

**MANUEL**  
**OF**  
**“HYBRID” FINITE ELEMENT/BOUNDARY ELEMENT PROGRAM**

**September 2009**

Written by Pooneh MAGHOUL

## *Welcome in HYBRID!*

Dans ce travail, un code de calcul nommé HYBRID, initialement écrit par **Gatmiri** et amélioré par **Kamalian** (2000), **Nguyen** (2005) et **Dehghan** (2005) a été modifié et complété. Ce code est capable de résoudre des problèmes bidimensionnels (en déformation plane) statiques ou dynamiques transitoires. Il est particulièrement efficace pour traiter des problèmes dynamiques en géotechnique dans des domaines (semi-)infinis comme la propagation d'ondes, l'interaction dynamique sol-structure, le comportement dynamique des ouvrages et des fondations, l'effet local de site.

Le code HYBRID combine la méthode des éléments finis (FE) et la méthode des éléments frontières (BE). **Chaque partie, FE ou BE, peut être utilisée séparément** ou elles peuvent être couplées l'une avec l'autre pour analyser des problèmes plus complexes.

La partie FE est constituée d'**éléments finis de volume quadratiques à 8 noeuds secs**, qui ont un comportement purement mécanique sans influence de la pression d'eau interstitielle, ou des éléments saturés qui obéissent à la théorie des milieux poreux saturés, comportant donc un couplage hydro-mécanique ou des éléments non-saturés qui obéissent à la théorie des milieux poreux non-saturés, comportant donc un couplage hydro (eau)- hydro (air)-mécanique.

Cette partie FE peut modéliser différents types de comportement matériel: élastique linéaire, élastique non-linéaire de type hyperbolique (modèle [Duncan et Chang, 1970] et [Hardin et Drnevich, 1972]) ou élasto-plastique (modèle [Prevost, 1985]).

La partie BE, quant à elle, constituée d'**éléments frontières linéiques quadratiques à 3 noeuds secs**, saturés ou non-saturés élastiques linéaires (les éléments saturés utilisent la solution fondamentale simplifiée proposée par Nguyen (2005) et les éléments non-saturés utilisent la solution fondamentale proposée dans ce travail).

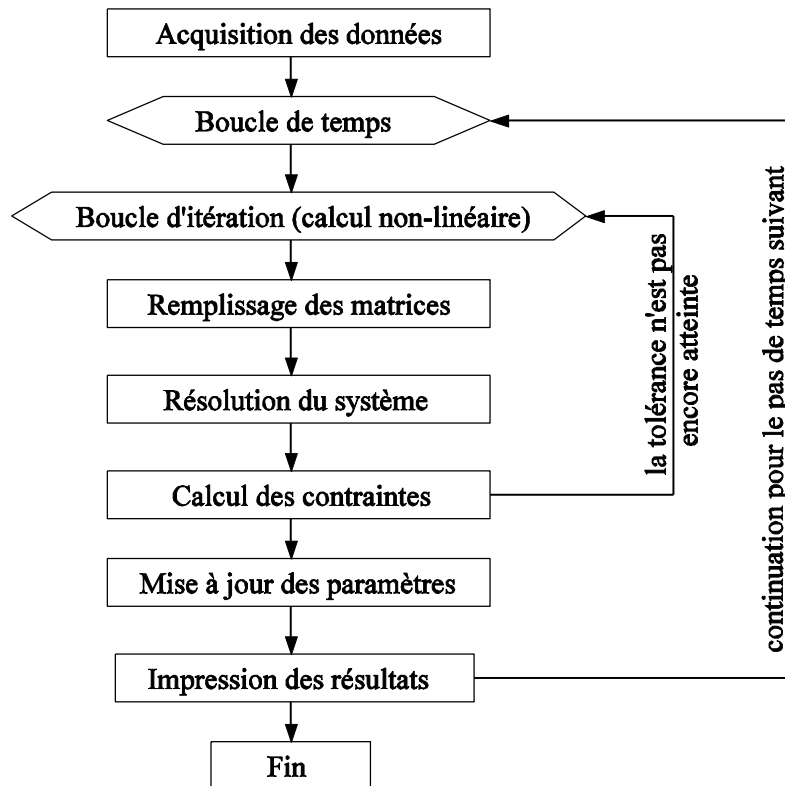
Grâce à ce programme, les déformations, les contraintes et les pressions interstitielles de l'eau et de l'air (suction) sont calculées.

En tenant compte des avantages et des inconvénients de chaque méthode, la partie FE sert à modéliser l'ouvrage et le sol alentour, qui peuvent avoir un comportement complexe, alors que la partie BE est utilisée pour modéliser le sol dans le champ lointain, supposé élastique linéaire.

Le code est écrit en Fortran (norme 90). Il s'agit d'un programme exécutable, qui est alimenté par un fichier de données en texte et fournit en sortie un fichier de résultats en texte. Ce programme se compose de 4 parties fonctionnelles distinctes :

- 1) Acquisition des données de base: il s'agit des informations, lues à partir d'un fichier de données en texte, concernant le maillage, le chargement, les matériaux, les conditions aux limites, les conditions initiales et d'autres paramètres. Le stockage de ces données est regroupé dans un seul vecteur (a); l'accès aux différentes variables se fait donc par l'intermédiaire d'un système d'index flottants.
- 2) Remplissage des matrices: c'est une partie cruciale du programme. La tâche principale est la construction de la matrice résolvante (matrice de rigidité de la zone FE, matrice  $\mathbf{G}^I$  de la zone BE ou matrice de rigidité équivalente de la zone FE/BE). Le traitement est différent selon les éléments (FE ou BE) et le comportement associé à l'élément (élastique ou plastique, linéaire ou non). Cette matrice est stockée selon la méthode dite 'ligne de ciel' (skyline).
- 3) Résolution du système: la méthode pour résoudre le système d'équations est fonction du type de problème considéré. Si seuls des éléments finis secs sont utilisés, alors la méthode de résolution sera la méthode de Mondkar & Powel, applicable aux matrices symétriques. Par contre, s'il existe des éléments finis saturés ou non-saturés ou des éléments frontières, on devra utiliser la méthode de Taylor applicable aux matrices asymétriques.
- 4) Traitement des résultats, mise à jour des données: une subroutine calcule les vecteurs des contraintes incrémentales et totales; une autre met à jour les déplacements nodaux et les pressions interstitielles pour l'itération suivante ; et un ensemble de subroutines envoie les résultats vers différents fichiers de sortie.

La structure globale du programme est donnée par la figure suivante.



On s'intéresse par la suite à la structure générale du code formé d'un programme principal, qui à son tour appelle 20 sous-programmes et 66 sous-programmes secondaires. Le corps de ce code traite les points fondamentaux suivants:

## 1. Phase d'acquisition des données

Lors de cette étape toutes les données concernant la mode de calcul, la méthode de résolution, la définition géométrique des éléments, les caractéristiques mécaniques des matériaux, le chargement, les conditions aux limites, les conditions initiales et d'autres paramètres sont lus et stockés.

### i) Subroutine **DIM1** : Lit et stocke des données principales

- Les données principales telle que, le titre du programme (*title*), le nombre total des nœuds (*nnp*), le mode du calcul (FE, BE ou FE/BE) (*ifem*, *ibem*), le nombre total des éléments drainés (*ne8d*) saturés (*ne8c*) ou bien non saturés (*ne8u*), le nombre

total des zones D1 (*nbed1*) et le nombre total des nœuds dans chacune de ces zones (*nnpbed1 (1:nbed1)*) , le nombre total des zones D2 (*nbed2*) et le nombre total des nœuds dans chacune de ces zones (*nnpbed2 (1:nbed2)*), le nombre total des zones D3 (*nbed3*) et le nombre total des nœuds dans chacune de ces zones (*nnpbed3 (1:nbed3)*), le nombre des matériaux drainés dans la zone FE (*m8d*) saturés (*m8c*) ou bien non saturés (*m8u*) et leurs types de comportement (*m8dep (1:m8d)*, *m8cep (1:m8c)*, *m8uep (1:m8u)*), le type de chargement (*idyn*), le code de séisme (*ieaq*), le nombre des étapes de chargement (*nload*), le nombre des pas de temps (*ntime*), l'indicateur du type de solution (symétrique ou asymétrique) (*isolv*), le code d'analyse (Newton-Raphson ou Newton-Raphson modifié) (*nr*), le minimum et le maximum des pas de temps (*dtimei* et *dtimeu*), le constant time-integration (*ptoll1*), la tolérance permmissible (*ptoll3*), le code de K-correction (*kcorf*), le point du cutting de l'histoire proche (*mtime*) et la tolérance de K-correction (*rmtol*)

- les conditions aux limites nulles comme le nombre des conditions aux limites nulles à chaque direction (*nbcx*, *nbcy*), les conditions aux limites nulles concernant la partie mécanique, l'eau (*nbcw*) et l'air (*nbcu*) et le nombre des pas de temps (*anew1*, *anew2*, *wil*, *intb*)
- les poids volumiques de l'eau et de l'air, la pression atmosphérique, l'accélération due à la pesanteur.

En plus, la première série des dimensions dynamique du vecteur *a* (11-138) est déterminée dans ce subroutine.

**ii) Subroutine INPUT :** 19 subroutines sont appelés dans ce sous-programme pour lire et stocker

- **COORC** (*nnp*, *x*, *nmiss*) : les coordonnées des noeuds dans le système cartésien (*x (1:2, n)*).
- **COORP** (*nnp*, *x*, *nmiss*) : les coordonnées des noeuds dans le système polaire (*x (1:2, n)*) et se convertir au système cartésien.

- **ELM8D** (ne8d, ie8d) : les caractéristiques géométriques (*ie8d (1:8, ne8d1)*) et mécanique (le type de matériau) (*ie8d (9, ne8d1)*) des éléments drainés dans la zone FE.
- **ELM8C** (ne8c, ie8c) : les caractéristiques géométriques (*ie8c (1:8, ne8c1)*) et mécanique (le type de matériau) (*ie8c (9, ne8c1)*) des éléments saturés dans la zone FE.
- **ELM8U** (ne8u, ie8u) : les caractéristiques géométriques (*ie8u (1:8, ne8u1)*) et mécanique (le type de matériau) (*ie8c (9, ne8u1)*) des éléments non-saturés dans la zone FE.
- **BLOCD** (nnp, x, ne8d, ie8d, nmiss) : les caractéristiques géométriques (*ie8d (1:8, ne8d1)*) et mécanique (le type de matériau) (*ie8d (9, ne8d1)*) des super-éléments drainés dans la zone FE. Si cette subroutine s'utilise pour lire les données, éviter de définir ces nœuds et éléments dans COORC, COORP et ELM8D.
- **BLOCC** (nnp, x, ne8c, ie8c, nmiss) : les caractéristiques géométriques (*ie8c (1:8, ne8c1)*) et mécanique (le type de matériau) (*ie8c (9, ne8c1)*) des super-éléments saturés dans la zone FE. Si cette subroutine s'utilise pour lire les données, éviter de définir ces nœuds et éléments dans COORC, COORP et ELM8C.
- **BLOCU** (nnp, x, ne8u, ie8u, nmiss) : les caractéristiques géométriques (*ie8u (1:8, ne8u1)*) et mécanique (le type de matériau) (*ie8u (9, ne8u1)*) des super-éléments non-saturés dans la zone FE. Si cette subroutine s'utilise pour lire les données, éviter de définir ces nœuds et éléments dans COORC, COORP et ELM8U.
- **BEMD1** (ibem, nnp, nbed1, nnpbed1, ie3d1): les nœuds des zones finies (D1) de BE *ie3d1 (1, nbed11)*. Cette subroutine n'est pas vérifiée et testée.
- **BEMD2** (ibem, nnp, nbed2, nnpbed2, ie3d2): les nœuds des zones infinies (D2) de BE *ie3d2 (1, nbed21)*. Cette subroutine n'est pas vérifiée et testée.
- **BEMD3** (ibem, nnp, nbed3, nnpbed3, ie3d3): les nœuds des zones semi-infinies (D3) de BE (*ie3d3 (1:nnpbed3)*). le caractéristique mécanique (le type de matériau ; [1] : BE sec, [2] : BE saturé, [3] : BE non saturé) (*ie3d3 (nnpbed3+1)*).
- **BEMEN** (nnpn, xencl, yencl): les nœuds des zones clôturée (enclosed elm.) *xencl (nnpn), yencl (nnpn)*.

- **MATED** (m8d, sm8d, m8dep, ibem, ne8d): les propriétés des matériaux drainées *sm8d (60, m8d)*
- **MATEC** (m8c, sm8c, m8cep, ibem, ne8c): les propriétés des matériaux saturées *sm8c (60, m8c)*
- **MATEU** (m8u, sm8u, m8uep, ibem, ne8u): les propriétés des matériaux non saturées *sm8u (60, m8u)*
- **BOUND** (nnp, nbcx, nbcy, nbcw, nbca, kodex, kodey, kodew, kodea): cette subroutine lit et engendre la degré de liberté fixe *kodex, kodey, kodew, kodea*.
- **PRTOUT**: cette subroutine lit le numéro des nodes lesquels on veut imprimer leurs résultats *init, iprint, iterp, nnpo, ne8do, ne8co, ne8uo, itprt, inpout, ie8dout, ie8cout, ie8uout*.
- **PHASE** (nload, iconst): cette subroutine lit les phases de construction *iconst*.
- **INITI** (idyn): cette subroutine lit les valeurs *npwi, nnpwi, vnpwi; npai, nnpai, vnpai; ndispi, nndispi, vxdispi, vydispi; nveli, nnveli, vxveli, vyveli; nacci, nnacci, vxacci, vyacci*

iii) **Subroutine DIM2**: Lit et détermine la degré de liberté en considérant des conditions aux limites (*mdof*) et sans considérant des conditions aux limites (*mdofn*) et la deuxième série de la dimension dynamique des vecteurs (*139-171*).

iv) Subroutine **DETNA** :