

Développement d'un module de prémaillage automatique utilisant un réseau neuronal pour le logiciel MAGNET.

**Rapport de Diplôme d'Etude Approfondies
en Génie Electrique au Laboratoire d'Electrotechnique
de Grenoble**

Jean-Charles PAPAZIAN

Responsables :

**Prof. D.A. LOWTHER
D. DYCK**
de l'université Mc Gill de Montréal

J. BIGEON
du Laboratoire d'Electrotechnique
de Grenoble

Stage effectué au CAD Lab,
département de Génie Electrique
de l'université Mc Gill (Canada)
Septembre 1993

Remerciements :

Je tiens à remercier le professeur Lowther de m'avoir accepté dans son laboratoire, ainsi que Derek Dyck qui m'a lancé et aidé à faire mon projet.

Enfin je tiens à remercier Jean Bigeon qui lors de son passage à Montréal a soulevé des questions intéressantes sur mon projet lors de discussions enrichissantes.

Sommaire

Introduction : présentation du projet.....	3
I Description du module de prémaillage préexistant et modifications demandées.....	4
I.1. Insertion du module de prémaillage dans le logiciel.	
I.2 Travail demandé.	
II Présentation et algorithme du programme de traitement de données géométriques : main_skk.....	7
II. 1 Utilisation.	
II.2 Algorithme général.	
II.3 Structure de données utilisée.	
III Utilisation du réseau neuronal.....	14
III.1 Notions générales.	
III.2 Description du réseau neuronal de type "background propagation".	
III.2.1 Structure générale.	
III.2.2 Méthode numérique utilisée pour la phase d'entraînement.	
III.2.3 Décodage de la sortie.	
IV Représentation de la géométrie locale pour les entrées du réseau neuronal.....	23
IV.1 Critère de maillage idéal.	
IV.2 Problème de la mise à l'échelle.	
IV.2.1 Propriété d'homothétie du maillage.	
IV.2.2 Choix du facteur d'échelle.	
IV.3 Choix des entrées.	
V Essais en vue d'obtenir un module de prémaillage opérationnel.....	30
V.1 Conditions expérimentales.	
V.2 Considérations de base sur le maillage conduisant au choix des entrées.	
V.3 Essais conduisant les choix des entrées.	
V.4 Liste des entrées.	
V.5 Critères de choix des entrées.	
VI Résultats obtenus.....	36
VI.1 Modèle d'entraînement.	
VI.1.1 Prémaillage obtenu dans l'air.	
VI.1.2 Prémaillage obtenu dans le fer.	
VI.2 Modèle de référence inconnu pour le réseau neuronal.	
VI.2.1 Prémaillage obtenu dans l'air.	
VI.1.2 Prémaillage obtenu dans le fer.	
VI.3 Autres exemples.	
Conclusion.....	44

Annexe.....46
Bibliographie.....48

Introduction : présentation du projet.

Le but de mon travail été de perfectionner un module de prémaillage existant pour le logiciel de CAO MAGNET d'Infolytica.

Le prémaillage obtenu n'est pas, et ce pour plusieurs raisons, le maillage définitif du problème.

D'une part, ce qui est obtenu en sortie du module sont les densités relatives de maillage en certains points du plan de résolution du problème, requis par le générateur de maillage de MAGNET. Ces densités relatives sont ensuite transformées par le générateur de maillage en densités absolues dépendant du degré de précision que l'on désire sur les éléments finis.

D'autre part, ces différentes densités sont ensuite transformées en un maillage effectif par le générateur de maillage, avec lequel le solveur adaptatif de MAGNET calcule les solutions du problème. Si des erreurs sur le maillage sont détectées, des corrections sur celui-ci sont effectuées par le solveur, qui renvoie des densités de maillage au générateur et ce lors d'itérations successives.

Un prémaillage a l'aide d'un réseau neuronal fournit au solveur adaptatif un maillage correct mais néanmoins imparfait, et ce très économiquement en termes de temps de calcul. Ce maillage permet alors d'économiser des itérations très coûteuses pour obtenir le maillage définitif, par rapport à un maillage de départ uniforme (qui est utilisé actuellement par MAGNET). L'originalité de cette démarche repose sur le fait que le réseau neuronal est utilisé pour obtenir des valeurs numériques et non pas pour classifier, ce qui est généralement son utilisation.

Le module de prémaillage a été conçu pour ne résoudre dans un premier temps uniquement des problèmes de magnétostatique à flux fermés, avec des matériaux linéaires de type fer et cuivre (les aimants permanents ne peuvent être encore utilisés).

I Description du module de prémaillage préexistant et modifications demandées.

I 1. Insertion du module de prémaillage dans le logiciel.

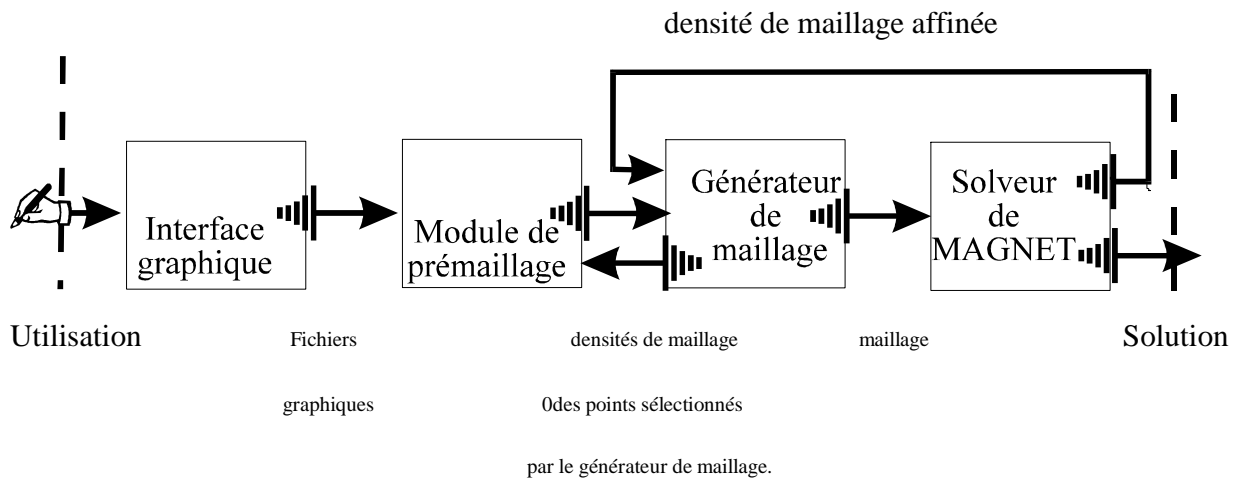
Des travaux de recherches ont été effectués dans le laboratoire de CAO de l'Université Mac-Gill [D. Dick 1990] et ont débouché sur l'élaboration d'un réseau neuronal de type "background propagation" qui permettait d'obtenir un prémaillage à partir de fichiers graphiques expérimentaux, ne comportant que des segments droits. Ceci représentait la première application des réseaux neuronaux au maillage automatique.

Lors de ce chapitre, nous assimilerons un réseau neuronal à une fonction à plusieurs entrées et à une sortie. Pour déterminer la taille des éléments finis en un point, les entrées du réseau neuronal sont la géométrie locale du problème autour de ce point. Il est clair que cette représentation est incomplète, et que seules les équations de Maxwell peuvent permettre de déterminer un maillage optimal.

Néanmoins, pour des problèmes ne comportant que des segments de droite, le prémaillage obtenu différait de moins de 15% avec le maillage obtenu par le solveur adaptatif de MAGNET, ce qui était bien supérieur à un simple prémaillage uniforme.

Mais à cause de sa représentation "rectiligne" de la géométrie, ce module n'est pas exploitable par MAGNET, car ses fichiers graphiques comportent des segments de droite et des arcs de cercle.

Mon travail a donc été d'élaborer un sous-module transformant les fichiers graphiques de la prochaine version de MAGNET en entrées décrivant la géométrie locale du problème en tout point d'une grille de taille d'éléments finis. Ainsi, le module de prémaillage peut être intégré dans MAGNET de manière effective (Cf fig I.1.a).



Architecture du logiciel MAGNET

Figure I.1.a

La représentation de cette géométrie pour le réseau neuronal a du être repensée en conséquence. En effet, la représentation des entrées des travaux précédents faisait appel aux distances et aux différents angles entrant en jeu entre les différents coins. Il est clair que cette représentation était plus valide avec des formes arrondies, sachant de plus que mon sous-module supprime les coins en les remplaçant par des arcs de cercles de courbure élevée (Ce choix sera détaillé plus tard).

Le module de prémaillage comporte à cet effet deux parties (Cf. fig . I.1.b) :

- transformation des fichiers géométriques de MAGNET en entrées "optimales" pour le réseau neuronal (en terme de représentation de la géométrie locale) et en chaque point de la grille.
- transformation des entrées déterminées précédemment en une taille d'élément fini (puis pour la version définitive en densité de maillage) par le réseau neuronal élaboré au laboratoire.

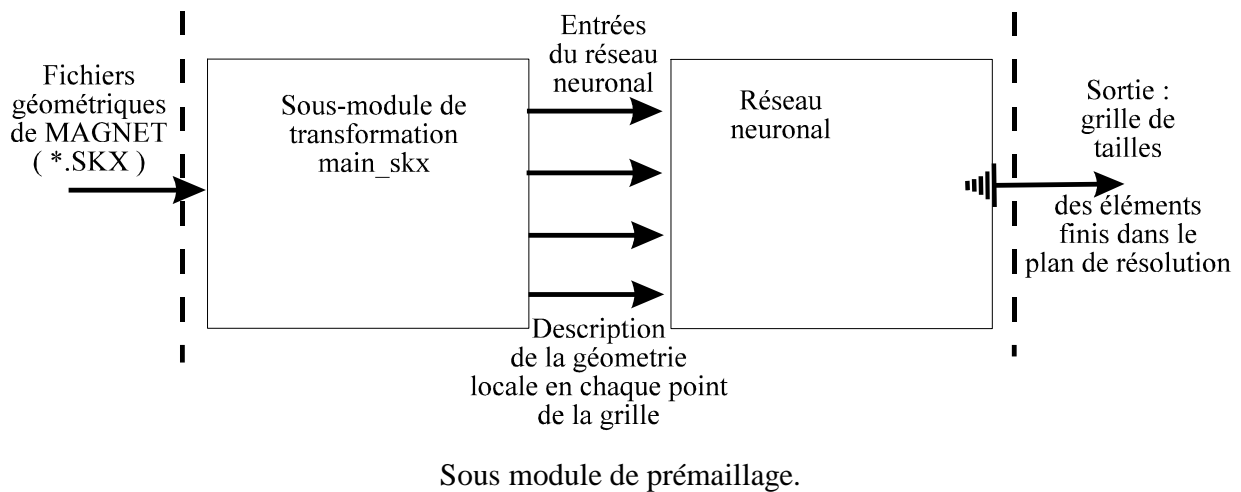


Figure I.1.b

I.2 Travail demandé.

Ma contribution au module de prémaillage est constituée de 3 parties :

- Le programme principal `main_skx` qui permet d'obtenir un set d'entrées (une procédure existe spécialement a cet effet) pour le réseau neuronal. Les entrées de ce programme se composent d'un fichier géométrique de MAGNET de type SKX¹ et d'une grille de positions de points où seront affectées les tailles des éléments finis. Dans la version utilisée par MAGNET, on n'utilise plus cette grille, mais on veut une densité de maillage en certains points du plan déterminés par le générateur de maillage.

Ce programme comporte 17 procédures et représente environs 2000 lignes sources en C.

- Choix des entrées du réseau neuronal, ayant accès à toute la géométrie du problème avec la procédure précédente. C'est de cette partie que dépend l'efficacité du prémaillage. Cette partie fait appel a la connaissance du fonctionnement des réseaux neuronaux et à des notions de base en maillage.

- Elaboration d'une interface graphique optionnelle permettant de visualiser la grille de maillage. Celle-ci est très utile, car on ne disposait précédemment que d'un pourcentage global fourni par le sous-module réseau neuronal. Cette interface permet donc de détecter les carences du prémaillage et de modifier les entrées en conséquence.

¹Se referer à l'exemple en annexe n°2.

II Présentation et algorithme du programme de traitement de données géométriques : main_skx.

Ce chapitre n'a pas pour objet de présenter les détails de programmation de ce programme (disponible dans le rapport technique annexe), mais d'expliquer son fonctionnement général, les principaux algorithmes et les structures de données utilisées. Les tests ont été effectués avec une grille de points et non pas en certains point comme c'est le cas pour la version qui sera utilisée par MAGNET.

II. 1 Utilisation.

L'objet de ce programme est de pouvoir savoir en un point quelconque M_0 du plan des informations du type suivant :

- segment le plus proche entourant du fer, avec dans la structure du segment (type superseg) la distance a ce segment, le point appartenant au segment le plus proche de M_0 , les caractéristiques géométriques du segment ainsi que l'adresse des segments adjacents a celui-ci.
- second segment le plus proche entourant du fer, avec les mêmes informations que précédemment.
- segment le plus proche entourant du cuivre.

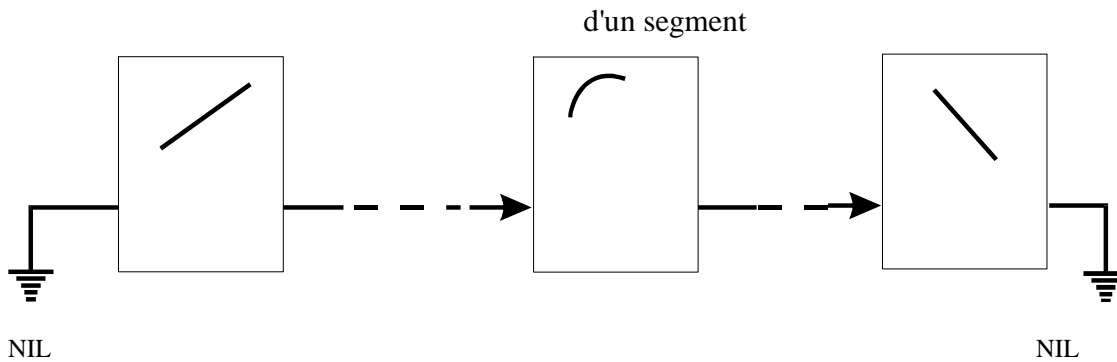
Avec ces informations, on est alors capable de déterminer la géométrie locale du problème pour constituer l'ensemble des entrées du réseau neuronal.

II.2 Algorithme général.

- Lecture des fichiers graphiques .SKX de MAGNET, et constitution d'une chaîne de pointeurs linéaires (Cf. fig. II.2.a) vers des structures de type superseg² décrivant complètement un segment, vu d'un point de la grille.

structures décrivant la géométrie

²Cf annexe 1



Chaîne de pointeurs de type superseg.

Figure II.2.a

- Détermination des grandeurs caractéristiques :

- Courbure maximale de la géométrie existant dans le problème : maxir.
- Points extrêmes de la machine, permettant par la suite de calculer la grandeur caractéristique du problème dmax.
- courbure des arcs remplaçant les coins déduits des deux grandeurs précédentes :

$$\text{courbure } ir = \frac{1}{\text{rayon de courbure}}$$

La courbure des coins irmax est définie comme suit :

$$\frac{10^3}{d_{\max}} \leq ir \leq \frac{10^5}{d_{\max}} \text{ et si c'est possible, } ir = 50 \times \text{maxir}$$

Ces caractéristiques ont été choisies de sorte que pour un problème dont la grandeur caractéristique est 1m, le rayon de courbure soit inférieur à 1mm.

- Constitution d'une chaîne de pointeurs vers des structures de segment (type superseg) regroupant tous les segments constituant une pièce fermée (Cf . fig. II.2.b) :

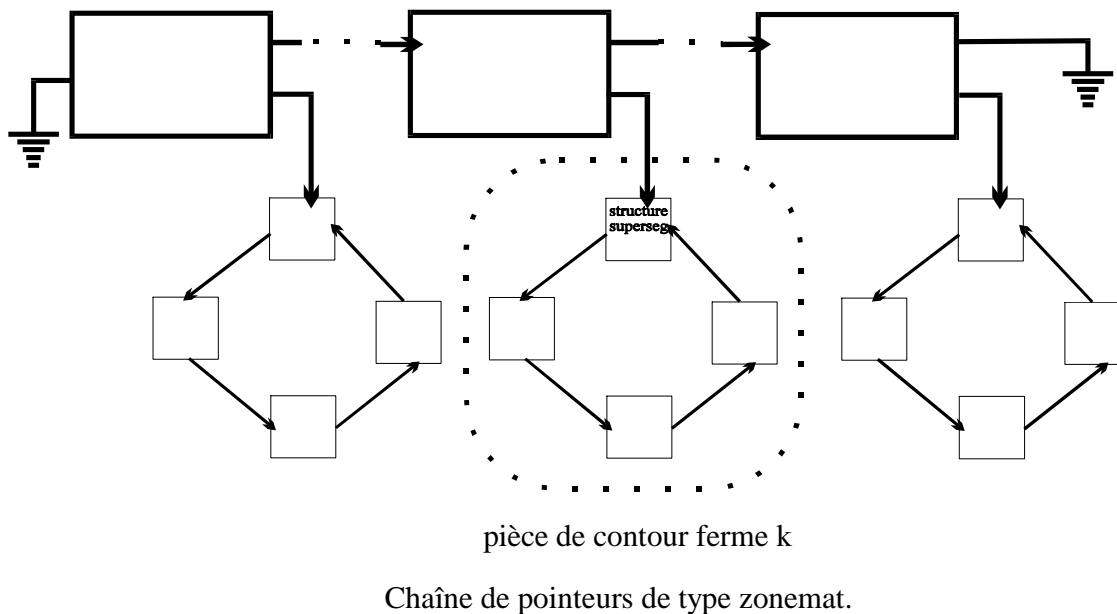


Figure II.2.b

Il est à noter que deux structures segment consécutives dans la chaîne correspondent à deux segments consécutifs dans la géométrie, sachant que tous les contours sont parcourus dans le sens trigonométrique.

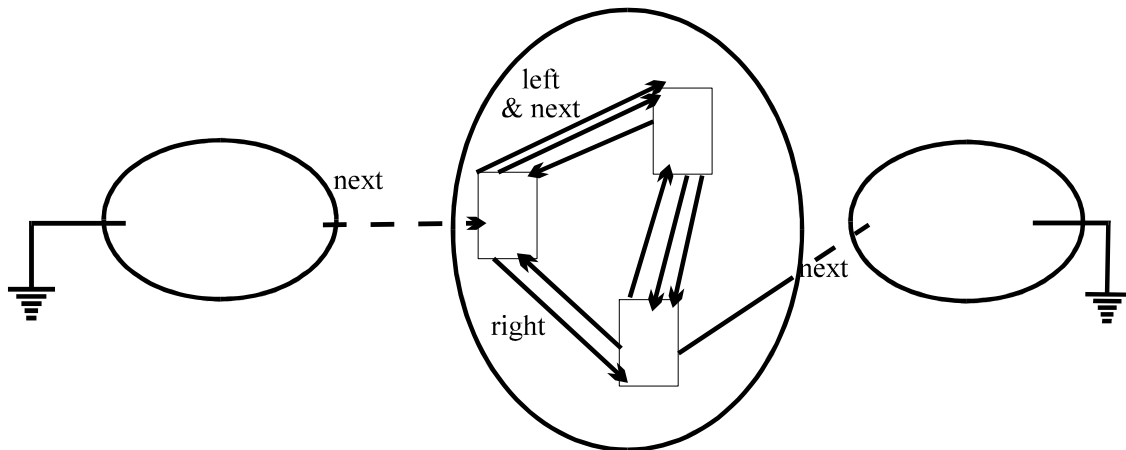
- Les contours sont ensuite lissés. En effet si deux segments consécutifs dans une pièce de contour fermé ne sont pas tangents, il est alors rajouté un arc de liaison de courbure maximale irmax. La raison du rajout de cet angle est triple.

Tout d'abord, cette représentation de la géométrie est plus proche de la réalité : quel que soit le cas étudié, la courbure d'un coin est toujours finie.

Ensuite, cette représentation permet d'uniformiser la géométrie en transformant un coin en un arc de cercle de courbure élevée. Ce type de représentation continue est préférable pour le réseau neuronal utilisé.

Enfin, l'algorithme utilise le fait que pour qu'un point soit à l'intérieur d'une pièce il faut que les segments qui la constituent soient tangents.

- Deux chaînes de pointeurs contenant les segments constituant toutes les pièces fermées du fer et du cuivre, avec des triples liens entre deux structures (Cf . fig. II.2.c):



Chaîne de pointeurs de type superseg utilisée en fin de programme.

Figure II.2.c

Une structure segment pointe vers le segment de gauche, de droite et vers le prochain segment de la chaîne, qui est identique au segment de gauche, sauf lorsque l'on change de pièce.

- Détermination des entrées du réseau neuronal, correspondant a un point de la grille :

On détermine tout d'abord le segment le plus proche, le second segment le plus proche non adjacent au précédent, pour le cuivre et pour le fer.

On utilise 2 prémaillages différents selon que Mo est dans l'air ou dans le fer : en effet, les critères de maillages sont totalement différents dans ces deux cas. Il est par conséquent impératif de savoir si Mo est dans l'air ou dans le fer (le cuivre est considéré dans ce cas comme de l'air, au niveau de son maillage).

Pour le savoir, il suffit de connaître l'orientation du segment le plus proche entourant une pièce de fer, vu du point Mo. Si cette orientation est positive (au sens trigonométrique), le point est à l'intérieur de la pièce, à l'extérieur sinon.

D'autre part, comme on peut le voir sur la figure II.2.d, dans une géométrie à tangentes continues, la droite passant par Mo et le point le plus proche du segment est toujours perpendiculaire au segment. C'est à dire que si le segment le plus proche est un arc de cercle, alors Mo est nécessairement dans le secteur angulaire de l'arc, si c'est un segment de droite, Mo est toujours entre les deux droites perpendiculaires au segment passant par Mfinal et Minitial. De fait, cette représentation simplifie énormément l'algorithme pour déterminer si Mo est à l'intérieur ou non du fer.

Cas où le segment le plus proche est un segment de droite

$$Mo \text{ est à l'intérieur} \Leftrightarrow \left| \vec{a} \times \vec{b} \right| \geq 0$$

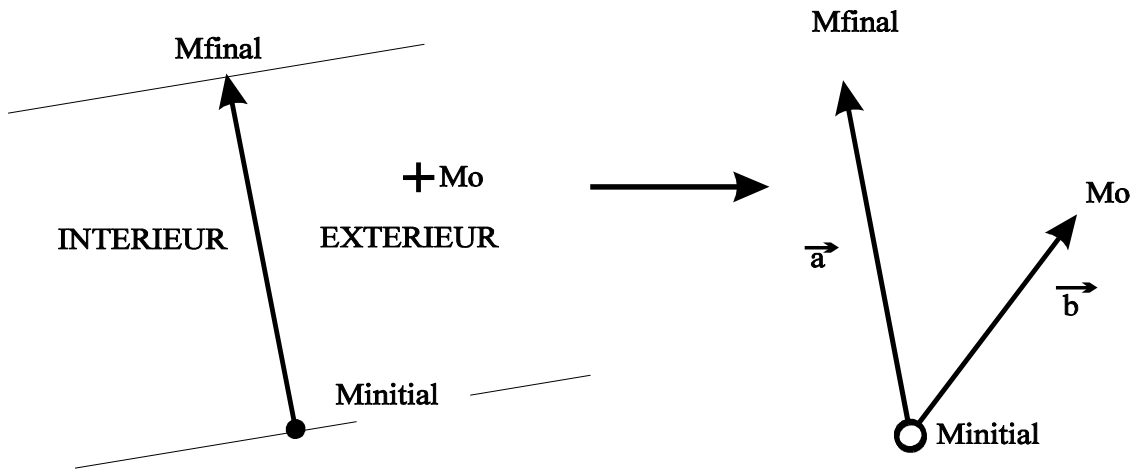


Figure II.2.d

Cas où le segment le plus proche est un arc de cercle (Cf. fig. II.2.e).

Si la courbure est positive

Si Mo est à l'intérieur du cercle de rayon $1/r$, alors Mo est à l'intérieur de la pièce, à l'extérieur sinon.

Sinon

Si Mo est à l'intérieur du cercle de rayon $1/r$, alors Mo est à l'extérieur de la pièce, à l'intérieur sinon.

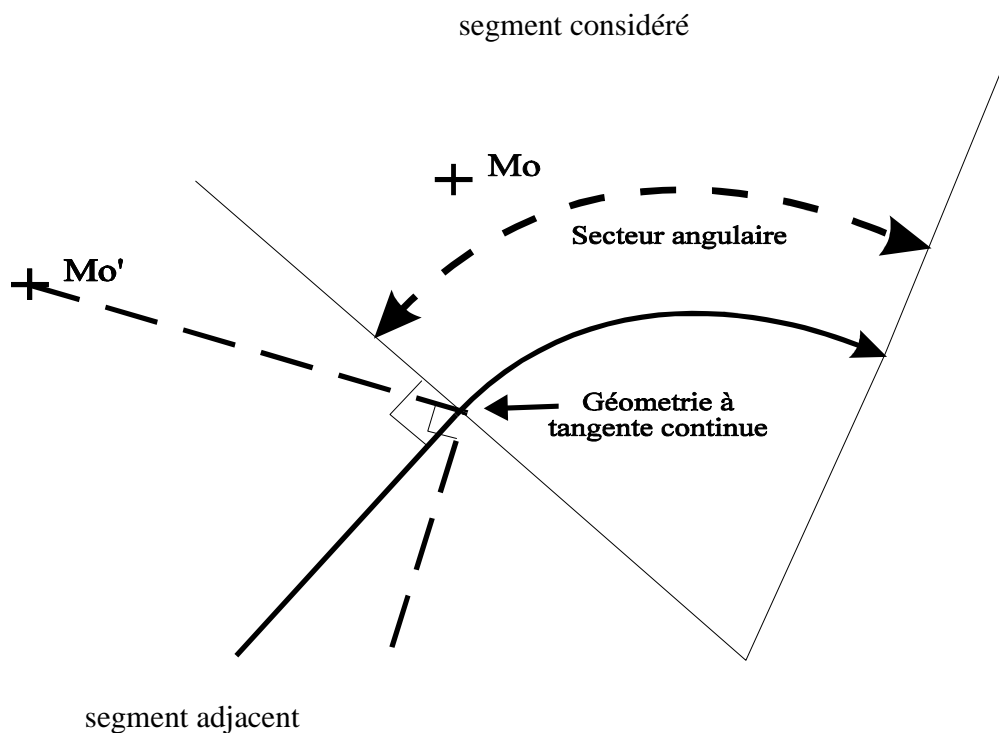


Figure II.2.e

De là, ayant accès pour tout point à sa géométrie locale, on peut alors déterminer dans une procédure prévue à cet effet, une combinaison d'entrées pour le réseau neuronal. Ces entrées devront être en sorte que celui-ci puisse leur associer une densité de maillage de façon correcte.

II.3 Structure de données utilisée.

La structure utilisée pour représenter le segment est la structure de type superseg³. Cette structure peut se décomposer en trois parties principales :

- données relatives aux caractéristiques du segment. Ces données sont définies en grandes parties au début du programme, lors de la lecture du fichier géométrique de MAGNET.
- données reliant les différents segments les uns aux autres : ces relations permettent en fait de reconstituer la géométrie du problème. Ces données sont définies alors que le programme reconstitue les contours fermes et les pièces.
- données décrivant les relations existant entre le segment et le point de prémaillage (distance ...). Cette partie de la structure va être donc changée en permanence, alors qu'on définit des entrées

³Les detail de cette structure sont en annexe n°1

pour beaucoup de points de prémaillage. Ces données sont définies en fin de programme alors que les autres données de la structure ne changent plus.

Avant d'entrer plus en détail dans le choix des entrées, il convient de détailler dans un premier temps la structure du réseau neuronal utilisée.

III Utilisation du réseau neuronal.

III.1 Notions générales.

Un réseau neuronal est une fonction non linéaire de plusieurs entrées et plusieurs sorties. Sa propriété principale reconnue expérimentalement mais non démontrée est son habileté à reproduire des sorties similaires si les entrées sont similaires.

Ainsi, les applications de ceux-ci sont principalement de la reconnaissance de formes. Cette reconnaissance peut se faire de manière à séparer en classes les différents types de formes (réseau de Kohonen), ou de manière à associer des valeurs en continu pour différentes formes. C'est un réseau neuronal du second type que j'avais à disposition, avec pour but d'associer une densité de maillage à une forme correspondant à la géométrie locale du problème autour d'un point.

Un réseau neuronal, de la même manière qu'un système expert, ne possède a priori aucune connaissance des problèmes qu'il doit résoudre. C'est pour cela, qu'il y a une phase d'apprentissage pour celui-ci, avant son utilisation proprement dite. A cet effet, on dispose d'un exemple de référence dont on connaît les entrées et les sorties. Cet exemple va permettre de transférer la connaissance que l'expert a utilisé pour le concevoir vers le réseau neuronal. Ainsi, si l'exemple choisi est pertinent, le réseau va être capable par la suite d'appliquer cette connaissance acquise à d'autres exemples, cette fois-ci inconnus, dans la mesure où l'expert serait capable de résoudre celui-ci avec la même approche.

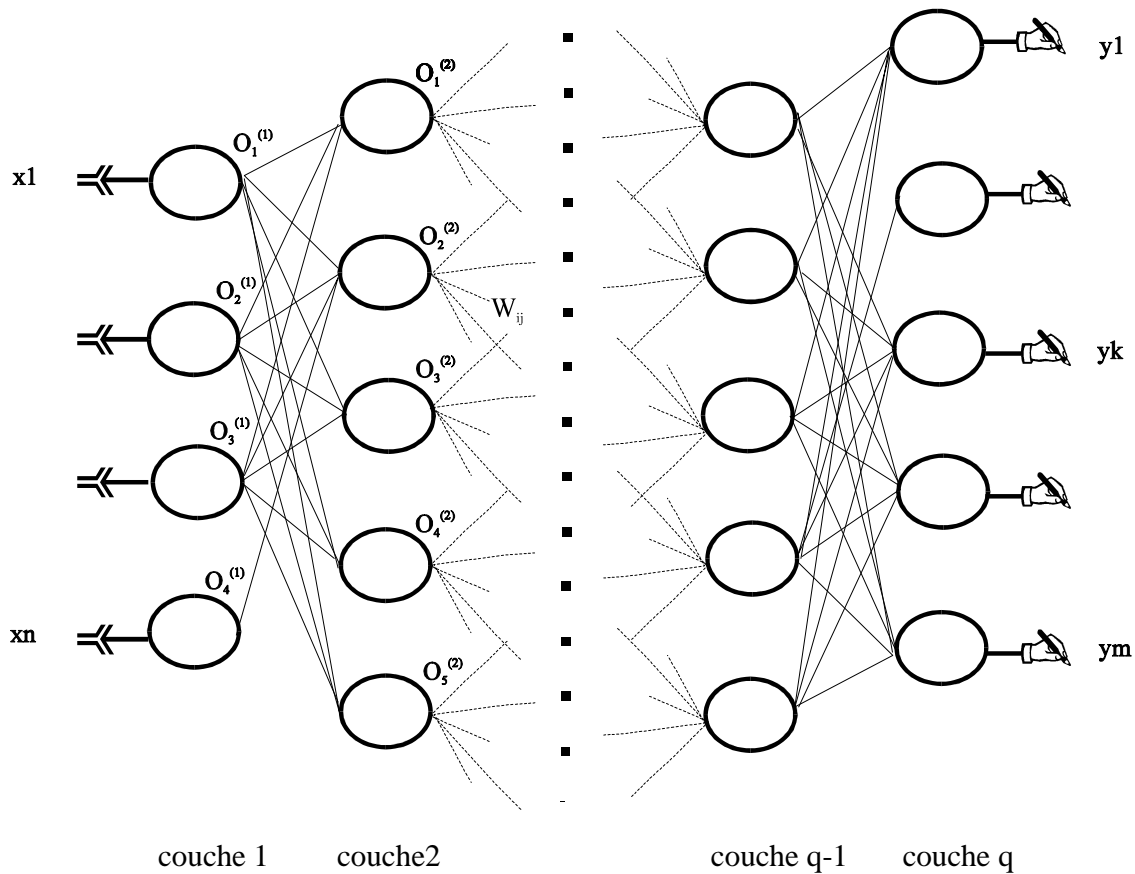
Néanmoins, comme nous le verrons par la suite, un réseau neuronal ne peut intrinsèquement pas donner de résultats parfaits. L'analyse du réseau neuronal de type "background propagation"⁴ que j'avais à disposition permet d'expliquer ce problème.

⁴traduction : propagation arrière

III.2 Description du réseau neuronal de type "background propagation".

III.2.1 Structure générale.

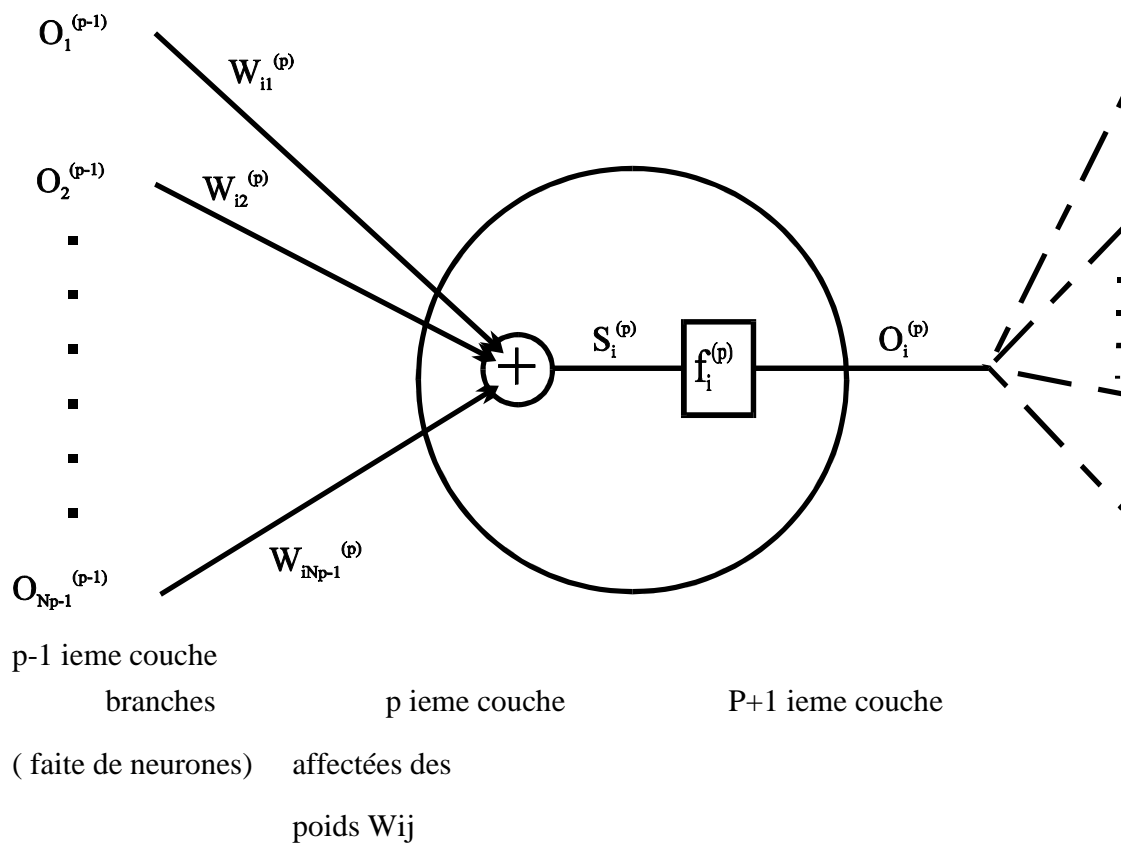
Ce réseau est constitué d'entrées, de sorties, de neurones (ou noeuds) et liens reliant tous les neurones d'une couche à chaque neurone de la couche suivante (Cf. fig.III.2.1.a).



Structure du réseau neuronal "background propagation".

figure III.2.1.a

A chaque branche reliant le j -ième neurone la $p-1$ -ième au i ième neurone de la p -ième couche on affecte un poids $W_{ij}^{(p)}$ (fig. III.2.1.b).

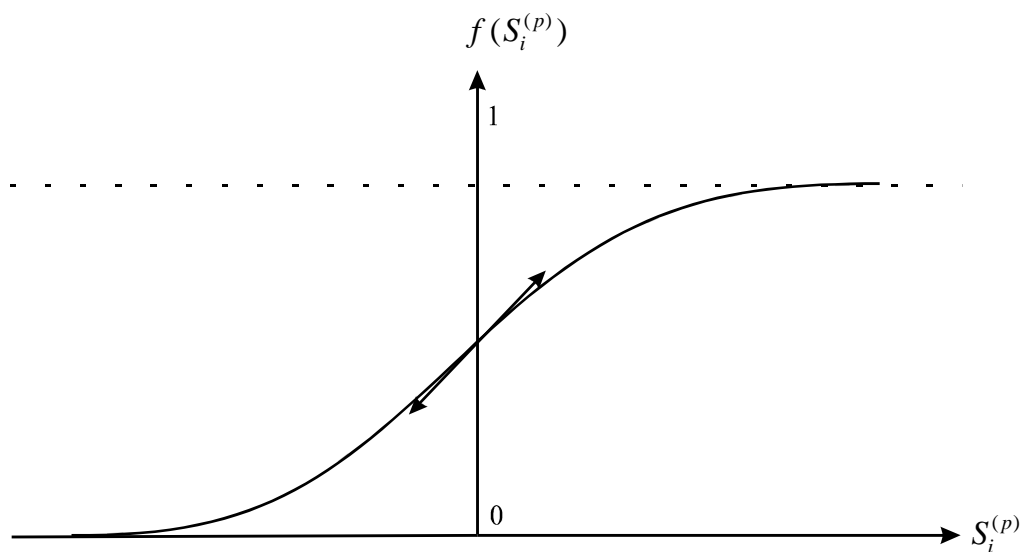


Structure d'un neurone.

figure III.2.1.b

Le calcul de la sortie $O_i^{(p)}$ du i ieme neurone de la p ième couche s'effectue de la manière suivante :

$$O_i^{(p)} = f(S_i^{(p)}), \text{ avec } S_i^{(p)} = \sum_j W_{ji}^{(p)} O_j^{(p-1)}, \text{ et } f(S_i^{(p)}) = \frac{1}{1 + \exp^{-S_i^{(p)}}}$$



Fonction de transfert d'un neurone.

Cela signifie en définitive que la fonction multidimensionnelle r associée au réseau neuronal peut s'écrire sous la forme :

$$r : R^n \rightarrow R^m$$

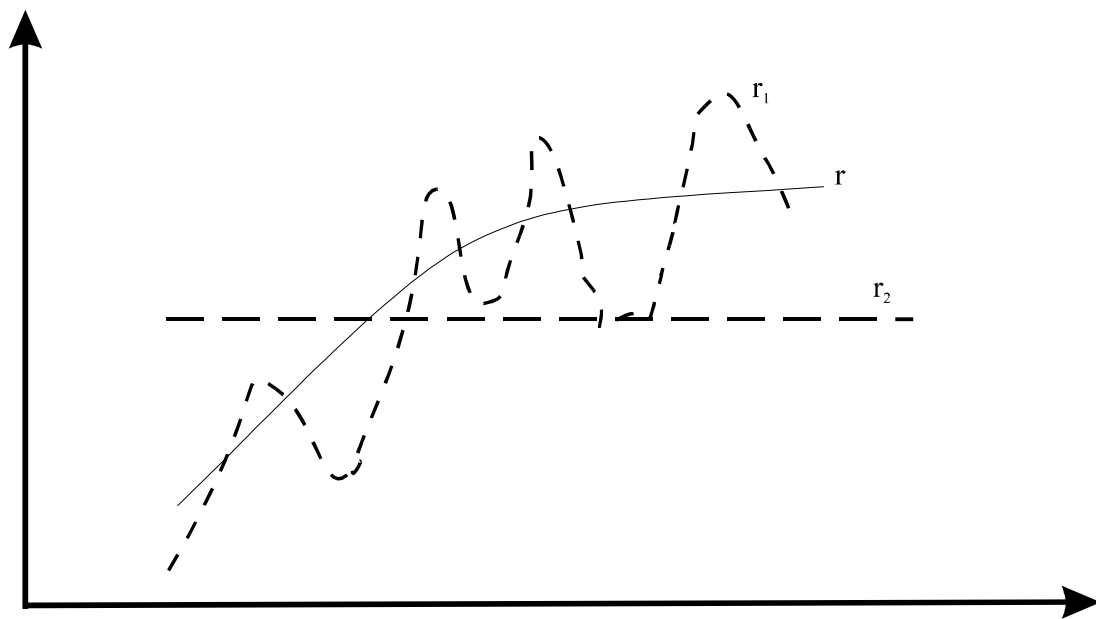
$$\underline{X} \rightarrow \underline{Y} = \underline{\underline{r(W_i)}} \cdot \underline{X}$$

Cette fonction est dans ce cas très complexe, et non linéaire. On pourrait imaginer pour d'autres applications une fonction f de transfert linéaire. Dans ce cas, le réseau neuronal pourrait se mettre sous forme matricielle associée à une application linéaire. Dans le cadre de mon projet, le réseau comportait 4 couches de neurones, avec plus de 10 neurones en moyenne par couche. Il apparaît alors clairement que l'expression exacte de r en fonction des W_{ij} est trop compliquée. Ainsi, le calcul des sorties se fait numériquement, couches par couches et neurones par neurones. La valeur calculée se propage dans le réseau des couches arrières (des entrées) aux couches avant (des sorties), d'où le nom du type du réseau neuronal.

On entrevoit ici que le but de l'entraînement du réseau neuronal est d'estimer r par interpolation de type moindre carres, avec comme modèle les points de l'exemple de référence.

C'est pour cette raison que le résultat ne peut être parfait :

Si le réseau est "trop bien" entraîné sur le modèle de référence, son habileté à reconnaître d'autres exemples se trouvera diminuée. Cela correspond à une interpolation qui calque précisément le modèle d'entraînement diminue l'aptitude du réseau neuronal à traiter les cas différents de celui-ci (Cf fig. III.2.1.c).



Fonctions d'interpolation

figure III.2.1.c

L'exemple de la figure illustre le problème de l'interpolation avec trop ou trop peu de degrés de liberté.

La fonction r représente la fonction associée au réseau neuronal.

La fonction r_1 représente la fonction r estimée avec trop de degrés de liberté. Inversement, la fonction r_2 représente la fonction r avec trop peu de degrés de liberté.

Les degrés de liberté d'un réseau neuronal sont représentés par les W_{ij} . Il est donc clair que la configuration du réseau neuronal (nombre de couches, nombre de neurones par couches) est un facteur important. Néanmoins, même si le réseau a une structure complexe, mais que ses entrées ne décrivent pas précisément l'exemple d'entraînement, l'interpolation ne sera pas correcte. Du choix des entrées et de leur nombre dépendra donc la réussite du réseau neuronal entraîné.

III.2.2 Méthode numérique utilisée pour la phase d'entraînement.

Les poids initiaux du réseau neuronal W_{ij} sont aléatoires. Le principe de l'entraînement consiste à trouver les W_{ij} qui minimise la fonction d'erreur quadratique entre les sorties de l'exemple de référence et les sorties du réseau neuronal correspondant aux entrées de l'exemple de référence :

$$\text{On cherche } \min(E = \sum_k \|y^k + t^k\|^2)$$

où

y^k représente la sortie du réseau neuronal pour un point de la grille de l'exemple de référence (ou d'entraînement) k.

t^k représente la sortie de référence du réseau neuronal associée à un point de la grille de l'exemple k.

Minimiser la fonction d'erreur revient à déterminer les W_{ij} tels que la dérivée de l'erreur par rapport à ceux-ci soit nulle :

$$\frac{\partial E}{\partial W_{ij}} = 0$$

Ceci correspond à un minimum local de cette fonction.. Le problème est de trouver si possible sinon le minimum global, l'un des minima le plus bas.

C'est pour cela que la méthode du "gradient descent"⁵ a été choisie par mon prédécesseur.

Durant la phase d'entraînement, chaque poids W_{ij} est choisi pour être modifié au hasard, et on peut alors calculer la dérivée de l'erreur par rapport à celui-ci. Les poids sont ainsi modifiés de la manière suivante :

$$\Delta W_{ij}^k = \alpha \cdot \Delta W_{ij}^{k-1} + \beta \cdot \frac{\partial E}{\partial W_{ij}}$$

ou α et β sont des coefficients ajustables par le concepteur lors de la phase d'optimisation du réseau neuronal.

Cette méthode peut se représenter à une dimension comme une bille (le coefficient W_{ij}) dévalant pente d'erreur avec au temps cas une vitesse d'inertie $\alpha \cdot \Delta W_{ij}^{k-1}$ et une vitesse due à

l'accélération de la pente d'erreur $\beta \cdot \frac{\partial E}{\partial W_{ij}}$ (Cf. fig. III.2.2.a)

Valeur de la fonction d'erreur

⁵ traduction : descente par gradient.

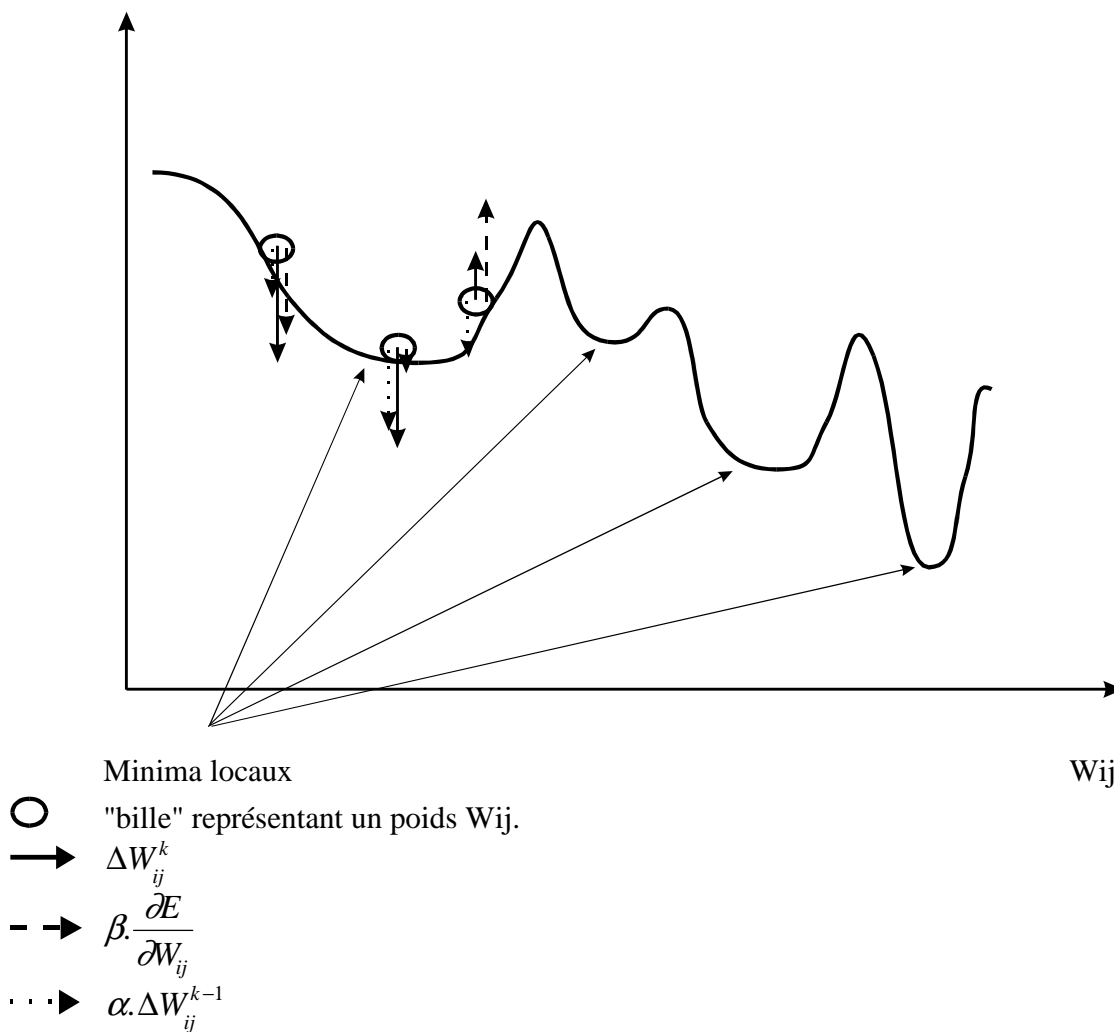


Figure III.2.2.a Evolution des coefficients W_{ij} en fonction de la pente d'erreur

L'avantage de cette méthode réside dans le fait que la fonction d'erreur peut "s'échapper" dans la plupart des cas des minimums locaux et ce pour deux raisons.

D'une part, la topologie de la fonction d'erreur change constamment en fonction des W_{ij} . C'est à dire qu'un minimum local pour un W_{ij} n'en sera pas un pour le prochain qui est choisi aléatoirement.

D'autre part, la fonction d'erreur n'est pas forcément enfermée dans un minimum local (à moins qu'il ne soit trop profond) à cause de la "vitesse d'inertie" $\alpha \cdot \Delta W_{ij}^{k-1}$.

Le but de cette méthode n'est pas que la fonction d'erreur soit "enfermée" dans le minimum global, mais seulement qu'elle passe par celui-ci, quitte à en sortir. En effet, dans le programme, la configuration du réseau minimisant l'erreur est sauvegardée durant la phase d'entraînement.

Néanmoins, cette méthode est relativement peu rapide. En effet, la phase d'entraînement dure environs deux heures sur un SUN sparc ipx pour 1 000 000 d'itérations. D'autre part, il n'y a pas de garantie dans ce cas de passer par le minimum local. Il est donc très important que le concepteur ajuste de manière correcte les coefficients α et β . De toute façon, il n'est pas envisageable d'utiliser une méthode de type Newton pour minimiser la fonction d'erreur. Pour un réseau neuronal classique, il y a au minimum 400 coefficients W_{ij} à déterminer, et cela n'est pas possible avec la technologie actuelle.

III.2.3 Décodage de la sortie.

Bien que l'on attende du réseau neuronal une valeur numérique (la densité de maillage), il est clair que de par son mode de calcul cette sortie sera bruitée. En effet, cette valeur numérique se propage couches par couches, et les poids W_{ij} pouvant être négatifs, de nombreuses soustractions consécutives sont effectuées. Ainsi, il est préférable d'utiliser les propriétés de classification des réseaux neuronaux qui sont très fortes, pour déterminer la valeur de la sortie, plutôt que le résultat numérique brut. Il est à noter que les réseaux neuronaux sont principalement utilisés pour leur habileté à classifier, et notamment en reconnaissance de formes.

On connaît a priori la dynamique des valeurs de densités de maillage relatives. Il suffit alors de diviser cette plage en intervalles réguliers associés à chaque neurone. Cette répartition des sorties sur plusieurs neurones a donné de très bons résultats [cf. D. Dick 1990] en amenuisant en grande partie les bruits de calculs inhérents au mode de calcul.

La valeur de la sortie décodée est dans l'intervalle associée au neurone ayant la plus forte sortie. Les valeurs obtenues dans ce cas sont discrétisées. Pour que lisser ces valeurs soient, on décode de la manière suivante la sortie :

$$S = \frac{O_{n-1}I_{n-1} + O_n I_n + O_{n+1}I_{n+1}}{O_{n-1} + O_n + O_{n+1}}$$

où S représente la sortie décodée.

O_{n-1}, O_n, O_{n+1} représentent la valeur des sorties des neurones n-1, n et n+1 de la dernière couche, telle que O_n soit la valeur de sortie la plus forte.

I_{n-1}, I_n, I_{n+1} représentent la valeur du centre de chaque intervalle n-1, n et n+1 associées aux neurones correspondants de la dernière couche.

Dans l'exemple de la figure ci-dessous, le réseau neuronal comporte 10 neurones en dernière couche. Admettons que la dynamique de la densité de maillage soit de 0 à 10, les intervalles correspondants aux sorties des neurones sont donc de type [n, n+1]. Dans l'exemple ci-dessous,

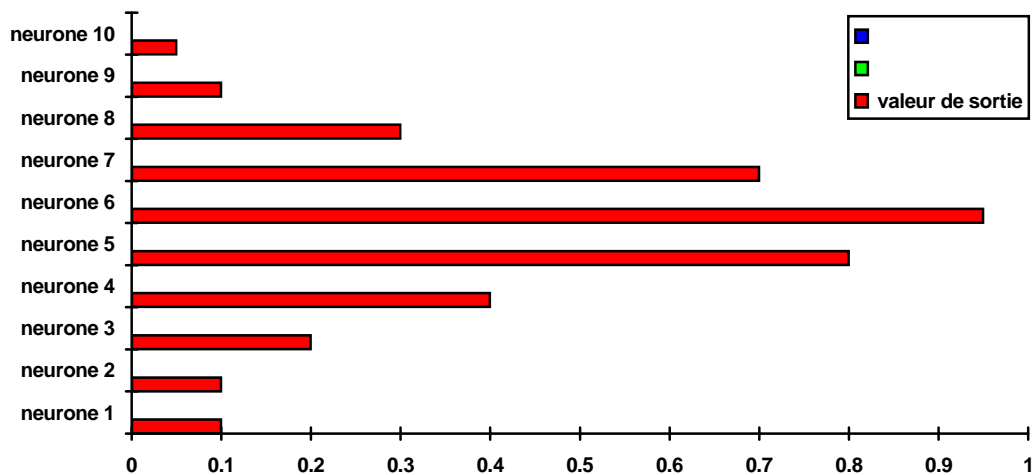
la sortie la plus forte qui correspond au neurone 6 vaut 0.95. L'intervalle correspondant est [5, 6], et sa valeur centrale est 5.5.

Finalement, la sortie décodée vaut :

$$S = \frac{0.8 * 4.5 + 0.95 * 5.5 + 0.7 * 6.5}{0.8 + 0.95 + 0.7}$$

C'est à dire

$$S = 5.46.$$



Exemple de sortie codée pour un réseau neuronal comportant 10 noeuds en dernière couche.

IV Représentation de la géométrie locale pour les entrées du réseau neuronal.

IV.1 Critère de maillage idéal.

Le maillage idéal tel qu'il est calculé dans l'exemple de référence doit se soumettre aux deux contraintes suivantes :

- Minimiser le nombre total éléments finis.
- Les erreurs de calcul après simulation doivent être inférieures à un seuil d'erreur E_0 , fixé arbitrairement selon les besoins de l'application, sur tous les éléments finis.

On peut en déduire de manière évidente que le maillage répondant à ces deux contraintes est tel que l'erreur sur chaque triangle du maillage soit E_0 précisément.

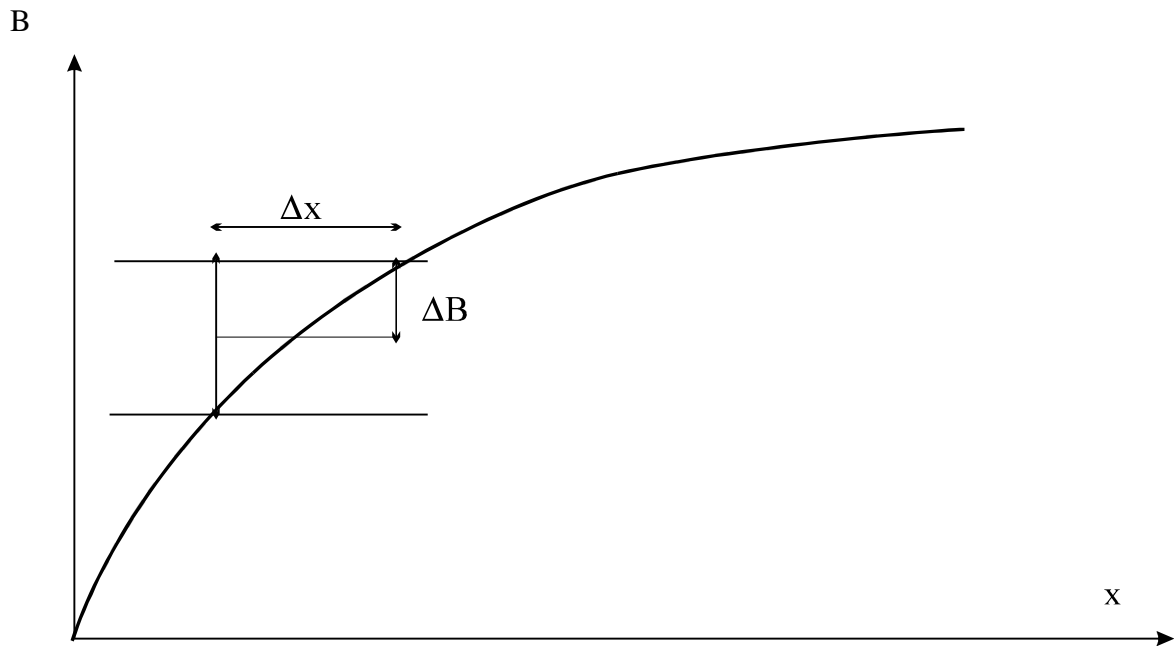
IV.2 Problème de la mise a l'échelle.

IV.2.1 Propriété d'homothétie du maillage.

Le réseau neuronal doit prémailler de la même manière un problème d'une géométrie fixée mais à échelle variable, à un facteur près. Il est donc très important de normaliser les entrées, de manière à ce que le réseau neuronal "voie" un problème à différentes échelles de la même manière. Cette propriété est d'autant plus nécessaire que les fonctions de transfert des neurones, des sigmoïdes, ont une plage où leur dérivée n'est pas négligeable entre $[-1;1]$ environ. Avec la méthode numérique utilisée, on dérive l'erreur par rapport à chaque W_{ij} . Si le neurone est saturé, c'est à dire que sa valeur d'entrée n'est pas comprise entre $[-1;1]$, une petite variation entraînée par celle de W_{ij} , n'entraînera pas de variation significative de la sortie.

Les sorties n'ont pas à être remises à l'échelle, car on ne s'intéresse qu'aux tailles relatives des éléments finis. Néanmoins, ceci sous-entend que le problème (et surtout la répartition des densités de maillage) est invariant par homothétie, ce qui n'est pas évident a priori.

La définition de l'erreur du champ magnétique B sur un élément fini est l'écart absolu entre sa valeur moyenne et sa valeur extrême sur un élément. On peut illustrer ceci par une représentation à une dimension, avec la grandeur caractéristique du triangle représentée par Δx (Cf. fig. IV.2.1.a).



Représentation a une dimension de l'erreur de maillage sur un élément fini.

Figure IV.2.1.a

En faisant l'approximation que B varie linéairement sur un élément fini à une dimension (ce qui doit être le cas si E_0 est suffisamment petit), on a

$$\frac{\partial B}{\partial x} \approx \frac{\Delta B}{\Delta x}$$

Or, la définition de l'erreur est dans ce cas :

$$E_0 = \frac{\partial B}{\partial x} \cdot \Delta x$$

L'erreur E_0 est donc constante si

$$\frac{\frac{\partial B(x_1)}{\partial x} \cdot \Delta x_1}{\frac{\partial B(x_2)}{\partial x} \cdot \Delta x_2} = 1 \quad \forall (x_1, x_2), \text{ points du problème.}$$

Si on effectue une homothétie du problème par rapport à l'origine des coordonnées du plan d'étude (pour simplifier) et de rapport h, alors

$$x \rightarrow h \cdot x.$$

L'Equation précédente devient donc :

$$\frac{\frac{\partial B_h(h.x_1)}{\partial(hx)} \cdot h\Delta x_1}{\frac{\partial B_h(h.x_2)}{\partial(hx)} \cdot h\Delta x_2} = 1$$

Or, dans un problème à flux fermé, B est proportionnel au courant i . Dans le cas d'une homothétie, les surfaces augmentant au carré, on a :

$$i \rightarrow h^2 i, \text{ donc } B_h(hx) = h^2 B(x)$$

L'équation précédente devient alors

$$\frac{\frac{h^2 \partial B(x_1)}{h \partial x} \cdot h\Delta x_1}{\frac{h^2 \partial B(x_2)}{h \partial x} \cdot h\Delta x_2} = 1$$

Ce qui est, simplifications faites équivalent à la condition initiale sur l'erreur.

Finalement, h se simplifiant, on se ramène au premier cas, avec $h.x_1$ et $h.x_2$ qui peuvent être considérés comme deux points particuliers du plan de résolution.

Le même raisonnement s'applique en deux dimensions, en prenant pour définition de l'erreur :

$$E_0 = \frac{\partial^2 B(x, y)}{\partial x \partial y} \Delta x \Delta y$$

où $\Delta x \Delta y$ représente l'élément de surface.

Finalement, on a démontré que le problème est invariant par homothétie. Ceci signifie, que toutes les grandeurs ayant trait à l'échelle du problème peuvent être normalisées en entrées du réseau.

IV.2.2 Choix du facteur d'échelle.

Le facteur d'échelle est un critère très important lors de la phase de conception des entrées. On peut avoir deux approches quand à ce choix. Mon prédécesseur a opté pour un facteur d'échelle local, qui est la distance au coin le plus proche au point de prémaillage. Ce choix se justifie d'une part par le fait que l'on considère une géométrie locale pour les entrées, et donc qu'il peut être légitime d'utiliser un facteur d'échelle local.

Néanmoins ce choix est fortement corrélé à la représentation de la géométrie qui avait été adoptée. En effet, à moins de modéliser une ligne droite sans épaisseur (ce qui est de nos jours assez rare), une géométrie ne comportant que des segments de droite possède au moins un coin. De plus, toutes les entrées du réseau neuronal qui sont mises à l'échelle par ce facteur tendent vers l'infini lorsque le point de prémaillage est proche d'un coin. En fait mettre à l'échelle les entrées par la distance au coin le plus proche revient à supputer que la taille des éléments finis est globalement proportionnelle à la distance au coin le plus proche. Ainsi on peut obtenir de bons résultats, le réseau déterminant en fait le coefficient de proportionnalité correspondant. Le réseau avait dans ce cas tendance à mailler trop finement près des coins et trop largement loin de ceux-ci.

L'approche que j'ai adoptée dans le choix de mon facteur d'échelle a été d'une part de considérer qu'avec une géométrie comportant des courbes, l'importance des coins n'était pas aussi prédominante que précédemment. Dans ce cas, le seul facteur d'échelle local viable est la distance au segment (courbe ou rectiligne) le plus proche. Dans ce cas, tous les points proches de la frontière fer-air auraient vu certaines de leurs entrées (celles mises à l'échelle) tendre vers l'infini, et donc saturer numériquement leur neurone d'entrée correspondant. En effet, la fonction de transfert⁶ du neurone différencie difficilement les valeurs élevées. Cela a pour conséquence de ne plus pouvoir différencier deux points plus ou moins proches de la frontière, et d'autre part de ralentir le processus d'apprentissage. En effet une variation du poids W_{ij} reliant l'entrée au premier neurone n'entraîne pas de variation de l'erreur globale. Le choix de prendre la courbure comme facteur d'échelle est encore moins envisageable, car elle peut être nulle, donc annuler toutes les entrées mises à l'échelle par ir (multipliées par ir , i.e divisées par le rayon de courbure), pour des points dont la géométrie locale est différente, mais dont le segment le plus proche est rectiligne.

Finalement, il semble que pour la représentation courbe, le facteur d'échelle le plus adapté soit une longueur caractéristique globale au problème. Ceci a l'avantage de ne pas trop avantager une des caractéristiques locales (distance au coin, distance au segment le plus proche) par rapport



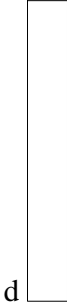
⁶ Se reporter à la description du réseau neuronal utilisé.

aux autres entrées. De plus, on n'assiste pas aux saturations des neurones entrées, lorsque le point de maillage est proche d'une singularité géométrique.

Le problème est de savoir quelle distance caractéristique du problème peut être adoptée. Les distances caractéristiques utilisées ont été la moyenne géométrique et la diagonale du rectangle inscrit à l'appareil modélisé.

Les essais ont été concluants avec les exemples à disposition, mais ce choix pouvait poser des problèmes dans des cas particuliers, du type tige. J'ai donc été amené à considérer plusieurs choix pour déterminer ce facteur d'échelle globale:

les extrema du rectangle inscrit à la machine étudiée sont $x_{max}, x_{min}, y_{max}, y_{min}$. Voici dans un tableau récapitulatif les possibilités envisagées :

Facteur d'échelle utilise	Application numérique à un carré de cotés(d*d)	Application numérique à un rectangle de cotés (d*2d)	Application numérique à un rectangle de cotés (d*0.1d)	Remarques
				
Longueur de la diagonale hypoténuse ($x_{max}-x_{min}, y_{max}-y_{min}$)	$d\sqrt{2}$	$d\sqrt{5}$	$d\sqrt{1.01}$	Ne différencie pas les géométries en privilégiant le plus grand coté
moyenne arithmétique $\frac{(x_{max} - x_{min}) + (y_{max} - y_{min})}{2}$	d	1.5d	0.55d	

$\min(x_{\max}-x_{\min}, y_{\max}-y_{\min})$	d	d	$0.1d$	Ne différencie pas assez les géométries
moyenne géométrique $\sqrt{(x_{\max} - x_{\min}) * (y_{\max} - y_{\min})}$	d	$d\sqrt{2}$	$0.316d$	

La longueur de la diagonale et le minimum du rectangle inscrit sont inadapés car ils différencient à peine un carré et une tige. D'autre part, la moyenne arithmétique ne rend moins compte du caractère "semi-infini" du troisième exemple. Finalement il apparaît après étude de ce tableau des différents choix de facteurs d'échelle, que la moyenne géométrique représente le meilleur compromis. Les essais ayant été effectués avec des problèmes tels que le rectangle inscrit à l'appareil était sensiblement un carré, la différence entre les différents facteurs d'échelle était difficile à évaluer. Le choix de la moyenne géométrique comme facteur d'échelle a donc été adopté *a priori*.

IV.3 Choix des entrées.

L'un des points de départ de ce projet est de considérer qu'il est possible de trouver des densités de maillage correctes d'un problème, uniquement en décrivant sa géométrie locale. Ceci est en toute rigueur inexact. Néanmoins, en examinant le maillage des deux exemples de référence, et il a été possible de trouver une corrélation entre les densités de maillage et d'éventuelles entrées représentant la géométrie locale.

Pour déterminer les entrées les plus adaptées, j'ai élaboré un programme permettant de visualiser la taille des éléments finis (l'inverse de sa densité de maillage) pour une grille de points. Ainsi on peut visualiser les tailles idéales de l'exemple d'entraînement et les tailles fournies par le réseau neuronal pour les entrées correspondantes. Lors du travail précédent, on ne disposait que d'un indice d'erreur global. Avec mon programme, j'ai donc pu plus facilement voir les carences de mes entrées et les modifier en conséquence pour faire coïncider les tailles idéales des éléments finis et celles fournies par le réseau.

Les réseaux neuronaux ayant été peu employés pour ce type d'applications, il n'existe pas de méthodologie permettant de maîtriser les paramètres tels que le nombre des entrées, le nombre de couches et de neurones par couches, ainsi que les paramètres α et β de vitesse de convergence de la fonction d'erreur vers son minimum. D'autre part, les moyens informatiques dont on dispose ne

permettent pas raisonnablement de déterminer tous ces paramètres de manière automatique avec une recherche systématique du minimum de la fonction d'erreur. C'est pour cela que dans un premier temps, j'ai utilisé la configuration optimale du réseau neuronal qu'avait trouvé mon prédécesseur lors de nombreuses simulations, pour me consacrer au choix des entrées. Une fois que j'ai trouvé le meilleur set d'entrées, j'ai fait des tentatives de modification de structure. Les meilleurs résultats ont été obtenus avec la même configuration que D. Dyck, mais avec les paramètres de vitesse de convergence deux fois inférieurs.

V Essais en vue d'obtenir un module de prémaillage opérationnel.

V.1 Conditions expérimentales.

Je disposais de deux exemples de référence. L'un des deux exemples, le "Ecore"⁷ (Cf. fig) sert à entraîner le réseau, avec 3997 points de prémaillage, c'est à dire autant d'exemples d'entraînement ou de points pour l'interpolation. L'autre exemple dont on connaît les sorties et les entrées correspondantes sert à déterminer l'erreur que fait le réseau neuronal sur un exemple inconnu "Gcore". Chaque essai se déroule donc avec un set d'entrées différentes déterminées pour les mêmes points que ces deux exemples, des paramètres différents, ainsi qu'un nombre d'itérations d'entraînement (comprises entre 100 000 et 2 500 000).

Quand l'essai est terminé, le programme retient la configuration (l'ensemble de tous les W_{ij}) où l'erreur test2 (sur Gcore) est minimum, et telle que celle l'erreur test1 (sur Ecore) soit inférieure a l'erreur test2. On considère au vu des essais et des visualisations sur l'interface graphique qu'une erreur de type :

erreur test1 : >20% erreur test2 : >30% est mauvaise. Ce type d'erreur permet d'éliminer les entrées non pertinentes. Néanmoins, il est très important d'avoir un set de base ou toutes les entrées sont correctes, car une seule entrée mal choisie peut conduire à de très mauvais résultats, même si les autres sont correctes. Par exemple, lors de mes essais, une de mes entrées était l'angle entre les deux segments les plus proches p_{close1} et p_{close2} . Cette entrée n'était pas absurde a priori, mais amenait des erreurs supérieures à 40% sur l'erreur test2. Ma démarche a donc été de déterminer en premier temps un set minimal d'entrées. Puis d'ajouter de nouvelles entrées en les gardant à condition qu'elles améliorent les test d'erreur.

V.2 Considérations de base sur le maillage conduisant au choix des entrées.

Les entrées du réseau neuronal sont purement géométriques. Leur détermination revient donc à établir une corrélation entre la géométrie du problème et la densité de maillage.

Globalement on peut établir trois relations assez évidentes, les relations plus complexes devant être apprises par le réseau neuronal lors de la phase d'entraînement avec l'exemple de référence.

⁷ Voir p.37 / 38

- Relation 1 : la densité de maillage est d'autant plus élevée au point de prémaillage M_0 que celui-ci est proche d'une singularité géométrique (coin aigu, segment de courbure élevée). Si M_0 est très proche d'une frontière air-fer mais que celle-ci est rectiligne, la densité de maillage n'est pas élevée si aucune des autres relations n'est mise en jeu.
- Relation 2 : la densité de maillage est d'autant plus élevée que M_0 est proche de l'entrefer (que ce soit pour l'air ou le fer).
- Relation 3 : La densité de maillage doit être d'autant plus élevée que l'on s'approche d'un conducteur de cuivre. En effet, dans le cas de l'air, les fuites se situent a priori au voisinage du conducteur de courant.

Ces trois relations n'apportent pas de solutions quantitatives pour le problème de détermination du maillage. Néanmoins, elles permettent de définir des entrées qui permettront au réseau neuronal d'interpoler la fonction liant ces entrées à la densité de maillage.

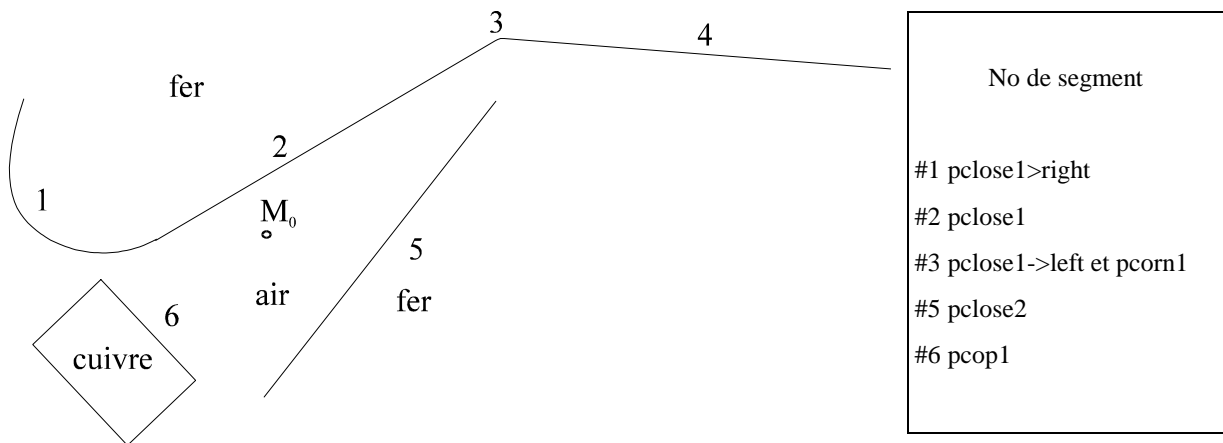
V.3 Essais conduisant les choix d'entrées.

Dans la lignée des travaux précédents, j'ai entraîné séparément 2 réseaux neuronaux, un pour l'air et un pour le fer. Mon but a été dans un premier temps de déterminer un set de base de six entrées. Le nombre de six entrées est un minimum, sachant que mon prédécesseur en utilisait huit en configuration optimale. Après plusieurs essais, où j'ai écarté les entrées non pertinentes et au vu du maillage optimal, je suis arrivé au set d'entrées décrit dans le paragraphe suivant.

V.4 Liste des entrées.

Configuration des entrées retenues après essais :

Les notations relatives aux différents segments correspondants à la géométrie locale sont les suivantes :



- Entrée no 1 : $\frac{pcorn1 \rightarrow d}{scal}$

L'examen de la grille de prémaillage de l'exemple d'entraînement -Ecore- et de l'autre exemple de référence -Gcore- (Cf. fig) m'ont conduit à trouver une forte corrélation entre la proximité d'un angle de la géométrie et l'augmentation de la densité de maillage (En accord avec le cahier des charges, le fichier graphique doit comporter au moins un coin). Néanmoins, cette densité semble être d'autant plus importante que cet angle est aigu Cette entrée se rapporte à la relation 1.

- Entrée no 2 : $\frac{\text{longueur du segment le plus proche}}{scal} = \frac{1}{scal * pclose1 \rightarrow id}$

Cette entrée se rapporte à la relation 1. En effet, cette entrée caractérise la "non-singularité" du segment le plus proche. Si cette entrée est grande, son influence sur la densité de maillage sera de la diminuer.

- Entrée no 3 : $\frac{pclose2 \rightarrow d - pclose1 \rightarrow d}{scal}$

L'effet sur la variation de l'induction magnétique autour de l'entrefer est très important dans une machine à flux magnétique fermé. Le choix de cette entrée est tel que lorsque le point de maillage M_o est entre deux parois de fer, cette entrée tend vers 0, sa contribution tend donc dans ce cas à augmenter la densité de maillage. Le seul autre cas principal où cette entrée tend vers 0, intervient lorsque le point M_o est équidistant à $pclose1$ et $pclose2$. Néanmoins, la probabilité d'être dans ce cas diminue à mesure que le point s'éloigne de la frontière air-fer.

$$\text{- Entrée no 4 : } 1 - \left(\frac{scal}{pclose2 \rightarrow d} * \frac{pclose2 \rightarrow ir}{ir \max} \right) * \left(1 - \frac{\left| \frac{pclose2 \rightarrow ir}{pclose2 \rightarrow id} \right|}{\pi} \right)$$

Cette entrée mesure l'effet de "singularité" en référence à la relation 1 du second segment le plus proche non adjacent. Cette entrée sera égale à 1 pour un segment rectiligne et diminuera d'autant pour un segment courbe si l'angle de celui-ci est faible et si l'angle de ce segment est faible (coin "pointu" si $ir=ir_{max}$), si M_o est proche de ce segment et ou si la courbure de ce segment est élevée.

$$\text{- Entrée no 5 : } \frac{pclose2 \rightarrow d}{scal}$$

La distance du second segment non adjacent n'a pas d'influence en tant que telle sur la densité de maillage. Néanmoins, si celle-ci est faible, alors la distance du segment le plus proche l'est encore plus. Dans ce cas, M_o est très proche de deux segments non sécants, et ceci caractérise sa proximité de l'entrefer. Cette entrée complète donc l'entrée 3 et correspond à la relation 2.

$$\text{- Entrée no 6 : } 10 * \frac{pcop1 \rightarrow d}{scal}$$

Cette entrée caractérise la relation 3. D'autres types d'entrées ont été imaginés, comme par exemple la distance au centre de gravité du conducteur (D. Dick 1990). Néanmoins, cette représentation avait le désavantage de ne pas être efficace dans certains cas. Le cas extrême étant un câble coaxial (bien que ce genre de cas n'entre pas dans le cadre de résolution du module de prémaillage). Le facteur 10 n'a priori aucune raison d'être, dans le cas où la méthode numérique employée lors de la phase d'apprentissage converge vers le minimum global. Malheureusement, cela n'étant pas nécessairement le cas, il est apparu lors des essais que l'influence de cette entrée rehaussée relativement aux autres améliorerait les résultats (la corrélation est assez évidente avec une visualisation de la grille des tailles d'éléments finis). Numériquement, le fait de

communiquer ce savoir aux entrées revient à rapprocher le point de départ des itérations du minima global.

$$\text{- Entrée no 7 : } 1 - \frac{\frac{pclose1 \rightarrow ir}{pclose1 \rightarrow id}}{\frac{\pi}{2}}$$

Cette entrée représente le fait qu'un segment est un coin pointu avec pour référence 1 si l'angle mesure 45 degrés, plus si l'angle est inférieur à cette valeur (cas d'un angle aigu), moins sinon. Ceci permet d'ajouter une information supplémentaire propre à la relation 1.

- Entrée no 8 :

$$0.5 * (1 - (0.1 * (\left| \frac{pclose1 \rightarrow left \rightarrow ir}{ir \max} \right| * (1 - \left| \frac{pclose1 \rightarrow left \rightarrow ir}{pclose1 \rightarrow left \rightarrow id} \right|) + \left| \frac{pclose1 \rightarrow right \rightarrow ir}{ir \max} \right| * (1 - \left| \frac{pclose1 \rightarrow right \rightarrow ir}{pclose1 \rightarrow right \rightarrow id} \right|))))$$

Cette entrée est de type relation 1. Elle est la somme de deux entrées identiques que l'entrée 4. Cette entrée représente l'influence du voisinage du segment le plus proche (segments de gauche de droite) en termes de singularité. Cette entrée étant d'une influence relative faible sur le maillage, elle a été multipliée par un coefficient réducteur, à l'inverse de l'entrée 6.

V.5 Critères de choix des entrées.

Ces critères de choix d'entrées ont été déterminés *a posteriori* à la suite de nombreux essais qui ont conduit au choix de la configuration définitive décrite ci-dessus. Ces critères n'ont aucun caractère définitif. Ils ne sont expérimentaux et pourront servir de base à de futurs travaux dans ce domaine tant au niveau théorique (à démontrer ou infirmer) qu'au niveau expérimental (cela peut servir à déterminer un set d'entrées).

- toutes les entrées qui ont amené une amélioration sensible sur les tests d'erreur, ont un point commun. Ces entrées sont telles que prises séparément elles ont un effet prévisible sur le maillage de type :

$$d = f (W_{ij}, X)_{xi},$$

ou $f(W_{ij}, X)$ est une fonction monotone (décroissante ou croissante).

X est l'ensemble des entrées.

d est la densité de maillage.

J'ai essayé de déterminer l'influence des entrées prises séparément. Toutes les entrées ont été successivement mises à 0 (entraînant a priori des éléments finis de petite taille) et toutes les autres à 1. La taille des éléments finis obtenus ainsi a été comparée à celle obtenue avec toutes les entrées mises à 1. Les résultats n'ont pas été convaincants : la plupart des entrées n'amenaient pas une diminution de taille par rapport à celle de référence. Le réseau neuronal effectue des combinaisons complexes entre les différentes entrées, et l'influence séparée de chacune d'entre elles n'a pas pu clairement être mise en évidence.'

- le réseau neuronal n'a pas réussi à interpoler des fonctions complexes reliant par exemple les angles entre les différents segments les plus proches ou de cuivre. Il semblerait que pour que le réseau neuronal puisse prendre en compte une entrée, il faut que le concepteur puisse prédire à l'avance quelle sera son influence sur la densité de maillage. Néanmoins, comme nous le verrons lors de l'examen des résultats des phases d'entraînement, la combinaison de différentes entrées peut amener des résultats imprévisibles, voire contradictoires lors de compétition entre différentes entrées pour certains exemples.

VI Résultats obtenus.

VI.1. Modèle d'entraînement.

Le modèle utilisé pour entraîner le réseau neuronal, le "Ecore", est censé pouvoir communiquer au réseau neuronal la plupart des cas représentatifs de maillage au niveau géométrique. La pertinence de cet exemple sera discutée par la suite. Les grilles présentées ici ont été obtenues avec la meilleure configuration du réseau neuronal pour ma représentation.

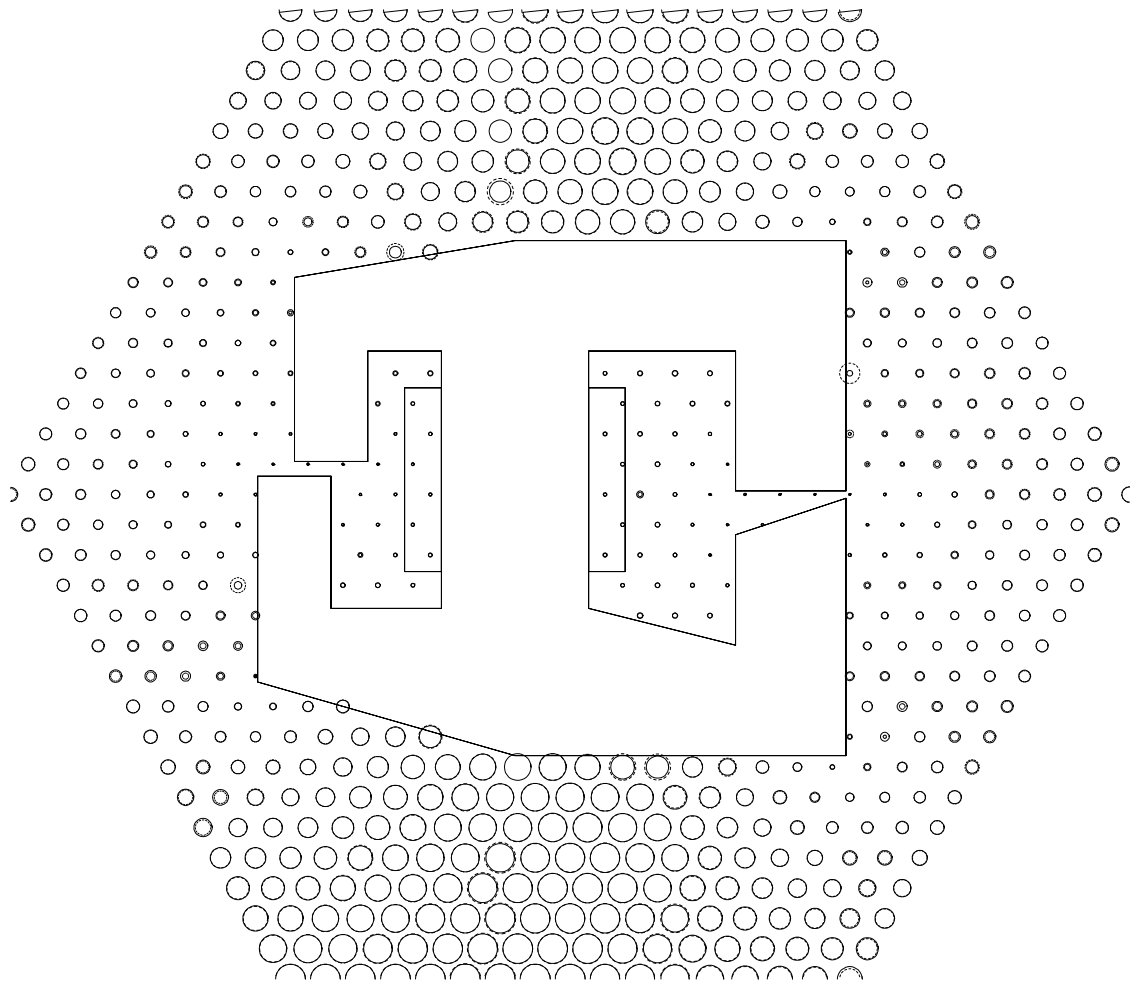
VI.1.1 Prémaillage obtenu dans l'air.

L'erreur obtenue sur l'exemple d'entraînement est de 8% et correspond à la configuration du réseau neuronal minimisant l'erreur sur le second exemple de référence, le "Gcore". Cet exemple-ci n'entraîne pas le réseau neuronal mais mesure l'efficacité sur un exemple inconnu. La configuration est retenue seulement si l'erreur sur le "Ecore" est inférieure à celle sur le "Gcore".

Taille des éléments finis :

_ Tailles de référence

... Tailles obtenues après entraînement du réseau neuronal. La surface des cercles représente la surface équivalente des triangles.

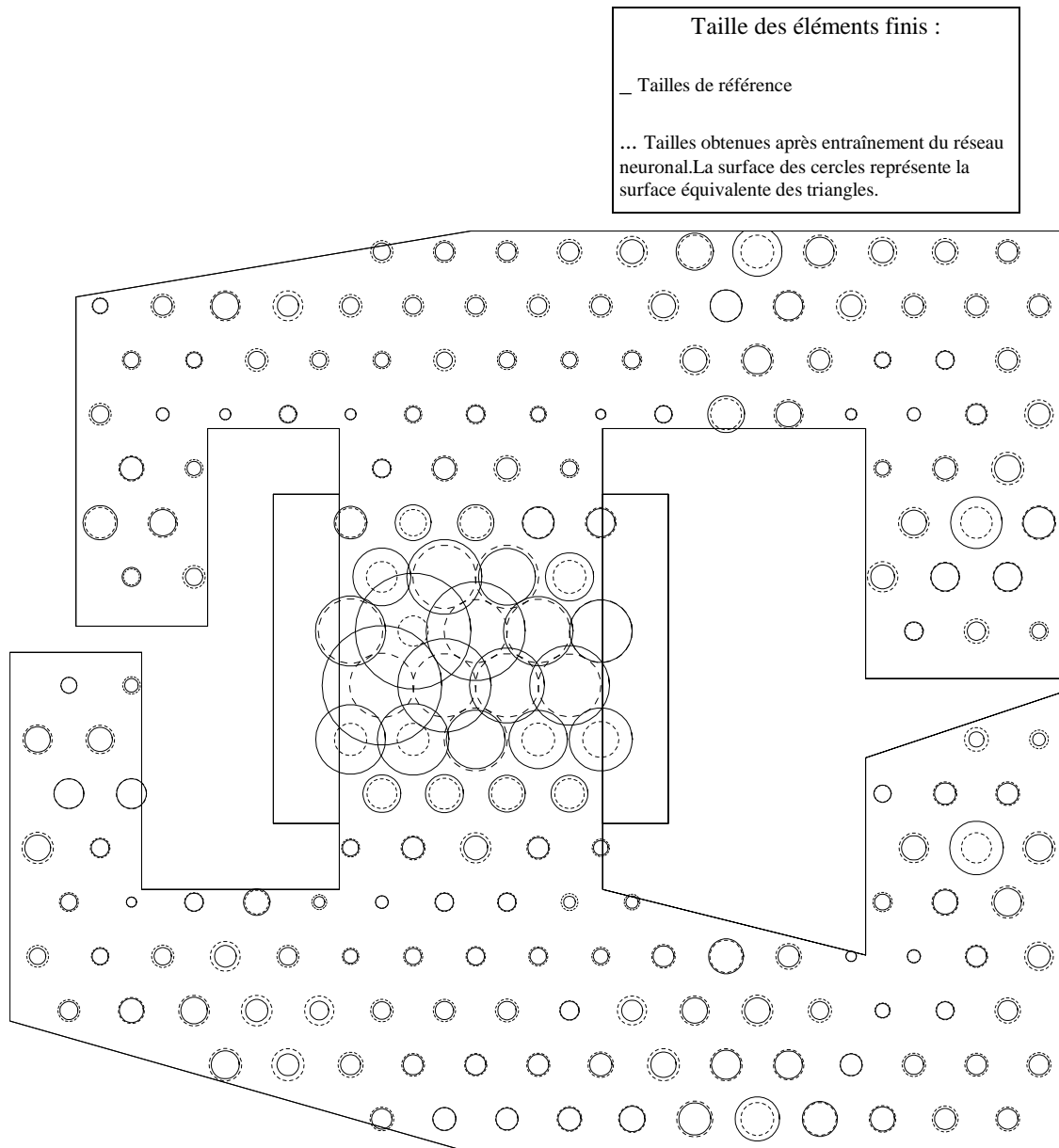


Grille de taille des éléments finis du Ecore pour l'air

Au niveau qualitatif, on peut constater que les principaux critères de maillage définis précédemment sont respectés : le maillage est fin aux alentours de l'entrefer, des singularités géométriques et près du cuivre. Néanmoins, près des conducteurs (dans les deux cavités), il est assez difficile (pour le réseau neuronal) de déterminer si le maillage est fin à cause de la proximité du cuivre, de l'entrefer ou des nombreuses singularités à cet endroit. On le constatera par la suite, cette ambiguïté amène des confusions entre les différentes entrées.

Les erreurs obtenues sur le maillage sont assez négligeables sauf pour des points isolés et situés à des emplacements peu importants pour le maillage (loin de l'entrefer et des coins).

VI.1.2 Prémaillage obtenu dans le fer.



Grille de taille des éléments finis du Ecore pour le fer

Le résultat obtenu pour le prémaillage du fer est de 11%. Ceci est moins bon que pour l'air, mais ce résultat peut s'expliquer par le fait que le prémaillage de référence amène une confusion lors de la phase d'entraînement. En effet, la taille des éléments finis est grande dans la colonne centrale

du "Ecore", et ce parce que le réseau neuronal a des difficultés à interpréter les entrées. Ces points sont très proches des conducteurs et les segments les plus proches sont relativement longs. Le réseau neuronal ne peut pas discerner l'origine de ce phénomène.

D'autre part, l'entrefer n'apparaît pas clairement dans cet exemple au niveau de la géométrie locale. Le maillage est satisfaisant dans ce cas, c'est à dire fin. Mais il est difficile de savoir à partir de l'exemple si cela est dû aux singularités (angles) dans cette région ou parce que les deux segments les plus proches non adjacents du point évoquent l'entrefer en raison de leur proximité.

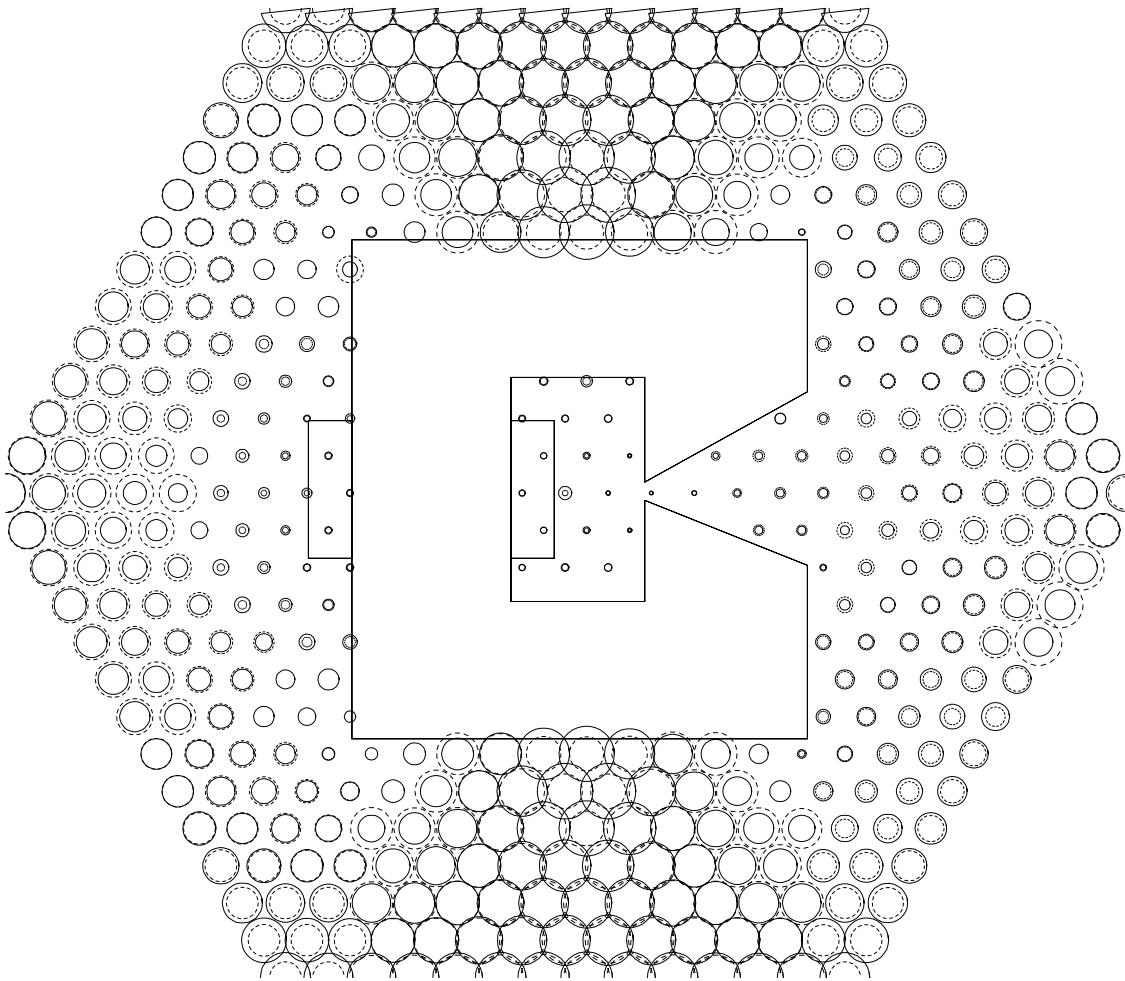
VI.2 Modèle de référence inconnu pour le réseau neuronal.

La méthode d'obtention de la meilleure configuration du réseau neuronal tient compte de cet exemple. En effet, la configuration retenue est telle que l'erreur sur Gcore soit minimale et que l'erreur sur Ecore soit inférieure à celle-ci. On peut donc considérer que même si l'exemple Gcore ne participe pas directement à l'entraînement du réseau neuronal, il y contribue de manière indirecte. Par conséquent il est clair que si on avait entraîné le réseau sur Ecore et retenu la configuration minimisant l'erreur sur cet exemple sans tenir compte de Gcore, on aurait eu des résultats d'erreur de moins de 5% sur Ecore, mais de plus de 15% sur Gcore. Par exemple j'ai obtenu avec 9 entrées une erreur test1 de 1.9% mais une erreur test2 de 16%. Cet essai correspond à une interpolation très précise de la fonction de prémaillage en fonction des entrées locales (le réseau neuronal), mais beaucoup moins adaptée à traiter d'autres exemples.

VI.2.1 Prémaillage obtenu dans l'air.

Ce prémaillage est correct, l'erreur obtenue est de 13.5 %. L'erreur de prémaillage n'est pas obtenue sur des points importants de la grille. On peut voir en effet que l'entrefer et les coins sont maillés correctement par rapport aux tailles de référence. Ce résultat a été obtenu à l'itération 2 879 000 lors de la phase d'entraînement. Ce programme a duré 6h environs sur la station de travail que j'utilisais.

Les légendes pour les figures suivantes sont identiques à celles des paragraphes précédents.

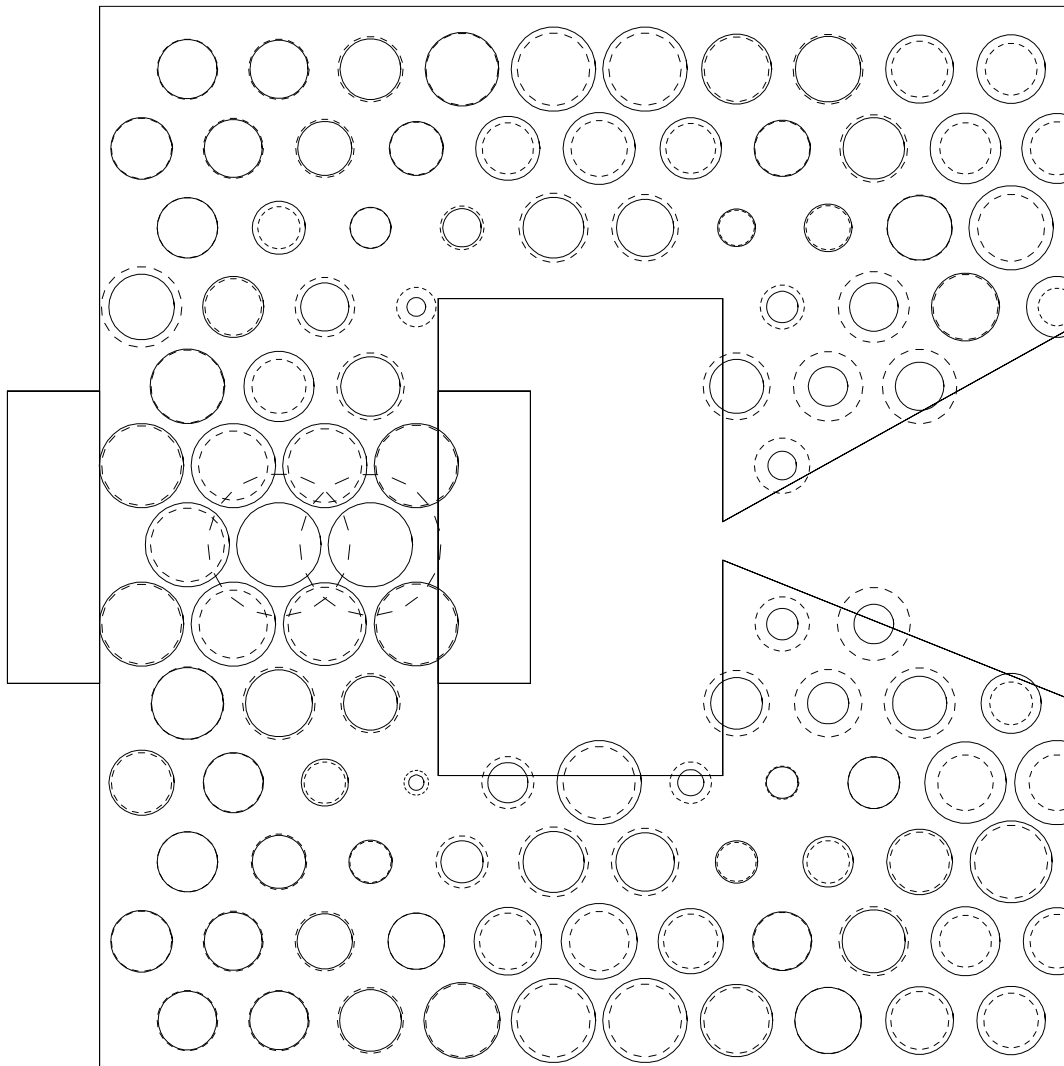


Grille de taille des éléments finis du Gcore pour l'air

VI.1.2 Prémaillage obtenu dans le fer.

Le prémaillage du fer de Gcore est correct à 13.2 % près. On assiste au même phénomène que pour le Ecore dans la colonne supportant les conducteurs. Les principales erreurs ont été obtenues au voisinage de l'entrefer et dans la colonne centrale. Ce n'est pas surprenant étant donné l'exemple d'entraînement (Ecore) :

La colonne d l'Ecore a une très faible densité de maillage, le réseau neuronal étant ainsi entraîné, le prémaillage obtenu sur le Gcore n'est pas surprenant. De même pour l'entrefer, l'exemple d'entraînement ne le fait pas apparaître assez clairement pour que le réseau neuronal l'applique au Gcore.



Grille de taille des éléments finis du Gcore pour le fer.

VI.3 Autres exemples.

Des essais ont été effectués sur des exemples dont le prémaillage idéal n'est pas connu. Dans les deux cas, pour le moteur à courant continu et le disjoncteur, les résultats sont corrects. L'entrefer est maillé correctement dans l'air pour l'un comme pour l'autre.

Pour le moteur à courant continu, le prémaillage du rotor (partie fer) n'est pas satisfaisante en certains points. Néanmoins, cet exemple comportant des arcs de cercle démontre la capacité de la représentation utilisée à généraliser pour des exemples très différents de l'exemple d'entraînement.

On peut voir aussi qu'en certains points, le réseau neuronal n'a pas pu déterminer de densité de maillage (points de petit diamètre). L'examen des sorties non décodées du réseau neuronal montre qu'elles sont de valeur trop faible (toutes inférieures au seuil de 0,2) pour être décodées.

Dans ce cas la valeur de sortie décodée est 0, car celles des sorties codées sont considérées comme des bruits de calcul.

Prémaillage obtenu pour un moteur à courant continu

Prémaillage obtenu pour un disjoncteur

Conclusion.

L'efficacité du réseau neuronal dépend énormément de l'exemple d'entraînement. Celui-ci se doit de couvrir si possible tous les cas de figure rencontrés lors d'un maillage. Mais il se doit en plus de posséder des vertus "pédagogiques" dans le sens que le réseau neuronal ne soit jamais embrouillé entre plusieurs entrées lors de la phase d'entraînement. De là, rien n'empêche d'utiliser un exemple totalement irréal, mais qui entraîne le mieux possible le réseau neuronal.

Néanmoins, aussi bon que soit cet l'exemple, il ne peut couvrir tous les exemples de problèmes de magnétostatique. Un expert ne maille pas avec la même démarche un disjoncteur ou un moteur, et on ne peut pas demander à un réseau neuronal entraîné avec un seul type d'appareil de le faire. Malgré le fait que les prémaillages obtenus sur le moteur et le disjoncteur semblent corrects, il faudrait disposer de différents réseaux neuronaux entraînés pour différentes classes de problèmes. De manière concrète, il suffirait de demander à l'utilisateur de choisir la classe de problème parmi un choix couvrant son domaine d'utilisation. Il serait aussi possible d'utiliser un réseau neuronal à apprentissage de type compétitif (de type Kohonen par exemple) pour le classifier.

La limite de précision du prémaillage semble être comprise (au vu des essais) entre 5 et 10%. Les critères de type géométrique ne pourront en aucun cas résoudre les équations de Maxwell, qui doivent être utilisées pour déterminer un maillage idéal. Néanmoins la détermination de ce prémaillage dure environs deux minutes sur une station de travail Sun Sparc Ipx, ce qui apporte un gain de temps non négligeable par rapport à celui pris par les itérations du solveur de Magnet.

Annexe

1.Detail de la structure de donnée utilisée pour décrire un segment :

```
struct superseg
```

```
{
```

```
[ Données relatives au segment ]
```

```
[ (x,y) et (xe,ye) sont les extrémités initiales et finales du segment associé]
```

```
float x ;
```

```
float y ;
```

```
float xe ;
```

```
float ye ;
```

```
[ alpha est l'angle de la tangente du début de segment ]
```

```
float alpha ;
```

```
[ ir est la courbure du segment. Ce choix s'impose par le fait que l'on peut alors représenter indifféremment un arc de cercle ou un segment de droite ( ir=0 dans ce cas ) ]
```

```
float ir ;
```

```
[ id représente l'inverse de la longueur du segment. Cette représentation permet une plus grande précision pour les petites longueurs, avec le format numérique choisi. ]
```

```
float id ;
```

```
[ rmat,lmat :type du matériau qui est a droite et a gauche du segment ]
```

```
int rmat ;
```

```
int lmat ;
```

```
[ Relations entre le segment et le point de maillage ]
```

```
[ (xh,yh) sont les coordonnées du point du segment, le plus proche du point ou on veut connaître la densité de prémaillage. d représente la distance entre ces deux points. ]
```

```
float xh ;
```

```
float yh ;
```

```
float d ;
```

```
[ Relations entre les différents segments, permettant de reconstituer la géométrie du problème ]
```

```
[ Prochain segment de la liste (cela a un signification différente selon ou on se trouve dans le programme ]
```

```
struct superseg *next ;
```

```
[ Segment oppose au segment courant ( ceci est utilise dans la reconstitution de contours fermes ) ]
```

```
struct superseg *op ;
```

[Segments se trouvant a gauche et a droite du segment dans un pièce de contour ferme. La pièce étant parcourue dans le sens trigonométrique lorsqu'on se trouve a l'intérieur, le segment *right est avant, et le segment *left est après]

```
    struct superseg *left ;
    struct superseg *right ;
};
```

2. Exemple de fichier graphique de type SKX (nouveau format graphique de MAGNET)

[type du segment : arc de cercle ou segment de droite]

LINE

[coordonnées (x,y) début de segment / (xe,ye) fin de segment matériau a droite / matériau a gauche du segment]

```
0.4094070  -0.6844300E-01  0.4870530  -0.3582210  0  2
```

LINE

```
0.4870530  -0.3582210  0.7768300  -0.2805750  0  2
```

LINE

```
-0.3887780  0.1454300  -0.2954420  0.5209400E-01  0  1
```

ARC

[coordonnées (xc,yc) du centre de l'arc / rayon de courbure de l'arc / angles en degrés du début et de la fin de l'arc, sachant que cette orientation est toujours positive dans le sens dans le sens trigonométrique]

```
0.8609920  -0.4014880  0.4999977E-01  -164.9999  -24.99991  0  1
```

ARC

```
0. 0. 0.9999999  -25.00004  55.00000  0  1
```

ARC

```
0.5448980  0.7781950  0.4999934E-01  54.99995  195.0001  0  1
```

Bibliographie.

A Collection of programs to simulate a neural network for the mesh discretization problem,
Derek Dyck, Dept of Electrical Engineering, McGill University, Montréal, Canada
1990.

Determining an approximate finite element mesh density using neural network techniques
D.N.Dyck, D.A.Lowther, S McFee, Dept of Electrical Engineering, McGill University, Montreal,
Canada, 1990.

Determining finite element mesh density from problem specification using neural network,
Derek Dyck, Master's thesis of Engineering, CAD laboratory, Dept. of Electrical engineering,
nov 1990.

Neurocomputing,
Robert Hecht-Nielsen, HNC, Inc. and university of California, San Diego,
Addison-Wesly Publishing compagny 1990.

Foundations of Neural Networks,
Tarun Khana, Addison-Wesley Publishing Compagny Inc 1990.