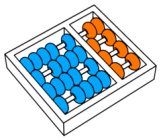

Adaptive Multiscale Function Approximation II: Regular Tensorial Bases

Gilcélia Regiâne de Souza *Jorge Stolfi*

Technical Report - IC-17-01 - Relatório Técnico
January - 2017 - Janeiro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Adaptive Multiscale Function Approximation II: Regular Tensorial Bases

Gilcélia Regiâne de Souza[1]
Jorge Stolfi[2]

Abstract

We apply the general top-down algorithm for adaptive multiscale approximation *HApp*, described in the Part I of this article, to a specific type of function approximation bases which we call regular multiscale bases. The algorithm guarantees a specified maximum approximation error at every sampling point. While the resulting adaptive basis is not necessarily minimal, it can be much smaller than the full basis, for target functions with localized detail at various spatial resolution scales.

The bases elements are tensorial splines with compact support. These bases are similar to standard wavelet bases, except that they provide explicit analytic formulas for the approximating function; that can be used, for example, for differentiation and interpolation between the sampling points.

In this part of the article, we assume a regular grid of sampling points and a box-like domain with toroidal topology. These choices allow considerable savings of computing time. We also use at each level a modified least squares analysis operator with Bayesian outlier rejection.

1 Introduction

In Part I of this article [16], we described a general algorithm for adaptive multiscale approximation (*HApp*) using a general basis reduction routine (*Reduce*). Both algorithms were very general, independent of the domain and mesh geometry, of the nature and arrangement of the basis elements, of the sampling points, and of the initial approximation operator *Analyze* used at each level. We also described a particular version of Analyze, namely iterative least squares with *Bayesian* rejection of outliers and rejection of "bad" cells.

In this second part of the article, we apply the the approximation method of Part I to *regular* bases, a specific type of basis that we found to be particularly useful and effective for a broad variety of target functions.

Namely, every element of a regular basis is the discrete sampling of a multi-dimensional *tensorial element*, which is the product of several univariate basis elements with bounded support, each depending on a single coordinate of the domain. A full *single-scale* regular basis consists of multiple translated copies of the same element, arranged in a regular grid. A full *multiscale* regular basis is a hierarchy of single-scale regular bases, where the elements at each level are similar to those in the next higher level, but are reduced in the spatial

[1]UFSJ, Ouro Branco, Brazil gilcelia@ufsj.edu.br
[2]UNICAMP, Campinas, SP, Brazil stolfi@ic.unicamp.br

directions (before sampling) and are arranged in a denser grid. An *adaptive* multiscale regular basis consists of a subset of each level of a full multiscale regular basis, up to some maximum level.

We assume a box-like domain with periodic boundary conditions, spanned by a regular array of sampling points aligned with the basis element grid. These choices allow considerable time savings in the computation of the basis and moment matrices. One could also use the regular bases described in this article with irregular sampling points; however, there will be no savings in computing time in that case.

Recall that the *HApp* algorithm guarantees a specified maximum approximation error at every sampling point, if it is allowed to use a sufficiently deep hierarchical basis. While the resulting adaptive basis is not necessarily minimal, it can be much smaller than the full basis, for target functions with localized detail at various spatial resolution scales.

1.1 Related work

A regular multiscale basis is similar to a standard wavelet basis [13], except that it provides explicit analytic formulas for the approximating function; that can be used, for example, for differentiation and interpolation between the sampling points.

A regular basis is also similar to a radial basis [1], but its elements are not radially symmetric. Their shape allows them to achieve a more uniform coverage of the domain when placed in a regular grid.

As observed in Part I, the *HApp* algorithm can be compared to the algorithms by Iske [10, 11, 9, 8]. However, due to its top-down approach, *HApp* is more efficient for identifying the reduced basis. Our regular single-scale bases include the multiscale B-spline [3, 14] bases as a special case which we use in our tests (see section 3). Our multiscale regular bases are also similar to the bases used by Castro et al [2].

2 Detailed description

2.1 Tensor bases

The reader is referred to section 2 of Part I for the notation and conventions. Here we assume that each element σ_j of a discrete basis S (in particular, of any level $S^{(\ell)}$ of the full hierarchical basis) is obtained by sampling a *reconstruction element*, a function ϕ_j from the domain $\mathbb{D}$ to $\mathbb{R}$, at the sampling points p_k. Let $\widehat{\phi}_1, \widehat{\phi}_2, \ldots$ be the reconstruction elements of the reduced basis $\widehat{S}_1, \widehat{S}_2, \ldots$ chosen by *HApp*, and $\widehat{\alpha}_1, \widehat{\alpha}_2 \ldots$ be the corresponding coefficients. The *reconstruction* of the sampled values f_i will then be the function $\widehat{\xi}$ from $\mathbb{D}$ to $\mathbb{R}$ defined by

$$\xi(x) = \sum \widehat{\alpha}_j \widehat{\phi}_j(x) \tag{1}$$

for all $x \in \mathbb{D}$. Note that the discrete approximation ξ returned by *HApp* is the sampling of function $\widehat{\xi}$.

2.2 Mother pulses

Each reconstruction element ϕ_j, is derived in turn from a single *mother pulse* Φ, a function from $\mathbb{R}$ to $\mathbb{R}$. Typically, Φ is an even function (symmetric around 0), positive when the argument is zero, and zero when the argument is sufficiently far from 0. See figure 1. Specifically we assume that the mother pulse Φ is zero outside some *mother support interval*

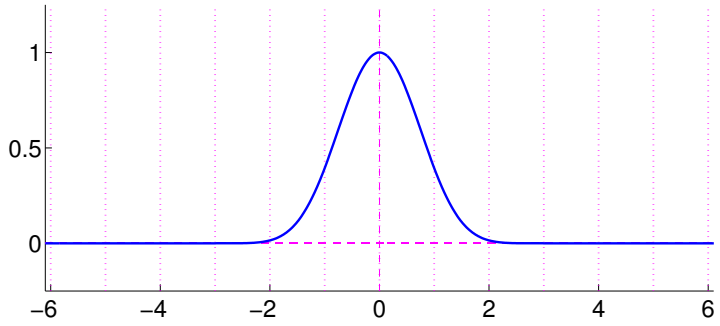

Figure 1: A typical mother pulse.

$\sup(\Phi) = [-H/2, +H/2]$. The parameter H will be called the *support width* of Φ.

2.3 Tensorial pulses

Each reconstruction element that we use is a d-dimensional *tensorial pulse* ϕ_j which is a function from $\mathbb{R}^d$ to $\mathbb{R}$, derived from the chosen mother pulse Φ by the formula

$$\phi_j(x) = \prod_{i=1}^{d} \Phi\Big(\frac{x_i - c_{ji}}{\rho_i}\Big), \tag{2}$$

for a generic point $x = (x_1, x_2, \ldots, x_d) \in \mathbb{D}$. Here $c_j = (c_{j1}, c_{j2}, \ldots, c_{jd})$ is a point of $\mathbb{D}$, the *center* of the pulse, and each ρ_i is a positive real *stretching factor* for the pulse along coordinate axis i. See figure 2. For problems where the coordinate axes have different characters (e. g., time vs. space) it may be appropriate to use a different mother pulse Φ_i for each axis. This generalization would not require any other changes to the method, and would not imply any significant extra cost.

2.4 Basis element placement

We assume that the domain of approximation $\mathbb{D}$ is a box in the cartesian space $\mathbb{R}^d$, for some dimension d, aligned with the coordinate axes.

We assume that the centers c_j are arranged in a regular orthogonal grid in the box $\mathbb{D}$, with Q_i elements along each axis i, whose centers are separated by a *basis grid step* ζ_i. That is, the center c_j is $\Big((u_{j1} + \frac{1}{2})\zeta_1, (u_{j2} + \frac{1}{2})\zeta_2, \ldots, (u_{jd} + \frac{1}{2})\zeta_d\Big)$, where each u_{ji} is an integer in $\{0..Q_i - 1\}$. The integer tuple $u_j = (u_{j1}, u_{j2}, \ldots, u_{jd})$ is the *position* of the element ϕ_j in the basis. Typically, the stretching factors ρ_i are proportional to the steps ζ_i. The domain box $\mathbb{D}$ has length $L_i = \zeta_i Q_i$ along axis i that is, $\mathbb{D} = [0, L_1] \times [0, L_2] \times \cdots \times [0, L_d]$.

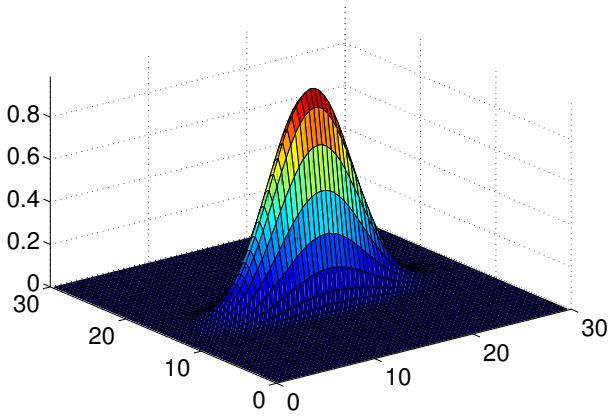

Figure 2: A tensorial pulse ϕ_j derived from the mother pulse Φ of figure 1, with center $c_j = (15, 15)$ and stretching factors $\rho_j = (2, 6)$.

2.5 The basis mesh

As detailed in the Part I of this article, a single-scale basis ϕ has an associated *mesh* $\mathbf{K}$. Each cell $\mathbf{K}_j$ of this mesh is a region in $\mathbb{D}$ where element ϕ_j nominally dominates. In the case of a regular basis, $\mathbf{K}_j$ is, by definition, a box with length ζ_i along each axis i, centered at the point c_j of $\mathbb{D}$. The *position* of the cell $\mathbf{K}_j$ is that of its associated reconstruction element ϕ_j. See figure 3

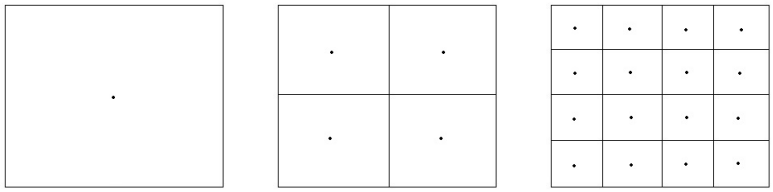

Figure 3: Cells (rectangles) and element centers (dots) for the first three levels $\phi^{(0)}$, $\phi^{(1)}$, $\phi^{(2)}$ of the 2-dimensional hierarchical regular basis.

2.6 Element supports

Note that each reconstruction element ϕ_j will be zero outside the d-dimensional box with center c_j and width $\rho_i H$ along each axis i. We define the *support box* $\sup(\phi_j)$ of element ϕ_j to be the set of all mesh cells in which the element is not entirely zero. If u_j is the position of ϕ_j, and $\rho_i H \leq L_i$, then the support box comprises cells with indices ranging from $u_{ji} - r$ to $u_{ji} + r$, modulo Q_i, along each axis i; where $r = \lceil (\rho_i H/\zeta_i - 1)/2 \rceil$. In other words, the support box is a block of $\big(2r + 1\big)^d$ cells centred on cell $\mathbf{K}_j$. See Figure 4. If $\rho_i H > L_i$, on the other hand, the support box is the whole mesh $\mathbf{K}$ (see section 2.8).

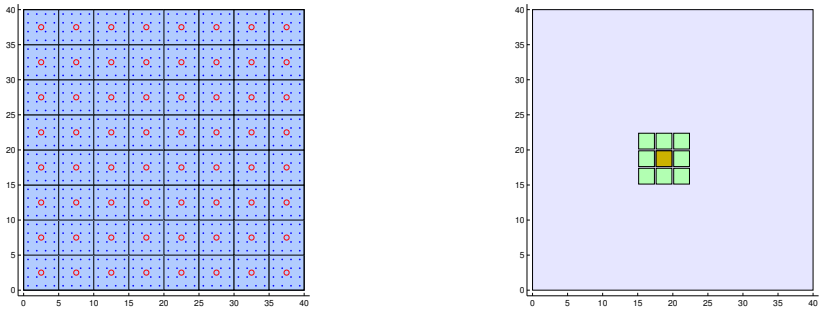

Figure 4: At left, illustration of a single-level regular basis with $d = 2$, $L = (40, 40)$, $Q = (8, 8)$, $\zeta = (5, 5)$, showing the element centers c_j (red dots), the sampling points p_k (small blue dots) with oversampling factors $t = (4, 4)$, and the support box of the element ϕ_j with position $(3, 3)$, $H = 3$, $\rho = (1, 1)$, and center $(17.5, 17.5)$.

2.7 Toroidal domain

A recurrent complication in the analysis of approximation methods on a bounded domain $\mathbb{D}$ is that the errors near the border of $\mathbb{D}$ are usually different from those in its interior. To avoid this complicating effect, and for efficiency reasons (see section 3.1), we will assume that each face of the domain box $\mathbb{D}$ is identified with the opposite face. In other words, we take $\mathbb{D}$ to be the quotient of $\mathbb{R}^d$ by the equivalence relation that relates two points x', x'' iff $x'_i - x''_i$ is an integer multiple of L_i for every i.

With this assumption, the domain $\mathbb{D}$ becomes a d-dimensional manifold without border, a *d-dimensional flat torus*, that retains the (flat) Euclidean local geometry and differential structure of $\mathbb{R}^d$ at every point. Note that all points of $\mathbb{D}$ are equivalent under toroidal translations.

This assumption is equivalent to implicitly extending any function defined on $\mathbb{D}$ to a function defined over $\mathbb{R}^d$ that is periodic along every axis i with period L_i. One drawback of this assumption is that it creates spurious discontinuities in the target function F when the latter is not periodic. This problem can be avoided by mapping the domain of interest of F to a sufficiently small region $\mathbb{D}'$ inside the box $\mathbb{D}$, and redefining F in the surrounding region $\mathbb{D} \setminus \mathbb{D}'$ so that it is smooth and periodic in the whole box $\mathbb{D}$, including at points in the border. See for example the target functions used in the tests (section 3.1).

2.8 Element fold-over

The toroidal domain assumption implies that any parts of a tensorial pulse ϕ_j that would otherwise fall outside the box $\mathbb{D}$ are "folded over" into the box across the opposite face(s). If the width $\rho_i H$ of the element's support along some axis i is greater than the corresponding box width L_i, the folded-over pulse will overlap itself, and the folded-over parts must be added together. See figure 5.

More precisely, the generic basis element is not the tensorial pulse ϕ_j defined by formula (2), but is the sum of infintely many copies of that pulse, whose centers form an

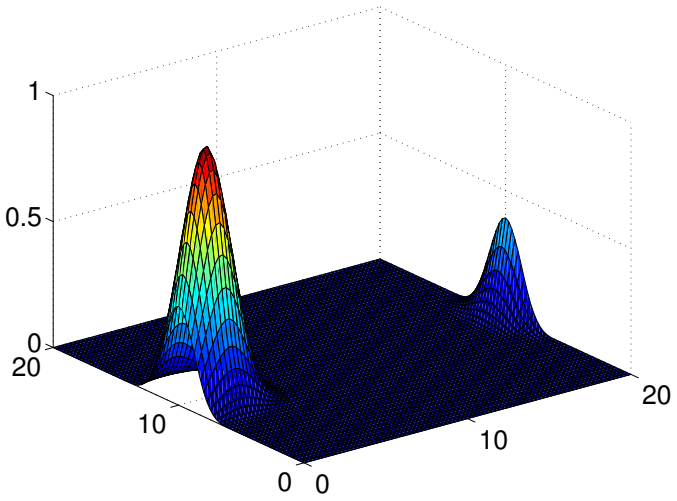

Figure 5: A sample reconstruction element on the domain $[0, 20] \times [0, 20]$

infinite regular grid with stride L_i along each axis i. That is, the generic element is

$$\phi_j(x) = \sum_{v \in \mathbb{Z}^d} \prod_{i=1}^{d} \Phi\Big(\frac{x_i - c_{ji} - v_i L_i}{\rho_i}\Big), \tag{3}$$

for every point x of $\mathbb{D}$. The sum (3) can be factored as follows

$$\phi_j(x) = \prod_{i=1}^{d} \left[\sum_{k \in \mathbb{Z}} \Phi\Big(\frac{x_i - c_{ji} - k L_i}{\rho_i}\Big) \right]. \tag{4}$$

Therefore, each basis element is still a tensorial element,

$$\phi_j(x) = \prod \Phi_i^*\Big(x_i - c_{ij}\Big), \tag{5}$$

where Φ_i^* is the *stretched and replicated mother function*

$$\Phi_i^*(z) \;=\; \sum_{k \in \mathbb{Z}} \Phi\Big(\frac{z - k L_i}{\rho_i}\Big) \;=\; \sum_{k \in \mathbb{Z}} \Phi\Big(\frac{z}{\rho_i} - k \frac{\zeta_i}{\rho_i} Q_i\Big). \tag{6}$$

These infinite summations always converge if the mother function has bounded support width H, since only a finite number of terms will be nonzero. Specifically, in the sum (6) we need only consider terms for which $-H/2 \le (z - kL_i)/\rho_i \le +H/2$, that is,

$$-\lfloor (\rho_i H/2 + z)/L_i \rfloor \;\le\; k \;\le\; \lfloor (\rho_i H/2 - z)/L_i \rfloor .$$

In particular, if $\rho_i H \le L_i$, the sum (6) has only one non-zero term, whose index k is z/L_i rounded to the nearest integer.

2.9 Efficient sampling

As for the sampling points p_k, we assume that they too form a regular rectangular grid, whose spacing along each axis i a sub-multiple $\delta_i = \zeta_i/t_i$ of the basis step ζ_i. The integer t_i is the *oversampling factor* for axis i. Each sampling point p_k is of the form

$$\left(\left(v_{k1} + \frac{1}{2}\right)\frac{\zeta_1}{t_1},\ \left(v_{k2} + \frac{1}{2}\right)\frac{\zeta_2}{t_2},\ \cdots,\ \left(v_{kd} + \frac{1}{2}\right)\frac{\zeta_d}{t_d}\right)$$

were each v_{ki} is an integer between 0 and $t_i Q_i - 1$. See Figure 6. In particular, if $t_i = 1$ for all i, the sampling points coincide with the element centers, the approximation problem becomes an interpolation one.

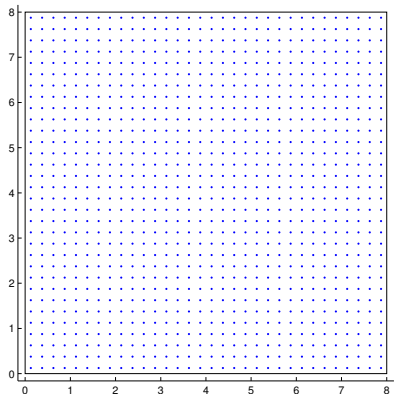

Figure 6: A regular grid of sample points with $Q = (8, 8)$ and $t = (2, 2)$.

One important consequence of all these definitions is that the *discretized* basis elements of a given level ℓ are all similar. Specifically, for any two reconstruction elements ϕ_r and ϕ_s, and any sample point p_k, the basis matrix element $S_{kr} = \phi_r(p_k)$ is equal to $S_{is} = \phi_s(p_i)$, where i is such that $p_i = p_k + c_s - c_r$.

This observation can be used to greatly reduce the time and memory needed to compute the pre-basis matrix $\widetilde{S} = \widetilde{S}^{(\ell)}$ for each level. Namely, it suffices to compute just the first column σ_1 of the full basis matrix S, since all columns of S (and of the pre-basis $\widetilde{S}$) will be permutations of that one. Therefore, the cost to build the pre-basis matrix $\widetilde{S}$ for $\widetilde{n}$ basis elements and m total sample points is roughly proportional to $m\beta(\tau_\phi + \widetilde{n}\kappa)$, where: β is the fraction of sampling points that are in the domain box of ϕ_1, which is at most $(2r + 1)^d/n$; τ_ϕ is the cost of evaluating one sample value $\phi_1(p_k)$; and κ is the cost of storing one sample into the matrix $\widetilde{S}$.

These choices also allow us to compute the least squares system matrix $M = \widetilde{S}^\top \widetilde{S}$ very efficiently. Namely, it suffices to compute the first row of M, since the other rows are just permutations of it. Each element M_{1k} of the first row is $\widetilde{\sigma}_1^\top \widetilde{\sigma}_k$, and only needs to be computed if the support boxes of ϕ_1 and ϕ_k intersect. The cost of computing M is therefore roughly proportional to $\widetilde{n}m\theta\mu$ to compute the first row, and $\widetilde{n}^2\beta\kappa$ to copy it to the other $\widetilde{n} - 1$ rows. Here θ is the fraction of elements ϕ_j of the basis whose support box intersects that of ϕ_1,

which is at most $(4r-1)^d/n$; and μ is the cost of one floating-point multiplication and one sum.

If the sampling points are irregular, these optimizations are not possible; the value of $\widetilde{S}_{ij} = \widetilde{\phi}_j(p_k)$ must be computed for every element $\widetilde{\phi}_j$ and every point p_k in support of $\widetilde{\phi}_j$, with cost $\widetilde{n}m\beta\tau_\phi$; and the matrix M must be computed by the formula $\widetilde{S}^\top\widetilde{S}$, with cost proportional to $\widetilde{n}^2 m\beta\mu$.

2.10 Hierarchical tensor pulse bases

Following Part I, we define a *multiscale* regular basis as a hierarchy of *levels* $S^{(0)}, S^{(1)}, \ldots$ where each level $S^{(\ell)}$ is a single-scale regular basis. By definition, the basis $S^{(\ell)}$ has $Q_i^{(\ell)} = 2^\ell$ elements along every axis i, with step $\zeta_i^{(\ell)} = L_i/2^\ell$. The corresponding mesh $\mathbf{K}^{(\ell)}$ has $Q_i^{(\ell)} = 2^\ell$ cells along each axis i, that span the domain $\mathbb{D}$. That is, level 0 has only one cell, $\mathbf{K}_1^{(0)} = \mathbb{D}$; and each cell of level ℓ is the union of 2^d cells of level $\ell+1$, in quadtree or octree fashion [6]. Note that all cells at all levels are similar in shape, and the element centers of all levels are all distinct. See Figure 3.

As explained in Part I, the pre-basis $S^{(\ell)}$ (step 14 of algorithm *HApp*) will be a subset of the full regular basis $S^{(\ell)}$, and the final basis $\widehat{S}^{(\ell)}$ (step 16) will be a subset of $\widetilde{S}^{(\ell)}$.

Note that, in step 10 of *HApp*, the set $\mathbf{B}^{(\ell)}$ of relevant cells is simply the union of all support boxes of all the reconstruction elements $\widehat{\phi}_j^{(\ell)}$ associated with the basis elements $\widehat{\sigma}_j^{(\ell)}$.

2.11 Starting level

As discussed in section 4.1 of Part I, many hierarchical approximation schemes described in the literature assume that each level space $\mathcal{S}^{(\ell)}$ is linearly independent, (or even orthogonal) to all previous levels. Moreover, in those schemes, the first level is usually defined to be some fairly detailed mesh, with hundreds or thousands of cells. Therefore, in order to capture global trends of the target function, those schemes must add to the basis, in that initial level, some extra elements with global support, such as polyharmonic splines and/or low-degree polynomials [7, 8, 9, 12].

As explained in Part I, our hierarchical basis use nested (rather than independent) approximation spaces at each level:

$$\mathcal{S}^{(\ell_{\min})} \subseteq \mathcal{S}^{(\ell_{\min}+1)} \subseteq \ldots \subseteq \mathcal{S}^{(\ell_{\max})}. \tag{7}$$

The regular spline bases that we use in this part of the article also satisfy this property. With this assumption, those extra ad-hoc elements in the first level are not needed, since the full basis $S^{(\ell)}$ at every level can generate the full space $\mathcal{S}^{(k)}$ for any previous level $k \le \ell$. In particular, the basis $\mathcal{S}^{(\ell_{\min})}$ can capture any global trends of the target function for any $\ell_{\min} \ge 0$.

Indeed, we can start the algorithm *HApp* at any level $\ell_{\min}$ above the finest level $\ell_{\max}$, even with $\ell_{\min} = \ell_{\max}$ (in which case the hierarchical basis reduces to a regular single-scale basis). Still, as show in section 4, starting at the appropriate level $\ell_{\min}$ between 0 and $\ell_{\max}-1$ usually results in a much smaller adaptive basis $\widehat{S}^{(*)}$, and considerable time savings compared to single-scale approximation ($\ell_{\min} = \ell_{\max}$).

3 Test parameters

In this section we describe the common settings of some parameters for all the tests of the *HApp* algorithm that we present in section 4. These tests are meant to illustrate the effectiveness of the method, namely its ability to find a small representation of the target function; and the effect of certain choices in the *HApp* and *Analyze* procedure.

3.1 Sampling points and values

We use a two-dimensional square domain $\mathbb{D}$ ($d = 2$) with $L_1 = L_2 = 8$ and a dense regular grid of sampling points as described in section 3.1 with $2^9 \times 2^9 = 262{,}144$ points evenly spaced in the box $\mathbb{D}$.

The sample values f_i were obtained from a *target function* F which was one of the four functions F_{O}, F_{C}, F_{G}, and F_{M} described in Appendix A. See figure 7.

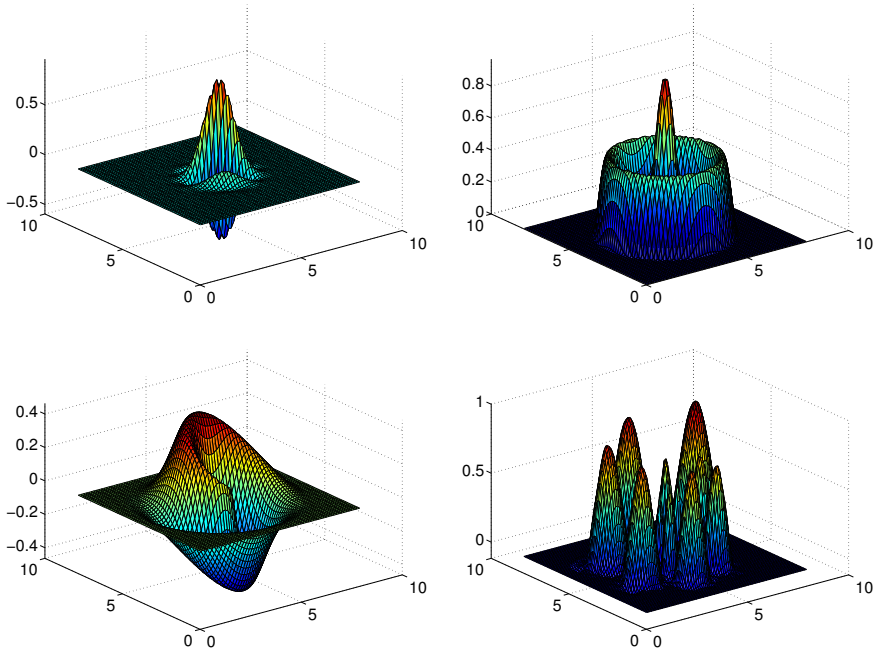

Figure 7: The target functions F_{O} and F_{C} (top), F_{G} and F_{M} (bottom), used in our tests.

3.2 Mother functions

In these tests, the reconstruction elements of all bases are tensor pulses derived from four mother functions Φ_g^{S} with $2 \le g \le 5$, which are the basis elements of uniform B-spline bases of degree g [4, 14, 15]. See figure 8. Each function Φ_g^{S} is a polynomial spline of degree g, symmetric around 0 and continuous to order $g-1$, consisting of $g+1$ polynomials restricted to $g+1$ intervals of length 1 centered at 0. See Appendix B 5 for the detailed formulas.

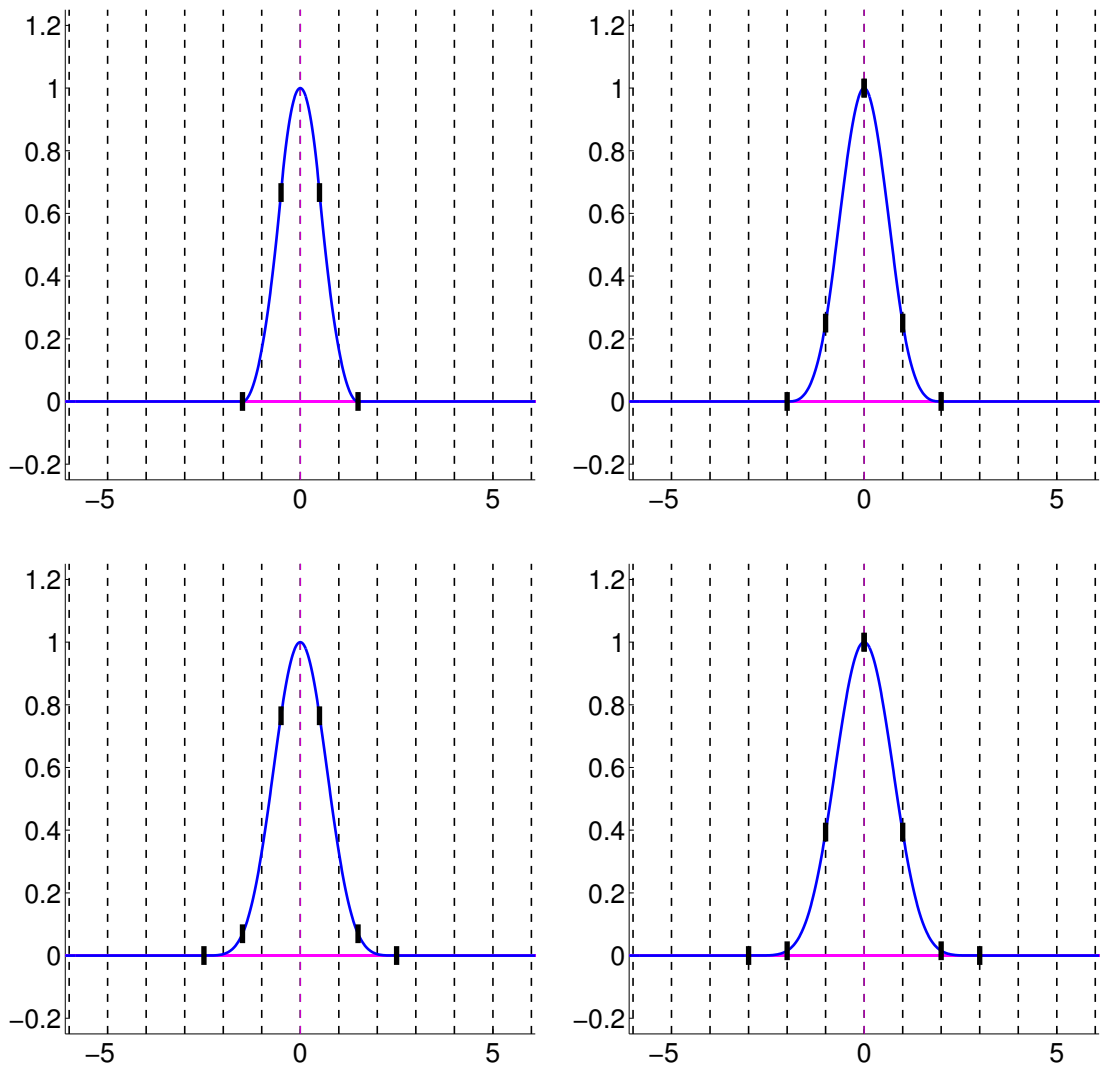

Figure 8: The B-spline mother pulses Φ_2^{S} and Φ_3^{S} (top), Φ_4^{S} and Φ_5^{S} (bottom). The support of Φ_g^{S} has width $H = g + 1$ and therefore is the interval $[-(g+1)/2, +(g+1)/2]$. The support of each pulse and the discontinuities in its derivative of order g are shown by the tics. The dotted vertical lines show the positions of the grid cell boundaries.

3.3 Multiscale basis

Each reconstruction element $\phi_j^{(\ell)}$ of the multiscale basis is the product of stretched and shifted copies of the unidimensional mother pulses Φ_g^{S}, according to formulas (5) and (6). We use $\rho_i = \zeta_i = 1$ for all i; so that the discontinuities in the derivative of order g of two elements $\phi_j^{(\ell)}$ and $\phi_k^{(\ell)}$, defined as above, coincide wherever the two elements overlap. Specifically, if g is odd, the discontinuities occur at cell boundaries, that is, at hyperplanes of $\mathbb{R}^d$ defined by the equations $x_i = k\zeta_i$ for $k \in \mathbb{Z}$. In this case, within each cell, each element ϕ_j is a single polynomial of the argument coordinates. If g is even, the discontinuities occur at hyperplanes that are displaced by $\zeta_i/2$ relative to the cell boundaries.

It turns out that the elements $\phi_j^{(\ell)}$ of a single level ℓ can reproduce any polynomial spline

of degree g or less, defined on the mesh $\mathbf{K}$, that is continuous to order $g-1$ and whose discontinuities of order g are contained in the hyperplanes described above. Note that the hyperplanes of discontinuity of level ℓ are a subset of those of level $\ell+1$. For these reasons, every element $\phi_j^{(\ell+1)}$ can be expressed as a combination of elements of level ℓ, satisfying assumption (7).

3.4 Other parameters

In most tests, except those of section 4.2, the starting level $\ell_{\min}$ was set to 3, so that the full basis $\phi^{(\ell_{\min})}$ has $2^3 \times 2^3 = 64$ elements. The maximum level $\ell_{\max}$ was set to 9, so that $\phi^{(\ell_{\max})}$ has $2^9 \times 2^9 = 262{,}144$ elements. Note that at level $\ell_{\max}$ there was only one sampling point per cell, i. e. $t^{(9)} = (1,1)$. However, in all tests the *HApp* procedure terminated before reaching that level. In what follows, for each test $\ell_{\max}$ is assumed to be the last level ℓ for which $\widehat{S}^{(\ell)}$ was not empty.

The error tolerance $\epsilon_{\max}$ for *Reduce* was set to 2.5×10^{-3}, in all tests except those of section 4.3. In all tests we used the *Analyze* procedure described section 6.2 of Part I; except in section 4.4, where the Bayesian outlier rejection was disabled.

3.5 Efficiency metrics

As in Part I, the effectiveness and efficiency of *HApp* on each test run are quantified by a *size metric* τ and a *cost metric* λ. The size metric is simply the ratio between the size $\widehat{n}^{(*)}$ of the adaptive multiscale basis $\widehat{S}^{(*)}$ chosen by *HApp* and the size $n_{\max} = n^{(\ell_{\max})}$ of the full regular basis $S^{(\ell)}$ at the maximum level ℓ present in $\widehat{\phi}^{(*)}$. The cost metric λ is the ratio between the total estimated running time $\widetilde{t}^{(*)}$ of the plain LS method applied to all reduced bases $\widetilde{S}^{(\ell)}$, and the estimated time $t^{(\ell_{\max})}$ of the plain LS method with the full basis $S^{(\ell_{\max})}$ for the last level. The time to solve one LS problem for a basis of n elements was assumed to be proportional to $n^2(n+q)$ where q was the average number of sampling points in the support of on basis element.

4 Test Results

4.1 Comparison of mother pulses

The first set of tests compares the effectiveness and efficiency (in terms of the size and cost metrics above) obtained with the four mother pulses defined in section 3.2.

The results of all the tests in this set are summarized in tables 1 to 4. In the first part of each table, $\widehat{n}^{(\ell)}$ is the size of the reduced basis $\widehat{S}^{(\ell)}$ of level ℓ, for $\ell = 0, 1, \ldots, \ell_{\max}$. In the second part, $\widehat{n}^{(*)}$ is the total size of the adaptive basis $\widehat{S}^{(*)}$ found by *HApp*; $n_{\max}$ is the size of the full basis $S^{(\ell)}$ for the last level ℓ present in $\widehat{S}^{(*)}$; τ is the size metric of the final basis $\widehat{S}^{(*)}$; and λ is the cost metric of the *HApp* algorithm.

Table 1: Results of *HApp* for $F = F_{\mathrm{O}}$ and different mother pulses.

F	Φ	$\widehat{n}^{(0)}$	$\widehat{n}^{(1)}$	$\widehat{n}^{(2)}$	$\widehat{n}^{(3)}$	$\widehat{n}^{(4)}$	$\widehat{n}^{(5)}$	$\widehat{n}^{(6)}$	$\widehat{n}^{(7)}$	$\widehat{n}^{(8)}$	$\widehat{n}^{(9)}$
F_{O}	Φ_2^{S}	-	-	-	20	72	184	274	18	8	-
F_{O}	Φ_3^{S}	-	-	-	20	60	202	222	62	-	-
F_{O}	Φ_4^{S}	-	-	-	20	60	208	36	-	-	-
F_{O}	Φ_5^{S}	-	-	-	20	62	208	-	-	-	-

F	Φ	$\widehat{n}^{(*)}$	$n_{\max}$	τ	λ
F_{O}	Φ_2^{S}	576	65536	0.0088	1.1×10^{-6}
F_{O}	Φ_3^{S}	566	16384	0.0345	1.1×10^{-4}
F_{O}	Φ_4^{S}	324	4096	0.0791	6.8×10^{-3}
F_{O}	Φ_5^{S}	290	1024	0.2832	9.7×10^{-2}

Table 2: Results of *HApp* for $F = F_{\mathrm{C}}$ and different mother pulses.

F	Φ	$\widehat{n}^{(0)}$	$\widehat{n}^{(1)}$	$\widehat{n}^{(2)}$	$\widehat{n}^{(3)}$	$\widehat{n}^{(4)}$	$\widehat{n}^{(5)}$	$\widehat{n}^{(6)}$	$\widehat{n}^{(7)}$	$\widehat{n}^{(8)}$	$\widehat{n}^{(9)}$
F_{C}	Φ_2^{S}	-	-	-	32	232	700	977	468	48	-
F_{C}	Φ_3^{S}	-	-	-	32	244	812	1328	416	24	-
F_{C}	Φ_4^{S}	-	-	-	32	252	852	1226	416	56	-
F_{C}	Φ_5^{S}	-	-	-	32	252	884	1348	400	16	-

F	Φ	$\widehat{n}^{(*)}$	$n_{\max}$	τ	λ
F_{C}	Φ_2^{S}	2457	65536	0.0375	2.2×10^{-5}
F_{C}	Φ_3^{S}	2856	65536	0.0436	5.7×10^{-5}
F_{C}	Φ_4^{S}	2834	65536	0.0432	5.5×10^{-5}
F_{C}	Φ_5^{S}	2932	65536	0.0447	8.1×10^{-5}

Table 3: Results of *HApp* for $F = F_{\mathrm{G}}$ and different mother pulses.

F	Φ	$\widehat{n}^{(0)}$	$\widehat{n}^{(1)}$	$\widehat{n}^{(2)}$	$\widehat{n}^{(3)}$	$\widehat{n}^{(4)}$	$\widehat{n}^{(5)}$	$\widehat{n}^{(6)}$	$\widehat{n}^{(7)}$	$\widehat{n}^{(8)}$	$\widehat{n}^{(9)}$
F_{G}	Φ_2^{S}	-	-	-	52	196	118	70	22	14	20
F_{G}	Φ_3^{S}	-	-	-	52	174	96	44	20	18	14
F_{G}	Φ_4^{S}	-	-	-	52	180	88	42	14	32	18
F_{G}	Φ_5^{S}	-	-	-	52	188	94	44	10	28	12

F	Φ	$\widehat{n}^{(*)}$	$n_{\max}$	τ	λ
F_{G}	Φ_2^{S}	492	262144	0.0019	3.6×10^{-8}
F_{G}	Φ_3^{S}	418	262144	0.0016	5.6×10^{-8}
F_{G}	Φ_4^{S}	426	262144	0.0016	8.4×10^{-8}
F_{G}	Φ_5^{S}	428	262144	0.0016	1.2×10^{-7}

Table 4: Results of *HApp* for $F = F_{\mathrm{M}}$ and different mother pulses.

F	Φ	$\widehat{n}^{(0)}$	$\widehat{n}^{(1)}$	$\widehat{n}^{(2)}$	$\widehat{n}^{(3)}$	$\widehat{n}^{(4)}$	$\widehat{n}^{(5)}$	$\widehat{n}^{(6)}$	$\widehat{n}^{(7)}$	$\widehat{n}^{(8)}$	$\widehat{n}^{(9)}$
F_{M}	Φ_2^{S}	-	-	-	39	239	731	1404	1079	244	11
F_{M}	Φ_3^{S}	-	-	-	39	245	806	1505	979	173	5
F_{M}	Φ_4^{S}	-	-	-	39	251	853	1467	973	148	23
F_{M}	Φ_5^{S}	-	-	-	39	252	877	1545	1056	165	18

F	Φ	$\widehat{n}^{(*)}$	$n_{\max}$	τ	λ
F_{M}	Φ_2^{S}	3747	262144	0.0143	7.1×10^{-7}
F_{M}	Φ_3^{S}	3752	262144	0.0143	1.1×10^{-6}
F_{M}	Φ_4^{S}	3754	262144	0.0143	1.2×10^{-6}
F_{M}	Φ_5^{S}	3952	262144	0.0151	1.6×10^{-6}

4.1.1 Test function F_{C} with mother function Φ_2^{S}

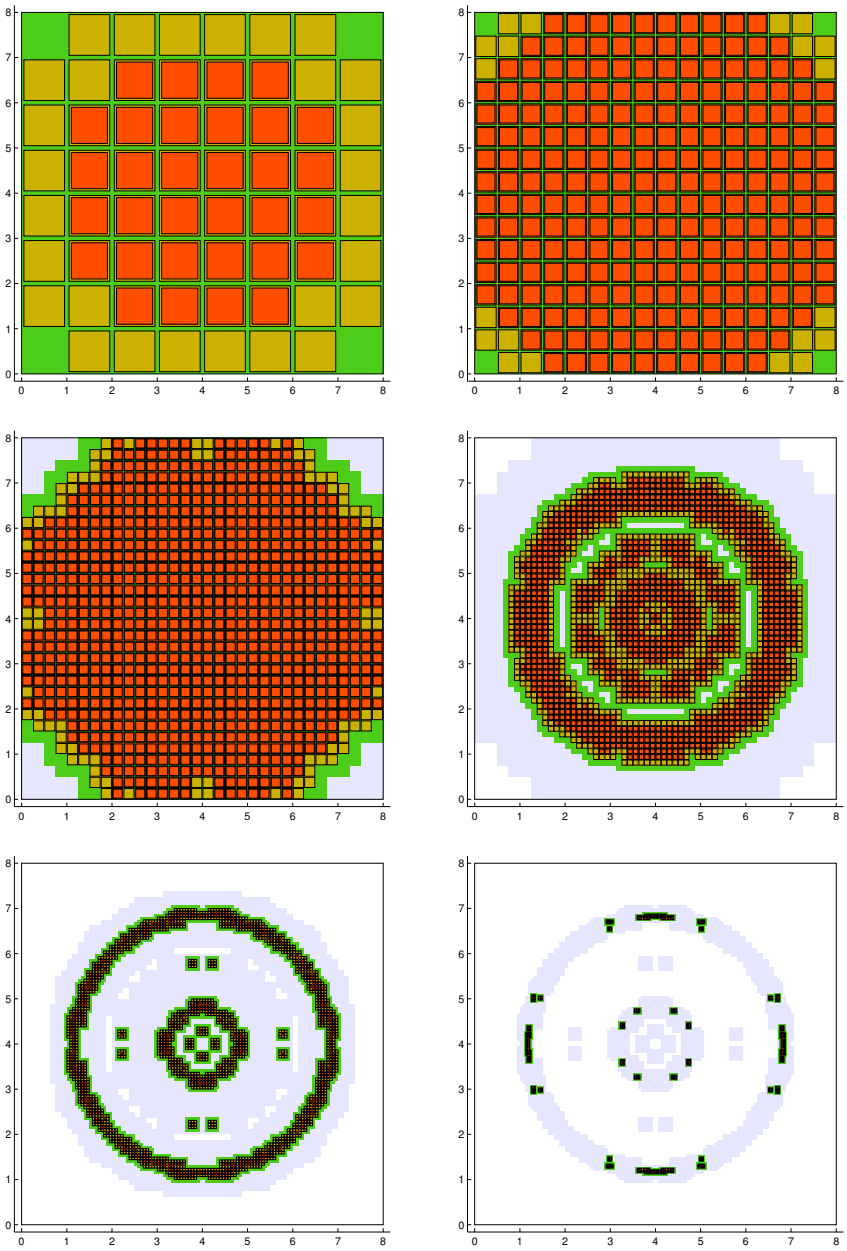

Figure 9: The cell sets $\mathbf{C}^{(\ell)}$ (orange), $\mathbf{B}^{(\ell)}$ (yellow), $\mathbf{R}^{(\ell)}$ (green), and $\mathbf{U}^{(\ell)}$ (blue) for each level ℓ in $\{\ell_{\min}..\ell_{\max}\}$.

4.1.2 Test function F_{M} with mother function Φ_2^{S}

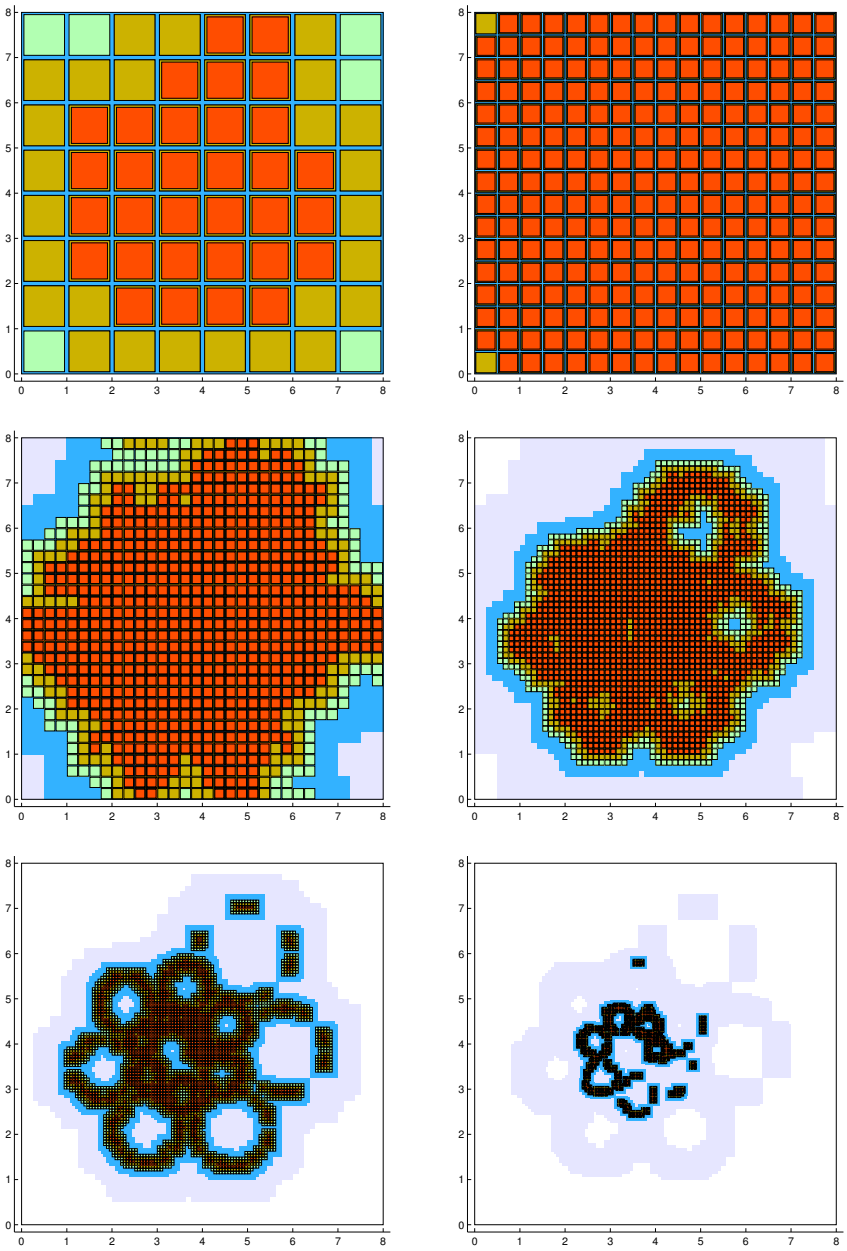

Figure 10: The cell sets $\mathbf{C}^{(\ell)}$ (orange), $\mathbf{B}^{(\ell)}$ (yellow), $\mathbf{R}^{(\ell)}$ (green), and $\mathbf{U}^{(\ell)}$ (blue) for each level ℓ in $\{\ell_{\min}..\ell_{\max}\}$.

Note that the final number $\widehat{n}^{(*)}$ of elements in the reduced basis is much smaller than the size $n_{\max} = n^{(\ell_{\max})} = 2^{\ell_{\max}} \times 2^{\ell_{\max}}$ of the complete uniform base for the last level used by *HApp*. Namely, the ratio $\tau = \widehat{n}^{(*)}/n_{\max}$ varies between 1.6% (f_{F} with Φ_2^{S}) to 11.6% (f_{C} with Φ_5^{S}). Note also that, the ratio $\widehat{n}^{(*)}/n_{\max}$ is often be much smaller than the area fraction of the domain where the target function is nonzero.

4.2 Advantage of multiscale approximation

To assess the advantages of multiscale vs. single-scale approximation, we should compare results of *HApp* with $\ell_{\min} < \ell_{\max}$ to the results of applying the basis reduction algorithm *Reduce* (see section 5.2 of Part I) to the LS approximation obtained with the single-level basis $S^{(\ell_{\max})}$; which is equivalent to applying *HApp* with $\ell_{\min} = \ell_{\max}$.

This test could not be performed for $\ell_{\max} = 8$ since it would require solving a $2^{16} \times 2^{16}$ system. The largest basis $\phi^{(\ell)}$ we could test was $\phi^{(5)}$ with $32 \times 32 = 1024$ elements. We performed this test for the function f_{O}, f_{C}, f_{G} and f_{M}, with mother function $\Phi = \Phi_2^{\mathrm{S}}$, and tolerance $\epsilon_{\max} = 2.5 \times 10^{-3}$. The results are shown in Table 5.

Table 5: Result for various F.

$F = F_{\mathrm{O}}$		$F = F_{\mathrm{C}}$		$F = F_{\mathrm{G}}$		$F = F_{\mathrm{M}}$	
$\ell_{\min}$	$\widehat{n}^{(*)}$	$\ell_{\min}$	$\widehat{n}^{(*)}$	$\ell_{\min}$	$\widehat{n}^{(*)}$	$\ell_{\min}$	$\widehat{n}^{(*)}$
3	324	3	2458	3	474	3	3547
4	302	4	2202	4	418	4	3384
5	172	5	2090	5	676	5	3173
6	638	6	2008	6	2346	6	3070
7	2596	7	5016	7	9346	7	5878
8	10446	8	19888	8	37518	8	22104

4.3 Effect of the error tolerance

In another series of tests, we compared the effectiveness of the method as a function of the error tolerance $\epsilon_{\max}$. We used the target functions F_{O} and F_{M} with Φ_2^{S} mother pulse, and $\ell_{\min} = 3$. The results are shown in Tables 6 and 7 and in Figures 11 and 12.

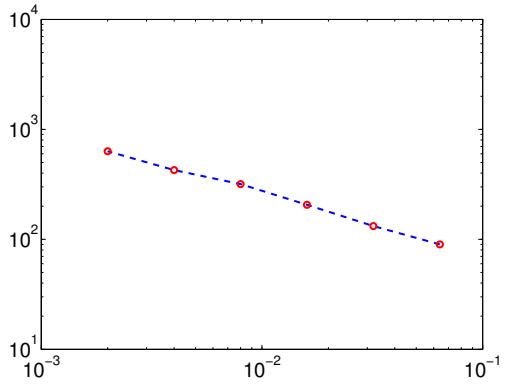

Figure 11: Plot of $\widehat{n}^{(*)}$ (vertical axis) as function of $\epsilon_{\max}$ (horizontal axis) in log-log scale.

Table 6: Results of *HApp* for $F = F_{\mathrm{O}}$ as a function of ϵ.

F	Φ	$\epsilon_{\max}$	$\widehat{n}^{(0)}$	$\widehat{n}^{(1)}$	$\widehat{n}^{(2)}$	$\widehat{n}^{(3)}$	$\widehat{n}^{(4)}$	$\widehat{n}^{(5)}$	$\widehat{n}^{(6)}$	$\widehat{n}^{(7)}$	$\widehat{n}^{(8)}$	$\widehat{n}^{(9)}$
F_{O}	Φ_2^{S}	6.4×10^{-2}	-	-	-	12	36	42	-	-	-	-
F_{O}	Φ_2^{S}	3.2×10^{-2}	-	-	-	16	42	72	2	-	-	-
F_{O}	Φ_2^{S}	1.6×10^{-2}	-	-	-	16	46	108	36	-	-	-
F_{O}	Φ_2^{S}	8.0×10^{-3}	-	-	-	16	54	142	106	-	-	-
F_{O}	Φ_2^{S}	4.0×10^{-3}	-	-	-	16	58	166	186	-	-	-
F_{O}	Φ_2^{S}	2.0×10^{-3}	-	-	-	24	76	194	284	46	8	-

F	Φ	$\epsilon_{\max}$	$\widehat{n}^{(*)}$	$n_{\max}$	τ	λ
F_{O}	Φ_2^{S}	6.4×10^{-2}	90	1024	0.0879	1.0×10^{-2}
F_{O}	Φ_2^{S}	3.2×10^{-2}	132	4096	0.0322	9.2×10^{-4}
F_{O}	Φ_2^{S}	1.6×10^{-2}	206	4096	0.0503	1.3×10^{-3}
F_{O}	Φ_2^{S}	8.0×10^{-3}	318	4096	0.0776	1.9×10^{-3}
F_{O}	Φ_2^{S}	4.0×10^{-3}	426	4096	0.1040	2.7×10^{-3}
F_{O}	Φ_2^{S}	2.0×10^{-3}	632	65536	0.0096	1.2×10^{-6}

Table 7: Results of *HApp* for $F = F_{\mathrm{M}}$ as a function of $\epsilon_{\max}$.

F	Φ	$\epsilon_{\max}$	$\widehat{n}^{(0)}$	$\widehat{n}^{(1)}$	$\widehat{n}^{(2)}$	$\widehat{n}^{(3)}$	$\widehat{n}^{(4)}$	$\widehat{n}^{(5)}$	$\widehat{n}^{(6)}$	$\widehat{n}^{(7)}$	$\widehat{n}^{(8)}$	$\widehat{n}^{(9)}$
F_{M}	Φ_2^{S}	6.4×10^{-2}	-	-	-	37	116	101	37	-	-	-
F_{M}	Φ_2^{S}	3.2×10^{-2}	-	-	-	37	158	221	95	1	-	-
F_{M}	Φ_2^{S}	1.6×10^{-2}	-	-	-	39	195	399	262	32	2	-
F_{M}	Φ_2^{S}	8.0×10^{-3}	-	-	-	39	219	567	586	204	-	-
F_{M}	Φ_2^{S}	4.0×10^{-3}	-	-	-	39	232	679	1026	586	51	-
F_{M}	Φ_2^{S}	2.0×10^{-3}	-	-	-	39	240	756	1584	1398	413	16

F	Φ	$\epsilon_{\max}$	$\widehat{n}^{(*)}$	$n_{\max}$	τ	λ
F_{M}	Φ_2^{S}	6.4×10^{-2}	291	4096	0.0710	3.6×10^{-3}
F_{M}	Φ_2^{S}	3.2×10^{-2}	512	16384	0.0312	1.3×10^{-4}
F_{M}	Φ_2^{S}	1.6×10^{-2}	929	65536	0.0142	4.8×10^{-6}
F_{M}	Φ_2^{S}	8.0×10^{-3}	1615	16384	0.0986	6.3×10^{-4}
F_{M}	Φ_2^{S}	4.0×10^{-3}	2613	65536	0.0399	2.4×10^{-5}
F_{M}	Φ_2^{S}	2.0×10^{-3}	4446	262144	0.0170	1.0×10^{-6}

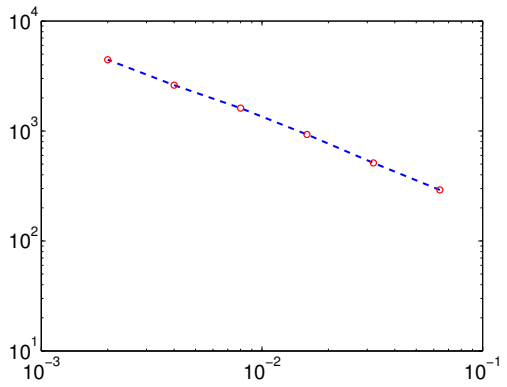

Figure 12: Plot of $\widehat{n}^{(*)}$ (vertical axis) as function of $\epsilon_{\max}$ (horizontal axis) in log-log scale.

4.4 Effect of Bayesian outlier elimination

In this section we compare the performance of *HApp* with and without the Bayesian outlier rejection (see section 6.2 of Part I), for selected target functions. See Table 2. The third column indicates the analysis operator $\mathcal{C}$ used: "—" for plain lest squares, "B" for iterated least squares with Bayesian rejection [17]. As we can see, the Bayesian elimination usually provides smaller bases.

Table 8: Results of *HApp* for all F with mother functions Φ_4^{S} and $\epsilon = 2.5 \times 10^{-3}$.

F	$\mathcal{C}$	$\widehat{n}^{(0)}$	$\widehat{n}^{(1)}$	$\widehat{n}^{(2)}$	$\widehat{n}^{(3)}$	$\widehat{n}^{(4)}$	$\widehat{n}^{(5)}$	$\widehat{n}^{(6)}$	$\widehat{n}^{(7)}$	$\widehat{n}^{(8)}$	$\widehat{n}^{(9)}$
F_{C}	—	—	—	—	32	252	864	1496	400	24	-
F_{C}	B	—	—	—	32	252	852	1226	416	56	-
F_{M}	—	—	—	—	39	254	861	1570	1100	102	5
F_{M}	B	—	—	—	39	251	853	1467	973	148	23

F	$\mathcal{C}$	$\widehat{n}^{(*)}$	$n_{\max}$	τ	λ
F_{C}	—	3068	65536	0.0468	8.8×10^{-5}
F_{C}	B	2834	65536	0.0432	5.5×10^{-5}
F_{M}	—	3931	262144	0.0150	1.5×10^{-6}
F_{M}	B	3754	262144	0.0143	1.2×10^{-6}

5 Conclusions

We combined several techniques — tensorial approximation elements based on B-spline basis elements, toroidal domain topology, and our hierarchical adaptive multiscale approximation, algorithm *HApp* — to obtain a method approximation of multiscale sample data target functions with pre-determined $\|\cdot\|_\infty$ error at the sampling points.

We implemented this algorithm for 2D data and verified empirically that it produces small approximation bases with considerable savings running time. Specifically, in the most demanding tests in section 4.1 (target function F_{M}), our method yielded approximation bases that were 1/100 of the full basis for the last level needed. This advantage was verified more directly in section 4.2.

The tests in section 4.4 showed that Bayesian outlier rejection has a modest but definite advantage over plain least squares.

Acknowledgments

This research was partially support by CNPq grant 310706/2015-7.

Appendix A: Definition of the test functions

In each case, the function is defined on the *natural domain* $\mathbb{U} = [-1/2, 1/2] \times [-1/2, 1/2]$ which is then mapped to the domain $\mathbb{D} = [0, L]^2$ by the formulas $X = x/L - 1/2$ and $Y = y/L - 1/2$, where (X, Y) are the natural arguments of the objective function and (x, y) a point of $\mathbb{D}$.

Test function F_{O}

Function F_{O} is a variant of Gabor's element [5], that is, a two-dimensional sinusoidal wave modulated by a Gaussian bell

$$F_{\mathrm{O}}(X, Y) = \exp\left(-\frac{1}{2R^2}(X^2 + Y^2) \right) \cos\Big((\alpha_x X + \alpha_y Y)\pi \Big), \tag{8}$$

where $R = 0.087$, $\alpha_x = 10$, and $\alpha_y = 5$.

Test function F_{C}

Function F_{C} consists of a central peak $f_0(X, Y)$ surrounded by a circular wall $f_1(X, Y)$:

$$F_{\mathrm{C}}(X, Y) = H_0 f_0(X, Y) + H_1 f_1(X, Y), \tag{9}$$

where

$$f_0 = \begin{cases} 0 & \text{if } X^2 + Y^2 \geq S_0^2 \\ (1 - v)^2 & \text{otherwise} \end{cases} \tag{10}$$

and

$$f_1 = \begin{cases} 0 & \text{if } X^2 + Y^2 \leq S_1 \text{ or } X^2 + Y^2 \geq S_2 \\ 4u(1 - u) & \text{otherwise} \end{cases} \tag{11}$$

in which

$$v = \frac{X^2 + Y^2}{S_0^2} \qquad \text{and} \qquad u = \frac{X^2 + Y^2 - S_1^2}{S_2^2 - S_1^2}.$$

with $S_0 = 0.1196$, $S_1 = 0.2208$, $S_2 = 0.4232$, $H_0 = 1.0$, and $H_1 = 0.5$.

Test function F_{G}

Function F_{G} is a spiral bump defined by the formula

$$F_{\mathrm{G}} = g(X, Y) \sin(t) \tag{12}$$

where

$$g(X, Y) = \begin{cases} \dfrac{v}{(1 - u)^2 \sqrt{1 + v^2}} & \text{if } u < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

with

$$u = \frac{X^2 + Y^2}{R_1^2} \qquad\qquad v = \frac{S^2}{R_0} \tag{14}$$

$$t = \sphericalangle(y, x) + k\frac{R_0}{r} \qquad\qquad S = \sqrt{R_0^2 + X^2 + Y^2}. \tag{15}$$

and $R_0 = 0.05$, $R_1 = 0.45$ e $k = 15$.

Test function F_{M}

Function F_{M} is the sum of $N = 15$ bell-shaped humps with varying widths and amplitudes arranged in spiral around point (-0.05, 0).

$$F_{\mathrm{M}}(X, Y) = \sum_{i=1}^{N} \frac{1}{2 - i/N}\, \psi\left(X, Y, \frac{i-1}{M}\right) \tag{16}$$

where

$$\psi(X, Y, s) = \Psi\left(\frac{\|(X, Y) - (\bar{X}, \bar{y})\|}{w}\right),$$

$$\Psi(z) = \begin{cases} 0 & \text{if} \quad z \geq 1 \\ (1 - z^2)^2 & \text{if} \quad z < 1 \end{cases}$$

and $M = 6.5$, $w = 0.40r$, $\bar{X} = r\cos(2\pi s) - 0.05$, $\bar{Y} = r\sin(2\pi s)$, and $r = 0.086 \times 2^s$.

Appendix B: Formulas for the mother pulses

The mother functions Φ that we used to construct the basis elements are defined by formulas (17) — (20) below.

$$\Phi_2^{\mathrm{S}}(r) = \begin{cases} 1 - \frac{4}{3}r^2 & \text{if} \quad |r| \le \frac{1}{2} \\ \frac{2}{3}(\frac{3}{2} - |r|)^2 & \text{if} \quad \frac{1}{2} < |r| < \frac{3}{2} \end{cases} \tag{17}$$

$$\Phi_3^{\mathrm{S}}(r) = \begin{cases} 1 - \frac{r^2}{4}(6 - 3|r|) & \text{if} \quad |r| \le 1 \\ \frac{(2-|r|)^3}{4} & \text{if} \quad 1 < |r| \le 2 \end{cases} \tag{18}$$

$$\Phi_4^{\mathrm{S}}(r) = \begin{cases} 1 - \frac{120}{115}r^2 + \frac{48}{115}r^4 & \text{if} \quad |r| \le \frac{1}{2} \\ \frac{22}{23} + \frac{8}{23}|r| - \frac{240}{115}r^2 + \frac{160}{115}|r|^3 - \frac{32}{115}r^4 & \text{if} \quad \frac{1}{2} < |r| \le \frac{3}{2} \\ \frac{16}{230}\Big(\frac{5}{2} - |r|\Big)^4 & \text{if} \quad \frac{3}{2} < |r| \le \frac{5}{2} \end{cases} \tag{19}$$

$$\Phi_5^{\mathrm{S}}(r) = \begin{cases} 1 - \frac{10}{11}r^2 + \frac{5}{11}r^4 - \frac{5}{33|r|^5} & \text{if} \quad |r| \le 1 \\ \frac{17}{22} + \frac{25}{22}|r| - \frac{35}{11}r^2 + \frac{25}{11}|r|^3 - \frac{15}{22}r^4 + \frac{5}{66}|r|^5 & \text{if} \quad 1 < |r| \le 2 \\ \frac{1}{66}(3 - |r|)^5 & \text{if} \quad 2 < |r| \le 3 \end{cases} \tag{20}$$

References

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

[2] Douglas Azevedo Castro. *AMultilevel approximation schemes and applications.* PhD thesis, Institute of Mathematics, Statistics and Scientific Computing, UNICAMP, October 2011. Advisor: Sônia Maria Gomes.

[3] C. de Boor. Splines as linear combinations of b-splines. *Approximation Theory II*, pages 1–47, 1976.

[4] C. de Boor. *A practical guide to splines.* Springer-Verlag, 1978.

[5] H. G. Feichtinger and T. Strohmer. Gabor analysis and algorithms. Springer, 1998.

[6] R. Finkel and J.L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.

[7] M. S. Floarter and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73:65–78, 1996.

[8] M. S. Floarter and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. In Tom Lyche and Larry L. Schumaker, editors, *Proceedings of the 2000 Conference on Mathematical Methods for Curves and Surfaces*, pages 211–220, 2000.

[9] A. Iske. Hierarchical scattered data filtering for multilevel interpolation schemes. In Tom Lyche and Larry L. Schumaker, editors, *Proceedings of the 2000 Conference on Mathematical Methods for Curves and Surfaces*, pages 211–220, 2000.

[10] A. Iske. *Multiresolution Methods in Scattered Data Modelling*. Springer, 2000.

[11] A. Iske and J. Levesley. Multilevel scattered data approximation by adaptive domain decomposition. *Numerical Algorithms*, 39:187–198, 2005.

[12] S. Müller. *Adaptive Multiscale Schemes for Conservation Laws*, volume 27 of *Lectures Notes in Computational Science and Engineering*. Springer, 2003.

[13] M. J. D Powell. *Advances in Numerical Analysis: Wavelets, Subdivision Algorithms, and Radial Basis Functions*. Oxford Science Publications.

[14] G. Wolberg S. Lee and S. Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE Transactions on visualization and computer graphics*, 3:228–244, 1997.

[15] I. J. Schoenberg. Cardinal interpolation and spline functions. *Approximation Theory II*, 2:167–206, 1969.

[16] G. R. Souza and J. Stolfi. Adaptive multiscale function approximation i: General discrete bases. *Advances in Computational Mathematics - Manuscript Number ACOM-D-16-00289*.

[17] G. R. Souza and J. Stolfi. Robust least squares by iterative bayesian data adjustment. *Institute of Mathematics, Statistics and Scientific Computing (IMECC), State University of Campinas (UNICAMP), Brazil*, Technical Report IC-15-06 - 2015.