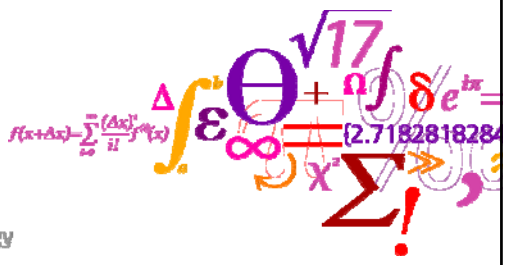
HAWC2

Soil spring modeling

Torben J. Larsen

Risø DTU
National Laboratory for Sustainable Energy



Soil spring forces – pile in sand

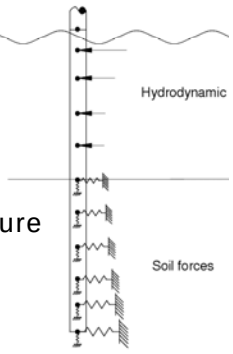
Lateral soil pressure

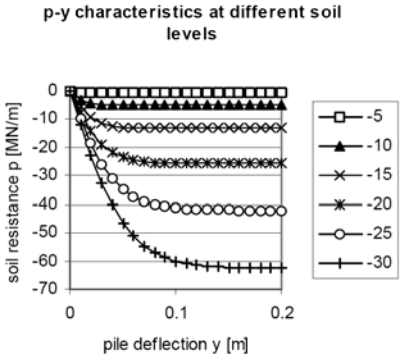
$$p(y) = A P_u \tanh\left(y k \frac{X}{A P_u} \right)$$

$$P_u = \min \begin{cases} (C_1 X + C_2 D)\gamma' X \\ C_3 D \gamma' X \end{cases}$$

Vertical soil pressure

$$t(z) = \frac{2G_0}{D \ln(Z_{IF})} z$$





γ'	[N/m³]	Submerged unit weight of soil
$C_{1\text{-}3}$	[-]	Coefficients as function of friction angle
D	[m²]	Cross sectional diameter
G_0	[Pa]	Initial shear modulus of the soil
K	[Pa/m]	Initial modulus of subgrade reaction
P_u	[N/m]	Ultimate lateral soil strength
Y	[m]	Lateral deflection
X	[m]	Height above mud level
Z_{IF}	[-]	Radius of influenced soil zone divided by pile radius

Soil spring example

```
begin soil;
  begin soil_element;
    mbdy_name pile_substructure;
    datafile ./data/soil.dat;
    soilsections uniform 30; Distribution of soil calculation points from mbdy node 1 to n
    damping_k_factor 0.01 ; Rayleigh damping based on soil stiffness matrix
    set 1;  lateral
    set 2;  axial
    set 3;  rotation_z
  end soil_element;
end soil;
```


Soil spring data file

```
This is a nonlinear soil spring demonstration file
#1
lateral                 (axial/lateral)
5 4                      nrow ndefl
        0.0     0.1      0.2     1.0    x1 x2 x3 .....   [m]
0.0     0       15       20      500    Z_G  P_1 P_2 P_3 .... F_ndefl [kN/m]
10.0    0       15       20      500
20.0    0       15       20      500
30.0    0       15       20      500
40.0    0       15       20      500
#2
axial                   (axial/lateral)
5 4                      nrow ndefl
        0.0     0.1      0.2     1.0    x1 x2 x3 .....   [m]
0.0     0       150      200     5000   Z_G  P_1 P_2 P_3 .... F_ndefl [kN/m]
10.0    0       150      200     5000
20.0    0       150      200     5000
30.0    0       150      200     5000
40.0    0       150      200     5000
#3
rotation_z              (axial/lateral/rotation_z)
5 4                      nrow ndefl
        0.0     0.1      0.2     1.0    x1 x2 x3 .....   [rad]
0.0     0       150      200     5000   Z_G M_1 M_2 M_3 .... M_ndefl [kNm/m]
10.0    0       150      200     5000
20.0    0       150      200     5000
30.0    0       150      200     5000
40.0    0       150      200     5000
```

External Force DLL :: Commands

- The following lines should be included in htc-file for a coupled soil spring analysis:

```
;----------------------------------------
begin FORCE ;
  begin DLL ;
    dll     ./externalforce/DiagonalStiffnessMatrixDLL.dll ;
    update  DiagonalStiffnessMatrixDLL ;
    mbdy    Foundation ;
    node    1 ;
  end DLL ;
end FORCE ;

;----------------------------------------
```


External Force DLL :: DLL prototype

```
SUBROUTINE DiagonalStiffnessMatrixDLL(time,x,xdot,xdot2,amat,omega,omegadot,F,M)
!DEC$ ATTRIBUTES DLLEXPORT::DiagonalStiffnessMatrixDLL
!DEC$ ATTRIBUTES ALIAS:'diagonalstiffnessmatrixdll' :: DiagonalStiffnessMatrixDLL
! input
DOUBLE PRECISION                    :: time             ! time
DOUBLE PRECISION ,DIMENSION(3)      :: x                ! global position of reference node
DOUBLE PRECISION ,DIMENSION(3)      :: xdot             ! global velocity of reference node
DOUBLE PRECISION ,DIMENSION(3)      :: xdot2            ! global acceleration of reference node
DOUBLE PRECISION ,DIMENSION(3)      :: omega            ! angular velocity of reference node (global base)
DOUBLE PRECISION ,DIMENSION(3)      :: omegadot         ! angular acceleration of reference node (global base)
DOUBLE PRECISION ,DIMENSION(3,3) :: amat             ! rotation matrix (body -> global)
! output
DOUBLE PRECISION ,DIMENSION(3)      :: F                ! External force in reference node (global base)
DOUBLE PRECISION ,DIMENSION(3)      :: M                ! External moment in reference node (global base)
! locals
LOGICAL, SAVE                       :: bInit = .FALSE. ! Initialisation flag
DOUBLE PRECISION, SAVE              :: kx,ky,kz         ! Stiffness in translation
DOUBLE PRECISION, SAVE              :: mx,my,mz         ! Stiffness in rotation
DOUBLE PRECISION, SAVE              :: x0(3)            ! Initial location of node where force is applied
!
! Initialise on first call
IF (.NOT.bInit) THEN
  bInit = .TRUE.
  ! Init stiffnesses
  open(20,file='externalforce\extforceparms.inp')
  read(20,*) kx      ! Vertical stiffness   [N/m]
  read(20,*) kz      ! Horizontal stiffness [N/m]
  read(20,*) mx      ! Rotational stiffness about horizontal axes [Nm/rad]
  read(20,*) mz      ! Rotational stiffness about vertical axis   [Nm/rad]
  read(20,*) x0(1)   ! Initial location of node where force is applied
  read(20,*) x0(2)
  read(20,*) x0(3)
  close(20)
  !-------------------------------------------------------------------------------
  ky = kx
  my = mx
ENDIF
!
! Calc. force

F = (/ kx*(x0(1) - x(1)), &
       ky*(x0(2) - x(2)), &
       kz*(x0(3) - x(3))  /)
!
M = (/-mx*amat(3,2), &
      -my*amat(1,3), &
      -mz*amat(2,1)  /)
!
END SUBROUTINE DiagonalStiffnessMatrixDLL
```