Rapport Tp2 Calcul Scientifique - Analyse de Données

Mdaa Saad // El Bennouri Abdeslam // Dahhoumi Mouad

Dpartement Sciences du Numérique - 1A
2019-2020

Question 1

```
smdaa@r2d2:~/1A/Projet_TP2_Fournitures/code_fourni/Fortran$ ./main -n 10 -v 11 -m 10

Calling the basic deflated power method
Eigenvalue    1:    10.000  9.19E-16
Eigenvalue    2:     9.000  8.82E-16
Eigenvalue    3:     8.000  9.87E-16
Eigenvalue    4:     7.000  8.78E-16
Eigenvalue    5:     6.000  9.15E-16
End of the basic deflated power method

percentage  0.70 reached with    5 eigenvalues

 ================================
 Time =    7.0000002160668373E-003
 ================================
```

```
smdaa@r2d2:~/1A/Projet_TP2_Fournitures/code_fourni/Fortran$ ./main -n 10 -v 10

Calling LAPACK DSYEV on A
End of LAPACK DSYEV

 ================================
 Time =    1.0000000474974513E-003
 ================================
```

On remarque que la méthode de puissance itérée est 7 fois plus lente que la méthode de Lapack dsyev.

Question 2

La méthode puissance est un algorithme simple, mais elle peut converger lentement. à cause de sa nature itérative : on calcule une valeur propre ,on applique l'étape de deflation puis on calcule la valeur propre suivante .

Question 3

On suppose qu'on applique l'algorithme 1 sur m vecteur (une matrice $V \in \mathbb{R}^{n,m}$) Donc

$$\begin{cases} V^0 \in \mathbb{R}^{n,m} \\ V^{k+1} = \frac{AV^k}{||AV^k||} \end{cases}$$

On suppose que A est diagonale de valeurs propres $\lambda_1, .., \lambda_n$, de vecteurs propres associés $\mu_1, .., \mu_n$, avec λ_1 la valeur propre dominante
soit c_i une colonne de V ($i \in [|1, m|]$) donc elle se décompose :

$$c_i = \sum_{j=1}^{n} a_j^{(i)} \mu_j$$

Donc

$$A^k c_i = \sum_{j=1}^{n} a_j^{(i)} A^k \mu_j = \sum_{j=1}^{n} a_j^{(i)} \lambda_j^k \mu_j$$

Alors

$$A^k c_i = \lambda_1^k \sum_{j=1}^{n} a_j^{(i)} (\frac{\lambda_j}{\lambda_1})^k \mu_j$$

On a pour $j \in [|2, n|]$

$$(\frac{\lambda_j}{\lambda_1})^k \to 0$$

Alors

$$A^k c_i \approx \lambda_1^k a_1^{(i)} \mu_1$$

Donc $\forall i \in [|1, m|]$ on

$$c_i \to \mu_1$$

Question 4

On a

$$H = V^T.A.V \in \mathbb{R}^{m,m}$$

Donc on est amenée à faire la décomposition spectrale à une matrice de dimension bien inférieur à la dimension de A . Par exemple dans l'application de l'ACP le nombre des vecteurs propres qu'on cherche est très inférieur à la dimension de la matrice de données.

Question 5

voir code.

Question 6

- Generate an initial set of m orthonormal vectors $V \in \mathbb{R}^{m,m}$

```
Vr = randn(n, m);
Vr = mgs(Vr);
```

- $V \leftarrow$ orthonormalisation of the columns of $Y = A.V$

```
Y = A*Vr;
Vr = mgs(Y);
```

- $Rayleigh - Ritz$ projection applied on matrix A and orthonormal vectors V

```
[Wr, Vr] = rayleigh_ritz_projection(A, Vr);
```

- *Convergence analysis step*

```
while(~analyse_cvg_finie),
    .. % ligne [68 -105]
end
```

- save eigenpairs that have converged and update *RercentReached*

```
if(conv)
    n_ev = nb_c;
    V = Vr(:, 1:n_ev);
    W = W(1:n_ev);
    it = k;
else
    W = zeros(1,1);
    V = zeros(1,1);
    n_ev = 0;
    it = k;
end
```

Question 7

- soit $A = (a_{ij}) \in \mathbb{R}^{n,n}$, on note $A^2 = (c_{ij})$ Donc $\forall i, j \in [|1, n|]$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} a_{kj}$$

Donc on a n multiplications et n additions pour le calcul d'un élément c_{ij} .Alors pour calculer A^2 on a besoin de :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} 2n = 2n^3$$

donc pour calculer A^p le nombre de flops nécessaire est :

$$\boxed{2(p-1)n^3}$$

• On note $A^p.V = (d_{ij})$ Donc $\forall i, j \in [|1, n|] * [|1, m|]$

$$d_{ij} = \sum_{k=1}^{n} a_{ik}^p v_{kj}$$

de même on a n multiplications et n additions pour le calcul d'un élément d_{ij}.Alors pour calculer $A^p.V$ on a besoin de :

$$\sum_{i=1}^{n}\sum_{j=1}^{m} 2n = \boxed{2mn^2}$$

•

Question 8

voir code.

Question 9

```
 ||A*V_i - Lambda_i*V_i||/||A||
 ==============================
Eigenvalue    1: 0.272E-15
Eigenvalue    2: 0.227E-15
Eigenvalue    3: 0.244E-15
Eigenvalue    4: 0.293E-15
Eigenvalue    5: 0.327E-15
Eigenvalue    6: 0.236E-15
Eigenvalue    7: 0.179E-15
Eigenvalue    8: 0.225E-15
Eigenvalue    9: 0.213E-15
Eigenvalue   10: 0.190E-15
Eigenvalue   11: 0.158E-15
Eigenvalue   12: 0.214E-15
Eigenvalue   13: 0.244E-15
Eigenvalue   14: 0.146E-15
Eigenvalue   15: 0.178E-15
Eigenvalue   16: 0.188E-15
Eigenvalue   17: 0.144E-15
Eigenvalue   18: 0.171E-15
Eigenvalue   19: 0.302E-15
Eigenvalue   20: 0.949E-15
```

• On remarque que la précision diffère pour certains des vecteurs en effet :la précision dépend du rapport de la plus grande valeur propre et la deuxième plus grande valeur propre .

Question 10

• Pour la version 3 la précision sera meilleur en effet on augmente la puissance de la matrice A donc on augmente le rapport de la plus grande valeur propre et la deuxième plus grande valeur propre .

Question 11

voir code.

Question 12

pour $p = 5$

```
smdaa@vador:~/1A/Projet_TP2_Fournitures/code_fourni/Fortran$ ./main -imat 3 -n 500 -v 2 -m 30 -disp 0

Calling the subspace iteration method v2

End the subspace iteration method v2
percentage  0.70 NOT reached with   30 eigenvalues
 n_ev =          30

 ==================================
 Time =    1.0650000572204590
 ==================================
```

pour $p = 6$

```
smdaa@vador:~/1A/Projet_TP2_Fournitures/code_fourni/Fortran$ ./main -imat 3 -n 500 -v 2 -m 30 -disp 0

Calling the subspace iteration method v2

End the subspace iteration method v2
percentage  0.70 NOT reached with   30 eigenvalues
 n_ev =          30

 ==================================
 Time =   0.92599999904632568
 ==================================
```

pour $p = 7$

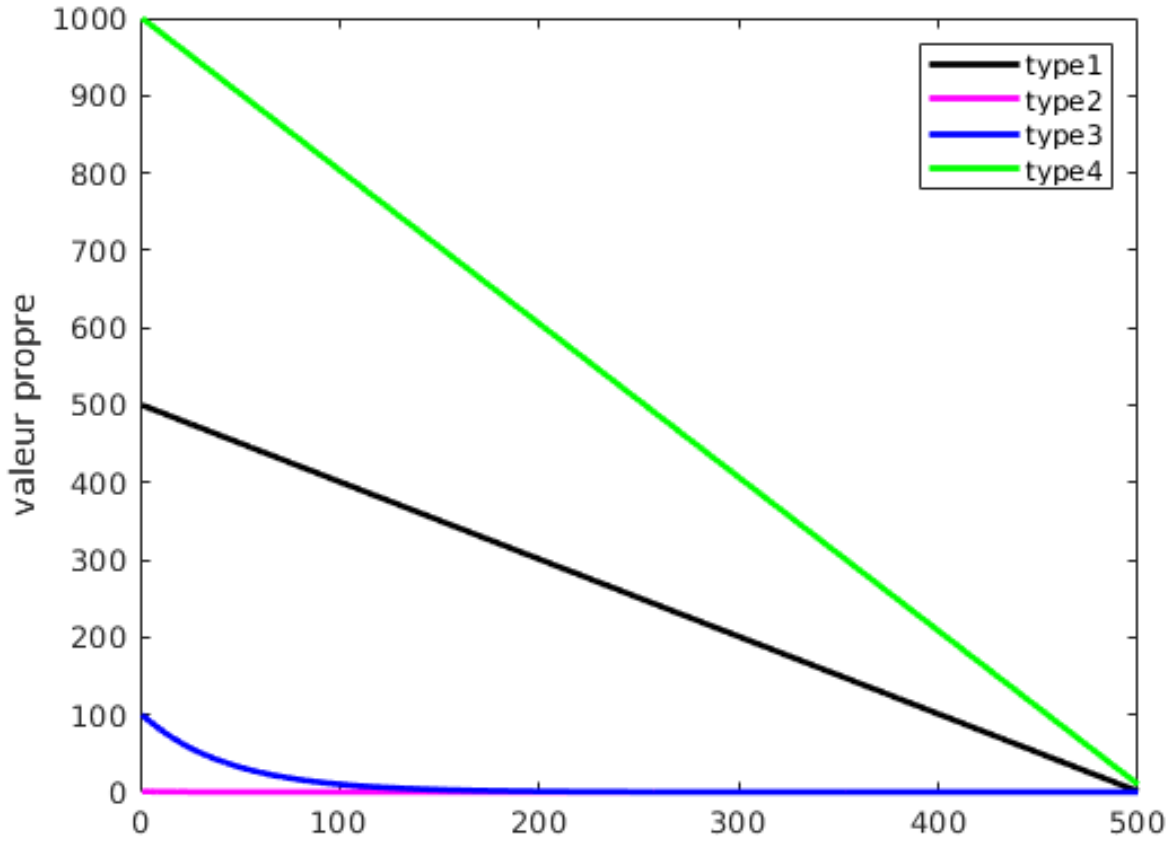
```
smdaa@vador:~/1A/Projet_TP2_Fournitures/code_fourni/Fortran$ ./main -imat 3 -n 500 -v 2 -m 30 -disp 0

Calling the subspace iteration method v2

End the subspace iteration method v2
percentage  0.70 NOT reached with   30 eigenvalues
 n_ev =          30

 ==================================
 Time =   0.84500002861022949
 ==================================
```

pour $p = 8$

```
smdaa@vador:~/1A/Projet_TP2_Fournitures/code_fourni/Fortran$ ./main -imat 3 -n 500 -v 2 -m 30 -disp 0

Calling the subspace iteration method v2

End the subspace iteration method v2
percentage  0.70 NOT reached with   30 eigenvalues
 n_ev =          30

 ==================================
 Time =   0.75800001621246338
 ==================================
```

• On remarque que lorsqu'on augmente p le temps diminue et la précision augmente aussi , en effet en augmentant p on augmente le rapport entre la plus grande valeur propre et la deuxième seconde valeur propre.

Question 13



• On remarque que la distribution des valeurs propre d'un matrice de type 1 et de type 4 est linéaire contrairement au type 2 et au type 3.
• Pour une matrice de type 3 la converge va être plus rapide car le rapport entre la plus grande valeur propre et la deuxième seconde valeur propre est plus important.

Question 14

pour $p = 5$
on exécute la commande : ./main -imat i -n j -v k -m 20 avec

$$i = 1, 3, 4$$

$$j = 50, 1000$$

$$k = 1, 2, 10, 3$$

On obtient les résultats suivantes pour $n = 50$:

matrice	v 1	v2	dseyv	v3
type 1	0,114 s	$1,79.10^{-2}$ s	1.10^{-3} s	$1,7.10^{-2}$ s
type 3	1.10^{-3} s	1,01 s	2.10^{-3} s	0.72 s
type 4	$9,3.10^{-2}$ s	$1,8.10^{-2}$ s	2.10^{-3} s	$1.9.10^{-2}$ s

On obtient les résultats suivantes pour $n = 1000$:

matrice	v 1	v2	dseyv	v3
type 1	98,09 s	$44,4$ s	$1,29$ s	$16,38$ s
type 3	$23,98$ s	$5,29$ s	$1,09$ s	$2,31$ s
type 4	98,78 s	$46,5$ s	$1,28$ s	$35,8$ s

• On remarque que sur les matrices de type 3 on obtient les meilleures performances en terme de temps.

• On remarque que la performance des matrice de type 1 et de type 4 sont très proches puisque les deux ont une distribution de valeurs propres linéaire.

• On remarque que pour une matrice de petite taille de type 3 la méthode la plus performante est la version 1 .

• On remarque que obtient une accélération plus importante pour la version 2 pour une matrice de type 3.

• On remarque que la méthode dseyv est la plus efficace en terme de temps et en terme de précision.

• On remarque que l'accélération obtenu par la version 3 est plus importante que la version 2 .