

## RESEARCH ARTICLE

# Adaptive Radial Basis Function-generated Finite Differences method for contact problems

Jure Slak<sup>1,2</sup> | Gregor Kosec\*<sup>1</sup>

<sup>1</sup>Parallel and Distributed Systems  
Laboratory, "Jožef Stefan" Institute,  
Slovenia

<sup>2</sup>Faculty of Mathematics and Physics,  
University of Ljubljana, Slovenia

**Correspondence**

\*Gregor Kosec, Email: gregor.kosec@ijs.si

**Present Address**

Department E6  
"Jožef Stefan" Institute  
Jamova ulica 39  
1000 Ljubljana  
Slovenia

**Summary**

This paper proposes an original adaptive refinement framework using Radial Basis Functions-generated Finite Differences method. Node distributions are generated with a Poisson Disk Sampling-based algorithm from a given continuous density function, which is altered during the refinement process based on the error indicator. All elements of the proposed adaptive strategy rely only on meshless concepts, which leads to great flexibility and generality of the solution procedure. The proposed framework is tested on four gradually more complex contact problems. First, a disk under pressure is considered and the computed stress field is compared to the closed form solution of the problem to assess the behaviour of the algorithm and the influence of free parameters. Second, a Hertzian contact problem is studied to analyse the proposed algorithm with an ad-hoc error indicator and to test both refinement and derefinement. A contact problem, typical for fretting fatigue, with no known closed form solution is considered and solved next. It is demonstrated that the proposed methodology produces results comparable with FEM without the need for manual refinement or any human intervention. In the last case, generality of the proposed approach is demonstrated by solving a 3-D Boussinesq's problem.

**KEYWORDS:**

Linear elasticity, Radial Basis Functions, Adaptive  $h$ -refinement, Cauchy Navier equations, Contact problems, Meshfree methods

## 1 | INTRODUCTION

Solving partial differential equations (PDEs) with Radial basis functions (RBFs) began in 1990 when Kansa suggested that besides scattered data interpolation, RBFs could also be used for solving PDEs<sup>1</sup>. In the beginning, only global collocation methods were used, achieving high order convergence and spectral accuracy<sup>2</sup>, however, at the cost of high computational complexity and possible ill-conditioning in solution of the global system, especially as the number of computational nodes increases. Shortcomings of the global methods encouraged the development of local approaches and soon several meshless methods based on approximation over local overlapping support domains appeared<sup>3,4</sup>. A popular variant among many local meshless methods is the Radial Basis Function-generated Finite Differences (RBF-FD) method<sup>5</sup>, which uses finite difference-like collocation weights on an unstructured set of nodes. The method has been successfully used in several problems and is still actively researched<sup>6,7,8</sup>. In the field of solid mechanics, where problems are traditionally tackled with the Finite Element Method (FEM)<sup>9</sup>, meshless methods surfaced as a response to the cumbersome meshing of realistic 3D domains required by FEM<sup>8,10,11,12</sup>. The development

began with weak form methods such as the Element Free Galerkin Method<sup>13</sup> and Meshless Local Petrov Galerkin Method<sup>14</sup>, followed by strong form collocation approach that emerged with Finite Point Method (FPM)<sup>15</sup> and further developed with similar local strong form methods<sup>8,11</sup>. One of the most attractive features of meshless methods is that they do not require an underlying mesh and fully define relations between nodes directly through their relative inter nodal positions, which allows to reduce the problem of constructing a mesh into a much simpler problem of positioning scattered nodes<sup>16,17</sup>. This simplification makes meshless methods good candidates for adaptive numerical analysis.

Adaptive refinement of domain discretisation is indispensable in problems where varying precision and resolution of the numerical solution are required in different areas of the computational domain. A typical example of such a problem in linear elasticity is a contact of two bodies where extremely high stress concentrations are present under and around the contact area. Solving such problems on uniform grids is intractable due to the amount of time and computational resources required. To obtain a numerical solution with satisfactory precision in the contact area within a reasonable time frame, significantly more nodes have to be placed under the contact than elsewhere. Since the areas of high stresses are generally not known a priori, and to eliminate the need for human intervention in construction of discretisations, a commonly adopted approach in such situations is to refine the discretisation iteratively, gradually increasing the resolution of the numerical solution in areas of interest with the ultimate goal to ensure relatively uniform error distribution throughout the domain, or in many cases solely to solve the problem. In addition to the PDE solution procedure itself, two more modules are required in an adaptive method, namely an indicator of the solution quality (an error indicator), and an appropriate refinement strategy.

The most widely accepted error indicator has been introduced by Zienkiewicz and Zhu in 1987<sup>18</sup> for FEM solutions of elasticity problems. The basic idea behind the proposed indicator, also known as  $ZZ$  or  $Z^2$  is based on the assumption that the error of the solution can be approximated as the difference between the numerically obtained solution and a recovered solution, i.e. a more accurate solution computed by appropriate post-processing. This approach has been later successfully extended to the meshless solutions of elasticity problems, both weak form, using meshless finite volume method<sup>19</sup>, NS-PIM<sup>20</sup>, and strong form, using FPM<sup>21</sup>. In RBF-FD context a  $ZZ$  type of error indicator has been discussed in a solution of Laplace equation in<sup>22</sup>. An alternative class of error indicators is available for commonly used least squares-based meshless methods, which relies on the least squares approximation residual<sup>23</sup>. The residual based error indicator has been successfully used in a meshless solution of an elasticity problem with a Discrete Least Squares Meshless method<sup>24</sup>. Furthermore, for specific problems, ad-hoc meshless error indicators have been reported in literature, inspired by a physical interpretation of the solution. These include indicators such as a maximal difference of values within the support domain, effectively acting as a maximal first derivative of the considered field<sup>25</sup>, or variance of the field values within the support domain<sup>26</sup>.

Once the knowledge about the error of the solution is extracted, two conceptually different adaptive approaches can be employed, namely the  $p$ -refinement or  $h,r$ -refinement. In the  $p$ -refinement scheme solution accuracy is varied by changing the order of the approximation. In meshless methods this results in an increase of the number of shape functions, modifications of the shape functions or an increase of the influence domain<sup>27</sup>. Second approach is to modify discretization of the domain, either with the  $r$ -refinement scheme, where nodes are repositioned to improve the approximation, leaving the total number of the nodes unchanged, or with  $h$ -refinement, which influences the total number of nodes. Successful  $r$ -refinement has been demonstrated in a meshless solution of elasticity problems with discrete least square method<sup>24</sup>, on the solution of an infinite plate with a circular hole and a circular disk under pressure. On the other hand,  $h$ -refinement changes the total number of nodes by placing more points in areas where needed and removing them from areas that are overpopulated or by recreating a new distribution of nodes based on the information from the original distribution. In mesh based methods, the adding and removing approach is also known as *mesh enrichment*, while creating new distribution approach is commonly referred to as a *re-meshing*<sup>19</sup>. In the meshless context,  $h$ -refinement has been successfully used with the global RBF Collocation Method<sup>28</sup> in a solution of nearly singular PDEs, as well as with local strong form meshless methods in a solution of Burgers' equation<sup>26</sup> and torsion problem<sup>29</sup>. It has also been successfully applied in meshless solutions of elasticity problems<sup>8,19,20,30,31</sup>.

Even though the existence of refinement algorithms seems to be an inherent property of all meshless methods, one has to be careful when interfering with the nodal configuration to avoid potential negative effects such as ill conditioning and other possible complications that arise with non smooth nodal distributions<sup>32,33</sup>. This task becomes even more challenging in a fully adaptive approach where the refinement algorithm and numerical solution have to be robust enough to fulfil the demands of the error indicator. In several published  $h$ -refinement meshless solutions, the authors used Voronoi diagrams, the dual construction to Delaunay triangulations, for insertion of new nodes<sup>23,19,20,30,21</sup>. This approach is often not feasible for 3-D problems and can be computationally demanding.

Davydov and Oanh<sup>25</sup> discussed a pure meshless  $h$ -refinement algorithm for strong form RBF-FD solution of Poisson equation. In their proposed algorithm, new nodes are added at midpoints between existing nodes. Since the algorithm does not guarantee smooth distributions, required by RBF-FD<sup>17</sup>, authors had to use special stencil selection algorithms to improve stability<sup>22</sup>. Similar  $h$ -refinement algorithm achieving a ratio of  $2^{17}$  between the densest and coarsest parts of the discretisation has been recently applied to elasticity problems<sup>8</sup>. No special stencil selection algorithms were needed in this case; however, a few steps of regularisation were needed to sufficiently improve the distribution.

This paper continues the evolution of adaptive meshless methods in the field of linear elasticity by presenting a robust purely meshless adaptive RBF-FD solution of contact problems in absence of meshing at any point of the solution procedure. The refinement procedure is also independent on the dimensionality of the considered domain, which is demonstrated by solving numerical examples from linear elasticity with pronounced differences in stress within the domain in 2-D and 3-D. The proposed methodology uses *re-meshing* type  $h$ -refinement with an ad-hoc error indicator.

The performance of the proposed adaptive method is demonstrated on four cases, governed by Navier-Cauchy equations, gradually increasing in complexity. In the first case, a disk under pressure is considered, which can be solved in closed form and is therefore ideal for testing, since the exact error can be computed during refinement. The second case considers Hertzian contact where an ad-hoc error indicator type<sup>25,26</sup> based on the characteristics of the underlying problem is used. The third case deals with numerical computation of stresses under fretting fatigue conditions, i.e. a contact with slip-stick zone that has been recently investigated with finite element analysis<sup>34</sup>. The last case serves as a presentation of the generality and effectiveness of the proposed algorithm in 3-D by solving the Boussinesq's problem of concentrated normal traction acting on an isotropic half-space.

All examples in this paper were computed using the in-house open source Medusa library<sup>35</sup>. All research data is freely available in a git repository<sup>36</sup>.

The rest of the paper is organized as follows: the RBF-FD method is described in section 2, the adaptivity algorithm and other relevant components are described in section 3, the results on four increasingly difficult cases are presented in section 4 and finally, the conclusions are presented in section 5.

## 2 | RBF-GENERATED FINITE DIFFERENCES

The RBF-FD method is used to obtain the discrete formulation of a PDE. Consider an elliptic boundary value problem with Dirichlet boundary conditions

$$\mathcal{L}u = f \quad \text{on } \Omega \quad (1)$$

$$u = u_0 \quad \text{on } \partial\Omega, \quad (2)$$

where  $f$  and  $u_0$  are known functions. To obtain a discrete representation,  $N$  nodes are placed in  $\Omega$ , of which  $N_i$  nodes in the interior and  $N_b$  nodes on the boundary. Let the set of all discretisation nodes be denoted by  $\mathcal{P}$ . Each node  $p_i$  is assigned  $n_i$  neighbours (including itself), denoted  $\mathcal{N}(p_i)$ . These neighbours are commonly called the node's *support domain* or, analogous to the Finite Difference Method (FDM), its *stencil*.

The operator  $\mathcal{L}$  at a point  $p_i$  is approximated as a weighted linear combination of function values at stencil points

$$(\mathcal{L}u)(p_i) \approx \sum_{p_j \in \mathcal{N}(p_i)} w_j^i u(p_j). \quad (3)$$

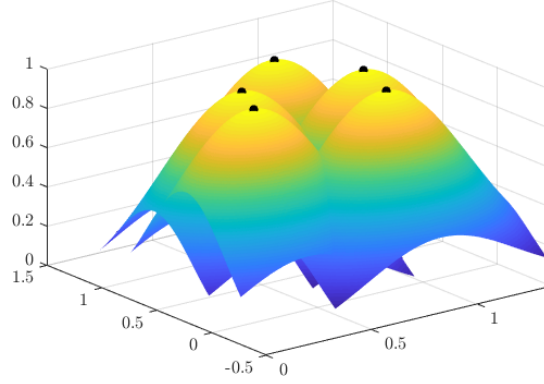
To determine the weights  $w_j^i$ , exactness of (3) is imposed for a certain set of functions. A common choice are monomials, as used in the FDM and FPM, however, in RBF-FD method, as the name suggests, the Radial Basis Functions (RBFs) are used. RBFs, positioned in centres  $P = \{p_1, \dots, p_n\}$  are functions

$$\{\varphi_i := \varphi(\|\cdot - p_i\|), \text{ for } p_i \in P\}, \quad (4)$$

where  $\varphi : [0, \infty) \rightarrow \mathbb{R}$  is an arbitrary function. In this work Gaussian RBFs will be used, defined as

$$\varphi_i(p) = \exp(-(\|p - p_i\|/\sigma_b)^2),$$

where  $\sigma_b$  stands for shape parameter. The Gaussian RBF are illustrated in Fig. 1. Many other types are known and used, with various recommendations<sup>37</sup>.



**FIGURE 1** Gaussian radial basis functions with  $\sigma_b = 0.5$  over a set of randomly generated nodes.

Imposing exactness of (3) for  $\varphi_k$ , we can write

$$(\mathcal{L}\varphi_k)(p_i) = \sum_{p_j \in \mathcal{N}(p_i)} w_j^i \varphi_k(p_j), \quad (5)$$

for all  $p_k \in \mathcal{N}(p_i)$ . Assembling all equations given by (5) in a matrix, the following system of equations is obtained:

$$\begin{bmatrix} \varphi(\|p_{j_1} - p_{j_1}\|) & \cdots & \varphi(\|p_{j_{n_i}} - p_{j_1}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|p_{j_1} - p_{j_{n_i}}\|) & \cdots & \varphi(\|p_{j_{n_i}} - p_{j_{n_i}}\|) \end{bmatrix} \begin{bmatrix} w_{j_1}^i \\ \vdots \\ w_{j_{n_i}}^i \end{bmatrix} = \begin{bmatrix} (\mathcal{L}\varphi_{j_1})(p_i) \\ \vdots \\ (\mathcal{L}\varphi_{j_{n_i}})(p_i) \end{bmatrix}, \quad (6)$$

where  $j_k$  are indices of nodes in  $\mathcal{N}(p_i)$ . Equation (6) is a system of  $n_i$  linear equations, compactly written as  $A_i \mathbf{w}_i = \mathbf{b}_i$ , where matrix  $A_i$  is symmetric. Furthermore, the properties of such collocation matrices arising from radial basis functions have been well studied<sup>38,37</sup>. If positive definite RBFs, such as Gaussian RBFs, are used, the matrices  $A_i$  are positive definite as soon as all nodes  $p_j$  are distinct<sup>37</sup>. Furthermore, to ensure consistency up to a certain order, RBF-FD approximation can be augmented with monomials. These consistency constraints are enforced using Lagrangian multipliers, resulting in a system similar to (6). For example, if a constant 1 is added to the basis set, the system for computing the weights reads

$$\begin{bmatrix} \varphi(\|p_{j_1} - p_{j_1}\|) & \cdots & \varphi(\|p_{j_{n_i}} - p_{j_1}\|) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \varphi(\|p_{j_1} - p_{j_{n_i}}\|) & \cdots & \varphi(\|p_{j_{n_i}} - p_{j_{n_i}}\|) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} w_{j_1}^i \\ \vdots \\ w_{j_{n_i}}^i \\ \lambda_1^i \end{bmatrix} = \begin{bmatrix} (\mathcal{L}\varphi_{j_1})(p_i) \\ \vdots \\ (\mathcal{L}\varphi_{j_{n_i}})(p_i) \\ 0 \end{bmatrix}, \quad (7)$$

effectively adding an additional constraint  $\sum_{i=1}^{n_i} w_{j_i} = 0$ . The effects of adding polynomial constraints to RBF-FD approximations have been studied recently by Bayona et al.<sup>7</sup>. In cases discussed in this paper, additional consistency constraints were not needed to obtain satisfying results. Another possible modification to the RBF-FD above procedure is to include more stencil points than basis functions, putting basis functions only on the closest  $m_i$  nodes. This turns (6) into an underdetermined system, however, it can be solved uniquely by imposing additional condition of minimizing the norm  $\|\mathbf{w}_i\|$  of the weights. The benefit of this approach is that is less sensitive to perturbations of nodal positions<sup>8</sup>. This modification will be used in this paper.

After computing the coefficients  $\mathbf{w}_i$  for all nodes, a sparse  $N \times N$  matrix with  $\sum_{i=0}^N n_i$  elements representing the governing problem can be assembled. The right hand side is computed from given  $f$  and  $u_0$  and the sparse system is solved to obtain an approximation for  $u$ . Vector PDEs are treated as a coupled system of scalar PDEs, resulting in a proportionally larger system. Boundary conditions that involve differential operators, such as Neumann or traction boundary conditions, are discretised analogously to operator  $\mathcal{L}$  above.

## 2.1 | Remarks on weight computation and shape parameter $\sigma_b$

It is worth noting that the selection of the shape parameter  $\sigma_b$  requires some attention. A known problem with computing the weights  $\mathbf{w}_i$  is that as the shape parameter  $\sigma_b$  tends towards  $\infty$ , the accuracy increases, but the matrices  $A_i$  become more and more ill conditioned. A common solution is to scale the shape parameter  $\sigma_b$  proportionally to the internodal distance and thus

keep condition numbers of  $A_i$  constant even as the node density increases. Another possible complication arises from the Runge Phenomenon, which comes into play with large  $\sigma_b$  and variable node densities. To avoid that, once again, a spatially variable shape parameter has been used as proposed by Fornberg and Zuev<sup>33</sup> and Flyer and Lehto<sup>39</sup>.

A downside to using a spatially variable shape parameter, is that the matrices  $A_i$  (Eq. 6) lose their symmetry properties. To retain the symmetry, the shape parameter is kept constant for all entries in each  $A_i$ , but varies with  $i$  as  $\sigma_{b,i} = \sigma_b \delta r_i$ , where  $\delta r_i$  is the distance to the closest neighbouring node to  $p_i$ . Another downside is that scaled shape parameters may cause stagnation errors and failure of convergence<sup>40</sup>, however, no such problems were detected in cases discussed in this paper and scaling the shape parameter sufficed to suppress the Runge Phenomenon and mitigate ill-conditioning.

### 3 | ADAPTIVITY

The concept of the adaptively refined solution of a PDE in this work is as follows: initially discretise the domain with (usually) uniform nodal distribution, compute approximate solution  $u$ , compute new desired nodal density function based on the previous density and the input from the chosen error indicator, redistribute the nodes and repeat as long as the error is not within desired range. The refinement procedure presented in this paper is built on the following preliminaries.

- (i) An a-posteriori error indicator  $\mathfrak{d}$  is known, where  $\mathfrak{d}(p)$  represents an indicator of error between an actual and computed value of some desired quantity at a point  $p$ . For convenience we define  $\mathfrak{d}(p_i) = \mathfrak{d}_i$  for all  $p_i \in \mathcal{P}$ . In a more general setting  $\mathfrak{d}$  can also represent any kind of refinement indicator, with larger values indicating areas where refinement is needed and lower values indicating areas where it is unnecessary or even derefinement is possible.
- (ii) Domain is initially filled with a known density  $\delta r^{(0)}$ .
- (iii) Refinement is an iterative process that alters current radius function  $\delta r$  with respect to the error indicator.
- (iv) The coarseness limit for  $\delta r$  is  $\delta r^u$ , i.e. the nodal radius during derefinement is bounded by  $\delta r_i^u$  from above.
- (v) The nodal radius function  $\delta r$  is changed with respect to the density increase factor ( $f_i$ ) that is evaluated based on the error indicator  $\mathfrak{d}_i$ , the maximal ( $\epsilon$ ) and minimal ( $\eta$ ) allowed error, and maximal ( $M$ ) and minimal ( $m$ ) value of the error indicator. The increase factor is limited to  $f \in [1/\beta, \alpha]$ , where  $\alpha, \beta \geq 1$ .

In the following subsections the parts of the above procedure are described in more detail.

#### 3.1 | Node placement

Although mesh generation is not needed in RBF-FD, certain care still has to be taken to avoid too close node placements, which cause ill-conditioning. In other words, we seek a distribution of nodes that conforms to the treated domain shape and target nodal density, with the nodes also being distributed as regularly as possible.

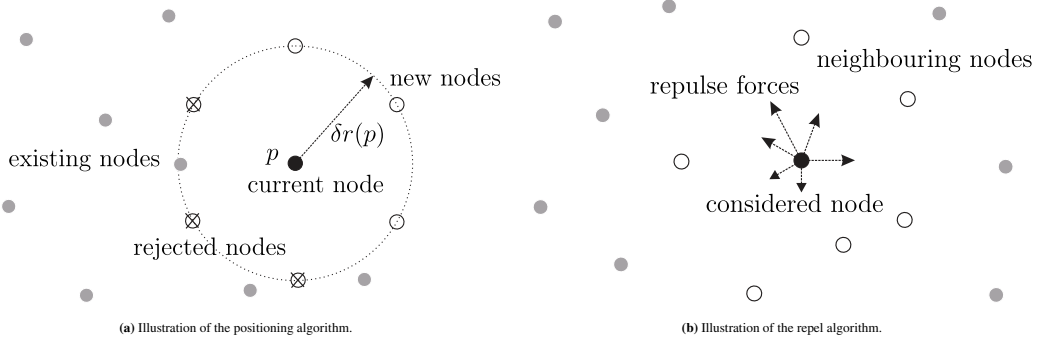
The basis for the node placement procedure is the Poisson Disc Sampling algorithm<sup>41</sup>. The procedure begins with adding a starting “seed node” into the domain, followed by an iterative spreading of the populated area of the domain. In each iteration new nodes are added regularly on a sphere (or circle) with centre in a randomly selected node  $p$  that has not been processed yet and radius  $\delta r(p)$ . The function  $\delta r$  is an external parameter and its value at point  $p$  represents the expected distance between neighbouring nodes around  $p$ , characterising the nodal density at that point. Some of these newly added nodes might be too close to the already positioned nodes, and thus a simple rejection criterion is used: if the distance to the closest node is closer than  $\zeta \cdot \delta r(p)$ , for  $\zeta \in [0, 1]$ , the node is removed. The proximity factor  $\zeta$  must be used due to numerical errors in distance computations, however its use also produces smoother distributions. It is typically set to 0.9 or 0.99. The algorithm is schematically presented in Fig. 2a.

After the domain is populated, the distribution is further regularised with an iterative algorithm that “diffuses” the nodes as if a repelling force was induced by their neighbours. The force is determined by summing the gradients of potentials in surrounding nodes, where the amplitude of the force is further normalised by the distance to the closest neighbour and  $\delta r$ . The algorithm also uses primitive “simulated annealing” techniques, i.e. the magnitude of translation is multiplied with an annealing factor that is

linearly dropping from  $Q_i$  to  $Q_f$ . More precisely, in each node a repulsing force is computed as

$$d(i) = -\xi \left( \frac{N_r - n_r}{N_r} (Q_i - Q_f) + Q_i \right) \sum_{p_j \in \mathcal{N}(p_i)} \left( \frac{\delta r(p_j)}{\|p_j - p_i\|} \right)^\kappa (p_j - p_i), \quad (8)$$

where  $\xi$  stands for normalization factor, usually distance to the closest node,  $N_r$  represents the number of regularization iterations,  $n_r$  current iteration,  $Q_i$  initial “heat”,  $Q_f$  final “heat”,  $\kappa$  potential power and  $\mathcal{N}(p)$  are neighbours of  $p$ . Typical set-up is  $N_r \in [3, 10]$ ,  $Q_i \in [0.8, 1.2]$ ,  $Q_f \in [0, 0.2]$ ,  $\kappa = 2$  and 3 closest neighbours are considered. The algorithm is schematically presented in Fig. 2b.



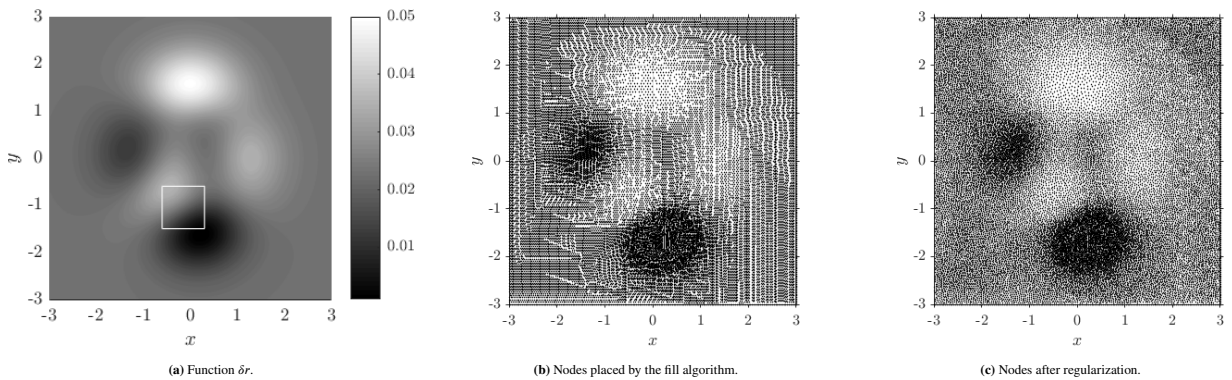
**FIGURE 2** Scheme of algorithms used for domain discretisation.

A simple demonstration of above algorithms is presented in Fig. 3. A scaled and shifted peaks function from Matlab was used as  $\delta r$ , specifically,

$$\delta r(x, y) = \delta x + (\Delta x - \delta x)(\text{peaks}(x, y) + 6.55)/14.66, \quad (9)$$

$$\text{peaks}(x, y) = 3(1 - x)^2 e^{-x^2 - (y+1)^2} - 10(x/5 - x^2 - y^5) e^{-x^2 - y^2} - \frac{1}{3} e^{-(x+1)^2 - y^2} \quad (10)$$

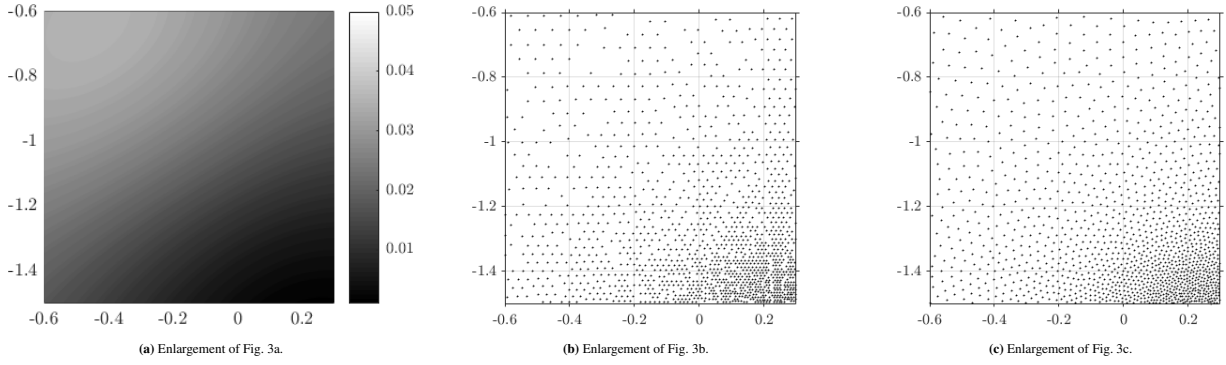
with  $\delta x = 0.007$  and  $\Delta x = 0.07$ . A square region  $[-3, 3] \times [-3, 3]$  was discretised, filled and regularised with described algorithms. The function  $\delta r$  is shown in Fig. 3a, and the initial and improved distributions are shown in Figures 3b and 3c, respectively. Zoomed-in versions are presented in Fig. 4 for easier visual assessment of the node quality.



**FIGURE 3** Illustration of domain discretisation with given function.

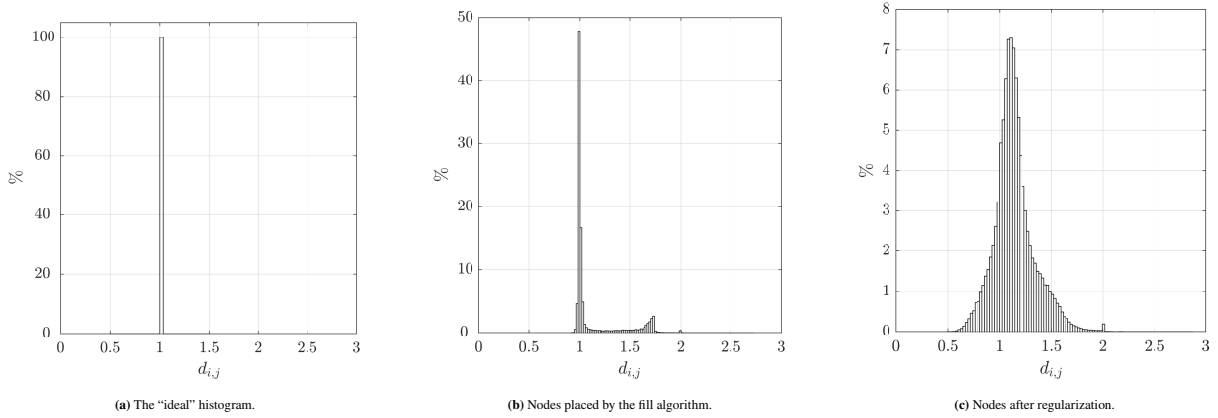
To present a quantitative measure of the node quality, the distances to the first layer of closest nodes were computed for each node. Specifically, if  $\{p_{i,j}\}_{j=1}^6$  are the six closest neighbours of  $p_i$  (excluding  $p_i$ ), the normalized distances  $d_{i,j}$  were computed as

$$d_{i,j} = \|p_{i,j} - p_i\| / \delta r(p_i). \quad (11)$$



**FIGURE 4** Enlarged versions of plots in Fig. 3.

Figure 5 shows the histograms of normalized distances  $d_{i,j}$  for node distributions before and after regularisation. The nodal distribution before regularisation is close to the “ideal” continuous case, where all  $d_{i,j}$  would be equal to 1. The tail peak of this histogram at around 1.7 has been investigated and corresponds to the nodes at the gaps in the distribution. A small portion of  $d_{i,j}$  at  $d_{i,j} = 2$  correspond to the nodes at the boundary where some of the six nodes fall in the second layer of closest nodes. The histogram of the regularised distribution is smoother with a peak at  $d_{i,j} = 1.2$  still corresponding to the nodes at the boundary. The peak at 1.7 before the regularisation, which corresponded to the gaps in the distribution, has been removed at the cost of greater spread.



**FIGURE 5** Histogram of normalized distances to 6 closest nodes.

Node generation is relatively undemanding from the computational point of view. The most expensive part of the fill algorithm is determining whether there are any proximity violations, which requires searching the closest nodes. However, this can be optimized with appropriate dynamic tree structures and managing the candidate list of possible nearby nodes. The regularization algorithm is a bit more expensive since it requires a full neighbourhood search in each iteration, but typically few iterations are needed to smooth out the irregularities.

Further analyses of the filling algorithm are out of the scope of this paper, but an interested reader is referred to a paper<sup>42</sup>, which provides a more quantitative analysis of the fill algorithm along with comparison to other existing techniques.

### 3.2 | Local density modification

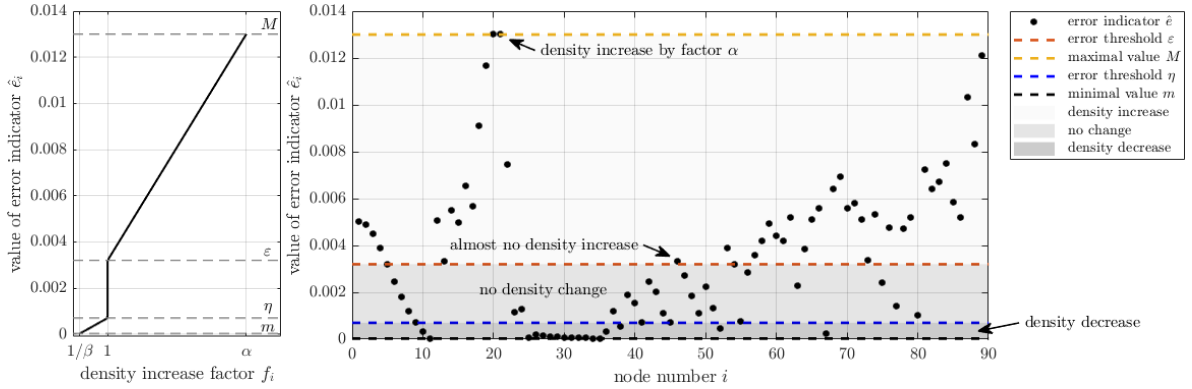
After the numerical solution of the PDE has been computed and the error indicator computed, the local nodal density is altered. This is achieved by computing new nodal radii  $\tilde{\delta r}_i$  from the existing ones as

$$\tilde{\delta r}_i = \min(\delta r_i / f_i, \delta r_i^u), \quad (12)$$

where  $\delta r_i$  is the distance to from  $p_i$  to the closest neighbouring node. This imposes the upper bound on radii from point (iv) from the preliminaries above. Similarly, a lower bound  $\delta r^\ell$  limiting refinement could also be imposed, however in practice that bound was not needed. The factor  $f_i$  is called a *density increase factor* and represents the relative change in nodal density at point  $p_i$ . It is computed as

$$f_i = \begin{cases} 1 + \frac{\eta - \mathfrak{d}_i}{\eta - m} \left( \frac{1}{\beta} - 1 \right), & \mathfrak{d}_i \leq \eta, \quad \text{i.e. decrease the density} \\ 1, & \eta < \mathfrak{d}_i < \varepsilon, \quad \text{i.e. no change in density} \\ 1 + \frac{\mathfrak{d}_i - \varepsilon}{M - \varepsilon} (\alpha - 1), & \mathfrak{d}_i \geq \varepsilon, \quad \text{i.e. increase the density} \end{cases} \quad (13)$$

where  $M = \max_{p_i \in \mathcal{P}} \mathfrak{d}_i$  is the maximal and  $m = \min_{p_i \in \mathcal{P}} \mathfrak{d}_i$  the minimal value of the error indicator. The plot of factor  $f_i$  with respect to the error indicator  $\mathfrak{d}_i$  can be seen in Fig. 6 (left).



**FIGURE 6** Construction of new density function  $\tilde{\delta r}$  from  $\delta r$  and  $\mathfrak{d}$ .

Note that  $\frac{1}{\beta} \leq f_i \leq \alpha$  always holds, as  $\frac{\mathfrak{d}_i - \varepsilon}{M - \varepsilon} \in [0, 1]$ , and  $\alpha \geq 1$ . If the node is on the error threshold, i.e.  $\mathfrak{d}_i = \varepsilon$ , the factor  $f_i$  equals 1 and the density will stay the same, ensuring compatibility with the case when  $\mathfrak{d}_i < \varepsilon$ . Additionally, if the node has the highest error, i.e.  $\mathfrak{d}_i = M$ , the increase factor will be maximal, i.e.  $\alpha$ . Symmetric observations hold for the derefine case. Setting  $\alpha = 1$  or  $\beta = 1$  disables refinement or derefinement, respectively.

Once the desired nodal radii  $\tilde{\delta r}_i$  have been evaluated in all nodes, a new continuous function  $\tilde{\delta r}$  is reconstructed using Modified Sheppard's method as described in the next section. The overall density modification procedure is illustrated conceptually in Fig. 6.

### 3.3 | Computing the new density function

After new desired local radii  $\tilde{\delta r}_i$  (12) have been computed, a continuous density function suitable for the fill algorithm in section 3.1 needs to be constructed. To construct a function  $\delta r(p)$  from discrete values  $\tilde{\delta r}_i$ , the Modified Sheppard's method is used. In a special case of  $\alpha = \beta = 1$  this becomes an inverse problem of the node positioning problem described in section 3.1. The problem of computing the desired internodal distance at an arbitrary point  $p \in \mathcal{P}$  translates to a problem of scattered data interpolation with known values  $\tilde{\delta r}_i$  at points  $p_i$ . Standard methods for this are well developed, described and compared by Franke<sup>43</sup>. Modified Shepard's method<sup>43, p. 185</sup> was used in this paper, where value of  $\delta r$  at a point  $p$  is computed as a weighted sum of  $n$  closest neighbours of  $p$

$$\delta r(p) = \begin{cases} \tilde{\delta r}_i, & \text{if } p = p_i \text{ for some } i \\ \frac{\sum_{i=1}^n w_i(p) \tilde{\delta r}_i}{\sum_{i=1}^n w_i(p)} & \text{else} \end{cases} \quad (14)$$

and the inverse distance weights  $w_i(p)$  are defined by

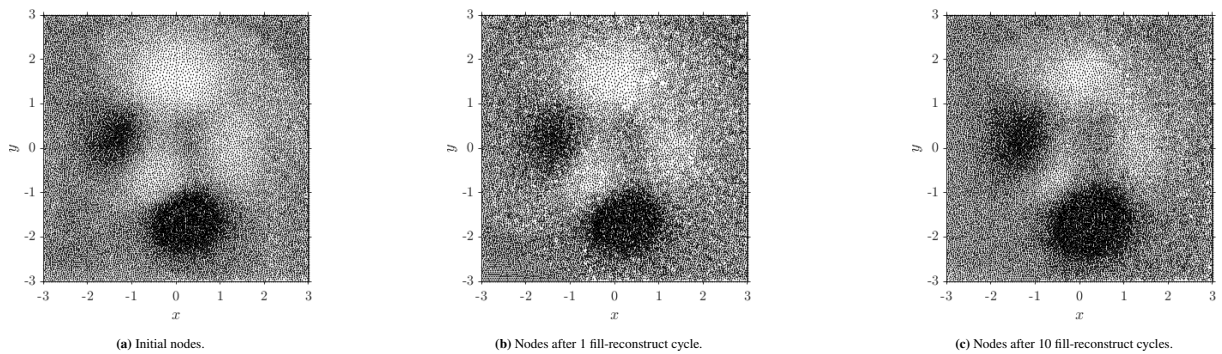
$$w_i(p) = \left( 1 - \frac{d_i}{d_n} \right) \frac{1}{d_i} \quad (15)$$

where  $d_i$  represents the Euclidean distance from  $p$  to its  $i$ -th neighbour and the neighbours are sorted according to  $d_i$ , thus making  $d_n$  the largest.

There is however no need for interpolation of values  $\tilde{\delta r}_i$ , as they themselves are only an approximation, and techniques from scattered data approximation<sup>44</sup> might be more appropriate. We have tested different approximation and interpolation methods for values  $\delta r_i$  with no significant changes in characteristics of node distributions upon refilling the domain. Modified Sheppard's method was chosen based on computational efficiency and ease of implementation.

### 3.3.1 | Analysis of repeated fill and reconstruct cycles

Every iteration of the adaptive procedure consists of filling the domain, solving the problem, and reconstructing and adapting the node density. Even without adaptation, these fill-reconstruction cycles do not guarantee the preservation of nodal density and it is therefore important to analyse their behaviour before implementing this approach into the refinement algorithm. To that end a continuation of the test case from section 3.1 was performed to better assess the effects of the repeated application of fill-reconstruction cycles. The effects of 10 successive cycles on the initial nodal distribution are demonstrated in Fig. 7c.



**FIGURE 7** Analysis of fill-reconstruct cycles.

Repeated application of fill-reconstruction cycles tends to slightly increase the nodal density, due to the proximity factor that is set to  $\zeta = 0.99$ . After 1 iteration, the minimal internodal distance is reduced from 0.00699727 to 0.00697768 and after 10 iterations to 0.00688613, i.e. a reduction of 1.5% over the course of 10 iterations, which is well within the allowed limit of 1% per iteration, as imposed by the fill tolerance factor  $\zeta$ . The most visible effect of successive application is the gradual spread and smear of high density areas, which expand with every successive iteration. This feature would be undesirable in certain situations and can be dealt with, however in numerical examples presented in this paper the phenomenon did not have considerable impact and is therefore not discussed further.

## 3.4 | Algorithm

The presented adaptive procedure is summarized as Algorithm 1. The DISCRETIZE procedure refers to the fill algorithm described in section 3.1, the SOLVE procedure refers to the application of the RBF-FD method for numerical solution of PDEs as described in section 2 and the ADAPT procedure refers to the local density modification and function reconstruction as described in sections 3.2 and 3.3.

## 3.5 | An illustrative adaptivity example

To help illustrate the proposed adaptive framework we use it on a simple test case of function approximation. The function

$$g(x) = 3(1 - x)^2 \exp(-x^2) + 3 \exp(-4(x - 1)^2) \quad (16)$$

is approximated on a given set of points using moving weighted least squares with basis  $\{1, x\}$ , 12 support nodes and Gaussian weight  $\exp(-x^2/d^2)$ , where  $d$  is the distance to the farthest support node. The resulting approximation is denoted  $\mathcal{f}$ .

**Algorithm 1** Adaptive solution procedure

**Input:** The problem, computation domain  $\Omega$ , initial density function  $\delta r : \Omega \rightarrow \mathbb{R}$ , the maximal number of iterations  $I_{\max}$  and adaptivity parameters  $\varepsilon, \eta, \alpha, \beta, \delta r^\mu$ .

**Output:** The numerical solution of the problem.

```

1: function ADAPTIVE_SOLVE(problem,  $\Omega$ ,  $\delta r$ ,  $I_{\max}$ ,  $\varepsilon$ ,  $\eta$ ,  $\alpha$ ,  $\beta$ ,  $\delta r^\mu$ )
2:   for  $i \leftarrow 0$  to  $I_{\max}$  do
3:      $\mathcal{P} \leftarrow$  DISCRETISE( $\Omega$ ,  $\delta r$ ) ▷ Discretises domain  $\Omega$  as described in section 3.1.
4:     solution  $\leftarrow$  SOLVE(problem,  $\mathcal{P}$ ) ▷ Solves the problem using discretisation  $\mathcal{P}$ , see section 2.
5:     error  $\leftarrow$  ESTIMATE_ERROR(solution,  $\mathcal{P}$ ) ▷ Error indicator computation, see (28) or (31).
6:     if MEAN(error)  $\geq \varepsilon$  then ▷ Other error reductions such as MAX can be used.
7:       return solution
8:     end if
9:      $\delta r \leftarrow$  ADAPT( $\delta r$ ,  $\mathcal{P}$ ,  $\varepsilon$ ,  $\eta$ ,  $\alpha$ ,  $\beta$ ,  $\delta r^\mu$ ) ▷ Adapt the discretisation as described in section 3.2.
10:  end for
11:  return solution
12: end function

```

Initially, the domain is filled with density

$$\delta r(x) = h_0(1 + 25|3 + x|), \quad h_0 = 0.005, \quad (17)$$

which results in a dense distribution on the negative part of the  $x$ -axis and a coarse distribution on the positive part (see Fig. 8, iteration 0). The upper bound for  $\delta r$  is set to  $\delta r^\mu(x) = 10h_0$ . The error is computed simply by taking the absolute difference between the actual and approximation value, multiplied by the local nodal spacing,  $e_i = |g(x_i) - \mathcal{g}(x_i)|\delta r(x_i)$ . Fig. 8 shows 3 iterations of the Algorithm 1 with error thresholds  $\varepsilon = 10^{-3}$ ,  $\eta = 10^{-4}$ , and aggressivenesses  $\alpha = 4$  and  $\beta = 4$ .

Initially, the function  $\mathcal{g}$  approximates  $g$  well around  $x = -3$  and poorly for  $x > -1$ , which is also reflected in the density increase factor  $f_i$ . It is greater than 1 and proportional to the error on the positive side, reaching a maximum of  $\alpha$ , where the error is greatest. Around  $x = -3$  the function  $g$  is approximated well, the error is below  $\eta$  and  $f_i$  is less than one and dropping to  $1/\beta$ , indicating derefinement is required. The node distribution in iteration 1 is adapted accordingly, with denser areas around peaks. This iteration continues and the  $L_1$  error  $\|g - \mathcal{g}\|_1$ , shown in figure titles, decreases accordingly. The number of nodes, along with the counts of refined and derefined nodes, is shown in Table 1.

**TABLE 1** Number of nodes during the course of adaptive iteration shown in Fig. 8.

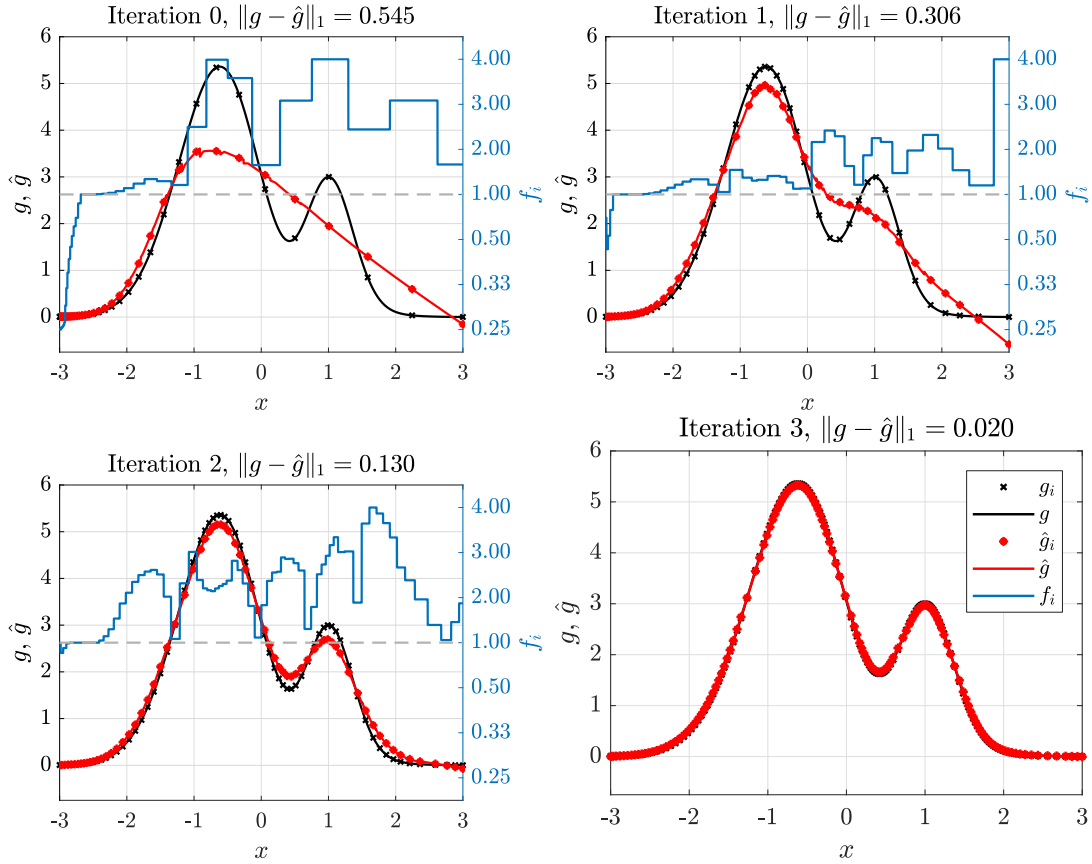
iteration	total nodes	refined	no change	derefined	derefined, but hit bound $\delta r^\mu$
0	41	18	4	11	8
1	50	38	8	2	2
2	72	59	10	1	2
3	159	/	/	/	/

## 4 | NUMERICAL RESULTS

### 4.1 | Governing problem

The solution procedure presented in previous section is applied to linear elasticity contact problems, more precisely to problems of deformation of elastic homogeneous isotropic materials under load. The problem is governed by Cauchy-Navier equations

$$(\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) + \mu\nabla^2 \mathbf{u} = \mathbf{f}, \quad (18)$$



**FIGURE 8** Illustration of adaptive function approximation using Algorithm 1.

where  $\mathbf{u}$  are unknown displacements,  $\mathbf{f}$  is the loading body force, and  $\lambda$  and  $\mu$  are Lamé parameters, often expressed in terms of Young's modulus  $E$  and Poisson's ratio  $\nu$ . Another important quantity is the stress tensor  $\sigma$ , related to strain via Hooke's law

$$\sigma = \lambda \operatorname{tr}(\epsilon)I + 2\mu\epsilon, \quad \epsilon = \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^T}{2}, \quad (19)$$

where  $\lambda$  and  $\mu$  are Lamé parameters from above and  $I$  is the identity tensor.

Two types of boundary conditions are considered: essential boundary conditions which specify displacements on some portion of the boundary of the domain, i.e.  $\mathbf{u} = \mathbf{u}_0$ , and traction boundary conditions, which prescribe surface traction  $\sigma \mathbf{n} = \mathbf{t}_0$ , with  $\mathbf{n}$  being an outside unit normal to the boundary of the domain.

In two dimensions, we will use the following componentwise notation for  $\mathbf{u}$  and  $\sigma$ , for sake of simplicity:

$$\mathbf{u} = (u, v) \quad \text{and} \quad \sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}. \quad (20)$$

## 4.2 | Disk under pressure

To demonstrate the adaptive strategy from section 3.2, a 2D example from linear elasticity, inspired by<sup>24</sup>, is chosen. A disk of radius  $R$  is subjected to diametrical compression with a point force of magnitude  $P$ , as illustrated in Fig. 9.

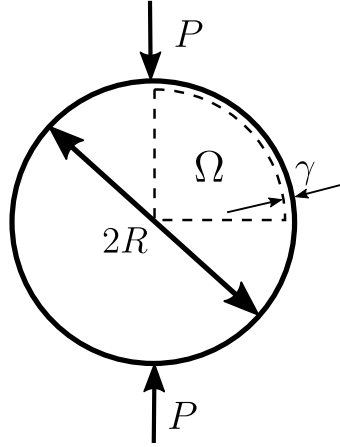


FIGURE 9 Disk under diametrical compression.

Exact closed form solution for stresses, assuming plane stress condition, is known, given by e.g. Sadd<sup>45, p. 197</sup> as:

$$\sigma_{xx} = -\frac{2P}{\pi} \left( \frac{x^2(R-y)}{r_1^4} + \frac{x^2(R+y)}{r_2^4} - \frac{1}{2R} \right) \quad (21)$$

$$\sigma_{yy} = -\frac{2P}{\pi} \left( \frac{(R-y)^3}{r_1^4} + \frac{(R+y)^3}{r_2^4} - \frac{1}{2R} \right) \quad (22)$$

$$\sigma_{xy} = \frac{2P}{\pi} \left( \frac{x(R-y)^2}{r_1^4} - \frac{x(R+y)^2}{r_2^4} \right) \quad (23)$$

where  $r_{1,2} = \sqrt{x^2 + (R \mp y)^2}$  are the distances to the poles. The stress distribution is singular at the poles.

Numerically, only a quarter of the disk is considered due to symmetry. Additionally, a small boundary layer of width  $\gamma$  is subtracted from the curved edge of the domain, to avoid the singularity at the poles. Therefore, the computational domain  $\Omega$ , shown in Fig. 9, is a quarter of a circle with radius  $R - \gamma$ . The parameter  $\gamma$  determines the peak stress magnitudes of order  $\frac{P}{2\pi\gamma}$  concentrated around the point  $x = 0, y = R - \gamma$ , allowing fine control over case difficulty. Thus the following problem is solved:

$$(\lambda + \mu)\nabla(\nabla \cdot \boldsymbol{\sigma}) + \mu\nabla^2 \boldsymbol{\sigma} = 0 \quad \text{in} \quad \Omega = \{(x, y) \in \mathbb{R}^2; x \geq 0, y \geq 0, x^2 + y^2 \leq (R - \gamma)^2\}. \quad (24)$$

Traction boundary conditions are imposed on the curved boundary, given by equations (21–23). Symmetry boundary conditions  $v = 0, \frac{\partial u}{\partial y} = 0$  and  $u = 0, \frac{\partial v}{\partial x} = 0$  are imposed on the bottom and left boundary, respectively.

Three possible values for  $\gamma$  will be considered,  $\gamma_1 = 0.2, \gamma_2 = 0.02$  and  $\gamma_3 = 0.002$ . The values of the stresses along the antidiagonal  $x = R - \gamma - y$  of  $\Omega$  are shown in Fig. 10.

The error between analytical stresses  $\sigma$  and computed stresses  $\boldsymbol{\phi}$  is measured in three different ways:

- Relative approximate  $L^\infty$  norm:

$$e_\infty = \frac{\|\boldsymbol{\phi} - \sigma\|_\infty}{\|\sigma\|_\infty}, \quad \|\sigma\|_\infty = \max_{p \in \Omega_h} \max\{\sigma_{xx}(p), \sigma_{yy}(p), \sigma_{xy}(p)\} \quad (25)$$

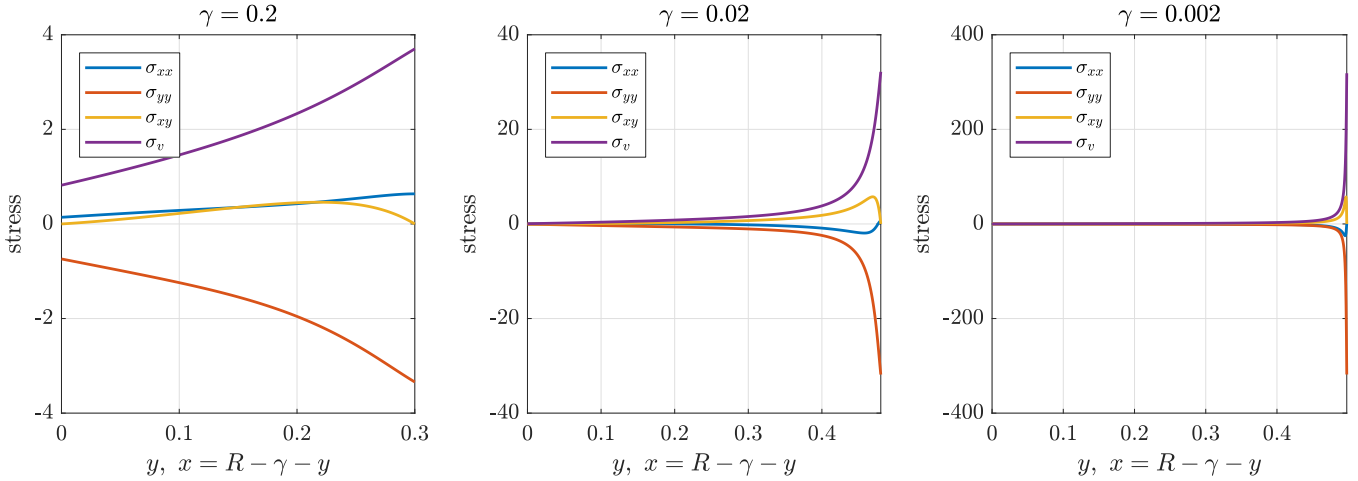
- Relative approximate  $L^1$  norm:

$$e_1 = \frac{\|\boldsymbol{\phi} - \sigma\|_1}{\|\sigma\|_1}, \quad \|\sigma\|_1 = \frac{1}{3|\Omega_h|} \sum_{p \in \Omega_h} (|\sigma_{xx}(p)| + |\sigma_{yy}(p)| + |\sigma_{xy}(p)|) \quad (26)$$

- Relative energy norm:

$$e_E = \frac{\|\boldsymbol{\phi} - \sigma\|_E}{\|\sigma\|_E}, \quad \|\sigma\|_E^2 = \int_{\Omega} \mathcal{E}_E(p) d\Omega, \quad \mathcal{E}_E(p) = \sigma(p) : S : \sigma(p), \quad (27)$$

where  $S$  is the fourth-order compliance tensor computed from  $E$  and  $\nu$ .



**FIGURE 10** Stress values along the antidiagonal  $x = R - \gamma - y$  for three different cases.

The norms are evaluated by discretising  $\Omega$  with a dense uniform grid  $\Omega_h$  with grid spacing  $h = 5 \cdot 10^{-4}$ , resulting in around 800 000 grid points. For  $e_\infty$  and  $e_1$  the maximum and the sum are taken over these grid points. The intermediate values of  $\sigma(p)$  for  $p \notin \mathcal{P}$  were evaluated using linear interpolation over the Delaunay triangulation induced by the points in  $\mathcal{P}$ . The integral in  $e_E$  was approximated using trapezoidal rule over  $\Omega_h$ .

Adaptive approach, described in section 3.2 is now employed to solve problem (24). The error indicator is constructed from the known analytical solution (21–23), using the kernel of the integral for energy norm (27) multiplied by the approximation of the volume element

$$\mathfrak{d}(p) = (\sigma(p) : S : \sigma(p)) \delta r(p)^d, \quad (28)$$

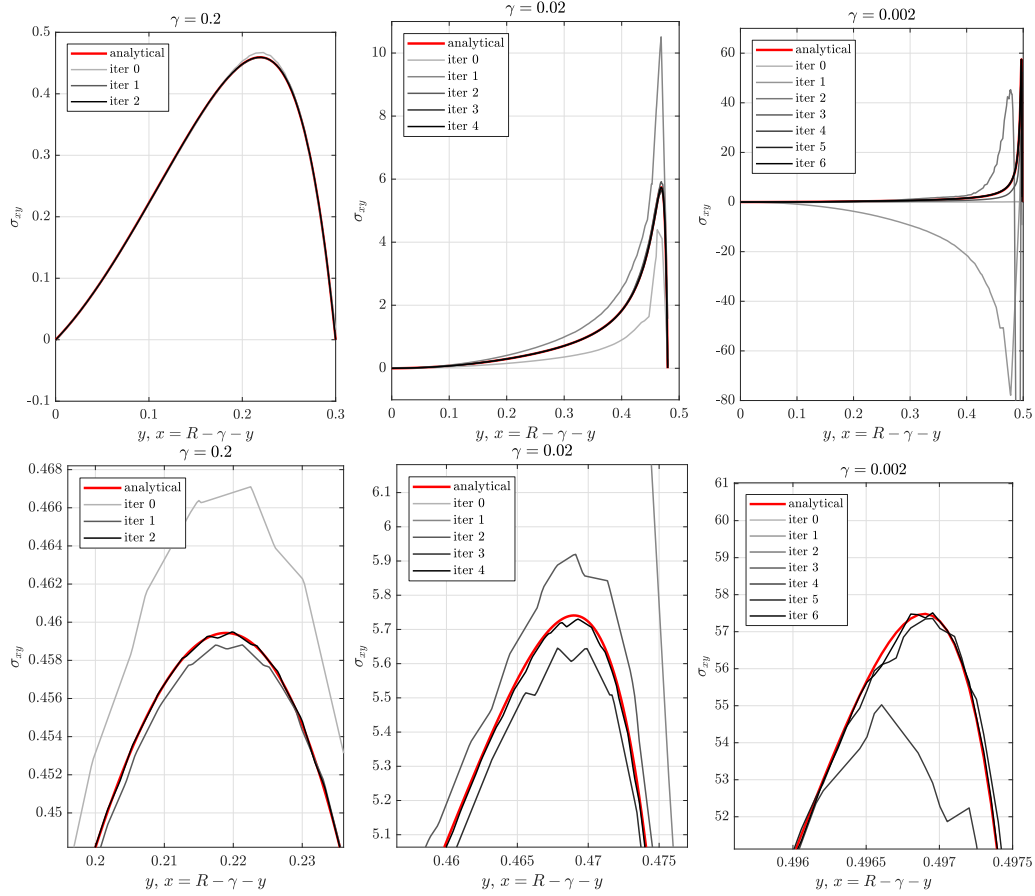
where  $d = 2$  is the dimension of the problem.

To discretise the domain, fill algorithm from section 3.1 with  $\zeta = 0.9$  was used and the distribution was regularised using repel algorithm with  $N_r = 10$  iterations,  $Q_i = 0.8$ ,  $Q_j = 0$ ,  $\kappa = 2$  and 3 neighbours. The domain was initially filled with constant density of  $\delta r^{(0)} = 0.02$ , amounting to approximately  $N = 600$  nodes in the domain. For the given discretisation, RBF-FD with 25 neighbouring nodes and 15 Gaussian RBFs was used. The shape parameter  $\sigma_b$  was varied proportionally to the internodal distance with base value  $\sigma_b = 100$ . For adaptive iteration,  $\varepsilon = 10^{-7}$ ,  $\eta = 10^{-9}$ ,  $\alpha = 5$  and  $\beta = 1.5$  were chosen. The reconstruction and evaluation of the new density function  $\tilde{\delta r}$  was done using 7 closest neighbours. Values of  $R = 0.5$ ,  $P = 1$ ,  $E = 1$  and  $\nu = 0.33$  were taken for physical parameters of the problem.

The overall procedure seems to depend on a lot of free parameters, however most are either well studied, such as the RBF shape parameters (see section 2.1) or their value is not very important to the overall procedure, such as the fill, repel and reconstruct parameters, described in more detail in sections 3.1 and 3.3. Standard values, as described above and as used by Slak and Kosec<sup>8</sup>, were taken for these parameters and they were kept the same for all runs, to avoid hyper-tuning the results. The parameters related to adaptivity are further analysed in section 4.3 and justify the choices above.

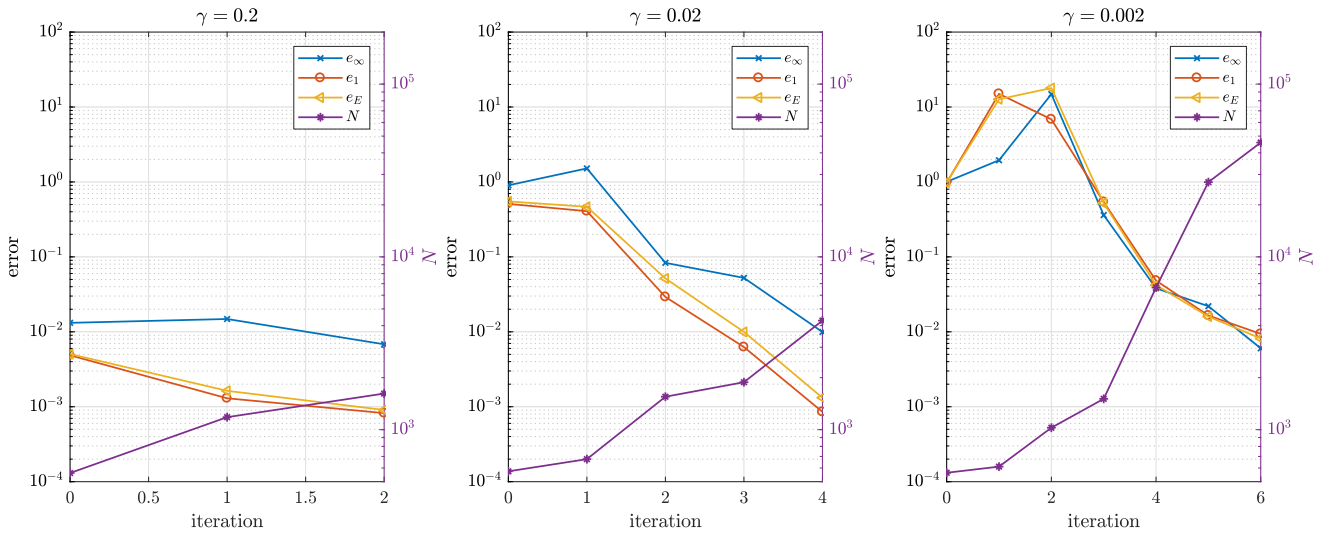
The stress profiles computed with the presented adaptive procedure for all three cases are shown in Fig. 11. To avoid cluttering, only  $\sigma_{xy}$  stress component is shown, but all behave similarly. In the first case, the initial distribution is almost sufficiently dense and thus adaptive procedure finishes after 2 iterations. The lower panel still reveals gradual improvement in both iterations. In the second case, the initial solution underestimates the peak stress value, the next iteration overestimates it, but then the profiles start to match with the analytic profile, as seen in the lower panel. This behaviour is even more prominent in the final case, where the initial distribution covers high stress area very coarsely and the first few iterations are only beginning to accurately cover the narrow stress peak. The lower panel once again shows that in the later iterations, when the solution behaviour has been well captured, the procedure behaves similarly to the other cases.

The errors over the course of the adaptive iteration are shown in Fig. 12 and confirm the behaviour observed in Fig. 11. All three error norms decrease as the algorithm proceeds, however no guarantee of monotonicity is given and sudden jumps in error can happen. In the second and more prominently in the third case, an initial rise in error can be observed, caused by low initial nodal densities. However in few iterations, the algorithm increases the density enough to accurately detect the error regions. Note that the error threshold  $\varepsilon$  has no direct connection to the final errors observed, only a general rule that lowering  $\varepsilon$  will also



**FIGURE 11** Computed  $\sigma_{xy}$  stress profiles for three considered cases. The lower three images show close-ups of the computed peaks.

decrease the measured errors holds. The number of nodes is also shown and behaves as expected: mode difficult cases require more iterations to obtain the final solution, which also uses more nodes for harder cases.



**FIGURE 12** Error of the numerical solution of the compressed disk problem during adaptive iteration.

To better visualise the adaptive results, a comparison between initial and final values of errors, error indicator and nodal densities are shown in Fig. 13.

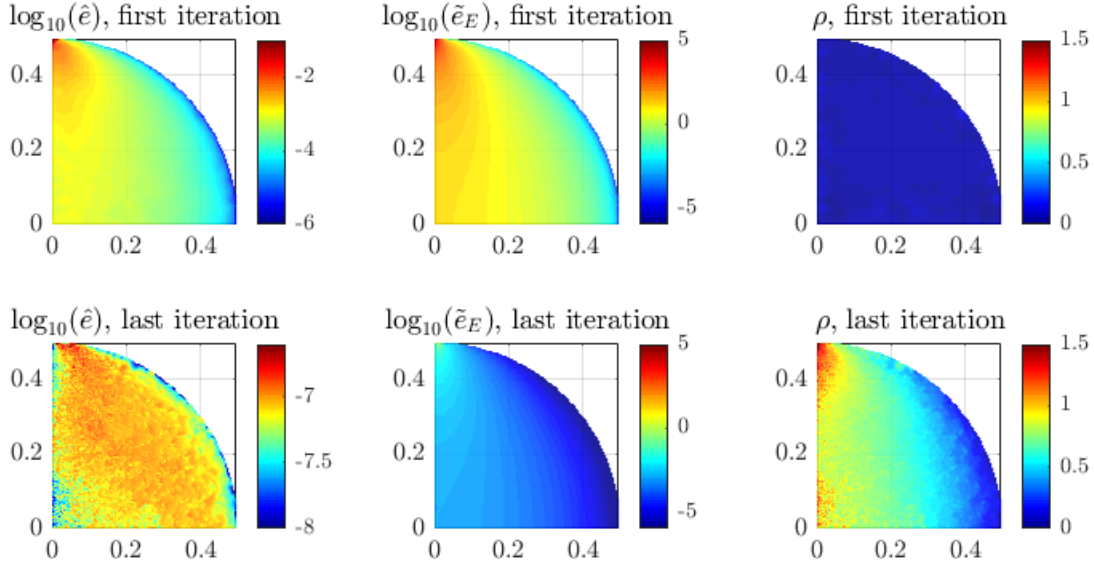


FIGURE 13 Error indicator value in the last iteration.

The measure of relative nodal density  $\rho$  at  $p_i$  is defined as

$$\rho_i = -\log_{10}(\delta r_i / \max_i(\delta r_i)), \quad (29)$$

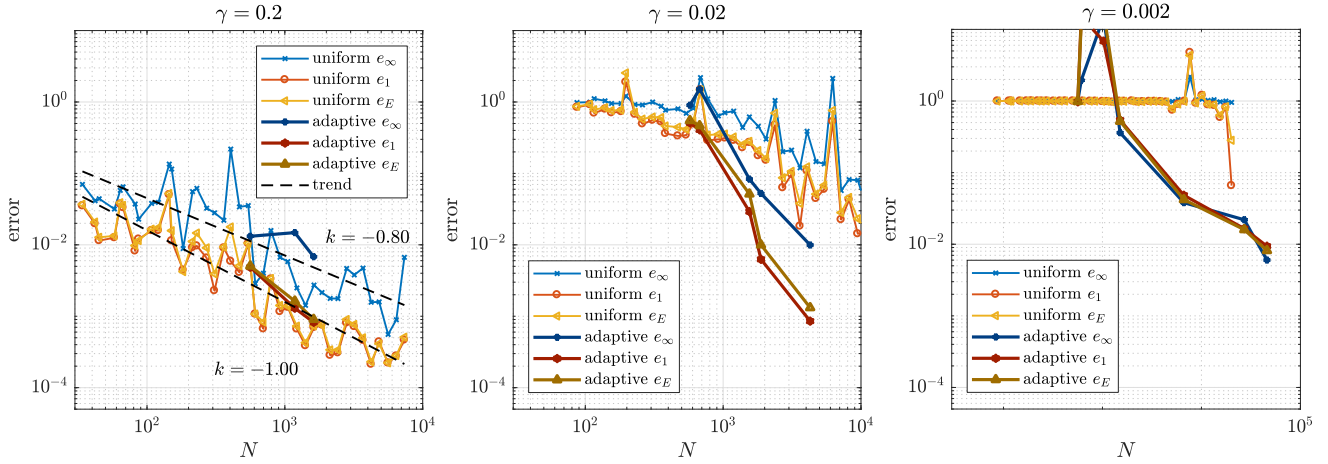
where  $\delta r_i$  is the distance to the closest neighbouring node. A value of e.g.  $\rho_i = 2$  means that nodal density around node  $p_i$  is  $10^2$  times greater than the lowest density in the domain. For the measure of error at node  $p_i$  the kernel of the energy error  $\mathcal{E}_E$  (27) was taken. Initially the nodal density is constant (i.e.  $\rho \approx 0$ ) and the error and error indicator approximately match (up to a factor). In the last iteration, the error indicator is more uniformly distributed over the whole domain and its value is lower in general, as expected. When comparing the error and nodal densities the same colour axis limits were used for the first and the last iteration to get a correct global impression. Over the course of the adaptive iteration the high error areas were repeatedly refined, gradually lowering the error to a much more uniform distribution. The final density distribution however transformed opposite to the error, from a uniform distribution to the final distribution that resembles the initial error distribution. This mutually reverse transition to and from uniformity between error and node density was observed in all three considered cases.

Finally, to justify testing the adaptivity on these cases at all, the problem (24) is solved with uniform node density, i.e. the density function  $\delta r$  is constant. Error of the numerical solution was analysed for  $\delta r$  ranging from 0.05 to 0.0033. The results for all three cases are shown in Fig. 14.

The value of parameter  $\gamma$  clearly influences the case difficulty and the need for adaptivity. In the easiest case uniformly distributed nodes perform well, and the observed order of convergence in  $e_\infty$  matches expectations provided by the theory for standard finite difference approximations. The middle case however exhibits worse convergence properties, and the method completely fails to even compute the most difficult case with satisfiable accuracy. Adaptivity is clearly needed in the last two cases and the tested procedure displays satisfactory error behaviour in all three cases.

### 4.3 | Analysis of adaptivity parameters

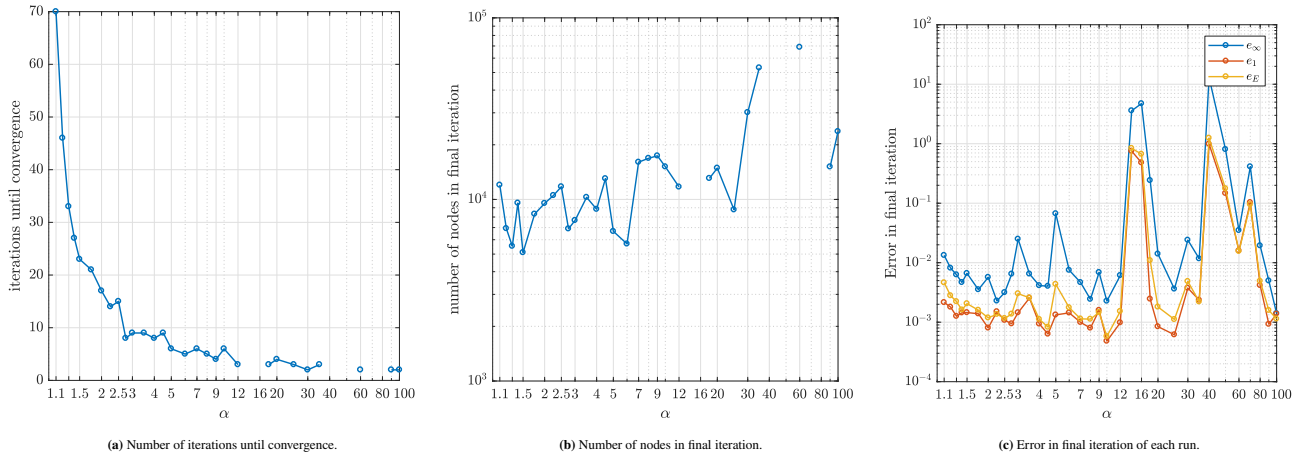
The two important parameters of the refine procedure are the error threshold  $\epsilon$  and refine aggressiveness  $\alpha$ . The error threshold parameter  $\epsilon$  is straightforward, and it generally holds that the lower the  $\epsilon$ , the lower the final error will be. Note that if  $\hat{\epsilon}$  is merely an error indicator and not an error estimator (i.e. it only gives an indication of areas with high errors and not upper bounds on errors), then the final error may be larger than  $\epsilon$ .



**FIGURE 14** Error of the numerical solution of the compressed disk problem with respect to number of nodes.

The parameter  $\alpha$  controls the aggressiveness of the refine procedure, as it bounds the density increase from above by a factor of at most  $\alpha$ . Setting e.g.  $\alpha = 2$  allows the internodal distance to at most halve and the number of nodes in a region to increase at most 4 times (in 2D). Intuitively, lower values of  $\alpha$  should result in slower and more controlled error decreases, while higher values lower the number of needed iterations but the error behaviour is more volatile.

To analyse the behaviour with respect to  $\alpha$  the case  $\gamma = \gamma_2$  was solved using the same parameters as in section 4.2, while  $\alpha$  ranged from 1.1 to 100. Fig. 15a shows the number of adaptive iterations until convergence, i.e. the number of while loop iterations on line 2 in algorithm 1. The missing data points indicate failure of convergence.



**FIGURE 15** Analyses of effect of aggressiveness parameter  $\alpha$ . Missing data points indicate a divergent solution.

As predicted, the number of iterations reduces steadily with larger  $\alpha$  up to around  $\alpha = 10$  when the behaviour becomes too unstable and failures of convergence occur often. In the stable region with  $\alpha \lesssim 10$  both the number of nodes in the final iteration (Fig. 15b) and the error in the final iteration (Fig. 15c) are mostly independent of  $\alpha$ , meaning that irrespective of how many iterations it took to get to a satisfiable solution, the solution has approximately the same error and uses approximately the same number of nodes. Therefore, to reduce the number of iterations necessary while being safely on the stable side,  $\alpha = 5$  was chosen.

The corresponding derefine parameters  $\beta$  and  $\eta$  behave similarly to  $\alpha$  and  $\epsilon$ . Too large values of  $\beta$  cause volatile behaviour, while too low values simply mean that some areas may be covered with too many nodes, which has little effect on convergence but unnecessarily increases computational costs. Usually the value of  $\beta = 1.5$  was chosen to derefine more conservatively. The

initial distribution  $\delta r^{(0)}$  was usually chosen for the coarseness bound  $\delta r^u$ . This means that derefine cannot reduce the density in the first few iterations, but can still correct possible increases in wrong areas later on.

The remaining free parameters are related to the density reconstruction (see sec. 3.3) using modified Sheppard's method. Varying the number of neighbours  $n$  from 3 to 15 had no significant effect on the solution procedure. Weighting over  $n = 7$  nodes was used in most runs.

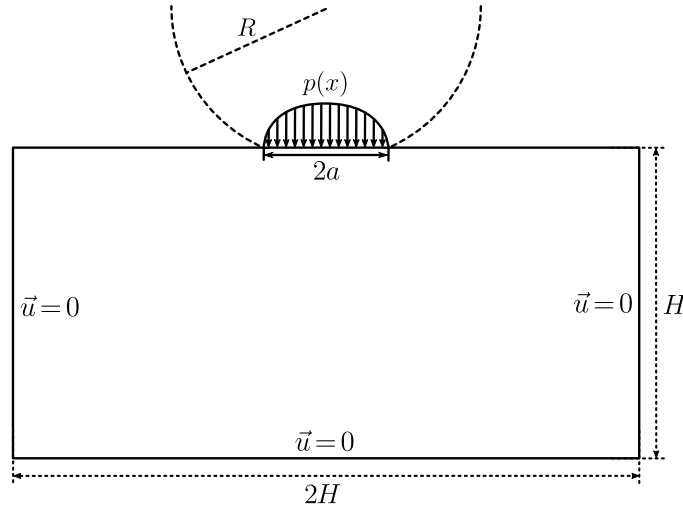
#### 4.4 | Hertzian contact

The next case considers Hertzian contact between an elastic cylinder and a half plane, as recently used by Slak and Kosec<sup>8</sup>. The analytical boundary condition on the top boundary is given by the known pressure distribution

$$p(x) = \begin{cases} p_0 \sqrt{1 - \frac{x^2}{a^2}}, & |x| \leq a \\ 0, & \text{otherwise} \end{cases}, \quad p_0 = \sqrt{\frac{PE^*}{\pi R}}, \quad a = 2\sqrt{\frac{PR}{\pi E^*}}, \quad (30)$$

where  $P$  is the pressure force,  $R$  is the radius of the cylinder and the combined Young's modulus  $E^*$  is given by  $\frac{1}{E^*} = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}$ , where  $E_1, \nu_1$  and  $E_2, \nu_2$  are the material properties of the cylinder and the half plane, respectively. The remaining boundary conditions are no-displacement conditions at infinity.

The problem is solved numerically by truncating the half plane to a rectangle  $[-H, 0] \times [-H, H]$  and applying the no displacement boundary conditions at all boundaries but the top, as illustrated in Fig. 16. For the choice  $R = 1$  m,  $P = 543$  N/m,  $E = 72.1$  GPa,  $\nu = 0.33$ , the contact width  $a$  is approximately 0.13 mm. A choice of  $H = 0.1$  m, which is approximately 770 times larger than the phenomenon of interest, is sufficiently large that the truncation error does not present a significant contribution<sup>8</sup>.



**FIGURE 16** Numerical domain and boundary conditions for the solution of Hertzian contact problem.

The large ratio between domain size and contact width presents a challenging case for adaptive refinement algorithms. The domain was initially filled with a constant discretisation step  $h$ , using the filling algorithm described in section 3.1. The initial step  $h$  needs to be small enough to capture the contact itself, while this same  $h$  might be unnecessarily small for the rest of the domain. This offers a chance to demonstrate adaptive derefinement as well.

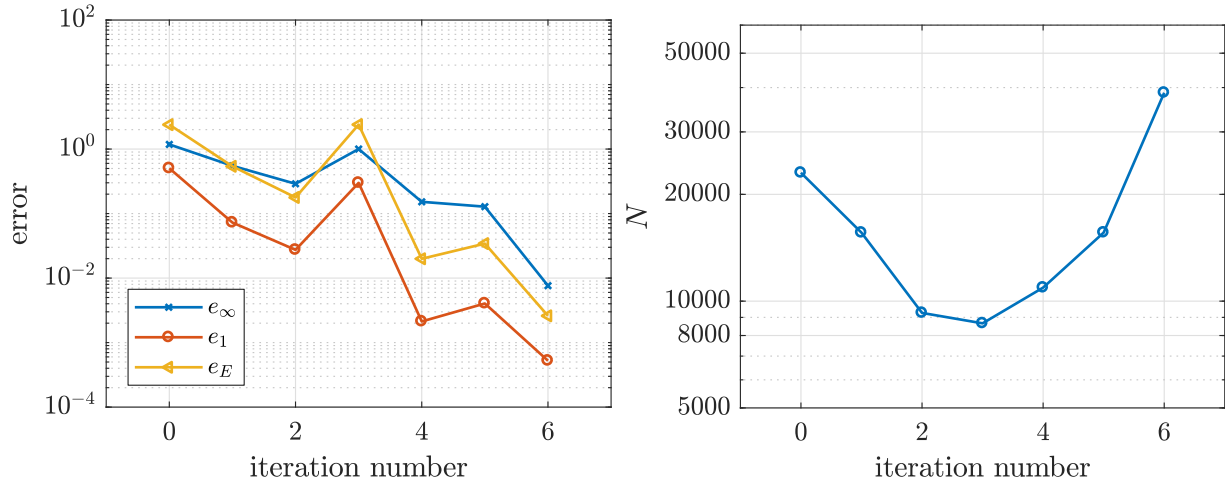
Additionally, instead of an analytical indicator given by (28), a simple ad-hoc error indicator was tested, to develop a functional indicator for cases where no analytical solution is available.

The construction of the indicator starts on the premise that the discretisation needs to be refined in areas with high gradients, i.e. the areas where the variability of the numerical solution is locally large. Similar indicators have been already reported for

adaptive solution of Poisson and Burgers's equation<sup>25,26</sup>. The error indicator is thus introduced as

$$\mathfrak{I}_i = \text{std}_{p_j \in N_i} \sigma_{xx}(p_j) + \text{std}_{p_j \in N_i} \sigma_{yy}(p_j) + \text{std}_{p_j \in N_i} \sigma_{xy}(p_j), \quad (31)$$

The adaptive procedure was run with  $\alpha = 5$ ,  $\varepsilon = 10^5$ ,  $\beta = 1.5$ ,  $\eta = 10^4$  and left to run for 6 iterations. The errors measured against a closed form solution<sup>8, eq. (54–56)</sup> and the number of nodes over the course of the iterative procedure are shown in Fig. 17. Additionally, data about node refinement for each iteration is shown in Table 2.



**FIGURE 17** Errors and node counts during adaptive iteration for the solution of the Hertzian problem.

iteration	total nodes	refined	no change	derefined	derefined, but hit bound $\delta r^u$
0	23 012	384	3 427	19 201	0
1	15 596	483	777	14 336	0
2	9 264	1 127	1 469	6 668	0
3	8 658	5 198	3 460	0	0
4	10 918	3 097	2 017	5 804	0
5	15 605	9 494	2 189	3 753	169
6	38 674	/	/	/	/

**TABLE 2** Number of refined and derefined nodes for each iteration.

In the beginning, the derefinement is substantial and the node count decreases, however, refinement is already present for nodes under contact. Later in the procedure, refinement becomes more pronounced and the total number of nodes increases. The computed top surface tractions over the course of the adaptive iteration are shown in Fig. 18.

It is interesting to compare these results to the manually refined solution obtained by Slak and Kosec<sup>8</sup>. The error in the final iteration of the adaptive procedure is in the same range as the errors obtained by manual refinement. Additionally, an interesting comparison of node densities under contact can be made and is shown in Fig. 19. The manual and adaptively obtained densities generally refine the same areas, however the adaptively obtained density is much smoother.

From the error plot in Fig. 17 and the density plot in Fig. 19 it can be concluded that the error indicator  $\mathfrak{I}$ , given by (31), works adequately in practice.

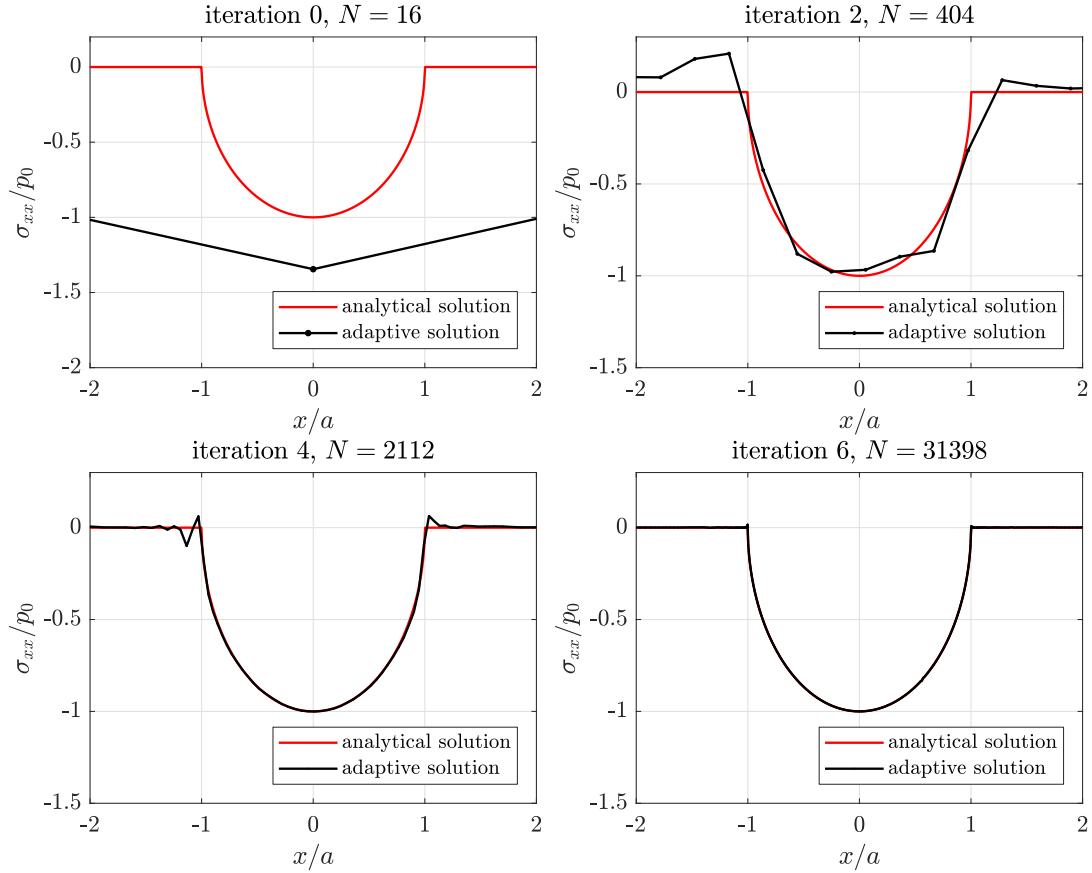


FIGURE 18 Top surface stress profiles during the iteration. Only even iterations are shown for brevity.

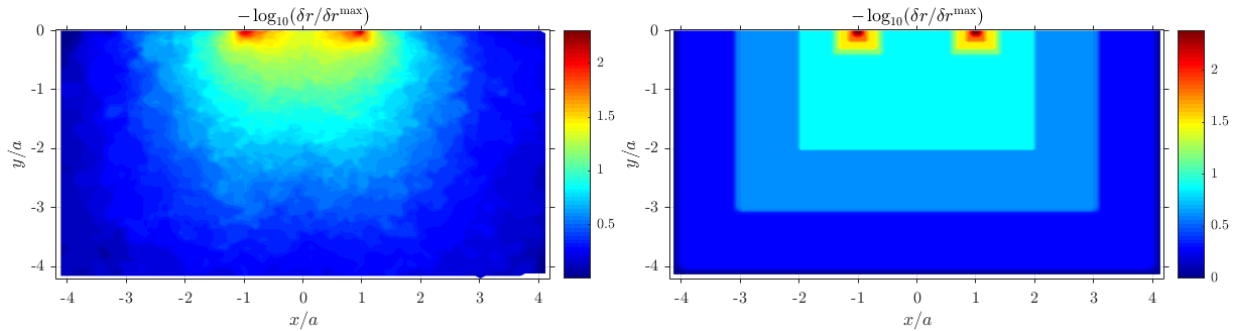


FIGURE 19 Comparing adaptive (left) and manual (right) nodal distributions densities for the Hertzian problem.

## 4.5 | Fretting fatigue contact

The usability of the described adaptive technique is further demonstrated on a case, originating from the study of fretting fatigue, described by Pereira et al.<sup>34</sup>, for which the authors believe no closed form solution is known. A small thin rectangular specimen of width  $W$ , length  $L$  and thickness  $t$  is stretched in one axis with axial traction  $\sigma_{ax}$  and compressed in another with force  $F$  by two oscillating cylindrical pads of radius  $R$ , inducing tangential force  $Q$ , as shown in Fig. 20a. The tractions induced by the pads are predicted using an extension of Hertzian contact theory, which splits the contact area into the stick and slip zones, depending on the coefficient of friction  $\mu$  and combined Young's modulus  $E^*$ , given by  $\frac{1}{E^*} = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}$ , where  $E_i$  and  $\nu_i$  represent the

Young's moduli and Poisson's ratios of the specimen and the pad, respectively. The theory predicts the contact half-width

$$a = 2\sqrt{\frac{FR}{t\pi E^*}}, \quad (32)$$

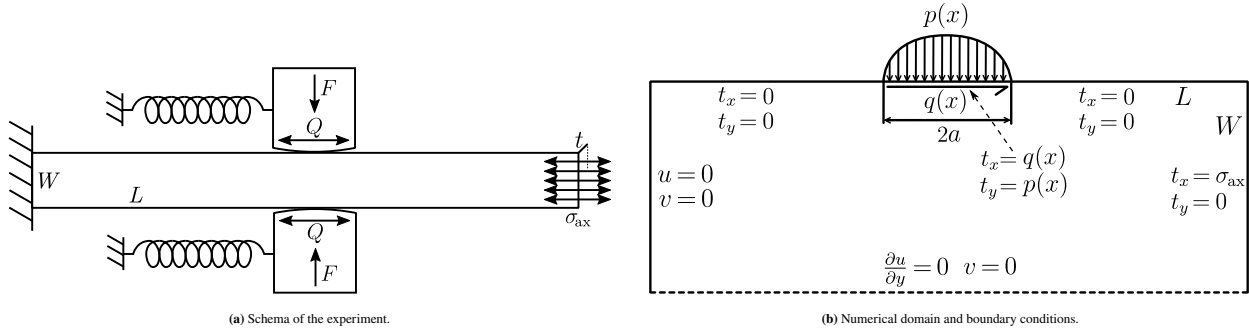
normal traction

$$p(x) = \begin{cases} p_0 \sqrt{1 - \frac{x^2}{a^2}}, & |x| < a \\ 0, & \text{else} \end{cases}, \quad p_0 = \sqrt{\frac{FE^*}{t\pi R}}, \quad (33)$$

and tangential traction

$$q(x) = \begin{cases} -\mu p_0 \left( \sqrt{1 - \frac{x^2}{a^2}} - \frac{c}{a} \sqrt{1 - \frac{(x-e)^2}{c^2}} \right), & |x - e| < c \\ -\mu p_0 \sqrt{1 - \frac{x^2}{a^2}}, & c \leq |x - e| \text{ and } |x| \leq a \\ 0, & \text{else} \end{cases} \quad (34)$$

where  $c = a\sqrt{1 - \frac{Q}{\mu F}}$  is the half-width of the slip zone, and  $e = \text{sgn}(Q)\frac{a\sigma_{ax}}{4\mu p_0}$  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, both of which will be satisfied in our case.

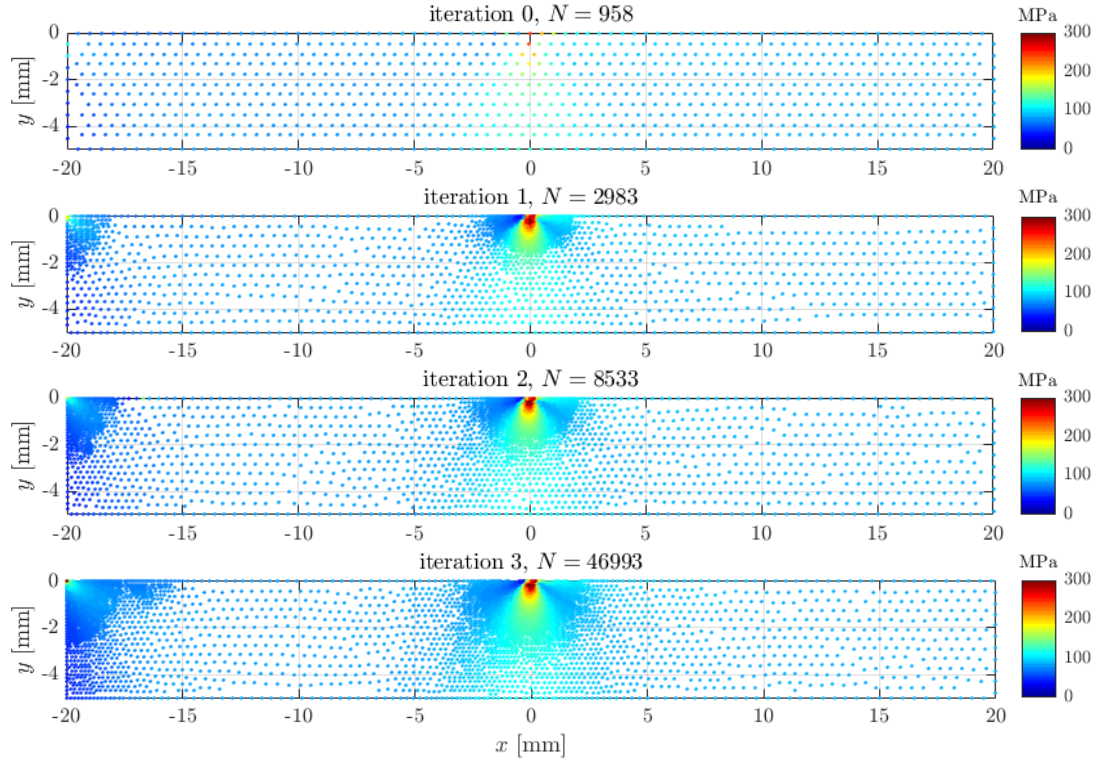


**FIGURE 20** Description of the considered fretting fatigue case. Ratios in the drawings are not to scale.

Plane strain is used to reduce the problem to two dimensions and symmetry along the horizontal axis is used to reduce the problem size. The region  $\Omega = [-L/2, L/2] \times [-W/2, 0]$  is taken as the problem domain and the boundary conditions are illustrated in Fig. 20b.

Values of  $E_1 = E_2 = 72.1$  GPa,  $\nu_1 = \nu_2 = 0.33$  were taken for the material parameters, coinciding with aluminium 2420-T3. Dimensions of the specimen were  $L = 40$  mm,  $W = 10$  mm and  $t = 4$  mm. Values  $F = 543$  N,  $Q = 155$  N,  $\sigma_{ax} = 100$  MPa,  $R = 10$  mm and  $\mu = 0.3$  were chosen for the model parameters. This means that the half-contact width  $a$  equals 0.2067 mm, which is approximately 200 times smaller than the domain width  $W$ , once again making it difficult to solve without adaptivity. The same ad-hoc error indicator as in section 4.4 was used to adaptively refine the solution. The initial distribution was generated using constant spacing  $\delta r^{(0)} = 0.5$  mm, putting exactly one node on the whole contact surface. 4 iterations of the adaptive solution procedure were run using  $\epsilon = 0.1$  MPa as the error threshold and  $\alpha = 10$ , resulting in a domain populated with 49 993 nodes. The sets of nodes  $\mathcal{P}$  and the corresponding numerical solutions obtained during the adaptive iteration are shown in Fig. 21. The results are compared to a solution obtained with the Finite Element Method (FEM) on a much denser mesh with more than 100 000 DOFs, using commercial ABAQUS<sup>®</sup> software for finite element analysis, where manual refinement has been used<sup>34</sup>, as well as to another FEM solution, obtained using the freely available FreeFem++<sup>46</sup> and its built-in adaptive refinement techniques.

The surface traction  $\sigma_{xx}$  is of particular interest, as it is often used to determine the location of crack initiation<sup>34</sup>. Its graph over the course of the adaptive iteration is shown in Fig. 22. In the initial solution, only one node is placed under the contact surface; however, after the final iteration, there are approximately 700 nodes under the contact surface and the results agree well with FEM solutions.



**FIGURE 21** Nodes of four subsequent adaptive iterations during the solution of the fretting fatigue problem, coloured by von Mises stress.

Additionally, error of the adaptive solution is shown in Fig. 23. The error was measured using  $e_1$  (see Eq. 26) on the contact surface, specifically on the interval  $[-2a, 2a]$ , where  $a$  is the contact half width. As no analytical solution is known, the error was measured using reference FEM solutions.

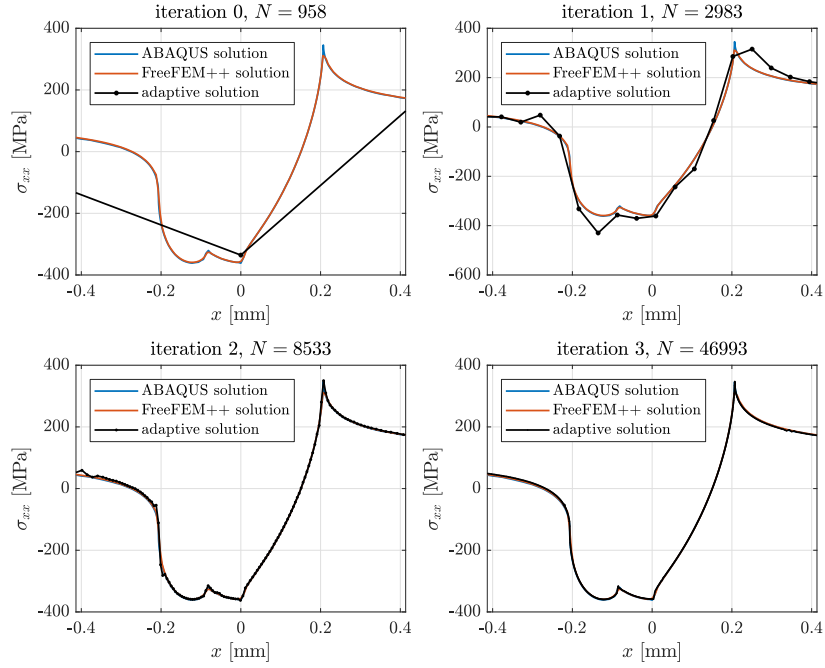
The two reference solutions were computed to also assess the variability between solutions obtained with established methods and implementations, and our adaptive implementation. The error shown in Fig. 23 decreases with each iteration, similarly to other already presented cases, and achieves satisfactory accuracy, nearing the reference solution variability.

Execution time was measured for both adaptive FEM and the proposed method. All time measurements were done on a laptop computer with Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz processor and 16 GB DDR4 RAM. Code was compiled using g++ (GCC) 8.1.1 for Linux with `-std=c++11 -O3 -DNDEBUG` flags and no parallelisation. The FEM solution was obtained using FreeFem++ version 3.61 compiled with full optimisations on the same machine. The execution time varies for both procedures with slight alterations of adaptivity parameters. The observed execution times for solutions obtained with FreeFem++ ranged from 6 s to 35 s and for the proposed adaptive procedure they ranged from 8 s to 100 s. For the specific solutions used in Fig. 23 the execution times were 12.9 s for FreeFem++ and 38.6 s for the proposed procedure. The reported times are total wall clock times which include generation of the initial discretization, repeated refinement, matrix assembly, solution of the linear system and data export. The majority of time is spent on solving the linear system, and in the meshless approach, on computing the shape functions.

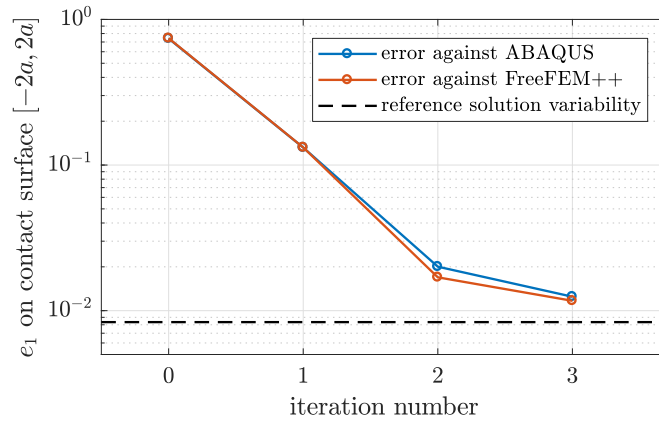
The execution times are reported only for reference and are not really comparable, since adaptive procedures used are different and our implementation served to demonstrate a proof-of-concept rather than being heavily focused on performance.

## 4.6 | Extension to 3-D problem

In addition to the examples given in the previous sections, we present a fully automatic adaptive solution of a 3-D problem. The Boussinesq's problem of the concentrated normal traction acting on an isotropic half-space is considered as described by



**FIGURE 22** Surface traction  $\sigma_{xx}$  under contact in four subsequent adaptive iterations during the solution of the fretting fatigue problem.



**FIGURE 23** Error of surface traction using two reference solutions.

Slaughter<sup>47</sup>. The closed form solution is given in cylindrical coordinates  $r$ ,  $\theta$  and  $z$  as

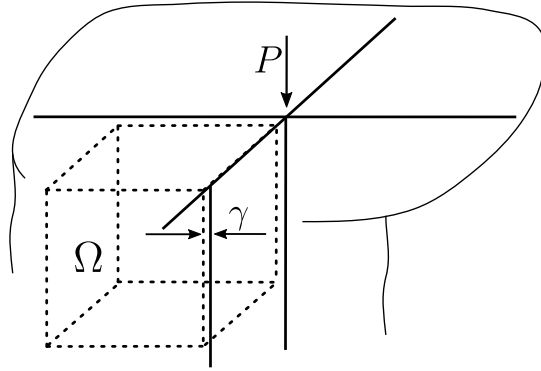
$$\begin{aligned}
 u_r &= \frac{Pr}{4\pi\mu} \left( \frac{z}{R^3} - \frac{1-2\nu}{R(z+R)} \right), & u_\theta &= 0, & u_z &= \frac{P}{4\pi\mu} \left( \frac{2(1-\nu)}{R} + \frac{z^2}{R^3} \right), \\
 \sigma_{rr} &= \frac{P}{2\pi} \left( \frac{1-2\nu}{R(z+R)} - \frac{3r^2z}{R^5} \right), & \sigma_{\theta\theta} &= \frac{P(1-2\nu)}{2\pi} \left( \frac{z}{R^3} - \frac{1}{R(z+R)} \right), \\
 \sigma_{zz} &= -\frac{3Pz^3}{2\pi R^5}, & \sigma_{rz} &= -\frac{3Prz^2}{2\pi R^5}, & \sigma_{r\theta} &= 0, & \sigma_{\theta z} &= 0,
 \end{aligned} \tag{35}$$

where  $P$  is the magnitude of the point force,  $\nu$  is the Poisson's ratio,  $\mu$  is the second Lamé parameter and  $R = \sqrt{r^2 + z^2}$  is the distance of given point from origin. This solution has a singularity at the origin and similarly to previous sections, we will consider a portion of a problem near the singularity. Numerically, we solve (19) with essential boundary conditions given by (35)

on a domain

$$\Omega = [-1, -\gamma] \times [-1, -\gamma] \times [-1, -\gamma] \quad (36)$$

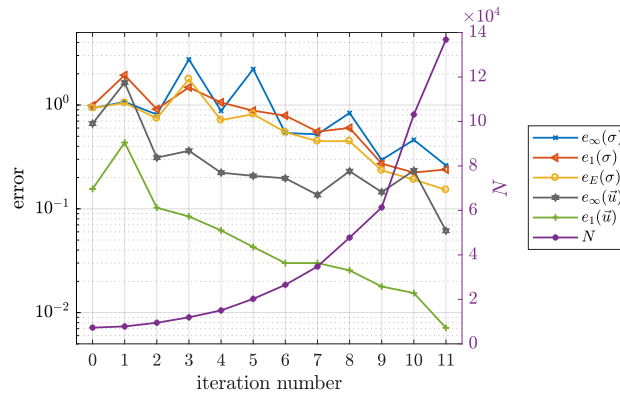
for  $\gamma = 0.01$ . The domain  $\Omega$  in an elastic half-space is shown in Fig 24. The similar stress profiles along the body diagonal of  $\Omega$  are similar to the ones in Fig. 10 and shown in Fig. 26.



**FIGURE 24** Boussinesq's problem and the computational domain  $\Omega$ .

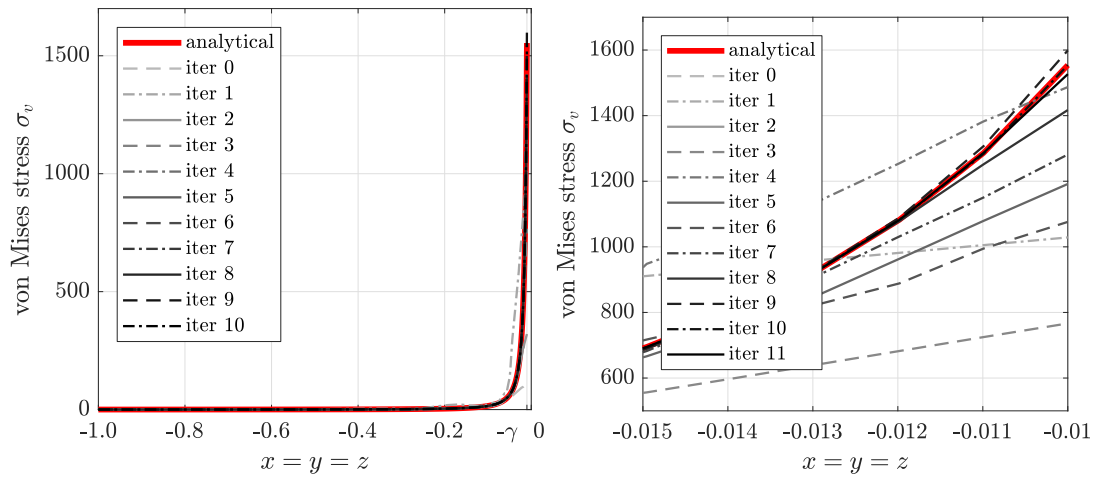
Although the solution is given in cylindrical coordinates, the discretisation and enforcement of boundary conditions are done in Cartesian coordinates, with the same implementation of the method as used in 2-D is used in 3-D as well. The domain was initially filled with constant density of  $\delta r^{(0)} = 0.05$ , amounting to  $N = 7289$  nodes. For the given discretisation, RBF-FD with 15 neighbouring nodes and 15 Gaussian RBFs was used. The shape parameter  $\sigma_b$  was varied proportionally to the internodal distance with base value  $\sigma_b = 100$ . For adaptive iteration the straightforward generalisation of ad-hoc error indicator (31) was used with  $\varepsilon = 5$ ,  $\eta = 0$ ,  $\alpha = 3$  and  $\beta = 1$ . The reconstruction and evaluation of the new density function  $\tilde{\delta r}$  was done using 15 closest neighbours. Values of  $P = -1$ ,  $E = 1$  and  $\nu = 0.33$  were taken for physical parameters of the problem.

The error was measured using direct generalisations of formulas (25–27) into 3-D. Since the closed form solution for displacement  $\mathbf{u}$  is known in this case as well, we will additionally show approximations of  $L^1$  and  $L^\infty$  errors for displacements as well, which are computed analogously to (25) and (26) and will be denoted as  $e_1(\mathbf{u})$  and  $e_\infty(\mathbf{u})$ , respectively. Fig. 25 shows the computed errors over the course of the adaptive iteration. Additionally, von Mises stress profiles along the body diagonal from  $[-1, -1, -1]$  to  $[-\gamma, -\gamma, -\gamma]$  are shown in Fig. 26.



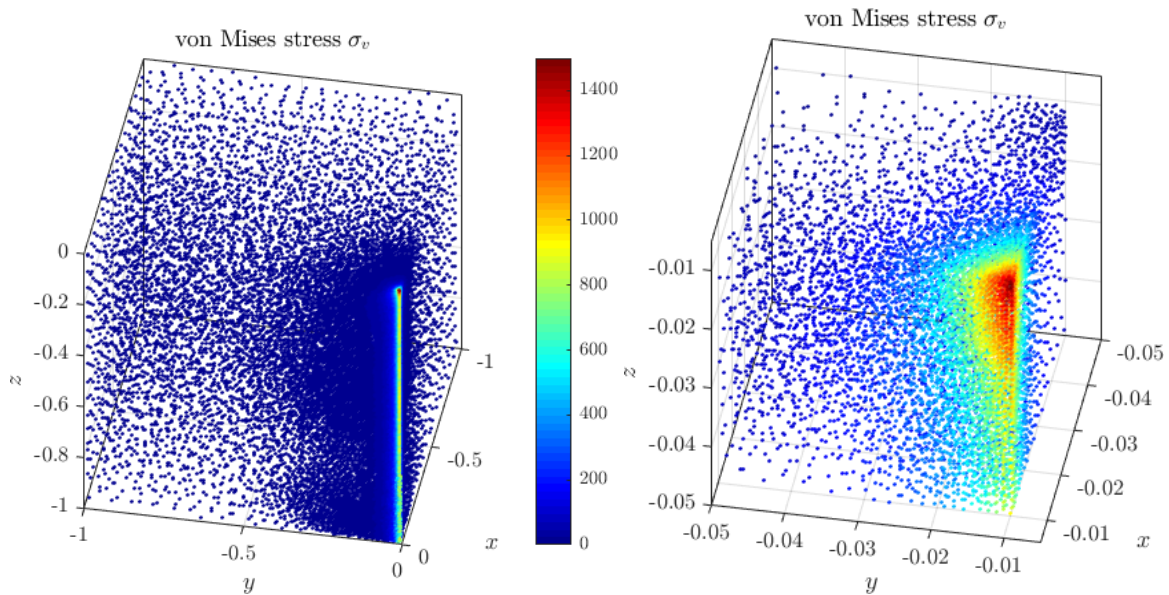
**FIGURE 25** Error with respect to the number of nodes in the adaptive iteration of the solution of the Boussinesq's problem. The right axis shows the number of computational nodes.

The behaviour of the method is similar to the two-dimensional compressed disk case. Initially, the error rises until the area around the contact is covered densely enough. The displacement errors are smaller than stress errors, as is customary in solid mechanics. Right panel in Fig. 26 shows more precisely how the peak stress is estimated.



**FIGURE 26** Stress profiles along the body diagonal from  $[-1, -1, -1]$  to  $[-\gamma, -\gamma, -\gamma]$  in adaptive iteration when solving the Boussinesq's problem.

Fig. 27 shows the solution in the final iteration of the adaptive procedure. The corner  $[-\gamma, -\gamma, -\gamma]$  is shown more closely in the right figure.



**FIGURE 27** The obtained solution in the final iteration with an enlarged portion around the contact area. Both solutions are plotted only in the computation nodes to also show the final nodal distribution. Nodes are coloured proportional to computed values of von Mises stress.

The execution time of the entire adaptive procedure to obtain the solution shown in Fig. 27 was 125 s with 1 s spent on the initial iteration with 7289 DOFs and 48 s spent on the last iteration with 138 632 DOFs. In the last iteration, the internodal distances between the densest parts are about 176 times smaller than in the coarsest parts.

## 5 | CONCLUSIONS

Adaptive refinement is a crucial part of a numerical solution procedure for any problem that exhibits extensive differences in intensity within the problem domain. In this paper, a modular adaptive refinement algorithm for solving elliptic boundary value problems was proposed. The refinement procedure consists of a Poisson Disc Sampling based nodal positioning algorithm, an ad-hoc error indicator, a RBF-FD meshless solution of the governing problem, the refinement strategy, and the reconstruction of the density function. The free parameters of the algorithm were determined by numerical experiments, and reasonable defaults along with acceptable intervals were proposed. Finally, it was clearly demonstrated that the proposed algorithm provides a stable solution for all considered cases without any hyper-tuning. This includes the solution of the fretting fatigue case, where complications arise due to the presence of the stick-slip zone at the contact interface, which causes extreme peaks in surface stress making it very difficult to solve numerically in a reasonable time without refinement. The proposed approach successfully adaptively refined initially uniform discretisation, captured both peaks, and numerically solved the problem in only four iterations, with excellent agreement towards the reference solution provided by commercial software using manual refinement.

All elements of the algorithm have no intrinsic dependency on the dimensionality of the space and generalisation to higher dimensions is possible. In the final example, the generality of the proposed method is demonstrated in the solution of 3-D Boussinesq's problem of the concentrated normal traction acting on an isotropic half-space.

Although all discussions in this paper were limited to the consideration of the Navier-Cauchy partial differential equation and contact problems, the presented methodology can also be applied in other areas where numerical discretisation is required to solve the governing problem. The only notable required change would be in the second step, i.e. the solution of the problem, while the other steps would remain the same.

The examples in this paper were computed using the in-house open source Medusa library<sup>35</sup> and the algorithms presented in this paper. All research data is freely available on a git repository<sup>36</sup>.

Future work will be focused on implementation of more error indicators, more efficient implementation, and extension to different problems.

## 6 | ACKNOWLEDGEMENTS

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

## References

1. Kansa EJ. Multiquadrics – A scattered data approximation scheme with applications to computational fluid-dynamics – I surface approximations and partial derivative estimates. *Computers & Mathematics with Applications* 1990; 19(8–9): 127–145. doi: 10.1016/0898-1221(90)90270-T
2. Franke C, Schaback R. Solving partial differential equations by collocation using radial basis functions. *Applied Mathematics and Computation* 1998; 93(1): 73–82. doi: 10.1016/s0096-3003(97)10104-7
3. Šarler B. From global to local radial basis function collocation method for transport phenomena. In: Berlin: Springer Netherlands. 2007 (pp. 257–282)
4. Prax C, Sadat H, Salagnac P. Diffuse approximation method for solving natural convection in porous media. *Transport in Porous Media* 1996; 22: 215–223. doi: 10.1007/bf01143516

5. Tolstykh AI, Shirobokov DA. On using radial basis functions in a “finite difference mode” with applications to elasticity problems. *Computational Mechanics* 2003; 33(1): 68–79. doi: 10.1007/s00466-003-0501-9
6. Fornberg B, Flyer N. Solving PDEs with radial basis functions. *Acta Numerica* 2015; 24: 215–258. doi: 10.1017/S0962492914000130
7. Bayona V, Flyer N, Fornberg B, Barnett GA. On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs. *Journal of Computational Physics* 2017; 332: 257–273. doi: 10.1016/j.jcp.2016.12.008
8. Slak J, Kosec G. Refined Meshless Local Strong Form solution of Cauchy–Navier equation on an irregular domain. *Engineering Analysis with Boundary Elements* 2018. doi: 10.1016/j.enganabound.2018.01.001
9. Zienkiewicz OC, Taylor RL. *The Finite Element Method: Solid Mechanics*. Butterworth-Heinemann . 2000.
10. Chen Y, Lee JD, Eskandarian A. *Meshless methods in solid mechanics*. New York, NY: Springer . 2006.
11. Mavrič B, Šarler B. Local radial basis function collocation method for linear thermoelasticity in two dimensions. *International Journal of Numerical Methods for Heat and Fluid Flow* 2015; 25: 1488–1510. doi: 10.1108/hff-11-2014-0359
12. Liu GR, Gu YT. *An Introduction to Meshfree Methods and Their Programming*. Dordrecht: Springer . 2005.
13. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; 37(2): 229–256. doi: 10.1002/nme.1620370205
14. Atluri SN, Zhu T. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; 22(2): 117–127. doi: 10.1007/s004660050346
15. Oñate E, Perazzo F, Miquel J. A finite point method for elasticity problems. *Computers & Structures* 2001; 79(22–25): 2151–2163. doi: 10.1016/s0045-7949(01)00067-0
16. Kosec G. A local numerical solution of a fluid-flow problem on an irregular domain. *Advances in Engineering Software* 2018; 120: 36–44. doi: 10.1016/j.advengsoft.2016.05.010
17. Fornberg B, Flyer N. Fast generation of 2-D node distributions for mesh-free PDE discretizations. *Computers & Mathematics with Applications* 2015; 69(7): 531–544. doi: 10.1016/j.camwa.2015.01.009
18. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering* 1987; 24(2): 337–357. doi: 10.1002/nme.1620240206
19. Thimnejad M, Fallah N, Khoei AR. Adaptive refinement in the meshless finite volume method for elasticity problems. *Computers & Mathematics with Applications* 2015; 69(12): 1420–1443. doi: 10.1016/j.camwa.2015.03.023
20. Tang Q, Zhang GY, Liu GR, Zhong ZH, He ZC. A three-dimensional adaptive analysis using the meshfree node-based smoothed point interpolation method (NS-PIM). *Engineering Analysis with Boundary Elements* 2011; 35(10): 1123–1135. doi: 10.1016/j.enganabound.2010.05.019
21. Angulo A, Pozo LP, Perazzo F. A posteriori error estimator and an adaptive technique in meshless finite points method. *Engineering Analysis with Boundary Elements* 2009; 33(11): 1322–1338. doi: 10.1016/j.enganabound.2009.06.004
22. Oanh DT, Davydov O, Phu HX. Adaptive RBF-FD Method for Elliptic Problems with Point Singularities in 2D. *Applied Mathematics and Computation* 2017; 313: 474–497. doi: 10.1016/j.amc.2017.06.006
23. Sang-Hoon P, Kie-Chan K, Sung-Kie Y. A posterior error estimates and an adaptive scheme of least-squares meshfree method. *International Journal for Numerical Methods in Engineering* 2003; 58(8): 1213–1250. doi: 10.1002/nme.817
24. Afshar MH, Naisipour M, Amani J. Node moving adaptive refinement strategy for planar elasticity problems using discrete least squares meshless method. *Finite Elements in Analysis and Design* 2011; 47(12): 1315–1325. doi: 10.1016/j.finel.2011.07.003

25. Davydov O, Oanh DT. Adaptive meshless centres and RBF stencils for Poisson equation. *Journal of Computational Physics* 2011; 230(2): 287–304. doi: 10.1016/j.jcp.2010.09.005
26. Kosec G, Šarler B. H-adaptive local radial basis function collocation meshless method. *CMC: Computers, Materials & Continua* 2011; 26(3): 227–253. doi: 10.3970/cmc.2011.026.227
27. Stevens D, Power H, Lees M, Morvan H. The use of PDE centres in the local RBF Hermitian method for 3D convective-diffusion problems. *Journal of Computational Physics* 2009; 228: 4606–4624. doi: 10.1016/j.jcp.2009.03.025
28. Libre AL, Emdadi A, Kansa EJ, Shekarchi M, Rahimian M. A fast adaptive wavelet scheme in RBF collocation for nearly singular potential PDEs. *CMES: Computer Modeling in Engineering & Sciences* 2008; 38(3): 263–284. doi: 10.3970/cmes.2008.038.263
29. Liu GR, Kee BBT, Chun L. A stabilized least-squares radial point collocation method (LS-RPCM) for adaptive analysis. *Computer Methods in Applied Mechanics and Engineering* 2006; 195(37–40): 4843–4861. doi: 10.1016/j.cma.2005.11.015
30. Afshar MH, Amani J, Naisipour M. A node enrichment adaptive refinement in Discrete Least Squares Meshless method for solution of elasticity problems. *Engineering Analysis with Boundary Elements* 2012; 36(3): 385–393. doi: 10.1016/j.enganabound.2011.08.012
31. Rabczuk T, Belytschko T. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering* 2007; 196: 2777–2799. doi: 10.1016/j.cma.2006.06.020
32. Driscoll TA, Heryudono ARH. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Computers & Mathematics with Applications* 2007; 53(6): 927–939. doi: 10.1016/j.camwa.2006.06.005
33. Fornberg B, Zuev J. The Runge phenomenon and spatially variable shape parameters in RBF interpolation. *Computers & Mathematics with Applications* 2007; 54(3): 379–398. doi: 10.1016/j.camwa.2007.01.028
34. Pereira K, Bordas S, Tomar S, et al. On the Convergence of Stresses in Fretting Fatigue. *Materials* 2016; 9(8): 639. doi: 10.3390/ma9080639
35. Medusa: coordinate free implementation of meshless methods. <http://e6.ijs.si/medusa/>; Accessed on 19<sup>th</sup> February 2019.
36. Slak J, Kosec G. Paper repository with algorithms and research data. <https://gitlab.com/e62Lab/2018AdaptivePaper>; Accessed on 19<sup>th</sup> February 2019.
37. Schaback R, Wendland H. Characterization and construction of radial basis functions. In: Dyn N, Leviatan D, Levin D, Pinkus A., eds. *Multivariate Approximation and Applications* Cambridge University Press. 2001 (pp. 1–24)
38. Buhmann MD. *Radial basis functions: theory and implementations*. 12 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge university press . 2003
39. Flyer N, Lehto E. Rotational transport on a sphere: Local node refinement with radial basis functions. *Journal of Computational Physics* 2010; 229(6): 1954–1969. doi: 10.1016/j.jcp.2009.11.016
40. Flyer N, Fornberg B, Bayona V, Barnett GA. On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy. *Journal of Computational Physics* 2016; 321: 21–38. doi: 10.1016/j.jcp.2016.05.026
41. Wei LY. Parallel Poisson disk sampling. *ACM Transactions on Graphics* 2008; 27(3): 20. doi: 10.1145/1360612.1360619
42. Slak J, Kosec G. On generation of node distributions for meshless PDE discretizations. *arXiv:1812.03160 [math.NA]* 2018. Preprint, available at <https://arxiv.org/abs/1812.03160>.
43. Franke R. Scattered data interpolation: tests of some methods. *Mathematics of Computation* 1982; 38(157): 181–200. doi: 10.1090/s0025-5718-1982-0637296-4
44. Wendland H. *Scattered data approximation*. 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge university press . 2004

- 
45. Sadd MH. *Elasticity: theory, applications, and numerics*. Elsevier/AP. 2nd ed. 2009.
  46. Hecht F. New development in FreeFem++. *Journal of Numerical Mathematics* 2012; 20(3-4): 251–265. doi: 10.1515/jnum-2012-0013
  47. Slaughter WS. *The Linearized Theory of Elasticity*. Birkhäuser Boston . 2002. Pages 351–352.

