Practical aspects and experiences

An element-by-element preconditioned conjugate gradient method implemented on a vector computer

Jocelyne Erhel a, Alice Traynard b and Marina Vidrascu b

a IRISA campus universitaire de Beaulieu, F-35042 Rennes Cedex, France
b INRIA Rocquencourt B.P. 105, F-78153 Le Chesnay Cedex, France

Received 10 November 1990
Revised 6 December 1990

Abstract


We consider the linear equation $Ax = b$ where $A$ is a sparse symmetric positive definite matrix arising from a finite element discretisation. We use the preconditioned conjugate gradient method to solve this equation, introducing an element-by-element preconditioner which is based on a Crout's decomposition of the element matrices and an element-by-element product of them. When the mesh is coloured, this preconditioner is largely vectorizable. We implement this method on a CRAY-2, and test it on 2D and 3D elastic and thermal problems and compare it to other classical preconditioners.

Keywords. Sparse matrix; iterative method; conjugate gradient; finite element; colouring; vectorization.

1. Introduction

In this work we are interested in solving a system of linear equations of the type:

$$Ax = b,$$  \hspace{1cm} (1)

where $A$ is a sparse matrix of order $n$.

In fact, many calculations require such a resolution which is, generally, a major and costly step. In particular, applying the finite element method to structural mechanics leads to an equation of type (1), where in this case $A$ is a symmetric, positive-definite matrix.

Direct methods based on the factorization of $A$ are the most rapid but are too memory-consuming, especially if the bandwidth of $A$ is large, as is the case for 3-dimensional problems. Therefore alternatives must be found using iterative methods. In 1952, Hestenes and Stiefel [5] presented the conjugate gradient method which may indeed be applied to symmetric positive-definite matrices. This method converges towards the exact solution $\bar{x}$ of (1) in a maximum of
n iterations. Moreover it has the particularity that at each iteration the approximated solution can be obtained without explicitly computing the matrix $A$: it is only necessary to calculate the $Av$ product of the matrix vector for a given vector $v$. This fully exploits the sparse character of the matrix. The drawback of this method is the slower convergence, which is even slower if the matrix is ill-conditioned (cf. Golub’s error estimation [3]). In the worst case, because of rounding errors (particularly in scalar products), the iterations can diverge. This may be partially remedied by using a preconditioner.

A matrix $B$ is chosen which is “close” to $A$, “easy to invert”, and (2) is solved instead of (1):

$$B^{-1}Ax = B^{-1}b.$$  

To accelerate the convergence, it is necessary to determine $B$ such that the conditioning of $B^{-1}A$ is in the neighbourhood of 1. In 1983, Hughes, Levit and Winget proposed an algorithm for element by element preconditioning [7–9]. This preconditioner has several advantages:

(i) The calculation of the inverse of the preconditioner $B$ is largely vectorizable.

(ii) The method does not require the explicit calculation of the matrix $B$, and the cost of storage only increase with the size of the problem.

The aim of this study is to evaluate the performances of this preconditioner, in particular the speed of convergence and the rate of execution, and to compare them with those of other preconditioners. We shall firstly describe the element-by-element (EBE) preconditioned conjugate gradient algorithm. Then we shall examine the results obtained on 2- and 3-dimensional thermal and elasticity problems.

2. The preconditioned conjugate gradient algorithm

We solve

$$Ax = b,$$

where $A$ is a symmetric positive-definite matrix, and $b$ is a given vector. $B$ is the preconditioner.

Step 1: initialization

$k = 0$
$r_0 = b - Ax_0$
$d_0 = B^{-1}r_0$
$p_0 = d_0$

Step 2: iteration on $k$

$$a_k = \frac{r_k d_k}{p_k A p_k}$$
$$x_{k+1} = x_k + a_k p_k$$
$$r_{k+1} = r_k - a_k A p_k$$

Step 3: check convergence

$$\|r_{k+1}\| \leq \varepsilon \|r_0\|$$

if

return
Step 4: compute the new conjugate direction
\[ \begin{align*}
    d_{k+1} &= B^{-1}r_{k+1} \\
    \beta_{k+1} &= \frac{r_{k+1}^T d_{k+1}}{r_k^T d_k} \\
    p_{k+1} &= d_{k+1} + \beta_{k+1} p_k \\
    k &= k + 1
\end{align*} \]

3. Constructing the EBE preconditioner

Basic idea [10]: Let
\[ \begin{align*}
    A &= I + \varepsilon \sum_{i=1}^{n} A_i, \\
    B &= \prod_{i=1}^{n} (I + \varepsilon A_i),
\end{align*} \]
then
\[ B - A = O(\varepsilon^2), \]
\varepsilon "small".
In our case, the Finite Element Method is used, therefore
\[ A = \sum_{e=1}^{N_e} A^e, \]
where \( N_e \) is the number of elements in the mesh, and \( A^e \) is the element matrix associated to the element \( e \) of the mesh [6].

Let \( W \) and \( W^e \) be the diagonal parts of \( A \) and \( A^e \). We have
\[ W = \sum_{e=1}^{N_e} W^e. \]
Thus
\[ A = W + \sum_{e=1}^{N_e} (A^e - W^e), \]
\[ = W^{1/2} \left( I + W^{-1/2} \sum_{e=1}^{N_e} (A^e - W^e) W^{-1/2} \right) W^{1/2}. \]
Let
\[ A^e = I + W^{-1/2} (A^e - W^e) W^{-1/2}, \]
let
\[ B = W^{1/2} \prod_{e=1}^{N_e} (I + W^{-1/2} (A^e - W^e) W^{-1/2}) W^{1/2}, \]
\[ \tilde{B} = W^{1/2} \left( \prod_{e=1}^{N_e} A^e \right) W^{1/2}. \]
\( \vec{A}^e \) is the Winget regularized element matrix of \( A^e \). \( \vec{A}^e \) is symmetric and invertible. It can be factorized in the form

\[ \vec{A}^e = L^e D^e L^e \quad \text{Crout's decomposition.} \]

Then

\[ \vec{B} = W^{1/2} \prod_{e=1}^{N_{el}} (L^e D^e L^e) W^{1/2}. \]

To retain the symmetry of the initial problem, a symmetric preconditioner is used. \( \vec{B} \) is made symmetric by reordering the element matrices, i.e.:

\[ B = W^{1/2} \left( \prod_{e=1}^{N_{el}} L^e \right) \left( \prod_{e=1}^{N_{el}} D^e \right) \left( \prod_{e=N_{el}}^{1} L^e \right) W^{1/2}. \]

Remarks.

Let us suppose that we have a mesh where all the elements are unconnected, in other words two distinct elements have neither a common edge nor a common vertex. Then during the assembly of \( A \), its coefficients arise from the contribution of a single element matrix \( A^e \), and the elementary matrices commute together. If, for example,

\[ \begin{pmatrix} \vec{a}^i \\ \vec{a}^j \end{pmatrix} = \begin{pmatrix} 1 & \ldots \\ \ldots & \ldots \ldots \\ \ldots & \ldots \ldots \\ 1 & \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots
Furthermore,

\[
A^i - W^i = \begin{pmatrix}
0 & (\ast) & & & \\
& 0 & \ddots & & \\
& & \ddots & 0 & \\
& & & 0 & (0) \\
& & & & 0
\end{pmatrix}
\]

\[
A^j - W^j = \begin{pmatrix}
0 & & & & (0) \\
& 0 & \ddots & & \\
& & \ddots & 0 & \\
& & & 0 & (\ast) \\
& & & & 0
\end{pmatrix}
\]

Then

\[
\forall i, j \quad 1 \leq i, j \leq N_{el} \quad i \neq j \quad (A^i - W^i)(A^j - W^j) = (A^i - W^j)(A^j - W^i) = 0.
\]

But

\[
\tilde{B} - A = W^{1/2} \prod_{e=1}^{N_{el}} (I + W^{-1/2}(A^e - W^e)W^{-1/2})W^{1/2} - \sum_{e=1}^{N_{el}} A^e
\]

\[
= W^{1/2} \left( \sum_{i=2}^{n} \sum_{1 \leq j_1 < j_2 < \cdots < j_i \leq n} \left( \prod_{k=1}^{i} W^{-1/2}(A^{j_k} - W^{j_k})W^{-1/2} \right) \right)W^{1/2}
\]

\[
= \sum_{i=2}^{n} \sum_{1 \leq j_1 < j_2 < \cdots < j_i \leq n} \prod_{k=1}^{i} (A^{j_k} - W^{j_k}).
\]

Thus

\[
\tilde{B} - A = 0,
\]

i.e.

\[
\tilde{B} = A.
\]

In general, the more the element will be unconnected, i.e. each element will have few neighbours, the closer \( \tilde{B} \) will be to \( A \). However, the elementary matrices are reordered to obtain \( B \). If they do not commute together, \( B \) may be greatly different from \( \tilde{B} \).
4. Vectorization

At each iteration it is necessary to solve the linear system

\[ Bx = b, \]

i.e.

\[
W^{1/2} \left( \prod_{e=1}^{N_e} L_e^T \right) \left( \prod_{e=1}^{N_e} D_e \right) \left( \prod_{e=N_e}^{1} L_e^T \right) W^{1/2} x = b.
\]

We introduce

\[
\Delta = \left( \prod_{e=1}^{N_e} D_e \right)^{-1},
\]

\( \Delta \) is a diagonal matrix computed once.

\[
y = W^{-1/2} b \quad \text{vector product},
\]

\[
z = \left( \prod_{e=1}^{N_e} L_e^T \right)^{-1} y \quad \text{triangular system},
\]

\[
t = \Delta z \quad \text{vector product},
\]

\[
u = \frac{1}{\prod_{e=N_e}^{1} L_e^T} t \quad \text{triangular system},
\]

\[
x = W^{-1/2} u \quad \text{vector product}.
\]

The part to be vectorized is the solution of a triangular system. A colouring technique is used to do this.

Suppose that \( i \) and \( j \) are two unconnected elements of the mesh. We have, for instance

\[
(L_i')^{-1} = \begin{pmatrix}
[I_i^{-1}] & 0 \\
0 & [I]
\end{pmatrix} \\
0 & 0 & [I]
\]

\[
(L_j')^{-1} = \begin{pmatrix}
[I] & 0 & 0 \\
0 & [I_j^{-1}] & 0 \\
0 & 0 & [I]
\end{pmatrix}
\]

and

\[
(L_i')^{-1} (L_j')^{-1} = (L_j')^{-1} (L_i')^{-1}
\]

and the blocks \( L_i^{-1} \) and \( L_j^{-1} \) are disjoint. The computations of \( (L_i')^{-1} y \) and \( (L_j')^{-1} y \) do not involve the same coordinates of \( y \).

\[
L_i^{-1} y = \begin{pmatrix}
[I_i^{-1}] & 0 & 0 \\
0 & [I] & 0 \\
0 & 0 & [I]
\end{pmatrix} \begin{pmatrix}
[y_i] \\
[y_j] \\
[y]
\end{pmatrix} = \begin{pmatrix}
[y_i] \\
[y_j] \\
[y]
\end{pmatrix},
\]

\[
L_j^{-1} y = \begin{pmatrix}
[I] & 0 & 0 \\
0 & [I_j^{-1}] & 0 \\
0 & 0 & [I]
\end{pmatrix} \begin{pmatrix}
[y_i] \\
[y_j] \\
[y]
\end{pmatrix} = \begin{pmatrix}
[y_i] \\
[y_i] \\
[y]
\end{pmatrix}.
\]
Both computations are therefore independent and may be carried out simultaneously. Thus $B$ can be written as follows:

$$B = W^{1/2} \left( \prod_{p=1}^{\text{NC}} \left( \prod_{e=1}^{L_p} L_e \right) \right) \Delta \left( \prod_{p=\text{NC}}^{1} \left( \prod_{e=L_p}^{1} L_e^* \right) \right) W^{1/2},$$

where NC is the number of colours in the mesh and $L_p$ is the number of elements of the colour $p$.

The solution of $Bx = b$ can be written

$$x = W^{-1/2}b$$

vector operation

\[ \text{do } 1p = 1, \text{NC} \]
\[ \text{do } 2e = 1, L_p \]
\[ x = (L^e)^{-1}x \]
\[ \text{endo } 2 \]
\[ \text{endo } 1 \]

loop2: vector operation

$$x = \Delta^{-1}x$$

vector operation

\[ \text{do } 3p = \text{NC}, 1, -1 \]
\[ \text{do } 4e = L_p, 1, -1 \]

$$x = (L^e)^{-1}x$$

loop4: vector operation

\[ \text{endo } 4 \]
\[ \text{endo } 3 \]

$$x = W^{-1/2}x$$

vector operation

The loops 1 and 3 on the colours remain sequential.

5. Colouring

5.1. Principle

Let $T$ be a mesh resulting from a discretization by finite elements. We consider the graph $(T, E)$ the vertices of which are the elements and the edges of which are defined by:

if $(i, j)$ are two elements of the mesh,

$((i, j) \in E) \iff (i$ and $j$ have a common degree of freedom).

The graph is coloured, i.e. we seek for a partitioning $T_k$ of the vertices of the graph (each colour is associated to each part $T_k$) such that each edge has limits of different colours. Thus, we obtain a partition

$$T = \bigcup_{k=1}^{\text{NC}} T_k,$$

NC: number of colours of the mesh.

As the colouring is used to vectorize the computations on the elementary matrices, we require the other condition that two vertices of the graph of the same colour be associated to elements of the same geometrical and topological natures. We then renumber the vertices of the graph by successive colours.
5.2. The algorithm of the colouring

An input is given:
- the mesh,
- LVECT: maximum number of elements by colour,
- the total amount and number of the surfaces or curved lines (if there is any).

An output is obtained:
- the mesh which is renumbered colour by colour (the element of colour 1 are numbered first, then those of colour 2, etc.),
- NC: the number of colour necessary for the colouring,
- an array associated to the mesh giving the number of the last element of each colour.

6. Numerical results

We carried out tests on three problems: one was an elasticity problem, the two others were 2D and 3D thermal problems; the classic test in 2D, and a thermic test with Fourier boundaries in 3D. All computations were carried out on the Cray-2, using the BLAS1 library.

The maximum size of a colour LVECT was successively fixed at 64, 128, 192, which induced slight variations of convergence. We compared the EBE preconditioner with:
- the simple conjugate's gradient (without preconditioner) (SCG),
- the diagonal preconditioner (Diagonal),
- the incomplete Choleski preconditioner (ICCG),

implemented in the MODULEF library [8]. We compared for the EBE preconditioner the CPU time if the computation was carried out in scalar or vector mode. The CPU time is divided in two parts:
1. the computation of the preconditioner,
2. the resolution of the linear system, itself divided into:
   (i) the inversion of the preconditioner,
   (ii) the complete resolution of the conjugate gradient (the loop on the iteration, the search of the new conjugate direction, etc.).

We remark, that in the EBE case, this last measurement (2.ii) includes an initialization phase, which implies a loss of time.

6.1. Elasticity problem (2D)

Let us consider a 2-dimensional beam. It is covered by a regular mesh of 4000 triangles. The beam is clamped at one end ($\Gamma_1$), and a vertical force $f = (0, -f)$ with $f = 1N$ is applied at the other end ($\Gamma_2$).

We consider a isotropic homogeneous linearly elastic material. Young's modulus and Poisson's coefficient are:

\begin{align*}
E &= 5 \times 10^4, \\
\nu &= 0.25,
\end{align*}

The equilibrium equations of this problem are:

\begin{align*}
\text{div} \sigma &= 0 \quad &\text{in } \Omega, \\
\sigma_{xx} &= 0 \quad &\text{on } \Gamma_1, \\
\sigma_{yy} &= -f \quad &\text{on } \Gamma_2,
\end{align*}

\footnote{The CRAY-2 of the C.C.V.R. at the Ecole Polytechnique.}
with

\[ \sigma_{ij} = \lambda tr(u) \delta_{ij} + 2\mu e_{ij}(u), \]
\[ e_{ij}(u) = \frac{1}{2}(\partial_i u_j + \partial_j u_i), \]
\[ \nu = \frac{\lambda}{2(\lambda + \mu)}, \]
\[ E = \frac{\mu (3\lambda + 2\mu)}{\lambda + \mu}, \]

where \( \lambda, \mu \) are Lame's constants.

The discretization by Finite Elements \( P_1 \) Lagrange leads us to solve a linear system with 4242 unknowns. The algorithm of Conjugate Gradient has converged when the residual becomes less than \( 10^{-4} \). Results are displayed in Figs. 1 and 2.

Moreover we have compared the results of the EBE preconditioner, taking the vector length \( LVECT \) equal to 64, 128, 192, and as large as possible \((\infty)\), and we have compared the cpu execution time both purely sequentially (inhibited vectorization) and vectorially (forced vectorization).

Moreover we have tested the influence of the renumbering on the convergence, but we do not notice any variation of the convergence.

We have tested the EBE preconditioner on the same problem, but with a regular mesh of 100000 triangles. The discretization by Finite Elements \( P_1 \) Lagrange leads us to solve a linear system with 101202 unknowns. The algorithm of Conjugate Gradient has converged when the residual becomes less than \( 10^{-4} \). The vector length \( LVECT \) is taken as large as possible \((\infty)\).
Fig. 2. Two-dimensional elastic test.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of iterations</th>
<th>Total cpu</th>
<th>Resolution</th>
<th>Preconditioning</th>
<th>Remainder</th>
<th>Computation of preconditioner</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICCG</td>
<td>85</td>
<td>7.555</td>
<td>3.880</td>
<td>3.336</td>
<td>0.339</td>
<td></td>
</tr>
<tr>
<td>SCG</td>
<td>617</td>
<td>24.434</td>
<td>-</td>
<td>24.434</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Diagonal</td>
<td>591</td>
<td>15.818</td>
<td>0.194</td>
<td>15.624</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>EBEoo</td>
<td>243</td>
<td>13.676</td>
<td>3.606</td>
<td>9.720</td>
<td>0.350</td>
<td></td>
</tr>
<tr>
<td>Ratio EBE/ICCG</td>
<td>2.86</td>
<td>1.16</td>
<td>0.93</td>
<td>2.89</td>
<td>1.03</td>
<td></td>
</tr>
<tr>
<td>Ratio Diag/EBE</td>
<td>2.43</td>
<td>1.74</td>
<td>0.054</td>
<td>1.61</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LVECT</th>
<th>No of iterations</th>
<th>Resolution</th>
<th>Preconditioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>245</td>
<td>12.641</td>
<td>3.478</td>
</tr>
<tr>
<td>128</td>
<td>246</td>
<td>13.333</td>
<td>4.074</td>
</tr>
<tr>
<td>192</td>
<td>245</td>
<td>13.355</td>
<td>3.763</td>
</tr>
<tr>
<td>∞</td>
<td>243</td>
<td>13.326</td>
<td>3.606</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complete resolution</th>
<th>Preconditioning</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 vectorized</td>
<td>12.641</td>
</tr>
<tr>
<td>64 sequential</td>
<td>118.576</td>
</tr>
<tr>
<td>64 ratio</td>
<td>9.2</td>
</tr>
</tbody>
</table>
An element-by-element preconditioned conjugate gradient method

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of iterations</th>
<th>Total cpu</th>
<th>Resolution</th>
<th>Computation of preconditioner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Preconditioning</td>
<td>Remainder</td>
</tr>
<tr>
<td>ICCG</td>
<td>422</td>
<td>502.6</td>
<td>270.5</td>
<td>227.5</td>
</tr>
<tr>
<td>SCG</td>
<td>2875</td>
<td>1537</td>
<td>--</td>
<td>1537</td>
</tr>
<tr>
<td>Diagonal</td>
<td>2839</td>
<td>1548.6</td>
<td>20.8</td>
<td>1527</td>
</tr>
<tr>
<td>EBEoo</td>
<td>1231</td>
<td>1362</td>
<td>400.5</td>
<td>952.5</td>
</tr>
<tr>
<td>Ratio EBE/ICCG</td>
<td>2.91</td>
<td>2.71</td>
<td>1.48</td>
<td>4.19</td>
</tr>
<tr>
<td>Ratio Diag/EBE</td>
<td>2.3</td>
<td>1.13</td>
<td>0.052</td>
<td>1.60</td>
</tr>
</tbody>
</table>

Complete resolution

<table>
<thead>
<tr>
<th>Method</th>
<th>Vectorized</th>
<th>Preconditioning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1353</td>
<td>400.5</td>
</tr>
<tr>
<td>Sequential</td>
<td>12456</td>
<td>11420</td>
</tr>
<tr>
<td>Ratio</td>
<td>9.20</td>
<td>28.5</td>
</tr>
</tbody>
</table>

6.2. Two-dimensional thermal problem

Let $\Omega$ be the square domain $[0, 1] \times [0, 1]$, with a uniform mesh of 1600 squares. $P_1$ Finite Elements are used. The equations of the problem are:

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

with

$$\forall (x, y) \in \Omega^2, f(x, y) = 32x(1 - x)y(1 - y).$$

Fig. 3. Two-dimensional thermic test.
We thus obtain a problem with 1600 degrees of freedom. The algorithm of Conjugate Gradient has converged when the residual becomes less than $10^{-4}$. Results are displayed in Figs. 3 and 4.

6.3. Three-dimensional thermal problem

Let $\Omega$ be the cube $[0, 1]^3$ meshed with $40^3$ cubes. We use the $Q_1$ hexahedra. Let $\partial \Omega_1$ be the three sides of $\Omega$ which are in the planes $(Ox, Oy)$, $(Ox, Oz)$, $(Oy, Oz)$. Let $\partial \Omega_2$ be the three other sides of $\Omega$. Let $\partial \Omega = \partial \Omega_1 \cup \partial \Omega_2$.

We consider the thermic test with Fourier boundaries:

$$\begin{cases} 
-\Delta u(x, y, z) = 0 & \text{in } \Omega \\
\partial u/\partial n + u = u_0 & \text{on } \partial \Omega,
\end{cases}$$

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of iterations</th>
<th>Total cpu</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Preconditioning</td>
</tr>
<tr>
<td>ICCG</td>
<td>13</td>
<td>0.469</td>
<td>0.203</td>
</tr>
<tr>
<td>SCG</td>
<td>43</td>
<td>0.616</td>
<td>–</td>
</tr>
<tr>
<td>Diagonal</td>
<td>43</td>
<td>0.611</td>
<td>0.006</td>
</tr>
<tr>
<td>EBE</td>
<td>25</td>
<td>0.588</td>
<td>0.059</td>
</tr>
<tr>
<td>Ratio EBE/ICCG</td>
<td>1.92</td>
<td>1.25</td>
<td>0.29</td>
</tr>
<tr>
<td>Ratio DIAG/EBE</td>
<td>1.72</td>
<td>1.05</td>
<td>0.10</td>
</tr>
</tbody>
</table>
Fig. 5. Three-dimensional thermic test.

Fig. 6. Three-dimensional thermic test.
where
\[ u_0 = \frac{1}{(x + y + z + \alpha)^2} + \frac{1}{x + y + z + \alpha} \text{ on } \partial \Omega_1, \]
\[ u_0 = -\frac{1}{(x + y + z + \alpha)^2} + \frac{1}{x + y + z + \alpha} \text{ on } \partial \Omega_2 \]
and \( \alpha = 0.1 \). The algorithm of Conjugate Gradient has converged when the residual becomes less than \( 10^{-4} \). Results are displayed in Figs. 5 and 6.

7. Conclusion

1. The EBE preconditioner can be very well vectorized. The ratio of sequential execution time (inhibited vectorization) to vectorized execution time (forced vectorization) is about 10.
2. In the bi-dimensional case, the more degrees of freedom the problem has, the more competitive the preconditioner is: in the case of our 2D thermal problem (with few degrees of freedom), the computation of the preconditioner is relatively much greater than in the 2D elasticity problem (with large number of degrees of freedom), but in this last problem, ICCG is better vectorized, because the arrays have longer rows.
3. the 3D thermal problem gives satisfactory results: EBE, while not being as good as the Incomplete Cholesky preconditioner, is much faster than the Diagonal preconditioner (cf. convergence curves).
4. Problems of 3D elasticity, which give ill-conditioned matrices, remain to be tested.

References

