UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Stabilizing the Hierarchical Basis by Approximate Wavelets, II: Implementation and Numerical Results

Panayot S. Vassilevski
Junping Wang

December 1995

CAM Report 95-48

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

Stabilizing the Hierarchical Basis by Approximate Wavelets, II: Implementation and Numerical Results

PANAYOT S. VASSILEVSKI[†] AND JUNPING WANG[‡]

Abstract. This paper is the second part of a work on stabilizing the classical hierarchical basis (HB) by using wavelet–like basis functions. Implementation techniques are discussed for the multilevel preconditioners proposed by the authors in the first part of the work [14]. Numerical results are presented to illustrate the convergence theory established in [14].

Key words. hierarchical basis, multilevel methods, preconditioning, finite element elliptic equations, approximate wavelets

AMS(MOS) subject classifications. 65F10, 65N20, 65N30

1. INTRODUCTION

In this paper we implement the wavelet-modified hierarchical basis method proposed in [14]. The method was based on the use of some approximate L^2–projections which stabilize the classical hierarchical basis ([16], [4]) by taking away from each hierarchical basis function its approximate L^2-projection on coarse levels. The modified or stabilized hierarchical basis shall be called AWM–HB (Approximate Wavelet Modified Hierarchical Basis). Applications in the multilevel preconditioning for finite element discretizations of elliptic partial differential equations are called AWM–HB methods.

The AWM–HB can be viewed as a stabilization of the HB in the sense that the resulting multilevel preconditioners are spectrally equivalent to the discretized elliptic operator for problems of two and three space variables. Other stabilizations of the HB methods, such as the AMLI methods presented earlier in [2] and [12], are not of V–cycle type (in the multigrid terminology), whereas the AWM–HB is of V–cycle type. The multiplicative AWM–HB method fits in the original framework as given in Vassilevski [11] (see also [12]) and the latter is a straightforward extension of the two–level method proposed by Bank and Dupont [3] and studied further by Axelsson and Gustafsson [1].

A survey on the subject of stabilizing the HB methods can be found in Vassilevski [13].

Other related (only partial) results in the use of L^2 and H^1 orthogonal direct decompositions for the finite element space can be found in Griebel and Oswald [7] and Stevenson [9, 10]. Our result is general; the AWM–HB method is of optimal order and it applies whenever standard HB decomposition of the finite element space exists.

Our objective in this paper is to implement the AWM–HB preconditioners. In particular, we propose some algorithms for computing the actions of the approximate L^2-projection operator Q^a_{k-1} on functions $v \in V^{(1)}_k$ (see Section 2.2 for details), where k indicates the number of levels with larger k corresponding to finer spaces. We also

[†] Center of Informatics and Computer Technology, Bulgarian Academy of Sciences, Acad. G. Bontchev street, Block 25 A, 1113 Sofia, Bulgaria. (panayot@iscbg.acad.bg). The work of the first author was partially supported by the U.S. National Science Foundation grant INT-95-06184 and also in part by Bulgarian Ministry for Education, Science and Technology under grant I-504, 1995.

[‡] Department of Mathematics, University of Wyoming, Laramie, WY 82071, USA. (wang@ schwarz.uwyo.edu). The second author was partially supported by the NSF grant INT-93-09286.

reformulate the algorithms of [14] in a matrix–vector form which, we believe, is suitable for practical computation.

The paper is organized as follows. In §2, we review the additive and multiplicative preconditioners arising from the AWM–HB. In §3, we discuss some examples of the approximate L^2-projection. In §4, we formulate the AWM–HB preconditioners in a matrix–vector form. In §5, we present some numerical results which illustrate the theory developed in the first part of this work [14].

2. Preliminaries

2.1. A model problem and its discretization. The bilinear form under consideration is given as follows:

$$a(\varphi, \psi) = \int_\Omega a \nabla \varphi \cdot \nabla \psi \qquad \forall \varphi,\ \psi \in H_0^1(\Omega). \tag{2.1}$$

Here $a = \{a_{ij}(x)\}$ is the coefficient matrix, which is assumed to be symmetric and positive definite uniformly in $x \in \Omega$ with bounded and measurable entries $a_{ij}(x)$.

To discretize the bilinear form $a(\cdot, \cdot)$, we use the routine successive (possibly local) refinement procedure to generate a sequence of finite element triangulations $\mathcal{T}_k$ for $k = 0, 1, \cdots, J$ with $\mathcal{T}_0$ being the initial triangulation. Let V_k be the conforming piecewise linear finite element space associated to $\mathcal{T}_k$. Denote by $A^{(k)} : V_k \to V_k$ the corresponding discretization of the bilinear form given by

$$(A^{(k)}\varphi, \psi) = a(\varphi, \psi) \qquad \forall \varphi, \psi \in V_k,$$

where $(\cdot, \cdot)$ stands for the standard L^2-inner product.

Each V_k is equipped with a standard Lagrangian (nodal) basis $\{\phi_i^{(k)},\ x_i \in \mathcal{N}_k\}$, where $\mathcal{N}_k$ is the node-set (the set of nodal degrees of freedom) of V_k. The basis functions satisfy $\phi_i^{(k)}(x_j) = \delta_{i,j}$ – the Kronecker symbol when x_j runs over the node-set $\mathcal{N}_k$. We assume that $\mathcal{N}_k \subset \mathcal{N}_{k+1}$. Due to the refinement process we also have $V_k \subset V_{k+1}$.

For the method to be defined we need the L^2-projection operators, $Q_k : L^2(\Omega) \to V_k$ defined in the standard way. To be more specific, for any $v \in L^2(\Omega)$, $Q_k v \in V_k$ is defined by solving the following mass (or Gramm) matrix problem:

$$(Q_k v, \psi) = (v, \psi) \quad \forall \psi \in V_k.$$

We also need the nodal interpolation operators $I_k : C(\bar{\Omega}) \to V_k$ given as follows:

$$I_k v = \sum_{x_i \in \mathcal{N}_k} v(x_i) \phi_i^{(k)}.$$

Finally, we need good approximations Q_k^a to Q_k that satisfy for some small tolerance $\tau \geq 0$, the estimate:

$$\|(Q_k - Q_k^a)v\|_0 \leq \tau \|Q_k v\|_0 \quad \forall v \in L^2(\Omega). \tag{2.2}$$

2.2. The AWM–HB preconditioners. The AWM–HB preconditioners exploit the following direct decomposition for each V_k:

$$V_k = (I - Q_{k-1}^a)(I_k - I_{k-1})V_k \oplus V_{k-1}.$$

By letting $V_k^1 = (I - Q_{k-1}^a)(I_k - I_{k-1})V_k = (I - Q_{k-1}^a)V_k^{(1)}$, where $V_k^{(1)} = (I_k - I_{k-1})V_k$ is the standard two–level hierarchical complement of V_{k-1} in the nearest fine space

V_k, the following direct decomposition of $V = V_J$ follows:

$$V = V_0 \oplus V_1^1 \oplus V_2^1 \oplus \ldots \oplus V_J^1.$$

Let $\mathcal{N}_k^{(1)} = \mathcal{N}_k \setminus \mathcal{N}_{k-1}$. The following set of functions

$$\{(I - Q_{k-1}^a)\phi_i^{(k)} \,:\, x_i \in \mathcal{N}_k^{(1)}\}, \tag{2.3}$$

forms a basis for V_k^1. The basis functions $\left\{(I - Q_{k-1}^a)\phi_i^{(k)}\right\}$ are clearly a modification of the classical HB functions of $V_k^{(1)}$. The modification was made by taking away from the HB function $\phi_i^{(k)}$ its approximate L^2–projection onto the nearest coarse space V_{k-1}.

The following operators are needed in the construction of the AWM–HB preconditioners:

- In each coordinate space V_k^1, we define the solution operators $A_{11}^{(k)} : V_k^1 \to V_k^1$ as the restriction of $A^{(k)}$ onto the subspace V_k^1; i.e.,

$$(A_{11}^{(k)}\psi^1, \varphi^1) = a(\psi^1, \varphi^1) \qquad \forall\varphi^1,\ \psi^1 \in V_k^1. \tag{2.4a}$$

- Similarly, we define $A_{12}^{(k)} : V_{k-1} \to V_k^1$ and $A_{21}^{(k)} : V_k^1 \to V_{k-1}$ by

$$(A_{12}^{(k)}\widetilde{\psi}, \varphi^1) = (\widetilde{\psi}, A_{21}^{(k)}\varphi^1) = a(\varphi^1, \widetilde{\psi}) \qquad \forall\widetilde{\psi} \in V_{k-1}, \varphi^1 \in V_k^1. \tag{2.4b}$$

Thus, the operator $A^{(k)}$ naturally admits the following two–by–two block structure:

$$A^{(k)} = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ A_{21}^{(k)} & A^{(k-1)} \end{bmatrix} \begin{matrix} \} & V_k^1 \\ \} & V_{k-1} \end{matrix}. \tag{2.4}$$

Let $B_{11}^{(k)}$ be given approximations (symmetric positive definite operators) to $A_{11}^{(k)}$ such that for some positive constant b_1 the following holds:

$$(A_{11}^{(k)}\varphi^1, \varphi^1) \le (B_{11}^{(k)}\varphi^1, \varphi^1) \le (1 + b_1)(A_{11}^{(k)}\varphi^1, \varphi^1) \qquad \forall\varphi^1 \in V_k^1. \tag{2.5}$$

Let $A = A^{(J)}$ be the operator of our main interest. Below, we define two major types of preconditioners B and D that exploit the two–by–two block structure of each $A^{(k)}$ in (2.4).

DEFINITION 1: MULTIPLICATIVE AWM–HB PRECONDITIONERS. Let $B = B^{(J)}$ be the multiplicative AWM–HB preconditioner of $A = A^{(J)}$. It is defined as follows:

- Set $B^{(0)} = A^{(0)}$.
- For $k = 1, \cdots, J$, set

$$B^{(k)} = \begin{bmatrix} B_{11}^{(k)} & 0 \\ A_{21}^{(k)} & B^{(k-1)} \end{bmatrix} \begin{bmatrix} I & B_{11}^{(k)^{-1}} A_{12}^{(k)} \\ 0 & I \end{bmatrix}.$$

DEFINITION 2: ADDITIVE AWM–HB PRECONDITIONERS. Let $D = D^{(J)}$ be the additive AWM–HB preconditioner of $A = A^{(J)}$. It is defined as follows:

- Set $D^{(0)} = A^{(0)}$.
- For $k = 1, \cdots, J$, set

$$D^{(k)} = \begin{bmatrix} B_{11}^{(k)} & 0 \\ 0 & D^{(k-1)} \end{bmatrix}.$$

2.3. Main results for the AWM–HB preconditioners. In the first part of this work [14] we have established a spectral equivalence between A and its preconditioners B and D. More precisely, the following result was derived:

$$c_1(Sv, v) \le (Av, v) \le c_2(Sv, v) \qquad \forall v \in V_J, \tag{2.6}$$

where $S = B^{(J)}$ or $D^{(J)}$. Here c_i are absolute constants independent of the mesh size h (or the level number J). The estimate (2.6) is based on the following assumptions:

(A) The tolerance τ in (2.2) must be sufficiently small, though independent of the mesh sizes h_i or the level number J;

(B) There exists a constant $\sigma_N > 0$ such that the following estimate holds:

$$|Q_0 v|_1^2 + \sum_{s=1}^{J} 2^{2s} \|(Q_s - Q_{s-1})v\|_0^2 \le \sigma_N \|v\|_1^2 \qquad \forall v \in V; \tag{a.i}$$

(C) There exist constants $\sigma_I > 0$ and $\delta \in (0, 1)$ (in fact, if $h_i = \frac{1}{2} h_{i-1}$, then $\delta = \frac{1}{\sqrt{2}}$) such that the following strengthened Cauchy–Schwarz inequality holds for any $i \le j$:

$$a(\varphi_i, \varphi_j)^2 \le \sigma_I \delta^{2(j-i)} a(\varphi_i, \varphi_i) \lambda_j \|\varphi_j\|_0^2 \qquad \forall \varphi_i \in V_i,\ \varphi_j \in V_j. \tag{a.ii}$$

Here $\lambda_j = O(h_j^{-2})$ is the largest eigenvalue of the operator $A^{(j)}$.

The assumptions (B) and (C) have been respectively verified by Oswald [8] and Yserentant [16, 17]. See, also [15], [5], and [6].

Another major fact, as shown in [14], is that the blocks $A_{11}^{(k)}$ are well–conditioned. In particular, they are spectrally equivalent to the diagonal part of their matrix representation with respect to the AWM-HB. Thus, the Jacobi preconditioner would be a good choice for $B_{11}^{(k)}$ (see (2.5)) in approximating $A_{11}^{(k)}$.

3. On the approximate L^2-projection

Let Q_{k-1}^a denote any approximate L^2-projections onto the subspace V_{k-1}. We begin with describing algorithms for computing the actions of Q_{k-1}^a.

For any $v \in V_k^{(1)}$, let $\mathbf{v} = \begin{array}{cl} \begin{bmatrix} \mathbf{v}_1 \\ 0 \end{bmatrix} & \begin{array}{l} \}\mathcal{N}_k \setminus \mathcal{N}_{k-1} \\ \}\mathcal{N}_{k-1} \end{array} \end{array}$ be its coefficient vector with respect to the standard nodal basis of V_k; the second block-component of $\mathbf{v}$ is zero since v vanishes on $\mathcal{N}_{k-1}$. The operator Q_{k-1}^a can be designed by approximately solving the following equation:

$$(Q_{k-1} v, w) = (v, w) \quad \forall w \in V_{k-1}. \tag{3.1}$$

Let $I_{k-1}^k = \begin{array}{cl} \begin{bmatrix} J_{12} \\ I \end{bmatrix} & \begin{array}{l} \}\mathcal{N}_k \setminus \mathcal{N}_{k-1} \\ \}\mathcal{N}_{k-1}. \end{array} \end{array}$ (with the abbreviation $J_{12} = J_{12}^{(k)}$) and $I_k^{k-1} = I_{k-1}^{k^T}$ be the natural coarse-to-fine, and respectively, fine-to-coarse transformation matrices. For example, if the nodal basis coefficient vector of a function $v_2 \in V_{k-1}$ in terms of the nodal basis of V_{k-1} is $\mathbf{v}_2$, then its coefficient vector with respect to the nodal basis of V_k (note that $v_2 \in V_{k-1} \subset V_k$) will be $I_{k-1}^k \mathbf{v}_2 = \begin{array}{cl} \begin{bmatrix} J_{12}\mathbf{v}_2 \\ \mathbf{v}_2 \end{bmatrix} & \begin{array}{l} \}\mathcal{N}_k \setminus \mathcal{N}_{k-1} \\ \}\mathcal{N}_{k-1} \end{array} \end{array}$.

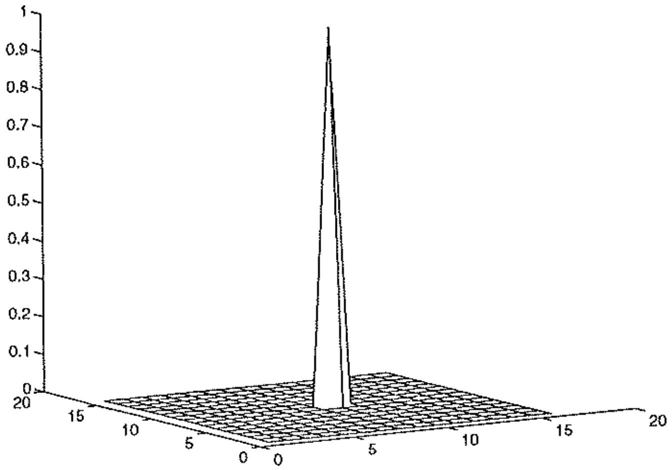

Figure 1. Plot of a HB function (no modification)

Denote now by $G_k = \{(\phi_j^{(k)}, \phi_i^{(k)})\}_{x_j, x_i \in \mathcal{N}_k}$ the mass (or Gramm) matrix at the k-th level. Then (3.1) admits the following matrix–vector form:

$$\mathbf{w}_2^T G_{k-1} \mathbf{v}_2 = (I_{k-1}^k \mathbf{w}_2)^T G_k \mathbf{v} \quad \forall \mathbf{w}_2 \cdot$$

Here $\mathbf{v}_2$ and $\mathbf{w}_2$ are, respectively, the nodal coefficient vectors of $Q_{k-1}v$ and $w \in V_{k-1}$ at the $(k-1)$th level. Therefore, one needs to solve the following mass matrix problem:

$$G_{k-1} \mathbf{v}_2 = I_k^{k-1} G_k \mathbf{v} \cdot \tag{3.2}$$

In other words, the exact L^2-projection $Q_{k-1}v$ is actually given by $G_{k-1}^{-1} I_k^{k-1} G_k \mathbf{v} \cdot$ Hence

$$\|Q_{k-1}v\|_0^2 = \left(G_{k-1}^{-1} I_k^{k-1} G_k \mathbf{v}\right)^T G_{k-1} \left(G_{k-1}^{-1} I_k^{k-1} G_k \mathbf{v}\right) = \|G_{k-1}^{-\frac{1}{2}} I_k^{k-1} G_k \mathbf{v}\|^2 . \tag{3.3}$$

Here and in what follows of this paper we use the notation $\|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{x}$.

To have a computationally feasible basis, we have to replace G_{k-1}^{-1} by some approximations $\widetilde{G}_{k-1}^{-1}$ whose action can be computed by simple iterative methods applied to (3.2). Such iterative methods lead to following polynomial approximations of G_{k-1}^{-1},

$$\widetilde{G}_{k-1}^{-1} = \left[I - \pi_m \left(G_{k-1}\right)\right] G_{k-1}^{-1},$$

where π_m is a polynomial of degree $m \geq 1$. The polynomial π_m also satisfies $\pi_m(0) = 1$ and $0 \leq \pi_m(t) < 1$ for $t \in [\alpha, \beta]$, where the latter interval contains the spectrum of the mass matrix G_{k-1}. Since G_{k-1} is well–conditioned, one can choose the interval $[\alpha, \beta]$ independent of k. Thus, the polynomial degree m can be chosen to be mesh-independent so that a given prescribed accuracy $\tau > 0$ in (2.2) is guaranteed. More precisely, given a tolerance $\tau > 0$, one can choose $m = m(\tau)$ satisfying

$$\begin{aligned} \|Q^a_{k-1}v - Q_{k-1}v\|_0 &= \|G^{\frac{1}{2}}_{k-1}\left(G^{-1}_{k-1} - \widetilde{G}^{-1}_{k-1}\right) I^{k-1}_k G_k \mathbf{v}\| \\ &= \|G^{\frac{1}{2}}_{k-1}\pi_m\left(G_{k-1}\right) G^{-1}_{k-1} I^{k-1}_k G_k \mathbf{v}\| \\ &\leq \max_{t\in[\alpha,\beta]} \pi_m(t) \|G^{-\frac{1}{2}}_{k-1} I^{k-1}_k G_k \mathbf{v}\| \\ &= \max_{t\in[\alpha,\beta]} \pi_m(t) \|Q_{k-1}v\|_0. \end{aligned}$$

Here we have used identity (3.3) and the properties of π_m. The last estimate implies the validity of (2.2) with

$$\tau \geq \max_{t\in[\alpha,\beta]} \pi_m(t).$$

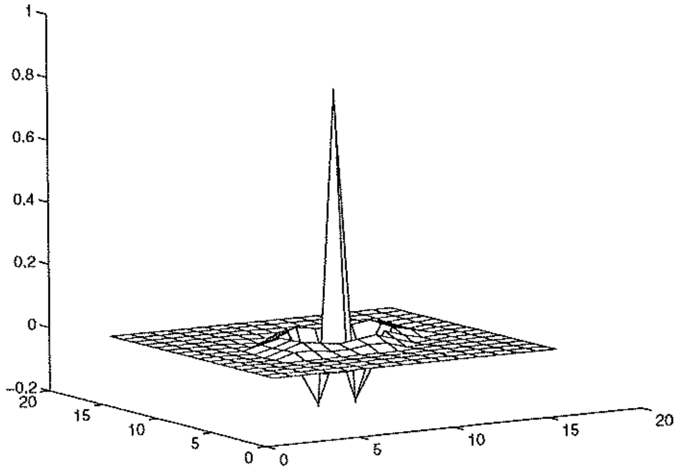

Figure 2. Plot of a wavelet modified HB function; $m = 2$

A simple choice of $\pi_m(t)$ is the truncated series

$$(1 - \pi_m(t))t^{-1} = p_{m-1}(t) \equiv \beta^{-1} \sum_{k=0}^{m-1} (1 - \frac{1}{\beta}t)^k, \tag{3.4}$$

which yields $\widetilde{G}^{-1}_{k-1} = p_{m-1}(G_{k-1})$. We remark that (3.4) was obtained from the following expansion:

$$1 = t\beta^{-1} \sum_{k=0}^{\infty} (1 - t\beta^{-1})^k, \qquad t \in [\alpha, \beta].$$

With the above choice on the polynomial $\pi_m(t)$, we have

$$\pi_m(t) = 1 - tp_{m-1}(t) = t\beta^{-1} \sum_{k\geq m} (1 - \beta^{-1}t)^k = (1 - \beta^{-1}t)^m.$$

It follows that

$$\max_{t\in[\alpha,\beta]} \pi_m(t) = \left(1-\frac{\alpha}{\beta}\right)^m .$$

In general, by a careful selection on π_m we have $\max\limits_{t\in[\alpha,\beta]} \pi_m(t) \leq Cq^m$ for some constants $C > 0$ and $q \in (0,1)$, both independent of k. Since the restriction on τ was that τ be sufficiently small, then one must have

$$m = O(\log \tau^{-1}). \tag{3.5}$$

The requirement (3.5) obviously imposes a very mild restriction on m. In practice, one expects to use reasonably small m (e.g., $m = 1, 2$). This observation is confirmed by our numerical experiments to be presented in Section 5.

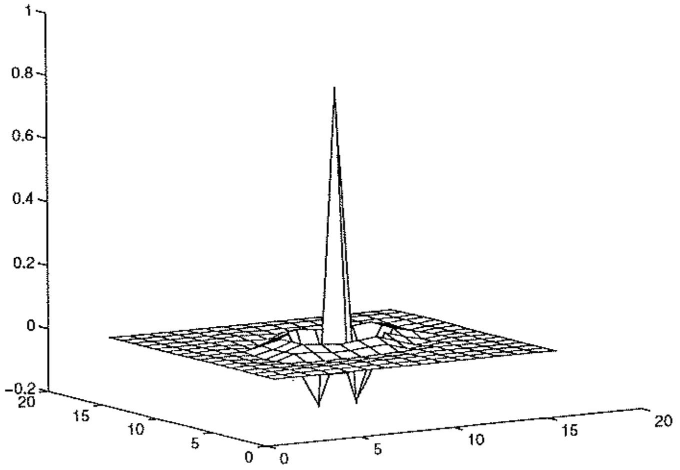

Figure 3. Plot of a wavelet modified HB function; $m = 4$

We show in Fig. 1 a typical plot of a nodal basis function of $V_k^{(1)}$ and its approximate wavelet modification for $m = 2$ in Fig. 2 and for $m = 4$ in Fig. 3. The cross section, for $m = 4$, is shown in Fig. 4. The conjugate gradient method was employed to provide approximations for the solution of the mass–matrix problem (3.2).

4. Matrix-vector representations of the AWM–HB methods

We now turn to the description of the multiplicative and additive AWM–HB methods in a matrix–vector form.

Let us first derive matrix representations for the operators $A_{11}^{(k)}$, $A_{12}^{(k)}$, and $A_{21}^{(k)}$ introduced in Section 2 (see (2.4a) and (2.4b)). In what follows of this section, capital letters without overhats will denote matrices corresponding to the standard nodal basis of the underlined finite element space. For example, $A^{(k)}$ denotes the standard nodal basis stiffness matrix with entries $\{a(\phi_i^{(k)}, \phi_j^{(k)})\}_{x_i,x_j\in\mathcal{N}_k}$.

For any $v \in V_k$ and its nodal coefficient vector $\mathbf{v}$, we decompose v as follows:

$$v = (I - Q^a_{k-1})(I_k - I_{k-1})v + w_2,$$

where $w_2 \in V_{k-1}$ is uniquely determined as $w_2 = I_{k-1}v + Q^a_{k-1}(I_k - I_{k-1})v$. Our goal is to find a vector representation for components of v. Since the above decomposition is direct, it is clear that there are vectors $\widehat{\mathbf{v}}_1$ and $\widehat{\mathbf{v}}_2$ satisfying

$$\mathbf{v} = (I - I^k_{k-1}\widetilde{G}^{-1}_{k-1}I^{k-1}_k G_k)\begin{bmatrix}\widehat{\mathbf{v}}_1\\ 0\end{bmatrix}\begin{matrix}\}\mathcal{N}_k \setminus \mathcal{N}_{k-1}\\ \}\mathcal{N}_{k-1}\end{matrix} + I^k_{k-1}\widehat{\mathbf{v}}_2. \tag{4.1}$$

The vectors $\widehat{\mathbf{v}}_1$ and $\widehat{\mathbf{v}}_2$ represent the two components of our wavelet-modified two–level HB coefficient vector $\widehat{\mathbf{v}} = \begin{bmatrix}\widehat{\mathbf{v}}_1\\ \widehat{\mathbf{v}}_2\end{bmatrix}$ of v.

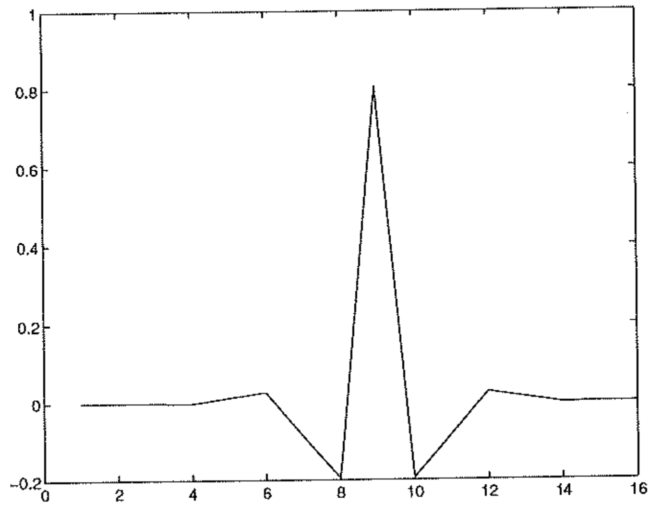

Figure 4. Cross-section plot of a wavelet-modified HB function, $m = 4$

Now, consider the following problem

$$A\mathbf{v} = \mathbf{d}, \tag{4.2}$$

which is in the standard nodal basis matrix–vector form. We transform it into the approximate wavelet modified two–level HB by testing (4.2) with the two components $(I - I^k_{k-1}\widetilde{G}^{-1}_{k-1}I^{k-1}_k G_k)\begin{bmatrix}\widehat{\mathbf{w}}_1\\ 0\end{bmatrix}$ and $I^k_{k-1}\widehat{\mathbf{w}}_2$ for arbitrary $\widehat{\mathbf{w}}_1$ and $\widehat{\mathbf{w}}_2$. By doing so, we get the following two–by–two block system for the approximate wavelet modified two–level HB components of $\widehat{\mathbf{v}}$ (denoted by $\widehat{\mathbf{v}}_1$ and $\widehat{\mathbf{v}}_2$),

$$\begin{bmatrix}\widehat{A}^{(k)}_{11} & \widehat{A}^{(k)}_{12}\\ \widehat{A}^{(k)}_{21} & \widehat{A}^{(k)}_{22}\end{bmatrix}\begin{bmatrix}\widehat{\mathbf{v}}_1\\ \widehat{\mathbf{v}}_2\end{bmatrix} = \begin{bmatrix}\widehat{\mathbf{d}}_1\\ \widehat{\mathbf{d}}_2\end{bmatrix}, \tag{4.3}$$

where

$$\begin{aligned}
\widehat{A}_{11}^{(k)} &= [I \quad 0]\left(I - G_k I_{k-1}^{k} \widetilde{G}_{k-1}^{-1} I_k^{k-1}\right) A^{(k)} \left(I - I_{k-1}^{k} \widetilde{G}_{k-1}^{-1} I_k^{k-1} G_k\right) \begin{bmatrix} I \\ 0 \end{bmatrix};\\
\widehat{A}_{12}^{(k)} &= [I \quad 0]\left(I - G_k I_{k-1}^{k} \widetilde{G}_{k-1}^{-1} I_k^{k-1}\right) A^{(k)} I_{k-1}^{k};\\
\widehat{A}_{21}^{(k)} &= I_k^{k-1} A^{(k)} \left(I - I_{k-1}^{k} \widetilde{G}_{k-1}^{-1} I_k^{k-1} G_k\right) \begin{bmatrix} I \\ 0 \end{bmatrix};\\
\widehat{A}_{22}^{(k)} &= I_k^{k-1} A^{(k)} I_{k-1}^{k} = A^{(k-1)}.
\end{aligned}$$

Note that having computed $\widehat{\mathbf{v}}_1$ and $\widehat{\mathbf{v}}_2$, the solution $\mathbf{v}$ of (4.2) can be recovered by using the formula (4.1), i.e.,

$$\mathbf{v} = Y_1 \widehat{\mathbf{v}}_1 + Y_2 \widehat{\mathbf{v}}_2,$$

where,

$$\begin{aligned}
Y_1 &= \left(I - I_{k-1}^{k} \widetilde{G}_{k-1}^{-1} I_k^{k-1} G_k\right) \begin{bmatrix} I \\ 0 \end{bmatrix},\\
Y_2 &= I_{k-1}^{k}.
\end{aligned}$$

We have,

$$\mathbf{v} = Y \widehat{\mathbf{v}}, \quad \widehat{\mathbf{v}} = \begin{bmatrix} \widehat{\mathbf{v}}_1 \\ \widehat{\mathbf{v}}_2 \end{bmatrix}, \ Y = [Y_1, \ Y_2], \ Y_1 = Y_1^{(k)}, \ Y_2 = Y_2^{(k)}.$$

The transformed right–hand side vectors of (4.3) read similarly as follows:

$$\begin{aligned}
\widehat{\mathbf{d}}_1 &= [I \quad 0]\left(I - G_k I_{k-1}^{k} \widetilde{G}_{k-1}^{-1} I_k^{k-1}\right) \mathbf{d} = Y_1^T \mathbf{d},\\
\widehat{\mathbf{d}}_2 &= I_k^{k-1} \mathbf{d} = Y_2^T \mathbf{d}\cdot
\end{aligned}$$

Therefore, the multiplicative AWM–HB preconditioner $B^{(k)}$ from Definition 1, starting with $B^{(0)} = A^{(0)}$, takes the following block–matrix form:

$$\widehat{B}^{(k)} = \begin{bmatrix} \widehat{B}_{11}^{(k)} & 0 \\ \widehat{A}_{21}^{(k)} & B^{(k-1)} \end{bmatrix} \begin{bmatrix} I & \widehat{B}_{11}^{(k)^{-1}} \widehat{A}_{12}^{(k)} \\ 0 & I \end{bmatrix}. \tag{4.4}$$

The preconditioner $B^{(k)}$ is related to $\widehat{B}^{(k)}$ in the same way as $A^{(k)}$ to $\widehat{A}^{(k)}$; namely, $\widehat{B}^{(k)} = [Y_1, Y_2]^T B^{(k)} [Y_1, Y_2]$ and therefore, $B^{(k)^{-1}} = [Y_1, Y_2] \widehat{B}^{(k)^{-1}} [Y_1, Y_2]^T$. We will show below that the inverse actions of $B^{(k)}$ can be computed only via the actions of $A^{(k)}$, Y_1, Y_2, and Y_1^T, Y_2^T in addition to the inverse actions of $\widehat{B}_{11}^{(k)}$.

We point out that (4.4) has precisely the same form as the algebraic multilevel method studied in Vassilevski [11] (see also Axelsson and Vassilevski [2] and Vassilevski [12]).

Observe that, in (4.4), $\widehat{B}_{11}^{(k)}$ is an appropriately scaled approximation of $\widehat{A}_{11}^{(k)}$. We have shown in [14] that $\widehat{A}_{11}^{(k)}$ is well–conditioned (see Lemma 4.3 there). Thus, it is possible to take some simple polynomial approximation $\widehat{B}_{11}^{(k)}$ for $\widehat{A}_{11}^{(k)}$ in the implementation. However, in order to take into account any possible jumps in the coefficient of the differential operator, it would be preferable to compute, for example, the diagonal part of $\widehat{A}_{11}^{(k)}$. This is computationally feasible since the basis functions of $V_k^1 = (I - Q_{k-1}^a) V_k^{(1)}$ (given in (2.3)) have reasonably narrow support if m is not too large, which should be the case in practice. Nevertheless, we employed in our numerical experiments the CG method to compute (fairly accurate) approximate actions of $\widehat{A}_{11}^{(k)^{-1}}$.

ALGORITHM I: COMPUTING INVERSE ACTIONS OF $B^{(k)}$. *The inverse actions of $B^{(k)}$ are computed by solving the system*

$$B^{(k)}\mathbf{w} = \mathbf{d},$$

with the change of basis $\mathbf{w} = Y\widehat{\mathbf{w}}$. *Namely, by setting*

$$\begin{aligned} \mathbf{w} &= Y_1\widehat{\mathbf{w}}_1 + Y_2\widehat{\mathbf{w}}_2 = [Y_1,\ Y_2]\begin{bmatrix}\widehat{\mathbf{w}}_1\\ \widehat{\mathbf{w}}_2\end{bmatrix}, \\ \widehat{\mathbf{d}}_1 &= Y_1^T\mathbf{d}, \\ \widehat{\mathbf{d}}_2 &= Y_2^T\mathbf{d}, \end{aligned}$$

$\mathbf{w} = B^{(k)^{-1}}\mathbf{d}$ *is computed via the solution of* $\widehat{B}^{(k)}\widehat{\mathbf{w}} = \widehat{\mathbf{d}}$ *as follows:*

- FORWARD RECURRENCE:

(1) *compute* $\widehat{\mathbf{z}}_1 = \widehat{B}_{11}^{(k)^{-1}}\widehat{\mathbf{d}}_1$;
(2) *change the basis; i.e., compute* $\mathbf{z} = Y_1\widehat{\mathbf{z}}_1$;
(3) *compute* $\widehat{\mathbf{d}}_2 := \widehat{\mathbf{d}}_2 - \widehat{A}_{21}^{(k)}\widehat{\mathbf{z}}_1 = Y_2^T(\mathbf{d} - A^{(k)}\mathbf{z})$;
(4) *compute* $\widehat{\mathbf{w}}_2 = B^{(k-1)^{-1}}\widehat{\mathbf{d}}_2$;
(5) *change the basis, i.e., compute* $\mathbf{v} = Y_2\widehat{\mathbf{w}}_2$;

- BACKWARD RECURRENCE:

(1) *update the fine–grid residual, i.e., compute*

$$\widehat{\mathbf{d}}_1 := \widehat{\mathbf{d}}_1 - \widehat{A}_{12}^{(k)}\widehat{\mathbf{w}}_2 = Y_1^T(\mathbf{d} - A^{(k)}Y_2\widehat{\mathbf{w}}_2) = Y_1^T(\mathbf{d} - A^{(k)}\mathbf{v});$$

(2) *compute* $\widehat{\mathbf{w}}_1 = \widehat{B}_{11}^{(k)^{-1}}\widehat{\mathbf{d}}_1$;
(3) *get the solution by* $\mathbf{w} = Y_1\widehat{\mathbf{w}}_1 + Y_2\widehat{\mathbf{w}}_2 = Y_1\widehat{\mathbf{w}}_1 + \mathbf{v}$.
End

Note that the above algorithm requires only the actions of the standard stiffness matrix $A^{(k)}$, the actions of the transformation matrices Y_1 and Y_2 and their transposition Y_1^T and Y_2^T, the inverse action of $\widehat{B}_{11}^{(k)}$, and some suitable approximations to the well–conditioned matrices $\widehat{A}_{11}^{(k)}$. Note that the actions of Y^{-1} are not required in the algorithm.

We now formulate the solution procedure for one preconditioning step using the multiplicative AWM–HB preconditioner $B = B^{(J)}$.

ALGORITHM II: MULTIPLICATIVE AWM–HB PRECONDITIONING. *Given the problem*

$$B\mathbf{v} = \mathbf{d}.$$

Initiate:

$$\mathbf{d}^{(J)} = \mathbf{d}.$$

(A) **Forward recurrence.** *For $k = J$ down to 1 perform:*
(1) *Compute:*

$$\widehat{\mathbf{d}}_1^{(k)} = [I \quad 0]\left(I - G_k I_{k-1}^k \widetilde{G}_{k-1}^{-1} I_k^{k-1}\right)\mathbf{d}^{(k)};$$

(2) *Solve:*

$$\widehat{B}_{11}^{(k)}\widehat{\mathbf{w}}_1 = \widehat{\mathbf{d}}_1^{(k)};$$

(3) *Transform basis:*

$$\mathbf{w} = \left(I - I_{k-1}^{k}\widetilde{G}_{k-1}^{-1}I_{k}^{k-1}G_{k}\right)\begin{bmatrix}\widehat{\mathbf{w}}_1 \\ 0\end{bmatrix}\begin{matrix}\}\mathcal{N}_k \setminus \mathcal{N}_{k-1} \\ \}\mathcal{N}_{k-1}\end{matrix};$$

(4) *Coarse–grid defect restriction:*

$$\begin{aligned}\mathbf{d}^{(k-1)} &= I_k^{k-1}\mathbf{d}^{(k)} - \widehat{A}_{21}^{(k)}\widehat{\mathbf{w}}_1 \\ &= I_k^{k-1}(\mathbf{d}^{(k)} - A^{(k)}\mathbf{w});\end{aligned}$$

(5) *Set $k = k-1$. If $k > 0$ go to (1), else:*

(6) *Solve on the* **coarsest level***:*

$$A^{(0)}\mathbf{x}^{(0)} = \mathbf{d}^{(0)};$$

(B) **Backward recurrence.**

(1) *Interpolate result: Set $k := k+1$ and compute*

$$\mathbf{x}^{(k)} = I_{k-1}^{k}\mathbf{x}^{(k-1)};$$

(2) *Update fine–grid residual:*

$$\begin{aligned}\widehat{\mathbf{d}}_1^{(k)} &:= \widehat{\mathbf{d}}_1^{(k)} - \widehat{A}_{12}^{(k)}\mathbf{x}^{(k-1)} \\ &= \widehat{\mathbf{d}}_1^{(k)} - [I \quad 0](I - G_k I_{k-1}^{k}\widetilde{G}_{k-1}^{-1}I_k^{k-1})A^{(k)}\mathbf{x}^{(k)} \\ &= [I \quad 0](I - G_k I_{k-1}^{k}\widetilde{G}_{k-1}^{-1}I_k^{k-1})(\mathbf{d}^{(k)} - A^{(k)}\mathbf{x}^{(k)});\end{aligned}$$

(3) *Solve:*

$$\widehat{B}_{11}^{(k)}\widehat{\mathbf{w}}_1 = \widehat{\mathbf{d}}_1^{(k)};$$

(4) *Change the basis:*

$$\mathbf{w} = (I - I_{k-1}^{k}\widetilde{G}_{k-1}^{-1}I_{k}^{k-1}G_{k})\begin{bmatrix}\widehat{\mathbf{w}}_1 \\ 0\end{bmatrix};$$

(5) *Finally set:*

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k)} + \mathbf{w}.$$

(6) *Set $k := k+1$. If $k < J$ go to step (1) of (B), else* set

$$\mathbf{v} = \mathbf{x}^{(J)}.$$

END

Similarly, one preconditioning solution step for the additive AWM–HB preconditioner $D = D^{(J)}$ takes the following form:

ALGORITHM III: ADDITIVE AWM–HB PRECONDITIONING. *Given the problem*

$$D\mathbf{v} = \mathbf{d}\cdot$$

Initiate:

$$\mathbf{d}^{(J)} = \mathbf{d}\cdot$$

(A) **Forward recurrence.** *For $k = J$ down to 1 perform:*

(1) *Compute:*

$$\widehat{\mathbf{d}}_1^{(k)} = [I \quad 0]\left(I - G_k I_{k-1}^{k}\widetilde{G}_{k-1}^{-1}I_k^{k-1}\right)\mathbf{d}^{(k)};$$

(2) *Solve:*

$$\widehat{B}_{11}^{(k)}\widehat{\mathbf{w}}_1 = \widehat{\mathbf{d}}_1^{(k)};$$

(3) *Transform basis:*

$$\mathbf{x}^{(k)} = \left(I - I_{k-1}^{k}\widetilde{G}_{k-1}^{-1}I_k^{k-1}G_k\right)\begin{bmatrix}\widehat{\mathbf{w}}_1\\ 0\end{bmatrix}\begin{matrix}\}\mathcal{N}_k\setminus\mathcal{N}_{k-1}\\ \}\mathcal{N}_{k-1}\end{matrix};$$

(4) *Coarse-grid defect restriction:*

$$\mathbf{d}^{(k-1)} = I_k^{k-1}\mathbf{d}^{(k)};$$

(5) *Set $k = k - 1$. If $k > 0$ go to (1), else :*

(6) *Solve on the* **coarsest level***:*

$$A^{(0)}\mathbf{x}^{(0)} = \mathbf{d}^{(0)};$$

(B) **Backward recurrence.**

(1) *Interpolate result: Set $k := k + 1$ and compute*

$$\mathbf{w} = I_{k-1}^{k}\mathbf{x}^{(k-1)};$$

(2) *Update at level k:*

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k)} + \mathbf{w};$$

(3) *Set $k := k + 1$. If $k < J$ go to step (1) of (B), else* set

$$\mathbf{v} = \mathbf{x}^{(J)}.$$

END

For both the additive and multiplicative preconditioners, it is readily seen that the above implementations require only actions of the stiffness matrices $A^{(k)}$, the mass matrices $G^{(k)}$, and the transformation matrices I_{k-1}^{k} and I_k^{k-1}. The approximate inverse actions of $\widehat{A}_{11}^{(k)}$ can be computed via some inner iterative algorithms. Similarly, the action of $\widetilde{G}_{k-1}^{-1}$ can be computed as approximate solutions of the corresponding mass–matrix problem using m steps of some simple iterative methods. Therefore, at each discretization level k, one performs a number of arithmetic operations proportional to the degrees of freedom at that level denoted by $\mathfrak{N}_k$. In the case of local mesh refinement, the corresponding operations involve only the stiffness and mass matrices computed for the subdomains where local refinement was made. Hence, even in the case of locally refined meshes, the cost of the AWM–HB methods is proportional to $\mathfrak{N} = \mathfrak{N}_J$. The proportionality constant depends linearly on $m = O(\log\tau^{-1})$, but is independent of J (or h).

The solution vector $\mathbf{v}$ in the additive algorithm is given by an expression of the form,

$$\mathbf{v} = D^{-1}\mathbf{d} = \left(R_0^T A^{(0)^{-1}} R_0 + \sum_{k=1}^{J} R_k^T B_{11}^{(k)^{-1}} R_k\right)\mathbf{d}, \tag{4.5}$$

where the matrices R_k are given by

$$\begin{aligned} R_0^T &= I_0^J, \quad \text{for } k = 0, \\ R_k^T &= I_k^J\left[I - I_{k-1}^{k}\widetilde{G}_{k-1}^{-1}I_k^{k-1}G_k\right]\begin{bmatrix} I\\ 0\end{bmatrix}, \quad \text{for } k \geq 1\cdot \end{aligned} \tag{4.6}$$

Here I_k^J stands for the natural coarse–to–fine transfer matrix from level k to the finest level.

It is clear from (4.5) and (4.6) that the additive version of the algorithm can be implemented in a parallel fashion, but this will require $O\left(\mathfrak{N}\log\mathfrak{N}\log\tau^{-1}\right)$ operations versus $O\left(\mathfrak{N}\log\tau^{-1}\right)$ operations in the consecutive implementation which is due to the more expensive simultaneous transfer of the same data from the finest grid directly to all the kth level grids for $k = J-1,\ldots,0$ and vice versa.

5. Numerical experiments

In this section we present some numerical results which demonstrate the performance of the additive and multiplicative AWM–HB algorithms described in Section 4.

For simplicity, the domain Ω was taken to be the unit square $(0,1)^2$. Also, the finite element spaces V_k contain piecewise linear continuous functions that vanish on $\Gamma_D \equiv \{(x,0):\ 0<x<1\}\cup\{(0,y):\ 0<y<1\}$. The spaces V_k correspond to uniform triangulations of Ω consisting of isosceles right triangles of size $h_k = 2^{-k}$ for $k = 0,1,2,\ldots,J$. The mass matrix problem involved in both algorithms are solved by applying $m \geq 0$ steps of the CG method (without preconditioning). The problem with $\widehat{A}_{11}^{(k)}$ was solved by the CG method until a prescribed residual tolerance is reached. That is, one may assume that the actions of $A_{11}^{(k)^{-1}}$ are practically exact. The diffusion coefficient $a = a(x,y)$ in the bilinear form (2.1) was given by

$$a(x,y) = 1 + x^2 + y^2 \cdot$$

In the test we varied the number of inner iterations $m = 0,2,4$ for solving the mass matrix problem in order to compute the actions of Q_{k-1}^a required in the AWM–HB methods. The multiplicative method with $m = 0$ corresponds to the method of Vassilevski [11], which coincides with the HB–MG method of Bank, Dupont, and Yserentant [4]. The additive method with $m = 0$ is then a variant of the HB method of Yserentant [16].

Table 1
HB Multilevel Preconditioners; (m = 0)

levels	*Additive*				*Multiplicative*			
J	λ_{min}	λ_{max}	ρ	*iter*	λ_{min}	λ_{max}	ρ	*iter*
3	0.462	5.167	0.435	25	1.000	2.677	0.127	10
4	0.396	7.674	0.566	38	1.000	3.459	0.234	14
5	0.358	10.52	0.640	48	1.000	4.433	0.298	17
6	0.333	13.26	0.690	59	1.000	5.522	0.347	19
7	0.316	16.09	0.726	69	1.000	6.732	0.383	22

In the Tables 1–3 we show the number of iterations, *iter*, in the preconditioned conjugate gradient method applied to solving

$$A\mathbf{x} = \mathbf{b},$$

$A = A^{(J)}$ for $J = 3, 4, 5, 6, 7$ (i.e., the mesh-size $h = \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$) with the same m in a given table. The right–hand side vector $\mathbf{b}$ was chosen to satisfy a prescribed solution $u(x, y)$.

The stopping criterion used is

$$\mathbf{r}^T W^{-1}\mathbf{r} \leq 10^{-18}\mathbf{r}_0^T W^{-1}\mathbf{r}_0,$$

where W is the preconditioner (i.e., B or D), $\mathbf{r}$ is the current residual, and $\mathbf{r}_0 = (I - AW^{-1})\mathbf{b}$ is the initial residual. We also show in the tables the following average convergence rate $\rho = \left[\sqrt{\frac{\mathbf{r}^T W^{-1}\mathbf{r}}{\mathbf{r}_0^T W^{-1}\mathbf{r}_0}}\right]^{\frac{1}{iter}}$.

Information on the minimum ($\lambda_{\min}$) and maximum ($\lambda_{\max}$) eigenvalues of ${A^{(k)}}^{-1}B^{(k)}$ and ${A^{(k)}}^{-1}D^{(k)}$ for $k = 3, \ldots, J$ can also be found in the tables. The Lanczos method was employed in the code to provide this information.

Notice that, in Tables 2–3, the number of iterations (as well as the estimated extreme eigenvalues) tends to be uniformly bounded from above. This is very well seen from the columns for the multiplicative AWM–HB preconditioner in Table 3. Thus, as proved in [16], the choice of m is practically not affected by J. For example, $m = 2$ (see Table 2) gives also weakly sensitive values of the number of iterations (as well as eigenvalues) when J varies from 3 to 7. An improvement over the pure HB method (see Table 1) is clearly demonstrated in this test. It is expected that a much better improvement can be seen for problems of three space variables. To conclude, the numerical tests do illustrate the convergence theory presented in the first part of this work.

TABLE 2

AWM–HB Multilevel Preconditioners; $m = 2$

levels	*Additive*				*Multiplicative*			
J	λ_{min}	λ_{max}	ρ	*iter*	λ_{min}	λ_{max}	ρ	*iter*
3	0.542	2.846	0.375	21	0.972	1.577	0.118	10
4	0.481	3.395	0.466	28	0.990	1.711	0.143	11
5	0.443	3.564	0.486	30	0.990	1.798	0.156	11
6	0.418	3.674	0.499	31	0.989	1.832	0.157	11
7	0.401	3.698	0.505	32	0.989	1.877	0.156	12

TABLE 3
AWM–HB Multilevel Preconditioners; $m = 4$

levels	*Additive*				*Multiplicative*			
J	λ_{min}	λ_{max}	ρ	*iter*	λ_{min}	λ_{max}	ρ	*iter*
3	0.544	2.862	0.364	21	0.997	1.572	0.098	9
4	0.481	3.393	0.447	26	0.999	1.724	0.130	10
5	0.442	3.633	0.668	28	0.998	1.808	0.143	11
6	0.417	3.722	0.484	30	0.999	1.856	0.147	11
7	0.399	3.769	0.498	32	0.999	1.905	0.147	11

BIBLIOGRAPHY

[1] O. AXELSSON AND I. GUSTAFSSON, *Preconditioning and two–level multigrid methods of arbitrary degree of approximations*, Math. Comp., 40 (1983), pp. 219–242.

[2] O. AXELSSON AND P.S. VASSILEVSKI, *Algebraic multilevel preconditioning methods, II*, SIAM J. Numer. Anal., 27 (1990), pp. 1569–1590.

[3] R.E. BANK AND T. DUPONT, *Analysis of a two–level scheme for solving finite element equations*, Report CNA–159, Center for Numerical Analysis, The University of Texas at Austin (1980).

[4] R.E. BANK, T. DUPONT AND H. YSERENTANT, *The hierarchical basis multigrid method*, Numer. Math., 52 (1988), pp. 427–458.

[5] F. BORNEMANN AND H. YSERENTANT, *A basic norm equivalence for the theory of multilevel methods*, Numer. Math., 64 (1993), pp. 455–476.

[6] W. DAHMEN AND A. KUNOTH, *Multilevel preconditioning*, Numer. Math., 63 (1992), pp. 315–344.

[7] M. GRIEBEL AND P. OSWALD, *Tensor product type subspace splittings and multilevel iterative methods for anisotropic problems*, Report No. 342, Institut für Informatik, TU Munich, pp. 1994.

[8] P. OSWALD, *Multilevel Finite Element Approximation. Theory and Applications. Teubner Skripten zur Numerik*, Teubner, Stuttgart, 1994.

[9] R. STEVENSON, *Robustness of the additive and multiplicative frequency decomposition multilevel method*, Computing, 54 (1995), pp. 331–346.

[10] R. STEVENSON, *A robust hierarchical basis preconditioner on general meshes*, Report # 9533, Department of Mathematics, University of Nijmegen, Nijmegen, The Netherlands, pp. 1995.

[11] P.S. VASSILEVSKI, *Nearly optimal iterative methods for solving finite element elliptic equations based on the multilevel splitting of the matrix*, Report # 1989–09, Institute for Scientific Computation, University of Wyoming, Laramie, WY, USA, 1989.

[12] P.S. VASSILEVSKI, *Hybrid V–cycle algebraic multilevel preconditioners*, Math. Comp., 58 (1992), pp. 489–512.

[13] P.S. VASSILEVSKI, *On two ways of stabilizing the hierarchical basis multilevel methods*, CAM Report # 1995–06, pp. Department of Mathematics, University of California at Los Angeles, CA, USA, 1995.

[14] P.S. VASSILEVSKI AND J. WANG, *Stabilizing the hierarchical basis by approximate wavelets, I: Theory*, Numer. Linear Alg. Appl. (submitted).

[15] J. XU, *Iterative methods by space decomposition and subspace correction*, SIAM Review, 34 (1992), pp. 581–613.

[16] H. YSERENTANT, *On the multilevel splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.

[17] H. YSERENTANT, *Old and new convergence proofs for multigrid algorithms*, Acta Numerica (1993), pp. 285–326.