Meshless Local Petrov-Galerkin (MLPG) Methods in Quantum Mechanics

*Williams L. Nicomedes, †Renato C. Mesquita and *Fernando J. S. Moreira
* Federal University of Minas Gerais, Dept. of Electronic Engineering, Belo Horizonte MG, Brazil
† Federal University of Minas Gerais, Dept. of Electrical Engineering, Belo Horizonte MG, Brazil
E-mail: wlnicomedes@yahoo.com.br

Abstract— **In this work, we apply a meshless method to some problems drawn from the context of quantum mechanics. We investigate eigenvalue problems - the time-independent Schrödinger's equation - and boundary value problems as well, these last illustrated by the Nonlinear Schrödinger equation. Meshless methods require absolutely no meshes, unlike the Finite Element Method (FEM), which turns out to be a great advantage when dealing with three-dimensional problems. In a particular class of meshless methods, the MLPG, the shape and test functions belong to different function spaces; the former are constructed via special numerical schemes, whereas there are a number of available options for the latter. The examples have been worked out through MLPG4, also known as Local Boundary Integral Equation Method (LBIE). The accuracy of this method is checked out with results coming from two and three-dimensional analytical problems.**

Index Terms— **Meshless, MLPG, MLS, Schrödinger.**

I. Introduction

Meshless (or meshfree) methods aim to find numerical solutions to differential equations defined in a region *without* relying on a mesh or a grid. These methods share some resemblances with FEM, like the operations with weak forms and the use of compactly supported shape functions, which leads to sparse stiffness matrices. The main difference, as the name indicates, is the complete lack of a mesh. Because there is no mesh, it is pointless to talk about elements, edges and connectivity arrays in this new approach. Meshless methods employ only a cloud of nodes scattered throughout the region of interest.

Meshless methods have successfully been applied in many areas of Computational Mechanics since the 1990's. This new 'numerical technology' is, as said in [1], in its infancy; many challenges still remain to be studied.

One particular meshless method, the MLPG, was devised by S. Atluri within the framework of Mechanics [2], and uses two kinds of functions, *shape functions* and *test functions*, which belong to two different function spaces. The shape functions are constructed numerically through procedures common to other meshless methods, whilst there are many choices available to the test functions. We are particularly interested in MLPG4, whose test function is a solution to Green's problem for Laplace's equation (as addressed in section IV). Our previous work in electrostatics and electromagnetic wave scattering corroborate the applicability of this method [3], [4].

Following the trend in the increasing miniaturization of semiconductor structures, the influence of quantum phenomena becomes seminal to the understanding of how nanodevices work. In this paper, we focus on the application of MLPG to some problems taken from the field of quantum mechanics. Later on, we expect to use these same meshless techniques to deal with situations where quantum mechanics and electromagnetism merge (e.g. self consistent Schrödinger-Poisson problems).

The first problem is concerned to the problem of finding the eigenvalues of the Hamiltonian operator. Given a spatial distribution of potential energy, we apply the MLPG4 to the three-dimensional time-independent Schrödinger equation, which leads to a generalized eigenvalue problem.

The second problem is also related to eigenvalues, but Schrödinger's equation is now stated in a three-dimensional region (unit cell) at which the boundary conditions are Bloch-periodic. This is a problem about the electronic band structure of solids. Through a naïve process, we show how to embed a feature of the problem (periodic boundary conditions) directly into the shape functions (periodic shape functions). As a result, the boundary conditions need not be imposed.

The third example is a two-dimensional boundary value problem, characterized by the Nonlinear Schrödinger equation. We employ a time-difference approximation and a predictor-corrector scheme (to deal with the nonlinearity) in conjunction with MLPG4 in order to find the numerical solutions.

II. Meshless Methods: Nodes & Domains

Let Ω be a region (whose global boundary is $\partial\Omega$) on which a given differential equation is to be solved. We begin by spreading nodes across Ω. The nodal distribution need not be uniform. The next step is to define shape functions associated to each node. These shape functions are compactly supported, i.e., they are different from zero only at a small region surrounding the node, called the node's *influence domain* Λ. This property is directly linked to the sparseness of the final stiffness matrix. So the collection of all shape functions ϕ_n (n runs from 1 to the total number of nodes N) forms a set $\mathcal{S}$ of compactly supported functions whose elements will be used to approximate ψ (the quantity of interest), i.e., given a point $\vec{x}$ where ψ^h shall be calculated (Fig.1) there follows:

$$\psi(\vec{x}) \sim \psi^h(\vec{x}) = \sum_{i=1}^{P} \phi_{c(i)}(\vec{x})\hat{u}_{c(i)} = \mathbf{\Phi}(\vec{x})\hat{\boldsymbol{u}}\,, \qquad \phi_{c(i)} \in \mathcal{S} \;\; \forall c(i), \qquad i = 1,2,\dots,P \qquad (1)$$

where the global index $c(i)$ runs through all P nodes whose influence domains include point $\vec{x}$ (in Fig.1, $c(1) = 3$, $c(2) = 7$, $c(3) = 9$, $c(4) = 17$, $c(5) = 20$) and each $\hat{u}_{c(i)}$ is a coefficient that shall be determined (also called *nodal parameter*). When spreading the nodes, one constraint must be satisfied: the union of the influence domains Λ from all nodes must cover the whole domain Ω, i.e., no *holes* can be left behind, in order to ensure the approximation ψ^h everywhere inside the domain. Overlapping influence domains of neighboring nodes i and j ($\Lambda_i \cap \Lambda_j \neq \emptyset$) is freely allowed.

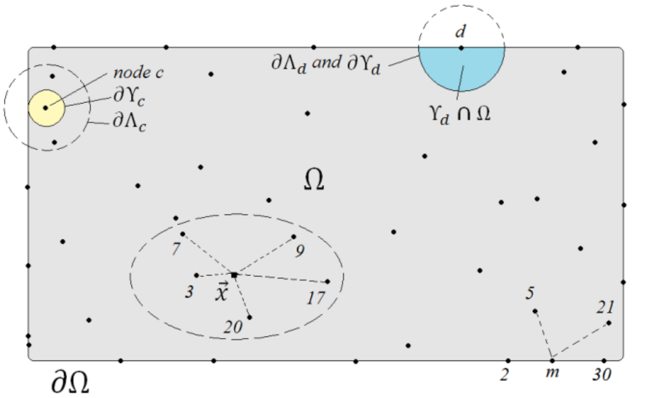

Fig.1. Some facts about a domain Ω and some nodes. *First:* Five nodes acting on point $\vec{x}$. *Second:* A node close to the global boundary, like c, has its test domain Υ_c just touching $\partial\Omega$, whereas its influence domain Λ_c extends over a larger region. *Third*: The shadowed region $\Upsilon_d \cap \Omega$ is the test domain that would be assigned to the boundary node d if a intersection with Ω had to be found out first. *Fourth*: Some nodes influencing a boundary node m.

III. The Shape Functions

In this work, the shape functions have been constructed by the Moving Least Squares (MLS) approximation. MLS is employed in many meshless methods, and the extensive numerical procedures that someone has to go through are omitted here. Details can be found in [1] and in our previous works [3] and [4]. It suffices to say that:

They are compactly supported (they are different from zero only at the node's influence domain Λ);

If one wants to calculate the shape functions at a point $\vec{x} = (x, y, z)$, one first finds out what nodes extend their influence domains over $\vec{x}$ (nodes 3,7,9,17 and 20 in Fig.1). Then one plugs the coordinates of the influencing nodes and those of point $\vec{x}$ in certain matrices. After some calculations, one ends up with a vector $\mathbf{\Phi}(\vec{x})$ whose elements are the values of the shape functions associated to the influencing nodes evaluated at $\vec{x}$. For example, in the case illustrated by Fig.1, one would get a vector of shape functions: $\mathbf{\Phi}(\vec{x}) = [\phi_3(\vec{x}), \phi_7(\vec{x}), \phi_9(\vec{x}), \phi_{17}(\vec{x}), \phi_{20}(\vec{x})]$.

Figure 2 shows a simple 2D MLS shape function ϕ associated to a node. The elements of $\mathcal{S}$ are all similar in form to this one, except that their peaks are located elsewhere, at each node on the domain Ω. From Fig.2 one sees that ϕ is smooth, which is a great advantage when calculating the derivatives of ψ^h. MLS shape functions do not satisfy the Kronecker delta property, i.e., $\phi_i(\vec{x}_j) \neq \delta_{ij}$, but this is not a matter of concern in MLPG4.

IV. The Local Boundary Integral Equation Method

A. *Test domains & Test functions*

We now proceed to lay down the mechanism of MLPG4/LBIE. We begin by spreading nodes across the computational domain Ω. The nodes inside Ω are called *interior nodes*, and those ones located exactly at $\partial\Omega$ are the *boundary nodes*. To each node i (interior and boundary nodes as well), a shape function is associated, whose compact support (the node's influence domain Λ_i) is a circle with radius r_0. In addition to the shape function, other function, called *test function* is associated *to interior nodes only*. This test function v_i acts in a specific region surrounding the node, called the node's *test domain* and represented by Υ_i (Fig. 1, node c). In LBIE, for 2D problems the test domain is required to be a *circle* centered at each interior node i (at $\vec{x}_i$). In 3D otherwise, it is a *sphere* centered at node i. Other requirements on v_i are:

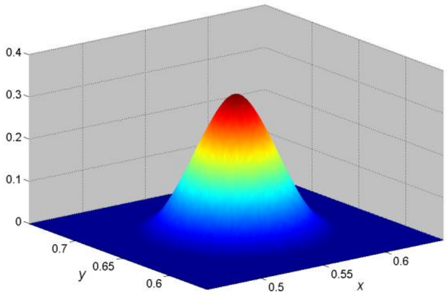

Fig.. 2. A 2D MLS shape function associated to a node at (0.55,0.65).

$$\nabla^2 v_i = -\delta(\vec{x} - \vec{x}_i) \text{ (a Dirac delta at } \vec{x}_i) \qquad (2)$$
$$v_i = 0 \text{ at the test domain boundary } \partial\Upsilon_i.$$

A function v_i centered at node i (an interior node) is therefore given by:

$$v_i(\vec{x}) = \frac{1}{2\pi} \ln\left(\frac{s_i}{\|\vec{x} - \vec{x}_i\|}\right), \qquad \text{for 2D problems} \qquad (3)$$

$$v_i(\vec{x}) = \frac{1}{4\pi}\left(\frac{1}{\|\vec{x} - \vec{x}_i\|} - \frac{1}{s_i}\right), \qquad \text{for 3D problems} \qquad (4)$$

where s_i is the radius of Υ_i. In general, for an interior node i, $r_i \neq s_i$.

The test domains are the regions on which the numerical integrations are carried out. In what regards boundary nodes, if they had been ascribed any sort of test domains, an intersection between the global domain Ω and the circle Υ_i would have to be found in order to carry out the numerical integration. Figure 3 shows this: had a test domain been assigned to node d, than the numerical integration would have to be performed at the shaded region. But trying to find intersections between curves is too cumbersome and hinders the whole process (we actually did it in [3]). This is the main reason why the approach that uses test domains for boundary nodes was dismissed in favor of the more efficient one described in this paper. So boundary nodes have no associated test domains at all.

Concerning interior nodes, in order for them not to have their test domains intersecting the global boundary $\partial\Omega$, one must make sure that if an internal node is close enough to the global boundary, its associated test domain is chosen in such a way that it just touches $\partial\Omega$ (node c in Fig.1).

B. *Imposing boundary conditions: The collocation method*

A simple scheme that does not require whatever kind of numerical integration is a *meshless collocation* scheme, based on the approximation described by (1). Let us suppose that a node m (coordinates $\vec{x}_m = (x_m, y_m, z_m)$) lies at a Dirichlet boundary $\partial\Omega$ whose prescribed condition is a known value $g(\vec{x}_m)$. Then $\psi(\vec{x}_m) = g(\vec{x}_m)$ or:

$$\sum_{i=1}^{Q} \phi_{c(i)}(\vec{x}_m)\hat{u}_{c(i)} = g(\vec{x}_m) \qquad (5)$$

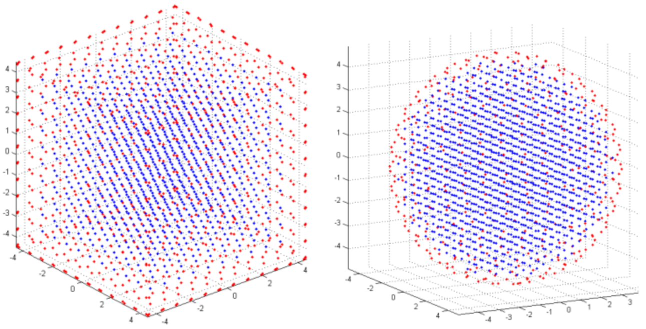

Fig. 3. 1864 nodes in the cubic domain (example 1) and 1868 nodes in the spherical domain (example 2). Each interior node (blue) is ascribed a test domain Υ where the weak forms are integrated. The boundary nodes (red) contribute with extra relations through the collocation procedure.

where the global index $c(i)$ runs through all Q nodes whose influence domains include point $\vec{x}_m$ (in Fig.1, $Q = 5$ and the global indices are $c(1) = 2$, $c(2) = 5$, $c(3) = 21, c(4) = 30$, and $c(5) = m$). Neumann conditions are imposed likewise, by considering a weighted sum of derivatives $\nabla\phi_{c(i)}(\vec{x}_m)\cdot\hat{\boldsymbol{n}}$ instead. This meshless collocation procedure renders the imposition of boundary conditions elegant and fairly simple; neither finding intersections between domains nor performing numerical integrations is necessary.

V. Worked Problems

A. Time-independent Schrödinger equation

Given a function $V(\vec{x})$ that describes the potential energy in a region Ω, we want to solve the time-independent Schröndiger equation (ψ is called wavefunction or probability amplitude):

$$-\frac{\hbar^2}{2m}\nabla^2\psi(\vec{x}) + V(\vec{x})\psi(\vec{x}) = E\psi(\vec{x}) \quad \text{in } \Omega \qquad (6)$$

i.e., we want to find the eigenvalues E of the Hamiltonian operator $\hat{H} = (-\hbar^2/2m)\nabla^2 + V$. Because the reduced Planck's constant ($\hbar$) and the electron mass (m) are quite tiny quantities, we rewrite (6) using Hartree atomic units (a.u.). As in the problems that we are about to solve the wavefunction ψ vanishes at the boundary $\partial\Omega$, the strong form then reads:

$$\begin{cases} -\frac{1}{2}\nabla^2\psi(\vec{x}) + V(\vec{x})\psi(\vec{x}) = E\psi(\vec{x}) & \text{in } \Omega \\ \psi(\vec{x}) = 0 & \text{at } \partial\Omega \end{cases} \qquad (7)$$

One of the ways in which a weak form for (7) can be found is through the weighted residual method. We take each interior node i, multiply the residual of (7) by the test function v_i and integrate over the test domain Υ_i:

$$\iiint_{\Upsilon_i} [\nabla v_i \cdot \nabla\psi + 2Vv_i\psi]\, dV = E\iiint_{\Upsilon_i} 2v_i\psi\, dV \qquad (8)$$

The other way around calls upon Green's second identity for the two functions ψ and v_i. The integrations are carried out at Υ_i, and taking the properties (2) into account, we get:

$$\psi(\vec{x}_i) + \oiint_{\partial\Upsilon_i} \frac{\partial v_i}{\partial n}\psi\, dS + \iiint_{\Upsilon_i} 2Vv_i\psi\, dV = \qquad (9)$$

$$= E\iiint_{\Upsilon_i} 2v_i\psi\, dV$$

where $\psi(\vec{x}_i)$ is the value of ψ evaluated at $\vec{x}_i$, the location of the interior node i. It is due to (9) that the method described in this paper also bears the name of Local Boundary Integral Equation (LBIE) method.

We begin by spreading N_I nodes at the interior of the computational domain Ω and N_B nodes at the boundary $\partial\Omega$, which amounts to a total of $N = N_I + N_B$ degrees of freedom (N unknown nodal parameters $\hat{u}$). We first choose a weak form ((8) or (9)) and impose it at each one of the N_I test domains Υ_i. The wavefunction ψ is expanded in shape functions like (1), and we arrive at a matrix system $A\hat{\boldsymbol{u}} = EB\hat{\boldsymbol{u}}$ where:

$$A_{ij} = \iiint_{\Upsilon_i} [\nabla v_i \cdot \nabla\phi_j + 2Vv_i\phi_j]\, dV \quad B_{ij} = \iiint_{\Upsilon_i} 2v_i\phi_j\, dV \qquad (10)$$

if the weak form (8) is used or

$$A_{ij} = \phi_j(\vec{x}_i) + \oiint_{\partial\Upsilon_i} \frac{\partial v_i}{\partial n}\phi_j\, dS + \iiint_{\Upsilon_i} 2Vv_i\phi_j\, dV$$

$$B_{ij} = \iiint_{\Upsilon_i} 2v_i\phi_j\, dV \qquad (11)$$

if (9) is used. The sparse matrices A and B have N_I rows and $N_I + N_B$ columns (because a given point $\vec{x}$ can be influenced by interior nodes and by those lying on the boundary as well). The collocation procedure (5) enforced at each one of the N_B boundary nodes generates N_B relations among $N_I + N_B$ variables. These relations are substituted back at A and B, and through some eliminations, new square matrices A' and B' ($N_I \times N_I$) are obtained. Finally, we get a generalized eigenvalue problem $A'\hat{\boldsymbol{u}} = EB'\hat{\boldsymbol{u}}$, which is readily solved for the E's.

The first example is the quantum harmonic oscillator. The sides of the cubic domain have been set to 9 a.u., and the potential energy is $V(\vec{x}) = \|\vec{x}\|^2/2$. Homogeneous Dirichlet conditions have been imposed on $\partial\Gamma$. The level n has energy $E_n = E_{n_x+n_y+n_z} = n + 3/2$, where $n_i = 0,1, \ldots$.Multiple values for E_n account for degenerate states. The second example is concerned to those levels that can exist inside a spherical infinite square well, i.e., $V(\vec{x}) = 0$ for $\|\vec{x}\| \le R$ and ∞ otherwise. The radius R of the spherical region has been set to 5 a.u. and homogeneous Dirichlet conditions have also been employed on $\partial\Gamma$. The allowed energy levels (in a.u.) are given by $E = u_{np}^2/2R^2$, where u_{np} are the p-th zeros of the spherical Bessel functions j_n. Figure 3 depicts the nodal distributions for both examples, and tables I and II shows the concordance between numerical and analytical solutions (at the second column, under the header *An.*) More precise results can be obtained either by increasing the number of nodes or by refining the numerical quadratures employed to integrate the weak forms.

B. The Kronig-Penney model

This example comes from solid-state physics, and deals with the calculation of the electronic band structure of solids. The potential energy $V(\vec{x})$ is periodic in the three-dimensional space, i.e., it replicates itself within each region called a *cell*.

TABLE I
FIRST EIGENVALUES FOR THE HARMONIC OSCILLATOR POTENTIAL

n	An.	Numerical – MLPG4/LBIE
0	1.5	1.4962
1	2.5	2.4934; 2.4956; 2.4956
2	3.5	3.4902; 3.4902; 3.4949; 3.4984; 3.5029; 3.5035
3	4.5	4.4856; 4.4911; 4.4927; 4.4958; 4.4958; 4.5031; 4.5031; 4.5151; 4.5257; 4.5257
4	5.5	5.4867; 5.4867; 5.4900; 5.4907; 5.4953; 5.5053; 5.5053; 5.5087; 5.5242; 5.5242; 5.5295; 5.5301; 5.5722; 5.5932; 5.5939

TABLE II
FIRST EIGENVALUES FOR THE SPHERICAL INFINITE SQUARE WELL

n	p	An.	Numerical – MLPG4/LBIE
0	0	0.1974	0.1966
1	0	0.4038	0.4018; 0.4023; 0.4025
2	0	0.6643	0.6606; 0.6607; 0.6614; 0.6623; 0.6632
0	1	0.7896	0.7864
3	0	0.9766	0.9687; 0.9725; 0.9728; 0.9730; 0.9730; 0.9733; 0.9747
1	1	1.1936	1.1878; 1.1890; 1.1895
4	0	1.3391	1.3294; 1.3301; 1.3314; 1.3333; 1.3349 1.3350; 1.3355; 1.3355; 1.3379

For the purposes of analysis, if this array of cells is taken to be infinite, the problems need to be solved only for a unique cell. The strong form is then imposed at a cell Ω as:

$$\begin{cases} -\nabla^2\psi(\vec{x}) + V(\vec{x})\psi(\vec{x}) = E\psi(\vec{x}) & \text{in } \Omega \\ \psi\left(\vec{x}+\vec{L}\right) = e^{j\vec{K}\cdot\vec{L}}\psi(\vec{x}) & \text{in } \partial\Omega \\ \dfrac{\partial\psi\left(\vec{x}+\vec{L}\right)}{\partial n} = e^{j\vec{K}\cdot\vec{L}}\dfrac{\partial\psi(\vec{x})}{\partial n} & \text{in } \partial\Omega \end{cases} \tag{12}$$

In (12), Schrödinger's equation (6) has been written using Rydberg atomic units. $\vec{K}$ is a vector called the Bloch vector, and $\vec{L}$ is the lattice vector. For a cubic cell, the boundary conditions expressed in (12) mean that ψ at a face is equal to ψ at the opposite face multiplied by an exponential term. If we invoke Bloch theorem:

$$\psi(\vec{x}) = e^{j\vec{K}\cdot\vec{x}}u(\vec{x}) \tag{13}$$

where $u(\vec{x})$ is a periodic function over a cell, we get a new strong form on $u(\vec{x})$:

$$\begin{cases} -\nabla^2 u(\vec{x}) - 2j\vec{K}\cdot\nabla u(\vec{x}) + \left(V(\vec{x}) + \left\|\vec{K}\right\|^2\right)u(\vec{x}) & \\ \qquad\qquad = Eu(\vec{x}) & in\ \Omega \\ u\left(\vec{x}+\vec{L}\right) = u(\vec{x}) & in\ \partial\Omega \\ \dfrac{\partial u\left(\vec{x}+\vec{L}\right)}{\partial n} = \dfrac{\partial u(\vec{x})}{\partial n} & in\ \partial\Omega \end{cases} \tag{14}$$

The boundary conditions stated in (14) mean that for a cubic cell, $u(\vec{x})$ at a face equals $u(\vec{x})$ at the opposite face. As the function u will be expanded in shape functions like (1), it is interesting if this periodicity were transferred to the ϕ's: boundary conditions would be unnecessary.

We have come across a way to do this, described in [5], who solves this very same problem using a Galerkin formulation (instead of a Petrov-Galerkin one). But we also have found a naïve approach of getting periodic shape functions, based on operations regarding the indices of the nodes. In this naïve approach, the MLS procedure is left untouched. Just to remember, in the MLS approximation, if one wants to calculate the shape functions at a point $\vec{x}$, what one has to do amounts to finding out which nodes influence $\vec{x}$ (nodes 3,7,9,17,20 in Fig.1) and to plugging their coordinates (together with those of $\vec{x}$) in certain matrices. After some calculations, one ends up with a vector $\mathbf{\Phi}$ whose elements are the shape functions associated to the influencing nodes evaluated at the desired point $\vec{x}$ ($[\phi_3(\vec{x}), \dots, \phi_{20}(\vec{x})]$).

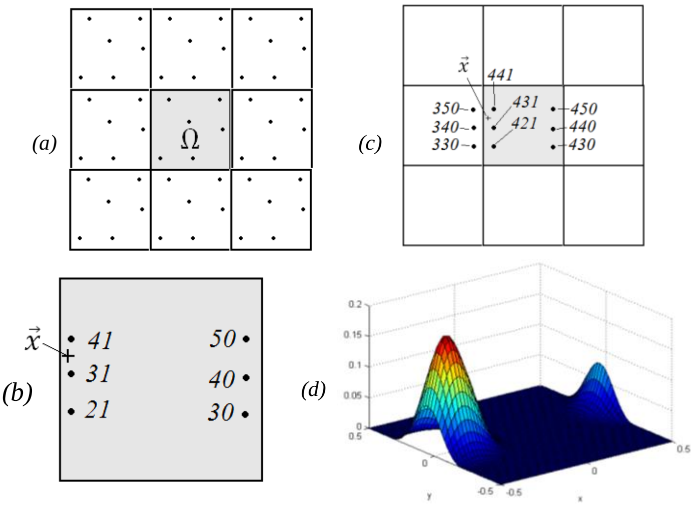

Fig. 4. *(a)* The extended domain Ω' (replicated nodal distributions). *(b)* 6 nodes influencing $\vec{x}$, 3 of which come from a neighbor cell. *(c)* Equivalent scenario for Ω: the value of ϕ calculated for node 330 (outside Ω) is transferred to its equivalent node inside Ω, node 430. Nodes 330 and 430 (global indices) both have index 30 in the new index scheme. *(d)* A periodic shape function (in Ω) associated to a node close to the left edge of Ω (431 in the global scheme, equivalent to 31 in the new scheme).

The process of finding periodic shape functions will be illustrated here for two-dimensional problems, just for the sake of easier visualization. Let us suppose that the problem is stated in a cell Ω. In this cell, we set up a nodal distribution. We then surround this cell with other 8 cells, and in each one of these extra cells, we assume a nodal distribution *identical* to that set up in Ω (i.e., we replicate it throughout). The situation is depicted in Fig.4a: nine cells, each one with the same nodal distribution within. This array of nine cells form an extended domain Ω', in the middle of which our original cell is situated. We then proceed to ascribe a global index i to each one of the nodes in Ω'. If there are, say, 100 nodes in the original cell Ω, then in Ω' the nodes in the first cell (top left) vary from 1 to 100; in the second cell (top middle), from 101 to 200, in the third (top right), from 201 to 300, and so on. The nodes in the innermost cell (Ω) vary from 401 to 500. Just to note: we use this bunch of extra nodes only when calculating the shape functions; as far as the problem (14) is concerned, only the innermost cell Ω and its nodes are considered.

Now in order to produce the periodic shape functions, nodes that occupy the same position within each cell throughout the 9 cells are considered equivalent. For example, nodes 2 (top left cell), 102 (top middle cell), 202 (top right cell), and so on, amounts to the same entity. It is as if the 9 nodes with global indices (2,102,202,…,802) were all equivalent to each other. As a result of the equivalence, each node in the extended domain Ω' can be mapped to a node inside Ω. This is carried out through a new index scheme: $I = i \bmod N$, where i is a node's global index (in Ω'), N is

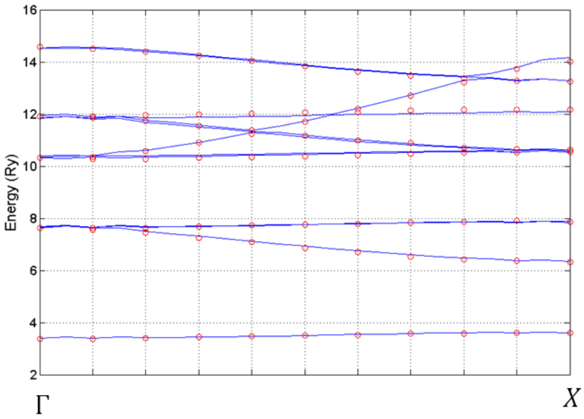

Fig. 5. The electronic band structure for the second example (Kronig-Penney model). Blue line: MLPG4/LBIE method. Red balls: results taken from [5].

the number of nodes inside a cell, and I is the node's global index mapped to the innermost cell Ω. Figure 4b illustrates this: suppose we want to calculate ϕ associated to node 431 (located close to the left edge) at $\vec{x}$. Given $\vec{x}$, we apply the MLS procedure considering also nodes from neighbor cells. This is accounted for by the global index scheme. For example, in the global scheme, the nodes influencing $\vec{x}$ are (330, 340, 350, 421, 431, 441). Nodes 330, 340 and 350 come from a neighbor cell. So information concerning these 6 nodes is fed into the matrices of the MLS approximation, and we get a vector whose elements are the shape functions $(\phi_{330}, \phi_{340}, \phi_{350}, \phi_{421}, \phi_{431}, \phi_{441})$ evaluated at $\vec{x}$. Now it is time to find the equivalent indices: the influencing nodes at $\vec{x}$ are then (30, 40, 50, 21, 31, 41). The equivalent scenario is illustrated in Fig.4(c). A subtlety should be noticed: in Fig. 4(b), nodes 340 and 440 are both equivalent (equivalent index 40), but the correct distance to be taken is the shorter distance from $\vec{x}$ to node 340, and not from $\vec{x}$ to 440. Just manipulating indices in this way avoids the issue of having to figure out the correct distance between points and nodes. The desired ϕ can be seen throughout Ω just by taking a set of points $\vec{x}$ covering Ω. The result is shown in Fig.4(d) and behold – we have just got a periodic shape function! The same profile is obtained at the left and right edges, i.e., $\phi(\vec{x}+\vec{L}) = \phi(\vec{x})$ and $\partial\phi(\vec{x}+\vec{L})/\partial n = \partial\phi(\vec{x})/\partial n$.

The extension of this procedure to three-dimensional problems is: we take a cubic cell Ω, set up a nodal distribution and replicate it throughout the 26 cells surrounding Ω. We form a global numbering scheme, do all the MLS calculations as if we were dealing with a larger problem and then map the global indices back to Ω (equivalent indices, represented here by uppercase letters). These new periodic shape functions form a vector space: a linear combination of them will also be periodic in a cell. Then the approximated ψ will also be periodic. Conclusion: the boundary conditions need not be imposed. Just take a cell Ω, spread some nodes, attach a spherical test domain Υ_I to each node I and enforce the weak form at them:

$$u(\vec{x}_I) + \oint\!\!\!\oint_{\partial\Upsilon_I} \frac{\partial v_I}{\partial n} u\, dS + \qquad (15)$$
$$\iiint_{\Upsilon_I} v_I \left[-2j\vec{K}\cdot\nabla u + \left(V + \|\vec{K}\|^2\right) u\right] dV = E \iiint_{\Upsilon_I} v_I u\, dV$$

where (15) came from Green's second identity. A different weak form could be obtained via weighted residual method (as explained earlier). After substituting u by an expansion like (1), one gets a generalized eigenvalue problem of the form $A\hat{\mathbf{u}} = EB\hat{\mathbf{u}}$.

In the example studied, Ω is a cube whose sides b measure 3 a.u., and the potential energy $V(\vec{x})$ is the three-dimensional Kronig-Penney potential:

$$V(x,y,z) = \bar{V}(x) + \bar{V}(y) + \bar{V}(z) \qquad (16)$$
$$\bar{V}(x) = \begin{cases} 0, & 0 \le x \le a \\ V_0, & a \le x \le b \end{cases}$$

where $a = 2$ a.u. and $V_0 = 6.5$ Ry. The eigenvalues (Rydberg a.u.) calculated as functions of $\vec{K}$ are shown in Fig. 5. The Bloch vector $\vec{K}$ varies from $[0;0;0]$ (point Γ) to $(\pi/b)[1;0;0]$ (point X). A total of 729 (9x9x9) nodes has been employed in the analysis, and a good concordance can be verified when the LBIE solutions are compared to the results provided by another numerical method [5].

C. How to solve boundary value problems: The nonlinear Schrödinger equation

We now turn our attention to boundary value problems, here illustrated by the two-dimensional nonlinear Schrödinger (NLS) equation. Let Ω be the computational domain. The NLS reads as

$$i\frac{\partial\psi(\vec{x},t)}{\partial t} + \nabla^2\psi(\vec{x},t) = \alpha(\vec{x},t) + \beta(\vec{x})|\psi(\vec{x},t)|^p\psi(\vec{x},t) \qquad (17)$$

The initial and boundary conditions are

$$\begin{cases} \psi(\vec{x},0) = f(\vec{x}), & \vec{x} \in \Omega \\ \psi(\vec{x},t) = g(\vec{x},t), & \vec{x} \in \partial\Omega_g \ \text{(Dirichlet)} \\ \dfrac{\partial\psi}{\partial n}(\vec{x},t) = h(\vec{x},t), & \vec{x} \in \partial\Omega_h \ \text{(Neumann)} \end{cases} \qquad (18)$$

where $\vec{x} = (x,y) \in \Omega$, $i = \sqrt{-1}$, α, β, f, g and h are known functions, $\partial\Omega = \partial\Omega_g \cup \partial\Omega_h$, and p is a positive real number. The time range is $0 \le t \le T$.

Dehgan and Mirzaei [6] solve this problem through MLPG5, a meshless method whose test function v is a Heaviside step function instead of a function like (3), which characterizes MLPG4. Furthermore, [6] does not use the collocation method when treating Neumann boundary conditions; intersections have to be found there. In order to solve (17) and (18), we take some approximations. First, a discretization in time: $t \to \{0, \Delta t, 2\Delta t, \ldots, k\Delta t, \ldots\}$, where $k \in \mathbb{N}$ and Δt is the time step. We use the shorthand $\psi^{(k)}(\vec{x}) = \psi(\vec{x}, k\Delta t)$. Second, $\psi(\vec{x},t) = \eta\psi^{(k+1)}(\vec{x}) + (1-\eta)\psi^{(k)}(\vec{x})$, where $0 < \eta \le 1$. Third, $\partial\psi(\vec{x},t)/\partial t = \left[\psi^{(k+1)}(\vec{x}) - \psi^{(k)}(\vec{x})\right]/\Delta t$. Fourth, $\alpha(\vec{x},t) = \alpha^{(k+1)}(\vec{x})$. Fifth, in order to apply a predictor-corrector scheme (to be explained later), the nonlinearity is "approximated", i.e., $\beta(\vec{x})|\psi(\vec{x},t)|^p\psi(\vec{x},t) \ \to\ \beta(\vec{x})\left|\psi_{ap}(\vec{x},t)\right|^p\psi(\vec{x},t)$. After inserting all these approximations at (17), we get a strong form:

$$\eta\nabla^2\psi^{(k+1)} + \frac{i}{\Delta t}\psi^{(k+1)} - \eta\alpha^{(k+1)}\psi^{(k+1)} - \eta\beta\left|\psi_{ap}\right|^p\psi^{(k+1)}$$
$$= -(1-\eta)\nabla^2\psi^{(k)} + \frac{i}{\Delta t}\psi^{(k)} + (1-\eta)\alpha^{(k)}\psi^{(k)}$$
$$+(1-\eta)\beta\left|\psi_{ap}\right|^p\psi^{(k)} \qquad (19)$$

(19) is organized in such a way that, if we know ψ at time step k, then ψ at time step $k+1$ can be known.

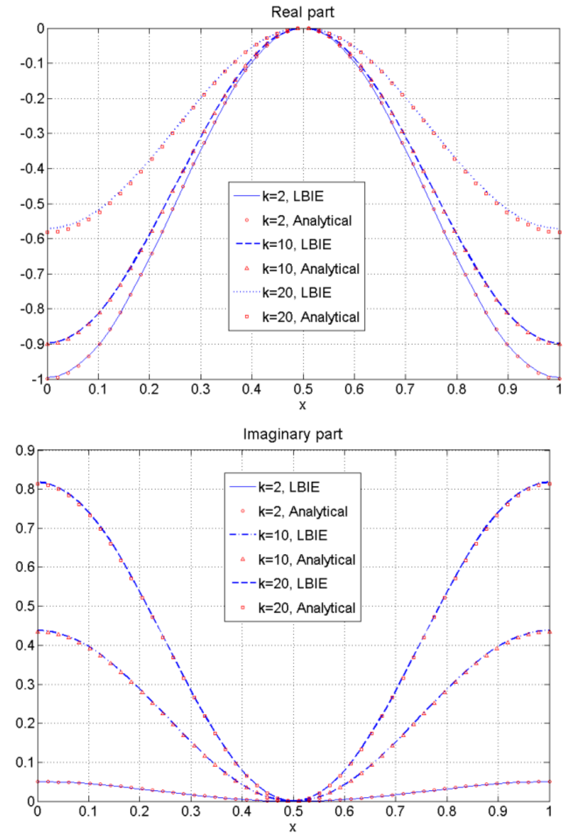

Fig. 6. The solution to NLS at 3 time steps (out of 20) along the line $x + y = 1$. 285 nodes have been scattered throughout Ω.

After spreading N_I interior nodes at Ω and N_B at $\partial\Omega$, we seek for a weak form. From Green's second identity we get:

$$-\eta\psi^{(k+1)}(\vec{x}_i) - \eta \oint_{\partial\Upsilon_i} \frac{\partial v_i}{\partial n}\psi^{(k+1)}dl \tag{20}$$

$$+ \iint_{\Upsilon_i} \left[\frac{i}{\Delta t} - \eta\alpha^{(k+1)} - \eta\beta\left|\psi_{ap}\right|^p\right] v_i\psi^{(k+1)}dS =$$

$$= (1-\eta)\psi^{(k)}(\vec{x}_i) + (1-\eta)\oint_{\partial\Upsilon_i} \frac{\partial v_i}{\partial n}\psi^{(k)}dl +$$

$$+ \iint_{\Upsilon_i} \left[\frac{i}{\Delta t} + (1-\eta)\alpha^{(k+1)} + (1-\eta)\beta\left|\psi_{ap}\right|^p\right] v_i\psi^{(k)}dS$$

After substituting $\psi^{(k+1)} = \sum_{j=1}^{N}\phi_j(\vec{x})\hat{u}_j^{(k+1)}$ and $\psi^{(k)} = \sum_{j=1}^{N}\phi_j(\vec{x})\hat{u}_j^{(k)}$ (the $\hat{u}_j^{(k)}$ are known from the previous iteration) and imposing (20) at each interior node, we get a matrix G (N_I rows and $(N_I + N_B)$ columns) and a vector Q (N_I rows). The other N_B equations come from the collocation at each boundary node. This information can be assembled in a matrix H (N_B rows and $(N_I + N_B)$ columns) and a vector R (N_B rows). Joining the matrices G and H into a matrix M ($(N_I + N_B)$ rows and $(N_I + N_B)$ columns) and the vectors Q and R into a vector F ($(N_I + N_B)$ elements) we form a system:

$$M(t,\psi_{ap})\widehat{\boldsymbol{u}}^{(k+1)} = F(t,\psi_{ap}) \tag{21}$$

where the matrix M and the vector F depend on the time and on the "approximated" term ψ_{ap}. The predictor-corrector scheme works as follows, assuming that the nodal parameters for the last iteration $\widehat{\boldsymbol{u}}^{(k)}$ are known:

First estimate for $\widehat{\boldsymbol{u}}^{(k+1)}$: $\widehat{\boldsymbol{u}}^{(k+1),0} = \widehat{\boldsymbol{u}}^{(k)}$
First estimate for $\widehat{\boldsymbol{u}}_{ap}$: $\widehat{\boldsymbol{u}}_{ap} = \widehat{\boldsymbol{u}}^{(k)}$
Calculate M and F.
Next estimate for $\widehat{\boldsymbol{u}}^{(k+1)}$: $\widehat{\boldsymbol{u}}^{(k+1),1} = M^{-1}F$
Next estimate for $\widehat{\boldsymbol{u}}_{ap}$: $\widehat{\boldsymbol{u}}_{ap} = 0.5[\widehat{\boldsymbol{u}}^{(k+1),1} + \widehat{\boldsymbol{u}}^{(k)}]$
Calculate M and F (with new $\widehat{\boldsymbol{u}}_{ap}$)
Next estimate for $\widehat{\boldsymbol{u}}^{(k+1)}$: $\widehat{\boldsymbol{u}}^{(k+1),2} = M^{-1}F$
Next estimate for $\widehat{\boldsymbol{u}}_{ap}$: $\widehat{\boldsymbol{u}}_{ap} = 0.5[\widehat{\boldsymbol{u}}^{(k+1),2} + \widehat{\boldsymbol{u}}^{(k)}]$
Calculate M e F (with new $\widehat{\boldsymbol{u}}_{ap}$)
Next estimate for $\widehat{\boldsymbol{u}}^{(k+1)}$: $\widehat{\boldsymbol{u}}^{(k+1),3} = M^{-1}F$
(…)
Repeat m times until

$$\left\|\widehat{\boldsymbol{u}}^{(k+1),m} - \widehat{\boldsymbol{u}}^{(k+1),m-1}\right\| \le \varepsilon$$

So

$$\widehat{\boldsymbol{u}}^{(k+1)} = \widehat{\boldsymbol{u}}^{(k+1),m}$$

We applied this time-domain LBIE to the NLS problem (17) in which $\Omega = [0,1] \times [0,1]$, $\alpha(x,y) = (1 - 2\pi^2)(1 - \cos^2\pi x \cos^2\pi y)$, $\beta(x,y) = 1 - 2\pi^2$, $p = 2$, $\psi(x,y,0) = \cos\pi x \cos\pi y$, $\partial\psi/\partial n = 0$ in $\partial\Omega$ and $\Delta t = 0.05$. The analytical solution to this problem is [6]:

$$\psi(x,y,t) = e^{-it}\cos\pi x \cos\pi y \tag{22}$$

In the simulations, the index k varies from 1 to 20 (20 time steps), and we have got again a good concordance between the numerical and analytical solutions, as Fig. 6 indicates.

VI. Conclusion

Through a number of situations coming from the field of quantum mechanics (two-dimensional, three-dimensional, eigenvalue, boundary value, time-domain and nonlinear problems) we have been able to verify the success of MLPG4. As a meshless method, it does not require any mesh, just a cloud of nodes. There are no elements to deal with. Furthermore, if the collocation procedure is employed when treating boundary conditions, the integrations of the weak forms are to be carried out in domains as simple as circles and spheres. In the future, when thoroughly examined and after all its properties and behaviors are unveiled, the MLPG can be pointed out as a serious alternative, or even a substitute for FEM.

References

[1] G. R. Liu, *Mesh Free Methods: Moving Beyond the Finite Element Method*. CRC Press, 2003.

[2] S. Atluri and S. Shen, "The Meshless Local Petrov-Galerkin Method: A simple & less-costly alternative to the finite-element and boundary element methods", *CMES*, vol. 3, no 1, pp. 11-51, 2002.

[3] Nicomedes, W. L., Mesquita, R. C., Moreira, F. J. S., "A Local Boundary Integral Equation (LBIE) Method in 2D Electromagnetic Wave Scattering, and a Meshless Discretization Approach". In: IMOC 2009 - SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference, November 03-06 2009

[4] Nicomedes, W. L., Mesquita, R. C., Moreira, F. J. S., "A Meshless Local Boundary Integral Equation Method for Three Dimensional Scalar Problems". In: 14th Biennial IEEE Conference on Electromagnetic Field Computation, IEEE CEFC 2010

[5] S. Jun, "Meshfree implementation for the real-space electronic-structure calculation of crystalline solids", International Journal for Numerical Methods in Engineering, v. 59, pp. 1909-1923, 2004.

[6] M. Dehghan and D. Mirzaei, "The Meshless Local Petrov-Galerkin (MLPG) method for the generalized two-dimensional nonlinear Schrödinger equation", Engineering Analysis with Boundary Elements", v. 32, pp. 747-756, 2008