

CODE CESC_FMM : manuel d'utilisation

N. Bartoli, F. Collino et F. Millot

CERFACS REPORT TR/EMC/04/33

Comment utiliser le code CESC_FMM ?

avril 2004

Ce document constitue la documentation technique relative au code de calcul FMM développé au CERFACS. Ce code a été installé au CNES en avril 2004.

Nous décrivons l'installation du code et son utilisation (en séquentiel ou parallèle), le préconditionneur et le module qui permet de calculer le champ crée par une antenne sur un satellite à partir de la connaissance de son champ lointain (mesures en chambre anéchoïde de l'antenne).

Nous remercions tout particulièrement I. d'Ast et M'B Fares pour l'aide qu'ils nous ont apportée lors de ces travaux de réalisation des logiciels FMM.

Equipe Electromagnétisme CERFACS

N. Bartoli bartoli@cerfacs.fr

F. Collino collino@cerfacs.fr

F. Millot millot@cerfacs.fr

Table des matières

1	Le code CESC_FMM	3
2	Comment installer le code ?	3
3	Comment compiler CESC_FMM ?	4
4	Descriptif sommaire de l'utilisation de CESC_FMM	6
5	Entrées-sorties du code	7
5.1	Entrées	7
5.2	Sorties à la fin du calcul du préconditionneur	7
5.3	Sorties à la fin de la résolution	7
6	Options d'exécution des programmes	8
6.1	Nombre de processeurs	8
6.2	Fichier de géométrie	8
6.3	Fichier de données	8
7	Fichier de données de type CESC.dat	8
7.1	Clés obligatoires de CESC.dat	8
7.2	Les valeurs par défaut de CESC.dat	10
7.3	Modifier les valeurs par défaut	12
7.3.1	Nom des fichiers de sortie	12
7.3.2	Nom des fichiers de courant	12
7.3.3	Règle de quadrature pour le calcul des intégrales	13
7.3.4	Règle de quadrature pour le calcul du second membre	13
7.3.5	Définition de l'octree	14
7.3.6	Discrétisation de la sphère unité	14
7.3.7	Choix de la méthode d'interpolation et de filtrage	16
7.3.8	Choix du type d'algorithme multipolaire	17
7.3.9	Choix de la méthode d'inversion	17
7.3.10	Restitution des temps de calcul	19
7.4	Gestion de la matrice des interactions proches	20
7.4.1	Définition des paramètres pour le calcul de la matrice proche	20
7.4.2	Fichiers de sortie	21
8	Le préconditionneur	22
8.1	Nature du préconditionneur	22
8.2	Fichier de sortie	22
8.3	Clés obligatoires pour le calcul du préconditionneur	23
8.4	Les valeurs par défaut de CESC.dat	23
8.5	Modifier les valeurs par défaut	23
8.5.1	Construction et stockage du préconditionneur	23
8.5.2	Description pour le préconditionneur de type topologique	24
8.5.3	Description pour le préconditionneur de type géométrique	24
8.5.4	Choix des bandes	25

9	Fichier de données de type CESC.dat pour le module antenne-satellite	26
9.1	Clé obligatoire de CESC.dat pour le module antenne-satellite	26
9.2	Clés optionnelles de CESC.dat pour le module antenne-satellite	27
9.3	Comparaison dans le cas d'un dipôle	29
10	Calcul des champs lointains	29
10.1	La clé de la SER bistatique	30
10.2	Cas d'un dipôle ou d'une antenne	31
10.3	Fichiers de sortie	31
11	Exemples de fichiers de données	31
11.1	Valeurs minimales	31
11.2	Valeurs maximales	34
12	Un exemple de validation dans le cas de la sphère	45
12.1	Les fichiers d'entrée	45
12.2	L'exécution des programmes	45
12.3	Les fichiers de sortie	47
12.4	Le fichier CESC.dat	47
12.5	La convergence	49
13	Un exemple de validation dans le cas du satellite Demeter	49
13.1	Les fichiers d'entrée	49
13.2	L'exécution des programmes	49
13.3	Les fichiers de sortie	50
13.4	Le fichier CESC.dat	50
13.5	La convergence	54
14	À partir d'un fichier CESC.dat	55

1 Le code CESC_FMM

CESSC_FMM est un code pour le calcul d'ondes électromagnétiques en régime fréquentiel. La méthode numérique utilisée s'appuie sur une équation intégrale que l'on résout par une méthode itérative couplée à des calculs multipôles. On renvoie aux références [11], [3], [8] et [6] pour une description de cette méthode. Le but de ce fascicule est de donner des indications sur l'installation et l'utilisation du code, ainsi que sur ses fonctionnalités.

Ce code de calcul possède une version out-of-core. Il peut fonctionner soit en mode séquentiel soit en mode parallèle. De plus, il a été implémenté sur de nombreuses architectures.

Présenté schématiquement, le code de calcul part d'un fichier de géométrie avec sa connectique (fichier de type *.Geomd ou de type *.Geom) et d'un fichier descriptif de l'expérience que l'on veut réaliser (fichier de type CESC.dat) : alimentation, fréquence, type de source, ... puis calcule les grandeurs suivantes

- les flux du courant électrique à travers les arêtes, pour les surfaces métalliques ouvertes ou fermées ;
- la SER bistatique.

Remarque 1 CESC_FMM possédant des fonctionnalités et des entrées communes avec le code généraliste CESC, il peut être intéressant de regarder la référence [2].

Remarque 2 Le code CESC_FMM ne possède pas toutes les fonctionnalités de CESC. Il ne peut pas traiter des diélectriques ni des fils par exemple. Il n'y a pas non plus de prise en compte des symétries. Ces limitations ne sont pas de nature méthodologique mais plus simplement dues à un manque de temps pour leur mise en œuvre.

2 Comment installer le code ?

Le code de calcul est contenu dans le fichier nommé `cesc_fmm.tar`. Pour installer ce code sur une machine (sous Unix), il convient de taper

- `cp /home/elec/FMM_CODES/CODE/cesc_fmm.tar`
- `tar xvf cesc_fmm.tar`,

Un répertoire CESC_FMM est créé. Ce dernier contient entre autres les fichiers "source" et aussi quelques exemples de fichiers de données permettant de faire les simulations. Sous le répertoire CESC_FMM, un certain nombre de répertoires sont créés ; ils sont schématisés sur les figures 1 et 2. On distingue :

- `geom` : un répertoire de géométrie contenant éventuellement les fichiers de géométrie.
- `cesc` : un répertoire de fichiers de données de type `CESSC.dat`.
- `result` : un répertoire où les résultats seront stockés.
- `config` : un répertoire de configuration qui permet de créer les bonnes options de compilation et d'exécution du programme en fonction de l'architecture choisie.
- `lib` : un répertoire où les bibliothèques vont être stockées.
- `bin` : un répertoire où les binaires vont être stockés.
- `src` : un répertoire contenant des sous-répertoires où tous les fichiers sources Fortran 90, Fortran 77 et C se trouvent.

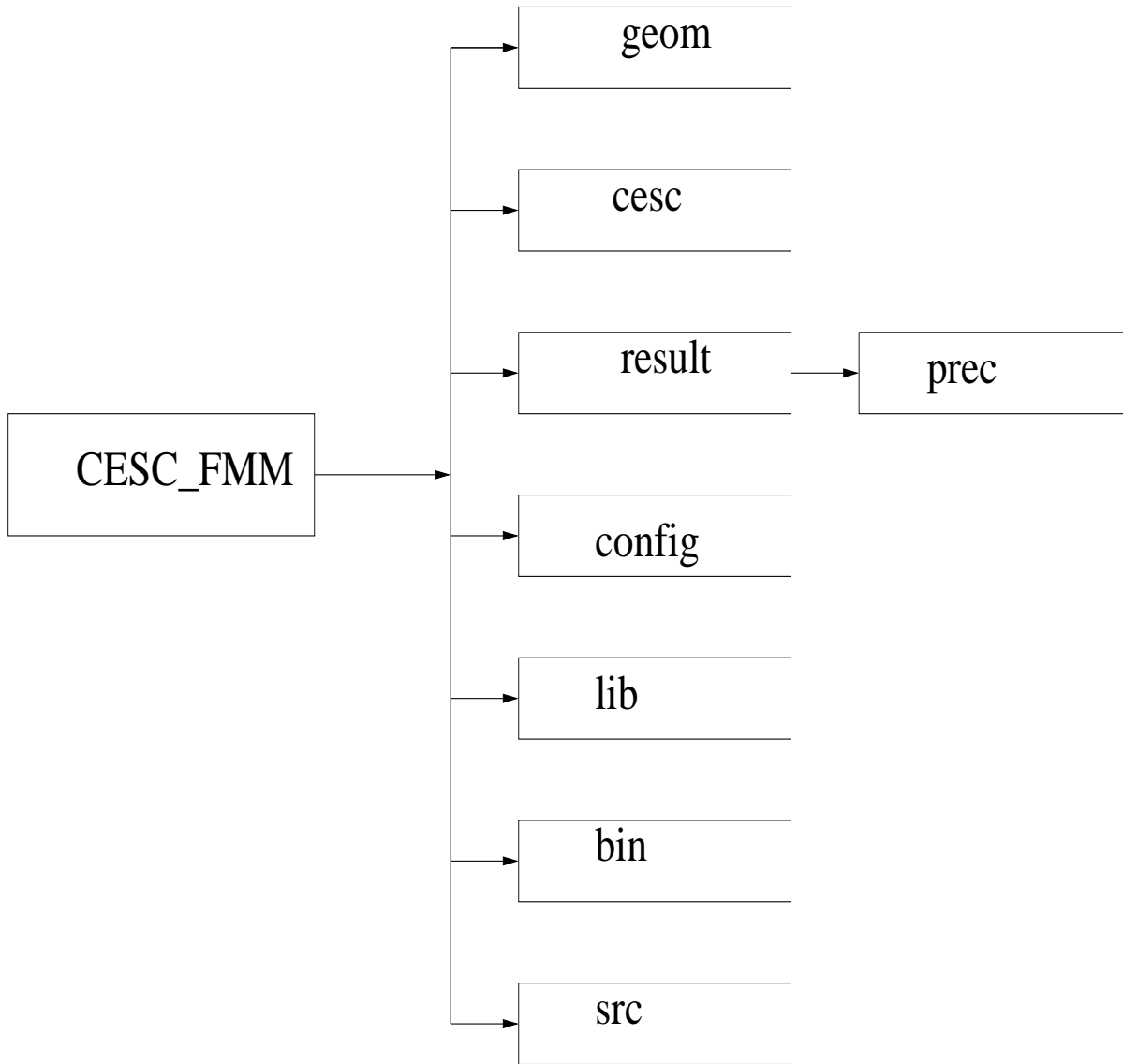


FIG. 1 – Les répertoires de CESC_FMM

Remarque 3 *On verra plus loin que les trois premiers répertoires ne sont pas forcément nécessaires car ils peuvent être redirigés par l'utilisateur.*

Le répertoire contient également des **Makefile** qui vont être utilisés pour la compilation. Tous ces **Makefile** sont écrits en **gnumake**. Pour compiler, il faut faire soit **make** ou **gmake**.

3 Comment compiler CESC_FMM ?

Pour compiler CESC_FMM, il faut faire

1. `cd CESC_FMM`
2. `make clean` ou `gmake clean`
3. `.config_exe machine mode`
4. `make` ou `gmake`

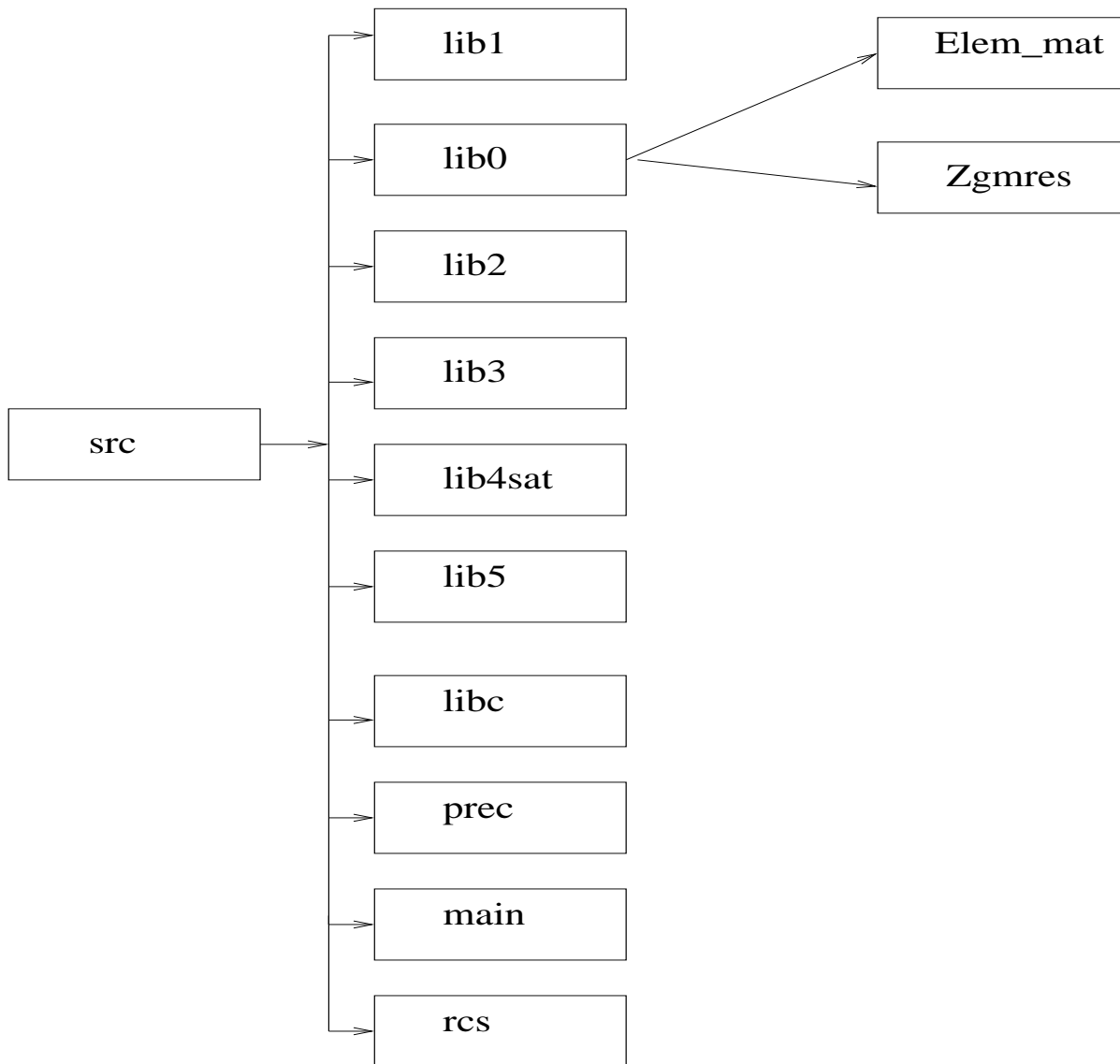


FIG. 2 – Les répertoires de CESC_FMM/src

- La chaîne `mode` doit être égale à `parallel` ou `sequentiel`.
- La chaîne `machine` peut être égale à `rsk6` (IBM), `o2000` (origin 2000) ou `compaq`. Pour toute autre machine, il faut créer soi-même le fichier `config/config.mk`. Ce fichier contient les commandes spécifiques à chaque compilateur ; par exemple le nom du compilateur fortran 90 (`f90` ou `pgf90` ou `rsk6`), le positionnement des bibliothèques ou encore la commande de redirection des modules (`-I` ou `-M`).
- Les bibliothèques `FFTW`, `LAPACK`, `MPI` ainsi que `SCALAPACK` sont utilisées par `CESC_FMM` : leur localisation doit être indiquée dans `config/config.mk`.

Remarque 4 *La bibliothèque FFTW est disponible à l'adresse suivante <http://www.fftw.org/>. On utilise la version `fftw-3.0.1`.*

Tous les exécutables seront créés sous le répertoire

`CESC_FMM/bin`

La liste des exécutables est :

mode séquentiel	mode parallèle	fonctionnalité
<code>cesc_prec_exe</code>	<code>pcesc_prec_exe</code>	Calcul du préconditionneur
<code>cesc_ite_exe</code>	<code>pcesc_ite_exe</code>	Résolution par multipôles en itératif
<code>cesc_rcs_exe</code>		Calcul de la bistatique

4 Descriptif sommaire de l'utilisation de CESC_FMM

Schématiquement, on part d'un fichier de géométrie avec sa connectique ; celui-ci contient le descriptif du maillage de l'objet diffractant. On suppose que le maillage est assez fin pour bien décrire l'interaction de l'objet avec l'onde à la fréquence d'utilisation (règle des 5 points par longueur d'onde minimum). Ensuite, on crée un fichier de type `CESC.dat` qui contient tout le descriptif de la simulation que l'on veut effectuer. La méthode numérique de résolution du (ou des) système(s) linéaire(s) étant itérative, il est judicieux de construire un préconditionneur pour accélérer la convergence ; c'est la première étape du calcul multipôle (on peut s'en affranchir mais ce n'est pas très conseillé). La seconde étape est le calcul des courants ; c'est l'étape de résolution du système linéaire. Enfin, la troisième étape est le calcul des champs lointains (qui se fait exactement de la même manière que pour `CESC`).

En résumé, pour utiliser `CESC_FMM` via les localisations des fichiers par défaut :

- Créer d'abord un fichier de géométrie que l'on range dans le répertoire `geom`.
- Créer le fichier de données `cesc/CESC.dat` ou éventuellement changer les données directement dans le fichier `cesc/CESC.dat` s'il existe déjà.
- Créer le préconditionneur en lançant l'exécutable `bin/cesc_prec_exe` (ou `bin/pcesc_prec_exe` pour un calcul en parallèle)
- Lancer alors le processus itératif avec l'exécutable `bin/cesc_ite_exe` (ou `bin/pcesc_ite_exe` pour un calcul en parallèle).

5 Entrées-sorties du code

5.1 Entrées

- *Le fichier de géométrie* : par défaut, on utilise le format allégé `*.Geom` mais il peut être aussi de type `*.Geomd`. L’extension du fichier de géométrie est définie avec les options d’exécution (cf. paragraphe §6).

Par défaut, ce fichier se trouve sous le répertoire `geom`. On peut toutefois changer cette localisation en définissant la variable d’environnement :

CESC_GEOM

(il suffit de taper “`setenv CESC_GEOM nom_de_repertoire`” ou “`export CESC_GEOM`” suivant le shell utilisé)

- *Le fichier de données* : par défaut, ce fichier est `/cesc/CESC.dat`. On peut néanmoins changer de fichier en définissant la variable d’environnement

CESC_DATA.

Ces deux fichiers sont indispensables pour l’exécution des trois programmes. On va ensuite distinguer les différents types de sortie en fonction du programme utilisé.

5.2 Sorties à la fin du calcul du préconditionneur

Nous utilisons un préconditionneur de type Sparse Approximate Inverse (SPAI). Il s’agit d’une matrice creuse complexe M approchant l’inverse de la matrice Z associée au système linéaire que l’on cherche à résoudre, [5], [4], [7].

La construction du préconditionneur est décrite dans le paragraphe §8. Pour mémoriser cette matrice creuse, nous avons choisi d’utiliser une structure `SPARSE`, reposant sur quatre caractéristiques. Ces dernières sont stockées sur quatre fichiers différents sous le répertoire

`/CESC_FMM/result/prec/`

Les noms des fichiers sont fixés suivant la simulation que l’on veut effectuer et seront décrits ultérieurement au paragraphe §8.2.

5.3 Sorties à la fin de la résolution

- Le fichier contenant les informations sur le déroulement de la simulation. Par défaut, le fichier est rangé dans le répertoire

`/CESC_FMM/result/`

On peut changer cette localisation en définissant la variable d’environnement :

CESC_RESULT

Le nom de ce fichier est décrit au paragraphe §7.2 et §7.3.1.

- Le fichier contenant la matrice proche. Par défaut, le fichier est rangé dans le répertoire

`/CESC_FMM/result/prec/`

On peut changer cette localisation en définissant la variable d’environnement :

CESC_RESULT/prec

Le nom de ce fichier est décrit au paragraphe §7.4.

- Le fichier contenant la valeur du courant. Ce fichier est rangé dans le répertoire

`/CESC_FMM/result/`

On peut changer de localisation en définissant la variable d'environnement :

CESC_CURRENT

Le nom de ce fichier est décrit au paragraphe §7.3.2.

6 Options d'exécution des programmes

6.1 Nombre de processeurs

On ne peut faire tourner la version parallèle `pcesc_ite_exe` que sur un nombre pair de processeurs. Si on ne dispose que d'un nombre impair de processeurs, il faut alors ajouter l'option `-mdebug`. Ce qui conduit à taper `pcesc_ite_exe -mdebug`

6.2 Fichier de géométrie

Les deux codes de calcul utilisent un fichier de géométrie avec sa connectique (fichier de type `*.Geomd` ou de type `Geom`). Par défaut on utilise un fichier de type `Geom`. Si on veut changer de type de fichier `*.Geomd`, il faut alors taper `pcesc_ite_exe -gGeomd`

6.3 Fichier de données

Par défaut le fichier de données est sous le répertoire `cesc` et porte le nom `CESC.dat`. On peut faire tourner les applications en changeant le nom du fichier de données. Par exemple si ce fichier de données s'appelle `CESC_min.dat`, on tape

```
pcesc_ite_exe -dCESC_min.dat
```

7 Fichier de données de type CESC.dat

Le fichier `CESC.dat` est un fichier à accès direct dont les accès sont pilotés par un certain nombre de mots clés tous commençant par le signe `#`. Ces clés peuvent être obligatoires (on doit les définir sinon la simulation ne peut pas s'effectuer) ou de type facultatif (dans ce cas, la simulation s'opérera avec des valeurs par défaut). Après chaque clé, doivent se trouver un certains nombre de lignes contenant des informations de type bien défini (entiers, chaîne de caractères, nombre réels, ...). Si on se trompe sur la nature ou le nombre de ces informations, normalement, un message d'erreur s'affiche à l'exécution. On se propose ici de donner un descriptif des clés et de leurs données associées.

7.1 Clés obligatoires de CESC.dat

Nous allons décrire ce que doit obligatoirement contenir le fichier de données de description de la simulation. Ce fichier doit obligatoirement contenir les valeurs suivantes.

- Le nom du fichier de géométrie. C'est l'objet de la carte `#INPUTFILE`. Par exemple

```
#INPUTFILE
```

```
'deux_spheres'
```

peut signifier que :

- le fichier de géométrie se nomme `FMM/geom/deux_spheres.Geom` par défaut ou `$CESC_GEOM/deux_spheres.Geom` si la variable d'environnement `CESC_GEOM` a été initialisée.
- ou le fichier de géométrie se nomme `FMM/geom/deux_spheres.Geomd` par défaut ou `$CESC_GEOM/deux_spheres.Geomd` si la variable d'environnement `CESC_GEOM` a été initialisée. Il faut alors lancer l'exécution avec l'option `-gGeomd`.

- Le nombre d'onde

```
#WAVENUMBER
```

```
1          ! nombre de nombre d'ondes
```

```
0.3       ! valeur du nombre d'onde
```

signifie que le nombre d'onde dans le vide k est fixé égal à 0.3 par mètre.

Si l'on ne connaît que la fréquence F en Hertz, on a

$$k = 2\pi \frac{F}{c_0}, \quad c_0 = 2.9992735610^8 \text{ kms}^{-1}.$$

- Le type d'équation intégrale choisie

debut du commentaire dans le fichier de donnees :

```
The type of equation to be solved
```

```
you must give
```

```
the type of equation
```

```
'MASS' 'WEDGE' 'EFIE' 'CFIE' 'MFIE' 'GENE'
```

```
if 'CFIE' you must give alpha_cfie
```

```
if 'GENE' you must give the 5 numbers :
```

```
c_mass c_wedge c_simple c_double c_double_wedge
```

```
CFIE=(1-alpha)*MFIE + alpha*EFIE
```

```
alpha = 1      =====> EFIE
```

```
alpha = 0      =====> MFIE
```

```
0 < alpha < 1 =====> CFIE ( alpha=0.2 )
```

```
fin du commentaire dans le fichier de donnees
```

Par exemple,

```
#EQUATION_TYPE
```

```
CFIE
```

```
0.2
```

signifie que l'on va résoudre une équation de type CFIE avec un coefficient de 0.2.

- Le choix de la source

```
- #PLANE_INCIDENTFIELD
```

```
<Number of excitations>
```

```
<Index> <THETA> <PHI> <Re(ETHETA)> <Im(ETHETA)> <Re(EPHI)>
```

```
<Im(EPHI)>
```

Dans ce cas, l'onde incidente est une onde plane qui attaque le profil par le haut et de biais

$$E^{inc}(\vec{x}, t) = (E_\theta \hat{\theta} + E_\varphi \hat{\varphi}) e^{-i(\omega t + k\hat{r}\cdot\vec{x})} \quad (1)$$

où le vecteur d'onde est $(-k\hat{r})$ et

$$\hat{r} = \begin{bmatrix} \sin \theta \cos \varphi \\ \sin \theta \sin \varphi \\ \cos \theta \end{bmatrix}, \quad \hat{\theta} = \begin{bmatrix} \cos \theta \cos \varphi \\ \cos \theta \sin \varphi \\ -\sin \theta \end{bmatrix}, \quad \hat{\varphi} = \begin{bmatrix} -\sin \varphi \\ \cos \varphi \\ 0 \end{bmatrix},$$

– **#DIPOLE_INCIDENTFIELD**

<Number of excitations>

<Index> <IA(1)> <IA(2)> <IA(3)> <X0(1)> <X0(2)> <X0(3)> <Medium>

Dans ce cas-ci, le second membre est le champ crée par un dipôle de la forme

$$I\delta(x - x_0)$$

- Les paramètres de la méthode GMRES. La méthode GMRES [9] est une méthode itérative qui nécessite comme données :
 - le nombre maximum d'itérations
 - un entier pour que l'écriture lors des itérations se fasse sur un fichier
 - la taille de projection ou restart (noté m) qui fixe le nombre de vecteurs à considérer dans la base de Krylov (l'algorithme est redémarré toutes les m itérations)
 - la valeur du résidu à atteindre

debut du commentaire dans le fichier de donnees :

GMRES inversion requires

1) an integer	N --> the maximum of iterations
2) an integer	1 --> write gmres diagnostic in some file
	0 --> not coded yet
3) an integer	M --> restart parameter
4) a real	eps-> the backward residual norm to achieve

fin du commentaire dans le fichier de donnees

Par exemple,

```
#GMRES
100
1
30
1.e-4
```

signifie que l'on va procéder à une résolution par GMRES en prenant un restart de 30. Le nombre maximal d'itérations est fixé à 100 et l'erreur que l'on désire atteindre est de 10^{-4} . Les résultats seront écrits sur un fichier.

7.2 Les valeurs par défaut de CESC.dat

On donne ici un descriptif des clés facultatives ainsi que les valeurs par défaut induites par l'absence de clé.

- ▶ Le nombre de points d'intégration pour le calcul de la matrice est choisi égal à 3 pour les triangles éloignés et 6 pour les triangles proches.
- ▶ La méthode d'inversion choisie est **GMRES**

- Le découpage en octree est fait de manière automatique. Le nombre de niveaux L_v est fixé tel que

$$\frac{kD}{2^{L_v}} < 1.5 < \frac{kD}{2^{L_v-1}} \quad (2)$$

où D est la taille du plus petit cube entourant l'obstacle. Une fois cet entier trouvé, on va élargir un peu la taille du cube entourant l'obstacle de telle sorte que

$$\frac{kD}{2^{L_v}} = 1.5 \quad (3)$$

où D est la nouvelle largeur.

- Le nombre de multipôles à chaque niveau et le nombre de points d'intégration sur la sphère unité sont choisis de telle sorte que

$$\begin{aligned} L &= kd + 2.15 \log(kd + \pi) \\ N_\theta &= (L + 1) \\ N_\phi &= 2(L + 1) \end{aligned} \quad (4)$$

- Lors des phases de changement de niveau, l'interpolation et le filtrage sont faits en utilisant une formulation semi-naïve de type Alpert et les transformées de Fourier sont faites via la librairie FFTW (cf paragraphe §3).
- L'algorithme multipolaire utilisé est de type **classique** avec :
 - quatre calculs pour la EFIE et CFIE
 - trois calculs pour la MFIE
- La matrice proche est calculée à chaque résolution. Elle est stockée sur disque en simple précision. La taille du buffer à la fois pour l'écriture de cette matrice et pour sa lecture est fixée égal à 1 000 000.
- Le calcul du second membre est fait en choisissant un nombre de points d'intégration égal à 3.
- Le programme va donner tous les temps de calcul liés au produit matrice-vecteur ainsi qu'à la résolution.
- Le nom du fichier contenant le déroulement de la simulation est par défaut, par exemple pour le cas séquentiel,

résolution multipôle	préconditionneur
PROCESS0000.out	PROCESS0000.prec

Pour le mode parallèle, chaque processeur crée le fichier qui lui est attaché. Ce qui donne pour l'exemple précédent pour un calcul en parallèle sur 4 processeurs,

résolution multipôle	préconditionneur
PROCESS0000.out	PROCESS0000.prec
PROCESS0001.out	PROCESS0001.prec
PROCESS0002.out	PROCESS0002.prec
PROCESS0003.out	PROCESS0003.prec

- Le nom du fichier contenant le courant est

currentPROCESS_0001 ou currentPROCESS_nocv0001 si la convergence n'a pas été atteinte

Le courant est écrit sous forme non formatée.

Remarque 5 *On peut résoudre le système pour différents seconds membres. Un fichier de courant pour chaque second membre est obtenu et est numéroté suivant l'ordre chronologique. Le fichier currentPROCESS_000k correspond au kième second membre.*

7.3 Modifier les valeurs par défaut

On peut cependant vouloir choisir d'autres paramètres que ceux définis par défaut. Ceci peut être fait en ajoutant des clés au fichier de données. Voici la liste de toutes les clés exceptée la clé définissant les calculs sur la matrice proche qui est décrite au paragraphe suivant §7.4.

7.3.1 Nom des fichiers de sortie

Le nom du fichier contenant le déroulement de la simulation peut être changé à l'aide de la clé #OUTPUTFILE.

debut du commentaire dans le fichier de donnees :

Name of the output files

fin du commentaire dans le fichier de donnees.

Si par exemple cette clé est égale à deuxpheres, les fichiers de sortie seront alors pour le cas séquentiel

résolution multipôle	préconditionneur
deuxpheres0000.out	deuxpheres0000.prec

Pour le mode parallèle, pour un calcul en parallèle sur 4 processeurs,

résolution multipôle	préconditionneur
deuxpheres0000.out	deuxpheres0000.prec
deuxpheres0001.out	deuxpheres0001.prec
deuxpheres0002.out	deuxpheres0002.prec
deuxpheres0003.out	deuxpheres0003.prec

7.3.2 Nom des fichiers de courant

Le stockage du courant peut être changé à l'aide de la clé #CURRENT.

debut du commentaire dans le fichier de donnees :

The current is assumed to be constructed

and the data for it is in a file formatted (val=2) or not (val = 1).

fin du commentaire dans le fichier de donnees.

Cette clé permet de définir le mode de stockage du courant. Si on veut un fichier sous forme formatée, il faut écrire

```
#CURRENT
```

```
1
```

```
toto
```

La deuxième valeur est une chaîne de caractères qui est utilisée éventuellement lors du calcul du champ lointain (cf paragraphe §10). D'autre part, si la clé #OUTPUTFILE est présente et est mise par exemple égale à deux_spheres, le courant sera stocké sous le fichier

currentdeux_spheres_0001 ou currentdeux_spheres_nocv0001 si la convergence n'a pas été atteinte

L'absence de clé est équivalente à

```
#CURRENT
```

```
1
```

7.3.3 Règle de quadrature pour le calcul des intégrales

On peut modifier le nombre de points d'intégration pour les interactions proches avec les deux clés #OPT_SIMPLE et #OPT_DOUBLE :

debut du commentaire dans le fichier de donnees :

```
The type of numerical integration technique is given here
if K, L are two triangles and c_K c_L the two centers of mass
of the triangles the rule is the following :
```

```
if ( d(c_g^K- c_g^L ) < tol*lambda --> '
    ns gauss points are used on K and L ': are used
if ( d(c_g^K- c_g^L ) > tol*lambda -->
    nf gauss points are used on K and L
    + a semi-analytical integration
```

You must give ns, nf for the

```
-- curl curl Green matrix (simple)
```

```
-- and for the curl Green matrix (double).
```

```
OPT_S(1): Number of integ. Pts. Gauss for 2 far triangles
```

```
OPT_S(2): Number of integ. Pts. Gauss for 2 near triangles
```

fin du commentaire dans le fichier de donnees

L'absence de clé est équivalente à

```
#OPT_SIMPLE
```

```
3 6
```

```
#OPT_DOUBLE
```

```
3 6
```

7.3.4 Règle de quadrature pour le calcul du second membre

On peut modifier le nombre de points d'intégration pour le calcul du second membre avec la clé #NPG_INTGS

debut du commentaire dans le fichier de donnees :

```
=====#NPG_INTGS of the RIGHT HAND SIDE=====
```

It is the number of Gauss points used on each triangle to make the numerical integration on the obstacle.

It is used to get the second term
fin du commentaire dans le fichier de donnees

L'absence de clé est équivalente à

```
#NPG_INTGS
3
```

7.3.5 Définition de l'octree

La méthode multipôle multiniveaux utilise une ventilation des points de Gauss du maillage dans un octree composé de boîtes de taille constante par niveau. On peut choisir aussi de modifier le découpage en boîtes par défaut grâce à la clé #BOXSPLIT

debut du commentaire dans le fichier de donnees :

```
=====#BOXSPLIT=====
```

```
Strategy for the splitting in boxes (automatic or given)
you must give
```

- ```
given : 1) the string automatic or given
 2) the number of levels for the splitting in boxes
 3) no
 4) the lower corner of the large box and its size (4 numbers)
automatic : 1) the number of levels for the splitting in boxes
```

fin du commentaire dans le fichier de donnees

La troisième carte est toujours fixée à no (cette carte correspond à un développement non encore abouti). Le nombre de niveaux doit toujours être supérieur à 3. Par exemple pour fixer en dur le nombre de niveaux à 5, on donnera

```
#BOXSPLIT
automatic ! automatic or given
5 ! levelnumber
no ! try to modify the center of the boxes
```

ou encore, si l'on veut 7 niveaux dans un cube de taille 3.2 et de coin inférieur  $(-1.2, -1.2, -1.2)$ , on écrira

```
#BOXSPLIT
given
7 ! levelnumber
no ! try to modify the center of the boxes
-1.2 -1.2 -1.2 3.2 !x-, y-, z-, size !
```

### 7.3.6 Discrétisation de la sphère unité

La méthode multipôle manipule les champs lointains et les champs proches associés à chaque boîte de l'octree. Ces champs sont des champs de vecteur définis sur la sphère unité (la sphère des directions d'ondes planes). On utilise alors une discrétisation de cette sphère qui diffère à chaque niveau. Pour cela, on utilise une discrétisation cartésienne

suisant les angles sphériques  $(\theta, \varphi)$  : on a  $N_\theta$  points en  $\theta$  et  $N_\varphi$  points en  $\varphi$ ,  $N_\phi$  et  $N_\theta$  dépendant de la taille de la boîte associée. Plus précisément, leurs valeurs dépendent du nombre de multipôles choisi

$$\begin{aligned} N_\theta &= (L + 1) \\ N_\phi &= 2(L + 1) \\ \text{avec } L &= \mathcal{F}(d) \end{aligned} \tag{5}$$

La loi  $\mathcal{F}$  fixant le nombre de multipôles en fonction de la taille de la boîte est choisie parmi trois possibilités

$$\begin{cases} L = kd + C \times \log(kd + \pi) + \mathcal{T} & (a) \\ L = kd + C(kd)^{1/3} + \mathcal{T} & (b) \\ \begin{cases} L = kd + C \times \max(\log(kd + \pi), (kd)^{1/3}) + \mathcal{T} & \text{si } kd \leq 7 \\ L = kd - 0.5 + \left(\frac{1}{2}\right)^{5/3} W^{2/3} \left(\frac{C(\alpha)kd}{4\epsilon^6}\right) (kd)^{1/3} & \text{sinon} \end{cases} & (c) \end{cases} \tag{6}$$

avec  $C(\alpha) = \frac{\alpha - 1}{\alpha + 1}$ ,  $\alpha = 2$ ,  $\epsilon = 10^{-2}$  et  $W$  est la fonction de Lambert définie par  $W(t)e^{W(t)} = t$ .

Le choix de la discrétisation sur la sphère unité à chaque niveau peut être modifié; il s'agit de la carte #SPHERES.

debut du commentaire dans le fichier de donnees :

```
=====#SPHERES=====
Strategy for the meshes of the unit sphere
5 possibilities (given or automatic)
1- first case: if given : you must type levelnumber lines with :
 lv, nb_theta, nb_phi, nb_mul
 (2*nb_mul+1<= nb_phi et nb_mul<=nb_theta)
 (remark: lv=1 corresponds to the larger mesh)
2- second case: if automatic : you must give the two constants C and T
 and the value sqrt(3)
and the type of law ('log' or 'lam' or 'pui')
 if it is 'log', the number of multipoles will be kd + C*(log(kd+pi) + T)
 if it is 'pui' , the number of multipoles will be kd+C*(kd)**(1/3)
 if it is 'lam', the same value when kd is smaller than 7 and after
 formule de quentin et collino
3- third case: if minus : the number of multipoles will be kd + (log(kd+pi)
4- fourth case: if mean : the number of multipoles will be kd + 2.15(log(kd+pi)

fin du commentaire dans le fichier de donnees.
```

Il existe plusieurs possibilités pour choisir la discrétisation sur la sphère.

- première possibilité : on choisit pour chaque niveau tous les paramètres de discrétisation de la sphère unité. On donnera par exemple, s'il y a 4 niveaux  
#SPHERES  
given

|   |     |     |     |
|---|-----|-----|-----|
| 1 | 160 | 320 | 159 |
| 2 | 84  | 168 | 83  |
| 3 | 46  | 92  | 45  |
| 4 | 26  | 52  | 25  |

- seconde possibilité : on veut utiliser pour tous les niveaux la même loi. Par exemple si on prend la loi (6-b) pour le nombre de multipôles, avec  $\mathcal{C} = 2.5$  et  $\mathcal{T} = 6$ , on écrira

```
#SPHERES
automatic !
2.15 6.
'pui'
```

- troisième possibilité : on prend la loi (6-a) pour le nombre de multipôles, avec  $\mathcal{C} = 1$  et  $\mathcal{T} = 0$ .
- quatrième possibilité : on prend la loi (6-a) pour le nombre de multipôles, avec  $\mathcal{C} = 2.15$  et  $\mathcal{T} = 0$ .
- cinquième possibilité : on prend la loi (6-a) pour le nombre de multipôles, avec  $\mathcal{C} = 3$  et  $\mathcal{T} = 0$ .

L'absence de clé est équivalente à

```
#SPHERES
automatic !
2.15 0.
'log'
```

### 7.3.7 Choix de la méthode d'interpolation et de filtrage

On peut remarquer que la discrétisation de la sphère unité dépend de la taille des boîtes et donc du niveau où on se trouve. C'est pourquoi lors des phases de changement de niveau, il est nécessaire de réaliser des étapes d'interpolation et de filtrage. La clé #INTERPOLATION permet de les définir.

```
debut du commentaire dans le fichier de donnees :
=====#INTERPOLATION=====
Strategy for the meshes of the unit sphere
2 possibilities (given or automatic)
if given : you must type levelnumber lines with :
 lv, type_interpolation, type_fft
 (remark: lv=1 corresponds to the larger mesh)
if automatic : you must give the two characters
type_interpolation , type_fft
fin du commentaire dans le fichier de donnees
```

Il n'est codé qu'un seul type d'interpolation et de filtrage. Ces phases d'interpolation et de filtrage utilisent une formulation semi-naïve de type Alpert et des transformées de Fourier. C'est pourquoi le premier caractère doit être mis égal à 'a' pour utiliser la formule d'Alpert.

En revanche, les transformées de Fourier peuvent être réalisées soit à l'aide de la librairie FFTW. Il faut alors mettre le deuxième caractère à 'f'. Nous avons aussi codé une transformée de Fourier. Pour faire appel à ces sous-routines, il faut alors mettre le deuxième caractère à 'b'.

**Remarque 6** Cette option de choix de transformée de Fourier permet de tester la bonne implémentation de la librairie FFTW.

L'absence de clé revient à

```
#INTERPOLATION
automatic
a f
```

et signifie que les transformées de Fourier sont faites via la librairie FFTW.

### 7.3.8 Choix du type d'algorithme multipolaire

Deux paramètres vont être maintenant définis. Tout d'abord, le nombre d'applications de l'algorithme multipolaire lors du calcul d'un produit matrice-vecteur va être choisi. En effet, classiquement, on réalise un produit matrice-vecteur en utilisant 4 fois l'algorithme multipolaire. On peut toutefois se ramener uniquement à 2 calculs multipolaires. On peut changer le nombre de calculs multipolaires à l'aide de la clé #FMMD.

On peut aussi remarquer que l'algorithme multipolaire lui-même peut être réalisé de deux manières. En effet, lors des phases d'interpolation ou de filtrage, on peut distinguer deux étapes de calcul :

- changement de niveau
- changement de centre de boîtes (multiplication par l'exponentielle)

Théoriquement, il faut d'abord effectuer la première étape et la seconde ensuite (notée **méthode 1**). Cependant, on remarque que si on fait le contraire (**méthode 2**), le temps de calcul des phases d'interpolation ou de filtrage est plus rapide et l'erreur engendrée est très faible.

debut du commentaire dans le fichier de donnees :

```
=====FMMD=====
Type of fmm computation: 2 types are coded
1)one integer 1: classic 'C' with 4 FMM calculations for EFIE or CFIE
and 3 for MFIE
2: bi _components '2' with 2 FMM calculations
2) a integer (1: method 1 or 2 : method 2) (2 is a good one)
Fin du commentaire dans le fichier de donnees.
```

L'absence de clé est équivalente à choisir

```
#FMMD
'C'
2
```

Ce qui signifie que l'on va faire un produit matrice-vecteur en appliquant 4 fois l'algorithme multipolaire et en utilisant la **méthode 2**.

### 7.3.9 Choix de la méthode d'inversion

Il existe plusieurs méthodes d'inversion du système. La première est une méthode d'inversion directe de type (LU). Les deux autres sont des méthodes dites de Krylov de type GMRES et flexible GMRES notée FGMRES. On peut choisir le type de méthode à l'aide de la clé #INVERSION.

debut du commentaire dans le fichier de donnees :

Après la carte inversion doit se trouver la chaine:

'NONE' or 'GMRES' or 'FGMRES' or 'FULL' (pour une solution LU)

fin du commentaire dans le fichier de donnees.

L'absence de clé revient à

```
#INVERSION
```

```
'GMRES'
```

Pour la méthode GMRES, il est nécessaire de donner les paramètres qui sont obligatoires et définis au paragraphe §7.1. Venons en maintenant à la méthode flexible GMRES. L'idée principale de la méthode flexible GMRES [10] est de préconditionner la matrice en utilisant quelques itérations d'une méthode de Krylov. On aboutit donc à un schéma avec deux boucles :

- une boucle externe qui utilise un GMRES
- une boucle interne qui sert à construire le préconditionneur et qui elle-aussi utilise un GMRES

**Boucle externe utilisant un GMRES :**

faire pour chaque itération

- faire un produit matrice-vecteur avec un algorithme multipolaire précis
- faire le produit par le préconditionneur :

**Boucle interne utilisant un GMRES :**

faire pour chaque itération

- ▶ faire un produit matrice-vecteur avec un algorithme multipolaire moins précis
- ▶ faire le produit par le préconditionneur de type SPAI

On est donc **obligé** de donner les paramètres suivants :

- le nombre maximum d'itérations
- le restart ou taille de projection dans la base de Krylov
- la valeur du résidu à atteindre

pour les deux GMRES emboîtés. ce qui est défini à l'aide de la clé #FLEXI\_GMRES.

debut de commentaire dans le fichier de donnees :

```
FLEXIBLE GMRES inversion requires
```

```
for the inner GMRES
```

- |               |                                           |
|---------------|-------------------------------------------|
| 1) an integer | N --> the maximum of iterations           |
| 2) an integer | 1 --> write gmres diagnostic in some file |
|               | 0 --> not coded yet                       |
| 3) an integer | M --> restart parameter                   |
| 4) a real     | eps-> the backward residual norm to       |
| achieve       |                                           |

```
5) a comment line
```

```
for the outer gmres
```

- |               |                                            |
|---------------|--------------------------------------------|
| 6) an integer | N --> the maximum of iterations            |
| 7) an integer | 1 --> write fgmres diagnostic in some file |
|               | 0 --> not coded yet                        |

8) an integer                    M --> restart parameter  
 9) a real                        eps-> the backward residual norm to

achieve

fin de commentaire dans le fichier de donnees

On a par exemple

```
#FLEXI_GMRES
30
1
15
1.e-1
---- data for the outer loop ---
300
1
30
1.e-4
```

Ce qui signifie que le préconditionneur sera construit en utilisant un GMRES avec un résidu de  $10^{-1}$  et un restart de 15. Remarquons que la boucle interne utilise une méthode itérative de type GMRES couplée à un calcul multipolaire. Cette dernière utilise une FMM peu précise et demande une autre définition de la discrétisation de la sphère. Ce qui est réalisé par la clé #FLEXIBLE\_SPHERES.

debut du commentaire dans le fichier de donnees

```
=====#FLEXIBLE SPHERES=====
Strategy for the meshes of the unit sphere for the inner GMRES
2 possibilities (given or automatic)
if given : you must type levelnumber lines with :
 lv, nb_theta, nb_phi, nb_mul
 (2*nb_mul+1<= nb_phi et nb_mul<=nb_theta)
 (remark: lv=1 corresponds to the larger mesh)
if automatic :you must give the two constants C and T
(the number of multipoles will be kd + C*(log(kd+pi) + T)
```

fin du commentaire dans le fichier de donnees

Dans ce cas-ci, on peut définir pour chaque niveau la discrétisation de la sphère unité ou utiliser pour tous les niveaux uniquement la loi (6-a) en choisissant les constantes  $C$  et  $T$ . Par exemple

```
#FLEXIBLE_SPHERES
automatic !
1. 0.
'log'
```

Ce qui signifie que pour la boucle interne, on utilise un calcul multipolaire avec un nombre de multipôles par niveau défini par  $L = kd + \log(kd + \pi)$ .

### 7.3.10 Restitution des temps de calcul

Le fichier de déroulement de simulation peut contenir des informations sur les temps de calcul. Le choix des informations sur l'écriture des temps CPU est fait à l'aide de la

clé #TIME.

```

debut de commentaire dans le fichier de donnees
=====#TIME=====
for having some informations on the CPU time
 1) 0 or 1 (1 : time for the matrix_vector, 0 :not)
 2) 0 or 1 (1: time for the inversion , 0:not)
 3) 0 or 1 (1:time for the I/o , 0: not)
#TIME
fin de commentaire dans le fichier de donnees

```

L'absence de clé est équivalente à

```

#TIME
1
1
0

```

## 7.4 Gestion de la matrice des interactions proches

### 7.4.1 Définition des paramètres pour le calcul de la matrice proche

Les boîtes du dernier niveau (qui ont la plus petite taille) forment une partition du domaine de calcul. On va isoler les interactions proches en fonction de la proximité de ces boîtes. Le calcul par FMM consiste à traiter les interactions proches par la méthode usuelle. Si le dernier niveau de boîtes est suffisamment élevé, ces interactions sont en faible nombre.

Bien que les interactions proches soient en petit nombre, l'assemblage de cette matrice proche engendre une quantité considérable de données. Plutôt que de jouer sur le nombre de niveaux pour diminuer le nombre d'interactions proches, on a préféré stocker cette matrice proche sur fichier. Une seconde piste pour libérer de la mémoire concerne la précision des données stockées. Comme la précision relative du produit matrice-vecteur de la FMM est typiquement de l'ordre de  $10^{-3}$ , nous envisageons de stocker la matrice proche avec des variables flottantes en simple précision (des réels codés sur 4 octets au lieu de la double précision usuelle). La clé #CLOSE permet de choisir le stockage de la matrice proche.

```

debut du commentaire dans le fichier de donnees :
=====#CLOSE MATRIX=====
Strategy for the close matrix
given 1) the integer (0 : to construct the close matrix,
 1: the close matrix is assumed to be constructed)
2) a integer (0 or 1 or 2)
 0: the data are not stored
 1:the data of the close matrix stored in
 a file unformatted
 2:the data of the close matrix stored in
 a file formatted
3) taille du buffer

```

4) simple ou double precision pour stocker la matrice proche  
fin du commentaire dans le fichier de donnees

Si on veut stocker la matrice proche sous forme non formatée en simple précision avec une taille de buffer en lecture et écriture de 1 000 000, on écrira

```
#CLOSE
0
1
1000000
simple
```

**Remarque 7** *Le produit matrice proche-vecteur est calculé avec les fichiers de stockage de la matrice proche. La matrice proche n'est jamais toute entière en mémoire.*

Si la matrice proche a déjà été construite et stockée sous forme non formatée en simple précision avec une taille de buffer de 1 000 000 et si on désire relancer un calcul, on donnera

```
#CLOSE
1
1
1000000
simple
```

ce qui évitera de reconstruire cette matrice. Si en revanche, le nombre d'éléments de cette matrice est inférieur à la taille du buffer, l'écriture sur fichier sera réalisée et pour la relecture de la matrice, il faudra indiquer la taille du buffer. L'absence de clé est équivalente à

```
#CLOSE
0
1
1000000
simple
```

**Remarque 8** *Cette clé permet d'éviter de recalculer la matrice proche et elle s'avère très utile en pratique.*

#### 7.4.2 Fichiers de sortie

Pour stocker une telle matrice creuse, nous avons choisi de ne stocker que les valeurs des éléments de la matrice. À chaque calcul, on reconstruit les données nécessaires pour faire un produit matrice proche-vecteur.

Les valeurs des éléments de la matrice sont stockées sur un fichier sous le répertoire

/CESC\_FMM/result/prec/

Le nom du fichier est obligatoirement fixé à

nom du fichier de géométrie concaténé avec le type d'équation intégrale  
et aussi le type d'algorithme suivi d'un suffixe

Le suffixe est défini par .close. Par exemple pour le cas séquentiel, on a

sphere270EFIEC.close

---

Ce qui correspond à une construction et au stockage de la matrice proche pour résoudre une équation intégrale de type EFIE avec un calcul multipolaire à 4 composantes sur un maillage “sphere 270”.

Pour le mode parallèle, chaque processeur crée le fichier qui lui est attaché. Ce qui donne pour l'exemple précédent pour un calcul en parallèle sur 10 processeurs,

```
sphere270EFIEC0000.close sphere270EFIEC0000.close....
..... sphere270EFIEC0009.close
```

## 8 Le préconditionneur

### 8.1 Nature du préconditionneur

Il s'agit d'une matrice creuse complexe  $M$  approchant l'inverse de la matrice  $Z$  associée au système linéaire que l'on cherche à résoudre, [5], [4], [7]. Concrètement, la taille mémoire de  $M$  est proportionnelle au nombre de degrés de liberté, et quasiment proportionnelle au nombre moyen d'éléments non nuls contenus dans chaque colonne. Rappelons que la construction du préconditionneur se fait à l'aide de l'exécutable `cesc_prec_exe` ou `pcesc_prec_exe` et avant de lancer le calcul de la solution.

La construction du préconditionneur se fait non pas en considérant tout l'obstacle mais en découpant celui-ci en tranche de largeur fixée suivant une direction prédéfinie. Chaque tranche est traitée indépendamment.

La construction du préconditionneur regroupe les coefficients colonne par colonne.

### 8.2 Fichier de sortie

Pour mémoriser une telle matrice creuse, nous avons choisi d'utiliser une structure SPARSE, reposant sur quatre caractéristiques :

- le vecteur `SPARSE%permut` qui donne les indices des colonnes de la matrice,
- le vecteur `SPARSE%MAT` des coefficients non nuls de la matrice,
- le tableau `SPARSE%COLUMN` qui donne les indices des lignes de chaque coefficient,
- le tableau `SPARSE%INDEX` qui donne les indices marquant le début de chaque colonne.

Ces quatre vecteurs sont stockés sur quatre fichiers différents sous le répertoire

```
/CESC_FMM/result/prec/
```

Les noms des fichiers sont obligatoirement fixés à

nom du fichier de géométrie concaténé avec le type d'équation intégrale  
et aussi le type d'algorithme suivi d'un suffixe

Les suffixes sont définis par `.permut .valprec .indexprec .columnprec`  
par exemple pour le cas séquentiel

```
sphere270EFIEC.permut
sphere270EFIEC.valprec
sphere270EFIEC.columnprec
sphere270EFIEC.indexprec
```

Ce qui correspond à une construction d'un préconditionneur de type SPAI pour résoudre une équation intégrale de type EFIE avec un calcul multipolaire à 4 composantes sur un maillage “sphere 270” .

Pour le mode parallèle, chaque processeur crée les quatre fichiers qui lui sont attachés. Ce qui donne pour l'exemple précédent pour un calcul en parallèle sur 10 processeurs,

|                                        |                                                                    |
|----------------------------------------|--------------------------------------------------------------------|
| sphere270EFIEC0000.permut<br>.....     | sphere270EFIEC0000.permut....<br>sphere270EFIEC0009.permut         |
| sphere270EFIEC0000.indexprec<br>.....  | sphere270EFIEC0000.indexprec....<br>sphere270EFIEC0009.indexprec   |
| sphere270EFIEC0000.columnprec<br>..... | sphere270EFIEC0000.columnprec....<br>sphere270EFIEC0009.columnprec |
| sphere270EFIEC0000.valprec<br>.....    | sphere270EFIEC0000.valprec....<br>sphere270EFIEC0009.valprec       |

### 8.3 Clés obligatoires pour le calcul du préconditionneur

Pour créer le préconditionneur, le fichier de données doit nécessairement contenir les clés obligatoires que nous avons décrites pour la résolution, (cf. paragraphe 7.1).

### 8.4 Les valeurs par défaut de CESC.dat

On donne ici un descriptif des clés facultatives ainsi que les valeurs par défaut induites par l'absence de clé nécessaire pour la construction du préconditionneur.

- ▶ le type de préconditionneur choisi est de type géométrique
- ▶ le choix de la plus petite boîte

$$d = \frac{\lambda}{10} \tag{7}$$

- ▶ la taille de bandes est fixée égale à  $3 \times d$
- ▶ la direction de découpage est la plus grande direction de l'obstacle.

Cela correspond aux informations contenues dans les clés #PREC, #PRECB et #PRECBOX.

### 8.5 Modifier les valeurs par défaut

#### 8.5.1 Construction et stockage du préconditionneur

Il existe deux façons de construire le préconditionneur. On peut soit construire un préconditionneur de type topologique ou de type géométrique.

La manière de construire le préconditionneur et de le stocker sur fichier est piloté par la clé #PREC

```
debut du commentaire dans le fichier de donnees
=====#PREC=====
The preconditionner is assumed to be constructed
and the data for it is in 4 files. Those 4 files
can be formatted (val=2) or not (val = 1).
If you don't want to precondition the system, val=0.
2) type of preconditioning
3) a size of buffer
#PREC
```

fin de commentaire dans le fichier de donnees

Par exemple, si on désire construire un préconditionneur de type géométrique avec un stockage sur fichier formaté et dont la taille du buffer est de 1000, on écrira

```
2
box
1000 ! buffer sizes
```

### 8.5.2 Description pour le préconditionneur de type topologique

Quand la géométrie est relativement régulière, seuls les degrés de liberté (ddl) proches les uns aux autres au sens topologique peuvent entrer en interaction forte. La construction du profil de  $M$  peut s'appuyer sur ces informations topologiques. La structure de  $M$  est définie à travers la conception des voisins de niveau  $k$  (cf. FIG.3). Au niveau 1, les voisins du ddl sont le ddl et les quatre ddl appartenant aux deux triangles connectés à l'arête sur laquelle ce ddl s'appuie. Au niveau 2, les voisins sont tous les voisins du niveau 1, plus les ddl appartenant aux triangles qui sont voisins aux deux triangles considérés au niveau 1, et ainsi de suite... Cette stratégie est connue sous le nom de stratégie topologique. En pratique, 3 niveaux de voisinage suffisent pour construire un préconditionneur  $M$  efficace. Ce préconditionneur est apparu peu efficace pour des géométries complexes et est en pratique peu utilisé.

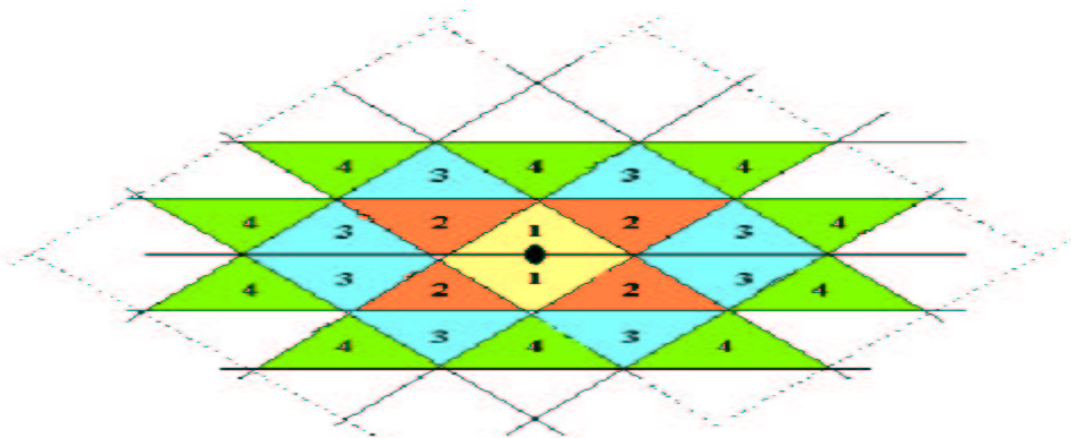


FIG. 3 – Schéma pour la stratégie topologique

### 8.5.3 Description pour le préconditionneur de type géométrique

Contrairement au cas précédent où la construction du préconditionneur s'appuyait sur des considérations topologiques (pour un degré de liberté  $j$  donné, on ne considérait que les degrés de liberté connectés à  $j$  via le maillage), on va maintenant utiliser plutôt une notion de distance (pour un degré de liberté  $j$  donné, on ne considère que les degrés de liberté qui lui sont proches).

Rappelons que la FMM sépare le calcul des interactions entre éléments proches de celui des interactions entre éléments éloignés.

Une implémentation naturelle du préconditionneur géométrique va s'inspirer de la structure de boîtes de la méthode FMM. Nous dirons qu'un ddl appartient à une boîte si le centre de l'arête sur laquelle s'appuie le ddl, est contenu dans la boîte. Dès lors, en ne considérant que le dernier niveau, nous envisagerons comme *proches* les éléments contenus dans deux boîtes ayant au moins un sommet en commun. Autrement dit un ddl est voisin au sens géométrique des ddl de cette même boîte et avec les ddl des 26 boîtes qui l'entourent.

Ces informations de voisinage sont utilisées, avec profit, pour calculer le profil du préconditionneur. Si le profil de  $M$  se révélait trop creux, il suffira donc de changer la taille des boîtes au dernier niveau pour augmenter la densité du préconditionneur.

La taille des boîtes choisie pour le préconditionneur est indépendante de celle choisie pour faire un calcul FMM. Elle est donc définie à partir de la clé #PRECBOX. La définition de cette clé est comparable à celle donnée pour la définition de l'octree (cf. paragraphe 7.3.5).

```
debut de commentaire dans le fichier de donnees
=====#PRECBOX=====
For constructing the preconditionner using the definition of the boxes
you must give: 1) the string automatic or given
 -if given: 1) the number of levels for the splitting in boxes
 2) the lower corner of the large box and its size (4 numbers)
 -if automatic: 1) the number of levels for the splitting in boxes
Caution: The number of levels must be greater or equal to 3
fin de commentaire dans le fichier de donnees

#PRECBOX
automatic
 4 ! levelnumber
-1.2 -1.2 -1.2 2.7 !x-, y-, z-, size ! if the option is given
```

### 8.5.4 Choix des bandes

Comme cela a déjà été dit, la construction du préconditionneur se fait non pas en considérant tout l'obstacle mais en découpant celui-ci en tranche de largeur fixée suivant une direction prédéfinie. La clé #PRECB permet de choisir la largeur de bandes ainsi que la direction

```
debut de commentaire
For constructing the preconditionner of type B
A splitting of the mesh in bands of width hmax is made
along the direction dir (optimization of the memory)
you must give : 1) hmax, then dir(1:3) in the first line
 2) the number of layers of triangles in the second line
 3) 1 or 0 for the 3rd line (1 :to store the preconditionner)

fin de commentaire

#PRECB
0.50 0 0 1 !
3 ! triangles number
```

---

0 ! store or not in full form

Dans ce cas, on prend une largeur de bande de 0.5 dans la direction de l'axe Oz.

## 9 Fichier de données de type CESC.dat pour le module antenne-satellite

Pour utiliser le module antenne-satellite, certaines clés sont à préciser dans le fichier CESC.dat. Les clés qui sont présentées ici se rajoutent à celles du module FMM. Elles peuvent être placées n'importe où dans le fichier CESC.dat. Les sources relatives à ce module sont placées dans le répertoire src/lib4sat.

### 9.1 Clé obligatoire de CESC.dat pour le module antenne-satellite

Pour utiliser les mesures du champs lointain comme source du système, la clé #ANTENNA\_FARFIELD doit être présente. Cette clé contient 3 champs donnés par les 3 lignes suivantes :

- le nom du fichier (chemin complet) contenant les coefficients d'interpolation  $\vec{\alpha}_{l,m}$
- les coordonnées cartésiennes du centre de phase (centre de l'antenne)
- l'information 'stop' ou 'go' suivant si l'utilisateur souhaite arrêter ou non le calcul après la formation du second membre.

**Remarque 9.1** *Certains réglages doivent être faits pour former le second membre à partir des mesures de l'antenne, par exemple un octree va être construit et le nombre de niveaux est un paramètre important. En effet, dès lors que la distance antenne-satellite est proche, le nombre de niveaux doit être relativement élevé de manière à avoir des tailles de boîtes de l'ordre du quart de longueur d'onde. Il est donc conseillé de mettre l'option 'stop' le temps de régler les différents paramètres avant d'effectuer la résolution du système itératif.*

debut du commentaire dans le fichier de donnees :

```
=====#ANTENNA_FARFIELD=====
#ANTENNA_FARFIELD
alpha/alpha_threshold.dat
0.0 0.0 0.0
'stop'
```

1st line: Name of the file where the coefficients alpha must be read

2nd line: coordinates of the antenna center = phase center of the farfield

3rd line: 'stop' or 'go' : if the computation stops

after the RHS or after the inversion

fin du commentaire dans le fichier de donnees

Si la clé #ANTENNA\_FARFIELD est absente, le programme utilisera la source spécifiée par la clé #INCIDENTFIELD (onde plane ou dipôle). Si la clé est activée, ce message apparaît :

```
=====
Right Hand side done from FAR2NEAR of antenna
=====
```

## 9.2 Clés optionnelles de CESC.dat pour le module antenne-satellite

Deux clés peuvent être rajoutées pour préciser certains paramètres dans le module antenne-satellite.

- la taille de l'antenne. C'est l'objet de la clé #ANTENNA\_DICHOTOMY. Elle comprend 2 champs donnés par les 2 lignes suivantes :

- l'information 'yes' or 'no'
- le numéro de la loi pour effectuer la dichotomie ou la taille de l'antenne.

Il s'agit de déterminer la taille de l'antenne soit à partir d'un calcul par dichotomie ('yes') soit directement en donnant sa valeur ('no').

'yes' Si l'option 'yes' a été choisie, la taille de l'antenne va être déterminée à partir du nombre d'harmoniques sphériques  $l_{\max}$  utilisées dans l'interpolation de son champ lointain. Il faut alors préciser la loi retenue pour effectuer le calcul de la taille de l'antenne  $d$  :

$$1) f_1(d) = kd + C_1 \log(kd + \pi) + C_2$$

$$2) f_2(d) = kd + C_1(kd + \pi)^{1/3} + C_2$$

où  $k$  est le nombre d'onde et les constantes  $C_1$  et  $C_2$  ont été choisies dans la clé #SPHERES (ou données par défaut si cette clé est absente). Le principe de la dichotomie consiste à chercher la valeur de  $d$  qui annule la fonction

$$f_i(d) - l_{\max} \text{ pour } i = 1 \text{ ou } 2.$$

'no' Si l'option 'no' a été sélectionnée, l'utilisateur doit alors donner sur la ligne suivante la taille de l'antenne (nombre entier ou réel).

**Remarque 9.2** La clé #ANTENNA\_DICHOTOMY étant optionnelle, le choix par défaut est de faire de la dichotomie avec la loi 1 (fonction  $f_1$ ).

**Remarque 9.3** La taille de l'antenne s'appuie sur la connaissance de  $l_{\max}$ , si  $l_{\max}$  est trop grand (même après l'étape de seuillage), le calcul de  $d$  par dichotomie risque d'être faux. L'idéal est de calculer cette taille à partir de plusieurs échantillons de mesures et de vérifier qu'elle est toujours du même ordre de grandeur.

debut du commentaire dans le fichier de donnees :

```
=====ANTENNA_DICHOTOMY=====
```

```
ANTENNA_DICHOTOMY (optional)
```

```
'yes'
```

```
1
```

If the card #ANTENNA\_DICHOTOMY is absent:

By default, the values are

```
'yes'
```

```
1
```

1st line for #ANTENNA\_DICHOTOMY: 'yes' or 'no'

The purpose is to determine the number of multipoles to construct the RHS with the FMM algorithm:

If 'yes' : size of antenna given by an algorithm based on dichotomy

```

2nd line : number of the law relative to the dichotomy algorithm
dichotomy law
 1) kd+C_1*ln(kd+pi)+C_2
 2) kd+C_1*(kd+pi)^(1/3)+C_2
 3)
the constants C_1 and C_2 are given by #SPHERES
If 'no' : the size of the antenna is directly read on the next line
2nd line: antenna size (real number)
fin du commentaire dans le fichier de donnees

```

- la construction des boîtes. C'est l'objet de la clé #ANTENNA\_NUMBERLEVEL qui fixe le nombre de niveaux lors de la construction de l'octree pour le calcul du second membre. Cette clé comprend 1 seul champ.

Ce nombre de niveaux est par défaut égal à celui précisé par la clé #BOXSPLIT ou calculé lorsque cette clé est absente. Pour assurer la convergence du système itératif, le nombre de niveaux  $L_v$  est généralement fixé tel que

$$\frac{kD}{2^{L_v}} < 1.5 < \frac{kD}{2^{L_v-1}}$$

où  $D$  est la taille du plus petit cube entourant l'obstacle. Le nombre  $L_v$  obtenu n'est pas forcément suffisant lorsque l'antenne et le satellite sont très proches (quart de longueur d'onde). Dans ce cas un message d'erreur apparaît pour préciser que le critère de distance nécessaire à la FMM n'est pas vérifié. Ce message est le suivant :

```

Problem with box : 37 at level 7
Distance criterium not satisfied
Box Center : -0.148441 -0.190997 -0.088441
Antenna Center : 0.000000 0.000000 0.000000
If the card #ANTENNA_NUMBERLEVEL exists in CESC.dat
==> we must increase the number of levels
If the card #ANTENNA_NUMBERLEVEL does not exist in CESC.dat
==> we must add it and specify the number of levels
====> the minimum number of levels (antenna octree)
 must be at least = 8

```

il faut alors ajouter la clé #ANTENNA\_NUMBERLEVEL si elle n'existait pas ou augmenter sa valeur afin de tester successivement  $L_v + 1$ ,  $L_v + 2$ , ...

```

debut du commentaire dans le fichier de donnees :
=====#ANTENNA_NUMBERLEVEL=====
#ANTENNA_NUMBERLEVEL (optional)
10

```

To precise the number of levels for the construction of the octree relative to the system: antenna+satellite.

By default, this number is equal to the number of levels given by #BOXSPLIT

```

fin du commentaire dans le fichier de donnees

```

**Remarque 9.4** Une fois que le nombre de niveaux est suffisant pour la détermination du second membre, le calcul est effectué. Le message suivant apparaît :

```
=====
RHS computed from antenna farfield is done
Potential written in file "RHS_antenna.dat"
=====

END of the COMPUTATION for the RHS
To solve the final system
Change the card #ANTENNA_FARFIELD with 'go' on the 3rd line
```

*l'option 'go' peut alors être activée dans la clé #ANTENNA\_FARFIELD.*

### 9.3 Comparaison dans le cas d'un dipôle

Si les mesures d'antenne correspondent à celles d'un dipôle, alors il est possible de comparer le second membre obtenu à partir des coefficients d'interpolation  $\vec{\alpha}_{l,m}$  à celui obtenu lorsque la clé #DIPOLE\_INCIDENTFIELD est activée. Pour effectuer cette comparaison, il faut :

- activer la clé #ANTENNA\_FARFIELD en précisant le nom du fichier des coefficients d'interpolation, la position de l'antenne
- activer la clé #DIPOLE\_INCIDENTFIELD en précisant l'orientation et la position du dipôle.

Le programme va calculer les deux second membres et fournit l'erreur relative entre ces deux résultats.

**Remarque 9.5** Dans le cas du dipôle, la taille de l'antenne calculée par dichotomie peut être "fausse", dans le sens où elle est trop grande. Si cette taille est trop grande, l'antenne est alors trop proche de l'obstacle et le critère de distance minimale (un quart de longueur d'onde) n'est jamais satisfait : la méthode ne peut être appliquée. Pour éviter un tel problème, mieux vaut rentrer une taille d'antenne directement dans le fichier de données en activant la clé #ANTENNA\_DICHOTOMY. Par exemple, ici on indique que la taille de l'antenne est 0.1.

```
=====#ANTENNA_DICHOTOMY=====
#ANTENNA_DICHOTOMY (optional)
'no'
0.1
```

**Remarque 9.6** Lors du calcul du champ lointain de la structure (antenne+obstacle), le champ lointain de l'antenne est ajouté à partir de la connaissance des coefficients d'interpolation (cf section 10.2).

## 10 Calcul des champs lointains

La SER bistatique est calculée après la résolution ou à l'aide de l'exécutable `cesc_rcs_exe`. Cela est réalisé si dans le fichier de données, la clé #BISTATIC est définie. Si cette clé n'est pas présente, le calcul du champ lointain n'est pas fait après la résolution.

## 10.1 La clé de la SER bistatique

La clé #BISTATIC est décrite par 3 lignes.

debut de commentaire dans le fichier de donnees

```
=====#BISTATIC=====
```

All angles in degree.

1) an integer

2)variation on theta

N angle1 angle2

N= number of subdivisions(angles)

theta varies between <angle1> and <angle2>

theta\_k = angle1 + (angle2-angle1)\*(k-1)/(N)

with k varying between 1 to N+1

3) variation on phi

N angle1 angle2

PHi varies between <angle1> and <angle2>

phi\_k = angle1 + (angle2-angle1)(k-1)/(N)

with k varying between 1 to N+1

The bistatic pattern is given according to the following rules:

Bistatic(s\_k) = 10 log<sub>10</sub>|Far(s\_k)|, k=1,...,N

( s\_k=(theta\_k, phi\_k) in spherical coordinates)

#BISTATIC

fin de commentaire dans le fichier de donnees

On donne pour chaque variation d'angle : le nombre de division  $N$  et l'angle de début  $\alpha_0$  et celui de fin  $\alpha_1$ . Le champ lointain est calculé pour  $\alpha = \alpha_0 + \frac{(\alpha_1 - \alpha_0)(k - 1)}{N}$  avec  $k$  variant de 1 à  $N + 1$ . Lorsque  $N$  vaut 0, on ne fait qu'un seul calcul pour  $\alpha_0$ .

Un exemple est donné ci-dessous :

#BISTATIC

1

180      0.0      180.0

79    0.0      355.5

Cela correspond à un calcul de SER (numéro 1), où  $\theta$  varie de 0 à 180 degrés et  $\phi$  de 0 à 355.5 degrés, soit un pas de 1 degré en  $\theta$  et de 45 degrés en  $\phi$ .

Dans le cas où on lance l'exécutable `cesc_rcs_exe`, on doit partir d'un fichier de géométrie et d'un fichier de courant ainsi que du même fichier de données. Précisons que le nom du fichier de courant (stocké lors de l'étape de résolution) est donné par la clé `CURRENT`. La première ligne précise si la lecture doit se faire de manière formatée ou non et la deuxième ligne le nom du fichier que l'on doit aller lire.

```
=====#CURRENT=====
```

The current is assumed to be constructed

1) integer :the data for it is in a file formatted (val=2) or not (val = 1)

2) name of the file

#CURRENT

1

currentPROCESS0001

## 10.2 Cas d'un dipôle ou d'une antenne

Lorsque la source est donnée par un dipôle (clé #DIPOLE\_INCIDENTFIELD) ou par une antenne (#ANTENNA\_FARFIELD), le champ lointain que l'on calcule correspond au champ total : somme du champ diffracté et du champ incident. On rajoute donc suivant le cas :

- le champ lointain du dipôle par une formule analytique qui tient compte de son orientation et de sa position (terme de phase), ou
- le champ lointain de l'antenne recomposé à partir des coefficients d'interpolation (fichier contenant  $\vec{\alpha}_{l,m}$ ) pour chaque direction de la sphère unité qui est demandée (angles demandés par la clé #BISTATIC).

## 10.3 Fichiers de sortie

Il y a trois fichiers créés. Par défaut, ils sont rangés dans le répertoire

/CESC\_FMM/result/

On peut changer cette localisation en définissant la variable d'environnement :

**\$CESC\_RESULT/**

- le fichier contenant les valeurs de la SER en décibel. Son nom est : rcs\_logPROCESS ou rcs\_log\_nom de l'outputfile si la clé #OUTPUTFILE est activée
- le fichier contenant le module du champ lointain rcsPROCESS ou rcs\_nom de l'outputfile si la clé #OUTPUTFILE est activée
- le fichier contenant les valeurs des champs. Son nom est rcs\_farfieldPROCESS ou rcs\_farfiel\_nom de l'outputfile si la clé #OUTPUTFILE est activée

Le paragraphe §11 suivant donne deux exemples de fichiers (contenu minimum et contenu maximum). Les deux derniers paragraphes décrivent un exemple de validation du code sur un cas académique et un exemple plus réaliste défini pour des applications CNES.

# 11 Exemples de fichiers de données

## 11.1 Valeurs minimales

```
=====
=====
COMMON DATA of the two F90 Programs : CESC code and FMM code
=====
=====

=====#INPUTFILE=====
Name of the input geometry file :
the file name.Geom must exist
#INPUTFILE
```

```

sphere5880
===== #WAVENUMBER =====
<Number-of-wave-number> following by the list of wave number.
<list-of-wave-number>
if((Number-of-wave-number)<0) then read the file : konde.dat
#WAVENUMBER
1
6.67

===== #EQUATION =====
The type of equation to be solved
you must give
the type of equation
'MASS' 'WEDGE' 'EFIE' 'CFIE' 'MFIE' 'GENE'
if 'CFIE' you must give alpha_cfie
if 'GENE' you must give the 5 numbers :
c_mass c_wedge c_simple c_double c_double_wedge
CFIE=(1-alpha)*MFIE + alpha*EFIE
alpha = 1 ==> EFIE
alpha = 0 ==> MFIE
0 < alpha < 1 ==> CFIE (alpha=0.2)
#EQUATION_TYPE
EFIE
0.2

===== #INCIDENTFIELD =====
Incident field : PLANE,DIPOLE,DELTA GAP,MAGFRILL
To choose the source term: put the character # in front of the name
!----- for PLANE----- !----- DIPOLE -----
!#PLANE ! !#DIPOLE !
!THETA PHI ! !IA(1) IA(2) IA(3) !
!Re(ETHETA) Im(ETHETA) ! !X0(1) X0(2) X0(3) !
!Re(EPhi) Im(EPhi) ! !Medium !
!-----!-----
!----- DELTATGAP -----!
!#DELTA GAP_INCIDENTFIELD !
!Index node (on the wire) !
!-----!-----
!----- EDGES -----!
!#EDGES_INCIDENTFIELD !
!Index Edge alimentation(on the body) !
!-----!-----
!----- COAX -----!
!#COAX_INCIDENTFIELD !
!-- a b h medium !
!-- a : rayon interne b : rayon externe !
!-- h : hauteur medium : milieu !

```

!-----!

PLANE\_INCIDENTFIELD Plane wave

1  
1 0.0 0.0 1.0 0.0 0.0 0.0

DELTAGAP\_INCIDENTFIELD delta-gap

1  
1 2

EDGES\_INCIDENTFIELD

1  
1 10 134 136 140 142 146 148 152 154 158 160

DIPOLE\_INCIDENTFIELD

1  
1 0.0 0.0 1.0 0. 0. 2. 0

COAX\_INCIDENTFIELD

0.00412 0.0012 0.07 0

MULTIPLANE\_INCIDENTFIELD

36  
1 0.0 0.0 1.0 0.0 0.0 0.0

=====#GMRES=====

GMRES inversion requires

- 1) an integer N --> the maximum of iterations
- 2) an integer 1 --> write gmres diagnostic in some file  
0 --> not coded yet
- 3) an integer M --> restart parameter
- 4) a real eps-> the backward residual norm to achieve

#GMRES

100  
1  
30  
0.10E-03

=====#BISTATIC=====

All angles in degree with angle\_k = angle1 + (angle2-angle1)(k)/(N)

- 1) first line : an integer
- 2) second line: variation in theta  
N= number of subdivisions(angles)  
theta varies between <angle1> and <angle2>
- 3) third line: variation in phi  
N= number of subdivisions(angles)  
phi varies between <angle1> and <angle2>

Output :

- 1:S.E.R (Section Equivalente Radar)
- 2:Etheta, EPHI

The bistatic pattern is given according to the following rules:

$$\text{Bistatic}(s_k) = 10 \log_{10} |\text{Far}(s_k)|, k=1, \dots, N$$

(  $s_k=(\text{theta}_k, \text{phi}_k)$  in spherical coordinates)

#BISTATIC

```
1
180 0.0 180.0
79 0.0 355.5
```

```
=====#ANTENNA CHARACTERISTICS=====
 3 CARDS to fill to compute the RHS from the antenna farfield
 ==#ANTENNA_FARFIELD
==#ANTENNA_DICHOTOMY (optional)
==#ANTENNA_NUMBERLEVEL (optional)
=====#ANTENNA_FARFIELD=====
#ANTENNA_FARFIELD
/home/bartoli/CESC_FMM/data_sat/alpha_threshold.dat_helice_4
-90.0 0.0 0.0
'go'
```

- 1st line: Name of the file where the coefficients alpha must be read
- 2nd line: coordinates of the antenna center = phase center of the farfield
- 3rd line: 'stop' or 'go' : if the computation stops  
after the RHS or after the inversion

## 11.2 Valeurs maximales

```
=====
=====
COMMON DATA of the two F90 Programs : CESC code and FMM code
=====
=====
=====#INPUTFILE=====
Name of the input geometry file :
the file name.Geom must exist
#INPUTFILE
sphere270
=====#OUTPUTFILE=====
```

```

Name of the output fileyes
#OUTPUTFILE
sphere270
=====#EQUATION=====
The type of equation to be solved
you must give
 the type of equation
 'MASS' 'WEDGE' 'EFIE' 'CFIE' 'MFIE' 'GENE'
 if 'CFIE' you must give alpha_cfie
 if 'GENE' you must give the 5 numbers :
 c_mass c_wedge c_simple c_double c_double_wedge
CFIE=(1-alpha)*MFIE + alpha*EFIE
alpha = 1 ==> EFIE
alpha = 0 ==> MFIE
0 < alpha < 1 ==> CFIE (alpha=0.2)
#EQUATION_TYPE
EFIE
0.2

=====#NUMERICAL INTEGRATION=====
The type of numerical integration technique is given here
if K, L are two triangles and c_K c_L the two centers of mass
of the triangles the rule is the following :
 if (d(c_g^K- c_g^L) < tol*lambda --> '
 ns gauss points are used on K and L ': are used
 if (d(c_g^K- c_g^L) > tol*lambda -->
 nf gauss points are used on K and L
 + a semi-analytical integration
You must give ns, nf for the
-- curl curl Green matrix (simple)
-- and for the curl Green matrix (double).

OPT_S(1): Number of integ. Pts. Gauss for 2 far triangles
OPT_S(2): Number of integ. Pts. Gauss for 2 near triangles
#OPT_SIMPLE
3 6
#OPT_DOUBLE
3 6

=====#WAVENUMBER=====
<Number-of-wave-number> following by the list of wave number.
<list-of-wave-number>
if((Number-of-wave-number)<0) then read the file : konde.dat
#WAVENUMBER
1
1.5d0
=====#INCIDENTFIELD=====

```

```

Incident field : PLANE,DIPOLE,DELTAGAP,MAGFRILL
To choose the source term: put the character # in front of the name
!----- for PLANE----- ----- DIPOLE -----
!#PLANE ! !#DIPOLE !
!THETA PHI ! !IA(1) IA(2) IA(3) !
!Re(ETHETA) Im(ETHETA) ! !X0(1) X0(2) X0(3) !
!Re(EPhi) Im(EPhi) ! !Medium !
!-----!-----
!----- DELTATGAP -----!
!#DELTAGAP_INCIDENTFIELD !
!Index node (on the wire) !
!-----!
!----- EDGES -----!
!#EDGES_INCIDENTFIELD !
!Index Egde alimentation(on the body) !
!-----!
!----- COAX -----!
!#COAX_INCIDENTFIELD !
!-- a b h medium !
!-- a : rayon interne b : rayon externe !
!-- h : hauteur medium : milieu !
!-----!

#PLANE_INCIDENTFIELD Plane wave
1
1 114.0 90.0 1.0 0.0 0.0 0.0

DELTAGAP_INCIDENTFIELD delta-gap
1
1 2

EDGES_INCIDENTFIELD
1
1 10 134 136 140 142 146 148 152 154 158 160

DIPOLE_INCIDENTFIELD
1
1 0.0 0.0 1.0 0. 0. 0.6 0

COAX_INCIDENTFIELD
0.00412 0.0012 0.07 0

MULTIPLANE_INCIDENTFIELD
9
1 0.0 90.0 1.0 0.0 0.0 0.0
40.0 0.0
=====#BISTATIC=====

```

All angles in degree with  $\text{angle}_k = \text{angle1} + (\text{angle2}-\text{angle1})(k)/(N)$

- 1) first line : an integer
- 2) second line: variation in theta  
 $N =$  number of subdivisions(angles)  
 theta varies between  $\langle \text{angle1} \rangle$  and  $\langle \text{angle2} \rangle$
- 3) third line: variation in phi  
 $N =$  number of subdivisions(angles)  
 phi varies between  $\langle \text{angle1} \rangle$  and  $\langle \text{angle2} \rangle$

Output :

- 1:S.E.R (Section Equivalente Radar)
- 2:Etheta, EPHI

The bistatic pattern is given according to the following rules:

$$\text{Bistatic}(s_k) = 10 \log_{10} |\text{Far}(s_k)|, \quad k=1, \dots, N$$

(  $s_k=(\text{theta}_k, \text{phi}_k)$  in spherical coordinates)

#BISTATIC

```
1
180 0.0 180.0
79 0.0 355.5
```

```
=====
=====
DATA of the CESC code
=====
=====
```

```
=====#IMPEDANCE=====
IMPEDANCE
(3.2,-0.2)
(0.0,-0.1)
```

```
=====#GRIDPROCS=====
SCALAPACK Grid procs for BLACS
#GRIDPROCS
3 3
```

```
=====#SIZEBLOCK=====
SCALAPACK block size pour cyclic distribution
#SIZEBLOCK
32
```

```
=====#CONTRASTS=====
```

<Number-of-mediums>

<list-of-contrasts>

#CONTRASTS

1

(1.0,0.0) (1.0,0.0)

(2.0,1.0) (1.0,0.0)

=====#MONOSTATIC=====

Direction U1,U2,U3

type excitation : PLANE,DIPOL

No-of-excitation

#MONOSTATIC

0.0 0.0 1.0

PLANE

1

=====#NEARFIELDS=====

type excitation : PLANE,DIPOLE

No-of-wave-number No-of-excitation

#NEARFIELDS

DIPOL

1 1

=====#RESTITCURRENT=====

type excitation : PLANE,DIPOL

No-of-wave-number No-of-excitation No-of-medium

#RESTITCURRENT

DIPOL

1 1 0

=====

=====

DATA of the FMM code

=====

=====

=====#NPG\_INTGS of the RIGHT HAND SIDE=====

It is the number of Gauss points used on each triangle

to make the numerical integration on the obstacle.

It is used to get the second term

#NPG\_INTGS

3

=====#CLOSE MATRIX=====

Strategy for the close matrix

given 1) the integer (0 : to construct the close matrix,

1: the close matrix is assumed to be constructed)  
 2) a integer (0 or 1 or 2)  
     0: the data are not stored  
     1: the data of the close matrix stored in  
         a file unformatted  
     2: the data of the close matrix stored in  
         a file formatted  
 3) taille du buffer  
 4) simple ou double precision pour stocker la matrice proche

```
#CLOSE
0
2
200000
simple
```

=====#BOXSPLIT=====

Strategy for the splitting in boxes (automatic or given)  
 you must give

    given : 1) the number of levels for the splitting in boxes  
           2) no  
           3) the lower corner of the large box and its size (4 numbers)  
     automatic : 1) the number of levels for the splitting in boxes  
                 2) no

Caution: The number of levels must be greater or equal to 3

```
#BOXSPLIT
automatic !given !automatic ! automatic or given
4 ! levelnumber
no ! try to modify the center of the boxes
-1.2 -1.2 -1.2 2.4 !x-, y-, z-, size ! if the option is given
```

=====#FMMD=====

Type of fmm computation 2 types are coded  
 1) one integer 1: classic 'C' with 4 FMM calculations for EFIE or CFIE  
               and 3 for MFIE  
               2: bi \_components '2' with 2 FMM calculations  
               3: direct : in case of sinus kernel, it is possible to use  
                   an algo with no translation (not tested yet)  
 2) a integer (1 or 2) (2 is a good one)

```
#FMMD
'C'
2
```

=====#KERNEL\_TYPE=====

The type of equation to be solved  
 you must give  
     the Green function :

```

cos(kr)/(4 pi kr) --> 'COS'
sin(kr)/(4 pi kr) --> 'SIN'
exp(ikr)/(4 pi kr) --> 'EXP'
#KERNEL_TYPE
EXP

=====#SPHERES=====
Strategy for the meshes of the unit sphere
5 possibilities (given or automatic)
1- first case: if given : you must type levelnumber lines with :
 lv, nb_theta, nb_phi, nb_mul
 (2*nb_mul+1<= nb_phi et nb_mul<=nb_theta)
 (remark: lv=1 corresponds to the larger mesh)
2- second case: if automatic : you must give the two constants C and T
 and the value sqrt(3)
and the type of law ('log' or 'lam' or 'pui')
 if it is 'log', the number of multipoles will be kd + C*(log(kd+pi) + T)
 if it is 'pui' , the number of multipoles will be kd+C*(kd)**(1/3)
 if it is 'lam', the same value when kd is smaller than 7 and after
 formule de quentin et collino
3- third case: if minus : the number of multipoles will be kd + (log(kd+pi)
4- fourth case: if mean : the number of multipoles will be kd + 2.15(log(kd+pi)
5- fifth case: if plus : the number of multipoles will be kd +3. (log(kd+pi)
#SPHERES
automatic !
2.15 0. 1.73
'log'
lv nb_theta nb_phi nb_mul

=====#FLEXIBLE SPHERES=====
Strategy for the meshes of the unit sphere for the inner GMRES
2 possibilities (given or automatic)
1- first case: if given : you must type levelnumber lines with :
 lv, nb_theta, nb_phi, nb_mul
 (2*nb_mul+1<= nb_phi et nb_mul<=nb_theta)
 (remark: lv=1 corresponds to the larger mesh)
2-second case: if automatic: you must give the two constants C and T
(the number of multipoles will be kd + C*(log(kd+pi) + T)
#FLEXIBLE_SPHERES
automatic !
1. 0. 1.732
'log'
lv nb_theta nb_phi nb_mul

=====#INTERPOLATION=====
Strategy for the meshes of the unit sphere
2 possibilities (given or automatic)

```

```

if given : you must type levelnumber lines with :
 lv, type_interpolation, type_fft
 (remark: lv=1 corresponds to the larger mesh)
if automatic : you must give the two characters
 type_interpolation , type_fft
#INTERPOLATION
automatic !
a f
lv type_interpol type_fft

=====#FULLMATRIX=====
If you want to compute the full matrix (computed by the FMM methods)
put 'yes' after the card #FULLMATRIX
#FULLMATRIX
'no'

=====#INVERSION=====
'NONE' or 'GMRES' or 'FGMRES' or 'FULL'(pour faire un LU sur la matrice pleine)
#INVERSION
'GMRES'

=====#PRECB=====
For constructing the preconditionner of type B
A splitting of the mesh in bands of width hmax is made
along the direction dir (optimization of the memory)
you must give : 1) hmax, then dir(1:3) in the first line
 2) the number of layers of triangles in the second line
 3) 1 or 0 for the 3rd line (1 :to store the preconditionner)

#PRECB
0.50 0 0 1 !0.1 1 0 0
3 ! triangles number
0 ! store or not in full form

=====#PREC_WEIGHT=====
For constructing the preconditionner of type B
A splitting of the mesh in bands of width hmax is made
along the direction dir (optimization of the memory)
you must give : 1) 1 (weight) 0 no thing

#PRECWEIGHT
0

=====#PRECBOX=====
For constructing the preconditionner using the definition of the boxes
you must give: 1) the string automatic or given
 -if given: 1) the number of levels for the splitting in boxes

```

2) the lower corner of the large box and its size (4 numbers)

-if automatic: 1) the number of levels for the splitting in boxes

Caution: The number of levels must be greater or equal to 3

#PRECBOX

automatic ! automatic or given

4 ! levelnumber

-1.2 -1.2 -1.2 2.7 !x-, y-, z-, size ! if the option is given

=====#PREC=====

The preconditionner is assumed to be constructed  
and the data for it is in 2 files. Those 2 files  
can be formatted (val=2) or not (val = 1).

If you don't want to precondition the system, val=0.

2) type of preconditioning

3) a size of buffer

#PREC

2

box

1000 ! buffer sizes

=====#GMRES=====

GMRES inversion requires

- |               |                                                |
|---------------|------------------------------------------------|
| 1) an integer | N --> the maximum of iterations                |
| 2) an integer | 1 --> write gmres diagnostic in some file      |
|               | 0 --> not coded yet                            |
| 3) an integer | M --> restart parameter                        |
| 4) a real     | eps-> the backward residual norm to<br>achieve |

#GMRES

100

1

30

1.e-4

=====#FLEXIBLE-GMRES=====

FLEXIBLE GMRES inversion requires

for the inner GMRES

- |               |                                                |
|---------------|------------------------------------------------|
| 1) an integer | N --> the maximum of iterations                |
| 2) an integer | 1 --> write gmres diagnostic in some file      |
|               | 0 --> not coded yet                            |
| 3) an integer | M --> restart parameter                        |
| 4) a real     | eps-> the backward residual norm to<br>achieve |

for the outer flexible gmres

- |               |                                 |
|---------------|---------------------------------|
| 1) an integer | N --> the maximum of iterations |
|---------------|---------------------------------|



```

=====#MIE=====
The bistatic pattern is compu You must give:
 1) 'yes' or 'no' if required
 2) an integer that is the number of poles
 if negative the program will compute it
 according to a truncation error of runcation error of 10^{-6}
 in the Jacobi-Anger serie
#MIE
'no'
-30
=====#TIME=====
for having some informations on the CPU time
 1) 0 or time for the matrix_vector, 0 :not)
 2) 0 or 1 (1: time for the inversion , 0:not)
 3) 0 or 1 (1:time for the I/o , 0: not)
#TIME
1
1
0

=====#ANTENNA CHARACTERISTICS=====
 3 CARDS to fill to compute the RHS from the antenna farfield
 ==#ANTENNA_FARFIELD
 ==#ANTENNA_DICHOTOMY (optional)
 ==#ANTENNA_NUMBERLEVEL (optional)
=====#ANTENNA_FARFIELD=====
#ANTENNA_FARFIELD
/home/bartoli/CESC_FMM/data_sat/alpha_threshold.dat_helice_4
-90.0 0.0 0.0
'go'

1st line: Name of the file where the coefficients alpha must be read
2nd line: coordinates of the antenna center = phase center of the farfield
3rd line: 'stop' or 'go' : if the computation stops
 after the RHS or after the inversion

=====#ANTENNA_DICHOTOMY=====

#ANTENNA_DICHOTOMY (optional)
'yes'
1

If the card #ANTENNA_DICHOTOMY is absent:
By default, the values are
 'yes'
 1

```

1st line for #ANTENNA\_DICHOTOMY: 'yes' or 'no'

The purpose is to determine the number of multipoles to construct the RHS with the FMM algorithm:

If 'yes' : size of antenna given by an algorithm based on dichotomy

2nd line : number of the law relative to the dichotomy algorithm

dichotomy law

1)  $kd+C_1*\ln(kd+pi)+C_2$

2)  $kd+C_1*(kd+pi)^{(1/3)}+C_2$

3) .....

the constants C\_1 and C\_2 are given by #SPHERES

If 'no' : the size of the antenna is directly read on the next line

2nd line: antenna size (real number)

=====#ANTENNA\_NUMBERLEVEL=====

ANTENNA\_NUMBERLEVEL (optional)

2

To precise the number of levels for the construction of the octree relative to the system: antenna+satellite.

By default, this number is equal to the number of levels given by #BOXSPLIT

## 12 Un exemple de validation dans le cas de la sphère

Nous avons repris l'exemple de la page 8 du rapport [1].

### 12.1 Les fichiers d'entrée

- Le fichier de géométrie (satellite) est `sphereR1_Tx_100.Geom` : c'est une sphère unité dont le centre est placé au point (100,0,0).
- Le nombre d'onde est  $k = 1.5$ .
- Le champ incident est celui d'une antenne hélice à 4 brins située à  $23\lambda$  de la sphère, c'est à dire que le centre de l'antenne est placé à l'origine.
- Le fichier du code CESC\_FMM : `CESC_sphereR1.dat`.
- Le fichier contenant les coefficients d'interpolation ( $l_{max}=7$  et  $m_{max}=4$ ) : `alpha_threshold.dat_helice_4`.

### 12.2 L'exécution des programmes

On lance le préconditionneur sur 4 processeurs.

```
> mpirun -np 4 bin/pcesc_prec_exe -dCESC_sphereR1.dat
```

Les commentaires à l'écran sont :

```
Je suis le processeur 0 sur 4 processus
```

```
inputfile ----> /home/bartoli/CESC_FMM_SAT/cesc//CESC_sphereR1.dat
```

```

listing ----> /home/bartoli/CESC_FMM_SAT/result//sphereR1_Tx_100.prec
preconditioning ----> /home/bartoli/CESC_FMM_SAT/result//prec
Je suis le processeur 3 sur 4 processus
Je suis le processeur 2 sur 4 processus
Je suis le processeur 1 sur 4 processus
=====
 THE END for the process 2
=====
=====
 THE END for the process 3
=====
=====
 THE END for the process 1
=====
=====
 THE END for the process 0
=====

```

>

On lance ensuite la formation du second membre et la résolution GMRES.

```
> mpirun -np 4 bin/pcesc_ites_sat_exe -dCESC_sphereR1.dat
```

```

Je suis le processeur 0 sur 4 processus
inputfile ----> /home/bartoli/CESC_FMM_SAT/cesc//CESC_sphereR1.dat
Je suis le processeur 2 sur 4 processus
Je suis le processeur 1 sur 4 processus
Je suis le processeur 3 sur 4 processus

```

```

The preconditionner is computed to be used OUT OF CORE.
It is and divided into 76 formatted blocks

```

```
Number of nonzero elements of preconditioning: 75236
```

WARNING GMRES :

```

For M = 100 optimal value
for LWORK = 60901

```

CONVERGENCE HISTORY FOR GMRES

|   |          |    |
|---|----------|----|
| 1 | 0.29E+00 | -- |
| 2 | 0.85E-01 | -- |
| 3 | 0.20E-01 | -- |
| 4 | 0.42E-02 | -- |
| 5 | 0.59E-03 | -- |
| 6 | 0.18E-03 | -- |
| 7 | 0.36E-04 |    |

current in

```
.../home/bartoli/CESC_FMM_SAT/result//currentsphereR1_Tx_1000001
```

>

### 12.3 Les fichiers de sortie

- Les fichiers de sortie classiques se trouvent dans le répertoire `result`.
- Le fichier `RHS_antenna.dat` est créé dans le répertoire courant (il contient le second membre du système).

### 12.4 Le fichier `CESC.dat`

Nous avons rappelé ici le contenu du fichier d'entrée `CESC_sphereR1.dat` en ne retenant que les clés importantes. Notons ici que le nombre de niveaux pour former le second membre et pour faire la résolution GMRES sont identiques : clé `#BOXSPLIT` égale à 3 niveaux et la clé `#ANTENNA_NUMBERLEVEL` est prise par défaut égale à cette valeur. Le fichier `CESC_sphereR1.dat` est donné ci-dessous.

```

=====
=====
COMMON DATA of the two F90 Programs : CESC code and FMM code
=====
=====

=====#INPUTFILE=====
Name of the input geometry file :
the file name.Geom must exist
#INPUTFILE
sphereR1_Tx_100

#=====#OUTPUTFILE=====
Name of the output files
#OUTPUTFILE
sphereR1_Tx_100

=====#WAVENUMBER=====
<Number-of-wave-number> following by the list of wave number.
<list-of-wave-number>
if((Number-of-wave-number)<0) then read the file : konde.dat
#WAVENUMBER
1
1.5

=====#EQUATION=====
The type of equation to be solved
you must give
the type of equation
'MASS' 'WEDGE' 'EFIE' 'CFIE' 'MFIE' 'GENE'
if 'CFIE' you must give alpha_cfie
if 'GENE' you must give the 5 numbers :
c_mass c_wedge c_simple c_double c_double_wedge

```

```
CFIE=(1-alpha)*MFIE + alpha*EFIE
alpha = 1 ==> EFIE
alpha = 0 ==> MFIE
0 < alpha < 1 ==> CFIE (alpha=0.2)
#EQUATION_TYPE
EFIE
0.2
```

```
=====
=====
DATA of the FMM code
=====
=====
```

```
=====#BOXSPLIT=====
Strategy for the splitting in boxes (automatic or given)
you must give

given : 1) the number of levels for the splitting in boxes
 2) no
 3) the lower corner of the large box and its size (4 numbers)
automatic : 1) the number of levels for the splitting in boxes
 2) no
Caution: The number of levels must be greater or equal to 3
```

```
#BOXSPLIT
automatic ! automatic or given
3 ! levelnumber
no ! try to modify the center of the boxes
-0.6 -11 -0.6 30 !x-, y-, z-, size ! if the option is given
```

```
=====#GMRES=====
GMRES inversion requires
1) an integer N --> the maximum of iterations
2) an integer 1 --> write gmres diagnostic in some file
 0 --> not coded yet
3) an integer M --> restart parameter
4) a real eps-> the backward residual norm to
 achieve
```

```
#GMRES
300
1
100
1.e-4
```



### 13.3 Les fichiers de sortie

- Les fichiers de sortie classiques se trouvent dans le répertoire `result`.
- Le fichier `RHS_antenna.dat` est créé dans le répertoire courant.

### 13.4 Le fichier `CESC.dat`

Nous avons rappelé ici le contenu du fichier d'entrée `CESC_demeter.dat` en ne retenant que les clés importantes. Les particularités sont reportées ci-dessous :

- le nombre de niveaux pour le préconditionneur est imposé à 8 au lieu de 9 par défaut, on accélère ainsi la convergence du GMRES avec un préconditionneur plus riche

```
#PRECBOX
 automatic
 8 ! levelnumber
```

- le nombre de niveaux pour former le second membre est imposé

```
#ANTENNA_NUMBERLEVEL (optional)
 12
```

- le nombre de niveaux pour l'inversion GMRES est imposé

```
#BOXSPLIT
 automatic ! automatic or given
 8 ! levelnumber
```

Le fichier `CESC_demeter.dat` est donné ci-dessous.

```
=====
=====
COMMON DATA of the two F90 Programs : CESC code and FMM code
=====
=====

=====#INPUTFILE=====
Name of the input geometry file :
the file name.Geom must exist
#INPUTFILE
demeter_t1s

#=====#OUTPUTFILE=====
Name of the output files
#OUTPUTFILE
helice_demeter_t1s

=====#WAVENUMBER=====
<Number-of-wave-number> following by the list of wave number.
<list-of-wave-number>
if((Number-of-wave-number)<0) then read the file : konde.dat
#WAVENUMBER
1
27

=====#EQUATION=====
The type of equation to be solved
you must give
the type of equation
'MASS' 'WEDGE' 'EFIE' 'CFIE' 'MFIE' 'GENE'
if 'CFIE' you must give alpha_cfie
if 'GENE' you must give the 5 numbers :
c_mass c_wedge c_simple c_double c_double_wedge
CFIE=(1-alpha)*MFIE + alpha*EFIE
alpha = 1 =====> EFIE
alpha = 0 =====> MFIE
0 < alpha < 1 =====> CFIE (alpha=0.2)
#EQUATION_TYPE
EFIE
0.2

=====
=====
DATA of the FMM code
=====
=====
```

=====#BOXSPLIT=====

Strategy for the splitting in boxes (automatic or given)  
you must give

- given : 1) the number of levels for the splitting in boxes
- 2) no
- 3) the lower corner of the large box and its size (4 numbers)
- automatic : 1) the number of levels for the splitting in boxes
- 2) no

Caution: The number of levels must be greater or equal to 3

#BOXSPLIT

automatic ! automatic or given

8 ! levelnumber

no ! try to modify the center of the boxes

-0.6 -11 -0.6 30 !x-, y-, z-, size ! if the option is given

=====#PRECBOX=====

For constructing the preconditionner using the definition of the boxes  
you must give: 1) the string automatic or given

- if given: 1) the number of levels for the splitting in boxes
- 2) the lower corner of the large box and its size (4 numbers)
- if automatic: 1) the number of levels for the splitting in boxes

Caution: The number of levels must be greater or equal to 3

#PRECBOX

automatic!given ! ! automatic or given

8 ! levelnumber

-1.2 -1.2 -1.2 2.7 !x-, y-, z-, size ! if the option is given

=====#GMRES=====

GMRES inversion requires

- 1) an integer                   N --> the maximum of iterations
- 2) an integer                   1 --> write gmres diagnostic in some file
- 0 --> not coded yet
- 3) an integer                   M --> restart parameter
- 4) a real                        eps-> the backward residual norm to  
                                  achieve

#GMRES

300

1

100

4.2e-4

=====#ANTENNA CHARACTERISTICS=====

3 CARDS to fill to compute the RHS from the antenna farfield

```
==#ANTENNA_FARFIELD
==#ANTENNA_DICHOTOMY (optional)
==#ANTENNA_NUMBERLEVEL (optional)
=====#ANTENNA_FARFIELD=====
#ANTENNA_FARFIELD
/home/emc_home/bartoli/CESC_FMM_SAT/data_sat/alpha_threshold.dat_k27
0.0 0.0 0.0
'go'
```

1st line: Name of the file where the coefficients alpha must be read  
2nd line: coordinates of the antenna center = phase center of the farfield  
3rd line: 'stop' or 'go' : if the computation stops  
after the RHS or after the inversion

```
=====#ANTENNA_DICHOTOMY=====
ANTENNA_DICHOTOMY (optional)
'yes'
1
```

If the card #ANTENNA\_DICHOTOMY is absent:  
By default, the values are

```
'yes'
1
```

1st line for #ANTENNA\_DICHOTOMY: 'yes' or 'no'  
The purpose is to determine the number of multipoles to construct the RHS  
with the FMM algorithm:

If 'yes' : size of antenna given by an algorithm based on dichotomy  
2nd line : number of the law relative to the dichotomy algorithm  
dichotomy law

- 1)  $kd + C_1 \ln(kd + \pi) + C_2$
- 2)  $kd + C_1 * (kd + \pi)^{(1/3)} + C_2$
- 3) .....

the constants C\_1 and C\_2 are given by #SPHERES

If 'no' : the size of the antenna is directly read on the next line  
2nd line: antenna size (real number)

```
=====#ANTENNA_NUMBERLEVEL=====
#ANTENNA_NUMBERLEVEL (optional)
12
```

To precise the number of levels for the construction of the octree relative  
to the system: antenna+satellite.

By default, this number is equal to the number of levels  
given by #BOXSPLIT

## 13.5 La convergence

Si le préconditionneur est calculé sur 8 niveaux, l'algorithme GMRES (restart=100) converge en 301 itérations avec un résidu final de  $0.42 \times 10^{-3}$ . La solution a été obtenue après 56 minutes (4 processeurs sur le COMPAQ AlphaServer). Si on utilise un flexible GMRES (FGMRES), il faut rajouter les clés suivantes dans le fichier CESC\_demeter.dat (clés #INVERSION #FLEXIBLE\_SPHERES et #FLEXI\_GMRES).

```

=====#INVERSION=====
'NONE' or 'GMRES' or 'FGMRES' or 'FULL'(pour faire un LU sur la matrice pleine)
#INVERSION
'FGMRES'

=====#FLEXIBLE SPHERES=====
Strategy for the meshes of the unit sphere for the inner GMRES
2 possibilities (given or automatic)
1- first case: if given : you must type levelnumber lines with :
 lv, nb_theta, nb_phi, nb_mul
 (2*nb_mul+1<= nb_phi et nb_mul<=nb_theta)
 (remark: lv=1 corresponds to the larger mesh)
2-second case: if automatic: you must give the two constants C and T
(the number of multipoles will be kd + C*(log(kd+pi) + T)
#FLEXIBLE_SPHERES
automatic !
1. 0. 1.732
'log'
lv nb_theta nb_phi nb_mul

=====#FLEXIBLE-GMRES=====
FLEXIBLE GMRES inversion requires
for the inner GMRES
 1) an integer N --> the maximum of iterations
 2) an integer 1 --> write gmres diagnostic in some file
 0 --> not coded yet
 3) an integer M --> restart parameter
 4) a real eps-> the backward residual norm to
 achieve
for the outer flexible gmres

 1) an integer N --> the maximum of iterations
 2) an integer 1 --> write fgmres diagnostic in some file
 0 --> not coded yet
 3) an integer M --> restart parameter
 4) a real eps-> the backward residual norm to
 achieve
#FLEXI_GMRES
300

```

```
1
100
1.e-1
---- data for the outer loop ---
300
1
30
1.e-3
```

La résolution avec FGMRES nécessite 5 itérations pour la boucle extérieure avec un résidu égal à  $0.36 \times 10^{-3}$ . La solution a été obtenue après 59 minutes (4 processeurs sur le COMPAQ AlphaServer).

## 14 À partir d'un fichier CESC.dat

On peut modifier un fichier standard CESC.dat utilisé par le code généraliste CESC [2]. Pour cela, il suffit

1. d'enlever les deux clés #OPT\_SIMPLE et #OPT\_DOUBLE
2. de modifier la clé #EFIE ou #MFIE de telle sorte qu'on ait

```
#EQUATION_TYPE
EFIE
```

3. d'ajouter la clé #GMRES

```
#GMRES
100
1
30
1.e-4
```

## Références

- [1] N. Bartoli, F. Collino, and F. Dodu. Etude de l'interaction d'une antenne et d'une structure métallique par la "Fast Multipole Method - partie 3". Contract Report CR/EMC/02/101, CERFACS, Toulouse, France, 2002.
- [2] A. Bendali and M'B. Fares. Cerfacs electromagnetics solver code. Technical Report TR/EMC/99/52, CERFACS, Toulouse, France, 1999.
- [3] Q. Carayol. *Développement et analyse d'une méthode multipôle multiniveau pour l'électromagnétisme*. PhD thesis, 2002.
- [4] B. Carpentieri, I. S. Duff, and L. Giraud. Sparse pattern selection strategies for robust Frobenius-norm minimization preconditioners in electromagnetism. *Numerical Linear Algebra with Applications*, 7(7-8) :667–685, 2000.
- [5] B. Carpentieri, I.S. Duff, and L.Giraud. Experiments with sparse preconditioning of dense problems from electromagnetic applications. Technical Report TR/PA/00/04, CERFACS, Toulouse, France, 1999.
- [6] F. Collino and F. Millot. La méthode multipôle à deux composantes pour l'électromagnétisme. Technical Report TR/EMC/01/22, CERFACS, Toulouse, France, 2001.
- [7] F. Collino and F. Millot. Mise en place d'un préconditionneur appliqué à un problème d'électromagnétisme. Technical Report TR/EMC/02/122, CERFACS, Toulouse, France, 2003.
- [8] Eric Darve. The fast multipole method : numerical implementation. *J. Comput. Phys.*, 160(1) :195–240, 2000.
- [9] V. Frayssé, L. Giraud, and S. Gratton. A set of GMRES routines for real and complex arithmetics. Technical report, Cerfacs TR/PA/97/49, Toulouse, France, 1997.
- [10] V. Frayssé, L. Giraud, and S. Gratton. A set of Flexible GMRES routines for real and complex arithmetics. Technical report, Cerfacs TR/PA/98/20, Toulouse, France, 1998.
- [11] G. Sylvand. *Résolution itérative de formulation intégrale pour Helmholtz 3D : applications de la méthode multipôle à des problèmes de grande taille*. PhD thesis, Ecole Nationale des Ponts et Chaussées, 2002.