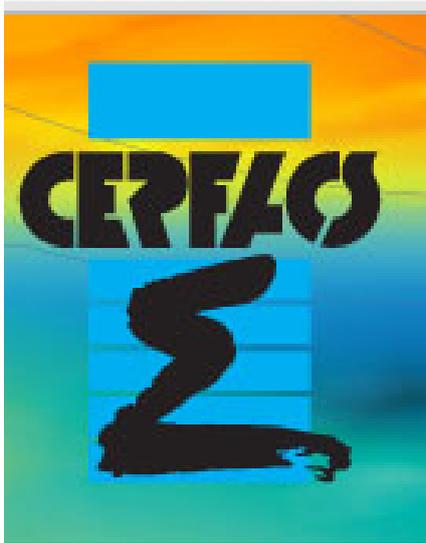


ÉCOLE NATIONALE DE LA MÉTÉOROLOGIE

CERFACS



Stage IENM 1 : rapport technique

Utilisation de la librairie ESMF pour calculer les poids d'interpolation dans OASIS-MCT

Mathieu Fructus

Encadrants :
Christophe Cassou
Laure Coquart
Sophie Valcke

Juillet 2014

Remerciements

Merci beaucoup à toute l'équipe d'accueil et en particulier à Christophe Cassou, Laure Coquart et Sophie Valcke pour leur soutien et pour avoir accepté d'aménager mon emploi du temps. Cela m'a permis de pouvoir être présent auprès de ma mère quand cela a été nécessaire. C'était très important pour moi et je vous en remercie encore. Je remercie aussi Bernard Iche pour avoir validé ces changements, mais aussi pour avoir accepté un format de stage particulier : je souhaitais pouvoir apercevoir dès que possible ce à quoi pouvait ressembler un poste d'ingénieur de recherche. Merci pour avoir su entendre ce projet professionnel et pour m'avoir donné les moyens de l'explorer.

J'en profite aussi pour remercier les membres de l'équipe, permanents ou temporaires, qui ont bien voulu répondre à mes interrogations sur ce que pouvait être le métier d'ingénieur de recherche, le métier de chercheur, et les différences entre les deux. Merci aussi aux thésards et post-doc pour leurs improbables sujets de discussion à la Sodex.

Sommaire

1	Introduction	2
2	Détermination du détail de l'algorithme HPR	4
2.1	Une introduction : quadrature des fonctions réelles	4
2.2	Quelques mots sur la méthode SPR pour les éléments finis	5
2.3	L'algorithme HPR de l'ESMF	5
3	Comparaison des méthodes d'interpolation Bilin et HPR	8
3.1	Algorithme général	8
3.1.1	Calcul de l'erreur d'interpolation	8
3.1.2	Calcul de l'erreur d'interpolation pour le gradient	8
3.1.3	Idée générale de l'algorithme de calcul d'erreur pour une seule méthode	8
3.1.4	Algorithme final	9
3.2	Choix des fonctions connues à tester	10
3.3	Résultats obtenus	12
3.3.1	BGGD vers LMDT	12
3.3.2	BGGD vers LMDZ	13
3.3.3	LMDT vers BGGD	13
3.3.4	LMDT vers LMDZ	14
3.3.5	LMDZ vers BGGD	14
3.3.6	LMDZ vers LMDT	15
3.3.7	LMDZ vers ORCA025	15
3.4	Limites de mon travail	16
4	Conclusion	18

Chapitre 1

Introduction

Le CERFACS a commencé à développer, dès 1991, le logiciel coupleur OASIS (Ocean, Atmosphere, Sea Ice, Soil). Ce logiciel est destiné à l'échange de données entre programmes. Par exemple, si un premier programme modélise l'atmosphère et un second programme modélise l'océan, OASIS va leur permettre d'échanger des informations, typiquement sur l'évolution des conditions aux limites dans ce cas-là. Le coupleur ne se limite pas à l'échange strict des données. Il peut effectuer des calculs intermédiaires et, en particulier, déterminer des interpolations entre maillages différents. Pour reprendre l'exemple, il n'y a aucune raison pour que les points de grille des deux programmes soient les mêmes. Les données calculées par le premier programme sur sa grille ("grille source") vont être interpolés par OASIS pour définir des valeurs sur la grille du second programme ("grille cible"). Cette problématique est illustrée sur la figure 1.1. Il s'agit de savoir si l'on peut donner une valeur "naturelle" sur les losanges (qui constituent les points de maille de la grille cible) à partir des valeurs sur une grille source représentée ici par les cercles.

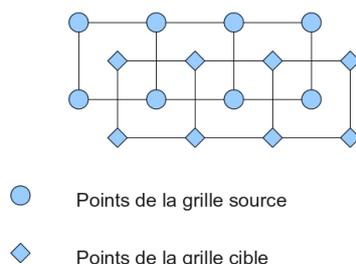


FIGURE 1.1 – *Exemple de grille source et cible*

Actuellement, le coupleur OASIS en est à la version OASIS3-MCT où la partie OASIS est développé par le CERFACS et le CNRS alors que la partie MCT (Model Coupling Toolkit) est développé par l'Argonne National Laboratory (laboratoire de recherche dépendant de l'Université de Chicago, USA).

Dans le coupleur OASIS3-MCT, plusieurs choix de méthodes sont possibles ("Plus Proches Voisins à Poids", "Interpolation Bilinéaire", "Interpolation Bicubique", "Interpolation Conservative"). La question se pose de développer d'autres propositions de méthodes d'interpolation.

En particulier, un logiciel de couplage différent, ESMF (Earth System Modeling Framework), développé aux USA, propose la méthode "High Order Patch Recovery". Mon projet de stage porte sur cette méthode d'interpolation. Plus précisément, j'ai effectué deux tâches successives :

- comprendre (puis expliquer !) le détail de l'algorithme utilisé dans cette méthode d'interpolation.
- Effectuer une comparaison entre la méthode "High Order Patch Recovery" (HPR dans la suite) et la méthode "Interpolation Bilinéaire" (Bilin dans la suite)

Chapitre 2

Détermination du détail de l'algorithme HPR

Le travail fourni dans cette partie a essentiellement consisté en du travail bibliographique : en effet, la méthode HPR développée par l'ESMF est une adaptation au couplage de données de la méthode SPR (Super-convergent Patch Recovery) pour les éléments finis. Même si, au final, il n'est pas nécessaire de connaître la méthode SPR, j'ai dû passer du temps dessus pour comprendre les idées en jeu dans la méthode HPR pour le couplage.

2.1 Une introduction : quadrature des fonctions réelles

Lorsque l'on recherche à approcher la valeur d'une intégrale, on voit souvent une approximation du type $\int_{-1}^1 f(t)dt \approx \sum_{i=0}^N w_i f(x_i)$. Tout le problème consiste alors à déterminer les poids w_i et les points d'interpolation x_i pour que l'approximation soit la meilleure possible. Plus précisément, par "meilleure méthode", on désire obtenir une formule qui soit exacte pour des polynômes de degré le plus élevé possible.

La première idée, probablement la plus naturelle, est de prendre des points d'interpolation x_i équidistants. Ce sont les méthodes de "Newton-Cotes" (contenant entre autres les méthodes des Trapèzes, méthodes de Simpson, etc...). Dans ce cas, prendre N points d'interpolation permet d'obtenir une méthode exacte jusqu'aux polynômes de degré N (ou $N - 1$ en fonction de la parité), autrement dit, du même ordre de grandeur.

L'évolution des techniques mathématiques a permis de faire un lien entre les méthodes d'interpolation et l'analyse hilbertienne : on peut exprimer l'approximation de l'intégrale d'une fonction à l'aide de polynômes orthogonaux pour des produits scalaires bien choisis. Il est alors possible d'obtenir, à l'aide des racines des polynômes de Legendre $L_n(x) = ((x^2 - 1)^n)^{(n)}$, des points d'interpolation x_i et des poids w_i . Dans ce cas, prendre N points d'interpolation permet d'obtenir une méthode exacte jusqu'aux polynômes de degré $2N - 1$. Ceci permet d'obtenir un gain d'ordre quadratique sur l'erreur d'interpolation. C'est ce phénomène de "super-convergence" qui est utilisé dans les méthodes de type SPR pour les éléments finis.

Cette méthode est appelée "méthode de Gauss-Legendre" (en France) ou bien "méthode de Gauss". Les points d'interpolation x_i sont appelés "Points de Gauss".

Les détails mathématiques de cette introduction sont développés dans [3, Chapitre II et III].

2.2 Quelques mots sur la méthode SPR pour les éléments finis

Les méthodes de type éléments finis consistent en une résolution approchée d'une équation aux dérivées partielles sur un domaine. Pour cela, on effectue une discrétisation du domaine pour permettre une résolution numérique. Pour contrôler la qualité de la solution tout en optimisant les ressources numériques, on cherche à raffiner la grille de discrétisation de manière adaptative. En effet, on cherche à obtenir une erreur uniformément distribuée et contrôlée. Si l'erreur est trop importante, on modifie la grille en déplaçant et/ou en rajoutant des noeuds. Une des méthodes possibles est la méthode des patch de recouvrement SPR.

Si j'ai bien compris :

- Il s'agit de déterminer une estimation de l'erreur entre la solution discrète calculée sur un maillage et la solution exacte (que l'on ne connaît pas). Pour cela, on lisse la solution discrète en effectuant une interpolation polynomiale. Plus précisément, on définit des régions (d'où l'appellation "Patch") qui se recouvrent partiellement (d'où l'appellation "Recovery"). Sur chaque région, on interpole par minimisation de type "moindre carré" au niveau des points de Gauss. Enfin, on fait la moyenne des valeurs obtenues sur chaque région pour avoir une valeur lissée. C'est cette valeur lissée qui va permettre de déterminer où il est le plus intéressant d'effectuer le raffinement.
- De plus, les points de Gauss vont être les noeuds du maillage raffiné. Le choix de ces noeuds permet d'obtenir une meilleure convergence de la solution approchée vers la solution exacte sur le noeud initial N que ce que l'on pourrait attendre pour un raffinement contenant le même nombre de points. C'est ce gain de convergence qui justifie l'appellation de "Superconvergent". Cette superconvergence a été démontrée pour certains cas particuliers de maillage, mais n'a pas été démontrée en toute généralité.

Il est possible d'obtenir plus de détail sur cette méthode dans la thèse de Martin Kempeneers [4, page 49 et suivantes] ou celle de Ramzy Boussetta [2, page 77 et suivantes]. Je me dois également de remercier Alexandre Bérard pour les qualités pédagogiques de sa thèse [1] : j'ai ainsi eu un éclairage rapide, compréhensible et bienvenu sur les transferts de champs entre maillages.

La page web http://www.si.ens-cachan.fr/accueil_V2.php?page=affiche_ressource&id=116 donne aussi des informations sur la méthode SPR (elle porte plutôt sur la maîtrise de l'erreur de discrétisation dans les méthodes d'éléments finis)

2.3 L'algorithme HPR de l'ESMF

On considère deux grilles. Une grille source (S) et une grille cible (C).

1. Pour un point M de la grille C , on détermine ses plus proches voisins V_i dans la grille S comme dans la figure 2.1
2. Pour chacun des voisins V_i de S :

- (a) Déterminer un "patch", c'est à dire un voisinage (dans S) centré autour du voisin V_i (pour le point V_1 , on a la figure 2.2)
- (b) Calculer (par interpolation bilinéaire) les valeurs en les points de Gauss de chaque sous-structure du patch
- (c) Déterminer un polynôme d'interpolation (de degré 2) sur tout le patch. Il est obtenu par minimisation (au sens des moindres carrés) des valeurs aux points de Gauss

On dispose ainsi, pour chaque voisin V_i , d'un polynôme P_i d'interpolation.

3. La valeur au point M de la grille C est alors obtenue en moyennant les valeurs des polynômes P_i au point C. (c'est une moyenne pondérée, elle est pondérée avec les mêmes poids que ceux de la méthode bilinéaire).

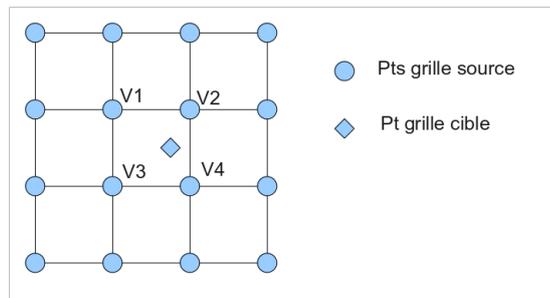


FIGURE 2.1 – *Les voisins de grille source d'un point cible*

J'ai utilisé une présentation de David Neckels du 12 mars 2008 dans la liste des présentations de l'ESMF [5] et dans une moindre mesure la présentation de Robert Oehmke [6]. De plus, j'ai pu avoir un échange de mails avec Robert Oehmke pour obtenir certaines précisions.

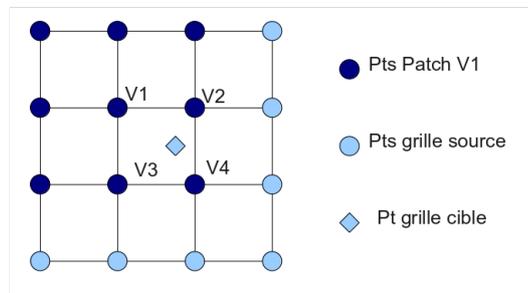


FIGURE 2.2 – *Le Patch du voisin V1*

Chapitre 3

Comparaison des méthodes d'interpolation Bilin et HPR

3.1 Algorithme général

3.1.1 Calcul de l'erreur d'interpolation

L'idée générale est d'utiliser une fonction connue f . En effet, on peut ainsi calculer cette fonction sur la grille source avant d'effectuer l'interpolation f_i sur la grille cible. Il est alors possible de déterminer l'erreur d'interpolation : il suffit pour cela de comparer l'interpolation obtenue avec les valeurs de la vraie fonction sur la grille cible. Plus précisément, j'ai calculé une erreur relative pour chaque point (θ, ϕ) de la grille cible : $\left| \frac{f(\theta, \phi) - f_i(\theta, \phi)}{f(\theta, \phi)} \right|$. Il a bien entendu fallu choisir des fonctions ne s'annulant pas pour éviter une explosion de l'erreur relative pour des raisons numériques (division par zéro).

3.1.2 Calcul de l'erreur d'interpolation pour le gradient

La démarche générale est exactement la même. Ceci dit, les phénomènes géophysiques étant périodiques, il est naturel dans les tests d'introduire des fonctions ondulatoires de type cosinus et sinus. Or leur gradient peut s'annuler. De plus, pour vérifier que le gradient est bien interpolé, il faut s'assurer que chacune des composantes du vecteur gradient l'est.

Ainsi, d'une part l'algorithme calcule les composantes en latitude et en longitude du vecteur gradient ; d'autre part, pour le calcul d'erreur relative, on prend l'expression $(\|\nabla_{\text{lat}} f + 0.001\|)$ (réciproquement $(\|\nabla_{\text{lon}} f + 0.001\|)$ pour l'autre composante) pour éviter toute explosion numérique non significative quand le gradient s'annule.

Au niveau du numérateur, les termes en 0.001 s'annulent. Ainsi, il n'y a pas de biais supplémentaire dans le calcul d'erreur relative.

3.1.3 Idée générale de l'algorithme de calcul d'erreur pour une seule méthode

1. Calculer fonction sur la grille source
2. Calculer fonction sur la grille cible
3. Calculer l'interpolation de 1) vers la grille cible

4. Calculer l'erreur entre les résultats 2) et 3)

Il faut faire de même pour le gradient dans la direction de latitude et pour le gradient dans la direction de longitude. On obtient ainsi 12 étapes dans l'algorithme de calcul d'erreur pour une seule méthode.

3.1.4 Algorithme final

Notre objectif est de comparer les méthodes bilin et HPR. Il faut donc effectuer ces deux interpolations avant de les comparer. Ainsi, dans l'algorithme général, les étapes 3-4, 7-8 et 11-12 s'appliquent deux fois : une fois avec l'interpolation bilinéaire et une fois avec l'interpolation High-order Patch Recovery. On détermine enfin la différence entre les erreurs relatives de chaque méthode : c'est-à-dire que l'on détermine l'erreur absolue entre les deux erreurs relatives. J'ai choisi de séparer cette erreur absolue en deux termes : "gain" quand l'erreur relative HPR est plus faible (d'au moins 1% pour éviter les "bruits numériques") que l'erreur relative bilin ; "perte" quand l'erreur relative HPR est plus forte que l'erreur relative bilin (d'au moins 1% pour éviter les "bruits numériques").

On obtient ainsi un algorithme général de la forme :

1. Calculer la fonction sur la grille source
2. Calculer la fonction sur la grille cible
3. Déterminer les deux gradients (lat et lon) sur la grille cible
4. Prendre $|\text{grad}|+0.001$ pour éviter des annulations du gradient (lat et lon)

5. Interpolation bilin de 1) sur la grille cible
6. Déterminer l'erreur relative entre l'interpolation bilin (5) et la fonction sur la grille cible (2)
7. Calcul du gradient (lat et lon) de l'interpolation bilin (5)
8. Prendre $|\text{grad}|+0.001$ pour éviter des annulations du gradient (lat et lon)
9. Déterminer l'erreur relative pour les gradients (ie entre gradient (8) par rapport au gradient (4))

10. Interpolation HPR de 1) sur la grille cible
11. Déterminer l'erreur relative entre l'interpolation HPR (5) et la fonction sur la grille cible (2)
12. Calcul du gradient (lat et lon) de l'interpolation HPR (5)
13. Prendre $|\text{grad}|+0.001$ pour éviter des annulations du gradient (lat et lon)
14. Déterminer l'erreur relative pour les gradients (ie entre gradient (8) par rapport au gradient (4))

15. Calcul des gains (ie HPR meilleur que bilin d'au moins 1%) pour fonction et gradient (on compare (11) et (6), ainsi que (14) et (9))

16. Calcul des pertes (ie HPR moins bon que bilin d'au moins 1%) pour fonction et gradient (idem)
17. Ecrire tous ces résultats dans des fichiers.

3.2 Choix des fonctions connues à tester

Pour comparer les deux interpolations, j'ai choisi de tester des fonctions dans trois catégories différentes :

- Des fonctions standards (de type polynomial : linéaire, quadratique, cubique, ordre 4)
- Des fonctions adaptées à ce qui pourrait ressembler à un champ géophysique (fonctions issues d'une documentation SCRIP)
- Des fonctions mathématiques au comportement plus chaotique, adaptées aux problèmes d'optimisation (beaucoup d'oscillations et beaucoup de minimums locaux)

Plus précisément, j'ai choisi les fonctions suivantes (avec θ pour la longitude et ϕ pour la latitude)

1. $P_1(\theta, \phi) = 20 + 0.05 * \phi$
 $P_2(\theta, \phi) = 22 + 0.05 * \phi - 0.05 * 0.04 * \phi^2 - 0.03 * \theta + 0.02 * 0.08 * \theta^2 - 0.06 * 0.005 * \theta * \phi$
 $P_3(\theta, \phi) = 500000 + 0.03 * \phi^2 * \theta - 0.02 * \theta^3$
 $P_4(\theta, \phi) = 1 + 0.003 * \phi^2 * \theta^2 + 0.002 * \theta^4$
2. $S_1(\theta, \phi) = 2 - \cos(\pi * \arccos(\cos(\phi) * \cos(\theta)))$
 $S_2(\theta, \phi) = 2 + \cos(\theta)^2 * \cos(2\phi)$
 $S_3(\theta, \phi) = 2 + \sin(\theta)^{16} * \cos(16 * \phi)$
3. $f_{\text{rastrigin}}(\theta, \phi) = 22 + \left(\frac{\phi}{10}\right)^2 - 10 * \cos\left(\pi * \frac{\phi}{5}\right) + \left(\frac{\theta}{20}\right)^2 - 10 * \cos\left(\pi * \frac{\theta}{10}\right)$ ainsi qu'une fonction $f_{\text{rastrigin}}$ modifié où les coefficients devant les cosinus valent 1 plutôt que 10 (pour atténuer les oscillations). Le graphe est sur la figure 3.1
 $f_{\text{eggholder}}(\theta, \phi) = 1000 - (6\phi + 47) * \sin\left(\sqrt{\|6\phi + 1.5\theta + 47\|}\right) - 3 * \theta * \sin\left(\sqrt{\|3\theta - (6\phi + 47)\|}\right)$.
 Le graphe est sur la figure 3.2
 $f_{\text{griewank}}(\theta, \phi) = 2 + \frac{\phi^2 + \theta^2}{4000} - \cos(\phi) * \cos\left(\frac{\theta}{\sqrt{2}}\right)$. Le graphe est sur la figure 3.3

Ces fonctions sont testées à l'aide de quatre grilles différentes :

- une grille standard (LMDZ)
- une grille de même taille dont les noeuds sont décalés par rapport à LMDZ : c'est la grille LMdT
- une grille de résolution 2 fois plus importante dans chaque coordonnées : BGGD
- Enfin, une grille de très haute résolution ORCA025 (8 fois la résolution BGGD environ pour la latitude comme pour la longitude)

Les six possibilités d'appariement "grille source - grille cible" sont testés pour les trois premières grilles : Il s'agit de voir les effets de la liaison entre des grilles de même taille ainsi qu'entre des grilles de taille différente. De plus, j'ai aussi traité les 2 appariements "LMDZ-ORCA025" et "BGGD-ORCA025" pour voir s'il y a gain (ou pas) pour la méthode HPR quand il y a beaucoup

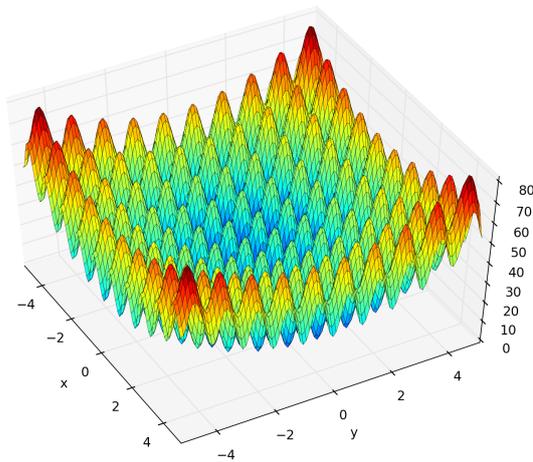


FIGURE 3.1 – Graphe de la fonction Rastrigin

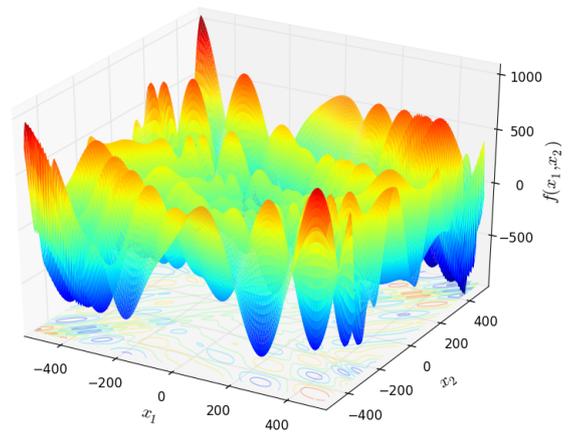


FIGURE 3.2 – Graphe de la fonction Eggholder

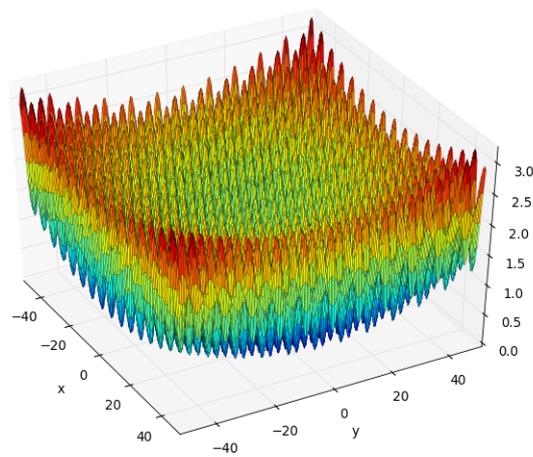


FIGURE 3.3 – Graphe de la fonction Griewank

de points cibles entre les points sources.

3.3 Résultats obtenus

Un problème important de l'ensemble des tests est d'arriver à synthétiser l'information : il y a 11 fonctions différentes, pour lesquelles on s'intéresse à la fois à la fonction, au gradient de latitude et au gradient de longitude. Ainsi, pour une unique configuration "grille cible - grille source", il y a 33 informations concernant la méthode HPR et 33 concernant la méthode Bilin. Enfin, comme le but est de comparer ces méthodes, il faut faire 33 comparaisons. Ainsi, pour chaque configuration "grille cible - grille source", il faut faire apparaître 99 informations. C'est sous la forme de tableau qu'il m'a paru le plus pertinent d'indiquer les résultats. Pour chaque configuration, on a un tableau avec 11 lignes (une pour chaque fonction) et 9 colonnes. Les 3 premières colonnes correspondent à la qualité d'interpolation de la méthode Bilin : TB si l'erreur est inférieure à 1 % pour l'ensemble des points de grille, B si l'erreur est comprise entre 1 et 5 %, si l'erreur est supérieure à 5 %, rien n'est indiqué dans la case correspondante. On a besoin de 3 colonnes puisque l'on fait la comparaison pour la fonction, le gradient latitude, le gradient longitude. Les 3 colonnes suivantes (colonne 4, 5 et 6) donnent les mêmes informations, mais pour la méthode HPR. Enfin, les 3 dernières colonnes indiquent s'il y a perte ou gain entre les deux méthodes (gain si HPR est meilleur, perte si Bilin est meilleur).

Il est important de comprendre que si une fonction est très bien interpolée par les deux méthodes, il ne peut y avoir ni gain ni perte : les deux méthodes conviennent pour l'interpolation entre grilles.

Les notations des fonctions mathématiques sont :

M1 : Rastrigin modifié (ie avec oscillations amorties)

M2 : Rastrigin

M3 : Eggholder

M4 : Griewank

3.3.1 BGGD vers LMDT

La grille source possède beaucoup plus de points que la grille cible : on s'attend donc à ce que l'interpolation soit bonne. Du coup, comme les deux méthodes vont bien comprendre les fonctions testées, on ne voit que peu de différence entre les méthodes (ie peu de gain et peu de perte).

Grille Source BGGD et Grille Cible LMDT									
Fction	Bilin			HPR			Gain ou Perte		
	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo
<i>P1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>P2</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>P3</i>	<i>TB</i>	<i>B</i>	<i>B</i>	<i>TB</i>				<i>Perte</i>	<i>Perte</i>
<i>P4</i>									<i>Gain</i>
<i>S1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S2</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S3</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>M1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>M2</i>		<i>B</i>	<i>B</i>		<i>B</i>	<i>B</i>			
<i>M3</i>									
<i>M4</i>		<i>B</i>	<i>B</i>		<i>B</i>	<i>B</i>			

3.3.2 BGGD vers LMDZ

Là aussi, la grille source possède beaucoup plus de points que la grille cible. On s'attend encore à ce que l'interpolation soit bonne. Ceci dit, le changement de grille cible amène des différences et la qualité générale d'interpolation est moins bonne. En particulier, il y a des phénomènes de bord qui apparaissent dans cette configuration : on peut avoir par exemple un gain sur le bord et une perte au centre (typiquement pour les fonctions mathématiques).

Grille Source BGGD et Grille Cible LMDZ									
Fction	Bilin			HPR			Gain ou Perte		
	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo
<i>P1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>P2</i>			<i>TB</i>			<i>B</i>	<i>Gain</i>	<i>Gain</i>	<i>Perte</i>
<i>P3</i>								<i>Gain</i>	<i>Perte</i>
<i>P4</i>								<i>Gain</i>	
<i>S1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S2</i>		<i>TB</i>	<i>TB</i>		<i>TB</i>	<i>TB</i>			
<i>S3</i>		<i>TB</i>	<i>TB</i>		<i>TB</i>	<i>TB</i>	<i>Perte</i>		
<i>M1</i>			<i>TB</i>			<i>TB</i>	<i>Gain</i>	<i>Gain</i>	
<i>M2</i>			<i>B</i>			<i>B</i>			
<i>M3</i>									
<i>M4</i>			<i>B</i>			<i>B</i>			

3.3.3 LMDT vers BGGD

La grille source possède moins de points que la grille cible. C'est normalement là que l'on attend une différence entre les méthodes, en faveur de la méthode HPR. Ce n'est pourtant pas le cas, et on observe aussi des phénomènes de bord : fonction (ou gradient) bien moins compris sur le bord.

Grille Source LMDT et Grille Cible BGGD									
Fction	Bilin			HPR			Gain ou Perte		
	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo
<i>P1</i>	<i>B</i>	<i>TB</i>	<i>TB</i>	<i>B</i>	<i>TB</i>	<i>TB</i>			
<i>P2</i>							<i>Gain</i>		
<i>P3</i>									
<i>P4</i>									
<i>S1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S2</i>		<i>B</i>	<i>B</i>		<i>B</i>	<i>B</i>			
<i>S3</i>		<i>B</i>	<i>B</i>		<i>B</i>	<i>B</i>			
<i>M1</i>									
<i>M2</i>									
<i>M3</i>									
<i>M4</i>							<i>Perte</i>		

3.3.4 LMDT vers LMDZ

Les deux grilles possèdent un nombre similaire de points mais sont décalées. Il n'y a pas d'a priori sur la qualité d'une méthode par rapport à une autre. On va cependant observer, comme précédemment, des phénomènes de bord.

Grille Source LMDT et Grille Cible LMDZ									
Fction	Bilin			HPR			Gain ou Perte		
	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo
<i>P1</i>	<i>B</i>	<i>TB</i>	<i>TB</i>	<i>B</i>	<i>TB</i>	<i>TB</i>			
<i>P2</i>							<i>Gain</i>		
<i>P3</i>							<i>Perte</i>		
<i>P4</i>									<i>Perte</i>
<i>S1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S2</i>		<i>B</i>	<i>B</i>		<i>B</i>	<i>B</i>	<i>Perte</i>		
<i>S3</i>		<i>B</i>	<i>TB</i>		<i>B</i>	<i>B</i>	<i>Gain</i>		
<i>M1</i>									
<i>M2</i>							<i>Perte</i>		
<i>M3</i>							<i>Perte</i>		
<i>M4</i>							<i>Perte</i>		

3.3.5 LMDZ vers BGGD

De nouveau, une grille source avec beaucoup moins de points que la grille cible. On s'attend à du gain pour la méthode HPR. Ce sera très légèrement le cas (et il y a aussi des phénomènes de bord).

Grille Source LMDZ et Grille Cible BGGD									
Fction	Bilin			HPR			Gain ou Perte		
	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo
<i>P1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>P2</i>								<i>Gain</i>	
<i>P3</i>									
<i>P4</i>									
<i>S1</i>	<i>B</i>	<i>TB</i>	<i>TB</i>	<i>B</i>	<i>TB</i>	<i>TB</i>			
<i>S2</i>		<i>B</i>	<i>TB</i>		<i>B</i>	<i>TB</i>			
<i>S3</i>		<i>B</i>	<i>TB</i>		<i>B</i>	<i>TB</i>			
<i>M1</i>									
<i>M2</i>								<i>Gain</i>	
<i>M3</i>							<i>Gain</i>		
<i>M4</i>								<i>Gain</i>	

3.3.6 LMDZ vers LMDT

De nouveau, des grilles de taille semblable. Dans ce sens, les fonctions interpolées sont mieux comprises. De manière surprenante, on va plutôt observer des pertes.

Grille Source LMDZ et Grille Cible LMDT									
Fction	Bilin			HPR			Gain ou Perte		
	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo
<i>P1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>P2</i>	<i>B</i>	<i>TB</i>	<i>B</i>		<i>TB</i>	<i>B</i>			
<i>P3</i>								<i>Perte</i>	<i>Perte</i>
<i>P4</i>							<i>Perte</i>	<i>Perte</i>	<i>Perte</i>
<i>S1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S2</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S3</i>	<i>B</i>	<i>TB</i>	<i>TB</i>	<i>B</i>	<i>TB</i>	<i>TB</i>			
<i>M1</i>	<i>B</i>	<i>TB</i>	<i>TB</i>	<i>B</i>	<i>TB</i>	<i>TB</i>			
<i>M2</i>			<i>B</i>				<i>Perte</i>		
<i>M3</i>							<i>Perte</i>		
<i>M4</i>			<i>B</i>				<i>Perte</i>		

3.3.7 LMDZ vers ORCA025

Dans cette simulation, la grille source contient beaucoup moins de points que la grille cible (en ordre de grandeur, 10 fois moins dans chaque coordonnées). C'est dans cet exemple que l'on espère voir apparaître un gain pour la méthode HPR. Il n'en est rien :

Grille Source LMDZ et Grille Cible ORCA 025									
Fction	Bilin			HPR			Gain ou Perte		
	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo	<i>Fct</i>	∇la	∇lo
<i>P1</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>P2</i>							<i>Perte</i>		
<i>P3</i>									
<i>P4</i>							<i>Perte</i>		
<i>S1</i>	<i>B</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>	<i>TB</i>			
<i>S2</i>	<i>B</i>	<i>TB</i>	<i>TB</i>		<i>TB</i>	<i>B</i>			
<i>S3</i>		<i>TB</i>	<i>B</i>		<i>TB</i>	<i>B</i>			
<i>M1</i>	<i>B</i>								
<i>M2</i>									
<i>M3</i>									
<i>M4</i>									

3.4 Limites de mon travail

- La limite la plus importante dans mon travail consiste probablement dans le peu de grilles testées. Sur des grilles à meilleure résolution, a priori, l'approximation bilinéaire est meilleure, d'où un moindre intérêt pour des méthodes d'interpolation plus élaborées. Ceci dit, il aurait pu être intéressant de quantifier le gain (ou pas) de la méthode HPR sur des grilles plus fines. Plus précisément, on aurait pu comparer le comportement des deux méthodes sur des grilles cibles et sources de taille similaire, en faisant varier la résolution (ie faire l'étude LMDZ-LMDT puis BGGD-BGGDT, puis ORCA025-ORCA025T et aussi pour une grille de taille intermédiaire entre BGGD et ORCA025). On pourrait ainsi voir l'évolution du gain en fonction de la résolution.
- Aucune étude n'a été faite non plus sur la "géométrie" des grilles : on aurait pu par exemple tester la variation de gain pour une méthode donnée quand on change la répartition des points de grille. Dans d'autres domaines où les interpolations mathématiques interviennent (traitement d'images par exemple), il vaut mieux prendre "plus de points vers les bords" et en mettre moins au centre. Ceci dit, les points de grille sont choisis dans les modélisations et il n'est peut-être pas très réaliste d'envisager des méthodes où ces points sont modifiés.
- Un autre point délicat est l'aspect subjectif de l'appréciation globale de gain ou de perte pour la comparaison des deux méthodes d'interpolation. Il faudrait trouver un critère plus fin qu'une simple moyenne. En effet, prendre la moyenne des gains ou des pertes n'aurait que peu de sens pour évaluer la qualité d'une méthode. Il faut trouver une formule de pondération faisant intervenir à la fois l'erreur-relative mais aussi le pourcentage de la surface où elle s'applique. Il doit être possible de fabriquer un critère à partir des 5 données suivantes : Surface relative de gain et moyenne de gain sur cette surface, surface relative où les 2 méthodes sont équivalentes, surface relative de perte et moyenne de perte sur cette surface.
- Une autre limite, moins gênante sur le fond, est la "naïveté" du code NCL que j'ai développé. Au fur et à mesure de l'avancement du stage, j'ai testé de plus en plus de fonctions, ce qui finit par générer un code avec beaucoup de "copier-coller". Une approche itérative indexée sur le nom des fonctions paraît plus pertinente. Vu le temps imparti pour le stage, je ne vois pas comment j'aurais pu à la fois avancer sur la comparaison "Bilin vs HPR" et

reprendre l'architecture du code. Du coup, le code n'est guère élégant (mais il a l'avantage d'être facilement compréhensible).

- Enfin, le choix d'ajouter 0.001 au gradient pour éviter les explosions numériques est arbitraire. Je n'ai pas vérifié l'influence de ce terme (qui n'a d'importance qu'aux points où le gradient s'annule, et il n'est pas apparu de phénomènes particulier en ces points là).

Chapitre 4

Conclusion

Contrairement à ce que j'attendais, la méthode HPR ne me paraît pas amener de gain véritable sur la qualité d'interpolation entre grille cible et grille source par rapport à la méthode bilinéaire. D'une certaine manière, soit les fonctions sont bien comprises par les deux méthodes et il n'y a pas de gain (en particulier, c'est le cas pour les fonctions géophysiques), soit les fonctions sont mal comprises par les deux méthodes (typiquement les fonctions tourmentées de type Rastrigin-Griewank) et dans ce cas, les gains comme les pertes sont entremêlés dans le domaine cible, répartis avec un aspect aléatoire.

Lorsque j'observe que même dans le cas "grille source LMDZ - grille cible ORCA 025", les fonctions géophysiques sont bien comprises avec la méthode d'interpolation bilinéaire, j'ai le sentiment qu'il n'est pas nécessaire d'aller chercher des algorithmes plus complexes et qu'il vaut mieux concentrer les moyens de calcul ailleurs.

Bibliographie

- [1] Alexandre Bérard, *Transferts de champs entre maillages de type éléments finis et applications numériques en mécanique non linéaire des structures*, thèse soutenue le 16 septembre 2011 à l'Université de Franche-Comté
- [2] Ramzy Boussetta, *Estimateurs d'erreur et remaillage adaptatif : application à la simulation 3D des procédés de mise en forme des matériaux*, thèse soutenue le 18 mars 2005 à l'École des Mines de Paris
- [3] Jean-Pierre Demailly, *Analyse Numérique et Equations Différentielles*, EDP Sciences, 2006 (troisième édition)
- [4] Martin Kempeneers, *Eléments finis statistiquement admissibles et estimation d'erreur par analyse duale*, thèse soutenue le 31 mars 2006 à l'Université de Liège
- [5] David Neckels, présentation disponible à l'adresse
http://www.earthsystemmodeling.org/presentations/pres_0803_neckels.pdf
- [6] Robert Oehmke, présentation disponible à l'adresse
http://www.earthsystemmodeling.org/presentations/pres_1109_see_regridding.ppt