Meshless local Petrov-Galerkin (MLPG) methods in quantum mechanics

Williams L. Nicomedes, Renato C. Mesquita and Fernando J.S. Moreira

Department of Electronic Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil

Abstract

Purpose – The purpose of this paper is to solve both eigenvalue and boundary value problems coming from the field of quantum mechanics through the application of meshless methods, particularly the one known as meshless local Petrov-Galerkin (MLPG).

Design/methodology/approach – Regarding eigenvalue problems, the authors show how to apply MLPG to the time-independent Schrödinger equation stated in three dimensions. Through a special procedure, the numerical integration of weak forms is carried out only for internal nodes. The boundary conditions are enforced through a collocation method. The final result is a generalized eigenvalue problem, which is readily solved for the energy levels. An example of boundary value problem is described by the time-dependent nonlinear Schrödinger equation. The weak forms are again stated only for internal nodes, whereas the same collocation scheme is employed to enforce the boundary conditions. The nonlinearity is dealt with by a predictor-corrector scheme.

Findings – Results show that the combination of MLPG and a collocation scheme works very well. The numerical results are compared to those provided by analytical solutions, exhibiting good agreement.

Originality/value – The flexibility of MLPG is made explicit. There are different ways to obtain the weak forms, and the boundary conditions can be enforced through a number of ways, the collocation scheme being just one of them. The shape functions used to approximate the solution can incorporate modifications that reflect some feature of the problem, like periodic boundary conditions. The value of this work resides in the fact that problems from other areas of electromagnetism can be attacked by the very same ideas herein described.

Keywords Meshless methods, Quantum mechanics, Schrödinger equation, Eigenvalues and Eigenfunctions, MLPG, Numerical analysis

Paper type Research paper

1. Introduction

Meshless (or meshfree) methods comprise a large class of numerical procedures whose seminal idea underlying all its members is, as its name indicates, to be able to build numerical solutions to differential equations defined in a certain geometrical domain without the need of setting up a mesh or a grid in this domain. These methods share some resemblances with FEM, like the operation with weak forms and the use of compactly supported shape functions, which leads to sparse stiffness matrices. The main difference is the complete absence 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.

COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering
Vol. 30 No. 6, 2011
pp. 1763-1776

0332-1649
DOI 10.1108/03321641111168084

Meshless methods have successfully been applied in many areas of computational mechanics since the 1990s. This new "numerical technology", as said in Liu (2002), is in its infancy; many challenges still remain to be studied.

One particular meshless method, the meshless local Petrov-Galerkin (MLPG), was devised by Atluri within the framework of mechanics (Atluri and Shen, 2002), 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, whereas there are many choices available to the test functions. We are particularly interested in MLPG4 (also called the local boundary integral equation (LBIE) method), whose test function is a solution to Green's problem for Laplace's equation (as addressed in Section 4). Our previous work in electrostatics and electromagnetic wave scattering corroborates the applicability of this method (Nicomedes *et al.*, 2009, 2011).

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 task 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.

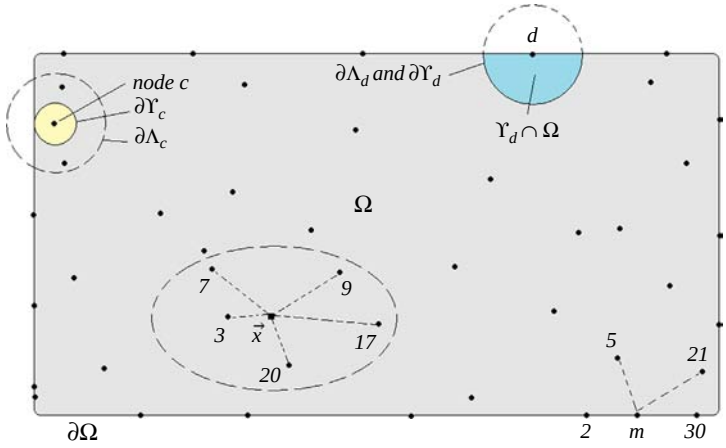
The second problem is also related to eigenvalues, but Schrödinger equation is now stated in a three-dimensional region (unit cell) in which the boundary conditions are Bloch-periodic. This is a problem about the electronic band structure of solids. Through a naïve process, we show how to embed a feature 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 (NLS) 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.

2. Meshless methods: nodes and domains

Let Ω be a region (whose global boundary is $\partial\Omega$) in 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 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 (Figure 1) there follows:

$$\psi(\vec{x}) \sim \psi^h(\vec{x}) = \sum_{i=1}^{p} \phi_{c(i)}(\vec{x})\hat{u}_{c(i)} = \mathbf{\Phi}(\vec{x})\hat{\boldsymbol{u}} \tag{1}$$



Notes: First: five nodes acting on point $\vec{x}$; second: a node close to the global boundary, like *c*, has its test domain Υ_c shrunk in such a way that it just touches $\partial\Omega$, 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 an intersection with Ω had to be found first; fourth: some nodes influencing a boundary node *m*

Figure 1.
Some facts about a domain Ω and its nodes

where the global index $c(i)$ runs through all P nodes whose influence domains include point $\vec{x}$ (in Figure 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 computational domain Ω:

$$\Omega \subseteq \bigcup_{i=1}^{N} \Lambda_i \tag{2}$$

Expression (2) above asserts that 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 ($\Lambda_i \cap \Lambda_j \neq \phi$) are freely allowed.

3. 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 one has to go through in order to calculate the shape functions are omitted here. Details can be found in Liu (2002) and in our previous works (Nicomedes *et al.*, 2009, 2011). It suffices to say that they are compactly supported. If one wants to calculate the shape functions at a point $\vec{x} = (x, y, z)$, one must first find which nodes extend their influence domains until $\vec{x}$ (nodes 3, 7, 9, 17 and 20 in Figure 1). Then, one plugs the coordinates of the influencing nodes and those of point $\vec{x}$ as well 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 shown in Figure 1, one would get a

vector of shape functions: $\Phi(\vec{x}) = [\phi_3(\vec{x}), \phi_7(\vec{x}), \phi_9(\vec{x}), \phi_{17}(\vec{x}), \phi_{20}(\vec{x})]$. Figure 2 shows a simple 2D MLS shape function ϕ associated to a node. The elements of S are all similar in form to this one, except that their peaks are located elsewhere, at each node in the domain Ω. From Figure 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.

4. The LBIE method

4.1 Test domains and 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, another function, the 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 Y_i (Figure 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:

$$\begin{aligned} \nabla^2 v_i &= -\delta(\vec{x} - \vec{x}_i) \text{ (a Dirac delta at } \vec{x}_i) \\ v_i &= 0 \text{ at the test domain boundary } \partial Y_i. \end{aligned} \tag{3}$$

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

$$v_i(\vec{x}) = \frac{1}{2\pi} \ln\left(\frac{s_i}{\|\vec{x} - \vec{x}_i\|}\right), \quad \text{for 2D problems} \tag{4}$$

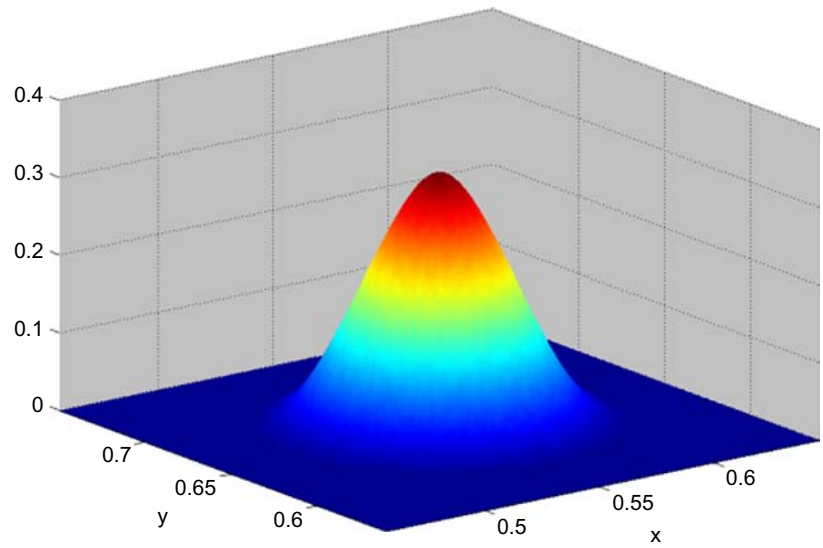


Figure 2.
A 2D MLS shape function associated to a node at (0.55,0.65)

$$v_i(\vec{x}) = \frac{1}{4\pi}\left(\frac{1}{\|\vec{x} - \vec{x}_i\|} - \frac{1}{s_i}\right), \quad \text{for 3D problems} \tag{5}$$

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

The test domains are the regions in 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 Y_i would have to be found in order to carry out the numerical integrations. Figure 1 shows this: had a test domain been assigned to node d, then 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 (as we did in Nicomedes *et al.* (2009)). 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 Figure 1). The radius of the test domain assigned to node i can be retrieved from a simple calculation: from a set of two elements $S_i = \{r_i,\ d_i\}$, where r_i is the radius of the influence domain Λ_i and d_i is the distance between node i and the global boundary $\partial\Omega$, we take s_i to be the smallest element in S_i. As said earlier, for the sake of simplicity, the influence domains of all nodes have the same value for their radii, i.e. $\forall i\ \ r_i = r_0$.

4.2 Imposing boundary conditions: the collocation method

A simple scheme that does not require any kind of numerical integration is a meshless collocation scheme, based on the approximation described by equation (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:

$$\sum_{i=1}^{Q} \phi_{c(i)}(\vec{x}_m)\hat{u}_{c(i)} = g(\vec{x}_m) \tag{6}$$

where the global index $c(i)$ runs through all Q nodes whose influence domains include point $\vec{x}_m$ (in Figure 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.

5. Worked problems

5.1 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 Schrödinger equation (ψ is called wavefunction or probability amplitude):

$$-\frac{\hbar^2}{2m}\nabla^2\psi(\vec{x}) + V(\vec{x})\psi(\vec{x}) = E\psi(\vec{x}) \quad \text{in}\ \Omega \tag{7}$$

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 equation (7) 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:

$$\begin{cases} -\frac{1}{2}\nabla^2\psi(\vec{x}) + V(\vec{x})\psi(\vec{x}) = E\psi(\vec{x}) & \text{in } \Omega \\ \psi(\vec{x}) = 0 & \text{at } \partial\Omega \end{cases} \tag{8}$$

One of the ways in which a weak form for equation (8) can be found is through the weighted residual method. We take each interior node i, multiply the residual of equation (8) by the test function v_i and integrate over the test domain Υ_i:

$$\iiint_{\Upsilon_i} [\nabla v_i \cdot \nabla\psi + 2Vv_i\psi]dV = E\iiint_{\Upsilon_i} 2v_i\psi\, dV \tag{9}$$

The other way is based on Green's second identity for the two functions ψ and v_i. The integrations are carried out in Υ_i, and taking the properties (3) into account, we get:

$$\psi(\vec{x}_i) + \oiint_{\partial\Upsilon_i} \frac{\partial v_i}{\partial n}\,\psi\, dS + \iiint_{\Upsilon_i} 2Vv_i\psi\, dV = E\iiint_{\Upsilon_i} 2v_i\psi\, dV \tag{10}$$

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 equation (10) that the method described in this paper also bears the name of LBIE method.

We begin by spreading N_I nodes throughout 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 (equation (9) or (10)) and impose it for each one of the N_I test domains Υ_i. The wavefunction ψ is expanded in shape functions like equation (1), and then we arrive at a matrix system $A\hat{\boldsymbol{u}} = EB\hat{\boldsymbol{u}}$ where:

$$A_{ij} = \iiint_{\Upsilon_i} [\nabla v_i \cdot \nabla\phi_j + 2Vv_i\phi_j]dV \quad \text{and} \quad B_{ij} = \iiint_{\Upsilon_i} 2v_i\phi_j\, dV \tag{11}$$

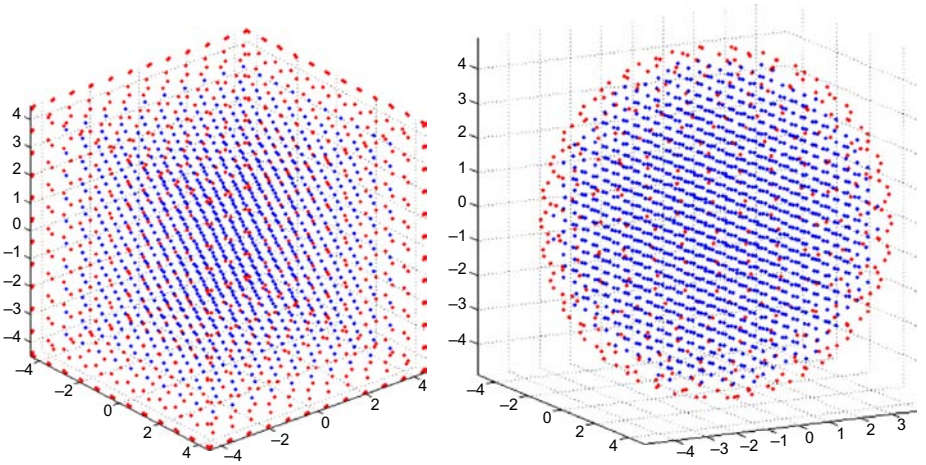
if the weak form (9) is used or:

$$A_{ij} = \phi_j(\vec{x}_i) + \oiint_{\partial\Upsilon_i} \frac{\partial v_i}{\partial n}\,\phi_j\, dS + \iiint_{\Upsilon_i} 2Vv_i\phi_j\, dV \quad \text{and} \quad B_{ij} = \iiint_{\Upsilon_i} 2v_i\phi_j\, dV \tag{12}$$

if equation (10) 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 (6) 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 at $\partial\Omega$. 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 $\|\vec{x}\| \leq R$ and ∞ otherwise. The radius R of the spherical region has been set to 5 a.u. and homogeneous Dirichlet conditions have also been imposed at $\partial\Omega$. The allowed energy levels (in a.u.) are given by $E = u_{np}^2/2R^2$, where u_{np} are the pth zeros of the spherical Bessel functions j_n. Figure 3 shows the nodal distributions for both examples, and Tables I and II show the



Notes: Each interior node (black) is ascribed a test domain Υ where the weak forms are integrated; the boundary nodes (grey) contribute with extra relations through the collocation procedure

Figure 3.
1,864 nodes in the cubic domain (example 1) and 1,868 nodes in the spherical domain (example 2)

n	Analytical	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 I.
First eigenvalues for the harmonic oscillator potential

n	p	Analytical	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

Table II.
First eigenvalues for the spherical infinite square well

concordance between numerical and analytical solutions. 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.

5.2 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. For the purpose 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:

$$\begin{cases} -\nabla^2\psi(\vec{x}) + V(\vec{x})\psi(\vec{x}) = E\psi(\vec{x}) & \text{in } \Omega \\ \psi(\vec{x}+\vec{L}) = e^{j\vec{K}\cdot\vec{L}}\psi(\vec{x}) & \text{at } \partial\Omega \\ \dfrac{\partial\psi(\vec{x}+\vec{L})}{\partial n} = e^{j\vec{K}\cdot\vec{L}}\dfrac{\partial\psi(\vec{x})}{\partial n} & \text{at } \partial\Omega \end{cases} \tag{13}$$

In equation (13), Schrödinger equation (7) 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 equation (13) mean that ψ at a face is equal to ψ at the opposite face multiplied by an exponential term. If we invoke Bloch theorem:

$$\psi(\vec{x}) = e^{j\vec{K}\cdot\vec{x}}u(\vec{x}) \tag{14}$$

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

$$\begin{cases} -\nabla^2 u(\vec{x}) - 2j\vec{K}\cdot\nabla u(\vec{x}) + \left(V(\vec{x}) + \left\|\vec{K}\right\|^2\right)u(\vec{x}) = Eu(\vec{x}) & \text{in } \Omega \\ u(\vec{x}+\vec{L}) = u(\vec{x}) & \text{at } \partial\Omega \\ \dfrac{\partial u(\vec{x}+\vec{L})}{\partial n} = \dfrac{\partial u(\vec{x})}{\partial n} & \text{at } \partial\Omega \end{cases} \tag{15}$$

The boundary conditions stated in equation (15) 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 equation (1), it is interesting if this periodicity were transferred to the ϕ's: boundary conditions would then be unnecessary.

We have come across a way to do this, described in Jun (2004), who solves this very same problem using a Galerkin formulation (instead of a Petrov-Galerkin one). But we have also devised an approach for getting periodic shape functions, based solely on operations regarding the indices of the nodes. In this somehow 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 which nodes influence $\vec{x}$ (nodes 3, 7, 9, 17 and 20 in Figure 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 $\vec{x}([\phi_3(\vec{x}), \ldots, \phi_{20}(\vec{x})])$.

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 eight 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 shown in Figure 4(a): nine cells, each one with the same nodal distribution within. This array of nine cells forms 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 (15) 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 nine cells are considered equivalent. For example, nodes 2 (top-left cell), 102 (top-middle cell), 202 (top-right cell), and so on,

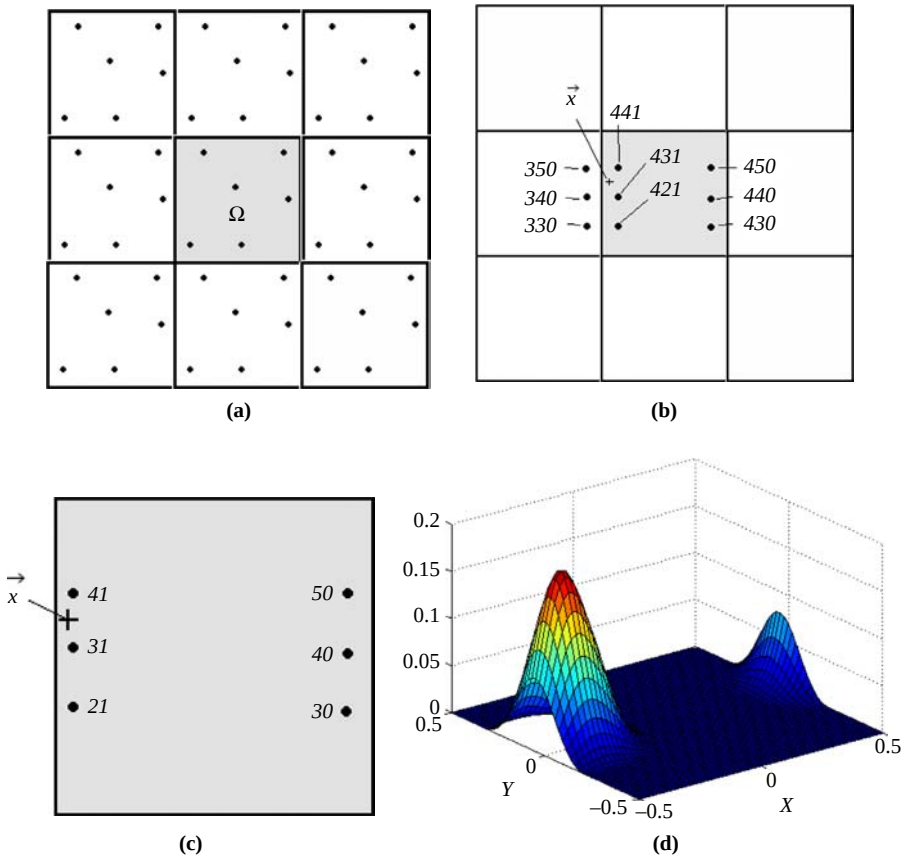


Figure 4.
(a) The extended domain Ω' (replicated nodal distributions); (b) six nodes influencing $\vec{x}$, three of which come from a neighbor cell; (c) equivalent scenario for Ω: the value of ϕ calculated for node 330 (outside Ω) is transferred to its equivalent node inside Ω, node 430; nodes 330 and 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)

amounts to the same entity. The nine nodes with global indices (2, 102, 202, ..., 802) are 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 the number of nodes inside a cell, and I is the node's global index mapped to the innermost cell Ω. Figure 4(b) shows this: suppose we want to calculate ϕ associated to node 431 (located close to the left edge of Ω) 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 six nodes is fed into the matrices of the MLS approximation, and we get a vector whose elements are the shape functions (ϕ_{330}, ϕ_{340}, ϕ_{350}, ϕ_{421}, ϕ_{431}, ϕ_{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 shown in Figure 4(c). A subtlety should be noticed: in Figure 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 node 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 by taking a set of points $\vec{x}$ covering Ω. The result is shown in Figure 4(d), which clearly indicates that we have 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 straightforward: 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 Y_I to each node I and enforce the weak form at them:

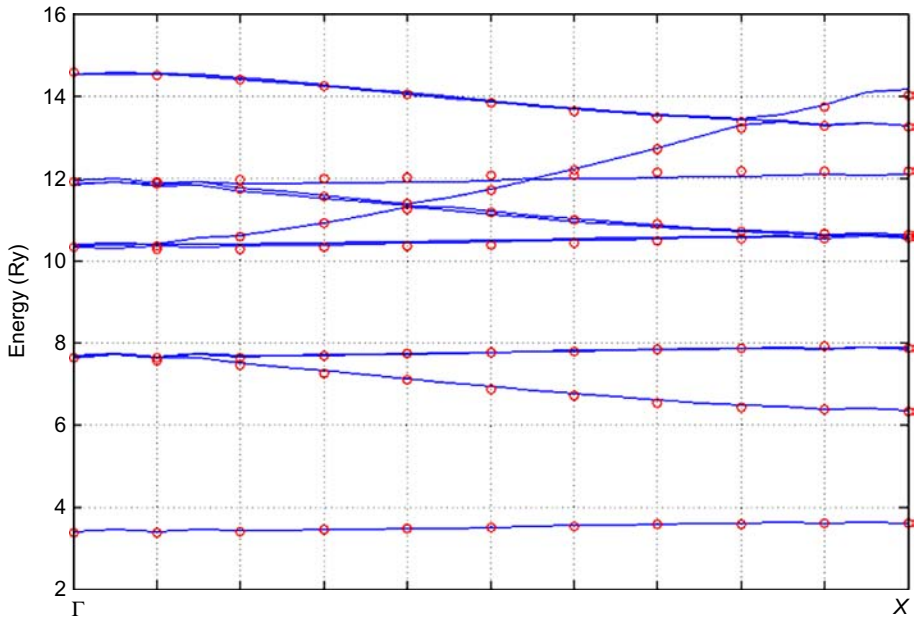
$$u(\vec{x}_I) + \oint_{\partial Y_i} \frac{\partial v_I}{\partial n} u \, dS + \iiint_{Y_i} v_I \left[-2j\vec{K}\cdot\nabla u + \left(V + \left\| \vec{K} \right\|^2 \right) u \right] dV = E \iiint_{Y_i} v_I u \, dV \quad (16)$$

where equation (16) 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 equation (1), one gets a generalized eigenvalue problem of the form $A\hat{\boldsymbol{u}} = EB\hat{\boldsymbol{u}}$.

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

$$\begin{aligned} V(x,y,z) &= \bar{V}(x) + \bar{V}(y) + \bar{V}(z) \quad \bar{V}(x) \\ &= \begin{cases} 0, & 0 \le x \le a \\ V_0, & a \le x \le b \end{cases} \end{aligned} \quad (17)$$

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 Figure 5. The Bloch vector $\vec{K}$ varies from [0; 0; 0] (point Γ)



Notes: Blue line: MLPG4/LBIE method; red balls: results taken from Jun (2004)

Figure 5.
The electronic band structure for the second example (Kronig-Penney model)

to $(\pi/b)[1; 0; 0]$ (point X). A total of 729 ($9 \times 9 \times 9$) 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 (Jun, 2004).

5.3 How to solve boundary value problems: the NLS equation

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

$$i\frac{\partial \psi(\vec{x},t)}{\partial t} + \nabla^2\psi(\vec{x},t) = \alpha(\vec{x},t) + \beta(\vec{x})\left|\psi(\vec{x},t)\right|^p\psi(\vec{x},t) \tag{18}$$

The initial and boundary conditions are:

$$\begin{cases} \psi(\vec{x},0) = f(\vec{x}), & \vec{x} \in \Omega \\ \psi(\vec{x},t) = g(\vec{x},t), & \vec{x} \in \partial\Omega_g \text{ (Dirichlet)} \\ \dfrac{\partial \psi}{\partial n}(\vec{x},t) = h(\vec{x},t), & \vec{x} \in \partial\Omega_h \text{ (Neumann)} \end{cases} \tag{19}$$

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 $0 \leq t \leq T$.

In Dehgan and Mirzaei (2008), this problem is solved through MLPG5, a meshless method whose test function v is a Heaviside step function instead of a function like equation (4), which characterizes MLPG4. Furthermore, the procedure described in Dehgan and Mirzaei (2008) does not use the collocation method when treating

Neumann boundary conditions; intersections have to be found there. In order to solve equations (18) and (19), we take some approximations.

First, a discretization in time: $t = \{0,\Delta t,2\Delta t,\ldots,k\Delta t,\ldots\}$, where $k \in \mathbf{N}$ 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 $0 < \eta \leq 1$.

Third, $\partial\psi(\vec{x}, t)/\partial t = [\psi^{(k+1)}(\vec{x}) - \psi^{(k)}(\vec{x})]/\Delta t$.

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. $\beta(\vec{x})|\psi(\vec{x}, t)|^p\psi(\vec{x}, t) \sim \beta(\vec{x})|\psi_{ap}(\vec{x}, t)|^p\psi(\vec{x}, t)$.

After inserting all these approximations at equation (18), we get a strong form:

$$\begin{aligned}\eta\nabla^2\psi^{(k+1)} + \frac{i}{\Delta t}\,\psi^{(k+1)} - \eta\alpha^{(k+1)}\psi^{(k+1)} - \eta\beta|\psi_{ap}|^p\psi^{(k+1)} \\ = -(1 - \eta)\nabla^2\psi^{(k)} + \frac{i}{\Delta t}\,\psi^{(k)} + (1 - \eta)\alpha^{(k)}\psi^{(k)} + (1 - \eta)\beta|\psi_{ap}|^p\psi^{(k)}\end{aligned} \tag{20}$$

Expression (20) 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. After spreading N_I interior nodes across Ω and N_B at $\partial\Omega$, we seek for a weak form. From Green's second identity we get:

$$\begin{aligned}&- \eta\psi^{(k+1)}(\vec{x}_i) - \eta\oint_{\partial\Upsilon_i}\frac{\partial v_i}{\partial n}\,\psi^{(k+1)}dl \\ &\quad + \iint_{\Upsilon_i}\left[\frac{i}{\Delta t} - \eta\alpha^{(k+1)} - \eta\beta|\psi_{ap}|^p\right]v_i\psi^{(k+1)}dS \\ &= (1 - \eta)\psi^{(k)}(\vec{x}_i) + (1 - \eta)\oint_{\partial\Upsilon_i}\frac{\partial v_i}{\partial n}\,\psi^{(k)}dl \\ &\quad + \iint_{\Upsilon_i}\left[\frac{i}{\Delta t} + (1 - \eta)\alpha^{(k+1)} + (1 - \eta)\beta|\psi_{ap}|^p\right]v_i\psi^{(k)}dS\end{aligned} \tag{21}$$

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 $\hat{u}_j^{(k)}$ coefficients are known from the previous iteration) and imposing equation (21) 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:

$$M(t, \psi_{ap})\hat{\boldsymbol{u}}^{(k+1)} = F(t, \psi_{ap}) \tag{22}$$

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 $\hat{\boldsymbol{u}}^{(k)}$ are known:

First estimate for $\hat{\boldsymbol{u}}^{(k+1)}$: $\hat{\boldsymbol{u}}^{(k+1),0} = \hat{\boldsymbol{u}}^{(k)}$

First estimate for $\hat{\boldsymbol{u}}_{ap}$: $\hat{\boldsymbol{u}}_{ap} = \hat{\boldsymbol{u}}^{(k)}$

Calculate M and F.

Next estimate for $\hat{\boldsymbol{u}}^{(k+1)}$: $\hat{\boldsymbol{u}}^{(k+1),1} = M^{-1}F$

Next estimate for $\hat{\boldsymbol{u}}_{ap}$: $\hat{\boldsymbol{u}}_{ap} = 0.5\left[\hat{\boldsymbol{u}}^{(k+1),1} + \hat{\boldsymbol{u}}^{(k)}\right]$

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

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

So:

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

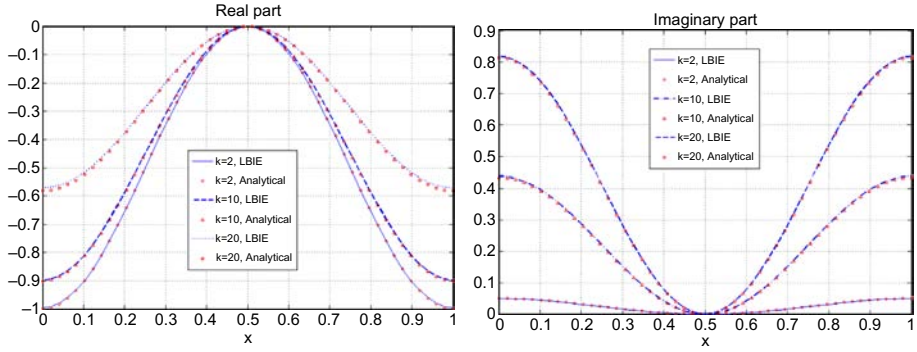
We applied this time-domain LBIE to the NLS problem (18) in which $\Omega = [0,1] \times [0,1]$, $\alpha(x, y) = (1 - 2\pi^2)(1 - \cos^2\pi x \cos^2\pi y)$, $\beta(x, y) = 1 - 2\pi^2$, $p = 2$, $\psi(x, y, 0) = \cos\pi x \cos\pi y$, $\partial\psi/\partial n = 0$ in $\partial\Omega$ and $\Delta t = 0.05$. The analytical solution to this problem is:

$$\psi(x, y, t) = e^{-it} \cos \pi x \cos \pi y \tag{23}$$

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 Figure 6 shows.

6. 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



Note: 285 nodes have been scattered throughout Ω

Figure 6.
The solution to NLS at three time steps (out of 20) along the line $x + y = 1$

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

Atluri, S. and Shen, S. (2002), "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.

Dehgan, M. and Mirzaei, D. (2008), "The meshless local Petrov-Galerkin (MLPG) method for the generalized two-dimensional nonlinear Schrödinger equation", *Engineering Analysis with Boundary Elements*, Vol. 32, pp. 747-56.

Jun, S. (2004), "Meshfree implementation for the real-space electronic-structure calculation of crystalline solids", *International Journal for Numerical Methods in Engineering*, Vol. 59, pp. 1909-23.

Liu, G.R. (2002), *Mesh Free Methods: Moving Beyond the Finite Element Method*, CRC Press, Boca Raton, FL.

Nicomedes, W.L., Mesquita, R.C. and Moreira, F.J.S. (2009), "A local boundary integral equation (LBIE) method in 2D electromagnetic wave scattering, and a meshless discretization approach", *Proceedings of the SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference*, pp. 133-7, available at: http://ieeexplore.ieee.org

Nicomedes, W.L., Mesquita, R.C. and Moreira, F.J.S. (2011), "A meshless local boundary integral equation method for three dimensional scalar problems", *IEEE Transactions on Magnetics*, Vol. 47 No. 5 (to appear in May 2011).

About the authors

Williams L. Nicomedes received his Bachelor's degree and Master's degree in Electrical Engineering from the Federal University of Minas Gerais (UFMG) in 2008 and 2011, respectively. He is interested in the application of meshfree methods to problems arising in engineering, physics, and other branches of applied mathematics. Williams L. Nicomedes is the corresponding author and can be contacted at: wlnicomedes@yahoo.com.br

Renato C. Mesquita is a Professor at the Electrical Engineering Department, Federal University of Minas Gerais, Brazil. He received his BE and MSc degrees from the Federal University of Minas Gerais, 1982 and 1987, and his PhD from the Federal University of Santa Catarina, Brazil, 1990. His main research interest is in the area of electromagnetic field computation.

Fernando J.S. Moreira received the BS and MS degrees in Electrical Engineering from the Catholic University, Rio de Janeiro, Brazil, in 1989 and 1992, respectively, and the PhD degree in Electrical Engineering from the University of Southern California in 1997. Since 1998, he has been with the Department of Electronics Engineering of the Federal University of Minas Gerais, Brazil, where he is currently an Associate Professor. His research interests are in the areas of electromagnetics, antennas and propagation. He has authored or co-authored over 100 journal and conference papers in these areas. He is a member of Eta Kappa Nu, IEEE Antennas and Propagation Society, and the Brazilian Microwave and Optoelectronics Society.