Projet d'Algèbre Linéaire Numérique :
Utilisation des EOFs pour prédire la température des océans

Philippe LELEUX
Naji BOULAMI
Arthur MANOHA
Groupe 19

15 Mai 2012

Résumé :
Le but de ce projet est d'implanter des EOFs dans le but de permettre la prédiction de la température des Océans. La première phase traitait de l'implantation elle même. Dans cette phase, on cherche à optimiser le calcul des paires propres et à étudier l'influence de la variance expliquée et du nombre de données que l'on ignore.

Table des matières

1 Introduction

1.1 Phase 1

Dans la première phase du sujet, nous avons introduit l'importance d'avoir un outil tel que les EOFs et la manière de les obtenir. Pour cela nous devions implanter l'algorithme de la puissance itérée qui permet de trouver les valeurs propres d'une matrice. C'est alors que nous avons aborder des problèmes tels que les conditions d'arrêt et avons préparer la phase 2

1.2 Phase 2

La deuxième phase se déroule en plusieurs étapes.
D'abord il fallait implanter 3 versions d'un algorithme sous fortran qui approfondissait la notion de puissance itérée. Chaque version de cette algorithme optimisait le calcul pour le rendre plus efficace et plus rapide.
Ensuite, l'implantation du source EOF.m sous matlab permettait en appelant les programmes fortran de trouver les EOFs, de les afficher et ensuite d'effectuer la prédiction.
S'en est suivi une phase de test, montrant plusieurs propriétés sur les EOFs et permettant d'analyser les phénomènes réels grâce à seulement quelques EOFs.

Nous allons présenter cette phase du projet en s'intéressant d'abord au calcul des paires propres dominantes codé en fortran puis à la méthodes pour effectuer le scénario de la prédiction codée en Matlab et enfin les résultats et l'analyse des résultats.

2 Calcul des paires propres dominantes

Dans cette partie, on a besoin de développer un algorithme permettant de trouver les m plus grandes paires propres. Comme vu dans la phase une, l'algorithme de la puissance itérée, ne permet pas le calcul des plus grandes paires propres, alors pour atteindre notre but, il faut alors trouver une autre méthode afin d'être satisfait.

2.1 Première version Proposée

Soit une matrice A, notre but comme on l'a expliqué ci-dessus c'est de calculer les m paires propres dominantes. Afin d'atteindre notre but nous générons sur matlab à l'aide d'une fonction prédéfini "Random" et un test sur le rang une matrice constitué de m vecteurs indépendants. En effet on essaye de tendre les valeurs propres de H vers les valeurs propres de A, puisque d'après la décomposition spectrale on a : AV=VH. Pour cela on utilise l'algorithme d'orthogonalisation de Schmidt afin d'orthogonaliser les colonnes de V, après un calcul de AV est nécessaire pour former le quotient de Rayleigh.

2.2 Deuxième version Proposée

L'algorithme précédent n'est pas tout à fait efficace, pour l'améliorer on utilise une méthode nommé projection de Rayleigh Ritz afin d'approximer les valeurs propres recherchées.

Explication de la projection de Rayleigh Ritz

Comme auparavant on possède une matrice A, et une autre V de dimension n dont les vecteurs sont orthogonaux. On calcule ensuite $H = V^T AV$, en appliquant la décomposition spectrale. En effet nous obtenons directement les valeurs propres de H qui sont ceux de A mais il est toujours nécessaire de calculer le produit HX pour les vecteurs propres associés.

Amélioration de l’algorithme

En réalité la matrice A est symétrique puisque c'est une matrice de variance-covariance, on calcule après la matrice V comme dans l'algorithme précédent, puis après on génère la variance exprimée et on calcul V / V=AV, puis on orthogonalise les colonnes de cette matrice, et on applique le théorème de Rayleigh sur A et V.

2.3 Troisième version Proposée

Il existe une manière très simple d'améliorer la rapidité de l'algorithme précédent qui consiste à simplement remplace V=AV par $V=A^pV$.

3 Scénario de la prédiction

3.1 Data Analysis

On considère que l'on dispose au début de l'algorithme :
- d'une matrice F à deux dimensions : nt dimension de temps et ns dimension d'espace.
- d'un paramètre percentReached qui caractérise la variance que vont expliquer les EOFs que l'on garde et donc influencer leur nombre

On commence par afficher les données dynamiquement grâce au code fourni. Ensuite on calcul la matrice d'anomalie Z et la matrice de covariance $S = Z^T Z$. L'étape suivante consiste à calculer les paires propres de S et ce en utilisant : soit la fonction eig définie dans matlab et en choisissant ensuite le nombre de vecteurs propres que l'on garde grâce au percentReached, soit en faisant appel aux fonctions codées en fortran.
On trace ensuite les EOFs et leurs PCs, ceci nous permettra de vérifier que l'on observe bien des phénomènes physiques réels (type mode d'oscillation).

Vérification de la qualité des EOFs

On va vérifier d'abord la qualité de la base d'EOFs trouvée en calculant $||ZVV^{T-Z}||/||Z||$, celui-ci doit être petit. Ensuite, en prenant une autre base de vecteurs pris aléatoirement on montre que celle-ci est une plus mauvaise base par le même calcul.

3.2 Prédiction

On considère que la matrice de données que l'on avait au départ contenait n_y années de données. A celle-ci on enlève la dernière année et on calcule les EOFs avec la méthode vu précédemment.

Dans l'année restante, on considère que certaines colonnes de données sont inconnues et on les enlève. Pour cela nous avons choisi de demander à l'utilisateur d'entrer le pourcentage de colonnes que l'on enlève de la matrice. Des colonnes à ce nombre sont ensuite enlevées aléatoirement de la matrice. On a alors sur les colonnes de données restantes $F = \alpha V^T$ où F est la matrice des données connues sur la dernière année et V est la matrice des EOFs tronquée des données inconnues (à chaque colonne de données enlevée on retire la ligne correspondante dans l'EOF). Il faut trouver α : on se retrouve confronté à un problème des moindres carrés.
La prédiction consisiste alors à considérer que α est également valable sur les colonnes de la matrice que l'on ne connait pas à priori. On retrouve alors la matrice de données entière :
$F' = (V \alpha)^T$

Vérification de la qualité de la prédiction

Pour vérifier la qualité de la prédiction, on calcul $||G-F||/||G||$ où G est la vraie matrice de donnée sur la dernière année Ensuite, en prenant une autre base de vecteurs pris aléatoirement on montre que celle-ci est une plus mauvaise base par la prédiction.

4 Résultats et analyse

4.1 Résultats

Pour les tests qu'on a utilisé on a choisit de varier la variance expliquée, on propose alors trois variations, 50, 90 et 99 % puis ensuite varier le pourcentage des colonnes enlevées les tableaux suivants montrent les résultats obtenu :

Variance expliquée	Les erreurs
50 %	Erreur avec les vrais EOFs $4{,}825908.10^{-1}$ Erreur avec une base aléatoire 491,4866
90 %	Erreur avec les vrais EOFs $9{,}466513.10^{-2}$ Erreur avec une base aléatoire 891,2265
99 %	Erreur avec les vrais EOFs $3{,}109980.10^{-2}$ Erreur avec une base aléatoire $2{,}5967.10^{3}$

Le même tableau avec le pourcentage des colonnes enlevées

Variance expliquée	Pourcentage des colonees enlevées
50 %	10 % Erreur avec les vrais Eofs $1{,}018514.10^{-1}$ Erreur avec une base aléatoire $9{,}810843.10^{-1}$ 50 %Erreur avec les vrais Eofs $1{,}018582.10^{-1}$ Erreur avec une base aléatoire $9{,}808984.10^{-1}$ 90 %Erreur avec les vrais Eofs $1{,}023532.10^{-1}$ Erreur avec une base aléatoire $9{,}834924.10^{-1}$
90 %	10 %Erreur avec les vrais Eofs $3{,}645637.10^{-1}$ Erreur avec une base aléatoire $9{,}912027.10^{-1}$ 50 %Erreur avec les vrais Eofs $3{,}648280.10^{-2}$ Erreur avec une base aléatoire $9{,}886859.10^{-1}$ 90 %Erreur avec les vrais Eofs $3{,}662424.10^{-2}$ Erreur avec une base aléatoire $9{,}888489.10^{-1}$
99 %	10 % Erreur avec les vrais Eofs $2{,}153762.10^{-2}$ Erreur avec une base aléatoire $9{,}949053.10^{-1}$ 50 % Erreur avec les vrais Eofs $2{,}161430.10^{-2}$ Erreur avec une base aléatoire $9{,}958142.10^{-1}$ 90 % Erreur avec les vrais Eofs $2{,}219351.10^{-2}$ Erreur avec une base aléatoire 9,956648.10−1

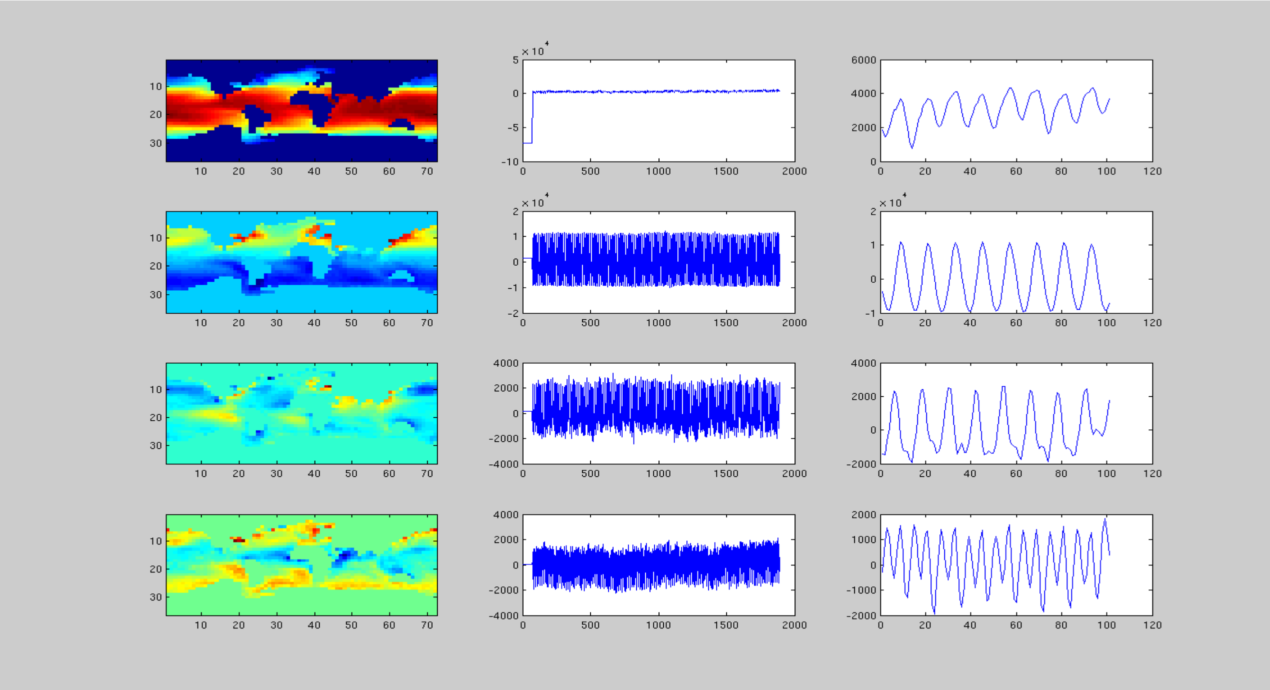

Image de la variance expliquée à 99 %

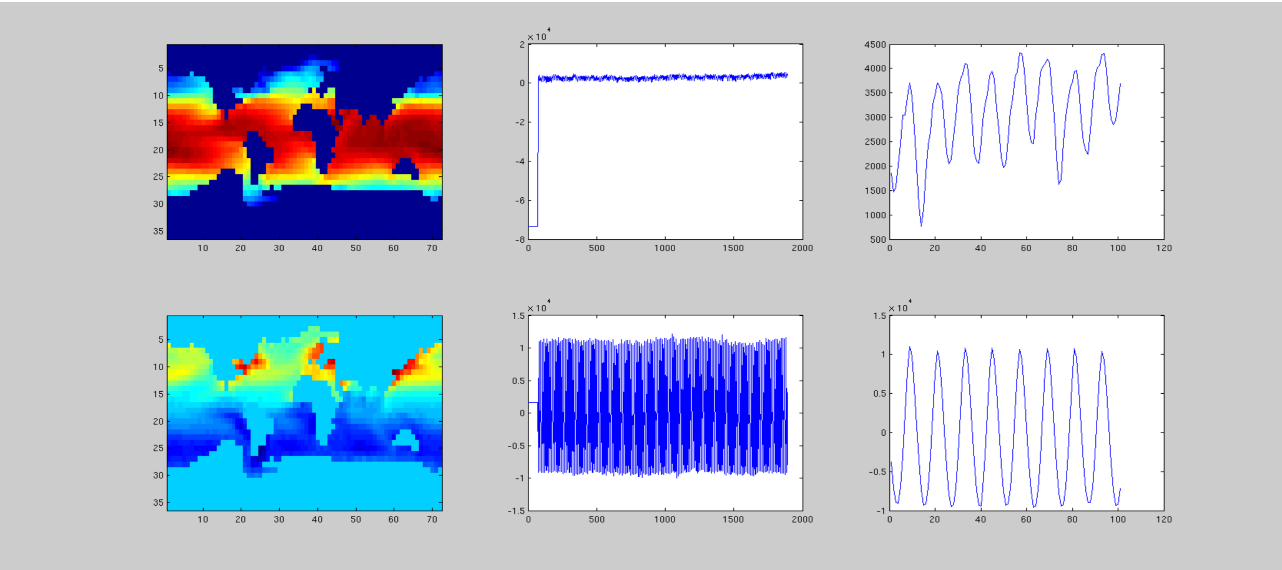

Image de la variance expliquée à 90 %

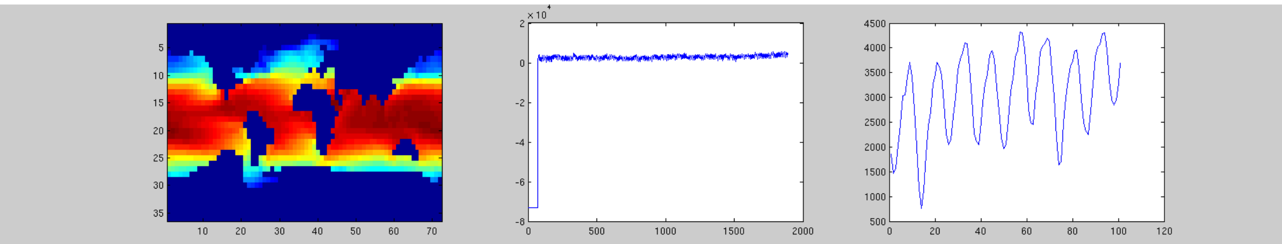

Image de la variance expliquée à 50 %

4.2 Analyse

Remarquons qu'il y a une différence entre l'erreur avec les vrais Eofs et l'erreur avec la base aléatoire : L'erreur avec une base aléatoire est énorme.

Lorsque l'on augmente la valeur de la variance ewpliquée, l'erreur diminue que ce soit au niveau de la qualité de la base ou la qualité de la prédiction. C'est normal car plus on explique la variance, plus les EOFs contiennent les « raisons » pour lesquelles la température des eaux varie, ainsi on obtient des données quasi exactes.

Lorsque l'on augmente le nombre de colonnes que l'on enlève de la dernière année, l'erreur augmente. Ce résultat est tout a fait logique, moins on dispose d'informations, plus la prédiction est inexacte.

Quand à l'analyse à partir de l'affichage des EOFs on vérifie :

- Que plus on se rapproche de l'équateur plus les températures sont chaude, ceci constitue la principal EOF, le principal motif de variation de la température de la mer.
- Que plusieurs autres motifs de variances sont observables, il faudrait les rapprocher de phénomènes réels : les courants océaniques ?
- Que globalement la température des océans augmente.

4 Conclusion

Même si nous avons trouvé que ce projet était difficile, notamment au niveau de la gestion du temps, celui-ci était intéressant dans le sens où on se rapproche de ce que peuvent faire des chercheurs en implantant une théorie, la testant et essayant d'analyser les résultats avec toujours un soucis d'optimisation.

Il reste cependant quelques points noirs à propos de notre travail, notamment au niveau de l'interface entre le fortran et le matlab, il faut l'avouer nous n'avons pas pu faire fonctionner EOF.m avec les programmes en fortran.

Ce projet nous a apporté de l'expérience quand à la programmation en fortran et matlab, languages que nous avons à peine effleuré au cours de l'année.