

Rapport de projet

Denis, Théo (27411800)

Résumé—Etude des performances et de l'exactitude d'un solveur bande dans le contexte de la simulation d'un moteur à réluctance variable.

I. INTRODUCTION :

Le but de ce projet est de simuler un moteur à réluctance variable. Pour ce faire, il faut appliquer les équations de Maxwell. Après développement des équations et d'hypothèses simplificatrices[2], nous obtenons l'équation suivante à résoudre :

$$\frac{\partial}{\partial x} \left(\frac{1}{\mu} \frac{\partial a}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{1}{\mu} \frac{\partial a}{\partial y} \right) + j_s = 0 \quad (1)$$

Où a est la valeur du potentiel magnétique du potentiel vecteur $\mathbf{A} = (0,0,a)$. Cette équation est un problème de diffusion où le terme source est la densité de courant j_s .

La section II abordera le choix du solveur et ses motivations. Nous étudierons la validité des résultats fournis par le solveur dans la section III. Les sections IV, V, VI et VII seront consacrées à l'analyse des performances et des optimisations faites. Finalement, la section VIII conclura le document.

II. CHOIX DU SOLVEUR

Le choix du solveur pour ce type de problème est crucial car la majorité du temps de calcul du programme est pris par la résolution du système d'équation. Une étude à l'aide de perf[3] a permis d'obtenir les pourcentages de temps d'exécution¹ suivant :

%	Fonction
93.52	motorComputeMagneticPotential
3.74	motorAdaptMesh
0.35	motorComputeCouple

Les fonctions `motorComputeCurrent` et `MotorFree` ne prennent pas assez de temps de calcul pour en tirer un pourcentage et le reste des pourcents sont des appels en tout genre.

Le choix s'est rapidement tourné vers un solveur bande. Le solveur plein n'est pas envisageable, le temps de calcul serait bien trop grand pour les plus gros maillage. Une autre possibilité serait les gradients conjugués pour faire un solveur itératif. Toutefois, leur performance sans préconditionneur est en dessous du solveur bande. De plus, le solveur bande offre des résultats précis.

1. Pour 1000 itérations sur `motor400.txt`

III. VALIDITÉ DU SOLVEUR :

Comme vu au cours[1], un moteur à réluctance variable possède la courbe couple en fonction de l'angle du rotor illustrée à la figure 1 si nous souhaitons maximiser le couple.

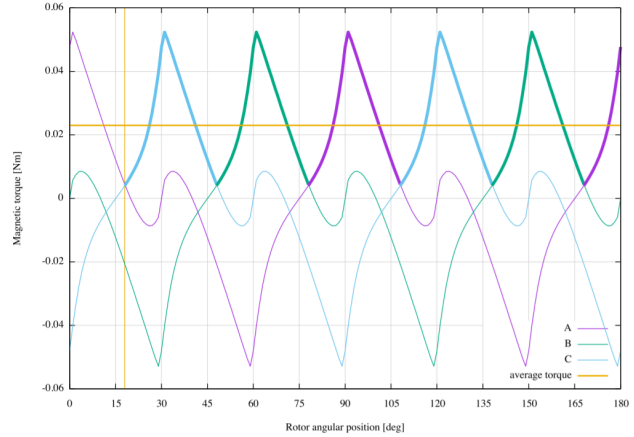


FIGURE 1: Couple en fonction de l'angle du rotor

En choisissant de faire le switch entre les bobines tous les 30° , le programme produit le résultat² fourni à la figure 2.

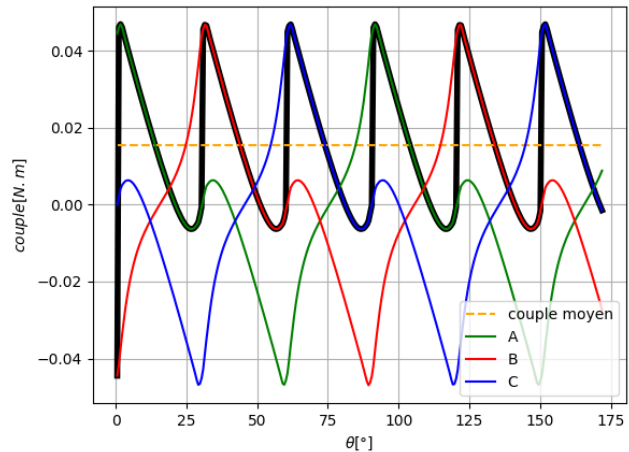


FIGURE 2: Couple en fonction de l'angle du rotor, asynchrone

Comme nous pouvons l'observer à la figure 2, le basculement des bobines n'est pas synchronisé. Il faut appliquer un décalage à la position de référence ou décider d'éteindre et allumer une paire sur un autre ensemble d'angle. Dans le code, nous choisissons la première option. Le couple obtenu

2. Pour le maillage `motor4424.txt`

est illustré à la figure 3. Nous pouvons y observer la justesse du moteur.

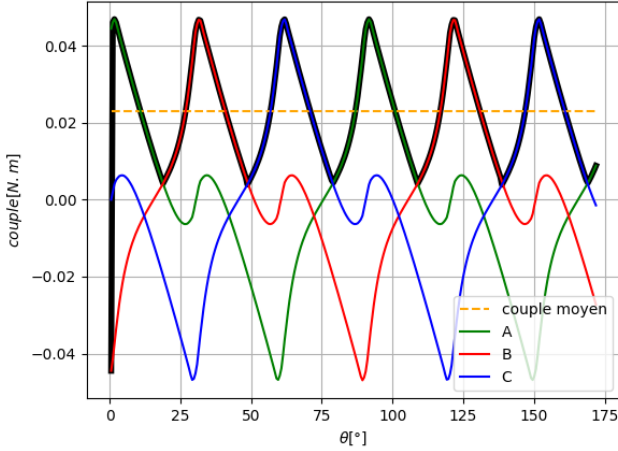


FIGURE 3: Couple en fonction de l'angle du rotor, synchronisé

Si nous prenons l'équation de rotation du rotor :

$$I \frac{d\omega}{dt} = C \quad (2)$$

L'inertie I est un terme constant. Par conséquent, à couple moyen constant, la vitesse angulaire augmente linéairement avec le temps. Nous pouvons l'observer à la figure 4 pour les différents maillages.

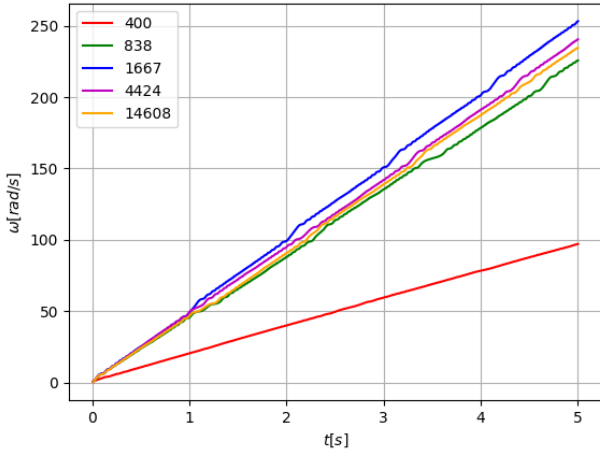


FIGURE 4: Vitesse angulaire en fonction du temps

La solution semble converger puisque les variations s'atténuent avec l'augmentation de la taille des maillages. Nous le remarquons plus particulièrement pour les maillages 4424 et 14608. Prendre des maillages avec plus de noeuds permettraient de se rapprocher davantage de la réalité. A cet égard, nous observons que le maillage avec 400 noeuds possède un résultat médiocre et éloigné des autres.

IV. PRÉCISION DES RÉSULTATS :

Puisque le programme respecte la physique et les équations, passons à l'étude de la précision des résultats. Nous supposons

que le maillage le plus grand donne la solution exacte du problème. Nous étudierons l'erreur des autres maillages sur base du maillage 14608.

La grandeur que nous étudions est le couple à la première itération puisque cette itération se fait dans les mêmes conditions pour chaque maillage. L'itération suivante est impactée par le déplacement du moteur qui dépend de la valeur du couple.

L'erreur relative pour chaque maillage est représentée à la figure 5. Un meilleur calcul d'erreur serait de faire l'erreur L2 qui demande une interpolation et des calculs plus lourds.

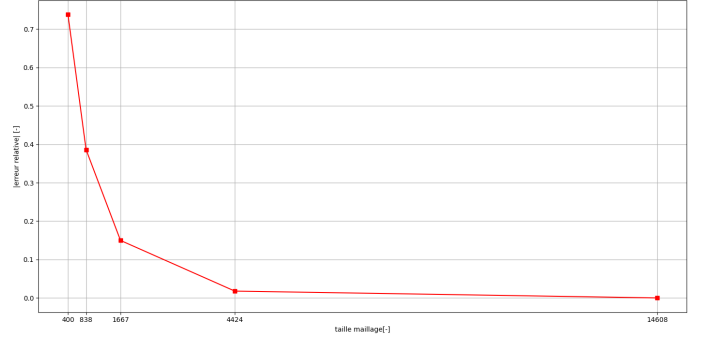


FIGURE 5: Erreur absolue relative du couple par rapport au maillage 14608

V. ORDRE DE LA MÉTHODE

Le temps pris par chaque itération³ reste une constante qui diffère en fonction de la taille du maillage comme nous pouvons l'observer à la figure 6.

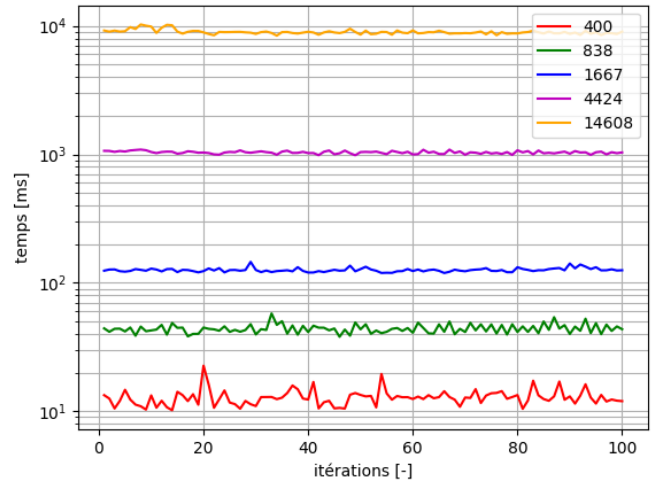


FIGURE 6: Log10 du temps pris par chaque itération

Si nous appelons w la largeur de bande, et n la taille de la matrice, la complexité temporelle de la méthode est $O(w^2n)$ qui est inférieur au solveur plein $O(n^3)$. Le solveur bande tient son avantage d'une bonne renumérotation qui permet de diminuer la taille de la bande w .

3. Pour une renumérotation en Y

VI. IMPACT DE LA RENUMÉROTATION

La renumérotation constitue une optimisation du solveur bande. Elle permet de diminuer la taille de la bande ce qui accélère le temps de calcul. Nous pouvons faire la moyenne des temps sur 100 itérations pour chaque maillage et comparer les 3 renumérotations(X, Y, Aucune).

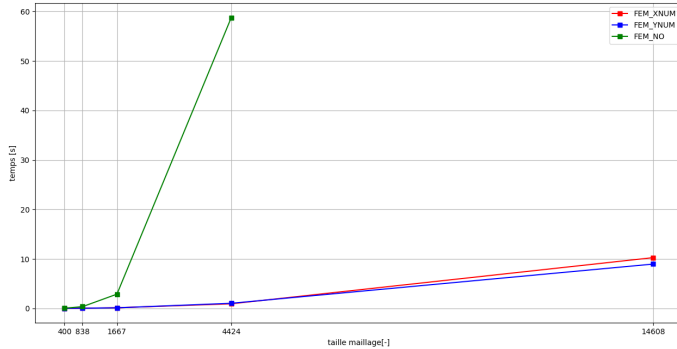


FIGURE 7: Temps moyen de calcul pour différente renumérotation

La renumérotation qui offre la plus grande rapidité sur les gros maillage est la renumérotation en Y. Elle est similaire à celle en X sur les petits maillages. Par contre, sans renumérotation, le solveur bande perd tous son intérêt.

VII. OPTIMISATION

Dans la fonction `motorComputeMagneticPotential`, plutôt qu'allouer à chaque passage dans la fonction les structures `femIntegration`, `femEdges`, ... Nous pouvons les initialiser uniquement lors de la première itération. La seule structure que nous devons réinitialiser est `femSolver` puisque le maillage aura tourné et la matrice doit être réassemblée. Ainsi, nous avons un gain de temps et de mémoire.

Dans la fonction `motorComputeCouple`, la seule optimisation consiste à ne calculer qu'une seule fois la valeur de `d` pour éviter la redondance des calculs à chaque itération.

Dans la fonction `motorAdaptMesh`, il a fallu changer de métrique. Le jacobien sur le périmètre ne fonctionnait pas pour le plus petit maillage. Il suffit de changer la métrique en un jacobien sur le périmètre au cube pour pénaliser les triangles avec un long périmètre.

VIII. CONCLUSION

En conclusion, le programme est capable de simuler avec justesse le comportement d'un moteur à réluctance variable. Cependant, il est plutôt lent pour de gros maillages. Un solveur itératif tel que gradient conjugué préconditionné écrit avec un système creux pourrait accélérer grandement les performances du code.

BONUS

Si vous souhaitez une présentation du projet, voici le lien d'une vidéo youtube le présentant : <https://youtu.be/YOmhyIKwCog>

REMERCIEMENTS

Remerciements à Jérôme Eertmans, mon tuteur qui m'a conseillé durant ce projet. Mais aussi à Louis Deville pour son travail combiné avec Jérôme pour sortir des notebooks de qualité.

RÉFÉRENCES

- [1] François HENROTTE. *Introduction aux machines électriques et à leur modélisation*. URL : <https://perso.uclouvain.be/vincent.legat/documents/epl1110/annotated-2021-slides/epl1110-cours10-machineselectriques.pdf> (visité le 08/05/2020).
- [2] Jean-François REMACLE. *Cours du 2 mars 2021 : Magnétostatique linéaire et non linéaire*. URL : <https://perso.uclouvain.be/vincent.legat/documents/epl1110/annotated-2021-slides/epl1110-cours9-magnetostatique.pdf> (visité le 09/05/2020).
- [3] UBUNTU. *Ubuntu Manpage :perf - Performance analysis tools for linux*. URL : <http://manpages.ubuntu.com/manpages/bionic/man1/perf.1.html> (visité le 08/05/2020).