

Refined Meshless Local Strong Form solution of Cauchy-Navier equation on an irregular domain

J. Slak, G. Kosec*

*Jozef Stefan Institute, Parallel and Distributed systems Laboratory, Jamova 39 1000
Ljubljana, Slovenia*

Abstract

This paper considers a numerical solution of a linear elasticity problem, namely the Cauchy-Navier equation, using a strong form method based on a local Weighted Least Squares (WLS) approximation. The main advantage of the employed numerical approach, also referred to as a Meshless Local Strong Form method, is its generality in terms of approximation setup and positions of computational nodes. In this paper, flexibility regarding the nodal position is demonstrated through two numerical examples, i.e. a drilled cantilever beam, where an irregular domain is treated with a relatively simple nodal positioning algorithm, and a Hertzian contact problem, where again, a relatively simple h-refinement algorithm is used to extensively refine discretization under the contact area. The results are presented in terms of accuracy and convergence rates, using different approximations and refinement setups, namely Gaussian and monomial based approximations, and a comparison of execution time for each block of the solution procedure.

Keywords: Meshless Local Strong Form Method, Weighted Least Squares, Shape functions, Cauchy-Navier equation, Cantilever beam, Hertzian contact, h-refinement, Irregular domain

*Corresponding author
Email address: gregor.kosec@ijs.si (G. Kosec*)

1. Introduction

Linear elasticity problems, governed by the Cauchy-Navier equation, are typically addressed in their weak form with the Finite Elements Method (FEM) [1]. However, the problem has also been addressed in its strong form in the past, e.g. component-wise iterative solution with the Finite Differences Method (FDM) [2] and with the Finite Volumes Method (FVM) [3]. Besides mesh based methods, meshless methods have also been employed for solving solid mechanics problems in strong and weak form [4, 5]. The conceptual difference between meshless methods and mesh based methods is in the treatment of relations between nodes. In the mesh based methods the nodes need to be structured into polygons (mesh) that cover the whole computational domain, while on the other hand, meshless methods fully define relations between nodes through the relative inter nodal positions [6], with an immediate consequence of greater generality of the meshless methods.

Strong form meshless methods can be understood as generalizations of FDM, where instead of predetermined interpolation over a local support, a more general approach with variable support and basis is used to evaluate partial differential operators [7], e.g. collocation using Radial Basis Functions [5, 8] or approximation with monomial basis [9]. There are many other methods with more or less similar methodology introducing new variants of the strong form meshless principle [10]. On the other hand, weak form meshless methods are generalizations of FEM. Probably the most known method among weak form meshless methods is the Meshless Local Petrov Galerkin Method (MLPG) [11], where for each integration point a local support is used to evaluate field values and weight functions of a Moving Least Squares (MLS) approximation are used as test functions. In last few decades there have been many variants of MLPG introduced to mitigate numerical instabilities and to improve accuracy and convergence rate, etc. [10].

In general, recent developments in meshless community are vivid, ranging from analyses of computer execution on different platforms [6, 12], reducing

31 computational cost by introducing a piecewise approximation [13] to implemen-
32 tation of more complex multi-phase flow [14], and many more.

33 This paper extends the spectra of published papers with a generalized for-
34 mulation of a local strong form meshless method, termed Meshless Local Strong
35 Form Method (MLSM) enriched with h-refinement [15] and ability to discretize
36 arbitrary domains [7].

37 The introduced meshless approach is demonstrated on a solution of a bench-
38 mark cantilever beam case [16] and a Hertzian contact problem [17]. The re-
39 sults are presented in terms of displacement and stress plots, comparison against
40 closed form solutions, convergence analyses, and execution time analyses.

41 The goal of this paper is to demonstrate generality of MLSM that is driven
42 by the fact that all the building blocks of the method depend only on the rela-
43 tive positions between the computational nodes. This is a very useful feature,
44 especially when dealing with problems in multidimensional spaces, complex ge-
45 ometries, and moving boundaries. This feature can be also exploited to write
46 elegant generic code [18].

47 The rest of the paper is organized as follows: in section 2 the MLSM prin-
48 ciple is explained, in section 3 the governing problem is introduced, section 4 is
49 focused on solution procedure, section 5 focuses on discussing the results, and
50 finally, the paper offers some conclusions and guidelines for future work in the
51 last section.

52 2. MLSM formulation

53 The core of the spatial discretization used in this paper is a local approx-
54 imation of a considered field over the overlapping local support domains, i.e.
55 in each node we use approximation over a small local subset of neighbouring n
56 nodes. The trial function \hat{u} is thus introduced as

$$\hat{u}(\vec{p}) = \sum_{i=1}^m \alpha_i b_i(\vec{p}) = \mathbf{b}(\vec{p})^T \boldsymbol{\alpha}, \quad (1)$$

57 with m , $\boldsymbol{\alpha}$, \mathbf{b} and \vec{p} standing for the number of basis functions, approximation
58 coefficients, basis functions and the position vector, respectively. In cases when

59 the number of basis functions and the number of nodes in the support domain
60 are the same, $n = m$, the determination of coefficients $\boldsymbol{\alpha}$ simplifies to solving
61 a system of n linear equations, resulting from evaluating equation (1) in each
62 support node and setting it equal to a true value $u(\vec{p}_j)$, for j from 1 to n :

$$u_j := u(\vec{p}_j) = \mathbf{b}(\vec{p}_j)^\top \boldsymbol{\alpha}, \quad (2)$$

63 where \vec{p}_j are positions of support nodes and u_j is the actual value of considered
64 field in the support node \vec{p}_j . The above system can be written in matrix form
65 as

$$\mathbf{u} = \mathbf{B}\boldsymbol{\alpha}, \quad (3)$$

66 where \mathbf{B} stands for coefficient matrix with elements $B_{ji} = b_i(\vec{p}_j)$. The most
67 known method that uses such an approach is the Local Radial Basis Function
68 Collocation Method (LRBFCM) that has been recently used in various prob-
69 lems [5, 8].

70 In cases when the number of support nodes is higher than the number of basis
71 functions ($n > m$) a WLS approximation is chosen as a solution of equation (3),
72 which becomes an overdetermined problem. An example of this approach is
73 DAM [9] that was originally formulated to solve fluid flow in porous media.
74 DAM uses six monomials for basis and nine noded support domains to evaluate
75 first and second derivatives of physical fields required to solve the problem at
76 hand, namely the Navier Stokes equation. To determine the approximation
77 coefficients $\boldsymbol{\alpha}$, a norm

$$R^2 = \sum_j^n w(\vec{p}_j) (u(\vec{p}_j) - \hat{u}(\vec{p}_j))^2 = (\mathbf{B}\boldsymbol{\alpha} - \mathbf{u})^\top \mathbf{W}^2 (\mathbf{B}\boldsymbol{\alpha} - \mathbf{u}), \quad (4)$$

78 is minimized, where \mathbf{W} is a diagonal matrix with elements $W_{jj} = \sqrt{w(\vec{p}_j)}$ with

$$w(\vec{p}) = \exp \left(- \left(\frac{\|\vec{p}_0 - \vec{p}\|}{\sigma p_{\min}} \right)^2 \right), \quad (5)$$

80 where σ stands for weight shape parameter, \vec{p}_0 for centre of support domain
81 and p_{\min} for the distance to the nearest support domain node. There are differ-
82 ent computational approaches to minimizing (4). The most intuitive and also

83 computationally effective is to simply compute the gradient of R^2 with respect
 84 to α and setting it to zero, resulting in a positive definite system

$$\mathbf{B}^\top \mathbf{W}^2 \mathbf{B} \alpha = \mathbf{B}^\top \mathbf{W}^2 \mathbf{u}. \quad (6)$$

85 The problem of this approach is bad conditioning, as the condition number of
 86 $\mathbf{B}^\top \mathbf{W}^2 \mathbf{B}$ is the square of the condition number of $\mathbf{W} \mathbf{B}$, unnecessarily increas-
 87 ing numerical instability. A more stable and more expensive approach is QR
 88 decomposition. An even more stable approach is SVD decomposition, which is
 89 of course even more expensive. Nevertheless, the solution of equation (6) can
 90 be written generally in matrix form as

$$\alpha = (\mathbf{W} \mathbf{B})^+ \mathbf{W} \mathbf{u}, \quad (7)$$

91 where \mathbf{A}^+ stands for a Moore–Penrose pseudo inverse of matrix \mathbf{A} . By explicitly
 92 inserting equation (7) for α into (1), the equation

$$\hat{u}(\vec{p}) = \mathbf{b}(\vec{p})^\top (\mathbf{W} \mathbf{B})^+ \mathbf{W} \mathbf{u} = \chi(\vec{p}) \mathbf{u}, \quad (8)$$

93 is obtained, where $\chi = \mathbf{b}(\vec{p})^\top (\mathbf{W} \mathbf{B})^+ \mathbf{W}$ is called a shape function. Now, we
 94 can apply a partial differential operator \mathcal{L} to the trial function, and get

$$(\mathcal{L} \hat{u})(\vec{p}) = (\mathcal{L} \chi)(\vec{p}) \mathbf{u}. \quad (9)$$

In this paper we deal with a Cauchy-Navier equation and therefore following
 shape functions are needed, expressed explicitly as

$$\chi^{\partial x}(\vec{p}) = \frac{\partial \mathbf{b}}{\partial x}(\vec{p})^\top (\mathbf{W} \mathbf{B})^+ \mathbf{W}, \quad (10)$$

$$\chi^{\partial y}(\vec{p}) = \frac{\partial \mathbf{b}}{\partial y}(\vec{p})^\top (\mathbf{W} \mathbf{B})^+ \mathbf{W}, \quad (11)$$

$$\chi^{\partial x \partial x}(\vec{p}) = \frac{\partial^2 \mathbf{b}}{\partial x^2}(\vec{p})^\top (\mathbf{W} \mathbf{B})^+ \mathbf{W}, \quad (12)$$

$$\chi^{\partial x \partial y}(\vec{p}) = \frac{\partial^2 \mathbf{b}}{\partial x \partial y}(\vec{p})^\top (\mathbf{W} \mathbf{B})^+ \mathbf{W}, \quad (13)$$

$$\chi^{\partial y \partial y}(\vec{p}) = \frac{\partial^2 \mathbf{b}}{\partial y^2}(\vec{p})^\top (\mathbf{W} \mathbf{B})^+ \mathbf{W}. \quad (14)$$

95 The shape functions depend only on the numerical setup, namely nodal dis-
 96 tribution, shape parameter, basis and support selection, and can as such be
 97 precomputed for a specific computation.

98 **3. Governing problem**

99 The goal in this paper is to numerically determine the stress and displace-
 100 ment distributions in a solid body subjected to the applied external force. To
 101 obtain a displacement vector field \vec{u} throughout the domain, a Cauchy-Navier
 102 equation is solved, which can be expressed concisely in vector form as

$$(\lambda + \mu)\nabla(\nabla \cdot \vec{u}) + \mu\nabla^2\vec{u} = 0, \quad (15)$$

where μ and λ stand for Lamé constants. In two dimensions we express $\vec{u} = (u, v)$ and the equation reads

$$(\lambda + \mu)\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \quad (16)$$

$$(\lambda + \mu)\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = 0 \quad (17)$$

103 Two types of boundary conditions are commonly used when solving these types
 104 of problems, namely essential boundary conditions and traction (also called
 105 natural) boundary conditions. Essential boundary conditions specify displace-
 106 ments on some portion of the boundary of the domain, i.e. $\vec{u} = \vec{u}_0$, while traction
 107 boundary conditions specify surface traction $\sigma\vec{n} = \vec{t}_0$, where \vec{n} is an outside unit
 108 normal to the boundary of the domain and

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \quad (18)$$

is the stress tensor. In terms of displacement vector \vec{u} the traction boundary conditions read

$$t_{01} = \mu n_2 \frac{\partial u}{\partial y} + \lambda n_1 \frac{\partial v}{\partial y} + (2\mu + \lambda)n_1 \frac{\partial u}{\partial x} + \mu n_2 \frac{\partial v}{\partial x} \quad (19)$$

$$t_{02} = \mu n_1 \frac{\partial u}{\partial y} + (2\mu + \lambda)n_2 \frac{\partial v}{\partial y} + \lambda n_2 \frac{\partial u}{\partial x} + \mu n_1 \frac{\partial v}{\partial x} \quad (20)$$

109 where t_{0i} and n_i denote the Cartesian components of \vec{t}_0 and \vec{n} .

110 **4. Solution procedure**

111 *4.1. Discretization of the problem*

112 The elliptic boundary value problem at hand is discretized into a linear
 113 system of $2N$ algebraic equations by approximating the differential operations
 114 using MLSM, as described in section 2. A block system of linear equations for
 115 two vectors \mathbf{u} and \mathbf{v} of unknowns representing values $u(\vec{p}_i)$ and $v(\vec{p}_i)$, respec-
 116 tively, is constructed. This system is a discrete analogy of PDE (15) and can
 117 symbolically be represented as

$$\begin{bmatrix} U1 & V1 \\ U2 & V2 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \quad (21)$$

118 where \mathbf{u} and \mathbf{v} stand for unknown displacements, \mathbf{b}_1 and \mathbf{b}_2 for values of
 119 boundary conditions and blocks $U1, V1, U2, V2$ contain precomputed shape
 120 functions (10–14). With $\mathcal{N}(i)$ standing for a list of indices of the chosen n
 121 neighbours of a point \vec{p}_i , as introduced in the beginning of section 2, we can,
 122 for all indices i of internal nodes, express

$$\left. \begin{aligned} U1_{i,\mathcal{N}(i)_j} &= [(\lambda + 2\mu)\mathcal{X}^{\partial x \partial x}(\vec{p}_i) + \mu\mathcal{X}^{\partial y \partial y}(\vec{p}_i)]_j \\ V1_{i,\mathcal{N}(i)_j} &= [(\lambda + \mu)\mathcal{X}^{\partial x \partial y}(\vec{p}_i)]_j \\ \mathbf{b}_{1i} &= 0 \end{aligned} \right\}, \quad (22)$$

123

$$\left. \begin{aligned} U2_{i,\mathcal{N}(i)_j} &= [(\lambda + \mu)\mathcal{X}^{\partial x \partial y}(\vec{p}_i)]_j \\ V2_{i,\mathcal{N}(i)_j} &= [\mu\mathcal{X}^{\partial x \partial x}(\vec{p}_i) + (\lambda + 2\mu)\mathcal{X}^{\partial y \partial y}(\vec{p}_i)]_j \\ \mathbf{b}_{2i} &= 0 \end{aligned} \right\}, \quad (23)$$

124 for each $j = 1, \dots, n$. Note that equation (22) represents direct discrete analogue
 125 of (16) and, likewise, (23) of (17).

126 Similarly, for all indices i of boundary nodes with traction boundary condi-
 127 tions we express

$$\left. \begin{aligned} U1_{i,\mathcal{N}(i)_j} &= [\mu n_2 \mathcal{X}^{\partial y}(\vec{p}_i) + (2\mu + \lambda)n_1 \mathcal{X}^{\partial x}(\vec{p}_i)]_j \\ V1_{i,\mathcal{N}(i)_j} &= [\lambda n_1 \mathcal{X}^{\partial y}(\vec{p}_i) + \mu n_2 \mathcal{X}^{\partial x}(\vec{p}_i)]_j \\ \mathbf{b}_{1i} &= t_0(\vec{p}_i)_1 \end{aligned} \right\}, \quad (24)$$

$$\left. \begin{aligned} U2_{i,\mathcal{N}(i)_j} &= [\mu n_1 \mathcal{X}^{\partial y}(\vec{p}_i) + \lambda n_2 \mathcal{X}^{\partial x}(\vec{p}_i)]_j \\ V2_{i,\mathcal{N}(i)_j} &= [\mu n_1 \mathcal{X}^{\partial x}(\vec{p}_i) + (2\mu + \lambda)n_2 \mathcal{X}^{\partial y}(\vec{p}_i)]_j \\ \mathbf{b}_{2i} &= t_0(\vec{p}_i)_2 \end{aligned} \right\}, \quad (25)$$

128 for each $j = 1, \dots, n$, where n_i are the Cartesian components of the outside unit
 129 normal to the boundary in node \vec{p}_i . Again, equation (24) is a direct analogue
 130 of (19) and (25) of (20). And finally, for indices i of nodes with essential
 131 boundary condition, we express

$$\begin{aligned} U1_{i,i} &= 1 & \text{and} & & U2_{i,i} &= 1 \\ \mathbf{b}_{1i} &= u_0(\vec{p}_i)_1 & & & \mathbf{b}_{2i} &= u_0(\vec{p}_i)_2 \end{aligned} \quad (26)$$

132 System (21) is sparse with nonzero ratio of less than $2n/N$. An example
 133 of the matrix of this system for the cantilever beam problem described in sec-
 134 tion 5.1 is shown in Figure 1, where the block structure and different patterns
 135 for boundary and internal nodes are clearly visible.

136 4.2. Positioning of nodes in a complex domain

137 Meshless methods are advertised as the methods that do not require any
 138 topological relations among nodes. That implies that even randomly distributed
 139 nodes could be used [19]. However, it is well-known that with regularly dis-
 140 tributed nodes one achieves much better results in terms of accuracy and sta-
 141 bility [20]. This has also been recently reported for MLSM in a solution of
 142 a Navier-Stokes problem [7]. The reason behind the sensitivity regarding the
 143 distribution of nodes lies in the generation of shape functions. To construct
 144 a stable method well balanced support domains are needed, i.e. the nodes in
 145 support domain need to be distributed evenly enough [7]. This condition is
 146 obviously fulfilled in regular nodal distributions, but when working with more
 147 interesting geometries, the positioning of nodes requires additional treatment.
 148 In literature one can find several algorithms for distributing the nodes within
 149 the domain of different shapes [21, 22]. In this paper we will use an extremely
 150 simple algorithm, introduced in [7] to minimize the variations in distances be-
 151 tween nodes in the support domain. The basic idea is to “relax” the nodes based

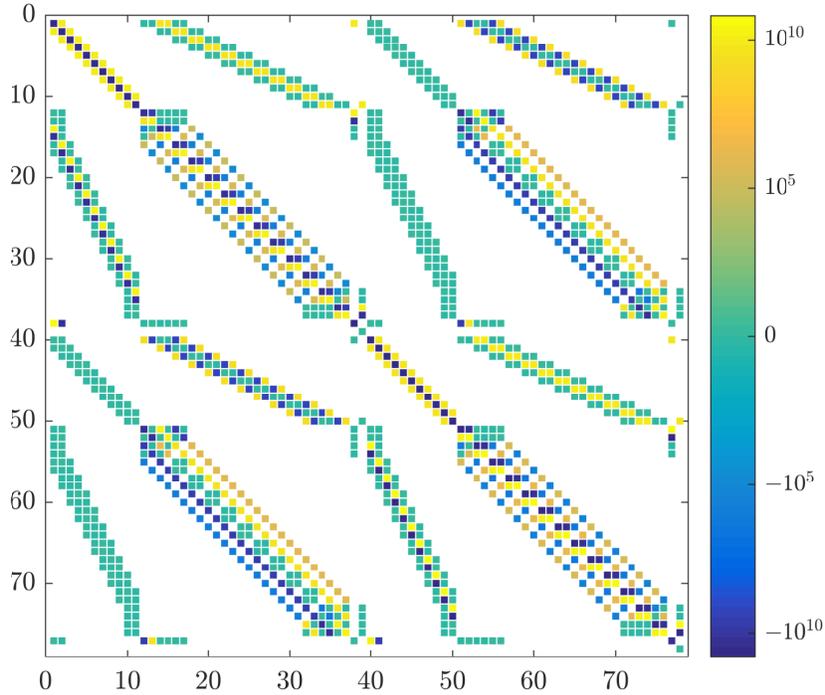


Figure 1: Matrix of the final system of equations in cantilever beam case with $N = 39$ and 22% nonzero elements.

152 on a potential between them. Since a Gaussian function is a suitable potential
 153 and already used as weight in the shape functions, the nodes are translated
 154 simply as

$$\delta\vec{p}(\vec{p}) = -\sigma_k \sum_{i=1}^{N_S} \nabla w(\vec{p} - \vec{p}_i), \quad (27)$$

155 where $\delta\vec{p}$, \vec{p}_i , σ_k and N_S stand for the translation step of the node,
 156 of i -th support node, relaxation parameter and number of support nodes,
 157 respectively (Figure 2a). After offsets in all nodes are computed, the nodes are
 158 repositioned as

$$\vec{p} \leftarrow \vec{p} + \delta\vec{p}(\vec{p}). \quad (28)$$

159 Presented iterative process procedure begins by positioning the boundary nodes,
 160 which are considered as the definition of the domain and are kept fixed through-
 161 out the process.

162 *4.3. h-refinement*

163 Besides flexibility regarding the shape of the domain, nodal refinement is of-
164 ten mandatory to achieve desired accuracy in cases with pronounced differences
165 in stress within the domain. A typical example of such situation is a contact
166 problem [17]. To mitigate the error in areas with high stresses the h-refinement
167 scheme is used., which has already been introduced into different meshless solu-
168 tions [23, 24]. In context of local RBF approximation the h-refinement has been
169 used in the solution of the Burger’s equation [15], where a quad-tree based algo-
170 rithm has been used to add and remove child nodes symmetrically around the
171 parent node in transient solution of Burgers’ equation. However, the algorithm
172 presented in [15] supported only regular nodal distribution. In this paper we
173 generalize it also to irregular nodal distribution.

174 In each node to be refined, new nodes are added on the half distances between
175 the node itself and its support nodes

$$\vec{p}_j^{new} = \frac{\vec{p} + \vec{p}_j}{2}, \quad (29)$$

176 where index j indicates j -th support node. When adding new nodes, checks are
177 performed if the newly added node is too close to any of the existing nodes;
178 in that case the node is not added. Moreover, if the refined node and support
179 node are both boundary nodes, newly added node is positioned on the boundary
180 (Figure 2b). This procedure can be repeated several times if an even more
181 refined domain is desired. These subsequent refinements will be called levels of
182 refinement and will be denoted as level i for the refinement that resulted from
183 i applications of the described algorithm.

184 The described algorithm follows the concept of meshless methods and as
185 such does not require any special topological relations between nodes to refine
186 a certain part of the computation domain. It is also flexible regarding the
187 dimensionality of the domain, i.e. there is no difference in implementation of 2D
188 or 3D variant of the algorithm.

189 An example on a non-trivial refinement is demonstrated in Fig. 3, where a
190 domain with a hole is considered. The vicinity of the hole is four times refined

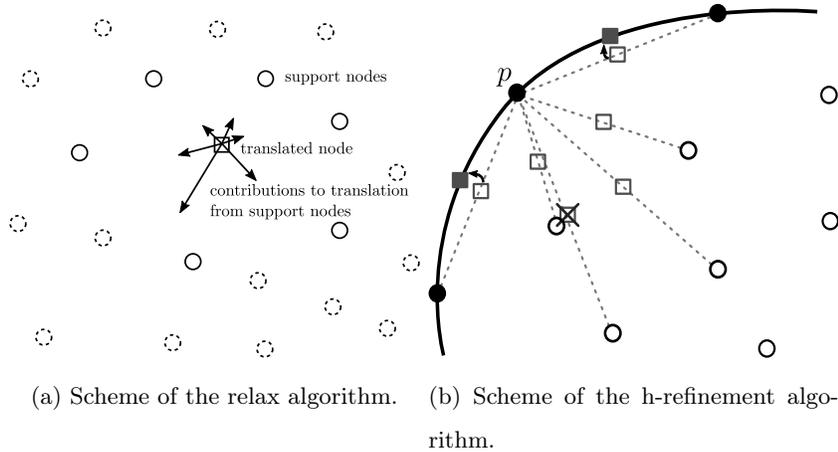


Figure 2: Schemes of algorithms used to improve the quality of the discretization.

191 and then, to mitigate possible irregularities during refinement, relaxed.

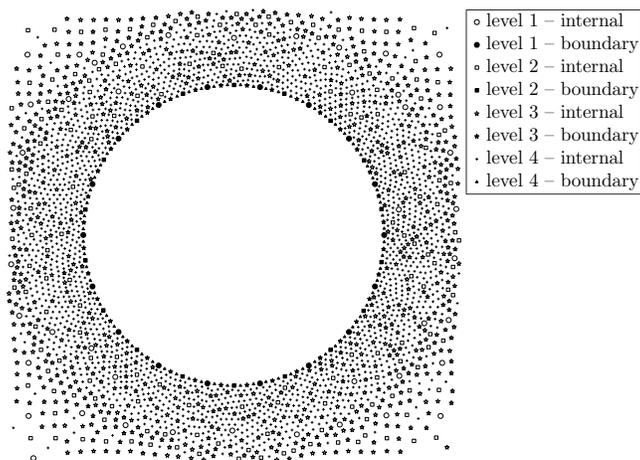


Figure 3: Four levels of the refinement algorithm applied around a hole in a domain after relaxation.

192 *4.4. Asymptotic complexity of MLSM*

193 The asymptotic complexity analysis begins with an assumption that eval-
 194 uations of basis functions, weights, linear operators and boundary conditions
 195 take $O(1)$ time. For simple domain discretization, such as the uniform grid in

196 a rectangle or random positioning, $O(N)$ time is required, where N stands for
 197 number of computational nodes. To find the neighbours of each point, a tree
 198 based data structure such as kd-tree [6], taking $O(N \log N)$ time to construct
 199 and $O(n \log N)$ time to query n closest nodes, is used. The relaxation of nodal
 200 positions (see section 4.2) with I iterations costs additional $O(InN \log^2 N)$ time.
 201 Re-finding the support nodes by rebuilding the tree and querying for support
 202 nodes once again, requires another $O((N + n) \log N)$ time. Calculation of the
 203 shape functions requires N SVD decompositions, each taking $O(nm^2)$ time, as
 204 well as some matrix and vector multiplication of lower complexity. Assembling
 205 the matrix takes $O(nN)$ time and assembling the right hand side takes $O(N)$
 206 of time. Then, the system is solved using BiCGSTAB iterative algorithm. The
 207 final time complexity is thus $O(InN \log^2 N + (N + n) \log N + m^2 nN) + T$, where
 208 T stands for the time spent by BiCGSTAB.

209 For comparison, the complexity of a well-known weak form Element Free
 210 Galerking method (EFG) [25] differs from MLSM in construction of the shape
 211 functions, whose computation requires $O(Nn_q m^2 n)$ time using EFG method,
 212 with n_q standing for the number of Gauss integration points per node. Addi-
 213 tionally, the number of nonzero elements in the final system of EFG is of order
 214 n_q times higher than that of MLSM, again increasing the complexity of EFG.

215 5. Numerical examples

216 5.1. Cantilever beam

First, the standard cantilever beam test is solved to assess accuracy and
 stability of the method. Consider an ideal thin cantilever beam of length L and
 height D covering the area $[0, L] \times [-D/2, D/2]$. Timoshenko beam theory offers
 a closed form solution for displacements and stresses in such a beam under plane
 stress conditions and a parabolic load on the left side. The solution is widely
 known and derived in e.g. [16], giving stresses in the beam as

$$\sigma_{xx} = \frac{Pxy}{I}, \quad \sigma_{yy} = 0, \quad \sigma_{xy} = \frac{P}{2I} \left(\frac{D^2}{4} - y^2 \right), \quad (30)$$

and displacements as

$$\begin{aligned} u &= \frac{Py(3D^2(\nu+1) - 4(3L^2 + (\nu+2)y^2 - 3x^2))}{24EI}, \\ v &= -\frac{P(3D^2(\nu+1)(L-x) + 4(L-x)^2(2L+x) + 12\nu xy^2)}{24EI}, \end{aligned} \quad (31)$$

217 where $I = \frac{1}{12}D^3$ is the moment of inertia around the horizontal axis, E is
218 Young's modulus, ν is the Poisson's ratio and P is the total load force.

In the numerical solution, traction free boundary conditions are used on the top and bottom of the domain, essential boundary conditions given by (31) are used on the right and traction boundary conditions given by (30) on the left

$$u(L, y) = \frac{Py(2D^2(1+\nu) - 4(2+\nu)y^2)}{24EI} \quad (32)$$

$$v(L, y) = -\frac{LvPy^2}{2EI} \quad (33)$$

$$\mu \frac{\partial u}{\partial y}(x, D/2) + \mu \frac{\partial v}{\partial x}(x, D/2) = 0 \quad (34)$$

$$\lambda \frac{\partial u}{\partial x}(x, D/2) + (\lambda + 2\mu) \frac{\partial v}{\partial y}(x, D/2) = 0 \quad (35)$$

$$-\mu \frac{\partial u}{\partial y}(x, -D/2) - \mu \frac{\partial v}{\partial x}(x, -D/2) = 0 \quad (36)$$

$$-\lambda \frac{\partial u}{\partial x}(x, -D/2) - (\lambda + 2\mu) \frac{\partial v}{\partial y}(x, -D/2) = 0 \quad (37)$$

$$-\lambda \frac{\partial v}{\partial y}(0, y) - (\lambda + 2\mu) \frac{\partial u}{\partial x}(0, y) = 0 \quad (38)$$

$$-\mu \frac{\partial u}{\partial y}(0, y) - \mu \frac{\partial v}{\partial x}(0, y) = \frac{P}{2I}((D/2)^2 - y^2). \quad (39)$$

219 The problem is solved using MLSM method with $n = 9$ or $n = 13$ support
220 nodes and Gaussian weight with $\sigma = 1$ (see (5)). Two sets of basis functions are
221 considered, 9 monomials

$$\mathbf{b} = \{1, x, y, x^2, y^2, xy, x^2y, xy^2, x^2y^2\} \quad (40)$$

222 and 9 Gaussian basis functions (see (5) for definition) centred in support nodes.
223 In the following discussions these two choices of basis functions will be referred
224 to as M9 and G9, respectively.

225 System (21) is solved with BiCGSTAB iterative algorithm [26] with ILUT
 226 preconditioning [27]. Values of $L = 30$ m, $D = 5$ m, $E = 72.1$ GPa, $\nu = 0.33$
 227 and $P = 1000$ N/m were chosen as physical parameters of the problem.

228 The acquired numerical solution of the cantilever beam problem is shown in
 229 Figure 4.

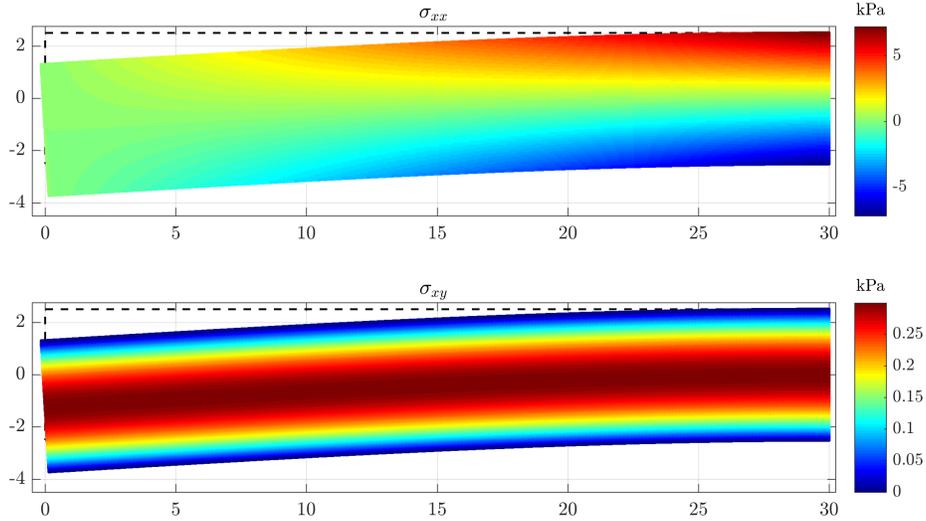


Figure 4: Numerical solution of cantilever beam case. Note that for the sake of visibility the displacements are multiplied by factor 10^5 .

The error of the numerical approximation of stresses and displacements is measured in relative discrete L_∞ norm, using

$$e_\infty(\vec{u}) = \frac{\max_{x \in X} \{\max\{|u(x) - \hat{u}(x)|, |v(x) - \hat{v}(x)|\}\}}{\max_{x \in X} \{\max\{|u(x)|, |v(x)|\}\}} \quad \text{and} \quad (41)$$

$$e_\infty(\sigma) = \frac{\max_{x \in X} \{\max\{|\sigma_{xx}(x) - \hat{\sigma}_{xx}(x)|, |\sigma_{yy}(x) - \hat{\sigma}_{yy}(x)|, |\sigma_{xy}(x) - \hat{\sigma}_{xy}(x)|\}\}}{\max_{x \in X} \{\max\{|\sigma_{xx}(x)|, |\sigma_{yy}(x)|, |\sigma_{xy}(x)|\}\}}, \quad (42)$$

230 as error indicators, with X representing the set of all nodes. Convergence with
 231 respect to the number of computational nodes is shown in Figure 5. The nu-
 232 merical approximations converge towards the correct solution in stress ($e_\infty(\sigma)$)
 233 norm as well, with approximately the same convergence rate.

234 It can be seen that monomials converge very regularly with order 1 as

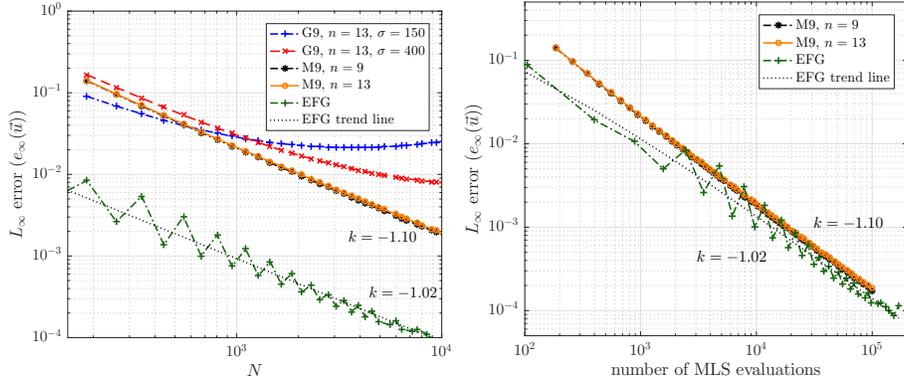


Figure 5: Accuracy of different MLSM setups compared to EFG per number of computational nodes (left) and number of MLS evaluations (right).

235 expected, while Gaussian functions exhibit slightly worse convergence. Such
 236 behaviour has already been reported in solution of diffusion equation, where
 237 MLSM with Gaussian basis failed to obtain accurate solution with a high number
 238 of computational nodes. More details about the phenomenon and further
 239 reading can be found in [28].

240 The method was compared to the standard Element Free Galerkin (EFG)
 241 method [29]. The EFG method used circular domains of influence with radius
 242 d_I equal to 3.5 times internodal distance, a cubic spline

$$w(\vec{p}) = \tilde{w}\left(\frac{\|\vec{p} - \vec{p}_i\|}{d_I}\right), \quad \tilde{w}(r) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & 0 \leq r < \frac{1}{2} \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3 & \frac{1}{2} \leq r < 1 \\ 0 & 1 \leq r \end{cases} \quad (43)$$

243 for a weight function, $n_q = 4$ Gaussian points for approximation of line integrals
 244 and $n_q = 16$ points for approximating area integrals. Lagrange multipliers were
 245 used to impose essential boundary conditions.

246 The performance of EFG with respect to the number of nodes is much better
 247 than MLSM. However, a more fair comparison would take into account also a
 248 higher complexity of the EFG. This can be achieved by comparing error with
 249 respect to the number of MLS evaluations, which is the most time consuming
 250 part of the solution procedure. In Figure 5 it is demonstrated that although

251 EFG provides much better results in comparison to MLSM at a given number of
 252 nodes, its accuracy becomes comparable to MLSM, when compared per number
 253 of MLS evaluations.

254 To assess the stability of the method regarding the nodal distribution, the
 255 following analysis was performed. A regular distribution of points as used in
 256 the solution in Figure 4 was distorted by adding a random perturbation to each
 257 internal node. Its position is altered by

$$\hat{\vec{p}} \leftarrow \vec{p} + \sigma \vec{U}, \quad \vec{U} \sim \text{Uniform}([0, \delta]^2), \quad (44)$$

258 where δ is the distance to the closest node, and measuring the accuracy of the
 259 solution with respect to σ , representing magnitude of the perturbation. An
 260 example of original and perturbed node distributions are shown in Figure 6.

261 Accuracy of the solution with respect to the perturbation magnitude is pre-
 262 sented in Figure 7. It is demonstrated that using monomials as a basis with 9
 263 support nodes results in an unstable setup. On the other hand monomials with
 264 13 support nodes are much more stable and equally accurate, while using Gaus-
 265 sian basis with high shape parameter is the most unstable setup. To mitigate
 266 the stability issue, a lower shape parameter can be chosen, however, at the cost
 267 of accuracy. Regardless of the setup one can expect the solution to be stable
 268 at least up to $\sigma \approx 0.1$. Note that using more nodes in support domain can also
 269 increase stability. Refer to [7] for more details.

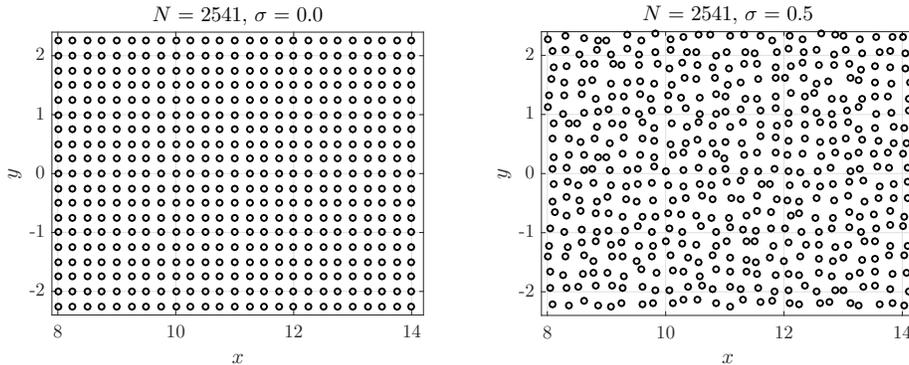


Figure 6: Regular and perturbed node positions, as used in stability analysis.

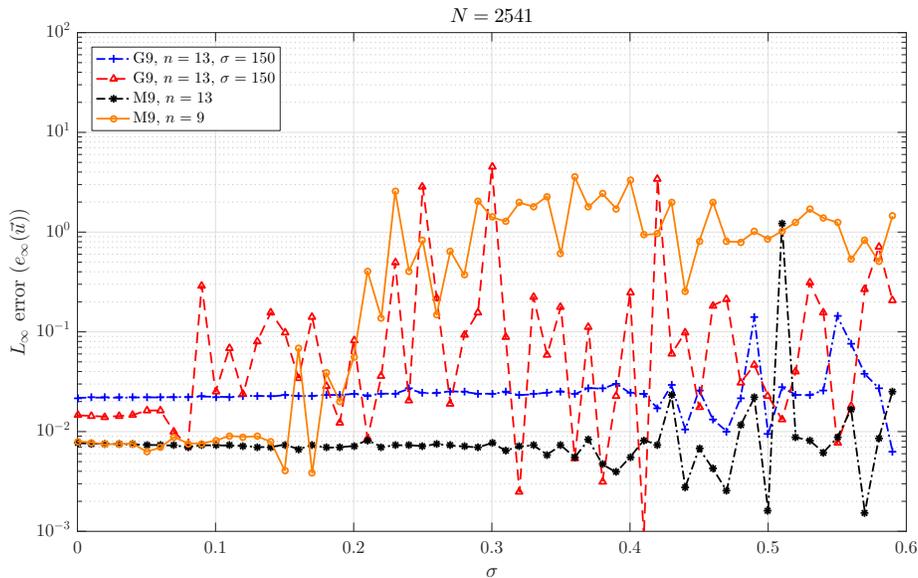


Figure 7: Stability of MLSM with respect to nodal perturbations.

270 Time spent on each part of the solution procedure is shown in Figure 8.
 271 All measurements were performed on a laptop computer with an Intel(R)
 272 Core(TM) i7-4700MQ @2.40GHz CPU and with 16 GiB of DDR3 RAM. MLSM
 273 is implemented in C++ [18] and compiled using g++ 7.1.7 for Linux with
 274 `-std=c++14 -O3 -DNDEBUG` flags. It can be seen that solving the system (21)
 275 makes up for more than 50% of total time spent. Around 70% of that time
 276 is spent on computing the preconditioner. The only other significant factor is
 277 computing the shape functions taking approximately 40% of total time. Domain
 278 construction and matrix assembly take negligible amounts of time, matching the
 279 predictions made by complexity analysis in section 4.4.

280 To emphasize the generality of MLSM method, a “drilled” domain is consid-
 281 ered in the next step. Arbitrarily positioned holes are added to the rectangular
 282 domain. The positioning algorithm described in section 4.2 and h-refinement
 283 algorithm described in section 4.3 are used to distribute the nodes inside the
 284 domain and refine the areas around the holes. The boundary conditions in this
 285 example are $\vec{u} = 0$ on the right, traction free on the inside of the holes and on

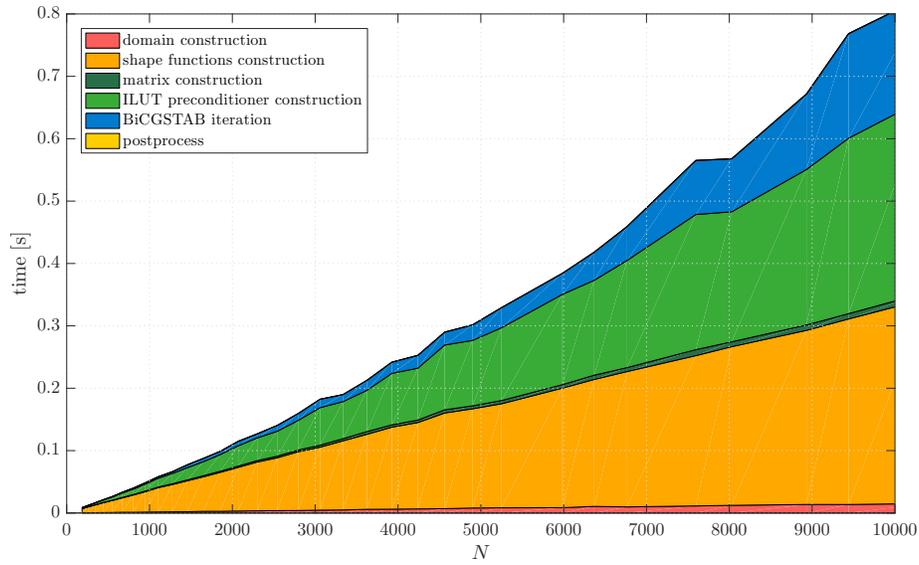


Figure 8: Execution time for different parts of the solution procedure with respect to the number of computational nodes.

286 top and bottom and uniform load of P/D on the left. The computed solution is shown in Figure 9 along with the ordinary cantilever beam example. Both

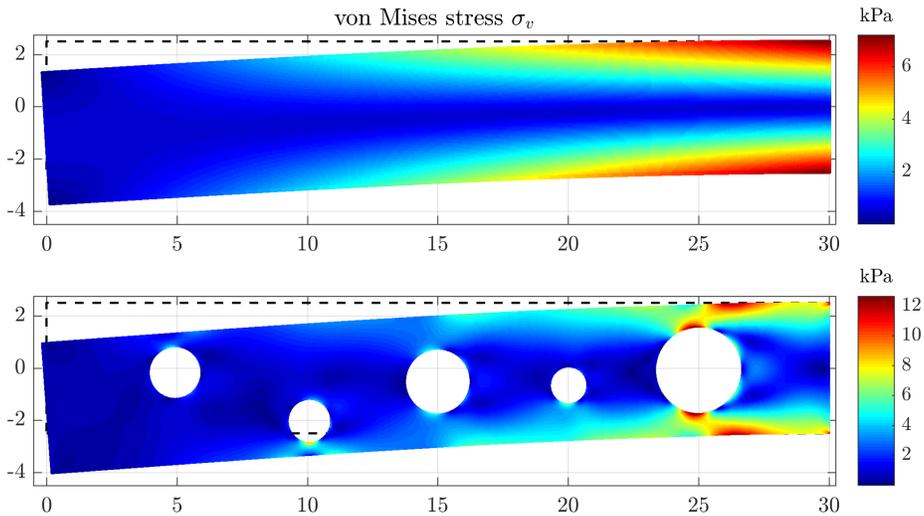


Figure 9: Numerical solution of a drilled cantilever beam case using $N = 177618$ nodes. Note that for the sake of visibility the displacements are multiplied by factor 10^5 .

287

288 solutions are coloured using von Mises stress σ_v , computed for the plane stress
 289 case as

$$\sigma_v = \sqrt{\sigma_{xx}^2 - \sigma_{xx}\sigma_{yy} + \sigma_{yy}^2 + 3\sigma_{xy}^2}. \quad (45)$$

290 To further illustrate the generality of the method, an even more deformed
 domain is considered (Figure 10).

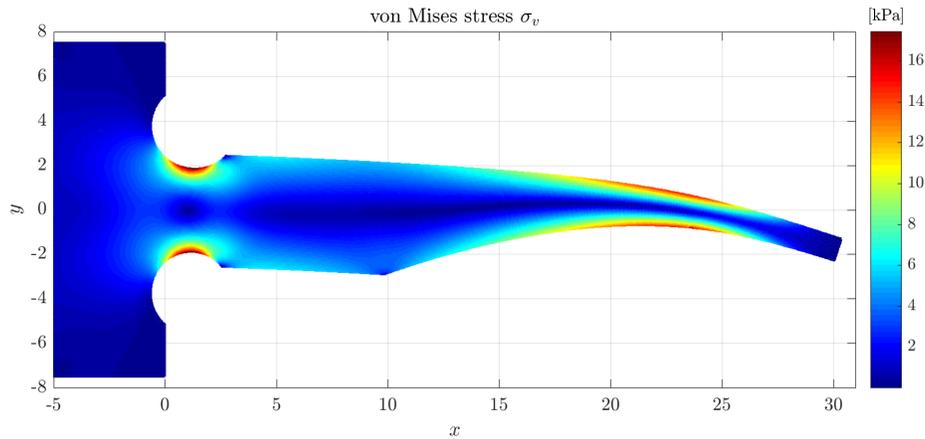


Figure 10: Numerical solution of a irregular cantilever beam using $N = 67887$ nodes. Note that for the sake of visibility the displacements are multiplied by factor 10^5 .

291

292 5.2. Hertzian contact

293 Another more interesting case arises from basic theory of contact mechanics,
 294 called Hertzian contact theory [30]. Consider two cylinders with radii R_1 and
 295 R_2 and parallel axes pressed together by a force per unit length of magnitude
 296 P . The theory predicts they form a small contact surface of width $2b$, where

$$b = 2\sqrt{\frac{PR}{\pi E^*}} \quad (46)$$

and

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}, \quad (47)$$

$$\frac{1}{E^*} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2}. \quad (48)$$

297 Elastic modulus and Poisson's ratio for the first material are denoted with E_1
 298 and ν_1 , and with E_2 and ν_2 for the second material. The pressure distribution
 299 between the bodies along the contact surface is semi-elliptical, i.e. of the form

$$p(x) = \begin{cases} p_0 \sqrt{1 - \frac{x^2}{b^2}}; & |x| \leq b \\ 0; & \text{otherwise} \end{cases}, \quad p_0 = \sqrt{\frac{PE^*}{\pi R}}. \quad (49)$$

300 A problem can be reduced to two dimensions using plane stress assumption. A
 301 special case of this problem is when $E_1 = E_2$, $\nu_1 = \nu_2$ and $R_2 \rightarrow \infty$, describing
 302 a contact of a cylinder and a half plane. This is the second numerical example
 303 tackled in this paper. The setup is ideal for testing the refinement, since a
 304 pronounced difference in behaviour of numerical solution near the contact in
 305 comparison to the rest of the domain is expected.

A displacement field \vec{u} satisfying (15) on $(-\infty, \infty) \times (-\infty, 0)$ with boundary conditions

$$\vec{t}(x, 0) = -p(x)\vec{j} \quad (50)$$

$$\lim_{x, y \rightarrow \infty} \vec{u}(x, y) = 0. \quad (51)$$

is sought. Vector \vec{t} represents traction force on the surface and $\vec{j} = (0, 1)$ the upwards direction. Analytical solution for internal stresses in the plane in general point (x, y) is calculated using the method of complex potentials [31] and the stresses are given in terms of m and n , defined as

$$m^2 = \frac{1}{2} \left(\sqrt{(b^2 - x^2 + y^2)^2 + 4x^2y^2} + b^2 - x^2 + y^2 \right), \quad (52)$$

$$n^2 = \frac{1}{2} \left(\sqrt{(b^2 - x^2 + y^2)^2 + 4x^2y^2} - (b^2 - x^2 + y^2) \right), \quad (53)$$

where $m = \sqrt{m^2}$ in $n = \text{sgn}(x)\sqrt{n^2}$. The stresses are then expressed as

$$\sigma_{xx} = -\frac{p_0}{b} \left[m \left(1 + \frac{y^2 + n^2}{m^2 + n^2} \right) + 2y \right] \quad (54)$$

$$\sigma_{yy} = -\frac{p_0}{b} m \left(1 - \frac{y^2 + n^2}{m^2 + n^2} \right) \quad (55)$$

$$\sigma_{xy} = \sigma_{yx} = \frac{p_0}{b} n \left(\frac{m^2 - y^2}{m^2 + n^2} \right). \quad (56)$$

306 Numerically the problem is solved by truncating the infinite domain to a rect-
 307 angle $[-H, H] \times [-H, 0]$ for large enough H and setting the essential boundary
 308 conditions $\vec{u} = 0$ everywhere but on the top boundary. The top boundary has a
 309 traction boundary condition with normal traction given by $p(x)$ and no tangen-
 310 tial traction. An illustration of the problem domain along with the boundary
 311 conditions is given in Figure 11.

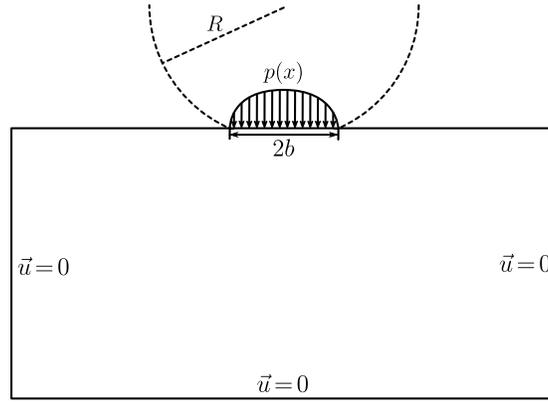


Figure 11: Domain and boundary conditions of considered contact problem.

The described contact problem is solved numerically and the error is mea-
 sured between calculated and given stresses in relative L_∞ norm as before, using

$$e_\infty = \max_{x \in X} \{ \max \{ |\sigma_{xx}(x) - \hat{\sigma}_{xx}(x)|, |\sigma_{yy}(x) - \hat{\sigma}_{yy}(x)|, |\sigma_{xy}(x) - \hat{\sigma}_{xy}(x)| \} \} / p_0$$

312 as an error indicator. Values $P = 543 \text{ N/m}$, $E_1 = E_2 = 72.1 \text{ GPa}$, $\nu_1 = \nu_2 =$
 313 0.33 , $R_1 = R = 1 \text{ m}$ were chosen for the physical parameters of the problem.
 314 These values yield contact half-width $b = 0.13 \text{ mm}$ and peak pressure $p_0 =$
 315 2.6 MPa . A value of $H = 10 \text{ mm}$ for domain height is chosen, approximately 38
 316 times greater than width of the contact surface. Convergence of the method is
 317 shown in Figure 12.

318 It is clear that the convergence of the method is very irregular and slow. This
 319 is to be expected, as $N = 10^6$ means only approximately 30 nodes positioned
 320 within the contact surface, and that naturally leads to large changes as a change
 321 of a single node bears a relatively high influence. Another problem is that the

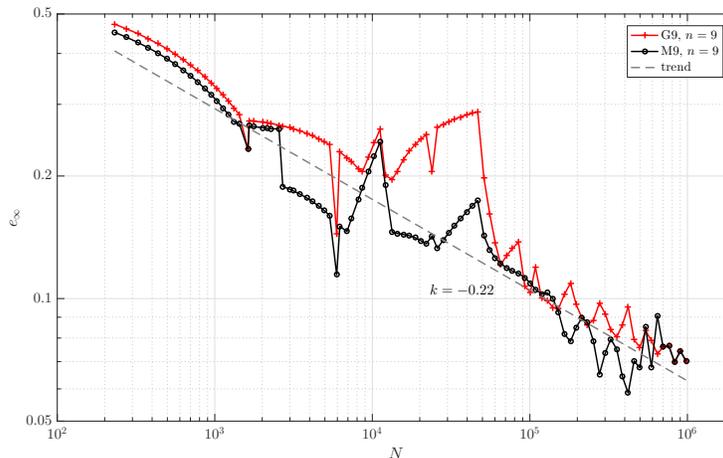


Figure 12: Convergence of MLSM when solving the described Hertzian contact problem.

322 boundary conditions are only continuous, and exhibit no higher regularity, not
 323 even Lipschitz continuity. The accuracy of the approximation may seem bad,
 324 but is in fact comparable to the cantilever beam case. Using the comparable
 325 value of $N = 30 \cdot 15 = 450$ nodes in the contact area $[-b, b] \times [-b, 0]$ it can
 326 be seen from Figure 5 that the approximation using this number of nodes in
 327 cantilever beam case achieved very similar results.

328 The total error of the approximation is composed of two main parts, the
 329 truncation error due to the non-exact boundary conditions and the discretization
 330 error, due to solving a discrete problem instead of the continuous one. First, we
 331 analyse the total error in terms of domain height H . A graph showing the total
 332 error with respect to domain height H is shown in Figure 13.

333 The total error decreases as domain height increases, regardless of the dis-
 334 cretization density used. However, as soon as truncation error becomes lower
 335 than discretization error, increasing the height further yields little to no gain
 336 in total error. The higher the discretization density is, the later this happens.
 337 When convergence of a method stops or significantly decreases in order, an error
 338 limit imposed by the truncation error was reached.

339 It soon becomes impossible to uniformly increase discretization density due

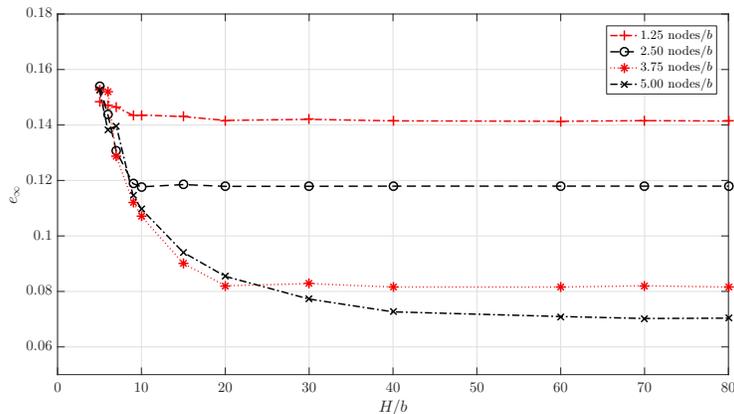


Figure 13: Total error with respect to domain size at different discretization densities.

340 to limited resources, and the immediate solution is to refine the discretization
 341 in the contact area with the h-refinement algorithm introduced in section 4.3.
 342 A domain of height $H = 1 \text{ m} \approx 75\,000b$ is chosen. Primary refinement is done
 343 in rectangle areas of the form

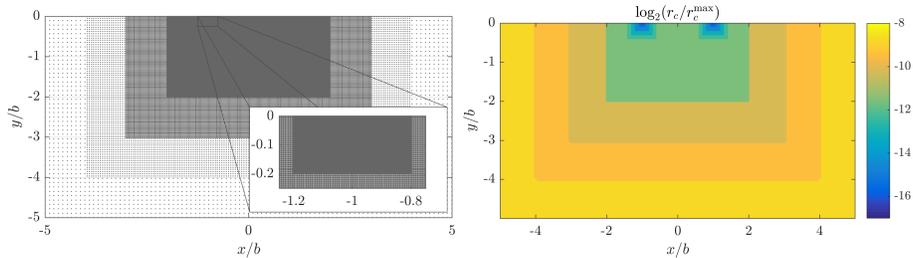
$$[-hb, hb] \times [0, hb], \text{ for } h \in \{1000, 500, 200, 100, 50, 20, 10, 5, 4, 3, 2\},$$

344 and secondary refinement around points $\pm b$ on the surface is done in rectangle
 345 areas

$$[c - hb, c + hb] \times [-hb, 0], \text{ for } c = \pm b \text{ and } h \in \{0.4, 0.3, 0.2, 0.1, 0.05, 0.0025\}.$$

346 The refined domain as described above is shown in Figure 14. This domain
 347 was used to solve the considered contact problem. Different levels of secondary
 348 refinement were tested to prove that refinement helps with accuracy. Conver-
 349 gence of the method on the refined domain is shown in Figure 15.

350 Comparing Figure 15 to Figure 12, it can be seen that refinement greatly
 351 improves the accuracy of the method. Using $N = 10^6$ nodes without refinement
 352 yields worse results than $N = 10^4$ nodes with only primary refinement. Each
 353 additional level of secondary refinement helps to decrease the error even further
 354 while keeping the same order of convergence. A solution of the problem on the
 355 final mesh is shown in Figure 16.



(a) A part of the refined domain. (b) Discretization density of the refined domain.

Figure 14: An example of 17-times refined domain used in solution of the described Hertzian contact problem.

356 6. Conclusions

357 A MLSM solution of a linear elasticity problem on regular and irregular do-
 358 mains with a refined nodal distribution of two different numerical examples is
 359 presented in this paper. The method is analysed in terms of accuracy by com-
 360 parison against available closed form solutions and by comparison against weak
 361 form EFG method. The convergence of the method is evaluated with respect
 362 to the number of computational nodes, selection of different basis functions,
 363 different refinement strategies and different boundary conditions. MLSM is also
 364 analysed from complexity point of view, first, theoretically, and then also ex-
 365 perimentally by timing the computer execution time of all main blocks of the
 366 method. It is clearly demonstrated that the method is accurate and stable.
 367 Furthermore, it is demonstrated that nodal adaptivity is mandatory when solv-
 368 ing contact problems in order to obtain accurate results and that the proposed
 369 MLSM method can handle extensive refinement with the smallest internodal dis-
 370 tance being 2^{17} times smaller than the initial one. It is also demonstrated that
 371 proposed MLSM configuration can handle computations in complex domains.

372 In our opinion the presented meshless setup can be used, not only to solve
 373 academic cases with the sole goal to show excellent convergences, but also in
 374 more complex engineering problems. The C++ implementation of presented

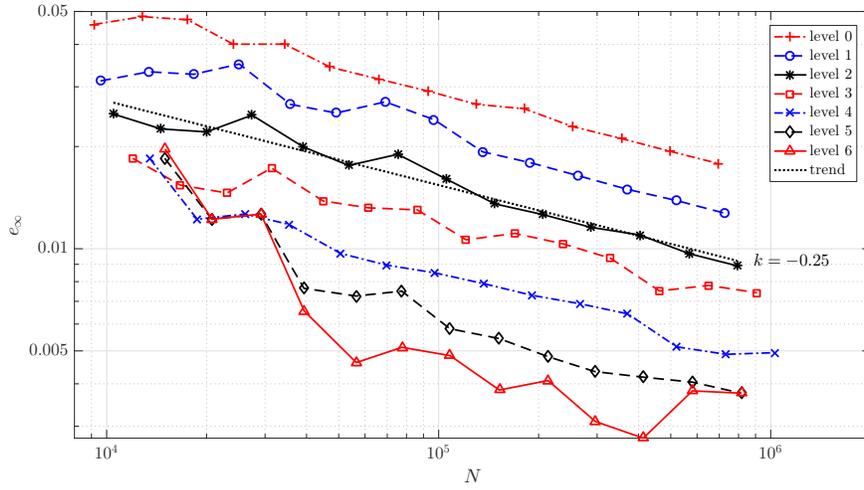


Figure 15: Convergence of MLSM at different refine levels.

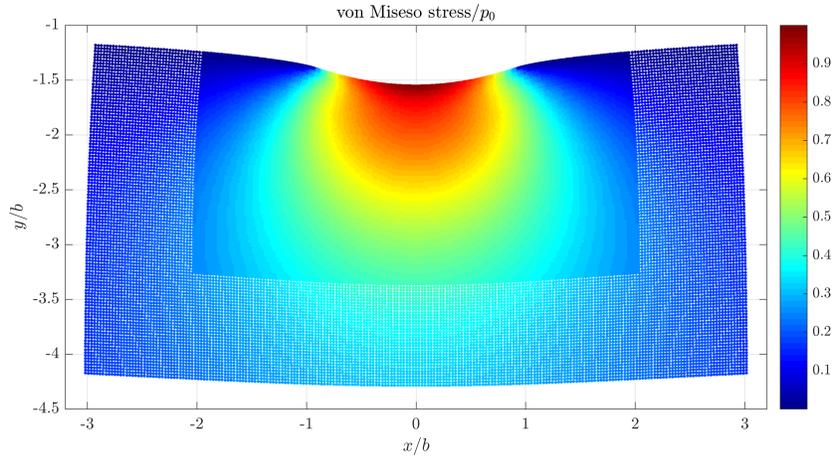


Figure 16: Numerical solution of the described Hertzian contact problem. Note that for the sake of visibility the displacements are multiplied by factor of $5 \cdot 10^3$.

375 MLSM is freely available at [18].

376 In future work we will continue to develop a meshless solution of a contact
 377 problem with a final goal to simulate a crack propagation due to the fretting
 378 fatigue [17] in a general 3D domain with added p-adaptivity to treat singularities
 379 near the crack tip.

380 **Acknowledgement**

381 The authors would like to acknowledge the financial support of the Research
382 Foundation Flanders (FWO), The Luxembourg National Research Fund (FNR)
383 and Slovenian Research Agency (ARRS) in the framework of the FWO Lead
384 Agency project: G018916N Multi-analysis of fretting fatigue using physical and
385 virtual experiments and the ARRS research core funding No. P2-0095.

386 **References**

- 387 [1] O. C. Zienkiewicz, R. L. Taylor, *The Finite Element Method: Solid Me-*
388 *chanics*, Butterworth-Heinemann, 2000.
- 389 [2] J. H. Hattel, P. N. Hansen, A control volume-based finite difference method
390 for solving the equilibrium equations in terms of displacements, *Applied*
391 *Mathematical Modelling* 19 (1994) 210–243.
- 392 [3] Y. D. Freyer, C. Bailey, M. Cross, C.-H. Lai, A control volume procedure for
393 solving the elastic stress-strain equations on an unstructured mesh, *Applied*
394 *Mathematical Modelling* 15 (1991) 639–645.
- 395 [4] Y. Chen, J. D. Lee, A. Eskandarian, *Meshless methods in solid mechanics*,
396 Springer, New York, NY, 2006.
- 397 [5] B. Mavrič, B. Šarler, Local radial basis function collocation method for lin-
398 ear thermoelasticity in two dimensions, *International Journal of Numerical*
399 *Methods for Heat and Fluid Flow* 25 (2015) 1488–1510.
- 400 [6] R. Trobec, G. Kosec, *Parallel scientific computing: theory, algorithms, and*
401 *applications of mesh based and meshless methods*, Springer, 2015.
- 402 [7] G. Kosec, A local numerical solution of a fluid-flow problem on an irregular
403 domain, *Advances in Engineering Software*.
- 404 [8] G. Kosec, R. Trobec, Simulation of semiconductor devices with a local nu-
405 merical approach, *Engineering Analysis with Boundary Elements* 50 (2015)
406 69–75.

- 407 [9] C. A. Wang, H. Sadat, C. Prax, A new meshless approach for three dimen-
408 sional fluid flow and related heat transfer problems, *Computers and Fluids*
409 69 (2012) 136–146.
- 410 [10] S. Li, K. W. Liu, Meshfree and particle methods and their applications,
411 *Applied Mechanics Reviews* 55 (2002) 1–34.
- 412 [11] S. N. Atluri, *The Meshless Method (MLPG) for Domain and BIE Dis-*
413 *cretization*, Tech Science Press, Forsyth, 2004.
- 414 [12] J. Zhang, H. Chen, C. Cao, A graphics processing unit-accelerated mesh-
415 less method for two-dimensional compressible flows, *Engineering Ap-*
416 *plications of Computational Fluid Mechanics* 11 (1) (2017) 526–543.
417 arXiv:<http://dx.doi.org/10.1080/19942060.2017.1317027>, doi:10.
418 1080/19942060.2017.1317027.
419 URL <http://dx.doi.org/10.1080/19942060.2017.1317027>
- 420 [13] W. Li, G. Song, G. Yao, Piece-wise moving least squares ap-
421 proximation, *Applied Numerical Mathematics* 115 (2017) 68–81.
422 doi:10.1016/j.apnum.2017.01.001.
423 URL [http://linkinghub.elsevier.com/retrieve/pii/](http://linkinghub.elsevier.com/retrieve/pii/S0168927417300016)
424 [S0168927417300016](http://linkinghub.elsevier.com/retrieve/pii/S0168927417300016)
- 425 [14] K. Kovářík, S. Masarovičová, J. Mužík, D. Sitányiová, A mesh-
426 less solution of two dimensional multiphase flow in porous media,
427 *Engineering Analysis with Boundary Elements* 70 (2016) 12–22.
428 doi:10.1016/j.enganabound.2016.05.008.
429 URL [http://linkinghub.elsevier.com/retrieve/pii/](http://linkinghub.elsevier.com/retrieve/pii/S0955799716300911)
430 [S0955799716300911](http://linkinghub.elsevier.com/retrieve/pii/S0955799716300911)
- 431 [15] G. Kosec, B. Šarler, H-adaptive local radial basis function collocation mesh-
432 less method, *CMC: Computers, Materials, and Continua* 26 (3) (2011) 227–
433 253.

- 434 [16] W. S. Slaughter, The linearized theory of elasticity, Springer Science &
435 Business Media, 2012.
- 436 [17] K. Pereira, S. Bordas, S. Tomar, R. Trobec, M. Depolli, G. Kosec, M. Ab-
437 del Wahab, On the convergence of stresses in fretting fatigue, *Materials*
438 9 (8) (2016) 639.
- 439 [18] G. Kosec, M. Kolman, J. Slak, Utilities for solving pdes with meshless
440 methods (2016).
441 URL <https://gitlab.com/e62Lab/e62numcodes.git>
- 442 [19] K. Reuther, B. Sarler, M. Rettenmayr, Solving diffusion problems on an
443 unstructured, amorphous grid by a meshless method, *International Journal*
444 *of Thermal Sciences* 51 (2012) 16–22.
- 445 [20] J. Amani, M. Afshar, M. Naisipour, Mixed discrete least squares meshless
446 method for planar elasticity problems using regular and irregular nodal dis-
447 tributions, *Engineering Analysis with Boundary Elements* 36 (2012) 894–
448 902.
- 449 [21] R. Löhner, E. Oñate, A general advancing front technique for filling space
450 with arbitrary objects, *International Journal for Numerical Methods in*
451 *Engineering* 61 (12) (2004) 1977–1991.
- 452 [22] Y. Liu, Y. Nie, W. Zhang, L. Wang, Node placement method by bubble
453 simulation and its application, *CMES: Computer Modeling in Engineering*
454 *& Sciences* 55 (1) (2010) 89.
- 455 [23] N. A. Libre, A. Emdadi, E. J. Kansa, M. Rahimian, M. Shekarchi, A
456 stabilised rbf collocation scheme for neumann type boundary conditions.,
457 *CMES: Computer Modeling in Engineering & Sciences* 24 (2008) 61–80.
- 458 [24] G. C. Bourantas, E. D. Skouras, G. C. Nikiforidis, Adaptive support domain
459 implementation on the moving least squares approximation for mfree meth-
460 ods applied on elliptic and parabolic pde problems using strong-form de-

- 461 description, CMES: Computer Modeling in Engineering & Sciences 43 (2009)
462 1–25.
- 463 [25] T. Belytschko, Y. Y. Lu, L. Gu, Element-free galerkin methods, Int. J.
464 Numer. Methods Eng. 37 (2) (1994) 229–256.
- 465 [26] H. A. Van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant
466 of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal
467 on scientific and Statistical Computing 13 (2) (1992) 631–644.
- 468 [27] Y. Saad, ILUT: A dual threshold incomplete LU factorization, Numerical
469 linear algebra with applications 1 (4) (1994) 387–402.
- 470 [28] G. Kosec, P. Zinterhof, Local strong form meshless method on multiple
471 graphics processing units, CMES: Computer Modeling in Engineering &
472 Sciences 91 (5) (2013) 377–396.
- 473 [29] V. P. Nguyen, T. Rabczuk, S. Bordas, M. Duflot, Meshless methods: a
474 review and computer implementation aspects, Mathematics and computers
475 in simulation 79 (3) (2008) 763–813.
- 476 [30] J. A. Williams, R. S. Dwyer-Joyce, Contact between solid surfaces, Modern
477 tribology handbook 1 (2001) 121–162.
- 478 [31] E. M’Ewen, Stresses in elastic cylinders in contact along a generatrix (in-
479 cluding the effect of tangential friction), The London, Edinburgh, and
480 Dublin Philosophical Magazine and Journal of Science 40 (303) (1949) 454–
481 459. doi:10.1080/14786444908521733.