Dimension splitting algorithm for a three-dimensional elliptic equation

Yanren Hou and Hongbo Wei*

School of Science, Xian Jiaotong University, Xian 710049, China

(Received 15 July 2011; revised version received 18 September 2011; accepted 29 September 2011)

This paper presents a finite-element dimension splitting algorithm (DSA) for a three-dimensional (3D) elliptic equation in a cubic domain. The main idea of DSA is that a 3D elliptic equation can be transformed into a series of two-dimensional (2D) elliptic equations in the $X$–$Y$ plane along the $Z$-direction. The convergence speed of the DSA for a 3D elliptic equation depends mainly on the mesh scale of the $Z$-direction. $P_2$ finite-element discretization in the $Z$-direction for DSA is adopted to accelerate the convergence speed of DSA. The error estimates are given for DSA applying $P_1$ or $P_2$ finite-element discretization in the $Z$-direction. Finally, some numerical examples are presented. We apply the parallel solving technology to our numerical examples and obtain good parallel efficiency. These numerical experiments test and verify theoretical results.

Keywords: dimension splitting algorithm; elliptic equation; finite-element method; parallel solving technology; error estimate

2000 AMS Subject Classifications: 65Y05; 65N30

1. Introduction

With the development of parallel computers and technology, the domain decomposition method has been widely used to solve the partial differential equations, especially in computational fluid dynamics. There are numerous works devoted to the development of the domain decomposition method [11]. Domain decomposition methods are techniques of solving partial differential equations based on decomposing the spatial domain of the problem into several subdomains. The principal advantages enhance parallelism and localized treatment of complex and irregular geometries, singularities and anomalous regions. These techniques can frequently be applied directly to the partial differential equations and discretizations of the differential equations (either by finite-difference method, finite-element method (FEM) or spectral method). There are many efficient numerical methods for the three-dimensional (3D) boundary-value problems [1,9]. Recently, a new method called the dimension split method is presented by Li and Huang [4]. Li and Huang [4] derived a nonlinear boundary-value problem satisfied by the stream function defined on the
stream surface and its finite-element approximation. Kaitai et al. [3] and Li and Shen [7] presented a dimensional splitting method for the linear elastic shell based on differential geometry and tensor analysis. Li and coworkers [5, 6, 8] applied the dimensional splitting method to the compressible Navier–Stokes equations. The dimensional splitting method is different from the domain decomposition method. We have to solve a 3D problem in each subdomain in the domain decomposition method, but we only solve a two-dimensional (2D) problem in each subdomain.

In this paper, we consider the following 3D elliptic equation with homogenous Dirichlet boundary condition defined in a cubic domain

$$\Omega_1 = [0, 1]^3 \subset R^3$$

$$-\Delta u + \kappa u = f \quad \forall (x, y, z) \in \Omega,$$

$$u|_{\partial \Omega} = 0 \quad \forall (x, y, z) \in \partial \Omega,$$

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

Remark 1 For parabolic equations, for example, the heat equation

$$u_t - \Delta u = f \quad \forall (x, y, z) \in \Omega, \quad 0 < t \leq T,$$

$$u = 0 \quad \forall (x, y, z, t) \in \partial \Omega \times (0, T],$$

$$u(x, y, z, 0) = u^0(x, y, z) \quad \forall (x, y, z) \in \Omega,$$

a backward Euler discretization with respect to time variable leads to

$$\frac{u^n - u^{n-1}}{k_n} - \Delta u^n = f(t_n),$$

where $k_n = t_n - t_{n-1}, n = 0, 1, \ldots$. For every time step $n$, Equation (3) can be viewed as the elliptic equation like Equation (1), with $\kappa = 1/k_n$.

In this paper, we introduce a dimension splitting algorithm (DSA) for the 3D elliptic equation in a cubic domain. The basic idea of DSA is that the 3D elliptic equation is transformed into a series of 2D elliptic equations with the $P_1$ finite-element discretization in the $Z$-direction and the $P_1$ finite-element discretization in the $X$–$Y$ plane direction. $P_2$ finite-element discretization in the $Z$-direction for DSA is adopted to accelerate the convergence speed of DSA. One of the advantage of DSA is that a series of 2D elliptic equations can be easily solved in parallel with different processors. Another advantage of DSA is that only the 2D meshes are involved, which is much easier to be constructed than the 3D meshes.

The outline of the paper is as follows. In Section 2, we give the formulations of the DSA for the 3-D elliptic equation in detail. We find that the Jacobi-like iterative scheme of DSA is convergent in both $L^2(\Omega)$ space and $H^1(\Omega)$ space. The error analysis is also given at the same time in this section. In Section 3, some numerical examples and numerical implementation using parallel solving technology will be reported. We end the paper with a short concluding section.

2. Dimension splitting algorithm

For the given index set $I \subset \{0, 1, 2, \ldots\}$, let us denote

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

In our case, $\omega = \Omega_i = [0, 1]^2, I_d = [0, 1]$. We denote

$$u^i(x, y) = u(x, y, z)|_{z=d_i} \quad \forall (x, y) \in \omega.$$
If we denote
\[ \Delta_2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad \nabla_2 = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)^\top, \]

\( u^i \) satisfies the following 2D elliptic equation:
\[ -\Delta_2 u^i + \kappa u^i = f^i + \frac{\partial^2 u^i}{\partial z^2} \forall (x, y) \in \omega, \]
\[ u^i |_{\partial \omega} = 0 \forall (x, y) \in \partial \omega, \tag{4} \]

where \( f^i = f(x, y, z) |_{z=d_i} \), \( \partial^2 u^i / \partial z^2 = \partial^2 u / \partial z^2 |_{z=d_i} \), and \( i \in I \).

We first consider a simple case that \( d_i = i * d, \) where \( d = 1/N \) and \( N > 0 \) is a given positive integer. Suppose \( T_h^2 = \{ K \} \) is a quasi-regular triangulation of \( \omega \) with mesh size \( h = \max \{ h_K : K \in T_h^2 \} \) and \( h_K = \text{diam} K \) and \( X_h \subset X \) is the \( P_1 \) finite-element subspace. Let \( T_d^1 \) be a quasi-regular triangulation of \([0, 1]\) with mesh size \( d \), and \( Z_d = \text{span}(\phi_i, i \in I) \) is the corresponding piecewise linear one-dimensional (1D) finite-element subspace (or piecewise \( P_2 \) continuous finite-element subspace), where \( \phi_i \) is the piecewise linear basis function (or piecewise \( P_2 \) basis function) at node \( i \). We define a 3D finite-dimensional subspace \( X_{hd} = X_h \times Z_d \subset H_0^1(\omega) \).

### 2.1 \( P_1 \) finite-element discretization in the Z-direction

Let us denote by \( I_h^2 \) the 2D finite-element interpolation operator on \( \omega \) and by \( I_d^1 \) the 1D finite-element interpolation on \([0, 1]\). For the given function \( u \in H_0^1(\Omega) \), we define its 3D finite-element interpolation as
\[ I_{hd}u = I_h^2(I_d^1u) = \sum_{i \in I} u_{i,h} \phi_i, \tag{5} \]

with \( I_d^1u = \sum_{i \in I} u(x, y, d_i) \phi_i \subset Z_d \) and \( u_{i,h} = I_h^2u(x, y, d_i) \in X_h \), where \( \phi_i \) is the piecewise linear basis function on \( I_d \).

#### 2.1.1 Numerical iterative scheme

First, we choose \( P_1 \) finite element in the Z-direction to an approximate solution of Equation (4). Now, our goal is to seek the finite-element approximate solution of Equation (4) in the following form:
\[ u_{hd} = \sum_{i \in I} u_{i,h} \phi_i, \quad \text{with } u_{i,h} \in X_h, \tag{6} \]

such that \( \forall \phi_i \in Z_d = \{ v_d \in C^0([0, 1]) | v_d|_K \in P_1 \ \forall K \in T_d^1 \} \ \forall v \in X_h \), the following equation holds:
\[ \left( \left( \frac{\partial u_{hd}}{\partial z}, \phi_i \right)_z , v \right)_x + a((u_{hd}, \phi_i)_z, v)_x + \kappa((u_{hd}, \phi_i)_z, v)_x = ((f, \phi_i)_z, v)_x, \tag{7} \]

where \( \langle \cdot, \cdot \rangle_z \) is the scalar product in \( I_d \) and \( \langle \cdot, \cdot \rangle_{xy} \) is the scalar product in \( \omega \) and \( a(\cdot, \cdot)_{xy} = (\nabla_2, \nabla_2)_{xy} \).
We apply the Jaccobi iterative method [10] to solve Equation (9). Assuming that $u_n$.
Then, from Equation (10), we have

Let us denote

The convergence and error analysis

Simple calculation admits

\[
\left( \left( 2 + \frac{2\kappa d^2}{3} \right) u_{i,h}^{n+1}, v \right)_{xy} + \frac{2d^2}{3} a(u_{i,h}^{n+1}, v)_{xy} = d((f, \phi_i)_z, v)_{xy} + \left( 1 - \frac{\kappa d^2}{6} \right) u_{i-1,h}^{n} + \left( 1 - \frac{\kappa d^2}{6} \right) u_{i+1,h}^{n}, v \right)_{xy}
\]

\[- \frac{d^2}{6} a(u_{i+1,h}^{n} + u_{i-1,h}^{n}, v)_{xy} \quad \forall i \in I, \ n = 0, 1, 2, \ldots.
\]

Remark 2 By computation, the first term on the right-hand side of the above equation may be approximated by

\[
d((f, \phi_i)_z, v)_{xy} \approx \frac{d^2}{6} (4f_{i,h} + f_{i-1,h} + f_{i+1,h}, v)_{xy}.
\]

2.1.2 The convergence and error analysis

Let us denote

\[
u_{i,h}^{n+1} = \sum_{i \in I} u_{i,h}^{n+1}, \phi_i
\]

as the $(n + 1)$th iterative solution. In this section, we will show that, for arbitrary iterative initial value $u^n_{i,h} \in X_{ih}$, $\{u^n_{i,h}\}_{n=0}^{\infty}$ is a convergent sequence in both $L^2(\Omega)$ and $H^1(\Omega)$. For simplicity, we denote

\[
e_{i,h}^{n+1} = u_{i,h}^{n+1} - u_{i,h}^{n} \quad \forall i \in I.
\]

Then, from Equation (10), we have

\[
\left( \left( 2 + \frac{2\kappa d^2}{3} \right) e_{i,h}^{n+1}, v \right)_{xy} + \frac{2d^2}{3} a(e_{i,h}^{n+1}, v)_{xy} = \left( 1 - \frac{\kappa d^2}{6} \right) e_{i-1,h}^{n} + \left( 1 - \frac{\kappa d^2}{6} \right) e_{i+1,h}^{n}, v \right)_{xy}
\]

\[- \frac{d^2}{6} a(e_{i+1,h}^{n} + e_{i-1,h}^{n}, v)_{xy} \quad \forall i \in I, \ n = 0, 1, 2, \ldots.
\]
Taking \( v = e_{i,h}^{n+1} \) in Equation (14) and denoting \( E_0^{n+1} = \max_{ief} \| e_{i,h}^{n+1} \|^2, E_1^{n+1} = \max_{ief} \| \nabla e_{i,h}^{n+1} \|^2, \) where \( \| \cdot \| \) denotes the \( L^2(\omega) \)-norm.

We admit

\[
 \left( 2 + \frac{2 \kappa d^2}{3} \right) \| e_{i,h}^{n+1} \|^2 + \frac{2 d^2}{3} \| \nabla e_{i,h}^{n+1} \|^2 \leq \left( 2 - \frac{\kappa d^2}{3} \right) \sqrt{E_0^n} \| e_{i,h}^n \| + \frac{d^2}{3} \sqrt{E_1^n} \| \nabla e_{i,h}^n \|. \tag{15}
\]

Therefore,

\[
 \left( 2 + \frac{2 \kappa d^2}{3} \right) E_0^{n+1} + \frac{2 d^2}{3} E_1^{n+1} \leq \left( 1 - \frac{\kappa d^2}{6} \right) (E_0^n + E_0^{n+1}) + \frac{d^2}{6} (E_1^n + E_1^{n+1}). \tag{16}
\]

Let

\[
 -\kappa d^2 \leq \alpha \leq 2 - \frac{4 \kappa d^2}{3}, \quad \beta = \frac{1}{6 - \kappa d^2} > 0,
\]

we have

\[
 (1 + \frac{5}{6} \kappa d^2 + \alpha)(E_0^{n+1} + \beta d^2 E_1^{n+1}) \leq (1 - \frac{1}{6} \kappa d^2)(E_0^n + \beta d^2 E_1^n). \tag{18}
\]

The Gronwall inequality [10] guarantees

\[
 E_0^{n+1} + \beta d^2 E_1^{n+1} \leq (1 + \gamma)^{-n}(E_0^1 + \beta d^2 E_1^1) \quad \forall n \geq 0,
\]

where

\[
 \gamma = (\kappa d^2 + \alpha)6\beta > 0.
\]

From inequality [19], one has

\[
 \sqrt{E_0^{n+1}} + \sqrt{\beta d \sqrt{E_1^{n+1}} \leq \sqrt{2}(1 + \gamma)^{-n/2} \sqrt{(E_0^1 + \beta d^2 E_1^1)}, \quad \forall n \geq 0.
\]

Now, for any given \( p \in \mathcal{N} \)

\[
 \| u_{i,h}^{n+p+1} - u_{i,h}^n \| + \sqrt{\beta d} \| \nabla (u_{i,h}^{n+p+1} - u_{i,h}^n) \|
\]

\[
 \leq \sum_{j=0}^p \left( \| u_{i,h}^{n+j+1} - u_{i,h}^{n+j} \| + \sqrt{\beta d} \| \nabla (u_{i,h}^{n+j+1} - u_{i,h}^{n+j}) \| \right)
\]

\[
 \leq \sum_{j=0}^p \left( \sqrt{E_0^{n+j+1}} + \sqrt{\beta d \sqrt{E_1^{n+j+1}}} \right)
\]

\[
 \leq \sqrt{2}(E_0^1 + \beta d^2 E_1^1)(1 + \gamma)^{-n/2} \sum_{j=0}^p (1 + \gamma)^{-j/2}
\]

\[
 \leq \sqrt{2}(E_0^1 + \beta d^2 E_1^1)\gamma^{-1}(1 + \gamma + \sqrt{1 + \gamma})(1 + \gamma)^{-n/2}. \tag{20}
\]

It is obvious that the sequence \( \{ u_{i,h}^n \}_{n=0}^\infty \) converges to certain \( u_{hd} \in X_{hd} \) in both \( L^2(\Omega) \) space and \( H_0^1(\Omega) \) space. Therefore, we have the following theorem.

**Theorem 2.1** The sequence \( \{ u_{i,h}^n \}_{n=0}^\infty \), with arbitrary iterative initial value \( u_{0}^0 \in X_{hd} \), is a convergent sequence in both \( L^2(\Omega) \) space and \( H_0^1(\Omega) \) space, and

\[
 \| (u_{i,h}^n - u_{hd}) \| + \| \nabla (u_{i,h}^n - u_{hd}) \| = C(O(1 + \gamma)^{-n/2}) \tag{21}
\]

holds.
Remark 3  For given $\kappa \geq 0$, we have

$$\gamma = (\kappa d^2 + \alpha)6\beta.$$  \hspace{1cm} (22)

Obviously, the iterative procedure converges quite fast with large $\kappa > 0$, while the iterative procedure might converge very slowly for small $d$.

In the following part, we give the error analysis of $|u - u_{hd}|_{1,\Omega}$ for the DSA with the $P_1$ finite-element discretization in the $Z$-direction and the $P_1$ finite-element discretization in the $X$-$Y$ plane direction. Let $I_d$ denote the domain $[0,1]$ in the $Z$-direction, then $\Omega = \omega \times I_d$. Let $\varphi_j(x, y)$ ($j = 1, 2, \ldots, M$) be the piecewise linear basis function of $X_h$ on $\omega$ and $\phi_i(z)$ ($i = 1, 2, \ldots, N - 1$) is the piecewise linear basis function of $Z_d$; Equation (5) can also be rewritten as follows:

$$I_{hd}u = I_d^1(I_h^2u) = \sum_{i=1}^{i=N-1} \sum_{j=1}^{j=M} u(x_j, y_j, z_i)\varphi_j(x, y)\phi_i(z),$$  \hspace{1cm} (23)

where $u(x_j, y_j, z_i)$ are the nodes on $\Omega$.

For each $u \in C^0(\Omega)$, we define interpolation operator $\Pi u = \sum_{i=1}^{i=N} u(a_i)\Psi_i$, where $a_i$ are the nodes on $\Omega$, and $\Psi_i$ are the corresponding shape functions on $X_h$ or $Z_d$. It is well known that the interpolation operator $\Pi$ satisfies the following properties [10], for any $u \in H^{k+1}(\Omega)$, where $\Omega = \omega$ or $I_d$, $| \cdot |_{1,\Omega}$ denotes the $H^1$-seminorm

$$|u - \Pi u|_{1,\Omega} \leq ch^k |u|_{k+1,\Omega},$$  \hspace{1cm} (24)

$$\|u - \Pi u\|_{0,\Omega} \leq ch^{k+1} |u|_{k+1,\Omega}.$$  \hspace{1cm} (25)

Therefore, applying $k = 1$ for $P_1$ finite element to Equations (24) and (25) on $\omega$ and $I_d$, respectively, we have

$$|u - I_d^1u|_{1,\omega}^2 \leq ch^2 |u|_{2,\omega}^2,$$  \hspace{1cm} (26)

$$|u - I_d^1u|_{1,I_d}^2 \leq cd^2 |u|_{2,I_d}^2.$$  \hspace{1cm} (27)

Integrating both sides of Equations (26) and (27) with respect to $I_d$ and $\omega$, respectively, we obtain

$$|u - I_d^1u|_{1,\Omega}^2 \leq ch^2 |u|_{2,\Omega}^2,$$  \hspace{1cm} (28)

$$|u - I_d^1u|_{1,\Omega}^2 \leq cd^2 |u|_{2,\Omega}^2.$$  \hspace{1cm} (29)

Noting that $|I_h^2u|_{2,\Omega} - |u|_{2,\Omega} \leq |u - I_h^2u|_{2,\Omega} \leq ch |u|_{3,\Omega}$, and

$$|u - I_{hd}u|_{1,\Omega}^2 \leq c|u - I_h^2u|_{1,\Omega}^2 + |I_h^2u - I_d^1(I_h^2u)|_{1,\Omega}^2 \leq ch^2 |u|_{2,\Omega}^2 + cd^2 |I_h^2u|_{2,\Omega}^2 \leq c(h^2 + d^2) |u|_{2,\Omega}^2 + cd^2 h^2 |u|_{3,\Omega}^2,$$  \hspace{1cm} (30)

we have

$$|u - I_{hd}u|_{1,\Omega} \leq c(h + d) |u|_{2,\Omega} + cdh |u|_{3,\Omega}.$$  \hspace{1cm} (31)

Using the similar method above, we also have

$$\|u - I_{hd}u\|_{0,\Omega} \leq c(h^2 + d^2) |u|_{2,\Omega} + cd^2 h |u|_{3,\Omega}.$$  \hspace{1cm} (32)
We consider the auxiliary problem of Equation (35): given \( g \) Taking \( /\Psi_1 \) Galerkin approximation to Equation (35) reads: find \( u_{hd} \)

Equation (1) reads as follows: given \( f \) In the following part, error estimation of \( \parallel g \parallel_{\Omega_1} \) and (38) yield \( h_d \)

Considering \( h_d \) is smaller than \( h + d \) when \( h \) and \( d \) are small enough, therefore, we have

\[
\parallel g \parallel_{\Omega_1} \leq c(h + d)\parallel u \parallel_{\Sigma_2}\.
\]

In the following part, error estimation of \( \parallel g \parallel_{\Omega_0} \) will be given. The weak formulation of Equation (1) reads as follows: given \( f \in L^2(\Omega) \), find

\[
u \in H^1_0(\Omega) : a(u, v) = (f, v) \quad \forall v \in H^1_0(\Omega),
\]

where \( a(u, v) = \int_\Omega ((\partial u/\partial x)(\partial v/\partial x) + (\partial u/\partial y)(\partial v/\partial y) + (\partial u/\partial z)(\partial v/\partial z) + \kappa uv) \, dx \, dy \, dz \). The Galerkin approximation to Equation (35) reads: find

\[
u_{hd} \in X_{hd} : a(u_{hd}, v) = (f, v) \quad \forall v \in X_{hd}.
\]

We consider the auxiliary problem of Equation (35): given \( g \in L^2(\Omega) \), find

\[
\Psi(g) \in H^1_0(\Omega) : a\Psi(g) = (g, v) \quad \forall v \in H^1_0(\Omega).
\]

For the solution of the auxiliary problem, one has regularity assumption as follows:

\[
\parallel \Psi(g) \parallel_{\Omega_2} \leq c\parallel g \parallel_{\Omega_0}.
\]

Using Equation (37), we can write

\[
\parallel g \parallel_{\Omega_0} = \sup_{g \in L^2(\Omega)} \frac{(g, u - u_{hd})}{\parallel g \parallel_{\Omega_0}} = \sup_{g \in L^2(\Omega)} \frac{a(u - u_{hd}, \Psi(g))}{\parallel g \parallel_{\Omega_0}}.
\]

For any arbitrary \( \Psi_h \in X_{hd} \), one has

\[
 a(u - u_{hd}, \Psi(g)) = a(u - u_{hd}, \Psi(g) - \Psi_h),
\]

thus

\[
 (g, u - u_{hd}) \leq c\parallel g \parallel_{\Omega_0} \parallel \Psi(g) - \Psi_h \parallel_{\Omega_1}.
\]

Taking \( \Psi_h = \Pi(\Psi(g)) \) and using Equation (31), from Equation (40) one obtains

\[
 (g, u - u_{hd}) \leq c\parallel g \parallel_{\Omega_2}((h + d)\parallel \Psi(g) \parallel_{\Omega_2} + c \, dh \parallel \Psi(g) \parallel_{\Omega_3}).
\]

Considering \( h_d \) is smaller than \( h + d \) when \( h \) and \( d \) are small enough, and using Equations (39), (41) and (38) yield

\[
\parallel g \parallel_{\Omega_0} \leq c(h^2 + d^2)\parallel u \parallel_{\Sigma_2}.
\]

Finally, we obtain the following error estimations.

**Theorem 2.2**  If \( u \) is the solution to Equation (1), and \( u_{hd} \) is the Galerkin approximate solution to Equation (7), then there exists a constant \( c \), independent of \( h \) and \( d \), such that Equations (34) and (42) hold for \( \Omega = \omega \times I_d \).
2.2 $P_2$ finite-element discretization in the $Z$-direction

From Remark 3, we know that $u_{hd}$ converges very slowly to exact solutions when using the $P_1$ finite-element discretization in the $Z$-direction. Then, we choose the $P_2$ finite-element discretization in the $Z$-direction in order to improve the convergence speed.

2.2.1 Numerical iterative scheme

For the $P_2$ case, we adapt

$$
\phi_i(z) = \begin{cases} 
\frac{(z - d_{i-1})(2z - d_{i-1} - d_i)}{d}, & \text{when } d_{i-1} \leq z \leq d_i, \\
\frac{(d_{i+1} - z)(d_i + d_{i+1} - 2z)}{d}, & \text{when } d_i < z \leq d_{i+1}, \\
0, & \text{otherwise},
\end{cases} \quad (43)
$$

$$
\phi_{i+1/2}(z) = \begin{cases} 
\frac{4(z - d_i)(d_{i+1} - z)}{d^2}, & \text{when } d_i < z \leq d_{i+1}, \\
0, & \text{otherwise},
\end{cases} \quad (44)
$$

to the approximate solution of Equation (4) in the following form:

$$
I_{hd}u = u_{hd} = \sum_{i \in I} u_{i,h} \phi_i + u_{i+1/2,h} \phi_{i+1/2}, \quad \text{with } u_{i,h}, u_{i+1/2,h} \in X_h,
$$

(45)

such that $\forall \phi_i, \phi_{i+1/2} \in Z_d = \{v_d \in C^0([0, 1]) \mid v_{d,k} \in P_2 \forall K \in T_d \} \forall v \in X_h$

$$
\left(\frac{\partial u_{hd}}{\partial z}, \phi_i\right)_x + a((u_{hd}, \phi_i)_z, v) + \kappa((u_{hd}, \phi_i)_z, v)_y = ((f, \phi_i)_z, v)_y. \quad (46)
$$

A series of simple calculations admit

$$
\left(\frac{14}{3d} + \frac{4d}{15} \kappa\right) u_{i,h} + \frac{4d}{15} a(u_{i,h}, v)_x y
$$

$$
= (f, \phi_i)_z + \left(\frac{1}{3d} - \frac{d}{30} \kappa\right) u_{i-1,h} + \left(\frac{1}{3d} - \frac{d}{30} \kappa\right) u_{i+1,h} + \left(\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i+1/2,h}
$$

$$
+ \left(\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i-1/2,h} + v\right)_x y
$$

$$
- a \left(\frac{d}{30} u_{i+1,h} + \frac{d}{15} u_{i+1/2,h} + \frac{d}{15} u_{i-1/2,h} + \frac{d}{15} u_{i-1,h}, v\right)_x y, \quad \phi(z) = \phi_i(z) \forall i \in I
$$

(47)

and

$$
\left(\frac{16}{3d} + \frac{8d}{15} \kappa\right) u_{i+1/2,h} + \frac{8d}{15} a(u_{i+1/2,h}, v)_x y
$$

$$
= (f, \phi_{i+1/2})_z + \left(\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i,h} + \left(\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i+1,h} + v\right)_x y
$$

$$
- a \left(\frac{d}{15} u_{i,h} + \frac{d}{15} u_{i+1,h}, v\right)_x y, \quad \phi(z) = \phi_{i+1/2}(z) \forall i \in I.
$$

(48)
Now the Jaccobi-like iterative schemes for Equations (47) and (48) are
\[
\left(\frac{14}{3d} + \frac{4d}{15} \kappa\right) u_{i,h}^{n+1} + \frac{4d}{15} a(u_{i,h}^{n+1}, v)_{xy}
\]
\[
= ((f, \phi_{i})_{z}, v)_{xy} - \left(\frac{1}{3d} - \frac{d}{30} \kappa\right) u_{i-1,h}^{n} + \left(\frac{1}{3d} - \frac{d}{30} \kappa\right) u_{i+1,h}^{n} + \left(-\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i-1/2,h}^{n}
\]
\[
+ \left(-\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i-1/2,h}^{n} v_{xy}
\]
\[
- a\left(\frac{d}{30} u_{i+1,h}^{n} - \frac{d}{30} u_{i-1,h}^{n} + \frac{d}{15} u_{i+1/2,h}^{n} + \frac{d}{15} u_{i-1/2,h}^{n} v_{xy}\right) \forall i \in I, n = 0, 1, 2, \ldots
\]

and
\[
\left(\frac{16}{3d} + \frac{8d}{15} \kappa\right) u_{i+1/2,h}^{n+1} + \frac{8d}{15} a(u_{i+1/2,h}^{n+1}, v)_{xy}
\]
\[
= ((f, \phi_{i+1/2})_{z}, v)_{xy} - \left(-\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i+1,h}^{n} + \left(-\frac{8}{3d} + \frac{d}{15} \kappa\right) u_{i+1/2}^{n} v_{xy}
\]
\[
- a\left(\frac{d}{15} u_{i+1,h}^{n} + \frac{d}{15} u_{i+1/2,h}^{n} v_{xy}\right) \forall i \in I, n = 0, 1, 2, \ldots
\]

2.2.2 The convergence and error analysis

Let us denote
\[
u_{i,h}^{n+1} = \sum_{i \in \Omega} u_{i,h}
\]
as the \((n + 1)\)th iterative solution. In the following section, we will give that, for arbitrary iterative initial value \(v_{i,h}^{0} \in X_{i,h}, \{u_{i,h}^{n}\}_{n=0}^{\infty}\) is a convergent sequence in both \(L^{2}(\Omega)\) and \(H^{1}(\Omega)\). For simplicity, we denote
\[
\varepsilon_{i,h}^{n+1} = u_{i,h}^{n+1} - u_{i,h}^{n} \forall i \in I.
\]
Taking \(v = \varepsilon_{i,h}^{n+1}\) in Equation (14) and denoting \(E_{0}^{n+1} = \max_{i \in \Omega} \{||\varepsilon_{i,h}^{n+1}||^{2}, ||\varepsilon_{i+1/2,h}^{n+1}||^{2}\}, E_{1}^{n+1} = \max_{i \in \Omega} \{||\nabla^{2}e_{i,h}^{n+1}||^{2}, ||\nabla^{2}e_{i+1/2,h}^{n+1}||^{2}\}, E_{i}^{n+1} = \max_{i \in \Omega} \{||\nabla^{2}e_{i,h}^{n+1}||^{2}, ||\nabla^{2}e_{i+1/2,h}^{n+1}||^{2}\}, E_{i}^{n+1} = \max_{i \in \Omega} \{||\nabla^{2}e_{i,h}^{n+1}||^{2}, ||\nabla^{2}e_{i+1/2,h}^{n+1}||^{2}\},\) from Equations (49) and (50), we have
\[
\left(\frac{14}{3d} + \frac{4d}{15} \kappa\right) \varepsilon_{i,h}^{n+1}^{2} + \frac{4d}{15} ||\nabla^{2}e_{i,h}^{n+1}||^{2} \leq 2 \left|\frac{1}{3d} - \frac{d}{30} \kappa\right| \sqrt{E_{0}^{n}} ||\varepsilon_{i,h}^{n+1}||^{2}
\]

\[
+ 2 \left|\frac{8}{3d} + \frac{d}{15} \kappa\right| \sqrt{E_{1}^{n}} ||\varepsilon_{i,h}^{n+1}|| + 3d \frac{15}{15} \sqrt{E_{1}^{n}} ||\nabla^{2}e_{i,h}^{n+1}|| \forall i \in I, n = 0, 1, 2, \ldots
\]

and
\[
\left(\frac{16}{3d} + \frac{8d}{15} \kappa\right) \varepsilon_{i+1/2,h}^{n+1}^{2} + \frac{8d}{15} ||\nabla^{2}e_{i+1/2,h}^{n+1}||^{2} \leq 2 \left|\frac{8}{3d} + \frac{d}{15} \kappa\right| \sqrt{E_{0}^{n}} ||\varepsilon_{i+1/2,h}^{n+1}||^{2}
\]

\[
+ 2d \frac{15}{15} \sqrt{E_{1}^{n}} ||\nabla^{2}e_{i+1/2,h}^{n+1}|| \forall i \in I, n = 0, 1, 2, \ldots
\]
Supposing $1/3d - \kappa d/30 \geq 0$ and $8/3d - \kappa d/15 \geq 0$ for convenience, i.e. $\kappa d^2 \leq 10$, and adding Equations (53) and (54), we have
\[
\alpha^{n+1} E_0^{n+1} + b^{n+1} E_1^{n+1} \leq \alpha^n E_0^n + b^n E_1^n,
\]
where $\alpha^{n+1} = 13/3d + 29d/30, b^{n+1} = 19d/30, \alpha^n = 17/3d - 5d/30, b^n = 5d/30$. Let $\alpha \leq 0$, from Equation (55), one has
\[
(a^{n+1} + \alpha) \left( E_0^{n+1} + \frac{b^n}{\alpha} E_1^{n+1} \right) \leq a^n \left( E_0^n + \frac{b^n}{\alpha} E_1^n \right). \tag{56}
\]
If
\[
\frac{(a^{n+1} + \alpha) b^n}{a^n} \leq b^n + 1, \quad \frac{a^n}{(a^{n+1} + \alpha)} < 1, \tag{57}
\]
then $(E_0^{n+1} + (b^n/\alpha) E_1^{n+1})$ is a contractive sequence. Supposing $\kappa d^2 > 20/17$, Equation (57) will hold due to $b^n < b^{n+1}$. Finally, we have the following theorem like Theorem 2.1.

**Theorem 2.3** The sequence $\{u_{hd}^n\}_{n=0}^\infty$, with the arbitrary iterative initial value $u_{hd}^0 \in X_{hd}$, is a convergent sequence in both $L^2(\Omega)$ space and $H_0^1(\Omega)$ space
\[
\|u_{hd}^n - u_{hd}\| + \|\nabla (u_{hd}^n - u_{hd})\| = O(1 + \gamma)^{-n/2}.
\]

We can also obtain the following error estimations of the DSA of $P_2$ finite-element discretization in the $Z$-direction by means of similar methods, which deduce inequalities (34) and (42)
\[
|u - u_{hd}|_{1,\Omega} \leq c(h + d^2)|u|_{1,\Omega}, \tag{59}
\]
\[
\|u - u_{hd}\|_{0,\Omega} \leq c(h^2 + d^3)|u|_{2,\Omega}. \tag{60}
\]

**Theorem 2.4** If $u$ is the solution to Equation (1) and $u_{hd}$ is the Galerkin approximate solution to Equation (46), then there exists a constant $c$, independent of $h$ and $d$, such that Equations (59) and (60) hold for $\Omega = \omega \times I_d$.

### 3. Numerical experiments

In this section, we will test the viability of DSA that has been proposed above. The numerical examples are broadly divided into two parts. The first part shows the numerical results and the rate of convergence of the method with different mesh scales. We compare the DSA with the 3D Galerkin FEM solving directly the 3D elliptic equation by the piecewise linear finite element ($P_1$ element) with 4-node tetrahedron. In the second part, we implement DSA in parallel with several processors and illustrate the parallel efficiency of DSA, while we only use one processor for 3D FEM. All algorithms are implemented using finite-element software FREEFEM++ [2].

We consider a cube of unit volume with domain $X_{hd} = X_h \times Z_d = [0, 1]^2 \times [0, 1]$. The exact solution of Equation (1) is
\[
u(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z), \tag{61}
\]
\[
f(x, y, z) = (3\pi^2 + \kappa) \sin(\pi x) \sin(\pi y) \sin(\pi z).
\]
In our test cases, we assume that the condition of reaching convergence in iterations is $\|u_{hd}^{n+1} - u_{hd}^n\|_{0,\Omega} < \text{Tol}$, where Tol = $10^{-6}$, with the initial value of iteration $u_{h, 0}^0 = 0$. 


3.1 Rate of convergence and parallel efficiency for $P_{1}$-FEM in the Z-direction

In this part, we take only one processor to solve Equation (10) by the DSA and the 3D FEM, respectively, and let $\kappa = 0, 100, h = d$. The $L^2$-norm and $H^1$-seminorm relative errors, numerical convergence rate, iterative steps and iterative times for DSA and 3D FEM, respectively, are presented in Tables 1–4. Denoting $L_{DSA}$ as $L^2$-norm relative errors for DSA, $L_{3DFEM}$ as $L^2$-norm relative errors for 3D FEM, $H_{DSA}$ as $H^1$-norm relative errors for DSA, $H_{3DFEM}$ as $H^1$-norm relative errors for 3D FEM, comparing Table 1 with Table 3, $L_{DSA}/L_{3DFEM} \simeq 0.60$, $H_{DSA}/H_{3DFEM} \simeq 0.68$ for $\kappa = 0$ in the same mesh scale and $L_{DSA}/L_{3DFEM} \simeq 0.65$, $H_{DSA}/H_{3DFEM} \simeq 0.67$ for $\kappa = 100$ in the same mesh scale. It shows that we can obtain more better approximation solutions by DSA than by the 3D FEM. Numerical convergence rate of $L^2$-norm and $H^1$-seminorm for DSA is almost up to the theoretical convergence rates: second order for $L^2$-norm and first order for $H^1$-seminorm, which are the same as the 3D FEM.

When taking $h = d = 1/17$ and different $\kappa(0, 0.01, 0.1, 1, 10, 100)$, iterative steps and iterative times for DSA are shown in Table 5. Numerical results show that more large $\kappa$ contributes to fast convergence speed, which is conformed by theoretical analysis in Remark 3. However, it takes more CPU times to reach convergence solutions for DSA than for 3D FEM, which is because that we need to solve a series of 2D elliptic equations for DSA, while we deal with only one 3D elliptic equation for 3D FEM.

In the following part, we introduce the parallel technology to solve Equation (10). ∀ $i$, $u_{i, h}^{n+1}$ is only related to $u_{i-1, h}^{n}$ and $u_{i+1, h}^{n}$ in Equation (10), therefore $i \in I$ may be assigned fairly to different processors to finish the computation and exchange the data among the different processors. The following allocation scheme for $I$ is adopted in our numerical cases:

$$1 + j \cdot P \leq i \leq (j + 1) \cdot P,$$

### Table 1. DSA for $P_{1}$-FEM $\kappa = 0$. 

| $h = d$ | $\frac{|u-u_{h,d}|_{L^2(\Omega)}}{|u|_{L^2(\Omega)}}$ | Order | $\frac{|u-u_{h,d}|_{H^1(\Omega)}}{|u|_{H^1(\Omega)}}$ | Order | Steps | CPU(s) |
|---|---|---|---|---|---|---|
| 1/9 | 0.0332892 | – | 0.154158 | – | 33 | 8.02 |
| 1/17 | 0.0094984 | 1.9719 | 0.0815261 | 1.0017 | 104 | 83.59 |
| 1/25 | 0.0044532 | 1.9641 | 0.0554254 | 1.0006 | 206 | 422.39 |
| 1/33 | 0.0026276 | 1.9903 | 0.0419861 | 1.0001 | 338 | 1357.71 |

### Table 2. DSA for $P_{1}$-FEM $\kappa = 100$. 

| $h = d$ | $\frac{|u-u_{h,d}|_{L^2(\Omega)}}{|u|_{L^2(\Omega)}}$ | Order | $\frac{|u-u_{h,d}|_{H^1(\Omega)}}{|u|_{H^1(\Omega)}}$ | Order | Steps | CPU(s) |
|---|---|---|---|---|---|---|
| 1/9 | 0.0155579 | – | 0.154086 | – | 11 | 6.76 |
| 1/17 | 0.0043831 | 1.9919 | 0.0815261 | 1.0012 | 30 | 54.15 |
| 1/25 | 0.0020379 | 1.9858 | 0.0554197 | 1.0004 | 57 | 207.98 |
| 1/33 | 0.0011832 | 1.9583 | 0.0549821 | 1.0002 | 93 | 605.76 |

### Table 3. 3D FEM $\kappa = 0$. 

| $h = d$ | $\frac{|u-u_{h,d}|_{L^2(\Omega)}}{|u|_{L^2(\Omega)}}$ | Order | $\frac{|u-u_{h,d}|_{H^1(\Omega)}}{|u|_{H^1(\Omega)}}$ | Order | CPU(s) |
|---|---|---|---|---|---|
| 1/9 | 0.0553402 | – | 0.224925 | – | 0.4 |
| 1/17 | 0.0158985 | 1.9612 | 0.119226 | 0.9980 | 2.89 |
| 1/25 | 0.0020379 | 1.9858 | 0.0554197 | 1.0004 | 57 | 207.98 |
| 1/33 | 0.0011832 | 1.9583 | 0.0549821 | 1.0002 | 93 | 605.76 |
Table 4. 3D FEM $\kappa = 100$.

| $h=d$ | $\frac{|u-u_h|_{L^2(\Omega)}}{|u|_{L^2(\Omega)}}$ | Order | $\frac{|u-u_h|_{1,\Omega}}{|u|_{1,\Omega}}$ | Order | CPU(s) |
|-------|---------------------------------|-------|---------------------------------|-------|--------|
| 1/9   | 0.0245318                       | –     | 0.234564                        | –     | 0.4    |
| 1/17  | 0.00670578                      | 2.0389| 0.120699                        | 1.0447| 2.63   |
| 1/25  | 0.00308386                      | 2.0149| 0.0815619                       | 1.0163| 8.42   |
| 1/33  | 0.00176607                      | 2.0078| 0.0616457                       | 1.0084| 19.61  |

Table 5. Different $\kappa$ for DSA.

<table>
<thead>
<tr>
<th>$h=d = 1/17$</th>
<th>$\kappa$</th>
<th>Steps</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\kappa$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>104</td>
<td>91.58</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>104</td>
<td>91.63</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>103</td>
<td>90.99</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>101</td>
<td>89.36</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>80</td>
<td>81.33</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>61.06</td>
<td></td>
</tr>
</tbody>
</table>

where $j$ stands for the $j$th ($j = 0, 1, \ldots$) processor, $P = (N - 1)/np$, $np$ for the number of processor. Tables 6–9 for $\kappa = 0$ and Tables 11–14 for $\kappa = 100$ show the numerical results when applying $np = 1, 2, 4, 8, 16$ processors with different mesh scales $h=d = 1/9, 1/17, 1/25, 1/33$. Parallel solving method can also be up to the theoretical convergence rates. The $\kappa = 0$ case in Table 10 and the $\kappa = 100$ case in Table 15 show the CPU times and speed-up ratio with different mesh scales $h = d = 1/9, 1/17, 1/25, 1/33$ for $np = 1, 2, 4, 8, 16$, respectively. The average speed-up ratio for $np = 1, 2, 4, 8, 16$ and good acceleration effects are obtained for DSA.

Table 6. DSA for two processors, $\kappa = 0$.

| $h=d$ | $\frac{|u-u_h|_{L^2(\Omega)}}{|u|_{L^2(\Omega)}}$ | Order | $\frac{|u-u_h|_{1,\Omega}}{|u|_{1,\Omega}}$ | Order | Steps | CPU(s) |
|-------|---------------------------------|-------|---------------------------------|-------|-------|--------|
| 1/9   | 0.0332895                       | –     | 0.154158                        | –     | 41    | 4.54   |
| 1/17  | 0.0095008                      | 1.9715| 0.081526                        | 1.0017| 116   | 47.47  |
| 1/25  | 0.0044538                      | 1.9645| 0.055425                        | 1.0006| 223   | 232.54 |
| 1/33  | 0.0026281                      | 1.9000| 0.041986                        | 1.0002| 359   | 754.78 |

Table 7. DSA for four processors, $\kappa = 0$.

| $h=d$ | $\frac{|u-u_h|_{L^2(\Omega)}}{|u|_{L^2(\Omega)}}$ | Order | $\frac{|u-u_h|_{1,\Omega}}{|u|_{1,\Omega}}$ | Order | Steps | CPU(s) |
|-------|---------------------------------|-------|---------------------------------|-------|-------|--------|
| 1/9   | 0.0332895                       | –     | 0.154158                        | –     | 49    | 2.71   |
| 1/17  | 0.0095005                      | 1.9716| 0.081526                        | 1.0017| 129   | 27.09  |
| 1/25  | 0.0044544                      | 1.9640| 0.055425                        | 1.0006| 240   | 127.85 |
| 1/33  | 0.0026286                      | 1.8998| 0.041986                        | 1.0002| 380   | 414.87 |

Table 8. DSA for eight processors, $\kappa = 0$.

| $h=d$ | $\frac{|u-u_h|_{L^2(\Omega)}}{|u|_{L^2(\Omega)}}$ | Order | $\frac{|u-u_h|_{1,\Omega}}{|u|_{1,\Omega}}$ | Order | Steps | CPU(s) |
|-------|---------------------------------|-------|---------------------------------|-------|-------|--------|
| 1/9   | 0.0332918                       | –     | 0.154158                        | –     | 63    | 1.58   |
| 1/17  | 0.0095032                      | 1.9712| 0.081526                        | 1.0017| 153   | 16.5   |
| 1/25  | 0.0044567                      | 1.9634| 0.055425                        | 1.0006| 273   | 77.97  |
| 1/33  | 0.0026334                      | 1.8950| 0.041986                        | 1.0002| 420   | 245.75 |
Table 9. DSA for 16 processors, $\kappa = 0$.

<table>
<thead>
<tr>
<th>$h = d$</th>
<th>$\frac{|u - u_h|<em>{0, \Omega}}{|u|</em>{0, \Omega}}$</th>
<th>Order</th>
<th>$\frac{|u - u_h|<em>{1, \Omega}}{|d|</em>{1, \Omega}}$</th>
<th>Order</th>
<th>Steps</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/17</td>
<td>0.00941915</td>
<td>–</td>
<td>0.0839469</td>
<td>–</td>
<td>201</td>
<td>12.67</td>
</tr>
<tr>
<td>1/33</td>
<td>0.00261074</td>
<td>1.9345</td>
<td>0.0430116</td>
<td>1.0082</td>
<td>501</td>
<td>165.27</td>
</tr>
</tbody>
</table>

Table 10. DSA for two processors, $\kappa = 100$.

<table>
<thead>
<tr>
<th>$h = d$</th>
<th>$\frac{|u - u_h|<em>{0, \Omega}}{|u|</em>{0, \Omega}}$</th>
<th>Order</th>
<th>$\frac{|u - u_h|<em>{1, \Omega}}{|d|</em>{1, \Omega}}$</th>
<th>Order</th>
<th>Steps</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>0.0155583</td>
<td>–</td>
<td>0.154086</td>
<td>–</td>
<td>13</td>
<td>3.93</td>
</tr>
<tr>
<td>1/17</td>
<td>0.0043847</td>
<td>1.9913</td>
<td>0.081513</td>
<td>1.0012</td>
<td>33</td>
<td>30.9</td>
</tr>
<tr>
<td>1/25</td>
<td>0.0020382</td>
<td>1.9863</td>
<td>0.055419</td>
<td>1.0005</td>
<td>62</td>
<td>118.23</td>
</tr>
<tr>
<td>1/33</td>
<td>0.0011837</td>
<td>1.9573</td>
<td>0.041982</td>
<td>1.0002</td>
<td>99</td>
<td>324.04</td>
</tr>
</tbody>
</table>

Table 11. DSA for four processors, $\kappa = 100$.

<table>
<thead>
<tr>
<th>$h = d$</th>
<th>$\frac{|u - u_h|<em>{0, \Omega}}{|u|</em>{0, \Omega}}$</th>
<th>Order</th>
<th>$\frac{|u - u_h|<em>{1, \Omega}}{|d|</em>{1, \Omega}}$</th>
<th>Order</th>
<th>Steps</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>0.0155582</td>
<td>–</td>
<td>0.154086</td>
<td>–</td>
<td>15</td>
<td>2.03</td>
</tr>
<tr>
<td>1/17</td>
<td>0.0043850</td>
<td>1.9912</td>
<td>0.081513</td>
<td>1.0012</td>
<td>36</td>
<td>15.49</td>
</tr>
<tr>
<td>1/25</td>
<td>0.0020391</td>
<td>1.9854</td>
<td>0.055419</td>
<td>1.0005</td>
<td>66</td>
<td>60.74</td>
</tr>
<tr>
<td>1/33</td>
<td>0.0011849</td>
<td>1.9479</td>
<td>0.041982</td>
<td>1.0002</td>
<td>104</td>
<td>171.89</td>
</tr>
</tbody>
</table>

Table 12. DSA for eight processors, $\kappa = 100$.

<table>
<thead>
<tr>
<th>$h = d$</th>
<th>$\frac{|u - u_h|<em>{0, \Omega}}{|u|</em>{0, \Omega}}$</th>
<th>Order</th>
<th>$\frac{|u - u_h|<em>{1, \Omega}}{|d|</em>{1, \Omega}}$</th>
<th>Order</th>
<th>Steps</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>0.0155582</td>
<td>–</td>
<td>0.154086</td>
<td>–</td>
<td>19</td>
<td>1.11</td>
</tr>
<tr>
<td>1/17</td>
<td>0.0043856</td>
<td>1.9910</td>
<td>0.081513</td>
<td>1.0012</td>
<td>42</td>
<td>8.91</td>
</tr>
<tr>
<td>1/25</td>
<td>0.0020387</td>
<td>1.9862</td>
<td>0.055419</td>
<td>1.0005</td>
<td>75</td>
<td>35.35</td>
</tr>
<tr>
<td>1/33</td>
<td>0.0011871</td>
<td>1.9479</td>
<td>0.041982</td>
<td>1.0002</td>
<td>114</td>
<td>96.79</td>
</tr>
</tbody>
</table>

Table 13. DSA for 16 processors, $\kappa = 100$.

<table>
<thead>
<tr>
<th>$h = d$</th>
<th>$\frac{|u - u_h|<em>{0, \Omega}}{|u|</em>{0, \Omega}}$</th>
<th>Order</th>
<th>$\frac{|u - u_h|<em>{1, \Omega}}{|d|</em>{1, \Omega}}$</th>
<th>Order</th>
<th>Steps</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/17</td>
<td>0.00434554</td>
<td>–</td>
<td>0.0839479</td>
<td>–</td>
<td>55</td>
<td>5.49</td>
</tr>
<tr>
<td>1/33</td>
<td>0.00117655</td>
<td>1.9698</td>
<td>0.0430091</td>
<td>1.0083</td>
<td>135</td>
<td>60.30</td>
</tr>
</tbody>
</table>

Table 14. Speed-up ratio for DSA, $\kappa = 0$.

<table>
<thead>
<tr>
<th>$h = d$</th>
<th>$np = 1$</th>
<th>$np = 2$</th>
<th>$np = 4$</th>
<th>$np = 8$</th>
<th>$np = 16$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>8.02</td>
<td>4.54(1.77)</td>
<td>2.71(2.96)</td>
<td>1.58(5.08)</td>
<td>–</td>
</tr>
<tr>
<td>1/17</td>
<td>83.59</td>
<td>47.47(1.76)</td>
<td>27.09(3.09)</td>
<td>16.5(5.07)</td>
<td>12.67(6.59)</td>
</tr>
<tr>
<td>1/25</td>
<td>422.39</td>
<td>232.54(1.82)</td>
<td>127.85(3.30)</td>
<td>77.47(5.42)</td>
<td>–</td>
</tr>
<tr>
<td>1/33</td>
<td>1357.71</td>
<td>754.78(1.80)</td>
<td>414.87(3.26)</td>
<td>245.75(5.23)</td>
<td>165.27(8.22)</td>
</tr>
<tr>
<td>Average</td>
<td>1.79</td>
<td>3.15</td>
<td>5.2</td>
<td>7.41</td>
<td></td>
</tr>
</tbody>
</table>

3.2 Rate of convergence and parallel efficiency for $P_2$-FEM in the Z-direction

In this part, we show the numerical results when applying the $P_2$ finite-element discretization in the Z-direction and taking the larger mesh scale in the Z-direction than the X–Y plane direction. According to Equations (59) and (60), the convergence rate of $H^1$-seminorm error should be first
Table 15. Speed-up ratio for DSA, $\kappa = 100$.

<table>
<thead>
<tr>
<th>$h = d$</th>
<th>$np = 1$</th>
<th>$np = 2$</th>
<th>$np = 4$</th>
<th>$np = 8$</th>
<th>$np = 16$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/9$</td>
<td>6.76</td>
<td>3.93(1.72)</td>
<td>2.03(3.33)</td>
<td>1.11(6.09)</td>
<td>–</td>
</tr>
<tr>
<td>$1/17$</td>
<td>54.16</td>
<td>30.9(1.75)</td>
<td>15.49(3.50)</td>
<td>8.91(6.08)</td>
<td>5.49(9.87)</td>
</tr>
<tr>
<td>$1/25$</td>
<td>207.98</td>
<td>118.23(1.76)</td>
<td>60.74(3.42)</td>
<td>35.35(5.88)</td>
<td>–</td>
</tr>
<tr>
<td>$1/33$</td>
<td>605.76</td>
<td>324.04(1.87)</td>
<td>171.89(3.24)</td>
<td>96.79(6.26)</td>
<td>60.30(10.05)</td>
</tr>
<tr>
<td>Average</td>
<td>1.78</td>
<td>3.75</td>
<td>6.07</td>
<td>9.96</td>
<td></td>
</tr>
</tbody>
</table>

Table 16. $H^1$-seminorm relative error of DSA using $P_2$-FEM.

<table>
<thead>
<tr>
<th>$d = h^{1/2}$</th>
<th>$\kappa = 0$</th>
<th></th>
<th>$\kappa = 100$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\frac{|u - u_h|_1}{|u|_1}$</td>
<td>Order</td>
<td>Steps</td>
<td>$\frac{|u - u_h|_1}{|u|_1}$</td>
</tr>
<tr>
<td>$h = 1/9$</td>
<td>0.143472</td>
<td>–</td>
<td>20</td>
<td>0.150654</td>
</tr>
<tr>
<td>$h = 1/16$</td>
<td>0.0804318</td>
<td>1.0059</td>
<td>34</td>
<td>0.0833069</td>
</tr>
<tr>
<td>$h = 1/25$</td>
<td>0.0514072</td>
<td>1.0030</td>
<td>50</td>
<td>0.0527253</td>
</tr>
<tr>
<td>$h = 1/33$</td>
<td>0.0356755</td>
<td>1.0018</td>
<td>69</td>
<td>0.0363513</td>
</tr>
<tr>
<td>$h = 1/49$</td>
<td>0.0262004</td>
<td>1.0013</td>
<td>91</td>
<td>0.0265789</td>
</tr>
</tbody>
</table>

Table 17. $L^2$-norm relative error of DSA using $P_2$-FEM.

<table>
<thead>
<tr>
<th>$d = h^{2/3}$</th>
<th>$\kappa = 0$</th>
<th></th>
<th>$\kappa = 100$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\frac{|u - u_h|<em>{L^2}}{|u|</em>{L^2}}$</td>
<td>Order</td>
<td>Steps</td>
<td>$\frac{|u - u_h|<em>{L^2}}{|u|</em>{L^2}}$</td>
</tr>
<tr>
<td>$h = 1/8$</td>
<td>0.0259306</td>
<td>–</td>
<td>33</td>
<td>0.00845083</td>
</tr>
<tr>
<td>$h = 1/27$</td>
<td>0.002342</td>
<td>1.9767</td>
<td>142</td>
<td>0.000666311</td>
</tr>
<tr>
<td>$h = 1/64$</td>
<td>0.0005159</td>
<td>1.7529</td>
<td>396</td>
<td>0.000134568</td>
</tr>
</tbody>
</table>

order if $d = h^{1/2}$; the convergence rate of $L^2$-norm error should be second order if $d = h^{2/3}$. Tables 16 and 17 show the $L^2$-norm and $H^1$-seminorm relative errors, convergence rate and iterative steps with different mesh scales $h, d$ for $\kappa = 0, 100$, respectively. It is easy to find that the numerical results are in accordance with the theoretical previous results.

In order to show the efficiency of DSA using the $P_2$-FEM in the $Z$-direction, we choose the proper mesh scale between $h$ and $d$ so that the DSA using $P_2$-FEM in the $Z$-direction and 3D FEM reaches the almost same $L^2$-norm relative error and $H^1$-seminorm relative error. Tables 18 and 19 show the CPU times for $\kappa = 0$ and $\kappa = 100$, respectively, when the DSA and 3D FEM reaches the same $L^2$-norm relative error and $H^1$-seminorm relative error with a series of difficult mesh scales. Moreover, the CPU times using several processors for DSA using $P_2$-FEM are also given in Tables 18 and 19. In Table 18, for every group of mesh scale, such as $h = 1/21, d = 1/5$ for DSA and $h = d = 1/33$ for DSA, we give the CPU times using several processors for DSA. DSA takes more CPU times using one processor than the 3D FEM, but if DSA is implemented in parallel with several processors, it can save more CPU times. The DSA’s storage of computations is only 0.05 times averagely of the 3D FEM at the same time, which is also one of the advantages of the DSA. From Tables 18 and 19, it is obvious that DSA for $\kappa = 100$ takes less CPU times using several processor than $\kappa = 0$. This is because that DSA converges fast for large $\kappa$ due to previous theoretical analysis. We should note that the DSA with smaller mesh scales has more advantages than the 3D FEM.
Table 18. DSA using $P_2$-FEM and 3D FEM, $\kappa=0$.

<table>
<thead>
<tr>
<th>Steps</th>
<th>CPU(s)</th>
<th>$|u-u_h|_{0,\Omega}$</th>
<th>$|u-u_h|_{1,\Omega}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/21(1/5)1 cpu</td>
<td>0.0043508</td>
<td>0.061175</td>
<td>50</td>
</tr>
<tr>
<td>1/21(1/5)2 cpu</td>
<td>0.0043526</td>
<td>0.061176</td>
<td>59</td>
</tr>
<tr>
<td>1/21(1/5)4 cpu</td>
<td>0.0043551</td>
<td>0.061175</td>
<td>69</td>
</tr>
<tr>
<td>1/53</td>
<td>0.0042498</td>
<td>0.061443</td>
<td>22.4</td>
</tr>
<tr>
<td>1/33(1/7)1 cpu</td>
<td>0.0017235</td>
<td>0.038888</td>
<td>91</td>
</tr>
<tr>
<td>1/33(1/7)4 cpu</td>
<td>0.0017303</td>
<td>0.038535</td>
<td>127</td>
</tr>
<tr>
<td>1/33(1/7)6 cpu</td>
<td>0.0017462</td>
<td>0.038535</td>
<td>127</td>
</tr>
<tr>
<td>1/73</td>
<td>0.00016503</td>
<td>0.03826</td>
<td>95.96</td>
</tr>
<tr>
<td>1/60(1/11)1 cpu</td>
<td>0.00054678</td>
<td>0.021379</td>
<td>203</td>
</tr>
<tr>
<td>1/60(1/11)5 cpu</td>
<td>0.00055686</td>
<td>0.021379</td>
<td>245</td>
</tr>
<tr>
<td>1/60(1/11)10 cpu</td>
<td>0.00055927</td>
<td>0.028217</td>
<td>200</td>
</tr>
<tr>
<td>1/93</td>
<td>0.00053638</td>
<td>0.027776</td>
<td>548.02</td>
</tr>
</tbody>
</table>

Table 19. DSA using $P_2$-FEM and 3D FEM, $\kappa=100$.

<table>
<thead>
<tr>
<th>Steps</th>
<th>CPU(s)</th>
<th>$|u-u_h|_{0,\Omega}$</th>
<th>$|u-u_h|_{1,\Omega}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/21(1/6)1 cpu</td>
<td>0.00168239</td>
<td>0.0615853</td>
<td>21</td>
</tr>
<tr>
<td>1/21(1/6)3 cpu</td>
<td>0.00168249</td>
<td>0.0615852</td>
<td>26</td>
</tr>
<tr>
<td>1/21(1/6)5 cpu</td>
<td>0.00168283</td>
<td>0.0615828</td>
<td>29</td>
</tr>
<tr>
<td>1/73</td>
<td>0.001676612</td>
<td>0.0616461</td>
<td>19.31</td>
</tr>
<tr>
<td>1/34(1/8)1 cpu</td>
<td>0.00655153</td>
<td>0.037872</td>
<td>33</td>
</tr>
<tr>
<td>1/34(1/8)3 cpu</td>
<td>0.00655227</td>
<td>0.037872</td>
<td>39</td>
</tr>
<tr>
<td>1/34(1/8)6 cpu</td>
<td>0.0067547</td>
<td>0.037338</td>
<td>45</td>
</tr>
<tr>
<td>1/53</td>
<td>0.00683444</td>
<td>0.038309</td>
<td>93.75</td>
</tr>
<tr>
<td>1/46(1/10)1 cpu</td>
<td>0.00333313</td>
<td>0.027937</td>
<td>48</td>
</tr>
<tr>
<td>1/46(1/10)3 cpu</td>
<td>0.00333364</td>
<td>0.027937</td>
<td>55</td>
</tr>
<tr>
<td>1/46(1/10)6 cpu</td>
<td>0.0034779</td>
<td>0.027436</td>
<td>61</td>
</tr>
<tr>
<td>1/73</td>
<td>0.0036011</td>
<td>0.027797</td>
<td>219.52</td>
</tr>
<tr>
<td>1/59(1/12)1 cpu</td>
<td>0.0019688</td>
<td>0.021763</td>
<td>66</td>
</tr>
<tr>
<td>1/59(1/12)3 cpu</td>
<td>0.0019826</td>
<td>0.021763</td>
<td>73</td>
</tr>
<tr>
<td>1/59(1/12)6 cpu</td>
<td>0.0020680</td>
<td>0.021362</td>
<td>79</td>
</tr>
<tr>
<td>1/93</td>
<td>0.0022201</td>
<td>0.021814</td>
<td>461.92</td>
</tr>
</tbody>
</table>

4. Conclusion

In this paper, the DSA for the 3D elliptic equation in a cubic domain is proposed. The greatest strength of DSA is that a series of 2D elliptic equations can be easily implemented in parallel with different processors. At the same time, DSA has fast convergence speed using several processors with parallel technology and save a great number of storage of computations than the 3D standard finite-element method. Finally, some numerical examples are given to illustrate the effectiveness of our method.

Further more, some other approximate methods such as the domain decomposition method, the variational multiscale method, can be applied to 2D elliptic equations in the $X–Y$ plane direction.
In addition, parallel technology can be used to deal with computations in the $X$–$Y$ plane direction in order to improve the convergence speed.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (11171269, 10871156) and Jiaoda Foundation (2009jxjuc30). Especially, we thank the anonymous referees very much for their helpful comments and suggestions, which led to substantial improvements of the presentation.

References