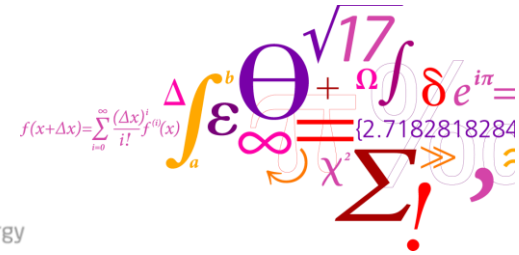
HAWC2

Hydrodynamic modeling



Risø DTU
National Laboratory for Sustainable Energy

Hydrodynamic model in HAWC2

Hydrodynamic kinematics:

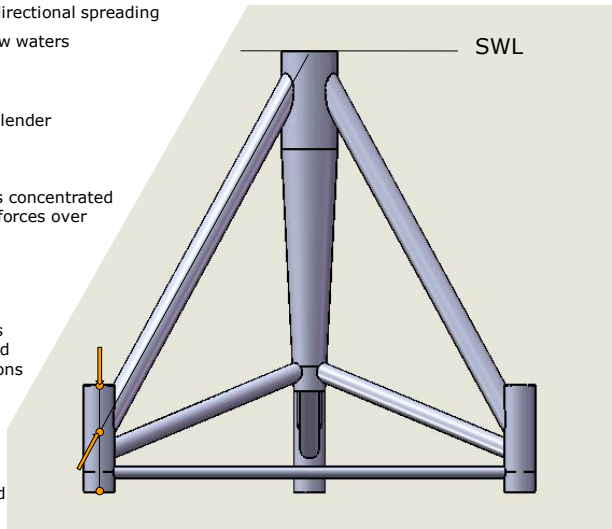
- Regular and iregular Airy waves, directional spreading
- Wheeler stretching used for shallow waters

Hydrodynamic loads:

- Morisons formula (assumption of slender elements)
- Axial damping term in end nodes
- Axial dynamic pressure inserted as concentrated force on end nodes and distributed forces over conical sections

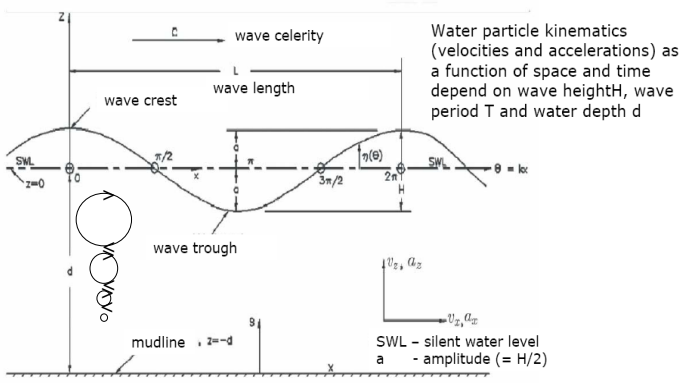
Buoyancy loads from static pressure:

- Axial dynamic pressure inserted as concentrated force on end nodes and distributed forces over conical sections
- Distributed perpendicular force contribution on angled elements
- Restoring moments distribution on conical sections
- Influence of flooded water included



Wave kinematics

Linear wave



Water particle kinematics (velocities and accelerations) as a function of space and time depend on wave heightH, wave period T and water depth d

Linear wave

The elevation is described as a sinus function

$$\eta_i(t,x) = A_i \sin\left(\omega_i t - k_i x + \varphi_i\right)$$

Where the relation between wave number and frequency is given by the dispersion relation

$$\omega_i^2 = g k_i \tanh(k_i z_0)$$

The horizontal and vertical velocities

$$u_i(t,x,z) = \omega_i \frac{\cosh\left[k_i\left(z+z_0\right)\right]}{\sinh\left[k_i z_0\right]} \eta(t,x) \qquad w_i(t,x,z) = \omega_i \frac{\sinh\left[k_i\left(z+z_0\right)\right]}{\sinh\left[k_i z_0\right]} A_i \cos\left(\omega_i t - k_i x + \varphi_i\right)$$

Linear wave (2)

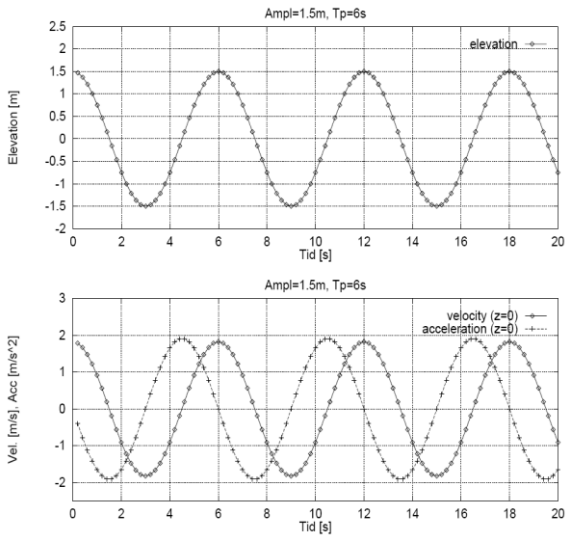
The particle accelerations

$$\dot{u}_i(t,x,z) = \omega_i^2 \frac{\cosh[k_i(z+z_0)]}{\sinh[k_i z_0]} A_i \cos(\omega_i t - k_i x + \varphi_i) \qquad \dot{w}_i(t,x,z) = -\omega_i^2 \frac{\sinh[k_i(z+z_0)]}{\sinh[k_i z_0]} \eta(t,x)$$

And the dynamic variation in pressure

$$p_i(t,x,z) = \rho g \frac{\cosh[k_i(z+z_0)]}{\cosh[k_i z_0]} \eta(t,x)$$

Linear wave (3)



Wheeler stretching

The velocity profile is extrapolated to the actual surface level by a stretching technique known as Wheeler stretching.

$$z' = \frac{z+d}{1+\frac{\eta(t)}{d}}$$

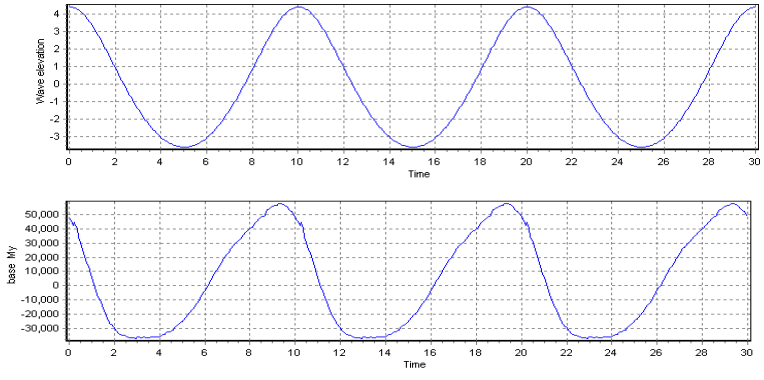
This is inserted in the calculation of velocities instead of z+d. It ensures that the velocities calculated at z=0 without stretching is inserted on the structure at the elevation level.

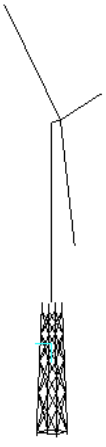
$$u_i(t,x,z) = \omega_i \frac{\cosh[k_i(z')]}{\sinh[k_i z_0]} \eta(t,x)$$

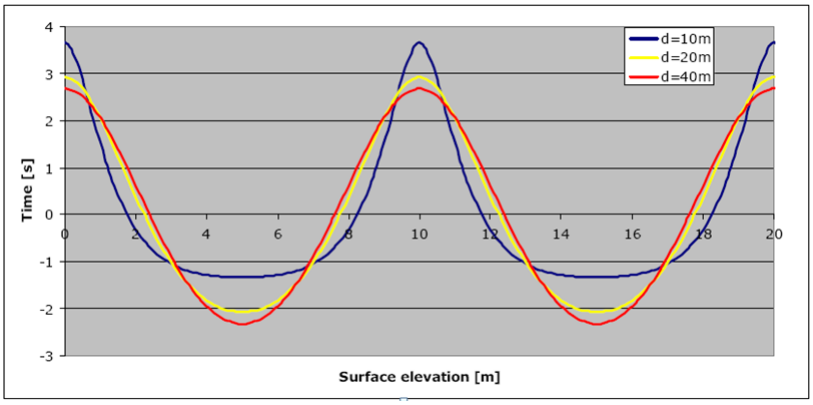
Stream function wave

- Stream function waves now added to the wave kinematics generator. (Missing feature for a long time).
- Method by Chaplin, Southhampton University
  - Dyn. pressure set to zero so far.

Stream function wave at 44m depth. H=8m, T=10s

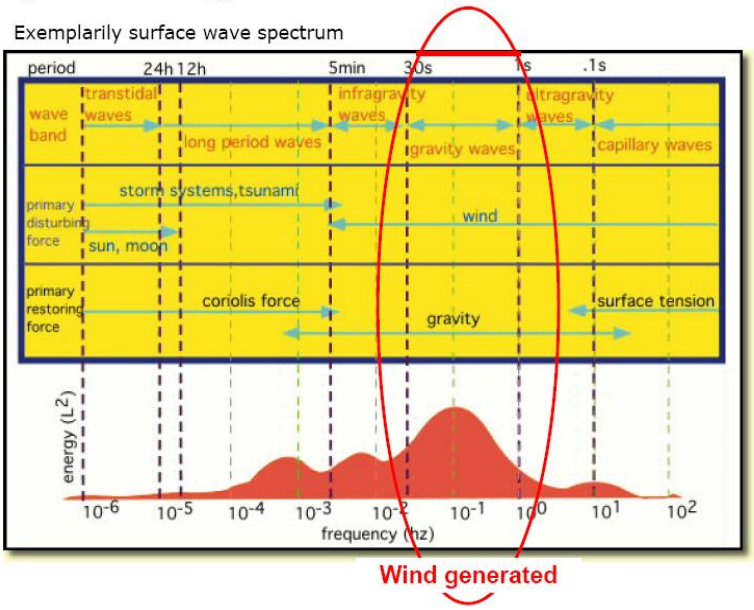






Irregular Waves(I)

Exemplarily surface wave spectrum

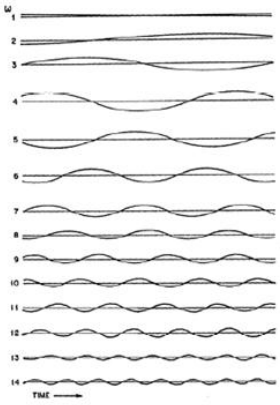


Water kinematics, iregular with spreading directions

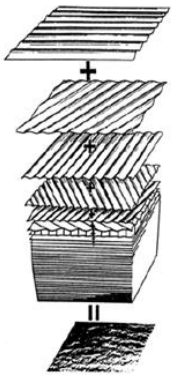
$$\alpha_i = \omega_i t - k_i x \cos\theta_i - k_i y \sin\theta_i + \varphi_i \qquad \omega_i^2 = g k_i \tanh(k_i z_0)$$

$$\eta_i(t,x,y) = A_i \sin\alpha_i \qquad \eta(t,x,y) = \sum_{i=1}^{N} \eta_i(t,x,y)$$

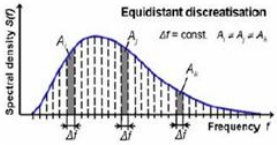
Individual linear waves



Superposition (different directions)



Wave spectrum



1. Discretization
2. Single wave amplitudes from spectrum energy
3. Random phases
4. (Directions)
5. Superposition

Jonswap spectrum for iregular waves:

$$G(\omega) = \frac{\alpha \cdot g^2}{\omega^5} \cdot \exp\left[-\frac{5}{4}\cdot\left(\frac{\omega_p}{\omega}\right)\right]^4 \gamma^{\exp\left(-\frac{(\omega-\omega_p)^2}{2\cdot\sigma^2\cdot\omega_p^2}\right)}$$

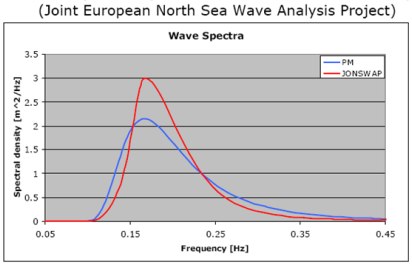
$$\alpha = \frac{5}{16}\frac{H_s^2 \omega_p^4}{g^2}(1 - 0.287\ln(\gamma))$$

$$\omega_p = \frac{2\pi}{T_p}$$

σ=0.07 for $\omega \le \omega_m$, σ= 0.09 for $\omega > \omega_m$.

Fully developed sea-state: (Fetch, Duration) — Pierson-Moskowitz spectrum

Not fully developed sea-state: (Joint European North Sea Wave Analysis Project) — JONSWAP spectrum



Exemplary wave spectra for equal H_s and T_p

H_s = 2.0 m

T_p = 6.0 s

γ (JONSWAP) =3.3

Water kinematics, iregular with spreading directions

$$\alpha_i = \omega_i t - k_i x \cos\theta_i - k_i y \sin\theta_i + \varphi_i \qquad \omega_i^2 = g k_i \tanh(k_i z_0)$$

$$\eta_i(t,x,y) = A_i \sin\alpha_i$$

$$\eta(t,x,y) = \sum_{i=1}^{N} \eta_i(t,x,y)$$

velocity

$$u_i(t,x,y,z) = \omega_i \frac{\cosh[k_i(z+z_0)]}{\sinh[k_i z_0]} A_i \sin\alpha_i \cos\theta_i \qquad u(t,x,y,z) = \sum_{i=1}^{N} u_i(t,x,y,z)$$

$$v_i(t,x,y,z) = \omega_i \frac{\cosh[k_i(z+z_0)]}{\sinh[k_i z_0]} A_i \sin\alpha_i \sin\theta_i \qquad v(t,x,y,z) = \sum_{i=1}^{N} v_i(t,x,y,z)$$

$$w_i(t,x,y,z) = \omega_i \frac{\sinh[k_i(z+z_0)]}{\sinh[k_i z_0]} A_i \cos\alpha_i \qquad w(t,x,y,z) = \sum_{i=1}^{N} w_i(t,x,y,z)$$

acceleration

$$\dot{u}_i(t,x,y,z) = \omega_i^2 \frac{\cosh[k_i(z+z_0)]}{\sinh[k_i z_0]} A_i \cos\alpha_i \cos\theta_i \qquad \dot{u}(t,x,y,z) = \sum_{i=1}^{N} \dot{u}_i(t,x,y,z)$$

$$\dot{v}_i(t,x,y,z) = \omega_i^2 \frac{\cosh[k_i(z+z_0)]}{\sinh[k_i z_0]} A_i \cos\alpha_i \sin\theta_i \qquad \dot{v}(t,x,y,z) = \sum_{i=1}^{N} \dot{v}_i(t,x,y,z)$$

$$\dot{w}_i(t,x,y,z) = -\omega_i^2 \frac{\sinh[k_i(z+z_0)]}{\sinh[k_i z_0]} A_i \sin\alpha_i \qquad \dot{w}(t,x,y,z) = \sum_{i=1}^{N} \dot{w}_i(t,x,y,z)$$

pressure

$$p_i(t,x,y,z) = \rho g \frac{\cosh[k_i(z+z_0)]}{\cosh[k_i z_0]} A_i \sin\alpha_i \qquad p(t,x,y,z) = \sum_{i=1}^{N} p_i(t,x,y,z)$$

Directional spreading

The directional spreading distribution function

$$f(\theta) = K_{2s} \cos^{2s}\theta \qquad K_{2s} = \frac{2^{2s-1} s!(s-1)!}{\pi(2s-1)!}$$

$s := 2$

$$K2s := \frac{2^{(2\cdot s-1)}\cdot s!\cdot(s-1)!}{\pi\cdot(2\cdot s-1)!}$$

The integrated function is calculated

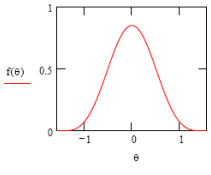
$$F(\theta) = \int_{-\frac{\pi}{2}}^{\theta} f(\theta)\,d\theta$$

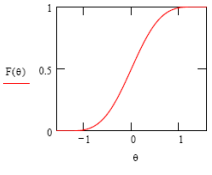
$$f(\theta) := K2s\cdot(\cos(\theta))^{2\cdot s}$$

$$F(\theta) := \int_{-\frac{\pi}{2}}^{\theta} f(\theta)\,d\theta$$

The inverse of this function is calculated so theta can be found with basis of F value.

The uniformly distributed phase angles are used as input to find a corresponding direction angle for each coefficient.

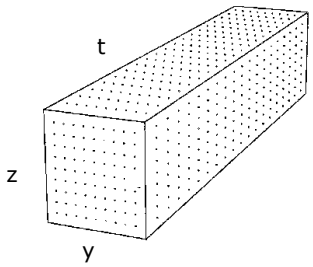




Wave kinematics

- In the previous version all fourrier summations were done in the exact position and time of lookup.
- Now a pregenerated field is created. 3 dimensions when spreading is included. Only height and time resolved for 2D waves.
- 15 points in the height is used. Time resolution is $T_{min}/10$.
- Linear interpolation is used.

- A 10 min time series on a jacket that used to take 2hours for simulations, is now ready after 12minutes!

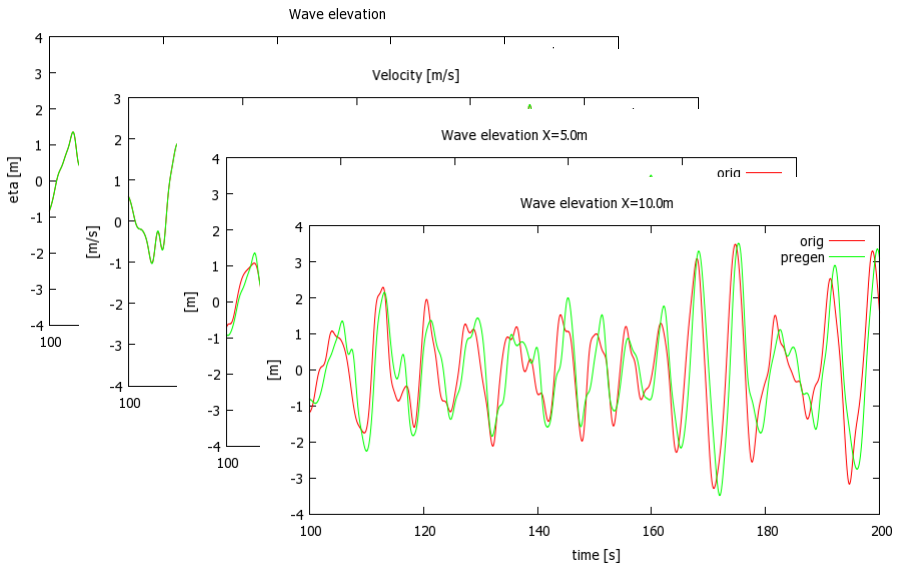


The assumed relation between time and x pos is based on the group velocity of waves.

$$C_G = \frac{1}{2}\sqrt{\frac{g}{k}}\left[\frac{kh + \tanh(kh) - kh\,tanh^2(kh)}{\sqrt{\tanh(kh)}}\right]$$

$$t_{lookup} = t - \frac{x}{C_G}$$

Comparison



Morison formula

For flexible slender structures D/L<0.2

$$dF = \rho A\dot{U} + \rho C_a A_R \dot{U}_{rel} + \tfrac{1}{2}\rho D C_d U_{rel}\left|U_{rel}\right|$$

Froud-Krylov force — **water added mass** — **drag force**

Which for stiff slender structures decouples to

$$dF = \rho \tfrac{\pi D^2}{4} C_M \dot{U} + \tfrac{1}{2}\rho D C_d U_{rel}\left|U_{rel}\right|, \qquad C_M = 1 + C_a$$

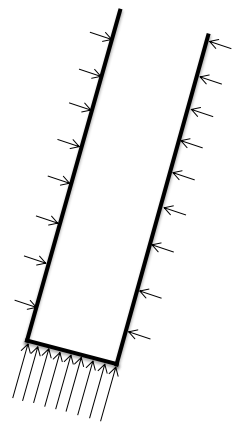
- hydrodynamic drag (cd) and inertia (cm=1+ca) coefficients depend on surface roughness, Reynolds number & Keulegan-Carpenter number
- hydrodynamic damping from drag term relevant only for very flexible structures (e.g. Riser, cable)

Buoyancy model in HAWC2

Boyancy implemented as the result of integration of external pressure

- Buoyancy, distributed load contributions:

$$\vec{F}_b = -g\rho \begin{Bmatrix} A_{3,1}S \\ A_{3,2}S \\ -\frac{\partial S}{\partial z}(z - z_0) + \frac{\partial S}{\partial z} p_{dyn} \end{Bmatrix} \qquad \vec{M}_b = -g\rho \begin{Bmatrix} -A_{3,2}\frac{\partial r}{\partial z}\pi r^3 \\ A_{3,1}\frac{\partial r}{\partial z}\pi r^3 \\ 0 \end{Bmatrix}$$



- Buoyancy and drag contributions at end nodes

$$F_{b,end}(3) = \rho g S(z - z_0) + S\, p_{dyn} + \frac{1}{2}\rho C_{d,axial} S\, u_{rel}|u_{rel}|$$

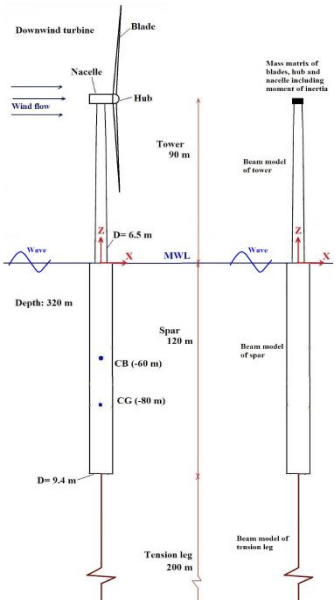
Static pressure — Dynamic pressure — Viscous drag

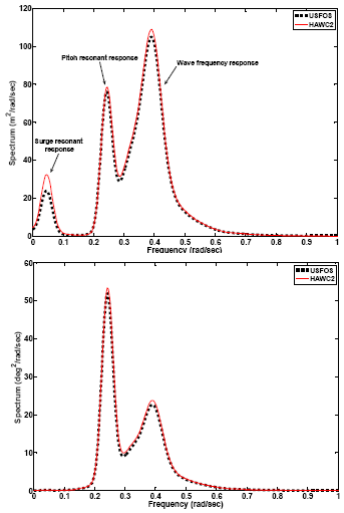
A : orientation matrix
S : area
r : radius

g : gravity
P : pressure
ρ : density

z : vertical pos
u : velocity
dr/dz : conicity

Validation for a spar bouy



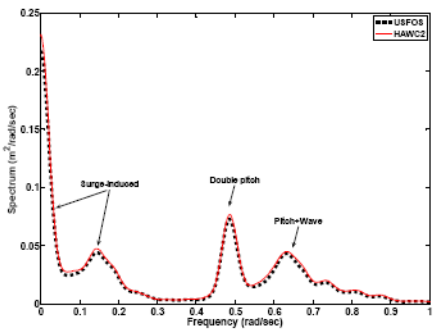


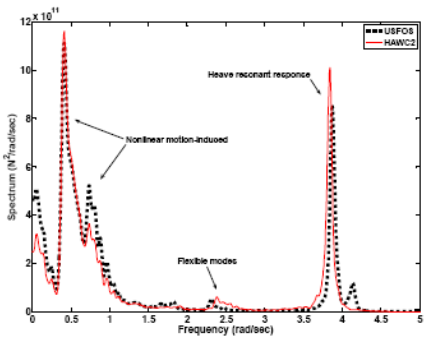
Surge

Pitch

Madjid Karimirad, Quentin Meissonnier, Zhen Gao, Torgeir Moan. HYDROELASTIC CODE-TO-CODE COMPARISON FOR A TENSION LEG SPAR-TYPE FLOATING WIND TURBINE. Submitted to Marine Structures journ. 2011.

Spar bouy (2)



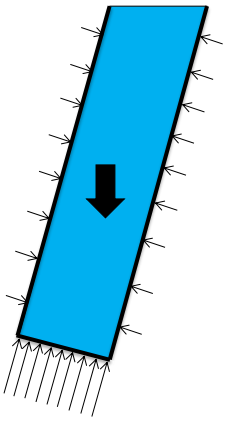


Heave

Tension

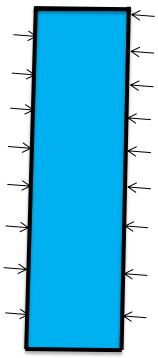
What about flooded members?

Important for piles in the jacket



- Classical approach in existing aeroelastic codes
  - Internal water applied as extra ”steel” mass to ensure correct inertia
- This is not correct since eg. gravity force will accumulate wrongly in the steel structure
- Another method is proposed that is valid for flooded members in compartments without free inner surfaces.

Flooded members – added mass



We start with the Morisons formula and include the flooded water

$$F_{waves} = m_{steel}\dot{u}_{cyl} + m_{water}\dot{u}_{cyl}$$

$$\Rightarrow \rho A\dot{u} + \rho C_m A_R \dot{u}_{rel} + \tfrac{1}{2}\rho D C_d u_{rel}\left|u_{rel}\right| = m_{steel}\dot{u}_{cyl} + \rho A_i \dot{u}_{cyl}$$

$$\dot{u}_{rel} = \dot{u} - \dot{u}_{cyl}$$

$$\rho(A - A_i)\dot{u} + \rho(C_m A_R + A_i)\dot{u}_{rel} + \tfrac{1}{2}\rho D C_d u_{rel}\left|u_{rel}\right| = m_{steel}\dot{u}_{cyl}$$

And end with a new modified Morisons formula for flooded members

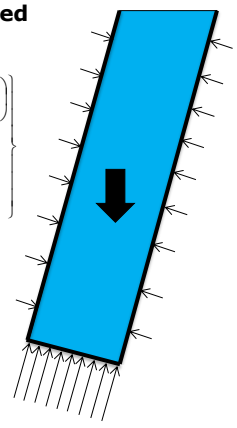
The benefit is an added mass not affected by gravity.

Buoyancy for flooded members

- **The boyancy is still a result of integration of external pressure**
- **The flooded water will apply an internal pressure at same location as outer pressure from water**
- **The result is that the inner area should be subtracted**

  - Buoyancy, distributed load contributions:

$$\vec{F}_b = -g\rho \begin{Bmatrix} A_{3,1}(S-S_i) \\ A_{3,2}(S-S_i) \\ -\frac{\partial S}{\partial z}(z-z_0)+\frac{\partial S_i}{\partial z}(z-z_0)+\frac{\partial S}{\partial z}p_{dyn} \end{Bmatrix} \qquad \vec{M}_b = -g\rho \begin{Bmatrix} -A_{3,2}\pi\left(\frac{\partial r}{\partial z}r^3-\frac{\partial r_i}{\partial z}r_i^3\right) \\ A_{3,1}\pi\left(\frac{\partial r}{\partial z}r^3-\frac{\partial r_i}{\partial z}r_i^3\right) \\ 0 \end{Bmatrix}$$



  - Buoyancy and drag contributions at end nodes

$$F_{b,end}(3) = \rho g(S-S_i)(z-z_0) + S\,p_{dyn} + \tfrac{1}{2}\rho C_{d,axial} S u_{rel}|u_{rel}|$$

Static pressure

Dynamic pressure

Viscous drag

Also applied at inner nodes when jump in plate thickness occurs

g : gravity
P : pressure
ρ : density

Dynamic pressure assumed not to change inner pressure (depends on design though)

Improvements of solver

A.M. Hansen

- External hydro added mass derived as an analytical integration of the external load.
- Separation of effects from large rotations/movements and local deformation.

Position of a point on a structure

$$\mathbf{u} = \mathbf{R} + \mathbf{A}(\mathbf{r_c} + \mathbf{N_c q})$$

Acceleration of point

$$\ddot{\mathbf{u}} = \ddot{\mathbf{R}} - \mathbf{A}\left[\{\mathbf{r_c}\}\times\mathbf{I}\right]\dot{\bar{\omega}} + \mathbf{A}\mathbf{N_c}\ddot{\mathbf{q}}$$

External force

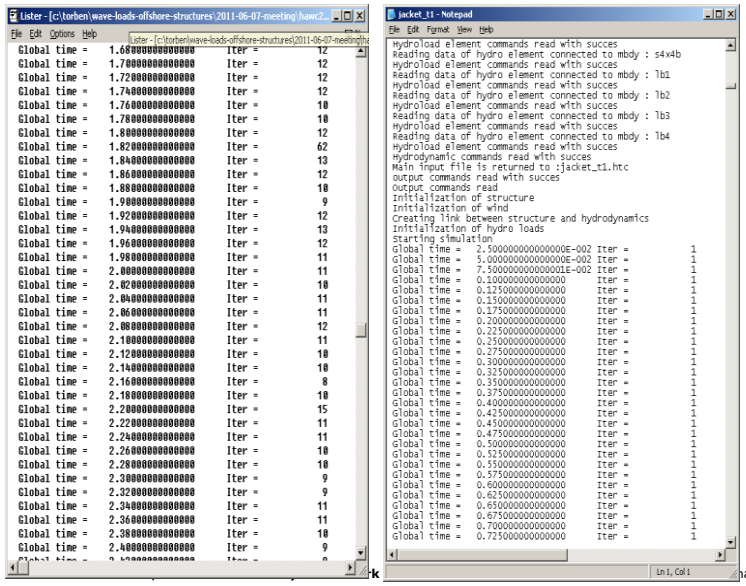
$$\mathbf{Q} = -\mathbf{A}\mathbf{T_{AS}}\mathbf{C_M}\mathbf{T_{AS}}^T\mathbf{A}^T\ddot{\mathbf{u}}$$

$\mathbf{C}_M$ is the section added mass matrix

$$\mathbf{M_A} = \int_L \begin{bmatrix} \mathbf{A}\mathbf{C}_\mathbf{M}^\mathbf{S}\mathbf{A}^T & \mathbf{A}\mathbf{C}_\mathbf{M}^\mathbf{S}\left[\{\mathbf{r_c}\}\times\mathbf{I}\right] & \mathbf{A}\mathbf{C}_\mathbf{M}^\mathbf{S}\mathbf{N_c} \\ & -\left[\{\mathbf{r_c}\}\times\mathbf{I}\right]\mathbf{C}_\mathbf{M}^\mathbf{S}\left[\{\mathbf{r_c}\}\times\mathbf{I}\right] & \left[\{\mathbf{r_c}\}\times\mathbf{I}\right]\mathbf{C}_\mathbf{M}^\mathbf{S}\mathbf{N_c} \\ sym & & \mathbf{N_c}^T\mathbf{C}_\mathbf{M}^\mathbf{S}\mathbf{N_c} \end{bmatrix} dz$$

where $\mathbf{C}_\mathbf{M}^\mathbf{S} = \mathbf{T_{AS}}\mathbf{C_M}\mathbf{T_{AS}}^T$. Since $\mathbf{A}$ is part of the added mass matrix and $\mathbf{A}$ is time dependent, the added mass matrix also becomes time dependent, however, $\mathbf{A}$ is the *only* time dependent part of the matrix. This means that the added mass matrix have to be updated each time step, but only by pre- and post multiplication by $\mathbf{A}$ - the remainder of the matrix is integrated only once.

Better convergence from version 10.4



HAWC2 example

```
begin HYDRO ;
   begin WATER_PROPERTIES ;
    rho 1025 ; [kg/m^3]
    gravity 9.816 ; [m/s^2]
    mwl 0.000; [m]
    mudlevel 14.500 ; [m]
    water_kinematics_dll ./wkin_dll.dll   ./hydrohtc/reg_airy_1.htc ;
   end WATER_PROPERTIES  ;
;
 begin HYDRO_ELEMENT ;
    mbdy_name L1 ;
     buoyancy 1 ;
     update_states  1 ; (0: no dynamic interaction, 1: fully coupled solution
     hydrosections auto 4 ; dist, of hydro calculation points from 1 to nsec
    nsec  9 ;  z       Cm   Cd    A          Aref        width dr/dz Cd_a_(quad) Cm_a Cd_a_lin Ai
      sec     0.000   1    1     1.1309734    1.1309734   1.20  0.0 0.0 0.0 0.0     0.9503318 ;
      sec     5.005   1    1     1.1309734    1.1309734   1.20  0.0 0.0 0.0 0.0     0.9503318 ;
      sec     5.015   1    1     1.5393804    1.5393804   1.40  0.0 0.0 0.0 0.0     0.9503318 ;
      sec    20.408   1    1     1.5393804    1.5393804   1.40  0.0 0.0 0.0 0.0     0.9503318 ;
      sec    20.418   1    1     1.5393804    1.5393804   1.40  0.0 0.0 0.0 0.0     1.0028749 ;
      sec    43.046   1    1     1.5393804    1.5393804   1.40  0.0 0.0 0.0 0.0     1.0028749 ;
      sec    43.056   1    1     1.1309734    1.1309734   1.20  0.0 0.0 0.0 0.0     1.0028749 ;
      sec    49.431   1    1     1.1309734    1.1309734   1.20  0.0 0.0 0.0 0.0     1.0028749 ;
      sec    61.215   1    1     1.1309734    1.1309734   1.20  0.0 0.0 0.0 0.0     1.0028749 ;
   end HYDRO_ELEMENT ;
end HYDRO ;
```

Wkin_dll input file example, regular airy

```
begin wkin_input ;
  wavetype 0 ; 0=regular, 1=irregular, 2=deterministic
  wdepth 220.0 ;
;
  begin reg_airy ;
    stretching 0; 0=none, 1=wheeler
    wave 9 12.6;  Hs,T
  end;
end;
;
exit ;
```


Wkin_dll input file example, irregular airy

```
begin wkin_input ;
  wavetype 1 ; 0=regular, 1=irregular, 2=deterministic
  wdepth 220.0 ;
;
  begin ireg_airy ;
    stretching 0;   0=none, 1=wheeler
    spectrum   1;  (1=jonswap)
    jonswap    9 12.6 3.3 ; (Hs, Tp, gamma)
    coef       200 1 ; (coefnr, seed)
    spreading 1 2; (type(0=off 1=on), s parameter (pos. integer min 1)
  end;
end;
;
exit ;
```

Wkin_dll input file example, deterministic airy

```
begin wkin_input ;
  wavetype 2 ; 0=regular, 1=irregular, 2=deterministic
  wdepth 220.0 ;
;
  begin det_airy ;
    stretching 0; 0=none, 1=wheeler
    file ..\waves\elevation.dat ;
    nsamples 32768 ;
    nskip 1 ;
    columns 1 2 ; time column, elevation column
  end;
end;
exit ;
```

..\waves\elevation.dat file example

```
time, elevation
0.0   0.0
0.1   0.2
0.2   0.4
0.3   0.6
0.4   0.8
0.5   0.9
0.6   1.0
0.7   0.9
0.8   0.8
0.9   0.6
```


Wkin_dll input file example, stream function

```
begin wkin_input ;
  wavetype 3 ; 0=regular, 1=irregular, 2=deterministic, 3=stream function
  wdepth 40.0 ;
;
 begin strf ;
    wave 8.0 10.0 0.0 ;     Hs,T,current
  end;
end;
;
exit ;
```