

A RESCALED LOCALIZED RADIAL BASIS FUNCTION INTERPOLATION ON NON-CARTESIAN AND NONCONFORMING GRIDS*

SIMONE DEPARIS[†], DAVIDE FORTI[†], AND ALFIO QUARTERONI[‡]

Abstract. In this paper we propose a rescaled localized radial basis function (RL-RBF) interpolation method, based on the use of compactly supported radial basis functions. Starting from the classical RBF interpolation technique, we introduce a rescaling that allows for exact interpolation of constant fields between nonconforming meshes without the use of an extra polynomial term. We also present two-dimensional and three-dimensional numerical examples on arbitrary finite element meshes to show that the RL-RBF interpolation leads to accurate results, fast evaluation, and easy parallelization of the algorithm. All the computations are carried out using the open source finite element library LifeV.

Key words. compactly supported radial basis functions, interpolation, nonconforming meshes, parallel algorithms

AMS subject classifications. 41A05, 68W10

DOI. 10.1137/130947179

1. Introduction. Nowadays, radial basis functions (RBFs) have gained popularity in many different areas of scientific computing such as scattered data interpolation problems [12], neural networks [9], mesh free methods for solving PDEs [14, 16, 7], and shape parametrization [11]. It is well known that under certain mild conditions on the location of the data sites, the interpolation problem is uniquely solvable for a wide range of basis functions, including thin plate splines, multiquadratics, and Gaussian [5, 4].

We focus on interpolation problems: given a set $\Xi := \{\boldsymbol{\xi}_m\}_{m=1}^M$ of M distinct sites, at which we know the data $\{f_m = f(\boldsymbol{\xi}_m)\}_{m=1}^M$, the RBF interpolation defines a global interpolant $\Pi_f(\mathbf{x})$ in the whole domain $\Omega \subset \mathbb{R}^d$:

$$(1.1) \quad \Pi_f(\mathbf{x}) = \sum_{m=1}^M \gamma_m^f \phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r),$$

where the value of the interpolant at a generic location \mathbf{x} is obtained as a weighted sum of RBFs $\phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r)$. The coefficients $\{\gamma_m^f\}_{m=1}^M$ will depend on $\{f_m\}_{m=1}^M$. Here $\|\cdot\|$ is the euclidean norm, and r is a reference radius. In literature both locally and globally supported RBFs have been used to model $\phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r)$. Globally supported basis functions have the drawback of being computationally expensive: direct methods for computing the RBF interpolant (1.1) need $O(M^3)$ operations and each evaluation of the interpolant takes another $O(M)$ operations. This becomes

*Submitted to the journal's Methods and Algorithms for Scientific Computing section December 2, 2013; accepted for publication (in revised form) September 18, 2014; published electronically November 20, 2014. This work was supported by the Swiss National Foundation (SNF) under project 140184 and by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID s475.

<http://www.siam.org/journals/sisc/36-6/94717.html>

[†]CMCS - Modelling and Scientific Computing, MATHICSE - Mathematics Institute of Computational Science and Engineering, EPFL - Ecole Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland (simone.deparis@epfl.ch, davide.forti@epfl.ch).

[‡]MOX - Modellistica e Calcolo Scientifico, Dipartimento di matematica "F. Brioschi", Politecnico di Milano, via Bonardi 9, 20133 Milano, Italy (on leave) (alfio.quarteroni@epfl.ch).

an issue for problems with $M \approx 10^5$ (or more) interpolation points, inducing many people to consider convenient RBFs for small size problems only. In recent years much effort was made for the development of fast solvers for interpolation problems [1, 2]. Nevertheless, these methods are known to be “rather delicate to implement” [17].

Besides the high costs related to the use of globally supported basis functions, it is well known that they yield ill-conditioned discrete linear systems as the number of data points increases [5]. The ill-conditioning can be alleviated by introducing a compactly supported RBF truncated by polynomials that yield strictly positive definite matrices [15].

In this paper we propose a new consistent rescaled localized interpolation suitable for the solution of large-scale problems based on compactly supported RBFs. We start by investigating the approximation properties of the interpolant using compactly supported RBFs at a specific point \mathbf{x} : we observe that locally we need only those data points ξ_m that are close to \mathbf{x} . Relying on compactly supported RBFs, to control the number of neighbor points one needs to consider, it is usually necessary to tune the value of the support r . This approach is not suitable in those cases when the data points, within the domain, are clustered with locally different densities. Indeed, in these situations, the use of a unique value for the radius of the support does not represent a suitable choice, since small radii induce good results near the interpolation points, but large interpolation errors where they are not so densely located.

In this context we normalize (1.1) by means of the interpolant $\Pi_1(\mathbf{x})$ of the constant function $g(\mathbf{x}) \equiv \mathbf{1}$. The introduced normalization allows for the exact interpolation of constant fields between nonconforming meshes. Furthermore, we show that with respect to classical RBF interpolation based on compactly supported basis functions, the normalization above leads to a smoother interpolant which allows for better interpolation results in the whole domain even for small values of the radii of the support.

Thanks to the introduced normalization, we propose a new strategy to select locally, i.e., for each interpolation point ξ_m , the shape factor r in such a way that it takes into account the data distribution in the domain. Hence, to get the interpolant at the generic point \mathbf{x} in the domain, we rely on the selection of its neighbor points lying within a certain number of its closest circles. In this way we choose a local radius r such that the number of interpolation points in the support of the basis is roughly constant. Of course, when the data points are uniformly distributed within the domain our approach does not differ from operating an a priori selection of the radius. On the other hand, our numerical examples show that when the data points are not uniformly distributed within the domain, our approach leads to more accurate interpolation results even for different numbers of neighbor circles considered.

The paper is organized as follows: section 2 is a reminder of the RBF interpolation technique. In section 3 we introduce our method and we show that the rescaling of the RBF interpolant allows for the exact interpolation of constants. In section 4 we detail the local approach introduced to select the shape factor r , whereas in section 5 we provide a description of the interpolation algorithm and discuss its parallel performances. In section 6, with the help of numerical experiments, we highlight the efficiency and the accuracy of the proposed interpolant when used as an inter-grid interpolation operating between two nonconforming tetrahedral grids. Finally, conclusions are drawn in section 7.

2. Radial basis function interpolation. In this section we briefly review the RBF interpolation method. Given a set of M interpolation points $\Xi := \{\xi_m\}_{m=1}^M$ in

TABLE 1
Some examples of radial basis function $\phi(\|\mathbf{x}\|, r)$.

Name	$\phi(\ \mathbf{x}\ , r)$
Wendland C^2 [15]	$\left(1 - \frac{\ \mathbf{x}\ }{r}\right)_+^4 \left(1 + 4 \frac{\ \mathbf{x}\ }{r}\right)$
Thin-plate splines (TPS) [6]	$(\ \mathbf{x}\ /r)^2 \ln(\ \mathbf{x}\ /r)$
Inverted multiquadratic (IMQ) [5]	$(\ \mathbf{x}\ ^2 + r^2)^{-1/2}$
Gaussian splines [5]	$e^{-\ \mathbf{x}\ ^2/r^2}$

\mathbb{R}^d , and a corresponding set of values $\{f_m = f(\boldsymbol{\xi}_m)\}_{m=1}^M$ (for a suitable function f), the RBF interpolation defines a global interpolant $\Pi_f(\mathbf{x})$ in the whole domain $\Omega \subset \mathbb{R}^d$ as follows:

$$(2.1) \quad \Pi_f(\mathbf{x}) = \sum_{m=1}^M \gamma_m^f \phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r).$$

We denote by $\phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r)$ the RBF, which can be either globally or locally supported; $r \in \mathbb{R}$ is a shape factor which is usually selected empirically. The weights of the interpolant $\{\gamma_m^f\}_{m=1}^M$ are determined by imposing the interpolation constraints

$$(2.2) \quad \Pi_f(\boldsymbol{\xi}_m) = f_m \quad \text{for } m = 1, \dots, M.$$

When we are interested in interpolating a function $f(\mathbf{x})$ on a set $\Lambda := \{\boldsymbol{\zeta}_n\}_{n=1}^N$ of N possibly different points, (2.1) yields

$$(2.3) \quad \Pi_f(\boldsymbol{\zeta}_n) = \sum_{m=1}^M \gamma_m^f \phi(\|\boldsymbol{\zeta}_n - \boldsymbol{\xi}_m\|, r) \quad \text{for } n = 1, \dots, N.$$

In Table 1 we report a list of the most popular RBFs proposed and adopted in the literature: the Wendland basis function has a compact support, while the others are globally supported. The subscript $+$, for the Wendland basis, indicates that only positive values have to be considered (negative values are set to zero) and r is the radius of the support of the basis; for globally supported basis functions, the shape factor r controls the shape of the basis.

For TPS, a polynomial of order $k - 1$ (generally with $k = 2$) has to be added to the interpolant:

$$(2.4) \quad \Pi_f(\mathbf{x}) = \sum_{m=1}^M \gamma_m^f \phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r) + p(\mathbf{x}).$$

When considering the polynomial term, the additional degrees of freedom are obtained by requiring that the weight vector $\{\gamma_m^f\}_{m=1}^M$ is orthogonal to the nodal values of the polynomial space \mathbb{P}_d^{k-1} [5], that is,

$$(2.5) \quad \sum_{m=1}^M \gamma_m^f q(\boldsymbol{\xi}_m) = 0 \quad \forall q \in \mathbb{P}_d^{k-1}.$$

Condition (2.5) guarantees the positive definiteness of the RBF interpolation matrix Φ_{MM}^P ; see (2.6), (2.7). Using TPS basis functions, condition (2.5) implies that, when a linear polynomial term is adopted, the interpolation points cannot lie on a straight line. For multiquadratics and Gaussian bases, such a geometric condition on

the data points is not needed; the linear system defined through the interpolation conditions (2.2) and (2.5) is uniquely solvable for all $M > 1$ and N under the sole condition that the interpolation points are distinct. For both locally and globally supported RBFs the accuracy of the interpolation results highly depends on r , the value of the radius of the support [13]. Hence, depending on the problem at hand, a tuning of r allows for better accuracy.

The interpolation constraints (2.2) can be written in matrix form as follows:

$$(2.6) \quad \Phi_{MM} \gamma^f = \mathbf{f}_\xi,$$

where we denoted by $\mathbf{f}_\xi = \{f_m\}_{m=1}^M$ the data vector at the M interpolation points $\xi = \{\xi_m\}_{m=1}^M$, and by Φ_{MM} an $M \times M$ matrix whose entries are $\Phi_{MM(i,j)} = \phi(\|\xi_i - \xi_j\|, r)$. When considering the polynomial term (2.4), combining (2.2) and (2.5) yields

$$(2.7) \quad \underbrace{\begin{pmatrix} \Phi_{MM} & P_M \\ P_M^T & 0 \end{pmatrix}}_{\Phi_{MM}^P} \begin{pmatrix} \gamma^f \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\xi \\ \mathbf{0} \end{pmatrix}.$$

P_M is an $M \times 4$ matrix whose i th row reads $P_{M_i} = [1, \xi_{i,1}, \xi_{i,2}, \xi_{i,3}]$ and α is the vector containing the string of coefficients of $p(\mathbf{x})$ (e.g., for a trilinear polynomial, $p(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3$). Without the polynomial term, the values of the RBF interpolant on a set of N different points ζ_n , $n = 1, \dots, N$, reads

$$(2.8) \quad \mathbf{f}_\zeta = \Phi_{NM} \gamma^f = \Phi_{NM} \Phi_{MM}^{-1} \mathbf{f}_\xi,$$

Φ_{NM} being an $N \times M$ matrix whose entries are $\Phi_{NM(i,j)} = \phi(\|\zeta_i - \xi_j\|, r)$. When using the polynomial term the previous equation becomes

$$(2.9) \quad \mathbf{f}_\zeta = (\Phi_{NM} \ P_N) \begin{pmatrix} \Phi_{MM} & P_M \\ P_M^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}_\xi \\ \mathbf{0} \end{pmatrix}.$$

The use of globally supported basis functions leads to dense matrices Φ_{MM} and Φ_{NM} , while by using compactly supported bases, such matrices are sparse. The shape factor r has a significant impact on both the accuracy and stability of the interpolation. The larger r is, the wider is the radius of influence of each interpolation point. The optimal value of this parameter can be computed analytically only in very simple cases [4]. However, it is common practice to consider a value of r large enough that, for each data point, the support of the basis function includes at least the nearest neighbor interpolation points. Generally, for globally supported RBFs the choice of r is limited from above by the fact that the condition number of the matrix to be inverted in (2.6) and (2.7) increases exponentially with r [4]; in fact, if the basis graphs are very wide and flat, it will be hard to distinguish the contribution of every single interpolation point on the neighboring points and the matrix Φ_{MM} will be almost singular.

In Figure 1 we show the behavior of the condition number of the matrices Φ_{MM} and Φ_{MM}^P (corresponding to either disregarding or considering a linear polynomial term, respectively). We considered a unit square geometry $\Omega = [0, 1]^2$ discretized by two structured meshes made up of quadrilateral elements: the continuous lines refer to the mesh whose element size is $h = 0.1$, while the dashed ones to the finer mesh, with $h = 0.025$. In both Figure 1(a) and Figure 1(b) we notice that as the mesh size decreases the condition number of the matrices increases. The use of the polynomial term does not increase significantly the condition number of the interpolation matrix

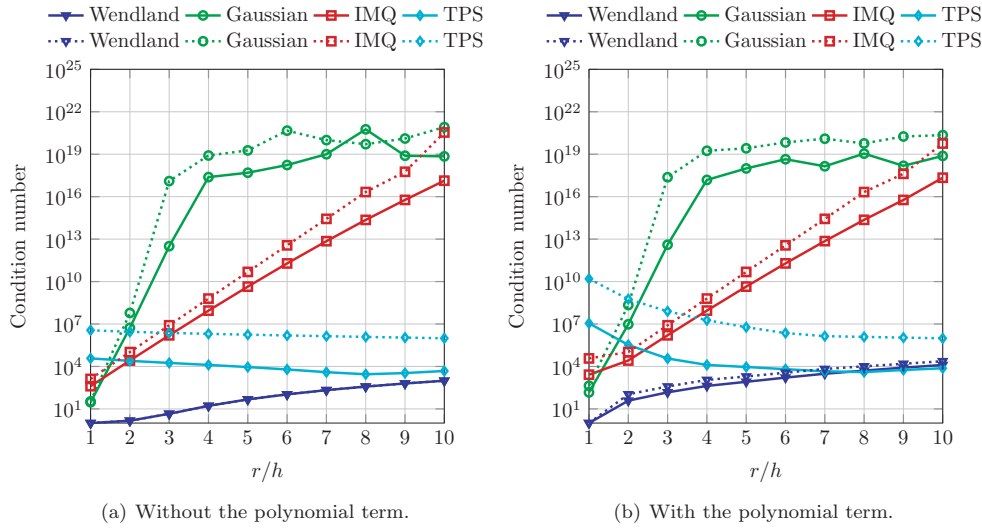


FIG. 1. Condition number of the matrices Φ_{MM} and Φ_{MM}^P without (left-hand side) and with (right-hand side) a linear polynomial term on a unit square geometry for two different mesh sizes ($h = 0.1$ continuous line and $h = 0.025$ dashed line).

when using IMQ basis functions. The Gaussian basis leads to a condition number of Φ_{MM} and Φ_{MM}^P which grows exponentially when r increases. The condition number related to TPS basis functions is almost constant without the use of the polynomial term, otherwise it gets smaller for larger shape factors r . Finally, we remark that the compactly supported Wendland basis functions lead to interpolation matrices whose condition number on the one hand slightly depend on the mesh size, on the other hand it is much smaller with respect to the other bases.

3. Rescaling the RBF interpolant. Our approach is based on the use of compactly supported Wendland RBFs without considering the polynomial term. Compactly supported RBFs lead to sparse interpolation matrices: the radius of the support r allows for controlling the sparsity of both Φ_{MM} and Φ_{NM} . In particular, as r increases, the number of nonzero entries of the interpolation matrices increases too. When dealing with large scale interpolation problems, sparse interpolation matrices help reduce memory requirements, CPU time, and, moreover, they increase scalability.

The use of a Wendland basis leads to an interpolation problem (2.6) that is uniquely solvable Φ_{MM} being a positive definite matrix [15]. Moreover, compactly supported RBFs allow for high accuracy of the interpolation under the condition that a sufficiently large radius of the support is considered.

In Figure 2 we show the results obtained by using a Wendland basis to interpolate the function $f(x) = x^3 + 2x + 2$ defined at a set Ξ of equidistributed points on $[0, 1]$ (whose distance h_A is 0.1); we interpolate a set Λ made of equidistributed points whose distance is $h_B = 0.05$. Good accuracy of the interpolation results is achieved within the whole domain using large r ; smaller values of the radius of the support lead to good approximations only near the interpolation points. Although the use of compactly supported RBFs is primarily motivated by the interest in dealing with sparse matrices, the need for accuracy leads to the use of larger radii of the support, with the backside effect of increasing the number of nonzero entries in the interpolation matrices.

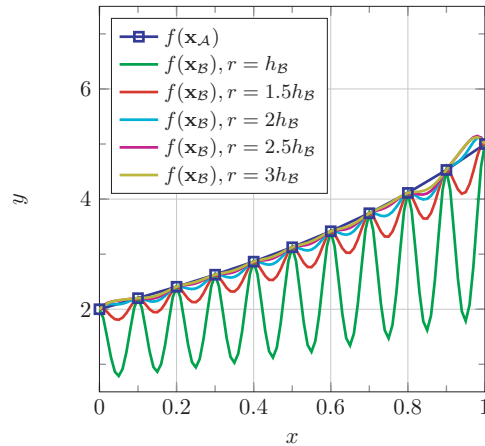


FIG. 2. One-dimensional (1D) example: comparison of the results achieved by the Wendland basis functions for different values of the radius of the support (symbol explanation in the text).

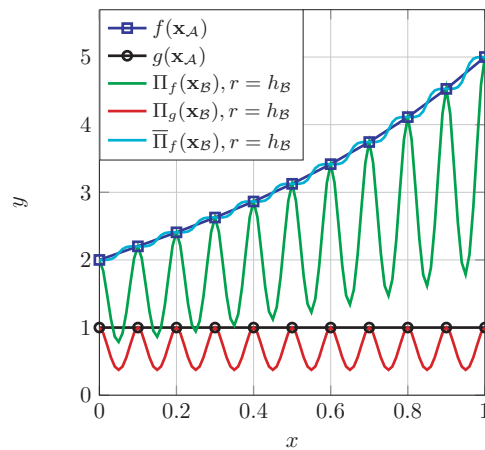


FIG. 3. 1D example: comparison of the results achieved by RBF ($\Pi_f(\mathbf{x})$) and by the rescaled RBF ($\bar{\Pi}_f(\mathbf{x})$) interpolations.

Introducing a rescaling of (2.1) allows for accurate interpolation results even by using a relatively small radius. The idea relies on rescaling (2.1) by the interpolant $\Pi_1(\mathbf{x})$ of the constant function $g(\mathbf{x}) \equiv 1$ that takes a value equal to one at each interpolation point:

$$(3.1) \quad \bar{\Pi}_f(\mathbf{x}) = \frac{\Pi_f(\mathbf{x})}{\Pi_1(\mathbf{x})} = \frac{\sum_{m=1}^M \gamma_m^f \phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r)}{\sum_{l=1}^M \gamma_l^g \phi(\|\mathbf{x} - \boldsymbol{\xi}_l\|, r)}.$$

In Figure 3 we compare Π_f with the new interpolant $\bar{\Pi}_f$. While $\Pi_1(\mathbf{x})$ oscillates at the same frequency as $\Pi_f(\mathbf{x})$ (and both are exact at the interpolation points), the proposed interpolant $\bar{\Pi}_f$ features much smaller oscillations and is more accurate,

even by using small radii of the support. Note that $\bar{\Pi}_f(\mathbf{x})$ is still interpolatory, indeed $\bar{\Pi}_f(\boldsymbol{\xi}_m) = f(\boldsymbol{\xi}_m) \forall \boldsymbol{\xi}_m \in \Xi$, since at the interpolation points $\Pi_1(\boldsymbol{\xi}_m) = 1$.

PROPOSITION 3.1. *If f is constant, then $\bar{\Pi}_f(\mathbf{x}) = f(\mathbf{x}) \forall \mathbf{x} \in \mathbb{R}^n$ such that $\Pi_1(\mathbf{x}) \neq 0$.*

Proof. Consider an arbitrary set of interpolation points $\Xi := \{\boldsymbol{\xi}_m\}_{m=1}^M$ and evaluate the interpolant $\bar{\Pi}_f$ of the function $f = C$ at an arbitrary point \mathbf{x} . We first impose the interpolation constraints, which in the matrix form read

$$(3.2) \quad \Phi_{MM} \boldsymbol{\gamma}^f = \mathbf{f}_\xi,$$

$$(3.3) \quad \Phi_{MM} \boldsymbol{\gamma}^g = \mathbf{1}_\xi.$$

Since $f(\boldsymbol{\xi}_m) = C = C g(\boldsymbol{\xi}_m)$, we have that $\mathbf{f}_\xi = C \mathbf{g}_\xi = C \mathbf{1}_\xi$ and $\boldsymbol{\gamma}^f = C \boldsymbol{\gamma}^g$, where $\mathbf{1}_\xi$ is a vector of length M with all entries equal to one. Then, recalling (3.1),

$$(3.4) \quad \bar{\Pi}_f(\mathbf{x}) = \frac{\Pi_f(\mathbf{x})}{\Pi_1(\mathbf{x})} = \frac{\sum_{m=1}^M C \gamma_m^g \phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r)}{\sum_{l=1}^M \gamma_l^g \phi(\|\mathbf{x} - \boldsymbol{\xi}_l\|, r)} = C. \quad \square$$

4. Local choice of the radius of the support. The choice of the radius of the support r is crucial: depending on the problem at hand, a tuning of r is necessary to get precise results. This has been shown for the IMQs and Gaussian RBFs (see, e.g., [8, 13]). For compactly supported radial basis functions, the value of r is generally chosen to guarantee that, on the one hand, for each interpolation point there are enough neighboring points under the support of the basis, and, on the other hand, that points being far away have no influence (see [3]). For compactly supported basis functions, the use of a large value of r allows for good interpolation, although it reduces the sparsity of the interpolation matrix; conversely, a small value of r leads to a more sparse matrix at the expense of worse accuracy.

In addition to the rescaling, we propose an adaptive strategy to select the radius of the support: we consider a value of r which is not unique but that can change locally for each interpolation point $\boldsymbol{\xi}_m$. In fact, the main idea is to take into account the way the data points are distributed within the domain such that, for each of them, the number of interpolation points in the support of the basis is about constant.

To introduce the adaptive selection of the radius of the support, we observe that for compactly supported RBFs, the region that is influenced by a data point is represented in \mathbb{R}^3 as a sphere of radius r centered in each interpolation point that has to include, at least, its neighboring points. Therefore, Ξ being the set of M distinct data sites $\{\boldsymbol{\xi}_m\}_{m=1}^M$, we identify for each of them, by means of the mesh connectivity structure, a set of K_m^c neighboring points $\Theta_m^c \subset \Xi$ with $\Theta_m^c := \{\boldsymbol{\xi}_{m,k}\}_{k=1}^{K_m^c}$, that are connected through at most c links to $\boldsymbol{\xi}_m$ ($c = 1$ in Figure 4(a)). Following this approach, instead of choosing a priori r as in classic RBF interpolation, we select the number of neighboring points within the support of the basis thanks to c . Once we fix c , for each interpolation point $\boldsymbol{\xi}_m$ we define a local measure of the radius of the support r_m^h as follows:

$$(4.1) \quad r_m^h = \max_{\boldsymbol{\xi}_{m,k} \in \Theta_m^c} \|\boldsymbol{\xi}_m - \boldsymbol{\xi}_{m,k}\|.$$

In Algorithm 1 we summarize the computation of the local radii r_m^h . The value of the radius of the support r_m^h therefore changes at each interpolation point (as shown

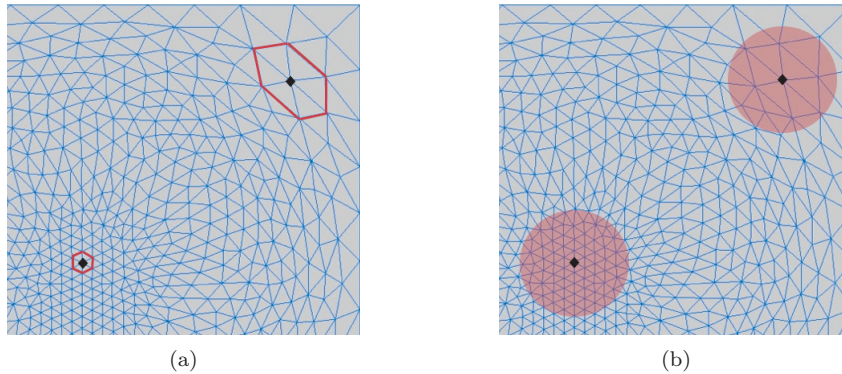


FIG. 4. Choice of neighbors by links ($c = 1$ in this example) (a) and by means of a user-defined radius (b).

in Figure 4(a)) and it has to be evaluated for the construction of the interpolation operators Φ_{MM} and Φ_{NM} .

ALGORITHM 1. COMPUTATION OF r_m^h FOR $m = 1, \dots, M$.

Require: $c \geq 1$

for $m = 1, \dots, M$ **do**

- Identify by the mesh connectivity structure a set $\Theta_m^c := \{\xi_{m,k}\}_{k=1}^{K_m^c} \subset \Xi$ of neighboring points of ξ_m , that are connected through at most c links to ξ_m (see Figure 4(a), $c = 1$).
- Compute $r_m^h = \max_{\xi_{m,k} \in \Theta_m^c} \|\xi_m - \xi_{m,k}\|$.

end for

Combining the rescaling algorithm with the adaptive selection of the radius yields the rescaled localized radial basis function (RL-RBF) interpolant

$$(4.2) \quad \bar{\Pi}_f(\mathbf{x}) = \frac{\Pi_f(\mathbf{x})}{\Pi_1(\mathbf{x})} = \frac{\sum_{m=1}^M \gamma_m^f \phi(\|\mathbf{x} - \xi_m\|, r_m^h)}{\sum_{l=1}^M \gamma_l^g \phi(\|\mathbf{x} - \xi_l\|, r_l^h)}.$$

With respect to (3.4), we remark that due to the introduced normalization, the solution of the problem is obtained dividing the following vectors element by element:

$$(4.3) \quad \hat{\mathbf{f}}_\zeta = \Phi_{NM} \Phi_{MM}^{-1} \mathbf{f}_\zeta,$$

$$(4.4) \quad \mathbf{g}_\zeta = \Phi_{NM} \Phi_{MM}^{-1} \mathbf{1}_\zeta,$$

where in (4.4) we indicated by \mathbf{g}_ζ a vector of dimension N with the interpolated values of $\mathbf{1}_\zeta$. Then, to compute

$$(4.5) \quad f_{\zeta_i} = \frac{\hat{f}_{\zeta_i}}{g_{\zeta_i}} \quad \text{for } i = 1, \dots, N,$$

two properties need to be a priori verified: (i) the matrix Φ_{MM} is invertible and (ii) $g_{\zeta_i} \neq 0$ for $i = 1, \dots, N$. In the numerical tests performed, both conditions have always been met and we have never experienced situations wherein either Φ_{MM} was singular or $g_{\zeta_i} = 0$ for some $i \in \{1, \dots, N\}$. In section 6.3, we study the condition

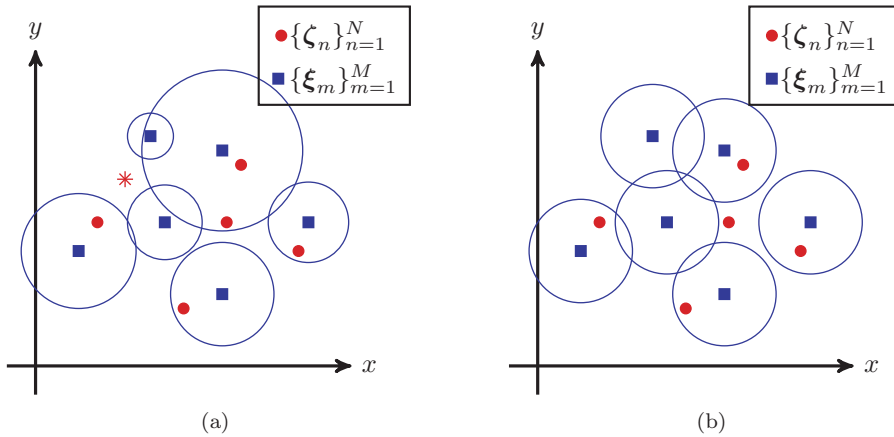


FIG. 5. Visualization of the support (blue circles) of the basis centered at each point ξ_m when a variable (a) or a unique (b) value of the radius is used.

number of the interpolation matrix Φ_{MM} associated with the proposed interpolant and with classical RBF interpolation. Although RL-RBF is based on the Wendland basis, they still differ in the way the number of points in the support is chosen. We observe that the proposed approach to compute the local radius of the support does not affect significantly the behavior of the condition number with respect to classical RBF interpolation based on the Wendland basis; see Figure 14.

Note that a necessary but not sufficient condition that allows for the fulfillment of (ii) is that the rectangular matrix Φ_{NM} has no rows with all zero entries. Such a condition is equivalent to ensuring that each ζ_n lies in the support of at least one basis function centered at the interpolation points. In Figure 5 we illustrate two different situations: where condition (ii) is fulfilled, Figure 5(a), and where it is not, Figure 5(b). As is shown in Figure 5(b) the use of a fixed value of r does not always guarantee the fulfillment of (ii). In contrast, the use of a local radius can allow each ζ_n to lie in the support of at least one basis function centered at the interpolation points. It is not possible to define the interpolant at the point represented by a star in Figure 5(a) since Π_1 is equal to zero.

5. Algorithm and parallel implementation of RL-RBF. In this section we describe a suitable parallel algorithm that implements the RL-RBF interpolation. With this aim, we consider the computational domain Ω_h decomposed into an arbitrary number N_D of subdomains. We recall that the use of a local radius of the support r_m^h requires, for each interpolation point ξ_m , the identification of the neighboring points that are connected through at most c links to ξ_m . For this reason we consider our subdomains to overlap with c layers of overlap (as shown in Figure 6). This property guarantees that for each interpolation point belonging to the i th subdomain, its neighbors (within c circles) belong to the i th subdomain as well.

Once Ω_h has been decomposed, the parallel assembly of the interpolation matrices Φ_{MM} and Φ_{NM} is performed. Then, to solve the interpolation problem, we need to impose the interpolation constraints

$$(5.1) \quad \Phi_{MM}\gamma^f = \mathbf{f}_\xi,$$

$$(5.2) \quad \Phi_{MM}\gamma^g = \mathbf{g}_\xi$$

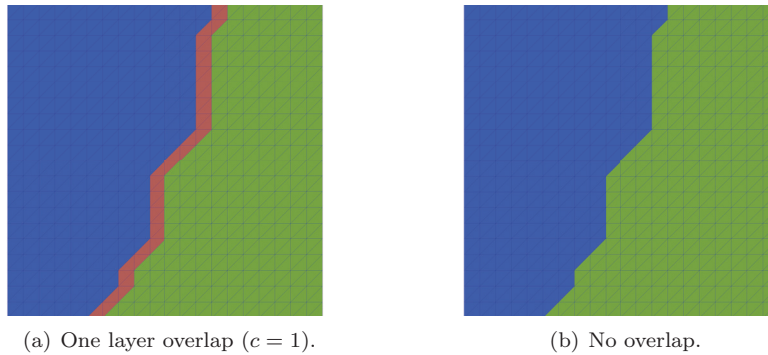


FIG. 6. Partition of Ω_h into two subdomains with one layer overlap (a) and without overlap (b).

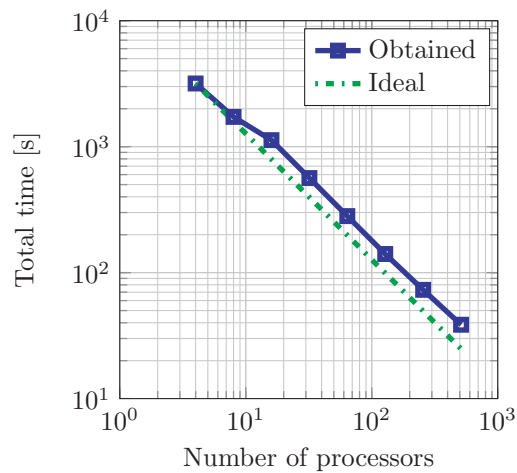


FIG. 7. Computational time for different numbers of processors used.

for the function to be interpolated and for the constant function to be equal to one, respectively. We observe that due to the rescaling, the RL-RBF strategy involves the solution of two linear systems instead of only one. Finally, thanks to the evaluation of the weights γ^f and γ^g , the solution of the interpolation problem is computed by

$$(5.3) \quad \mathbf{f}_{\zeta_n} = \frac{(\Phi_{NM}\gamma^f)_n}{(\Phi_{NM}\gamma^g)_n} \quad \text{for } n = 1, \dots, N.$$

To assess the parallel performance of the RL-RBF implementation, a scalability test has been carried out by evaluating the interpolation process on different numbers of processors. In this test case we have considered an interpolation problem whose dimensions are $M = 1\,002\,001$ (number of interpolation points) and $N = 1\,442\,401$ (number of evaluation points). Figure 7 illustrates the results obtained: the computational times reported refer to the sum of the time spent to assemble the interpolation matrices Φ_{MM} and Φ_{NM} , to impose the interpolation constraints (5.1) and (5.2), and to evaluate the solution (5.3).

6. Numerical examples. In this section we introduce three examples to compare the introduced rescaled localized strategy to the RBF interpolation (using both

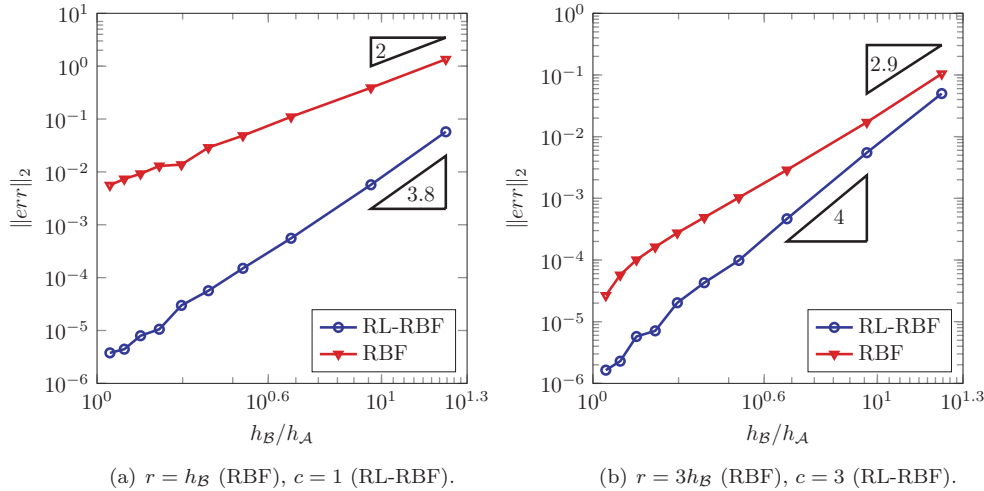


FIG. 8. Comparison between RL-RBF and RBF interpolations considering different values of c (for RL-RBF) and of r (for RBF).

locally and globally supported basis functions for the latter) in terms of accuracy of the achieved results and of the computational costs.

In the first preliminary tests we show the convergence of the RL-RBF technique solving a two dimensional interpolation problems using non-conforming meshes for the unit square that are either structured and unstructured. In the second example we consider a two-dimensional (2D) domain with unstructured mesh, where the data points are not uniformly distributed within the domain. Finally, we show the results obtained using the RL-RBF technique to solve a three-dimensional (3D) problem involving interpolation between two nonconforming meshes that have been obtained by starting from two slightly different geometries.

6.1. Example on structured grids. In this example we compare the rescaled localized interpolation with the RBF strategy (using, for the latter, compactly supported Wendland basis functions). In this way the two strategies differ by the rescaling and by the choice of the radius of the support.

We consider a structured mesh Ω_h^A of the unit square $\Omega = [0, 1]^2$ and a set of nonconforming meshes Ω_h^B of the same square. We suppose the data to be known at the mesh vertices of Ω_h^B . In detail, the known values at the interpolation points are obtained by evaluating the function $f(x, y) = 1 + x + y$. The goal is to interpolate the data at the finer mesh Ω_h^A from a set of coarser meshes Ω_h^B . For the simulations we fixed the mesh size of Ω_h^A to $h_A = 10^{-3}$, and we considered different values for h_B within the range $h_B \in [1.1 \cdot 10^{-3}, 1.68 \cdot 10^{-2}]$. Figure 8 shows the results obtained, where the computed error is defined as follows:

$$(6.1) \quad \|err\|_2 = \sqrt{\sum_{n=1}^N \frac{|f(\zeta_n) - \mathbf{f}_n|^2}{|f(\zeta_n)|^2}}.$$

We denote by $f(\zeta_n)$ the evaluation of the function $f(x, y)$ at the nodes of Ω_h^B , while by \mathbf{f}_n we denote the solutions obtained by RL-RBF and RBF interpolations at the same nodes.

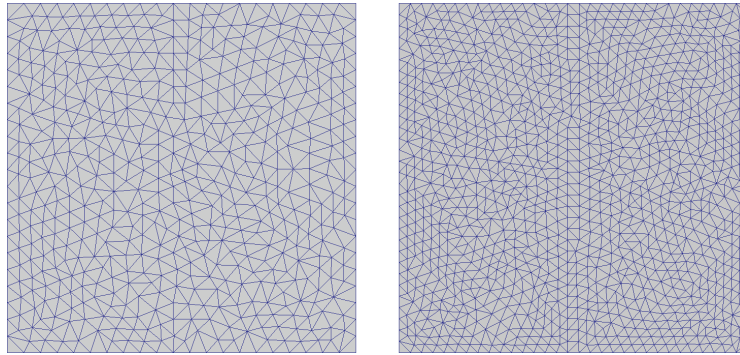


FIG. 9. Visualization of the two nonconforming unstructured meshes $\Omega_{h,2}^{\mathcal{B}}$ (left-hand side) and $\Omega_{h,2}^{\mathcal{A}}$ (right-hand side) used for $k = 2$.

As shown in both Figure 8(a) and 8(b), the use of the rescaled localized strategy leads to more accurate interpolation errors. In particular we observe that the results obtained by RL-RBF for $c = 1$ are better even if compared to the one of the RBF for the larger radius r considered ($r = 3h_{\mathcal{B}}$). Moreover, we report, with respect to the mesh size ratio $h_{\mathcal{B}}/h_{\mathcal{A}}$, the rates of convergence for both strategies: the RL-RBF interpolation leads to better convergence rates than RBF even when considering, for the RBF technique, larger radii of the support.

6.2. Example on unstructured grids. In this example we still consider the unit square $\Omega = [0, 1]^2$ and two families of unstructured meshes $\Omega_{h,k}^{\mathcal{A}}$ and $\Omega_{h,k}^{\mathcal{B}}$ for $k = 1, \dots, 8$. We suppose that the data points are the vertices of the mesh $\Omega_{h,k}^{\mathcal{B}}$, whereas the evaluation ones are the vertices of $\Omega_{h,k}^{\mathcal{A}}$. The mesh sizes used for $\Omega_{h,k}^{\mathcal{A}}$ and $\Omega_{h,k}^{\mathcal{B}}$ are $h_{\mathcal{A},k} = (10(k+1))^{-1}$ and $h_{\mathcal{B},k} = (10k)^{-1}$, respectively. The data values are obtained evaluating the function $f(x, y) = \sin(2\pi x) \cos(3\pi y) + e^{xy}$. We interpolate the data at the finer meshes $\Omega_{h,k}^{\mathcal{A}}$ from the set of coarser meshes $\Omega_{h,k}^{\mathcal{B}}$. In Figure 9 we illustrate the meshes used for $k = 2$.

The results obtained are illustrated in Figure 10: we report the interpolation error defined in (6.1) and the condition number of the interpolation operator when considering $c = 1$ and $c = 2$. On the one hand, we notice that the rate of convergence obtained, i.e., $p \approx 2.2$, with respect to the results illustrated in Figure 8, is different: the effect is due to the fact that now the number of points in the support of the basis function changes from one point to the other and it is not constant anymore as it was for structured meshes. On the other hand, in agreement with the results obtained for structured meshes, we observe that convergence rates obtained for different values of c are roughly the same although the results gained for higher values of c are more accurate.

6.3. Two-dimensional test with nonuniform distribution of data points.

In this example we test the robustness of the proposed strategy on a 2D problem involving interpolation between nonconforming meshes where the data points are not uniformly distributed within the domain. In this test we compare the RL-RBF to RBF interpolation using for the latter both globally and locally supported basis functions. We suppose that the interpolation points are the nodes of the mesh $\Omega_h^{\mathcal{A}}$ of $\Omega = [0, 1]^2$: in Figure 11(a) we can observe the nonuniform distribution of the interpolation points in the domain. The interpolatory function is still $f(x, y) = \sin(2\pi x) \cos(3\pi y) + e^{xy}$.

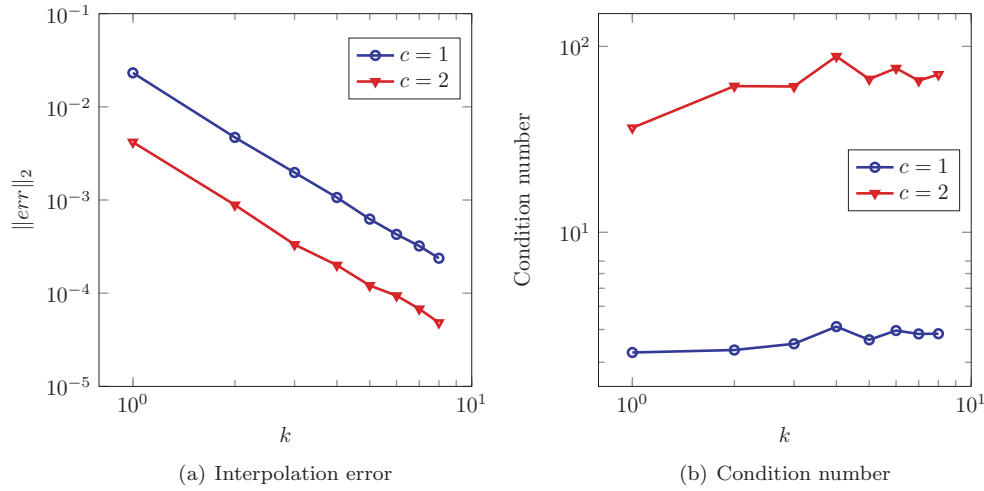


FIG. 10. Error and condition number of RL-RBF (using $c = 1, 2$) interpolation between the 2D unstructured meshes $\Omega_{h,k}^B$ and $\Omega_{h,k}^A$ for $k = 1, \dots, 8$.

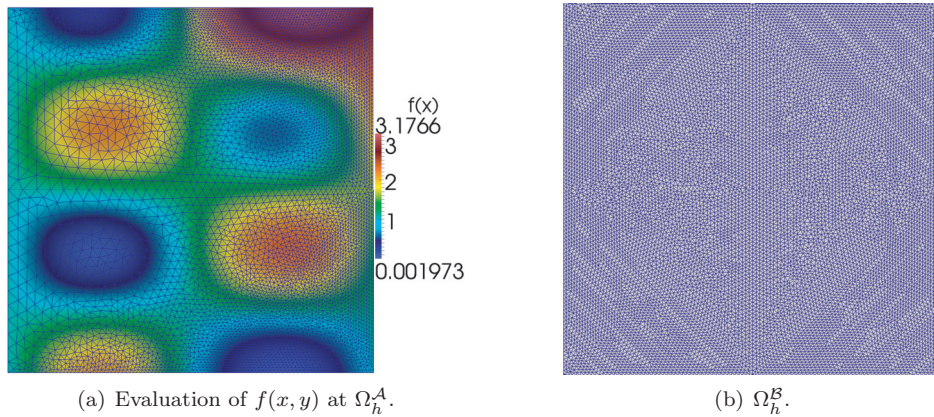


FIG. 11. Visualization of the two nonconforming meshes used.

We compute the values of the RL-RBF interpolant at the finer mesh Ω_h^B whose vertices are uniformly distributed within the domain; see Figure 11(b). With such an irregular distribution of the data points within the domain, the choice of a proper value of r represents a delicate and critical aspect for RBF interpolation. As a matter of fact, using a compactly supported basis, the value of r has to be chosen such that in the region where the data points have a coarser distribution, at least the nearest neighbors of each interpolation point lie within the support of the basis. On the other hand, once we select a value of r , it would be too large in the regions of Ω_h^A characterized by a high data density (as schematically illustrated in Figure 4(b)).

Figures 12(a) and 12(b) report the results obtained using both the RL-RBF and classical RBF interpolation strategies. Accurate results are achieved by the RL-RBF technique considering one circle to compute the local values of the radius of the support. The results of RBF interpolation with compactly supported basis functions are in line with previous works [10], since only for sufficiently large values of r are accurate results achieved.

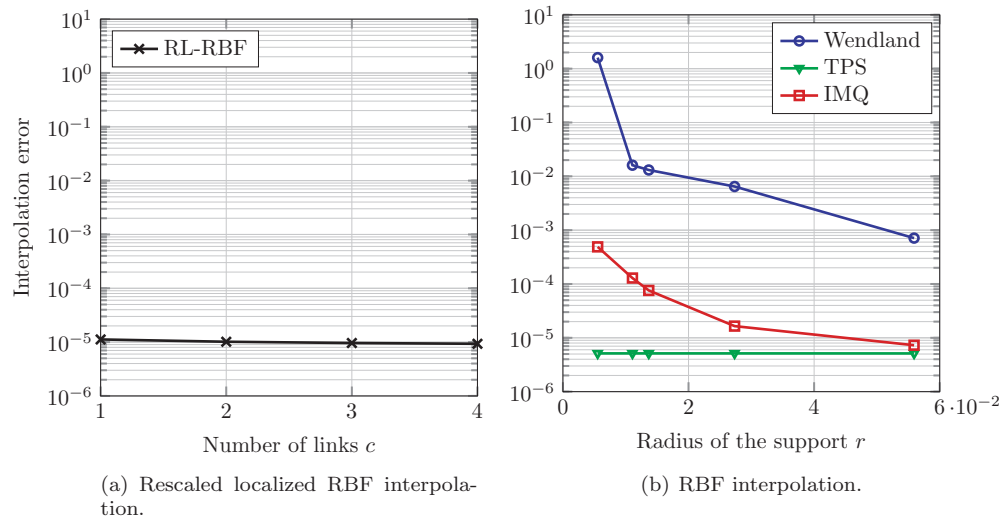


FIG. 12. Interpolation error for the function $f(x, y) = \sin(2\pi x) \cos(3\pi y) + e^{x,y}$ on a set of interpolation points not uniformly distributed within $\Omega = [0, 1]^2$.

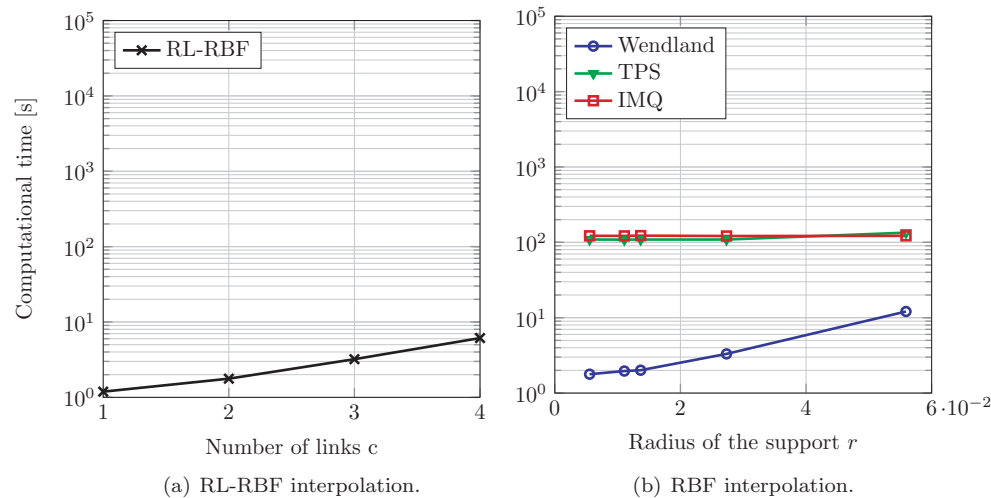


FIG. 13. Comparison of the computational costs between the proposed rescaled localized approach and the classical RBF interpolation using both compactly and globally supported basis functions.

The use of globally supported basis functions for RBF leads to more accurate results with respect to the ones obtained by the Wendland basis: TPS drives to precise results that slightly depend on r , while the IMQs requires a large value of r to get accurate results. In terms of accuracy, both the RL-RBF and the RBF with globally supported basis functions provide satisfactory results. Nevertheless, by the comparison of the computational costs shown in Figure 13, we see the RL-RBF approach leads to a much faster solution of the interpolation problem.

In Figures 14(a) and 14(b) we show the behavior of the condition number of the matrix Φ_{MM} for RL-RBF and RBF, respectively. We observe that the condition number related to TPS basis functions is almost constant with r , whereas for the IMQ

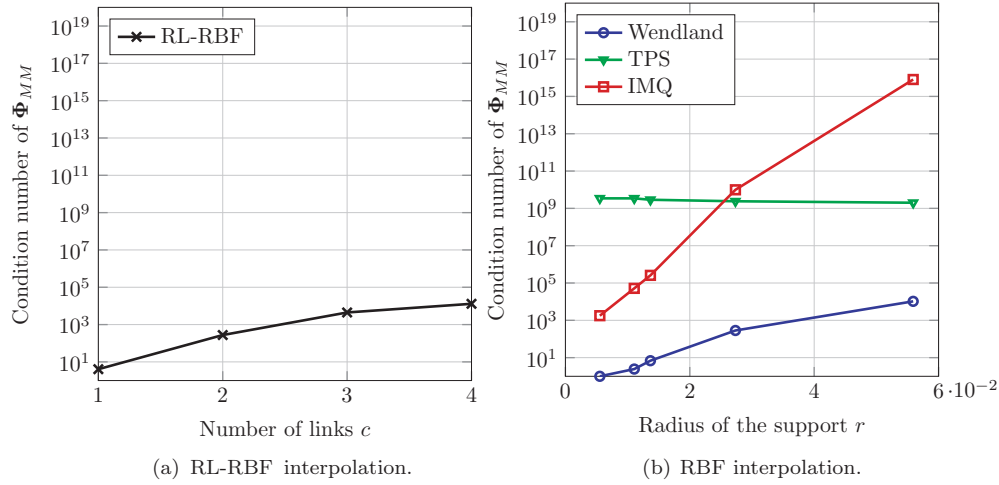


FIG. 14. Comparison of the condition numbers between the proposed rescaled localized approach and the classical RBF interpolation using both compactly and globally supported basis functions.

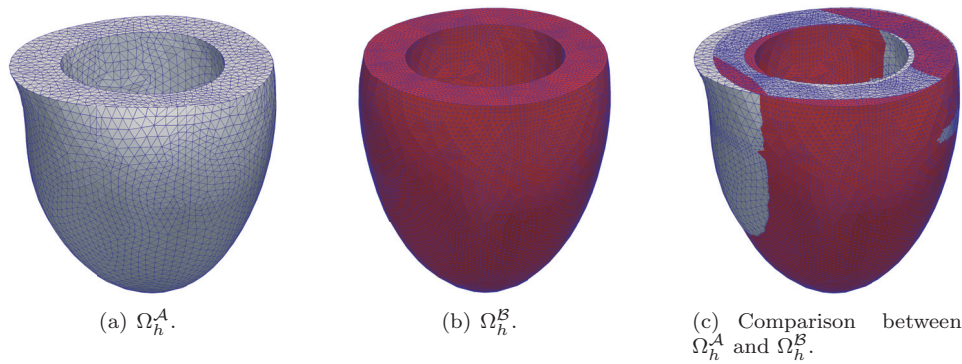


FIG. 15. Visualization and comparison of the meshes considered.

the condition number increases exponentially. The condition numbers obtained for RL-RBF interpolation and for RBF based on compactly supported Wendland basis functions are comparable: although RL-RBF is based on the Wendland basis, they still differ in the way the number of points in the support are chosen. Hence, we observe that the proposed approach to compute the local radius of the support does not influence significantly the condition number of Φ_{MM} with respect to classical RBF based on Wendland basis functions.

In conclusion the RL-RBF strategy leads to interpolation results of the same order of accuracy as TPS with a computational effort that is considerably lower.

6.4. A three-dimensional example on nonconforming geometries. In this example we consider a 3D interpolation problem which involves the data transfer between two nonconforming meshes generated from two slightly different geometries reported in Figure 15. The aim is to test the robustness of the RL-RBF strategy on nonconforming 3D geometries. In detail, at the $M = 11\,390$ nodes of the mesh Ω_h^A (Figure 15(a)) the data are defined by the evaluation of the func-

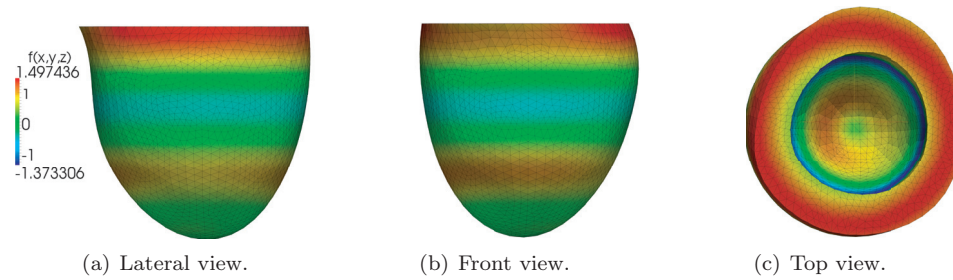


FIG. 16. 3D example: imposed data.

TABLE 2

Results of the 3D interpolation between nonconforming meshes: T_4 has been obtained on an Intel(R) Xeon(R) CPU E5-2630L (2.00 GHz) using 4 MPI processes, while T_{256} on 8 nodes of the cluster Bellatrix at the EPF Lausanne, each characterized by 2 Sandy Bridge processors (2.20 GHz) with 8 cores each.

Number of links (c)	$\ err\ _2$	T_4 [s]	T_{256} [s]
1	$2.4982 \cdot 10^{-4}$	31	0.998
2	$2.4651 \cdot 10^{-4}$	33	1.012
3	$2.4598 \cdot 10^{-4}$	38	1.031
4	$2.4539 \cdot 10^{-4}$	40	1.098

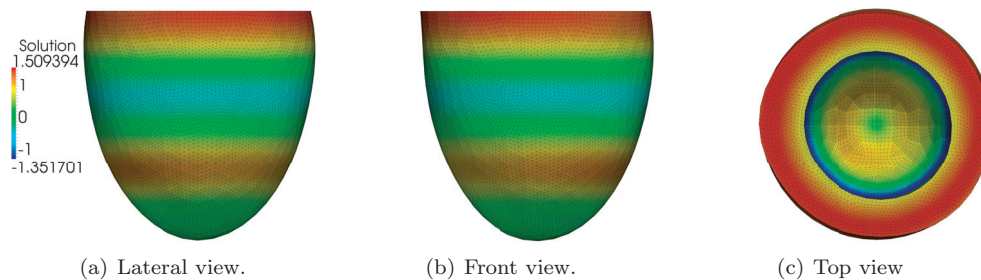


FIG. 17. Results on a 3D example.

tion $f(x, y, z) = \sin(z) + \sin(\sqrt{x^2 + y^2}) \cos(\sqrt{x^2 + y^2})$ (see Figure 16), while the $N = 56\,789$ points where the interpolant is evaluated are the nodes of the mesh Ω_h^B (Figure 15(b)): we are thus interpolating f from a coarse to a finer mesh.

Table 2 and Figure 17 show the results obtained for different values of c used to define the support radius: we note that the proposed strategy allows for accurate interpolation results even when considering $c = 1$.

7. Conclusions. In this work we have introduced an RL-RBF strategy for the solution of large-scale interpolation problems between nonconforming meshes. After reviewing the classical RBF (both locally and globally supported), we have analyzed the effect of considering an additional linear polynomial term for RBF on the condition number of the interpolation matrix, for different types of basis functions.

Then, we have focused on the use of the compactly supported Wendland basis functions and have introduced a rescaling to the RBF interpolation that allows for the exact interpolation of constant fields between nonconforming meshes. We have

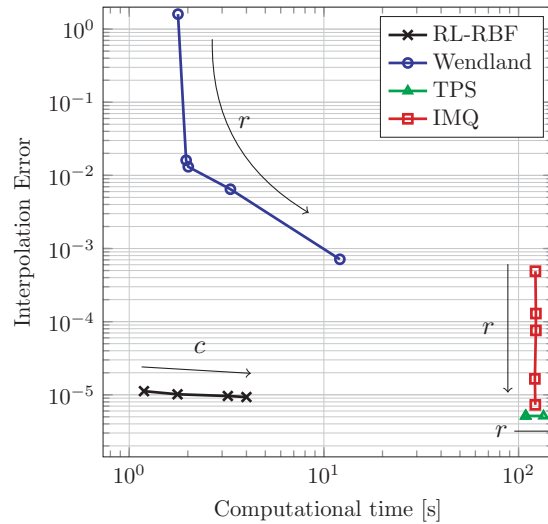


FIG. 18. Comparison of the RL-RBF and RBF interpolation strategies on a 2D example with nonuniform distribution of data points. The direction of the arrows indicates an increasing value of r and of c . For the RL-RBF we considered, $c = 1, \dots, 4$.

shown that the introduced rescaling allows for a smoother interpolant with respect to classical RBF ones. Then, thanks to the introduced rescaling, a new approach to compute the local radius of the support has been proposed.

We have compared the classical RBF with the new RL-RBF interpolation in terms of accuracy and sensitivity with respect to the number of links c of the RL-RBF, and of the radius r of the RBF, respectively. The results are summarized in Figure 18 for the numerical example proposed in section 6.3. Furthermore, thanks to its excellent parallel performances, we highlighted that the interpolation technique proposed can be successfully used to solve large-scale interpolation problems.

These properties have also been shown on large-scale problems which may possibly involve nonconforming geometries.

Acknowledgment. We gratefully acknowledge the CSCS for providing us the CPU resources for our simulations.

REFERENCES

- [1] R. K. BEATSON, J. B. CHERRIE, AND C. T. MOUAT, *Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration*, Appl. Math. Comput., 11 (1999), pp. 253–270.
- [2] R. K. BEATSON AND G. N. NEWSAM, *Fast evaluation of radial basis functions: I*, Comput. Math. Appl., 24 (1992), pp. 7–19.
- [3] A. BECKERT AND H. WENDLAND, *Multivariate interpolation for fluid-structure-interaction problems using radial basis functions*, Aerosp. Sci. Technol., 5 (2001), pp. 125–134.
- [4] M. D. BUHMANN, *Radial basis functions*, Acta Numer., 9 (2000), pp. 1–38.
- [5] M. D. BUHMANN, *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, New York, 2003.
- [6] J. DUCHON, *Splines minimizing rotation invariant semi-norms in Sobolev spaces*, in Constructive Theory of Functions of Several Variables, W. Schempp and K. Zeller, eds., Lecture Notes in Math. 571, Springer, Berlin, 1977, pp. 85–100.
- [7] C. FRANKE AND R. SCHABACK, *Solving partial differential equations by collocation using radial basis functions*, Appl. Math. Comput., 93 (1988), pp. 73–82.

- [8] R. FRANKE, *Scattered data interpolation: Tests of some methods*, Math. Comp., 38 (1982), pp. 181–200.
- [9] J. A. S. FREEMAN AND D. SAAD, *Learning and generalisation in radial basis function networks*, Neural Comput., 7 (1995), pp. 1000–1020.
- [10] M. LOMBARDI, N. PAROLINI, AND A. QUARTERONI, *Radial basis functions in FSI problems: Interpolation and mesh motion*, Internat. J. Numer. Methods Fluids, 256 (2013), pp. 117–131.
- [11] A. MANZONI, A. QUARTERONI, AND G. ROZZA, *Model reduction techniques for fast blood flow simulation in parametrized geometries*, Internat. J. Numer. Methods Biomed. Engrg., 28 (2012), pp. 604–625.
- [12] B. S. MORSE, T. S. YOO, D. T. CHEN, P. RHEINGANS, AND K. R. SUBRAMANIAN, *Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions*, in ACM SIGGRAPH 2005 Courses, Los Angeles, CA, 2005.
- [13] S. RIPPA, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comput. Math., 11 (1999), pp. 193–210.
- [14] J. G. WANG AND G. R. LIU, *A point interpolation meshless method based on radial basis functions*, Internat. J. Numer. Methods Eng., 54 (2002), pp. 1623–1648.
- [15] H. WENDLAND, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Adv. Comput. Math., 4 (1995), pp. 389–396.
- [16] H. WENDLAND, *Meshless Galerkin methods using radial basis functions*, Math. Comput., 68 (1999), pp. 1521–1531.
- [17] H. WENDLAND, *Fast evaluation of radial basis functions: Methods based on partition of unity*, in Approximation Theory X: Wavelets, Splines, and Applications, 2002, Vanderbilt University Press, Nashville, TN, pp. 473–483.