

Improved MLPG Method for Potential Flow Problem



Rituraj Singh and Roman Trobec

1 Introduction

Many meshless methods have been developed to resolve the problem of re-meshing in conventional numerical methods such as FEM and FDM. The meshless local Petrov–Galerkin (MLPG) method seems to be the most successful meshless method among them [1]. It is based on a local weak form, and numerical integration is done over small local domains. A background mesh is generated locally rather than globally which makes the numerical integration relatively easier than by the global weak form-based meshless methods. The meshless shape functions in the MLPG method can be generated by meshless shape function generation schemes such as moving least squares (MLS), radial basis function (RBF), and several other approaches. The MLS scheme is mostly used due to its accuracy; however, it has some drawbacks such as the absence of Kronecker delta property (KDP) is computationally expensive, uses complicated polynomials, and could be unstable for particular arrangements of nodal points. There is a motivation to further develop the MLPG method as two major problems remain: high computational cost and difficulty in the imposition of essential boundary conditions.

R. Singh (✉)

Department of Mechanical Engineering, National Institute of Technology Jamshedpur,
Jamshedpur 831014, India
e-mail: rituraj.7am@gmail.com

R. Trobec

Department of Communication System, Jozef Stefan Institute, 1000 Ljubljana, Slovenia

2 Literature Review and Objective

Many researchers have reported a slightly improved versions of the MLS scheme which overcome the drawbacks of the original MLS scheme. They have tested these MLS variants in element free Galerkin (EFG) method, global boundary node method (GBNM), and some other meshless methods [2]. To enable the KDP, modified weight functions in the MLS scheme have been proposed [3], which closely satisfy the KDP. Another variant, namely interpolating MLS (IMLS) scheme has been proposed in [4, 5], where a singular weight function is used in the IMLS scheme. It exactly satisfies the KDP, however, on the computational nodes, the value of the singular weight function becomes infinite, and thus, the shape functions cannot be calculated. The IMLS is successful only if the evaluation points do not coincide with the computational nodes. This problem is removed in the improved version of IMLS, named the improved interpolation moving least squares (IIMLS) scheme [6]. In IMLS and IIMLS variants, the inverse of a moment matrix has to be calculated in the procedure of computation of shape functions, which makes the scheme computationally expensive. Another variant of the MLS scheme, which is based on the orthogonal weight function (OMLS), has been proposed in [6]. In this scheme, the moment matrix becomes a diagonal matrix that eliminates the calculation for inversion of moment matrix, which saves a substantial part of computational time.

The objective of this paper is to develop the MLPG method based on the above listed improved versions of MLS scheme and test their performance for the potential flow problem. The tested MLPG methods based on the four improved versions of MLS, i.e. MLS_k (MLS with the modified weight function), IIMLS, OMLS, and OMLS_k (OMLS with modified weight function) are named here: MLPG_k, IIMPLPG, OMLPG, and OMLPG_k, respectively. In the next section, the MLPG formulation for the Laplace equation, which governs the potential flow, is introduced followed by numerical results, discussion, and conclusion.

3 Materials and Methods

In this section, we first present the procedure of MLS and OMLS approaches followed by the formulation of the MLPG method for the Laplace equation. Detailed information on the procedure of shape function generation by the above listed MLS schemes has been discussed in [2].

3.1 MLS Approximation

The MLS approximation for unknown variable $\mathbf{u}(\mathbf{x})$ at an evaluation point \mathbf{x} can be written as

$$u^h(\mathbf{x}) = \sum_{j=1}^m p_j(\mathbf{x})a_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}), \quad (1)$$

where $\mathbf{p}^T(\mathbf{x})$ is a row vector containing m basis functions and $a_j(\mathbf{x})$ are unknown coefficients. The elements of the vector $\mathbf{p}^T(\mathbf{x})$ in 2D can be written as

$$\mathbf{p}^T(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_1x_2, x_2^2]. \quad (2)$$

There are six basis functions ($m = 6$) in 2D. Similarly, the vector can be written for 1D and 3D [2]. The coefficient vector $\mathbf{a}(\mathbf{x})$ is derived by minimising a weighted residual using l_2 norm

$$\sum_{i=1}^{ns} w(\mathbf{x}, \mathbf{x}_i) [u^h(\mathbf{x}, \mathbf{x}_i) - u(\mathbf{x}_i)]^2 = \sum_{i=1}^{ns} w(\mathbf{x}, \mathbf{x}_i) [\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - u(\mathbf{x}_i)]^2, \quad (3)$$

where $w(\mathbf{x}, \mathbf{x}_i)$ is a weight function

$$w(\mathbf{x}, \mathbf{x}_i) = \begin{cases} 1 - 6d^2 + 8d^3 - 3d^4 & \text{if } 0 \leq d \leq 1 \\ 0 & \text{if } d > 1 \end{cases}. \quad (4)$$

$d = \|\mathbf{x} - \mathbf{x}_i\|/r_w$ and r_w is the radius of the support domain in which the weight function is nonzero. There are n_s number of nodes in the support domain, and $u(\mathbf{x}_i)$ is the nodal parameter of the field variable at the discretization node \mathbf{x}_i .

The m linear equations are obtained by differentiating Eq. (3) with respect to coefficient vector $\mathbf{a}(\mathbf{x})$ and equating it to zero

$$\sum_{i=1}^{ns} w(\mathbf{x}, \mathbf{x}_i) [[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - u(\mathbf{x}_i)]]\mathbf{p}(\mathbf{x}_i) = 0, \quad (5)$$

which can be written in matrix form as

$$\mathbf{A}(\mathbf{x})_{m \times m} \mathbf{a}(\mathbf{x})_{m \times 1} = \mathbf{B}(\mathbf{x})_{m \times ns} \mathbf{u}_{ns \times 1}, \quad (6)$$

where

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^{ns} w(\mathbf{x}, \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \quad (7)$$

is defined as the moment matrix, and

$$\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \dots, a_m(\mathbf{x})]^T$$

$$\begin{aligned}\mathbf{B}(\mathbf{x}) &= \sum_{i=1}^{ns} w(\mathbf{x}, \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i), \\ \mathbf{u} &= [\bar{u}_1, \bar{u}_2, \dots, \bar{u}_{ns}]^T,\end{aligned}\quad (8)$$

where \bar{u}_i is the value of the field variable at the discretization point \mathbf{x}_i . The coefficient vector $\mathbf{a}(\mathbf{x})$ is obtained from Eq. (6) and inserted in Eq. (1) to obtain the final MLS approximation

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}) \mathbf{u}. \quad (9)$$

Denoting $\mathbf{p}^T \mathbf{A}^{-1} \mathbf{B}$ by ϕ , we obtain the vector of n_s MLS shape functions. Using a vector notation, the partial derivatives of ϕ can be obtained as

$$\phi_x = \mathbf{p}_x^T (\mathbf{A}^{-1} \mathbf{B}) + \mathbf{p}^T (\mathbf{A}_x^{-1} \mathbf{B} + \mathbf{A}^{-1} \mathbf{B}_x),$$

where

$$\mathbf{A}_x^{-1} = -\mathbf{A}^{-1} \mathbf{A}_x \mathbf{A}^{-1}.$$

To enable KDP, the modified weight function used in [2] can be used in MLS.

$$w(\mathbf{x}, \mathbf{x}_i) = \frac{\tilde{w}(\mathbf{x}, \mathbf{x}_i)}{\sum_{k=1}^{ns} \tilde{w}(\mathbf{x}, \mathbf{x}_k)}, \quad (10)$$

where

$$\tilde{w}(\mathbf{x}, \mathbf{x}_i) = \begin{cases} \frac{(d^2 + \varepsilon^2)^{-2} - (1 + \varepsilon)^{-2}}{\varepsilon^{-2} - (1 + \varepsilon)^{-2}} & \text{if } 0 \leq d \leq 1 \\ 0 & \text{if } d \geq 1 \end{cases}. \quad (11)$$

ε is a small constant, e.g. 10^{-5} [2]. The modified weight function makes the MLS approximation very close to having the KDP.

3.2 OMLS Approximation

The weighted orthogonal basis functions were proposed in the MLS approximation [5], which makes the moment matrix a diagonal matrix. Let, f and g be functions of \mathbf{x} . The inner product $(f \cdot g)$ is defined as

$$(f \cdot g)_x = \sum_{k=1}^{ns} w(\mathbf{x}, \mathbf{x}_k) f(\mathbf{x}_k) g(\mathbf{x}_k), \quad (12)$$

where $w(\mathbf{x}, \mathbf{x}_i)$ is a weight function. The weighted orthogonal basis function is defined as

$$\tilde{p}_i(\mathbf{x}) = p_i(\mathbf{x}) - \sum_{k=1}^{i-1} \frac{(p_i, \tilde{p}_k)}{(\tilde{p}_k, \tilde{p}_k)} \tilde{p}_k, \quad i = 2, 3, \dots, m, \quad (13)$$

where $\tilde{p}_i(\mathbf{x})$ is the new, orthogonal basis function of the OMLS scheme and $p_i(\mathbf{x})$ is the monomial basis function of MLS from Eq. (2). In the present paper, the notations are same as in the MLS procedure, except for \tilde{p} . For example, the first and second elements of the weighted orthogonal basis function ($\tilde{p}_i(\mathbf{x})$) are given as

$$\begin{aligned} \tilde{p}_1(\mathbf{x}) &= 1 \\ \tilde{p}_2(\mathbf{x}) &= x_1 - \frac{(p_2, \tilde{p}_1)}{(\tilde{p}_1, \tilde{p}_1)} p_1 \\ &= x_1 - \sum_{k=1}^{ns} w(\mathbf{x}, \mathbf{x}_k)(x_{1k}) / \sum_{k=1}^{ns} w(\mathbf{x}, \mathbf{x}_k). \end{aligned} \quad (14)$$

In the same way, other weighted orthogonal basis functions for the OMLS can be obtained. The approach for obtaining the OMLS shape functions is the same as in the MLS. The only difference is the diagonal moment matrix (\mathbf{A}), which can be written as.

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} (\tilde{p}_1 \cdot \tilde{p}_1) & 0 & \dots & 0 \\ 0 & (\tilde{p}_2 \cdot \tilde{p}_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (\tilde{p}_m \cdot \tilde{p}_m) \end{bmatrix}. \quad (15)$$

To enable the KDP in the OMLS (OMLS_k), a modified weight function of Eq. (10) can be used.

We further use shifted and scaled polynomials to improve the conditioning of the moment matrix. Its details can be found in [2].

3.3 MLPG Formulation

The Laplace equation governs the potential flow. The global domain and boundary are represented as Ω and Γ , respectively.

$$\begin{aligned} \nabla^2 u &= 0 \quad \text{in } \Omega \\ u(\mathbf{x}) &= \bar{u}(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma_e \\ \partial u(\mathbf{x}) / \partial n &= \bar{q}(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma_n, \end{aligned} \quad (16)$$

u is the potential function, \bar{u} is the specified value of potential function on the global Essential boundary Γ_e , \bar{q} is the specified value on the global Natural boundary Γ_n , where $\Gamma_e \cup \Gamma_n = \Gamma$. The weighted residual statement for the above equation can be written as

$$\int_{\Omega_Q} \nabla^2 u v d\Omega = 0, \quad (17)$$

where Ω_Q is a local quadrature domain and v is the test function. The above equation can be written in the weak form by applying the divergence theorem to reduce the continuity requirement. The equation in the vector notation can be written as

$$\int_{\Gamma_Q} u_{,x} n_x v d\Gamma - \int_{\Omega_Q} u_{,x} v_{,x} d\Omega = 0, \quad (18)$$

where Γ_Q denotes the boundary of Ω_Q . We use vector notation, subscript, x denotes derivative with respect to the spatial coordinate. In the first term, the weak MLPG form can be written for three possible types of Γ_Q

$$\int_{\Gamma_{Qe}} u_{,x} n_x v d\Gamma + \int_{\Gamma_{Qn}} u_{,x} n_x v d\Gamma + \int_{\Gamma_{Qin}} u_{,x} n_x v d\Gamma - \int_{\Omega_Q} u_{,x} v_{,x} d\Omega = 0, \quad (19)$$

where Γ_{Qe} is a part of Γ_Q that intersects Γ_e , Γ_{Qn} is a part of the local domain boundary that intersects Γ_n , and Γ_{Qin} is a part of the local domain boundary that does not intersect Γ . The test function in the MLPG method is chosen so that it vanishes at Γ_{Qin} , then the line integral term for Γ_{Qin} from Eq. (19) becomes zero. A discretization of the MLPG form leads to the system of algebraic equations given by

$$\mathbf{K} \mathbf{u} = \mathbf{F}, \quad (20)$$

where \mathbf{K} is the global system matrix (traditionally named the stiffness matrix), \mathbf{u} is the vector of unknown field variable, and \mathbf{F} is the force vector. The elements of the matrix \mathbf{K} and the vector \mathbf{F} are as follows:

$$K_{ij} = \begin{cases} \phi_j(\mathbf{x}_i) & \mathbf{x}_i \in \Gamma_e \\ \int_{\Gamma_{Qe}} \phi_{j,x}(\mathbf{x}) n_x v(x, \mathbf{x}_i) d\Gamma - \int_{\Omega_Q} \phi_{j,x}(\mathbf{x}) v_{,x}(\mathbf{x}, \mathbf{x}_i) d\Omega & \mathbf{x}_i \notin \Gamma \end{cases}, \quad (21)$$

$$F_i = \begin{cases} \bar{u}(\mathbf{x}_i) & \mathbf{x}_i \in \Gamma_e \\ - \int_{\Gamma_{Qn}} \bar{q} v(\mathbf{x}, \mathbf{x}_i) d\Gamma & \mathbf{x}_i \notin \Gamma \end{cases}. \quad (22)$$

The indices i and j represent computational node number and number of nodes in the support domain of the node i , respectively. In the MLPG method based on MLS and OMLS, the direct interpolation method is used to impose essential boundary condition, and in the MLPG method based on IIMLS, MLS $_k$ and OMLS $_k$, the EBC can be directly imposed, in the same way as in the FEM.

Test Function

The MLPG test function is a quadratic function proposed in [7]

$$v(\mathbf{x}, \mathbf{x}_i) = \begin{cases} \left(1 - \left(\frac{x_1 - x_{1i}}{a_Q(\mathbf{x}_i)/2}\right)\right)^2 \cdot \left(1 - \left(\frac{x_2 - x_{2i}}{a_Q(\mathbf{x}_i)/2}\right)\right)^2 & \text{if } (x_1, x_2) \in \Omega_{Qi} \\ 0 & \text{if } (x_1, x_2) \notin \Omega_{Qi}, \end{cases} \quad (23)$$

where $a_Q(\mathbf{x}_i) = \beta_Q r_s(\mathbf{x}_i)$, $a_Q(\mathbf{x}_i)$ is the edge of the local quadrature domain, β_Q is the proportionality parameter, and $r_s(\mathbf{x}_i)$ is the radius of the support domain of point \mathbf{x}_i .

4 Results and Discussion

Four improved MLPG methods have been developed, namely, MLPG $_k$, IIMLPG, OMLPG, and OMLPG $_k$ based on MLS scheme with modified weight function (MLS $_k$), improved interpolating MLS (IIMLS), MLS based on orthogonal basis function (OMLS) and OMLS with modified weight function (OMLS $_k$), respectively. Their performance is analysed and compared with the original MLPG method, based on a standard MLS scheme, regarding accuracy level, convergence behaviour, and computational speed for the potential flow problem. The MLPG computer code is developed on the C/C++ platform. The Laplace equation has been solved on two-dimensional non-rectangular domain with both EBC and Natural boundary conditions as shown in Fig. 1.

Results of the MLPG code have been verified with the exact analytical solution of the problem which is

$$u = x_2 [1.0 - 1.0 / \{(x_2)^2 - (x_1 - L)^2\}], \quad (24)$$

where u is the potential function, L is the length of the domain, x_1 and x_2 are spatial coordinates. Two error norms have been used in the current work, absolute l_1 norm is defined as

$$\text{Err}_- l_1 = |u_i - u_i^e|, \quad (25)$$

where u_i and u_i^e are approximate and exact values at the evaluation point i , respectively, and l_2 norm is defined as

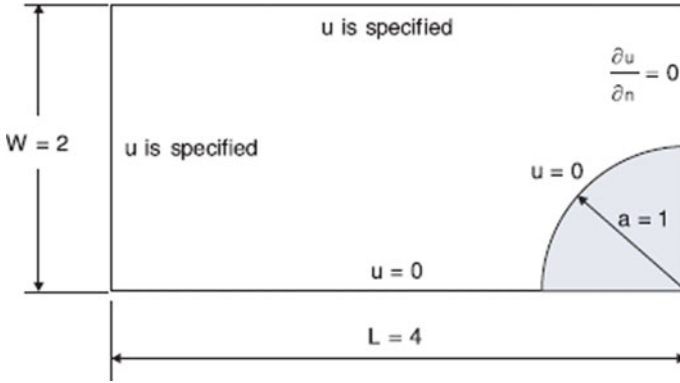


Fig. 1 Schematic diagram of the 2D non-rectangular domain with boundary conditions [8]

$$Rerr_{l_2} = 100 \times \sqrt{\sum_{i=1}^N (u_i - u_i^e)^2} / \sqrt{\sum_{i=1}^N (u_i^e)^2}, \tag{26}$$

where N is the number of discretization nodes.

Often, there are two types of grid, discretisation grid, and evaluation grid. The MLPG solution is obtained on discretised grid points and further re-evaluated on evaluation grid points. In Fig. 2, the distribution of 30 discretised and 114 evaluation grid points in the domain are shown.

Tests have been performed for different sets of parameters such as the different number of nodes in the support domain n_s , different sizes of the local support domain by varying the proportionality parameter and the different number of Gaussian points in the local integration domain. The quadratic basis has been chosen in the MLS variants as it appears to be an optimum. Results obtained with optimised parameters are shown here, i.e. 3×3 Gaussian points in the local integration domain, proportionality

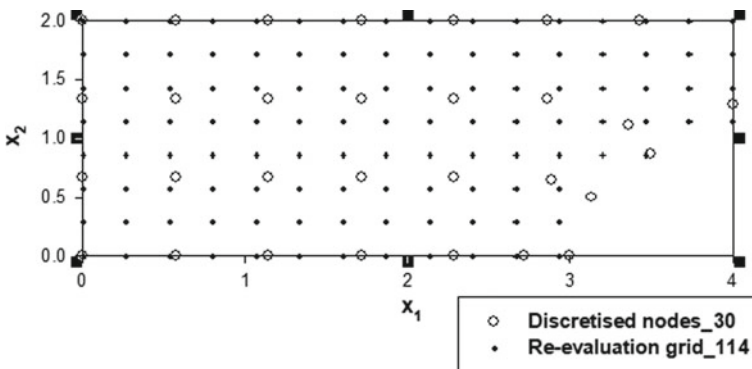


Fig. 2 Schematic diagram of the 2D non-rectangular domain with boundary conditions [2]

parameter $\beta_Q = 0.7$ and $n_s = 13$. First, the MLPG solution is obtained on the discretised grid, and then it is re-evaluated on the evaluation grid. Numerical experiments have been performed for different sizes of grids.

In Fig. 3, a comparison of analytical solution (top) with numerical MLPG solution (bottom) for 168 discretised and 717 evaluation points respectively is shown. Contour plots of constant potential lines are shown in the domain. Satisfactory result is found even for the coarser grid. The solution contour plots of other MLPG variants are visually similar to the MLPG method; thus, they are not shown.

The computational efficiency, of all the selected MLS variants in the MLPG method, relative to the original MLS, has been tested (Fig. 4). The IIMLS method is the most computationally expensive due to its complicated shape function generation procedure. The OMLS method is the fastest and almost 10% more efficient as the original MLS method. As it does not possess KDP, the use of the modified weight function in the OMLS enables KDP, and based on it, the OMLS_k method is

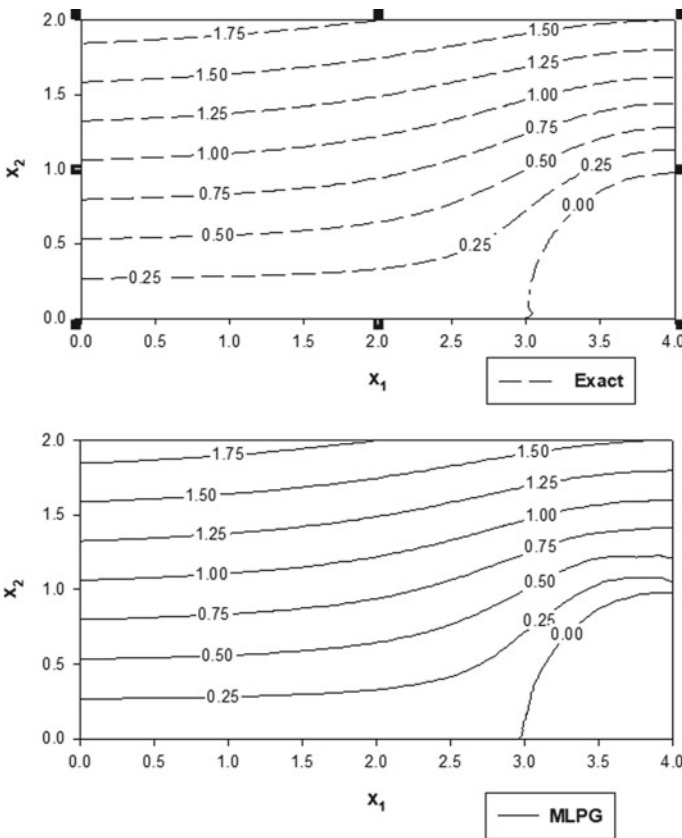


Fig. 3 Comparison of analytical solution (top) with numerical MLPG solution (bottom)

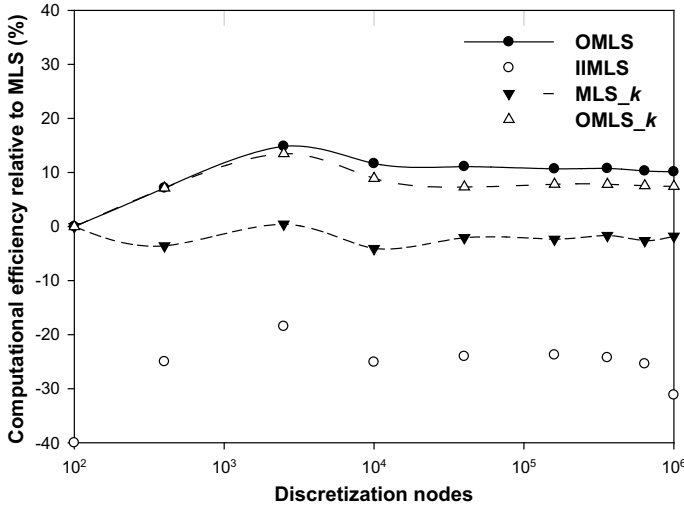


Fig. 4 Computational efficiency of MLPG with different MLS variants relative to MLS [2]

approximately 8% more efficient than the original MLS method. There is the little trade-off between possessing KDP and computational efficiency.

The results of MLPG and OMLPG are identical, and similarly, the results of MLPG_k and OMLPG_k are identical. The OMLPG_k seems to be optimum, and thus, we show only its convergence with the original MLPG method in Fig. 5; however, the IIMLPG method exhibits similar convergence behaviour. The x -axis shows the number of discretised grid points, and the y -axis shows the error. The convergence behaviour of both methods based on the l_2 norm (top) and l_1 norm (bottom) is shown in Fig. 5. Slight waviness in convergence can be seen, nevertheless, good and almost similar convergence behaviour and good accuracy level are found.

The accuracy level and convergence behaviour of all the selected methods are found almost similar and satisfactory. The IIMLPG possesses KDP; however, it is more computationally expensive than MLPG and MLPG_k. The OMLPG is almost 10% faster than the MLPG method; however, it does not possess KDP. The use of modified weight function in the OMLPG method enables KDP at the cost slight decrease in computational speed. Hence, OMLPG_k seems to be the optimum method which resolves both the problems, imposition of essential boundary conditions and moderate computational speed.

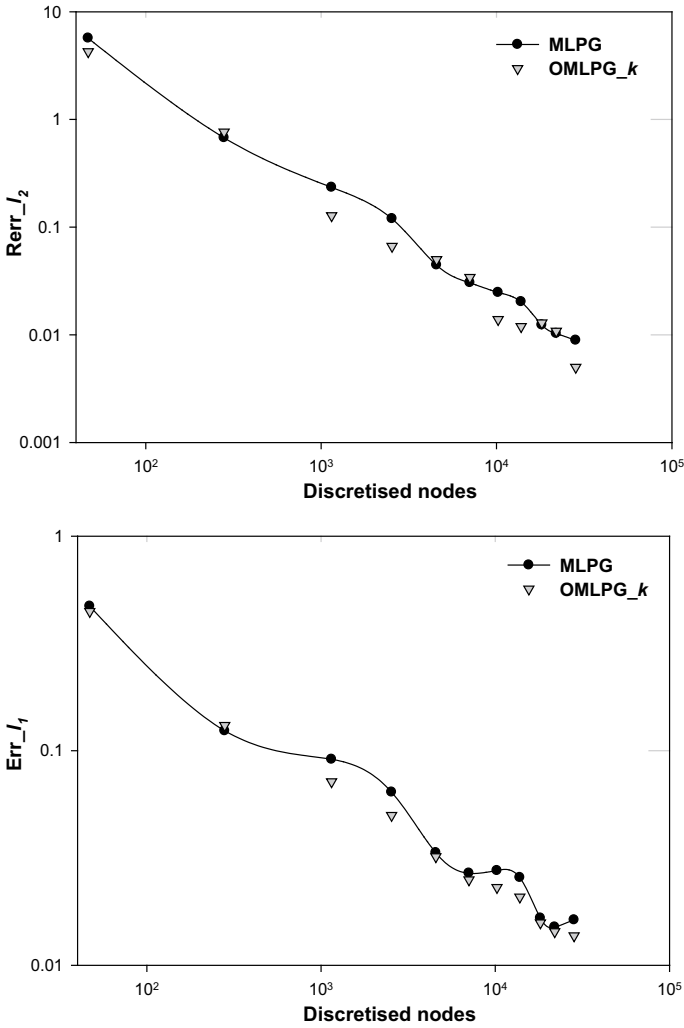


Fig. 5 Convergence of MLPG and OMLPG_k methods based on l_2 norm (top) and l_1 norm (bottom)

5 Conclusions

Four improved MLPG methods have been developed, i.e. MLPG_k, IIMLPG, OMLPG, and OMLPG_k based on MLS scheme with modified weight function (MLS_k), improved interpolating MLS (IIMLS), MLS based on orthogonal basis function (OMLS) and OMLS with modified weight function (OMLS_k), respectively. Their performance is analysed and compared with the original MLPG method based on accuracy level, convergence behaviour, and computational speed for potential flow problem. The accuracy level and convergence behaviour of all the selected

methods are found almost similar and satisfactory. The IIMPLPG possesses KDP; however, it is more computationally expensive than MLPG and MLPG_{*k*}. The OMLPG is almost 10% faster than the MLPG method; however, it does not possess KDP. The use of the modified weight function in the OMLPG method enables KDP at the cost of slight decrease in the computational speed. Hence, OMLPG_{*k*} seems to be the optimum method which resolves both the problems, imposition of essential boundary condition and increase of computational speed.

Acknowledgements The authors acknowledge the financial support from the Slovenian Research Agency under the Grant P2-0095.

Nomenclature

α_Q	Edge of the local quadrature domain
β_Q	Proportionality parameter
ϕ	Shape function
Γ	Global boundary
Γ_e	Global boundary on which essential boundary condition is applied
Γ_n	Global boundary on which natural boundary condition is applied
Ω_Q	Local quadrature domain
v	Test function
n_s	Number of nodes in support domain
r_s	Radius of support domain
m	Basis function
u	Potential function
x	Spatial coordinate
KDP	Kronecker delta property
IIMLS	Improved interpolating moving least squares
MLS	Moving least squares
MLPG	Mehsless local Petrov Galerkin
OMLS	Moving least squares with orthogonal basis function

References

1. Sladek J, Stanak P, Han ZD, Sladek V, Atluri SN (2013) Applications of the MLPG method in engineering & sciences: a review. *Comput Model Eng Sci* 92(5):423–475
2. Singh R, Trobec R (2022) Engineering analysis with boundary elements analysis of the MLS variants in the meshless local Petrov-Galerkin method for a solution to the 2D Laplace equation. *Eng Anal Bound Elem* 135:115–131
3. Most T, Bucher C (2005) A moving least squares weighting function for the elementfree Galerkin method which almost fulfills essential boundary conditions. *Struct Eng Mech* 21:315–332

4. Netuzhylov H (2008) Enforcement of boundary conditions in meshfree methods using interpolating moving least squares. *Eng Anal Bound Elem* 32(6):512–516
5. Ren HP, Cheng YM, Zhang W (2009) An improved boundary element-free method (IBEFM) for two-dimensional potential problems. *Chinese Phys B* 18(10):4065–4073
6. Wang JF, Sun FX, Cheng YM (2012) An improved interpolating element-free Galerkin method with a nonsingular weight function for two-dimensional potential problems. *Chinese Phys B* 21(9):090204
7. Sterk M, Trobec R (2008) Meshless solution of a diffusion equation with parameter optimization and error analysis. *Eng Anal Bound Elem* 32(7):567–577
8. Zhu T, Zhang JD, Atluri SN (1998) A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. *Comput Mech* 21(3):223–235