

Méthodes de sous espaces de Krylov

1 Méthode d'Arnoldi

1.1 Procédé d'orthogonalisation

La méthode d'Arnoldi permet de calculer une base orthonormale d'un espace de Krylov $K_m(A; v) = \text{Vect}(v, Av, \dots, A^{m-1}v)$. On procède comme suit :

On pose $v_1 = \frac{v}{\|v\|}$ et on génère une base orthonormale en mettant en œuvre un procédé de Gram-Schmidt :

Pour $k = 1, \dots, m$	
Calculer	$h_{i,k} = \langle v_i, Av_i \rangle, \quad i = 1, \dots, k$
Poser	$w_k = Av_k - \sum_{i=1}^k h_{i,k} v_i, \quad h_{k+1,k} = \ w_k\ $
Si $w_k = 0$	STOP
Sinon	Poser $v_{k+1} = \frac{w_k}{\ w_k\ }$

Si V_m est la matrice dont les colonnes sont les vecteurs v_i ainsi construits, alors

$$V_m^T A V_m = H_m$$

On a de plus la relation $V_{m+1}^T A V_m = \hat{H}_m$, avec $\hat{H}_m$ $(m+1) \times m$ matrice dite de Hessenberg supérieure ; H_m est définie par le processus.

1.2 Arnoldi pour les systèmes linéaires

L'idée consiste à imposer la condition $r^k \perp K_k(A; r^0)$, $k \geq 1$. Ainsi, disposant d'une base orthonormée de $K_k(A; r^0)$, cela revient à imposer

$$V_k^T r^k = 0$$

Une méthode de sous-espace de Krylov consiste à chercher $x^k \in x_0 + K_k(A; r^0)$, donc sous la forme

$$x^k = x^0 + V_k z^k.$$

On a donc la relation $r^k = r^0 - A V_k z^k$. Du coup

$$V_k^T r^0 - V_k^T A V_k z^k = 0$$

or r^0 est le premier vecteur de la base de l'espace de Krylov et donc la première colonne de V_k , $k \geq 1$. Il en découle que

$$V_k^T r^0 = \|r^0\| e_1$$

On calcule donc z^k comme solution du système

$$H_k z^k = \|r^0\| e_1$$

La méthode d'Arnoldi consiste donc à résoudre la suite des systèmes linéaires $H_k z^k = \|r^0\| e_1$.

Théorème 1

La méthode d'Arnoldi converge en au plus n itérations.

Preuve.

- Il est clair que si la méthode s'arrête à la n -ième itération, alors $x^n = x$ puisque $K_n(A, r^0) = \mathbb{R}^n$.
- Si la méthode s'arrête à la m -ième itération, $m < n$, on parle alors de *breakdown*, montrons que $x^m = x$. Nous avons en effet par définition

$$x^m = x^0 + V_m z^m$$

avec $z^m = H_m^{-1}(\|r^0\| e_1)$ et $H_m = V_m^T A V_m$. Du coup

$$x^m = x^0 + V_m H_m^{-1} V_m^T r^0 = A^{-1} b.$$

■

2 GMRES

L'idée directrice de la méthode GMRES (Generalized Minimal RESidual) est de définir x^k de sorte à minimiser la norme euclidienne du résidu. En écrivant x^k dans une base orthonormée de l'espace de Krylov $K(A; r^0)$, on a

$$r^k = r^0 - A V_k z^k$$

avec $r^0 = \|r^0\| v_1$. Ainsi, puisque $V_{k+1}^T A V_k = \hat{H}_k$, il vient

$$r^k = V_{k+1} \left(\|r^0\| e_1 - \hat{H}_k z^k \right)$$

Le vecteur z^k est alors calculé de sorte à minimiser $\| \|r^0\| e_1 - \hat{H}_k z^k \|$. On peut montrer le résultat suivant

Théorème 2

La méthode GMRES converge en au plus n itérations.

3 Exercices

Exercice 1 (Orthogonalisation)

Soit A une matrice inversible $n \times n$ (de petite taille, par exemple $n = 5$) et v un vecteur, $v \neq 0$.

1. Construire l'espace de Krylov $K_n(A; v)$. Calculer la dimension de cet espace, on pourra utiliser la commande `rank`.
2. La base obtenue est-elle orthonormée ? On notera V la matrice dont les colonnes sont les vecteurs produits par le procédé d'Arnoldi.
Ind. On pourra vérifier l'orthogonalité que par un simple calcul sur V .
3. Appliquer la méthode d'Arnoldi pour en extraire une base orthonormée. Vérifier que la base obtenue est bien orthonormée.
4. Calculer la matrice $V^T A V$. Que constatez-vous ?
5. Comparer $V^T A V$ à la matrice H .

Exercice 2 (Résolution d'un système linéaire par Arnoldi)

Soit A une matrice inversible $n \times n$.

1. Soit `A=2*eye(N,N)-diag(ones(N-1,1),1)-diag(ones(N-1,1),-1);`. Appliquer la méthode d'Arnoldi pour les systèmes linéaires, on tracera l'historique des résidus.
2. Même question avec `A=kron(Id,Axx)+kron(Axx,Id)` ; où `Id=speye(N,N)` ;.

Exercice 3 (Résolution d'un système linéaire par GMRES)

Soit A une matrice inversible $n \times n$.

1. Soit `A=2*eye(N,N)-diag(ones(N-1,1),1)-diag(ones(N-1,1),-1)`. Appliquer la méthode GMRES pour les systèmes linéaires, on tracera l'historique des résidus.
2. Même question avec `A=kron(Id,Axx)+kron(Axx,Id)` ; où `Id=speye(N,N)` ;.

4 Annexe

4.1 Arnoldi

```
function [V,H]=AlgoArnoldi(A,v,m)
v=v/norm(v,2);
V=[v]; H=[];k=0;
while k <=m-1
[k,V,H]=GrmSchArnoldi(A,m,k,V,H);
end
endfunction
```

```
function [k,V,H]=GrmSchArnoldi(A,m,k,V,H)
epsilon=1.e-10;
k=k+1;
H=[H,V(:,1:k)']*A*V(:,k)];
s=0;
for i=1:k
s=s+H(i,k)*V(:,i);
end
w=A*V(:,k)-s;
H(k+1,k)=norm(w,2);
if(H(k+1,k)>=epsilon) & (k<m)
V=[V,w/H(k+1,k)];
else
k=m+1;
end
endfunction
```

```

function [x,k]=METHArnoldi(A,b,m,x0,toll)
r0=b-A*x0; nr0=norm(r0,2);
if nr0 ~= 0
v1=r0/nr0;
V=[v1];
H=[]; k=0;
istop=0;
while (k <= m-1) & (istop == 0)
[k,V,H] = GrmSchArnoldi(A,m,k,V,H);
[nr,nc]=size(H);
e1=eye(nc);
y=(e1(:,1)')*nr0)/H(1:nc,:);
residual = H(nr,nc)*abs(y*e1(:,nc));
if residual <= toll
istop = 1;
y=y';
end
end
if istop==0
[nr,nc]=size(H);
e1=eye(nc);
y=(e1(:,1)')*nr0)/H(1:nc,:);
y=y';
end
x=x0+V(:,1:nc)*y;
else
x=x0;
end
endfunction

```

4.2 GMRES

```
function [x,k,resid]=gmres(A,b,m,toll,x0)
r0=b-A*x0;
nr0=norm(r0,2);
if nr0~=0
v1=r0/nr0;
V=[v1];
H=[];
k=0;
residual=1;
resid=[residual];
while k <= m-1 & residual > toll,
[k,V,H] = GrmSchArnoldi(A,m,k,V,H);
[nr,nc]=size(H); y=(H'*H) (H'*nr0*[1;zeros(nr-1,1)]);
x=x0+V(:,1:nc)*y;
residual = norm(b-A*x,2)/nr0;
resid=[resid,residual]; end
else
x=x0;
end
endfunction
```