Appendix 1: M File

```
clear; clc; close all;

 %Input Data

u = 0; % Wave Velocity Initial Value.
a = 0; % Wave Acceleration Initial Value.
n = 0; % n-th Moment Spectrum Power Initial Value.
tmax = 10800; % Max Run Time "Storm Duration". "input"
fmax = 1; % Max Wave Frequency ( Hz )."input"

nt = 0.1; % Delta t. "input"

nf = (fmax.*nt)./tmax; % Delta f.
t = 0:nt:tmax; % Run Time ( 10800 sec. = 3 hrs. ) "Storm Duration".
g = 9.81; % Gravitational Acceleration ( m/sq. sec.). "input"

Density = 1029; % Sea Water Density ( Kg/m3 ).
Ca = 2.29; % added mass coefficient "input" (as per potential flow theory
Sorenson et.al 1986)
Cm = Ca+1; % Inertia Coefficient.

imax = length(t);
L = 1; % Pipe/Cable Length (m).
Z0 = 0.6; % SeaBed Soil Roughness (N/m/m)."input"

gammas = 10000; % Submerged unit soil for sand normally range from 7000
N/m3 (very loose) to 13500 N/m3 (very dense)

%to be replaced with correct correction cofficients

Clc = 1; %lift force correction factor due to cable movement
Cdc = 1; %Drag force correction factor due to cable movement

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

D = 14; % Water Depth ( m )."input"
Hs = 16;% Surface Wave Significant Wave Height ( m ) (Empirical
data)."input"
Tp = 18.5; % Surface Wave Height Peak Period ( sec. ) (Empirical
data)."input"
d = 0.161; % Pipe/Cable Diameter ( m )."input"
V = 0.55; %Currunt Velocity at 1m Above Sea Bed [10 years return period]
(m/sec.)

%Before added mass

mass = 44.4; % Pipe/Cable Mass (kg/m). ''input''
Volcbefore = (pi./4).*(d.^2).*L; %CableVolume  (m3/m)
mswawaterbefore = Volcbefore.*Density; %mass of displaced seawater (Kg/m)
Fbuoyantbefore = mswawaterbefore.*g; %Buoyant force (N/m)
Wsbefore = (mass.*g) - Fbuoyantbefore; %Cable Submerged weight (N/m)
msbefore = Wsbefore./g; % Pipe/Cable Submerged Mass (kg/m).

%After added mass
```

```
percentage = 1350; % Added Chain Mass percentage from orginal mass.
mchain = mass .*(percentage./100); % Added Chain Mass (Kg/m)
ma = mass.*(1+(percentage./100)); % New Pipe/Cable Mass (kg/m).

Volc = (pi./4).*(d.^2).*L; %CableVolume  (m3/m)
mswawater = Volc.*Density; %mass of displaced seawater (Kg/m)
Fbuoyant = mswawater.*g; %Buoyant force (N/m)
Ws = (ma.*g) - Fbuoyant; %Cable Submerged weight (N/m)
ms = Ws./g; % Pipe/Cable Submerged Mass (kg/m).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %Peak Enhancement Factor.

TpHs = (Tp/(sqrt(Hs)));

if TpHs <= 3.6; Gamma = 5;
elseif 3.6 < TpHs && TpHs < 5; Gamma = 3.3;
elseif 5 <= TpHs; Gamma = 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x0 = 0; %cable inial horizontal postion
y0 = 0; %cable inial vertical postion, cable intial penetration due to
movement in first iteration is 0 (=Zpm)

ux0 = 0; %cable intial horizontal velocity
uy0 = 0; %cable intial vertical velocity

ax0 = 0; %cable intial horizontal acceleration
ay0 = 0; %cable intial vertical acceleration

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ss = [];
Su = [];
m = [];
ww = [];
ff = [];
phaseangle = [];
GG = [];
kk = [];
sigmaa = [];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

FHmabs = [];
Wss = [];

tt = [];
x = []; %Cable horizontal displacement Time History (m)
y = []; %Cable vertical displacement Time History (m)

zpi = [];
zpm =[];
zp = [];
```

```
ue = []; %effective wave partical velocity after firt iteration due to pipe
movement (m/sec.)
ux = []; %pipe horizontal velocity (m/sec)
ax = []; %pipe horizontal acceleration (m/sec2)
uy = []; %pipe vertical velocity (m/sec)
ay = []; %pipe vertical acceleration (m/sec2)

FIc = []; %hydrodynamic corretion inertia force due to movement (N/m)
FDc = []; %hydrodynamic corretion drag force due to movement (N/m)
FVc = []; %hydrodynamic corretion vertical force due to movement (N/m)

FDm = []; %hydrodynamic correted drag force due to movement (N/m)
FVm = []; %hydrodynamic correted vertical force due to movement (N/m)
FIm = []; %hydrodynamic correted inertia force due to movement (N/m)
FHm = []; %hydrodynamic correted total horizontal force due to movement
(N/m)

FHnet = []; %net horizontal force acting on cable (N/m)
FVnet = []; %net vertical force acting on cable (N/m)

Fpr = []; %Passive Resistance Force (N/m)
Fcl = []; % Colomb Firction Resistance Force (N/m)
FR = []; %total soil resistance force

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %Wave Elevation Power Density Spectrum "PSD" (JONSWAP), Velocity Power
Desntiy Spectrum, Transfer Function, Velocity & Acceleration Time History &
Random Phase Angle Generation

for f = 0.00000000000000001:nf:fmax;

    w = 2.*pi.*f; % Wave Radial Frequency ( rad/sec. ).
    wp = 2*pi/Tp; % Peak Radial Frequency ( rad/sec. ).

    k = w./(sqrt(g.*D));% Wave Number.
    G = w./(sinh((w.*(sqrt(D))./(sqrt(g))));% Transfer Function for 14m
Depth "Despeirson Relation".

    GG = [GG G];
    kk = [kk k];

    if w <= wp  ; sigma = 0.07; % Spectral width parameter.
    else              sigma = 0.09;
    end

    Sss = ((5.*(Hs.^2).*(wp.^4))./(16.*(w.^5))).*(1-
(0.287.*(log(Gamma)))).*(exp((-1.25).*((wp./w).^4))).*(Gamma.^(exp((-
0.5).*(((w-wp)./(sigma.*wp)).^2)))); %old equation modified on 10/4/2017
DNV VERSION
    Suu = ((5.*(Hs.^2).*(wp.^4).*(w.^(-
3)))./(16.*((sinh((w.*(sqrt(D)))./(sqrt(g)))).^2))).*(1-
(0.278.*(log(Gamma)))).*(exp((-1.25).*((wp./w).^4))).*(Gamma.^(exp((-
0.5).*(((w-wp)./(sigma.*wp)).^2)))); %old equation modified on 10/4/2017
DNV VERSION

    Ss = [Ss Sss]; % Wave Elevation Density Spectrum at Sea Surface (
m2/rad/sec. ).
```

```
    Su = [Su Suu]; % Velocity power spectrum at Pipeline/Cable Level  [14m
Depth] ( m2/rad/sec. ).
    ww = [ww w];
    ff = [ff f];
    sigmaa = [sigmaa sigma];

    Ab = sqrt(2.*Suu.*nf);
    phi = rand_extended(-pi,pi); % Random phase angle "white noise".

    etb = Ab.*(cos((2.*pi.*f.*t)+phi));
    eetb = u+etb;
    u = eetb; % sum of Velocity time history at Pipeline/Cable Level (
m/sec. ).

    atb = -2.*Ab.*f.*pi.*(sin(phi +(2.*pi.*f.*t)));
    aatb = a+atb;
    a = aatb; % sum of Acceleration time history at Pipeline/Cable Level (
m/sec.2 ).

    phaseangle = [phaseangle phi];

end

%nophaseangles = length(phaseangle);

%n-th Spectral Moments

for n = 0:4;

        Suuu = @(w) ((5.*(Hs.^2).*(wp.^4).*(w.^(n-
3))./(16.*((sinh((w.*(sqrt(D)))./(sqrt(g)))).^2))).*(1-
(0.278.*(log(Gamma)))).*(exp((-5./4).*((w./wp).^(-4)))).*(Gamma.^(exp((-
1./2).*(((w-wp)./(sigma.*wp)).^2))));%old equation modified on 10/4/2017

        moment = integral(Suuu,0,inf);

        m = [m moment];

end

m0 = m(1);
m1 = m(2);
m2 = m(3);
m3 = m(4);
m4 = m(5);

us = 2.*(sqrt(m0)); %Significant Flow Velocity Amplitude at Pipeline/Cable
Level.
tp = 2.*pi.*(sqrt(m2./m4)); %Average Period Between Successive Crests at
Pipeline/Cable Level [Peak Period].
tu = 2.*pi.*(sqrt(m0./m2)); %Mean Zero Up-Crossing Period of Oscillating
flow at Pipeline/Cable Level.
M = V/us; %Current to Wave Velocity Ratio
KC = (us*tp)/d; %Keulegan-Carpenter Number

 %Fourier Coefficients

if KC <= 10;
```

```
CD0 = (2.5469*(M^5))-(6.6661*(M^4))+(3.3485*(M^3))+(2.7547*(M^2))-
(0.5156*M)+0.0052;
CD1 = (-85.484*(M^6))+(350.71*(M^5))-(524.83*(M^4))+(352.68*(M^3))-
(103.04*(M^2))+(10.714*M)+1.2885;
CD2 = (-3.8522*(M^5))+(12.047*(M^4))-
(11.868*(M^3))+(4.0869*(M^2))+(0.2123*M)+0.0169;
CD3 = (-59.444*(M^6))+(236.3*(M^5))-(340.31*(M^4))+(218.9*(M^3))-
(61.259*(M^2))+(5.9688*M)+0.1253;
CD4 = (-24.542*(M^6))+(97.865*(M^5))-(140.62*(M^4))+(89.066*(M^3))-
(23.814*(M^2))+(2.1731*M)-0.001;
CD5 = (-7.9032*(M^6))+(31.744*(M^5))-(47.266*(M^4))+(33.096*(M^3))-
(11.053*(M^2))+(1.4396*M)+0.0219;
PHYD1 = (-293.15*(M^6))+(713.48*(M^5))-(157.19*(M^4))-
(590.96*(M^3))+(367.92*(M^2))-(16.33*M)+11.384;
PHYD2 = (4642.6*(M^6))-(21284*(M^5))+(36829*(M^4))-
(30153*(M^3))+(11994*(M^2))-(2247.5*M)+227.16;
PHYD3 = (103.51*(M^6))-(804.89*(M^5))+(1970.5*(M^4))-
(2015.5*(M^3))+(843.18*(M^2))-(47.672*M)-28.22;
PHYD4 = ((8.7234*(10^5))*(M^8))-
((4.7425*(10^6))*(M^7))+((1.0956*(10^7))*(M^6))-
((1.3986*(10^7))*(M^5))+((1.0736*(10^7)*(M^4))-
((5.0429*(10^6))*(M^3))+((1.4052*(10^6))*(M^2))-((2.1046*(10^5))*M))+12767;
PHYD5 = (1520.1*(M^6))-(6459.2*(M^5))+(10232*(M^4))-
(7694.5*(M^3))+(2888.1*(M^2))-(433.4*M)-23.189;

CL0 = (-64.161*(M^6))+(249.57*(M^5))-(350.46*(M^4))+(217.7*(M^3))-
(56.144*(M^2))+(4.6146*M)+2.4016;
CL1 = (-32.936*(M^6))+(114.59*(M^5))-
(129.19*(M^4))+(49.836*(M^3))+(4.9471*(M^2))-(1.6548*M)+0.043;
CL2 = (2.6872*(M^5))-(5.575*(M^4))-(0.1735*(M^3))+(4.066*(M^2))-
(0.9836*M)+1.8293;
CL3 = (-22.081*(M^6))+(95.127*(M^5))-(149.97*(M^4))+(105.33*(M^3))-
(31.64*(M^2))+(3.5259*M)-0.0031;
CL4 = (43.244*(M^6))-(148.9*(M^5))+(191.58*(M^4))-
(112.31*(M^3))+(28.885*(M^2))-(2.6455*M)+0.2826;
CL5 = (-23.077*(M^6))+(91.615*(M^5))-(129.89*(M^4))+(79.68*(M^3))-
(20.102*(M^2))+(1.7807*M)+0.0005;
PHYL1 = (2297.6*(M^6))-(10716*(M^5))+(18957*(M^4))-
(15883*(M^3))+(6323.3*(M^2))-(1014.5*M)+37.297;
PHYL2 = (697.42*(M^6))-(3102.4*(M^5))+(5097*(M^4))-
(3842.4*(M^3))+(1314.2*(M^2))-(146.49*M)-27.245;
PHYL3 = (-2672.2*(M^6))+(13664*(M^5))-(27004*(M^4))+(25868*(M^3))-
(12274*(M^2))+(2580.6*M)-230.54;
PHYL4 = (4169.7*(M^5))-(8992*(M^4))+(6352.5*(M^3))-
(1671.7*(M^2))+(147.39*M)+33.364;
PHYL5 = (-2648.4*(M^6))+(11771*(M^5))-(20134*(M^4))+(16908*(M^3))-
(7195.2*(M^2))+(1409.7*M)-71.083;

elseif 10 < KC && KC <= 15;

CD0 = (2.3062*(M^5))+(10.298*(M^4))-(16.724*(M^3))+(12.15*(M^2))-
(2.0828*M)+0.0777;
CD1 = ( -27.036*(M^6))+(110.37*(M^5))-(160.83*(M^4))+(99.748*(M^3))-
(23.94*(M^2))+(1.9305*M)+1.14409;
CD2 = (-0.77*(M^5))+(2.1338*(M^4))-(1.666*(M^3))+(0.2328*(M^2))+(0.548*M)-
0.0108;
```

```
CD3 = (-17.008*(M^6))+(70.215*(M^5))-(105.89*(M^4))+(71.748*(M^3))-
(21.873*(M^2))+(2.6356*M)+0.2072;
CD4 = abs((665.47*(M^8))-(3650.9*(M^7))+(8125.5*(M^6))-
(9443.3*(M^5))+(6125.3*(M^4))-(2183*(M^3))+(385.78*(M^2))-
(24.572*M)+(1.2093*(10^(-11))));%%%%%%%%%%%%%%
CD5 = abs((-146.66)*(M^8))+(753.1*(M^7))-(1529.2*(M^6))+(1559.4*(M^5))-
(829.28*(M^4))+(209.63*(M^3))-(15.932*((M^2))-
(1.2391*M)+0.18501);%%%%%%%%%%%%%%
PHYD1 = (495.03*(M^6))-(2110.1*(M^5))+(3327.2*(M^4))-
(2425*(M^3))+(829.96*(M^2))-(99.582*M)+5.9886;
PHYD2 = (5507.3*(M^6))-(25176*(M^5))+(43617*(M^4))-
(35864*(M^3))+(14423*(M^2))-(2750.8*M)+256.23;
PHYD3 = (546.99*(M^6))-(2231*(M^5))+(3325.6*(M^4))-
(2307.6*(M^3))+(781.29*(M^2))-(71.678*M)-28.267;
PHYD4 = (-11068*(M^6))+(52973*(M^5))-(97835*(M^4))+(88379*(M^3))-
(40793*(M^2))+(9050.1*M)-714.96;
PHYD5 = (596.23*(M^6))-(2842.5*(M^5))+(5188.9*(M^4))-
(4484.6*(M^3))+(1751.6*(M^2))-(220.09*M)+31.514;

CL0 = (-2.5645*(M^5))+(12.86*(M^4))-(22.074*(M^3))+(16.083*(M^2))-
(3.6822*M)+2.2136;
CL1 = (-8.4135*(M^6))+(18.938*(M^5))+(5.1042*(M^4))-
(36.391*(M^3))+(26.968*(M^2))-(4.0123*M)+0.004;
CL2 = (-27.33*(M^6))+(117.38*(M^5))-184.69*M^4+129.42*M^3-
39.217*M^2+4.5482*M+1.3836;
CL3 = abs((699.8*(M^8))-(4037.8*(M^7))+(9515.9*(M^6))-
(11794*(M^5))+(8225.3*(M^4))-(3191*(M^3))+(630.35*(M^2))-
(48.687*M)+0.3462);%%%%%%%%%%%%%%
CL4 = (-1.172*(M^5))+(3.4219*(M^4))-(2.6866*(M^3))+(0.2617*(M^2))-
(0.2048*M)+0.5123;
CL5 = (-11.685*(M^6))+(47.062*(M^5))-(67.961*(M^4))+(42.394*(M^3))-
(10.957*(M^2))+(1.1283*M)-0.0019;
PHYL1 = (4635.2*(M^6))-(21948*(M^5))+(39799*(M^4))-
(34579*(M^3))+(14584*(M^2))-(2586.6*M)+83.992;
PHYL2 = (728.25*(M^6))-(2963.1*(M^5))+(4395.8*(M^4))-
(2964*(M^3))+(943.68*(M^2))-(114.75*M)-33.662;
PHYL3 = (-2144.5*(M^6))+(11080*(M^5))-(23072*(M^4))+(24700*(M^3))-
(14290*(M^2))+(4235.9*M)-457.49;
PHYL4 = (-436.55*(M^6))+(1576.3*(M^5))-
(1836.1*(M^4))+(671.5*(M^3))+(1.829*(M^2))-(1.1744*M)+25.732;
PHYL5 = (-5216.4*(M^6))+(24566*(M^5))-(44842*(M^4))+(40467*(M^3))-
(19141*(M^2))+(4567.4*M)-431.32;

elseif 15 < KC && KC <= 20;

CD0 = (-22.951*(M^6))+(45.243*(M^5))-(23.308*(M^4))-
(4.5342*(M^3))+(7.5908*(M^2))-(0.9272*M)+0.0004;
CD1 = (-23.519*(M^6))+(92.528*(M^5))-(128.53*(M^4))+(75.332*(M^3))-
(17.017*(M^2))+(1.4558*M)+1.2811;
CD2 = (0.7545*(M^6))-(4.0341*(M^5))+(7.5028*(M^4))-
(5.9842*(M^3))+(2.1111*(M^2))+(0.0816*M)-0.0003;
CD3 = abs(((3.654*(10^5))*(M^10))-
((2.016*(10^6))*(M^9))+((4.8039*(10^6))*(M^8))-
((6.4794*(10^6))*(M^7))+((5.442*(10^6))*(M^6))-
((2.9454*(10^6))*(M^5))+((1.0248*(10^6))*(M^4))-
((2.2059*(10^5))*(M^3))+(26669*(M^2))-(1409.1*M)+7.0309);%%%%%%%%%%%%%%
```

```
CD4 = abs(((-1164.1)*(M^10))+(8527.1*(M^9))-(26350*(M^8))+(44835*(M^7))-
(45987*(M^6))+(29240*(M^5))-(11394*(M^4))+(2581.7*(M^3))-
(302.11*(M^2))+(14.021*M)+(6.3974*(10^-7)));%%%%%%%%%%%%%%
CD5 = abs(((-4401.7)*(M^10))+(30696*(M^9))-
(91714*(M^8))+((1.5385*(10^5))*(M^7))-
((1.5928*(10^5))*(M^6))+((1.0519*(10^5))*(M^5))-
(44116*(M^4))+(11271*(M^3))-(1583.8*(M^2))+(93.878*M)-
0.21476);%%%%%%%%%%%%%%
PHYD1 = ((-624.7)*(M^6))+(3196.9*(M^5))-(6359.4*(M^4))+(6235.5*(M^3))-
(3150.5*(M^2))+(785.07*M)-69.592;
PHYD2 = (4634.2*(M^6))-(20969*(M^5))+(35469*(M^4))-
(27982*(M^3))+(10559*(M^2))-(1894.7*M)+193.07;
PHYD3 = ((-471.91)*(M^6))+(2066.1*(M^5))-(3192.5*(M^4))+(2030*(M^3))-
(449.02*(M^2))+(51.369*M)-34.917;
PHYD4 = ((-6386.9)*(M^6))+(29784*(M^5))-(53396*(M^4))-
(21042*(M^2))+(4593.6*M)-354.44;
PHYD5 = (((-2.9392)*(10^6))*(M^10))+((1.941*(10^7))*(M^9))-
((5.4406*(10^7))*(M^8))+((8.4861*(10^7))*(M^7))-
((8.1005*(10^7))*(M^6))+((4.891*(10^7))*(M^5))-
((1.8594*(10^7))*(M^4))+((4.267*(10^6))*(M^3))-
((5.332*(10^5))*(M^2))+(27609*M)-12.84;

CL0 = (-6.4906*(M^5))+(25.813*(M^4))-(36.479*(M^3))+(22.834*(M^2))-
(5.0858*M)+1.9476;
CL1 = (19.027*(M^6))-(83.078*(M^5))+(132.83*(M^4))-
(96.172*(M^3))+(33.94*(M^2))-(3.8174*M)+0.0137;
CL2 = (-2.1898*(M^5))+(11.125*(M^4))-(19.506*(M^3))+(13.263*(M^2))-
(2.7771*M)+1.3355;
CL3 = (-13.181*(M^6))+(53.959*(M^5))-(78.212*(M^4))+(47.643*(M^3))-
(11.55*(M^2))+(1.5148*M)-0.0086;
CL4 = (1.8538*(M^5))-(7.8154*(M^4))+(11.571*(M^3))-
(6.711*(M^2))+(0.7214*M)+0.4733;
CL5 = (-2.339*(M^5))+(7.731*(M^4))-
(7.8541*(M^3))+(2.1956*(M^2))+(0.3159*M)-0.0116;
PHYL1 = (859.38*(M^6))-(3541.1*(M^5))+(4837*(M^4))-(1899*(M^3))-
(948.11*(M^2))+(843.73*M)-169.86;
PHYL2 = (-46.272*(M^6))+(275.65*(M^5))-(556.43*(M^4))+(427.67*(M^3))-
(77.119*(M^2))-(2.923*M)-34.91;
PHYL3 = (-4638.9*(M^6))+(22697*(M^5))-(43484*(M^4))+(41641*(M^3))-
(20963*(M^2))+(5299.1*M)-502.47;
PHYL4 = (1045.3*(M^6))-(4647.6*(M^5))+(7664.1*(M^4))-
(5777.5*(M^3))+(1906.8*(M^2))-(211.51*M)+24.89;
PHYL5 = (-1736.1*(M^6))+(7998.2*(M^5))-(13612*(M^4))+(10438*(M^3))-
(3415.8*(M^2))+(271.71*M)+37.006;

elseif 20 < KC && KC <= 30;

CD0 = (6.1568*(M^6))-(26.039*(M^5))+(41.332*(M^4))-
(31.215*(M^3))+(12.063*(M^2))-(0.9422*M)+0.0049;
CD1 = (-0.6898*(M^6))+(3.1349*(M^5))-(5.318*(M^4))+(45662*(M^3))-
(1.9509*(M^2))+(0.642*M)+1.2542;
CD2 = (1.0191*(M^6))-(4.622*(M^5))+(7.4148*(M^4))-
(5.2563*(M^3))+(1.7479*(M^2))+(0.1529*M)-0.0017;
CD3 = abs(((-1949.7)*(M^9))+(11314*(M^8))-(27141*(M^7))+(34954*(M^6))-
(26220*(M^5))+(11608*(M^4))-(2923.9*(M^3))+(378.26*(M^2))-
(19.235*M)+0.22984);%%%%%%%%%%%%%%
CD4 = abs(((-3913.9)*(M^10))+(27496*(M^9))-
(82279*(M^8))+((1.3698*(10^5))*(M^7))-
```

```
((1.3896*(10^5))*(M^6))+(88331*(M^5))-(34793*(M^4))+(8061.6*(M^3))-
(975.3*(M^2))+(45.887*M)-(6.4001*(10^(-10))));%%%%%%%%%%%%%%
CD5 = abs(((-1758.9)*(M^9))+(10188*(M^8))-(24346*(M^7))+(31162*(M^6))-
(23189*(M^5))+(10169*(M^4))-(2533.5*(M^3))+(323.62*(M^2))-
(16.651*M)+0.24207);%%%%%%%%%%%%%%
PHYD1 = (-66.944*(M^6))+(277.63*(M^5))-
(356.2*(M^4))+(87.047*(M^3))+(87.153*(M^2))-(15.119*M)-3.8888;
PHYD2 = (2834.2*(M^6))-(12582*(M^5))+(20707*(M^4))-
(15584*(M^3))+(5425.5*(M^2))-(936.1*M)+135.29;
PHYD3 = (-1332.1*(M^6))+(5100.3*(M^5))-(6923.8*(M^4))+(4015.9*(M^3))-
(929.33*(M^2))+(117.66*M)-42.706;
PHYD4 = ((-35399)*(M^8))+((1.8886*(10^5))*(M^7))-
((4.1476*(10^5))*(M^6))+((4.9101*(10^5))*(M^5))-
((3.4345*(10^5))*(M^4))+((1.4495*(10^5))*(M^3))-(35615*(M^2))+(4563.6*M)-
204.18;
PHYD5 = (((-7.2902)*(10^5))*(M^9))+((4.1942*(10^6))*(M^8))-
((9.9552*(10^6))*(M^7))+((1.2671*(10^7))*(M^6))-
((9.3967*(10^6))*(M^5))+((4.1235*(10^6))*(M^4))-
((1.0355*(10^6))*(M^3))+((1.3505*(10^5))*(M^2))-(7116.6*M)-13.641;

CL0 = ((-3.4527)*(M^6))+(16169*(M^5))-(26.695*(M^4))+(18.157*(M^3))-
(2.9662*(M^2))-(0.4877*M)+1.4399;
CL1 = ((-1.4864)*(M^6))+(14.109*(M^5))-(37.634*(M^4))+(38.571*(M^3))-
(12.48*(M^2))+(1.6006*M)+0.0017;
CL2 = ((-24.725)*(M^6))+(102.89*(M^5))-(155.48*(M^4))+(102.96*(M^3))-
(28.844*(M^2))+(3.1805*M)+1.8694;
CL3 = ((-12.854)*(M^5))+(48.018*(M^4))-(60.697*(M^3))+(28.308*(M^2))-
(2.5816*M)+0.0509;
CL4 = (1.6882*(M^5))-(6.4457*(M^4))+(8.0758*(M^3))-(3.2265*(M^2))-
(0.4015*M)+0.4433;
CL5 = abs((867.17*(M^9))-(5002.4*(M^8))+(11913*(M^7))-
(15193*(M^6))+(11224*(M^5))-(4832.3*(M^4))+(1151.8*(M^3))-
(134.03*(M^2))+(6.0733*M)-0.0079575);%%%%%%%%%%%%%%
PHYL1 = (3654.7*(M^6))-(1699*(M^5))+(29987*(M^4))-
(25648*(M^3))+(10851*(M^2))-(2036.9*M)+92.088;
PHYL2 = (29.325*(M^60))-(123.88*(M^5))+(275.02*(M^4))-
(408.45*(M^3))+(307.31*(M^2))-(54.816*M)-37.886;
PHYL3 = ((-4553.4)*(M^6))+(22393*(M^5))-(42888*(M^4))+(40533*(M^3))-
(19698*(M^2))+(4657.5*M)-391.79;
PHYL4 = ((4.2537*(10^5))*(M^9))-
((2.4392*(10^6))*(M^8))+((5.7561*(10^6))*(M^7))-
((7.2509*(10^6))*(M^6))+((5.2807*(10^6))*(M^5))-
((2.2465*(10^6))*(M^4))+((5.3582*(10^5))*(M^3))-
(64322*(M^2))+(2964.9*M)+8.3076;
PHYL5 = (3206.9*(M^6))-(14231*(M^5))+(24215*(M^4))-
(19997*(M^3))+(830.2*(M^2))-(1627.1*M)+108.58;

elseif 30 < KC && KC <= 40;

CD0 = (31.818*(M^6))-(113.67*(M^5))+(152.5*(M^4))-
(94.882*(M^3))+(27.451*(M^2))-(1.9097*M)-(1*(10^(-5)));
CD1 = (-2.5896*(M^6))+(10.219*(M^5))-
(13.174*(M^4))+(5.2554*(M^3))+(1.5643*(M^2))-(0.916*M)+1.2595;
CD2 = (16.239*(M^6))-(43.562*(M^5))+(37.158*(M^4))-(8.7452*(M^3))-
(1.5947*(M^2))+(0.8757*M)-0.0003;
CD3 = (-35.687*(M^6))+(124.01*(M^5))-(160.55*(M^4))+(94.924*(M^3))-
(25.519*(M^2))+(2.852*M)+0.0601;
```

```
CD4 = (-20.88*(M^5))+(55.01*(M^4))-(48.478*(M^3))+(15.491*(M^2))-
(0.9129*M)-0.0011;
CD5 = (267.98*(M^7))-(1083.3*(M^6))+(1731.8*(M^5))-
(1385.9*(M^4))+(576.28*(M^3))-(115.14*(M^2))+(8.1518*M)+ 0.11219;
PHYD1 = (-474.89*(M^6))+(1491.6*(M^5))-(1618.2*(M^4))+(656.66*(M^3))-
(57.983*(M^2))+(18.713*M)-9.4595;
PHYD2 = (25168*(M^6))-(97259*(M^5))+(146662*(M^4))-
(109125*(M^3))+(41785*(M^2))-(7864.5*M)+617.35;
PHYD3 = (3664.9*(M^6))-(11146*(M^5))+(12513*(M^4))-
(6527.9*(M^3))+(1581.2*(M^2))-(69.436*M)-45.475;
PHYD4 = (-58411*(M^6))+(214654*(M^5))-(307047*(M^4))+(217143*(M^3))-
(79663*(M^2))+(14359*M)-970.2;
PHYD5 = (37775*(M^6))-(138347*(M^5))+(194006*(M^4))-
(129418*(M^3))+(40748*(M^2))-(4858.1*M)+33.913;

CL0 = (-510.09*(M^6))+(1736.2*(M^5))-(2223.6*(M^4))+(1316*(M^3))-
(346.14*(M^2))+(29.684*M)+1.1449;
CL1 = (25.152*(M^6))-(76.835*(M^5))+(83.566*(M^4))-
(41.023*(M^3))+(12.915*(M^2))-(1.2792*M)-0.0001;
CL2 = (3.5556*(M^6))-(17.625*(M^5))+(32.3*(M^4))-
(27.251*(M^3))+(10.41*(M^2))-(1.4023*M)+0.8478;
CL3 = (-17.722*(M^5))+(50.384*(M^4))-(48.686*(M^3))+(17.173*(M^2))-
(0.8976*M)-0.0029;
CL4 = abs((9113.6*(M^8))-(37858*(M^7))+(64364*(M^6))-
(57904*(M^5))+(29664*(M^4))-(8610.9*(M^3))+(1305.1*(M^2))-
(80.997*M)+0.67945);%%%%%%%%%%%%%%
CL5 = (23.47*(M^6))-(79.211*(M^5))+(101.56*(M^4))-
(58.636*(M^3))+(13.929*(M^2))-(0.6516*M)-0.0003;
PHYL1 = (-8761.6*(M^5))+(29727*(M^4))-(38332*(M^3))+(23452*(M^2))-
(6812.8*M)+731.87;
PHYL2 = (-1214.1*(M^6))+(4140.1*(M^5))-(5226.5*(M^4))+(2912.4*(M^3))-
(654.92*(M^2))+(74.236*M)-42.81;
PHYL3 = (-17942*(M^6))+(68879*(M^5))-(103769*(M^4))+(77484*(M^3))-
(29693*(M^2))+(5448.6*M)-341.32;
PHYL4 = (13844*(M^6))-(45496*(M^5))+(55175*(M^4))-
(30099*(M^3))+(7039.5*(M^2))-(534.93*M)+16.142;
PHYL5 = (12501*(M^6))-(48589*(M^5))+(73699*(M^4))-
(55032*(M^3))+(20935*(M^2))-(3789.9*M)+240.5;

elseif 40<KC && KC<=50;

CD0 = (-87.671*(M^6))+(227.81*(M^5))-(215.62*(M^4))+(89.348*(M^3))-
(14.394*(M^2))+(1.3416*M)-0.0002;
CD1 = (30.113*(M^6))-(72.945*(M^5))+(61.144*(M^4))-
(20.05*(M^3))+(2351*(M^2))+(0.0563*M)-1.1644;
CD2 = (49.689*(M^6))-(133.6*(M^5))+(133.9*(M^4))-
(61.308*(M^3))+(12.28*(M^2))-(11977*M)+2.9985;
CD3 = (-158.65*(M^6))+(281.45*(M^5))-
(156.59*(M^4))+(21.937*(M^3))+(4.7956*(M^2))-(1.3609*M)+2.9627;
CD4 = abs((3288.8*(M^7))-(10217*(M^6))+(12498*(M^5))-
(7612.3*(M^4))+(2395.5*(M^3))-(362.15*(M^2))+(20.417*M)+(5.649*(10^(-
13))));%%%%%%%%%%%%%%
CD5 = (-84.073*(M^6))+(229.05*(M^5))-(233.96*(M^4))+(110.07*(M^3))-
(23.021*(M^2))+(1.3883*M)+0.1247;
PHYD1 = (-14248*(M^6))+(38822*(M^5))-(39577*(M^4))+(18587*(M^3))-
(3958.2*(M^2))+(319.44*M)-7.981;
PHYD2 = (56859*(M^6))-(180821*(M^5))+(226037*(M^4))-
(14087*(M^3))+(45976*(M^2))-(7645.5*M)+569.43;
```

```
PHYD3 = (-18109*(M^6))+(52408*(M^5))-(57131*(M^4))+(28879*(M^3))-
(6687.6*(M^2))+(650.09*M)-49.968;
PHYD4 = (-912.87*(M^6))-(2267.6*(M^5))+(7988.6*(M^4))-
(7292.4*(M^3))+(2903.3*(M^2))-(479.35*M)+39.967;
PHYD5 = (12449*(M^6))-(30341*(M^5))+(28502*(M^4))-
(13327*(M^3))+(3201.1*(M^2))-(226.49*M)-59.504;

CL0 = (26.585*(M^6))-(37.917*(M^5))-(0.6721*(M^4))+(20.219*(M^3))-
(6.8596*(M^2))+(0.3612*M)+1.0786;
CL1 = (-198*(M^6))+(556.44*(M^5))-(588.88*(M^4))+(288.05*(M^3))-
(60.895*(M^2))+(5.0385*M)+0.0009;
CL2 = (-5.6308*(M^6))+(68.914*(M^5))-(136.83*(M^4))+(102.12*(M^3))-
(31.412*(M^2))+(3.5162*M)+0.6833;
CL3 = abs((4162.2*(M^8))-(11725*(M^7))+(12071*(M^6))-
(5046.8*(M^5))+(226.89*(M^4))+(433.86*(M^3))-(113.34*(M^2))+(9.1587*M)-
(8.8789*(10^(-13))));%%%%%%%%%%%%%%
CL4 = (31.67*(M^5))-(72.571*(M^4))+(57.717*(M^3))-
(17.73*(M^2))+(1.069*M)+0.3194;
CL5 = (-49.798*(M^5))+(97.246*(M^4))-(62.941*(M^3))+(14.027*(M^2))-
(0.2765*M)+0.0003;
PHYL1 = (-8431.5*(M^6))+(27467*(M^5))-(36055*(M^4))+(24741*(M^3))-
(9616*(M^2))+(2091.9*M)-225.81;
PHYL2 = (-5873.2*(M^6))+(17014*(M^5))-(18533*(M^4))+(9240.4*(M^3))-
(2020.7*(M^2))+(178.61*M)-37.769;
PHYL3 = (2522.8*(M^6))-(7441.4*(M^5))+(9111.9*(M^4))-
(6319.2*(M^3))+(2738.3*(M^2))-(630.95*M)+81.456;
PHYL4 = (40214*(M^6))-(119780*(M^5))+(134737*(M^5))+(134737*(M^4))-
(70199*(M^3))+(16395*(M^2))-(1346.7*M)+24.873;
PHYL5 = (59083*(M^6))-(189908*(M^5))+(238986*(M^4))-
(148709*(M^3))+(47531*(M^2))-(7353*M)+416.06;

elseif 50 < KC && KC <= 60;

CD0 = (-1.8657*(M^3))+(2.7593*(M^2))+(0.0785*M)-0.0043;
CD1 = (78.99*(M^5))-(167*(M^4))+(125.3*(M^3))-
(37.912*(M^2))+(3.6712*M)+1.1556;
CD2 = (-3.8957*(M^5))+(2.6823*(M^4))+(2.249*(M^3))-
(1.1871*(M^2))+(0.2785*M)-(3*(10^(-5)));
CD3 = (60.139*(M^5))-(121.29*(M^4))+(85.077*(M^3))-
(24.438*(M^2))+(2.7897*M)+0.0038;
CD4 = abs((54.346*(M^6))+(140.75*(M^4))-(131.88*(M^3))+(42.241*(M^2))-
(3.9915*M)+(5.8016*(10^(-15))));%%%%%%%%%%%%%%
CD5 = abs((64.132*(M^6))+(153.71*(M^4))+(139.51*(M^3))-
(44.557*(M^2))+(4.6312*M)+0.072433);%%%%%%%%%%%%%%
PHYD1 = (211.67*(M^5))-(289.25*(M^4))-(64.359*(M^3))+(172.69*(M^2))-
(24.191*M)-8.9551;
PHYD2 = (-593.19*(M^4))+(720.46*(M^3))+(116.44*(M^2))-(439.41*M)+139.62;
PHYD3 = (-3238.3*(M^5))+(8648.7*(M^4))-(8086.8*(M^3))+(3056.5*(M^2))-
(311.5*M)-42.431;
PHYD4 = (-2577.6*(M^4))+(5292.9*(M^3))-(3727.7*(M^2))+(1122.6*M)-107.31;
PHYD5 = (-3871.9*(M^5))+(9985.8*(M^4))-(9491.8*(M^3))+(3813.7*(M^2))-
(460.61*M)-56.045;

CL0 = (81.376*(M^5))-(153.77*(M^4))+(102.83*(M^3))-
(26.537*(M^2))+(2.0689*M)+0.9885;
CL1 = (-1.5041*(M^3))+(4.6154*(M^2))-(0.3948*M)-0.0095;
CL2 = (11.779*(M^5))-(42.376*(M^4))+(40.571*(M^3))-
(13.123*(M^2))+(12456*M)+0.6781;
```

```
CL3 = abs(((-68)*(M^6))+((1.9*(10^2))*(M^4))-
((1.8*(10^2))*(M^3))+(62*(M^2))-(6.1*M)-(2.5*(10^(-14))));%%%%%%%%%%%%%%
CL4 = (133.43*(M^5))-(293.98*(M^4))+(231.97*(M^3))-
(76.743*(M^2))+(8.6387*M)+0.19;
CL5 = (-35.21*(M^5))+(65.572*(M^4))-(40.513*(M^3))+(8.7167*(M^2))-
(0.1604*M)-(1*(10^(-11)));
PHYL1 = (-3124.8*(M^4))+(6081.1*(M^3))-(4305.7*(M^2))+(1340*M)-177.93;
PHYL2 = (2270.3*(M^5))-(4086.5*(M^4))+(2347.5*(M^3))-
(404.54*(M^2))+(18.27*M)-35.082;
PHYL3 = (903603*(M^5))-((2*(10^6))*(M^4))+((2*(10^6))*(M^3))-
(916406*(M^2))+(194103*M)-15543;
PHYL4 = (-27089*(M^5))+(57311*(M^4))-(42504*(M^3))+(12784*(M^2))-
(1320.4*M)+22.019;
PHYL5 = (314.97*(M^4))+(37.721*(M^3))-(559.21*(M^2))+(285.65*M)-53.479;

else

CD0 = (3.0868*(M^6))+(107.38*(M^5))-(150.72*(M^4))+(63.15*(M^3))-
(6.5127*(M^2))+(0.4477*M)-0.0001;
CD1 = (-10.977*(M^4))+(12.76*(M^3))-(3.7683*(M^2))+(0.1967*M)+1.0882;
CD2 = (-107.76*(M^6))+(131.44*(M^5))-(58.185*(M^4))+(16.774*(M^3))-
(3.4535*(M^2))+0.4629*M+3*10^-5;
CD3 = (-83.814*(M^5))+(90.003*(M^4))-
(25.727*(M^3))+(1.0075*(M^2))+(0.4515*M)+0.0357;
CD4 = abs(((-53965)*(M^7))+((1.0066*(10^5))*(M^6))-
(72614*(M^5))+(25721*(M^4))-(4672.1*(M^3))+(406.03*(M^2))-
(12.381*M)+(4.585*(10^(-13))));%%%%%%%%%%%%%%
CD5 = (603.5*(M^6))-(762.92*(M^5))+(344.64*(M^4))-
(61.011*(M^3))+(2.1814*(M^2))-(0.0267*M)+0.1012;
PHYD1 = (78982*(M^6))-(136715*(M^5))+(87611*(M^4))-
(25642*(M^3))+(3375.5*(M^2))-(138.4*M)-10.855;
PHYD2 = ((7*(10^6))*(M^6))-((1*(10^7))*(M^5))+((9*(10^6))*(M^4))-
((3*(10^6))*(M^3))+(560857*(M^2))-(49111*M)+1679;
PHYD3 = (574050*(M^6))-(937681*(M^5))+(561940*(M^4))-
(151743*(M^3))+(18159*(M^2))-(723.52*M)-29.679;
PHYD4 = (451364*(M^6))-(772154*(M^5))+(501425*(M^4))-
(156527*(M^3))+(24631*(M^2))-(1832.1*M)+61.283;
PHYD5 = (-986.97*(M^6))+(7605.6*(M^5))-(11958*(M^4))+(6598.6*(M^3))-
(1214.8*(M^2))+(109.14*M)+9.2713;

CL0 = (191.12*(M^6))-(368.34*(M^5))+(235.88*(M^4))-
(57.525*(M^3))+(5.097*(M^2))-(0.1406*M)+0.8635;
CL1 = (1762.5*(M^6))-(2810.9*(M^5))+(1640.7*(M^4))-
(418.8*(M^3))+(48.404*(M^2))-(1.6158*M)-(6*(10^(-5)));
CL2 = abs(((3.5341*(10^5))*(M^7))-
((6.5244*(10^5))*(M^6))+((4.7021*(10^5))*(M^5))-
((1.6937*(10^5))*(M^4))+(32317*(M^3))-(3145.9*(M^2))+(131.29*M)-
0.55438);%%%%%%%%%%%%%%
CL3 = (86.723*(M^5))-(123.21*(M^4))+(57.281*(M^3))-
(10.534*(M^2))+(1.3933*M)-0.001;
CL4 = abs((-1942.4*(M^6))+(3027*(M^5))-(1712.2*(M^4))+(438.82*(M^3))-
(52.427*(M^2))+(2.1768*M)+0.2213);%%%%%%%%%%%%%%
CL5 = abs(((-30991)*(M^7))+(60758*(M^6))-(46749*(M^5))+(18024*(M^4))-
(3655.7*(M^3))+(364.27*(M^2))-(13.011*M)-(1.4494*(10^(-
12))));%%%%%%%%%%%%%%
PHYL1 = (255724*(M^6))-(550236*(M^5))+(466714*(M^4))-
(19694*(M^3))+(42461*(M^2))-(4137.9*M)+83.736;
```

```
PHYL2 = ((-4.0588*(10^6))*(M^7))+((8.043*(10^6))*(M^6))-
((6.2858*(10^6))*(M^5))+((2.4718*(10^6))*(M^4))-
((5.1433*(10^5))*(M^3))+(53769*(M^2))-(2264.7*M)-21.073;
PHYL3 = ((-2.0325*(10^7))*(M^7))+((4.424*(10^7))*(M^6))-
((3.9097*(10^7))*(M^5))+((1.8157*(10^7))*(M^4))-
((4.7785*(10^6))*(M^3))+((7.1114*(10^5))*(M^2))-(55279*M)+1753;
PHYL4 = (189268*(M^6))-(290573*(M^5))+(161047*(M^4))-
(40901*(M^3))+(4828.6*(M^2))-(205.03*M)+19.217;
PHYL5 = ((1*(10^6))*(M^6))-((2*(10^6))*(M^5))+((1*(10^6))*(M^4))-
(504506*(M^3))+(89881*(M^2))-(7810*M)+240.9;

end

%Fourier Vertical & Horizontal Forces Equations

FDf = (0.5).*(Density).*(u.^2).*(d).*((CD0+(CD1.*(cos(((1.*ww.*t)-
PHYD1))))+(CD2.*(cos(((2.*ww.*t)-PHYD2))))+(CD3.*(cos(((3.*ww.*t)-
PHYD3))))+(CD4.*(cos(((4.*ww.*t)-PHYD4))))+(CD5.*(cos(((5.*ww.*t)-
PHYD5)))))); % Fourier Horizontal DRAG Force Analysis ( N/m ).
FI = (pi./4).*(Cm).*(d.^2).*(Density).*(a); % Horizontal INERTIA Force
Analysis ( N/m ).
FHf = FDf+FI; % Fourier Total Horizontal Force Analysis ( N/m ).
FLf = (((0.5).*(Density).*(u.^2).*(d).*((CL0+(CL1.*(cos(((1.*ww.*t)-
PHYL1))))+(CL2.*(cos(((2.*ww.*t)-PHYL2))))+(CL3.*(cos(((3.*ww.*t)-
PHYL3))))+(CL4.*(cos(((4.*ww.*t)-PHYL4))))+(CL5.*(cos(((5.*ww.*t)-
PHYL5)))))))); % Fourier Total Vertical Force Analysis Including Pipe/Cable
Submerged Weight ( N/m ).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Cable-Soil Interaction Model + Force Correction Model

for    i=1:imax;

    uee =
((((abs(FDf(i))./(0.5.*Density.*d))).^(0.5)).*((abs(FDf(i))./FDf(i)); %
Effective wave particel velocity at cable level (m/sec.)

    if i == 1, Fddc = 0;
     Fvvc = 0;
     Fiic = 0;
    else Fddc = 0.5.*Density.*d.*Cdc.*((uee.*(abs(uee)))-((uee-
ux0).*(abs(uee-ux0)))); %Drag correction force (N/m)
      Fvvc = 0.5.*Density.*d.*Clc.*((2.*uee.*ux0)-(ux0.^2)); %Vertical
correction force (N/m)
      Fiic = Ca.*(pi./4).*(d.^2).*ax0; %Inertia correction force (N/m)
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %force correction model
if i == 1, Fddm = 0;
else if y0 >= 0;    if Fddc == 0, Fddm = FDf(i);
                elseif Fddc > 0, if Fddc < FDf(i), Fddm = FDf(i) - Fddc;
                      elseif FDf(i) <= Fddc, Fddm = 0;
                                             end
                       elseif Fddc < 0, if FDf(i)  == 0 , Fddm = 0;
```

```
                    elseif FDf(i) > 0, if FDf(i) > (abs(Fddc)), Fddm =
FDf(i) +Fddc;
elseif FDf(i) <= (abs(Fddc)), Fddm = 0;
end
                    elseif FDf(i) < 0, if (abs(FDf(i))) <= (abs(Fddc)),
Fddm = 0;
elseif (abs(FDf(i))) > (abs(Fddc)), Fddm = FDf(i) - Fddc;

end
   end
      end
elseif y0 < 0, FDfi = (1-((d-(abs(y0)))/d)).*FDf(i);
                     if Fddc == 0, Fddm = FDfi;
              elseif Fddc > 0, if Fddc < FDfi, Fddm = FDfi - Fddc;
              elseif FDfi <= Fddc, Fddm = 0;
                                  end
              elseif Fddc < 0, if FDfi  == 0 , Fddm = 0;
              elseif FDfi > 0, if FDfi > (abs(Fddc)), Fddm = FDfi +Fddc;
              elseif FDfi <= (abs(Fddc)), Fddm = 0;
end
              elseif FDfi < 0, if (abs(FDfi)) <= (abs(Fddc)), Fddm = 0;
elseif (abs(FDfi)) > (abs(Fddc)), Fddm = FDfi - Fddc;

end
   end
      end
         end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if i == 1, Fvvm = 0;
else if y0 >= 0;    if Fvvc == 0, Fvvm = FLf(i);
                     elseif Fvvc > 0, if Fvvc < FLf(i), Fvvm = FLf(i) -
Fvvc;
                     elseif FLf(i) <= Fvvc, Fvvm = 0;
                      end
                      elseif Fvvc < 0, if FLf(i)  == 0 , Fvvm = 0;
                      elseif FLf(i) > 0, if FLf(i) > (abs(Fvvc)), Fvvm =
FLf(i) +Fvvc;
elseif FLf(i) <= (abs(Fvvc)), Fvvm = 0;
end
                      elseif FLf(i) < 0, if (abs(FLf(i))) <= (abs(Fvvc)),
Fvvm = 0;
elseif (abs(FLf(i))) > (abs(Fvvc)), Fvvm = FLf(i) - Fvvc;
end
   end
      end
elseif y0 < 0, FLfi = (1-((d-(abs(y0)))/d)).*FLf(i);
           if Fvvc == 0, Fvvm = FLfi;
               elseif Fvvc > 0, if Fvvc < FLfi, Fvvm = FLfi - Fvvc;
               elseif FLfi <= Fvvc, Fvvm = 0;
                end
               elseif Fvvc < 0, if FLfi  == 0 , Fvvm = 0;
               elseif FLfi > 0, if FLfi > (abs(Fvvc)), Fvvm = FLfi +Fvvc;
elseif FLfi <= (abs(Fvvc)), Fvvm = 0;
end
                elseif FLfi < 0, if (abs(FLfi)) <= (abs(Fvvc)), Fvvm = 0;

elseif (abs(FLfi)) > (abs(Fvvc)), Fvvm = FLfi - Fvvc;
```

```
end
   end
      end
         end
end
%%%%%%%%%%%%%%%%%%%%%%%%
if i == 1, Fiim = 0;
else if y0 >= 0;
if Fiic == 0, Fiim = FI(i);
elseif Fiic > 0, if Fiic < FI(i), Fiim = FI(i) - Fiic;
elseif FI(i) <= Fiic, Fiim = 0;
                      end
elseif Fiic < 0, if FI(i)  == 0 , Fiim = 0;
elseif FI(i) > 0, if FI(i) > (abs(Fiic)), Fiim = FI(i) +Fiic;
elseif FI(i) <= (abs(Fiic)), Fiim = 0;
end
elseif FI(i) < 0, if (abs(FI(i))) <= (abs(Fiic)), Fiim = 0;
elseif (abs(FI(i))) > (abs(Fiic)), Fiim = FI(i) - Fiic;
end
   end
      end
elseif y0 < 0, FIi= (1-((d-(abs(y0)))/d)).*FI(i);
if Fiic == 0, Fiim = FIi;
elseif Fiic > 0, if Fiic < FIi, Fiim = FIi - Fiic;
elseif FIi <= Fiic, Fiim = 0;
       end
elseif Fiic < 0, if FIi  == 0 , Fiim = 0;
elseif FIi > 0, if FIi > (abs(Fiic)), Fiim = FIi +Fiic;
elseif FIi <= (abs(Fiic)), Fiim = 0;
end
elseif FIi < 0, if (abs(FIi)) <= (abs(Fiic)), Fiim = 0;
elseif (abs(FIi)) > (abs(Fiic)), Fiim = FIi - Fiic;

end
   end
      end
         end
end

Fhhm = Fiim + Fddm; % Total Moving horizontal force after correction (N/m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Soil Resistance
Fc = Ws - Fvvm;
Ks = gammas.*(d.^2)./Fc;
zpii = 0.037.*((abs(Ks)).^(-0.67)).*((abs(Ks))/Ks).*d; % Inital penetration
(m)
zpmm = -y0; % Penetration due to movement (m)
zpp = zpii+zpmm; % Tota penetration (m)
if zpp <= 0 , zpp = 0;
else zpp = zpii+zpmm;
end
if y0 > 0, Frr = 0;
Fcll = 0;
Fppr = 0;
else Fcll = Z0.*Fc; %Coulomb friction force (N/m)
```

```
if Ks <= 26.7 , Fppr = ((5.*Ks) - (0.15.*(Ks.^2))).*((zpp./d).^1.25).*Fc;
%Passive resisitance force (N/m)
else Fppr = (Ks.^2).*(((abs(zpp))./d).^1.25).*((abs(zpp))/zpp).*Fc;
end
Frr = Fcll+Fppr; %Total resisitance force (N/m)
if Frr < 0, Frr = abs(Frr);
else Frr = Frr;
end
  end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %Horizontal motion analysis

if Fhhm == 0, if y0 > 0, Fhhnet = 0;
    axx = 0;
ux1 = 0;
x1 = x0;
  elseif y0 <= 0, if abs(y0) >= d, Fhhnet = 0;
axx = 0;
ux1 = 0;
x1 = x0;
 else if ux0 == 0, Fhhnet = 0;
axx =0;
ux1 = 0;
x1 = x0;
  elseif ux0 > 0, if i ==1, Fhhnet = -Frr;
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
  elseif i > 1, if (Fhhm(i-1)) > Frr, Fhhnet = (Fhhm(i-1))-Frr;

axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
   elseif (Fhhm(i-1)) <= Frr, Fhhnet = ((Fhhm(i-1))-Frr)*(1-((Frr-(Fhhm(i-
1)))/Frr));
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;

end
    end
   elseif ux0 < 0, if i ==1, Fhhnet = Frr;

axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
   elseif i > 1, if abs(Fhhm(i-1)) > Frr, Fhhnet = (Fhhm(i-1))+Frr;

axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;

elseif abs(Fhhm(i-1)) <= Frr, Fhhnet = ((Fhhm(i-1))+Frr)*(1-((Frr+(Fhhm(i-
1)))/Frr));
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
```

```
x1 = (ux1.*nt)+x0;
        end
  end
    end
       end
           end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif Fhhm > 0, if y0 > 0, Fhhnet = Fhhm;
    axx = Fhhnet./ms;
    ux1 = (axx.*nt)+ux0;
    x1 = (ux1.*nt)+x0;
   elseif y0 <= 0, if abs(y0) >= d, Fhhnet = 0;
    axx = 0;
     ux1 = 0;
       x1 = x0;
         else if Fhhm > Frr, Fhhnet = Fhhm-Frr;
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
   else if ux0 == 0, Fhhnet = 0;

axx = 0;
ux1 = 0;
x1 = x0;
     elseif ux0 > 0, Fhhnet = (Fhhm-Frr)*(1-((Frr-Fhhm)/Frr));
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
     elseif ux0 < 0, Fhhnet = (-(Fhhm-Frr))*(1-((Frr-Fhhm)/Frr));
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
                end
                  end
                    end
                       end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif Fhhm < 0, if y0 > 0, Fhhnet = Fhhm;
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
              elseif y0 <= 0, if abs(y0) >= d, Fhhnet = 0;
axx = 0;
ux1 = 0;
x1 = x0;
           else if (abs(Fhhm)) > Frr, Fhhnet = Fhhm+Frr;
                                                                    axx
= Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
               else if ux0 == 0, Fhhnet = 0;

axx = 0;
ux1 = 0;
x1 = x0;
              elseif ux0 > 0, Fhhnet = ((abs(Fhhm))-Frr)*(1-
((Frr+Fhhm)/Frr));
axx = Fhhnet./ms;
```

```
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
              elseif ux0 < 0, Fhhnet = (-((abs(Fhhm))-Frr))*(1-
((Frr+Fhhm)/Frr));
axx = Fhhnet./ms;
ux1 = (axx.*nt)+ux0;
x1 = (ux1.*nt)+x0;
                           end
                            end
                              end
                                end
                                  end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %Vertical motion analysis

if Fvvm == 0, if y0 > 0, Fvvnet = -Ws+Frr;
ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = 0;
                 elseif y0 <= 0, Fvvnet = 0;
ayy = 0;
uy1 = 0;
y1 = y0;
                       end
                      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif Fvvm > 0, if y0 == 0, if Fvvm > Ws, Fvvnet = Fvvm-Ws;
ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;
       elseif Fvvm <= Ws, Fvvnet = 0;

ayy = 0;
uy1 = 0;
y1 = y0;
                  end
              elseif y0 > 0, if Fvvm > Ws, Fvvnet = Fvvm-Ws;

ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;
              elseif Fvvm == Ws, Fvvnet = Fvvm-Ws;
ayy = 0;
uy1 = 0;
y1 = y0;
               elseif Fvvm < Ws, if (Fvvm-Ws) <= Frr, Fvvnet = (Fvvm-
Ws)+Frr;
ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = 0;

elseif (Fvvm-Ws) > Frr, Fvvnet = (Fvvm-Ws)+Frr;
ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;
                          end
                            end
```

```
      elseif y0 < 0, if (abs(y0)) >= d,  Fvvnet = 0;

ayy = 0;
uy1 = 0;
y1 = y0;
                 else if Fvvm > Frr, if (Fvvm-Frr) > Ws, Fvvnet = (Fvvm-
Frr)-Ws;
ayy = Fvvnet./ms;

uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;
elseif (Fvvm-Frr) <= Ws, Fvvnet = (Fvvm-Frr)-Ws;
ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = 0;
end
                   elseif Fvvm <= Frr, if uy0 == 0, Fvvnet = 0;
ayy = 0;
uy1 = 0;
y1 = y0;

      elseif uy0 > 0, Fvvnet = Fvvm-Frr;
ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;

      elseif uy0 < 0, Fvvnet = -(Fvvm-Frr);

ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;

      end
           end
               end
                  end
%%%%%%%%%%%%%%%%%%%%
elseif Fvvm < 0, if y0 > 0, if (abs(Fvvm-Ws)) > Frr, Fvvnet = (Fvvm-
Ws)+Frr;

ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;
                   elseif (abs(Fvvm-Ws)) <= Frr, Fvvnet = (Fvvm-Ws)+Frr;

ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = 0;
              end
                   elseif y0 <= 0, if (abs(y0)) >= d,  Fvvnet = 0;

ayy = 0;
uy1 = 0;
y1 = y0;
                 else if abs(Fvvm) > Frr, Fvvnet = Fvvm+Frr;

ayy = Fvvnet./ms;
```

```
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;
    elseif abs(Fvvm) <= Frr, if uy0 == 0, Fvvnet = 0;

ayy = 0;
uy1 = 0;
y1 = y0;
  elseif uy0 > 0, Fvvnet = -(Fvvm+Frr);

ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;

 elseif uy0 < 0, Fvvnet = Fvvm+Frr;

ayy = Fvvnet./ms;
uy1 = (ayy.*nt)+uy0;
y1 = (uy1.*nt)+y0;

               end
                   end
                      end
                          end
                             end

if y1 > D, y1 = D;
else y1 = y1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xx = x1; %accumelated horizontal displacement (m)
    yy = y1; %accumelated vertical displacement (m)

    tt = [tt t(i)];
    x = [x x1]; %Cable horizontal displacement Time History (m)
    y = [y y1]; %Cable vertical displacement Time History (m)

    zpi = [zpi zpii];
    zpm =[zpm zpmm];
    zp = [zp zpp];

    ue = [ue uee];
    ux = [ux ux1];
    ax = [ax axx];
    uy = [uy uy1];
    ay = [ay ayy];

    FIc = [FIc Fiic];
    FDc = [FDc Fddc];
    FVc = [FVc Fvvc];

    FDm = [FDm Fddm];
    FVm = [FVm Fvvm];
    FIm = [FIm Fiim];
    FHm = [FHm Fhhm];

    FHnet = [FHnet Fhhnet];
```

```
    FVnet = [FVnet Fvvnet];
    Wss = [Wss Ws];

    Fpr = [Fpr Fppr];
    Fcl = [Fcl Fcll];
    FR = [FR Frr];

    x0 = x1;
    y0 = y1;

    ux0 = ux1;
    uy0 = uy1;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %Plotting & Representation of Data & Results

figure(1)
subplot(1,2,1); plot(ww,Ss,ww,Su)
xlabel('W (rad/sec.)');
ylabel('S(w) (m2.sec)');
grid on;
legend('Wave Elevation Spectum at Sea Surface (PSD)',sprintf('Velocity
Power Spectrum at Cable Level [% 3.0f m Depth ]',D));
title('JONSWAP Spectrum');

subplot(1,2,2); plot(ww,GG)
xlabel('W (rad/sec.)');
ylabel('G(w)');
grid on;
title(sprintf('Transfer Function for Depth (D) = % 3.0f m',D));

figure(2)
subplot(2,1,1); plot(t,u,'k',t,ue,'g');
xlabel('t (sec.)');
ylabel('Wave Partical Velocity (m/sec)');
grid on;
title(sprintf('Wave Partical Velocities Time History at Cable Level for %
3.0f sec. Storm Duration',tmax));
legend('Wave Partical Velocity (m/sec)','Wave Partical Effective Velocity
Due-to Cable Movement (m/sec)');

subplot(2,1,2); plot(t,a,'r');
xlabel('t (sec.)');
ylabel('Wave Partical Acceleration (m/sec.2)');
title(sprintf('Wave Partical Acceleration Time History at Cable Level for %
3.0f sec. Storm Duration',tmax));

figure(3)
plot(ff,phaseangle,'r');
xlabel('F (Hz)');
ylabel('Phase Angle');
grid on;
title(sprintf('Random Phase Angle (% 3.0f Element )',imax));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure (4)
subplot(2,1,1); plot(t,FDc,'r',t,FDf,'b')
legend('Correction Due To Movement','During Fixed Analysis');
xlabel('t (sec.)');
ylabel('FD (N/m)');
grid on;
title(sprintf('Drag Force Time Histroy for % 3.0f sec. Storm
Duration',tmax));

subplot(2,1,2); plot(t,FDm,'k')
xlabel('t (sec.)');
ylabel('FD (N/m)');
grid on;
title(sprintf('Corrected Drag Force Time Histroy for % 3.0f sec. Storm
Duration',tmax));

figure (5)
subplot(2,1,1); plot(t,FVc,'r',t,FLf,'b')
legend('Correction Due To Movement','During Fixed Analysis');
xlabel('t (sec.)');
ylabel('FL (N/m)');
grid on;
title(sprintf('Lift Force Time Histroy for % 3.0f sec. Storm
Duration',tmax));

subplot(2,1,2); plot(t,FVm,'k')
xlabel('t (sec.)');
ylabel('FL (N/m)');
grid on;
title(sprintf('Corrected Lift Force Time Histroy for % 3.0f sec. Storm
Duration',tmax));

figure (6)
subplot(2,1,1); plot(t,FIc,'r',t,FI,'b')
legend('Correction Due To Movement','During Fixed Analysis');
xlabel('t (sec.)');
ylabel('FI (N/m)');
grid on;
title(sprintf('Inertia Force Time Histroy for % 3.0f sec. Storm
Duration',tmax));

subplot(2,1,2); plot(t,FIm,'k')
xlabel('t (sec.)');
ylabel('FI (N/m)');
grid on;
title(sprintf('Corrected Inertia Force Time Histroy for % 3.0f sec. Storm
Duration',tmax));

figure (7)
subplot(2,1,1); plot(t,FHm,'r',t,FR,'b',t,(-FR),'b');
legend('Total Horizontal Force (Corrected Drag Force + Corrected Inertia
Force)','Soil Resistance Force');
xlabel('t (sec.)');
ylabel('Horizontal Forces (N/m)');
grid on;
```

```
title(sprintf('Horizontal Force Time Histroy Analysis for % 3.0f sec. Storm
Duration',tmax));

subplot(2,1,2); plot(t,FHnet,'k');
xlabel('t (sec.)');
ylabel('Horizontal Forces (N/m)');
grid on;
title(sprintf('Net Horizontal Force Time Histroy Analysis for % 3.0f sec.
Storm Duration',tmax));

figure (8)
subplot(2,1,1); plot(t,FVm,'r',t,Wss,'b',t,(-FR),'g');
legend('Lift Force','Submerged Weight','Soil Resistance Force Due to
Penetration');
xlabel('t (sec.)');
ylabel('Vertical Forces (N/m)');
grid on;
title(sprintf('Vertical Force Time Histroy Analysis for % 3.0f sec. Storm
Duration',tmax));

subplot(2,1,2); plot(t,FVnet,'k');
xlabel('t (sec.)');
ylabel('Vertical Forces (N/m)');
grid on;
title(sprintf('Net Vertical Force Time Histroy Analysis for % 3.0f sec.
Storm Duration',tmax));

figure (9)
subplot(2,1,1); plot(t,ux,'r');
xlabel('t (sec.)');
ylabel('Cable Horizontal Velocity (m/sec.)');
grid on;
title(sprintf('Cable Horizontal Velocity After % 3.0f sec.',tmax));

subplot(2,1,2); plot(t,ax,'b');
xlabel('t (sec.)');
ylabel('Cable Horizontal Acceleration (m/sec.2)');
grid on;
title(sprintf('Cable Horizontal Acceleration After % 3.0f sec.',tmax));

figure (10)
subplot(2,1,1); plot(t,uy,'r');
xlabel('t (sec.)');
ylabel('Cable Vertical Velocity (m/sec.)');
grid on;
title(sprintf('Cable Vertical Velocity After % 3.0f sec.',tmax));

subplot(2,1,2); plot(t,ay,'b');
xlabel('t (sec.)');
ylabel('Cable Vertical Acceleration (m/sec.2)');
grid on;
title(sprintf('Cable Vertical Acceleration After % 3.0f sec.',tmax));

figure (11)
plot(t,x);
xlabel(' t (sec.) ');
ylabel(' X (m) ');
grid on;
```

```
title(fprintf('Horizontal Displacement after % 3.0f sec.',tmax));

figure (12)
plot(t,y);
xlabel('t (sec.)');
ylabel(' Y (m)');
grid on;
title(fprintf('Vertical Displacement after % 3.0f sec.',tmax));

figure (13)
plot(t,zpi ,'k-',t,zpm,'r-',t,zp,'b-');
legend('Intial Penetration (m)','Penetration Due-to Movement (m)','Total
Penetration (m)');
xlabel('t (sec.)');
ylabel('m');
grid on;

figure (14)
plot((x./d),Fpr,'r--');
ylabel('Passive Resistance Force (N/m)');
xlabel('Horizontal Displacement to Diameter Ratio');
grid on;

figure (15)
plot(x,y,'o--r');
xlabel('X (m)');
ylabel('Y (m)');
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 %Data presentation & checking for design safety

meu=mean (u);
meue=mean (ue);
mea=mean (a);
meFDf=mean (FDf);
meFI=mean (FI);
meFHf=mean (FHf);
meFLf=mean (FLf);

mxu=max(u);
mxue=max(ue);
mxa=max(a);
mxFDf=max(FDf);
mxFI=max(FI);
mxFHf=max(FHf);
mxFLf=max(FLf);

mxxu=min(u);
mxxue=min(ue);
mxxa=min(a);
mxxFDf=min(FDf);
mxxFI=min(FI);
mxxFHf=min(FHf);

mxxxu=abs(mxxu);
mxxxue=abs(mxxue);
```

```
mxxxa=abs(mxxa);
mxxxFDf=abs(mxxFDf);
mxxxFI=abs(mxxFI);
mxxxFHf=abs(mxxFHf);

fprintf('Fourier Analysis Force Model "Irregular Wave & Current"\n');
fprintf('\n');
fprintf('Irregular wave elevation spectrum properties @ sea surface [Tp =
%3.3f sec. ',Tp);
fprintf('& Hs = %3.3f m]',Hs);
fprintf(', Depth (D) = %3.3f m',D);
fprintf(', Storm duration (t) = %3.0f sec.',tmax);
fprintf(', Time step (Delta t) = %3.3f sec.',nt);
fprintf(', Cable/pipeline dry mass (m) = %3.3f kg/m ',mass);
fprintf(', submerged mass (m) = %3.3f kg/m ',msbefore);
fprintf(', submerged weight (Ws) = %3.3f N/m ',Wsbefore);
fprintf(', length (L) = %2.0f m ',L);
fprintf('& diameter (d) = %3.2f m.\n',d);
fprintf('\n');
fprintf('Wave velocity mean value = %3.3f m/sec.\n',meu);
fprintf('Effective wave velocity mean value = %3.3f m/sec.\n',meue);
fprintf('Wave acceleration mean value = %3.3f m/sq. sec.\n',mea);
fprintf('\n');
if mxue>mxxxue  ;
fprintf('Wave velocity maximum value = % 3.3f m/sec.\n',mxu);
else
fprintf('Wave velocity maximum value = % 3.3f m/sec.\n',mxxu);
                    end
if mxu>mxxxu  ;
fprintf('Effective wave velocity maximum value = % 3.3f m/sec.\n',mxue);
else
fprintf('Effective wave velocity maximum value = % 3.3f m/sec.\n',mxxue);
end
if mxa>mxxxa ;
fprintf('Wave acceleration maximum value = % 3.3f m/sq. sec.\n',mxa);
else
fprintf('Wave acceleration maximum value = % 3.3f m/sq. sec.\n',mxxa);
end
fprintf('\n');
fprintf('n-th Moment Spectrum Results:\n')
fprintf('\n');
fprintf('Significant flow velocity amplitude at cable level (Us) = % 3.3f
m/sec.\n',us);
fprintf('Average period between successive crests at cable/pipeline level
"Peak Period" (Tp) = % 3.3f sec.\n',tp);
fprintf('Mean zero up-crossing period of oscillating flow cable/pipeline
level (Tu) = % 3.3f sec.\n',tu);
fprintf('Cable roughness : Fine \n');
fprintf('Seabed roughness : Fine \n');
if KC<=10 ;
fprintf('Keulegan-Carpenter number (Us*Tp/d = 10) =% 3.3f  \n',KC);
elseif 10<KC && KC<=15 ; fprintf('Keulegan-Carpenter number (Us*Tp/d = 15)
= % 3.3f \n',KC);
elseif 15<KC && KC<=20 ; fprintf('Keulegan-Carpenter number (Us*Tp/d = 20)
= % 3.3f \n',KC);
elseif 20<KC && KC<=30 ; fprintf('Keulegan-Carpenter number (Us*Tp/d = 30)
= % 3.3f \n',KC);
```

```
elseif 30<KC && KC<=40 ; fprintf('Keulegan-Carpenter number (Us*Tp/d = 40)
= % 3.3f \n',KC);
elseif 40<KC && KC<=50 ; fprintf('Keulegan-Carpenter number (Us*Tp/d = 50)
= % 3.3f \n',KC);
elseif 50<KC && KC<=60 ; fprintf('Keulegan-Carpenter number (Us*Tp/d = 60)
= % 3.3f \n',KC);
else
fprintf('Keulegan-Carpenter number (Us*Tp/d = 70) = % 3.3f  \n',KC);
                                end
fprintf('Current to wave velocities ratio (Uc/Us) = % 12.5f \n',M);
fprintf('\n');
fprintf('Fourier Coefficicents:\n')
fprintf('\n');
fprintf('CD0 = % 3.3f  \n',CD0);
fprintf('CD1 = % 3.3f  \n',CD1);
fprintf('CD2 = % 3.3f  \n',CD2);
fprintf('CD3 = % 3.3f  \n',CD3);
fprintf('CD4 =% 3.3f  \n',CD4);
fprintf('CD5 = % 3.3f  \n',CD5);
fprintf('PHYD1 = % 3.3f  \n',PHYD1);
fprintf('PHYD2 = % 3.3f  \n',PHYD2);
fprintf('PHYD3 = % 3.3f  \n',PHYD3);
fprintf('PHYD4 = % 3.3f  \n',PHYD4);
fprintf('PHYD5 = % 3.3f  \n',PHYD5);
fprintf('\n');
fprintf('CL0 = % 3.3f  \n',CL0);
fprintf('CL1 = % 3.3f  \n',CL1);
fprintf('CL2 = % 3.3f  \n',CL2);
fprintf('CL3 = % 3.3f  \n',CL3);
fprintf('CD4 = % 3.3f  \n',CL4);
fprintf('CL5 = % 3.3f  \n',CL5);
fprintf('PHYL1 = % 3.3f  \n',PHYL1);
fprintf('PHYL2 = % 3.3f  \n',PHYL2);
fprintf('PHYL3 = % 3.3f  \n',PHYL3);
fprintf('PHYL4 = % 3.3f  \n',PHYL4);
fprintf('PHYL5 = % 3.3f  \n',PHYL5);
fprintf('\n');
fprintf('Calculated Forces:\n');
fprintf('\n');
fprintf('Iinertia force mean value = %3.3f N/m\n',meFI);
fprintf('\n');
fprintf('Fourier drag force mean value = %3.3f N/m\n',meFDf);
fprintf('Fourier total horizontal force mean value = %3.3f N/m\n',meFHf);
fprintf('Fourier vertical force mean value = %3.3f N/m\n',meFLf);
fprintf('\n');
if mxFI>mxxxFI  ;
fprintf('Inertia force maximum value = % 3.3f N/m\n',mxFI);
else
fprintf('Inertia force maximum value = % 3.3f N/m\n',mxxFI);
end
fprintf('\n');
if mxFDf>mxxxFDf  ;
fprintf('Fourier drag force maximum value = % 3.3f N/m\n',mxFDf);
else
fprintf('Fourier drag force maximum value = % 3.3f N/m\n',mxxFDf);
end
if mxFHf>mxxxFHf ;
```

```
fprintf('Fourier total horizontal force maximum value = % 3.3f
N/m\n',mxFHf);
else
fprintf('Fourier total horizontal force maximum value = % 3.3f
N/m\n',mxxFHf);
end
fprintf('Fourier vertical force maximum value = % 3.3f N/m\n',mxFLf);
fprintf('\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

limit = (10.*d);
xmaxp = max(x);
xmin = min(x);
absxmin = abs(xmin);
if absxmin <= xmaxp, xmax = xmaxp;
else xmax = absxmin;
end
ymaxp = max(y);
ymin = min(y);
absymin = abs(ymin);
if absymin <= ymaxp, ymax = ymaxp;
else ymax = absymin;
end
zpmax = max(zp);
fprintf('Post Analysis Results ( @ %3.0f Percent of Orginal Mass):
\n',percentage);
fprintf('\n');
if xmax > limit , fprintf('Design : Horizontally not safe');
else
fprintf('Design : Horizontally safe');
end
fprintf('\n');
if ymax > 0;
fprintf('Design : Vertically not safe');
else
fprintf('Design : Vertically safe');
end
fprintf('\n');
fprintf('\n');
if ymax <= 0 && xmax < limit, if percentage == 0;
fprintf('Required Added Mass = %3.3f Kg/m ',mchain);
fprintf('Diameter = %3.3f m \n',d);
fprintf('Mass = %3.3f Kg/m \n',ma);
fprintf('Submerged Mass = %3.3f Kg/m \n',ms);
fprintf('Submerged Weight = %3.3f N/m \n',Ws);
  else fprintf('Required Added Mass = %3.3f Kg/m \n',mchain);
fprintf('New Diameter = %3.3f m \n',d);
fprintf('New Mass = %3.3f Kg/m \n',ma);
fprintf('New Submerged Mass = %3.3f Kg/m \n',ms);
fprintf('New Submerged Weight = %3.3f N/m \n',Ws);
end
elseif ymax > d,
fprintf('Someting is Wrong!!!!!!!!! \n');
else
fprintf('Design Is Not Safe \n');
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Appendix 2: Function File
function M=rand_extended(a,b,n,m)
%rand_extended computes a random number in the interval [a,b]
%It is basically an extended version of the command rand
%Inputs:
       %a: the lower bound of the interval
       %b: the upper bound of the interval
       %m,n(optional): dimensions of the matrix of random numbers
       %to be generated
%Outputs:
   %M: random number/matrix in the interval [a,b]
%
 %Example1:
  %M=rand_extended(1,2,2,2)
%
 %Example2:
  %M=rand_extended(1,2))
%This function is written by :
  %Nassim Khaled, Ph.D.
%If you have any comments or face any problems, please feel free to leave
%your comments and i will try to reply to you as fast as possible.
if nargin==2
    M=(b-a).*rand(1)+a;
else
    M=(b-a).*rand(m,n)+a;
end
```