ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

MATHICSE
Mathematics Institute of Computational Science and Engineering
School of Basic Sciences - Section of Mathematics

MATHICSE Technical Report
Nr. 37.2013
November 2013

A rescaled localized radial basis functions interpolation on non-cartesian and non-conforming grids

S. Deparis, D. Forti, A. Quarteroni

http://mathicse.epfl.ch

Phone: +41 21 69 37648

Address:
EPFL - SB - MATHICSE (Bâtiment MA)
Station 8 - CH-1015 - Lausanne - Switzerland

Fax: +41 21 69 32545

A rescaled localized radial basis functions interpolation on non-cartesian and non-conforming grids

Simone Deparis[1], Davide Forti[1], and Alfio Quarteroni[1,2]

[1]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

[2]MOX - Modellistica e Calcolo Scientifico, Dipartimento di matematica "F. Brioschi",
Politecnico di Milano, via Bonardi 9, 20133 Milano, Italy

Abstract

In this paper we propose a Rescaled Localized Radial Basis Functions (RL-RBF) interpolation method, based on the use of compactly supported radial basis functions. Starting from classic RBF interpolation technique, we introduce a rescaling that recovers the partition of unity combined with a new algorithm to select the support of the basis. The proposed rescaling generates a set of basis functions that allows for exact interpolation of constant fields between non-conforming meshes without the use of an extra polynomial term. We also present 2D and 3D 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.

Keywords: Compactly supported radial basis functions, interpolation, non-conforming meshes, partition of unity, parallel algorithms.

1 Introduction

Nowadays radial basis functions 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, multi-quadratics 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 $\mathbf{f}_\xi = f(\boldsymbol{\xi})$, the radial basis function interpolation defines a global interpolant $\Pi_f(\mathbf{x})$ in the whole domain $\Omega \subset \mathbb{R}^d$:

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

where the value of the interpolant at a generic location $\mathbf{x}$ is obtained as a weighted sum of radial basis functions $\phi(\| \mathbf{x}-\boldsymbol{\xi}_m \|, r)$. Here $\|\cdot\|$ is the euclidean norm, and r is a reference radius. In literature both locally and globally supported radial basis functions have been used to model $\phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r)$. Globally supported basis functions are in general more precise, but they have the drawback of being computational expensive: direct methods for computing the RBF interpolant (1) need $O(M^3)$ operations and each evaluation of the interpolant takes another $O(M)$ operations. This is too much for problems with $M \approx 10^5$ (or more) interpolation points, for this reason many people consider radial basis functions to be useful for small problems only. Much work aimed at the development of fast solvers [1, 17] for the interpolation problems, and for the set up of fast evaluation of the interpolant has been done in recent years. Nevertheless, these methods [2] are rather difficult to implement.

Besides the high costs related to the use of globally supported basis functions, it is well known that they yield to ill-conditioned discrete linear system as the number of data points increases [5]. The ill-conditioning of the RBF method has been reduced by introducing a compactly supported radial basis function truncated by polynomials [15] that are strictly positive definite.

In this paper we propose a new consistent rescaled localized interpolation suitable for the solution of large-scale problems based on compactly supported radial basis functions. To introduce the new approach we start by investigating the approximation properties of the interpolant at a specific point $\mathbf{x}$: we observe that locally we need only those data points $\boldsymbol{\xi}_m$ that are close to $\mathbf{x}$. Relying on compactly supported radial basis functions, to control the number of neighbor points 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 local 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 on the one hand good results near the interpolation points, on the other hand large interpolation errors where they are not so densely located.

In this context we normalize (1) by means of the interpolant $\Pi_g(\mathbf{x})$ of the constant function $g(\mathbf{x}) \equiv \mathbf{1}$. The introduced normalization allows for the exact interpolation of constant fields between non-conforming meshes and it generates a set of basis functions that satisfy the partition of unity property. Furthermore, we show that with respect to RBF interpolation based on compactly supported basis functions, the normalization above leads to a smoother interpolant which allows

for better interpolation results within 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 $\boldsymbol{\xi}_m$, the shape factor r such that it takes into account the way the data is distributed 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 the introduced approach is similar to an a priori selection of the radius. On the other hand, at the basis of the numerical examples proposed, we show that when dealing with data points not uniformly distributed within the domain, our approach leads to more accurate interpolation results even for different number of neighbor circles considered.

The paper has been organized as follows: Section 2 is a reminder of the radial basis function interpolation technique. In Section 3 we introduce our method and we show that the rescaling of the RBF interpolant allows for the recovery of a set of basis functions satisfying the partition of unity. 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 towards the intergrid interpolation between non-conforming tetrahedral grids. Finally, conclusions are drawn in Section 7.

2 Radial basis function interpolation

In this Section we briefly review the radial basis function interpolation method. Given a set of M interpolation points $\Xi := \{\boldsymbol{\xi}_m\}_{m=1}^{M}$ in $\mathbb{R}^d$, and a corresponding set of values $\mathbf{f}_{\boldsymbol{\xi}_m} = f(\boldsymbol{\xi}_m)$ (for a suitable function f), the radial basis function interpolation defines a global interpolant $\Pi_f(\mathbf{x})$ in the whole domain $\Omega \subset \mathbb{R}^d$ as follows:

$$\Pi_f(\mathbf{x}) = \sum_{m=1}^{M} \gamma_m^f \, \phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r). \tag{2}$$

We denoted by $\phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r)$ the radial basis function, 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 $\boldsymbol{\gamma}^f$ are determined by imposing the interpolation constraints:

$$\Pi_f(\boldsymbol{\xi}_m) = f_{\boldsymbol{\xi}_m}, \quad \text{for } \; m = 1, \ldots, M. \tag{3}$$

Once the weights of the interpolant $\boldsymbol{\gamma}^f$ have been computed, the interpolated value $\Pi_f(\mathbf{x})$ at a point $\mathbf{x} \in \mathbb{R}^d$ is computed by (2). When we are interested in

Name	$\phi(\|\mathbf{x}\|, r)$
Beckert & Wendland (B&W) [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 Multi-Quadratic (IMQ) [5]	$(\|\mathbf{x}\|^2+r^2)^{-1/2}$
Gaussian splines [5]	$e^{-\|\mathbf{x}\|^2/r^2}$

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

interpolating a function $f(\mathbf{x})$ on a set $\Lambda := \{\boldsymbol{\zeta}_n\}_{n=1}^N$ of N possibly different points, the interpolation is written as:

$$\Pi_f(\boldsymbol{\zeta}_n) = \sum_{m=1}^{M} \gamma_m^f \, \phi(\| \, \boldsymbol{\zeta}_n - \boldsymbol{\xi}_m \, \|, \, r) \quad \text{for} \;\; n = 1, \ldots, N. \tag{4}$$

In Table 1 we report a list of the most popular radial basis functions proposed and adopted in literature: the Beckert & Wendland basis function has a compact support, while the others are globally supported. Within the Beckert & Wendland basis the subscript $+$ 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 thin-plate splines, a polynomial of order $k-1$ (generally with $k=2$), with $p(\mathbf{x}) \in \mathbb{P}_M^{k-1}(\mathbf{x})$, has to be added to the interpolant:

$$\Pi_f(\mathbf{x}) = \sum_{m=1}^{M} \gamma_m^f \, \phi(\| \, \mathbf{x} - \boldsymbol{\xi}_m \, \|, \, r) + p(\mathbf{x}). \tag{5}$$

When considering the polynomial term, the additional degrees of freedom are obtained by requiring that the weight vector $\boldsymbol{\gamma}^f$ is orthogonal to the polynomial space $\mathbb{P}_M^{k-1}(\mathbf{x})$ [5]:

$$\sum_{m=1}^{M} \gamma_m^f q(\boldsymbol{\xi}_m) = 0, \qquad \forall q \in \mathbb{P}_M^{k-1}(\mathbf{x}). \tag{6}$$

Condition (6) guarantees the positive definiteness of the RBF interpolation problem. Using thin-plate splines basis functions, condition (6) implies that, when a linear polynomial term is adopted, the interpolation points must not lie on a straight line. For multi-quadratics and gaussian basis, such a geometric condition on the data points is not needed: the linear system defined through the interpolation conditions (3) and (6) is uniquely solvable for all $M > 1$ and N under the only condition that the interpolation point are distinct. For both locally and

globally supported radial basis functions the accuracy of the interpolation results highly depend on the value of the radius of the support r [13]. Hence, depending on the problem at hand, a tuning of r allows for better accuracy.

The interpolation constraints (3) can be written in matrix form as follows

$$\Phi_{MM}\,\boldsymbol{\gamma}^f = \mathbf{f}_\xi, \tag{7}$$

where we denoted by $\boldsymbol{\gamma}^f$ the vector of weights of the interpolant, by $\mathbf{f}_\xi$ the data vector at the M interpolation points $\boldsymbol{\xi}$, and by Φ_{MM} an $M \times M$ matrix whose entries are $\Phi_{MM_{(i,j)}} = \phi(\parallel \boldsymbol{\xi}_i - \boldsymbol{\xi}_j \parallel, r)$. When considering the polynomial term (5), combining (3) and (6) yields:

$$\underbrace{\begin{pmatrix} \Phi_{MM} & P_M \\ P_M^T & 0 \end{pmatrix}}_{\Phi^P_{MM}} \begin{pmatrix} \boldsymbol{\gamma}^f \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\xi \\ \mathbf{0} \end{pmatrix}. \tag{8}$$

P_M is a $M \times 4$ matrix whose i-th row reads $P_{M_i} = [1,\, \xi_{i,1},\, \xi_{i,2},\, \xi_{i,3}\,]$ and $\boldsymbol{\alpha}$ is the vector containing the string of coefficients of $p(\boldsymbol{\xi})$, (e.g. for a linear polynomial, $p(\boldsymbol{\xi}) = \alpha_0 + \alpha_1\,\xi_1 + \alpha_2\,\xi_2 + \alpha_3\,\xi_3$). Without the polynomial term, the values of the RBF interpolant on a set of N different points $\boldsymbol{\zeta}_n$, $n = 1, \ldots, N$, reads:

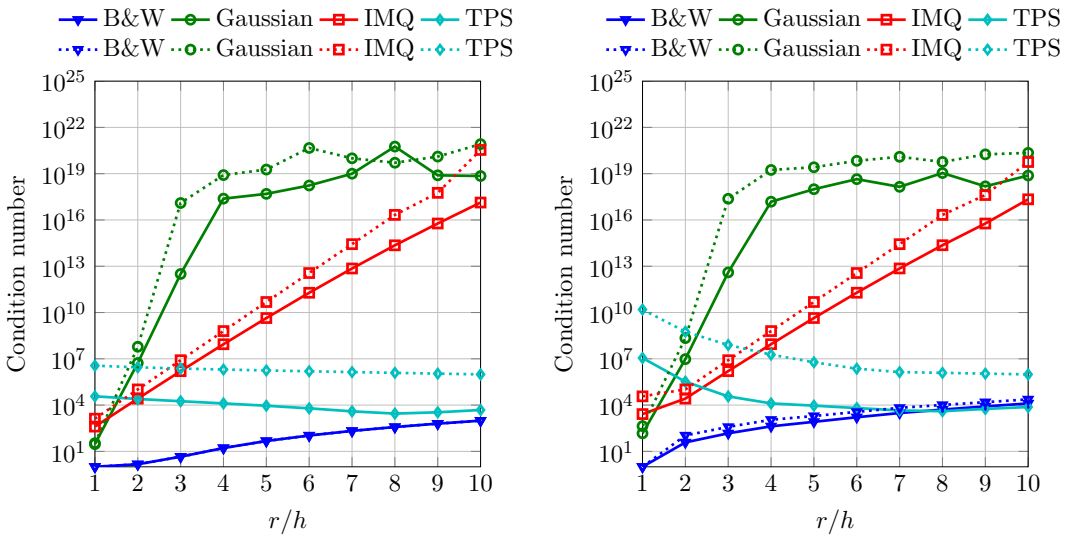
$$\mathbf{f}_\zeta = \Phi_{NM}\,\boldsymbol{\gamma}^f = \Phi_{NM}\,\Phi_{MM}^{-1}\,\mathbf{f}_\xi, \tag{9}$$

being Φ_{NM} an $N \times M$ matrix whose entries are $\Phi_{NM_{(i,j)}} = \phi(\parallel \boldsymbol{\zeta}_i - \boldsymbol{\xi}_j \parallel, r)$. When using the polynomial term the previous equation becomes:

$$\mathbf{f}_\zeta = (\Phi_{NM} \;\; P_N) \begin{pmatrix} \Phi_{MM} & P_M \\ P_M^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}_\xi \\ \mathbf{0} \end{pmatrix}. \tag{10}$$

The use of globally supported basis functions leads to dense matrices Φ_{MM} and Φ_{NM}, while by using compactly supported basis, such matrices are sparse. The shape factor r has a significant impact on the accuracy and stability of the interpolation. The larger r, the wider 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 radial basis functions the choice of r is limited from above by the fact that the condition number of the matrix to be inverted in (7) and (8) 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^P_{MM} (i.e. not-considering and considering a linear polynomial term,



(a) Without the polynomial term.

(b) With the polynomial term.

Figure 1: Condition number of the matrices Φ_{MM} and Φ^P_{MM} 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).

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 when using IMQ basis functions. Gaussian basis leads to a condition number of Φ_{MM} and Φ^P_{MM} 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 Beckert & Wendland basis functions lead to interpolation matrices whose condition number on the one hand slightly depend on the mesh size, on the other hand that is much smaller with respect to the other basis.

3 Rescaling the RBF interpolant

Our approach is based on the use of compactly supported Beckert & Wendland radial basis functions without considering the polynomial term. Compactly sup-

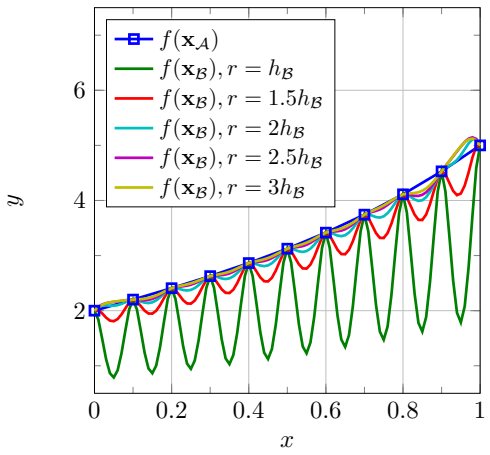


Figure 2: 1D example: comparison of the results achieved by Beckert and Wendland basis functions for different values of the radius of the support (symbol explanation in the text).

ported radial basis functions 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 we increase r, we increase the number of nonzero entries of the interpolation matrices as well. When dealing with large scale interpolation problems, sparse interpolation matrices help reducing memory requirements, CPU time and increase scalability, therefore compactly supported radial basis functions are essential.

On the one hand, the use of Beckert & Wendland basis leads to an interpolation problem (7) that is uniquely solvable being Φ_{MM} a positive definite matrix [15]. On the other hand compactly supported radial basis functions allows 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 Beckert & 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 $h_\mathcal{A}$ is 0.1); we interpolate an a set Λ made of equidistributed points whose distance $h_\mathcal{B} = 0.05$. A 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 nearby the interpolation points. Although the use of compactly supported radial basis functions is primarily motivated by the interest of 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.

Introducing a rescaling of (2) allows for accurate interpolation results even by using a radius of the support smaller then the one used for classic radial basis

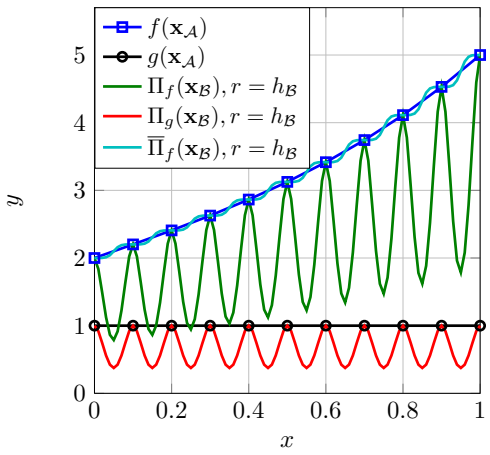


Figure 3: 1D example: comparison of the results achieved by RBF ($\Pi_f(\mathbf{x})$) and by the rescaled RBF ($\overline{\Pi}_f(\mathbf{x})$) interpolations.

interpolation. The main idea relies in rescaling (2) by the interpolant $\Pi_g(\mathbf{x})$ of the constant function $g(\mathbf{x}) = 1$ that has values equal to one at each interpolation point:

$$\overline{\Pi}_f(\mathbf{x}) = \frac{\Pi_f(\mathbf{x})}{\Pi_g(\mathbf{x})} = \frac{\displaystyle\sum_{m=1}^{M} \gamma_m^f \, \phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r)}{\displaystyle\sum_{l=1}^{M} \gamma_l^g \, \phi(\| \mathbf{x} - \boldsymbol{\xi}_l \|, r)}. \tag{11}$$

In Figure 3 we compare Π_f with the new interpolant $\overline{\Pi}_f$. Indeed, with the help of Figure 3, we show that the interpolant $\Pi_g(\mathbf{x})$ oscillates at the same frequency of $\Pi_f(\mathbf{x})$ and both are exact at the interpolation points. The proposed interpolant $\overline{\Pi}_f$ shows much smaller oscillations and it is more accurate, even by using small radii of the support. Note that $\overline{\Pi}_f(\mathbf{x})$ is still interpolatory, indeed $\overline{\Pi}_f(\boldsymbol{\xi}_m) = f(\boldsymbol{\xi}_m)$ $\forall \boldsymbol{\xi}_m \in \Xi$, since at the interpolation points $\Pi_g(\boldsymbol{\xi}_m) = 1$.

Proposition 3.1. *If f is constant, then $\overline{\Pi}_f(\mathbf{x}) = f(\mathbf{x})$.*

Proof. Consider an arbitrary set of interpolation points $\Xi := \{\boldsymbol{\xi}_m\}_{m=1}^M$ and compute the value obtained from the interpolant of $f = C$ at a point $\mathbf{x}$. We firstly impose the interpolation constraints, which in the matrix form read:

$$\Phi_{MM} \boldsymbol{\gamma}^f = \mathbf{f}_\xi = C \, \mathbf{1}_\xi, \tag{12}$$

$$\Phi_{MM} \boldsymbol{\gamma}^g = \mathbf{1}_\xi. \tag{13}$$

Then, recalling (9), the values of the interpolant at the point $\mathbf{x}$ is

$$\overline{\Pi}_f(\mathbf{x}) = C \, \frac{(\Phi_{NM} \Phi_{MM}^{-1} \mathbf{1}_\xi)}{(\Phi_{NM} \Phi_{MM}^{-1} \mathbf{1}_\xi)} = C \tag{14}$$

where we denoted by $\mathbf{1}_\xi$ a vector of length M with all entries equal to one. □

Proposition 3.2. *The introduced rescaling can be seen as a RBF interpolation whose basis functions satisfy the partition of unity.*

Proof. Interpolant (11) can be rewritten as:

$$\overline{\Pi}_f(\mathbf{x}) = \sum_{m=1}^{M} \gamma_m^f \sum_{l=1}^{M} \frac{\phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r)}{\gamma_l^g \phi(\| \mathbf{x} - \boldsymbol{\xi}_l \|, r)}. \tag{15}$$

Hence, by introducing $\tilde{\gamma}_m^f = \dfrac{\gamma_m^f}{\gamma_m^g}$ for $m = 1, \ldots, M$, we obtain:

$$\overline{\Pi}_f(\mathbf{x}) = \sum_{m=1}^{M} \tilde{\gamma}_m^f \sum_{l=1}^{M} \frac{\gamma_m^g \phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r)}{\gamma_l^g \phi(\| \mathbf{x} - \boldsymbol{\xi}_l \|, r)} = \sum_{m=1}^{M} \tilde{\gamma}_m^f \tilde{\phi}(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r). \tag{16}$$

In (16), we denoted by $\tilde{\phi}(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r)$ the new set of basis functions that satisfy the partition of unity:

$$\sum_{m=1}^{M} \tilde{\phi}(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r) = \sum_{m=1}^{M} \sum_{l=1}^{M} \frac{\gamma_m^g \phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r)}{\gamma_l^g \phi(\| \mathbf{x} - \boldsymbol{\xi}_l \|, r)} = 1. \tag{17}$$

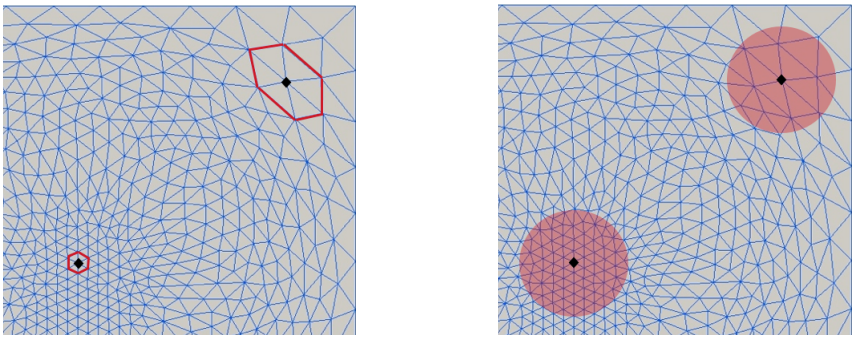
□

Therefore, with respect to classical radial basis function interpolation, we obtain a smoother interpolant that interpolates exactly any constant field ($\overline{\Pi}_C(\mathbf{x}) = C \ \forall \mathbf{x}$, where C is a constant), and allows for a better accuracy by using smaller radius of the support.

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 inverted multi-quadratics and gaussian radial basis functions (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, 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 a 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



(a) (b)

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

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 roughly constant.

To introduce the adaptive selection of the radius of the support, we observe that for compactly supported radial basis functions, 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, being Ξ the set of M distinct data sites, 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 $\Theta_m^c := \{\boldsymbol{\xi}_{m,k}\}_{k=1}^{K_m^c}$ (for $m = 1, \ldots, M$) 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 like 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:

$$r_m^h = \max_{\boldsymbol{\xi}_{m,k} \in \Theta_m^c} \| \boldsymbol{\xi}_m - \boldsymbol{\xi}_{m,k} \| . \tag{18}$$

The value of the radius of the support r_m^h therefore changes at each interpolation point (as shown in Figure 4(b)) and it has to be evaluated during the construction of the interpolation operators Φ_{MM} and Φ_{NM}.

Combining the rescaling algorithm with the adaptive selection of the radius

yields to the Rescaled Localized Radial Basis Function (RL - RBF) interpolant:

$$\overline{\Pi}_f(\mathbf{x}) = \frac{\Pi_f(\mathbf{x})}{\Pi_g(\mathbf{x})} = \frac{\sum_{m=1}^{M} \gamma_m^f \, \phi(\| \mathbf{x} - \boldsymbol{\xi}_m \|, r_m^h)}{\sum_{l=1}^{M} \gamma_l^g \, \phi(\| \mathbf{x} - \boldsymbol{\xi}_l \|, r_m^h)}. \tag{19}$$

With respect to (14), we remark that due to the introduced normalization, the solution of the problem is obtained dividing point by point the following vectors:

$$\hat{\mathbf{f}}_\zeta = \Phi_{NM} \Phi_{MM}^{-1} \mathbf{f}_\xi, \tag{20}$$

$$\mathbf{g}_\zeta = \Phi_{NM} \Phi_{MM}^{-1} \mathbf{1}_\xi, \tag{21}$$

where in (21) we indicated by $\mathbf{g}_\zeta$ a vector of dimension N with the interpolated values of $\mathbf{1}_\xi$. Therefore, to evaluate:

$$\mathbf{f}_{\zeta_i} = \frac{\hat{\mathbf{f}}_{\zeta_i}}{\mathbf{g}_{\zeta_i}} \qquad \text{for } i = 1, \ldots, N, \tag{22}$$

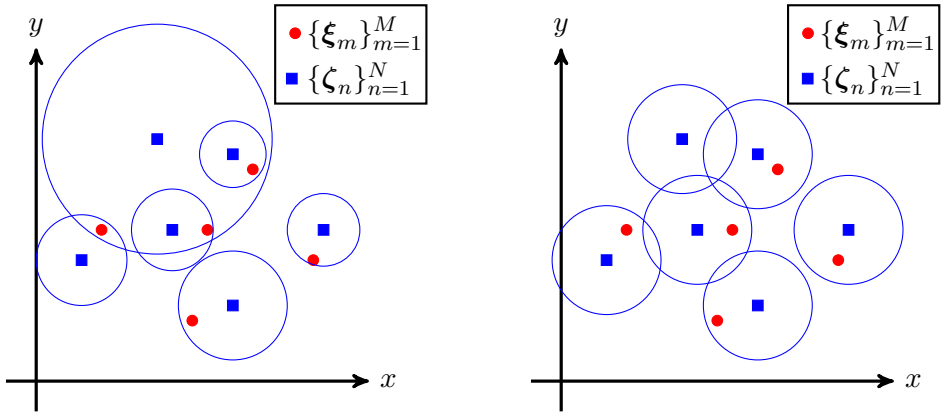
the following condition has to be satisfied:

$$\mathbf{g}_{\zeta_i} \neq 0 \qquad \text{for } i = 1, \ldots, N. \tag{23}$$

The fulfillment of (23) implies that the rectangular matrix Φ_{NM} has no rows with all zero entries. Such a condition imposes that, for each point $\boldsymbol{\zeta}_n$, r_n^h is large enough to have at least one interpolation point $\boldsymbol{\xi}_m$ in the support of the basis. In Figure 5 we illustrate two different situations where condition (23) is fulfilled, Figure 5(a), and it is not, Figure 5(b). As it is shown in Figure 5(b) the use of a fixed value of r does not guarantee always the fulfillment of (23). In contrast, the use of a local radius that includes, for each $\boldsymbol{\zeta}_n$, at least one interpolation point $\boldsymbol{\xi}_m$, satisfies condition (23).

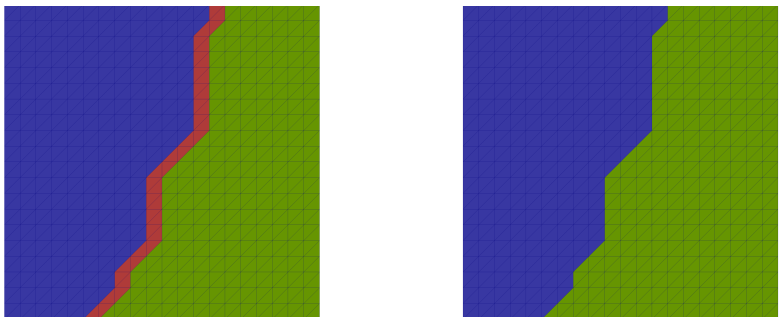
5 Algorithm and parallel implementation of RL-RBF

In this section we describe the parallelization of the rescaled localized radial basis function interpolation. We recall that the use of a local radius of the support r_h^m requires, for each interpolation point $\boldsymbol{\xi}_m$, the identification of the neighboring points that are connected through at most c links to $\boldsymbol{\xi}_m$, we consider overlap between sub-domains. For that, we consider the computational domain Ω_h decomposed in N_D sub-domains with c layers of overlap. In this way, as illustrated in Figure 6, by using c overlapping layers we guarantee that for each interpolation point belonging to the i-th sub-domain, its neighbors (within c circles) belong to the i-th sub-domain as well.



(a) (b)

Figure 5: Visualization of the support (blue circles) of the basis centered at each point $\boldsymbol{\zeta}_n$ when a variable (a) or a unique (b) value of the radius is used.



(a) One layer overlap. (b) No overlap.

Figure 6: Partition of Ω_h in two sub-domains with one layer overlap (Figure 6(a)) and without overlap (Figure 6(b)).

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:

$$\Phi_{MM}\boldsymbol{\gamma}^f = \mathbf{f}_\xi, \tag{24}$$
$$\Phi_{MM}\boldsymbol{\gamma}^1 = \mathbf{1}_\xi, \tag{25}$$

for the function to be interpolated and for the constant function 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 since we need to impose the interpolation constraints for the constant function $g(\mathbf{x}) = \mathbf{1}$ as well. Finally, thanks to the evaluation of the weights $\boldsymbol{\gamma}^f$ and $\boldsymbol{\gamma}^1$, the solution of the interpolation

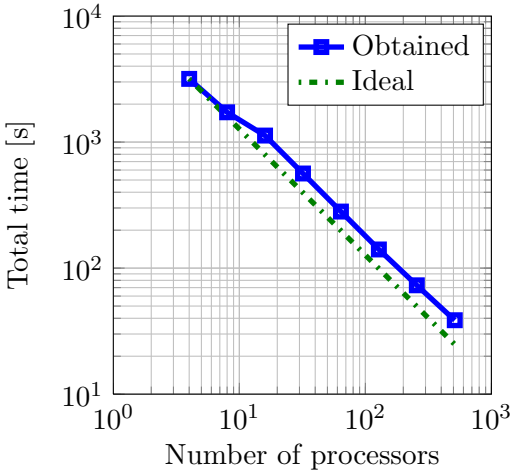


Figure 7: Computational time for a different number of used processors.

problem is computed by:

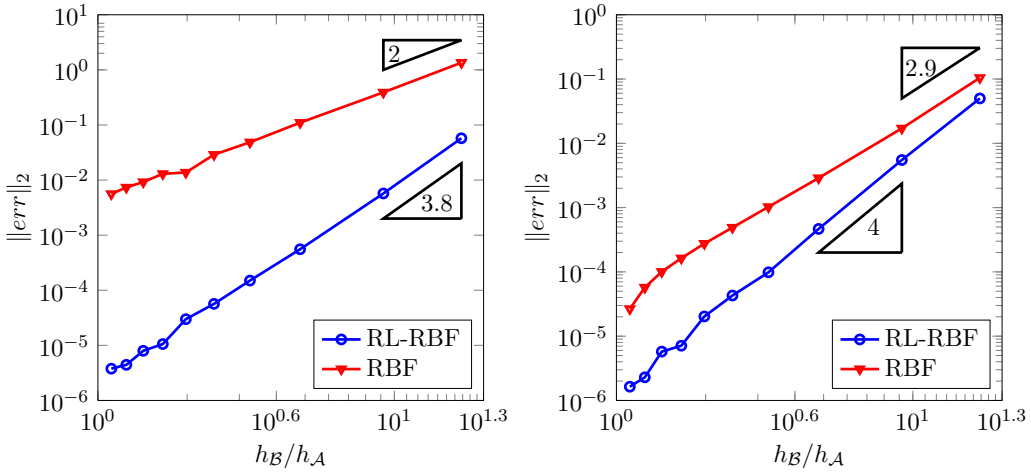
$$\mathbf{f}_{\zeta\, n} = \frac{(\Phi_{NM}\boldsymbol{\gamma}^f)_n}{(\Phi_{NM}\boldsymbol{\gamma}^1)_n} \qquad \text{for } n = 1, \ldots, N. \tag{26}$$

To assess the parallel performance of the RL-RBF implementation, a scalability test has been carried out by evaluating the interpolation process on different number 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 points where the interpolant is evaluated). 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 (24) and (25), and to evaluate the solution (26).

6 Numerical examples

In this section we introduce two examples to compare the introduced rescaled localized strategy to the radial basis function interpolation (using both 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 test we compare RL-RBF and RBF (using for both techniques the compactly supported Beckert & Wendland basis functions) against a two dimensional interpolation problem between two structured non-conforming meshes. In the second example we propose the results obtained for a two-dimensional test using unstructured meshes, where the data points are not uniformly distributed within the domain. Finally, we show the results obtained using the



(a) $r = h_{\mathcal{B}}$ (RBF), $c = 1$ (RL-RBF).

(b) $r = 3h_{\mathcal{B}}$ (RBF), $c = 3$ (RL-RBF).

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

RL-RBF technique to solve a three dimensional problem involving interpolation between two non-conforming meshes that have been obtained by starting from two slightly different geometries.

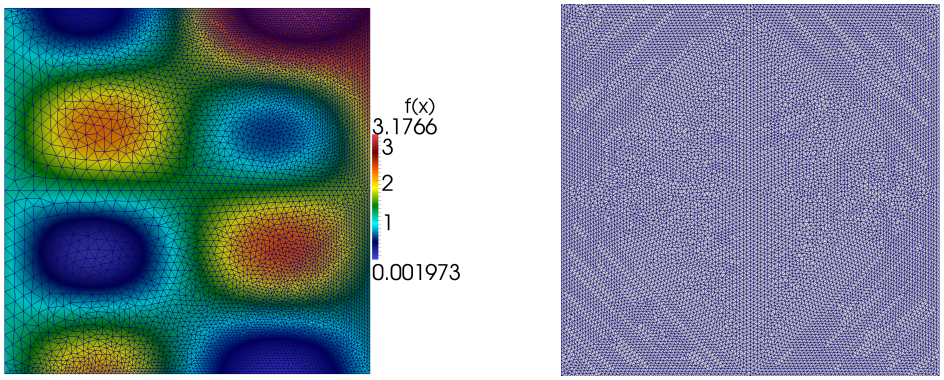
6.1 Preliminary example on structured grids

In this example we compare the rescaled localized interpolation with the radial basis function strategy (using for the latter compactly supported Beckert & 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 $\Omega_h^{\mathcal{A}}$ of a geometry that consists in a unit square $\Omega = [0,\, 1]^2$ and a set of non-conforming meshes $\Omega_h^{\mathcal{B}}$ of Ω. We suppose the data to be known at the mesh points of $\Omega_h^{\mathcal{B}}$. In detail, the known values at the interpolation points are obtained by evaluating the function $f(x, y) = 1 + x + y$. Therefore, the goal is to interpolate the data at the finer mesh $\Omega_h^{\mathcal{A}}$ from a set of coarser meshes $\Omega_h^{\mathcal{B}}$. For the simulations we fixed the mesh size of $\Omega_h^{\mathcal{A}}$ to $h_{\mathcal{A}} = 10^{-3}$, and we considered different values for $h_{\mathcal{B}}$ within the range $h_{\mathcal{B}} \in [1.1 \cdot 10^{-3}, 1.68 \cdot 10^{-2}]$. Figure 8 shows the results obtained, where the computed error is defined as follows:

$$\|err\|_2 = \sqrt{\sum_{n=1}^{N} \frac{|f(\boldsymbol{\zeta}_n) - \mathbf{f}_n|^2}{|f(\boldsymbol{\zeta}_n)|^2}}. \tag{27}$$

We denote by $f(\boldsymbol{\zeta}_n)$ the evaluation of the function $f(x, y)$ at the nodes of $\Omega_h^{\mathcal{B}}$,



(a) Evaluation of $f(x, y)$ at $\Omega_h^{\mathcal{A}}$. (b) $\Omega_h^{\mathcal{B}}$.

Figure 9: Visualization of the two non-conforming meshes used.

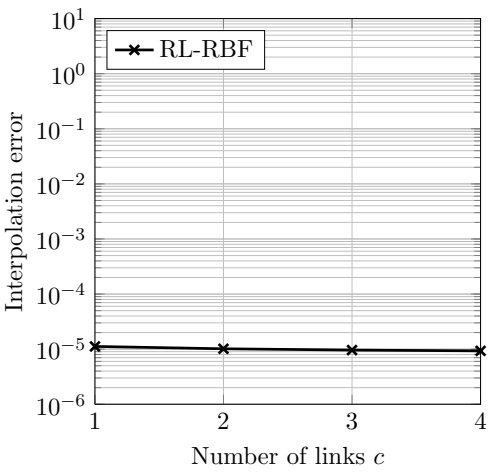
while by $\mathbf{f}_n$ we denote the solutions obtained by RL-RBF and RBF interpolations at the same nodes.

As shown in both Figure 8(a) and 8(b), the use of the rescaled localized strategy leads to more accurate results in terms of interpolation error. In particular we observe that the result obtained by RL-RBF for $c = 1$ are better even if compared to the one gained by RBF for the larger radius r considered ($r = 3h_{\mathcal{B}}$). Moreover we report, with respect to the 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.

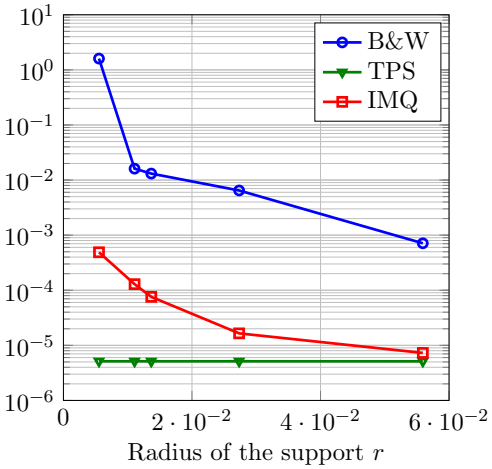
6.2 Two-Dimensional test with non uniform distribution of data points

In the second example we test the robustness of the proposed strategy for a two dimensional problem involving interpolation between non-conforming 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 $\Omega = [0, 1]^2$: thanks to Figure 9(a) we can observe the non-uniform distribution of the interpolation points in the domain. At the interpolation points the data is given by evaluating the function $f(x, y) = \sin(2\pi x)\cos(3\pi y) + e^{xy}$.

We study the behavior of both RL-RBF and RBF strategies to interpolate $f(x, y)$ at the finer mesh $\Omega_h^{\mathcal{B}}$ (whose nodes are uniformly distributed within the domain, see Figure 9(b)) from mesh $\Omega_h^{\mathcal{A}}$. In this example involving 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



(a) Rescaled localized RBF interpolation.



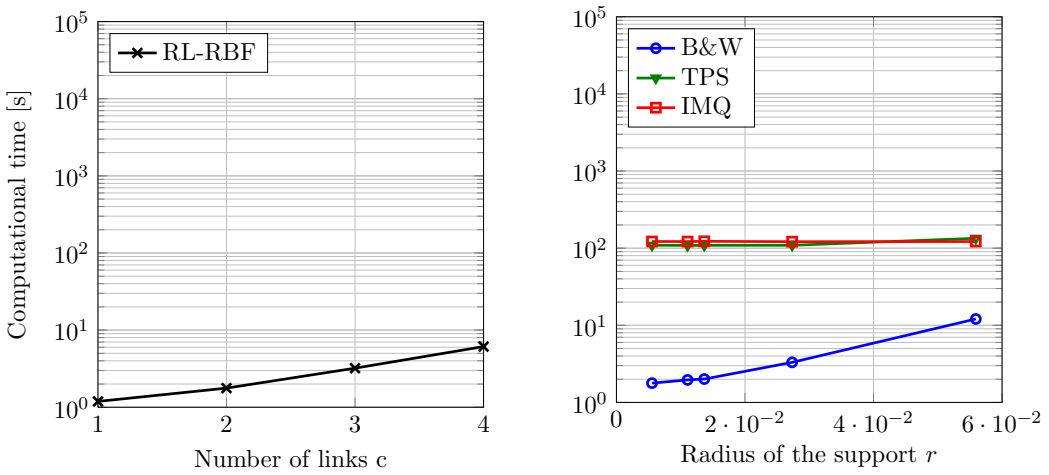
(b) RBF interpolation.

Figure 10: 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 uniformely distributed within $\Omega = [0, 1]^2$.

of fact, using 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 is 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 $\Omega_h^{\mathcal{A}}$ characterized by a high data density (as schematically illustrated in Figure 4(b)).

Figure 10(a) and 10(b) report the results obtained using the rescaled localized and the radial basis function interpolation strategies. Accurate results, in terms of interpolation error, 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 accurate results are achieved.

The use of globally supported basis functions for RBF leads to more accurate results with respect to the ones obtained by the Beckert & Wendland basis: thin-plate splines drives to precise results that slightly depend on r, while the inverted multi-quadratics 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 11, we see the RL-RBF approach leads to a much faster solution of the interpolation problem. In conclusion, we highlight that the RL-RBF strategy leads to interpolation results of the same order of accuracy of thin-plate splines with a computational effort that is considerably lower.



(a) RL-RBF interpolation.

(b) RBF interpolation.

Figure 11: Comparison of the computational costs between the rescaled localized approach proposed and the classical RBF interpolation using both compactly and globally supported basis functions.

6.3 Three-dimensional example on non-conforming geometries

In this example we consider a three-dimensional interpolation problem which involves the data transfer between two non-conforming meshes generated from two different geometries reported in Figure 12. The aim is to test the robustness of the LR-RBF strategy towards an interpolation problem involving non-conforming geometries. In detail, at the $M = 11\,390$ nodes of the mesh $\Omega_h^{\mathcal{A}}$ (Figure 12(a)) the data are defined by the evaluation of the function $f(x, y, z) = \sin(z) + \sin(\sqrt{x^2 + y^2})\cos(\sqrt{x^2 + y^2})$, while the $N = 56\,789$ points where the interpolant is evaluated are the nodes of the mesh $\Omega_h^{\mathcal{B}}$ (Figure 12(b)): therefore, we are interpolating f from a coarse to a finer mesh.

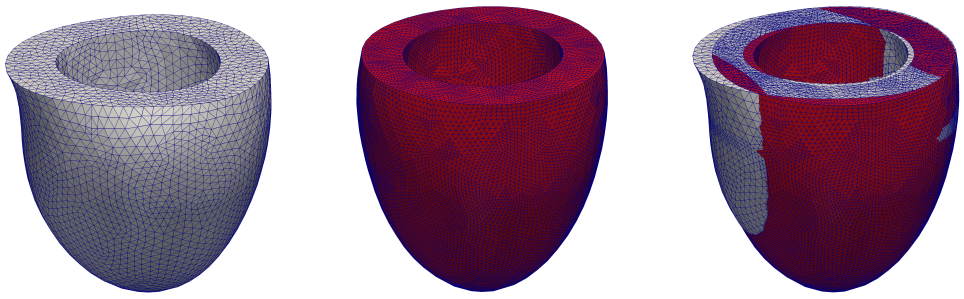
Table 2 shows the results obtained for different values of c to define the support radius: the proposed strategy allows for accurate interpolation results even when considering $c = 1$.

7 Conclusions

In this work we have introduced a rescaled localized radial basis function strategy for the solution of large-scale interpolation problems between non-conforming meshes. In the first section, a review of the classic radial basis function interpolation has been carried out, focusing on both the use of locally and globally supported basis functions. Moreover, we have analyzed the effect of considering an additional linear polynomial term for RBF on the condition number of the

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

Table 2: Results of the three-dimensional interpolation between non-conforming 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.



(a) $\Omega_h^{\mathcal{A}}$. (b) $\Omega_h^{\mathcal{B}}$. (c) Comparison between $\Omega_h^{\mathcal{A}}$ and $\Omega_h^{\mathcal{B}}$.

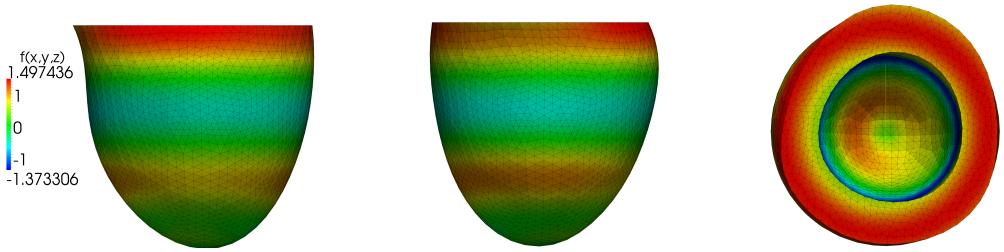
Figure 12: Visualization and comparison of the meshes considered.

interpolation matrix, for different types of basis functions.

In the second part of this paper, we have focused on the use of the compactly supported Beckert & Wendland basis functions. We have introduced a rescaling to RBF interpolation that allows for the exact interpolation of constant fields between non-conforming meshes and that recovers a set of basis functions that satisfy the partition of unity. We have shown that the introduced rescaling allows for a smoother interpolant with respect to classic RBF interpolation. Then, thanks to the introduced rescaling, a new approach to compute the local radius of the support has been proposed.

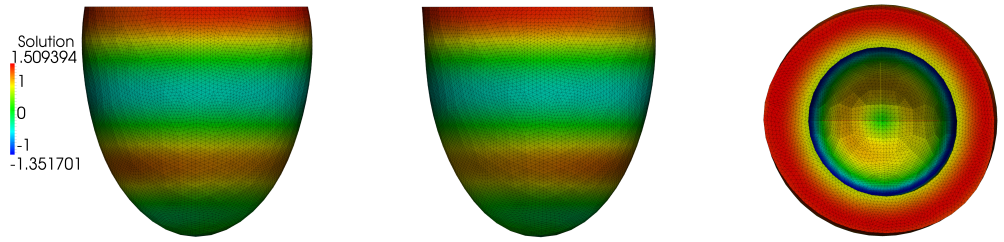
We have compared the classical RBF with the new RL-RBF interpolation in terms of accuracy and sensitivity with respect to the number of links c of the RL-RBF, and of the radius r of the RBF, respectively. The results are summarized in Figure 15, for the numerical example proposed in Section 6.2. Furthermore, by means of the parallel performances of the algorithm we highlighted that the interpolation technique proposed can be successfully used to solve large-scale interpolation problems.

We have shown the high accuracy, the efficiency and the parallel performances



(a) Lateral view. (b) Front view. (c) Top view.

Figure 13: Three-dimensional example: imposed data.



(a) Lateral view. (b) Front view. (c) Top view

Figure 14: Resuts on a three-dimensional example.

achieved by RL-RBF when dealing with large-scale problems which may also involve non-conforming geometries.

References

[1] R. K. Beatson, J. B. Cherrie, and C. T. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned gmres iteration. *Applied Mathematics and Computation*, 11:253–270, 1999.

[2] R. K. Beatson and G. N. Newsam. Fast evaluation of radial basis functions: I. *Computers & Mathematics with Applications*, 24(12):7 – 19, 1992.

[3] A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5:125–134, 2001.

[4] M. D. Buhmann. Radial basis functions. *Acta Numerica 2000*, 9:1–38, 2000.

[5] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, New York, NY, USA, 2003.

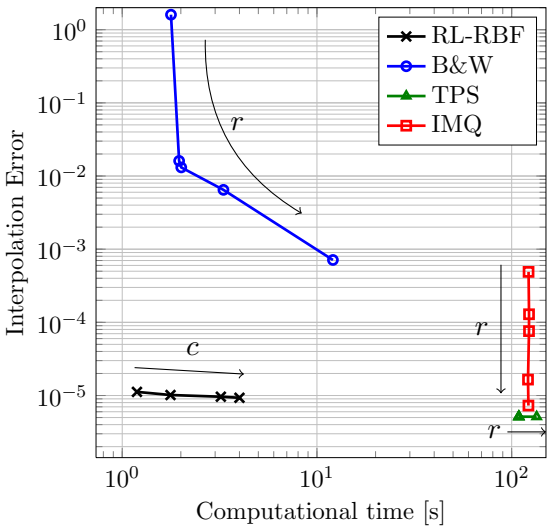


Figure 15: Comparison of the RL-RBF and RBF interpolation strategies on a 2D example with nonuniform distribution of data points. The direction of the arrows indicates an increasing value of r and of c. For the RL-RBF we considered $c = 1, \ldots, 4$.

[6] J. Duchon. Splines minimizing rotation invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer, Berlin, 1977.

[7] C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Applied Mathematics and Computation*, 93:73–82, 1998.

[8] R. Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 38:181–200, 1982.

[9] J. A. S. Freeman and D. Saad. Learning and generalisation in radial basis function networks. *Neural computation*, 7:1000–1020, 1995.

[10] M. Lombardi, N. Parolini, and A. Quarteroni. Radial basis functions in FSI problems: interpolation and mesh motion. *International Journal for Numerical Methods in Fluids*, 256:117–131, 2013.

[11] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *International Journal for Numerical Methods in Biomedical Engineering*, 28:604–625, 2012.

[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 *Proceedings of the International Conference on Shape Modeling & Applications*, SMI '01, Washington, DC, USA, 2001. IEEE Computer Society.

[13] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2-3):193–210, 1999.

[14] J. G. Wang and G. R. Liu. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, 54:1623–1648, 2002.

[15] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

[16] H. Wendland. Meshless galerkin methods using radial basis functions. *Mathematics of Computation*, 68:1521–1531, 1999.

[17] H. Wendland. Fast evaluation of radial basis functions: Methods based on partition of unity. In *Approximation Theory X: Wavelets, Splines, and Applications*, pages 473–483. Vanderbilt University Press, 2002.

Recent publications :

MATHEMATICS INSTITUTE OF COMPUTATIONAL SCIENCE AND ENGINEERING
Section of Mathematics
Ecole Polytechnique Fédérale
CH-1015 Lausanne

25.2013 D. DEVAUD, A. MANZONI, G. ROZZA:
A combination between the reduced basis method and the ANOVA expansion: on the computation of sensitivity indices

26.2013 M. SHAO:
On the finite section method for computing exponentials of doubly-infinite skew-Hermitian matrices

27.2013 A. ABDULLE, G. VILMART, K. C. ZYGALAKIS:
High order numerical approximation of the invariant measure of ergodic SDEs

28.2013 S. ROSSI, T. LASSILA, R. RUIZ-BAIER, A. SEQUEIRA, A. QUARTERONI:
Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics

29.2013 F. BONIZZONI, F. NOBILE:
Perturbation analysis for the Darcy problem with log-normal permeability

30.2013 Z. LI, A. USCHMAJEW, S. ZHANG:
On convergence of the maximum block improvement method

31.2013 R. GRANAT, B. KAGSTRÖM, D. KRESSNER, M. SHAO:
Parallel library software for the multishift QR algorithm with aggressive early deflation

32.2013 P. CHEN, A. QUARTERONI:
Weighted reduced basis method for stochastic optimal control problems with elliptic PDE contstraint

33.2013 P. CHEN, A. QUARTERONI, G. ROZZA:
Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by Stokes equations

34.2013 A. ABDULLE, M. J. GROTE, C. STOHRER:
Finite element heterogeneous multiscale method for the wave equation: long time effects

35.2013 A. CHKIFA, A. COHEN, G. MIGLIORATI, F. NOBILE, R. TEMPONE:
Discreate least squares polynomial approximation with random evaluations-application to parametric and stochastic elliptic PDEs

36.2013 N. GUGLIELMI, D. KRESSNER, C. LUBICH:
Computing extremal points of symplectic pseudospectra and solving symplectic matrix nearness problems

37.2013 S. DEPARIS, D. FORTI, A. QUARTERONI:
A rescaled localized radial basis functions interpolation on non-cartesian and non-conforming grids