Numerical Algorithms (2005) **38:** 285–303

Alternating oblique projections for coupled linear systems

Luis Manuel Hernández-Ramos

Laboratoire "Jacques-Louis Lions", Université "Pierre et Marie Curie" (Paris VI), Boite courrier 187, 75252 Paris Cedex 05, France
Centro de Calculo Cientifico y Tecnologico, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela
E-mail: luish@kuaimare.ciens.ucv.ve

Received 14 November 2002; accepted 20 September 2004
Communicated by G. Meurant

In this work we propose the use of alternating oblique projections (AOP) for the solution of the saddle points systems resulting from the discretization of domain decomposition problems. These systems are called coupled linear systems. The AOP method is a descent method in which the descent direction is defined by using alternating oblique projections onto the search subspaces. We prove that this method is a preconditioned simple gradient (Uzawa) method with a particular preconditioner. Finally, a preconditioned conjugate gradient based version of AOP is proposed.

Keywords: domain decomposition, Uzawa, alternating projections, conjugate gradient method, preconditioner

AMS subject classification: 65F10, 65N22, 65Y05

1. Problem definition

We are interested in the system of linear equations:

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{1}$$

where

H1. $B \in \Re^{m\times n}$ is a matrix with $\operatorname{rank}(B) = m$, $m \leqslant n$,

H2. $A \in \Re^{n\times n}$ is a symmetric positive definite matrix.

Proposition 1.1. Under the hypotheses H1, H2 the system (1) has a unique solution $(x, \lambda)^{\mathrm{T}}$ which satisfies:

$$\begin{cases} f - Ax \perp \ker B, \\ x \in \ker B. \end{cases} \tag{2}$$

Proof. Under hypotheses H1, H2 the system

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ 0 & BA^{-1}B^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ BA^{-1}f \end{pmatrix}, \tag{3}$$

is equivalent to (1) and $BA^{-1}B^{\mathrm{T}}$ is a positive definite matrix. Consequently the system (1) has a unique solution. This solution x verifies $f - Ax = B^{\mathrm{T}}\lambda \perp \ker B$. □

Let V be the linear variety defined by

$$V = \{x / f - Ax \perp \ker B\}. \tag{4}$$

Under the hypotheses H1, H2, the solution $(x, \lambda)^{\mathrm{T}}$ of (1) is such that x is the only element in the intersection of V and $\ker B$,

$$\{x\} = V \cap \ker B. \tag{5}$$

Let $x_u = A^{-1}f$, the system (2) becomes

$$\begin{cases} \forall y \in \ker B, \quad \langle A(x_u - x), y\rangle = 0, \\ x \in \ker B. \end{cases} \tag{6}$$

When A is a symmetric positive definite matrix, the mapping $\{x, y\} \mapsto \langle Ax, y\rangle$ is a scalar product denoted by $\langle .,. \rangle_A$ and the system (1) is equivalent to:

$$\begin{cases} x_u = A^{-1}f, \\ x_u - x \perp_A \ker B, \\ x \in \ker B. \end{cases} \tag{7}$$

It means that x is the orthogonal projection of $x_u = A^{-1}f$ onto $\ker B$, in the sense of the scalar produit $\langle .,. \rangle_A$. The notation $x_u - x \perp_A \ker B$ means $\langle x_u - x, y\rangle_A = 0, \forall y \in \ker B$. In this case we can write the linear variety as $V = \{x_u\} + (\ker B)^{\perp_A}$.

2. Alternating oblique projections

From this point on we assume the hypotheses H1, H2. All the subsequent theoretical results are based on those two hypotheses. But, in practice, the resulting methods will be only used when:

- the computational cost of solving $Ay = b$ is reasonable, and
- the orthogonal projection P onto $\ker B$ is easily obtained.

We shall concentrate on the second point.

Definition 2.1 (AOP method). We define the alternating oblique projections method as,

$$\begin{cases} x_0 \in V, \\ x_{k+1} = x_k + \alpha_k d_k, \end{cases} \tag{8}$$

with

$$d_k = \widetilde{Q} P x_k - x_k, \tag{9}$$

where

- the descent direction d_k is an approximation of $e_k = x - x_k$,
- α_k is the stepsize,
- P is the orthogonal projection matrix onto $\ker B$,
- $\widetilde{Q}$ is a projection matrix onto the linear variety V.

The orthogonal projection onto the linear variety $V = x_u + \ker B^{\perp_A}$ is very expensive to calculate because this involves computing the orthogonal projection $\overline{R} = A^{-1}B^{\mathrm{T}}(BA^{-2}B^{\mathrm{T}})^{-1}BA^{-1}$ onto $\ker B^{\perp_A}$. Hence, we propose to use the oblique projection $\widetilde{Q}$ onto V defined by

$$\forall y \in \Re^n, \quad \widetilde{Q}y := A^{-1}Pf + Qy, \tag{10}$$

where

$$\begin{cases} R = I - P, & \text{the orthogonal projection matrix onto } (\ker B)^{\perp}, \\ Q = A^{-1}RA, \end{cases} \tag{11}$$

thus

$$\widetilde{Q}y = A^{-1}Pf + A^{-1}(I - P)Ay \tag{12}$$

$$= A^{-1}P(f - Ay) + y. \tag{13}$$

Lemma 2.1. Under the hypotheses (H1), $Q = A^{-1}RA$ satisfies,

1. $Q^2 = Q$, thus Q is a projection,
2. $\forall y \in \Re^n$, $Qy \perp_A \ker B$,
3. $\forall y \in \Re^n$, $(y - Qy) \perp_A (\ker B)^{\perp}$.

Proof.

$$\begin{aligned} &Q^2 = A^{-1}RAA^{-1}RA = A^{-1}R^2A = A^{-1}RA, \\ &Qy = A^{-1}RAy, && \text{then } AQy = RAy \in (\ker B)^{\perp} \quad \text{and} \quad Qy \in (\ker B)^{\perp_A}, \\ &y - Qy = A^{-1}PAy, && \text{thus } A(y - Qy) = PAy \in \ker B, \\ & && \text{then } A(y - Qy) \perp \ker B^{\perp} \end{aligned}$$

and, finally,

$$(y - Qy) \perp_A \ker B^{\perp}. \qquad \square$$

Comments. It is easy to verify that $\widetilde{Q}$ is a projection onto V. Moreover $(f - A\widetilde{Q}y) \perp \ker B$ and $\widetilde{Q}^2 = \widetilde{Q}$.

3. Analysis of the method

M1. We analyze the following method:

Initialization

$x_0 \in V$

iterations: for $k = 0, 1, \ldots$ **do**

- $d_k = -A^{-1}RARx_k$
- $\alpha_k = \langle -x_k, d_k\rangle_A / \langle d_k, d_k\rangle_A = -\langle Ad_k, x_k\rangle / \langle Ad_k, d_k\rangle$
- $x_{k+1} = x_k + \alpha_k d_k$

end

In lemma 3.1, item 4, we will show that M1 is equivalent to the AOP method given by (8) and (9).

3.1. Properties

Lemma 3.1. Under hypotheses H1, H2, method M1 satisfies: for all $k \geqslant 0$,

1. $d_k \in \ker B^{\perp_A}$,
2. $x_k \in V$,
3. $\langle Ad_k, d_k\rangle = 0 \Leftrightarrow x_k = x$; thus the sequence is stationary,
4. $d_k = \widetilde{Q}Px_k - x_k$, then method M1 becomes the AOP method given by (8) and (9).

Proof. 1. We have

$$d_k = -A^{-1}RARx_k, \tag{14}$$

then

$$\begin{aligned} Ad_k = -RARx_k \in (\ker B)^{\perp} \quad &\Rightarrow \quad \forall y \in \ker B, \langle Ad_k, y\rangle = 0 \\ &\Leftrightarrow \quad d_k \in (\ker B)^{\perp_A}. \end{aligned}$$

2. Let

$$x_0 \in V; \tag{15}$$

by induction, we suppose $x_k \in V$. As $d_k \in (\ker B)^{\perp_A}$ then $\alpha_k d_k \in (\ker B)^{\perp_A}$ therefore $x_{k+1} \in V$.

3. $\langle Ad_k, d_k\rangle = 0 \Rightarrow d_k = 0$ because A is symmetric definite positive,

$$\begin{aligned} &\Rightarrow RARx_k = 0 \\ &\Rightarrow \langle ARx_k, Rx_k\rangle = 0 \\ &\Rightarrow Rx_k = 0 \\ &\Rightarrow x_k = Px_k \end{aligned}$$

$$\Rightarrow x_k \in \ker B \text{ and } x_k \in V$$
$$\Rightarrow x_k = x,$$

and the sequence is stationary.

If $x_k = x$ then $Rx_k = 0$, which implies $d_k = 0$ and $\langle Ad_k, d_k\rangle = 0$.

4. We have

$$\widetilde{Q}Px_k - x_k = A^{-1}Pf + A^{-1}RAPx_k - x_k, \tag{16}$$

which implies

$$\begin{aligned} A\big(\widetilde{Q}Px_k - x_k\big) &= Pf + RAPx_k - Ax_k \\ &= Pf + RAx_k - RARx_k - Ax_k \\ &= Pf - PAx_k - RARx_k \\ &= P(f - Ax_k) - RARx_k, \end{aligned}$$

but $f - Ax_k \perp \ker B$ because $x_k \in V$, thus $P(f - Ax_k) = 0$ therefore,

$$A\big(\widetilde{Q}Px_k - x_k\big) = -RARx_k = Ad_k, \tag{17}$$

finally,

$$\widetilde{Q}Px_k - x_k = d_k. \tag{18}$$

□

Lemma 3.2. Under hypotheses H1, H2, method M1 satisfies: for all $k \geqslant 0$,

1. $\langle x, d_k\rangle_A = 0$,
2. $\langle e_k, d_k\rangle_A = \langle -x_k, d_k\rangle_A$, with $e_k = x - x_k$,
3. $\langle e_{k+1}, d_k\rangle_A = 0$,
4. $\langle e_k, d_k\rangle_A = \|Rx_k\|_A^2$.

Proof. 1. We have $x \in \ker B$ thus $\langle Ad_k, x\rangle = 0$ by lemma 3.1, item 1.

2. By definition,

$$\langle e_k, d_k\rangle_A = \langle x - x_k, d_k\rangle_A = \langle x, d_k\rangle_A - \langle x_k, d_k\rangle_A, \tag{19}$$

but $\langle x, d_k\rangle_A = 0$ thus,

$$\langle e_k, d_k\rangle_A = \langle -x_k, d_k\rangle_A. \tag{20}$$

3. Next

$$\begin{aligned} \langle e_{k+1}, d_k\rangle_A &= \langle e_k, d_k\rangle_A - \alpha_k\langle d_k, d_k\rangle_A \\ &= -\langle x_k, d_k\rangle_A + \langle x_k, d_k\rangle_A = 0. \end{aligned}$$

So in this method, $\alpha_k d_k$ is the projection of e_k onto d_k, thus it is an error projection method (cf. [15]).

4. Finally,

$$\begin{aligned}\langle e_k, d_k\rangle_A &= \langle -x_k, Ad_k\rangle \\ &= \langle -x_k, -RARx_k\rangle \\ &= \langle Rx_k, ARx_k\rangle \\ &= \|Rx_k\|_A^2. \end{aligned}$$

□

Lemma 3.3. Under hypotheses H1, H2, method M1 satisfies: for all $k \geqslant 0$,

1. $\|e_{k+1}\|_A^2 = \|e_k\|_A^2 - \|Rx_k\|_A^4/\|d_k\|_A^2 = \|e_k\|_A^2 - \alpha_k^2\|d_k\|_A^2$,
2. $\|e_{k+1}\|_A \leqslant \|e_k\|_A$,
3. $\|e_{k+1}\|_A = \|e_k\|_A$ iff $x = x_k$.

Proof. 1. We have

$$\begin{aligned}\|e_{k+1}\|_A^2 &= \langle e_{k+1}, e_k - \alpha_k d_k\rangle_A \\ &= \langle e_{k+1}, e_k\rangle_A \quad \text{by the lemma 3.2, item 3.}\end{aligned}$$

Thus

$$\|e_{k+1}\|_A^2 = \|e_k\|_A^2 - \alpha_k\langle e_k, d_k\rangle_A; \tag{21}$$

by substituting $\alpha_k = \langle -x_k, d_k\rangle_A/\langle d_k, d_k\rangle_A$ and by the use of lemma 3.2, item 4, we have,

$$\begin{aligned}\|e_{k+1}\|_A^2 &= \|e_k\|_A^2 - \frac{\langle x_k, d_k\rangle_A^2}{\langle d_k, d_k\rangle_A} \\ &= \|e_k\|_A^2 - \frac{\|Rx_k\|_A^4}{\|d_k\|_A^2} = \|e_k\|_A^2 - \alpha_k^2\|d_k\|_A^2.\end{aligned}$$

2. As corollary of part (1),

$$\|e_{k+1}\|_A \leqslant \|e_k\|_A. \tag{22}$$

3. From (1)

$$\|e_{k+1}\|_A = \|e_k\|_A \quad \Rightarrow \quad \|Rx_k\|_A = 0, \tag{23}$$

but

$$\|Rx_k\|_A = 0 \quad \Rightarrow \quad Rx_k = 0 \quad \Rightarrow \quad x_k \in \ker B. \tag{24}$$

From lemma 3.1, item 2, $x_k \in V$, then $x_k \in \ker B \cap V$ and, finally, $x_k = x$. □

Lemma 3.4. Under hypotheses H1, H2 and if $e_k \neq 0$, method M1 satisfies: for all $k \geqslant 0$,

1. $\|e_{k+1}\|_A = \|e_k\|_A\sqrt{1-\rho_k^2}$, where $\rho_k = \|Rx_k\|_A^2/(\|d_k\|_A\|e_k\|_A)$,

2. $\rho_k \geqslant (\|Rx_k\|_A/\|e_k\|_A)(1/\|Q\|_A)$, where $Q = A^{-1}RA$ is the oblique projection into $(\ker B)^{\perp_A}$.
3. $\|Rx_k\|_A/\|e_k\|_A \geqslant 1$,
4. $\|e_{k+1}\|_A \leqslant \|e_k\|_A\sqrt{1-\rho^2}$, where $\rho = 1/\|Q\|_A > 0$. Then the method converges.

Proof. 1. From lemma 3.3, item 1, we have for $e_k \neq 0$,

$$\|e_{k+1}\|_A^2 = \|e_k\|_A^2\left[1 - \frac{\|Rx_k\|_A^4}{\|d_k\|_A^2\|e_k\|_A^2}\right], \tag{25}$$

thus,

$$\|e_{k+1}\|_A = \|e_k\|_A\sqrt{1-\rho_k^2}, \tag{26}$$

where $\rho_k = \|Rx_k\|_A^2/(\|d_k\|_A\|e_k\|_A)$.

Remark. By lemma 3.2, item 4

$$\frac{\|Rx_k\|_A^2}{\|d_k\|_A\|e_k\|_A} = \frac{\langle e_k, d_k\rangle_A}{\|d_k\|_A\|e_k\|_A} = \cos\angle_A(e_k, d_k). \tag{27}$$

2. We have

$$\|d_k\|_A = \left\|A^{-1}RARx_k\right\|_A \leqslant \|Q\|_A\|Rx_k\|_A \tag{28}$$

and

$$\rho_k \geqslant \frac{\|Rx_k\|_A}{\|e_k\|_A}\frac{1}{\|Q\|_A}. \tag{29}$$

3. We write

$$\begin{aligned}\|Rx_k\|_A^2 &= \|x_k - Px_k\|_A^2\\ &= \|x_k - x\|_A^2 + \|x - Px_k\|_A^2 + 2\langle x_k - x, x - Px_k\rangle_A.\end{aligned}$$

But $x \in \ker B$ and $Px_k \in \ker B$ and $x_k - x \perp_A \ker B$. Thus

$$\|Rx_k\|_A^2 - \|e_k\|_A^2 = \|x - Px_k\|_A^2, \tag{30}$$

hence

$$\|Rx_k\|_A^2 \geqslant \|e_k\|_A^2 \quad\Rightarrow\quad \|Rx_k\|_A \geqslant \|e_k\|_A \tag{31}$$

and, finally,

$$\frac{\|Rx_k\|_A}{\|e_k\|_A} \geqslant 1. \tag{32}$$

Remark. This result gives us one practical criterion for stopping the algorithm. We can use $\|Rx_k\|_A \leqslant \varepsilon$, $\varepsilon > 0$ to stop it and we guarantee that $\varepsilon \geqslant \|Rx_k\|_A \geqslant \|e_k\|_A$.

4. From parts 2 and 3 we choose

$$\rho_k \geqslant \rho = \frac{1}{\|Q\|_A} > 0, \tag{33}$$

thus

$$\|e_{k+1}\|_A \leqslant \|e_k\|_A \sqrt{1-\rho^2}, \tag{34}$$

and method M1 converges. □

4. Relationship with Uzawa's method

The Uzawa method preconditioned by any matrix Q_B^{-1} is written as:

$$\begin{cases} \lambda_0 \in \Re^m, \\ x_k = A^{-1}\big(f - B^{\mathrm{T}}\lambda_k\big), \\ \lambda_{k+1} = \lambda_k + \alpha_k Q_B^{-1} B x_k. \end{cases} \tag{35}$$

From (35) we have

$$x_{k+1} = A^{-1}\big(f - B^{\mathrm{T}}\lambda_{k+1}\big), \tag{36}$$

but

$$\lambda_{k+1} = \lambda_k + \alpha_k Q_B^{-1} B x_k, \tag{37}$$

thus, by substitution,

$$x_{k+1} = A^{-1}\big(f - B^{\mathrm{T}}\lambda_k - \alpha_k B^{\mathrm{T}} Q_B^{-1} B x_k\big) \tag{38}$$

$$= A^{-1}\big(f - B^{\mathrm{T}}\lambda_k\big) - \alpha_k A^{-1} B^{\mathrm{T}} Q_B^{-1} B x_k \tag{39}$$

$$= x_k - \alpha_k A^{-1} B^{\mathrm{T}} Q_B^{-1} B x_k. \tag{40}$$

We choose the preconditioning matrix $Q_B^{-1} = (B^+)^{\mathrm{T}} A B^+ \approx (BA^{-1}B^{\mathrm{T}})^{-1} = S^{-1}$, where $B^+ = B^{\mathrm{T}}(BB^{\mathrm{T}})^{-1}$ is the pseudoinverse of B. This matrix approaches the inverse of the Schur complement matrix $S = BA^{-1}B^{\mathrm{T}}$ of the system (1). Hence by (40), the preconditioned Uzawa method becomes

$$x_{k+1} = x_k - \alpha_k A^{-1} B^{\mathrm{T}} Q_B^{-1} B x_k \tag{41}$$

$$= x_k - \alpha_k A^{-1} B^{\mathrm{T}} (B^+)^{\mathrm{T}} A B^+ B x_k, \tag{42}$$

but $B^+ B = R$ (cf. [11]) thus,

$$x_{k+1} = x_k - \alpha_k A^{-1} R A R x_k \tag{43}$$

$$= x_k + \alpha_k d_k, \tag{44}$$

with the descent direction $d_k = -A^{-1} R A R x_k$, which coincides with the AOP method proposed in definition 2.1. Therefore, the AOP method is equivalent to using the Uzawa method preconditioned by $Q_B^{-1} = (B^+)^{\mathrm{T}} A B^+$.

From (35), we get the approximation error of λ_k which comes from (cf. [4])

$$E_\lambda^{k+1} = \lambda - \lambda_{k+1} = \left(I - \alpha_k Q_B^{-1} S\right) E_\lambda^k, \tag{45}$$

and with the choice $\alpha_k = \langle -x_k, z_k\rangle / \langle d_k, z_k\rangle$ the AOP method proposed in M1, we minimize $\|E_\lambda^{k+1}\|_S$ in the descent direction d_k (the same choice of α_k is frequently used for the Uzawa method in symmetric case).

5. Acceleration of the conjugate gradient method

If the matrix A is symmetric positive definite, the preconditioning matrix $Q_B^{-1} = (B^+)^{\mathrm{T}} A B^+$ satisfies:

- $(Q_B^{-1})^{\mathrm{T}} = (B^+)^{\mathrm{T}} A B^+ = Q_B^{-1}$, thus the matrix is symmetric.
- $\forall x \in \Re^m$: $\langle x, Q_B^{-1} x\rangle = \|B^+ x\|_A$. Hence, if $\langle x, Q_B^{-1} x\rangle = 0$, then $\|B^+ x\|_A = 0$ and $B^+ x = 0$. When the range of B is maximal then $x = 0$ and the preconditioning matrix is positive definite.

Hence, we propose here to accelerate the conjugate gradient method applied to the condensed system $BA^{-1}B^{\mathrm{T}}\lambda = BA^{-1}f$, with this preconditioner.

5.1. The preconditioned conjugate gradient algorithm

We write here the preconditioned conjugate gradient algorithm with $Q_B^{-1} = (B^+)^{\mathrm{T}} A B^+$. We take advantage of the relation $B^{\mathrm{T}} Q_B^{-1} B x_k = B^{\mathrm{T}} (B^+)^{\mathrm{T}} A B^+ B x_k = RARx_k$:

1. **Initialization:**
   - $x_0 \in V$
   - $w_0 = RARx_0$
   - $y_0 = w_0$
2. **Iteration: For** $k = 0, 1, \ldots,$ **Do**
   - $d_k = A^{-1} y_k$
   - $\alpha_k = \langle x_k, w_k\rangle / \langle d_k, y_k\rangle$
   - $x_{k+1} = x_k - \alpha_k d_k$
   - $w_{k+1} = RARx_{k+1}$
   - $\beta_{k+1} = \langle x_{k+1}, w_{k+1}\rangle / \langle x_k, w_k\rangle$
   - $y_{k+1} = w_{k+1} + \beta_{k+1} y_k$
3. **End**

Remark. Note that:

- The algorithm does not explicitly need to calculate either λ_k or the vector residual.
- The right-hand side f occurs in the choice of the initial vector $x_0 \in V = \{x/f - Ax \perp_A \ker B\}$.

6. Numerical experiments

6.1. Introduction

In this section we describe the results obtained when solving system (1), arising from the domain decomposition method applied to the Poisson equation:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{46}$$

where Ω is a bounded rectangular domain of $\mathbb{R}^2$. Let $\partial\Omega$ be its boundary and f a given function.

The domain Ω is subdivided into four subdomains Ω_i, $i = 1, \ldots, 4$, separated by one interface Γ. We define the matrices A_i, $i = 1, \ldots, 4$, each one corresponding to the discretization of (46) independently in each subdomain Ω_i, $i = 1, \ldots, 4$. This discretization is carried out with the help of the 5 point-stencil on a uniform grid with $n_i \times n_i$ nodes. The matrix A is then defined from the matrix A_i as:

$$\begin{pmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 \\ 0 & 0 & A_3 & 0 \\ 0 & 0 & 0 & A_4 \end{pmatrix}. \tag{47}$$

A is a block diagonal matrix which satisfies hypothesis H2.

The matrix B is related to the matching condition on the interface Γ. B is a "coupling" matrix in the sense of the definition proposed in [11]. We subdivide Γ into subinterfaces Γ_{ij}, $i, j = 1, \ldots, 4$, defined as the part of the interface Γ which separates the subdomains Ω_i and Ω_j; thus, the matrix B is formed by the matrix B_{ij} related to the matching condition on the interface Γ_{ij}. The matrix B_{ij} comes from the discretization of the coupling condition

$$\forall v_i \in V_i,\ v_j \in V_j,\ w_{ij} \in W_{ij}: \quad \int_{\Gamma_{ij}} (v_i - v_j) w_{ij}\, \mathrm{d}x = 0, \tag{48}$$

where

- v_i is the trace of the local solution defined in the subdomain Ω_i on Γ_{ij},
- $V_i = \{v_i \in H'(\Omega_i);\ v_i\ |_{\partial\Omega} = 0\}$ is the space where this local solution is defined,
- w_{ij} is a test function in a proper subspace $W_{ij} = H^{-1/2}(\Gamma_{ij})$.

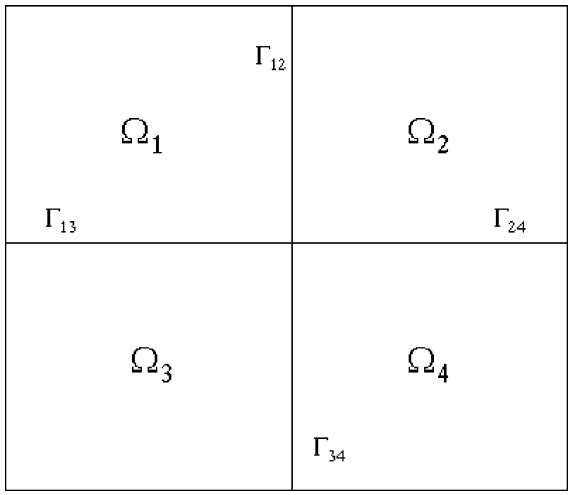

Figure 1. Decomposition of domain Ω.

Thus, in the discretization, the matrix B_{ij} expresses a condition of the type:

$$\forall v_i \in V_i^h,\ v_j \in V_j^h,\ w_{ij} \in W_{ij}^h: \quad \int_{\Gamma_{ij}} (v_i - v_j) w_{ij}\, \mathrm{d}x = 0. \tag{49}$$

For the numerical experiments we have chosen the following space

$$V_i^h = \left[\left\{\Phi_k^{(i)}\right\}_{k=1}^{n_i}\right] \quad \text{the space generated by } \left\{\Phi_k^{(i)}\right\}_{k=1}^{n_i}, \tag{50}$$

where

- n_i is the number of nodes on the interface Γ coming from the subdomain Ω_i.
- Let $\{x_k^{(j)}\}_{k=1}^{n_j}$ be the nodes of the interface which corresponds to the subdomain Ω_j. The functions $\Phi_k^{(j)}$, $j = 1, \ldots, 4$; $k = 1, \ldots, n_j$ are defined by

$$\Phi_k^{(j)}(x) = \begin{cases} \dfrac{x - x_{k-1}^{(j)}}{x_k^{(j)} - x_{k-1}^{(j)}} & \text{if } x \in \left[x_{k-1}^{(j)}, x_k^{(j)}\right], \\ \dfrac{x_{k+1}^{(j)} - x}{x_{k+1}^{(j)} - x_k^{(j)}} & \text{if } x \in \left[x_k^{(j)}, x_{k+1}^{(j)}\right], \\ 0 & \text{elsewhere} \end{cases} \tag{51}$$

for $j = 2, \ldots, n_j - 1$,

$$\Phi_1^{(j)}(x) = \begin{cases} \dfrac{x_2^{(j)} - x}{x_2^{(j)} - x_1^{(j)}} & \text{if } x \in \left[x_1^{(j)}, x_2^{(j)}\right], \\ 0 & \text{elsewhere} \end{cases} \tag{52}$$

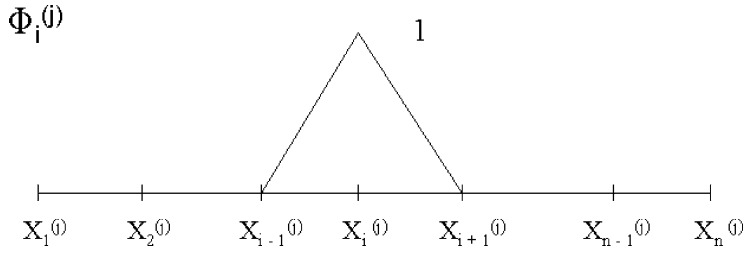

Figure 2. Function $\Phi_i^{(j)}$.

and

$$\Phi_{n_j}^{(j)}(x) = \begin{cases} \dfrac{x - x_{n_j-1}^{(j)}}{x_{n_j}^{(j)} - x_{n_j-1}^{(j)}} & \text{if } x \in \left[x_{n_j-1}^{(j)}, x_{n_j}^{(j)}\right], \\ 0 & \text{elsewhere.} \end{cases} \tag{53}$$

For W_{ij}^h we choose the space which has the smallest dimension between V_i^h and V_j^h, the nodes being distributed uniformly on the interface of each subdomain.

Remark. The mesh in each Ω_i can be independent.

6.2. *Results*

We compare the performance of the AOP method, and the CG preconditioned with the AOP method (CG-AOP) for system (1) with the following methods:

- Uzawa,
- Conjugate Gradient (CG),
- Conjugate Gradient Preconditioned by $Q_B^{-1} = BAB^{\mathrm{T}}$ (CG-Schur) (cf. [13]),
- GMRES preconditioned by the Skew-Hermitian Splitting Iteration (HSS) recently proposed by Benzi and Golub that depends on a parameter $\alpha > 0$ [2,3].

The vector f of (1) corresponds to the exact solution of equation (46): $u(x, y) = \sin(K_1\pi x)\sin(K_2\pi y) + 3$, with $K_1 = 1.7$, $K_2 = 2.3$. The experiments correspond to the so-called conforming case where $n_1 = n_2 = n_3 = n_4$.

All our experiments were run on a Pentium IV using MATLAB 6.1.

6.2.1. *Comparison of the methods*

Figures 3, 4, 5 show the evolution of $\|Rx_k\|_2$ in relation to the number of iteration, in the case where $n_i = 10, 20, 30$, respectively, for the following methods: Uzawa, CG, AOP, CG-AOP.

Comments. The CG-AOP method provides the best convergence results in all the experiments carried out. The convergence rate of the CG-AOP method is clearly superior to those of the other methods.

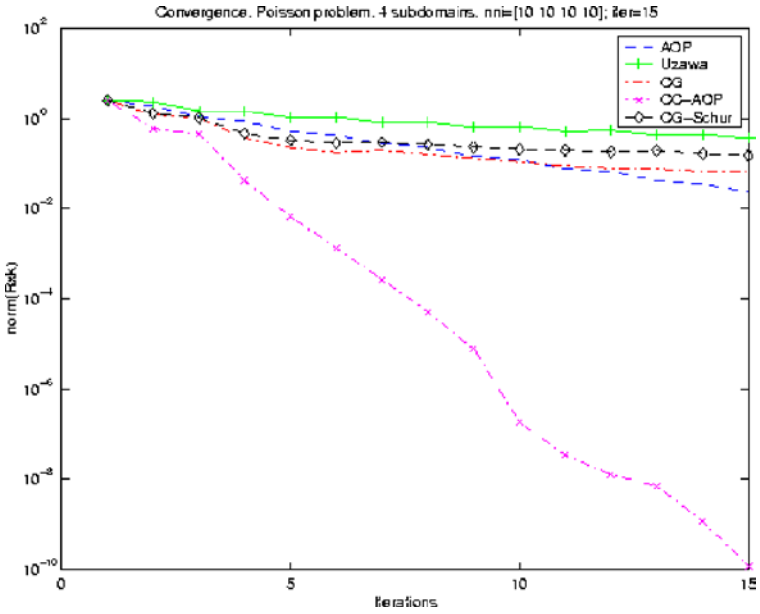

Figure 3. Value of $\|Rx_k\|$ for the problem with $n_i = 10$, $i = 1, \ldots, 4$.

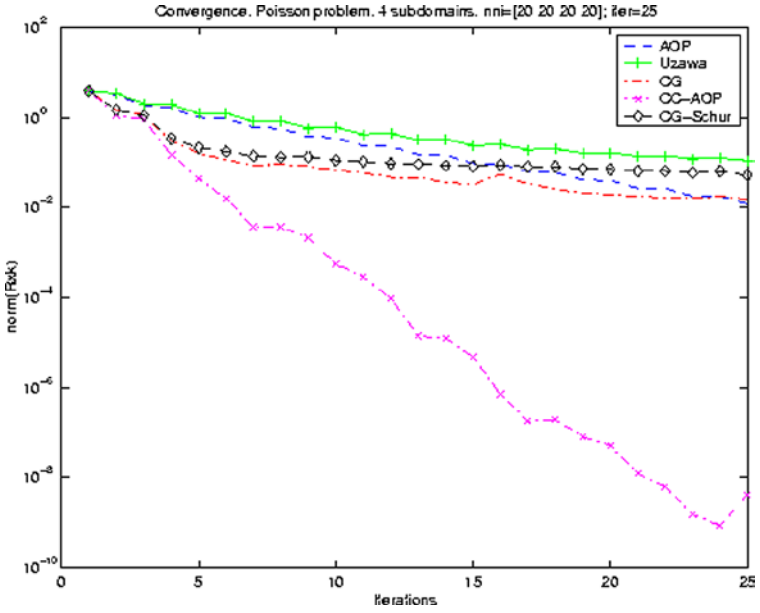

Figure 4. Value of $\|Rx_k\|$ for the problem with $n_i = 20$, $i = 1, \ldots, 4$.

Figures 6 and 7 compares the evolution of $\|Rx_k\|_2$ for the GC-AOP method with GMRES preconditioned by the HSS iteration (Benzi–Golub) for $n_i = 20$ and $n_i = 30$, respectively, and for $\alpha = 10^{-2}$, 10^{-3}, 10^{-4} and 10^{-5}, as suggested in [17].

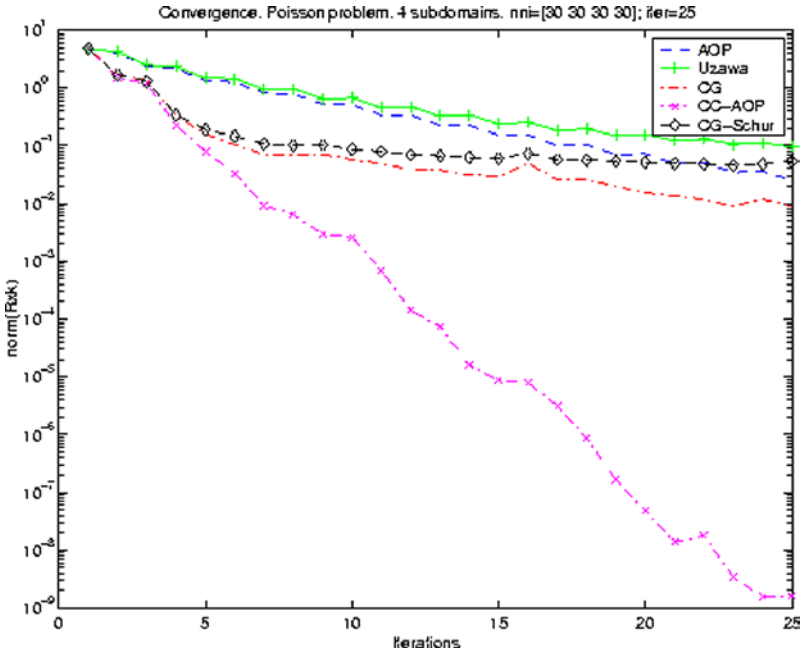

Figure 5. Value of $\|Rx_k\|$ for the problem with $n_i = 30$, $i = 1, \ldots, 4$.

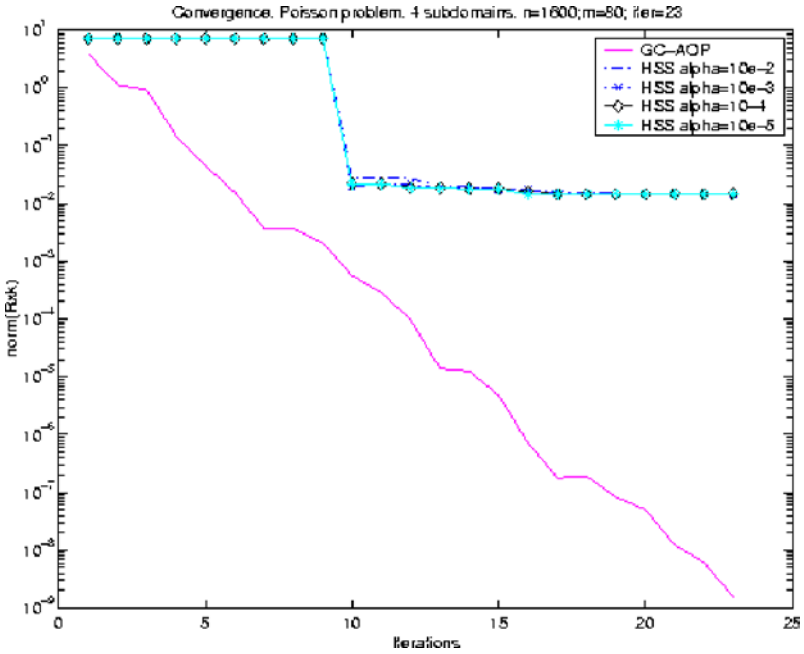

Figure 6. Value of $\|Rx_k\|$ for the problem with $n_i = 20$, $i = 1, \ldots, 4$.

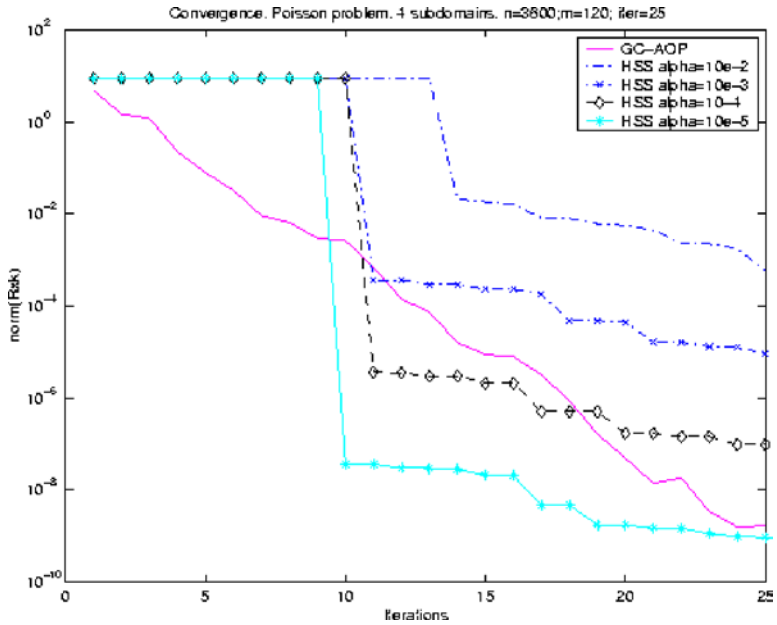

Figure 7. Value of $\|Rx_k\|$ for the problem with $n_i = 30$, $i = 1, \ldots, 4$.

Comments.

- For $n_i = 20$ ($n = 1600$ and $m = 80$) the GC-AOP clearly outperforms the HSS methods for any choice of α. We also observe in figure 6 that the HSS iteration has a similar behavior for $10^{-5} \leqslant \alpha \leqslant 10^{-2}$.
- For $n_i = 30$ ($n = 3600$ and $m = 120$) the best result is obtained by the HSS iteration when $\alpha = 10^{-5}$. Nevertheless, it is interesting to observe that for any choice of α the HSS method shows a significant reduction at some specific iteration but it tends to stagnate before and after that iteration. On the other hand, the GC-AOP method tends to reduce $\|Rx_k\|_2$ uniformly during the whole process. This feature seems to be convenient for inner–outer schemes, associated to nonlinear problems (e.g., KKT-matrix for nonlinear programming), in which low precision in very few iterates is highly recommendable. Interesting discussions on the choice of the parameter α for the HSS method can be found in [2,17].

Value of $\|Rx\|_2$. Figures 8, 9, 10 show the value of $\|Rx\|_2$ for the CG-AOP method with $n_i = 10, 20, 30$, $i = 1, \ldots, 4$. These figures indicate the efficiency of the CG-AOP method in the solution of the coupled linear system solution (1). This does not take into account the discretization error by domain decomposition. Here we realize that the value of $\|Rx_i\|_2$ is higher on the interfaces, but the order of these values is still quite small. Thus allowing us to validate the efficiency of the method in the solution of the coupled linear system (1).

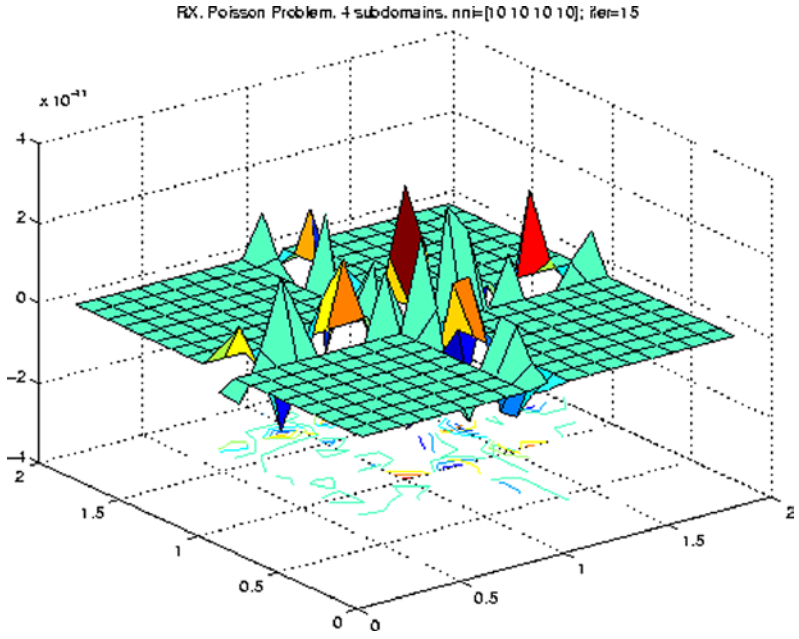

Figure 8. $\|Rx\|_2$ for the CG-AOP method with $n_i = 10, i = 1, \ldots, 4$.

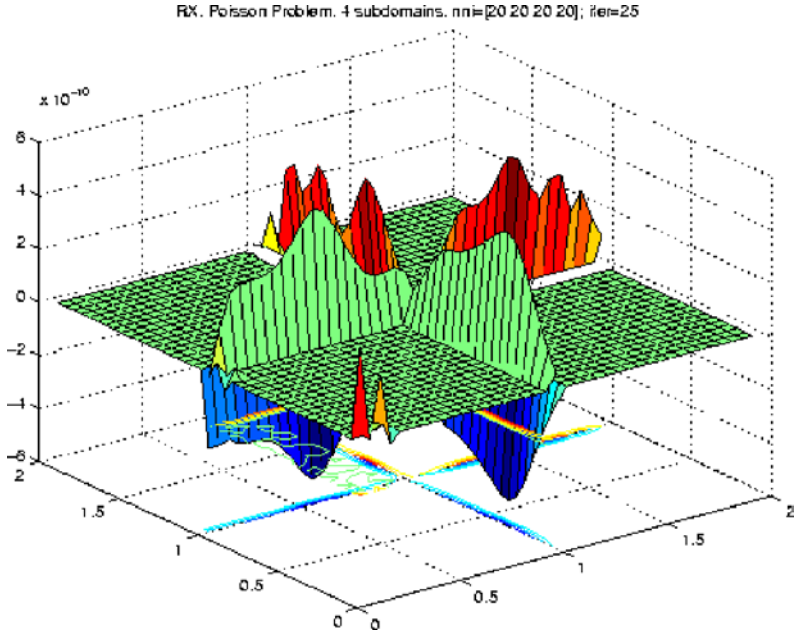

Figure 9. $\|Rx\|_2$ for the CG-AOP method with $n_i = 20, i = 1, \ldots, 4$.

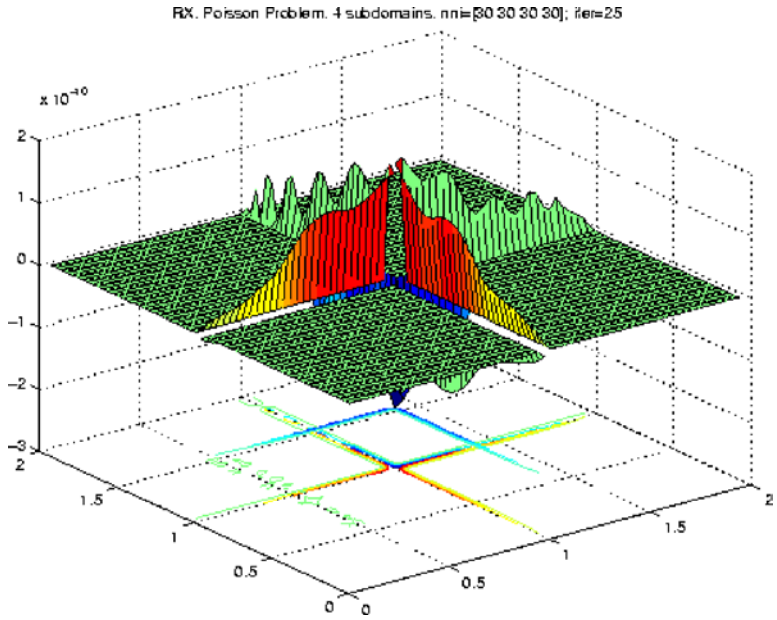

Figure 10. $\|Rx\|_2$ for the CG-AOP method with $n_i = 30$, $i = 1, \ldots, 4$.

Trace on the interface. Figures 11, 12 show the evolution of the trace error in 10 iterations of the CG-AOP method with $n_i = 10, 20, 30$. In figure 11 we show the error on the Γ_{13} and Γ_{24} interfaces. Figure 12 shows the error on the interfaces Γ_{12} and Γ_{34}. Here we still realize that if the number of nodes increases, the approximation has improved. The difference between the horizontal trace and the vertical trace is due to the fact that the form of the error is similar to that of the exact solution. We recall that this solution is given by $u(x, y) = \sin(K_1 \pi x) \sin(K_2 \pi y) + 3$, with $K_1 = 1.7$ and $K_2 = 2.3$, thus the vertical trace is different from the horizontal trace.

7. Conclusions

The results of the experiments presented in this work show that the AOP method is competitive with recent methods, and has some interesting new features. The iteration cost of AOP depends on the cost of the P projection onto the kernel of B. In some cases, as in the case of domain decomposition, the matrix B has a particular structure that can be used to compute P at a reasonable cost (cf. [11]). In this case, the AOP method is a valuable tool for solving the coupled linear systems.

As a continuation of this research, we would like to:

- Experiment with this method on non-conforming cases.
- Compare the computational time of several methods for parallel machines.
- Extend the experiments to several subdomains.
- Extend the alternating oblique projections idea to nonsymmetric matrices.

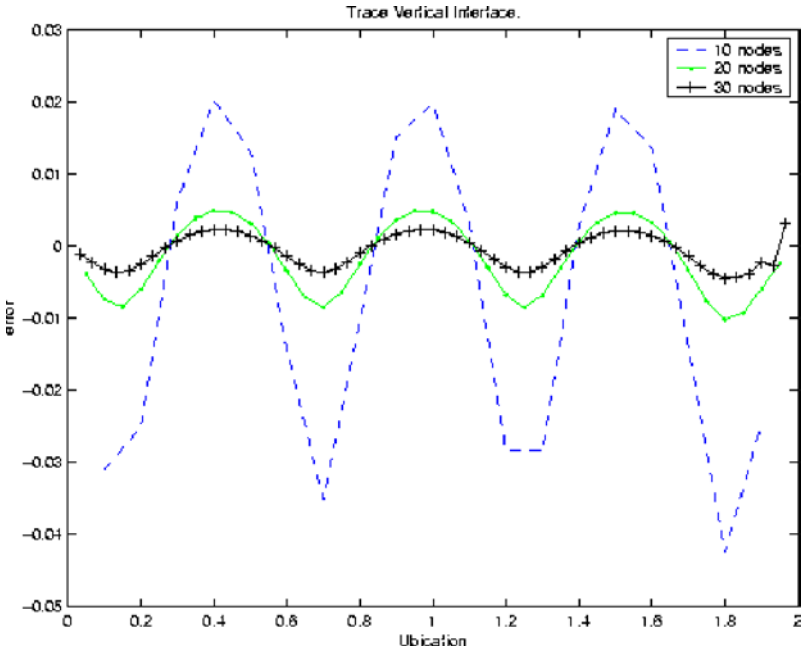

Figure 11. The vertical cut shows the error e_{10} on the interface for the CG-AOP method.

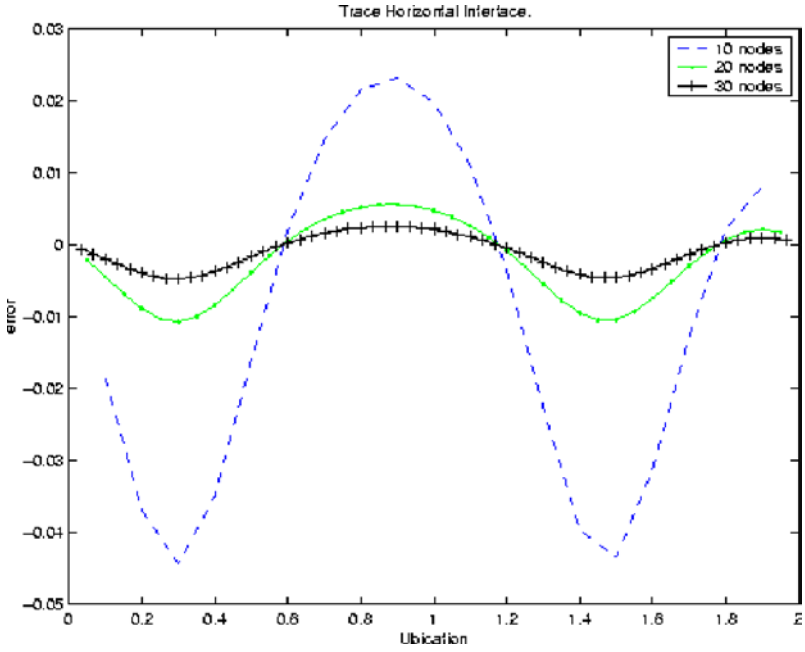

Figure 12. The horizontal cut shows the error e_{10} on the interface for the CG-AOP method.

Acknowledgements

The author thanks an anonymous referee and the associate editor G. Meurant for valuable suggestions, and for pointing out references [2,3]. He is indebted to Michele Benzi for a careful reading of the first version of the paper. He would also like to thank Vivette Girault, Pascal Joly and Marcos Raydan for interesting discussions on this topic.

References

[1] H. Bauschke and J. Borwein, On the convergence of Von Neumann's alternating projection algorithm for two sets, Set Value Anal. 1 (1993) 185–212.
[2] M. Benzi, M.J. Gander and G.H. Golub, Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle point problems, BIT Numer. Math. (2003) acepted.
[3] M. Benzi and G. Golub, An iterative method for generalized saddle point problems, SIAM J. Matrix Anal. Appl. (2002) submitted.
[4] J. Bramble, J. Pasciak and A. Vassilev, Analysis of the inexact Uzawa algorithm for saddle point problems, SIAM J. Numer. Anal. 34(3) (2000) 1072–1092.
[5] J. Bramble, J. Pasciak and A. Vassilev, Uzawa type algorithm for saddle point problems, Math. Comp. 69(230) (2000) 667–689.
[6] C. Brezinski, Multiparameter descent methods, `http://ano.univ-lille1.fr/`.
[7] F. Deutsh, The method of alternating orthogonal projections, in: *Approximation Theory, Splines Functions and Applications* (1992) pp. 105–121.
[8] R. Escalante and M. Raydan, Alternating projections methods: Theory and applications, in preparation.
[9] V. Girault and P. Raviart, *Finite Element Method for Navier–Stokes Equations. Theory and Algorithms* (Springer, Berlin, 1986).
[10] L.M. Hernández, Precondicionamiento de grandes sistemas lineales acoplados para descomposición de dominios, Trabajo Especial de Grado de Magister Scientarum en Computación, Universidad Central de Venezuela, Caracas (1999).
[11] L.M. Hernández, Un opérateur de projection pour les systèmes linéaires couplés, Thèse du diplôme d'Etudes Doctorales de l'Université "Pierre et Marie Curie", Paris (2000).
[12] P. Joly, *Analyse Numérique Matricielle. Cours 2001/2002* (Laboratoire d'Analyse Numérique Paris 6, Paris, 2001).
[13] X. Juvigny, Résolution de grands systèmes linéaires sur des machines massivement parallèles, Thèse de doctorat de l'Université "Pierre et Marie Curie", Paris (1997).
[14] Y. Saad, *Iterative Methods for Sparse Linear Systems* (PSW, 1996).
[15] Y. Saad, Further analysis of minimum residual iterations, Numer. Linear Algebra Appl. 7(2) (2000) 67–93.
[16] Y. Saad and H. Van der Vorst, Iterative solution of linear systems in the 20th century, Numerical Analysis 2000, Vol. III, Linear Algebra, J. Comput. Appl. Math. 12(1/2) (2000) 1–33.
[17] V. Simoncini and M. Benzi, Spectral properties of the Hermitian and skew Hermitian splitting preconditioner for saddle point problems, SIAM J. Matrix Anal. Appl. (2003) to appear.
[18] J. Xu and L. Zikatanov, The method of alternating projections and the method of subspace corrections in Hilbert space, Report No. AM223, Penn State, Department of Mathematics (2000).