

# Numerical Integration with Constraints for Meshless Local Petrov-Galerkin Methods

L. Sun<sup>1</sup>, G. Yang<sup>2</sup> and Q. Zhang<sup>3</sup>

**Abstract:** We propose numerical integration rules for meshless local Petrov-Galerkin methods (MLPG) employed to solve elliptic partial differential equations (PDE) with Neumann boundary conditions. The integration rules are required to satisfy an integration constraint condition of Green's formula type (GIC). GIC was first developed in [Babuska, Banerjee, Osborn, and Zhang (2009)] for Galerkin meshless method, and we will show in this paper that it has better features for MLPG due to flexibility of MLPG in choosing different trial and test function spaces. A general constructive algorithm is presented to design the integration rules satisfying GIC. We also present a useful situation, where GIC holds automatically for Gaussian rules. According to this, we conclude that the conical weight is suggested to adopt in MLPG from viewpoint of reducing integration complexity. Approach to extending GIC in [Babuska, Banerjee, Osborn, and Zhang (2009)] to more general elliptic PDE, such as elasticity equation, is discussed. The 1D and 2D numerical results illuminate that GIC reduces the errors in the approximate solutions of MLPG significantly.

**Keywords:** Meshless Petrov-Galerkin method (MLPG), numerical integration, integration constraint, Green's formula, correction algorithm, conical weight.

## 1 Introduction

For last decades, a lot of progress has been made in the development of meshless methods (MM). These methods have been extensively applied to solve engineering problems, especially those where mesh-generation is complicated, e.g., the problems with crack propagation or large deformation. We refer to [Atluri and

---

<sup>1</sup> School of Science, Guangdong University of Petrochemical Technology, Maoming, Guangdong, 525000, China.

<sup>2</sup> School of Computer Science, Zhongyuan University of Technology, Zhengzhou, 450007, China.

<sup>3</sup> Corresponding author. Guangdong Province Key Laboratory of Computational Science and Department of Scientific Computing & Computer Applications, Sun Yat-Sen University, Guangzhou, 510275, China. E-mail: zhangqh6@mail.sysu.edu.cn.

Shen (2002); Atluri (2004); Babuska, Banerjee, and Osborn (2003); Belytschko, Krongauz, Organ, Fleming, and Krysl (1996); Belytschko, Lu, and Gu (1994); Fries and Matthies (2004); Li and Liu (2002); Liu, Jun, and Zhang (1995); Schaback and Wendland (2006); Sladek, Stanak, Han, Sladek, and Atluri (2013)] for various details on MM. There are many classes of MM in practice, such as element-free Galerkin methods (EFGM) [Belytschko, Krongauz, Organ, Fleming, and Krysl (1996); Belytschko, Lu, and Gu (1994)] reproducing kernel particle methods (RKPM) [Liu, Jun, and Zhang (1995)], particle-partition of unity methods (PPUM) [Griebel and Schweitzer (2002)], meshless local Petrov-Galerkin methods (MLPG) [Atluri and Zhu (1998); Atluri and Zhu (2000); Atluri and Shen (2002); Atluri (2004); Sladek, Stanak, Han, Sladek, and Atluri (2013)], generalized finite element method (GFEM) [Babuska, Banerjee, and Osborn (2003)], collocation meshless methods [Schaback and Wendland (2006); Fries and Matthies (2004); Zhang, Dong, Alotaibi, and Atluri (2013)], etc.

Among these methods, MLPG has caught much attention due to its two main advantages: (a) flexibility in designing different trial and test spaces; (b) easy implementation of numerical integration, which could be done in supports of the test shape function without an auxiliary integration mesh (“truly” meshless method). As in all the Galerkin meshless methods (GMM) [Babuska, Banerjee, Osborn, and Li (2008); Babuska, Banerjee, Osborn, and Zhang (2009); Beissel and Belytschko (1996); Carpinteri, Ferro, and Ventura (2002); Chen, Wu, Yoon, and You (2001) De and Bathe (2000); Dolbow and Belytschko (1999); Duan and Belytschko (2009); Fries and Belytschko (2008); Griebel and Schweitzer (2002); Liu and Belytschko (2010); Sze, Chen, Sheng, and Liu (2004); Zhang (2011); Zhang and Banerjee (2012)], numerical integration posed big challenge in MLPG because its shape functions (especially the trial shape functions) are complex or even without explicit expressions. In fact, if numerical integration is not carried out carefully, MLPG may fail to simulate physical phenomena correctly. Lots of brilliant ideas have been developed to deal with the issue, such as [Atluri and Shen (2002); Mazzia, Ferronato, Pini, and Gambolati (2007); Mazzia and Pini (2010); Pecher (2006); Avila, Han, and Atluri (2011); Sellountos, Sequeira, and Polyzos (2010); Zhang, Dong, Alotaibi, and Atluri (2013)]. Moreover, many of integration schemes proposed in GMM [Babuska, Banerjee, Osborn, and Zhang (2009); Liu and Belytschko (2010); Zhang and Banerjee (2012)] could also be introduced in MLPG. In our opinion, however, the issue has not yet been addressed efficiently.

It has been recognized from the latest researches that the integration errors in GMM could be reduced remarkably by certain *integration constraint* (IC) conditions. In [Chen, Wu, Yoon, and You (2001); Sze, Chen, Sheng, and Liu (2004)], an IC condition based on divergence theorem is proposed for nodal integration rules in GMM

to obtain a linear exactness of the approximate solutions. A strain smoothing stabilization scheme is developed to meet the IC condition there. In [Babuska, Banerjee, Osborn, and Li (2008); Zhang (2011)], IC assumes a zero row-sum condition of stiffness matrix to reduce the integration errors of GMM, namely, it requires that sum of the entries of every row in the stiffness matrix is zero. The IC condition of zero row-sum is achieved by correcting the diagonal elements of the stiffness matrix. The divergence free IC is advised for EFGM in [Liu and Belytschko (2010)] based on the so-called *support integration*—the integration rule where numerical integration is carried out on supports of the shape functions. According to the support integration also, a more general IC condition, called as IC of *Green's formula type* (GIC), is developed in [Babuska, Banerjee, Osborn, and Zhang (2009); Zhang and Banerjee (2012)], where the integration rules are required to satisfy certain discrete Green's formulae. The above-mentioned IC in [Chen, Wu, Yoon, and You (2001); Sze, Chen, Sheng, and Liu (2004); Babuska, Banerjee, Osborn, and Li (2008); Zhang (2011); Liu and Belytschko (2010)] could be viewed as the special examples of GIC in some sense. Most importantly, GIC works for moving least-square (MLS) trial functions of any degree  $k$ , where positive integer  $k$  means the degree of polynomials reproduced by the MLS functions; while the other IC conditions in [Chen, Wu, Yoon, and You (2001); Sze, Chen, Sheng, and Liu (2004); Babuska, Banerjee, Osborn, and Li (2008); Zhang (2011); Liu and Belytschko (2010)] only work for  $k = 1$ , i.e., the linear exactness. As pointed out in [Babuska, Banerjee, Osborn, and Li (2008); Babuska, Banerjee, Osborn, and Zhang (2009); Zhang (2011); Zhang and Banerjee (2012)], the integration rules that do not satisfy any IC may cause the “growing up” errors in the approximate solutions of GMM, namely, the errors may increase as spacing of particles decreases. This phenomenon, more or less, interprets why the conventional integration rules, e.g., Gaussian integration or Trapezoidal rule, behave badly in GMM. We mention that convergence of GMM with IC was mathematically analyzed in [Babuska, Banerjee, Osborn, and Li (2008); Babuska, Banerjee, Osborn, and Zhang (2009); Zhang (2011); Zhang and Banerjee (2012)].

To our best knowledge, IC has not been studied in MLPG yet. It is noteworthy that GIC or IC in [Liu and Belytschko (2010)] are set up based on the support integration, which is just one of the major advantages of MLPG. In addition, constructing the integration rules for GIC or IC in [Liu and Belytschko (2010)] depends on the test functions, which are the MLS and RKP functions in EFGM and RKPM respectively, both are difficult to evaluate. On the other hand, it is possible for MLPG to choose simple test functions such that the construction of the integration rules for GIC get easier. Due to these two aspects, designing the integration rules with IC for MLPG is more promising. In this paper, we propose the integration al-

gorithms with GIC in [Babuska, Banerjee, Osborn, and Zhang (2009); Zhang and Banerjee (2012)] for MLPG, applied to solve elliptic PDE with Neumann boundary conditions. The trial functions in MLPG is the moving least-square (MLS) basis functions reproducing polynomials of degree  $k$ , where  $k$  is an arbitrary positive integer; while the weight functions, used to construct the MLS functions, serve as the test shape functions. This class of MLPG is referred to as MLPG1 in the literature [Atluri and Shen (2002); Mazzia and Pini (2010); Pecher (2006)], which is the most popular MLPG in practice. We present a general constructive algorithm to correct the conventional integration rules (such as Gaussian rules, Trapezoidal rule, even Riemann sum rule) to satisfy GIC. This correction algorithm is designed according to the weight functions and does not increase extra computation complexity. When the conical weight with rectangular support is applied in MLPG, GIC will hold for Gaussian rules automatically. This nice feature is attributed to the flexibility of MLPG in choosing the test functions and never occurs in EFGM or RKPM. Therefore, we suggest to make use of the conical weight in MLPG from viewpoint of reducing integration complexity. We also discuss approach to extending GIC in [Babuska, Banerjee, Osborn, and Zhang (2009); Zhang and Banerjee (2012)] (for Poisson equation) to other general elliptic PDE, by taking a plane elasticity problem as example. We numerically investigate effects of GIC for  $k = 1, 2$  in 1D and 2D problems. Without satisfaction of GIC, the errors in the approximate solutions of MLPG may grow up as the particle density is higher; while GIC reduces the errors in MLPG significantly. Ratios of the errors in MLPG satisfying GIC and not satisfying GIC are between 0.011% and 2.3% at the highest particle density, and between 9.4% and 50% at the lowest particle density. In particular, the advantages of GIC in MLPG are more notable for the higher exactness  $k$  and particle density.

We organize the paper as follows. We first introduce a model problem and its local Petrov-Galerkin weak formulations in Section 2. The trial and test spaces of MLPG are reviewed in Section 3. Section 4 is the main part of the paper where we propose GIC for MLPG and present a constructive algorithm to satisfy GIC and a useful situation that GIC holds automatically for Gaussian rules. We also discuss approach to extending GIC to other general elliptic PDE there. The numerical experiments and conclusion are presented in Section 5 and 6, respectively.

## 2 Model problem and local Petrov-Galerkin formulation

For a domain  $D \subset \mathbb{R}^d$ , we denote by  $L^2(D)$ ,  $C(D)$ ,  $C^k(D)$  the spaces of square integrable, continuous and  $k$ -continuous functions, respectively. Polynomial space of degree  $k$  is denoted by  $\mathcal{P}_k$ . Below, the letter  $k$  will always mean the degree of polynomials reproduced by the MLS functions.

Let  $\Omega \subset \mathbb{R}^d$  be a bounded domain with Lipschitz continuous boundary  $\Gamma := \partial\Omega$ .

For the model problem, we consider the Neumann problem

$$\begin{aligned} -\nabla \cdot (A \nabla u) + cu &= f, & \text{in } \Omega \\ A \nabla u \cdot \vec{n} &= g, & \text{on } \Gamma \end{aligned} \tag{1}$$

where  $A(x) = [a_{ij}(x)]_{1 \leq i, j \leq d}$  is a symmetric matrix-valued function,  $a_{ij} \in C^k(\bar{\Omega})$ ,  $c \in C(\bar{\Omega})$ ,  $f \in L^2(\Omega)$ ,  $g \in L^2(\Gamma)$  and  $\vec{n}$  is the outward unit normal vector to  $\Gamma$ . We assume that there is a constant  $\beta > 0$  such that

$$\sum_{i,j=1}^d u_i a_{ij}(x) u_j \geq \beta \sum_{i=1}^d u_i^2, \forall u \in \mathbb{R}^d \text{ and } c(x) \geq \beta, \quad \forall x \in \Omega.$$

**Remark 2.1** *To highlight the integration algorithms, we consider natural boundary conditions because essential boundary conditions in MM need to be imposed with care [Fernandez and Huerta (2004); Babuska, Banerjee, and Osborn (2003)]. We have in [Zhang (2014)] investigated a Nitsche’s approach to enforce the essential boundary conditions, where the integration algorithms in this article can apply without difficulty. Furthermore, the ideas developed here can be extended to other elliptical PDE, for instance, the elasticity problem, see Section 4.4 below. ■*

Instead of the global weak forms on  $\Omega$  in the conventional Galerkin methods, MLPG employ the local variational formulations to solve the problem (1) [Atluri and Zhu (1998); Atluri and Zhu (2000); Atluri and Shen (2002); Atluri (2004)]. Assume  $\Omega_i$  is a sub-domain of  $\Omega$  and  $v_i$  is a test function associated to it. Multiplying  $v_i$  on both sides of (1) and integrating on  $\Omega_i$ , we have

$$\int_{\Omega_i} (-\nabla \cdot (A \nabla u) + cu) v_i dx = \int_{\Omega_i} f v_i dx.$$

Using divergence theorem and the boundary condition in (1), we get

$$\begin{aligned} \int_{\Omega_i} (A \nabla u \cdot \nabla v_i + cu v_i) dx &= \int_{\Omega_i} f v_i dx + \int_{\partial \Omega_i} A \nabla u \cdot \vec{n}_i v_i ds \\ &= \int_{\Omega_i} f v_i dx + \int_{\Gamma_i} g v_i ds + \int_{L_i} A \nabla u \cdot \vec{n}_i v_i ds, \end{aligned} \tag{2}$$

where  $\Gamma_i := \partial \Omega_i \cap \Gamma$ ,  $L_i := \partial \Omega_i \setminus \Gamma_i$ , and  $\vec{n}_i$  is the outward unit normal vector to  $\partial \Omega_i$ . The formulation (2) is referred to as the local Galerkin weak formulation. MM that approximate (2) using different trial and test spaces are referred to as meshless Petrov-Galerkin methods (MLPG). Different choices of the test function  $v_i$  lead to

many variants of MLPG. If the test function  $v_i$  is chosen to have compact support such that  $v_i = 0$  on  $L_i$ , then (2) is reduced to

$$\int_{\Omega_i} (A \nabla u \cdot \nabla v_i + c u v_i) dx = \int_{\Omega_i} f v_i dx + \int_{\Gamma_i} g v_i ds \quad (3)$$

MLPG based on (3) is called as MLPG1, which is the original version of MLPG [Atluri and Zhu (1998)]. If  $v_i$  is taken as constant 1, (2) becomes

$$-\int_{L_i} A \nabla u \cdot \vec{n}_i ds + \int_{\Omega_i} c u dx = \int_{\Omega_i} f dx + \int_{\Gamma_i} g ds,$$

which leads to MLPG5. MLPG1 and MLPG5 seem to be most promising in the literature. Without carrying out volume integrations in stiffness matrix, MLPG5 have less computation complexity. However, MLPG5 are less stable than MLPG1 in a sense that its stiffness matrix may be diagonally less dominant than those of MLPG1 and cause serious computational difficulties in solving linear systems [Mazzia, Ferronato, Pini, and Gambolati (2007); Mazzia and Pini (2010)]. Therefore, we focus on MLPG1 (3) in this paper. Below, we assume that  $v_i$  has compact support and vanishes on  $L_i$  so that we work with the variational formulation (3) in stead of (2). We mention that it is not a restrict requirement and can be implemented easily. As we will see in next sections,  $v_i$  is generally taken as the weight function of the MLS scheme, and  $\Omega_i$  is the intersection of  $\Omega$  and support of  $v_i$  so that the requirement is fulfilled naturally.

### 3 Moving least-square schemes and discretization

In this section, we specify the trial and test functions and the subdomains  $\Omega_i$  in meshless approach to discretize the weak formulation (3). There are several schemes in the literature, for instance, moving least-square method (MLS) [Belytschko, Lu, and Gu (1994)], reproducing kernel particle method (RKPM) [Babuska, Banerjee, and Osborn (2003); Liu, Jun, and Zhang (1995)], partition of unity (PU) [Babuska, Banerjee, and Osborn (2003); Griebel and Schweitzer (2002)], etc. In these methods, the shape functions are produced through a set of particles scattered in the domain, therefore, MM are also referred to as particle methods. In this paper, the MLS approach [Belytschko, Lu, and Gu (1994)] is employed to produce the trial space; while the weight functions used to construct the MLS functions serve as the test functions.

#### 3.1 Moving least-square (MLS) method

For a positive integer  $N$ , denote by  $I_N$  to be the number set  $\{1, 2, \dots, N\}$ . Let  $X_N = \{z_i : i \in I_N\}$  be a set of particles scattered in  $\Omega$ , ordered by the index set  $I_N$ .

Every particle  $z_i$  is associated to a weight (or window) function  $w_i(x) \geq 0$  that is of compact support and centered at  $x_i$ . In general,  $w_i(x)$  can be obtained through dilation and translation of a master weight function  $w(x)$ , namely,

$$w_i(x) = w\left(\frac{x - z_i}{h_i}\right), \tag{4}$$

where  $h_i$  is a dilation parameter to adjust size of the support of  $w_i(x)$ . In 1D, several common weight functions  $w(x)$  apply, for instance,

- (a) Gaussian:  $w^1(x) = \begin{cases} e - e^{(\frac{x}{R})^2}, & |x| \leq R \\ 0, & |x| > R, \end{cases}$
- (b) cubic spline:  $w^1(x) = \begin{cases} \frac{2}{3} - 4(\frac{x}{R})^2 + 4(\frac{|x|}{R})^3, & |x| \leq \frac{R}{2} \\ \frac{4}{3} - 4\frac{|x|}{R} + 4(\frac{x}{R})^2 - \frac{4}{3}(\frac{|x|}{R})^3, & \frac{R}{2} < |x| \leq R \\ 0, & |x| > R, \end{cases}$
- (c) Wendland function:  $w^1(x) = \begin{cases} (1 - \frac{|x|}{R})^5 [8(\frac{x}{R})^2 + 5\frac{|x|}{R} + 1], & |x| \leq R \\ 0, & |x| > R, \end{cases}$
- (d) conical:  $w^1(x) = \begin{cases} [(1 - \frac{|x|}{R})^2]^l, & |x| \leq R \\ 0, & |x| > R, \end{cases} \quad l = 1, 2, \dots,$

where  $R > 0$  is a scaling constant adjusting the support of  $w^1(x)$ . In high dimension  $\mathbb{R}^d$ , the weight function  $w(x)$  can be constructed through the 1D weight function  $w^1(x)$  as follows:

$$w(x) = w^1(\|x\|) \text{ or } w(x) = \prod_{i=1}^d w^1(x_i), \tag{5}$$

where  $x = [x_i]_{i=1}^d$  is Cartesian coordinates in  $\mathbb{R}^d$  and  $\|x\|$  is Euclidean distance of  $x$  and the origin. We note that the former in (5) is radial and has circle and sphere supports in 2D and 3D respectively; while the later is the tensor product of  $w^1$  and has square and cubic supports in 2D and 3D respectively.

For every  $i \in I_N$ , let  $\{p_i^s : s \in I_Q\}$  be a basis of polynomial space  $\mathcal{P}_k$ . For a given point  $x \in \Omega$ , the MLS function  $\phi_i(x)$ , associated to  $x_i$ , is defined by

$$\phi_i(x) = w_i(x) \sum_{s \in I_Q} a_s(x) p_i^s(x), \tag{6}$$

where  $[a_s]_{s \in I_Q}$  are the undetermined coefficients, which are determined by imposing the so-called property reproducing polynomials of degree  $k$  of  $\phi_i$ :

$$\sum_{i \in I_N} p(x_i) \phi_i(x) = p(x), \quad x \in \Omega \quad p \in \mathcal{P}_k. \tag{7}$$

This gives rise to a linear system in  $[a_s]_{s \in I_Q}$ , namely,

$$\sum_{s \in I_Q} \left[ \sum_{i \in I_N} w_i(x) p_i^s(x) p_i^t(x) \right] a_s(x) = p_i^t(x), t \in I_Q. \tag{8}$$

A sufficient condition that the system (8) is solvable is that the particles  $\{z_i \in X_N : x \in \text{supp } w_i\}$  are  $\mathcal{P}_k$ -unisolvent. The functions  $\phi_i$  in (6) are referred to as the MLS functions, which are employed to be the trial functions in this paper. We note from (6) that the function  $\phi_i$  possesses the same support as the weight function  $w_i$ . Denote  $\omega_i = \text{supp } \phi_i \cap \Omega$  that is the support of  $\phi_i$  constrained in  $\Omega$ .

### 3.2 Discretization

Assume  $u_N = \sum_{j \in I_N} c_j \phi_j$  is an approximate solution to the exact solution  $u$  of (1), then replacing  $u$  by  $u_N$  in the formulation (3), we get an equation

$$\int_{\Omega_i} (A \nabla u_N \cdot \nabla v_i + c u_N v_i) dx = \int_{\Omega_i} f v_i dx + \int_{\Gamma_i} g v_i ds, \tag{9}$$

that gives an linear equation with the unknowns  $c_j$  as follows:

$$\sum_{j=1}^N (\gamma_{ij} + \sigma_{ij}) c_j = f_i + g_i \tag{10}$$

where

$$\gamma_{ij} := \int_{\Omega_i} A \nabla \phi_j \cdot \nabla v_i dx, \sigma_{ij} := \int_{\Omega_i} c \phi_j v_i dx, f_i := \int_{\Omega_i} f v_i dx, \text{ and } g_i := \int_{\Gamma_i} g v_i ds. \tag{11}$$

We note that each pair of  $\Omega_i$  and  $v_i$  is associated to a linear equation (10). To get a square linear system with the unknowns  $c_j$ ,  $N$  pairs of  $\Omega_i$  and  $v_i, i \in I_N$ , are needed. In practice MLPG1 take  $[v_i]_{i \in I_N}$  to be the weight functions  $w_i$  used to produce  $\phi_i$  (see (6)). But, the support of  $v_i$  could be different from  $w_i$ , specifically, we assume that

$$v_i(x) = w \left( \frac{x - z_i}{r_i} \right), \tag{12}$$

where  $r_i$  is a parameter other than the one in (4). For every  $i \in I_N$ , let  $\Omega_i = \text{supp } v_i \cap \Omega$  that is the support of  $v_i$  constrained in  $\Omega$ . With such  $v_i$  and  $\Omega_i$ , we get the linear system (10), where  $[\gamma_{ij}]$ ,  $[\sigma_{ij}]$ ,  $[f_i]$ , and  $[g_i]$ , defined in (11), are called as the stiffness matrix, mass matrix, volume load vector, boundary load vector, respectively.

Theoretically, to ensure a reliable approximate solution from the associated discretization problem, the union of all sub-domains  $\Omega_i$  should cover the domain  $\Omega$ , namely,

$$\Omega = \bigcup_{i \in I_N} \Omega_i.$$

In our numerical implementation, however, we find out that an incomplete covering could also provide the numerical solutions of high precision, see also [Atluri and Zhu (1998)].

**Remark 3.1** *The parameters  $h_i$  and  $r_i$  have significant influence on MLPG, and it is not easy to get optimal size of these parameters [Nie, Atluri, and Zuo (2006)]. This, however, is not a task in this paper. Below, we always employ a simple relationship  $h_i = r_i$  to present our integration algorithm. We stress that this simplicity does not cause any extra trouble. In our computation, we find out that the results for the different relationships turn out to behave similarly. ■*

For convenience, we denote by  $I'_N$  the index set of those weight functions  $v_i$  such that  $\text{supp } v_i \cap \Omega = \emptyset$ . The particles indexed by  $I'_N$  are called as *interior particles*, and the weight functions  $v_i$ ,  $i \in I'_N$  are referred to as *interior weight functions*. It is obvious that  $\Omega_i = \text{supp } v_i$  for the interior weight functions.

#### 4 Numerical integration for MLPG

The definite integrals in the linear system (10) have to be computed numerically during implementation, and we solve an approximate system

$$\sum_{j \in I_N} (\gamma_{ij}^* + \sigma_{ij}^*) c_j = f_i^* + g_i^*, \quad \forall i \in I_N, \tag{13}$$

where,  $\gamma_{ij}^*$ ,  $\sigma_{ij}^*$ ,  $f_i^*$ , and  $g_i^*$  are numerical approximations of  $\gamma_{ij}$ ,  $\sigma_{ij}$ ,  $f_i$ , and  $g_i$ , respectively, via numerical integration. From (11), we naturally define

$$\gamma_{ij}^* := \int_{\Omega_i}^s A \nabla \phi_j \cdot \nabla v_i dx, \sigma_{ij}^* := \int_{\Omega_i}^m c \phi_j v_i dx, f_i^* := \int_{\Omega_i}^f f v_i dx, \text{ and } g_i^* := \int_{\Gamma_i}^g g v_i ds, \tag{14}$$

where  $f_D^s$ ,  $f_D^m$ ,  $f_D^f$ , and  $f_{\partial D}^g$  mean numerical integration formulae for the stiffness matrix, mass matrix, volume load vector, and boundary load vector, on a set  $D$ , respectively.

**Remark 4.1** We note that for every  $i$ , the integrations for  $\gamma_{ij}^*$ ,  $\sigma_{ij}^*$  are carried out on  $\Omega_i$  and  $\Gamma_i$  for all  $j \in \{j \in I_N : \omega_j \cap \Omega_i \neq \emptyset\}$ , i.e., we use the same integration formulae for the  $i$ th line in the stiffness and mass matrices in (13). Recall that  $\Omega_i$  is the support of the weight function (12), which possesses regular shape. Therefore, a background mesh is not necessary to implement numerical integrations. This is the reason that MLPG is known as “truly” meshless method. We emphasize that such integration schemes are also employed in EFGM recently, which are called as the support integration [Liu and Belytschko (2010)], see [Babuska, Banerjee, and Osborn (2003); Zhang and Banerjee (2012); Liu and Belytschko (2010)] in detail.

■

**4.1 Integration constraint in MLPG, based on Green’s formula**

Now, we present the main assumptions for the integration formulae.

**IC1.** For every  $i \in I_N$ , the integration formulae on  $\Omega_i$  and  $\Gamma_i$  are required to satisfy the following condition:

$$\int_{\Omega_i} \tilde{p} \cdot \nabla v_i dx + \int_{\Omega_i} \nabla \cdot \tilde{p} v_i dx - \int_{\Gamma_i} \tilde{p} \cdot \tilde{n} v_i ds = 0, \forall \tilde{p} \in \mathcal{P}_{k-1}^d, \tag{15}$$

where  $\tilde{p} = [p_i]_{i=1}^d$  is a vector-valued polynomial function with its components  $p_i \in \mathcal{P}_{k-1}$ ,  $i = 1, 2, \dots, d$ .

**Remark 4.2** We note that when the numerical integrations in (15) are exact, (15) will hold automatically according to Green’s formula. Therefore, we refer the integration rule (15) to as integration constraint of Green’s formula-type (GIC). It is noteworthy that the conventional integration rules, such as Gaussian or Trapezoidal rules, do not satisfy GIC. This provides a possible interpretation why the conventional rules work badly in MLPG (also in other MM [Babuska, Banerjee, Osborn, and Zhang (2009); Mazzia, Ferronato, Pini, and Gambolati (2007); Zhang (2011); Zhang and Banerjee (2012)]). We will show this phenomenon in the numerical experiments later.

■

**IC2.** For every  $i \in I_N$ , we assume

$$\int_{\Omega_i}^m = \int_{\Omega_i}^f. \tag{16}$$

**Remark 4.3** The condition **IC2** is easy to satisfy so long as for every  $\Omega_i$  we employ the same integration formula to compute the entries  $\sigma_{ij}^*$  in  $i$ th line of the mass matrix and the  $i$ th component  $f_i^*$  of the volume load vector. Hence, GIC (15) is the main

condition. Recently, It has been realized in MM (mainly, EFGM) that IC properly imposed will reduce the integration errors significantly. In [Babuska, Banerjee, Osborn, and Li (2008); Zhang (2011)], the zero row-sum IC is developed:

$$\sum_{j \in I_N} \gamma_{ij}^* = 0, \forall i \in I_N; \tag{17}$$

In the stabilized conformal nodal integration (SCNI) [Chen, Wu, Yoon, and You (2001); Sze, Chen, Sheng, and Liu (2004)] and the support integration [Liu and Belytschko (2010)], IC is employed in the form:

$$\int_{\Omega_i} \nabla \phi_i dx = \int_{\Gamma_i} \vec{n} \phi_i ds; \tag{18}$$

In [Babuska, Banerjee, Osborn, and Zhang (2009); Zhang and Banerjee (2012)], (15) with  $v_i$  replaced by  $\phi_i$  are used in the situation of EFGM. We mention that these IC could all be viewed as the special cases of (15). In particular, (17) will hold automatically by the support integration scheme (14); (18) is GIC (15) in the case  $k = 1$ ; while the GIC in [Babuska, Banerjee, Osborn, and Zhang (2009); Zhang and Banerjee (2012)] is the special situation of (15) where the test function  $v_i$  is taken as the trial function  $\phi_i$ . We emphasize that applying GIC in MLPG is more advantageous than in EFGM because the choices of the test function  $v_i$  are more flexible in MLPG so that either construction of GIC becomes more simple or GIC will hold automatically by Gaussian rules when  $v_i$  is chosen to be polynomial weight function (such as the conical function). We will specify it below. ■

**Remark 4.4** To understand (15) intuitively, we present its simple cases:

(a) 1D case. Let  $\Omega_i = [a_i, b_i]$ , then for  $k = 1$ , replacing  $\tilde{p} = 1$  in (15), we have

$$\int_{[a_i, b_i]} v_i' dx = v_i(b_i) - v_i(a_i). \tag{19}$$

For  $k = 2$ , replacing  $\tilde{p} = 1$  and  $x$  in (15), we get (19) and

$$\int_{[a_i, b_i]} xv_i' dx = b_i v_i(b_i) - a_i v_i(a_i) - \int_{[a_i, b_i]} v_i dx, \tag{20}$$

therefore, for  $k = 1$ , the integration rules must satisfy (19), while for  $k = 2$ , the integration rules must satisfy (19) and (20).

(b) 2D case. For  $k=1$ , replacing  $\tilde{p} = [1, 0]$  and  $[0, 1]$  in (15), we have

$$\int_{\Omega_i} \frac{\partial v_i}{\partial x_1} dx = \int_{\Gamma_i} n_1 v_i ds \text{ and } \int_{\Omega_i} \frac{\partial v_i}{\partial x_2} dx = \int_{\Gamma_i} n_2 v_i ds, \tag{21}$$

where  $\vec{n} = [n_1, n_2]^T$ . For  $k = 2$ , replacing  $\tilde{p} = [1, 0], [0, 1], [x_1, 0], [0, x_1], [x_2, 0],$  and  $[0, x_2]$ , we get (21) and

$$\int_{\Omega_i}^s x_1 \frac{\partial v_i}{\partial x_1} dx = \int_{\Gamma_i}^g x_1 n_1 v_i ds - \int_{\Omega_i}^f v_i dx, \quad \int_{\Omega_i}^s x_1 \frac{\partial v_i}{\partial x_2} dx = \int_{\Gamma_i}^g x_1 n_2 v_i ds, \quad (22)$$

$$\int_{\Omega_i}^s x_2 \frac{\partial v_i}{\partial x_1} dx = \int_{\Gamma_i}^g x_2 n_1 v_i ds, \quad \int_{\Omega_i}^s x_2 \frac{\partial v_i}{\partial x_2} dx = \int_{\Gamma_i}^g x_2 n_2 v_i ds - \int_{\Omega_i}^f v_i dx, \quad (23)$$

therefore, for  $k = 1$ , the integration rules must satisfy (21), while for  $k = 2$ , the integration rules must satisfy (21), (22), (23). ■

### 4.2 Construction for GIC

We present a constructive algorithm to fulfill the conditions **IC1** and **IC2** (mainly GIC). Firstly, for every  $i \in I_N$  we employ the conventional integration rules (such as Gaussian rules) for the volume load integration  $f_i^*$  (also for the entries  $\sigma_{ij}^*$  of the mass matrix such that **IC2** holds) and the boundary load integration  $g_i^*$ , respectively. Secondly, we correct the integration rules for the entries  $f_{\Omega}^s$  in the stiffness matrix to satisfy GIC (15).

Let  $\tilde{p}_r, r \in I_M$  be a basis of polynomials space  $\tilde{\mathcal{P}}_{k-1}$ , where  $M$  is dimensionality of  $\tilde{\mathcal{P}}_{k-1}$ . For instance, in 1D,  $M = 1$  for  $k = 1$  and  $M = 2$  for  $k = 2$ ; while in 2D  $M = 2$  for  $k = 1$  and  $M = 6$  for  $k = 2$ . In general,  $M = d \times C_{k+d-1}^{k-1}$ . Then, (15) is equivalent to

$$\int_{\Omega_i}^s \tilde{p}_r \cdot \nabla v_i dx = - \int_{\Omega_i}^f \nabla \cdot \tilde{p}_r v_i dx + \int_{\Gamma_i}^g \tilde{p}_r \cdot \vec{n} v_i ds, \quad \forall r \in I_M, \quad (24)$$

For every  $i \in I_N$ , let  $Q_i^s$  and  $Q_i^f$  be integration rules on  $\Omega_i$  for the entries in the stiffness matrix and the volume load vector, respectively, and assume  $Q_i^g$  is an integration rule on  $\Gamma_i$  for the boundary load integration, namely,

$$Q_i^s(\xi) = \int_{\Omega_i}^s \xi dx, \quad Q_i^f(\xi) = \int_{\Omega_i}^f \xi dx, \quad \text{and} \quad Q_i^g(\xi) = \int_{\Gamma_i}^g \xi ds.$$

We mention that  $Q_i^s, Q_i^f,$  and  $Q_i^g$  could be any conventional rules, such as Gaussian rules, Trapezoidal formula, and even Riemann sum rule. We fix  $Q_i^f$  (same as  $f_{\Omega_i}^m$ ) and  $Q_i^g$ , then correct  $Q_i^s$  to satisfy the condition (24). To this end, let  $Q_i^s$  be an  $p$ -point integration rule with the integration points and weights  $\{y_s, w_s\}_{s \in I_p}$ . We correct them as follows:

$$\begin{cases} y_{s,c} &= y_s \\ w_{s,c} &= w_s + w_s \sum_{r=1}^M \theta_r \tilde{p}_r(y_s) \cdot \nabla v_i(y_s) \end{cases}, \quad s \in I_p \quad (25)$$

such that the new integration rule with the points and weights  $\{y_{c,s}, w_{c,s}\}_{s \in I_p}$  satisfies the condition (24), where  $\theta_r, r \in I_M$  are determined by the condition (24). This rule is referred to as *k-corrected integration rule*, denoted by  $Q_i^{s,c}$ . To determine the parameters  $\theta_r$ , integrating (24) by  $Q_i^{s,c}$ ,  $Q_i^f$ , and  $Q_i^g$ , we get

$$Q_i^{s,c}(\tilde{p}_t \cdot \nabla v_i) = -Q_i^f(\nabla \cdot \tilde{p}_t v_i) + Q_i^g(\tilde{p}_t \cdot \tilde{n} v_i), \forall t \in I_M. \tag{26}$$

Replacing the integration points and weights (25) of  $Q_i^{s,c}$  in LHS in (26), we get a linear system with the unknowns  $\theta_r$  as follows:

$$\sum_{r \in I_M} Q_i^s((\tilde{p}_t \cdot \nabla v_i)(\tilde{p}_r \cdot \nabla v_i)) \theta_r = -Q_i^f(\nabla \cdot \tilde{p}_t v_i) + Q_i^g(\tilde{p}_t \cdot \tilde{n} v_i) - Q_i^s(\tilde{p}_t \cdot \nabla v_i), \forall t \in I_M. \tag{27}$$

We note that the coefficients matrix  $[Q_i^s((\tilde{p}_r \cdot \nabla v_i)(\tilde{p}_t \cdot \nabla v_i))]_{r,t \in I_M}$  can be expressed by

$$\left[ \sum_{s \in I_p} [\tilde{p}_r(y_s) \cdot \nabla v_i(y_s)] [\tilde{p}_t(y_s) \cdot \nabla v_i(y_s)] w_s \right]_{r,t \in I_M},$$

which is Gram matrix of the vectors in  $\mathbb{R}^p$

$$[\tilde{p}_r(y_s) \cdot \nabla v_i(y_s)]_{s \in I_p}, r \in I_M \tag{28}$$

under a weighted inner product

$$\langle m, n \rangle_w = \sum_{s \in I_p} m_s n_s w_s.$$

Therefore, the linear system (27) is solved uniquely as long as the vectors (28) is linearly independent. Solving the linear system (27), we get the parameters  $\theta_r$  and further get the *k-corrected integration rule*  $Q_i^{s,c}$ , which satisfy GIC (15).

**Remark 4.5** *In our computation, we find out that in almost all cases the linear independence of (28) is ensured by the condition  $p \geq M$ , namely, the number of the integration points is not less than the dimensionality of  $\tilde{\mathcal{P}}_{k-1}$ . Indeed, there are extremely exceptional cases where the vectors (28) may be linearly dependent for  $p \geq M$ . Fortunately, even in those cases, the linear system (27) is compatible and can be solved efficiently by the singular value decomposition method. We note that for each  $\Omega_i$ , we only correct  $Q_i^s$  one time to get  $Q_i^{s,c}$  by solving a linear system of degree  $M$  (27). Therefore, the correction algorithm above does not increase extra complexity in computation. In addition, the integration algorithms constructed above are problem-independent, namely, they do not depend on the coefficients  $A$  and  $c$  of the equation (1). ■*

### 4.3 A useful special situation of GIC

In fact, the constructive algorithm developed above is a quite general approach and can be applied to the various weight function  $v_i$ , the arbitrary shapes of  $\Omega_i$  and integration rules on  $\Omega_i$  (even Riemann sum rule). As we comment in Introduction, applying GIC in MLPG is more advantageous than in other MM (such as EFGM or RKPM) thanks to its flexibility in choosing the different test functions. Importance of GIC in MLPG consists in that it provides us with a direction to choose the test functions from viewpoint of reducing integration complexity.

Indeed, it is possible for us to propose a MLPG that both maintains the underlying approximation properties and possesses the less integration complexity. We have investigated a useful but not trivial situation of GIC. Using the conical weight function  $w^1$  and constructing  $v_i$  through the tensor product of  $w^1$  (5), we obtain the polynomial weight function  $v_i$  that has square support in 2D and cubic support in 3D. If  $\text{supp } v_i \subset \Omega$ , then  $\Omega_i = \text{supp } v_i$ , and such a  $v_i$  is called as interior weight function. For every interior weight function  $v_i$ , the conventional Gaussian rules make GIC (15) holds automatically because the integrands in (15) are all polynomials, which could be integrated exactly by Gaussian rules. This case also occurs to those  $v_i$ , which intersect the boundary  $\Gamma$  but have the regular integration domains  $\Omega_i$  such that Gaussian rules could integrate the integrands in GIC (15) exactly.

It is concluded from this discussion that the weight functions of the tensor product of the conical weight  $w^1$  are suggested to be used in MLPG from viewpoint of computation complexity. In this case, GIC (15) is satisfied automatically for the interior weight functions (the majority of the weight functions) by using Gaussian rules, while for few remaining  $v_i$ , the constructive algorithm in Subsection 4.2 is applied to construct the integration rules to satisfy GIC. As a consequence, the efficiency of numerical integration in MLPG is increased significantly.

### 4.4 Extension to elasticity problem

We extend GIC (15) to 2D elasticity problem, and the extensions to other elliptic PDE could be carried out in the same idea. Denote by  $\mathbf{u}$  and  $[\boldsymbol{\tau}]$  a vector-valued function  $[u_1(x), u_2(x)]^T$  and a matrix-valued function  $[\tau_{ij}]_{i,j=1,2}$ , respectively. For every  $\mathbf{u}$  and  $[\boldsymbol{\tau}]$ , we define

$$\text{grad } \mathbf{u} = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} \end{bmatrix} \text{ and } \text{div } [\boldsymbol{\tau}] = \begin{bmatrix} \frac{\partial \tau_{11}}{\partial x_1} + \frac{\partial \tau_{12}}{\partial x_2} \\ \frac{\partial \tau_{21}}{\partial x_1} + \frac{\partial \tau_{22}}{\partial x_2} \end{bmatrix}.$$

For a displacement function  $\mathbf{u}$ , we denote  $\sigma^u$  to be the stress tensor relative to  $\mathbf{u}$ . For any  $[\tau]$  and  $[\eta]$ , we define

$$[\tau] : [\eta] = \sum_{i=1}^2 \sum_{j=1}^2 \tau_{ij} \eta_{ij}.$$

Below, we will use the notations in Section 2, without repeating their definition again.

Consider the plane linear elasticity problem in  $\Omega$  with traction boundary condition:

$$\begin{aligned} -\operatorname{div} \sigma^u &= \mathbf{f}, & \text{in } \Omega \\ \sigma^u \vec{n} &= \mathbf{g}, & \text{on } \Gamma, \end{aligned} \tag{29}$$

where  $\mathbf{f}$  and  $\mathbf{g}$  are the body force and the boundary condition, respectively. Multiplying  $\mathbf{v}_i$  on both sides of (29) and integrating on  $\Omega_i$ , we have

$$\int_{\Omega_i} -\operatorname{div} \sigma^u \cdot \mathbf{v}_i dx = \int_{\Omega_i} \mathbf{f} \cdot \mathbf{v}_i dx.$$

Using divergence theorem and the traction boundary condition, we get

$$\begin{aligned} \int_{\Omega_i} \sigma^u : \operatorname{grad} \mathbf{v}_i dx &= \int_{\Omega_i} \mathbf{f} \cdot \mathbf{v}_i dx + \int_{\partial\Omega_i} \sigma^u \vec{n}_i \cdot \mathbf{v}_i ds \\ &= \int_{\Omega_i} \mathbf{f} \cdot \mathbf{v}_i dx + \int_{\Gamma_i} \mathbf{g} \cdot \mathbf{v}_i ds + \int_{L_i} \sigma^u \vec{n}_i \cdot \mathbf{v}_i ds. \end{aligned} \tag{30}$$

As specified above, we assume that the test function  $\mathbf{v}_i$  has compact support such that  $\mathbf{v}_i = 0$  on  $L_i$ , then (30) is reduced to

$$\int_{\Omega_i} \sigma^u : \operatorname{grad} \mathbf{v}_i dx = \int_{\Omega_i} \mathbf{f} \cdot \mathbf{v}_i dx + \int_{\Gamma_i} \mathbf{g} \cdot \mathbf{v}_i ds. \tag{31}$$

Motivated from (31), we propose GIC for the plane elasticity problem as follows:

**EIC.** For every  $i \in I_N$ , the integration formulae on  $\Omega_i$  and  $\Gamma_i$  are required to satisfy the condition:

$$\int_{\Omega_i}^s [\tau] : \operatorname{grad} \mathbf{v}_i dx + \int_{\Omega_i}^f \operatorname{div} [\tau] \cdot \mathbf{v}_i dx - \int_{\Gamma_i}^g [\tau] \vec{n} \cdot \mathbf{v}_i ds = 0, \forall [\tau] \in [\mathcal{P}_{k-1}], \tag{32}$$

where  $[\mathcal{P}_{k-1}]$  denotes the space of symmetric matrix-valued polynomials of degree  $k - 1$ :

$$[\mathcal{P}_{k-1}] := \left\{ \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} : p_1, p_2, p_3 \in \mathcal{P}_{k-1} \right\}.$$

We note that GIC (32) is problem-independent in a sense that the integration algorithms derived from (32) do not depend on the Lamé constants involved in the stress tensor  $\sigma^u$  of (29). The algorithms proposed in Subsections 4.2 and 4.3 can also be employed to construct the integration algorithms to satisfy GIC (32), and we will not specify it here.

### 5 Numerical experiments

We present 1D and 2D numerical results to demonstrate the advantages of the integration rules developed in Section 4.2 and 4.3. We consider  $\Omega = (0, 1)$  and  $\Omega = (0, 1) \times (0, 1)$  in 1D and 2D examples, respectively, for simplicity. However, we mention that the integration algorithms in Section 4.2 and 4.3 do not depend on shapes of the domain  $\Omega$ . In 1D, we assume in (1) the exact solution  $u(x) = e^{2x}$ ,  $A(x) = (1 + x^3)$ , and  $c(x) = 1 + \sin^2 x$ ; while in 2D, let  $u(x_1, x_2) = e^{2x_1 + x_2}$ ,  $A(x) = I$ ,  $c(x) = 1$ , where  $I$  is the identity matrix in  $\mathbb{R}^2$ . The body force  $f$  and the boundary condition  $g$  are derived in the initial equation (1) from  $u$ ,  $A$ , and  $c$ .

The uniformly distributed particles  $\{z_i, i \in I_N\}$  are employed to discretize the domain  $\Omega$ , where positive integer  $N$  represents number of the particles. The particle spacing is denoted by  $h$ , which equals to  $\frac{1}{N-1}$  in 1D and  $\frac{1}{\sqrt{N-1}}$  in 2D. The weight function  $v_i(x)$  is defined by  $v_i(x) = \omega(\frac{x-z_i}{h})$ . In 2D, we get  $v_i$  from the tensor product of the 1D weight function  $w^1$  (see (5)) so that the support of  $v_i$  is a square with side length  $2Rh$ . We produce the MLS trial functions  $\phi_i$  based on  $v_i$  by following the procedure in Subsection 3.1. In this setting, it is shown in [Babuska, Banerjee, and Osborn (2003)] that existence of the MLS shape functions  $\phi_i$  is under the condition  $R > \frac{k+1}{2}$  in both 1D and 2D. In these numerical experiments, we take  $R = 1.4$  for  $k = 1$  and  $1.8$  for  $k = 2$ , respectively. Recall that  $I'_N \subset I_N$  denotes the index set of the interior weight functions, namely,  $\Omega_i = \text{supp } v_i$  for  $i \in I'_N$ .

Below, once writing  $p$ - $Q$  integration rule (such as  $p$ -Gaussian rule) on  $\Omega_i$  in 1D case, we mean that we employ  $p$ -point  $Q$  integration rule on  $\Omega_i$  with its integration points and weights  $\{y_s, w_s\}_{s \in I_p}$ . If we write  $p$ - $Q$  integration rule in 2D case, we are using the  $p \times p$ -point  $Q$  integration rule on  $\Omega_i$ , where its integration points and weights are obtained from tensor product of  $p$ -point  $Q$  rule in 1D, namely, they are  $\{(y_s, y_t), w_s \times w_t\}_{s,t \in I_p}$ . In addition, in both 1D and 2D, if we write  $k$ -corrected  $p$ - $Q$  rule on  $\Omega_i$ , we integrate the entries  $f_i$  and  $g_i$  in the load vector and the entries  $m_{ij}$  in the mass matrix by  $p$ - $Q$  rule, and then we correct  $p$ - $Q$  rule on  $\Omega_i$  for the entries  $\gamma_{ij}$  in the stiffness matrix to satisfy GIC (15), by following the constructive algorithm in Subsection 4.2.

In 1D case, for every  $i \in I'_h$ , we employ the conventional Gaussian integration on  $\Omega_i := (a_i, b_i)$ , and because of symmetry of Gaussian rule and anti-symmetry of the

derivative of  $v_i$  on  $(a_i, b_i)$ , we get

$$\int_{a_i}^{b_i} v_i' dx = v_i(b_i) - v_i(a_i) = 0, \forall i \in I'_N. \quad (33)$$

It means that there is a coincidence for Gaussian rule in 1D such that GIC for  $k = 1$  is satisfied automatically for the interior particles, see (15) or (19). Therefore, it is not convenient for us to show the advantage of GIC in this case. To avoid this coincidence, we develop an non-symmetric Gaussian rule based on Gaussian rule. To this end, for every  $i \in I_N$ , we consider a mapping  $T_i : (a_i, b_i) \rightarrow (a_i, b_i)$ , defined by

$$z = T_i(y) = y + \frac{0.2}{b_i - a_i} \left[ \left( y - \frac{a_i + b_i}{2} \right)^2 - \left( \frac{b_i - a_i}{2} \right)^2 \right].$$

For a smooth function  $f$ , we have

$$\int_{a_i}^{b_i} f(z) dz = \int_{a_i}^{b_i} f(T_i(y)) T_i'(y) dy.$$

Using  $p$ -point Gaussian rule with the integration points and weights  $\{y_s, w_s\}_{s \in I_p}$  on RHS of the above equality, we get a new integration formula on  $(a_i, b_i)$  with the points and weights  $\{T_i(y_s), T_i'(y_s) w_s\}_{s \in I_p}$ , which is called as non-symmetric Gaussian rule on  $(a_i, b_i)$ . It is known that algebraic precision of  $p$ -point Gaussian rule is  $2p - 1$ . It can be shown that algebraic precision of such non-symmetric Gaussian rule, presented above, is  $p - 1$ . For the non-symmetric Gaussian rules, the integration equality (33) will not hold automatically.

The solution, computed from the system (13) with numerical integration, is denoted by  $u_N^*$ , and we will compare the relative energy error

$$E_N := \frac{\|u - u_N^*\|_{H^1(\Omega)}}{\|u\|_{H^1(\Omega)}}$$

for the various integration rules.

**The case  $k = 1$**

We employ the following integration rules and the weight functions

- Rule 1:  $p$ -point non-symmetric Gaussian rule and the cubic spline weight function;
- Rule 2: 1-corrected  $p$ -point non-symmetric Gaussian rule and the cubic spline weight function;

- Rule 3:  $p$ -point non-symmetric Gaussian rule and the conical weight function;

to compute the approximation solution  $u_N^*$  and draw the loglog plot of the energy errors  $E_N$  with respect to  $N$  in Fig. 1 (a) and (b) for 1D and 2D cases, respectively. We note that Rule 1 does not satisfy GIC; while GIC holds for Rule 2 and 3 because of the constructive algorithm in Subsection 4.2 and the polynomial property of the conical weight (see Subsection 4.3), respectively.

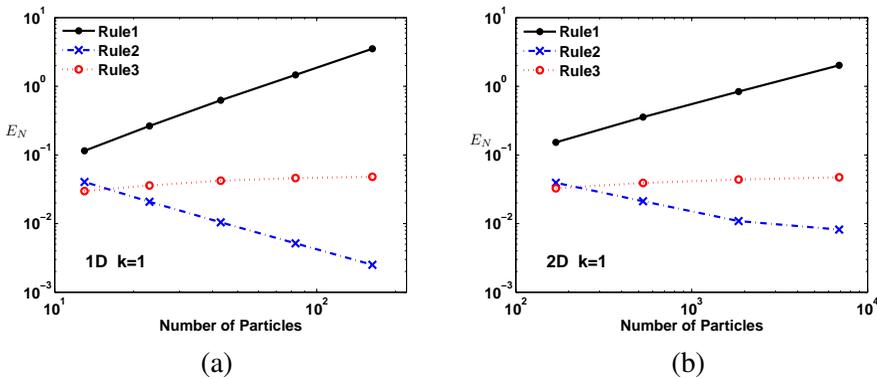


Figure 1: The loglog plot of  $E_N$  with respect to  $N$  for  $k = 1$  in 1D (a) and 2D (b).  $u_N^*$  is computed using 10-point non-symmetric Gaussian rule and the cubic weight; 1-corrected 10-point non-symmetric Gaussian rule and the cubic weight; and 10-point non-symmetric Gaussian rule and the conical weight. The later two satisfy GIC (15) for  $k = 1$ .

It is observed that without satisfaction of GIC, the energy errors  $E_N$  increase as  $N$  increases, on the contrary, the errors are reduced by Rule 2 and Rule 3 significantly due to GIC. At the highest particle density, the ratios of  $E_N$  of Rule 2 and 3 to Rule 1 are 0.071% and 1.4% respectively in 1D, and are 0.4% and 2.3% in 2D respectively; while at the lowest particle density, they are 35% and 26% in 1D respectively, and are 26% and 21% in 2D respectively. This implies that GIC is more advantageous at the higher particle density. We do not use symmetric Gaussian rule to demonstrate the advantage of GIC because in this case GIC will hold automatically for the interior weight functions due to symmetry of Gaussian rules and anti-symmetry of  $\nabla v_i$  on  $\Omega_i$ .

**The case  $k = 2$**

We compute the approximate solutions  $u_N^*$  using Rule 1, Rule 3 and

- Rule 4: 2-corrected  $p$ -point non-symmetric Gaussian rule and the cubic spline weight function

and present the loglog curves of the errors  $E_N$  with respect to  $N$  in Fig. 2 (a) for 1D and (b) for 2D cases, respectively. Similarly, when GIC does not hold, the errors grow up as  $N$  increases. After we correct Rule 1 or replace the cubic weight with the conical weight such that GIC for  $k = 2$  satisfies, Rule 4 and Rule 3 will reduce the errors greatly. We note that in the case  $k = 2$  improvement of the errors is more notable than in the case  $k = 1$ . At the highest particle density, the ratios of  $E_N$  of Rule 4 and 3 to Rule 1 are 0.075% and 0.011% respectively in 1D, and are 0.25% and 0.038% respectively in 2D; while at the lowest particle density, they both are 14% 1D, and are 12% and 9.4% respectively in 2D. Therefore, GIC behaves more notably at the higher particle density.

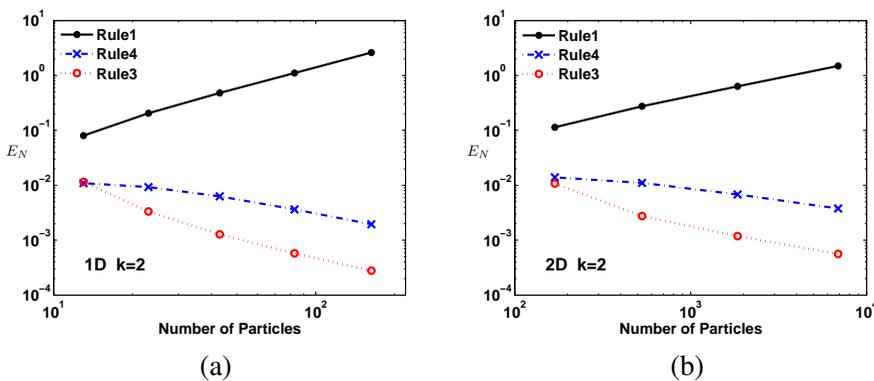


Figure 2: The loglog plot of  $E_N$  with respect to  $N$  for  $k = 2$  in 1D (a) and 2D (b).  $u_N^*$  is computed using 10-point non-symmetric Gaussian rule and the cubic weight; 2-corrected 10-point non-symmetric Gaussian rule and the cubic weight; and 10-point non-symmetric Gaussian rule and the conical weight. The later two satisfy GIC (15) for  $k = 2$ .

We also check the situations for symmetric Gaussian rules. We apply

- Rule 5:  $p$ -point symmetric Gaussian rule and the cubic spline weight function;
- Rule 6: 2-corrected  $p$ -point symmetric Gaussian rule and the cubic spline weight function;
- Rule 7:  $p$ -point symmetric Gaussian rule and the conical weight function;

to compute the approximate solutions  $u_N^*$  and present the loglog curves of the errors  $E_N$  with respect to  $N$  in Fig. 3 (a) for 1D and (b) for 2D cases, respectively. For Rule 5, the errors  $E_N$  are controlled (not growing up) as  $N$  increases because it satisfies GIC for  $k = 1$ . But, Rule 5 does not present an asymptotical convergence behavior as  $N$  increases because it does not satisfy GIC for  $k = 2$ . Again, after the correction scheme or replacement of the weight function, the associate Gaussian rules improve the errors markedly. At the highest particle density, the ratios of  $E_N$  of Rule 6 and 7 to Rule 5 are 0.14% and 0.074% respectively in 1D, and are 0.51% and 0.54% respectively in 2D; while at the lowest particle density, they are 32% and 50% respectively in 1D, and are 34% and 43% respectively in 2D. Again, we see the better feature of GIC at the higher particle density.

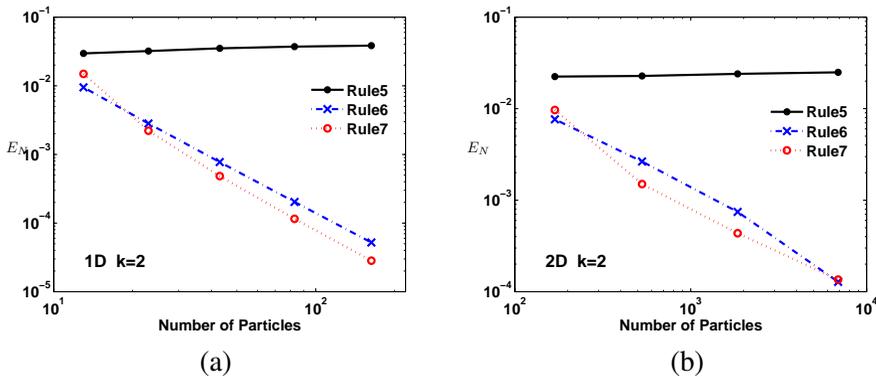


Figure 3: The loglog plot of  $E_N$  with respect to  $N$  for  $k = 2$  in 1D (a) and 2D (b).  $u_N^*$  is computed using 10-point symmetric Gaussian rule and the cubic weight; 2-corrected 10-point symmetric Gaussian rule and the cubic weight; and 10-point symmetric Gaussian rule and the conical weight. The first one satisfies GIC (15) only for  $k = 1$ ; while the later two satisfy the GIC (15) for  $k = 2$ .

We also repeated these numerical experiments by changing the support size  $R$  of the weight functions and the number  $p$  of the integration points, and the results are very similar to the situation above. We will not present them here.

## 6 Conclusion

An integration constraint condition, GIC, is proposed for MLPG applied to solve elliptic PDE with Neumann boundary conditions. Designing the integration rules with GIC in MLPG is more beneficial than in other Galerkin meshless methods,

such as EFGM or RKPM, thanks to the advantages of MLPG of the support integration and the flexibility in choosing the test functions. A constructive algorithm not increasing extra computation complexity is presented to correct the conventional integration rules, for instance, Gaussian rules, Trapezoidal rule, and even Riemann sum rule, to satisfy GIC. We also present a useful situation where GIC holds automatically for Gaussian rules. This nice but not trivial property only occurs in MLPG and never in EFGM and RKPM. From this reason, we suggest use of the conical weight in MLPG in viewpoint of reducing integration complexity. GIC in [Babuska, Banerjee, Osborn, and Zhang (2009); Zhang and Banerjee (2012)] only works for Poisson equations, and approach to extending it to more general elliptic PDE is addressed in this paper, by taking a plane elasticity problem as example. The numerical experiments demonstrate that without GIC the errors in the approximate solutions of MLPG may increase as the particles are denser; while the integration rules with GIC reduce the errors significantly. GIC behaves more notably for the higher exactness  $k$  and particle density.

**Acknowledgement:** This research was partially supported by the Natural Science Foundation of China under grant 11001282, Guangdong Provincial Natural Science Foundation of China under grant S2011040003030, and the Fundamental Research Funds for the Central Universities. Much of work reported here was completed when Q. Zhang was a postdoctoral fellow at the University of Hong Kong. Q. Zhang wish to express sincere appreciation to Professor K. Y. Sze for his insightful discussion and suggestion on this paper.

## References

- Atluri, S. N.** (2004): *The Meshless Method (MLPG) for Domain and BIE Discretizations*. Tech Science Press, Forsyth, GA, USA.
- Atluri, S. N.; Shen, S.** (2002): *The Meshless Local Petrov Galerkin Method*. Tech. Sci. Press, Duluth.
- Atluri, S. N.; Zhu, T.** (1998): A new meshless local petrov-galerkin (mlpg) approach in computational mechanics. *Computational Mechanics*, vol. 22, pp. 117–127.
- Atluri, S. N.; Zhu, T.** (2000): New concepts in meshless methods. *Int. J. Numer. Meth. Engng.*, vol. 47, pp. 537–556.
- Avila, R.; Han, Z.; Atluri, S. N.** (2011): A novel MLPG-finite-volume mixed method for analyzing Stokesian flows & study of a new vortex mixing flow. *CMES: Computer Modeling in Engineering & Sciences*, vol. 71, pp. 363–396.

**Babuska, I.; Banerjee, U.; Osborn, J. E.** (2003): Survey of meshless and generalized finite element methods: a unified approach. *Acta Numerica*, vol. 12, pp. 1–125.

**Babuska, I.; Banerjee, U.; Osborn, J. E.; Li, Q.** (2008): Quadrature for meshless methods. *Int. J. Numer. Meth. Engng.*, vol. 76, pp. 1434–1470.

**Babuska, I.; Banerjee, U.; Osborn, J. E.; Zhang, Q.** (2009): Effect of Numerical Integration on Meshless Methods. *Comput. Methods Appl. Mech. Engrg.*, vol. 198, pp. 2886–2897.

**Beissel, S.; Belytschko, T.** (1996): Nodal integration of the element-free Galerkin method. *Comput. Methods Appl. Mech. Engrg.*, vol. 139, pp. 49–74.

**Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P.** (1996): Meshless methods: An overview and recent developments. *Comp. Meth. Appl. Mech. Engrg.*, vol. 139, pp. 3–47.

**Belytschko, T.; Lu, Y.; Gu, T.** (1994): Element-free Galerkin methods. *Int. J. Numer. Meth. Engng.*, vol. 37, pp. 229–256.

**Carpinteri, A.; Ferro, G.; Ventura, G.** (2002): The partition of unity quadrature in meshless methods. *Int. J. Numer. Meth. Engng.*, vol. 54, pp. 987–1006.

**Chen, J.S.; Wu, C.T.; Yoon, S.; You, Y.** (2001): A stabilized conformal nodal integration for a Galerkin mesh-free method. *Int. J. Numer. Meth. Engng.*, vol. 50, pp. 435–466.

**De, S.; Bathe, K. J.** (2000): The method of finite spheres. *Computational Mechanics*, vol. 25, pp. 329–345.

**Dolbow, J.; Belytschko, T.** (1999): Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics*, vol. 23, pp. 219–230.

**Duan, Q.; Belytschko, T.** (2009): Gradient and dilatational stabilizations for stress-point integration in the element-free galerkin method. *Int. J. Numer. Meth. Engrg.*, vol. 77, pp. 776–798.

**Fernandez-Mendez, S.; Huerta, A.** (2004): Imposing essential boundary conditions in mesh-free methods. *Comput. Methods Appl. Mech. Engrg.*, vol. 193, pp. 1257–1275.

**Fries, T. P.; Belytschko, T.** (2008): Convergence and stabilization of stress-point integration in mesh-free and particle methods. *Int. J. Numer. Meth. Engrg.* vol. 74, pp. 1067–1087.

**Fries, T. P.; Matthies, H. G.** (2004): *Classification and overview of meshfree methods*. Technical report, Technical University Braunschweig, Brunswick, Germany.

- Griebel, M.; Schweitzer, M. A.** (2002): A particle-partition of unity method. ii. efficient cover construction and reliable integration. *SIAM J. Sci. Comput.*, vol. 23, pp 1655–1682.
- Li, S.; Liu, W. K.** (2002): Meshfree and particle methods and their application. *Applied Mechanics Review*, vol. 55, pp. 1–34.
- Liu, W. K.; Jun, S.; Zhang, Y. F.** (1995): Reproducing kernel particle methods. *Int. J. Numer. Meth. Fluids.*, vol. 20, pp. 1081–1106.
- Liu, Y.; Belytschko, T.** (2010): A New support integration scheme for the weak-form in meshfree methods. *Int. J. Numer. Meth. Engng.*, vol. 82, pp. 699–715.
- Mazzia, A.; Ferronato, M.; Pini, G.; Gambolati, A.** (2007): A comparison of numerical integration rules for the Meshless Local Petrov-Galerkin method. *Numerical Algorithms*, vol. 45, pp. 61–74.
- Mazzia, A.; Pini, G.** (2010): Product Gauss quadrature rules vs. cubature rules in the meshless local Petrov Galerkin method. *Journal of Complexity*, vol. 26, pp. 82–101.
- Nie, Y. F.; Atluri, S. N.; Zuo, C. W.** (2006): The optimal radius of the support of radial weights used in moving least squares approximation. *CMES: Computer Modeling in Engineering & Sciences*, vol. 12, pp. 137–147.
- Pecher, R.** (2006): Efficient cubature formulae for MLPG and related methods. *Int. J. Numer. Meth. Engng.*, vol. 65, pp. 566–593.
- Schaback, R.; Wendland, H.** (2006): Kernel techniques: from machine learning to meshless methods. *Acta Numer.*, vol. 15, pp. 543–639.
- Sellountos, E. J.; Sequeira, A.; Polyzos, D** (2010): Solving Elastic Problems with Local Boundary Integral Equations (LBIE) and Radial Basis Functions (RBF) Cells. *CMES: Computer Modeling in Engineering & Sciences*, vol. 57, pp. 109.
- Sladek, J.; Stanak, P.; Han, Z. D.; Sladek, V.; Atluri, S. N.** (2013): Applications of the MLPG Method in Engineering & Sciences: A Review. *CMES: Computer Modeling in Engineering & Sciences*, vol. 92, pp. 423–475.
- Sze, K. Y.; Chen, J. S.; Sheng, N.; Liu, X. H.** (2004): Stabilized conforming nodal integration: exactness and variational justification. *Finite Elements in Analysis and Design*, vol. 41, pp. 147–171.
- Zhang, Q.** (2011): Theoretical analysis of numerical integration in the Galerkin meshless methods. *BIT Numerical mathematics*, vol. 51, pp. 459–480.
- Zhang, Q.** (2014): Quadrature for meshless Nitsche’s method. *Numerical Methods for Partial Differential Equations*, vol. 30, pp. 265–288.

**Zhang, Q.; Banerjee, U.** (2012): Numerical integration in Galerkin meshless methods, applied to elliptic Neumann problem with non-constant coefficients. *Advances in Computational Mathematics*, vol. 37, pp. 453–492.

**Zhang, T.; Dong, L.; Alotaibi, A.; Atluri, S. N.** (2013): Application of the MLPG Mixed Collocation Method for Solving Inverse Problems of Linear Isotropic/Anisotropic Elasticity with Simply/Multiply-Connected Domains. *CMES: Computer Modeling in Engineering & Sciences*, vol. 94, pp. 1–28.