Université de Picardie Jules Verne
Master 2 AAM
Calcul Scientifique
2019-2020

Gradient, Gradient Conjugué/Préconditionné

1 Gradient Conjugué - Préconditionné

1.1 Méthode de Projection simple

Soit A SDP. La méthode de Richardson (1910) est obtenue pour $M = \frac{1}{\alpha} Id$

$$x^{(k+1)} = x^{(k)} + \alpha(b - Ax^{(k)}) = u^{(k)} + \alpha r^{(k)} \to r^{(k)} = (Id - \alpha A)^k r^{(0)}$$

La méthode converge SSI $0 < \alpha < \frac{2}{\lambda_{max}}$ et $\rho(I - \alpha A)$ est minimisé pour $\alpha^* = \frac{2}{\lambda_{min}+\lambda_{max}}$.

Généralisation pour α variable : $x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}$ (D. Young, 1954, avec aussi choix cycliques de α_k). On a

$$r^{(k+1)} = \left(\prod_{k=0}^{k} (Id - \alpha_k A)\right) r^{(0)} = P_{k+1}(A) r^{(0)}.$$

La méthode de la plus profonde descente due à L-A Cauchy (1847) consiste à prendre α_k de sorte à minimiser $\|e^{(k+1)}\|_A$.
Le problème de ces méthodes est qu'elles convergent lentement.

Pour la méthode de plus profonde descente, on a

$$\|e_k\|_A \leq \left(\frac{cond_2(A) - 1}{cond_2(A) + 1}\right)^k \|e_0\|_A$$

où $e_k = u_k - x$ et $\|e\|_A^2 =< e, Ae >$.
Pour accélérer la convergence, on réécrit là encore les iérations sous la forme

$$\mathbf{B} y^{(k)} = r^{(k)} \text{ puis } x^{(k+1)} = x^{(k)} + \alpha_k y^{(k)}$$

Finalement, on modifie l'algorithme original en insérant une étape de préconditionnement et l'on obtient l'algorithme préconditionné.

Exercice 1 (Plus profonde descente préconditionnée)
Programmer la méthode en prenant pour A la matrice de $-\Delta$ sur le carré unité et pour péconditionneur B une factorisation LU incomplète.

1.2 Méthode du gradient conjugué (Stiefel 1952)

Gradient Conjugué

`poser` $r^0 = b - Au^0,\ p^0 = r^0$
`pour` $k = 0, \cdots$
`poser` $z^k = Ap^k$
`poser` $\alpha_k = \frac{<p^k, r^k>}{<p^k, z^k>}$
`poser` $u^{k+1} = u^k + \alpha_k p^k$
`poser` $r^{k+1} = r^k - \alpha_k z^k$
`poser` $\beta_{k+1} = \frac{<z^k, r^{k+1}>}{<p^k, z^k>}$
`poser` $p^{k+1} = r^{k+1} - \beta_{k+1} p^k$

$$\|e^k\|_A \leq \frac{2c^k}{1+2c^{2k}} \|e^0\|_A, \text{ avec } c = \frac{\sqrt{K_2(A)} - 1}{\sqrt{K_2(A)} + 1}.$$

1.3 Gradient Conjugué préconditionné

Gradient Conjugué préconditionné

`poser` $r^0 = b - Au^0, \mathbf{B} z^0 = r^0, d_0 = z_0$
`pour` $k = 0, \cdots$
`poser` $\alpha_k = \frac{<z^k, r^k>}{<d^k, Ad^k>}$
`poser` $u^{k+1} = u^k + \alpha_k d^k$
`poser` $r^{k+1} = r^k - \alpha_k A d^k$
`résoudre` $\mathbf{B} z^{k+1} = r^{k+1}$
`poser` $\beta_{k+1} = \frac{<z^{k+1}, r^{k+1}>}{<z^k, r^k>}$
`poser` $d^{k+1} = z^{k+1} + \beta_{k+1} d^k$

Remarque 1
Si $B = SS^T$ alors le Gradient Conjugué Préconditionné par B appliqué à A équivaut au gradient conjugué appliqué à $S^{-1}AS^{-T}$ (exercice).

De la sorte l'estimation d'erreur devient

$$\|e^k\|_{S^{-1}AS^{-T}} \leq \frac{2c^k}{1+2c^{2k}} \|e^0\|_A, \text{ avec } c = \frac{\sqrt{K_2(S^{-1}AS^{-T})} - 1}{\sqrt{K_2(S^{-1}AS^{-T})} + 1}.$$

L'intérêt du préconditionnement est alors évident.

Exercice 2
On considère le système $Ax = b$ où A est la matrice de discrétisation de $-\dfrac{\partial^2}{\partial x^2}$ sur $[0, 1]$, avec conditions aux limites de type Dirichlet homogènes.

1. Ecrire un programme Scilab pour résoudre le système linéaire.

2. Executer le code lorsque `b=ones(N*N,1)` . Que remarquez-vous ?

Exercice 3 (Construction de Factorisations incomplètes et seuillées)
On considère ici le système $Ax = b$ où A est la matrice de discrétisation de $-\Delta$ sur $[0, 1]2$, avec conditions aux limites de type Dirichlet homogènes.

1. Utiliser IUL(0) pour calculer $\mathbf{B} = (D - E)D^{-1}(D - F)$, pour définir IC(0).

2. Partir de `b=rand(N*N,1)` et résoudre le système obtenu par

   - Gradient conjugué
   - Gradient conjugué préconditionné par **B**.

3. Comparer l'historique des résidus. Conclusion ?

4. Même question avec `b=ones(N*N,1)`

Exercice 4 (Best of)
Pour A, la matrice de discrétisation de $-\Delta$ sur $[0,1]2$, avec conditions aux limites de type Dirichlet homogènes et `b=ones(N*N,1)` , comparer l'historique des méthodes : plus profonde descente, plus profonde descente préconditionnée, Gradient conjugué, gradient conjugué préconditionné.

2 Annexe

2.1 Gradient conjugué

```
function [x,R]=CG(A,b,x0,epsilon,Maxit)
x=x0;
r=b-A*x;
p=r;
k=0;
res=norm(r);
R=[res];
while k<Maxit & res > epsilon
z=A*p;
alpha=p'*r/(p'*z);
x=x+alpha*p;
r=r-alpha*z;
beta=z'*r/(p'*z);
p=r-beta*p;
res=norm(r);
R=[R res];
end
endfunction
```

2.2 Gradient conjugué préconditionné par $\mathbf{B} = S^T S$

```
function [x,R]=PCG(A,SPrec,b,x0,epsilon,Maxit)
x=x0;
r=b-A*x;
z=SPrec'\(SPrec\ r);
d=z;
k=0;
res=norm(r);
R=[res];
while k<Maxit & res > epsilon
Ad=A*d;
a=z'*r;
alpha=a/(d'*Ad);
x=x+alpha*d;
r=r-alpha*Ad;
z=SPrec'\(SPrec\ r);
beta=(r'*z)/a;
d=z+beta*d;
res=norm(r);
R=[R res];
end
endfunction
```