Available online at www.sciencedirect.com
ScienceDirect
C. R. Acad. Sci. Paris, Ser. I 345 (2007) 107–112
COMPTES RENDUS MATHEMATIQUE
http://france.elsevier.com/direct/CRASS1/

Numerical Analysis

Accelerating the method of finite element patches using approximately harmonic functions

Jiwen He [a], Alexei Lozinski [a,1], Jacques Rappaz [b]

[a] *Dept. of Mathematics, University of Houston, 4800 Calhoun Road, Houston, TX 77204-3008, USA*
[b] *Section of Mathematics, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland*

Received 11 May 2007; accepted 5 June 2007

Available online 13 July 2007

Presented by Olivier Pironneau

Abstract

We present a new variant of a domain decomposition method with complete overlap and not necessarily nested grids to solve numerically elliptic problems with multi-scale data. The novelty of the method consists in restricting finite element functions on the coarse grid to be approximately harmonic inside the subdomain where a finer triangulation is applied. Numerical experiments confirm an increase in the convergence rate over a previously proposed method. ***To cite this article: J. He et al., C. R. Acad. Sci. Paris, Ser. I 345 (2007).***

Résumé

Acceleration de la méthode des patches d'éléments finis en utilisant des fonctions approximativement harmoniques. Nous présentons une nouvelle variante de la méthode de décomposition de domaines avec un recouvrement complet et des maillages non nécessairement emboîtés pour la résolution numérique des problèmes elliptiques avec des données multi-échelle. La nouveauté de la méthode consiste dans la restriction de l'espace des fonctions éléments finis grossières à être des approximations des fonctions harmoniques dans le sous-domaine sur lequel une triangulation fine est appliquée. Des expériences numériques confirment une augmentation du taux de convergence par rapport à une méthode proposée précédemment. ***Pour citer cet article : J. He et al., C. R. Acad. Sci. Paris, Ser. I 345 (2007).***

Version française abrégée

Soit $\Omega \subset \mathbb{R}^2$ un domaine polygonal borné et $\Lambda \Subset \Omega$ un autre domaine polygonal « beacoup plus petit » que Ω. On considère le problème de trouver $u \in H_0^1(\Omega)$ tel que

$$a(u, v) = \langle f | v \rangle, \quad \forall v \in H_0^1(\Omega), \tag{1}$$

E-mail addresses: jiwenhe@math.uh.edu (J. He), alozinski@gmail.com (A. Lozinski), jacques.rappaz@epfl.ch (J. Rappaz).
[1] Present address: EPFL SB IACS ASN, bâtiment MA, Station 8, CH-1015 Lausanne, Switzerland.

1631-073X/$ – see front matter
doi:10.1016/j.crma.2007.06.006

où $a(u,v)=\int_\Omega \nabla u\cdot\nabla v\,\mathrm{d}\mathbf{x}$ et $f\in H^{-1}(\Omega)$ est une fonction donnée. On suppose que la solution u varie brutalement dans Λ de sorte que l'on ne puisse pas discrétiser le problème par une méthode d'éléments finis sur une grille qui est de même résolution sur $\bar\Omega\setminus\Lambda$ que sur $\bar\Lambda$. On introduit donc deux espaces d'éléments finis : V_H – l'espace des fonctions continues, s'annulant sur $\partial\Omega$ et polynômiales par morceaux sur une triangulation régulière $\mathcal{T}_H$ de $\bar\Omega$ et V_h – l'espace des fonctions continues, s'annulant sur $\partial\Lambda$, polynômiales par morceaux sur une triangulation régulière $\mathcal{T}_h$ de $\bar\Lambda$ ($h\ll H$) et qui sont prolongées par zéro sur $\bar\Omega\setminus\Lambda$. En posant $V_{Hh}=V_H+V_h$ on cherche une approximation $u_{Hh}\in V_{Hh}$ de u qui satisfait (1) pour toutes $v\in V_{Hh}$. En pratique, il est impossible de construire une base standard d'éléments finis pour l'espace V_{Hh} et de résoudre le problème discrétisé directement. C'est pourquoi un algorithme itératif pour calculer u_{Hh} a été proposé dans [3] (voire Algorithm 1 ci-dessous) dans lequel on ne résout que des problèmes standards sur V_H et V_h. Cet algorithme s'avère être très efficace dans les situations où la grille $\mathcal{T}_h$ est emboîtée dans $\mathcal{T}_H$ (c'est-à-dire chaque triangle de $\mathcal{T}_h$ est contenu dans un seul triangle de $\mathcal{T}_H$), mais son taux de convergence peut être extrêmement faible pour certains placements de $\mathcal{T}_h$ par rapport à $\mathcal{T}_H$. Cette observation peut être expliquée par le fait que le taux de convergence est géré par l'angle abstrait (par rapport au produit scalaire $a(\cdot,\cdot)$) entre V_h et $\tilde V_H$ où $\tilde V_H$ est le complément a-orthogonal de $V_H\cap V_h$ dans V_H. Dans le cas emboîté, $\tilde V_H$ ne contient que les fonctions qui approchent des fonctions harmoniques dans Λ. L'espace $\tilde V_H$ est donc approximativement a-orthogonal à V_h, ce qui explique la rapide convergence de la méthode dans ce cas. Si, au contraire, $\mathcal{T}_h$ est placé sur $\mathcal{T}_H$ de sorte à ce que $V_H\cap V_h=0$, les espaces V_h et $\tilde V_H=V_H$ peuvent être presque « parallèles », ce qui implique une mauvaise convergence. Le but de cette Note est de transférer partiellement les bonnes propriétés de convergence du cas emboîté au cas général. L'idée est de remplacer l'espace V_H dans l'algorithme précédent par son sous-espace $\bar V_H$ qui est le complément a-orthogonal du V_H^0 où V_H^0 est le sous-espace de V_H composé des fonctions qui s'annulent dans $\bar\Omega\setminus\Lambda$. On obtient alors des problèmes non standards posés dans $\bar V_H$ qui peuvent être résolus efficacement moyennant l'introduction de problèmes auxiliaires et peu coûteux posés dans V_H^0. Cette nouvelle variante de l'algorithme (voir Algorithm 2 ci-dessous) converge vers une approximation $\bar u_{Hh}\in\bar V_{Hh}$ de u (avec $\bar V_{Hh}=\bar V_H+V_h$) qui satisfait (1) pour tout $v\in\bar V_{Hh}$. On établit l'estimation a priori pour $\bar u_{Hh}$ du type

$$|u-\bar u_{Hh}|_1\leqslant C\big(H^r\|u\|_{H^q(\Omega\setminus\bar\Lambda)}+h^s\|u\|_{H^q(\Lambda)}\big),$$

où r et s sont les degrés des polynômes utilisés dans la construction de V_H et V_h respectivement et $q=\max(r,s)+1$. On conclut la Note par les expériences numériques qui montrent une convergence très rapide de notre nouvelle méthode.

Let $\Omega\subset\mathbb{R}^2$ be a bounded polygonal domain and $\Lambda\Subset\Omega$ be another polygonal domain which is 'much smaller' than Ω. Given an $f\in H^{-1}(\Omega)$, we consider the Poisson–Dirichlet problem of finding u such that $-\Delta u=f$ in Ω and $u=0$ on $\partial\Omega$, whose weak formulation is: find $u\in H_0^1(\Omega)$ such that

$$a(u,v)=\langle f|v\rangle,\quad \forall v\in H_0^1(\Omega), \tag{2}$$

where $a: H_0^1(\Omega)\times H_0^1(\Omega)\to\mathbb{R}$ is symmetric, continuous and coercive bilinear form

$$a(u,v)=\int_\Omega \nabla u\cdot\nabla v\,\mathrm{d}\mathbf{x}. \tag{3}$$

We are interested in the 'multi-scale' situation where the solution u of (2) has a very strong variation inside Λ so that the discretization of (2) by a finite element method cannot be done using grids with the same resolution on $\bar\Omega\setminus\Lambda$ and $\bar\Lambda$. Therefore we introduce two finite element spaces: $V_H=\{g\in C_0(\bar\Omega)$ such that $g|_K\in\mathbb{P}_r(K),\forall K\in\mathcal{T}_H$ and $g=0$ on $\partial\Omega\}$ where $\mathcal{T}_H$ is a regular triangulation of $\bar\Omega$ and $\mathbb{P}_r(K)$ is the space of polynomials of degree $\leqslant r$ on a triangle $K\in\mathcal{T}_H$, and $V_h=\{g\in C_0(\bar\Omega)$ such that $g|_K\in\mathbb{P}_s(K),\forall K\in\mathcal{T}_h$ and $g=0$ on $\bar\Omega\setminus\Lambda\}$ where $\mathcal{T}_h$ is a regular triangulation of $\bar\Lambda$. Setting $V_{Hh}=V_H+V_h$ we search as approximation for u the function $u_{Hh}\in V_{Hh}$ satisfying

$$a(u_{Hh},v)=\langle f|v\rangle,\quad \forall v\in V_{Hh}. \tag{4}$$

A priori $V_H\cap V_h$ does not necessarily reduce to the element zero and it is impossible, practically speaking, to exhibit a 'finite element'-type basis of the space V_{Hh}. That is why the following algorithm to compute u_{Hh} was suggested in [3]:

Algorithm 1.

(1) Set $u_H^0 = 0$, $u_h^0 = 0$.
(2) For $n = 1, 2, 3, \ldots$ find
  (i) $u_H^n \in V_H$ such that $a(u_H^n, v) = \langle f|v\rangle - a(u_h^{n-1}, v),\ \forall v \in V_H$,
  (ii) $u_h^n \in V_h$ such that $a(u_h^n, v) = \langle f|v\rangle - a(u_H^n, v),\ \forall v \in V_h$, set $u_{Hh}^n = u_H^n + u_h^n$.

Remark 1. Algorithm 1 can be viewed as a block Gauss–Seidel method to solve (4). It was presented in [3] in a different and more general form with possible use of over-relaxation making it similar to the SOR method. The present form makes evident the connection of this algorithm to the Chimera method [1]. Note also that the coarse update (i) and fine update (ii) are swapped here in comparison with the original version of [3]. This is motivated by the numerical results presented below.

Letting $P_h : V_{Hh} \to V_h$ and $P_H : V_{Hh} \to V_H$ be the orthogonal (with respect to $a(\cdot,\cdot)$) projectors from V_{Hh} to V_h and V_H respectively, it is easy to see that

$$u_{Hh} - u_{Hh}^n = (I - P_h)(I - P_H)\big(u_{Hh} - u_{Hh}^{n-1}\big).$$

We recall the following result from [2]:

Lemma 1. *Let V be an inner product space of finite dimension, V_1 and V_2 be two subspaces of V such that $V = V_1 + V_2$ and $P_j : V \to V_j$, $j = 1, 2$, be orthogonal projectors from V to V_j. Let $V_0 = V_1 \cap V_2$ and suppose that $V_1 \neq V_0$, $V_2 \neq V_0$. Define $\tilde\gamma = \tilde\gamma(V_1, V_2)$, $0 \leqslant \tilde\gamma < 1$ to be the cosine of the abstract angle between V_1 and V_2 so that*

$$\tilde\gamma(V_1, V_2) = \sup_{\substack{v_1 \in V_1 \cap V_0^\perp,\ v_1 \neq 0 \\ v_2 \in V_2 \cap V_0^\perp,\ v_2 \neq 0}} \frac{(v_1, v_2)}{\|v_1\|\|v_2\|} \tag{5}$$

where $(\cdot,\cdot)$ denotes the inner product in V and $\|\cdot\|$ is the norm induced by it. Then the spectral radius of operator $B = (I - P_2)(I - P_1)$ is $\rho(B) = \tilde\gamma^2$.

Applying this result in the case $V = V_{Hh}$ equipped with the inner product $a(\cdot,\cdot)$, $V_1 = V_H$, $V_2 = V_h$ we readily see that Algorithm 1 converges and its asymptotic convergence rate is given by $\tilde\gamma(V_H, V_h)^2$, hence the convergence is slow if the parameter $\tilde\gamma(V_H, V_h)$ is close to 1. Note, that even a more general version of Algorithm 1 using an over-relaxation can allow one to improve the convergence rate to only $1 - \sqrt{8(1-\tilde\gamma)} + \mathrm{O}(1-\tilde\gamma)$ as $\tilde\gamma \to 1$, see [2]. Very poor convergence was indeed observed in practice in some cases [2,6,8].

There is, however, at least one situation in which Algorithm 1 converges reasonably fast, namely when the triangulations $\mathcal{T}_H$ and $\mathcal{T}_h$ are nested, i.e. each triangle of $\mathcal{T}_h$ is contained in only one triangle of $\mathcal{T}_H$. This experimental observation is easy to interpret in view of Lemma 1. The intersection space $V_0 = V_H \cap V_h$ can be described in this case as $V_0 = V_H^0$ where V_H^0 is the subspace of V_H defined by

$$V_H^0 = \big\{v_H \in V_H : v_H(x) = 0\ \forall x \in \Omega \setminus \bar\Lambda\big\}, \tag{6}$$

so the a-orthogonal compliment to V_0 in V_H is

$$\bar V_H = \big\{v_H \in V_H : a(v_H, \phi_H) = 0\ \forall \phi_H \in V_H^0\big\}. \tag{7}$$

It consists of finite element functions of V_H that are approximately harmonic inside Λ on triangulation $\mathcal{T}_H$. The space $\bar V_H$ is therefore approximately a-orthogonal to the space V_h, since $a(\phi_H, \phi_h)$ should be small for any $\phi_H \in \bar V_H$ and any $\phi_h \in V_h$ vanishing on the boundary $\partial\Lambda$ by definition of V_h. Hence we can indeed expect that the parameter $\tilde\gamma(V_H, V_h) = \tilde\gamma(\bar V_H, V_h)$ is small in the case of nested triangulations. Note that V_h is a subspace of V_H^0 so that the approximation u_{Hh} given by (4) would not change if we replace $V_{Hh} = V_H + V_h$ by $\bar V_{Hh} = \bar V_H + V_h$ there. Moreover, hypothetically we could replace the space V_H by $\bar V_H$ on all the iterations of Algorithm 1 and this would produce the same iterates u_{Hh}^n.

We inspire ourselves from the last remark to modify Algorithm 1 by replacing the space V_H by its subspace consisting of functions which are approximately harmonic inside Λ in the hope to transfer partially the good convergence

properties of the nested case to the general setting of arbitrary triangulations. More specifically, we keep the notations (6) and (7) to define $\bar{V}_{Hh} = \bar{V}_H + V_h$ also in a nonnested case. Note that V_H^0 consists now of all the functions from V_H with supports inside $\bar{\Lambda}_H$ where $\bar{\Lambda}_H$ is the union of (closed) triangles from $\mathcal{T}_H$ that lie completely inside $\bar{\Lambda}$. We are now looking for the approximation $\bar{u}_{Hh} \in \bar{V}_{Hh}$ to u in (2) such that

$$a(\bar{u}_{Hh}, v) = \langle f|v\rangle, \quad \forall v \in \bar{V}_{Hh}. \tag{8}$$

In order to run Algorithm 1 with V_H replaced by $\bar{V}_H$, we should be able to solve the problems like the following one: given $g \in V_H'$ find $u_H \in \bar{V}_H$ such that

$$a(u_H, v) = \langle g|v\rangle, \quad \forall v \in \bar{V}_H. \tag{9}$$

In practice, we do not have an explicit finite element basis for the space $\bar{V}_H$ so that a direct implementation of (9) is impossible. To circumvent this difficulty we consider two auxiliary finite element problems:

(a) Find $\lambda_H \in V_H^0$ such that $a(\lambda_H, \mu) = \langle g|\mu\rangle,\ \forall \mu \in V_H^0$.
(b) Find $u_H \in V_H$ such that $a(u_H, v) = \langle g|v\rangle - a(\lambda_H, v),\ \forall v \in V_H$.

One can easily see that u_H in (b) gives the solution of (9) and both problems (a) and (b) are straightforward to implement. We can now announce a modified version of Algorithm 1 in which V_H is implicitly replaced by $\bar{V}_H$ as:

Algorithm 2.

(1) Set $u_H^0 = 0$, $u_h^0 = 0$.
(2) For $n = 1, 2, 3, \ldots$ find
  (i) $\lambda_H^n \in V_H^0$ such that $a(\lambda_H^n, \mu) = (f|\mu) - a(u_h^{n-1}, \mu),\ \forall \mu \in V_H^0$,
  (ii) $u_H^n \in V_H$ such that $a(u_H^n, v) = \langle f|v\rangle - a(u_h^{n-1}, v) - a(\lambda_H^n, v),\ \forall v \in V_H$,
  (iii) $u_h^n \in V_h$ such that $a(u_h^n, v) = \langle f|v\rangle - a(u_H^n, v),\ \forall v \in V_h$, set $u_{Hh}^n = u_H^n + u_h^n$.

Note that auxiliary problem (i) in the algorithm above is normally very cheap and its matrix is just a submatrix of that in (ii) corresponding to the basis functions of V_H whose supports are inside $\bar{\Lambda}_H$. Applying Lemma 1 with $V_1 = \bar{V}_H$, $V_2 = V_h$ we see immediately that Algorithm 2 converges to the solution $\bar{u}_{Hh}$ of (8). Although the space $\bar{V}_{Hh}$ in (8) is poorer than V_{Hh} in (4), its solution $\bar{u}_{Hh}$ approximates u with the optimal order as shows the following proposition the proof of which will be found in [4]:

Proposition 2. *Let $q = \max(r, s) + 1$ and suppose that the solution u of (2) is in $H^q(\Omega)$. If the patch Λ is convex then the solution $\bar{u}_{Hh}$ of* (8) *satisfies*

$$|u - \bar{u}_{Hh}|_1 \leqslant C\big(H^r \|u\|_{H^q(\Omega\setminus\bar{\Lambda})} + h^s \|u\|_{H^q(\Lambda)}\big), \tag{10}$$

where H and h are the maximum diameters of the triangles in $\mathcal{T}_H$ and $\mathcal{T}_h$ respectively and C is a constant independent of H and h.

Algorithm 2 is particularly efficient in the case $\bar{\Lambda} = \bar{\Lambda}_H$. This situation will be referred to as 'boundary conforming' triangulations. We prove in [4] that the convergence rate of Algorithm 2 can be estimated independently of H and h in this case.

Numerical example. We illustrate the two algorithms presented above on the example of the Poisson–Dirichlet problem (2) in $\Omega = (-1, 1)^2$ with the exact solution

$$u = \cos(k\pi x_1)\cos(k\pi x_2) + \eta\chi(r)\exp\left(\frac{1}{\epsilon^2} - \frac{1}{|\epsilon^2 - r^2|}\right)$$

where $r = \sqrt{x_1^2 + x_2^2}$ and $\chi(r) = 1$ if $r \leqslant \epsilon$, $\chi(r) = 0$ if $r > \epsilon$; $k = \frac{1}{2}$, $\eta = 20$ and $\epsilon = 0.3$. In a region close to $(0, 0)$ where the solution is peaking, we apply a patch with a finer grid. We choose P_1 finite elements for both V_H and V_h. The software `Freefem++` [5] is used to generate the grids and to implement the algorithm. The 'mixed terms' of the

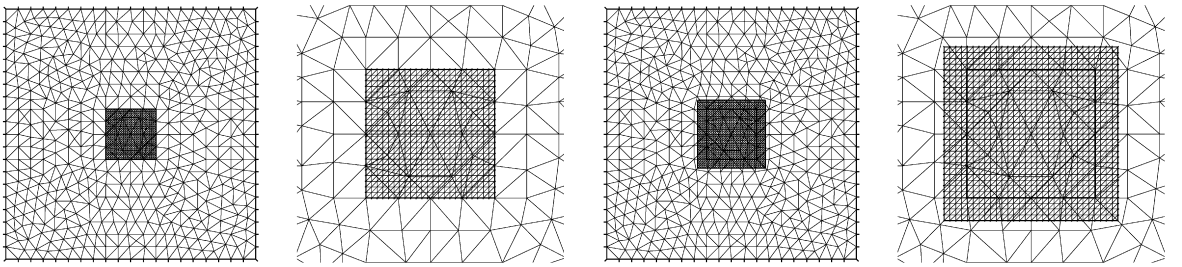

Fig. 1. (a) Grids $\mathcal{T}_H$ and $\mathcal{T}_h$ in the boundary conforming situation; (b) zoom of (a) to the rectangle $(-0.4, 0.4)^2$; (c) Grids $\mathcal{T}_H$ and $\mathcal{T}_h$ in the boundary nonconforming situation; (d) zoom of (c) to the rectangle $(-0.4, 0.4)^2$.

type $a(u_H, v_h)$ with $u_H \in V_H$ and $u_h \in V_h$ are calculated exactly using the numerical integration on the auxiliary grid representing the intersections of $\mathcal{T}_H$ and $\mathcal{T}_h$ and constructed with the aid of software `Triangle` [7].

We consider two situations:

(i) Boundary conforming case with an unstructured grid $\mathcal{T}_H$ having the grid size H_b on the boundary and structured grid $\mathcal{T}_h$ that covers the patch $\Lambda = (-0.2, 0.2)^2$. The grid $\mathcal{T}_H$ is constructed so that $\partial\Lambda$ is a union of edges of some triangles of $\mathcal{T}_H$, see Fig. 1(a), (b) with $H_b = 1/10$ and $\mathcal{T}_h$ consisting of 24×24 nodes. The integrals on the right-hand side $\langle f|v_h\rangle$ for $v_h \in V_h$ are approximated by 7-point quadrature formulas on $\mathcal{T}_h$. Their coarse triangulation

Table 1
Relative H^1- and L^2-error on iterations and for the converged solution, number of iterations until convergence and numerically computed asymptotic convergence rate in the case of boundary conforming grids

	Algorithm 1						Algorithm 2					
H_b	1/10		1/20		1/40		1/10		1/20		1/40	
iter.	H^1	L^2	H^1	L^2	H^1	L^2	H^1	L^2	H^1	L^2	H^1	L^2
0.5	3.82E−1	1.44E−1	2.77E−1	6.42E−2	1.41E−1	1.78E−2	9.98E−1	8.46E−1	9.98E−1	8.48E−1	9.98E−1	8.49E−1
1	4.31E−2	9.58E−3	4.58E−2	2.60E−3	2.43E−2	7.03E−4	1.76E−2	9.50E−3	4.18E−2	1.94E−3	9.12E−4	5.30E−4
1.5	7.86E−2	4.33E−2	5.14E−2	1.49E−2	2.61E−2	5.01E−3	9.59E−3	4.39E−3	2.28E−3	1.02E−3	5.33E−4	2.50E−4
2	3.83E−2	7.37E−3	4.30E−2	2.30E−3	2.38E−2	5.45E−4	8.18E−3	4.21E−3	1.99E−3	1.01E−3	5.16E−4	2.49E−4
2.5	7.06E−2	3.92E−3	4.81E−2	1.40E−3	2.54E−2	4.88E−3	7.94E−3	4.23E−3	1.95E−3	1.00E−3	5.14E−4	2.50E−4
conv.	1.12E−2	3.69E−3	1.25E−2	1.97E−3	1.22E−2	5.03E−4	7.87E−3	4.20E−3	1.94E−3	1.00E−3	5.13E−4	2.49E−4
iter.	54		68		65		5		4		3	
$\tilde{\gamma}^2$	0.9565		0.9927		0.9967		0.2006		0.2046		0.2046	

Table 2
Same results as in Table 1 for the case of boundary nonconforming grides

	Algorithm 1						Algorithm 2					
H_b	1/10		1/20		1/40		1/10		1/20		1/40	
iter.	H^1	L^2	H^1	L^2	H^1	L^2	H^1	L^2	H^1	L^2	H^1	L^2
0.5	4.14E−1	1.44E−1	2.72E−1	6.41E−2	1.39E−1	1.78E−2	1.00	8.50E−1	9.98E−1	8.48E−1	9.98E−1	8.49E−1
1	7.55E−2	7.81E−3	3.94E−2	1.97E−3	2.28E−2	5.77E−4	1.03E−2	6.18E−3	2.18E−3	1.25E−3	6.08E−4	4.05E−4
1.5	8.87E−2	3.55E−2	4.71E−2	1.54E−2	2.49E−2	5.17E−3	1.01E−2	5.22E−3	2.36E−3	1.27E−3	6.42E−4	4.36E−4
2	6.85E−2	7.94E−3	3.71E−2	1.96E−3	2.22E−2	5.09E−4	8.93E−3	4.79E−3	2.10E−3	1.11E−3	5.56E−4	3.01E−4
2.5	7.91E−2	3.18E−2	4.41E−2	1.45E−2	2.42E−2	5.04E−3	9.40E−3	5.09E−3	2.16E−3	1.17E−3	5.91E−4	3.72E−4
conv.	1.53E−2	4.11E−3	1.17E−2	2.32E−3	1.16E−2	4.31E−4	8.72E−3	4.89E−3	2.09E−3	1.09E−3	5.51E−4	2.87E−4
iter.	76		61		61		11		4		3	
$\tilde{\gamma}^2$	0.9687		0.9907		0.9969		0.8236		0.9339		0.9698	

counterparts $\langle f|v_H\rangle$ for $v_H \in V_H$ are approximated using both coarse triangulation $\mathcal{T}_H$ in $\bar{\Omega} \setminus \Lambda$ and fine triangulation $\mathcal{T}_h$ in Λ.

(ii) Boundary nonconforming case with the same unstructured grid $\mathcal{T}_H$ as above and structured grid $\mathcal{T}_h$ that covers the patch $\Lambda = (-0.27, 0.27)^2$, see Fig. 1(c), (d) with $H_b = 1/10$ and $\mathcal{T}_h$ consisting of 31×31 nodes so that the grid size h is approximately the same as in the case (i). Unlike situation (i), the integrals $\langle f|v_H\rangle$ for $v_H \in V_H$ are approximated using only coarse triangulation $\mathcal{T}_H$ because of algorithmic difficulties with the evaluation of integrals on $\Lambda \setminus \bar{\Lambda}_H$.

Table 1 represents the results obtained in situation (i) on the grids as in Fig. 1(a), (b) and two consecutive twofold refinements thereof. We report the evolution of the error $e^n = u^n_{Hh} - u$ and $e^{n-1/2}_{Hh} = u^n_H + u^{n-1}_h - u$ in H^1 and L^2 norms on first 2 iterations, the error after convergence, the stopping criterion being $|u^n_{Hh} - u^{n-1}_{Hh}|_1 < 10^{-4}|u^n_{Hh}|_1$ and the number of iterations needed to reach the convergence. In order to determine numerically the asymptotic convergence rate we have also run our algorithms with $f = 0$ starting from some non-zero u^0_H, u^0_h until the error reduction per iteration quotient attains a stable value (with variation below 10^{-6}). This quantity is equal to $\tilde{\gamma}^2$ according to Lemma 1 and we report it in the last line of Table 1. Numerical results clearly demonstrate the superiority of our new Algorithm 2. Note also that the error in Algorithm 1 tends to be smaller after the fine update than after the coarse update (look at the iterations with numbers $n - 1/2$).

Analogous numerical experiments in boundary nonconforming situation (ii) are reported in Table 2. The numerically calculated abstract angle $\tilde{\gamma}$ governing the asymptotic convergence rate of Algorithm 2 is no longer as good as in the boundary conforming case. However, the experiments with 'multi-scale' non-zero right-hand side f show again the superiority of the new algorithm both in terms of the number of iterations needed to reach convergence and the accuracy of the converged solution.

Algorithm 2 can be easily extended to general elliptic problems in $\mathbb{R}^d$ for any positive integer d and also to the situations with multiple patches. We note finally that another way to increase the convergence rate of Algorithm 1 is to use it as a preconditioner for a Conjugate Gradient (CG) method [6]. One can expect therefore to obtain yet faster convergence by incorporating Algorithm 2 into the CG approach.

Acknowledgements

The authors are very grateful to Prof. Olivier Pironneau for the interest in their work and helpful discussions and to Prof. Frédéric Hecht for the help with the software `Freefem++`.

References

[1] F. Brezzi, J.-L. Lions, O. Pironneau, Analysis of a chimera method, C. R. Acad. Sci. Paris, Ser. I 332 (2003) 655–660.
[2] R. Glowinski, J. He, A. Lozinski, J. Rappaz, J. Wagner, Finite element approximation of multi-scale elliptic problems using patches of elements, Numer. Math. 101 (4) (2005) 663–687.
[3] R. Glowinski, J. He, J. Rappaz, J. Wagner, Approximation of multi-scale elliptic problems using patches of finite elements, C. R. Acad. Sci. Paris, Ser. I 337 (2003) 679–684.
[4] J. He, A. Lozinski, J. Rappaz, Some results on the method of finite element patches, Technical report, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2007, in preparation.
[5] F. Hecht, O. Pironneau, K. Ohtsuka, `Freefem++`, ver. 4.12 http://www.freefem.org.
[6] V. Rezzonico, Multiscale algorithm with patches of finite element and applications, PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2007.
[7] J.R. Shewchuk, `Triangle`, http://www.cs.cmu.edu/quake/triangle.html.
[8] J. Wagner, Finite element methods with patches and applications, PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2006.