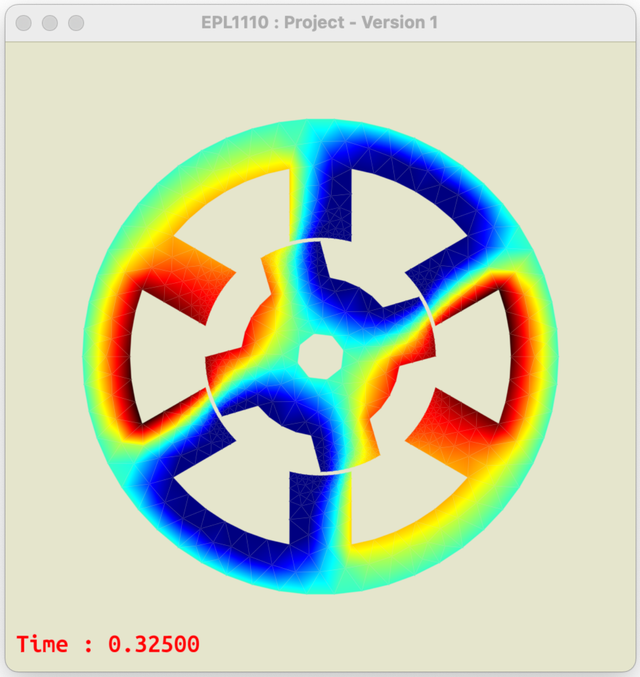
Simulation d'un moteur à reluctance variable par la méthode des éléments finis



L'objectif du projet est de vous initier aux difficultés de la mise au point et de la certification d'une application numérique. Rassurez-vous : il ne s'agit nullement de vous transformer en experts de l'architecture de grandes applications numériques ! Nous nous limiterons à l'écriture d'un tout petit programme C pour simuler le comportement d'un moteur électrique *Switched Reluctance Motor*.

1 Notre modèle mathématique :-)

1.1 Calcul du potentiel magnétique : une petite équation de Poisson...

A chaque instant, nous allons résoudre une équation de poisson pour obtenir le potentiel magnétique ou plus exactement de l'unique composante $A_z = a$ de ce potentiel vecteur $\mathbf{A}$. En d'autres mots, on fait l'hypothèse que le problème est bi-dimensionnel : ce qui est évidemment une approximation, mais c'est une approximation parfaitement raisonnable pour faire du dimensionnement et comprendre la physique du problème. Nous allons aussi supposer que les courants induits sont négligeables : ce qui est aussi une simplification acceptable car on utilise des tôles feuilletées pour éviter la création de grandes boucles de courants induits et limiter les pertes, comme Jean-François Remacle vous l'a expliqué.

Il faut donc résoudre le problème suivant sur la géométrie complète du moteur.

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

On imposera une valeur nulle $a = 0$ sur le rayon extérieur du stator et sur l'arbre du rotor qui n'a pas été maillé. Cela consiste à imposer que $\mathbf{B} \cdot \mathbf{n} = 0$ ou que la frontière est isolée d'un point de vue magnétique. Il n'a pas de flux de fuite. Nous n'avons pas maillé l'arbre du rotor car le maillage a été générée par une duplication symétrique et que cela nons ennuyait un peu de le mailler. On suppose également qu'il est isolé d'un point de vue magnétique et on impose la même constante $a = 0$ sinon, même en l'absence de tout courant, on observerait une induction magnétique [1]

Le terme des courants source j_s dans le stator et la valeur de la perméabilité dépendent évidemment du sous-domaine considéré du maillage. Pour obtenir la solution, on utilisera des triangles linéaires continus et on obtiendra donc la solution numérique :

$$a^h = \sum_{i=1}^{n} A_i \tau_i(x, y)$$

où A_i sont les n valeurs nodales associées à chaque sommet du maillage fourni.

Le maillage fourni comprend une série de sous-domaines associés au rotor, au stator, à l'espace entre le rotor et le stator, à l'espace vide entre les dents du rotor et finalement aux zones d'induction ou bobines traversées par le bobinage en cuivre dans le stator (*coils* en anglais). Les termes sources j_s avec le signe adéquat n'existeront que dans ces zones dans lesquels circulent un courant électrique à l'instant considéré. La perméabilité magnétique relative des inducteurs en cuivre est vraiment très proche de la valeur unitaire de celle de l'air. On utlisera donc $\mu = \mu_0$ dans tous les sous-domaines autres que les parties en acier du rotor et du stator où la perméabilité sera celle de l'acier doux $\mu = \mu_r\ \mu_0$ où μ_r est la perméabilité relative de l'acier doux [2].

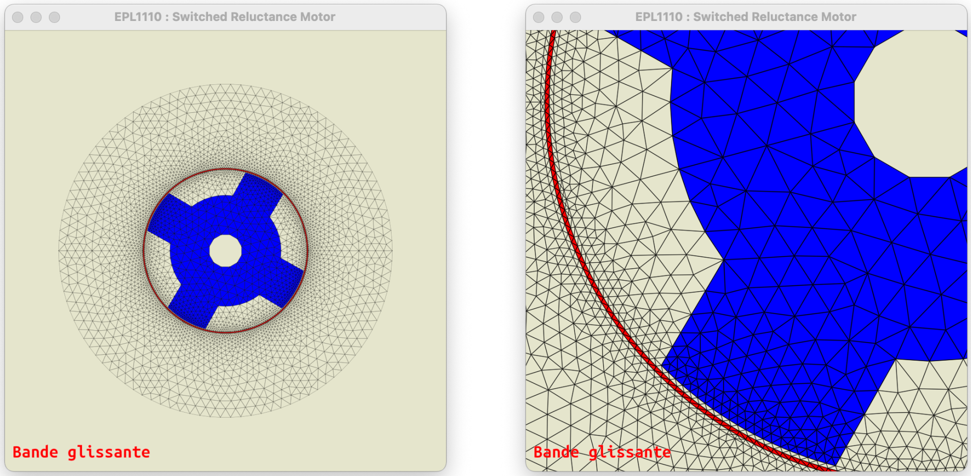
Cette toute première partie est assez élémentaire et ne devrait pas poser de difficultés majeures. Il faut toutefois veiller à obtenir la solution de la manière la plus précise, la plus efficace et rapide possible.

1.2 Mais le rotor tourne...

La seconde étape du projet consistera à faire tourner le rotor. Dans cette optique, un sous-domaine particulier a été défini dans le maillage, il s'agit d'une couronne d'éléments reliant des sommets situés sur deux cercles proches. Pour un angle de rotation donné du rotor alors que les éléments du stator restent immobiles, nous allons évidemment observer une déformation inacceptable de ces éléments. Il faudra donc de supprimer les éléments de cette bande intermédiaire et en recréer de nouveaux lorsqu'on effectue une rotation du maillage.

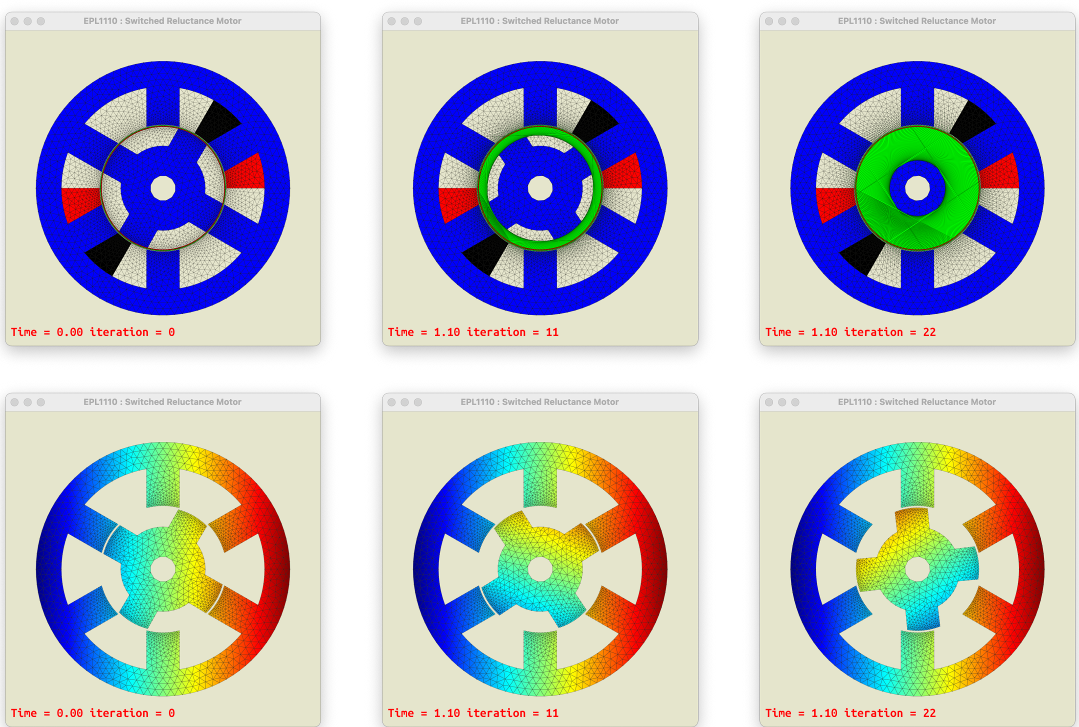
[1] Si vous aviez trouvé cela tout seuls, Jean-François Remacle prétend que c'était une question à 20 euros. Avec 350 inscrits au cours, cela risque de lui coûter cher. Je ne vais pas mettre de smileys car il estime que j'en abuse : bon, on se refait pas à mon âge :-)

[2] Oui : la perméabilité relative du cuivre est proche de celle de l'air : `https://fr.wikipedia.org/wiki/Permeabilite_magnetique` Et pourtant, j'étais pas convaincu a priori :-) On découvre des choses tous les jours !



En d'autres mots, il s'agira de mailler cette zone intermédiaire afin d'obtenir les triangles les plus proches possibles de triangles équilatéraux pour que la simulation par éléments finis soit la plus précise possible. La qualité d'un élément triangulaire Ω_e sera défini par le rapport entre le jacobien et le périmètre du triangle. Un jacobien négatif sera le signe d'un élément qui a été retourné, un jacobien nul sera le cas d'un élément totalement plat. A périmètre constant, le jacobien sera maximal pour un triangle équilatéral :-)

Il s'agira donc d'implémenter une méthode robuste et efficace pour mailler la bande glissante : on supposera toutefois qu'il sera toujours possible d'avoir un maillage conforme avec les sommets définis sur les deux cercles délimitant la bande glissante et qu'il ne sera jamais nécessaire de devoir ajouter des sommets. C'est la partie purement informatique du projet !



Dans l'ébauche du projet, on effectue la rotation du rotor sans remailler la bande glissante qui fait apparaître des éléments totalement déformés :-) La valeur de l'inconnue a été initialisée comme l'abscisse des coordonnées initiales de points : on peut donc bien voir la rotation du rotor.

Mon ami François m'explique ici qu'utiliser le terme bande glissante pourrait faire croire que les parties fixe et mobile glissent l'une par rapport à l'autre et que le raccord de continuité est effectué par une méthode de multiplicateurs de Lagrange. J'aurais donc dû plutôt parler de bande de mouvement ou de bande à remailler. Comme je n'ai pas envie de refaire les figures : nous allons donc conserver mon appelation erronée pour le moment. On est donc bien d'accord : la méthode présentée réalise le raccord entre la partie mobile et fixe du maillages, en remaillant cette bande intermédiaire à chaque itération temporelle.

1.3 Calcul du couple qui fait tourner le rotor !

La troisième partie du projet consistera à calculer le couple qui s'exerce sur le rotor Ω_r produit par la force de Lorentz induite par le champ d'induction magnétique créé par les courants qui passe dans les bobines du rotor.

$$C = L \int_{\Omega_r} \mathbf{r} \times \underbrace{\boldsymbol{\nabla} . \boldsymbol{\sigma}}_{\mathbf{f}} \; d\Omega$$

où $\mathbf{r}$ est le vecteur position et $\mathbf{f}$ la densité de force de Lorentz. Dans l'air ou le vide, l'effet local de la force de Lorentz peut s'exprimer sous la forme de la divergence du tenseur de Maxwell $\mathbf{f} = \boldsymbol{\nabla} . \boldsymbol{\sigma}$ avec ce dernier défini comme suit

$$\boldsymbol{\sigma} = \frac{1}{\mu_0}\mathbf{B}\mathbf{B} + \frac{2}{\mu_0}(\mathbf{B} \cdot \mathbf{B}) \; \boldsymbol{\delta}$$

où $\mathbf{B} = \boldsymbol{\nabla} \times \mathbf{A} = (a_{,y}, -a_{,x}, 0)$ et $\boldsymbol{\delta}$ représente le tenseur identité. Evidemment, on m'objectera que le rotor est en acier et que ce n'est pas le vide. Mais, nous allons observer que dans la suite, nous allons appliquer cette relation dans l'air qui entoure notre rotor et donc que c'est bien cette relation que nous allons utiliser. Dans un matériau quelcpnq

On observe immédiatement que calculer la divergence d'un rotationnel du champ $a(x, y)$ ne sera pas évident avec une représentation linéaire par morceaux de ce champ. Heureusement, nous disposons du théorème de la divergence et nous pouvons remplacer l'intégrale de volume de la divergence du tenseur de Maxwell par une intégrale du tenseur de Maxwell contracté avec la normale sur la surface extérieure du rotor et écrire :

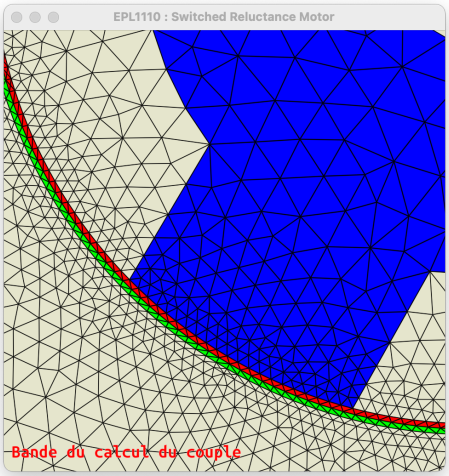
$$C = L \int_{\partial\Omega_r} \mathbf{r} \times (\boldsymbol{\sigma} \cdot \mathbf{n}) \; ds$$

où $\mathbf{n}$ est la normale sortante. Ici, il n'est plus nécessaire de calculer la divergence du tenseur de Maxwell, mais un problème subsiste car nous devons estimer les deux dérivées partielles de a sur une courbe le long de laquelle, nous ne disposons que de la variation tangentielle.

C'est ici qu'intervient une idée particulièrement astucieuse que François Henrotte m'a expliqué avec beaucoup de patience[3] :-) On peut calculer le couple exercé en appliquant le théorème de la divergence

[3]Note interne : *Calcul du Couple - formulation A 2D par François Henrotte - 12 mars 2021*

sur la surface physique de la pièce mobile, mais également sur n'importe quelle surface fermée englobant la pièce mobile et de l'air ou tout autre milieu électromagnétiquement neutre. Considérons maintenant le domaine Ω_g qui est une région annulaire de largueur constante d entourant le rotor et qui représente le tiers de l'entrefer dans notre maillage.



On peut alors voir cette région Ω_g comme une famille de cylindres concentriques et appliquer notre calcul du couple à chacun de ces cylindres et ensuite effectuer une moyenne le long de la direction radiale des intégrales le long des cercles qui donnent toutes en théorie la même valeur. Et faire cette moyenne se fait tout simplement en effectuant une intégrale dans la direction radiale et en divisant le résultat par d. On obtient ainsi une expression du couple qui implique une intégrale de surface. Cette expression est nettement plus robuste et plus simple à implémenter dans un code d'éléments finis.

$$C = \frac{L}{d} \int_{\Omega_g} \mathbf{r} \times (\boldsymbol{\sigma} \cdot \mathbf{n}) \; d\Omega$$

Pour obtenir l'expression finale à implémenter pour notre moteur, il faut maintenant considérer les vecteurs de base orthonormés d'un système cylindrique $\mathbf{e}_r$, $\mathbf{e}_\theta$ et $\mathbf{e}_z$ et écrire les expressions qui apparaissent dans le calcul du couple. En notant que la normale est radiale $\mathbf{n} = \mathbf{e}_r$, on peut en déduire mécaniquement :

$$\begin{aligned}
\mathbf{B} &= B_r \; \mathbf{e}_r + B_\theta \; \mathbf{e}_\theta \\
\boldsymbol{\sigma} \cdot \mathbf{n} &= \frac{1}{\mu_0} \mathbf{B} \; B_r + \frac{2}{\mu_0} (\mathbf{B} \cdot \mathbf{B}) \; \mathbf{e}_r \\
\mathbf{r} \times (\boldsymbol{\sigma} \cdot \mathbf{n}) &= \frac{B_r \; r}{\mu_0} (\mathbf{e}_r \times \mathbf{B}) \;=\; \frac{B_r B_\theta \; r}{\mu_0} \mathbf{e}_z
\end{aligned}$$

On obtient finalement l'expression dite de Arkkio.

$$C = \frac{L}{\mu_0 d} \int_{\Omega_g} B_r \; B_\theta \; r \; d\Omega = -\frac{L}{\mu_0 d} \int_{\Omega_g} \frac{\partial a}{\partial \theta} \; \frac{\partial a}{\partial r} \;\; d\Omega$$

En pratique, on calculera B_x et B_y : ensuite, on fera un changement de base pour obtenir les composantes normale B_r et radiale B_θ de ce vecteur :-)

1.4 Et maintenant, l'équation de la rotation du rotor !

La dernière pièce du puzzle est d'écrire l'équation de la rotation du rotor :

$$I \, \frac{d\omega}{dt} = C$$

Ensuite, il s'agit d'effectuer le mouvement de rotation de la partie du rotor du maillage et de passer à l'itération temporelle suivante. Enfin, c'est presque terminé, il faut aussi implémenter le basculement du courant entre les 3 paires de bobines du notre moteur en fonction de la position angulaire du rotor. Et voilà, le modèle qu'il s'agit de construire dans le projet d'éléments finis.

Paramètres matériels

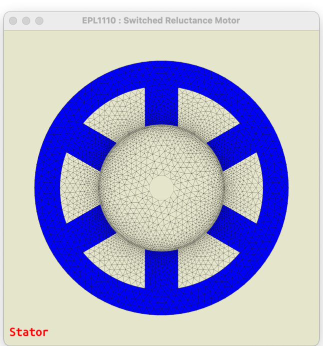
$$\boxed{\begin{aligned} j_s &= 8.8464\ 10^5\ [\mathrm{A\ m^{-2}}] \\ \mu_0 &= 4\ \pi\ 10^{-7}\ [\mathrm{kg\ m\ A^2 s^{-2}}] \\ \mu_r &= 1000 \\ I &= 5\ 10^{-4}\ [\mathrm{kg\ m^2}] \\ L &= 6\ [\mathrm{cm}] \end{aligned}}$$

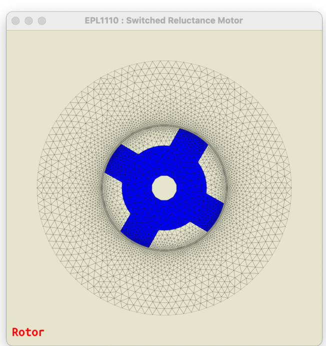
2 Ce que vous devrez réaliser !

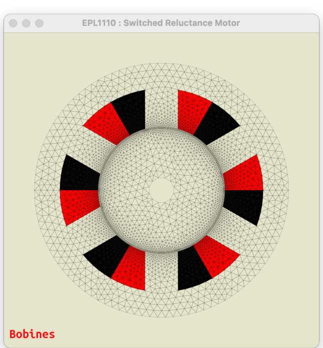
L'objet du projet est de réaliser un petit code d'éléments finis permettant de simuler un moteur électrique et de valider votre code en le confrontant à d'autres simulations et à l'expérience : il s'agit d'un petit laboratoire numérique où vous allez être confrontés à des comportements numériques un petit peu inhabituels et où il s'agira de les interpréter correctement : il y a peu à programmer et beaucoup à comprendre !

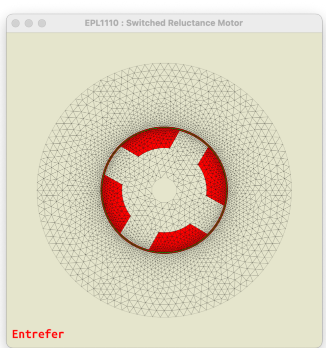
Tout d'abord, vous allez recevoir un maillage généré avec notre mailleur favori `gmsh`.
C'est un maillage d'éléments triangulaires avec 12 sous-domaines bien particuliers.

`Stator_core`	partie en acier formant le stator : so easy !
`Rotor_core`	partie en acier formant le rotor :-)
`Coil_AP`	partie de la paire A de bobines avec une densité de courant positive
`Coil_AN`	partie de la paire A de bobines avec une densité de courant négative
`Coil_BP`	partie de la paire B de bobines avec une densité de courant positive
`Coil_BN`	partie de la paire B de bobines avec une densité de courant négative
`Coil_CP`	partie de la paire C de bobines avec une densité de courant positive
`Coil_CN`	partie de la paire C de bobines avec une densité de courant négative
`Rotor_gap`	partie de l'entrefer proche du rotor
`Air_gap`	bande glissante de l'entrefer (la zone à remailler :-)
`Stator_gap`	partie de l'entrefer proche du stator
`Rotor_air`	zone entre les dents du rotor









Ensuite, nous avons créé deux structures élémentaires de données qui contiennent le maillage, les paramètres matériels et la solution courante du problème.

```
typedef struct {
    int *elem;
    double *X;
    double *Y;
    int nElem;
    int nNode;
    int nLocalNode;
    int nDomain;
    int *nElemDomain;
    string *nameDomain;
    int *domain;
} motorMesh;

typedef struct {
    int size;
    motorMesh *mesh;
    double *a;
    double time;
    double theta;
    double omega;
    int *movingNodes;
    double inertia;
    double L;
    double *js;
    double *mu;
    int nonLinearFlag;
    int nHystereticCurve;
    const double *hystereticCurveH;
    const double *hystereticCurveB;
} motor;
```

La structure `motorMesh` contient le maillage ainsi que les 11 sous-domaines caractérisés par un nom, un nombre d'éléments. On observera que tous les éléments des sous-domaines sont numérotés séquentiellement. Les premiers éléments appartienennent au premier sous-domaine et ainsi de suite...

La structure `motor` contient tous les paramètres matériels et géométriques, ainsi que le vecteur `theMotor->a` qui contient les valeurs nodales A_i du potentiel magnétique. Pour chaque sous-domaine, nous avons aussi la perméabilité magnétique et la densité de courant source. Une courbe entre **B** et **H** est également fournie pour la partie non-linéaire.

L'implémentation se fera avec des éléments triangulaires linéaires et un régle d'intégration de Hammer à trois points. Dans l'exemple fourni, on utilise un méthode d'Euler explicite pour calculer la nouvelle vitesse du rotor ω ainsi que la nouvelle position angulaire θ de celui-ci,mais on pourrait évidemment imaginer d'autres intégrateurs numériques pour cette équation. La position angulaire de référence est celle de la position du rotor dans les maillages fournis.

Votre projet consistera d'un unique fichier avec l'implémentation des quatre fonctions suivantes. Si vous avez développé votre projet avec plusieurs fichiers : il suffit de tous les concaténer afin d'obtenir un fichier unique[4]. L'objectif est évidemment de faciliter la tâche pour la correction automatique, mais aussi manuelle de votre code. Il est permis d'utiliser les fonctions fournies pour les devoirs précédents contenues en les incluant dans votre soumission, mais c'est aussi pour vous l'occasion de réfléchir si il n'existe pas d'autres manières pour obtenir une implémentation plus efficace et/ou plus élégante. Dans ce projet, il est permis et encouragé de développer votre créativité en termes de génie logiciel et autres. La seule contrainte est que votre projet doit être totalement autonome et ne pas utiliser d'autres librairies que celles prévues dans le commande de compilation du serveur.

```
void motorComputeMagneticPotential(motor *myMotor);
double motorComputeCouple(motor *myMotor);
void motorAdaptMesh(motor *myMotor, double dtheta);
void motorComputeCurrent(motor *myMotor, double theta);
void motorFree(motor *myMotor);
```

- La première fonction calculera le potentiel magnétique sur le maillage courant.
- La seconde fonction calculera le couple sur le maillage courant avec la solution courante.
- La troisième fonction effectuera une rotation d'un angle $\Delta\theta = \omega\Delta t$ du maillage courant et adaptera le maillage de la bande de glissement : il s'agira bien de supprimer tous les éléments de la bande de glissement et de les remplacer par de nouveaux éléments. Observer -au passage- toutefois que le nombre d'éléments restera constant.
- La quatrième fonction implémentera de manière virtuelle l'électronique de notre moteur en calculant dans quelle couple de bobine, le courant source circulera pour un position angulaire θ du rotor. Il s'agira ici de réfléchir à sélectionner de manière astucieuse les angles critiques pour faire le switch d'une bobine à l'autre. On considère en général des fonctions échelons, mais on pourrait aussi imaginer d'autres dispositifs pour contrôler le courant source dans le stator.
- La cinquième fonction désalloue l'espace mémoire de la structure `myMotor` ainsi que toutes les structures de données internes que vous avez alloué pour récupérer les résultats qu'il n'est pas nécessaire de recalculer à chaque itération. Typiquement, ce sont des structures de données que vous auriez conservés sous la forme de variables statiques dans votre fichier `motor.c`

Il vous est demandé :

1. De concevoir un programme permettant la simulation d'un moteur à reluctance variable des éléments triangulaires continus linéaires.
2. D'optimiser votre programme afin qu'il soit le plus rapide possible.
3. Le choix du pas de temps est fixé par nos soins.
4. De rédiger une note de synthèse d'au maximum 5 pages pour le Service de Magnétostatique appliquée Fournir une estimation de l'ordre de précision du résultat obtenu en expliquant comment vous avez validé votre code numérique. Produire quelques illustrations pertinentes pour l'analyse de la solution. Expliquer comment vous avez optimisé votre programme afin qu'il soit le plus rapide possible. Ne pas recopier les développements théoriques du syllabus, ne pas recopier l'énoncé du problème, ne pas fournir des diagrammes incompréhensibles, ne pas donner des tableaux de chiffres indigestes. L'orthographe, le soin et la présentation seront conformes à celles d'une note fournie par un bureau d'études professionnel.

[4] A cet égard, la commande unix `cat` est d'une efficacité redoutable :-)

(option) Effectuer une animation graphique originale de votre simulation en utilisant la librairie graphique BOV. Ce serait vraiment cool de voir les lignes de champs : c'est pas bien compliqué à implémenter pourtant !

(option) Analyser l'effet de la magnétostatique non linéaire

(option) Faire un podcast court sur votre projet sur youTube :-)

L'entièreté de votre code sera inclus dans un unique fichier `motor.c` qui sera compilé avec le programme `main.c` fourni. Il vous est loisible de reprendre tout ce que vous souhaitez dans les codes fournis dans les 8 devoirs. Votre programme doit toutefois être écrit en pur C et devra être compilé sur le serveur. Lors de la soumission, un calcul d'un nombre limité de pas de temps sera effectué afin de vous permettre d'estimer la rapidité de votre programme. Si vous avez effectué une implémentation graphique originale, vous pourrez soumettre également un projet complet avec un fichier `CMakeLists.txt` et des instructions de compilation précises. Pour qu'on puisse évaluer votre implémentation graphique, il est essentiel que les assistants puissent la compiler et il est donc impératif de n'utiliser que les dépendances fournies dans les devoirs.

Soyez particulièrement attentifs de citer toutes vos sources d'inspiration ! Il n'est pas interdit de s'inspirer d'autres codes, mais il est alors essentiel de citer correctement vos sources.

Toutes les soumissions seront soumises à un logiciel anti-plagiat. En cas de fraude flagrante, les cas de plagiat seront soumis au Jury des examens. Vous êtes invités à consulter la page web de l'Université pour avoir une petite idée des sanctions possibles dans ce cas !

2.1 Evaluation du projet

L'évaluation du projet se fera sur la base suivante sur un total de 20 points :

Programme qui fonctionne :-)	4
Précision des résultats	4
Rapidité du code de calcul	4
Style, soin et esthétique de l'implémentation	4
Rapport	4
Implémentation graphique	4
Analyse de la magnétostatique non-linéaire	4
Réaliser un podcast sur votre projet	4

Comme vous pourrez le constater, il est parfaitement possible d'obtenir 20/20 sans réaliser l'implémentation graphique. C'est donc bien une option qu'il n'est pas indispensable de réaliser. Pour vous aider, on vous a fourni un exemple de programme graphique permettant de faire tourner le moteur : c'est une version minimaliste de l'implémentation graphique : vous avez donc déjà une version de base pour le résultat optionnel...

2.2 Evaluation du cours

La note finale du cours sera pondérée comme suit. Un tiers est la note des devoirs, un tiers est la note du projet et le dernier tiers est la note de l'examen (sur le contenu théorique du cours, le projet et les exercices des séances). Il y a toutefois une contrainte supplémentaire : il faut impérativement réussir le projet ET l'examen pour bénéficier de la moyenne. La formule finale est donc

$$\begin{aligned} Note &= \max\left(\frac{Devoirs + Projet + Examen}{3}, \frac{Projet + Examen}{2}\right) && \text{si } Projet > 8 \text{ et } Examen > 8 \\ Note &= \min\left(\frac{Devoirs + Projet + Examen}{3}, Projet, Examen\right) && \text{si } Projet \leq 8 \text{ ou } Examen \leq 8 \end{aligned}$$

Ne pas faire le projet ne peut jamais être récupéré à l'examen !
Le projet de cette année-ci n'est pas particulièrement compliqué et le minimum syndical est vraiment simple à obtenir. Dans le but de laisser la réussite possible même pour les étudiants qui ont eu beaucoup de difficultés avec les devoirs, nous avons légèrement modifié la formule afin de ne pas trop pénaliser anticipativement des étudiants ayant une très mauvaise moyenne pour les 8 devoirs. Il est donc toujours encore possible de se rattraper même si vous n'avez encore rien fait à ce jour : ce qui n'était pas une très bonne idée, au passage.

3 Grand Prix International 2021 de l'Elément le Plus Fini

Le programme correct le plus rapide recevra le Grand Prix de l'Elément le Plus Fini d'un montant de 100 euros. Ce prix a été rendu possible grâce à la contribution anonyme d'un ancien étudiant soucieux de promouvoir la qualité de la formation. La proclamation des résultats se fera sur le web avant le 15 juin 2021. Les gagnants du prix seront invités à prendre contact avec les titulaires du cours. La mesure de la rapidité du programme sera effectuée par deux tests indépendants avec le maillage le plus raffiné avec l'option non-linéaire. Le résultat final sera la moyenne des deux mesures. En cas de timings aberrants ou manifestement parasites, des runs complémentaires seront effectués.

Les auteurs de l'implémentation informatique graphique la plus remarquable gagneront le prix spécial des assistants d'un montant de 50 euros. Cette seconde partie du concours est également ouverte à l'équipe didactique et à tous les membres de l'Ecole Polytechnique.

Sans oublier le prix spécial de la ville d'Ottignies-Louvain-la-neuve pour le plus beau podcast d'un montant exceptionnel de 50 euros.

Noter au passage que c'est largement supérieur aux prix que vous pouviez gagner lors des divers petits jeux de la Revue virtuelle des ingénieurs du 1er avril.

Version 2 du 30-4-2021 : le projet n'est pas un poisson d'avril :-)