```
                                          //     copie du sous tableaux
  Rnmk Bsub(B(FromTo(1,2),FromTo(1,3),FromTo(0,3)));
  B(SubArray(2,1),SubArray(3,1),SubArray(4,0))   = -1;
  B(SubArray(2,1),SubArray(3,1),SubArray(4,0)) += -1;
  cout << " B       = " << B << endl;
  cout << Bsub << endl;

  return 0;
}
```

### 6.2.4 La résolution d'un systéme linéaire avec le gradient conjugué

L'algorithme du gradient conjugué présenté dans cette section est utilisé pour résoudre le système linéaire $Ax = b$, où $A$ est une matrice symétrique positive $n \times n$. Cet algorithme est basé sur la minimisation de la fonctionnelle quadratique $E : \mathbb{R}^n \to \mathbb{R}$ suivante :

$$E(x) = \frac{1}{2}(Ax, x) - (b, x),$$

La version préconditionnée avec un matrice $C$ correspond après avec un changement de variable $x = Cy$ à la resolution de ${}^tCACy = {}^tCb$, et à la minimisation de la fonctionnelle :

$$E_c(y) = \frac{1}{2}(ACy, y)_C - (b, y)_C,$$

où $(.,.)_C$ est le produit scalaire associer à une une matrice $C$, symétrique définie positive de $\mathbb{R}^n$. et car la matrice $CAC$ est symétrique positive.

**Algorithme 6.** **Le gradient conjugué preconditionné**

soient $x^0 \in \mathbb{R}^n$, $\varepsilon$, $C$ donnés
- $G^0 = Ax^0 - b$
- $H^0 = -CG^0$
- *si* $(G^0, G^0)_C < \varepsilon$ *stop*
- *pour* $i = 0$ *à* $n$
  - $\rho = -\dfrac{(G^i, H^i)}{(H^i, AH^i)}$
  - $x^{i+1} = x^i + \rho H^i$
  - $G^{i+1} = G^i + \rho AH^i$
  - $\gamma = \dfrac{(G^{i+1}, G^{i+1})_C}{(G^i, G^i)_C}$
  - $H^{i+1} = -CG^{i+1} + \gamma H^i$
  - *si* $(G^{i+1}, G^{i+1})_C < \varepsilon$ *stop*

**Théorème 6.1.** *Notons $\mathcal{E}_C(x) = \sqrt{(Ax - \overline{x}, x - \overline{x})_C}$, l'erreur dans la norme de $A$ preconditionné, où $\overline{x}$ est la solution du problème. Alors l'erreur à l'intération $k$ un gradient conjugué est majoré :*

$$\mathcal{E}_C(x^k) \leq 2\left(\frac{\sqrt{K_C(A)} - 1}{\sqrt{K_C(A)} + 1}\right)^k \mathcal{E}_C(x^0)$$