Université de Jijel
Département d'électrotechnique
Master II Machines électriques
Module : Conception assistée par ordinateur des machines électriques

TP N°1
Architecture d'un logiciel de CAO à base d'éléments finis
-sous environnement Matlab-

I. Architecture d'un logiciel de CAO

Au niveau logiciel, un environnement de CAO est constitué de trois étapes qui sont:
- Pré-processeur;
- Processeur de résolution;
- Post-processeur ou processeur d'exploitation.

I.1. Le Pré-processeur

Le rôle de ce module est la réalisation des fonctions suivantes sur l'objet à modéliser:
- Dessin de la géométrie;
- Maillage;
- Affectation des caractéristiques physiques.

Le dessin de la géométrie est réalisée par des opérations (union, intersection,….) sur des objets de base (polygone, cercle, cylindre…).

Le maillage consiste à déterminer l'ensemble des nœuds et des éléments qui composent le domaine d'étude. Cette opération est réalisée par le mailleur.

L'affectation des caractéristiques physiques sert à:

-définir le type du problème (magnétostatique, magnétodynamique,….);

-décrire les caractéristiques physiques des matériaux, telles que: perméabilité magnétique, conductivité électrique, densité de courant,….etc

-décrire les conditions aux limites (conditions de Dirichlet, conditions de Neumann,….).

I.2. Le Processeur

C'est le module chargé de la résolution du système d'équations linéaires ou non linéaires. Il produit les valeurs des inconnues en chaque nœud du maillage.

I.3. Le Post-processeur

Une fois la résolution est terminée, on dispose aux nœuds du maillage de la variable d'état, celle-ci n'est pas directement exploitable. Donc, le post-processeur permet de visualiser des grandeurs locales scalaires ou vectorielles (induction magnétique, champ magnétique,…etc), il permet aussi de calculer des grandeurs globales comme les forces et les couples qui s'exercent sur une région ou sur tout le domaine.

II. Matlab pour la résolution des EDP

Matlab est un logiciel d'analyse par simulation qui permet de manipuler les opérations mathématiques et contient des bases de calcul matriciel très développées.

Matlab est un outil de modélisation à base de la méthode des éléments finis grasse à la boite à outils PDE Toolbox. Elle dispose des fonctions Matlab qui permettent de résoudre les équations aux dérivées partielles sur une surface bidimensionnelle en utilisant la méthode des éléments finis.

II.1. La géométrie

PDE Toolbox permet de modéliser une très grande variété de dispositifs électromagnétiques. Une surface 2D peut être réalisée par l'utilisation des droites et des cercles.

On désigne par 1 les cercles et par 2 les segments.

Cercle et arc=[1 Xdébut Xfin Ydébut Yfin région_gauche région_droit Xcentre Ycentre rayon]';
Segment=[2 Xdébut Xfin Ydébut Yfin région_gauche région_droit]';

<u>Exemple :</u>

```
L1=[2 -1e-3 -1e-3 -0.6 1 0 1]';
L2=[2 -1e-3 1 1 1 0 1]';
L3=[2 1 1 1 -0.6 0 1]';
L4=[2 1 -1e-3 -0.6 -0.6 0 1]';
L=[L1 L2 L3 L4];
figure (1)
pdegplot(L);
```

II.2. Le maillage

Le maillage de la géométrie d'un dispositif est décrit par les trois matrices de nœuds, de connectivités et de triangles (p,e,t).

```
[p,e,t]=initmesh(L);
[p,e,t]=refinemesh(L,p,e,t,5);
x=p(1,:);
y=p(2,:);
```

```
nbtrig=size(t,2);
% size(t)=4 2343
figure (2)
pdeplot(p,e,t);
```

II.3. Les coefficients

Il faut affecter à chaque élément triangulaire du maillage les coefficients correspondants (perméabilité, conductivité et courant) qui diffèrent d'une région à une autre.

```
c=zeros(1,nbtrig);
a=zeros(1,nbtrig);
f=zeros(1,nbtrig);
for m=1:nbtrig
      if (t(4,m))==1
       c(1,m)=nu0;
      end
      if (t(4,m))==2
       c(1,m)=nu1;
%****
      if(t(4,m))==1
      f(1,m)=f1;
      end
end
```

II.4. Construction des matrices globales

Les matrices élémentaires sont assemblées afin de construire les matrices globales du système algébrique.

```
[K,M,F]=assema(p,t,c,a,f);
U=K+M;
```

II.5. Les conditions aux limites

La condition issue de la méthode développée, a été programmée sur toute la périphérie extérieure du domaine d'étude.

```
i1=find(p(2,:)==-0.6 & p(1,:)~=-1e-3);
i2=find(p(1,:)== 1 & p(2,:)~=-0.6);
i3=find(p(2,:)== 1 & p(1,:)~= 1);
i4=find(p(1,:)== -1e-3 & p(2,:)~=1);
i0=[i1 i2 i3 i4];
```

```
for j1=1:length(i0)
   A(i0(j1))=0;
   U(:,i0(j1))=0;
   U(i0(j1),:)=0;
   U(i0(j1),i0(j1))=1;
   F=F-U(:,i0(j1))*A(i0(j1));
end
```

II.6. La résolution

La résolution est représentée par un vecteur qui donne les valeurs du potentiel vecteur magnétique dans tous les nœuds du maillage.

```
A=U\F;
pdeplot(p,e,t,'xydata',A,'xystyle','off','contour','on','levels',50,'colorbar','on')
```

II.7. L'exploitation

L'équation magnétostatique ou magnétodynamique est formulée en terme de potentiel vecteur A, ce dernier ne constitue pas une grandeur physique directement exploitable. Il est utilisé pour construire toutes les grandeurs d'exploitation locales ou globales qui nous intéressent.

A partir des valeurs du potentiel vecteur, on peut déterminer les valeurs de l'induction magnétique en procédant comme suit :

```
[DXA,DYA]=pdegrad(p,t,A);
Bx=DYA;
By= -DXA;
CBX=pdeprtni(p,t,Bx);
CBY=pdeprtni(p,t,By);
CBT=sqrt(CBX.^2+CBY.^2);
xx=0:0.0001:0.25;
yy=0.3:0.001:0.31;
BBX=TRI2GRID(p,t,CBX,xx,yy);
BBY=TRI2GRID(p,t,CBY,xx,yy);
BBT=sqrt(BBX.^2+BBY.^2);
```