frac{x^6}{72}\right]_0^1 = \left[\frac{x^3}{3}\right]_0^1 \\ c_1\left[\frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{10}\right]_0^1 + c_2\left[\frac{x^4}{8} + \frac{x^6}{36}\right]_0^1 + c_3\left[-\frac{x^4}{12} + \frac{x^5}{5} + \frac{x^7}{84}\right]_0^1 = \left[\frac{x^4}{4}\right]_0^1 \end{array}\right\}$$

$$\left.\begin{array}{c} \left(1 - \frac{1}{2} + \frac{1}{6}\right)c_1 + \left(\frac{1}{4} + \frac{1}{24}\right)c_2 + \left(-\frac{1}{6} + \frac{1}{3} + \frac{1}{60}\right)c_3 = \frac{1}{2} \\ \left(\frac{1}{2} - \frac{1}{3} + \frac{1}{8}\right)c_1 + \left(\frac{1}{6} + \frac{1}{30}\right)c_2 + \left(-\frac{1}{9} + \frac{1}{4} + \frac{1}{72}\right)c_3 = \frac{1}{3} \\ \left(\frac{1}{3} - \frac{1}{4} + \frac{1}{10}\right)c_1 + \left(\frac{1}{8} + \frac{1}{36}\right)c_2 + \left(-\frac{1}{12} + \frac{1}{5} + \frac{1}{84}\right)c_3 = \frac{1}{4} \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{c} \frac{2}{3}c_1 + \frac{7}{24}c_2 + \frac{11}{60}c_3 = \frac{1}{2} \\ \frac{7}{24}c_1 + \frac{1}{5}c_2 + \frac{11}{72}c_3 = \frac{1}{3} \\ \frac{11}{60}c_1 + \frac{11}{72}c_2 + \frac{9}{70}c_3 = \frac{1}{4} \end{array}\right\} \Rightarrow \left.\begin{array}{c} c_1 = -0.0137 \\ c_2 = +2.0196 \\ c_3 = -0.4358 \end{array}\right\} \Rightarrow y(x) \approx -0.0137 + 2.0196x - 0.4358x^2$$

d) Method of Least Square:

We evaluate $c_1\left(1 - x + \frac{x^2}{2}\right) + c_2\left(\frac{x}{2} + \frac{x^3}{6}\right) + c_3\left(-\frac{x}{3} + x^2 + \frac{x^4}{12}\right) \approx x$ at 0, 0.25, 0.5, 0.75 and 1 (refer to the above point b) for the evaluation at 0, 0.5 and 1) and calculate the weighting matrix S* by:

$$\left.\begin{array}{c} c_1 = 0 \\ c_1\left(1 - \frac{1}{4} + \frac{1}{32}\right) + c_2\left(\frac{1}{8} + \frac{1}{384}\right) + c_3\left(-\frac{1}{12} + \frac{1}{16} + \frac{1}{3072}\right) = \frac{1}{4} \\ \frac{5}{8}c_1 + \frac{13}{48}c_2 + \frac{17}{192}c_3 = \frac{1}{2} \\ c_1\left(1 - \frac{3}{4} + \frac{9}{32}\right) + c_2\left(\frac{3}{8} + \frac{27}{384}\right) + c_3\left(-\frac{1}{4} + \frac{9}{16} + \frac{81}{3072}\right) = \frac{3}{4} \\ \frac{1}{2}c_1 + \frac{2}{3}c_2 + \frac{3}{4}c_3 = 1 \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{c} c_1 = 0 \\ \frac{25}{32}c_1 + \frac{49}{384}c_2 - \frac{63}{3072}c_3 = \frac{1}{4} \\ \frac{5}{8}c_1 + \frac{13}{48}c_2 + \frac{17}{192}c_3 = \frac{1}{2} \\ \frac{17}{32}c_1 + \frac{171}{384}c_2 + \frac{1041}{3072}c_3 = \frac{3}{4} \\ \frac{1}{2}c_1 + \frac{2}{3}c_2 + \frac{3}{4}c_3 = 1 \end{array}\right\} \Rightarrow$$

$$S^* = DS = \begin{bmatrix} \frac{1}{12} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{12} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \frac{25}{32} & \frac{49}{384} & -\frac{63}{3072} \\ \frac{5}{8} & \frac{13}{48} & \frac{17}{192} \\ \frac{17}{32} & \frac{171}{384} & \frac{1041}{3072} \\ \frac{1}{2} & \frac{2}{3} & \frac{3}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{12} & 0 & 0 \\ \frac{25}{96} & \frac{49}{1152} & -\frac{21}{3072} \\ \frac{5}{48} & \frac{13}{288} & \frac{17}{1152} \\ \frac{17}{96} & \frac{57}{384} & \frac{347}{3072} \\ \frac{1}{24} & \frac{1}{18} & \frac{1}{16} \end{bmatrix}$$

where D is the matrix of weighting coefficients according to Simpson's rule at 5 odd points: $D = \frac{h}{3}\{1,4,2,4,1\} = \frac{(1-0/(5-1)}{3}\{1,4,2,4,1\} = \frac{1}{12}\{1,4,2,4,1\} = \{\frac{1}{12}, \frac{1}{3}, \frac{1}{6}, \frac{1}{3}, \frac{1}{12}\}$.

The required three linear equations of the system are found by calculating $S^{*T}S_A$, where S_A is the augmented matrix of the above system. With the use of MS Excel, we obtain:

TABLE 1 - LEAST SQUARE APPROXIMATION

	0.46680	0.16808	0.09514	0.29167	c1=	-0.0123
$S^{*T}S_A$ =	0.16808	0.12079	0.09509	0.20009	c2=	2.0147
	0.09514	0.09509	0.08660	0.15289	c3=	-0.4333

Hence the approximation $y(x) \approx -0.0123 + 2.0147x - 0.4333x^2$.

e) Method of the Approximation of the Kernel:

Now, if we assume the kernel is such that $K(x,\varepsilon) \approx c_1 + c_2x + c_3x^2$. The evaluation of $\int_0^1 K(x,\varepsilon)dx$ gives $\int_0^1 (c_1 + c_2x + c_3x^2)dx \approx c_1 + \frac{c_2}{2} + \frac{c_3}{3}$ and $\int_0^1 K(x,\varepsilon)d\varepsilon = \int_0^x \varepsilon d\varepsilon +$

$\int_x^1 xd\varepsilon = x - \frac{x^2}{2}$. (a) The integral of the kernel must equal its approximation over (0,1), hence $c_1 + \frac{c_2}{2} + \frac{c_3}{3} = x - \frac{x^2}{2}$. Additionally, the kernel approximation is exact at ends points 0 and 1; (b) $K(0,\varepsilon) = x|_{x=0} = 0 = c_1$ and (c) $K(1,\varepsilon) = \varepsilon|_{x=1} = \varepsilon = c_1 + c_2 + c_3$. Conditions (a), (b) and (c) lead to the system

$$\left.\begin{array}{c} c_1 + \frac{c_2}{2} + \frac{c_3}{3} = \varepsilon - \frac{\varepsilon^2}{2} \\ c_1 = 0 \\ c_1 + c_2 + c_3 = \varepsilon \end{array}\right\} \Rightarrow \left.\begin{array}{c} c_1 = 0 \\ 3c_2 + 2c_3 = 3\varepsilon(2-\varepsilon) \\ c_3 = \varepsilon - c_2 \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{c} c_1 = 0 \\ c_2 = \varepsilon(4-3\varepsilon) \\ c_3 = 3\varepsilon(\varepsilon-1) \end{array}\right\} \Rightarrow K(x,,\varepsilon) \approx \varepsilon(4-3\varepsilon)x + 3\varepsilon(\varepsilon-1)x^2$$

Using the method of *separable kernel*, the following steps lead to the approximation:

$y(x) = x + \int_0^1 K(x,,\varepsilon)y(\varepsilon)d\varepsilon$ becomes $y(x) = x + x\int_0^1 [\varepsilon(4-3\varepsilon)]y(\varepsilon)d\varepsilon + x^2\int_0^1 [3\varepsilon(\varepsilon-1)]y(\varepsilon)d\varepsilon$

Let $a_1 = \int_0^1 [\varepsilon(4-3\varepsilon)]y(\varepsilon)d\varepsilon$ and $a_2 = \int_0^1 [3\varepsilon(\varepsilon-1)]y(\varepsilon)d\varepsilon$; we have $y(x) = x + a_1 x + a_2 x^2$ (I)

Multiplying (I) successively by $g_1(x) = x(4-3x)$ and $g_2(x) = 3x(x-1)$ and integrating over (0,1), we obtain the system:

$$\left.\begin{array}{l} \int_0^1 x(4-3x)y(x)dx = a_1 = \int_0^1 x^2(4-3x)dx + a_1\int_0^1 x^2(4-3x)dx + a_2\int_0^1 x^3(4-3x)dx \\ \int_0^1 3x(x-1)y(x)dx = a_2 = \int_0^1 3x^2(x-1)dx + a_1\int_0^1 3x^2(x-1)dx + a_2\int_0^1 3x^3(x-1)dx \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{l} a_1 = \frac{7}{12} + \frac{7}{12}a_1 + \frac{2}{5}a_2 \\ a_2 = -\frac{1}{4} - \frac{1}{4}a_1 - \frac{3}{20}a_2 \end{array}\right\} \Rightarrow \left.\begin{array}{l} \frac{5}{12}a_1 - \frac{2}{5}a_2 = \frac{7}{12} \\ \frac{1}{4}a_1 + \frac{23}{20}a_2 = -\frac{1}{4} \end{array}\right\} \Rightarrow$$

$$\left.\begin{array}{l} a_1 = \frac{137}{139} \approx 0.9856 \\ a_2 = -\frac{60}{139} \approx -0.4317 \end{array}\right\}$$

$\Rightarrow y(x) \approx x + 0.9856x - 0.4317x^2 \approx 1.9856x - 0.4317x^2$.

f) Comparison table and graph:

$y_{exact}(x) = \frac{sin(x)}{cos\,(1)}$

$y_{collocation} = 1.98795x - 0.43373x^2$

$y_{Galerkin}(x) \approx -0.0137 + 2.0196x - 0.4358x^2$

$y_{least\ square}(x) \approx -0.0123 + 2.0147x - 0.4333x^2$

$y_{kernel\ approximation}(x) \approx 1.9856x - 0.4317x^2$

IV. IMPLEMENTATION

This section is based on ([3], p.68-112), inspired by ([1], p.337-457).

Mesh Generator:

We have developed our *Delaunay Triangulations (*interior of the circumscribed sphere does not contain any vertex of the triangulations*)* and non-Delaunay options with insertion of triangle centers.

Alternative Mesh Generator

There are many types of triangle centers. We use them to generate the mesh; this is a non-Bowyer-Watson mesh generation. However, it can be shown that a Delaunay mesh is easily reached from a square diagonally split, then by simple successive insertion of the circumcenter. This is our fundamental results for building a mesh.

Modified Bowyer-Watson Algorithm:

The algorithm inserts successive points inside a triangle and split is in finer pieces, refining the mesh to a better representation of the domain. The more points are inserted, the more represented is the domain and the more accurate is the solution. This is similar to approaching a cercle with a polygon to which the number of sides is gradually increased.

Modified Delaunay Algorithm

To generate a mesh, first we construct a background mesh that will be attached to the domain at its interception points with that domain. The elements that overlap the boundary will be replaced by their parts that are interior to the domain when the parts that are exterior will be deleted from the mesh. And, if the part that is interior is not a triangle, it will be triangulated. The mesh can be initially variable in each direction (for instance, for 2D mesh, Ni intervals in x direction and Nj intervals in y direction).

Illustrative Example of Mesh of Complex 2D Polygon Shape

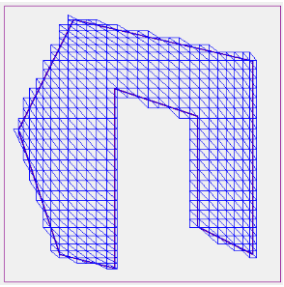

Fig. 1 - Meshing a U Shape Reservoir before Refinement

Quadratures, Assembling and Storage

Assembling

This refers to the phase in the finite element program where the entries of the stiffness matrix and the right-hand side vector are computed.

Storage and Sparse Matrices

Let $\mathcal{A} \in \mathbb{R}^{N,N'}$. Denote by *nnz* the number of non-zero entries in $\mathcal{A}$. The matrix $\mathcal{A}$ is said to be sparse if $nnz \ll N \times N'$. CSR or CSC format or Ellpack-Itpack format methods are used to condition the matrix.

Linear Algebra

We used conditioning and reordering techniques to save memory and accelerate the convergence to the solution of the system. We used LU factorization, LDL^T factorization and Choleski factorization.

Posteriori Error Estimates and Adaptive Meshes

The goal is to assess the error between the exact solution and its finite element approximation in term of known quantities only, i.e., the size of the mesh cells, the problem data and the approximate solution. The *a posteriori estimate* assesses the quality of the approximation solution and provides information to construct a new mesh, a process that can be repeated several times, thereby generating a sequence of so-called *adaptive meshes*.

Adaptive Mesh Generation

We use the local errors indicators to generate a new mesh:

$$|\mathcal{E}| \leq \left(\sum_{K\in\mathfrak{T}_h} \eta_K(u_K,f)^2\right)^{\frac{1}{2}}$$

The quantities $\eta_K(u_K,f)$ are error indicators and can be readily evaluated when using residual–based or hierarchical

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

error estimates. For duality techniques, $\eta_K(u_K, f)$ requires solving a dual problem.

Implementation of Iterative Techniques:
We used "Direct Methods" (*Gaussian Pivot and Gauss-Jordan Pivot* [2], p.1; *Crout Method* [2], p.339*)* and "Indirect Methods" (Jacobi, Gauss-Seidel [5]; Krylov Spaces: CG and Bi-CG Stabilized [1], p.401-413 and [6], p.11-64).

Example of Time-Marching Algorithms (Case of PDEs):
The Implicit Euler is given by the formulation

$$\frac{1}{\Delta t}(u_h^{n+1} - u_h^n, v_h)_L + (Au_h^{n+1}, v_h) = (f^{n+1}, v_h)_L, \forall\, v_h \in V_h$$

Other time-marching algorithms programmed are *Explicit Euler, Leap-Frog and Backward-Differential.* More details could be found in [1], p.279-334.

The Finite Element Calculator (FEC©)
A software was specifically developed to illustrate the application of the method of finite element ([3], p.68-112). It is named the "Finite Element Calculator©" which is self-explanatory. Our objective is to build a library of solutions of real life problems such as heat transfer, reservoir simulation, earthquake, economical predictions, quantum mechanics, etc.

Example 1: Simulated Atm. Pressure along a cockpit:
Using a 15x15 on a [0,1]x[0,1] background mesh, we obtain the following:

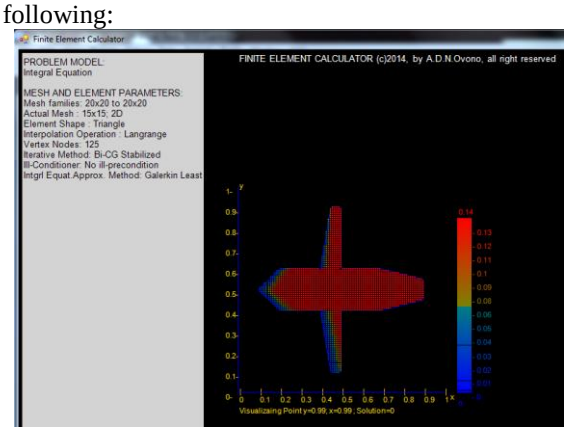

Fig. 2 – Profile of Atm. Depression along the Cockpit

Example 2: Water Saturation Profile in U shape Reservoir:

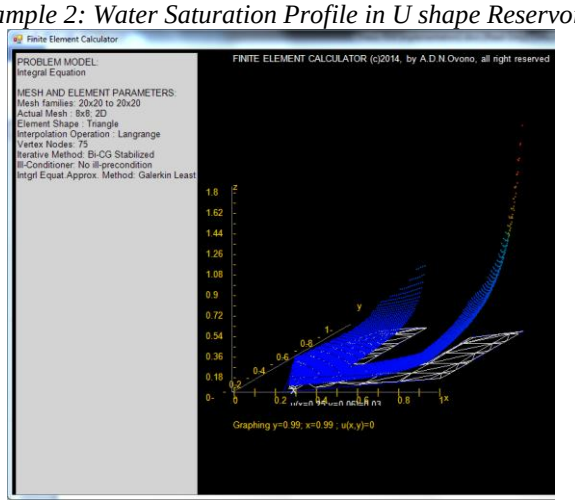

Fig. 3 – U Shape Reservoir Layer – Water Saturation Profile, perspective view

V. CONCLUSION

Complex natural phenomena and engineering applications can often be represented by Integral Equations (IE).

Because it could be challenging to solve some of these equations using conventional and/or analytical methods, we use numerical techniques by mean of computers. The most powerful method of the latter is the so called the Finite Element Method (FEM). The FEM refers not only to the partitioning of a domain in smaller pieces called cells or elements, constituting the mesh, but also the use of mathematical polynomials as shape functions, the discretization of the equations, leading to a system of linear equations in matrix format, more suitable for computer iterative solving techniques.

This paper presented our implementation of the FEM using enhanced adaptive meshes to solve and simulate a broad range of real life complex problems that can be formulated in the forms of Integral Equations.

TABLE 2 - SYMBOLS

Symbol	Description
$L^p(\Omega)$	Functions whose p-th power is Lebesgue integrable
$\|f\|_X =$	Norm of f in the normed space X
$W^{s,p}(\Omega)$	Function whose derivatives up to order s are in $L^p(\Omega)$
$\partial^\alpha u$	$\partial_{x1}^{\alpha 1 \cdots} \partial_{xd}^{\alpha d} u$ is a multi-index
$\mathcal{L}_m^k(s_l)$	Lagrange Polynomial
$\hat{g}_i$	*geometric reference nodes*
$\hat{\psi}_1$	*geometric reference shape functions*
δ_{ml}	*Kronecker symbol*

ACKNOWLEDGEMENT

This paper is submitted as a continuation of my works started during my PhD in Applied Mathematics related to finite element implementation. Therefore, I would like to take this opportunity to express my gratitude to those that have been very supportive and of great help, many of my great colleagues, friends, former teachers, parents and close family, my mother Clementine Avome Ndong and my uncle, Paul Ovono Ndong for their moral encouragements throughout my life and studies.

REFERENCES

[1] Alexandre.Ern, Jean-Luc Guermond, "Theory and Practice of Finite Element", Springer, Vol.159, 2010
[2] Francis B. Hildebrand, "Methods of Applied Mathematics", Prentice Hall, The Dover edition, 1992
[3] Dieudonne N. Ovono, PhD Thesis "Implementation of Advanced Finite Element Solution to Complex Problems: Partial Differential, Mixed Problem and Integral Equations", B.I.U, 2014, p.13-17, p.18-25, p.31-42, p.68-112
[4] C.Antonini, J-F Quint, P.Borgnat, J.Berard, E.Lebeau, E.Souche, A.Chatean, O.Teytaud, "Les Mathématiques pour l'Agrégation (Translation: Mathematics for Aggregation)", Mai 2002. p.154-161, p.473-499.
[5] R.L. Burden & J.D. Faires, "Numerical Analysis - 9th Edition", Brooks/Cole, 2011, Cengage Learning
[6] C.T. Kelly, Iterative Methods for Linear and Nonlinear Equations, North Carolina, States University.