Engineering with Computers (2024) 40:1027–1047
https://doi.org/10.1007/s00366-023-01843-6

ORIGINAL ARTICLE

Strong form mesh-free *hp*-adaptive solution of linear elasticity problem

Mitja Jančič[1,2] · **Gregor Kosec**[1]

Received: 25 October 2022 / Accepted: 4 May 2023 / Published online: 27 May 2023

Abstract

We present an algorithm for *hp*-adaptive collocation-based mesh-free numerical analysis of partial differential equations. Our solution procedure follows a well-established iterative solve–estimate–mark–refine paradigm. The solve phase relies on the Radial Basis Function-generated Finite Differences (RBF-FD) using point clouds generated by advancing front node positioning algorithm that supports variable node density. In the estimate phase, we introduce an Implicit-Explicit (IMEX) error indicator, which assumes that the error relates to the difference between the implicitly obtained solution (from the solve phase) and a local explicit re-evaluation of the PDE at hand using a higher order approximation. Based on the IMEX error indicator, the modified Texas Three Step marking strategy is used to mark the computational nodes for *h*-, *p*- or *hp*-(de-)refinement. Finally, in the refine phase, nodes are repositioned and the order of the method is locally redefined using the variable order of the augmenting monomials according to the instructions from the mark phase. The performance of the introduced *hp*-adaptive method is first investigated on a two-dimensional Peak problem and further applied to two- and three-dimensional contact problems. We show that the proposed IMEX error indicator adequately captures the global behaviour of the error in all cases considered and that the proposed *hp*-adaptive solution procedure significantly outperforms the non-adaptive approach. The proposed *hp*-adaptive method stands for another important step towards a fully autonomous numerical method capable of solving complex problems in realistic geometries without the need for user intervention.

Keywords RBF-FD · *hp*-adaptivity · Mesh-free · Linear elasticity · Error indicator

1 Introduction

Many natural and technological phenomena are modelled through Partial Differential Equations (PDEs), which can rarely be solved analytically—either because of geometric complexity or because of the complexity of the model at hand. Instead, realistic simulations are performed numerically. There are well-developed numerical methods that can be implemented in a more or less effective numerical solution procedure and executed on modern computers to perform virtual experiments or simulate the evolution of various natural or technological phenomena. Nonetheless, despite the immense computing power at our disposal, which allows us to solve ever more complex problems numerically, the development of efficient numerical approaches is still crucial. Relying solely on brute force computing often leads to unnecessarily long computations—not to mention wasted energy.

Most numerical solutions are obtained using mesh-based methods such as the Finite Volume Method (FVM), the Finite Difference Method (FDM), the Boundary Element Method (BEM) or the Finite Element Method (FEM). Modern numerical analysis is dominated by FEM [1] as it offers a mature and versatile solution approach that includes all types of adaptive solution procedures [2] and well understood error indicators [3]. Despite the widespread acceptance of FEM, the meshing of realistic 3D domains, a crucial part of FEM analysis where nodes are structured into polyhedrons covering the entire domain of interest, is still a problem that often requires user assistance or development of domain-specific algorithms [4].

✉ Mitja Jančič
mitja.jancic@ijs.si

Gregor Kosec
gregor.kosec@ijs.si

[1] Parallel and Distributed Systems Laboratory, Institute Jožef Stefan, Jamova Cesta 39, Ljubljana 1000, Slovenia

[2] Jožef Stefan International Postgraduate School, Jamova Cesta 39, Ljubljana 1000, Slovenia

In response to the tedious meshing of realistic 3D domains, required by FEM, and the geometric limitations of FDM and FVM, a new class of mesh-free methods [5] emerged in the 1970s. Mesh-free methods do not require a topological relationship between computational nodes and can therefore operate on scattered nodes, which greatly simplifies the discretisation of the domain [6], regardless of its dimensionality or shape [7, 8]. Just recently, they have also been promoted to Computer Aided Design (CAD) geometry aware numerical analysis [9]. Moreover, the formulation of mesh-free methods is extremely convenient for implementing h-refinement [10], considering different approximations of partial differential operators in terms of the shape and size of the *stencil* [11, 12] and the local approximation order [13]. However, they tend to be more computationally intensive as they require larger stencils for stable computations [13, 14] and have limited preprocessing capabilities [15]. This may make them less attractive from a computational point of view, but the ability to work with scattered nodes and easily control the approximation order makes them good candidates for many applications in science and industry [16, 17].

Adaptive solution procedures are essential in problems where the accuracy of the numerical solution varies spatially and are currently subject of intensive studies. Two conceptually different adaptive approaches have been proposed, namely p-adaptivity or h-, r-adaptivity. In p-adaptivity, the accuracy of the numerical solution is varied by changing the order of approximation, while in h- and r-adaptivity, the resolution of the spatial discretisation is adjusted for the same purpose. In the h-adaptive approach, nodes are added or removed from the domain as needed, while in the r-adaptive approach the total number of nodes remains constant – the nodes are only repositioned with respect to the desired accuracy. Ultimately, h- and p-adaptivities can be combined to form the so-called hp-adaptivity [18–20], where the accuracy of the solution is controlled with the order of the method and the resolution of the spatial discretisation.

Since the regions where higher accuracy is required are often not known *a priori*, and to eliminate the need for human intervention in the solution procedure, a measure of the quality of the numerical solution, commonly called a posterior error indicator, is a necessary additional step in an adaptive solution procedure [4]. The most famous error indicator, commonly referred to as the ZZ-type error indicator, was introduced in 1987 by Zienkiewicz and Zhu [21] in the context of FEM and it is still an active research topic [22]. The ZZ-type error indicator assumes that the error of the numerical solution is related to the difference between the numerical solution and a locally recovered solution. The ZZ-type error indicator has also been employed in the context of mesh-free solutions of elasticity problems using the mesh-free Finite Volume Method [23] in both weak and strong form using the Finite Point Method [24]. Furthermore, it also served as an inspiration in the context of Radial Basis Function-Generated Finite Difference (RBF-FD) solution to Laplace equation [25]. Moreover, a residual-based class of error indicators [26] has been demonstrated in the elasticity problems using a Discrete Least Squares mesh-free method [27]. Nevertheless, the most intuitive error indicators are based on the physical interpretation of the solution, usually evaluating the first derivative of the field under consideration [11] or calculating the variance of the field values within the support domain [10].

The advent of hp-adaptive numerical analysis began with FEM in the 1980s [28]. In hp-FEM, for example, one has the option of splitting an element into a set of smaller elements or increasing its approximation order. This decision is often considered to be the main difficulty in implementing the hp-adaptive solution procedure and was already studied by Babuška [28] in 1986. Since then, various decision-making strategies, commonly referred to as marking strategies, have been proposed [2, 29]. The early works use a simple Texas Three Step algorithm, originally proposed in the context of BEM [30], where the refinement is based on the maximum value of the error indicator. The first true hp-strategy was presented by Ainsworth [31] in 1997, since then many others have been proposed [2, 29]. In general, p- in FEM is more efficient when the solution is smooth. Based on this observation, most authors nowadays use the local Sobolev regularity estimate to choose between the h- and the p-refinement [32–34] for a given finite element. Moreover, in [35] local boundary values are solved, while the authors of [36, 37] use minimisation of the global interpolation error methods.

For mesh-free methods, h-adaptivity comes naturally with the ability to work with scattered nodes, and as such has been thoroughly studied in the context of several mesh-free methods [38–40]. Only recently, the popular Radial Basis Function-generated Finite Differences (RBF-FD) [41] have been used in the h-adaptive solution of elliptic problems [25, 42] and linear elasticity problems [10, 43]. Researchers have also reported the combination of h- and r-adaptivity, which form a so-called hr-adaptive solution procedure [44]. The p-adaptive method, on the other hand, is still quite unexplored in the mesh-free community. However, the authors of [45] approach the p-adaptive RBF-FD method in solving Poisson's equation with the idea of varying the order of the augmenting monomials to maintain the global order of convergence over the domain regardless of the potential variations in the spatial discretisation distance. It should also be noted that some authors reported p-adaptive methods by locally increasing the number of shape functions, changing the interpolation basis functions, or simply increasing the stencil size [46–48]. These approaches are all to some extent p-adaptive, but not in their true essence. The authors of [49] have introduced a p-refinement with spatially variable local approximation order and come closest to a true p-adaptive solution procedure on scattered nodes. However, this work lacks an automated marking and refinement strategy for the local approximation order,

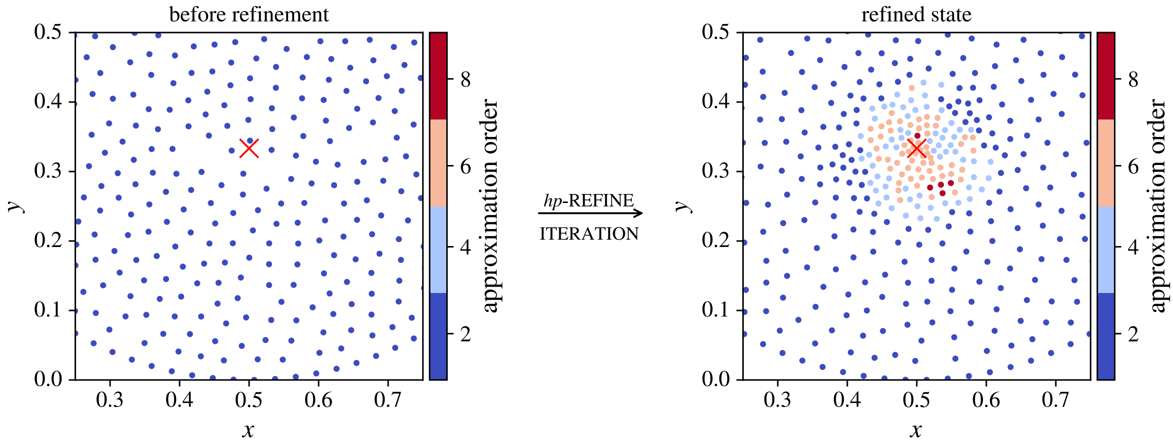


Fig. 1 A sketch of a single hp-refinement iteration for a two-dimensional problem. Note that the exponentially strong source (marked with red cross) is set at $\boldsymbol{p} = \left(\frac{1}{2}, \frac{1}{3}\right)$. The refined state has been obtained by employing h- and p-refinement strategies, thus the number of nodes and the local approximation orders in the neighbourhood of the strong source have been modified. Closed form solution has been used to indicate the error in the estimate module

e.g. based on an error indicator. The automated marking and refinement strategies were used with the weak form h-p adaptive clouds [50], where the authors use grid-like h-enrichment to improve the local field description.

In this paper, we present our attempt to implement the hp-adaptive strong form mesh-free solution procedure using the mesh-free RBF-FD approximation on scattered nodes. Our solution procedure follows a well-established paradigm based on an iterative loop. To estimate the accuracy of the numerical solution, we employ original IMEX error indicator. The marking strategy used in this work is based on the Texas Three Step algorithm [34], where the basic idea is to estimate the smoothness or analyticity of the numerical solution. Our refinement strategy is based on the recommendations of [10], where the authors were able to obtain satisfactory results using a purely h-adaptive solution procedure for elasticity problems. Although the chosen refinement and marking strategies are not optimal [36], the obtained results clearly outperform the non-adaptive approach.

2 *hp*-adaptive solution procedure

In the present work, we focus on the implementation of mesh-free hp-adaptive refinement, which combines the advantages of h- and p-refinement procedures. The proposed hp-adaptive solution procedure follows the well-established paradigm based on an iterative loop, where each iteration step consists of four modules:

1. **Solve** – A numerical solution $\widehat{u}$ is obtained.
2. **Estimate** – An error indication of the obtained numerical solution.
3. **Mark** – Marking of nodes for refinement/de-refinement.
4. **Refine** – Refinement/de-refinement of the spatial discretisation and local approximation order of the numerical method.

The workings of each module are further explained in the following subsections, while a full hp-adaptive solution procedure algorithm is given in Algorithm 1. For clarity, Fig. 1 also graphically sketches the ultimate goal of a single refinement iteration.

Algorithm 1 hp-adaptive solution procedure

Input: The problem, computational domain Ω, initial nodal density function $h:\ \Omega \to \mathbb{R}$, initial approximation order distribution $m:\ \Omega \to \mathbb{N}$, the maximal number of iterations $I_{\max}$ and adaptivity parameters $\alpha_{h,p}, \beta_{h,p}, \lambda_{h,p}, \vartheta_{h,p}$.
Output: The hp-refined numerical solution of the problem.

```
function ADAPTIVE_SOLVE(problem, $\Omega, h, m, I_{\max}, \alpha_{h,p}, \beta_{h,p}, \lambda_{h,p}, \vartheta_{h,p}$)
    for $i \leftarrow 0$ to $I_{\max}$ do
        $\Omega^\star \leftarrow$ DISCRETISE($\Omega, h$)    ▷ Discretises domain using nodal density function $h$.
        solution ← SOLVE(problem, $\Omega^\star, m$)    ▷ Obtains a numerical solution to the problem.
        indicator ← IMEX(problem, solution, $\Omega^\star, m$)    ▷ Error indicator computation.
        if STOPPING_CRITERIA then
            return solution
        end if
        $h, m \leftarrow$ ADAPT(indicator, $h, m, \Omega^\star, \alpha_{h,p}, \beta_{h,p}, \lambda_{h,p}, \vartheta_{h,p}$)    ▷ Refine the nodes and approximation orders.
    end for
    return solution
end function
```

2.1 The SOLVE module

First, a numerical solution $\widehat{u}$ to the governing problem must be obtained. In general, the numerical treatment of a system of PDEs is done in several steps. First, the domain is

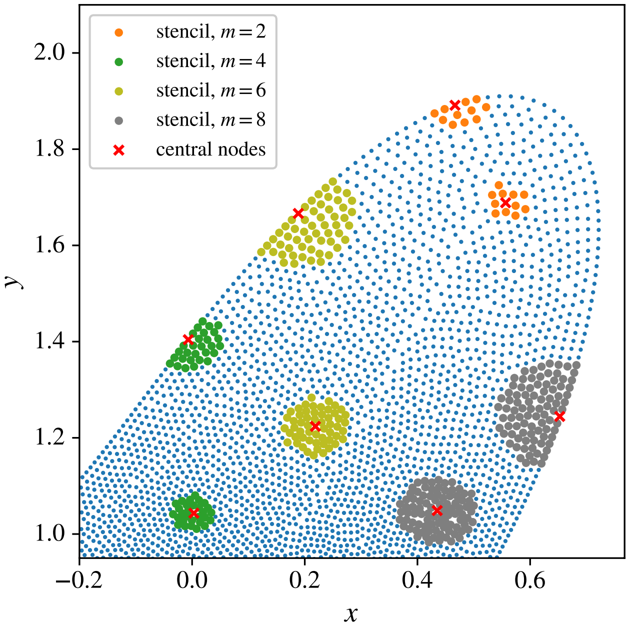


Fig. 2 An example of domain discretisation with scattered nodes and variable node density. Example stencils are also shown for different approximation orders m on the domain boundary and its interior

discretised by positioning the nodes, then the linear differential operators in each computational node are approximated, and finally the system of PDEs is discretised and assembled into a sparse linear system. To obtain a numerical solution $\hat{u}$, the sparse system is solved.

2.1.1 Domain discretisation

While traditional mesh-based methods discretise the domain by building a mesh, mesh-free methods simplify this step to the positioning of nodes, as no information about internodal connectivity is required. With the mathematical formulation of the mesh-free methods being dimension-independent, we accordingly choose a dimension-independent algorithm for node generation based on Poisson disc sampling [51]. Conveniently, the algorithm also supports spatially variable nodal densities required by the h-adaptive refinement methods. An example of a variable node density discretisation can be found in Fig. 2.

Interested readers are further referred to the original paper [51] for more details on the node generation algorithm, its stand-alone C++ implementation in the *Medusa library* [52], and follow-up research focusing on its parallel implementation [53] and parametric surface discretisations [54].

2.1.2 Approximation of linear differential operators

Having discretised the domain, we proceed to the approximation of linear differential operators. In this step, a linear differential operator $\mathcal{L}$ is approximated over a set of neighbouring nodes, commonly referred to as *stencil nodes*.

To derive the approximation, we assume a central point $\boldsymbol{x}_c \in \Omega$ and its stencil nodes $\{\boldsymbol{x}_i\}_{i=1}^n = \mathcal{N}$ for stencil size n. A linear differential operator in $\boldsymbol{x}_c$ is then approximated over its stencil with the following expression

$$(\mathcal{L}u)(\boldsymbol{x}_c) \approx \sum_{i=1}^{n} w_i u(\boldsymbol{x}_i), \tag{1}$$

for an arbitrary function u and yet to be determined weights $\boldsymbol{w}$ which are computed by enforcing the equality of approximation (1) for a chosen set of basis functions.

In this work, we use Radial Basis Functions (RBFs) augmented with monomials. To eliminate the dependency on a shape parameter, we choose Polyharmonic Splines (PHS) [14] defined as

$$f(r) = \begin{cases} r^k, & k \text{ odd} \\ r^k \log r, & k \text{ even} \end{cases}, \tag{2}$$

for Eucledian distance r. The chosen approximation basis effectively results in what is commonly called the RBF-FD approximation method [41].

Furthermore, it is necessary that the stencil nodes form a so-called polynomial unisolvent set [55]. In this work, we follow the recommendations of Bayona [14] and define the stencil size as twice the number of augmenting monomials, i.e.

$$n = 2\binom{m+d}{m} \tag{3}$$

for monomial order m and domain dimensionality d. This, in practice, results in large enough stencil sizes to satisfy the requirement, so that no special treatment was needed to assure unisolvency. While special stencil selection strategies showed promising results [11, 56], a common choice for selecting a set of stencil nodes $\mathcal{N}$ is to simply select the nearest n nodes. The latter approach was also used in this work. Figure 2 shows example stencils for different approximation orders m on domain boundary and its interior.

It is important to note that the augmenting monomials allow us to directly control the order of the local approximation method. The approximation order corresponds to the highest augmenting monomial order m in the approximation basis. However, the greater the approximation order the greater the computational complexity due to larger stencil sizes [13]. Nevertheless, the ability to control the local order of the approximation method sets the foundation for the p-adaptive refinement.

To conclude the solve module, the PDEs of the governing problem are discretised and assembled into a global sparse system. The solution of the assembled system stands for the numerical solution $\widehat{u}$.

2.2 The ESTIMATE module (Implicit-Explicit error indicator)

In the estimation step, critical areas with high error of the numerical solution are identified. Identifying such areas is not a trivial task. In rare cases where a closed form solution to the governing problem exists, we can directly determine the accuracy of the numerical solution. Therefore, other objective metrics, commonly referred to as *error indicators*, are needed to indicate areas with high error of the numerical solution.

2.2.1 IMplicit-EXplicit (IMEX) error indicator

In this work we will use an error indicator based on the implicit-explicit [57] evaluation of the considered field. IMEX makes use of the implicitly obtained numerical solution and explicit operators (approximated by a higher order basis) to reconstruct the right-hand side of the governing problem. To explain the basic idea of IMEX, let us define a PDE of type

$$\mathcal{L}u = f_{RHS}, \tag{4}$$

where $\mathcal{L}$ is a differential operator applied to the scalar field u and f_{RHS} is a scalar function. To obtain an error indicator field η, the problem (4) is first solved implicitly using a lower order approximation $\mathcal{L}^{im}$ of operators $\mathcal{L}$, obtaining the solution u^{im} in the process. The explicit high order operators $\mathcal{L}^{ex}$ are then used over the implicitly computed field u^{im} to reconstruct the right-hand side of the problem (4) obtaining f^{ex}_{RHS} in the process. The error indication is then calculated as $\eta = |f_{RHS} - f^{ex}_{RHS}|$. The calculation steps of the IMEX error indicator are also shown in Algorithm 2.

Algorithm 2 IMEX error indicator

```
Input: The problem, domain $\Omega$, differential operators $\mathcal{L}$, low-order approximation basis $\xi$, high order approximation basis $\zeta$.
Output: Error indicator field $\eta$.
function INDICATE_ERROR(problem, $\Omega$, $\mathcal{L}$, $\xi$, $\zeta$)
    $\mathcal{L}^{im} \leftarrow$ APPROXIMATE($\Omega, \xi$)    ▷ Obtain low-order approximation of differential operators $\mathcal{L}$.
    $u^{im}, f_{RHS} \leftarrow$ SOLVE(problem, $\Omega, \mathcal{L}^{im}$)    ▷ Obtain a numerical solution to the problem.
    $\mathcal{L}^{ex} \leftarrow$ APPROXIMATE($\Omega, \zeta$)    ▷ Obtain high order approximation of differential operators $\mathcal{L}$.
    $f^{ex}_{RHS} \leftarrow$ EVALUATE(problem, $\Omega, \mathcal{L}^{ex}, u^{im}$)    ▷ Explicit re-evaluation.
    $\eta \leftarrow$ COMPUTE($f_{RHS}, f^{ex}_{RHS}$)    ▷ Obtain error indicator field.
    return $\eta$
end function
```

The assumption that the deviation of the explicit high order evaluation $\mathcal{L}^{ex}u^{im}$ from the exact f_{RHS} corresponds to the error of the solution u^{im} is similar to the reasoning behind the ZZ-type indicators, where the deviation of the recovered high order solution from the computed solution characterises the error. As long as the error in u^{im} is high, the explicit re-evaluation will not correctly solve the Equation (4). However, as the error in u^{im} decreases, the difference between f_{RHS} and f^{ex}_{RHS} will also decrease, assuming that the error is dominated by the inaccuracy of u_{im} and not by the differential operator approximation.

It is worth noting that the definition of IMEX is general in the sense that computing the error indication η does not distinguish between the interior and boundary nodes. In the boundary nodes, the error indicator η is calculated in the same way as in the interior nodes. In the case of Dirichlet boundary conditions, the error indicator is trivial because the solution fields are exactly imposed, i.e. the error indicator results in $\eta = 0$. However, in case of boundary conditions involving the evaluation of derivatives (Robin and Neumann), $\eta \neq 0$.

2.3 The MARK module

After the error indicator η has been obtained for each computational point in domain Ω, a marking strategy is applied. The main goal of this module is to mark the nodes with too high or too low values of the error indicator to achieve a uniformly distributed accuracy of the numerical solution and to reduce the computational cost of the solution procedure – by avoiding fine local field descriptions and high order approximations where this is not required. Moreover, the marking strategy not only decides whether or not (de-)refinement should take place at a particular computational node, but also defines the type of refinement procedure if there are several to choose from. In this work, we use a modified Texas Three Step marking strategy [30, 58], originally restricted to refinement (no de-refinement) with the h- and p-refinement types. This chosen strategy was also considered in one of the recent papers by Eibner [34], who showed that, although extremely simple to understand and implement, it can provide results good enough to demonstrate the advantages of mesh-based hp-adaptive solution procedures.

In each iteration of the adaptive procedure, the marking strategy starts by checking the error indicator values η_i for all computational nodes in the domain. Unlike the originally proposed marking strategy [34] that used only refinement, we additionally introduce de-refinement. Therefore, if η_i is greater than $\alpha\eta_{max}$ for the maximum indicator value η_{max} and a free model parameter $\alpha \in (0, 1)$, the node is marked for refinement. If η_i is less than $\beta\eta_{max}$ for a free model parameter $\beta \in (0, 1) \wedge \beta \leq \alpha$, the node is marked for de-refinement.

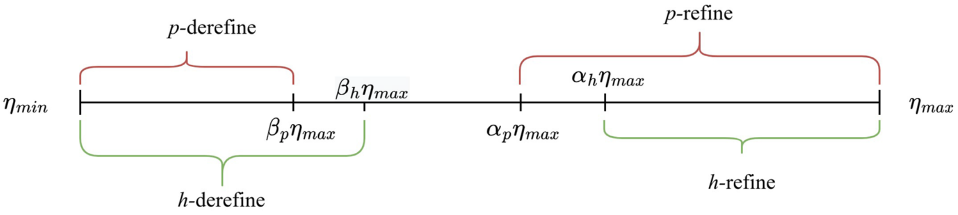


Fig. 3 A visual representation of h- and p-(de)refinement marking strategy

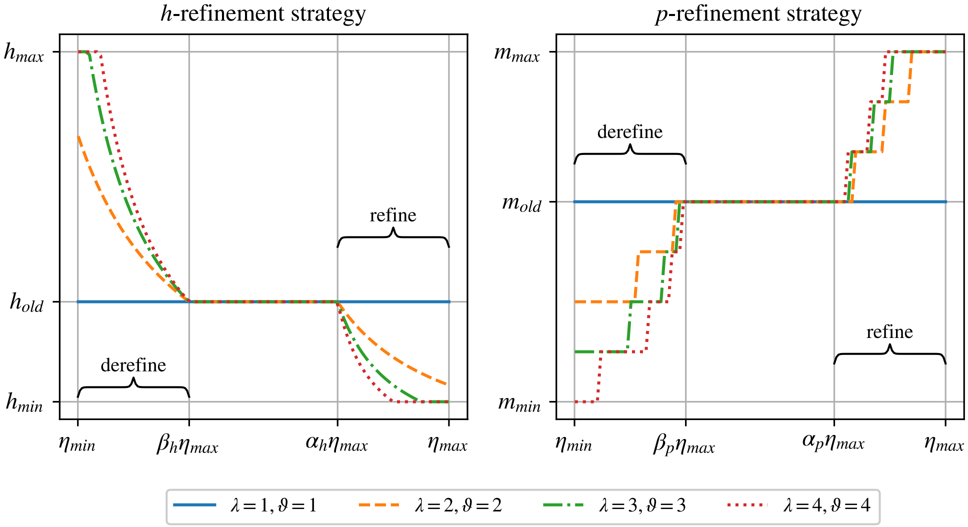


Fig. 4 A visual representation of the (de-)refinement strategies for different values of refinement aggressiveness λ and de-refinement aggressiveness ϑ. Notice that both refinement types also have lower and upper limits

Otherwise, the node remains unmarked, which means that no (de-)refinement should take place. The marking strategy can be summarised with a single equation

$$\begin{cases} \eta_i > \alpha\eta_{max}, & \text{refine} \\ \beta\eta_{max} \leq \eta_i \leq \alpha\eta_{max}, & \text{do nothing} \\ \eta_i < \beta\eta_{max}, & \text{de-refine} \end{cases}. \tag{5}$$

In the context of mesh-based methods, it has already been observed, that such marking strategy, although easy to implement, is far from optimal [2, 34]. Additionally, it has also been demonstrated that in case of smooth solutions p-refinement is preferred while h-refinement is preferred in volatile fields, e.g. in vicinity of a singularity in the solution [2, 36], which cannot be achieved with the chosen marking strategy. Additional discussion on this issue can be found in Sect. 4, where problems with singularity in the solution are discussed, and in Sect. 2.6.3 where we discuss some guidelines for possible work on improved marking strategies.

Since our work is focused on the implementation of hp-adaptive solution procedure rather than discussing the optimal marking strategy, we decided to secure full control over the marking strategy by treating h- and p-methods separately – but at the cost of higher number of free parameters. Therefore, the marking strategy is modified by introducing parameters $\left\{\alpha_h, \beta_h\right\}$ and $\left\{\alpha_p, \beta_p\right\}$ for separate treatment of h- and p-refinements, respectively (see Fig. 3 for clarification). Note that the proposed modified marking strategy can mark a particular node for h-, p- or hp-(de-)refinement if required, otherwise the computational node is left unchanged.

2.4 The REFINE module

After obtaining the list of nodes marked for modification, the refinement module is initialised. In this module, the local field description and local approximation order are left unchanged for the unmarked nodes, while the remaining nodes are further processed to determine other refinement-type-specific details – such as the amount of the (de-)refinement. Our h-refinement strategy is inspired by the recent h-adaptive mesh-free solution of elasticity problem [10], where the following h-refinement rule was introduced

$$h_i^{new}(\boldsymbol{p}) = \frac{h_i^{old}}{\frac{\eta_i - \alpha\eta_{max}}{\eta_{max} - \alpha\eta_{max}}\Big(\lambda - 1\Big) + 1} \tag{6}$$

for the dimensionless parameter $\lambda \in [1, \infty)$ allowing us to control the aggressiveness of the refinement – the larger the value, the greater the change in nodal density, as shown in Fig. 4 on the left. This refinement rule also conveniently refines the areas with higher error indicator values more than

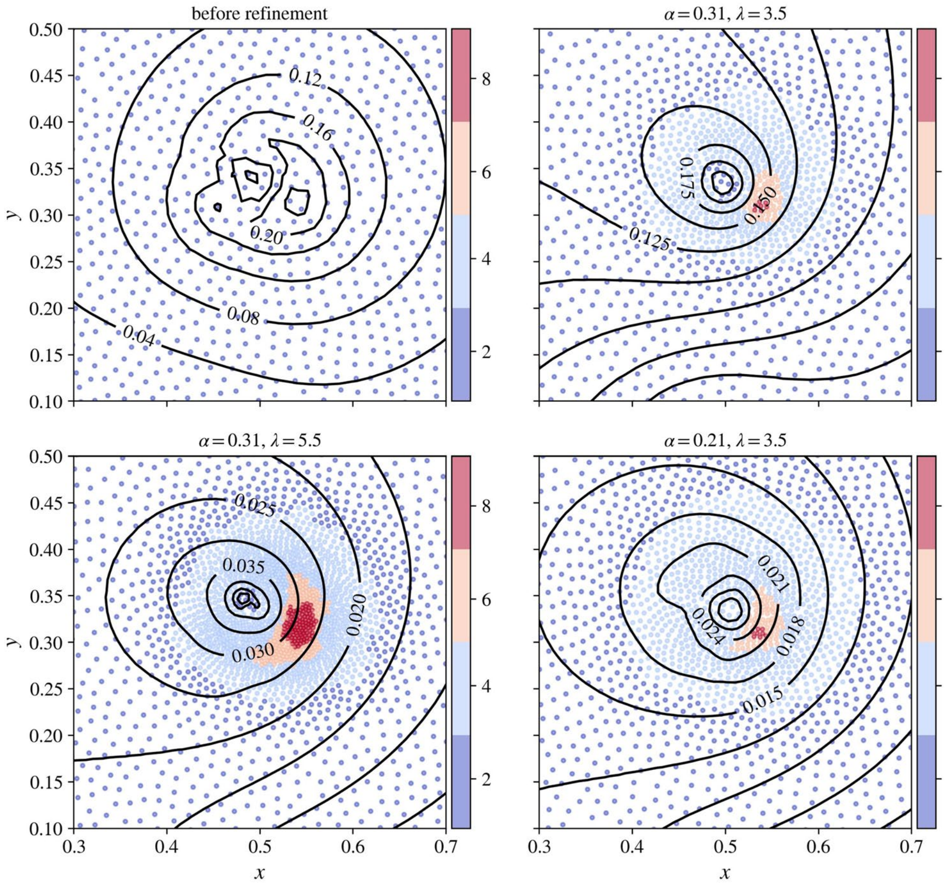


Fig. 5 Demonstration of *hp*-refinement for selected values of refinement parameters. The top left figure shows the numerical solution before its refinement, while the rest show its refined state for different values of refinement parameters. Contour lines are used to show the absolute error of the numerical solution. To denote the *p*-refinement, the nodes are coloured according to the local approximation order. For clarity, all figures are zoomed to show only the neighbourhood of an exponentially strong source $e^{-a\|\boldsymbol{x}-\boldsymbol{x}_s\|^2}$ positioned at $\boldsymbol{x}_s = \left(\frac{1}{2}, \frac{1}{3}\right)$

those closer to the upper refinement threshold $\alpha_h \eta_{max}$. Similarly, a de-refinement rule is proposed

$$h_i^{new}(\boldsymbol{p}) = \frac{h_i^{old}}{\frac{\beta\eta_{max}-\eta_i}{\beta\eta_{max}-\eta_{min}}\left(\frac{1}{\vartheta} - 1\right) + 1}, \tag{7}$$

where parameter $\vartheta \in [1, \infty)$ allows us to control the aggressiveness of de-refinement.

The same refinement (6) and de-refinement (7) rules are applied to control the order of local approximation (*p*-refinement), except that this time the value is rounded to the nearest integer, as shown in Fig. 4 on the right. Similarly, and for the same reasons as for the marking strategy (see Sect. 2.3), we consider a separate treatment of *h*- and *p*-adaptive procedures by introducing (de-)refinement aggressiveness parameters $\{\lambda_h, \vartheta_h\}$ and $\{\lambda_p, \vartheta_p\}$ for *h*- and *p*-refinement types respectively.

2.5 Finalization step

Before the 4 modules can be iteratively repeated, the domain is re-discretised taking into account the newly computed local internodal distances $h_i^{new}(\boldsymbol{p})$ and the local approximation orders $m_i^{new}(\boldsymbol{p})$. However, both are only known in the computational nodes, while global functions $\widehat{h}^{new}(\boldsymbol{p})$ and $\widehat{m}^{new}(\boldsymbol{p})$ over our entire domain space Ω are required.

We use Sheppard's inverse distance weighting interpolation using the closest n_s^h neighbours to construct $\widehat{h}^{new}(\boldsymbol{p})$ and the closest n_s^m neighbours to construct $\widehat{m}^{new}(\boldsymbol{p})$. In general, the proposed refinement strategy can introduce aggressive and undesirable local jumps in node density, which ultimately leads to a potential violation of the quasi-uniform internodal spacing requirement within the stencil. To mitigate this effect, we use relatively large $n_s^h = 30$ to smoothen such potential local jumps. The $\widehat{m}^{new}(\boldsymbol{p})$ is much less sensitive in this respect and therefore a minimum $n_s^m = 3$ is used.

Figure 5 schematically demonstrates 3 examples of *hp*-refinements. For demonstration purposes, the refinement parameters for *h*- and *p*-adaptivity are the same, i.e. $\{\alpha, \beta, \lambda, \vartheta\} = \{\alpha_h, \beta_h, \lambda_h, \vartheta_h\} = \{\alpha_p, \beta_p, \lambda_p, \vartheta_p\}$. Additionally, the de-refinement aggressiveness ϑ and the lower threshold β are kept constant, so that effectively only the upper limit of refinement α and the refinement aggressiveness λ are altered. We observe that the effect of the refinement parameters is somewhat intuitive. The greater the aggressiveness λ, the better the local field description and the greater the number of nodes with high approximation order. A similar effect is observed when manipulating the upper refinement threshold α, except that the effect comes at a smoother manner. Note also that all refined states were able to increase the accuracy of the numerical solution from the initial state.

2.6 Note on marking and refinement strategies

With the chosen marking and refinement strategies, a separate treatment of h- and p-refinement types turned out to be a necessary complication for a better overall performance of the solution procedure. Nevertheless, we have tried to simplify the solution procedure as much as possible. In the process, important observations have been made – some of which we believe should be highlighted. This section therefore opens a discussion on important remarks related to the proposed marking and refinement modules.

2.6.1 The error indicators

Since the h- and p-refinements are conceptually different, our first attempt was to employ two different error indicators – one for each type of refinement. We employed the previously proposed variance of field values [10] for marking the h-refinement and the approximation order based IMEX for the p-refinement. Unfortunately, no notable advantages of such solution procedure has been observed and was therefore discarded due to the increased implementation complexity. However, other combinations that might show more promising results should be considered in future work.

2.6.2 Free parameters

In the proposed solution procedure, each adaptivity type comes with 4 free parameters that need to be defined, i.e. $\left\{\alpha_{h,p}, \beta_{h,p}, \lambda_{h,p}, \vartheta_{h,p}\right\}$. This gives a total of 8 free parameters that can be fine-tuned to a particular problem. While we have tried to avoid any kind of fine-tuning, we have nevertheless observed that these parameters can have a crucial impact on the overall performance of the hp-adaptive solution procedure in terms of (i) the achieved accuracy of the numerical solution, (ii) the spatial variability of the error of the numerical solution, (iii) the computational complexity, and (iv) the stability of the solution procedure.

We observed that if the refinement aggressiveness λ_h is too high, the number of nodes can either diverge into unreasonably large domain discretisations or ultimately violate the quasi-uniform internodal spacing requirement, making the solution procedure unstable. Note that here we refer to the stability of the solution of the discretised PDEs, which ultimately governs the stability of the whole solution procedure. Furthermore, a large number of nodes combined with high approximation orders can lead to unreasonably high computational complexity in a matter of few iterations. However, when refinement aggressiveness λ_h and λ_p is set too low, the number of required iterations can increase to such an extent that the entire solution procedure becomes inefficient. On top of that, the lower and upper threshold multipliers α and β also play a crucial role. If α is too low, almost the entire domain is refined. Moreover, if α is too large, almost no refinement takes place and if it does, it is extremely local, which again has no beneficial consequences as it often leads to a violation of the quasi-uniform nodal distribution requirement.

In our tests, based on extensive experimental parameter testing, we have selected a reasonable combination of all 8 parameters that lead to a stable solution procedure while demonstrating the advantages of the proposed hp-adaptive approach. A thorough analysis of these parameters and their correlation would most likely lead to better results, as there is no guarantee that the selected parameters are optimal. However, such an analysis is beyond the scope of this paper, whose aim is to present an hp-adaptive solution procedure in the context of mesh-free methods and not to discuss the optimal marking and refinement strategies. Nevertheless, we have tried to reduce the number of free parameters using the same values for h- and p-adaptivity (see Fig. 5). While this approach also yielded satisfactory results that outperformed the numerical solutions obtained with uniform nodal and approximation order distributions in terms of accuracy, the full 8-parameter formulation easily yielded significantly better results.

2.6.3 A step beyond the artificial refinement strategies

As discussed in Sect. 2.3 and later in Sect. 4, the Texas Three Step based marking strategy cannot assure the optimal balance of h- and p-refinements due to missing local data regularity estimation [2]. In FEM, local Sobolev regularity estimate is commonly used to choose between the h- and the p-refinement [32–34]. Using an estimate for upper error bound [59, 60] one could generalise this approach to meshless methods, essentially upgrading the strategy with an information on the minimal internodal spacing required for local approximation of the partial differential operator of a certain order.

The refinement strategy could also be based on a specific knowledge about convergence rates and computational complexity in terms of internodal distance $h(\boldsymbol{p})$ and local approximation orders $m(\boldsymbol{p})$.

It has already been shown by Bayona [61] that the approximation error of mesh-free interpolant F is bounded by

$$\|F(\boldsymbol{p}) - u(\boldsymbol{p})\|_\infty \leq Ch^{m+1} \max_{\boldsymbol{p}\in\Omega} |\mathcal{L}^{(m+1)}(u(\boldsymbol{p}))|. \tag{8}$$

Note that the constant C present in Equation (8) depends on the stencil and on the approximation order, both of which are modified by the *hp*-adaptive solution procedure. Nevertheless, for the purpose of illustrating how a better marking strategy could be constructed, we decide to simplify the Equation (8) to saying that the error e is proportional to $h(\boldsymbol{p})^{m(\boldsymbol{p})}$. Knowing the target error e_t, we write the ratio of e_t/e_0 as

$$\frac{e_t}{e_0} \propto \frac{h^{m_t}}{h^{m_0}} = h^{m_t - m_0}, \tag{9}$$

where m_t is used to denote the target approximation order and m_0 is the current order of the approximation used to compute current error e_0.

From Equation (9) a smarter guess for target local approximation order can be obtained

$$m_t = m_0 + \ln \frac{e_t}{e_0}. \tag{10}$$

Such strategy would conveniently leave the approximation order unchanged when $e_t = e_0$, increase it when $e_t < e_0$ and decrease it when $e_t > e_0$.

A step even further could be to additionally consider the change in computational complexity, similar to what the authors of [35] and [45] have already shown. Therefore, we believe that future work should consider the minimum local computational complexity criteria. A rough computational complexity can be obtained with the help of

$$\chi \propto \frac{\binom{m_t + d}{d}^3 \left(\frac{1}{h_t}\right)^d}{\binom{m_0 + d}{d}^3 \left(\frac{1}{h_0}\right)^d}, \tag{11}$$

for domain dimensionality d and target and current internodal distances h_t and h_0 respectively.

2.7 Implementation note

The entire *hp*-adaptive solution procedure from Algorithm 1 is implemented in C++. All meshless methods and approaches used in this work are included in our in-house developed *Medusa library* [52]. The code[1] was compiled using g++ (GCC) 9.3.0 for Linux with -03 -DNDEBUG -fopenmp flags. Post-processing was done using Python 3.10 and Jupyter notebooks, also available in the provided git repository.

3 Demonstration on exponential peak problem

The proposed *hp*-adaptive solution procedure is first demonstrated on a synthetic example. We chose a 2-dimensional Poisson problem with exponentially strong source positioned at $\boldsymbol{x}_s = \left(\frac{1}{2}, \frac{1}{3}\right)$. This example is categorized as a difficult problem and is commonly used to test the performance of adaptive solution procedures [2, 29, 42, 62]. The problem has a tractable solution $u(\boldsymbol{x}) = e^{-a\|\boldsymbol{x}-\boldsymbol{x}_s\|^2}$, which allows us to evaluate the precision of the numerical solution $\widehat{u}$, e.g. in terms of the infinity norm

$$e_\infty = \frac{\|\widehat{u} - u\|_\infty}{\|u\|_\infty}, \quad \|u\|_\infty = \max_{i=1,\ldots,N} |u_i|. \tag{12}$$

Governing equations are

$$\nabla^2 u(\boldsymbol{x}) = 2ae^{-a\|\boldsymbol{x}-\boldsymbol{x}_s\|^2}(2a\|\boldsymbol{x} - \boldsymbol{x}_s\| - d) \quad \text{in } \Omega, \tag{13}$$

$$u(\boldsymbol{x}) = e^{-a\|\boldsymbol{x}-\boldsymbol{x}_s\|^2} \quad \text{on } \Gamma_d, \tag{14}$$

$$\nabla u(\boldsymbol{x}) = -2a(\boldsymbol{x} - \boldsymbol{x}_s)e^{-a\|\boldsymbol{x}-\boldsymbol{x}_s\|^2} \quad \text{on } \Gamma_n, \tag{15}$$

for a d-dimensional domain Ω and strength $a = 10^3$ of the exponential source. The domain boundary is split into two sets: Neumann $\Gamma_n = \left\{\boldsymbol{x}, x \leq \frac{1}{2}\right\}$ and Dirichlet $\Gamma_d = \left\{\boldsymbol{x}, x > \frac{1}{2}\right\}$ boundaries. An example *hp*-refined numerical solution is shown in Fig. 6.

In the continuation of this paper, the numerical solution of the final linear system is obtained by employing BiCGSTAB solver with a ILUT preconditioner from the Eigen C++ library [63]. Global tolerance was set to 10^{-15} with a maximum number of 800 iterations and drop-tolerance and fill-factor set to 10^{-5} and 50 respectively. While the initial adaptivity solution was obtained without the guess, all other iterations used the previous numerical solution $\widehat{u_{i-1}}$ as the guess for new numerical solution $\widehat{u}_i$, effectively reducing the number of iterations required by the BiCGSTAB solver.

3.1 Convergence analysis of unrefined solution

The problem is first solved on a two-dimensional unit disc without employing any refinement procedures, i.e. with uniform nodal and approximation order distributions. The shapes approximating the linear differential operators are

[1] The source code is available at: https://gitlab.com/e62Lab/public/2022_p_hp-adaptivity under tag *v1.2*.

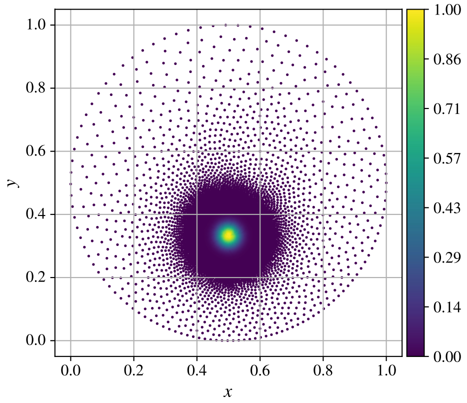


Fig. 6 Example *hp*-refined solution to exponential peak problem

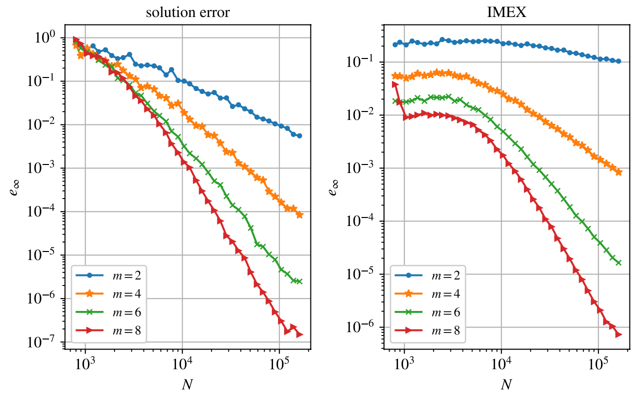


Fig. 7 Convergence of unrefined numerical solution (left) and IMEX error indicator (right). Figure only shows a median value after 50 runs with slightly different domain discretisations. Note that, the approximation order m in the right figure denotes the approximation order used to obtain the numerical solution, while the explicit operators employed by the IMEX error indicator are approximated with orders $m + 2$

computed using the RBF-FD with PHS order $k = 3$ and monomial augmentation $m \in \{2, 4, 6, 8\}$.

Figure 7 shows the results. Each plotted point is an average obtained after 50 consecutive runs with slightly different domain discretisations (a random seed for generating expansion candidates was changed, see [51] for more details). In this way, we can not only study the convergence behaviour, but also evaluate how prone the numerical method is to non-optimal domain discretisations. The convergence of the numerical solution for selected monomial augmentations is shown on the left. We observe that due to the strong source, the convergence rates no longer follow the theoretical prediction of being proportional to h^m. Instead, the convergence rates for a small number of computational nodes ($N \lessapprox 2000$) are significantly lower than that obtained for larger domain discretisations ($N \gtrapprox 3000$) for all approximation orders $m > 2$. Furthermore, the accuracy gain using higher order approximations with small domain discretisations is practically negligible. However, when the local field description is sufficient, both the numerical solution and the IMEX error indicator (Fig. 7 on the right) give reliable results. While we could have forced at least one node in the neighbourhood of the source, we do not use any special techniques in this work. Instead, further research is simply limited to sufficiently large domains so that this observation does not represent an issue.

Moreover, the behaviour of the IMEX error indicator is studied on the right side of Fig. 7. Here, the approximation order m means that the implicit numerical solution u^{im} was obtained with approximation order m, while the explicit operators $\mathcal{L}^{ex}$ from IMEX were approximated using monomials up to and including order $m + 2$. The observations show that the maximum value of the error indicator also converges with the number of computational nodes. Moreover, we can also observe the aforementioned change in the convergence rate of the numerical solution, since the maximum value of the error indicator for domain sizes $N \lessapprox 3000$ is approximately constant.

3.2 Analysis of *hp*-refined solution

The same problem is now solved by employing the *hp*-adaptivity. Free parameters are adjusted to each refinement type, as can be seen in Table 1. Adaptivity iteration loop is stopped after a maximum of N_{iter} iterations. For practical use, other stopping criteria could also be used, e.g. based on the maximum error indicator reduction

$$\frac{\eta^j_{max}}{\eta^0_{max}} \le \gamma, \qquad (16)$$

for the iteration index j. The shapes are computed with RBF-FD using the PHS with order $k = 3$ and local monomial augmentation restricted to choose between approximation orders $m \in \{2, 4, 6, 8\}$. Note that the IMEX error indicator increases the local approximation order by 2, effectively using monomial orders $m_{IMEX} \in \{4, 6, 8, 10\}$. Furthermore, to avoid unreasonably large number of computational nodes, the maximum number of allowed nodes N_{max} is defined. Once this number is reached, further h-refinement is prevented and

Table 1 Adaptivity parameters used to obtain solution to the peak problem

β_h	α_h	λ_h	ϑ_h	β_p	α_p	λ_p	ϑ_p	h_{max}	N_{max}	N_{iter}
0.175	0.225	2.625	1.01	10^{-4}	0.05	5	1.258	0.1	$2.5 \cdot 10^5$	70

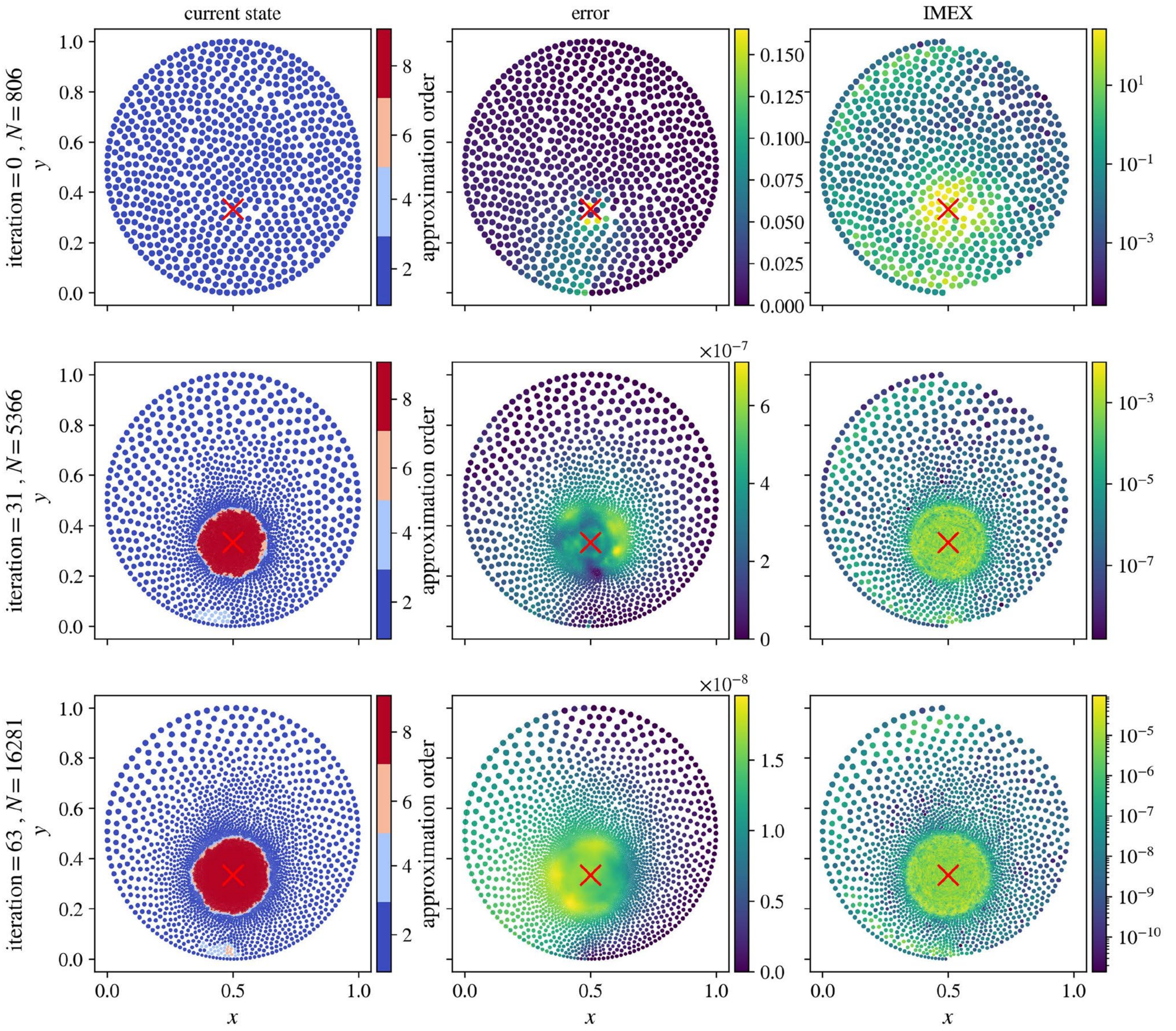


Fig. 8 Refinement demonstration. Initial iteration (top row), intermediate iteration (middle row) and best-performing iteration (bottom row) accompanied with solution error (middle column) and IMEX error indicator values (right column). The IMEX values for Dirichlet boundary nodes are not shown. A red cross is used to mark the location of the strong peak

only de-refinement is allowed, while the p-adaptive method retains its full functionality. To avoid insufficient local field description, the local nodal density is limited by an upper bound, i.e. $h(\boldsymbol{p}) \leq h_{max}$. The order of the PHS is left constant.

3.2.1 A brief analysis of IMEX error indicator

Figure 8 shows example indicator fields for the initial iteration, the intermediate iteration, and the iteration that achieved the best numerical solution accuracy – hereafter also referred to as the best-performing iteration or simply the best iteration. The third column shows the IMEX error indicator. We can see that the IMEX has successfully located the position of the strong source at $\boldsymbol{x}_s = \left(\frac{1}{2}, \frac{1}{3}\right)$ as the highest indicator values are seen in its vicinity. Furthermore, the second column shows that both the accuracy of the numerical solution and the uniformity of the error distribution were significantly improved by the hp-adaptive solution procedure, further proving that IMEX can be successfully used as a reliable error indicator.

The behaviour of IMEX over 70 adaptivity iterations is also studied in Fig. 9. We are pleased to find that the convergence limit of the indicator around iteration $N_{iter} = 60$ agrees well with the convergence limit of the numerical

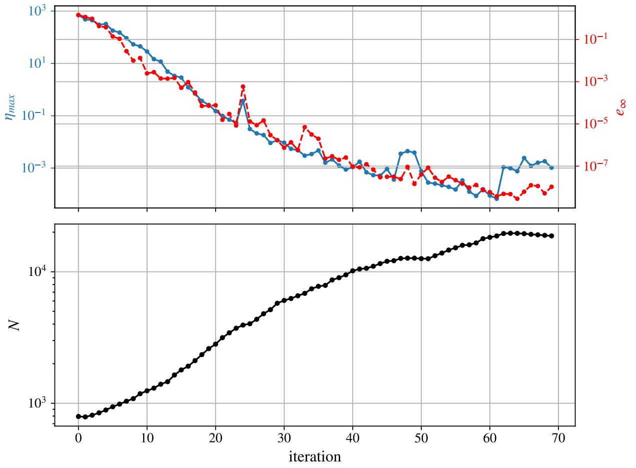


Fig. 9 In the top row convergence of IMEX error indicator (blue) and convergence of numerical solution (red) within 70 iterations is shown, while the total number of computational nodes is shown below

solution. This observation also makes the IMEX error indicator suitable for stopping criteria. Note that, in the process, the maximum error of the numerical solution has been reduced by about 9 orders of magnitude, while the maximum value of the error indicator has been reduced by about 7 orders of magnitude. In addition, Fig. 9 also shows the number of computational nodes with respect to the adaptivity iterations.

3.2.2 Approximation order distribution

The iterative adaptive procedure starts by obtaining the numerical solution of the unrefined problem setup. In this step, the approximation with the lowest approximation order, i.e. $m = 2$, is assigned to all computational nodes. Later, the approximation orders are changed according to the marking and refinement strategies. Figure 8 shows the approximation order distributions for 3 selected adaptivity iterations. We can observe that the highest approximation orders are all near the exponentially strong source. Moreover, due to h-adaptivity, the node density in the neighbourhood of the strong source is also significantly increased, i.e. $h_{max}/h_{min} \approx 52$ in the best-performing iteration.

After applying the p-refinement strategy in the refinement step, the approximation order in two neighbouring nodes may differ by more than one. While numerical experiments with FEM have shown that heterogeneity of polynomial order in FEM leads to undesired oscillations of the approximated solution [64], no similar behaviour was observed in our analyses with our setup using mesh-free methods. Thus, in contrast to p-FEM, where additional smoothing of the approximation order takes place within the refinement module, we have completely avoided such manipulations and allow the approximation order in two neighbouring nodes to differ by more than one.

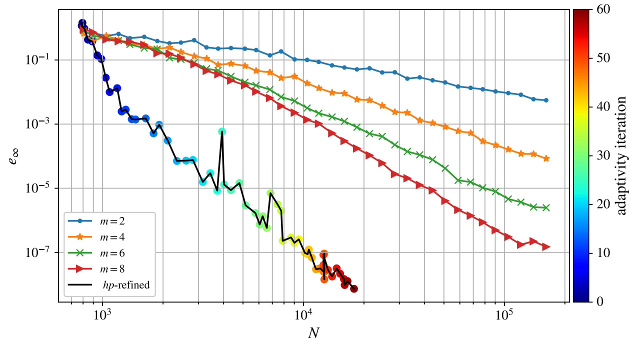


Fig. 10 Convergence of the hp-refined solution compared to the convergence of the unrefined solutions

3.2.3 Convergence rates of *hp*-adaptive solution procedure

Finally, the convergence behaviour of the proposed hp-adaptive solution procedure is studied. In addition to the convergence of a single hp-adaptive run, Fig. 10 shows the convergences obtained without the use of refinement procedures, i.e. solutions obtained with uniform internodal spacing and approximation orders over the entire domain. The figure clearly shows that a hp-adaptive solution procedure was able to significantly improve the numerical solution in terms of accuracy and computational points required.

As previously discussed by Eibner [34] and Demkowicz [36], we believe that a more complex marking and refinement strategies would further improve the convergence behaviour, but already the proposed hp-adaptive solution procedure significantly outperforms the unrefined solutions. Specifically, the refined solution is almost 4 orders of magnitude more accurate than the unrefined solution (for the highest approximation order $m = 8$ used) at about 10^4 computational nodes.

4 Application to linear elasticity problems

In this section we address two problems from linear elasticity that are conceptually different from the exponential peak problem discussed in Sect. 3. While the solution of exponential peak problem is infinitely smooth, these two problems both have a singularity in the solution.

In areas of smooth solution, the hp-strategy should favour p-refinement (assuming that the local discretization is sufficient, as briefly discussed in Sect. 3.1), while near the singularity, h-refinement should be preferred [2, 36]. However, the Texas Three Step based marking strategy used in this paper cannot trivially achieve this, since the strategy has no knowledge of the smoothness of the solution field. In addition, the strategy also cannot perform pure h- or pure p-refinement [34] (see Fig. 3), which would be ideal in the limiting situations. Instead, the strategy used enforces an increase in the approximation order by its design – even if the solution is not smooth and even if low-regularity data is being used to construct the approximation. Nevertheless, in our experiments we observed an increase of the approximation order near the singularity only in the first few iterations, while the following iterations were focused on improving the local field description with h-refinement. This observation is also in agreement with reports from the literature [2, 34], where authors justify the use of the Texas Three Step marking strategy also for problems with singularity in the solution.

4.1 Fretting fatigue contact

The application of the proposed hp-adaptive solution procedure is further expanded to study a linear elasticity problem. Specifically, we obtain a hp-refined solution to fretting fatigue contact problem [65] for which no closed form solution is known.

The problem dynamics is governed by the Cauchy-Navier equations

$$(\lambda + \mu)\nabla(\nabla \cdot \boldsymbol{u}) + \mu\nabla^2\boldsymbol{u} = \boldsymbol{f} \tag{17}$$

with unknown displacement vector $\boldsymbol{u}$, external body force $\boldsymbol{f}$ and Lamé parameters μ and λ. The domain of interest is a thin rectangle of width W, length L and thickness D. Axial traction σ_{ax} is applied to the right side of the rectangle, while a compression force is applied to the centre of the rectangle to simulate contact. The contact is simulated by a compressing force F generated by two oscillating cylindrical pads of radius R, causing a tangential force Q. The tractions introduced by the two pads are predicted using an extension of Hertzian contact theory, which splits the contact area into the stick and slip zones depending on the friction coefficient μ and the combined elasticity modulus $E^{*-1} = \left(\frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}\right)$, where E_i and ν_i are the Young's modulus and the Poisson's ratios of the sample and the pad, respectively. The problem is shown schematically in Fig. 11 together with the boundary conditions.

Theoretical predictions from [10] are used to obtain the contact half-width

$$a = 2\sqrt{\frac{FR}{t\pi E^*}}, \tag{18}$$

with normal traction

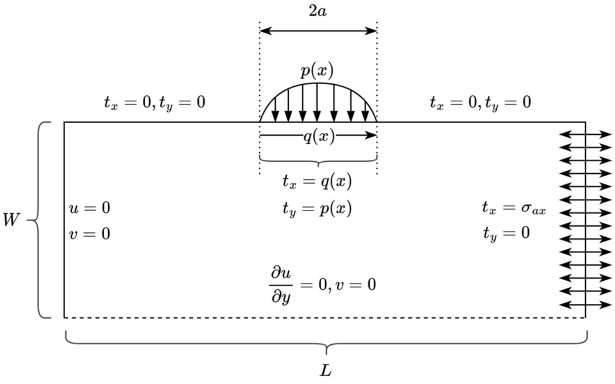


Fig. 11 Fretting fatigue contact problem scheme and boundary conditions

$$p(x) = \begin{cases} p_0\sqrt{1 - \frac{x^2}{a^2}}, & |x| < a \\ 0, & \text{else} \end{cases}, \qquad p_0 = \sqrt{\frac{FE^*}{t\pi R}}, \tag{19}$$

and tangential traction

$$q(x) = \begin{cases} -\mu p_0\left(\sqrt{1 - \frac{x^2}{a^2}} - \frac{c}{a}\sqrt{1 - \frac{(x-e)^2}{c^2}}\right), & |x - e| < c \\ -\mu p_0\sqrt{1 - \frac{x^2}{a^2}}, & c \leq |x - e| \text{ and } |x| \leq a \\ 0, & \text{else} \end{cases} \tag{20}$$

for $c = a\sqrt{1 - \frac{Q}{\mu f}}$ defined as the half-width of the slip zone and $e = \operatorname{sgn}(Q)\frac{a\sigma_{ax}}{4\mu p_0}$ is the eccentricity due to axial loading. Note that the inequalities $Q \leq \mu F$ and $\sigma_{ax} \leq 4\left(1 - \sqrt{1 - \frac{Q}{\mu F}}\right)$ must hold for these expressions to be valid.

Plane strain approximation is used to reduce the problem from three to two dimensions and symmetry along the horizontal axis is used to further halve the problem size. Finally, $\Omega = [-L/2, L/2] \times [-W/2, 0]$ is taken as the domain.

We take $E_1 = E_2 = 72.1\,\text{GPa}$, $\nu_1 = \nu_2 = 0.33$, $L = 40\,\text{mm}$, $W = 10\,\text{mm}$, $t = 4\,\text{mm}$, $F = 543\,\text{N}$, $Q = 155\,\text{N}$, $\sigma_{ax} = 100\,\text{MPa}$, $R = 10\,\text{mm}$ and $\mu = 0.3$ for the model parameters. With this setup, the half-contact width a is equal to $0.2067\,\text{mm}$, which is about 200 times smaller than the domain width W. For stability reasons, the 4 corner nodes were removed after the domain was discretised.

The linear differential operators are approximated with RBF-FD using the PHS with order $k = 3$ and local monomial augmentation limited to choose between approximation orders $m \in \{2, 4, 6, 8\}$. The PHS order was left constant during the adaptive refinement. The hp-refinement parameters used to obtain the numerical solution are given in Table 2.

Table 2 Adaptivity parameters used to obtain solution to fretting fatigue contact problem

β_h	α_h	λ_h	ϑ_h	β_p	α_p	λ_p	ϑ_p	h_{max}	N_{max}	N_{iter}
$5 \cdot 10^{-5}$	10^{-4}	5	1.05	10^{-3}	0.1	4	1.05	$2.5 \cdot 10^{-4}$	$5 \cdot 10^{5}$	19

Figure 12 shows an example of a *hp*-refined solution to fretting fatigue problem in the last adaptivity iteration with $N = 46\,626$ computational nodes. We see that the solution procedure has successfully located the two critical points, i.e. the fixed upper left corner with a stress singularity and the area in the middle of the upper edge where contact is simulated. Note that the highest stress values (about 2 times higher) were calculated in the singularity in the upper left corner, but these nodes are not shown as our focus is shifted towards the area under the contact.

4.1.1 Surface traction under the contact

For a detailed analysis, we consider the surface traction σ_{xx}, as it is often used to determine the location of crack initiation. The surface traction is shown in Fig. 13 for 6 selected adaptivity iterations. The mesh-free nodes are coloured according to the local approximation order enforced by the *hp*-adaptive solution procedure. The message of this figure is twofold. First, it is clear that the proposed IMEX error indicator can be successfully used in linear elasticity problems, and second, we find that the *hp*-adaptive solution procedure has successfully approximated the surface traction near the contact. In doing so, the local field description under the contact has been significantly improved and the local approximation orders have taken a non-trivial distribution.

The surface traction in Fig. 13 is additionally accompanied with the FEM results on a much denser mesh with more than 100,000 DOFs obtained with the commercial solver Abaqus® [65]. To calculate the absolute difference between the two methods, the mesh-free solution was interpolated to Abaqus's computational points using Sheppard's inverse distance weighting interpolation with 2 nearest neighbours. We see that the absolute difference under the contact decreases with the number of adaptivity iterations and eventually settles at approximately 2 % of the maximum difference from the initial iteration. As expected, the highest absolute difference is at the edges of the contact, i.e. around $x = a$ and $x = -a$, while the difference is even smaller in the rest of the contact area. The absolute difference between the two methods is further studied in Fig. 14, where the mean of $|\sigma_{xx}^{\mathrm{FEM}} - \sigma_{xx}^{\text{mesh-free}}|$ under the contact area, i.e. $-a \le x \le a$, is shown. We observe that the mesh-free *hp*-refined solution converges towards the reference FEM solution with respect to the adaptivity iterations. Moreover, Fig. 14 also shows the number of computational nodes with respect to the adaptivity iteration.

4.2 The three-dimensional Boussinesq's problem

As a final benchmark problem we solve the three-dimensional Boussinesq's problem, where a concentrated normal traction acts on an isotropic half-space [66].

The problem has a closed form solution given in cylindrical coordinates r, θ and z as

$$\begin{aligned}
u_r &= \frac{Pr}{4\pi\mu}\left(\frac{z}{R^3} - \frac{1-2\nu}{R(z+R)}\right), \qquad u_\theta = 0,\\
u_z &= \frac{P}{4\pi\mu}\left(\frac{2(1-\nu)}{R} + \frac{z^2}{R^3}\right),\\
\sigma_{rr} &= \frac{P}{2\pi}\left(\frac{1-2\nu}{R(z+R)} - \frac{3r^2z}{R^5}\right),\\
\sigma_{\theta\theta} &= \frac{P(1-2\nu)}{2\pi}\left(\frac{z}{R^3} - \frac{1}{R(z+R)}\right),\\
\sigma_{zz} &= -\frac{3Pz^3}{2\pi R^5}, \qquad \sigma_{rz} = -\frac{3Prz^2}{2\pi R^5},\\
\sigma_{r\theta} &= 0, \qquad \sigma_{\theta z} = 0,
\end{aligned} \tag{21}$$

where P is the magnitude of the concentrated force, ν is the Poisson's ratio, μ is the Lamé parameter and R is the Eucledian distance to the origin. The solution has a singularity at the origin, which makes the problem ideal for treatment with adaptive procedures. Furthermore, the closed form solution also allows us to evaluate the accuracy of the numerical solution.

In our setup, we consider only a small part of the problem, i.e. $\varepsilon = 0.1$ away from the singularity, as schematically shown in Fig. 15. From a numerical point of view, we solve the Navier–Cauchy Equation (17) with Dirichlet boundary conditions described in (21), where the domain Ω is defined as a box, i.e. $\Omega = [-1, -\varepsilon] \times [-1, -\varepsilon] \times [-1, -\varepsilon]$.

Although the closed form solution is given in cylindrical coordinate systems, the problem is implemented using cartesian coordinates. We employ the proposed mesh-free *hp*-adaptive solution procedure where the shapes are computed with RBF-FD using the PHS with order $k = 3$ and monomial augmentation restricted to choose between approximation orders $m \in \{2, 4, 6, 8\}$. Other *hp*-refinement related parameters are given in Table 3. For the physical parameters of the problem, the values $P = -1$, $E = 1$ and $\nu = 0.33$ were assumed.

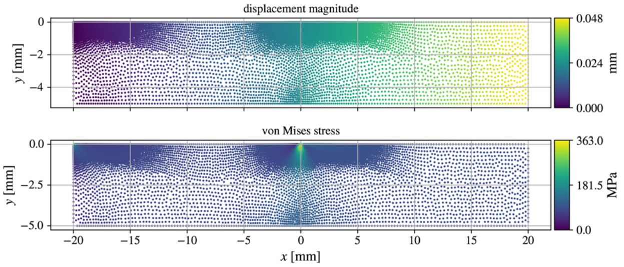
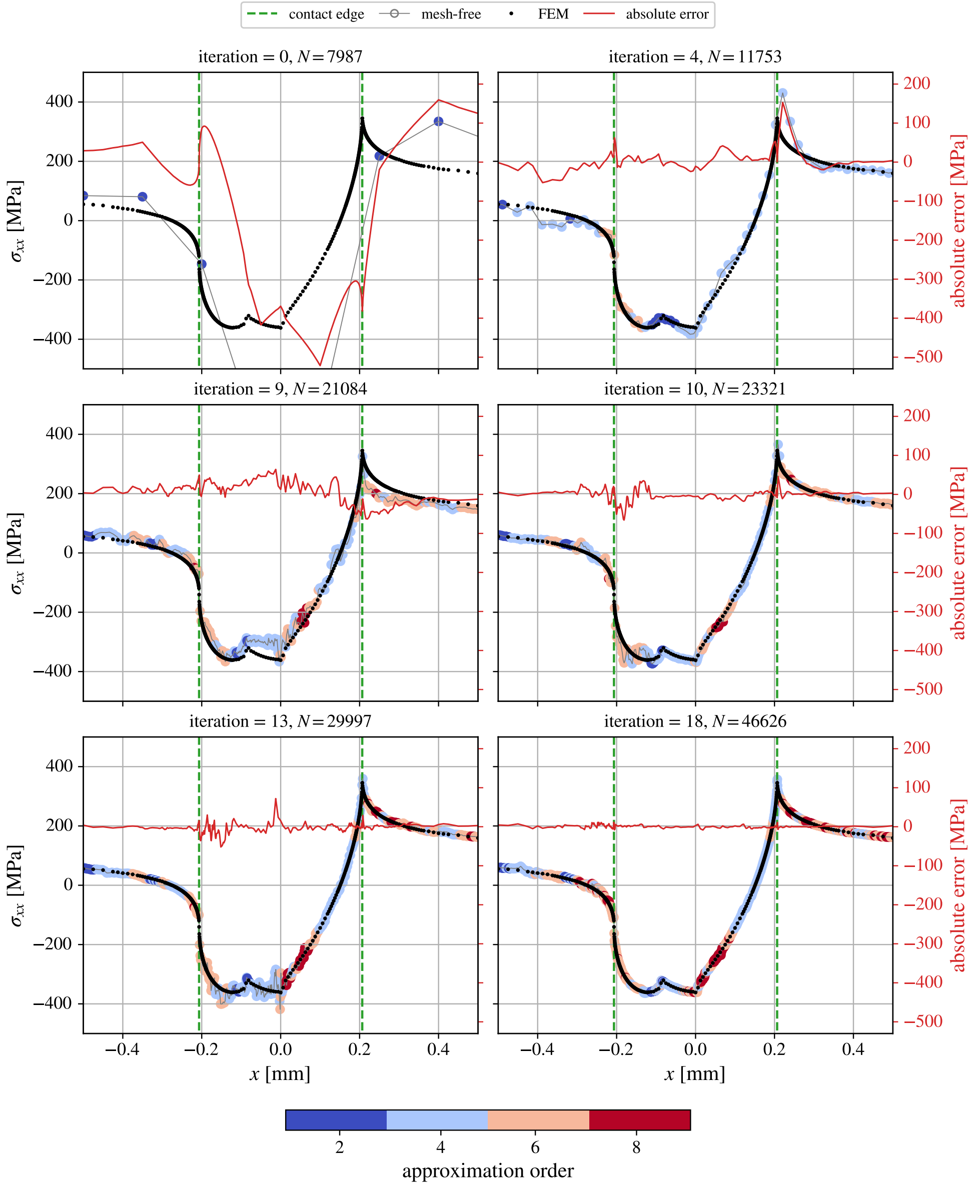


Fig. 12 Example *hp*-refined fretting fatigue contact solution

It is worth mentioning, that the final sparse system was solved using BiCGSTAB with ILUT preconditioner (employed with an initial guess obtained from the previous adaptivity iteration), where the global tolerance was set to 10^{-15} with a maximum number of 500 iterations and drop-tolerance and fill-factor set to 10^{-6} and 60 respectively. Other possible choices and their effect on the solution procedure are further discussed in Sect. 4.2.2.

Example *hp*-refined numerical solution is given in Fig. 16. We can see that the proposed *hp*-adaptive solution procedure is sufficiently robust to obtain a good solution even for three-dimensional problems with singularities. Additionally, we also observe that the IMEX error indicator successfully identified the singularity, effectively seen as an increase in the local field description in the neighbourhood of the concentrated force applied at the origin.

Fig. 13 Surface traction under the contact for selected iteration steps demonstrating the *hp*-adaptivity process. Colours are used to denote the local approximation orders. Numerical solution is additionally compared against the Abaqus FEM solution, where the red line is used to denote the absolute difference between the two methods. For clarity, the two dashed green lines show the edge contact



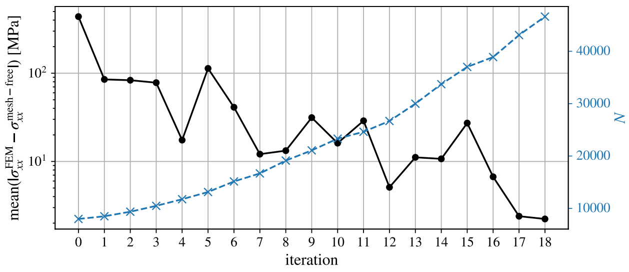


Fig. 14 Mean surface traction difference between the two methods under the contact area

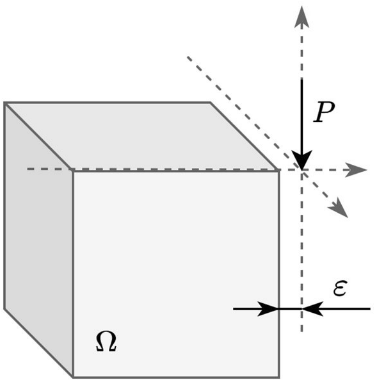


Fig. 15 Schematic presentation of Boussinesq's problem

4.2.1 The von Mises stress along the body diagonal

Figure 17 shows further evaluation of the *hp*-refined mesh-free numerical solution. Here, the von Mises stress at points near the body diagonal $(-1, -1, -1) \to (-\varepsilon, -\varepsilon, -\varepsilon)$ is calculated for selected 4 adaptivity iterations and compared to the analytical values in terms of relative error. In addition, the nodes are coloured according to the local approximation order enforced by the *hp*-adaptive solution procedure. We can see that the highest relative error of approximately 0.3 at the initial state is observed in the neighbourhood of the origin. In the final iteration, the relative error is reduced by about an order of magnitude. We also see that the *hp*-adaptive solution procedure has found a non-trivial order distribution and that the number of nodes in the neighbourhood of the corner $(-\varepsilon, -\varepsilon, -\varepsilon)$ has increased significantly.

A more quantitative analysis of the mesh-free solution is given in Fig. 18 where the ℓ_1, ℓ_2 and ℓ_∞ error norms and number of computational nodes vs. adaptivity iteration are shown. Compared to the initial state, the *hp*-adaptive solution procedure was able to achieve a numerical solution that was almost two orders of magnitude more accurate. In the process, the number of computational nodes increased from 10 500 in the initial state to about 80 000 in the final iteration. However, it is interesting to observe that with the configuration from Table 3, none of the computational nodes used the approximation with the highest order allowed ($m = 8$). Instead, in the final iteration, there were 130 nodes approximated with $m = 6$, and 5937 with $m = 4$, while the rest were approximated with the second order. Note that, as expected, most of the higher order approximations are near the concentrated force—which is difficult to represent visually, so we only give the descriptive data.

For reference, we take the *h*-refined solution by Slak et al. [10], who were able to reduce the infinity norm error by about an order of magnitude with $N \approx 140\,000$ nodes in the final iteration. It is perhaps naive to compare this result with ours, since the authors use different marking and refinement strategies and, more importantly, a different error indicator. Nevertheless, the infinity norm error of our *hp*-refined solution is in the neighbourhood of 10^{-3} compared to theirs at approximately 10^{-2} with almost twice as many computational nodes. We believe our results could be further improved by fine-tuning the free parameters, but we decided to avoid such an approach.

4.2.2 Additional discussion on solving the global sparse system

In all previous sections, we have completely neglected the importance of solving the global sparse system in the proposed *hp*-adaptive solution procedure with a suitable solver. However, inappropriate choice of solver can lead to inaccurate or even unstable behaviour and, most importantly, unreasonably large computational cost. To avoid such flaws, we compared an iterative BiCGSTAB and BiCGSTAB with ILUT preconditioner with two direct solvers—namely the SparseLU and the PardisoLU—on a *hp*-adaptive solution to the Boussinesq problem, performing 25 adaptivity iterations with approximately 10,000 initial nodes and 135,000 nodes after the last iteration. Note that the iterative BiCGSTAB solver with ILUT preconditioner was employed with an initial guess obtained from the previous adaptivity iteration.

In addition to the discussed solvers, we also tried the SparseQR. While its stability and accuracy were comparable to other solvers, its computational cost was significantly higher and was therefore removed from further analysis and from the list of potential candidates. For all performed tests we used the EIGEN linear algebra library [63].

Let us first examine the sparsity patterns of the systems assembled at different stages of the *hp*-adaptive process in Fig. 19, where we can see how the system increases in size and also becomes less sparse due to globally decreasing the internodal distance h and increasing the approximation order p. Additionally, the spectra of the matrices are shown in the bottom row of Fig. 19, where we can see that the ratios between the real and imaginary parts of the eigenvalues are in good agreement with previous studies [13, 14, 61].

Table 3 Adaptivity parameters used to obtain solution to Boussinesq's problem

β_h	α_h	λ_h	ϑ_h	β_p	α_p	λ_p	ϑ_p	h_{max}	N_{max}	N_{iter}
10^{-3}	10^{-3}	3.75	1.01	10^{-4}	10^{-2}	3	1.5	0.04	$7 \cdot 10^4$	20

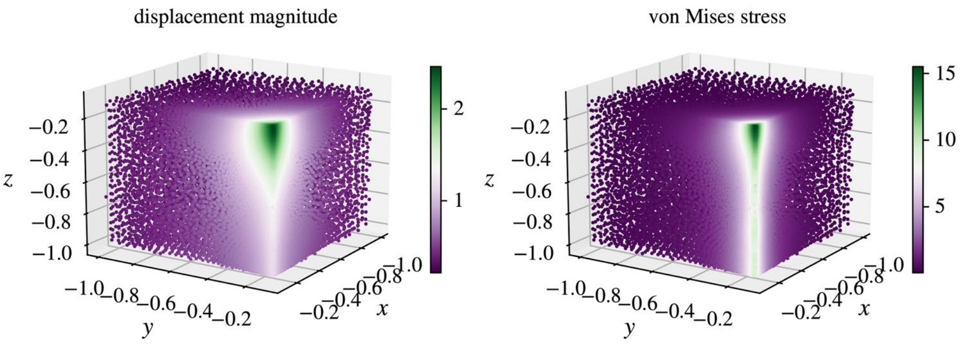


Fig. 16 Example *hp*-refined numerical solution to Boussinesq's problem

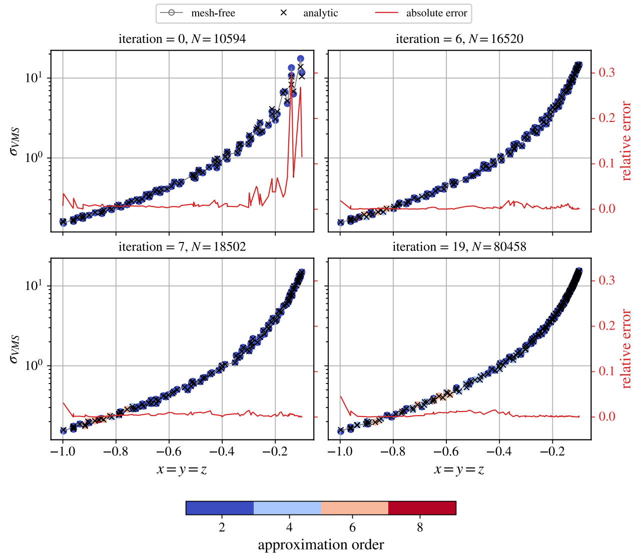


Fig. 17 Numerical solution compared to analytical solution at the nodes near the body diagonal $(-1,-1,-1) \rightarrow (-\varepsilon,-\varepsilon,-\varepsilon)$ for selected iterations

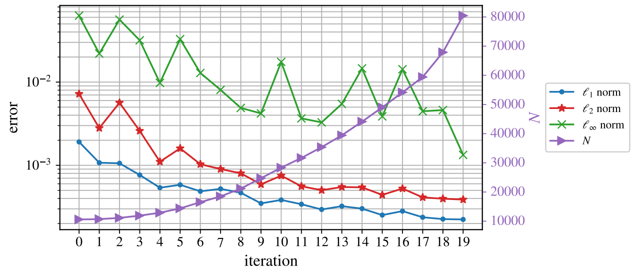


Fig. 18 Convergence of numerical solution along with number of computational nodes

Moreover, Fig. 20 presents three different views of the solvers' performance: (i) the achieved accuracy of the final solution for different solvers, (ii) the number of iterations a solver needs to converge, and (iii) the execution times of each solver, each with respect to the *hp*-adaptive iterations. The differences in final accuracy for different solvers are marginal. Perhaps the BiCGSTAB shows better stability behaviour (in terms of error scatter) compared to others. Nevertheless, it is important to observe, that the SparseLU only works until the 15th iteration with approximately 50 000 nodes, at which point our computer ran out of the available 12 Gb memory, which is to be expected due to the computational complexity or SparseLU. PardisoLU, on the other hand, remains stable through all adaptivity iterations.

Generally speaking, the number of iterations BiCGSTAB needs to converge increases with *hp*-adaptivity iterations due to the increasing non-zero elements in the global system. The BiCGSTAB with a ILUT precoditioner shows similar behaviour, but with approximately 2/3 less iterations required. Both direct solvers, of course, require only one "iteration". Finally, the analysis of the execution time shows that the PardisoLU solver is by far the most efficient among all considered candidates.

With all things considered, PardisoLU seems to be the the best candidate for *hp*-adaptive solution procedure. However, the last adaptivity iteration with approximately 115,000 nodes was coincidentally right at the limit of the available 12 Gb RAM memory—using approximately 10.5 Gb. It is therefore expected that like SparseLU, the PardisoLU would soon run out of memory for larger domains. To avoid such problems, we chose to work with a general purpose iterative BiCGSTAB solver with ILUT preconditioner employed with an initial guess, since it shows slightly better computational efficiency than the pure BiCGSTAB and required only 7.5 Gb of RAM for approximately 135,000 nodes in the final adaptivity iteration.

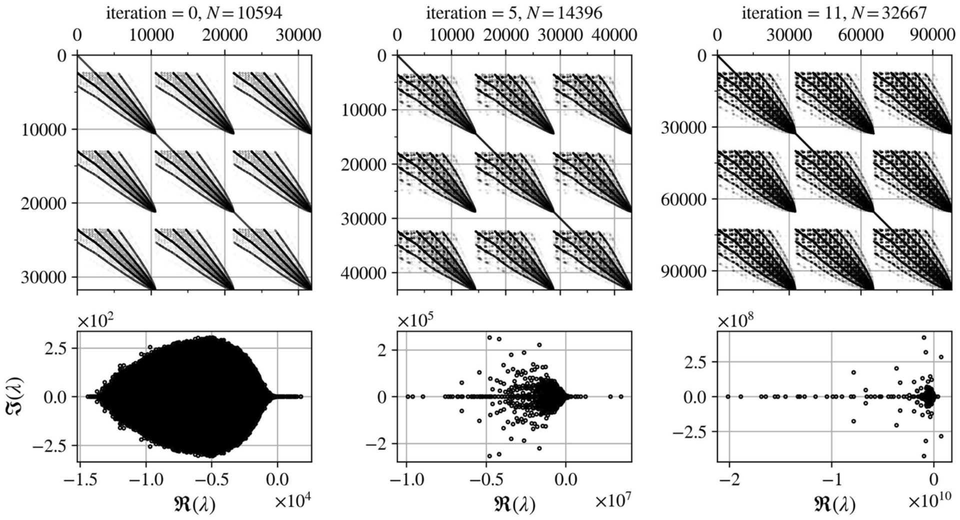


Fig. 19 Global sparse matrix plot (top row) and spectra of the matrices (bottom row) at three selected iterations of the *hp*-adaptive solution procedure. Note that the spectra are computed for the BiCGSTAB solver with an ILUT preconditioning using an estimate from the previous iteration

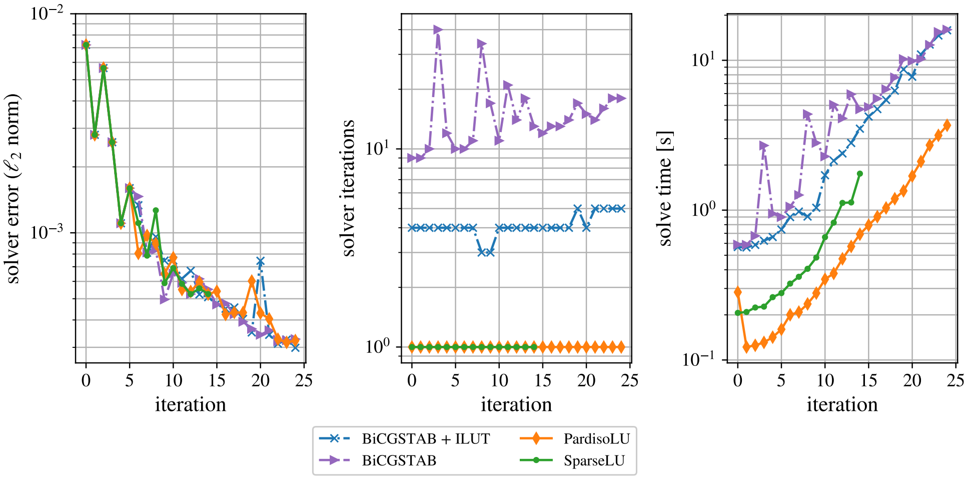


Fig. 20 Error of the final solution with respect to the adaptivity iteration for different solvers (left), number of solver iteration per adaptivity iteration (centre) and solver compute time for each adaptivity iteration (right)

5 Conclusions

In this paper we establish a baseline for *hp* strong form mesh-free analysis. We have formulated and implemented a *hp*-adaptive solution procedure and demonstrated its performance in three different numerical experiments.

The cornerstone of the presented *hp*-adaptive method is an iterative solve–estimate–mark–refine paradigm with the modified Texas Three Step marking strategy. The *h*-refine of the proposed method relies on an advancing front node positioning algorithm based on Poisson disc sampling, which enables dimension-independent node generation supporting spatially variable density distributions. For the adaptive order of the method, we exploit an elegant control over the order of the approximation via the augmenting monomials in the approximation basis.

We proposed an IMEX error indicator, where the implicit solution of the problem is processed with the higher order local explicit representation of PDE at hand, e.g. if the implicit solution is computed with a second order approximation, the explicit re-evaluation happens at fourth order. Our analyses show that the proposed error indicator successfully captures main characteristics of error distributions, which suffices for the proposed iterative adaptivity.

The proposed *hp*-adaptive solution procedure is first demonstrated on a two-dimensional Poisson problem with exponential source and mixed boundary conditions. Further demonstration focuses on linear elasticity problems. First, a 2D fretting fatigue problem – a contact problem with pronounced peaks in the surface stress, and second, a 3D Boussinesq's problem with stress singularity. In both cases, we have demonstrated the advantages of using the proposed *hp*-adaptive approach.

Although the *hp*-adaptivity introduces additional steps in the solution procedure and is therefore undoubtedly computationally more expensive per node than the non-adaptive approach, it is essential in problems that exhibit volatilities in solution in small regions of the domain. For example singularity in the contact problem require excessively detailed numerical analysis near the contact compared to the rest (the bulk) of the domain. Such cases are extremely difficult (or even impossible) to solve without adaptivity, since the minimal uniform h and p distribution required to capture these volatilities would lead to unreasonably high computational complexity. In cases of a smooth solution, however, the benefits of *hp*-adaptivity in most cases do not justify its computational overheads.

We are aware that there are many opportunities for improvement of presented methodology. The IMEX error indicator needs further clarification. Other error indicators should also be implemented and tested. During our experiments, we have found that a marking strategy with more free parameters leads to better accuracy, but is also more difficult to understand and control and can be case dependent. A smarter and more effective refinement and marking strategies are certainly part of future work. These should possibly take into account more information about the method itself, e.g. the dependence of the computational complexity on the approximation order, and most importantly local data regularity to choose between p and h refinement.

One of our goals in future work is also generalisation of the presented *hp*-adaptive solution procedure to time-dependent problems. The most straightforward approach to achieve that is to granularly adapt h and p throughout the simulation. In its simplest form, the proposed *hp*-adaptivity would be performed at each time step, starting with the *hp* distributions of the previous time step and using the same adaptivity parameters for all time steps. A more sophisticated approach would also take into account the desired accuracy during the simulation, resulting in time-dependent adaptivity parameters. For example, if one is only interested in a steady state solution, the desired accuracy would increase with time, reaching its maximum at steady state. Additionally, to perform proper adaptive analysis, the time step should also be adaptive, which requires an additional step in the *hp*-adaptive solution procedure.

Acknowledgements . The authors acknowledge the financial support from the Slovenian Research and Innovation Agency (ARIS) research core funding No. P2-0095, and research projects No. J2-3048 and No. N2-0275 (joint research project between National Science Centre, Poland and ARIS, where Polish research group is Funded by National Science Centre, Poland under the OPUS call in the Weave programme 2021/43/I/ST3/00228. This research was funded in whole or in part by National Science Centre (2021/43/I/ST3/00228). For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.).

Declarations

Conflict of interest The authors declare that they have no conflict of interest. All the co-authors have confirmed to know the submission of the manuscript by the corresponding author.

References

1. Upadhyay BD, Sonigra SS, Daxini SD (2021) Numerical analysis perspective in structural shape optimization: a review post 2000. Adv Eng Softw 155:102,992
2. Mitchell WF, McClain MA (2014) A comparison of hp-adaptive strategies for elliptic partial differential equations. ACM Trans Math Softw (TOMS) 41(1):1–39
3. Segeth K (2010) A review of some a posteriori error estimates for adaptive finite element methods. Math Comput Simul 80(8):1589–1600. https://doi.org/10.1016/j.matcom.2008.12.019. https://www.sciencedirect.com/science/article/pii/S0378475408004230. ESCO 2008 Conference
4. Liu GR, Gu YT (2005) An introduction to meshfree methods and their programming. Springer, Berlin
5. Liu GR (2002) Mesh free methods: moving beyond the finite element method. CRC Press, Boca Raton. https://doi.org/10.1201/9781420040586
6. Zienkiewicz OC, Taylor RL, Zhu JZ (2005) The finite element method: its basis and fundamentals. Elsevier, Amsterdam
7. van der Sande K, Fornberg B (2021) Fast variable density 3-d node generation. SIAM J Sci Comput 43(1):A242–A257
8. Shankar V, Kirby RM, Fogelson AL (2018) Robust node generation for mesh-free discretizations on irregular domains and surfaces. SIAM J Sci Comput 40(4):A2584–A2608
9. Jacquemin T, Suchde P, Bordas SP (2023) Smart cloud collocation: geometry-aware adaptivity directly from CAD. Comput Aided Design 154:103409. https://doi.org/10.1016/j.cad.2022.103409. https://linkinghub.elsevier.com/retrieve/pii/S0010448522001427
10. Slak J, Kosec G (2019) Adaptive radial basis function-generated finite differences method for contact problems. Int J Numer Methods Eng 119(7):661–686
11. Davydov O, Oanh DT (2011) Adaptive meshless centres and rbf stencils for poisson equation. J Comput Phys 230(2):287–304
12. Jacquemin T, Bordas SPA (2021) A unified algorithm for the selection of collocation stencils for convex, concave, and singular problems. Int J Numer Methods Eng 122(16):4292–4312
13. Jančič M, Slak J, Kosec G (2021) Monomial augmentation guidelines for rbf-fd from accuracy versus computational time perspective. J Sci Comput 87(1):1–18

14. Bayona V, Flyer N, Fornberg B, Barnett GA (2017) On the role of polynomials in rbf-fd approximations: Ii. numerical solution of elliptic pdes. J Comput Phys 332:257–273
15. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P (1996) Meshless methods: an overview and recent developments. Comput Methods Appl Mech Eng 139(1–4):3–47
16. Kosec G, Šarler B (2014) Simulation of macrosegregation with mesosegregates in binary metallic casts by a meshless method. Eng Anal Bound Elements 45:36–44. https://doi.org/10.1016/j.enganabound.2014.01.016. https://linkinghub.elsevier.com/retrieve/pii/S0955799714000290
17. Maksić M, Djurica V, Souvent A, Slak J, Depolli M, Kosec G (2019) Cooling of overhead power lines due to the natural convection. Int J Electrical Power Energy Syst 113:333–343. https://doi.org/10.1016/j.ijepes.2019.05.005. https://linkinghub.elsevier.com/retrieve/pii/S0142061518340055
18. Gui Wz, Babuska I (1985) The h, p and hp versions of the finite element method in 1 dimension. part 3. the adaptive hp version. Tech. rep., Maryland Univ College Park Lab for Numerical Analysis
19. Gui WZ, Babuška I (1986) The h, p and hp versions of the finite element method in 1 dimension. part ii. the error analysis of the h and hp versions. Numerische Mathematik 49(6):613–657
20. Devloo12 PR, Bravo CM, Rylo EC (2012) Recent developments in hp adaptive refinement
21. Zienkiewicz OC, Zhu JZ (1987) A simple error estimator and adaptive procedure for practical engineerng analysis. Int J Numer Methods Eng 24(2):337–357
22. González-Estrada OA, Natarajan S, Ródenas JJ, Bordas SP (2021) Error estimation for the polygonal finite element method for smooth and singular linear elasticity. Comput Math Appl 92:109–119
23. Thimnejad M, Fallah N, Khoei AR (2015) Adaptive refinement in the meshless finite volume method for elasticity problems. Comput Math Appl 69(12):1420–1443. https://doi.org/10.1016/j.camwa.2015.03.023
24. Angulo A, Pozo LP, Perazzo F (2009) A posteriori error estimator and an adaptive technique in meshless finite points method. Eng Anal Bound Elements 33(11):1322–1338. https://doi.org/10.1016/j.enganabound.2009.06.004
25. Oanh DT, Davydov O, Phu HX (2017) Adaptive rbf-fd method for elliptic problems with point singularities in 2d. Appl Math Comput 313:474–497
26. Sang-Hoon P, Kie-Chan K, Sung-Kie Y (2003) A posterior error estimates and an adaptive scheme of least-squares meshfree method. Int J Numer Methods Eng 58(8):1213–1250. https://doi.org/10.1002/nme.817
27. Afshar M, Naisipour M, Amani J (2011) Node moving adaptive refinement strategy for planar elasticity problems using discrete least squares meshless method. Finite Element Anal Design 47(12):1315–1325
28. Guo B, Babuška I (1986) The hp version of the finite element method. Comput Mech 1(1):21–41
29. Mitchell WF (2016) Performance of hp-adaptive strategies for 3d elliptic problems
30. Tinsley Oden J, Wu W, Ainsworth M (1995) In: Modeling, mesh generation, and adaptive numerical methods for partial differential equations (Springer), pp 347–366
31. Ainsworth M, Senior B (1997) Aspects of an adaptive hp-finite element method: adaptive strategy, conforming approximation and efficient solvers. Comput Methods Appl Mech Eng 150(1–4):65–87
32. Houston P, Senior B, Süli E (2003) In: Numerical mathematics and advanced applications (Springer), pp 631–656
33. Houston P, Süli E (2005) A note on the design of hp-adaptive finite element methods for elliptic partial differential equations. Comput Methods Appl Mech Eng 194(2–5):229–243
34. Eibner T, Melenk JM (2007) An adaptive strategy for hp-fem based on testing for analyticity. Comput Mech 39(5):575–595
35. Bürg M, Dörfler W (2011) Convergence of an adaptive hp finite element strategy in higher space-dimensions. Appl Numer Math 61(11):1132–1146
36. Demkowicz L, Rachowicz W, Devloo P (2002) A fully automatic hp-adaptivity. J Sci Comput 17(1):117–142
37. Rachowicz W, Pardo D, Demkowicz L (2006) Fully automatic hp-adaptivity in three dimensions. Comput Methods Appl Mech Eng 195(37–40):4816–4842
38. Benito J, Urena F, Gavete L, Alvarez R (2003) An h-adaptive method in the generalized finite differences. Comput Methods Appl Mech Eng 192(5–6):735–759
39. Liu G, Kee BB, Chun L (2006) A stabilized least-squares radial point collocation method (ls-rpcm) for adaptive analysis. Comput Methods Appl Mech Eng 195(37–40):4843–4861
40. Hu W, Trask N, Hu X, Pan W (2019) A spatially adaptive high-order meshless method for fluid-structure interactions. Comput Methods Appl Mech Eng 355:67–93
41. Tolstykh A, Shirobokov D (2003) On using radial basis functions in a "finite difference mode'' with applications to elasticity problems. Comput Mech 33(1):68–79
42. Oanh DT, Tuong NM (2022) An approach to adaptive refinement for the rbf-fd method for 2d elliptic equations. Appl Numer Math 178:123–154
43. Tóth B, Düster A (2022) h-adaptive radial basis function finite difference method for linear elasticity problems. Comput Mech:1–20
44. Fan L (2019) Adaptive meshless point collocation methods: investigation and application to geometrically non-linear solid mechanics. Ph.D. thesis, Durham University
45. Mishra PK, Ling L, Liu X, Sen MK (2020) Adaptive radial basis function generated finite-difference (rbf-fd) on non-uniform nodes using *p*-refinement. arXiv preprint arXiv:2004.06319
46. Milewski S (2021) Higher order schemes introduced to the meshless fdm in elliptic problems. Eng Anal Bound Elements 131:100–117
47. Albuquerque-Ferreira A, Ureña M, Ramos H (2021) The generalized finite difference method with third-and fourth-order approximations and treatment of ill-conditioned stars. Eng Anal Bound Elements 127:29–39
48. Liszka T, Duarte C, Tworzydlo W (1996) hp-meshless cloud method. Computer Methods Appl Mech Eng 139(1–4):263–288
49. Jančič M, Slak J, Kosec G (2021) In: 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech), pp 01–06. https://doi.org/10.23919/SpliTech52315.2021.9566401
50. Duarte CA, Oden JT (1996) An hp adaptive method using clouds. Computer Methods Appl Mech Eng 139(1–4):237–262
51. Slak J, Kosec G (2019) On generation of node distributions for meshless pde discretizations. SIAM J Sci Comput 41(5):A3202–A3229
52. Slak J, Kosec G (2021) Medusa: a c++ library for solving pdes using strong form mesh-free methods. ACM Trans Math Softw (TOMS) 47(3):1–25
53. Depolli M, Slak J, Kosec G (2022) Parallel domain discretization algorithm for RBF-FD and other meshless numerical methods for solving PDEs. Comput Struct 264:106773. (Publisher: Elsevier)
54. Duh U, Kosec G, Slak J (2021) Fast variable density node generation on parametric surfaces with application to mesh-free methods. SIAM J Sci Comput 43(2):A980–A1000
55. Wendland H (2004) Scattered data approximation. Cambridge Monographs on Applied and Computational Mathematics.

Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511617539

56. Davydov O, Oanh DT, Tuong NM (2023) Improved stencil selection for meshless finite difference methods in 3d. J Comput Appl Math 425:115031. https://doi.org/10.1016/j.cam.2022.115031. https://www.sciencedirect.com/science/article/pii/S037704272200629X
57. Jančič M, Strniša F, Kosec G (2022) In: 2022 7th International Conference on Smart and Sustainable Technologies (SpliTech), pp 01–04. https://doi.org/10.23919/SpliTech55088.2022.9854342
58. Heuer N, Mellado ME, Stephan EP (2001) hp-adaptive two-level methods for boundary integral equations on curves. Computing 67(4):305–334
59. Bayona V (2019) An insight into rbf-fd approximations augmented with polynomials. Comput Math Appl 77(9):2337–2353
60. Tominec I, Larsson E, Heryudono A (2021) A least squares radial basis function finite difference method with improved stability properties. SIAM J Sci Comput 43(2):A1441–A1471. https://doi.org/10.1137/20M1320079
61. Bayona V (2019) Comparison of moving least squares and rbf+poly for interpolation and derivative approximation. J Sci Comput 81(1):486–512
62. Daniel P, Ern A, Smears I, Vohralík M (2018) An adaptive hp-refinement strategy with computable guaranteed bound on the error reduction factor. Comput Math Appl 76(5):967–983
63. Guennebaud G, Jacob B, et al (2010) Eigen v3. http://eigen.tuxfamily.org
64. Wakeni MF, Aggarwal A, Kaczmarczyk L, McBride AT, Athanasiadis I, Pearce CJ, Steinmann P (2022) A p-adaptive, implicit-explicit mixed finite element method for diffusion-reaction problems. Int J Numer Methods Eng
65. Kosec G, Slak J, Depolli M, Trobec R, Pereira K, Tomar S, Jacquemin T, Bordas SP, Wahab MA (2019) Weak and strong from meshless methods for linear elastic problem under fretting contact conditions. Tribol Int 138, 392–402. (Publisher: Elsevier)
66. Slaughter WS (2002) Three-dimensional problems. Birkhäuser, Boston, pp 331–386. https://doi.org/10.1007/978-1-4612-0093-2_9

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.