USER'S GUIDE

FOR

"HYBRID" FINITE ELEMENT/BOUNDARY ELEMENT PROGRAM

September 2009

Written by Pooneh MAGHOUL

Welcome in HYBRID!

I. Basic data (DIM1.F)

1. TITLE (char*72)

   Name of the current problem

2. CARD (char*5)

   This character is equal to 'basic'

3. NNP, IFEM, IBEM (Integer)

   NNP : total number of nodes in problem (FE & BE zones)

   IFEM : FEM code [0, 1];

   [0]: program studies just the topographical effect by BEM (FE is not done),

   [1]: program studies geotechnical effect in a 2D topographical structure by hybrid FEM/BEM (FE is done)

   IBEM : BEM code, [0, 1, 2, 3];

   [0]: program doesn't consider BEM analysis. Except some simple examples IBEM must never be [0],

   [1]: BEM_dry,

   [2]: BEM_saturate,

   [3]: BEM_unsaturate

If IFEM=0, line 4 is ignored:

4. NE8D, NE8C, NE8U (Integer)

   NE8D : number of drained (dry) elements in FE zone

   NE8C : number of consolidated (saturated) elements in FE zone

   NE8U : number of unsaturated elements in FE zone

If IBEM=0, line 5 is ignored:

5. NBED1, NBED2, NBED3, NBEINT (Integer)

   NBED1 : number of D1 BEM areas (finite areas), it can be at maximum 5 zones

   NBED2 : number of D2 BEM areas (infinite areas), it can be at maximum 5 zones

   NBED3 : number of D3 BEM areas (semi-infinite area), it can be at maximum 1 zone

   NBEINT : number of internal nodes, it can be at maximum 10

If NBED1>0, line 6 if not, this line is ignored:

6. NNPBED1 (i), i=1,NBED1 (Integer (5))

   NNPBED1 : number of nodes of each D1 BEM area

If NBED2>0, line 7 if not, this line is ignored:

7. NNPBED2 (i), i=1,NBED2 (Integer (5))

   NNPBED2 : number of nodes of each D2 BEM area

If NBED3>0, line 8 if not, this line is ignored:

8. NNPBED3, NNPEN (Integer)

   NNPBED3 : number of nodes of one D3 BEM area

   NNPBED3INT : number of internal nodes in D3 BEM area

   NNPEN : number of nodes of fictitious enclosing elements in D3 BEM area

If IBEM ≠ 2, line 10 if not, this line is ignored:

9. KFSOL (Integer)

   KFSOL : kind of fundamental solution in saturated zone [1,2];

   [1]: numerical Laplace inverse,

   [2]: analytical Laplace inverse (incompressible component)

10. M8D, M8C, M8U (Integer)

    M8D : number of drained (dry) materials in FE & BE zones; in BE area the program is written for just one material (homogeneous space) then if we use a dry material in BE zone (by considering the dry materials in FE zone) the number of material of BE zone is always equal to M8D because the materials from "1" to "M8D-1" are used in FE zone

    M8C : number of consolidated (saturated) materials in FE & BE zones; in BE area the program is written for just one material (homogeneous space) then if we use a saturated material in BE zone (by considering the saturated materials in FE zone) the number of material of BE zone is always equal to M8C because the materials from "1" to "M8C-1" are used in FE zone

    M8U : number of unsaturated materials in FE & BE zones; in BE area the program is written for just one material (homogeneous space) then if we use an unsaturated material in BE zone (by considering the unsaturated materials in FE zone) the number of material of BE zone is always equal to M8U because the materials from "1" to "M8U-1" are used in FE zone

If M8D>0, line 12 if not, this line is ignored:

11. M8DEP (i), i=1,M8D (Integer (10))

    M8DEP : behaviour type of drained (dry) materials [1, 2, 3, 4];

    [1]: linear,

    [2]: hyperbolic for static,

    [3]: hyperbolic for dynamics,

    [4]: elastoplastic; if we use a dry material in BE zone then the behaviour type of BE zone's material is: M8DEP(m8d)

If M8C>0, line 13 if not, this line is ignored:

12. M8CEP (i), i=1,M8C (Integer (10))

    M8CEP : behaviour type of drained (dry) materials [1, 2, 3, 4];

    [1]: linear,

    [2]: hyperbolic for static,

    [3]: hyperbolic for dynamics,

    [4]: elastoplastic; if we use a dry material in BE zone then the behaviour type of BE zone's material is: M8CEP(m8c)

If M8U>0, line 14 if not, this line is ignored:

13. M8UEP (i), i=1,M8U (Integer (10))

    M8UEP : behaviour type of drained (dry) materials [1, 2, 3, 4];

    [1]: linear,

    [2]: hyperbolic for static,

    [3]: hyperbolic for dynamics,

[4]: elastoplastic; if we use a dry material in BE zone then the behaviour type of BE zone's material is: M8UEP(m8u)

14. IDYN, IEAQ, NLOAD, NTIME, ISOLVE, NR (Integer) DTIMEI, DTMEM, PTOLL1, PTOLL3 (Real*8)

IDYN : dynamic code or type of loading [0, 1, 2];

[0]: static,

[1]: consolidated (quasi-static),

[2]: dynamic problem,

IEAQ : earthquake code [0, 1, 2, 3];

[0]: static case,

[1]: acceleration,

[2]: imposed displacement,

[3]: incident wave, in most of the time IEAQ = [3]

NLOAD : number of load steps; load is applied in "nload" steps, in each "iload=1: nload" the characteristics of loading such as 'intensity - nstep - ...' are different

NTIME : number of time steps or total number of "nstep"s in all "nload"s. loading is divided into SUM[nstep]s, when the characteristics of "nstep"s are similar we put these "nsteps" in one group which is named iload

ISOLVE : solution type indicator

[1]: for a symmetric computation

[3]: for a non-symmetric computation in most of the case is solved by ISOLV = [3]

NR : type newton raphson code [0, 1];

[0] modified,

[1] normal,

DTIMEI : minimum time step

DTIMEM : maximal time step

PTOLL1 : time-integration constant-beta in FE formulation, for ISOLVE = 3, PTOLL1 > 0.5

PTOLL3 : permissible tollerance in FE formulation

15. KCORF, MTIME (Integer) RMTOL (Real*8)

KCORF : K-correction parameters: K-correction code [0, 1];
[0]: Wrobel method,
[1]: time truncated method

MTIME : K-correction parameters: cutting point of near history. Time integration is limited to the number of time steps (MTIME) in time truncation method

RMTOL : K-correction parameters: K-correction tolerance

16. NBCX, NBCY, NBCW, NBCA (Integer)

NBCX : number of (null) boundary conditions introduced for x-displacements

NBCY : number of (null) boundary conditions introduced for y-displacements

NBCW : number of (null) boundary conditions introduced for nodal water pressure

NBCA : number of (null) boundary conditions introduced for nodal air pressure

17. GAMMAW, GAMMAS, GAMMAA, GRAV, ATMP (Real*8)

GAMMAW : unit weight of water [ ]

GAMMAS : unit weight of solid [ ]

GAMMAA : unit weight of air [ ]

GRAV : acc. due to gravity [m.s-2]

ATMP : ambient atmospheric pressure (taken constant) [Pa]

18. ANEW1, ANEW2, WIL (Real*8) INTB (Integer)

ANEW1 : Gamma coefficient in Newmark direct integration method. This coefficient must be [>=0.5] because that the convergence be

unconditionally stable. In this program, it is chosen 0.5. This choice corresponds to the rule of the trapezoid

ANEW2 : Beta coefficient in Newmark direct integration method. This coefficient must be [>=0.25] because that the convergence be unconditionally stable. In this program, it is chosen 0.25. This choice corresponds to the rule of the trapezoid

WIL : θ coefficient in θ-Wilson method for the stability of numerical solution [>1]. The coefficient plays a numerical damping role of artificial wave propagation due to the perturbations. When θ = [1] it is meaning that θ-Wilson method is not considered

INTB : interpolation function of time [1, 2, 3];

[1]: constant: two fields of displacement and stress (u,t) are supposed to be constant in each time interval,

[2]: linear: two fields of displacement and stress (u,t) vary linearly in each time interval,

[3]: mix: displacement field (u) vary linearly while stress field (t) is constant in each time interval

19. LCQM, PCQM, EPSCQM (Real*8)

LCQM :

PCQM :

EPSCQM :

II. Meshing data (COORC, COORP, ELM8D, ELM8C, BLOCD, BLOCC, BEMD1, BEMD2, BEMD3, BEMEN)

20. N, NG (Integer), X (1, N), X (2, N) (Real*8) (N = 1, NNP)

N : node number

NG [0, 1] : increment code

[1]: it is created the new nodes automatically between N1 and N2 with the increment which is calculated in the program

[0]: there will not be created new nodes between N1 and N2, these two nodes are continuous

X (1, N) : x-coordinate of the N-th node

X (2, N) : y-coordinate of the N-th node

21. N1, N2, LG, IDIR, Xo, Yo (convertir des coordonnées polaires des noeuds)

22. N, NG, IE8D (I, N) (I = 1, 9 & N = 1, NE8D) (Integer)

N : drained element number

NG [0, 1] : increment code

IE8D (1: 8, N) : number of global nodes forming the current drained element corresponding with the local nodes in counter-clockwise direction and in increasing order

IE8D (9, N) : code defining the material type to characterize the behaviour of the current drained element

23. N, NG, IE8C (I, N) (I = 1, 9 & N = 1, NE8C) (Integer)

N : consolidated element number

NG [0, 1] : increment code

IE8C (1: 8, N) : node numbers of the nodes forming the current consolidated element

IE8C (9, N) : code defining the material type to characterize the behaviour of the

current consolidated element

24. N1, E1, NE1, NE2, MAT, X1, Y1, X2, Y2, X3, Y3, X4, Y4 (BLOCD)

N1 : node number

E1 : element number

NE1 : number of element of each cote

NE2 : number of element of each cote

25. N1, E1, NE1, NE2, MAT, X1, Y1, X2, Y2, X3, Y3, X4, Y4 (BLOCC)

N1 : node number

E1 : element number

NE1 : number of element of each cote

NE2 : number of element of each cote

26. IE3D1, N, NG, IZONE (Integer)

IE3D1 (NNPBED1 (I) +1, I), I = 1, NBED1:

N : node number in BEMD1

NG : increment code

IZONE : zone BEMD1 number

27. IE3D2, N, NG, IZONE (Integer)

IE3D2 (NNPBED2 (I) +1, I), I = 1, NBED2:

N : node number in BEMD2

NG : increment code

IZONE : zone BEMD2 number

28. IE3D3, N, NG (Integer)

IE3D3 (NNPBED3 (I) +1, I), I = 1, NBED3:

N : node number in BEMD3

NG : increment code

29. N, NG (Integer) XENCL, YENCL (Real*8)

N : node number in BEMEN

NG : increment code

XENCL (n) : X-coordinates of enclosing element nodes (nodal data)

YENCL (n) : Y-coordinates of enclosing element nodes (nodal data)

III. Material data (MATED, MATEC, MATEU)

Drained material data

If "NE4D = 0 and IBEM≠1", go to 32.

For I=1, M4D, write lines 30-31.

30. IMAT, ICPT (Integer)

IMAT : drained material number in MATED (from 1 to M8D)

ICPT [1, 2, 3, 4]: [1] linear elastic; [2] hyperbolic elastic for static; [3] hyperbolic elastic for dynamic; [4] elastoplastic

If ICPT = 1

31. SM8D (3, IMAT), SM8D (7, IMAT), SM8D (10, IMAT), SM8D (11, IMAT)

SM8D (3, IMAT) : K1, loading modulus used in non-linear elasticity (adimensional) $E = K_1 P_{atm}$

SM8D (7, IMAT) : K_b , bulk modulus (adimensional)

SM8D (10, IMAT): K_0

SM8D (11, IMAT): gamma γ_s, soil volumetric weight ($N.m^{-3}$)

Saturated material data

If "NE4C = 0 and IBEM≠2", go to 34.

For I=1, M4C, write lines 32-33.

32. IMAT, ICPT (Integer)

IMAT : consolidates material number in MATEC (from 1 to M8C)

ICPT [1, 2, 3, 4] : [1] linear elastic; [2] hyperbolic elastic for static; [3] hyperbolic elastic for dynamic; [4] elastoplastic

If ICPT = 1

33. SM8C (3, IMAT), SM8C (7, IMAT), SM8C (10, IMAT), SM8C (11, IMAT), SM8C (13, IMAT), SM8C (14, IMAT), SM8C (15, IMAT), SM8C (16, IMAT),

SM8C (3, IMAT) : K1, loading modulus used in non-linear elasticity (adimensional) $E = K_l P_{atm}$

SM8C (7, IMAT) : K_b , bulk modulus (adimensional)

SM8C (10, IMAT): K_0

SM8C (11, IMAT): gamma, soil volumetric weight ($N.m^{-3}$) $\gamma = \gamma_w n + \gamma_s (1-n)$

SM8C (13, IMAT): Kz, vertical water permeability (m.s-1)

SM8C (14, IMAT): α, Biot's coefficient

SM8C (15, IMAT): compm

SM8C (16, IMAT): k_x/k_z

Unsaturated material data

If "NE4U = 0 and IBEM≠3", go to 36.

For I=1, M4U, write lines 34-35.

34. IMAT, ICPT (Integer)

IMAT : unsaturated material number in MATEU (from 1 to M8U)

ICPT [1, 2, 3, 4] : [1] linear elastic; [2] hyperbolic elastic for static; [3] hyperbolic elastic for dynamic; [4] elastoplastic

If ICPT = 1

35. SM8C (3, IMAT), SM8C (5, IMAT), SM8C (6, IMAT), SM8C (7, IMAT), SM8C (11, IMAT), SM8C (14, IMAT), SM8C (15, IMAT), SM8C (16, IMAT), SM8C (17, IMAT), SM8C (19, IMAT), SM8C (20, IMAT), SM8C (21, IMAT), SM8C (22, IMAT), SM8C (23, IMAT), SM8C (24, IMAT), SM8C (25, IMAT), SM8C (27, IMAT), SM8C (28, IMAT), SM8C (31, IMAT), SM8C (33, IMAT)

SM4U(3, I) : K_1 , loading modulus used in non-linear elasticity (adimensional)

SM4U(5, I) : K_a , air bulk modulus (Pa)

SM4U(6, I) : K_w , air bulk modulus (Pa)

SM4U(7, I) : K_b , bulk modulus used in non-linear elasticity (adimensional)

SM4U(11, I) : Young modulus minimal value (Pa)

SM4U(14, I) : S_{w0}, initial saturation degree (adimensional)

SM4U(15, I) : a, constant ($m.s^{-1}$) used in the formula of water permeability

$$K_w = a.10^{\alpha_w.e}\left(\frac{S_w - S_{w,r}}{1 - S_{w,r}}\right)^3\left(\frac{\mu_w(T)}{\mu_w(T_0)}\right)$$

SM4U(16, I) : α_w, constant (adimensional) used in the formula of water permeability (See explanations for SM4U(15, I))

SM4U(17, I) : $S_{w,r}$, residual saturation degree (adimensional), used in the formula of water permeability (See explanations for SM4U(15, I))

SM4U(19, I) : b, constant (m^2) used in the formula of air permeability

$$K_a = \frac{b\gamma_a}{\mu_a}\left(e(1 - S_w)\right)^{\alpha_a}$$

SM4U(20, I) : α_a, constant (adimensional) used in the formula of air permeability (See explanations for SM4U(19, I))

SM4U(21, I) : "viscosity", dynamic air viscosity μ_a ($N.s.m^{-2}$), used in the formula of air permeability (See explanations for SM4U(19, I)). Generally, $\mu_a = 1{,}846.10^{-5}$ $N.s.m^{-2}$.

SM4U(22, I) : K_w^{max}, maximal authorized water permeability value ($m.s^{-1}$)

SM4U(23, I) : K_a^{max}, maximal authorized air permeability value ($m.s^{-1}$)

SM4U(24, I) : a_e, constant (adimensional) used in the formula of the void ratio state surface

$$e = (1 - e_0)\exp\left[-\frac{\left[a_e(p'/p_{atm}) + b_e(1 - p'/\sigma_e)(s/p_{atm})\right]^{1-d_e}}{K_b(1 - d_e)}\right]\exp\left[-c_e(T - T_0)\right] - 1$$

SM4U(25, I) : b_e, constant (adimensional) used in the formula of the void ratio state surface (See explanations for SM4U(24, I))

SM4U(27, I) : e_0, constant (adimensional) used in the formula of the void ratio state surface (See explanations for SM4U(24, I))

SM4U(28, I) : σ_e, maximal traction resistance (Pa) used in the formula of the void ratio state surface (See explanations for SM4U(24, I))

SM4U(31, I) : β_w, constant (Pa^{-1}) used in the formula of the saturation degree state surface (See explanations for SM4U(29, I))

SM4U(33, I) : "msuc-e"

IV. Boundary conditions (BOUND)

36. N1, N2, LG, KODE

N1 : node number

N2 : node number

LG : increment

KODE [1, 2, 3] : boundary condition code, [1]: U_x=0; [2]: U_y=0; [3]: P=0

V. Résultats à imprimer (PRTOUT)

37. INIT, IPRINT, ITERP, NNPO, NE8DO, NE8CO

INIT : code describing the type of initial conditions (initial stress code for soil)

- =0 if the results are taken from a previous analysis
- =1 if initial normal stresses are considered equal to the weight of the upper soil layers
- =2 if initial stresses are computed for the steady state of the system
- =3 if initial stresses are calculated as the weight of the upper soil layers, considering that the system has a mere parallelogram shape, and that the x-direction of the space-frame corresponds to the physical horizontal
- =4 if initial stresses are given on some elements (or on whole elements)

IPRINT : control code for the input and output files (initial data output print code)

- =0 to avoid the outprint of the mesh file
- =1 to print out the mesh file

ITERP : number of time steps for which the results are printed out in the output file

NNPO : number of total nodes that we need their outputs

NE8DO : number of drained elements that we need their outputs

NE8CO : number of consolidated elements that we need their outputs

38. ITPRT (I), I = 1, ITERP

ITPRT (I) : time steps numbers which are selected to print out the results

[0] if no printout is expected

39. INPOUT (I), I = 1, NNPO

INPOUT :

40. IE8DOUT (I), I = 1, NE8DO

IE8DOUT :

41. IE8DOUT (I), I = 1, NE8CO

IE8COUT :

VI. Loading data for each load step (PHASE)

42. do I = 1, NLOAD: ICONST (J, I), J = 1, 3

ICONST (1, I) : intial timestep number of the current loadstep

ICONST (2, I) : number of loading substeps for the current loadstep

ICONST (3, I) : number of iterations for each substep

VII. Initial conditions (INITI)

43. NACCI, NVELI, NDISPI, NPWI

NACCI : number of nodes with initial acceleration values

NVELI : number of nodes with initial velocity values

NDISPI : number of nodes with initial displacement values

NPWI : number of nodes with initial p-water values

If NACCI ≠ 0

44. NNACCI (I), VXACCI (I), VYACCI (I), (I = 1, NACCI)

NNACCI : numbers of nodes with initial acceleration values

VXACCI : initial accelerations for the nodes with initial acceleration values (X-dir)

VYACCI : initial accelerations for the nodes with initial acceleration values (Y-dir)

If NVELI ≠ 0

45. NNVELI (I), VXVELI (I), VYVELI (I), (I = 1, NVELI)

NNVELI : numbers of nodes with initial velocity values

VXVELI : initial velocities for the nodes with initial velocity values (X-dir)

VYVELI : initial velocities for the nodes with initial velocity values (Y-dir)

If NDISPI ≠ 0

46. NNVELI (I), VXVELI (I), VYVELI (I), (I = 1, NVELI)

NNDISPI : numbers of nodes with initial displacement values

VXDISPI : initial displacements for the nodes with initial displacement values (X-dir)

VYDISPI : initial displacements for the nodes with initial displacement values (Y-dir)

If NPWI ≠ 0

47. NNVELI (I), VXVELI (I), VYVELI (I), (I = 1, NVELI)

NNPWI : numbers of the nodes on which an initial water pressure is imposed

VNPWI : initial water pressure values imposed on the preceding nodes