

Aeroacoustic source term computation based on radial basis functions

Stefan Schoder^{ID} | Klaus Roppert^{ID} | Michael Weitz | Clemens Junger |
Manfred Kaltenbacher

Institute of Mechanics and Mechatronics,
TU Wien, Vienna, Austria

Correspondence

Stefan Schoder, Institute of Mechanics and Mechatronics, TU Wien, 1060 Vienna, Austria.
Email: stefan.schoder@tuwien.ac.at

Funding information

Austrian Science Fund, I 3702; Deutsche Forschungsgemeinschaft, 391215328

SUMMARY

In low Mach number aeroacoustics, the known disparity of length scales makes it possible to apply well-suited simulation models using different meshes for flow and acoustics. The workflow of these hybrid methodologies include performing an unsteady flow simulation, computing the acoustic sources, and simulating the acoustic field. Therefore, hybrid methods seek for robust and flexible procedures, providing a conservative mesh to mesh interpolation of the sources while ensuring high computational efficiency. We propose a highly specialized radial basis function interpolation for the challenges during hybrid simulations. First, the computationally efficient local radial basis function interpolation in conjunction with a connectivity-based neighbor search technique is presented. Second, we discuss the computation of spatial derivatives based on radial basis functions. These derivatives are computed in a local-global approach, using a Gaussian kernel on local point stencils. Third, radial basis function interpolation and derivatives are used to compute complex aeroacoustic source terms. These ingredients are necessary to provide flexible source term calculations that robustly connect flow and acoustics. Finally, the capabilities of the presented approach are shown in a numerical experiment with a co-rotating vortex pair.

KEY WORDS

compactly supported functions, computational aeroacoustics, multivariate interpolation, radial basis functions

1 | INTRODUCTION

The demand for adapted interpolation techniques started with the need for topology interpolation. First attempts were made in terms of ordinary series representation. Trigonometric and polynomial series arise with an increasing computational effort and may not even represent typical surface elevations, such as hill tops and valleys. Regarding the needs, the application of radial basis functions began in the late 1960s in the field of geophysics. The appropriate treatment of scattered data is one of the main challenges when measuring air and sea temperature, wind speeds, or simply the topology of the earth. Having this scattered property of the data in mind, Hardy developed an approximation and interpolation method for surface fitting problems.¹ Since then, the multiquadric approximation of scattered data has been used in

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2019 The Authors. *International Journal for Numerical Methods in Engineering* published by John Wiley & Sons, Ltd.

geodesy, mapping, signal processing, digital terrain modeling, and hydrology.² In the field of aeroelasticity, fluid-structure interaction algorithms based on mesh-less radial basis function (RBF) interpolation are proposed as a computational efficient coupling strategy.³⁻⁵ The coupling of nonconforming grids with focus on computational performance is of great importance in aeroelasticity.⁶

Today, efficient interpolation techniques are especially important when dealing with computational aeroacoustics. Since the beginning of CAA, hybrid methods have been established as the most practical for fast and accurate aeroacoustic computations at low Mach numbers. The workflow of hybrid aeroacoustics involves three steps (see Figure 1): (i) perform unsteady flow computations on a restricted subdomain; (ii) compute the acoustic sources; (iii) simulate the acoustic field. For low Mach number applications, a key challenge in computational aeroacoustics (CAA) is the huge disparity of scales between flow structures and audible acoustic wavelengths. The scaling between the acoustic wavelength λ and the characteristic length l of a vortex is given by

$$\lambda \sim \frac{l}{\text{Ma}},$$

where Ma is the Mach number. Utilizing this disparity of scales, aeroacoustic analogies or perturbation techniques separate the flow computation from the acoustic computation (eg, References 7,8). Therefore, the sizes of the two computational grids are in general quite different.

For both physical fields, the individual optimal computational grid achieves the highest accuracy and the two grids differ according to the modeling criteria. On the one hand, the flow grid resolves boundary layers and is mostly coarsened toward outflow boundaries to dissipate vortices. On the other hand, the acoustic grid transports waves and therefore needs a uniform grid size all over the computational domain. In order to couple the different meshes, an interpolation scheme is necessary that satisfies the fundamental requirement of hybrid CAA: An accurate data transfer from the flow to the acoustic grid to minimize interpolation errors while conserving the energy. To cope with this task, different interpolation strategies can be applied, starting from low complexity nearest neighbor interpolation to complex source-term computation models with volume intersections between flow and acoustic grid. The simple nearest neighbor interpolation fails to compute the acoustic sources accurately (eg, References 9,10). In Reference 9, the aeroacoustic source term is computed by summing the contributions of all flow cells belonging to finite elements that surround an acoustic finite element node. It is assumed that the flow quantities used for the acoustic source-term computation are constant over each flow cell. This approach has also been used in Reference 11 for three-dimensional problems, where a grid dependency has been observed resulting in a too low sound pressure level over the whole frequency spectrum. A fully conservative approach has been used in Reference 12, where the acoustic sources within the finite element formulation are first computed on the fine flow grid. These so-called nodal loads are then interpolated by a conservative scheme to the acoustic grid. This approach is accurate in cases where the flow grid is much finer than the acoustic grid but fails in cases where the flow grid is coarser than the acoustic grid. As a solution to this problem, we derived a cut cell approach and successfully applied it to the aeroacoustic computation of an axial fan.¹³ Similar investigations have been performed in Reference 10, where for both the flow and the acoustic field a finite volume scheme has been used. However, most of these methods fail if the computational fluid dynamics (CFD) simulation only provides primary variables, such as the pressure p or the velocity \mathbf{u} , since the aeroacoustic sources are often nontrivial combinations of these primary variables (see Figure 1).

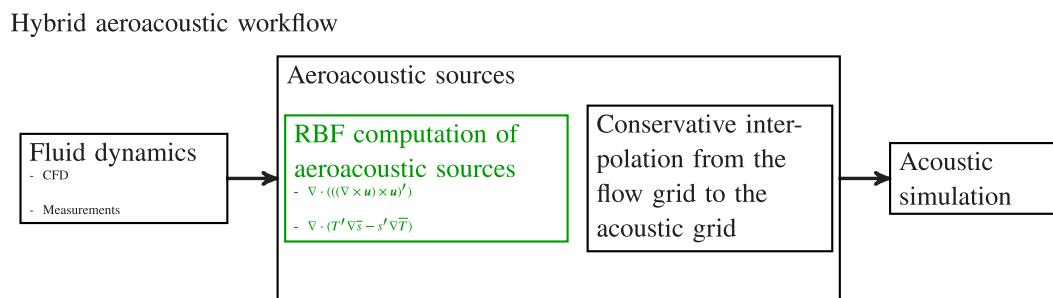


FIGURE 1 The hybrid aeroacoustic workflow consists of three main computational parts, where the presented radial basis function source-term computation is a crucial part if the CFD solver provides only primary results, such as the pressure p or the velocity \mathbf{u} . Here, a prime denotes the fluctuating part of a physical quantity and a bar over a physical quantity its time average [Color figure can be viewed at wileyonlinelibrary.com]

Desirably, an accurate, flexible, and conservative coupling scheme ensures a rigorous connection between fluid dynamics and acoustics within a hybrid aeroacoustic simulation. The properties of the desired coupling scheme, or interpolation, are summed up as follows:

- An accurate and fast interpolation technique.
- An interpolation technique that handles special grids, for example, grids to resolve boundary layers.
- A method to compute accurate derivatives of the primary flow variables, such as pressure p , velocity \mathbf{u} , density ρ , temperature T , and entropy s .
- A flexible algorithm that can be integrated into a standard product development cycle.
- A flexible algorithm that assembles different hybrid aeroacoustic source terms, for example, the divergence of the Lamb vector $\nabla \cdot ((\nabla \times \mathbf{u}) \times \mathbf{u})'$ or the divergence of an entropy source $\nabla \cdot (T' \nabla \bar{s} - s' \nabla \bar{T})$.¹⁴
- A conservative algorithm that transfers the desired amount of energy, defined by the aeroacoustic sources, from the flow discretization to the mesh of the acoustic simulation (see eg, Reference 13;) is not discussed here.

In this paper, we propose to compute the aeroacoustic sources directly from the primary CFD variables by applying an interpolation scheme based on RBFs in conjunction with RBF derivatives. We show the ideal setup for the algorithms, the choice of the kernel function and propose a selection of discretization natural neighbors for the computation based on the connectivity. This patch search technique guarantees the resolution of typical flow structures. The application of local RBFs provides promising capabilities in terms of computational efficiency, known from nearest neighbor algorithms. Furthermore, the computation of RBF derivatives can be carried out elegantly and accurately with a local-global approach.

The rest of the paper is organized as follows: In Section 2, we provide a general overview about the selected interpolation scheme, discuss convergence based on the parameters, the grid characteristics, the neighbor search, and typical fluid dynamic effects. Boundary layer interpolation motivates the proposed neighbor search technique and the superior performance is shown in a convergence study. Furthermore, the integration of constraints is discussed. Next, in Section 3, we discuss our approach to compute spatial derivatives based on local-global RBFs, where we investigate the sensitivity of the scaling parameter based on verification benchmarks. The derivatives are judged by analytic representations of flow structures, such as a vortical distribution, a sinusoidal shock, and a boundary layer profile. Finally, in Section 5 we conclude the findings and give additional remarks for future research.

2 | INTERPOLATION SCHEMES

RBFs are scalar-valued multivariant functions $\mathbb{R}^d \mapsto \mathbb{R}$ that depend on the radial measure r in terms of the Euclidean norm $r = \|\mathbf{x} - \mathbf{z}\|_2$ between a scattered data (source) point \mathbf{x} and the evaluation (target) point \mathbf{z} (see Figure 2).

Over the years, application-driven development of RBFs has modified function type, shape, and relevant support. The most prominent RBF kernels are the Gaussian, the multiquadric, and the inverse multiquadric.¹⁵

At this point, a clear distinction between local and global methods should be drawn. Global methods mainly using the Gaussian kernel are well suited for theoretical analyses since it can be shown that the solution converges to the exact solution under certain conditions.¹⁶ But on the downside, the condition number of the global interpolation matrix tends to infinity. There are some approaches to circumvent this issue¹⁷ using a QR-decomposition or a Taylor-expansion but they are computationally more expensive and less parallelizable than local ones.

With increasing number of ill-placed datapoints in datasets, the demand for RBFs with compact local support¹⁸ increases. Local representations reduce computational time when interpolating data and increase the condition number

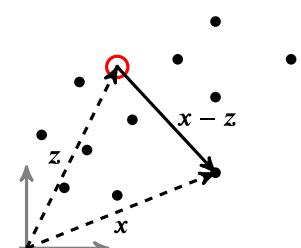


FIGURE 2 Geometric definitions of scattered data (\mathbf{x} , ●) and the target point (\mathbf{z} , ○). [Color figure can be viewed at wileyonlinelibrary.com]

of each local subproblem. Instead of inverting one ill-conditioned global system matrix, many small well-conditioned stencil-matrices are inverted. Using Wendland's compactly supported RBFs together with a modified Shepard's method as presented in Reference 19, high computational efficiency for large system can be achieved.

An important aspect of this work is the application of RBF interpolation schemes to scattered data distributions with bad quality, such as extremely anisotropic point distribution, for example, from boundary layer flow. We use a local collocation method that represents the data exactly in the prescribed point. In this sense no algorithmic uncertainty is present at the individual, deterministic, scattered data point. To handle anisotropic point distributions, a special search procedure is used to find the optimal local point distribution for interpolation, which will be presented in Section 2.2.

2.1 | RBF interpolation

As mentioned above, the main purpose of our method is to interpolate flow results, such as velocity \mathbf{u} or vorticity ω , from a CFD mesh to an acoustic finite element mesh (FEM). Since hybrid aeroacoustics deals with a large number of unknowns and unfortunate data distributions, for example, in boundary layers, we focus on a parallelizable algorithm to achieve high computational efficiency.

In this work, the local Wendland kernel Φ (1) together with a modified Shepard's method was chosen, similar to Reference 19. This is the most promising approach since it is both fast and capable of handling boundary layer meshes if the chosen kernel is combined with the patch search technique presented in Section 2.2.

The scaled Wendland kernel¹⁹ reads

$$\Phi(||\mathbf{x} - \mathbf{z}||_2, \alpha) = \left(1 - \frac{||\mathbf{x} - \mathbf{z}||_2}{\alpha}\right), \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the location of a scattered data point. For all scattered data points the associated indices $\omega_s \in 1, 2, \dots, N_s$ are defined, with N_s as the number of scattered data points, also called *source points*. The point $\mathbf{z} \in \mathbb{R}^d$ is the point at which the interpolation is evaluated. These points are called *target points* and form a set of target point indices $\omega_t \in 1, 2, \dots, N_t$, with N_t as the number of target points. The parameter α is responsible for the scaling of the compact support (1), which is especially important for bad source point distributions resulting in a high condition number of the interpolation matrices. For our investigations, the parameter α is chosen to be

$$\alpha \sim r_{\max} = \max_{i \in I_q} \{r_i\}, \quad (2)$$

with I_q as the indices of the N_q neighbors of the target point \mathbf{z} and define the patch set. It should be noted that with this RBF interpolation, a higher order of accuracy can be achieved by “flattening” the compact support of the kernel (see Section 3.1) or increasing the order of the local interpolation kernel.

2.1.1 | Local interpolation method

Following the approach of Lazzaro,¹⁹ two different scattered data patches around the target point at which the interpolant is evaluated are introduced. The set $X_q = \{\mathbf{x}_i \in \mathbf{x}, i \in I_q\}$, with I_q as the indices of the N_q neighbors of the target point \mathbf{z} , defines the patch set. The set $X_w = \{\mathbf{x}_i \in \mathbf{x}, i \in I_w\}$, with I_w as the indices of the N_w influence points, defines the influence of the different patches. The influence radius $r_{W_k} = \max_{k \in I_w} \{r_k\}$, with $r_k = ||\mathbf{x}_k - \mathbf{z}||_2$, is defined to be the maximum distance of a target point in X_w and $N_w \subset N_q$. The interpolant is given by

$$s(\mathbf{z}) = \sum_{k=1}^{N_w} \overline{W}_k(\mathbf{z}) R_k(\mathbf{z}), \quad (3)$$

where $R_k(\mathbf{z})$ defines the local interpolation system and \overline{W}_k the weight function. The local interpolation system is given by the algebraic system

$$R_k(\mathbf{x}_k) = \sum_{j=1}^{N_q} c_j \Phi(||\mathbf{x}_j - \mathbf{x}_k||_2), \quad (4)$$

which has to be initially solved for the N_q temporary (unscaled) interpolation weights c_j , where $R_k(\mathbf{x}_k)$ denotes the scattered data (field that is interpolated) at the patch source point \mathbf{x}_k . The weight function \overline{W}_k of the modified interpolant $s(\mathbf{x})$ is chosen to be

$$\overline{W}_k(\mathbf{x}) = \frac{\left(\max \left\{ \frac{(r_{W_k} - r_k)}{r_{W_k} r_k}, 0 \right\} \right)^p}{\sum_{l=1}^{N_q} \left(\max \left\{ \frac{(r_{W_k} - r_l)}{r_{W_k} r_l}, 0 \right\} \right)^p}, \quad (5)$$

with the exponent p as a measure of locality. The bigger p , the more local the approach making it less accurate but capable of resolving stronger gradients. Since the weights $\overline{W}_k(\mathbf{x})$ constitute a partition of unity, the following holds:

$$\sum_{k=1}^{N_w} \overline{W}_k(\mathbf{x}) = 1. \quad (6)$$

Then the interpolant $s(\mathbf{x})$ can be evaluated at every target point \mathbf{z} . The algorithm of the local RBF interpolation is written in pseudo code (Algorithm 1).

Algorithm 1. Local RBF Interpolation

```

1:  $s(\mathbf{z}) \leftarrow \text{RBFInterpolation}(\mathbf{x})$ 

2: function RBFInterpolation
3:   for  $k$  in  $\omega_t$  do
4:      $I_q = \text{GetNodeIndicesOfPatch}(k)$            //  $I_q$  holds indices of  $N_q$  source points in patch
5:      $\mathbf{x}_k = \text{GetCoordsOfSrcPointsForTrgPoint}(k)$ 
6:      $\mathbf{R}_k = \text{GetSourceData}(I_q)$            //get scattered data at patch source point
7:     Solve the  $N_q \times N_q$ -system for the patch coefficients  $\mathbf{c}^k$ , Eq. (4)
8:      $i_w = I_q[\text{influence-point}]$            //get index of influence point
9:      $r_{W_k} = \text{GetDistOfPoint}[i_w]$            //distance between target point and  $i_w$ -th source point
10:    Compute weights for every source point in patch  $k$ :  $\overline{W}_k$ , Eq. (5)
11:    Compute interpolant  $s(\mathbf{z}_k)$  at target point  $\mathbf{z}_k$ , Eq. (3)
12:   end for
13: end function

```

Proof of convergence of the interpolation. As described by Lazzaro in Reference 19, an upper bound for the global approximation error $E(\mathbf{x})$ reads

$$E(\mathbf{x}) = ||f(\mathbf{x}) - s(\mathbf{x})|| = \left\| \sum_{k=1}^{N_q} \overline{W}_k(\mathbf{x}) f(\mathbf{x}) - \sum_{k=1}^{N_q} \overline{W}_k(\mathbf{x}) R_k(\mathbf{x}) \right\| \leq \sum_{k=1}^{N_q} \overline{W}_k(\mathbf{x}) e_k(\mathbf{x}), \quad (7)$$

with $e_k(\mathbf{x}) = ||f(\mathbf{x}) - R_k(\mathbf{x})||$ as the approximation error of the interpolant $R_k(\mathbf{x})$ relative to the local interpolation system. Additionally, Madych has proven in Reference 16 that the following holds if the distance h between the source points is sufficiently small:

$$e_k(\mathbf{x}) = ||f(\mathbf{x}) - R_k(\mathbf{x})|| \leq C_1 e^{-C_2 \alpha} ||f||_2, \quad (8)$$

where $C_1, C_2 \in \mathbb{R}^+$ are constants. Thus for the shape parameter of the Wendland kernel (1) $\alpha \rightarrow \infty$, the approximation error of the local interpolation system $e_k(\mathbf{x})$ goes to zero and hence from Equation (7) it can be concluded that the same is true for the global approximation error $E(\mathbf{x})$. ■

2.2 | On the optimal choice of neighbors

An important part of the interpolation is the choice of source points \mathbf{x}_i in the patch X_q to avoid unphysical artifacts in the interpolated field. This is especially important in boundary layers, where cells or elements are distorted or stretched with high aspect ratios. A common approach to obtain a specific amount of nearest neighbors is to use a kd-tree search, for example, from computational geometry algorithms library²⁰ or fast library for approximate nearest neighbors.²¹

These fast kd-tree algorithms have to be provided with the coordinate of the target point for which the patch is defined and all coordinates of the source points. By doing this, the underlying mesh structure (connectivity) is ignored and the data is represented on discrete scattered points (see Figure 3A). Thus, important information to improve the quality of the interpolation might be missed. To understand the shortcoming of this approach, a simple example of a boundary layer flow in a channel is provided, where the cell data from a CFD simulation is interpolated to nodes of a different FEM mesh. The interpolation from a CFD Finite Volume mesh to an FEM mesh is our standard procedure; however, the findings are also true for other discretization types without loss of generality.

The interesting part is the area close to the wall, where a mesh refinement is necessary to properly resolve the boundary layer. The resulting patch of elements emerging from a kd-tree search for two example nodes on the boundary layer is shown in Figure 3A. There we can see the unfavorable choice of source points (black) for different target points (red circle). As shown in Figure 3B, this patch choice results in a poor interpolation in streamwise direction, which leads to numerical artifacts. The results become even worse if we provide a heavily distorted mesh. There we can observe that the patches only contain vertical neighbors and thus the patches have no connection in streamwise direction. A common approach to circumvent this issue is using anisotropic search algorithms and anisotropic kernel functions. In contrast to the herein described connectivity-based neighbor search, additional tuning parameters have to be determined for local mesh features.

However, if we use a connectivity preserving mesh-based neighbor search, the patch looks as displayed in Figure 4A, which results in an even spatial distribution and good interpolation results (see Figure 4B). This modified patch algorithm searches for the next directly connected neighbor elements over the common global nodes. By applying a “layer strategy,” the neighbors of the direct neighbors are also taken into account. This search can be carried out recursively in a layered manner until the preferred number of elements is reached. In the illustrated example, only a one-layer level search was applied.

Another drawback of a coordinate based kd-tree search is the interpolation along slender wing profiles and structures. It is possible that a kd-tree search at the upper side of the wing includes points from the lower side in the patch, which would be completely incorrect. In this case, the modified patch search based on the connectivity performs well since elements on the upper side have no direct connection to the elements on the lower side of the slender airfoil.

With the given interpolation formulation it is possible to impose exact boundary conditions, for example, a no-penetration condition $\mathbf{u} \cdot \mathbf{n} = 0$, where the corresponding entries of the interpolant can simply be set equal to zero.

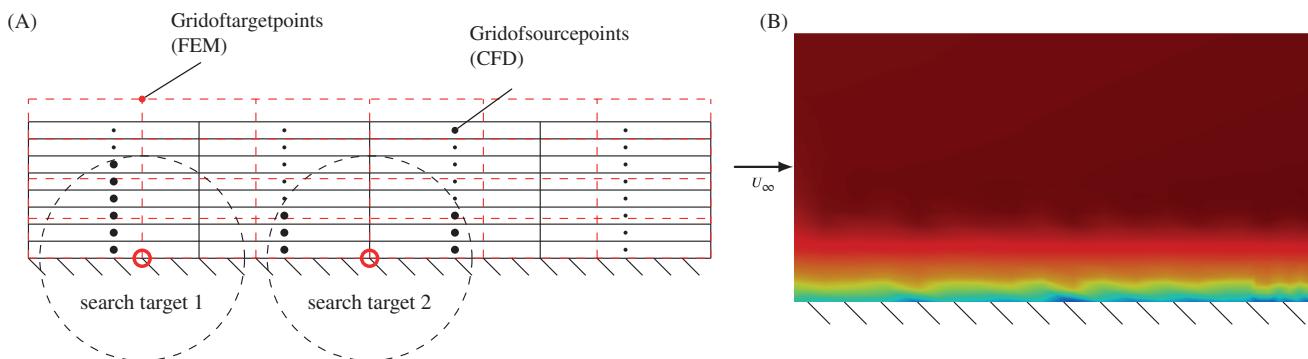


FIGURE 3 kd-tree search and resulting interpolated flow field. A, The search of a kd-tree patch is based on a metric (eg, euclidean) where all source points inside the influence domain (black dashed circle) surrounding the target point (red circle) are selected. For different mesh discretizations, different search patterns arise. These patterns severely influence the quality of the interpolation; B, The velocity field of a CFD simulation was interpolated to a different grid with the ordinary kd-tree search algorithm. Nonphysical artifacts occur in the boundary layer [Color figure can be viewed at wileyonlinelibrary.com]

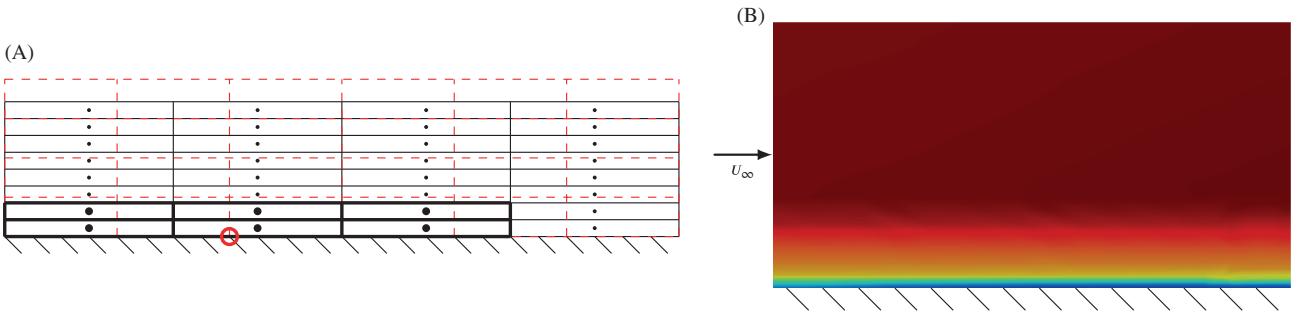


FIGURE 4 Modified patch search and resulting interpolated flow field. A, The modified patch search is based on the connectivity of the mesh. Only direct neighbors of the origin element are selected by a layer strategy. This ensures that the neighborhood discretizing the physical phenomena is used to interpolate the data and connections on meshes with high aspect ratios are maintained; B, The velocity field of a CFD simulation was interpolated to a different grid with the mesh based search algorithm. No artifacts occur in the boundary layer and the high quality of the flow field is maintained [Color figure can be viewed at [wileyonlinelibrary.com](#)]

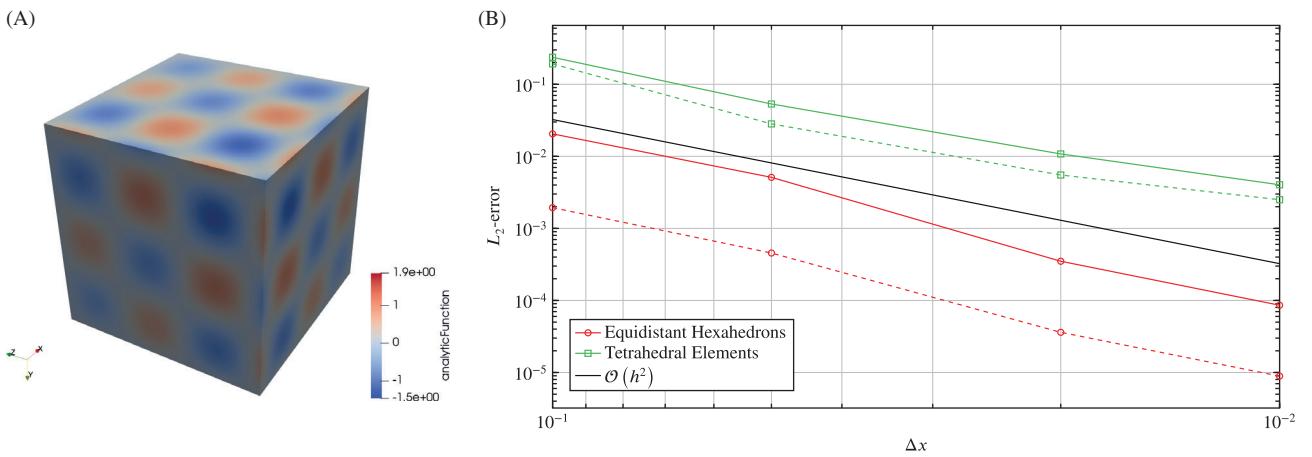


FIGURE 5 Convergence of the interpolation for the first analytic function $f_1(\mathbf{x})$. A, Representation of the first analytic function $f_1(\mathbf{x})$; B, Illustration of the L_2 -error for the interpolation of the first analytic function on different mesh types using a nearest neighbor search (solid) and patch search (dotted). The target mesh discretization is $\Delta z=0.1$ [Color figure can be viewed at [wileyonlinelibrary.com](#)]

2.3 | Convergence of interpolation

To analyze the convergence of the RBF interpolation and to quantify the more or less heuristic observations of correct choice of patches, we prescribe analytic functions on a unit cube $\Omega \in [0, 1]^3$ with different node distributions and observe the L_2 -error after interpolating to a different mesh with equidistant node distribution.

The first analytic function

$$f_1(\mathbf{x}) = (x^2 + y^2 + z^2) \sin(10x) \sin(10y) \sin(10z) \quad (9)$$

represents the transition of different vortex structures in shear layers (see Figure 5A). This smooth function is used to investigate the convergence and accuracy of the RBF interpolation since the majority of pressure fields encountered in low Mach number aeroacoustics are smooth. The second analytic function

$$f_2(\mathbf{x}) = \tanh(20(x + 0.3 \sin(-10y) - 0.3 \sin(-5(z - 0.1)))) , \quad (10)$$

is a representative of a continuous sinusoidal shock, which is similar to the numerical solution of a shock using a dissipative flow solver²² (see Figure 6A). This function is used to test the capabilities of RBF interpolation to handle strong gradients and sharp transitions.

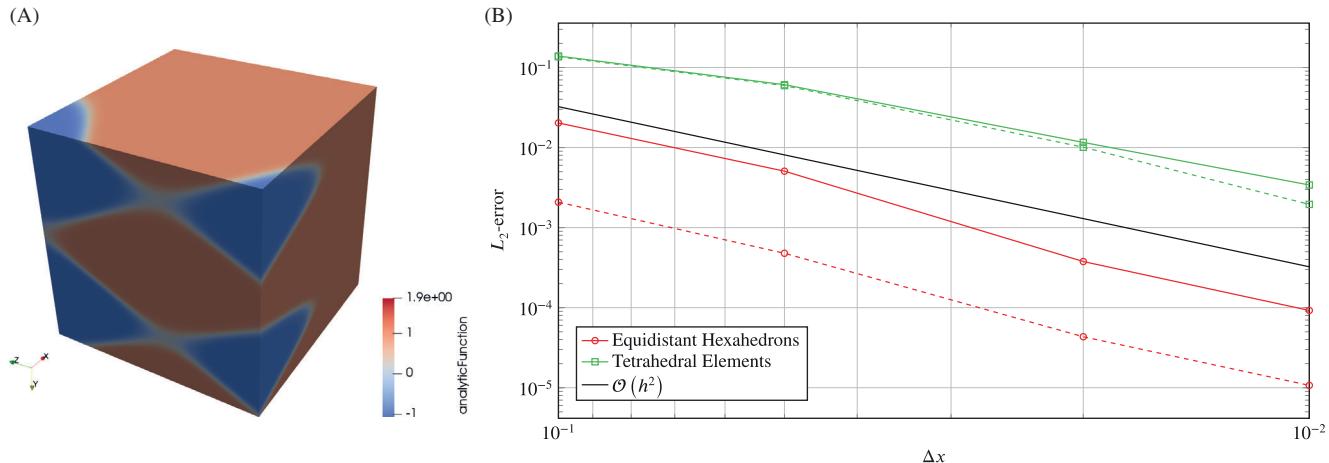


FIGURE 6 Convergence of the interpolation for the second analytic function $f_2(\mathbf{x})$. A, Representation of the second analytic function $f_2(\mathbf{x})$; B, Illustration of the L_2 -error for the interpolation of the second analytic function on different mesh types using a nearest neighbor search (solid) and patch search (dotted). The target mesh discretization is $\Delta z=0.1$ [Color figure can be viewed at wileyonlinelibrary.com]

These analytic functions were prescribed on a unit cube $\Omega \in [0, 1]^3$ consisting of equidistant hexahedrons, and tetrahedron elements. Then the analytic functions were interpolated to a different mesh consisting of equidistant hexahedrons with discretization distance Δz using a nearest neighbor search or the modified patch search, respectively.

Figures 5B and 6B show the convergence of the RBF interpolation with respect to the source discretization distance Δx , being the edge length of an element, for the first and second analytic function. Obviously, there is an influence resulting from the source data density, and the RBF interpolation performs well for both the nearest neighbor search and the modified patch search on all of the different mesh types. The larger L_2 -error of the tetrahedron elements is due to an unsymmetric interpolation point distribution around the target point, which is suboptimal for RBF and leads to higher errors. However, the order of convergence remains the same.

In order to further analyze the capability of the RBF interpolation to handle boundary layer data in combination with the different choices of neighbors, we use the function

$$f_3(\mathbf{x}) = \frac{10}{4.6151} (\log(y + 10^{-2}) - \log(10^{-2})) . \quad (11)$$

The analytic function $f_3(\mathbf{x})$ roughly describes the flow velocity component in inflow direction in a boundary layer (see Figure 7A). Here, the wall that is the reason for the boundary layer to develop is located at $y = 0$. As mentioned above, a CFD mesh to properly resolve the boundary layer is characterized by cells with a big aspect ratio in the region adjacent to the wall where the largest gradients occur. For our investigations, we used a mesh consisting of regular hexahedrons with different aspect ratios in the range of 2 to 512 for the cells at $y = 0$ (typical CFD boundary layer mesh²³). With growing distance y the mesh gradually becomes coarser. In order to realize different aspect ratios in the first cell row, the cell height is varied while the number of nodes on the edges stays the same.

As shown in Figure 7B, the kd-tree-based nearest neighbor search interpolation causes increasing errors with increasing aspect ratio. Hence, the interpolation error for large aspect ratios limits the application of the nearest neighbor search. However, the previously described patch search leads to better results at large aspect ratios where it significantly outperforms the kd-tree-based nearest neighbor search. Ideally, the L_2 -error is independent of the aspect ratio. One can see that this requirement is approximately met using patch search.

3 | DERIVATIVES BASED ON RBFS

Although the procedure to compute derivatives based on RBFs is similar to the RBF interpolation, there is one important difference. For derivatives, there is only one source patch set $X_q = \{\mathbf{x}_i \in \mathbf{x}, i \in I_q\}$, with I_q as the indices of the N_q neighbors of the target point \mathbf{z} .

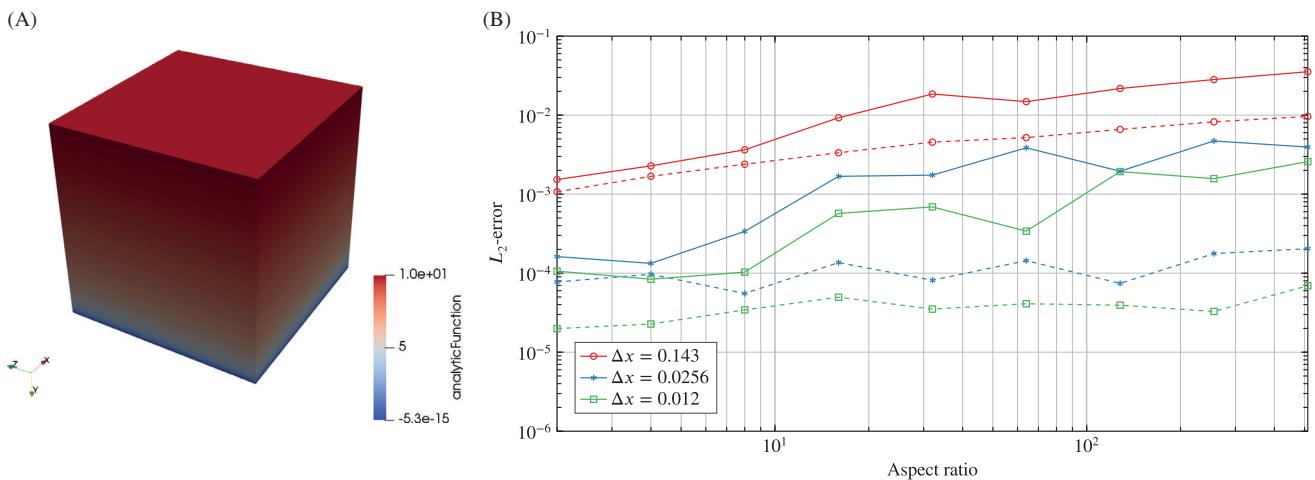


FIGURE 7 Convergence of the interpolation for the third analytic function $f_3(\mathbf{x})$ on a mesh to resolve a boundary layer. A, Representation of the third analytic function $f_3(x)$; B, Illustration of the L_2 -error for the interpolation of the third analytic function on hexahedron mesh with different source discretizations using a nearest neighbor search (solid) and patch search (dotted). The target mesh discretization is $\Delta z=0.1$ [Color figure can be viewed at wileyonlinelibrary.com]

To explain the derivative of a given function $s(\mathbf{z})$ at target point \mathbf{z} , let us start using matrix notation. We first need to interpolate the values at discrete datapoints \mathbf{x} to determine the interpolation weights σ . Thus we need to form the interpolant $\mathbf{s}(\mathbf{x})$

$$\mathbf{s}(\mathbf{x}) = \Phi(\mathbf{x}) \cdot \sigma, \quad (12)$$

and solving this system for the weights σ leads to

$$\sigma = \Phi^{-1}(\mathbf{x}) \cdot \mathbf{s}. \quad (13)$$

Since we want to apply the differential operator $\mathcal{L}[\]$ to the interpolant, we may write

$$\mathcal{L}[\mathbf{s}(\mathbf{z})] = \mathcal{L}[\Phi(\mathbf{z}) \cdot \sigma], \quad (14)$$

and inserting the weights from Equation (13) yields

$$\mathcal{L}[\mathbf{s}(\mathbf{z})] = \mathcal{L}[\Phi(\mathbf{z}) \cdot \Phi^{-1}(\mathbf{x}) \cdot \mathbf{s}]. \quad (15)$$

Now we can use the fact that $\sigma = \Phi^{-1}(\mathbf{x}) \cdot \mathbf{s}$ does not depend on \mathbf{z} and we can pull it out of the operator's range

$$\mathcal{L}[\mathbf{s}(\mathbf{z})] = \mathcal{L}[\Phi(\mathbf{z})] \cdot \underbrace{\Phi^{-1}(\mathbf{x}) \cdot \mathbf{s}}_{\sigma}, \quad (16)$$

and we can rewrite more compactly in the following way:

$$\mathcal{L}[\mathbf{s}(\mathbf{z})] = \underbrace{\mathcal{L}[\Phi(\mathbf{z})] \cdot \Phi^{-1}(\mathbf{x})}_{c(\mathbf{z})} \cdot \mathbf{s}. \quad (17)$$

The process from above can now be transformed to a more implementation-friendly form, based on Reference 24, where we first have to apply the spatial derivative operator \mathcal{L} to the RBF and evaluate it, as presented in Equation (18), where \mathbf{x}_k and \mathbf{x}_l are the coordinates of two source points with indices $k, l \in I_q$.

$$\mathbf{L}_\Phi^k = \mathcal{L}[\Phi(||\mathbf{z} - \mathbf{x}_k||_2)] = \sum_{l=1}^{N_q} c_l \Phi(||\mathbf{x}_l - \mathbf{x}_k||_2), \quad k = 1, \dots, N_q, \quad (18)$$

whereas the derivative of the Gaussian kernel was computed analytically. This system has to be inverted in order to obtain the weights c_l for the derivatives at the source points $l \in I_q$. Then the derivative of the given function $s(\mathbf{z})$ can be evaluated by

$$\mathcal{L}[s(\mathbf{z})] = \sum_{l=1}^{N_q} c_l s(\mathbf{x}_l) = \sum_{l=1}^{N_q} c_l R_l, \quad (19)$$

where $R_l = s(\mathbf{x}_l) \in \mathbb{R}^{N_q \times 1}$ is the scattered data value of the given function at the source point with index l .

Algorithm 2 shows how to calculate derivatives of scattered data distributions. Vector-valued datasets can be interpreted as a tuple of scalar components

$$\mathbf{s}(\mathbf{x}) = \begin{pmatrix} s_1(\mathbf{x}) \\ s_2(\mathbf{x}) \\ s_3(\mathbf{x}) \end{pmatrix}. \quad (20)$$

Then \mathbf{R}_k is a matrix $\mathbb{R}^{N_q \times d}$ with d as the spatial dimension and evaluating the last expression of Algorithm 2 leads to a Jacobian matrix (21), where each row i contains the derivatives of the scattered data vector.

Algorithm 2. Local RBF Derivative

```

1:  $s(\mathbf{z}) \leftarrow \text{RBFDerivative}(\mathbf{x})$ 

2: function RBFDIFFERENTIATION
3:   for  $t$  in  $\omega_t$  do
4:      $I_q = \text{GetNodeIndicesOfPatch}(t)$            //  $I_q$  holds indices of the  $N_q$  source points in the patch
5:      $\mathbf{x}_k = \text{GetCoordsOfSrcPointsOfTrgPoint}(t)$ 
6:     for  $k$  in  $I_q$  do
7:        $\mathbf{L}_\Phi^k = \text{CalcDerivativesOfRBF}(\mathbf{x}_i)$       //  $\mathcal{L}[\Phi(||\mathbf{x}_k - \mathbf{x}_i||_2)] \rightarrow \mathbf{L}_\Phi^k \in \mathbb{R}^{N_q \times d}$ 
8:        $\Phi = \text{EvalRBFat}(\mathbf{x}_k, \mathbf{x}_i)$           //  $\Phi \in \mathbb{R}^{N_q \times N_q}$ 
9:        $\mathbf{L}_\Phi^k = \Phi \cdot \mathbf{c}(\mathbf{x}_k, \mathbf{z})$         // Compute the weight for source point  $k$ , Eq. 18
10:    end for
11:     $\mathbf{R}_t = \text{GetSourceData}(I_q)$                 // Get the scattered data at patch source point
12:     $s(\mathbf{z}_t) = \mathbf{c} \cdot \mathbf{R}_t$                   // Derivatives  $s(\mathbf{z}_t)$  at  $\mathbf{z}_t$ , Eq. 19
13:  end for
14: end function
```

To show how to form the desired differential operator from the spatial differentiation matrix and to outline the workflow, a simple example is presented. Assume we have a scattered data vector field $(s_1, s_2, s_3)^T(\mathbf{z}) \in \mathbb{R}^3$. Then the rows of \mathbf{L}_Φ^i contain the spatial derivatives of the three-dimensional RBF in the three spatial directions $\left(\frac{\partial \Phi}{\partial z_1}, \frac{\partial \Phi}{\partial z_2}, \frac{\partial \Phi}{\partial z_3} \right)$. Additionally, the derivative coefficients $\mathbf{c}[i]$ for source point i are three-dimensional vectors. The final evaluation $\mathbf{S}(\mathbf{z}) = \mathbf{c} \cdot \mathbf{R}(\mathbf{z})$ leads to the following 3×3 matrix

$$\begin{pmatrix} \frac{\partial s_1}{\partial z_1} & \frac{\partial s_1}{\partial z_2} & \frac{\partial s_1}{\partial z_3} \\ \frac{\partial s_2}{\partial z_1} & \frac{\partial s_2}{\partial z_2} & \frac{\partial s_2}{\partial z_3} \\ \frac{\partial s_3}{\partial z_1} & \frac{\partial s_3}{\partial z_2} & \frac{\partial s_3}{\partial z_3} \end{pmatrix}. \quad (21)$$

3.1 | Kernel scaling

As mentioned above, we use a Gaussian kernel in a local approach. The local kernel has no numerical stability problem in its basic form and works well on regular grids. But as soon as the mesh becomes distorted (eg, boundary layer) the results worsen. An improvement can be achieved by scaling the Gaussian ansatz function in the following way:

$$\Phi(||\mathbf{x} - \mathbf{z}||_2, \alpha) = e^{\left(\frac{||\mathbf{x} - \mathbf{z}||_2}{\alpha}\right)^2}. \quad (22)$$

We found that the Gaussian kernel theoretically leads to high accuracy (in theory we can differentiate up to any given order) but the stability problem, similar to the global interpolation method, restricts the maximum flatness α . Roberts proposed a method using a Taylor's expansion in the flatness parameter α to circumvent this issue.²⁵ However, there are optimization strategies¹⁵ to find optimal values of the flatness parameter α .

It should be emphasized here that the parameter α is computed differently depending on whether an interpolation or derivation procedure is carried out. Our idea to find an optimal shape parameter for the RBF derivative procedure is to derive a relationship (a confident initial guess)

$$\alpha \sim f(r_{12}), \quad (23)$$

for the parameter α with respect to the value r_{12} as the smallest distance between two points in the stencil of the current target point. The basic approach is to minimize the following objective functional

$$E(\alpha, r_{12}) + \lambda(\kappa(\alpha, r_{12}) - \kappa_{\text{limit}}), \quad (24)$$

with the L_2 -error E , the Lagrange multiplier λ , the condition number κ , and the limit of the condition number κ_{limit} where the system becomes sensitive to floating-point round-off errors. The L_2 -error represents the deviation of the basis function from being one (absolutely flat, $\alpha \rightarrow \infty$) on the whole domain. This error measure can simply be obtained by integrating

$$E(\alpha, r_{12}) = \int_{\Omega_{2/3}} |1 - \Phi(||\mathbf{x} - \mathbf{z}||_2, \alpha)|^2 d\Omega. \quad (25)$$

The challenging task is to find the influence of the shape parameter α on the condition number of the resulting linear system. In general, the condition number κ of a matrix is defined as the ratio of the largest to the smallest eigenvalue:

$$\kappa(\alpha, r_{12}) = \frac{\sigma_{\max}(\alpha, r_{12})}{\sigma_{\min}(\alpha, r_{12})}. \quad (26)$$

Therefore, the smallest eigenvalue must be nonzero in order to stay invertible.

The minimization is trivial since the functional decreases monotonically in α while a critical condition number κ_{limit} guarantees the invertibility of the matrix. With this constraint, our confident initial guess α_{initial} is

$$\alpha_{\text{initial}} \sim 1/r_{12} \quad (27)$$

and from this starting point the shape parameter is iterated to the limit of the matrix invertibility, depending on the inversion algorithm.

As proven multiple times in the literature dealing with Gaussian RBF kernels (eg, in Reference 19), the L_2 -error decreases with increasing shape parameter α . However, this is true only up to a certain value, where numerical instabilities tend to increase the error again.

3.2 | Additional notes

During the computation, there is an easy way to assess how accurate the derivative will be. Looking at the last expression of Algorithm 2, where we multiply the weight matrix $\mathbf{c} \in \mathbb{R}^{d \times N_q}$ and the source data matrix $\mathbf{R} \in \mathbb{R}^{N_q \times d}$, we can state

that the rowsum of each row must be zero. This can be seen easily by considering a constant field, where all entries are equal $R_{ij} = \text{const.}$ and the resulting derivative must be zero (derivative of a constant function) $S_i(\mathbf{z}) = \sum_{j=0}^{N_q} c_{ij} R_j = 0$. For nontrivial coefficients, this only holds if the rowsum is zero $\sum_{j=0}^{N_q} c_{ij} = 0$ for $i = 1, \dots, d$. This rowsum condition is used for optimal neighbor consideration.

During the investigations, the local Wendland kernel $\Phi_{2,0} = (1 - \|\mathbf{x} - \mathbf{z}\|_2)^2$ works perfectly for generic test cases with moderately distorted elements. But as soon as the Wendland kernel is applied to more sophisticated problems with boundary layers, severe numerical artifacts are observed, especially when computing the divergence. Therefore a local approach with a Gaussian kernel (22) is used, which produces accurate results for a wide range of mesh discrepancies.

3.3 | Convergence of derivatives

To analyze the convergence and accuracy of the RBF derivatives, we again prescribe our analytic functions Equations (9) to (11) on a unit cube $\Omega \in [0, 1]^3$ with different node distributions using the patch search algorithm and observe the L_2 -error after computing the gradient of the field.

Figure 8A shows the convergence of the RBF derivatives with respect to the characteristic discretization distance Δx for the first and second analytic function. Similar to the RBF interpolation, there is an influence resulting from the data density. Although the RBF derivatives are accurate for both analytic functions, it is obvious that larger gradients (as encountered in the second analytic function) lead to larger deviations.

Additionally, the convergence of the derivatives in boundary layers was analyzed considering the analytic function (11). This analytic function was again prescribed on a mesh consisting of regular hexahedrons with different aspect ratios in the range of 2 to 128 for the cells in the first row.

As shown in Figure 8B, the L_2 -error decreases with increasing number of nodes in the region where the largest gradients occur (next to the wall). However, since the number of nodes on the edges perpendicular to the wall stays the same while the height of the cells is varied, a point is reached eventually where the nodes are not distributed well enough over the entire region of the largest gradients. This is the reason for the flattening of the curves for hexahedrons with large aspect ratios.

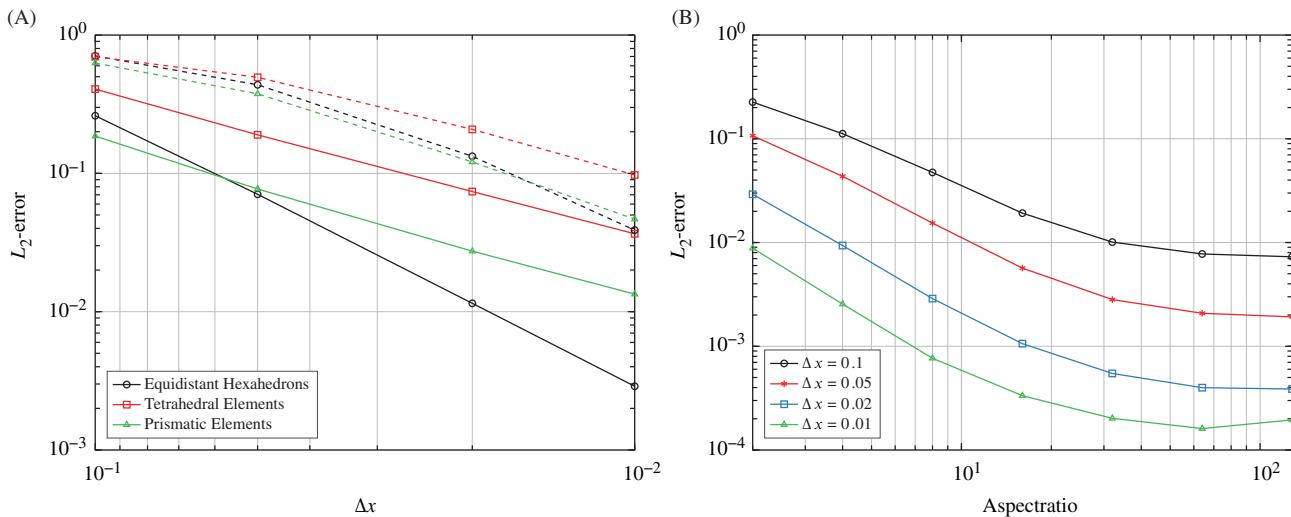


FIGURE 8 Convergence of the derivatives for the analytic functions f_1 (9), f_2 (10) and f_3 (11). A, Illustration of the L_2 -error for the gradient of the first analytic function f_1 (solid) and the second analytic function f_2 (dashed) with respect to the mesh discretization; B, Illustration of the L_2 -error for the gradient of the third analytic function f_3 on a hexahedron mesh with respect to the mesh discretization [Color figure can be viewed at wileyonlinelibrary.com]

4 | NUMERICAL EXAMPLE

The rotating vortex pair has been frequently used to determine the capabilities of aeroacoustic methodologies.²⁶⁻²⁹ This arrangement has the nature of a quadrupolar sound field.

Figure 9 illustrates the configuration of the vortex pairs. Both vortices are delta distributions and oppose each other at a distance of $2r_0$. The strength of each vortex is characterized by the circulation intensity Γ . The vortices rotate around the origin with a period of $T = 8\pi^2 r_0^2 / \Gamma$ imposing an angular rotating speed $\omega_r = \Gamma / (4\pi r_0^2) \cdot \mathbf{e}_3 = \omega_r \cdot \mathbf{e}_3$. Each vortex convects the other vortex by a velocity of $\mathbf{u}_\theta = \Gamma / (4\pi r_0) \cdot \mathbf{e}_t$, where \mathbf{e}_t is the unit vector in tangential direction. The Mach number in the circumferential direction is given by $M_\theta = u_\theta / c = \Gamma / (4\pi r_0 c)$. As already mentioned, the hydrodynamic field of the spinning vortex pair is expressed by a rotating quadrupole field. The potential flow theory can be used to determine the fundamental solution of the spinning vortex pair in terms of the complex flow potential function. In doing so, we introduce the transformation from Cartesian coordinates (x_1, x_2) to the complex plane with the complex coordinate $z = r \exp^{i\theta} = x_1 + ix_2$. The location of each vortex over time t is defined by $b = r_0 \exp^{i\omega_r t}$. Using the definitions, we express the incompressible, inviscid flow potential $\varphi(z, t)$ as

$$\Phi(z, t) = \frac{\Gamma}{2\pi i} \ln(z - b) + \frac{\Gamma}{2\pi i} \ln(z + b) = \frac{\Gamma}{2\pi i} \ln(z^2 - b^2). \quad (28)$$

From this field we can derive the primary variables representing the fluid dynamic quantities. Based on the pressure and the velocity, aeroacoustic sources are derived. The incompressible velocity field $\mathbf{u} = (u_1, u_2)^T$ of the spinning vortex pair is obtained by differentiating Equation (28) with respect to the complex coordinate z

$$u_1 - iu_2 = \frac{\partial \varphi(z, t)}{\partial z} = \frac{\Gamma}{\pi i} \frac{z}{z^2 - b^2}. \quad (29)$$

The fluid dynamic pressure p^{ic} is obtained by applying the unsteady form of Bernoulli's principle

$$p^{ic} = p_0 - \rho_0 \frac{\partial \mathbf{Re}\{\varphi(z, t)\}}{\partial t} - \frac{1}{2} \rho_0 (u_1^2 + u_2^2). \quad (30)$$

Müller and Obermeier³⁰ derived an analytic solution of the acoustic far-field, based on matched asymptotic expansion of the potential solution. Starting from the solution in form of a complex potential (28), matching the inner and the outer solution yields the far-field of the acoustic pressure fluctuation p' of the co-rotating vortex pair

$$p' = \frac{\rho_0 \Gamma^4}{64\pi^3 r_0^4 c^2} (J_2(2kr) \cos(2(\omega t)) - Y_2(2kr) \sin(2(\omega t))). \quad (31)$$

In Equation (31) $k = \omega/c$ denotes the wave number, $J_2(\star)$ the second-order Bessel function of first kind and $Y_2(\star)$ the second kind. This analytic pressure fluctuation p' is used to verify both aeroacoustic methodologies, that is, using the divergence of Lamb vector and the first substantial time derivative of the incompressible pressure as acoustic source

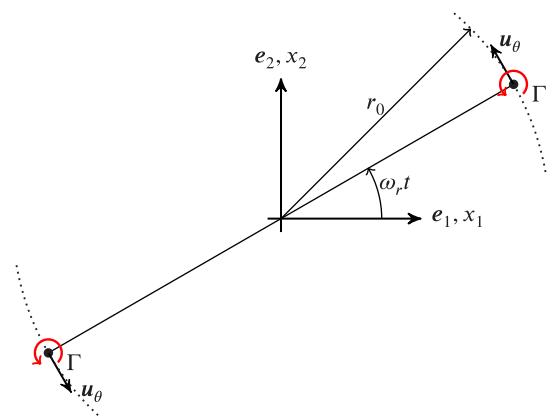


FIGURE 9 Schematic of the co-rotating vortex pair defining the main geometrical and physical characteristics [Color figure can be viewed at wileyonlinelibrary.com]

terms. It should be emphasized that this fluctuating pressure p' is not equal to the acoustic pressure p^a ; however, $p' \rightarrow p^a$ holds in the far-field.

4.1 | Numerical investigation

In order to judge the computational procedure,^{26–29} we compare the analytic solution of the test case with a numerically obtained solution based on the simulation procedure that is illustrated in Figure 1. This analytic validation has the benefit of mitigating experimentally induced errors, reducing the overall modeling uncertainty of the aeroacoustic model, and simply avoiding errors of a numerical benchmark routine. We simulated the co-rotating vortex pair on a stationary grid with moving sources induced by the vortical structures. An unstructured mesh is used to discretize the computational domain. In the source region, a characteristic mesh size of $h \approx 90$ cm is used. Each vortex distribution $\Gamma\delta(z - b)$ is approximated by a continuous multivariate normal distribution with equivalent circulation Γ and an isotropic variance of $\sigma^2 = 0.05$ m².

We assume that the analytic field (\mathbf{u}, p^{ic}) is represented on the flow grid and is based on a circulation strength of $\Gamma = 2\pi$ m²/s and a distance of $2r_0 = 2$ m between the vortices. The angular rotation induced by the vortices is $\omega_r = 0.5$ /s, the speed of sound $c = \sqrt{10}$ m/s and density $\rho_0 = 1$ kg/m³. Using this flow field, we apply the source term computation procedure and simulate the sound which is compared to the analytic solution in the far-field (31). We verify the interpolation methodology by two hybrid aeroacoustic simulations of the co-rotating vortex pair. The first simulation is based on vortex sound and the divergence of the Lamb vector as source term³¹

$$\frac{1}{c^2} \frac{\partial^2 p'}{\partial t^2} - \Delta p' = \rho_0 \nabla \cdot (\boldsymbol{\omega} \times \mathbf{u}) . \quad (32)$$

The second verification example is based on the perturbed convective wave equation (PCWE) with the substantial derivative of the incompressible pressure as aeroacoustic excitation¹³

$$\frac{1}{c^2} \frac{D^2 \psi^a}{Dt^2} - \Delta \psi^a = \frac{1}{\rho_0 c^2} \frac{D p^{ic}}{Dt} . \quad (33)$$

In Equation (33), ψ^a denotes the acoustic velocity potential and $\frac{D}{Dt} = \frac{\partial}{\partial t} + \bar{\mathbf{u}} \cdot \nabla$ the substantial derivative. The acoustic pressure can be computed as $p^a = \rho_0 \frac{D \psi^a}{Dt}$.

Based on the computational methodology of RBFs, we propose the following simulation workflow for the computation of the aeroacoustic source terms. First, the primary fields incompressible pressure p^{ic} , flow velocity \mathbf{u}^{ic} , and vorticity $\boldsymbol{\omega}$ are computed in a flow simulation on the CFD mesh. Second, if required, derivatives are computed by the RBF framework and the source term is assembled on the CFD mesh. Naturally, the RBF framework is carried out on the discretization of the flow simulation, since this discretization already resolves characteristic flow structures. As a third step, a conservative integration¹³ of the source terms ensures a nondissipative calculation of the aeroacoustic finite element sources that are applied to the finite element simulation. Finally, the acoustic simulation is carried out, typically on a much coarser CAA mesh.

The source term of Equation (32) involves the cross-product to determine the Lamb vector $\boldsymbol{\omega} \times \mathbf{u}$ and employing RBF derivatives yields the divergence of the Lamb vector $\nabla \cdot (\boldsymbol{\omega} \times \mathbf{u})$. Of course, the presented framework is capable of calculating the vorticity as the curl of the velocity field $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, with the source term $\nabla \cdot ((\nabla \times \mathbf{u}) \times \mathbf{u})$.

For the PCWE (33), the proposed workflow is as follows. The substantial derivative of the incompressible pressure is computed from primary CFD quantities. With the use of RBF derivatives, the gradient of the incompressible pressure is evaluated and the inner product with the velocity field is computed. We utilize a fifth-order backward differencing scheme for the time derivative of the pressure. Then the source term is assembled for the acoustic simulation.

For both acoustic simulations, a Newmark scheme with a time step size of $\Delta t = 0.09$ s is applied for the time discretization. The computational domain (260x260m) is structured in three separate conforming subdomains. A source domain consisting of a disk with a diameter of $5r_0$ (relatively high resolved unstructured triangular mesh, $h \approx 90$ cm) is embedded in a propagation region, in which we gradually increase the unstructured mesh size. The propagation region is surrounded by a structured perfectly matched layer region,³² which absorbs the radiating waves. The wave length $\lambda = \frac{2\pi}{k} \approx 20$ m is resolved (in the propagation region) with approximately 20 linear finite elements per wavelength.

FIGURE 10 Acoustic fields of the co-rotating vortex pair. A, Result of Equation 32 with the Lamb vector as source term; B, Result of Equation 33 with the substantial derivative of the incompressible pressure as source term [Color figure can be viewed at wileyonlinelibrary.com]

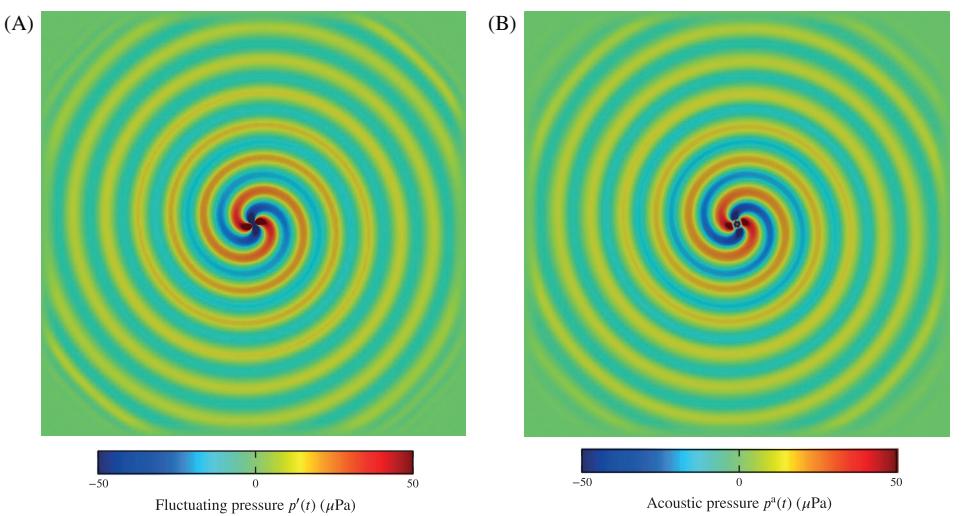


FIGURE 11 Fluctuating pressure p' or acoustic pressure p^a , respectively, as a function of the coordinate x_1 . Both the coarse and the fine discretization are simulated with the divergence of the Lamb vector as right-hand side of the wave equation and compared to the sound field of the perturbed convective wave equation as well as the analytic solution [Color figure can be viewed at wileyonlinelibrary.com]

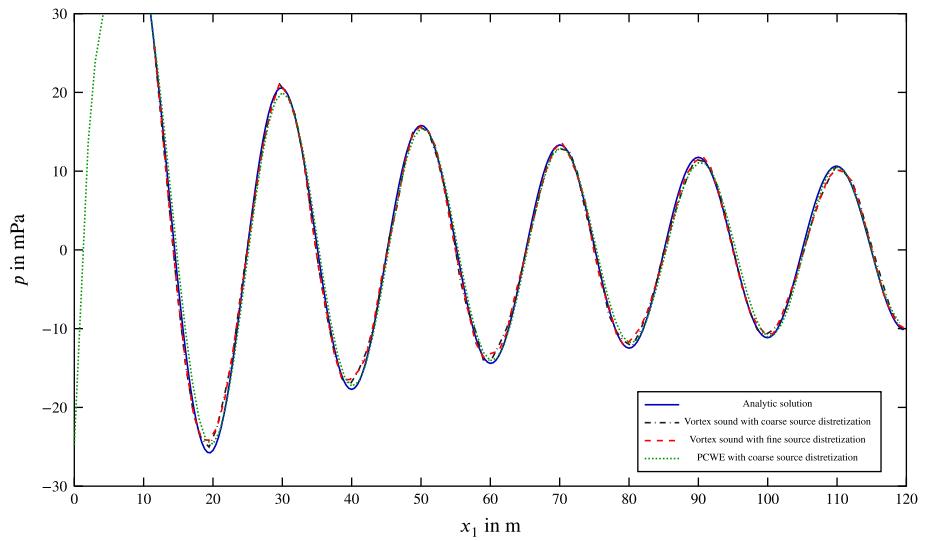


Figure 10A shows the acoustic field of co-rotating vortex pair, with the Lamb vector as source term. The acoustic field with the substantial derivative as source term is depicted in Figure 10B. In the acoustic near-field, differences occur due to the different solution quantities in the two aeroacoustic formulations (fluctuating pressure p' with the divergence of the Lamb vector as source term and acoustic pressure p^a for the PCWE). Both results have the characteristic radiation pattern of the co-rotating vortex pair.

The steepest descent of the Gaussian distribution is discretized by five linear triangular elements over 2σ . This coarse approximation of the vortical distribution shows the robustness of the RBF derivatives. A mesh refinement of the source region (see Figure 11) causes no significant increase in accuracy compared to the analytic solution. As depicted in Figure 11, one can clearly see the good accordance of numerical and analytic solution, even for the coarse grid.

5 | SUMMARY AND CONCLUSION

In this work, we have presented how to use RBFs in a local formulation for the interpolation of scattered data using a Wendland kernel in combination with a modified Shepard's method. Since hybrid aeroacoustics deals with a large number of unknowns and unfortunate data distributions, for example, in boundary layers, we focused on a parallelizable algorithm to achieve high computational efficiency in regards of algorithmic complexity. For the presented approach, an algorithmic complexity of $\mathcal{O}(N \cdot N_q)$ can be estimated, where N is the number of interpolation points and N_q the size of

the local stencils. Since for large datasets $N \gg N_q$, the scaling of the presented method behaves better in comparison to classical (global) RBF interpolation, which has complexity $\mathcal{O}(N^2)$.

The choice of the local interpolation patches is very important for the interpolation of distorted meshes, as they occur for example in boundary layers of flow simulations. Therefore, we proposed a connectivity preserving, mesh-based neighbor search. This modified patch algorithm searches for the next directly connected neighbor elements over the common global nodes. Additionally, the neighbors of the direct neighbors are also taken into account by applying a *layer strategy*. To analyze the convergence of the RBF interpolation and to quantify the choice of neighbors, analytic functions were prescribed on a unit cube with different node distributions and the L_2 -error was observed after interpolating to a different mesh with equidistant node distribution. In doing so, we found that both search methods perform well for well distributed data points. However, for large aspect ratios of boundary layer meshes, the connectivity preserving, mesh-based neighbor search significantly outperformed the kd-tree based nearest neighbor search. Furthermore, constraints and boundary conditions are incorporated to enhance the interpolation of boundary layers.

Additionally to RBF interpolation, we have presented how to compute spatial derivatives based on RBFs. The derivatives were computed in a local-global approach, using a Gaussian kernel on local point stencils. Investigations on the local-local approach, using a Wendland kernel on a local point stencil, showed reduced accuracy for skewed grids. Moreover, an optimization strategy to find an optimal value of the shape parameter scaling the Gaussian kernel was presented. Good convergence of this scaled exponential kernel was shown for the computation of the gradient of an analytic field that was prescribed on a unit cube with different node distributions.

Finally, a numerical investigation with a co-rotating vortex pair was carried out to show the proposed workflow from CFD results to aeroacoustics using RBFs. This example also included the comparison to an analytic solution, which showed excellent agreement even for relatively coarse aeroacoustic meshes.

ACKNOWLEDGEMENTS

The authors acknowledge support from the German Research Foundation (DFG) under DO 1247/10-1 (no. 391215328) and the Austrian Research Council (FWF) under no. I 3702.

ORCID

Stefan Schoder  <https://orcid.org/0000-0002-2148-6703>

Klaus Roppert  <https://orcid.org/0000-0002-2324-4361>

REFERENCES

1. Hardy RL. Multiquadric equations of topography and other irregular surfaces. *J Geophys Res.* 1971;76(8):1905-1915.
2. Hardy RL. Theory and applications of the multiquadric-biharmonic method. *Comput Math Appl.* 1990;19(8-9):163-208.
3. Rendall T, Allen C. Unified fluid-structure interpolation and mesh motion using radial basis functions. *Int J Numer Methods Eng.* 2008;74(10):1519-1559.
4. Wendland H. Hybrid methods for fluid-structure-interaction problems in aeroelasticity. *Meshfree Methods for Partial Differential Equations IV*. Berlin, Heidelberg: Springer; 2008:335-358.
5. Cordero-Gracia M, Gómez M, Ponsin J, Valero E. An interpolation tool for aerodynamic mesh deformation problems based on octree decomposition. *Aerosp Sci Technol.* 2012;23(1):93-107.
6. Cordero-Gracia M, Gómez M, Valero E. A radial basis function algorithm for simplified fluid-structure data transfer. *Int J Numer Methods Eng.* 2014;99(12):888-905.
7. Hardin JC, Hussaini MY, eds. Computational aeroacoustics for low Mach number flows. *Computational Aeroacoustics*. New York, NY: Springer-Verlag; 1992:50-68.
8. Wagner C, Hüttl T, Sagaut P, eds. *Large-Eddy Simulation for Acoustics*. Cambridge, England: Cambridge University Press; 2007.
9. Caro S, Detandt Y, Manera J, Mendonca F, Toppinga R. Validation of a new hybrid CAA strategy and application to the noise generated by a flap in a simplified HVAC duct. Paper presented at: AIAA 2009-3352; 2009.
10. Schröder T, Silkeit P, Estorff O. Influence of source term interpolation on hybrid computational aeroacoustics in finite volumes; 2016:1598-1608.
11. Piellard M, Bailly C. A hybrid method for computational aeroacoustic applied to internal flows. Paper presented at: NAGDAGA2009/369; 2009.
12. Kaltenbacher M, Escobar M, Ali I, Becker S. Numerical simulation of flow-induced noise using LES/SAS and Lighthill's acoustics analogy. *Int J Numer Methods Fluids.* 2010;63(9):1103-1122.
13. Kaltenbacher M, Hüppe A, Reppenagen A, Zenger F, Becker S. Computational aeroacoustics for rotating systems with application to an axial fan. *AIAA J.* 2017;55(11):3831-3838.

14. Ewert R, Schröder W. Acoustic perturbation equations based on flow decomposition via source filtering. *J Comput Phys.* 2003;188(2):365-398.
15. Mongillo M. Choosing basis functions and shape parameters for radial basis function methods; 2011. www.siam.org/students/siuro/vol4/S01084.pdf. Accessed May 22, 2018.
16. Madych WR. Miscellaneous error bounds for multiquadric and related interpolants. *Comput Math Appl.* 1992;24(12):121-138.
17. Fornberg B, Larsson E, Flyer N. Stable computations with Gaussian radial basis functions. *SIAM J.* 2011;33(2):869-892.
18. Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv Comput Math.* 1995;4(1):389-396.
19. Lazzaro D, Montefusco L. Radial basis functions for the multivariate interpolation of large scattered data sets. *J Comput Appl Math.* 2002;140:521-536.
20. The CGAL Project. CGAL User and Reference Manual. 4.10 ed. CGAL Editorial Board; 2017.
21. Muja M, Lowe DG. FLANN: fast library for approximate nearest neighbor.
22. Alauzet F. A parallel matrix-free conservative solution interpolation on unstructured tetrahedral meshes. *Comput Methods Appl Mech Eng.* 2016;299:116-142.
23. Casey M, Wintergerste T. *Best Practice Guidelines: ERCOFTAC Special Interest Group on “Quality and Trust in Industrial CFD”*. London, UK: ERCOFTAC; 2000.
24. Larsson E, Lehto E, Heryudono A, Fornberg B. Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions. *SIAM J Sci Comput.* 2013;35(4):A2096-A2119.
25. Roberts J. A method for calculating meshless finite difference weights. *Int J Numer Methods Eng.* 2008;74(2):321-336.
26. Lee D, Koo S. Numerical study of sound generation due to a spinning vortex pair. *AIAA J.* 1995;33(1):20-26.
27. Slimon S, Soteriou M, Davis D. Computational aeroacoustics simulations using the expansion about incompressible flow approach. *AIAA J.* 1999;37(4):409-416.
28. Hardin J, Pope D. An acoustic/viscous splitting technique for computational aeroacoustics. *Theor Comput Fluid Dyn.* 1994;6(5):323-340.
29. Munz C-D, Dumbser M, Zucchini M. The multiple pressure variables method for fluid dynamics and aeroacoustics at low Mach numbers. *Numer Methods Hyperbolic Kinet Probl.* 2003;7:335-359.
30. Müller E, Obermeier F. The spinning vortices as a source of sound. Paper presented at: AGARD CP-22; 1967.
31. Howe MS. *Theory of Vortex Sound*. Vol 33. Cambridge, UK: Cambridge University Press; 2003.
32. Kaltenbacher M. *Numerical Simulation of Mechatronic Sensors and Actuators: Finite Elements for Computational Multiphysics*. Berlin, Germany: Springer; 2015.

How to cite this article: Schoder S, Roppert K, Weitz M, Junger C, Kaltenbacher M. Aeroacoustic source term computation based on radial basis functions. *Int J Numer Methods Eng.* 2020;1-17.

<https://doi.org/10.1002/nme.6298>