Response to Reviewer's Comments

Authors appreciate careful reading of the manuscript by the reviewers, and are thankful for helpful suggestions for its improvement. Authors have modified the manuscript substantially in light of the reviewer's comments. All the modifications and changes made are highlighted in the Marked Revised Manuscript. Issues and concerns raised by the reviewers are discussed as follows:

Reviewer #1

In this paper the authors float an interesting idea of combining FEM for displacement and RKPM for pressure unknowns to approximate solution to incompressible elasticity problems. They derive an inf-sup constant estimator and develop an optimal constraint ratio that tells them how many meshfree functions they need to introduce in the problem domain to achieve stability and optimal convergence.

Incompressibility is an old issue, with many approaches in FEM introduced over the years. This is evident from the references cited in the paper. As a result, any innovation in this area, while welcome, must reach quite a high bar to make it into a journal like CMAME. While the idea of using the flexibility of RKPM to enforce incompressibility is interesting, the paper seems to be vague on several important details:

Comment 1. *RKPM functions used are cubic with C^2-continuity, while the displacements are either linear or quadratic in the present paper. The expense associated with: a. Constructing RKPM functions on top of the FEM mesh; b. Integrating these functions; and c. The likely very high condition number of the resulting LHS matrices does not really justify having a methodology whose accuracy is still limited to that of the (low-order) displacement discretization.*

Response: Authors thank the reviewer for this insightful comment regarding the order of the meshfree shape functions. The accuracy of the proposed method is indeed governed by the order of consistency satisfied by the reproducing kernel approximation, as specified in Eq. (55). This order is directly determined by the polynomial basis functions used in the reproducing kernel approximation, which, in this manuscript, is consistently chosen to match the order of the displacement approximation.

The reviewer's mention of "cubic order" correctly refers to the order of the kernel function itself, specifically the cubic B-spline kernel function in Eq. (54). It is a standard practice for ensuring the continuity and smoothness of meshfree shape functions, and is commonly employed with both linear and quadratic meshfree approximations, as detailed in [1]. The selection of this kernel function does not inherently increase the computational cost associated with constructing or integrating the meshfree shape functions, nor does it unilaterally dictate the accuracy of proposed method. To enhance clarity on this crucial point, we have incorporated the following statements into the revised manuscript:

Page 14 (line 328-332):

> " The consistency condition ensures that the reproducing kernel shape functions are able to reproduce the polynomial space spanned by the basis function $\boldsymbol{p}$, which is a fundamental requirement for the accuracy of the Galerkin method. Herein, the order of the basis function $\boldsymbol{p}$ is chosen to be the same as the order of the displacement approximation. "

Page 15 (line 362-366):

> " For pressure discretization, linear meshfree approximation with a normalized support size of 1.5 is employed for Tri3, Quad4, Tet4 and Hex8. For Tri6 and Quad8, a quadratic meshfree approximation with a normalized support size of 2.5 is utilized. "

Comment 2. *Numerical quadrature is another issue. The support of RKPM functions does not conform to the elements of the FEM mesh. As a result, using Gaussian quadrature on the FEM mesh has accuracy implication for the integration of the pressure-pressure and velocity-pressure matrices. Because the RKPM functions are C^2-continuous and non-conforming, increasing the number of Gaussian quadrature points will not improve accuracy over a given threshold. These issues are not addressed in the paper. It's not clear what quadrature rule is used in the calculations.*

Response: Authors are grateful to the reviewer for raising this important point regarding numerical quadrature. The issue of numerical integration indeed presents a well–known challenge for meshfree methods. This is primarily due to the overlapping nature of basis function supports and the rational form of meshfree shape functions, which can lead to a violation of the so-called integration constraint or variational consistency [2, 5]. The authors and their collaborators have actively researched this challenge, leading to the development of several methods designed to overcome this issue [3, 4].

Variational consistency generally demands that meshfree shape functions and their derivatives cooperatively reproduce the solution spanned by the basis function $\boldsymbol{p}$ within the Galerkin weak form. However, in the context of the present work, only the meshfree shape functions themselves appear in the weak form of Eq. (14). This fortunate characteristic implies that the proposed method is not subject to the stringent requirement of integration constraint. Consequently, the integration of all linear and bilinear forms within Eq. (14) can be efficiently performed using standard lower-order Gaussian quadrature rules, commonly employed in conventional finite element methods, without compromising optimal accuracy. A detailed discussion on the numerical integration strategy employed in the proposed method has been added to the revised manuscript on Page 15 (lines 348-354):

> “ The numerical integration issue caused by the loss of variational consistency between meshfree shape functions and their derivatives [60] would not appear in the mixed formulation of Eq. (14), this is due to the fact that Eq. (14) solely depends on the meshfree shape functions themselves. Therefore, the proposed method employs standard lower-order Gaussian quadrature rules, as commonly used in traditional finite element methods, while still maintaining its accuracy. ”

Comment 3. *While showing suggested placement of RKPM nodes on structured meshes, the authors do not say at all how they do this for arbitrary unstructured meshes. This is likely done by trial an error and the results are likely heavily dependent on the distribution of the RKPM nodes. (In that sense, the ”constraint ratio” alone is not enough.)*

Response: Authors appreciate the reviewer’s comment regarding the placement of the pressure nodes, particularly on unstructured meshes. The proposed method does not impose a strict limitation on the distribution of pressure nodes. Instead, it primarily requires that the chosen pressure nodes are well–distributed to avoid the local locking. For simplicity and computational efficiency, we have chosen to co-locate the pressure nodes with the displacement nodes.

For unstructured meshes, the procedure for generating pressure and displacement nodes varies depending on the element order. For linear elements, such as Tri3, pressure nodes are initially generated using conventional meshing techniques, like Delaunay triangulation. Subsequently, displacement nodes are then generated through a standard mesh refinement process. For quadratic elements, the pressure nodes are specifically chosen as the vertices of the elements, effectively implementing an “every other node” strategy. This approach ensures that the resulting constraint ratio consistently falls within the optimal range. We have added a discussion of these nodal placement strategies to the revised manuscript on Page 16 (line 383–389):

> “ For practical implementations of linear cases, the pressure nodes are initially generated using traditional approaches, such as Delaunay triangulation. Subsequently, the displacement nodes are then obtained through a standard mesh refinement process to the pressure nodes. For quadratic approximations in Tri6 and Quad8 elements, the element vertices are chosen as pressure nodes after displacement element generation. Consequently, all constraint ratios evaluated using the discretizations in Figure (10) fall within the optimal range. ”

Comment 4. *Finally, Cook’s membrane and Compression Block problems are only presented “qualitatively”. The authors are citing several papers where these problems are solved and comparison of several quantities of interest is shown. For Cook’s membrane one typically looks at the corner node displacement and its convergence under mesh refinement. For Compression Block one typically applies several levels of pressure loading and monitors the resulting displacement, also showing convergence under mesh refinement. Such a quantitative assessment is necessary to even consider the paper for publication.*

Response: Authors appreciate the reviewer’s suggestion to provide quantitative assessment for problems of Cook’s membrane and compression block, the corresponding results and discussions have been added on Page 25 and (line 460-464, Figure 18 in the revised manuscript) and Page 29 (line 479-482, Figure 24 in the revised manuscript), respectively. The results also evidenced the effectiveness of the proposed optimal constraint ratio in eliminating volumetric locking and enhancing the performance of mixed finite element and meshfree formulations.

In conclusion, the paper cannot be accepted as it is now. The comments above need to be addressed, which may require a significant effort, before further consideration for publication.

Response: Authors appreciate the reviewer’s comprehensive comments and constructive suggestions, which have undoubtedly enhanced the quality of this manuscript.

Reviewer #2

Summary and Contributions

The manuscript introduces a theoretically grounded "optimal volumetric constraint ratio" for mixed formulations of incompressible elasticity, derived via the inf–sup (LBB) condition. By re-casting the inf–sup constant as the smallest nonzero eigenvalue of a dilatational stiffness operator (Section 3.1) and constructing an estimator that counts the active polynomial modes (Section 3.2–3.4), deriving an explicit bound. They then implement this ratio in a mixed finite–element–meshfree framework (Section 4), using standard FE interpolation for displacement and reproducing–kernel shape functions for pressure, allowing arbitrary pressure–node placement to satisfy the proposed bound. Inf–sup tests on 2D and 3D canonical domains confirm that inf–sup stability breaks down as $n_p > n_s$.

Comments

Comment 1. *The term* n_u *appears to be used with slightly different meanings. In Table 1,* n_u *is listed as "number of displacement nodes" for a given polynomial order n. For example, for 2D linear* ($n = 1$), $n_u = 3$. *This seems to refer to the number of nodes required to define a scalar polynomial of that order. However, in Eq. 38,* $n_u = \dim V_{n_u}$ *. Here,* V_{n_u} *is the "polynomial displacement space"* $P_{n_u}^{n_d}$ *(line 218). If* P_{n_u} *is the space of scalar polynomials of dimension (say)* N_{poly}, *then* $\dim V_{n_u} = n_d \times N_{poly}$.

Response: Thanks for the reviewer's careful observation. This is indeed a mistake in Eq. (38), the term n_u in Eq. (38) should be replaced by $n_d \times n_u$, which represents the total number of DOFs within the displacement space V_{n_u}. The authors have corrected this mistake in the revised manuscript on Page 10 (Eq. (38)):

$$n_d \times n_u = \dim V_{n_u}$$

This correction ensures that the meaning of n_u is consistent throughout the paper.

Comment 2. *The paper proposes selecting "every other displacement node as a pressure node" (line 461-462) to achieve the optimal constraint ratio. While this is a practical and simple strategy, its generality and optimality require further justification. In particular, for higher-order elements, "every other node" might still lead to a relatively dense pressure field. Moreover, it would be interesting to state precisely how the pressure nodes have been chosen in Figure 3-8 and Figure 11.*

Response: Authors appreciate the reviewer's valuable comment regarding the selection strategy for pressure nodes. For the implementation of linear elements, the pressure nodes are initially generated using traditional meshing methods. Subsequently, the displacement nodes are generated through a standard mesh refinement process. For quadratic elements, specifically Tri6 and Quad8, selecting the vertices of the elements as pressure nodes effectively implements the ”every other node” strategy. We have clarified this discussion in the revised manuscript on Page 16 (lines 383-389):

> “ For practical implementations of linear cases, the pressure nodes are initially generated using traditional approaches, such as Delaunay triangulation. Subsequently, the displacement nodes are then obtained through a standard mesh refinement process to the pressure nodes. For quadratic approximations in Tri6 and Quad8 elements, the element vertices are chosen as pressure nodes after displacement element generation. Consequently, all constraint ratios evaluated using the discretizations in Figure (10) fall within the optimal range. ”

Furthermore, it’s important to clarify the pressure node distribution in Figures 3-8, 11, and 24. In these figures, the pressure nodes are not selected directly from the displacement nodes. Instead, they are uniformly distributed across the entire domain using a Cartesian product. To enhance clarity, we have added a new Figure 3 in the revised manuscript to visually illustrate this pressure node distribution. Corresponding statements have also been added on Page 16(lines 366-368).

Comment 3. *Though the global smoothness of the reproducing kernel functions is highlighted, no discussion is given on the conditioning of the resulting linear systems or the cost of assembling the mixed system. Quantitative data (e.g., condition numbers, solve times) would strengthen the work.*

Response: Authors appreciate the reviewer’s valuable comment regarding the conditioning of the linear systems and the associated computational costs. These are indeed crucial aspects for every numerical method, particularly for meshfree approaches.

We have actually investigated the condition number and efficiency for the Cook’s membrane problem presented in Section 5.3 (Figures 18-21 in revised manuscript). The results, summarized in Table 1, indicate that the condition numbers and CPU-times for solving the discrete governing equations with the proposed methods are indeed larger than those of traditional finite element methods. This is primarily attributed to the larger bandwidth of the stiffness matrices generated by meshfree approximation. However, it’s important to note that a significant portion of the computational expense for the proposed methods is spent on calculating the meshfree shape functions.

The result about the lower efficiency of the reproducing kernel shape functions is consistent with the pioneering literatures [4], and moreover, the primary objective of using the proposed mixed FE-meshfree method is to demonstrate the effectiveness of the optimal constraint ratio and to establish a general framework for enhancing arbitrary displacement elements to circumvent volumetric locking. Consequently, to maintain the conciseness of the manuscript, we opted not to include condition number and efficiency comparisons for all numerical examples in Section 5.

Table 1: Condition number and efficiency comparison for Cook's membrane problem

Method	Condition number	CPU–time (s) for		
		Shape function	Assembly	Solving
MINI	1.11E06	0.025	0.327	0.022
Tri3–RK($r = n_d$)	1.89E10	1.73	4.160	0.108
Tri3–RK($r = r_{opt}$)	1.13E08	1.290	1.720	0.052
T6P3	1.62E05	0.004	0.380	0.021
Tri6–RK($r = n_d$)	2.48E16	1.62	1.67	0.294
Tri6–RK($r = r_{opt}$)	3.69E10	1.110	0.634	0.077
Q4P1	5.75E12	0.011	0.344	0.021
Quad4–RK($r = n_d$)	5.21E10	2.1	4.89	0.122
Quad4–RK($r = r_{opt}$)	1.97E08	1.5	2.14	0.057
Q8P3	2.69E07	0.005	0.373	0.015
Quad8–RK($r = n_d$)	2.75E15	1.17	1.18	0.184
Quad8–RK($r = r_{opt}$)	8.67E10	0.847	0.471	0.065

Comment 4. *It would be valuable to compare error convergence and locking behavior against established stabilized methods (e.g., B-bar, MINI–element, Taylor–Hood, etc.) to demonstrate not only stability but also accuracy and efficiency. Namely, meshfree methods may be computationally expensive or ill-conditioned due to integration over background cells of functions with large support domains.*

Response: Thanks for the reviewer's comment. In the revised manuscript, the Section 5.1 has been added the error convergence studies of 4–node quadrilateral displacement element with the piece–wise constant pressure element (Q4P1), and 8–node quadrilateral displacement element with piece–wise linear pressure element (Q8P3), and Section 5.2 has been added the results of the MINI element, 6–node triangular displacement element with 3–node triangular pressure element (T6P3), which can better demonstrate the effectiveness of the proposed methods. As discussed in Comment 4, the efficiency or condition number of the proposed methods is not the main focus of this manuscript, so the authors have not provided these studies.

Comment 5. *The author should comment on how they enforce bound-*

ary conditions. In particular, it would be interesting to know how boundary conditions on the pressure field are applied due to the meshfree approximation.

Response: Authors appreciate the reviewer's comment, it is right that the imposition of essential boundary conditions is a key issue for meshfree methods, since the loss of Kronecker delta property of meshfree shape functions leads to the diffculty in directly imposing essential boundary conditions. However, as shown in Eq. (14), the mixed formulation studied in this manuscript does not include any essential boundary conditions on the pressure field. Only the displacement field is required to satisfy the essential boundary conditions, which is imposed by the standard penalty method in this study.

Minor Comments

Comment 1. *In Eq. (4), please define precisely the meaning of $\boldsymbol{u}\nabla$. Moreover $\nabla \cdot \boldsymbol{u}$ is a scalar, it should be multiplied by the identity tensor.*

Response: Authors thanks for the reviewer's careful observation. A second-order identity tensor **1** has been added in Eq. (4) to clarify the meaning of $\nabla^d \boldsymbol{u}$. And the Eq. (4) has been modified as follows in the revised manuscript on Page 5:

$$\nabla^d \boldsymbol{u} = \frac{1}{2}(\boldsymbol{u}\nabla + \nabla \boldsymbol{u}) - (\frac{1}{3}\nabla \cdot \boldsymbol{u})\mathbf{1} \tag{1}$$

Comment 2. *A few minor typos e.g. "Mixd formulation" in Keywords;*

Response: Authors appreciate the reviewer's careful reading of the manuscript, and have corrected the typo in the Keywords. And the authors have also double-checked the entire manuscript for any other typos and have corrected them accordingly.

Comment 3. *At lines 338 and 342 you use the Kroneker symbol for the cartesian product of intervals.*

Response: Authors appreciate the reviewer for pointing out this mistake. The symbol ⊗ has been replaced by the Cartesian product symbol × in the revised manuscript on Page 15 (line 349 and 351).

Comment 4. *Some figures make the document slow to load. I recommend either reducing the number of components to render or raster them.*

Response: Authors appreciate the reviewer's suggestion, we have made the effort to accelerate the loading speed of the document, such as reducing the size of the pdf file.

Recommendation

Subject to satisfactory revision along these lines, I recommend acceptance with major revisions.

Response: Authors are thankful for the reviewer's positive comments and suggestions, which definitely help the authors to improve this manuscript.

References

[1] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 1994.

[2] Jiun Shyan Chen, Cheng Tang Wu, Sangpil Yoon, and Yang You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466, January 2001.

[3] Dongdong Wang and Junchao Wu. An efficient nesting sub-domain gradient smoothing integration algorithm with quadratic exactness for Galerkin meshfree methods. *Computer Methods in Applied Mechanics and Engineering*, 298:485–519, 2016.

[4] Dongdong Wang and Junchao Wu. An inherently consistent reproducing kernel gradient smoothing framework toward efficient Galerkin meshfree formulation with explicit quadrature. *Computer Methods in Applied Mechanics and Engineering*, 349:628–672, June 2019.

[5] Junchao Wu and Dongdong Wang. An accuracy analysis of Galerkin meshfree methods accounting for numerical integration. *Computer Methods in Applied Mechanics and Engineering*, 375:113631, March 2021.