Augmented conjugate gradient. Application in an iterative process for the solution of scattering problems

M.O. Bristeau [a] and J. Erhel [b]

[a] *INRIA, B.P. 105, Rocquencourt, F-78153 Le Chesnay Cedex, France*
E-mail: marie-odile.bristeau@inria.fr

[b] *INRIA, Campus Universitaire de Beaulieu, F-35042 Rennes Cedex, France*
E-mail: jocelyne.erhel@irisa.fr

Received 19 January 1998; revised June 1998
Communicated by G. Meurant

We discuss the application of an augmented conjugate gradient to the solution of a sequence of linear systems of the same matrix appearing in an iterative process for the solution of scattering problems. The conjugate gradient method applied to the first system generates a Krylov subspace, then for the following systems, a modified conjugate gradient is applied using orthogonal projections on this subspace to compute an initial guess and modified descent directions leading to a better convergence. The scattering problem is treated via an Exact Controllability formulation and a preconditioned conjugate gradient algorithm is introduced. The set of linear systems to be solved are associated to this preconditioning. The efficiency of the method is tested on different 3D acoustic problems.

Keywords: conjugate gradient, Krylov subspace, least-squares, scattering

AMS subject classification: 65F10, 65C20

1. Introduction

In this paper, we discuss an application of the *Augmented Conjugate Gradient* algorithm proposed by Erhel and Guyomarc'h in [11]. This algorithm concerns a sequence of symmetric linear systems of the form

$$Ax^{(i)} = b^{(i)},$$

where A is an $N \times N$ symmetric positive definite matrix and where the different right-hand sides $b^{(i)}$ are computed sequentially; this situation arises for instance when a new right-hand side depends upon previous solutions.

The basic idea of this conjugate gradient method is to use a *Krylov subspace* generated by the first system to define an initial guess and modified descent directions to solve the following systems. An analogous idea has been applied to the Lanczos method in [21] and to the conjugate gradient in [13,20], see also [10] and the references therein. When the different right-hand sides are known a priori, then block-conjugate gradient methods are efficient. In [8], block-CG is combined with a seed projection method which generates a Krylov subspace for the so-called seed system and projects the residuals of other systems onto this subspace. Similar ideas are developed in [16].

The method studied in this paper has been validated in [11] on academic test problems; we intend here to test it on larger problems.

We discuss the application of the augmented conjugate gradient method in an iterative process applied to the solution of acoustic problems. They concern the scattering of harmonic planar waves by purely reflecting bodies. We propose an *Exact Controllability* approach; a least-squares formulation is introduced and solved by a preconditioned conjugate gradient algorithm (see [4,5]); so at each iteration of this outer conjugate gradient, we have to solve a linear system for the preconditioner. For 2D applications, these linear systems are solved by a direct method (Cholesky), the matrix being factorized once for all. For the 3D applications discussed here, the matrices are too large to allow a direct method, so an iterative method is necessary and we have used a conjugate gradient. In order to increase the efficiency of this conjugate gradient, we apply the augmented conjugate gradient method; for each of these linear systems, the matrix is the same and the right-hand sides depend on the previous solutions. Conjugate gradient is used at two levels, the inner is an augmented one.

The format of this paper is as follows. In section 2, we describe the augmented conjugate gradient. In section 3, we present the scattering problem and the solution algorithm; in section 4, we give some details on the numerical implementation. The last section is devoted to numerical results with emphasis on the efficiency of the augmented conjugate gradient to solve the linear systems associated to the preconditioner.

2. Augmented conjugate gradient algorithm

We present here the algorithm for two successive linear systems to be solved; let A be a symmetric positive definite matrix of $\mathbb{R}^{N,N}$, we consider the systems

$$Ay = c, \tag{2.1}$$

$$Ax = b. \tag{2.2}$$

We assume that a classical Conjugate Gradient algorithm (CG) is used to solve the first system (2.1). In what follows, $(\cdot,\cdot)$ denotes the Euclidean scalar product and $|\cdot|$ the associate norm; to set the notations, we write CG as follows:

Initialization

$$y_0 \textit{ given}, \tag{2.3}$$

compute

$$s_0 = c - Ay_0, \tag{2.4}$$

set

$$w_0 = s_0. \tag{2.5}$$

Iterations
For $j \geqslant 0$, y_j, s_j, w_j *being known, compute* y_{j+1}, s_{j+1}, w_{j+1} *as follows*:

$$\gamma_j = \frac{|s_j|^2}{(w_j, Aw_j)}, \tag{2.6}$$

$$y_{j+1} = y_j + \gamma_j w_j, \tag{2.7}$$

$$s_{j+1} = s_j - \gamma_j A w_j. \tag{2.8}$$

If

$$\frac{|s_{j+1}|}{|c|} \leqslant \varepsilon, \tag{2.9}$$

take $m = j + 1$, $y = y_{j+1}$; *else, compute*

$$\delta_{j+1} = \frac{|s_{j+1}|^2}{|s_j|^2}, \tag{2.10}$$

$$w_{j+1} = s_{j+1} + \delta_{j+1} w_j. \tag{2.11}$$

Do $j + 1 \to j$ *and return to* (2.6).

Let

$$S = (s_0, s_1, \ldots, s_m), \tag{2.12}$$

$$W = (w_0, w_1, \ldots, w_m) \tag{2.13}$$

be the set of residuals and descent directions generated. Recall that

$$S^*S = \Delta, \qquad W^*AW = D, \qquad \mathrm{span}(S) = \mathrm{span}(W) = K_m(A, s_0), \tag{2.14}$$

where Δ and D are diagonal matrices and $K_m(A, s_0)$ is the Krylov subspace of dimension $m + 1$ generated by the initial residual s_0.

We now want to use this information to speed up the solution of the second system (2.2), so we use the following Augmented Conjugate Gradient (AugCG) proposed in [11]:

Initialization

$$x_0 \textit{ given}, \tag{2.15}$$

compute

$$r_0 = b - Ax_0. \tag{2.16}$$

For $j = 1, \ldots, m$, *do*

$$x_0 = x_0 + \frac{(r_0, w_j)}{(w_j, Aw_j)} w_j, \tag{2.17}$$

$$r_0 = r_0 - \frac{(r_0, w_j)}{(w_j, Aw_j)} Aw_j. \tag{2.18}$$

Set

$$z_0 = r_0, \tag{2.19}$$

for $j = 1, \dots, m$, *do*

$$z_0 = z_0 - \frac{(z_0, Aw_j)}{(w_j, Aw_j)} w_j, \tag{2.20}$$

and set

$$p_0 = z_0. \tag{2.21}$$

Iterations
For $k \geqslant 0$, x_k, r_k, z_k, p_k *being known, compute* x_{k+1}, r_{k+1}, z_{k+1}, p_{k+1} *as follows*:

$$\alpha_k = \frac{(r_k, z_k)}{(p_k, Ap_k)}, \tag{2.22}$$

$$x_{k+1} = x_k + \alpha_k p_k, \tag{2.23}$$

$$r_{k+1} = r_k - \alpha_k Ap_k, \tag{2.24}$$

$$\mu_{k+1} = \frac{(r_{k+1}, Aw_m)}{(w_m, Aw_m)}, \tag{2.25}$$

$$z_{k+1} = r_{k+1} - \mu_{k+1} w_m. \tag{2.26}$$

If

$$\frac{|(r_{k+1}, z_{k+1})|^{1/2}}{|b|} \leqslant \varepsilon \tag{2.27}$$

take $x = x_{k+1}$; *else, compute*

$$\beta_{k+1} = \frac{(r_{k+1}, z_{k+1})}{(r_k, z_k)}, \tag{2.28}$$

$$p_{k+1} = z_{k+1} + \beta_{k+1} p_k. \tag{2.29}$$

Do $k + 1 \to k$ *and return to* (2.22).

This algorithm computes an initial guess x_0 such that the initial residual r_0 is orthogonal to the Krylov subspace $K_m(A, s_0)$ and an initial descent direction p_0 conjugate to the descent directions W; then the new descent direction p_{k+1} is enforced to be A-orthogonal to the last vector w_m of the initial system and to the previous descent direction p_k.

Remark 2.1. We do not store w_j and Aw_j, but directly $w_j/\sqrt{(w_j, Aw_j)}$ and $Aw_j/\sqrt{(w_j, Aw_j)}$. Generally to avoid a too important memory size increase, the vectors w_j, Aw_j, $j = 1, \dots, m$, are stored in a secondary memory. Then to compute the initial terms x_0, r_0, z_0, we have to read twice the vectors stored in the secondary memory, and only the last ones (w_m, Aw_m) are kept in main memory for the iterations.

Remark 2.2. As proposed in [11], we have used this form (2.15)–(2.29) of the Augmented Conjugate Gradient algorithm, using Modified Gram–Schmidt processes to ensure stability.

We refer to [11] for a study of the properties of AugCG and particularly for the demonstration that the asymptotic rate of convergence of AugCG is better than the classical CG one.

3. Application to 3D acoustic scattering problems

In this section, we shall discuss an application, in an iterative process, of the methodology introduced in the above section. It concerns the scattering of monochromatic incident waves by purely reflecting bodies.

3.1. Formulation of the scattering problems

Let us consider a scattering body B of boundary $\gamma = \partial B$, illuminated by an harmonic incident wave. The scattered field satisfies

$$u_{tt} - \Delta u = 0 \quad \text{in } \left(\mathbb{R}^3 \setminus \overline{B}\right) \times (0,T), \tag{3.1}$$

$$u = g \quad \text{on } \sigma = \gamma \times (0,T). \tag{3.2}$$

The unknown u represents the pressure. The boundary condition (3.2) can be replaced by a Neumann one.

We have bounded $\mathbb{R}^3 \setminus \overline{B}$ by an artificial boundary Γ and we denote by Ω the region of $\mathbb{R}^3$ between γ and Γ (see figure 1) and by Q the domain $\Omega \times (0,T)$.

We prescribe on Γ an *approximate Sommerfeld condition* such that

$$\frac{\partial u}{\partial n} + \frac{\partial u}{\partial t} = 0 \quad \text{on } \Sigma = \Gamma \times (0,T); \tag{3.3}$$

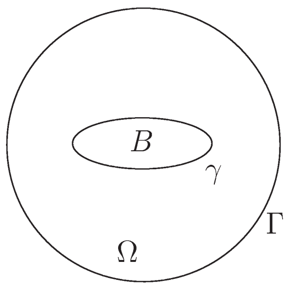

Figure 1.

the *first order* absorbing condition (3.3) can be replaced by a *second order* one (see, e.g., [5]) leading to better accuracy and efficiency.

Suppose that the harmonic wave hitting B is of period T; we have then $g(x,t) = -\mathrm{Re}(\mathrm{e}^{-\mathrm{i}kt}G(x))$ with $k = 2\pi/T$ and after some transition we shall reach a time-periodic regime of period T. From a computational point of view we have two classical possibilities.

With the *first one* we use a separation of variable method based on

$$u(x,t) = \mathrm{Re}\big(\mathrm{e}^{-\mathrm{i}kt}U(x)\big)$$

which leads to the following Helmholtz problem

$$\begin{cases} \Delta U + k^2 U = 0 & \text{in } \Omega, \\ U = G & \text{on } \gamma, \\ \dfrac{\partial U}{\partial n} = \mathrm{i}kU & \text{on } \Gamma; \end{cases} \tag{3.4}$$

we shall not pursue that direction; for classical Helmholtz equation solvers, see, e.g., [17] and the references therein.

The second approach consists in time-integrating the original problem (3.1)–(3.3) starting from initial conditions on u and u_t until a *T*-periodic regime has been reached. We shall follow this second approach; however, since the asymptotic convergence as $t \to +\infty$ can be slow, particularly for *nonconvex* reflectors, the idea is to speed it up using *control methods*.

The control methods will be applied to system (3.1)–(3.3) completed by the following *T*-periodicity conditions:

$$u(0) = u(T), \qquad u_t(0) = u_t(T). \tag{3.5}$$

3.2. *Exact controllability and least-squares formulations*

Solving problem (3.1)–(3.3), (3.5) is *equivalent* to finding a pair $\{e_0, e_1\}$ such that

$$\begin{cases} u(0) = e_0, & u_t(0) = e_1, \\ u(T) = e_0, & u_t(T) = e_1, \end{cases} \tag{3.6}$$

with u the solution of (3.1)–(3.3).

Problem (3.1)–(3.3), (3.6) is an *exact controllability problem* which can be solved by methods directly inspired by Lions' *Hilbert Uniqueness Method* (HUM) (see, e.g., [14,18,19]).

To apply these methods, the right choice for the space E containing $e = \{e_0, e_1\}$ is fundamental; an appropriate choice is

$$E = V_g \times L^2(\Omega), \tag{3.7}$$

with $V_g = \{\varphi \mid \varphi \in H^1(\Omega),\ \varphi \mid \gamma = g(0)\}$.

A *least-squares* formulation of (3.1)–(3.3), (3.6) is then given by

$$\min_{v\in E} J(v), \tag{3.8}$$

with

$$J(v)=\frac{1}{2}\int_\Omega\left[\left|\nabla\big(y(T)-v_0\big)\right|^2+\left|y_t(T)-v_1\right|^2\right]\mathrm{d}x,\quad \forall v=\{v_0,v_1\}, \tag{3.9}$$

where, in (3.9), the function y is *the* solution of

$$y_{tt}-\Delta y=0\quad\text{in } Q, \tag{3.10}$$

$$y=g\quad\text{on } \sigma, \tag{3.11}$$

$$\frac{\partial y}{\partial n}+\frac{\partial y}{\partial t}=0\quad\text{on } \Sigma, \tag{3.12}$$

$$y(0)=v_0,\qquad y_t(0)=v_1. \tag{3.13}$$

Remark 3.1. The choice of J as cost function is fairly natural once we realize that the *energy* $E(t)$ associated to the wave equation (3.10) is precisely

$$E(t)=\frac{1}{2}\int_\Omega\left[\left|\nabla y(t)\right|^2+\left|y_t(t)\right|^2\right]\mathrm{d}x.$$

Remark 3.2. It has been shown by Bardos and Rauch [1] that functional J may fail to be strongly "elliptic" (coercive) in some situations with trapping rays and they have proposed an alternative functional with better coercivity properties. However, the functional J has given satisfactory computational results as shown in [3–5] and its implementation is simpler, so we have here used the functional J. For comparisons of the two approaches, see [6].

Problem (3.8) can be solved by a preconditioned conjugate gradient algorithm, so we need the gradient $J'(v)$, $\forall v\in E$. We refer to [3,4] for the details of the computation of $J'(v)$, we give here only the result:

$$\begin{aligned}\big\langle J'(v),w\big\rangle={}&\int_\Omega\nabla\big(v_0-y(T)\big)\cdot\nabla w_0\,\mathrm{d}x+\int_\Gamma p(0)w_0\,\mathrm{d}\Gamma\\&-\int_\Omega p_t(0)w_0\,\mathrm{d}x+\int_\Omega\big(v_1-y_t(T)\big)w_1\,\mathrm{d}x+\int_\Omega p(0)w_1\,\mathrm{d}x,\\&\forall w=\{w_0,w_1\}\in E_0,\end{aligned} \tag{3.14}$$

with the adjoint state p defined by

$$p_{tt}-\Delta p=0\quad\text{in } Q, \tag{3.15}$$

$$p=0\quad\text{on } \sigma, \tag{3.16}$$

$$\frac{\partial p}{\partial n} - \frac{\partial p}{\partial t} = 0 \quad \text{on } \Sigma, \tag{3.17}$$

and the final conditions

$$p(T) = y_t(T) - v_1, \tag{3.18}$$

$$\int_\Omega p_t(T) z \, \mathrm{d}x = \int_\Gamma p(T) z \, \mathrm{d}\Gamma - \int_\Omega \nabla\big(y(T) - v_0\big) \cdot \nabla z \, \mathrm{d}x, \quad \forall z \in V_0. \tag{3.19}$$

In (3.14), $\langle \cdot, \cdot \rangle$ denotes the duality pairing between E_0' and E_0, where

$$E_0 = V_0 \times L^2(\Omega) \quad \text{with } V_0 = \big\{\varphi \mid \varphi \in H^1(\Omega),\ \varphi = 0 \text{ on } \gamma\big\}. \tag{3.20}$$

3.3. *Conjugate gradient solution of the least-squares problem* (3.8)

A conjugate gradient algorithm for the solution of problem (3.8) is described below:

Step 0: Initialization

$$e^0 = \{e_0^0, e_1^0\} \in E \quad \textit{is given.} \tag{3.21}$$

Solve the following forward wave problem:

$$y_{tt}^0 - \Delta y^0 = 0 \quad \textit{in } Q, \tag{3.22}$$

$$y^0 = g \quad \textit{on } \sigma, \tag{3.23}$$

$$\frac{\partial y^0}{\partial n} + \frac{\partial y^0}{\partial t} = 0 \quad \textit{on } \Sigma, \tag{3.24}$$

$$y^0(0) = e_0^0, \qquad y_t^0(0) = e_1^0. \tag{3.25}$$

Solve the following backward wave problem:

$$p_{tt}^0 - \Delta p^0 = 0 \quad \textit{in } Q, \tag{3.26}$$

$$p^0 = 0 \quad \textit{on } \sigma, \tag{3.27}$$

$$\frac{\partial p^0}{\partial n} - \frac{\partial p^0}{\partial t} = 0 \quad \textit{on } \Sigma, \tag{3.28}$$

with $p^0(T)$ and $p_t^0(T)$ given by

$$p^0(T) = y_t^0(T) - e_1^0, \tag{3.29}$$

$$\int_\Omega p_t^0(T) z \, \mathrm{d}x = \int_\Gamma p^0(T) z \, \mathrm{d}\Gamma - \int_\Omega \nabla\big(y^0(T) - e_0^0\big) \cdot \nabla z \, \mathrm{d}x, \quad \forall z \in V_0, \tag{3.30}$$

respectively.

Define next $g^0 = \{g_0^0, g_1^0\} \in E_0 = V_0 \times L^2(\Omega)$ *by*

$$\int_\Omega \nabla g_0^0 \cdot \nabla z \,\mathrm{d}x = \int_\Omega \nabla\big(e_0^0 - y^0(T)\big) \cdot \nabla z \,\mathrm{d}x - \int_\Omega p_t^0(0) z \,\mathrm{d}x + \int_\Gamma p^0(0) z \,\mathrm{d}\Gamma, \quad \forall z \in V_0, \tag{3.31}$$

$$g_1^0 = p^0(0) + e_1^0 - y_t^0(T), \tag{3.32}$$

and then

$$w^0 = g^0. \tag{3.33}$$

For $k \geqslant 0$, *suppose that* e^k, g^k, w^k *are known*; *we then compute their updates* e^{k+1}, g^{k+1}, w^{k+1} *as follows*:

Step 1: Descent

Solve

$$\bar{y}_{tt}^k - \Delta \bar{y}^k = 0 \quad \text{in } Q, \tag{3.34}$$

$$\bar{y}^k = 0 \quad \text{on } \sigma, \tag{3.35}$$

$$\frac{\partial \bar{y}^k}{\partial n} + \frac{\partial \bar{y}^k}{\partial t} = 0 \quad \text{on } \Sigma, \tag{3.36}$$

$$\bar{y}^k(0) = w_0^k, \qquad \bar{y}_t^k(0) = w_1^k. \tag{3.37}$$

Solve the following backward wave problem

$$\bar{p}_{tt}^k - \Delta \bar{p}^k = 0 \quad \text{in } Q, \tag{3.38}$$

$$\bar{p}^k = 0 \quad \text{on } \sigma, \tag{3.39}$$

$$\frac{\partial \bar{p}^k}{\partial n} - \frac{\partial \bar{p}^k}{\partial t} = 0 \quad \text{on } \Sigma, \tag{3.40}$$

with $\bar{p}^k(T)$ *and* $\bar{p}_t^k(T)$ *given by*

$$\bar{p}^k(T) = \bar{y}_t^k(T) - w_1^k, \tag{3.41}$$

$$\int_\Omega \bar{p}_t^k(T) z \,\mathrm{d}x = \int_\Gamma \bar{p}^k(T) z \,\mathrm{d}\Gamma - \int_\Omega \nabla\big(\bar{y}^k(T) - w_0^k\big) \cdot \nabla z \,\mathrm{d}x, \quad \forall z \in V_0, \tag{3.42}$$

respectively.

Next, define $\bar{g}^k = \{\bar{g}_0^k, \bar{g}_1^k\} \in V_0 \times L^2(\Omega)$ *by*

$$\int_\Omega \nabla \bar{g}_0^k \cdot \nabla z \,\mathrm{d}x = \int_\Omega \nabla\big(w_0^k - \bar{y}^k(T)\big) \cdot \nabla z \,\mathrm{d}x - \int_\Omega \bar{p}_t^k(0) z \,\mathrm{d}x + \int_\Gamma \bar{p}^k(0) z \,\mathrm{d}\Gamma, \quad \forall z \in V_0, \tag{3.43}$$

$$\bar{g}_1^k = \bar{p}^k(0) + w_1^k - \bar{y}_t^k(T), \tag{3.44}$$

and then ρ_k *by*

$$\rho_k = \frac{\int_\Omega [|\nabla g_0^k|^2 + |g_1^k|^2]\,\mathrm{d}x}{\int_\Omega (\nabla \bar{g}_0^k \cdot \nabla w_0^k + \bar{g}_1^k w_1^k)\,\mathrm{d}x}. \tag{3.45}$$

We update e^k *and* g^k *by*

$$e^{k+1} = e^k - \rho_k w^k, \tag{3.46}$$

$$g^{k+1} = g^k - \rho_k \bar{g}^k. \tag{3.47}$$

Step 2: Test of the convergence and construction of the new descent direction
If

$$\frac{\left(\int_\Omega \left(|\nabla g_0^{k+1}|^2 + |g_1^{k+1}|^2\right)\mathrm{d}x\right)^{1/2}}{\left(\int_\Omega \left(|\nabla g_0^0|^2 + |g_1^0|^2\right)\mathrm{d}x\right)^{1/2}} \leqslant \varepsilon_1$$

take $e = e^{k+1}$*; else, compute*

$$\gamma_k = \frac{\int_\Omega \left(|\nabla g_0^{k+1}|^2 + |g_1^{k+1}|^2\right)\mathrm{d}x}{\int_\Omega \left(|\nabla g_0^k|^2 + |g_1^k|^2\right)\mathrm{d}x} \tag{3.48}$$

and update w^k *by*

$$w^{k+1} = g^{k+1} + \gamma_k w^k. \tag{3.49}$$

Do $k = k + 1$ *and go to* (3.34).

Remark 3.3. Algorithm (3.21)–(3.49) looks complicated at first glance. In fact, it is not that complicated to implement since each iteration requires basically the solution of two wave equations such as (3.34)–(3.37) and (3.38)–(3.42) and of an elliptic problem such as (3.43).

4. Numerical implementation

4.1. Space–time discretization

The practical implementation of the above algorithm relies on fairly classical time discretization and finite element approximations. The wave equation is time discretized by a centered second order explicit finite difference scheme combined to piecewise linear finite element approximations for the space variables.

To be more precise, the basic wave problem

$$u_{tt} - \Delta u = 0 \quad \text{in } Q, \tag{4.1}$$

$$u = g \quad \text{on } \sigma, \tag{4.2}$$

$$\frac{\partial u}{\partial n} + \frac{\partial u}{\partial t} = 0 \quad \text{on } \Sigma, \tag{4.3}$$

$$u(0) = u_0, \qquad u_t(0) = u_1, \tag{4.4}$$

will be approximated as follows:

Step 1: We introduce the following *variational formulation* of problem (4.1)–(4.4):

Find u satisfying $u(t) \in H^1(\Omega)$, $\forall t \in [0, T]$, and

$$\int_\Omega u_{tt} z \,\mathrm{d}x + \int_\Omega \nabla u \cdot \nabla z \,\mathrm{d}x + \int_\Gamma \frac{\partial u}{\partial t} z \,\mathrm{d}\Gamma = 0, \quad \forall z \in V_0, \text{ a.e. on } (0, T), \tag{4.5}$$

relation (4.5) being completed by (4.2), (4.4). The space V_0 is still defined by (3.20).
Step 2: We introduce $\Delta t = T/N_t$ (N_t: a positive integer); we then (formally) *time-discretize* (4.2), (4.4), (4.5) by

$$u^0 = u_0, \qquad u^1 - u^{-1} = 2\Delta t u_1, \tag{4.6}$$

and for $n = 0, 1, \ldots, N_t$, we compute u^{n+1} via the solution of

$$\begin{cases} u^{n+1} \in V_{g^{n+1}}, \\ \displaystyle\int_\Omega \frac{u^{n+1} + u^{n-1} - 2u^n}{\Delta t^2} z \,\mathrm{d}x + \int_\Omega \nabla u^n \cdot \nabla z \,\mathrm{d}x + \int_\Gamma \frac{u^{n+1} - u^{n-1}}{2\Delta t} z \,\mathrm{d}\Gamma = 0, \\ \quad \forall z \in V_0, \end{cases} \tag{4.7}$$

with

$$V_{g^{n+1}} = \bigl\{ z \mid z \in H^1(\Omega),\ z = g\bigl((n+1)\Delta t\bigr) \text{ on } \gamma \bigr\}. \tag{4.8}$$

Scheme (4.6)–(4.7) is (formally) *second order accurate* with respect to Δt.
Step 3: We suppose – for simplicity – that Ω is a *bounded polygonal* domain of $\mathbb{R}^3$. With $\mathcal{T}_h$ a classical *finite element tetrahedrization* of Ω, we approximate $H^1(\Omega)$ and $L^2(\Omega)$ by

$$V_h = \bigl\{ z \mid z \in C^0(\overline{\Omega}), z|_T \in P_1,\ \forall T \in \mathcal{T}_h \bigr\}, \tag{4.9}$$

where P_1 is the space of polynomials in three variables of degree $\leqslant 1$, and where, as usual, h is the length of the largest edge(s) of $\mathcal{T}_h$. With obvious notation, we approximate (4.1)–(4.4) by

$$u_h^0 = u_{0h}, \qquad u_h^1 - u_h^{-1} = 2\Delta t u_{1h}. \tag{4.10}$$

and, for $n = 0, 1, \ldots, N_t$, by

$$\begin{cases} u_h^{n+1} \in V_{g_h^{n+1}}, \\ \displaystyle\int_\Omega \frac{u_h^{n+1} + u_h^{n-1} - 2u_h^n}{\Delta t^2} z \,\mathrm{d}x + \int_\Omega \nabla u_h^n \cdot \nabla z \,\mathrm{d}x + \int_\Gamma \frac{u_h^{n+1} - u_h^{n-1}}{2\Delta t} z \,\mathrm{d}\Gamma = 0, \\ \quad \forall z \in V_{0h}, \end{cases} \tag{4.11}$$

where, in (4.10), u_{0h} and u_{1h} are approximations of u_0 and u_1 belonging to V_h and where

$$V_{0h} = \{z \mid z \in V_h,\ z = 0 \text{ on } \gamma\} \ (= V_0 \cap V_h), \tag{4.12}$$

$$V_{g_h^{n+1}} = \{z \mid z \in V_h,\ z = g_h^{n+1} \text{ on } \gamma\}, \tag{4.13}$$

g_h^{n+1} being an approximation of $g((n+1)\Delta t)$ belonging to the boundary space span by the traces on γ of the functions of V_h.

Concerning the computation of u_h^{n+1} in (4.11), we apply mass lumping to obtain a fully explicit scheme. So, if a parallel computer is used, this wave equation solver can be easily parallelized as shown in [2] for 2D-cases.

We define, for the discrete problem, a least-squares formulation equivalent to (3.8)–(3.13) and a conjugate gradient algorithm analogous to (3.21)–(3.49).

After the discretization steps, the linear problems (3.31) and (3.43) are of the form

$$Ag_0^0 = b^0, \tag{4.14}$$
$$A\bar{g}_0^k = b^k, \quad k = 1, \dots, \tag{4.15}$$

with A a symmetric, positive definite matrix (A is the matrix of the Laplace operator).

In fact we do not solve directly the systems (4.14), (4.15), but we first apply a diagonal preconditioning; if we define the diagonal matrix D by

$$D_i = \sqrt{A_{ii}}, \quad i = 1, \dots, N, \tag{4.16}$$

with N the dimension of A, we replace each system such as

$$Ag = b \tag{4.17}$$

by

$$\begin{cases} D^{-1}AD^{-1}y = D^{-1}b, \\ g = D^{-1}y. \end{cases} \tag{4.18}$$

The algorithm defined in section 2 is applied to the solution of the elliptic problems of the form (4.14), (4.15) corresponding to (3.31) and (3.43).

4.2. Matrix storage

In the algorithm (3.21)–(3.49), the more time consuming steps are the integration of the wave equations (3.34)–(3.37) and (3.38)–(3.42) which after discretization are of the form (4.10), (4.11), and the solution of problem (3.43). In (4.11), the more expensive part is the computation of the matrix–vector product corresponding to the evaluation of the term $\int_\Omega \nabla u_h^n \cdot \nabla z \, dx$; the matrix is the same as the one of the discretized problem corresponding to (3.43). As an iterative process is used for the solution of this last problem, the main cost is also related to the matrix–vector product.

So, for the efficiency of the global algorithm, the matrix–vector product has to be as optimized as possible. The matrix being sparse and symmetric, only the non-zero terms of the lower triangular matrix are stored. The computations are done on a CRAY vector machine. If we consider row storage, the vectors are short and the vectorization not very efficient. That is why we have considered a diagonal storage (see [9,12]) for the non-zero terms, the length of the vectors is then the number of non-zero terms of the diagonals. To optimize this length, a renumbering algorithm minimizing the matrix bandwidth is previously applied.

5. Numerical results

We consider different test problems and, in this paper, we emphasize the results concerning the efficiency of the augmented conjugate gradient (AugCG) defined in section 2 and applied to the solution of (3.31), (3.43); for more details concerning the scattering results, we refer to [3–6]. The methodology is applied on three different geometries. The first two test cases concern spheres; these geometries have been chosen to validate the code since exact solutions can be computed. The third example concerns a nonconvex obstacle, this kind of obstacle being the aim of the control approach considered in this paper.

5.1. Scattering by spheres

The first obstacle is a sphere of diameter $\lambda/2$, the artificial boundary is located at a distance λ to ∂B. The mesh has 34,518 nodes and 198,264 tetrahedra. The field scattered by this sphere can be computed just integrating in time the wave equation, but we apply the control algorithm as a test. The linear problem (3.31), preconditioned by diagonal (see (4.18)), is solved by the conjugate gradient (CG) (2.3)–(2.11) (also preconditioned by diagonal) and then the problems (3.43) are solved by the augmented conjugate gradient (AugCG) (2.15)–(2.29).

With $\varepsilon = 10^{-3}$ in the stopping test (2.9), we obtain $m = 85$. In figure 2, we compare, for the first problem (3.43) ($k = 1$), the convergence history obtained either with the standard conjugate gradient (dotted line), or with the augmented conjugate gradient (continuous line). In AugCG, we also use $\varepsilon = 10^{-3}$ in the stopping test (2.27). Then, for the problems (3.43) associated to 10 iterations of the global algorithm (3.21)–(3.49), we compare, in table 1, the number of iterations necessary to reach convergence either with CG or with AugCG using the 85 $\{w_j, Aw_j\}$ stored vectors. During these 10 iterations of the global algorithm, the number of matrix–vector products devoted to the solution of (3.43) is reduced from 766 to 303 by using AugCG rather than CG, while the CPU time is reduced from 6.4 s to 3.6 s, the computations being done on one processor of a Cray C90. During the 10 iterations, the cost function J defined by (3.9) is reduced from 3.10^{-3} to 5.10^{-6}.

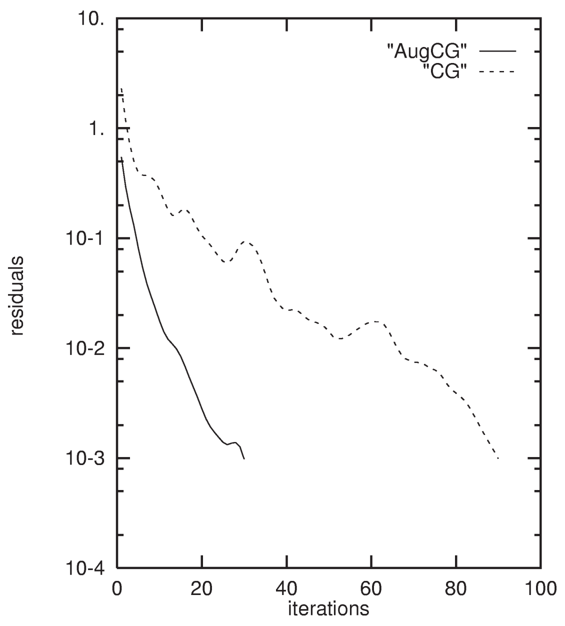

Figure 2. Convergence histories ($N = 34518$).

Table 1
Comparison of the number of iterations ($N = 34518$).

k	1	2	3	4	5	6	7	8	9	10
CG	90	85	56	71	85	84	73	85	66	61
AugCG	30	26	26	29	31	30	32	31	28	30

The second sphere considered is of diameter 2λ, the mesh has 87,512 nodes and 502,005 tetrahedra. With the angles defined as in figure 3, the incident wave is defined by $\varphi = 270°$, the right-hand side g in (3.2) is then defined by

$$g(x,t) = -\mathrm{Re}\big[\mathrm{e}^{-\mathrm{i}kt}\mathrm{e}^{\mathrm{i}KX}\big] \tag{5.1}$$

with

$$K = k\begin{pmatrix}\cos\theta\cos\varphi\\ \sin\theta\cos\varphi\\ \sin\theta\end{pmatrix}, \qquad X = \begin{pmatrix}x\\ y\\ z\end{pmatrix} \quad \text{and} \quad k = 2\pi/\lambda.$$

The exact scattered field with a cross-section by the plane ($x = 0$) is plotted in figure 4 and the computed one in figure 5. This solution is reached after 10 control iterations. The solutions are in good agreement with a discrepancy at the bottom of the plottings due to the radiation condition (3.3) which is only first order.

For the solution of (3.31), with the same stopping test as for the previous example, we obtain $m = 100$. In figure 6, we compare also, for the first problem (3.43)

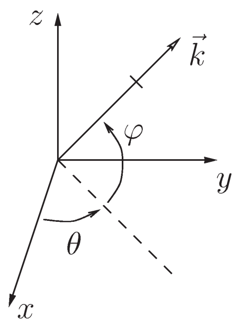

Figure 3. Definition of the angles.

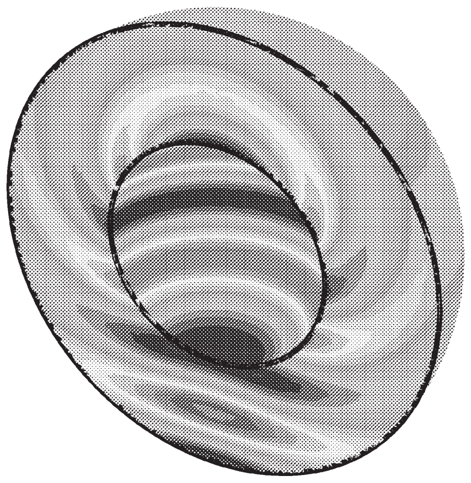

Figure 4. Exact solution.

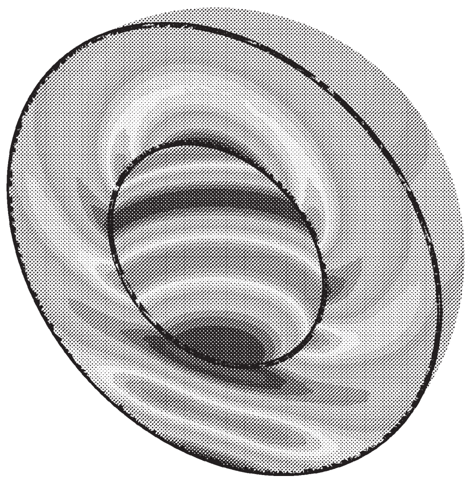

Figure 5. Computed solution.

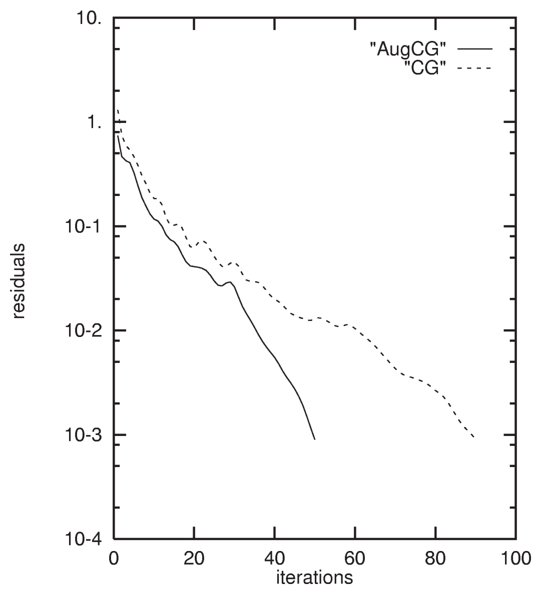

Figure 6. Convergence histories ($N = 87512$).

Table 2
Comparison of the number of iterations ($N = 87512$).

k	1	2	3	4	5	6	7	8	9	10
CG	90	92	92	70	86	88	90	88	84	103
AugCG	50	50	50	54	73	54	54	44	49	54

($k = 1$), the convergence history obtained either with the standard conjugate gradient (dotted line), or with the augmented conjugate gradient (continuous line). Then, for the solution of (3.43) with $k = 1, \dots, 10$, we give, in table 2, the number of iterations necessary to reach convergence either with CG or with AugCG. The number of matrix–vector products devoted to the solution of (3.43) for $k = 1, \dots, 10$ is reduced from 883 to 532 and the CPU time for this preconditioning step is reduced from 18.1 s to 16.2 s. The CPU time is reduced to 14.2 s if the vectors $\{w_j, Aw_j\}$ are kept in main memory but the memory requirement for the code then increases from 6 Mw to 24 Mw.

The last test case concerns a semi-open cylindrical cavity; the internal diameter is 0.5λ and the length of the cavity is λ, the wall thickness is 0.05λ. The mesh has about 230,000 nodes and 1,330,000 tetrahedra. Figure 7 shows the trace of the mesh on the cavity. With the angles defined as in figure 3, the incident wave is defined by $\theta = 150°$, $\varphi = 0°$. The convergence of the global algorithm (3.21)–(3.49) is reached after 20 iterations. In figure 8 the contours of the scattered field are shown in the cross section by the plane ($z = 0$).

In figure 9, we compare, for the first problem (3.43) ($k = 1$), the convergence history obtained either with the standard conjugate gradient (dotted line), or with the

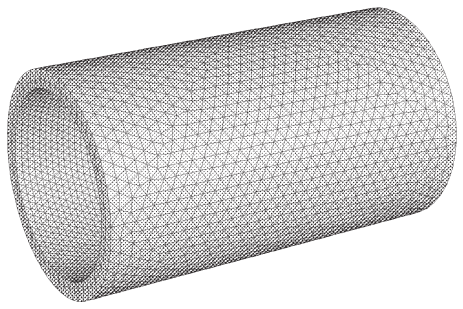

Figure 7. Trace of the mesh on the boundary of the cylindrical semi-open cavity.

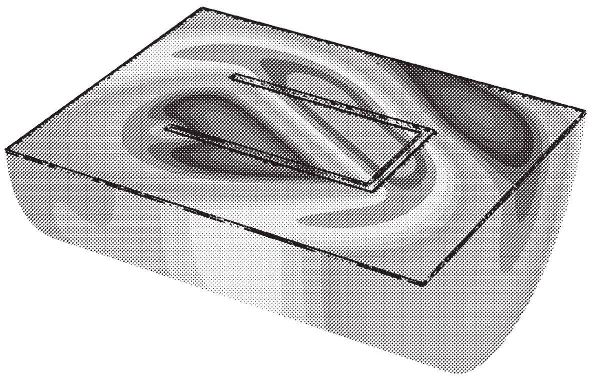

Figure 8. Cylindrical cavity: contours of the scattered field in a cross-section.

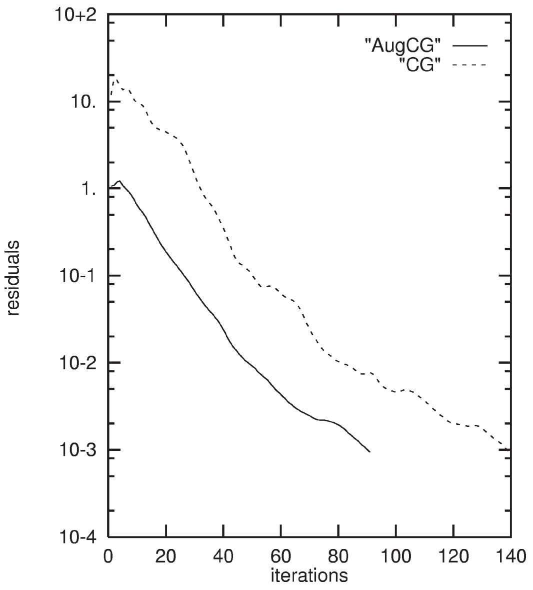

Figure 9. Convergence histories ($N = 231668$).

Table 3
Comparison of the number of iterations ($N = 231668$).

k	1	2	3	4	5	6	7	8	9	10
CG	139	133	111	129	138	131	116	121	114	116
AugCG	91	110	95	106	115	99	97	95	79	79

Table 4
Efficiency of AugCG versus the test problem.

N	34518	87512	231688
R	2.5	1.6	1.3

augmented conjugate gradient (continuous line). Table 3 allows to compare, as in the previous cases, the efficiency of CG and AugCG for the 10 first solutions of (3.43). The number of matrix–vector products is then reduced from 1248 (CG) to 966 (AugCG), but if we compare the CPU time, it is reduced, when using secondary memory saving, only from 63.8 s to 61.8 s, and to 56.4 s with main memory storage (memory requirement increasing from 16 Mw to 67 Mw).

We can summarize in table 4 the efficiency of AugCG versus the test problem. We have computed the average ratio R of the number of iterations with CG by the number of iterations with AugCG. We notice the decrease of the efficiency of the method with the size of the system.

Of course, we have noticed on examples not presented here that, for the same continuous operator, the efficiency of the preconditioning introduced by AugCG is related not only to the number of unknowns but also to the difficulty of the global problem, implying that the right-hand sides in (3.43) are more or less different.

Remark 5.1. For the initial guess, (2.15), we can choose either $x_0 = 0$ or the solution of the previous iteration. The two possibilities give rather equivalent results, one being better than the other depending on the cases. The results presented here correspond to

$$x_0^k = x^{k-1}, \quad k = 1, \ldots,$$

for the first and third examples, and to

$$x_0^k = 0, \quad k = 1, \ldots,$$

for the second example.

Remark 5.2. The m vectors $\{w_j, Aw_j\}$ stored are the ones of the first iteration and they are kept for the preconditioning of all the following iterations; we have tried to compute new ones after for instance 10 iterations but without convergence improvement of the solution of the following systems.

Another possibility is to store the vectors $\{w_j, Aw_j\}$ of the two or three first iterations; this approach has not proved efficient because the overhead due to the fact that the stored vectors are more numerous is not balanced by the decrease of the number of iterations which is very weak.

6. Concluding remarks

We have developed in this paper a real-life application of the Augmented Conjugate Gradient proposed in [11]; it has been tested on large systems up to more than 200,000 unknowns. One of the interesting points of this algorithm lies in its simplicity to implement and its small overhead; the algorithm AugCG does not induce new matrix–vector products, it is based mainly on dot-products which are easily vectorized. If we use secondary memory, the main part of the overhead is due to the reading of vectors for the computation of the initial guess and initial descent direction. If we avoid this overhead, then the memory requirement increase is important as shown in section 5. So depending on the machine to which the code is devoted, one can choose one of the two approaches.

Some improvements of the method are under development to keep its efficiency independent of the size of the system: choice of the vectors to be stored, coupling with domain decomposition. One idea to reduce the memory overhead is to augment the current Krylov subspace with some approximation to an invariant subspace associated with a few of the lowest eigenvalues, see, e.g., [7,22]. The advantage of domain decomposition methods is dealing with the interface problem. Therefore it is possible to store many vectors with a low memory requirement and a very good efficiency (see [15]).

Acknowledgements

The authors would like to thank R. Glowinski, F. Guyomarc'h and J. Périaux for their collaboration to different parts of this study.

References

[1] C. Bardos and J. Rauch, Variational algorithms for the Helmholtz equation using time evolution and artificial boundaries, Asymptotic Anal. 9 (1994) 101–117.

[2] M.O. Bristeau, J. Erhel, P. Feat, R. Glowinski and J. Périaux, Solving the Helmholtz equation at high-wave numbers on a parallel computer with a shared virtual memory, Internat. J. Supercomput. Appl. 9(1) (1995) 18–28.

[3] M.O. Bristeau, R. Glowinski and J. Périaux, Scattering waves using exact controllability methods, in: *Proc. of 31st AIAA Aerospace Sciences Meeting,* Reno, NV, AIAA Paper 93-0460.

[4] M.O. Bristeau, R. Glowinski and J. Périaux, Using exact controllability to solve the Helmholtz equation at high-wave numbers, in: *Mathematical and Numerical Aspects of Wave Propagation,* eds. R. Kleinman, T. Angell, D. Colton, F. Santosa and I. Strakgold (SIAM, Philadelphia, PA, 1993) pp. 113–127.

[5] M.O. Bristeau, R. Glowinski and J. Périaux, Exact controllability to solve the Helmholtz equation with absorbing boundary conditions, in: *Finite Element Methods,* eds. D. Krizek, P. Neittaanmaki and R. Stenberg (Marcel Dekker, New York, 1994) pp. 79–93.

[6] M.O. Bristeau, R. Glowinski and J. Périaux, Waves scattering using exact controllability, in: *Numerical Methods in Engineering 96*, eds. J.A. Desideri, P. Le Tallec, E. Onate, J. Périaux and E. Stein (Wiley, New York, 1996) pp. 96–103.

[7] K. Burrage and J. Erhel, On the performance of various adaptive preconditioned GMRES, Numer. Linear Algebra Appl. 5(1) (1998).

[8] T.F. Chan and W. Wan, Analysis of projection methods for solving linear systems with multiple right-hand sides, SIAM J. Sci. Comput. 18 (1997).

[9] J. Erhel, Sparse matrix multiplication on vector computers, Internat. J. High Speed Comput. 2(2) (1990) 101–116.

[10] J. Erhel, About Newton–Krylov methods, in: *Computational Science for the 21st Century*, eds. M.O. Bristeau, G. Etgen, W. Fitzgibbon, J.L. Lions, J. Périaux and M.F. Wheeler (Wiley, Chichester, UK, 1997) pp. 53–61.

[11] J. Erhel and F. Guyomarc'h, An augmented subspace conjugate gradient, INRIA Research Report No. 3278 (October 1997).

[12] J. Erhel and M. Vidrascu, Adapting algorithms and data structures to supercomputers in finite element calculations, in: *Calcul des Structures et Intelligence Artificielle,* Vol. 3, eds. J.M. Fouet, P. Ladevèze and R. Ohayon (Pluralis, 1989) pp. 403–417.

[13] C. Farhat, L. Crivelli and F.X. Roux, Extending substructure based iterative solvers to multiple load and repeated analyses, Comput. Methods Appl. Mech. Engrg. 117 (1994) 195–209.

[14] R. Glowinski and J.L. Lions, Exact and approximate controllability for distributed parameter systems, Part II, Acta Numerica (1996) 159–333.

[15] R. Glowinski, J. Périaux, Z.C. Shi and O. Widlund, eds., *Domain Decomposition Methods in Sciences and Engineering* (Wiley, Chichester, UK, 1995).

[16] P. Joly, Résolution de systèmes linéaires avec plusieurs seconds membres par la méthode du gradient conjugué, Technical Report R-91012, Laboratoire d'Analyse Numérique, Paris VI (Mars 1991).

[17] Y. Kuznetsov and K. Lipnikov, On the application of fictitious domain and domain decomposition methods for scattering problems on Cray Y-MP C98, Technical Report 9557, Department of Mathematics, University of Nijmegen, Nijmegen, The Netherlands (1995).

[18] J.L. Lions, Exact controllability, stabilization and perturbation for distributed systems, SIAM Rev. 30 (1988) 1–68.

[19] J.L. Lions, *Contrôlabilité Exacte, Perturbation et Stabilisation des Systèmes Distribués*, Vols. 1 et 2 (Masson, Paris, 1996).

[20] C. Rey, Développement d'algorithmes parallèles de résolution en calcul non-linéaire de structures hétérogènes: Application au cas de la butée acier-élastomère, Thèse de doctorat, Ecole Normale Supérieure de Cachan (1994).

[21] Y. Saad, On the Lanczos method for solving symmetric linear systems with several right-hand sides, Math. Comp. 178 (1987) 651–662.

[22] Y. Saad, Analysis of augmented Krylov subspace techniques, SIMAX 18(2) (1997) 435–449.