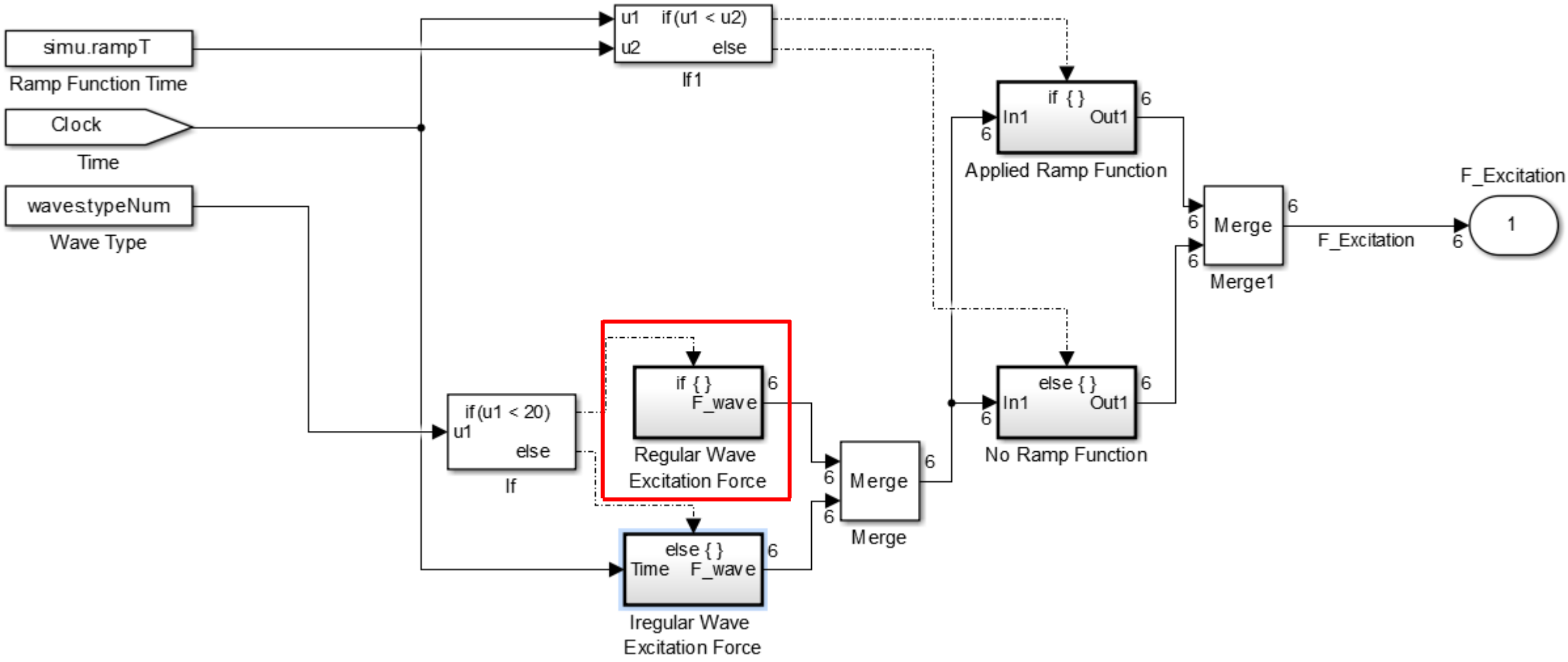
Excitation Force (Regular Wave)



Excitation Force (Regular Wave)

$$F_{ext} = \Re\left[R_f \frac{H}{2} F_x(\omega_r) e^{i(\omega_r t)}\right] = R_f \frac{H}{2} \Re\left[F_x(\omega_r) e^{i(\omega_r t)}\right]$$

$$F_x(\omega_r) e^{i(\omega_r t)} = (\Re F_x(\omega_r) + i Im F_x(\omega_r))(\cos(\omega_r t) + i \sin(\omega_r t))$$

$$\Re\left[F_x(\omega_r) e^{i(\omega_r t)}\right] = \Re F_x(\omega_r) \cos(\omega_r t) - Im F_x(\omega_r) \sin(\omega_r t)$$

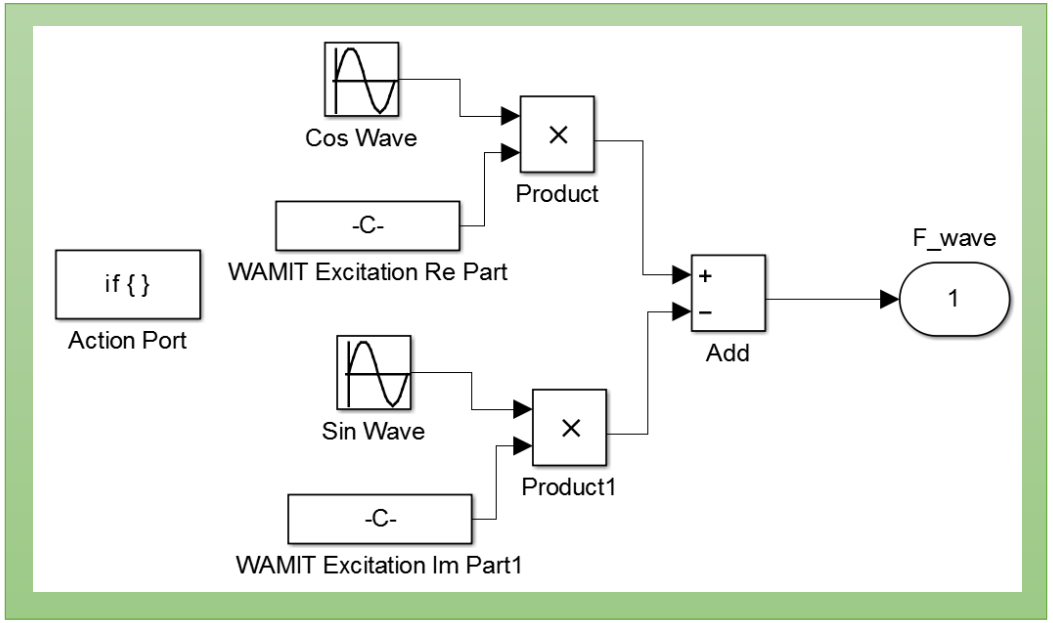
$\Re F_x(\omega_r)$: WAMIT Excitation Real Part

$Im F_x(\omega_r)$: WAMIT Excitation Imaginary Part

$$F_{ext} = R_f \frac{H}{2} [\Re F_x(\omega_r) \cos(\omega_r t) - Im F_x(\omega_r) \sin(\omega_r t)]$$

$$R_f = \begin{cases} \frac{1}{2}\left(1 + \cos\left(\pi + \frac{\pi t}{t_r}\right)\right), & \frac{t}{t_r} < 1 \\ 1, & \frac{t}{t_r} \geq 1 \end{cases}$$

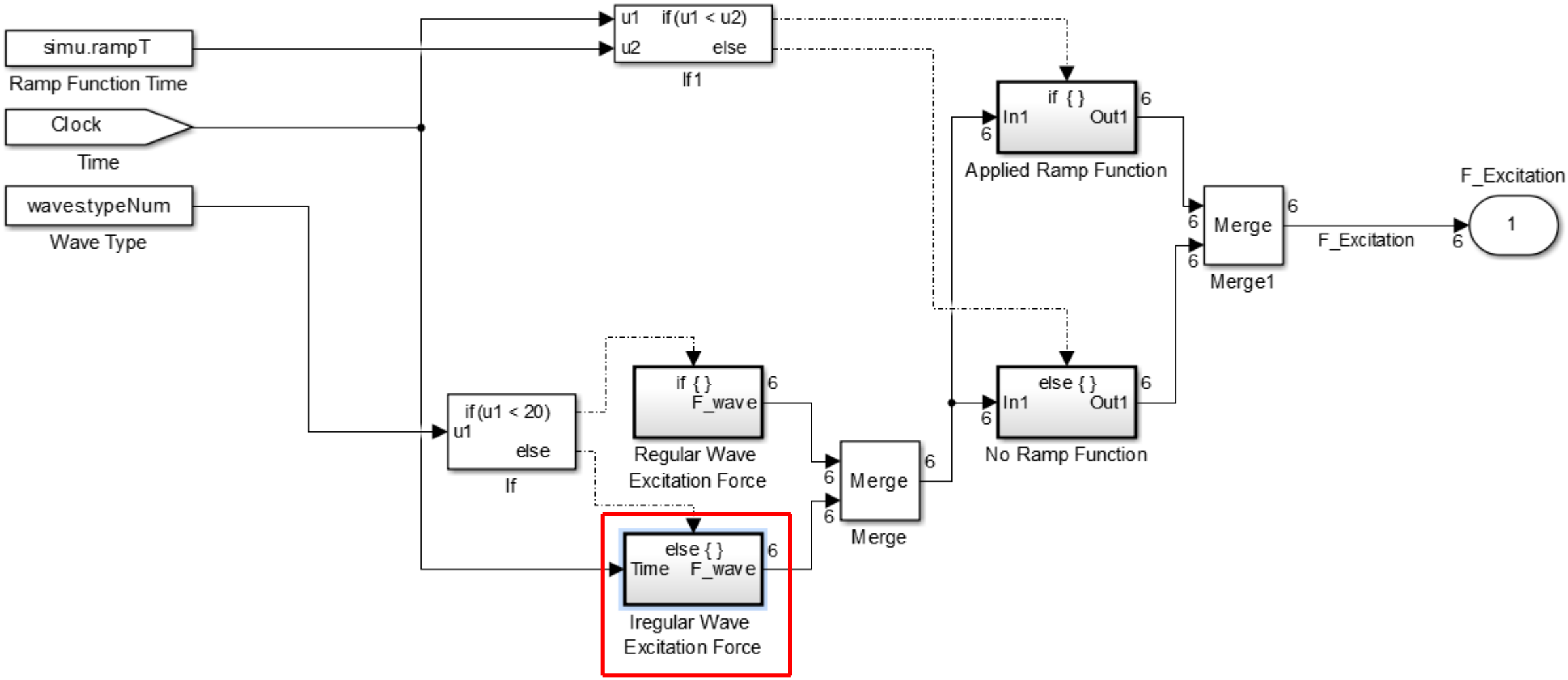
$$R_f = \begin{cases} \frac{1}{2}\left(1 + \sin\left(\pi + \frac{\pi t}{t_r} + \frac{\pi}{2}\right)\right), & \frac{t}{t_r} < 1 \\ 1, & \frac{t}{t_r} \geq 1 \end{cases}$$



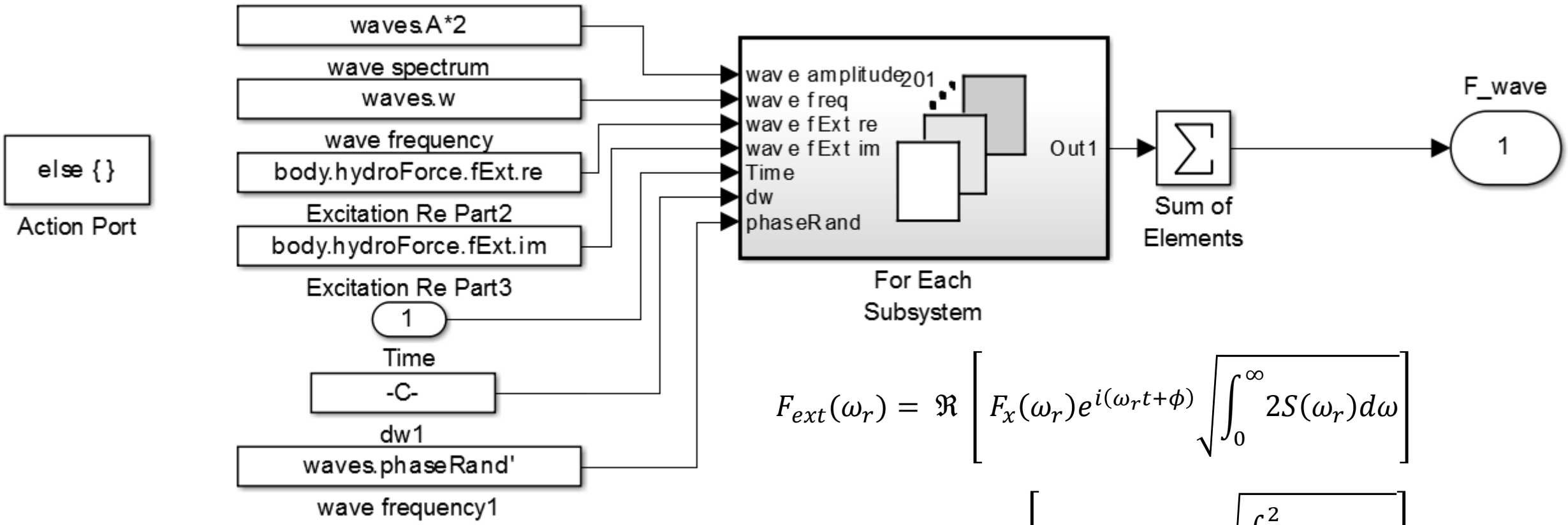
$$F_{ext} = \Re\left[R_f \frac{H}{2} F_x(\omega_r) e^{i(\omega_r t)}\right]$$

Where is H/2 ?

Excitation Force (Irregular Wave)



Excitation Force (Irregular Wave)

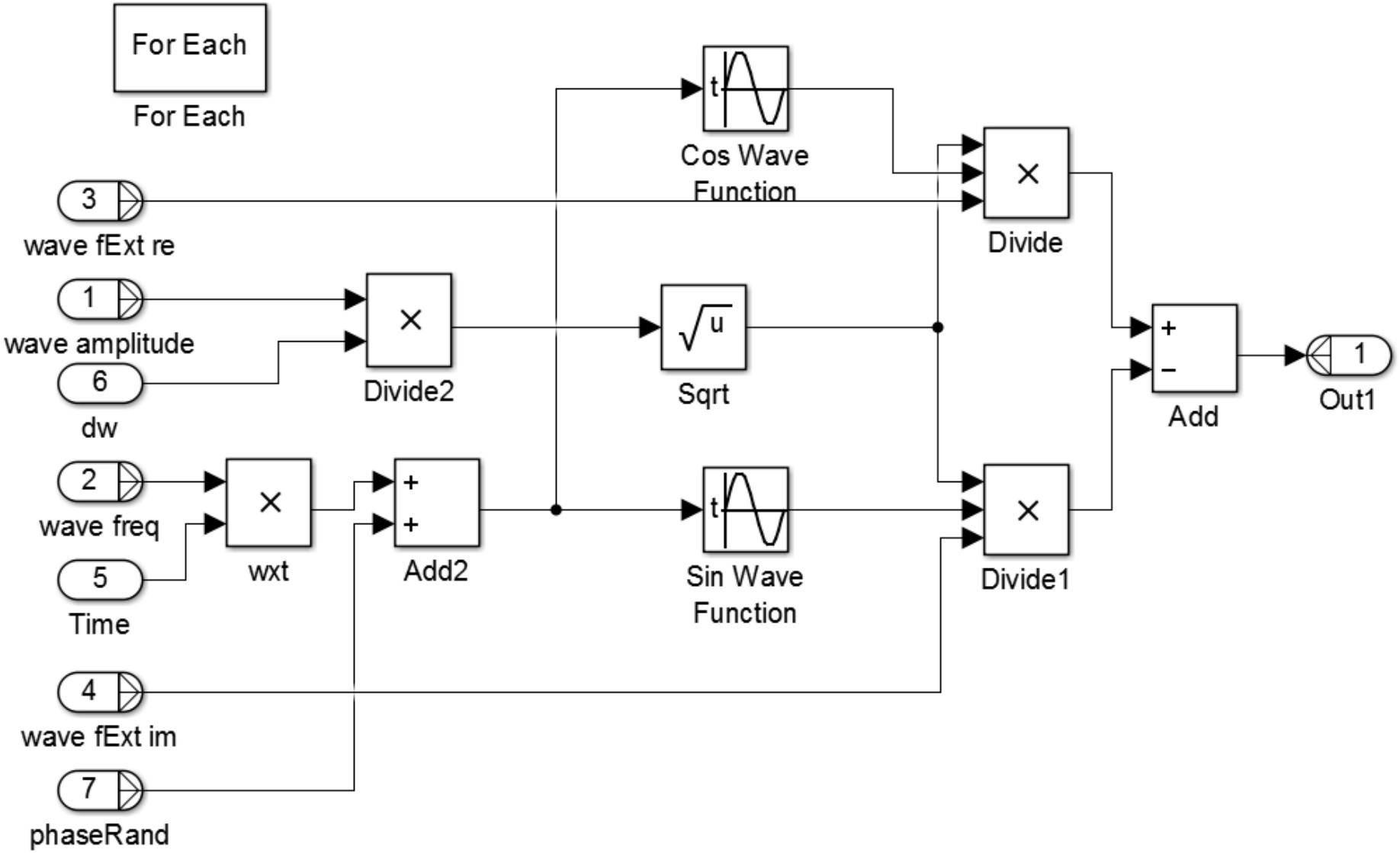


$$F_{ext}(\omega_r) = \Re\left[F_x(\omega_r)e^{i(\omega_r t+\phi)}\sqrt{\int_0^{\infty} 2S(\omega_r)d\omega}\right]$$

$$F_{ext}(\omega_r = 2) = \Re\left[F_x(2)e^{i(2t+\phi)}\sqrt{\int_0^{2} 2S(2)d\omega}\right]$$

Why is there a summation block after the square root block?

Excitation Force (Irregular Wave)



Excitation Force (Irregular Wave)

$$F_{ext}(t) = \eta(t) * f_e\,(t) = \int_{-\infty}^{\infty} \eta(\tau) f_e(t-\tau)\,d\tau$$

$$F_{ext}(\omega_r) = \eta(\omega_r) f_e(\omega_r)$$

$$f_e(\omega_r) = |f_e(\omega_r)| e^{i\theta} = F_x(\omega_r) = (\Re F_x(\omega_r) + i Im F_x(\omega_r)) \longrightarrow \text{From WAMIT}$$

$$\overline{\eta^2(x,y,t)} = \int_0^{\infty} S\,(f)\,df, where \quad S(f) = S_f(f) = 2\pi S_\omega(2\pi f) = 2\pi S_\omega(\omega) \longrightarrow \text{From Falnes}$$

$$\eta(x,y,t) = \sqrt{\int_0^{\infty} S\,(f)\,df}$$

$$\eta(x,y,\omega_r) = \int_{-\infty}^{\infty} \sqrt{\int_0^{\infty} 2\pi S(\omega_r)\,d\omega}\; e^{-i\omega t}\,dt$$

$$F_{ext}(\omega_r) = \Re\left[R_f F_x(\omega_r) e^{i(\omega_r t+\phi)} \sqrt{\int_0^{\infty} 2S(\omega_r)\,d\omega} \right]$$

waves.A*2

wave spectrum

$$F_{ext}(\omega_r) = \Re\left[R_f \underbrace{F_x(\omega_r)}_{f_e(\omega_r)} e^{i(\omega_r t+\phi)} \underbrace{\sqrt{\int_0^{\infty} 2S(\omega_r)\,d\omega}}_{\eta(\omega_r)} \right]$$

- What is eta(ω) in terms of s(ω)?
- $waves.\mathrm{A} = S(\omega_r), correct?$
- $waves.\mathrm{A} * 2 = \mathrm{Amplitude}, \mathrm{why?}$
- $\int_{-\infty}^{\infty} S(\omega_r)\,d\omega = \int_0^{\infty} 2S(\omega_r)\,d\omega$

wavesIrreg.m

```
%%  Calulate wave history
waveAmpTime = zeros(simu.maxIt+1,1);
% No wave ramping defined
if simu.rampT==0
    for i=1:simu.maxIt+1;
        t = (i-1)*simu.dt;
        for j=1:numFqs
            tmp=sqrt(2*Sf(j)*df);
            waveAmpTime(i) = waveAmpTime(i) + tmp*real(exp(sqrt(-1)*(frequency(j)*t + waves.phaseRand(j))));
        end

    end
% If wave ramping defined
else
    for i=1:maxRampIT
        t = (i-1)*simu.dt;

        for j=1:numFqs
            tmp=sqrt(2*Sf(j)*df);
            waveAmpTime(i) = waveAmpTime(i) + tmp*real(exp(sqrt(-1)*(frequency(j)*t + waves.phaseRand(j))));
        end
        waveAmpTime(i) = waveAmpTime(i)*(1+cos(pi+pi*(i-1)/maxRampIT))/2;
    end
    for i=maxRampIT+1:simu.maxIt+1
        t = (i-1)*simu.dt;

        for j=1:numFqs
            tmp=sqrt(2*Sf(j)*df);
            waveAmpTime(i) = waveAmpTime(i) + tmp*real(exp(sqrt(-1)*(frequency(j)*t + waves.phaseRand(j))));
        end
    end
end
```

$$\eta(\omega_r) = \left(\sqrt{2S(\omega_r)\Delta\omega_r}\right)\Re\left[e^{\sqrt{(-1)*(\omega_r t+\phi)}}\right]$$

$$\eta(\omega_r) = \left(\sqrt{2S(\omega_r)\Delta\omega_r}\right)\cos\left(\sqrt{(\omega_r t+\phi)}\right)$$

$$\eta(\omega_r) = e^{i(\omega_r t+\phi)}\sqrt{\int_0^{\infty} 2S(\omega_r)d\omega}$$

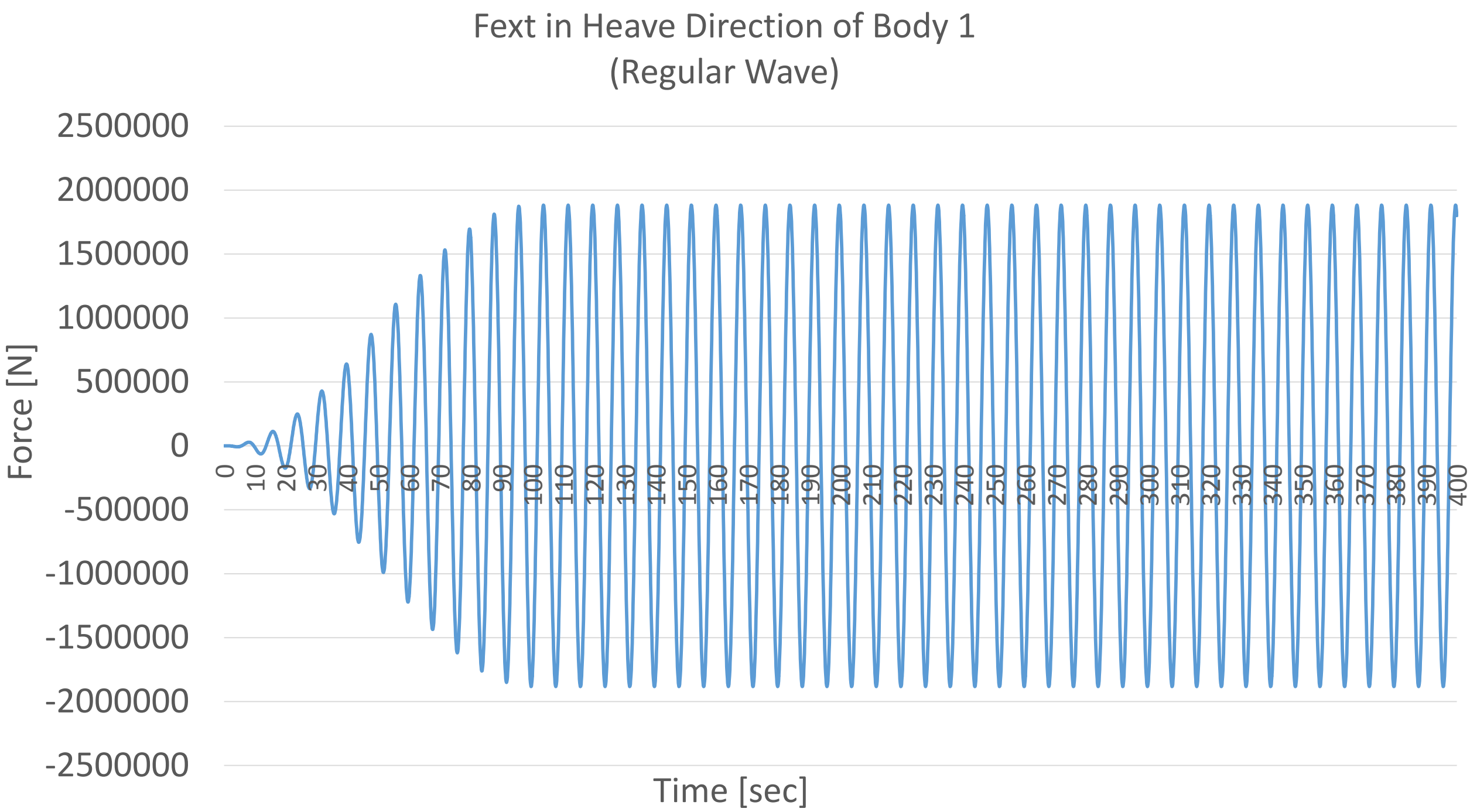
$$\eta(t) = \sum_{\omega=0,\,\phi=0}^{\infty}\left(\sqrt{2S(\omega_r)\Delta\omega_r}\right)\cos\left(\sqrt{(\omega_r t+\phi)}\right)$$

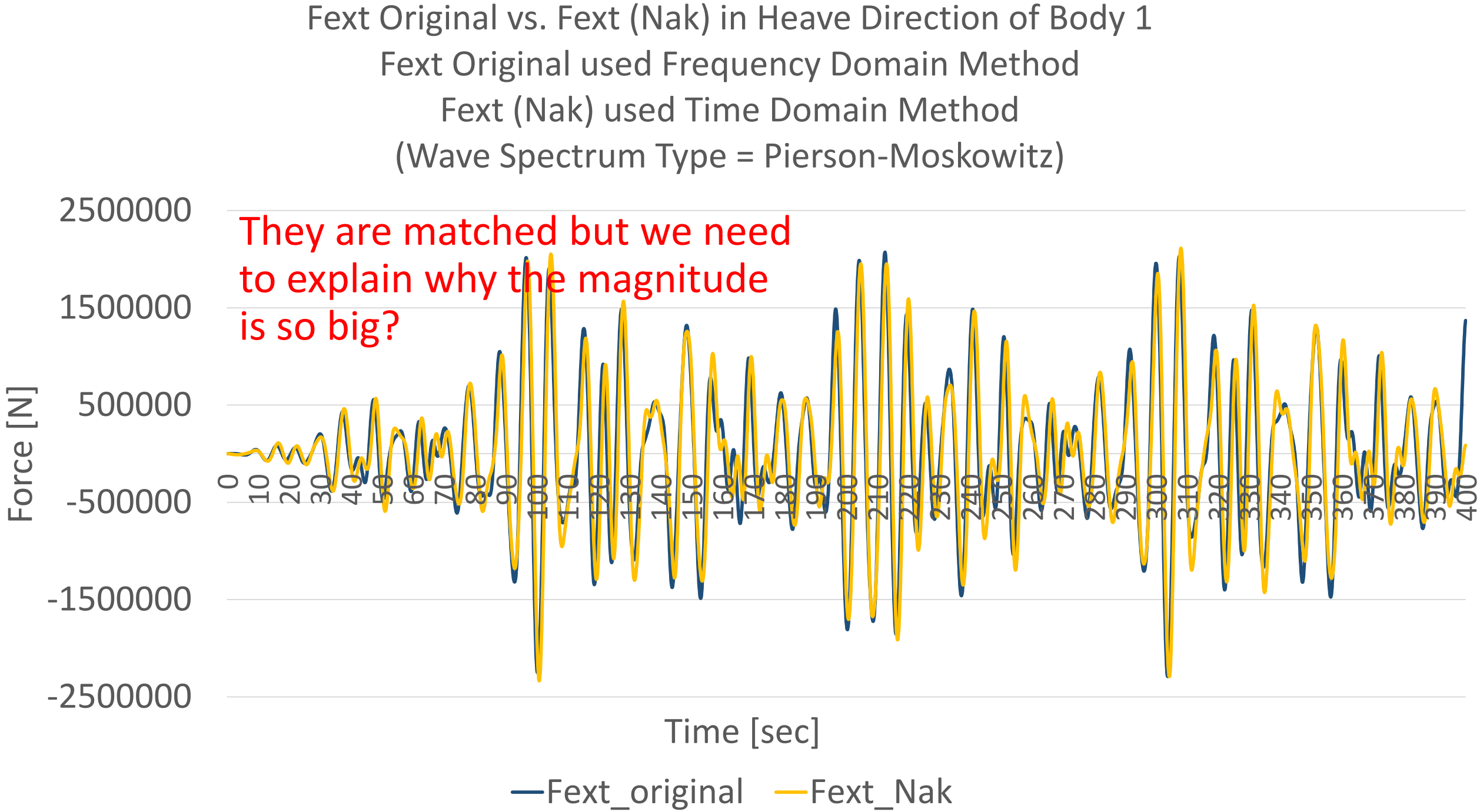
Fe_Nak

- Data from WEC-Sim
  - Fe from WAMIT (fExt.re & fExt.im)
  - Frequency (rad/s)
  - Fe(t) % for comparison
  - eta(t)
- $F_{ext}(t) = \eta(t) * f_e\ (t)$
- $F_{ext}(t) = \int_{-\infty}^{\infty} \eta(\tau) f_e(t-\tau) d\tau$
- Assumption(s):
  - eta(t) = waves.waveAmpTime

Fe_Original

- $F_{ext}(\omega_r) = \Re\left[R_f F_x(\omega_r) e^{i(\omega_r t+\phi)} \sqrt{\int_0^{\infty} 2S(\omega_r) d\omega}\right]$
- $F_{ext}(t) =$ …


Fext in Heave Direction of Body 1
(Regular Wave)
Force [N]
2500000
2000000
1500000
1000000
500000
0
-500000
-1000000
-1500000
-2000000
-2500000
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
Time [sec]


Fext Original vs. Fext (Nak) in Heave Direction of Body 1
Fext Original used Frequency Domain Method
Fext (Nak) used Time Domain Method
(Wave Spectrum Type = Pierson-Moskowitz)
They are matched but we need to explain why the magnitude is so big?
2500000
1500000
500000
-500000
-1500000
-2500000
Force [N]
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
Time [sec]
Fext_original
Fext_Nak

Excitation Force/Power Estimation

- Following McCormick:
  - Energy/(meter crest) = $\rho g^2 H^2 T^2 / 16\pi$ [J/m] (deep water approximation)
  - J/m = Nm/m = N
  - T = 8s, H=2.5m => 7.6582e+05 Nm/m
  - Body is 20m wide, => 1.5316e+07 N of excitation force hitting body.

- Fe reported from WEC-Sim = 1.88e6
- Is energy equation applicable?

Issue 1: class type conflict in simulationClass

Issue Description

- Date Reported: 7/28/2014
- Description: Matlab editor warning about a class usage conflict in the file ‘simulationClass.m’.
- Line 16:

```
classdef simulationClass<handle
```

- Line 152:

```
function obj =
set.simMechanicsFile(obj,value)
```

- Warning details indicate this conflict can cause code analyzer to be confused.

- See warning details for recommendations on corrections.

Issue Status

- Status: Reported
- Priority: Low
- Date Updated: 7/28/2014
- Status Details:

Issue 2: Hanging debug error messages in simulationClass.m

Issue Description

- Date Reported: 7/28/2014
- 4 function definitions in ‘simulationClass.m’ are either obsolete or broken. The ‘value’ in the error message is non-existent
- Each has the format:

```
function logFile = get.logFile(obj)
  if exist(obj.simMechanicsFile,'file') ~=4
    error('The simMechanics file, %s, does not exist in the case
directory',value)
  end
  logFile = [obj.caseDir filesep 'output' filesep
obj.simMechanicsFile(1:end-4) '_simulationLog.txt'];
end
```

- If error messages are kept, ‘value’ should be replaced with ‘obj.simMechanicsFile, and the ‘if’ statement should be promoted to outside the 4 function definitions.
- If error messages are not needed, they should be removed from the code.

Issue Status

- Status: Reported
- Priority: Low
- Status Details:

Issue 3: simulationClass ‘time’ parameter confusion

Issue Description

- Date Reported: 7/28/2014
- Description: ‘time’ parameter in ‘simulationClass.m’ currently a protected parameter, and set to 0.
- Assigned actual time-series values later in code.
- Should it be made a dependent parameter instead of a protected one?

Issue Status

- Status: Reported
- Priority: Low
- Date Updated:
- Status Details:

Issue

Issue Description

- Date Reported:
- Description:

Issue Status

- Status:
- Date Updated:
- Status Details: