International Journal of Computer Mathematics, 2014
Vol. 91, No. 12, 2535–2553, http://dx.doi.org/10.1080/00207160.2013.878022

GUIDE

Adaptive dimension splitting algorithm for three-dimensional elliptic equations

Hongbo Wei[a], Yanren Hou[a,b*], Lina Song[c]

[a]*School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China;* [b]*Center for Computational Geosciences, Xi'an Jiaotong University, Xi'an 710049, China;* [c]*College of Mathematics, Qingdao University, Qingdao 266071, China*

(*Received 27 December 2012; revised version received 4 April 2013; second revision received 3 July 2013; third revision received 8 November 2013; fourth revision received 17 December 2013; accepted 18 December 2013*)

An adaptive dimension splitting algorithm for three-dimensional (3D) elliptic equations is presented in this paper. We propose residual and recovery-based error estimators with respect to $X - Y$ plane direction and Z direction, respectively, and construct the corresponding adaptive algorithm. Two-sided bounds of the estimators guarantee the efficiency and reliability of such error estimators. Numerical examples verify their efficiency both in estimating the error and in refining the mesh adaptively. This algorithm can be compared with or even better than the 3D adaptive finite element method with tetrahedral elements in some cases. What is more, our new algorithm involves only two-dimensional mesh and one-dimensional mesh in the process of refining mesh adaptively, and it can be implemented in parallel.

Keywords: dimension splitting algorithm; elliptic equation; finite element method; adaptive technique; a posteriori error estimate

2000 AMS Subject Classifications: 65N15; 65N30

1. Introduction

Practical three-dimensional (3D) boundary value problems always lead to solving a large system. In particular, if problems are singular or associated with complex, irregular geometry, the computational complexity is also increased. A great number of efficient methods were proposed to treat such problems. One popular method is the domain decomposition method. It decomposes the spatial domain of the problem into several subdomains, which allows parallelization and localized treatment of the 3D problem [6,18]. Another method is the dimension splitting algorithm (DSA) presented by Li and Huang in ref. [9]. Instead of decomposing the spatial domain, it splits the 3D problem into a series of two-dimensional (2D) problems. Recently, Li and his coworkers [10–13] extended the dimensional splitting method to the compressible Navier–Stokes equations and the linear elastic shell based on differential geometry and tensor analysis. Hou and Wei [7] presented a finite element DSA for the 3D elliptic equation in cartesian coordinates. The algorithm decomposes the 3D elliptic equation into a series of 2D elliptic equations in X–Y plane along Z direction. It has two advantages. One is that the series of 2D elliptic equations can be easily

*Corresponding author. Email: yrhou@mail.xjtu.edu.cn

solved in parallel. The other is that only 2D meshes are involved, which are much easier to be constructed than 3D meshes.

The aim of this paper is to propose an adaptive algorithm for the DSA based on two error estimators. A reliable and efficient posteriori error estimator can provide the knowledge about the magnitude and distribution of the error. Moreover, it can adaptively control the refinement of grid to get the overall optimal accuracy. The articles by Babuska and Rheinboldt [2,3] represented the pioneering work in the field of adaptive finite element methods. In the past decades, a great number of researchers have developed the error estimation techniques (e.g. see [1,4,5,15,16,19–24]).

The main characteristics of the adaptive DSA (ADSA) are as follows.

(1) It is based on two error estimators with respect to X–Y plane and Z direction. It is efficient and feasible to deal with the singularity only in Z direction or X–Y plane direction. Comparing the 3D adaptive finite element method with tetrahedral elements, this algorithm avoids the construction of 3D adaptive meshes, thus saving a lot of central processing unit time and memory.
(2) It uses the recovery-based error estimator with respect to Z direction. In the DSA, the residual type of error estimator fails to estimate the error in Z direction. However, the recovery-based error estimator works well for the DSA due to its easy implementation and mild dependence upon the problem's data.
(3) It inherits all advantages of the DSA, such as implementation in parallel and usage of 2D meshes rather than 3D meshes.
(4) Posteriori error estimators are proposed to guarantee the efficiency and reliability of the algorithm. Moreover, numerical results confirm the theoretical results and show the advantages of the ADSA.

The remainder of this paper is arranged as follows. In Section 2, we briefly review the DSA. Posteriori error estimators both in X–Y plane direction and Z direction are discussed in Section 3. In Section 4, we present the ADSA based on the two error estimators. Numerical experiments are carried out in Section 5. Finally, we end the paper with a short concluding section.

2. Dimension splitting algorithm

We introduce briefly the DSA in ref. [7] for 3D elliptic equations in this section. We consider the following 3D elliptic equation with homogeneous Dirichlet boundary condition defined in a cubic domain $\Omega = [0,1]^3 \subset R^3$,

$$\begin{aligned} -\Delta u + \kappa u &= f \qquad \forall (x,y,z) \in \Omega, \\ u|_{\partial\Omega} &= 0 \qquad \forall (x,y,z) \in \partial\Omega, \end{aligned} \tag{1}$$

where $\kappa \geq 0$ is a constant.

For a given index set $I = \{0,1,2,\ldots,LZ\}$, where $LZ > 0$ is a given positive integer, let us denote

$$\Omega_i = \Omega \cap \{z = z_i\} \subset R^2 \quad \text{for given } i \in I \text{and } 0 \leq z_i \leq 1.$$

In the following, we denote $\omega = \Omega_i = [0,1]^2, I_d = [0,1]$, $\Delta_2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$. Suppose $\mathcal{T}_h = \{K\}$ is a quasi-regular triangulation of ω with grid size $h = \max\{h_K : K \in \mathcal{T}_h\}$, where $h_K = \text{diam}\{K\}$ and $X_h \subset X = H_0^1(\omega)$ is the P_1 finite element subspace. Let $\mathcal{T}_d = \{d_i\}$ be a quasi-regular triangulation of $[0,1]$ with grid size $d = \max\{d_i = z_i - z_{i-1} : i \in I\backslash\{0\}\}$ and $Z_d = \text{span}\{\phi_i,\ i \in I\} = \{v_d \in C^0([0,1]) : v_d|_{d_i} \in P_1, \forall d_i \in \mathcal{T}_d\}$ be the corresponding piecewise linear one-dimensional finite element subspace, where ϕ_i is the piecewise linear basis function at node

i. We define a 3D finite-dimensional subspace $X_{hd} = X_h \times Z_d \subset H_0^1(\Omega)$. Then, the finite element approximation to the solution of Equation (1) u_{hd} has the form

$$u_{hd} = \sum_{i \in I} u_{i,h}\phi_i \quad \text{with } u_{i,h} \in X_h. \tag{2}$$

Using the DSA, the variational formulation of problem (1) can be rewritten as the following equation. For given $i \in I$, $\forall \phi_i \in Z_d$ and $v \in X_h$, find $u_{i,h} \in X_h$ such that

$$\left(\left(\frac{\partial u_{hd}}{\partial z}, \phi_i'\right)_z, v\right)_{xy} + a((u_{hd}, \phi_i)_z, v)_{xy} + \kappa((u_{hd}, \phi_i)_z, v)_{xy} = ((f, \phi_i)_z, v)_{xy}, \tag{3}$$

where $(\cdot,\cdot)_z$ is the scalar product in I_d, and $(\cdot,\cdot)_{xy}$ is the scalar product in ω and $a(\cdot,\cdot)_{xy} = (\nabla_2\cdot, \nabla_2\cdot)_{xy}$.

By simple calculation, the above equation becomes

$$\begin{aligned}\left(\frac{3}{d_i d_{i+1}} + 1\right) u_{i,h} - \Delta_2 u_{i,h} = f_{i,h} &+ \frac{d_i}{2(d_i + d_{i+1})} f_{i-1,h} + \frac{d_{i+1}}{2(d_i + d_{i+1})} f_{i+1,h} \\ &+ \Delta_2\left(\frac{d_i}{2(d_i + d_{i+1})} u_{i-1,h} + \frac{d_{i+1}}{2(d_i + d_{i+1})} u_{i+1,h}\right) \\ &+ \frac{6 - d_i^2}{2(d_i + d_{i+1})d_i} u_{i-1,h} + \frac{6 - d_{i+1}^2}{2(d_i + d_{i+1})d_{i+1}} u_{i+1,h}, \quad \forall i \in I,\end{aligned} \tag{4}$$

where $f_{i,h}$ is the finite element interpolation of $f|_{z=z_i}$.

A Jacobi-like iterative scheme for the above equation is as follows.

$$\begin{aligned}\left(\frac{3}{d_i d_{i+1}} + 1\right) u_{i,h}^{n+1} - \Delta_2 u_{i,h}^{n+1} = f_{i,h} &+ \frac{d_i}{2(d_i + d_{i+1})} f_{i-1,h} + \frac{d_{i+1}}{2(d_i + d_{i+1})} f_{i+1,h} \\ &+ \Delta_2\left(\frac{d_i}{2(d_i + d_{i+1})} u_{i-1,h}^n + \frac{d_{i+1}}{2(d_i + d_{i+1})} u_{i+1,h}^n\right) \\ &+ \frac{6 - d_i^2}{2(d_i + d_{i+1})d_i} u_{i-1,h}^n + \frac{6 - d_{i+1}^2}{2(d_i + d_{i+1})d_{i+1}} u_{i+1,h}^n, \\ &\forall i \in I,\ n = 0, 1, 2, \ldots,\end{aligned} \tag{5}$$

where $u_{i,h}^{n+1}$ denotes the $(n+1)$th iterative approximation to $u_{i,h}$.

3. A posteriori error estimate for DSA

First of all, we obtain the initial approximate solution $u_{h,d}$ to Equation (1) by means of the scheme (5) with the initial mesh $\mathcal{T}_d^0 = \{d_i : d_i = 1/LZ\}$ in Z direction and $\mathcal{T}_{i,h}^0$ for every Ω_i in X–Y plane direction. In Figure 1, we present the initial mesh $\mathcal{T}_{i,h}^0$. In order to develop the ADSA, we construct the corresponding error estimators with respect to both Z direction and X–Y plane direction.

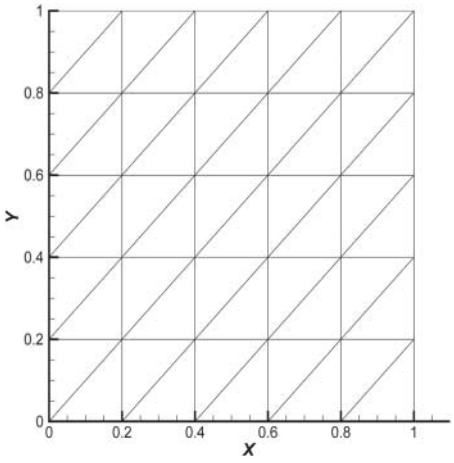

Figure 1. The initial mesh $\mathcal{T}_{i,h}^0$.

3.1 *Error estimator with respect to Z direction*

In Z direction, we consider the recovery-based error estimator

$$\eta_{I_i}^2 = \frac{1}{|\omega|}\int_\omega \|G_{I_d}(u_{h,d}) - \frac{\partial u_{h,d}}{\partial z}\|_{L^2(I_{d_i})}^2 \,\mathrm{d}\omega, \tag{6}$$

where G_{I_d} is the recovered gradient operator and $|\omega|$ denotes the area of ω, $I_{d_i} = [z_{i-1}, z_i]$. The framework for constructing recovery operator was addressed by Zienkiewicz and Zhu [21,22]. G_{I_d} is of the form

$$G_{I_d}(v) = \sum_{i\in I} g_i(v)\phi_i, \tag{7}$$

where $\phi_i \in Z_d$, g_i are linear functions.

Let $z_i (i \in I)$ be an element vertex, and consequently, the patch $\bar{I}_{d_i}$ consists of the elements sharing the vertex z_i. The sampling points are at the midpoint c_i of each element I_{d_i} in the patch $\bar{I}_{d_i}$, as is illustrated in Figure 2. The values of the recovered gradient at the sampling points are used to produce a recovered value at the patch assembly point by a function of the form $P(z)^\top \alpha_i$, where the $P(z)^\top$ is based on the P_1 finite element space, $P(z) = [1, z]^\top$, and α_i is a unknown vector. α_i is determined by a discrete least-squares fit based on the values at the sampling points: α_i is the unique minimizer of $\mathcal{J}(\alpha_i)$

$$\mathcal{J}(\alpha_i) = \frac{1}{|\omega|}\int_\omega \sum_{I_{d_i}\in\bar{I}_{d_i}} \left(\frac{\partial u_{h,d}}{\partial z}(c_i) - P(c_i)^\top \alpha_i\right)^2 \mathrm{d}\omega.$$

The condition for the minimizer reveals that α_i is the solution of the matrix equation

$$A\alpha_i = b,$$

where A is the $2 * 2$ matrix

$$A = \sum_{I_{d_i}\in\bar{I}_{d_i}} P(c_i)P(c_i)^\top \tag{8}$$

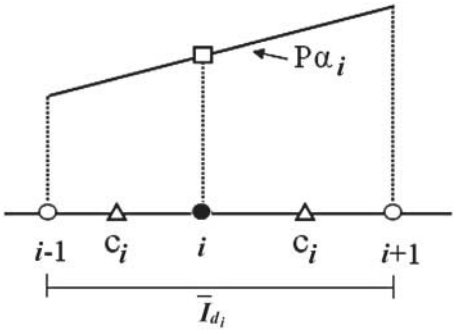

Figure 2. Patch $\tilde{I}_{d_i}$. ∘ Node; △ Sampling points; □ Nodal values determined by recovery procedure; • Patch assembly point.

and b is the vector

$$b = \frac{1}{|\omega|} \int_{\omega} \sum_{I_{d_i} \in \tilde{I}_{d_i}} P(c_i)^{\top} \frac{\partial u_{h,d}}{\partial z}(c_i)\, \mathrm{d}\omega. \tag{9}$$

Finally, the recovered values of the gradient operator sampled at the central node are defined by

$$g_i(u_{h,d}) = P(u_{i,h})^{\top} \alpha_i.$$

3.2 *Error estimator with respect to X–Y plane direction*

We consider the following 2D elliptic equation:

$$\begin{aligned} -\Delta_2 u + \kappa u &= f \quad \forall (x,y) \in \omega, \\ u|_{\partial\omega} &= 0 \quad \forall (x,y) \in \partial\omega, \end{aligned} \tag{10}$$

and suppose that $\mathcal{P}$ is a regular partition of the domain ω into triangular elements, and u_X is the Galerkin finite element approximation to u.

We define element and edge residuals by $r = f + \Delta_2 u_X - \kappa u_X$, and

$$R = \begin{cases} -\left[\dfrac{\partial u_X}{\partial n}\right], & \partial\mathcal{P} \backslash \partial\omega, \\ 0, & \partial K \cap \partial\omega, \end{cases} \tag{11}$$

where $[\cdot]$ represents the jump discontinuity in the approximation to the normal flux on the interface. Then, the classical local residual error estimator addressed by Verfurth [20] is

$$\eta_K^2 = h_K^2 \|r\|_{L^2(K)}^2 + \frac{1}{2} h_K \|R\|_{L^2(\partial K)}^2. \tag{12}$$

And the global error estimator $\eta^2 = \sum_{K \in \mathcal{P}} \eta_K^2$ really provides a two-sided estimation of the true error measured in energy norm.

We define the local posteriori error estimator $\eta_{i,K}^2$ for Equation (5):

$$\eta_{i,K}^2 = h_{i,K}^2 \|r'\|_{L^2(K)}^2 + \frac{1}{2} h_{i,K} \|R'\|_{L^2(\partial K)}^2, \tag{13}$$

where $r' = f' + \Delta_2 u_{i,h}^{n+1} - \kappa' u_{i,h}^{n+1}$ with $\kappa\ '= 3/d_i d_{i+1} + 1$, f' is the right-hand side of Equation (5),

$$R' = \begin{cases} -\left[\dfrac{\partial u_{i,h}^{n+1}}{\partial n}\right], & \partial\mathcal{P}\backslash\partial\omega, \\ 0, & \partial K \cap \partial\omega. \end{cases} \tag{14}$$

From Verfurth [20], there exists the following upper bound for the error $e_i = u(z_i) - u_{i,h}$ in Equation (5)

$$C_1 |||e_i|||_\omega^2 \le \sum_{K\in\mathcal{P}} \eta_{i,K}^2, \tag{15}$$

as well as the local lower bound for e_i

$$\eta_{i,K}^2 \le C_2 |||e_i|||_{\tilde{K}}^2, \tag{16}$$

where $\tilde{K}$ consists of the element K along with the elements sharing at least one common vertex with K, $|||e_i|||_\omega = (\int_\omega |\nabla_2 e_i|^2 + \kappa e_i^2 \,\mathrm{d}\omega)^{1/2}$, $\nabla_2 = (\partial/\partial x, \partial/\partial y)$.

To combine with two-sided bounds on every e_i, we deduce the following global two-sided bounds on the error $e = u - u_{h,d}$ in Ω.

THEOREM 3.1 *There exists a constant C, independent of d_i, such that*

$$|||u - u_{h,d}|||_\Omega^2 \le d \sum_{i=1}^{LZ-1} \sum_K \eta_{i,K}^2 + Cd^2. \tag{17}$$

Proof By using the trapezoidal integration formula and the two-sided bounds on e_i, we have

$$\begin{aligned} |||u - u_{h,d}|||_\Omega^2 &= \sum_{i=1}^{LZ} \int_{z_{i-1}}^{z_i} |||\, u - u_{h,d}\, |||_\omega^2 \,\mathrm{d}z \\ &\le \sum_{i=1}^{LZ} \left\{ \frac{d_i}{2} (|||\, u(z_i) - u_{h,d}(z_i)\, |||_\omega^2 + |||\, u(z_{i-1}) - u_{h,d}(z_{i-1})\, |||_\omega^2) + Cd_i^2 \right\} \\ &\le \frac{d}{2} \sum_{i=1}^{LZ} \{(|||\, u(z_i) - u_{i,h}\, |||_\omega^2 + |||\, u(z_{i-1}) - u_{i-1,h}\, |||_\omega^2)\} + Cd^2 \\ &= d \sum_{i=1}^{LZ-1} |||\, u(z_i) - u_{i,h}\, |||_\omega^2 + Cd^2 \\ &\le d \sum_{i=1}^{LZ-1} \sum_K \eta_{i,K}^2 + Cd^2. \end{aligned} \tag{18}$$

The proof is thus complete. ■

THEOREM 3.2 *There exists the following lower bound:*

$$\sum_{i=1}^{LZ-1} d_i \eta_{i,K}^2 \le \frac{3}{2} \left\{ |||u - u_{h,d}|||_\Omega^2 + d^4 \left\| \frac{\partial \nabla u}{\partial z} \right\|_{L^2(\Omega)}^2 + d^4 \|\kappa \frac{\partial u}{\partial z}\|_{L^2(\Omega)}^2 \right\}. \tag{19}$$

Proof For the piecewise linear basis function ϕ_i at node i in Z direction, we have

$$\int_{I_d} \phi_i^2 \,\mathrm{d}z = \frac{2}{3} d_i. \tag{20}$$

Starting from the definition of $\eta_{i,K}^2$ and applying the identities (20), (2) and triangle inequality, we have

$$\begin{aligned}
\sum_{i=1}^{LZ-1} d_i \eta_{i,K}^2 &= \sum_{i=1}^{LZ-1} d_i \int_\omega |\nabla_2 e_i|^2 + \kappa e_i^2 \,\mathrm{d}\omega, \\
&= \frac{3}{2} \sum_{i=1}^{LZ-1} \int_{I_d} \int_\omega |\nabla_2 e_i|^2 \phi_i^2 + \kappa e_i^2 \phi_i^2 \,\mathrm{d}\omega \,\mathrm{d}z, \\
&\le \frac{3}{2} \int_{I_d} \int_\omega |\nabla_2 \sum_{i=1}^{LZ-1} e_i \phi_i|^2 + \kappa \sum_{i=1}^{LZ-1} (e_i \phi_i)^2 \,\mathrm{d}\omega \,\mathrm{d}z, \\
&\le \frac{3}{2} \int_{I_d} \int_\omega |\nabla_2 \left(\sum_{i=1}^{LZ-1} u(z_i)\phi_i - u \right)|^2 + |\nabla_2 (u - u_{h,d})|^2 \,\mathrm{d}\omega \,\mathrm{d}z \\
&\quad + \kappa \left(\sum_{i=1}^{LZ-1} u(z_i)\phi_i - u \right)^2 + \kappa (u - u_{h,d})^2 \,\mathrm{d}\omega \,\mathrm{d}z, \\
&\le \frac{3}{2} \int_{I_d} \int_\omega |\nabla \left(\sum_{i=1}^{LZ-1} u(z_i)\phi_i - u \right)|^2 + |\nabla (u - u_{h,d})|^2 \,\mathrm{d}\omega \,\mathrm{d}z \\
&\quad + \kappa \left(\sum_{i=1}^{LZ-1} u(z_i)\phi_i - u \right)^2 + \kappa (u - u_{h,d})^2 \,\mathrm{d}\omega \,\mathrm{d}z, \\
&\le \frac{3}{2} \left\{ ||| \, u - u_{h,d} \, |||_\Omega^2 + d^4 \| \frac{\partial \nabla u}{\partial z} \|_{L^2(\Omega)}^2 + d^4 \| \kappa \frac{\partial u}{\partial z} \|_{L^2(\Omega)}^2 \right\},
\end{aligned} \tag{21}$$

where the interpolation property with respect to Z direction is used in the last inequality. ■

4. Adaptive DSA

In this section, we give the mesh-refinement strategies both in Z direction and in X–Y plane direction.

In Z direction, we adopt the maximum refining strategy and show the details as follows. It should be noted that other refinement strategies are also available for our method.

Process 1

Step 1: Give an initial triangulation $\mathcal{T}_d^0$ along Z direction and a constant $\theta \in (0, 1)$.

Step 2: Once we get $\mathcal{T}_d^{j-1}$, $j \ge 1$, compute

$$\eta_{I_{d_i},\max} = \max_{i \in I} \eta_{I_{d_i}}.$$

Step 3: If

$$\eta_{I_{d_i}} \ge \theta \eta_{I_{d_i},\max},$$

mark I_{d_i} as the element to be refined and put it into a set $\tilde{I}_{d_i}$. Dividing every element in $\tilde{I}_{d_i}$ into two elements, we obtain the new refined mesh $\mathcal{T}_d^j$. Go to Step 2 unless $\eta_{I_{d_i}}$ reaches the desired tolerance and stop. In numerical implementation, we take $\theta = 0.5$.

In X–Y plane direction, we give the main idea of the refinement strategy presented in reference [8]. Starting from the original triangulation $\mathcal{T}_{i,h}^0$ for every Ω_i, we construct a sequence of refined triangulations $\mathcal{T}_{i,h}^j$ as follows. Based on a given mesh $\mathcal{T}_{i,h}^j$, we compute the solution of scheme (5). Then we calculate the error estimator $\eta_{i,K}^j$ and give the new grid size by the following formula:

$$h_i^{j+1} = \frac{h_i^j}{f_i^j(\eta_{i,K}^j)},$$

where h_i^j is the previous 'mesh size' field at $z = z_i$, and f_i^j is a user function defined by

$$f_i^j = \min\left(\max\left(\frac{\eta_{i,K}^j}{(c\bar{\eta}_i{}^j)}, 1.00\right), 3.00\right),$$

where $\bar{\eta}_i^j$ means the mean value of $\eta_{i,K}^j$, c is a user coefficient in general close to 1 and the numbers 1.00 and 3.00 also can be changed by the user according to requirement. The two artificial numbers, namely 1.00 and 3.00, control to coarsen and refine the mesh, respectively. For the sake of convenience, for all latter numerical tests, we choose $c = 0.9$. Certainly, we can also consider other refinement strategies, such as that in reference [19]. There are no big differences between the choices of refinement strategies.

For simplicity, we control the refinement of mesh by means of choosing adaptive level numbers of refinement. The computation with adaptive strategy here is made of following steps.

Process 2

Step 1: Given the adaptive level number of refinement M in X–Y plane direction, start from the original triangulation $\mathcal{T}_{i,h}^0$ for every Ω_i with $m = 1$.

Step 2: Compute iteratively scheme (5) based on the refined mesh in X–Y plane direction. If $m > M$, stop. Otherwise, go to Step 3.

Step 3: Generate a new 'mesh size' field h_i^{j+1} by the above strategy based on error estimator $\eta_{i,K}^j$. Set $m = m + 1$ and go back to Step 2.

From the *Process* 1 and *Process* 2, we present the following adaptive dimension splitting algorithm for 3D elliptic equations.

Algorithm-ADSA

Denote by L the global adaptive level number, and give some initial information needed in Step 1 of *Process* 1 and *Process* 2, and take $l = 1$.

Step 1: Compute the final solution $u_{h,d}$ on the final refined mesh $\mathcal{T}_{i,h}^*$ in X–Y plane direction and $\mathcal{T}_d^*$ in Z direction. If $l > L$, stop. Otherwise, go to Step 2.

Step 2: Implement *Process* 1 and *Process* 2. Set $l = l + 1$, and go back to Step 1.

Remark 1 When implementing the ADSA in the lth level number, if $I_{d_i} \in \tilde{I}_{d_i}$, then we need to construct the mesh $\mathcal{T}_{i',h}^l$ at the added centre node z_i' of I_{d_i}. We consider the regular uniform mesh as the initial mesh $\mathcal{T}_{i',h}^0$, whose element numbers are one half of the sum of both $\mathcal{T}_{i,h}^l$ and $\mathcal{T}_{i-1,h}^l$. What is more, according to the previous DSA, it is important to note that $u_{i,h}$ in *Process 2* can be solved in parallel.

5. Numerical experiments

In this section, we demonstrate the efficiency and reliability of the error estimators presented in the ADSA by numerical results. Moreover, we also compare the ADSA with the 3D adaptive finite element method for the 3D elliptic equation with four-node tetrahedron elements, which is denoted by the 3DAFEM. All numerical experiments are implemented by using public domain finite element software [8], and the mesh of the three dimension can be refined adaptively using TetGen [14]. This software is a tetrahedral mesh generator of a 3D domain defined by its boundary. The method in TetGen to control the quality of the mesh is based on the Delaunay refinement method [17]. In this paper, we assume that the condition of reaching convergence in scheme (5) is $\|u_{h,d}^{n+1} - u_{h,d}^{n}\|_{L^2(\Omega)} < \mathrm{Tol}$, where $\mathrm{Tol} = 10^{-6}$.

For convenience of presentation, we introduce the following notations:

DOF_i: the number of elements for triangulation $\mathcal{T}_{i,h}$;

NV: the number of vertices for mesh;

EI_I: the effective index with respect to Z direction, i.e. the ratio between the related estimator and the true error,

$$\mathrm{EI}_I = \frac{(\sum_{i\in I}(1/|\omega|)\int_\omega \|G_{I_d}(u_{h,d}) - \nabla_z u_{h,d}\|^2_{L^2(I_{d_i})}\,\mathrm{d}\omega)^{1/2}}{(\sum_{i\in I}\frac{1}{|\omega|}\int_\omega \|\nabla_z u - \nabla_z u_{h,d}\|^2_{L^2(I_{d_i})}\,\mathrm{d}\omega)^{1/2}};$$

EI_{ω_i}: the effective index with respect to X–Y plane direction,

$$\mathrm{EI}_{\omega_i} = \frac{(\sum_{K\in\omega_i}\eta^2_{i,K})^{1/2}}{\|u(z_i) - u_{i,h}\|_{H^1(\omega_i)}}.$$

5.1 *Test problem 1*

In this part, we consider the case that there exist boundary layers only in Z direction. Let $\Omega = [0,1]^3$, and the exact solution of Equation (1) is

$$u(x,y,z) = \sin(\pi x)\sin(\pi y)(1 - \exp^{-(1-z)/e})\cos(\pi z), \tag{22}$$

where coefficient $e = 0.02$, the forcing term f is computed by $f = -\triangle u + \kappa u$, $\kappa = 100$.

In Table 1, we present the effective indices EI_I and EI_{ω_i}, the total number of elements for triangulation $\sum_i \mathrm{DOF}_i$, and the number of nodes LZ along Z direction from Levels 1–5 with initial mesh $LX = LY = LZ = 10$. EI_I varies from 0.59656 to 1.55541, which is close to 1, and $\mathrm{EI}_{\omega_i}(i \in I)$ tends to 3–5 successively level by level. They indicate that our error indicators are effective and acceptable. The relative errors of both the ADSA and the DSA in the L^2-norm and H^1-seminorm under almost the same number of elements are given in Table 2. Specifically, the relative errors of the ADSA in the L^2-norm and H^1-seminorm are much smaller than those of the DSA. It indicates that our adaptive DSA is more effective and accurate than the DSA with the uniform mesh.

Table 1. Test problem 1: convergence analysis for the ADSA.

Level	EI_I	EI_{ω_i}	$\sum_i \mathrm{DOF}_i$	LZ
1	0.59656	2.98954–5.02211	3297	13
2	0.99214	3.35397–5.19157	6532	14
3	1.4083	3.68445–5.12866	9800	15
4	1.55541	3.51295–5.14616	15711	17
5	1.37741	3.58392–5.13652	23885	20

Table 2. Test problem 1: the relative errors of both the ADSA and the DSA.

	$\sum_i \mathrm{DOF}_i$	LZ	$\|u-u_{h,d}\|_{L^2(\Omega)}/\|u\|_{L^2(\Omega)}$	$\|u-u_{h,d}\|_{H^1(\Omega)}/\|u\|_{H^1(\Omega)}$
ADSA method	23885	20	0.031695	0.12096
DSA method	23750	20	0.34171	0.67221

Figure 3 displays Z partition value of the exact solution (22) and the adaptive mesh along Z direction successively level by level. It shows that the gradient magnitude of domain close to plane $z = 1$ is bigger than others in Figure 3(a). In Figure 3(b), we observe that the initial mesh is refined heavily in the domain close to plane $z = 1$, which is due to the big magnitude of gradient at plane $z = 1$ nearby. The biggest grid size is $d_i = 0.1$ and the smallest $d_i = 0.003125$ in the process of refining mesh when Level $= 5$. Therefore, our adaptive refinement method along Z direction is efficient. The adaptive mesh, numerical solution and exact solution of different $\mathcal{T}_{i,h}(i = 14, 18)$ in the ADSA are reported in Figures 4 and 5. We find that though the domain near the plane $z = 1$ is refined heavily in Z direction, the mesh in $\mathcal{T}_{i,h}(i = 14, 18)$ is not refined obviously because of good regularity in X–Y plane direction, where

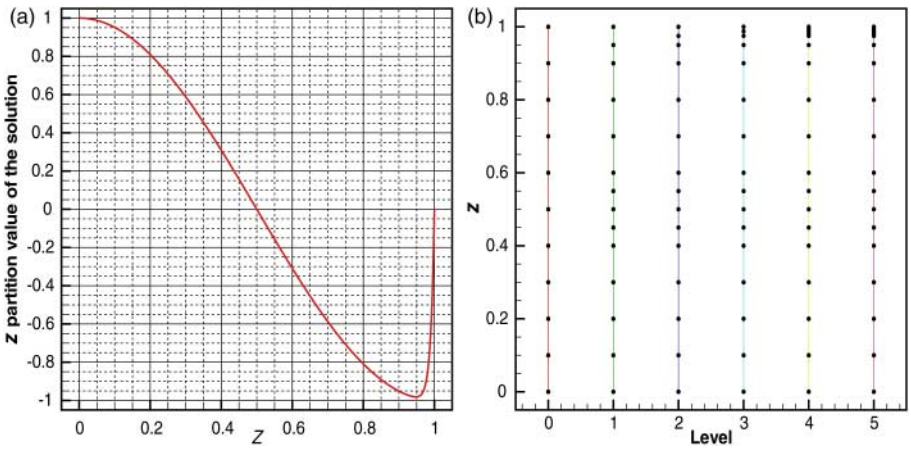

Figure 3. Test problem 1. (a) Z partition value of the true solution and (b) adaptive mesh for the Z direction.

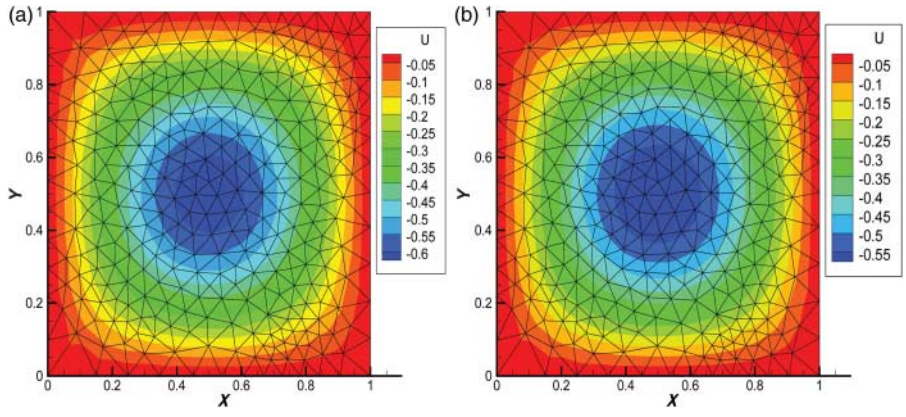

Figure 4. Test problem 1, adaptive mesh and solution for $\mathcal{T}_{14,h}$ (a) numerical solution and (b) exact solution.

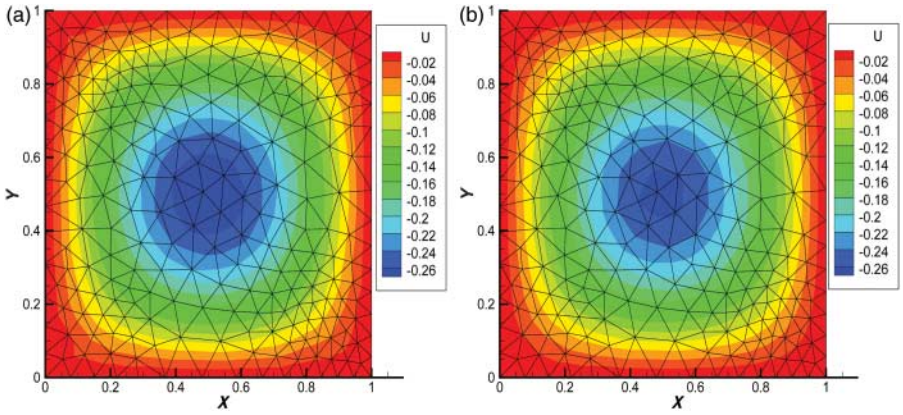

Figure 5. Test problem 1, adaptive mesh and solution for $\mathcal{T}_{18,h}$. (a) numerical solution and (b) exact solution.

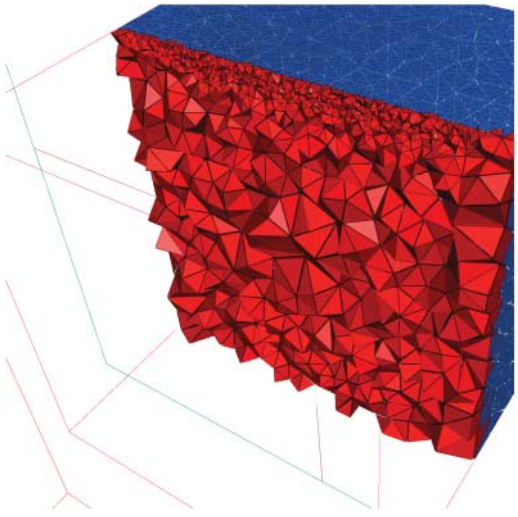

Figure 6. Test problem 1, adaptive mesh of refining tetrahedron element.

the smallest $h_{\mathrm{T}} = 0.038087$ (h_{T} denotes the longest edge of triangle in $\mathcal{T}_{i,h}$) in $\mathcal{T}_{14,h}$ and the smallest $h_{\mathrm{T}} = 0.043084$ in $\mathcal{T}_{18,h}$. This reveals that our adaptive refinement method is also efficient in X–Y plane direction.

The tetrahedron element mesh of the 3DAFEM by refining adaptively in longitudinal section $x = 0.5$ is demonstrated in Figure 6. The mesh of the domain near the plane $z = 1$ is refined heavily with the smallest $h_{\mathrm{T}'} = 0.0042481$ ($h_{\mathrm{T}'}$ denotes the longest edge of tetrahedron element). Moreover, in Figure 7, the relative errors of both the ADSA and the 3DAFEM as the number of vertices increases are provided. It is obvious that the relative error of the ADSA in the H^1-seminorm is much smaller than those of the 3DAFEM. However, when the number of vertices is small, the relative error of the ADSA in the L^2-norm is smaller than those of the 3DAFEM. When the number of vertices is getting bigger, the situation is just the reverse. In other words, the numerical results reflect our ADSA can obtain the more accurate approximate solution in the sense of the H^1-seminorm compared with the 3DAFEM in this case.

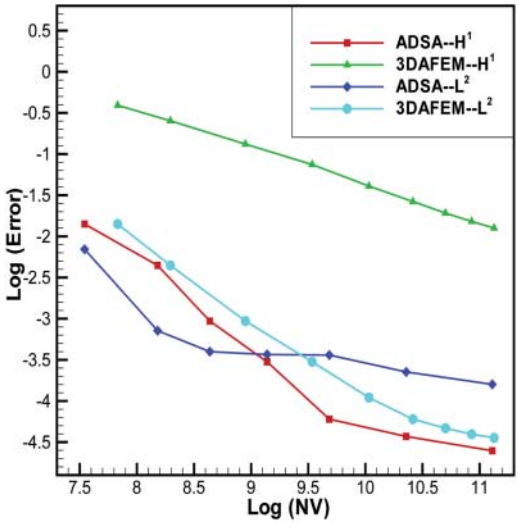

Figure 7. Test problem 1, comparisons of relative errors of both the ADSA and the 3DAFEM.

5.2 *Test problem 2*

In this part, we consider the case that there exist the boundary layers only in X–Y plane direction. Let $\Omega = [0, 1]^3$, and the exact solution of Equation (1) is

$$u(x, y, z) = (1 - \exp^{-(1-x)/e})(1 - \exp^{-(1-y)/e}) \cos(\pi(x + y)) \sin(\pi z), \tag{23}$$

where coefficient $e = 0.02$, the forcing term f is computed by $f = -\triangle u + \kappa u$, $\kappa = 100$.

In Table 3, we present the effective indexes EI_I and EI_{ω_i}, the total number of elements for triangulation $\sum_i \mathrm{DOF}_i$, and the number of nodes LZ along Z direction from Levels 1–5 with initial mesh $LX = LY = LZ = 10$. EI_I varies from 0.67395 to 0.85479, which is close to 0.7, and $\mathrm{EI}_{\omega_i} (i \in I)$ is included in the interval (1,3) successively level by level. It reveals that our error indicators are effective and acceptable. The relative errors of both the ADSA and the DSA in the L^2-norm and H^1-seminorm under almost the same number of elements are reported in Table 4. Specifically, it displays that the relative errors of the ADSA in the L^2-norm and H^1-seminorm are much smaller than those of the DSA. This implies that our ADSA is effective and gets the more accurate approximate solution than the DSA under the uniform mesh.

Figure 8 displays the adaptive mesh along Z direction successively level by level. The initial mesh of Z direction is not refined heavily, which is because that the Z partition of the exact solution has better regularity. The biggest grid size is $d_i = 0.1$ and the smallest $d_i = 0.025$ in the process of refining mesh when Level= 5. The X–Y partition value of the exact solution (23) is exhibited in Figure 9. From Figure 9 we can see that the magnitude of gradient at plane $x = 1$ and plane

Table 3. Test problem 2: convergence analysis for the ADSA.

Level	EI_I	EI_{ω_i}	$\sum_i \mathrm{DOF}_i$	LZ
1	0.74061	2.38393–2.60642	3756	12
2	0.77126	1.84961–2.43395	7674	14
3	0.67395	1.57341–2.24961	15005	20
4	0.72289	1.39863–2.30427	23057	22
5	0.85479	0.96182–1.61569	33007	24

Table 4. Test problem 2: the relative errors of both the ADSA and the DSA.

	$\sum_i \mathrm{DOF}_i$	LZ	$\frac{\|u-u_{h,d}\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\frac{\|u-u_{h,d}\|_{H^1(\Omega)}}{\|u\|_{H^1(\Omega)}}$
ADSA method	33007	24	0.186187	0.0325865
DSA method	33534	24	0.406214	0.0577752

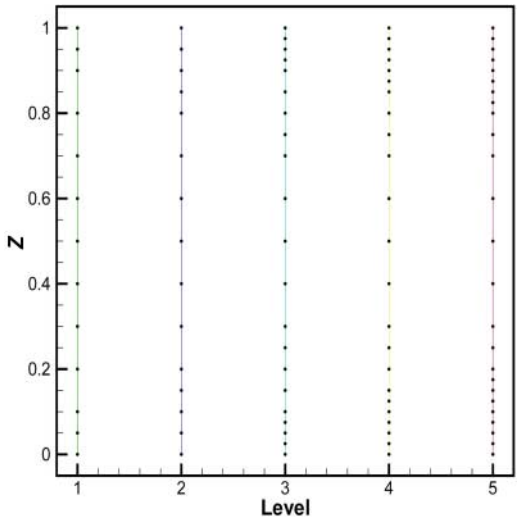

Figure 8. Test problem 2, adaptive mesh for the Z direction.

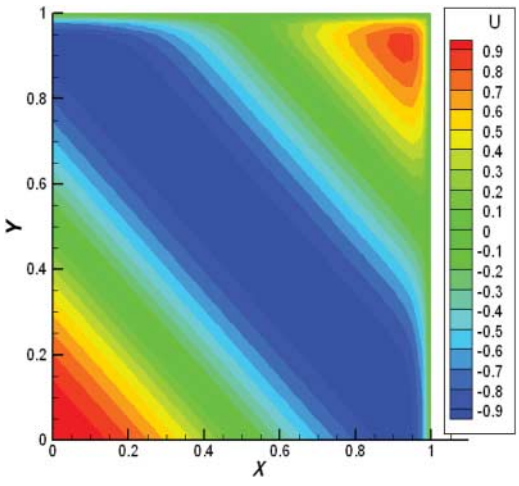

Figure 9. Test problem 2, exact solution.

$y = 1$ nearby is bigger than others. The adaptive mesh, numerical solution and exact solution of different $\mathcal{T}_{i,h}(i = 10, 20)$ in the ADSA are reported in Figures 10 and 11. It is observed that the mesh of the domain at both line $x = 1$ and line $y = 1$ nearby is refined heavily, where the smallest $h_{\mathrm{T}} = 0.011029$ in $\mathcal{T}_{14,h}$ and the smallest $h_{\mathrm{T}} = 0.012268$ in $\mathcal{T}_{18,h}$. It reveals that our ADSA method is able to recognize the regions with large gradient magnitude of the solution.

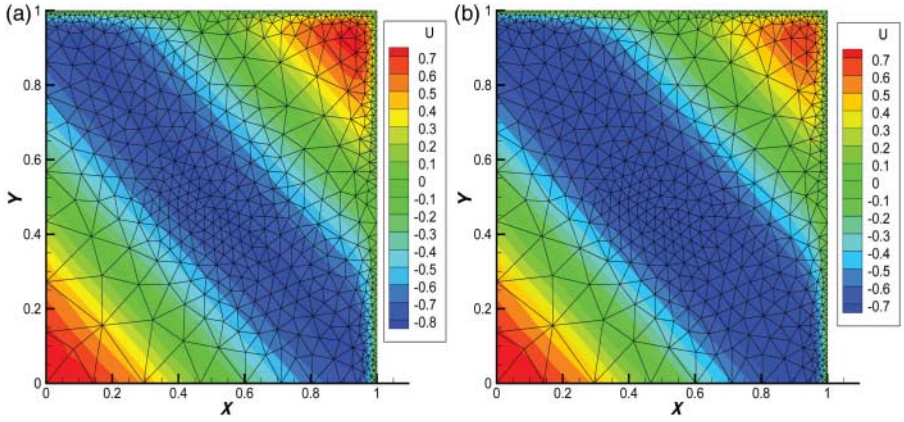

Figure 10. Test problem 2, adaptive mesh and solution for $\mathcal{T}_{10,h}$. (a) numerical solution and (b) exact solution.

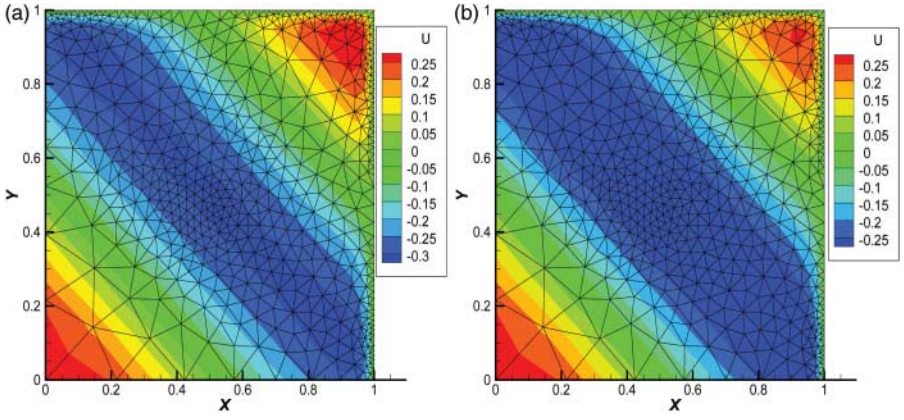

Figure 11. Test problem 2, adaptive mesh and solution for $\mathcal{T}_{20,h}$. (a) numerical solution and (b) exact solution.

The tetrahedron element mesh of the 3DAFEM by refining adaptively in cross section $z = 0.5$ is illustrated in Figure 12. The mesh of the domain near the plane $x = 1$ and plane $y = 1$ is refined heavily with the smallest $h_{T'} = 0.0028394$. Finally, the relative errors of both the ADSA and the 3DAFEM as the number of vertices increases are shown in Figure 13. From Figure 13, there is the same conclusion as Figure 7 in test problem 1. That is to say that our ADSA can get the more accurate approximate solution in the sense of the H^1-seminorm compared with the 3DAFEM in this case.

5.3 *Test problem 3*

In this part, we consider the case that there exists the local peak in X–Y plane direction and the boundary layers in Z direction. Let $\Omega = [0, 1]^3$, and the exact solution of Equation (1) is:

$$u(x, y, z) = \exp^{-\alpha((x-xc-0.5z)^2+(y-yc-0.5z)^2)}(1 - \exp^{-(1-z)/e})\cos(\pi z), \tag{24}$$

where coefficients $\alpha = 100, xc = 0.25, yc = 0.25, e = 0.02$, the forcing term f is computed by $f = -\triangle u + \kappa u, \kappa = 100$.

In Table 5, we show the effective indices EI_I and EI_{ω_i}, the total number of elements for triangulation $\sum_i \mathrm{DOF}_i$, and the number of nodes LZ along Z direction from Levels 1–5 with initial

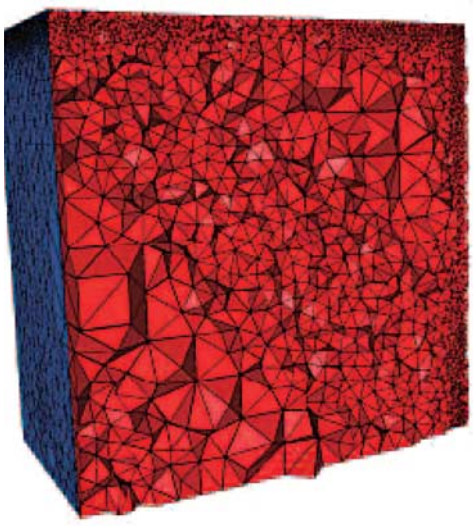

Figure 12. Test problem 2, adaptive mesh of refining tetrahedron element.

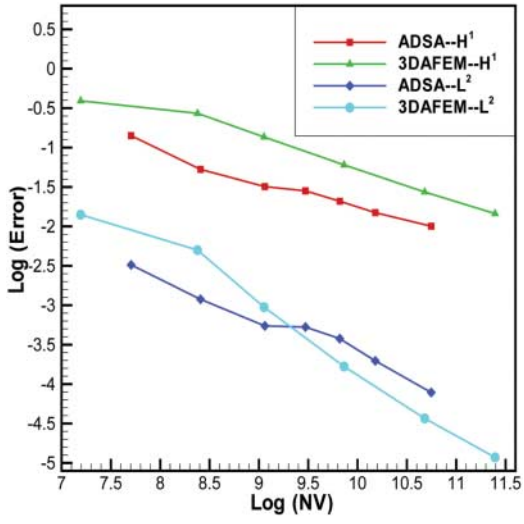

Figure 13. Test problem 2, comparisons of relative errors of both the ADSA and the 3DAFEM.

mesh $LX = LY = LZ = 10$. We find that EI_I varies from 1.23118 to 1.74526, which is close to a constant. And every $\mathrm{EI}_{\omega_i}(i \in I)$ tends to 1 nearby successively level by level. They imply that the error indicators presented by us are effective and acceptable. Table 6 reports the relative errors of both the ADSA and the DSA in the L^2-norm and H^1-seminorm under almost the same number of elements. Specifically, it displays that the relative errors of the ADSA in the L^2-norm and H^1-seminorm are much smaller than those of the DSA. It indicates that our ADSA is effective, and can obtain the more accurate approximate solution than the DSA under the uniform mesh.

Figure 14 shows the adaptive mesh along Z direction successively level by level. We observe that the initial mesh is refined in the domain close to plane $z = 1$, where the biggest $d_i = 0.1$ and the smallest $d_i = 0.003125$ when Level $= 5$, which is due to much bigger gradient magnitude of Z direction at plane $z = 1$ nearby. It reveals that our adaptive refinement method along Z direction

Table 5. Test problem 3: convergence analysis for the ADSA.

Level	EI_I	EI_{ω_i}	$\sum_i \mathrm{DOF}_i$	LZ
1	1.23118	1.00385–1.67666	3250	12
2	1.52287	0.63525–1.54375	5418	13
3	1.74526	0.61758–1.23754	8080	14
4	1.62548	0.62829–1.12359	11749	16
5	1.34355	0.550359–1.14391	19346	21

Table 6. Test problem 3: the relative errors of both the ADSA and the DSA.

	$\sum_i \mathrm{DOF}_i$	LZ	$\frac{\lVert u-u_{h,d}\rVert_{L^2(\Omega)}}{\lVert u\rVert_{L^2(\Omega)}}$	$\frac{\lVert u-u_{h,d}\rVert_{H^1(\Omega)}}{\lVert u\rVert_{H^1(\Omega)}}$
ADSA method	19346	21	0.103889	0.224799
DSA method	19360	21	0.600968	0.613576

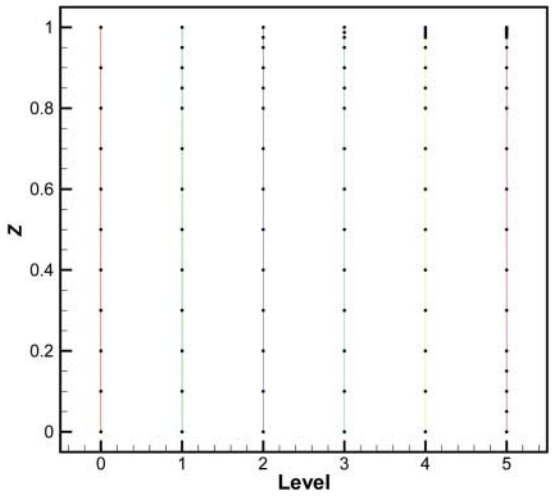

Figure 14. Test problem 3, adaptive mesh for the Z direction.

is efficient. The adaptive mesh, numerical solution and exact solution of different $\mathcal{T}_{i,h}(i = 8, 17)$ in the ADSA are shown in Figures 15 and 16. It is observed that the mesh of the domain near the point $(xc + 0.5z_i, yc + 0.5z_i)$ is refined heavily, and our adaptive method is able to recognize the regions with large gradients of the solution.

The tetrahedron element mesh of the 3DAFEM by refining adaptively in longitudinal section $x + y = 0$ is demonstrated in Figure 17. We found that the meshes of the domain near the line segment that joins the point $(xc + 0.5z_i, yc + 0.5z_i, 0)$ and the point $(xc + 0.5z_i, yc + 0.5z_i, 1)$ are refined heavily. Finally, the relative errors of both the ADSA and the 3DAFEM as the number of vertices increases are given in Figure 18. From Figure 18, we can see that the relative error of the ADSA in the H^1-seminorm is almost the same as those of the 3DAFEM, and the relative error of the ADSA in the L^2-norm is a little greater than those of the 3DAFEM. It indicates that for the problem with singularity both in X–Y plane direction and in Z direction, our ADSA method can also recognize the regions where the meshes need to be refined. Though in this case we obtain the almost same approximation solution in the H^1-seminormin like the 3DAFEM, only 2D meshes are involved in the ADSA. It saves an enormous amount of time and memory in implementation, which is a great merit compared with the 3DAFEM.

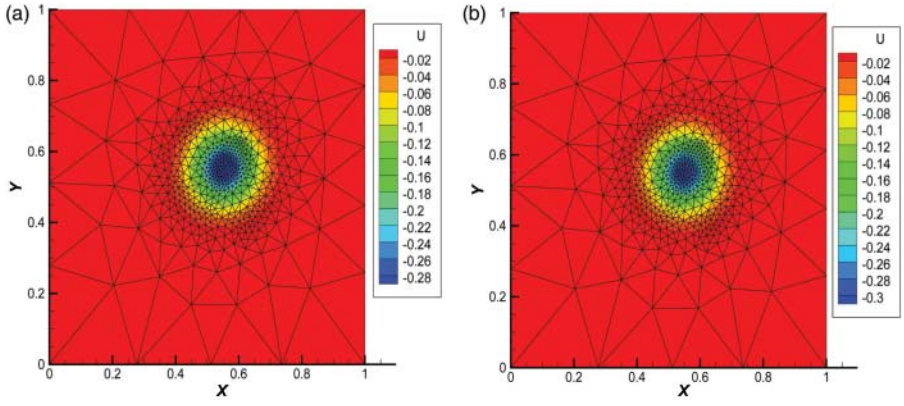

Figure 15. Test problem 3, adaptive mesh and solution for $\mathcal{T}_{8,h}$. (a) numerical solution and (b) exact solution.

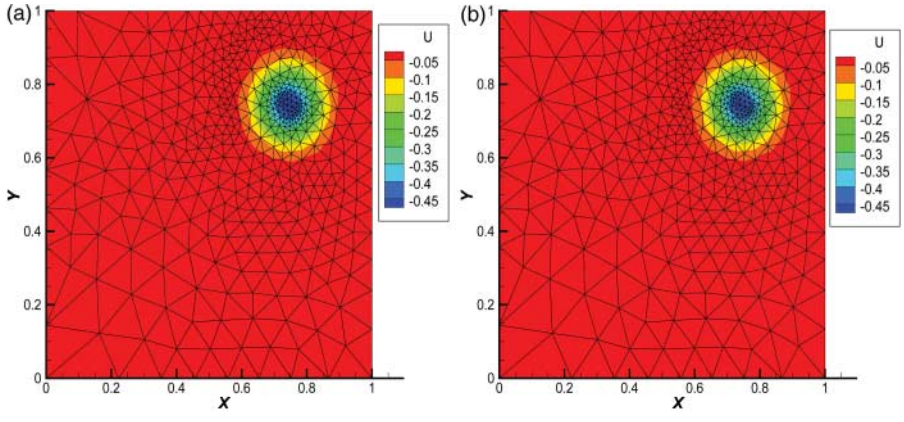

Figure 16. Test problem 3, adaptive mesh and solution for $\mathcal{T}_{17,h}$. (a) numerical solution and (b) exact solution.

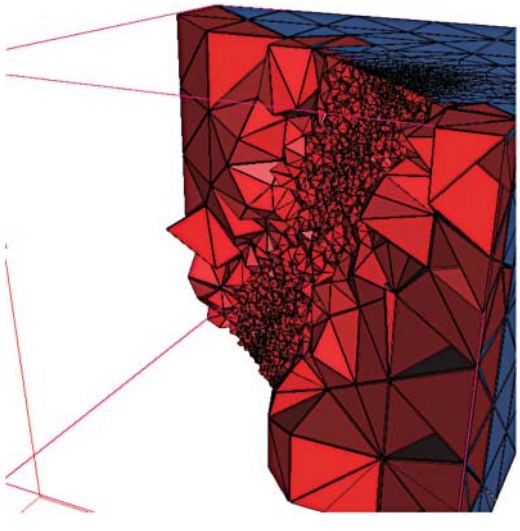

Figure 17. Test problem 3, adaptive mesh for refining tetrahedron element.

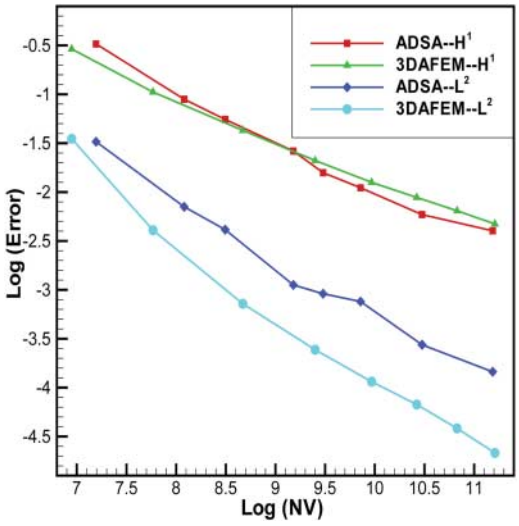

Figure 18. Test problem 3, comparisons of relative errors of both the ADSA and the 3DAFEM.

6. Conclusion

In this paper, we propose an ADSA for 3D elliptic equations based on two error estimators both in X–Y plane and in Z directions. We point out that the ADSA can be implemented in parallel like the previous DSA. What is more, we can extend the adaptive dimension splitting algorithm to other models such as the convection–diffusion equation, incompressible flow. Our algorithm is suitable only for domains of simple shape, but there are possibilities that the algorithm can be applied to problems with more complex geometries. For example, we can impose boundary conditions either via the fictitious domain method or via directional adjustment of the grid at the boundary. These issues will be under research.

Acknowledgements

This work is supported by NSFC (Grant No. 11171269), the Ph.D. Programs Foundation of Ministry of Education of China (20110201110027) and the Fundamental Research Funds for the Central Universities.

References

[1] M. Ainsworth and J.T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley & Sons, New York, 2000.
[2] I. Babuska and W.C. Rheinboldt, *A posteriori error estimates for the finite element method*, Internat. J. Numer. Methods Engrg. 12 (1978), pp. 1597–1615.
[3] I. Babuska and W.C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal. 15 (1978), pp. 736–754.
[4] I. Babuska, O.C. Zienkiewicz, J. Gago, and E.A.Oliviera, *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Wiley, New York, 1986.
[5] R.E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comp. 44 (1985), pp. 283–301.
[6] T.F. Chan and T.P. Mathew, *Domain decomposition algorithms*, Acta Numer. 3 (1994), pp. 61–143.
[7] Y. Hou and H. Wei, *Dimension splitting algorithm for three dimensional elliptic equation*, Int. J. Comput. Math. 89(1) (2012), pp. 112–127.
[8] F. Hecht and O. Pironneau, FreeFEM++(Version 3.9-2)(2010); available at http://www.freefem.org/ff++/ftp.

[9] K. Li and A. Huang, *Mathematical aspect of the stream-function equations of compressible turbomachinery flows and their finite element approximation using optimal control*, Comput. Methods Appl. Mech. Engrg. 41 (1983), pp. 175–194.
[10] K. Li, A. Huang, and W. Zhang, *A dimension split method for the 3-D compressible Navier-Stokes equations in turbomachine*, Comm. Numer. Methods Engrg. 18(1) (2002), pp. 1–14.
[11] K. Li, W. Zhang, and A. Huang, *An asymptotic analysis method for the linearly shell theory*, Sci. China Ser. A 49(8) (2006), pp. 1009–1047.
[12] K. Li and X. Shen, *A dimensional splitting method for linearly elastic shell*, Int. J. Comput. Math. 84 (2007), pp. 807–824.
[13] K. Li and D. Liu, *Dimension splitting method for 3D rotating compressible Navier-Stokes equations in turbomachinery*, Int. J. Numer. Anal. Model. 6 (2009), pp. 420–439.
[14] H. Si, *TetGen Users' guide: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator*; available at http://tetgen.berlios.de.
[15] L. Song, Y. Hou, and H. Zheng, *Adaptive local postprocessing finite element method for the Navier-Stokes equations*, J. Sci. Comput. 55 (2013), pp. 255–267.
[16] L. Song, Y. Hou and H. Zheng, *Adaptive variational multiscale method for the Stokes equations*, Int. J. Numer. Methods Fluids 71 (2013), pp. 1369–1381.
[17] J.R. Shewchuk, *Tetrahedral mesh generation by delaunay refinement proceeding of the fourteenth annual symposium on computational geometry*, Minneapolis, MN, 1998, pp. 86–95.
[18] A. Toselli and O. Widlund, *Domain Decomposition Methods-Algorithms and Theory*, Springer-Verlag, Berlin, 2005.
[19] R. Verfurth, *A posteriori error estimators for the Stokes equations*, Numer. Math. 55 (1989), pp. 309–325.
[20] R. Verfurth, *A posteriori error estimation and adaptive mesh refinement techniques*, J. Comput. Appl. Math. 50 (1994), pp. 67–83.
[21] O.C. Zienkiewicz and J.Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates, Part 1: The recovery technique*, Internat. J. Numer. Methods Engrg. 33 (1992), pp. 1331–1364.
[22] O.C. Zienkiewicz and J.Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates, Part 2: Error estimates and adaptivity*, Internat. J. Numer. Methods Engrg. 33 (1992), pp. 1365–1382.
[23] H. Zheng, Y. Hou, and F. Shi, *Adaptive variational multiscale methods for incompressible flow based on two local Gauss integrations*, J. Comput. Phys. 229 (2010), pp. 7030–7041.
[24] H. Zheng, Y. Hou, and F. Shi, *A posteriori error estimates of stabilization of low-order mixed finite elements for incompressible flow*, SIAM J. Sci. Comp. 32(3) (2010), pp. 1346–1361.