FBEM (Facet-Based SAR Altimeter Echo Model for Snow-Covered Sea Ice)
SIMULATION EXAMPLES

Citation: Landy, J.C., Tsamados, M. and Scharien, R.K., 2019. A Facet-Based Numerical Model for Simulating SAR Altimeter Echoes From Heterogeneous Sea Ice Surfaces. IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2018.2889763. https://ieeexplore.ieee.org/abstract/document/8625441

PULSE-LIMITED ECHOES WITH VARYING SEA ICE ROUGHNESS

SHELL.m

```
%% Model Variables (MODIFIABLE)

% Up to 3 variables can be vectors


% Geophysical parameters
sigma_s = 0.001; % snow rms height (default = 0.001 m)
l_s = 0.04; % snow correlation length (default = 0.04 m)
T_s = -20; % snow bulk temperature (default = -20 C)
rho_s = 350; % snow bulk density (default = 350 kg/m^3)
r_s = 0.002; % snow grain size (normal range from 0.0001 to 0.004 m, default 1 mm)
h_s = 0.2; % snow depth, m

sigma_si = 0.002; % sea ice rms height (default = 0.002 m)
l_si = 0.02; % sea ice correlation length (default = 0.02 m)
T_si = -2; % sea ice bulk temperature (default = -15 C)
S_si = 6; % sea ice bulk salinity (default = 6 ppt)

sigma_sw = 0.00001; % lead rms height (default = 0.00001 m)
T_sw = 0; % temperature of seawater (default = 0 C)
S_sw = 34; % salinity of seawater (default = 34 ppt)

% Antenna parameters
lambda = 0.0221; % radar wavelength (default = 0.0221, Ku-band e.g. Cryosat-2)
```

```
GP = whos('*'); % all parameters controlling scattering
signatures

op_mode = 1; % operational mode: 1 = pulse-limited, 2 = SAR
(PL-mode only feasible on high memory machines)
beam_weighting = 2; % weighting on the beam-wise azimuth FFT:
1 = rectangular, 2 = Hamming (defualt = Hamming)
P_T = 2.188e-5; % transmitted peak power (default = 2.188e-5
watts)

pitch = 0; % antenna bench pitch counterclockwise (up to
~0.001 rads)
roll = 0; % antenna bench roll counterclockwise (up to ~0.005
rads)

h = 720000; % satellite altitude (default = 720000 m)
v = 7500; % satellite velocity (default = 7500 m/s)
N_b = 64; % no. beams in synthetic aperture (default = 64,
e.g. Cryosat-2)
if op_mode==1 % single beam for PL echo
    N_b = 1;
else
end

prf = 18182; % pulse-repetition frequency (default = 18182
Hz, e.g. Cryosat-2)
bandwidth = 320*10^6; % antenna bandwidth (default = 320*10^6
Hz, e.g. Cryosat-2)
G_0 = 42; % peak antenna gain, dB
D_0 = 36.12; % synthetic beam gain, SAR mode

% Number of range bins
N_tb = 70; % (default = 70)

% Range bin at mean scattering surface, i.e. time = 0
t_0 = 15; % (default = 15)

% Time oversampling factor
t_sub = 2;

% Parameters of synthetic topography
topo_type = 2; % type of surface: 1 = Gaussian, 2 =
lognormal, 3 = fractal
sigma_surf = [0.2 0.5]; % large-scale rms roughness height
(default = 0.1 m)
l_surf = 5; % large-scale correlation length (default = 5 m)
H_surf = 0.5; % Hurst parameter (default = 0.5)
dx = 20; % resolution of grid, m (WARNING use dx>=10 for PL
mode and dx>=5 for SAR mode)
```

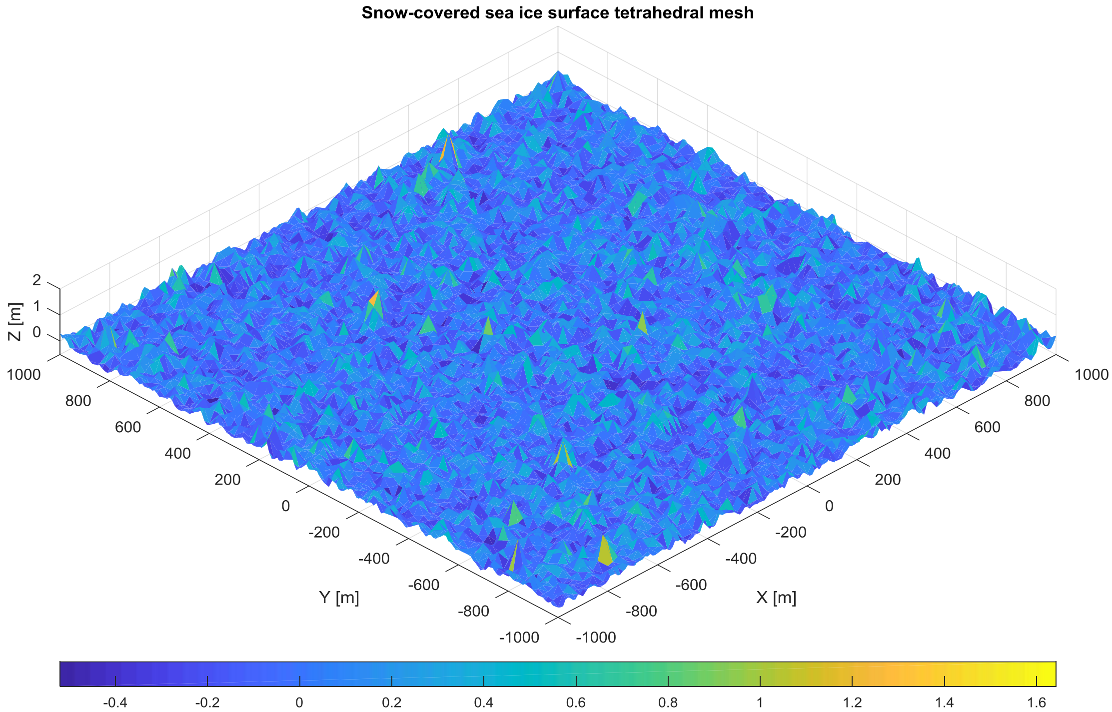
```
% Lead parameters (optional)
L_w = 0; % lead width (default = 100 m)
L_h = 0; % lead depth (default = 0.2 m)
D_off = 0; % distance off nadir (default = 0 m)

% Add melt ponds (optional)
T_fw = 0; % temperature of freshwater in pond (default = 0 C)
f_p = 0; % melt pond fraction (default = 0.5)
u_a = 4; % boundary-layer wind speed (default = 4 m/s)

save('FEM_Simulations');

% Optional Plotting
topo_plot = 1; % example plot of tetrahedral surface mesh
echo_plot = 1; % example plots of modelled echoes

%% Antenna Geometry

epsilon_b = lambda/(2*N_b*v*(1/prf)); % angular resolution of
beams from full look crescent (beam separation angle)


%% Loop Echo Model

% Use parallel processing
% parpool

% Identify vector variables
PARAMETERS = whos('*');
idS = find(cellfun(@(x) x(:,2),{PARAMETERS.size})>1);
idG = find(cellfun(@(x) x(:,2),{GP.size})>1);

if isempty(idS)
    vec1 = 1;
    vec2 = 1;
    vec3 = 1;
elseif numel(idS) == 1
    eval(['vec1 = ',PARAMETERS(idS(1)).name,';']);
    vec2 = 1;
    vec3 = 1;
elseif numel(idS) == 2
    eval(['vec1 = ',PARAMETERS(idS(1)).name,';']);
    eval(['vec2 = ',PARAMETERS(idS(2)).name,';']);
    vec3 = 1;
elseif numel(idS) == 3
    eval(['vec1 = ',PARAMETERS(idS(1)).name,';']);
    eval(['vec2 = ',PARAMETERS(idS(2)).name,';']);
    eval(['vec3 = ',PARAMETERS(idS(3)).name,';']);
```

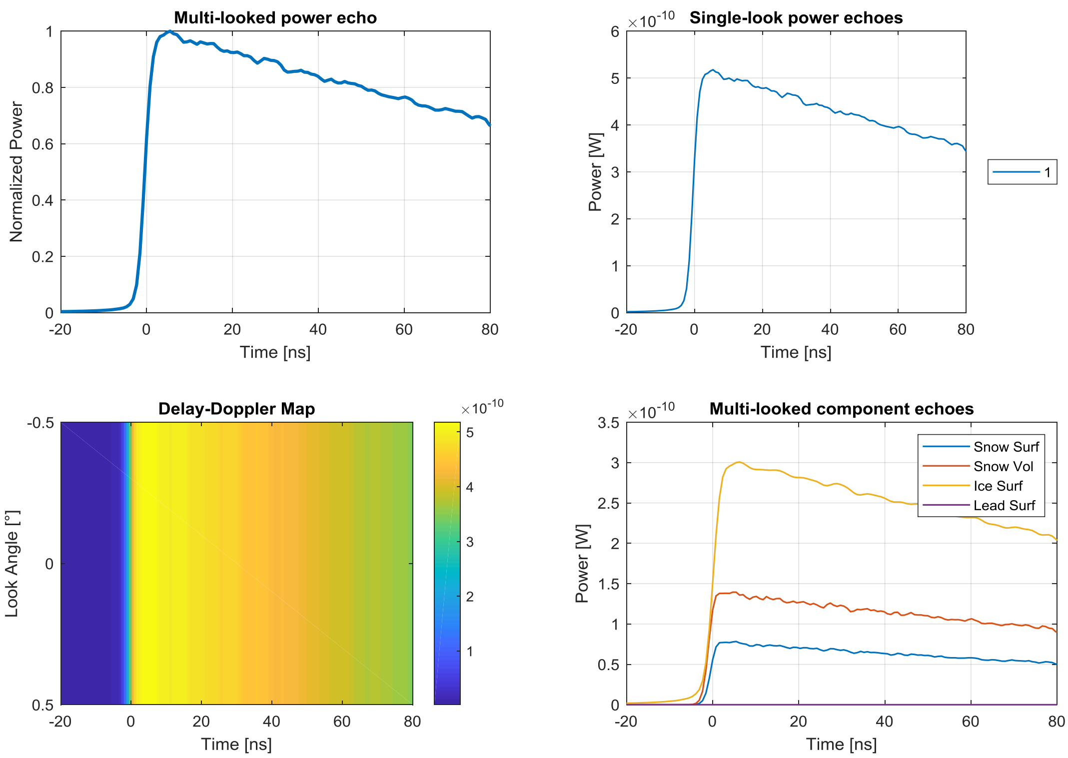
```
end


% Loop model over vector variables
P_t_full_range =
cell(length(vec1),length(vec2),length(vec3));
P_t_ml_range = cell(length(vec1),length(vec2),length(vec3));
P_t_full_comp_range =
cell(length(vec1),length(vec2),length(vec3));
P_t_ml_comp_range =
cell(length(vec1),length(vec2),length(vec3));

counter = 0;
for i = 1:length(vec1)

    for j = 1:length(vec2)

        for k = 1:length(vec3)

            for l = 1:length(idS)
                eval([PARAMETERS(idS(l)).name ' = vec'
num2str(l) '(i);']);
            end

            % Time domain
            t = (0.5/bandwidth)*((1:(1/t_sub):N_tb) - t_0);

            if counter<1 || ~isempty(idG) % skip if
scattering properties do not change between runs

                % Effective width of angular extent of
coherent component (TUNING PARAMETER FOR LEADS)
                beta_c = epsilon_b; % no wider than synthetic
beam angle epsilon_b, rads

                % Surface & volume backscattering properties

[theta,sigma_0_snow_surf,sigma_0_snow_vol,kappa_e,tau_snow,c_
s,epsr_ds] =
snow_backscatter(lambda,sigma_s,l_s,T_s,rho_s,r_s,h_s,beta_c)
;
                [~,sigma_0_ice_surf,~] =
ice_backscatter(lambda,sigma_si,l_si,T_si,S_si,h_s,beta_c,eps
r_ds);
                [~,sigma_0_lead_surf] =
lead_backscatter(lambda,sigma_sw,T_sw,S_sw,beta_c);
                [~,sigma_0_mp_surf] =
pond_backscatter(lambda,T_fw,beta_c,u_a);
```

```
            else
            end

            counter = counter + 1;

            itN = 3; % Number of iterations
            P_t_full = zeros(length(t),N_b,itN);
            P_t_ml = zeros(length(t),itN);
            P_t_full_comp = zeros(length(t),N_b,4,itN);
            P_t_ml_comp = zeros(length(t),4,itN);
            for l = 1:itN % Average over n iterations

                % Synthetic topography
                [PosT,surface_type] =
synthetic_topo_shell(op_mode,topo_type,sigma_surf,l_surf,H_su
rf,dx,L_w,L_h,D_off,f_p);

                % Run Facet-Based Echo Model

[P_t_full(:,:,l),P_t_ml(:,l),P_t_full_comp(:,:,:,l),P_t_ml_co
mp(:,:,l)] =
Facet_Echo_Model(op_mode,lambda,bandwidth,P_T,h,v,pitch,roll,
prf,beam_weighting,G_0,D_0,N_b,t,PosT,surface_type,sigma_0_sn
ow_surf,sigma_0_snow_vol,kappa_e,tau_snow,c_s,h_s,sigma_0_ice
_surf,sigma_0_lead_surf,sigma_0_mp_surf);

                fprintf(['Iteration ' num2str(l) '/'
num2str(itN) ', Simulation ' num2str(counter) '/'
num2str(length(vec1)*length(vec2)*length(vec3)) '\n']);

            end

            P_t_full_range{i,j,k} = nanmean(P_t_full,3);
            P_t_ml_range{i,j,k} = nanmean(P_t_ml,2);
            P_t_full_comp_range{i,j,k} =
nanmean(P_t_full_comp,4);
            P_t_ml_comp_range{i,j,k} =
nanmean(P_t_ml_comp,3);

            % Optional plotting
            if (topo_plot | echo_plot)>0

Plotting(topo_plot,echo_plot,PosT,t,P_t_ml_range{i,j,k},P_t_f
ull_range{i,j,k},P_t_ml_comp_range{i,j,k},N_b,epsilon_b);
            else
            end

        end
```

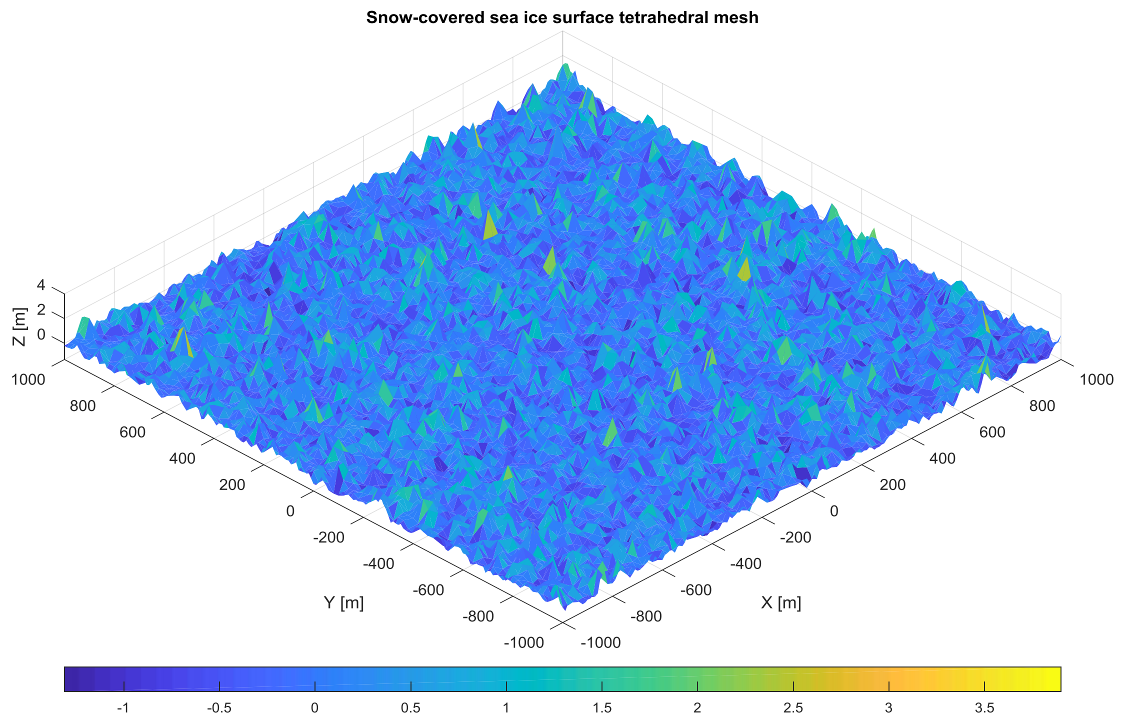
```
    end
end
```

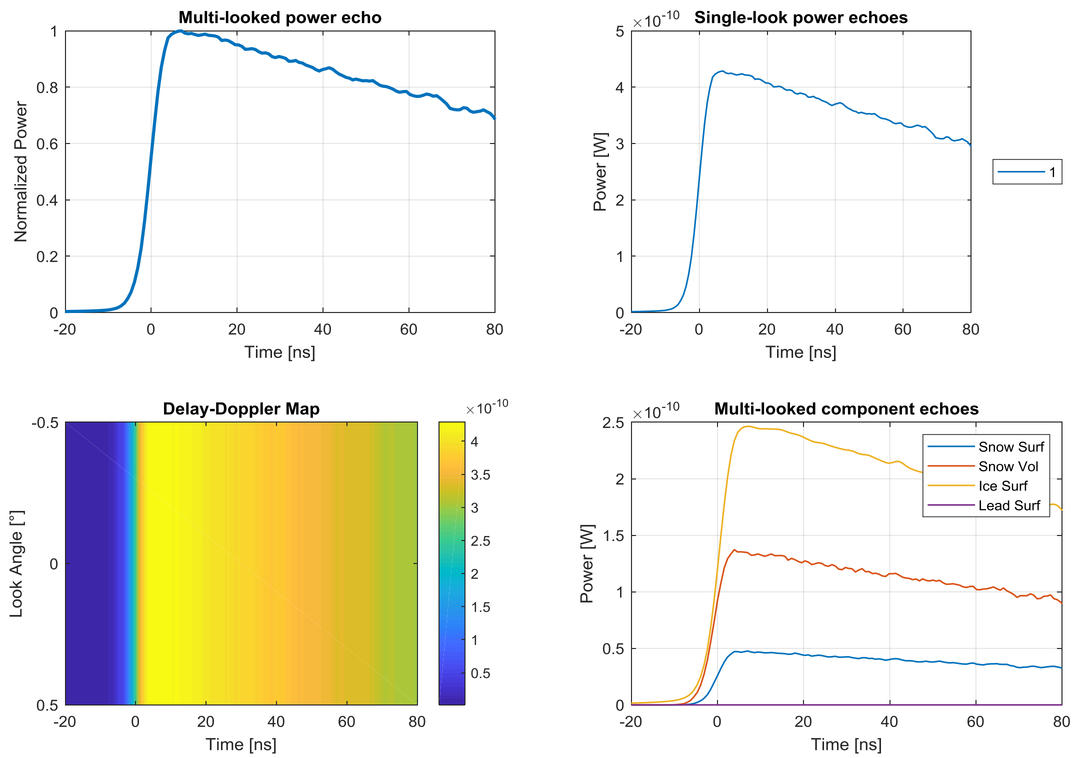
sigma_surf = 0.2





sigma_surf = 0.5





SAR ECHOES WITH LEADS AT VARYING OFF-NADIR DISTANCE

SHELL.m

```
%% Model Variables (MODIFIABLE)

% Up to 3 variables can be vectors


% Geophysical parameters
sigma_s = 0.001; % snow rms height (default = 0.001 m)
l_s = 0.04; % snow correlation length (default = 0.04 m)
T_s = -20; % snow bulk temperature (default = -20 C)
rho_s = 350; % snow bulk density (default = 350 kg/m^3)
r_s = 0.002; % snow grain size (normal range from 0.0001 to
0.004 m, default 1 mm)
h_s = 0.2; % snow depth, m

sigma_si = 0.002; % sea ice rms height (default = 0.002 m)
l_si = 0.02; % sea ice correlation length (default = 0.02 m)
T_si = -2; % sea ice bulk temperature (default = -15 C)
S_si = 6; % sea ice bulk salinity (default = 6 ppt)

sigma_sw = 0.0001; % lead rms height (default = 0.00001 m)
T_sw = 0; % temperature of seawater (default = 0 C)
S_sw = 34; % salinity of seawater (default = 34 ppt)

% Antenna parameters
lambda = 0.0221; % radar wavelength (default = 0.0221, Ku-
band e.g. Cryosat-2)
GP = whos('*'); % all parameters controlling scattering
signatures

op_mode = 2; % operational mode: 1 = pulse-limited, 2 = SAR
(PL-mode only feasible on high memory machines)
beam_weighting = 2; % weighting on the beam-wise azimuth FFT:
1 = rectangular, 2 = Hamming (default = Hamming)
P_T = 2.188e-5; % transmitted peak power (default = 2.188e-5
watts)

pitch = 0; % antenna bench pitch counterclockwise (up to
~0.001 rads)
roll = 0; % antenna bench roll counterclockwise (up to ~0.005
rads)

h = 720000; % satellite altitude (default = 720000 m)
v = 7500; % satellite velocity (default = 7500 m/s)
```

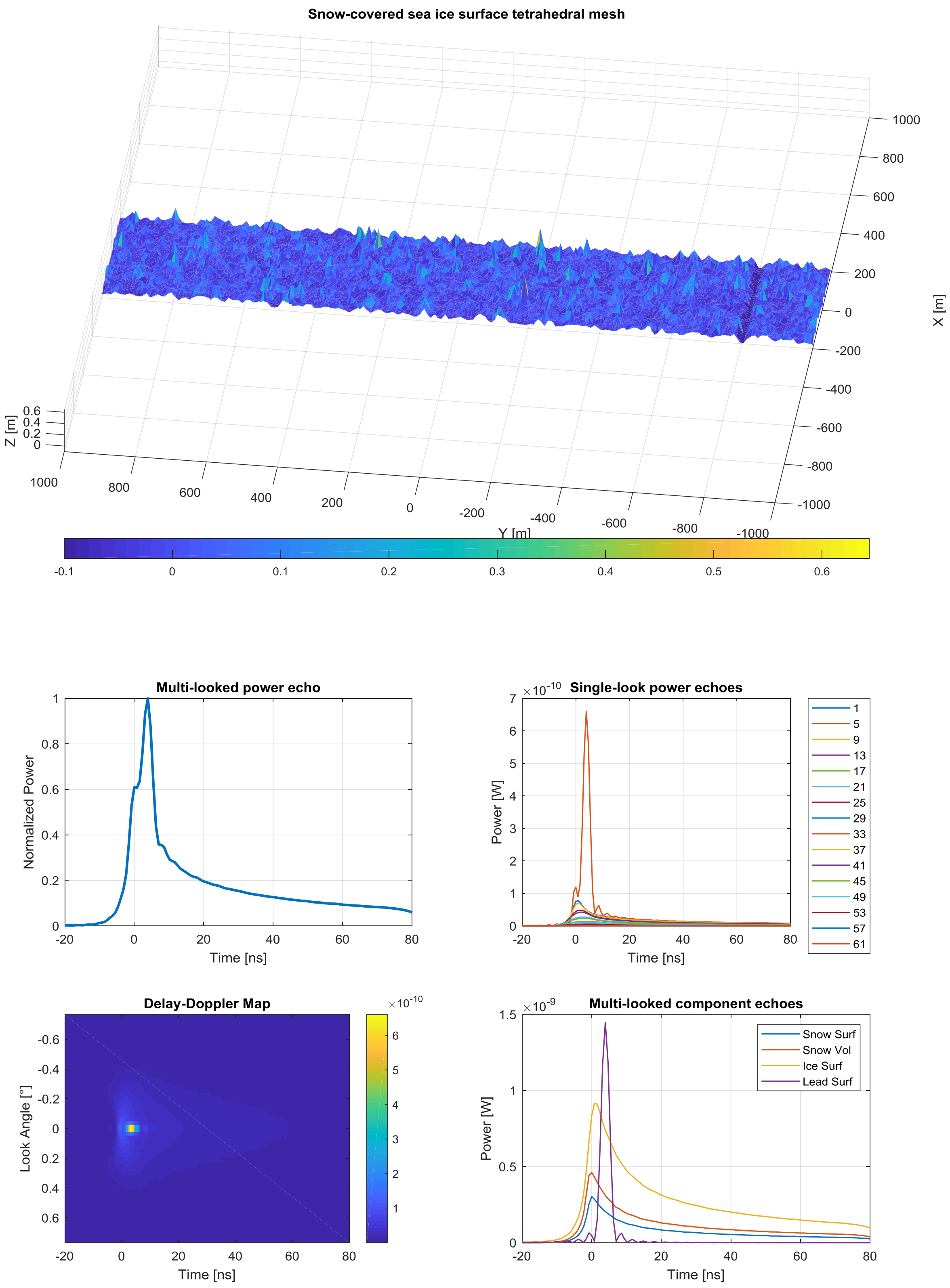
```
N_b = 64; % no. beams in synthetic aperture (default = 64,
e.g. Cryosat-2)
if op_mode==1 % single beam for PL echo
    N_b = 1;
else
end

prf = 18182; % pulse-repetition frequency (default = 18182
Hz, e.g. Cryosat-2)
bandwidth = 320*10^6; % antenna bandwidth (default = 320*10^6
Hz, e.g. Cryosat-2)
G_0 = 42; % peak antenna gain, dB
D_0 = 36.12; % synthetic beam gain, SAR mode

% Number of range bins
N_tb = 70; % (default = 70)

% Range bin at mean scattering surface, i.e. time = 0
t_0 = 15; % (default = 15)

% Time oversampling factor
t_sub = 2;

% Parameters of synthetic topography
topo_type = 2; % type of surface: 1 = Gaussian, 2 =
lognormal, 3 = fractal
sigma_surf = 0.05; % large-scale rms roughness height
(default = 0.1 m)
l_surf = 5; % large-scale correlation length (default = 5 m)
H_surf = 0.5; % Hurst parameter (default = 0.5)
dx = 10; % resolution of grid, m (WARNING use dx>=10 for PL
mode and dx>=5 for SAR mode)

% Lead parameters (optional)
L_w = 10; % lead width (default = 100 m)
L_h = 0.1; % lead depth (default = 0.2 m)
D_off = [800 1000]; % distance off nadir (default = 0 m)

% Add melt ponds (optional)
T_fw = 0; % temperature of freshwater in pond (default = 0 C)
f_p = 0; % melt pond fraction (default = 0.5)
u_a = 4; % boundary-layer wind speed (default = 4 m/s)

save('FEM_Simulations');

% Optional Plotting
topo_plot = 1; % example plot of tetrahedral surface mesh
echo_plot = 1; % example plots of modelled echoes

%% Antenna Geometry
```

```
epsilon_b = lambda/(2*N_b*v*(1/prf)); % angular resolution of
beams from full look crescent (beam separation angle)


%% Loop Echo Model

% Use parallel processing
% parpool

% Identify vector variables
PARAMETERS = whos('*');
idS = find(cellfun(@(x) x(:,2),{PARAMETERS.size})>1);
idG = find(cellfun(@(x) x(:,2),{GP.size})>1);

if isempty(idS)
    vec1 = 1;
    vec2 = 1;
    vec3 = 1;
elseif numel(idS) == 1
    eval(['vec1 = ',PARAMETERS(idS(1)).name,';']);
    vec2 = 1;
    vec3 = 1;
elseif numel(idS) == 2
    eval(['vec1 = ',PARAMETERS(idS(1)).name,';']);
    eval(['vec2 = ',PARAMETERS(idS(2)).name,';']);
    vec3 = 1;
elseif numel(idS) == 3
    eval(['vec1 = ',PARAMETERS(idS(1)).name,';']);
    eval(['vec2 = ',PARAMETERS(idS(2)).name,';']);
    eval(['vec3 = ',PARAMETERS(idS(3)).name,';']);
end


% Loop model over vector variables
P_t_full_range =
cell(length(vec1),length(vec2),length(vec3));
P_t_ml_range = cell(length(vec1),length(vec2),length(vec3));
P_t_full_comp_range =
cell(length(vec1),length(vec2),length(vec3));
P_t_ml_comp_range =
cell(length(vec1),length(vec2),length(vec3));

counter = 0;
for i = 1:length(vec1)

    for j = 1:length(vec2)

        for k = 1:length(vec3)
```

```
            for l = 1:length(idS)
                eval([PARAMETERS(idS(l)).name ' = vec'
num2str(l) '(i);']);
            end

            % Time domain
            t = (0.5/bandwidth)*((1:(1/t_sub):N_tb) - t_0);

            if counter<1 || ~isempty(idG) % skip if
scattering properties do not change between runs

                % Effective width of angular extent of
coherent component (TUNING PARAMETER FOR LEADS)
                beta_c = epsilon_b; % no wider than synthetic
beam angle epsilon_b, rads

                % Surface & volume backscattering properties

[theta,sigma_0_snow_surf,sigma_0_snow_vol,kappa_e,tau_snow,c_
s,epsr_ds] =
snow_backscatter(lambda,sigma_s,l_s,T_s,rho_s,r_s,h_s,beta_c)
;
                [~,sigma_0_ice_surf,~] =
ice_backscatter(lambda,sigma_si,l_si,T_si,S_si,h_s,beta_c,eps
r_ds);
                [~,sigma_0_lead_surf] =
lead_backscatter(lambda,sigma_sw,T_sw,S_sw,beta_c);
                [~,sigma_0_mp_surf] =
pond_backscatter(lambda,T_fw,beta_c,u_a);

            else
            end

            counter = counter + 1;

            itN = 3; % Number of iterations
            P_t_full = zeros(length(t),N_b,itN);
            P_t_ml = zeros(length(t),itN);
            P_t_full_comp = zeros(length(t),N_b,4,itN);
            P_t_ml_comp = zeros(length(t),4,itN);
            for l = 1:itN % Average over n iterations

                % Synthetic topography
                [PosT,surface_type] =
synthetic_topo_shell(op_mode,topo_type,sigma_surf,l_surf,H_su
rf,dx,L_w,L_h,D_off,f_p);

                % Run Facet-Based Echo Model

[P_t_full(:,:,l),P_t_ml(:,l),P_t_full_comp(:,:,:,l),P_t_ml_co
```

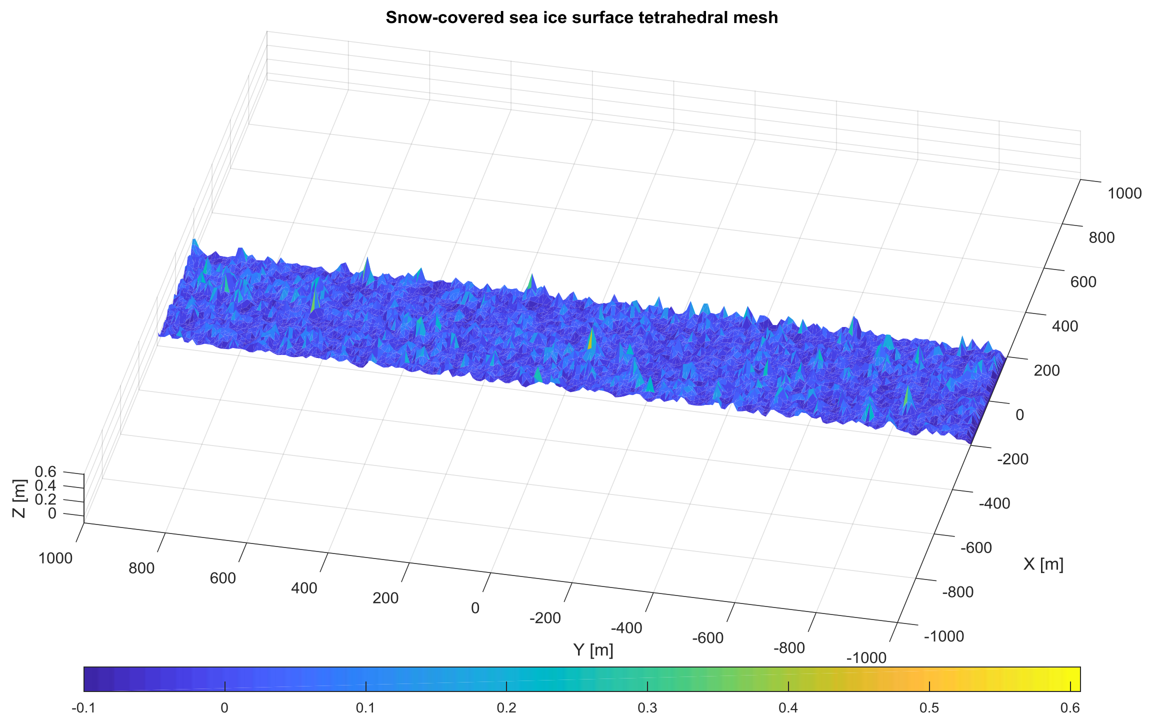
```
mp(:,:,l)] =
Facet_Echo_Model(op_mode,lambda,bandwidth,P_T,h,v,pitch,roll,
prf,beam_weighting,G_0,D_0,N_b,t,PosT,surface_type,sigma_0_sn
ow_surf,sigma_0_snow_vol,kappa_e,tau_snow,c_s,h_s,sigma_0_ice
_surf,sigma_0_lead_surf,sigma_0_mp_surf);

                fprintf(['Iteration ' num2str(l) '/'
num2str(itN) ', Simulation ' num2str(counter) '/'
num2str(length(vec1)*length(vec2)*length(vec3)) '\n']);

            end

            P_t_full_range{i,j,k} = nanmean(P_t_full,3);
            P_t_ml_range{i,j,k} = nanmean(P_t_ml,2);
            P_t_full_comp_range{i,j,k} =
nanmean(P_t_full_comp,4);
            P_t_ml_comp_range{i,j,k} =
nanmean(P_t_ml_comp,3);

            % Optional plotting
            if (topo_plot | echo_plot)>0

Plotting(topo_plot,echo_plot,PosT,t,P_t_ml_range{i,j,k},P_t_f
ull_range{i,j,k},P_t_ml_comp_range{i,j,k},N_b,epsilon_b);
            else
            end

        end

    end

end
```

```
D_off = 800
```

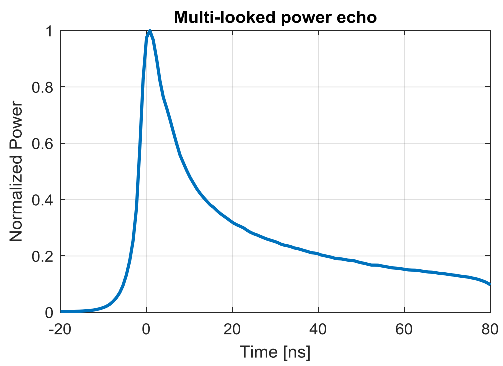


Snow-covered sea ice surface tetrahedral mesh
Z [m]
Y [m]
X [m]
Multi-looked power echo
Normalized Power
Time [ns]
Single-look power echoes
Power [W]
Time [ns]
Delay-Doppler Map
Look Angle [°]
Time [ns]
Multi-looked component echoes
Snow Surf
Snow Vol
Ice Surf
Lead Surf
Power [W]
Time [ns]

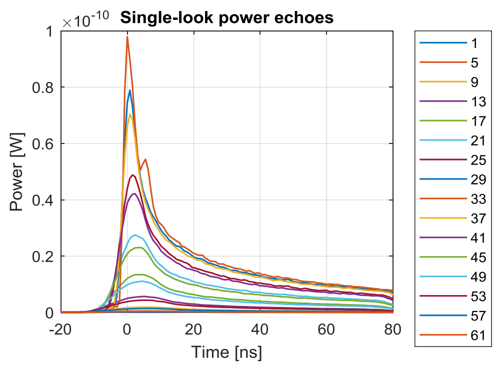
D_off = 1000



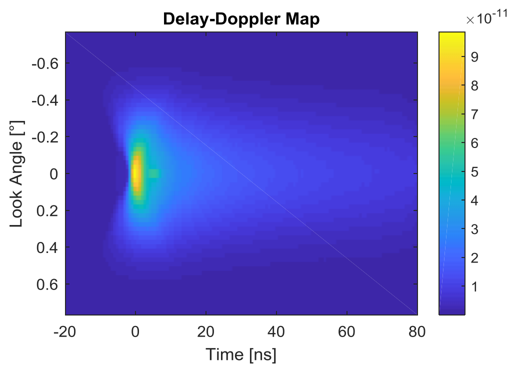
Snow-covered sea ice surface tetrahedral mesh
Z [m]
Y [m]
X [m]



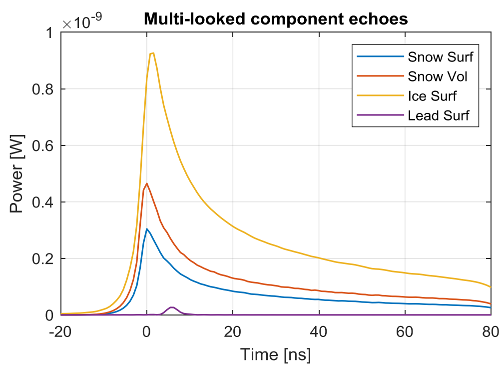
Multi-looked power echo
Normalized Power
Time [ns]



Single-look power echoes
Power [W]
Time [ns]



Delay-Doppler Map
Look Angle [°]
Time [ns]



Multi-looked component echoes
Snow Surf
Snow Vol
Ice Surf
Lead Surf
Power [W]
Time [ns]