NUMERICAL MATHEMATICS
A Journal of Chinese Universities (English Series)

Vol. **16** No. 1, pp. 14-27
February 2007

Incremental Unknowns Method for Solving Three-Dimensional Convection-Diffusion Equations†

Lunji Song* and Yujiang Wu

School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China.

Received June 2005; Accepted (in revised version) January 19, 2006

Abstract. We use the incremental unknowns method in conjunction with the iterative methods to approximate the solution of the nonsymmetric and positive-definite linear systems generated from a multilevel discretization of three-dimensional convection-diffusion equations. The condition numbers of incremental unknowns matrices associated with the convection-diffusion equations and the number of iterations needed to attain an acceptable accuracy are estimated. Numerical results are presented with two-level approximations, which demonstrate that the incremental unknowns method when combined with some iterative methods is very efficient.

Key words: Incremental unknowns; finite difference; convection-diffusion equation.

AMS subject classifications: 65M06, 65M60

1 Introduction

The incremental unknowns method was first introduced by Temam [1] to study long time integration of dissipative evolutionary equations when finite difference approximations are used. Incremental unknowns of different types have been proposed as a means to develop linear elliptic problem and nonlinear evolutionary equations. Garcia [2] studied the algebraic framework appropriate to one and two dimensional linear partial differential equations when several levels of discretization are considered. The hierarchical ordering of the nodal unknowns lead to a linear system $(A)(x) = (b)$, which can be written, with the use of the incremental unknowns ordered in the same way, as the equivalent system $[A][x] = [b]$, where $[A] = S^T(A)S$ and $[b] = S^T(b)$. Here, S stands for the transfer matrix from the incremental unknowns $[x]$ to the nodal unknowns (x), i.e., $(x) = S[x]$. Hereafter, we always solve the linear system $[A][x] = [b]$, instead of the linear system $(A)(x) = (b)$, with the use of the following iterative methods: the Conjugate Gradient method when $[A]$ is a symmetric and positive-definite matrix, the iterative methods such as MR,

*Correspondence to: Lunji Song, School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China. Email: song@lzu.edu.cn
†This work was supported by the Foundation of Gansu Natural Science (3ZS041-A25-011).

GCR, Orthomin(k) [7], HSS and PSS [13, 14] when $[A]$ is a real nonsymmetric and positive-definite matrix, and the Bi-CGSTAB method (see [5]) when $[A]$ is a nonsingular matrix. By specializing the PSS method to block triangular (or triangular) and skew-Hermitian splittings (BTSS or TSS), the PSS method naturally leads to a BTSS or TSS iteration method, so the BTSS (or TSS) method is a special case of the PSS method.

The article is organized as follows. In Section 2, we introduce the convection-diffusion equations. In Section 3, we report six iterative methods, i.e., MR, GCR, Orthomin(k), Bi-CGSTAB, HSS, BTSS, for computing a nonsymmetric linear system. In Section 4, using the construction of transfer matrix S (see [11]) and the utiliztion of space decomposition, we analyze some properties of incremental unknowns and the transfer matrix. In Section 5, we apply the incremental unknowns method to solve the convection-diffusion equations and estimate condition numbers of the incremental unknown matrices. Moreover, it will be demonstrated that at most k iterations are needed to obtain an acceptable solution with MR, GCR, Orthomin(k). Numerical results with two-level approximations are presented and analyzed in Section 6.

2 Convection-diffusion equations

We consider the three-dimensional convection-diffusion equations

$$\begin{cases} Lu = -\nabla \cdot (\nabla u + bu) = f, & in\ \Omega, \\ u(x) = 0, & on\ \partial\Omega, \end{cases}$$

where $b > 0,\ x = (x_1, x_2, x_3) \in \mathbb{R}^3$. When the central finite difference is used for the spatial multi-level discretization, we have the problem of approximating the solutions of large sparse systems of linear equations

$$AU = g, \tag{1}$$

where the vectors $U, g \in \mathbb{R}^N$, $A \in \mathbb{R}^{N\times N}$ is a nonsymmetric and positive-definite matrix of order N. The dimension N is the number of nodal points and U is the vector corresponding to the nodal values of the unknown function. In order to approximate the solution of (1), several iterative methods, such as Minimum Residual (MR), Generalized Conjugate Residual (GCR), Orthomin(k), Bi-CGSTAB methods, Hermitian and Skew-hermitian Splitting method (HSS) and Block Triangular and Skew-hermitian Splitting method (BTSS), can be considered.

We introduce incremental unknowns (IU) method to a linear system (see, [2, 3]). Let $\overline{U} \in \mathbb{R}^N$ be the $(l+1)$-level IU vector $(l > 0)$ of the form

$$\overline{U} = \begin{pmatrix} Y_0 \\ Z^l \end{pmatrix}, \quad Z^l = \begin{pmatrix} Z_1 \\ \vdots \\ Z_l \end{pmatrix},$$

where

(i). Y_0 is the properly ordered set of the approximate nodal values of u at the coarsest grid points;

(ii). Z_l is the properly ordered set of the incremental unknowns at the level l, which is the increment of u to the average of the values at the neighboring coarse grid points.

Moreover, let S be the transfer matrix from $\overline{U}$ to U

$$U = S\overline{U},$$

where S is nonsingular. Then (1) becomes

$$\overline{A}\,\overline{U} = \overline{g}, \tag{2}$$

where $\overline{A} = S^T A S$, $\overline{g} = S^T g$. Here $\overline{A}$ is called the incremental unknowns matrix of A and has a positive-definite symmetric part, namely $H = \frac{1}{2}(\overline{A}^T + \overline{A})$ is positive-definite, which will be confirmed in Section 5. The convergence properties of incremental unknowns method can be found in [1, 11].

3 Some iterative methods

We consider the following six well-known algorithms: MR, GCR, Orthomin(k), Bi-CGSTAB, HSS and BTSS for approximating the solution of a linear system (1) or (2).

The first three algorithms have some similarities, i.e., the MR algorithm is a special case of the Orthomin(k) algorithm with $k = 0$, and Orthomin(k) is the truncated GCR method (storing k directions). For the convergence rates of the first three methods, we have the following result proved in [7].

Theorem 3.1. *If $\{r(i)\}$ is the sequence of residuals generated by the MR, GCR, Orthomin(k) algorithms, then*

$$\|r(i)\|_2 \leq \left\{1 - \frac{1}{\nu(A)^2}\right\}^{i/2} \|r(0)\|_2. \tag{3}$$

where $\nu(A) = \lambda_{max}(A^T A)^{1/2}/\lambda_{min}(H)$ and $H = \frac{1}{2}(A + A^T)$ is the symmetric part of A.

From Theorem 3.1, the convergence rate depends on the number $\nu(A)$ which is the condition number of A when A is a matrix with a positive-definite symmetric part. In three space dimensions, Garcia has shown that the condition number of the incremental unknowns matrix associated with the Laplace operator is $\mathcal{O}(1/H^2)\mathcal{O}((1/h)|\log(h)|)$ [12], where H is the mesh size of the coarsest grid and h is the mesh size of the finest grid. Furthermore, if block diagonal scaling is used, the condition number of the preconditioned incremental unknowns matrix associated with the Laplace operator turns out to be $\mathcal{O}(1/h)$. In contrast, the condition number of the nodal unknowns matrix associated with the Laplace operator is $\mathcal{O}(1/h^2)$. Hence we observe an improvement in the case of incremental unknowns for the convergence rates.

Bi-CGSTAB is a finite termination method. The theoretical convergence properties of Bi-CGSTAB are very much the same as those of CGS, which can be found in [5, 15]. The essential difference is that often Bi-CGSTAB is more smoothly converging than CGS, i.e., its oscillations are less pronounced. We will give some numerical comparisons between Bi-CGSTAB and MR, GCR, Orthomin(k) for solving the convection-diffusion equations in Section 6.

Due to [13, Theorem 2.2] and [14, Theorem 2.3], we know the HSS and the BTSS iteration methods converge unconditionally to the exact solution of the positive-definite linear system (1). We call the HSSIU method an incremental unknowns method in conjunction with the HSS iterative method and the BTSSIU method an incremental unknowns method in conjunction with the BTSS iterative method for solving the linear system (2). Furthermore, we will give applications of BTSS to the linear system (1) and BTSSIU to the linear system (2) of the convection-diffusion equations whose coefficient matrices possess the block two-by-two structure:

$$A = \begin{pmatrix} W & F \\ E & N \end{pmatrix} = \begin{pmatrix} W & 0 \\ E + F^T & N \end{pmatrix} + \begin{pmatrix} 0 & F \\ -F^T & 0 \end{pmatrix} = T_1 + S_1.$$

The results will be reported in Section 6.

4 Properties of incremental unknowns

We consider $d+1$ levels of meshes in $\mathbb{R}^3$

$$h_0 = (h_{1,0},\, h_{2,0},\, h_{3,0}),$$
$$h_l = (h_{1,l},\, h_{2,l},\, h_{3,l}), \quad h_{i,l} = h_{i,0}/2^l,\ \ 0 \leqslant l \leqslant d,\ \ i = 1,2,3,$$

where $h_{i,0}$ represents the mesh size of the coarsest grid and $h_{i,d}$ represents the mesh size of the finest grid along the x_i direction. Different meshes are allowed in three directions x_1, x_2 and x_3. To these meshes we associate the grids $\Re_l = \Re_{h_l}$ consisting of points $(j_1 h_{1,l},\, j_2 h_{2,l},\, j_3 h_{3,l})$, where $j_1, j_2, j_3 \in \mathbb{Z}$. We denote by $\mathcal{U}_l$ the set of nodal points $\mathcal{U}_l = \Re_l \bigcap \bar{\Omega}$. For $j = (j_1, j_2, j_3) \in \mathbb{Z}^3$ we denote by $K_{j,l}$ the cube

$$K_{j,l} = (j_1 h_{1,l},\, (j_1+1)h_{1,l}) \times (j_2 h_{2,l},\, (j_2+1)h_{2,l}) \times (j_3 h_{3,l},\, (j_3+1)h_{3,l}),$$

and by $\mathcal{T}_l$ the set of all cubes $K_{j,l}$, $0 \leqslant l \leqslant d$.

For the sake of simplicity, we consider the case where Ω is a cuboid $(0,1)^3$ and $h_{i,0} = 1/N$, $i = 1,2,3$, $N \in \mathbb{N}$. We denote by V_l the space of continuous real functions on $\bar{\Omega}$ that are Q_1 (affine with respect to x_1, x_2 and x_3 respectively) on each cube $K_{j,l} \subset \Omega$. Since the cubes $\{K_{j,l} \in \mathcal{T}_l\}$ are obtained by dividing the cubes of $\mathcal{T}_{l-1}$ into eight equal cubes, we observe that $V_0 \subset V_1 \subset \cdots \subset V_d$.

Since $V_{d-1} \subset V_d$, it is useful to define a supplement W_l of V_{l-1} in V_l so that $V_l = V_{l-1} \oplus W_l$, $1 \leqslant l \leqslant d$. By reiteration we then have

$$V_d = V_0 \oplus W_1 \oplus \cdots \oplus W_d. \tag{4}$$

We can also define the decomposition of $u \in V_d$ corresponding to (4), which reads:

$$u = u_0 + \sum_{l=1}^{d} u_l, \qquad u_0 \in V_0, \quad u_l \in W_l. \tag{5}$$

An analysis shows that $u_l(x)$, $x \in \mathcal{U}_l \setminus \mathcal{U}_{l-1}$, $l = 1, \cdots, d$ are the incremental values of u at the different levels of discretization [6]. We then endow V_d with a semi-norm corresponding to the incremental values:

$$[u]_d^2 = \sum_{l=1}^{d} \sum_{x \in \mathcal{U}_l \setminus \mathcal{U}_{l-1}} |u_l(x)|^2.$$

For a function $u \in V_d$ we associate the step function $\tilde{u}$ such that

$$\tilde{u}(x) = u(j_1 h_d, j_2 h_d, j_3 h_d), \quad \forall x \in K_{j,d}, \quad j = (j_1, j_2, j_3), \quad 0 \leqslant j_i \leqslant 2^d N.$$

We observe that $\tilde{u}$ is defined in an extend region $\Omega^*_{h_d}$, where $\Omega^*_{h_d} = (0, 1+h_d)^3$. We first establish the following result:

Lemma 4.1. *For every u of the form (5), there exist constants c_1, c_2, c_3 and c_4 depending only on the shape of Ω such that for every $u \in V_d$,*

$$c_1 \left(\int_{\Omega^*_{h_d}} |\tilde{u}|^2 dx \right)^{\frac{1}{2}} \leqslant |u|_{L^2(\Omega)} \leqslant c_2 \left(\int_{\Omega^*_{h_d}} |\tilde{u}|^2 dx \right)^{\frac{1}{2}}, \tag{6}$$

$$|u|_{L^2(\Omega)} \leqslant c_3 \left\{ |u_0|^2_{L^2(\Omega)} + h_{1,0}h_{2,0}h_{3,0}[u]^2_d \right\}^{\frac{1}{2}}. \tag{7}$$

Consequently,

$$\left(\int_{\Omega^*_{h_d}} |\tilde{u}|^2 dx \right)^{\frac{1}{2}} \leqslant c_4 \left\{ |u_0|^2_{L^2(\Omega)} + h_{1,0}h_{2,0}h_{3,0}[u]^2_d \right\}^{\frac{1}{2}}. \tag{8}$$

Proof The results (6) and (7) are proved in [11, Lemma 8] and [11, Lemma 5]. The estimate (8) can be obtained by simply combining (6) and (7). ■

We define the finite difference operators $\nabla_{h_d} = (\nabla_{1,h_d}, \nabla_{2,h_d}, \nabla_{3,h_d})^T$:

$$\nabla_{i,h_d} u(x) = \frac{1}{h_{i,d}} \{ u(x + h_{i,d} e_i) - u(x) \}, \quad i = 1, 2, 3,$$

where $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$ and $e_3 = (0, 0, 1)$. It is easy to observe that ∇_{1,h_d}, ∇_{2,h_d} and ∇_{3,h_d} are defined in the extended region $\Omega^*_{1,d} = (0, 1) \times (0, 1 + h_d) \times (0, 1 + h_d)$, $\Omega^*_{2,d} = (0, 1 + h_d) \times (0, 1) \times (0, 1 + h_d)$ and $\Omega^*_{3,d} = (0, 1 + h_d) \times (0, 1 + h_d) \times (0, 1)$, respectively. Let S_d be the transfer matrix from the incremental unknowns $\overline{U}_d$ to the nodal unknowns U_d. Then we obtain the following lemma.

Lemma 4.2. *There exists a constant c_5 depending only on the shape of Ω, such that*

$$\|S_d\|_2 \leqslant c_5 \cdot 8^{d/2}. \tag{9}$$

Proof Observe

$$\|S_d\|^2_2 = \sup_{\overline{U}_d \neq 0} \frac{\langle S_d \overline{U}_d, S_d \overline{U}_d \rangle}{\langle \overline{U}_d, \overline{U}_d \rangle} = \sup_{\overline{U}_d \neq 0} \frac{\langle U_d, U_d \rangle}{\langle \overline{U}_d, \overline{U}_d \rangle},$$

where $\langle \cdot, \cdot \rangle$ denotes the usual Euclidean scalar product. Let $\tilde{u}$ be the step function associated with U_d. It follows from (8) that

$$\langle U_d, U_d \rangle = \frac{1}{h_{1,d}h_{2,d}h_{3,d}} \int_{\Omega^*_{h_d}} |\tilde{u}|^2 dx \leqslant \frac{c_4^2}{h_{1,d}h_{2,d}h_{3,d}} \left\{ |u_0|^2_{L^2(\Omega)} + h_{1,0}h_{2,0}h_{3,0}[u]^2_d \right\}.$$

Thanks to (6), we get

$$|u_0|^2_{L^2(\Omega)} \leqslant c_2^2 \int_{\Omega^*_{h_d}} |\tilde{u}_0|^2 dx \leqslant c_2^2 h_{1,0}h_{2,0}h_{3,0} \langle Y_0, Y_0 \rangle$$

and $[u]^2_d = \langle Z^d, Z^d \rangle$. Therefore,

$$\langle U_d, U_d \rangle \leqslant \frac{c_4^2 \cdot 8^d}{h_{1,0}h_{2,0}h_{3,0}} \left\{ c_2^2 h_{1,0}h_{2,0}h_{3,0} \langle Y_0, Y_0 \rangle + h_{1,0}h_{2,0}h_{3,0} \langle Z^d, Z^d \rangle \right\}.$$

By taking $c_5 = (c_4^2 \max\{c_2^2, 1\})^{\frac{1}{2}}$, we obtain

$$\langle U_d, U_d \rangle \leqslant c_5^2 \cdot 8^d \langle \overline{U}_d, \overline{U}_d \rangle,$$

which leads to (9). ■

5 Incremental Unknowns for convection-diffusion equations

We consider the convection-diffusion equations given in Section 2 on the cube $\Omega = (0, 1)^3$ with the homogeneous Dirichlet boundary conditions. Let d, N be nonnegative integers with $d \geqslant 1$ and $N \geqslant 2$, $h_0 = 1/N$, $h_d = h_0/2^d$. The discrete equation with mesh size h_d is:

$$\begin{aligned}(L_dU_d)_{m,n,e} &= \frac{1}{h_d^2}(2u_{m,n,e} - u_{m-1,n,e} - u_{m+1,n,e}) + \frac{1}{h_d^2}(2u_{m,n,e} - u_{m,n-1,e} - u_{m,n+1,e}) \\ &\quad + \frac{1}{h_d^2}(2u_{m,n,e} - u_{m,n,e-1} - u_{m,n,e+1}) + \frac{b}{2h_d}(u_{m+1,n,e} - u_{m-1,n,e}) \\ &\quad + \frac{b}{2h_d}(u_{m,n+1,e} - u_{m,n-1,e}) + \frac{b}{2h_d}(u_{m,n,e+1} - u_{m,n,e-1}) \\ &= f_{m,n,e}, \qquad m, n, e = 1, \cdots, 2^dN - 1, \qquad (10) \\ u_{m,n,e} &= 0, \qquad \text{if } m \text{ or } n \text{ or } e = 0 \text{ or } 2^dN, \qquad (11)\end{aligned}$$

where $u_{m,n,e}$ and $f_{m,n,e}$ are the approximate values of u and f at (mh_d, nh_d, eh_d) respectively. Let $A_d, B_{i,d}, C_{i,d}, i = 1, 2, 3$ be the matrix forms of the difference operators defined by

$$\begin{aligned}(A_dU_d)_{m,n,e} &= (2u_{m,n,e} - u_{m-1,n,e} - u_{m+1,n,e}) + (2u_{m,n,e} - u_{m,n-1,e} - u_{m,n+1,e}) \\ &\quad + (2u_{m,n,e} - u_{m,n,e-1} - u_{m,n,e+1}), \\ (B_{1,d}U_d)_{m,n,e} &= u_{m+1,n,e} - u_{m,n,e}, \quad (B_{2,d}U_d)_{m,n,e} = u_{m,n+1,e} - u_{m,n,e}, \\ (B_{3,d}U_d)_{m,n,e} &= u_{m,n,e+1} - u_{m,n,e}, \quad (C_{1,d}U_d)_{m,n,e} = u_{m,n,e} - u_{m-1,n,e}, \\ (C_{2,d}U_d)_{m,n,e} &= u_{m,n,e} - u_{m,n-1,e}, \quad (C_{3,d}U_d)_{m,n,e} = u_{m,n,e} - u_{m,n,e-1}.\end{aligned}$$

for $m, n, e = 1, \cdots, 2^dN - 1$.

Now we can write (10) and (11) in the following matrix form:

$$L_dU_d = \frac{1}{h_d^2}A_dU_d - \frac{b}{2h_d}[(B_{1,d} + C_{1,d}) + (B_{2,d} + C_{2,d}) + (B_{3,d} + C_{3,d})]U_d = f_d.$$

When $b \neq 0$, L_d is nonsymmetric. Using the boundary condition (11), we obtain by simple calculation that

$$\langle L_dU_d, U_d\rangle = \frac{1}{h_d^2}\langle A_dU_d, U_d\rangle.$$

Because A_d is positive-definite, L_d is also positive-definite. Consider the incremental unknowns $\overline{U}_d$. Let S_d be the transfer matrix from $\overline{U}_d$ to U_d such that $U_d = S_d\overline{U}_d$. Then we get

$$\overline{L}_d\overline{U}_d = \overline{f_d},$$

where $\overline{L}_d = S_d^TL_dS_d$ and $\overline{f}_d = S_d^Tf_d$.

Since S_d is a nonsingular matrix,

$$\begin{aligned}\langle\frac{1}{2}(\overline{L}_d + \overline{L}_d^T)\overline{U}_d, \overline{U}_d\rangle &= \langle\overline{L}_d\overline{U}_d, \overline{U}_d\rangle \\ &= \langle S_d^TL_dS_dS_d^{-1}U_d, S_d^{-1}U_d\rangle \\ &= \langle L_dU_d, U_d\rangle,\end{aligned}$$

it is evident that $\overline{L}_d$ has a positive-definite symmetric part. Due to Theorem 3.1 in Section 3 and [5,13,14], if the six iterative methods are used on the linear system

$$\overline{L}_d\overline{U}_d = \overline{f}_d,$$

then the incremental unknown methods are convergent and (3) holds for the MR, GCR and Orthomin(k) algorithms.

Theorem 5.1. *There exist constants c_6, c_7 and c_8 depending on the shape of Ω but not on h_d, such that*

$$\lambda_{\min}(H) \geqslant \frac{c_6}{d^3 h_d}, \tag{12}$$

$$\lambda_{\max}(\overline{L}_d^T\overline{L}_d)^{\frac{1}{2}} \leqslant c_7\frac{d}{h_d^2}(1 + bd^{-\frac{1}{2}}8^{\frac{d}{2}}h_d), \tag{13}$$

and hence,

$$\nu(\overline{L}_d) \leqslant c_8\log^4(h_d)\left\{h_d^{-1} + b[-\log(h_d)8^{\log(h_d)}]^{-\frac{1}{2}}\right\},$$

where $H = \frac{1}{2}(\overline{L}_d^T + \overline{L}_d)$ and $\nu(\overline{L}_d) = \lambda_{\max}(\overline{L}_d^T\overline{L}_d)^{\frac{1}{2}}/\lambda_{\min}(H)$. Therefore, to reduce the residual of the approximation solution by a factor ε while the residual is generated by one of the MR, GCR and Orthomin(k) algorithms, we need at most k iterations, where

$$k = 2c_8^2\log^8(h_d)\left\{h_d^{-1} + b[-\log(h_d)8^{\log(h_d)}]^{-\frac{1}{2}}\right\}^2 |\log(\varepsilon)|. \tag{14}$$

Proof Thanks to [11, Theorem 1], there exist two constants c_1' and c_2' which depend only on the shape of Ω, such that for any $\overline{U}_d$

$$\begin{aligned}\frac{c_1' h_d}{d^3} &\leqslant \frac{\langle A_dU_d, U_d\rangle}{\langle\overline{U}_d, \overline{U}_d\rangle} \\ &= \frac{\int_{\Omega^*_{1,h_d}}(\nabla_{1,h_d}\tilde{u})^2dx + \int_{\Omega^*_{2,h_d}}(\nabla_{2,h_d}\tilde{u})^2dx + \int_{\Omega^*_{3,h_d}}(\nabla_{3,h_d}\tilde{u})^2dx}{\langle\overline{U}_d, \overline{U}_d\rangle} \leqslant c_2'd,\end{aligned} \tag{15}$$

where $\tilde{u}$ is the step function associated with U_d. We obtain

$$\begin{aligned}\lambda_{min}(H) &= \inf_{\overline{U}_d\neq 0}\frac{\langle\frac{1}{2}(\overline{L}_d^T + \overline{L}_d)\overline{U}_d, \overline{U}_d\rangle}{\langle\overline{U}_d, \overline{U}_d\rangle} = \inf_{\overline{U}_d\neq 0}\frac{\langle\overline{L}_d\overline{U}_d, \overline{U}_d\rangle}{\langle\overline{U}_d, \overline{U}_d\rangle} \\ &= \inf_{\overline{U}_d\neq 0}\frac{1}{h_d^2}\frac{\langle A_dU_d, U_d\rangle}{\langle\overline{U}_d, \overline{U}_d\rangle} \geqslant \frac{c_1'}{d^3h_d},\end{aligned}$$

with $c_6 = c_1'$. Then the inequality (12) is proved.

If we define the matrices $\overline{A}_d = S_d^TA_dS_d$, $\overline{B}_{i,d} = S_d^TB_{i,d}S_d$ and $\overline{C}_{i,d} = S_d^TC_{i,d}S_d$, then

$$\begin{aligned}\lambda_{\max}(\overline{L}_d^T\overline{L}_d)^{\frac{1}{2}} = \|\overline{L}_d\|_2 \leqslant\ & \frac{1}{h_d^2}\|\overline{A}_d\|_2 + \frac{b}{2h_d}\big(\|\overline{B}_{1,d}\|_2 + \|\overline{C}_{1,d}\|_2 \\ & + \|\overline{B}_{2,d}\|_2 + \|\overline{C}_{2,d}\|_2 + \|\overline{B}_{3,d}\|_2 + \|\overline{C}_{3,d}\|_2\big).\end{aligned}$$

It follows from (15) that

$$\|\overline{A}_d\|_2 = \lambda_{\max}(\overline{A}_d) = \sup_{\overline{U}_d\neq 0}\frac{\langle A_dU_d, U_d\rangle}{\langle\overline{U}_d, \overline{U}_d\rangle} \leqslant c_2'd.$$

With the help of Lemma 4.2,

$$\|\overline{B}_{1,d}\|_2 = \|S_d^T B_{1,d} S_d\|_2 \leqslant \|S_d^T\|_2 \|B_{1,d} S_d\|_2 \leqslant c_5 \cdot 8^{d/2} \|B_{1,d} S_d\|_2.$$

Since

$$\begin{aligned}\|B_{1,d} S_d\|_2^2 &= \sup_{\overline{U}_d \neq 0} \frac{\langle B_{1,d} S_d \overline{U}_d,\, B_{1,d} S_d \overline{U}_d \rangle}{\langle \overline{U}_d,\, \overline{U}_d \rangle} = \sup_{\overline{U}_d \neq 0} \frac{\langle B_{1,d} U_d,\, B_{1,d} U_d \rangle}{\langle \overline{U}_d,\, \overline{U}_d \rangle} \\ &= \sup_{\overline{U}_d \neq 0} \frac{\int_{\Omega^*_{1,h_d}} (\nabla_{1,h_d} \tilde{u})^2 dx}{\langle \overline{U}_d,\, \overline{U}_d \rangle} \leqslant c_2' d,\end{aligned}$$

we obtain

$$\|B_{1,d} S_d\|_2 \leqslant (c_2' d)^{\frac{1}{2}} \quad , \quad \|\overline{B}_{1,d}\|_2 \leqslant c_5 \cdot 8^{d/2} (c_2' d)^{\frac{1}{2}}.$$

Similarly, we can obtain that the upper bounds of $\|\overline{C}_{1,d}\|_2$, $\|\overline{B}_{2,d}\|_2$, $\|\overline{C}_{2,d}\|_2$, $\|\overline{B}_{3,d}\|_2$ and $\|\overline{C}_{3,d}\|_2$ are $c_5 \cdot 8^{d/2} (c_2' d)^{\frac{1}{2}}$. Therefore,

$$\begin{aligned}\lambda_{\max}(\overline{L}_d^T \overline{L}_d)^{\frac{1}{2}} &\leqslant \frac{c_2' d}{h_d^2} + c_5 \frac{3b \cdot 8^{d/2}}{h_d} (c_2' d)^{\frac{1}{2}} \\ &\leqslant \frac{d}{h_d^2} \left\{ c_2' + 3 (c_2')^{\frac{1}{2}} c_5 d^{-\frac{1}{2}} b 8^{d/2} h_d \right\}.\end{aligned}$$

With $c_7 = \max\{c_2',\ 3(c_2')^{\frac{1}{2}} c_5\}$, the inequality (13) follows. Consequently,

$$\begin{aligned}\nu(\overline{L}_d) &= \lambda_{\max}(\overline{L}_d^T \overline{L}_d)^{\frac{1}{2}} / \lambda_{\min}(H) \leqslant c_7 / c_6 d^4 (h_d^{-1} + b d^{-\frac{1}{2}} 8^{\frac{d}{2}}) \\ &\leqslant c_8 \log^4(h_d) \left\{ h_d^{-1} + b[-\log(h_d) 8^{\log(h_d)}]^{-\frac{1}{2}} \right\}.\end{aligned}$$

To reduce the residual of the solution by a factor ε with k steps, using (3) with (12) and (13), it is sufficient to require

$$\left\{ 1 - \frac{1}{c_8^2 \log^8(h_d) \{ h_d^{-1} + b[-\log(h_d) 8^{\log(h_d)}]^{-\frac{1}{2}} \}^2} \right\}^{\frac{k}{2}} \leqslant \varepsilon,$$

or equivalently,

$$k \geqslant \frac{2|\log(\varepsilon)|}{|\log(1 - c_8^{-2} \log^{-8}(h_d) \{ h_d^{-1} + b[-\log(h_d) 8^{\log(h_d)}]^{-\frac{1}{2}} \}^{-2})|}.$$

Since $|\log(1-x)| \geqslant x$, there holds

$$k \geqslant 2 c_8^2 \log^8(h_d) \left\{ h_d^{-1} + b[-\log(h_d) 8^{\log(h_d)}]^{-\frac{1}{2}} \right\}^2 |\log(\varepsilon)|. \quad \blacksquare$$

Remark 5.1. If $b = 0$, then $(\overline{L}_d)$ is the incremental unknowns matrix associated with the Laplace problem in three space dimensions. As a result, we can recover the condition number $\mathcal{O}(h_d^{-1} \log^4(h_d))$ in [11].

6 Numerical results

In this section, we perform several examples to examine effectiveness and feasibility of the incremental unknown methods.

Example 1 Consider the system of linear equations (1), for which A_d is the matrix generated from the two-level finite difference discretization of the three-dimensional convection-diffusion equation on the unit cube $\Omega = [0,1]^3$ with the homogeneous Dirichlet boundary conditions. For our numerical tests, $b = 0.1$ is taken and the step sizes along three directions are the same, i.e., $h_d = 1/2^d n$, $d = 1$. The discrete values at nodal points are taken as those of $u(x,y,z) = 100xyz(1-x)(1-y)(1-z)$ which is the exact solution. The right-hand side f_d is obtained by computing $f_d = A_d U_d$, with U_d being the exact solution.

We use the two-level incremental unknowns method in conjunction with the four iterative methods: GCR, MR, Orthomin(1) and Bi-CGSTAB, to approximate the solution of the $(2n-1)^3$-by-$(2n-1)^3$ linear system (2). Without special claim, all our tests in this subsection are started from zero vector and terminated when the iteration satisfies $\|r^{(k)}\|_2/\|r^{(0)}\|_2 < 10^{-6}$, where $r^{(k)}$ is the residual of the kth iteration.

We illustrate the plots of the two level incremental unknowns ($n = 4$, $d = 1$) and the nodal unknowns ordered by different levels in Figs. 1-4 for the four algorithms. From Figs. 1-4, one can see that the incremental unknowns in the fine mesh are less than the nodal unknowns in the fine mesh, and the incremental unknowns in the coarsest mesh are equivalent to the nodal unknowns in the coarsest mesh, which correspond to their definition. Furthermore, the four-dimensional slices of the exact solution and the numerical solution are plotted in the left and the right of Fig. 5, respectively, which use a two level uniform $15 \times 15 \times 15$ mesh ($n = 8$, $d = 1$) and are approximate to each other.

The relative L^2 norm(log10) of the error is plotted against the iteration number in the left of Fig. 6. The effectiveness of the Bi-CGSTAB algorithms and other algorithms are compared with the same initial error on a uniform $9 \times 9 \times 9$ mesh. The left of Fig. 6 shows the decrease in error with respect to the iterations. We see that with the Bi-CGSTAB method, the error decreases faster than that of the MR, GCR, Orthomin(1) methods.

The relative L^2 norm(log10) of the error is plotted against the CPU time used in the right of Fig. 6. Here we use a uniform $9 \times 9 \times 9$ mesh. It is observed that the Bi-CGSTAB method uses the least CPU time with a given accuracy requirement.

We observe that using IU in conjunction with the Bi-CGSTAB iteration method can save CPU time and use less number of iteration with the required accuracy. Although the CPU time of the MR and Orthomin(1) iteration method is longer than that of the Bi-CGSTAB method, we can see that they are still two stable methods.

Example 2 We will compare HSS, BTSS, HSSIU and BTSSIU, and demonstrate the convergence behavior of the incremental unknowns method in conjunction with HSS and BTSS. All our numerical tests are started from zero vector and terminated when the iteration satisfies $\|r^{(k)}\|_2/\|r^{(0)}\|_2 < 10^{-4}$, where $r^{(k)}$ is the residual of the kth iteration. Here $b = 10$ is taken and the step sizes along three directions are the same, i.e., $h_d = 1/2^d n$, $d = 1$, $\Omega = [0,1]^3$. The discrete values at nodal points are those of $u(x,y,z) = 100xyz(1-x)(1-y)(1-z)$ which is taken as the exact solution. The right-hand side f_d is obtained by computing $f_d = A_d U_d$, where U_d is the exact solution and the size of the matrix A_d is $(2^d n)^3 \times (2^d n)^3$.

In Table 1, we list the experimentally optimal parameter α_{exp} and the corresponding spectral radii $\rho(M(\alpha_{exp}))$ of the iteration matrices $M(\alpha_{exp})$ of the HSS, BTSS, HSSIU and BTSSIU methods for different n. It is observed that when n is increasing, α_{exp} is decreasing and $\rho(M(\alpha_{exp}))$ is increasing for HSS, BTSS, and HSSIU. But $\rho(M(\alpha_{exp}))$ of BTSSIU has a little change when

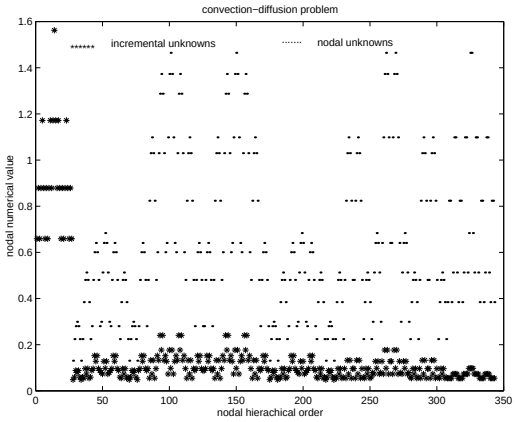

Figure 1: The approximate solution and the incremental unknowns ordered by different levels with the MR method, $7 \times 7 \times 7$ mesh.

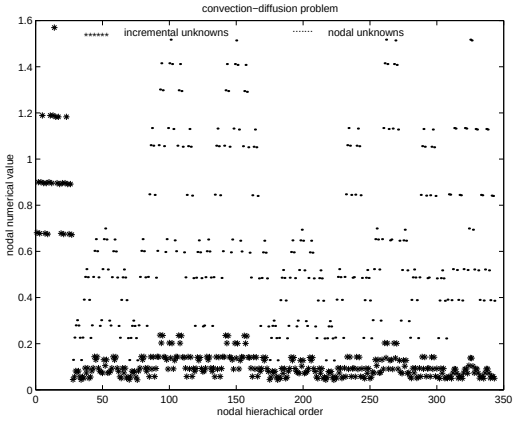

Figure 2: The approximate solution and the incremental unknowns ordered by different levels with the GCR method, $7 \times 7 \times 7$ mesh.

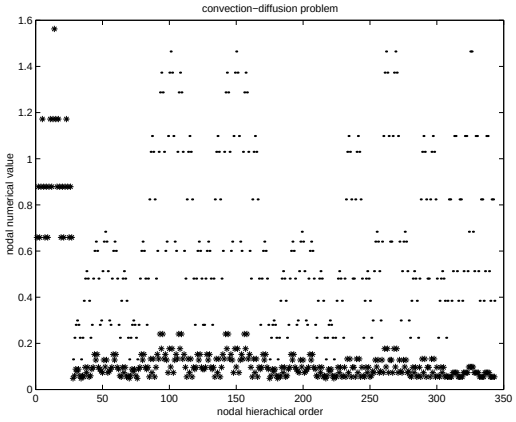

Figure 3: The approximate solution and the incremental unknowns ordered by different levels with the $Bi-CGSTAB$ method, $7 \times 7 \times 7$ mesh.

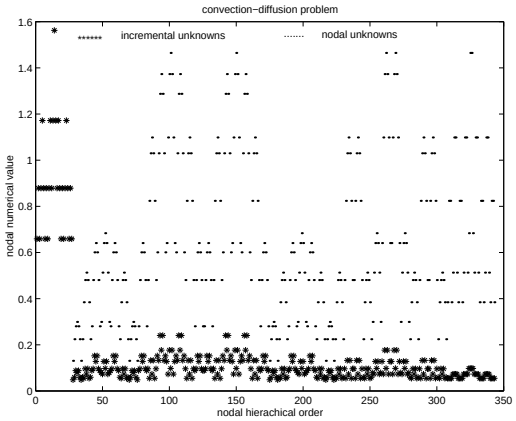

Figure 4: The approximate solution and the incremental unknowns ordered by different levels with the $Orthomin(1)$ method, $7 \times 7 \times 7$ mesh.

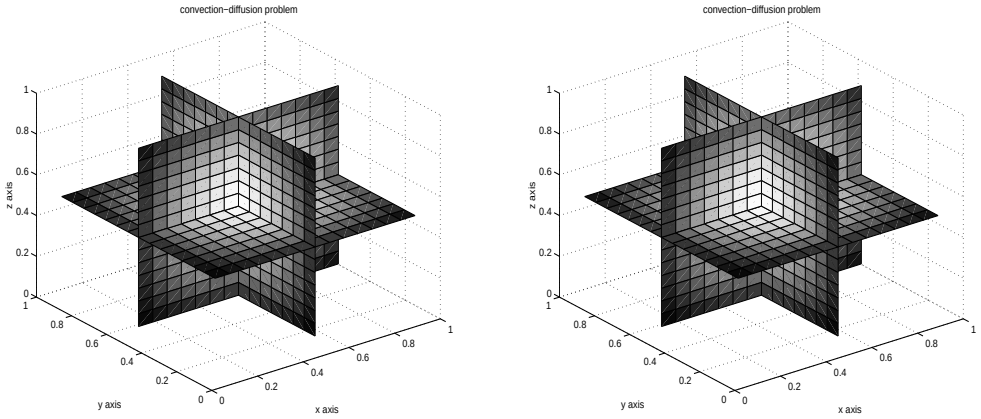

Figure 5: The four-dimensional slices of the exact solution (left) and the numerical solution (right), with a $15 \times 15 \times 15$ mesh.

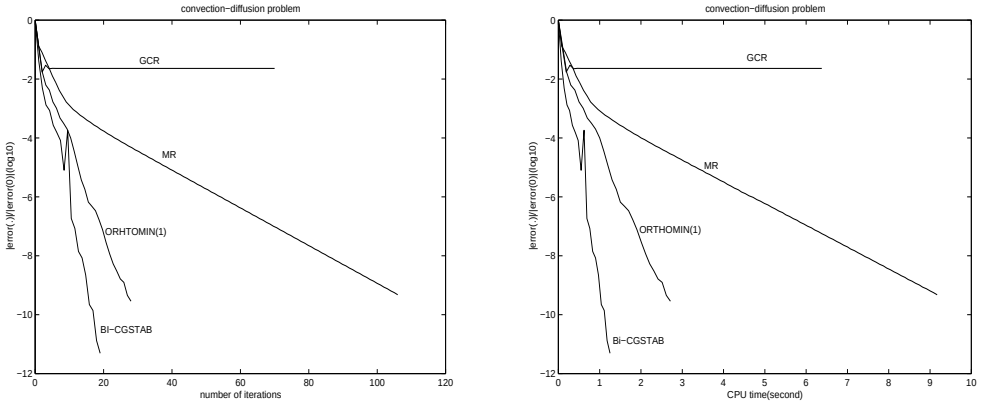

Figure 6: The relative L^2 norm(log10) of the error against the iteration number (left) and the CPU time (right) by using IU in conjunction with the Bi-CGSTAB algorithm and the MR, GCR, Orthomin(1) algorithms, $9 \times 9 \times 9$ mesh.

Table 1: α_{exp} and $\rho(M(\alpha_{exp}))$ for Example 2.

n		3	4	5	6
HSS	α_{exp}	3.9575	3.8050	2.9005	2.3996
	$\rho(M(\alpha_{exp}))$	0.3163	0.4043	0.4352	0.4971
BTSS	α_{exp}	4.1103	2.8032	2.5034	2.6116
	$\rho(M(\alpha_{exp}))$	0.3234	0.5204	0.6163	0.7544
HSSIU	α_{exp}	2.5968	2.3881	2.3103	2.202
	$\rho(M(\alpha_{exp}))$	0.4569	0.4916	0.5115	0.5293
BTSSIU	α_{exp}	3.7452	3.5003	3.1285	2.8957
	$\rho(M(\alpha_{exp}))$	0.6972	0.7210	0.7117	0.6939

Table 2: IT and CPU time for Example 2.

n		3	4	5	6
HSS	IT	13	17	18	27
	CPU	0.01560	1.4060	10.0620	40.3600
BTSS	IT	8	13	25	25
	CPU	0.0650	0.6250	7.5150	31.1720
HSSIU	IT	13	15	15	16
	CPU	0.1560	1.1250	7.7650	31.1250
BTSSIU	IT	10	11	11	11
	CPU	0.0940	0.5940	3.4530	13.9060

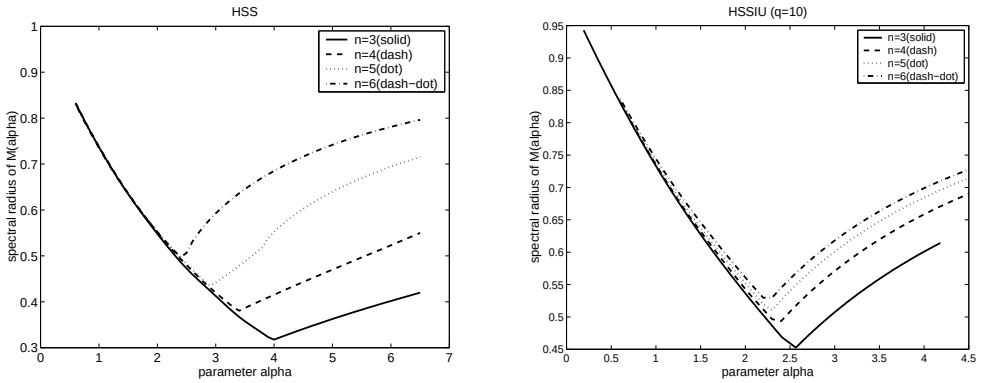

Figure 7: The curves of the spectral radii of the iteration matrices with respect to different α for the HSS (left) and HSSIU (right) iteration methods.

α_{exp} is decreasing. This is confirmed by Figs. 7 and 8, which depict the curves of the spectral radii of the iteration matrices with respect to different α for HSS, BTSS, HSSIU and BTSSIU. Evidently, from Fig. 7 we see that $\rho(M(\alpha_{exp}))$ attains the minimum at about α_{exp}, and monotonically increases when n is increasing, and that the increment of spectral radii of HSSIU is less than that of spectral radii of HSS. The same situation is observed in Fig. 8. In particular, we can see that $\rho(M(\alpha_{exp}))$ of BTSSIU is less than that of BTSS when n is large. Because BTSS has the same computational workload as BTSSIU, the actual computing CPU time of BTSS may be more than that of BTSSIU. These facts are further confirmed by the numerical results listed in Table 2. Because BTSS has a less computational workload than HSS at each IT, the actual

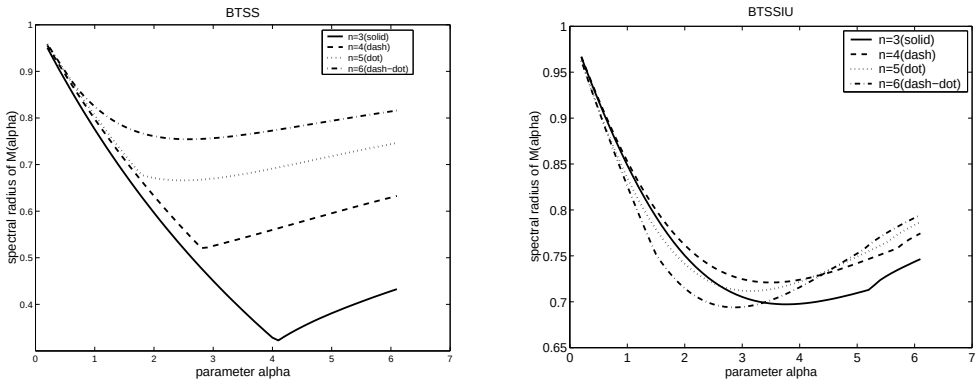

Figure 8: The curves of the spectral radii of the iteration matrices with respect to different α for the BTSS (left) and BTSSIU (right) iteration methods.

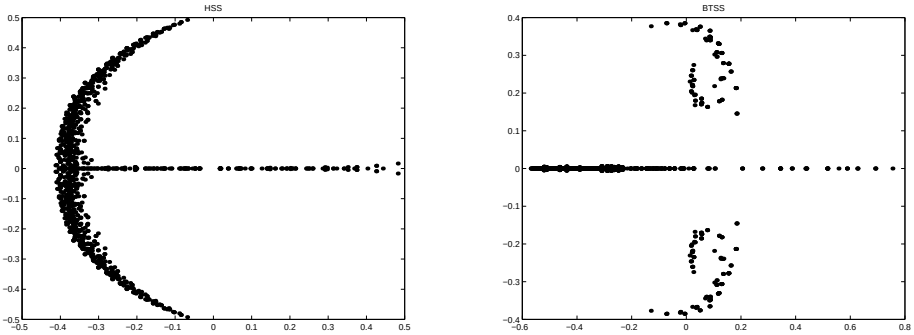

Figure 9: The eigenvalue distributions for the HSS (left) and BTSS (right) methods.

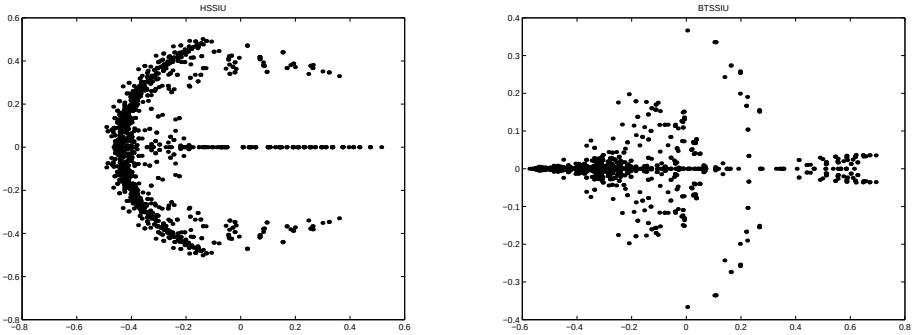

Figure 10: The eigenvalue distributions for the HSSIU (left) and BTSSIU (right) methods.

CPU time of BTSS may be much less than that of HSS. It is concluded from these figures and tables that BTSS will be more efficient than HSS. From Table 2, we can see that the IT (number of iteration) and CPU time of HSS are more than those of HSSIU respectively, which also occur for BTSS and BTSSIU.

In Figs. 9 and 10, we plot the eigenvalues of the iteration matrices of the HSS, BTSS, HSSIU and BTSSIU using the optimal parameter α when $b = 10$ and $n = 6$, $d = 1$. It is clear that the eigenvalue distributions of the four iteration matrices are quite different. The eigenvalues of the

HSS iteration matrix are tightly clustered around the real axis and a circular arc on the complex plane, while those of the BTSS iteration matrix are clustered on the real axis and two arcs on the complex plane. Moreover, the eigenvalues of the BTSS iteration matrix is considerably less than that of the HSS iteration matrix along the direction of the imaginary axis. However, the eigenvalue distributions of the BTSSIU and HSSIU iteration matrices become complex and keep their original figures partly while the eigenvalues appear to be clustered around the origin point.

We have applied the two-level incremental unknowns to the HSS and BTSS iteration methods, namely, the HSSIU and BTSSIU iteration methods with the two-level discretization. Due to the convergence of HSS and BTSS, HSSIU and BTSSIU inherit advantages of those methods. Numerical examples have been implemented to show that, in the sense of computational storage and CPU time, the HSSIU and BTSSIU methods are much more efficient than the HSS and BTSS methods. This confirms that the incremental unknowns method is an effective one, especially in conjunction with some iteration methods.

Acknowledgments

The authors would like to thank the referees for their useful suggestions.

References

[1] Temam R, Inertial manifolds and multigrid methods. SIAM J. Math. Anal., 1990, 21(1): 154-178.
[2] Garcia S, The matricial framework for the incremental unknowns methods. Numer. Func. Anal. Opt., 1993, 25(14): 25-44.
[3] Chen M, Temam R, Incremental unknowns for solving partial differential equations. Numer. Math., 1991, 59(3): 255-271.
[4] Garcia S, Numerical study of the incremental unknowns method. Numer. Math. Part. D. E., 1994, 10(1): 103-127.
[5] Van der Vorst H A, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 1992, 13(2): 631-644.
[6] Chen M, Temam R, Incremental unknowns in finite differences: condition number of the matrix. SIAM J. Matrix Anal. Appl., 1993, 14(2): 432-455.
[7] Eisenstat S C, Elman H C, Schultz M H, Variational iterative methods for nonsymmetric system of linear equations. SIAM J. Numer. Anal., 1983, 20: 345-357.
[8] Chen M, Temam R, Incremental unknowns for convection-diffusion equations. Appl. Numer. Math., 1993, 11: 365-385.
[9] Garcia S, Algebraic conditioning analysis of the incremental unknowns preconditioner. Appl. Math. Model, 1998, 22: 351-366.
[10] Golub G H, Van Loan C F, Matrix Computations. The John Hopkins University Press, Baltimore, 1989.
[11] Chen M, Miranville A, Temam R, Incremental unknowns in finite differences in space dimensions 3. Comput. Appl. Math., 1995, 14(3): 1-15.
[12] Garcia S, Incremental unknowns and graph techniques in three space dimensions. Appl. Numer. Math., 2003, 44: 329-365.
[13] Bai Z Z, Golub G H, NG M K, Hermitian and skew-hermitian splitting methods for non-hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl., 2003, 24(3): 603-626.
[14] Bai Z Z, Golub G H, Lu L Z, Yin J F, Block triangular and skew-hermitian splitting methods for positive-definite linear systems. SIAM J. Sci. Comput., 2005, 26(3): 844-863.
[15] Van der Vorst H A, The convergence behaviour of preconditioned CG and CG-S in the presence of rounding errors. Lect. Notes Math., 1990, 1457: 126-136.