

## HAWC2

Hydrodynamic modeling

$$f(x+\Delta x) = \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x)$$

$$\int_a^b \Theta + \Omega \int \delta e^{i\pi} = \sqrt{17} \int \infty \sum \gg \Sigma !$$

Risø DTU  
National Laboratory for Sustainable Energy

## Hydrodynamic model in HAWC2

### Hydrodynamic kinematics:

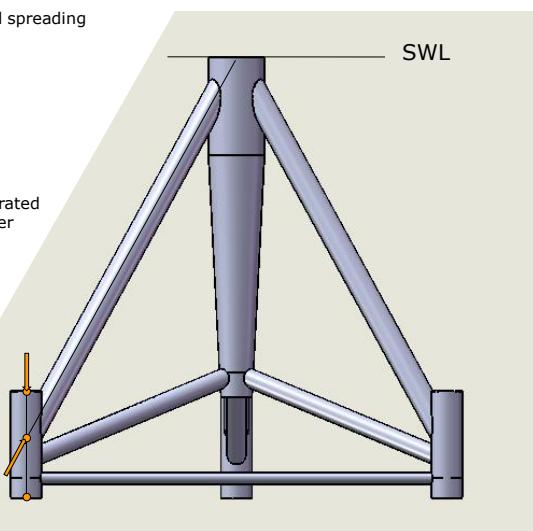
- Regular and irregular Airy waves, directional spreading
- Wheeler stretching used for shallow waters

### Hydrodynamic loads:

- Morison's 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

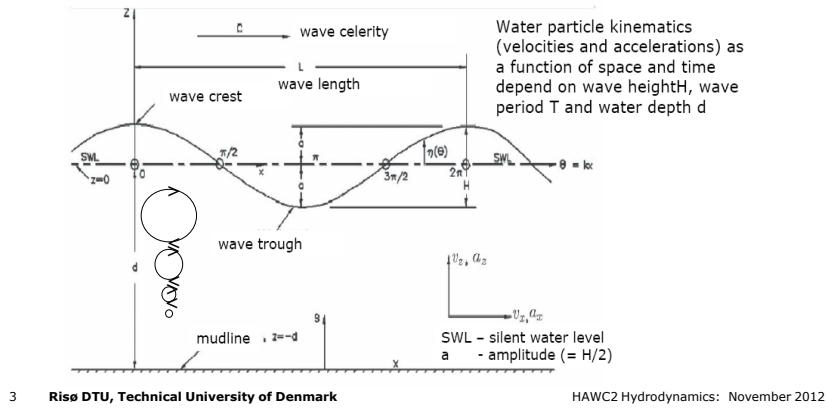
### Buoyancy loads from static pressure:

- 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



## Linear wave

The elevation is described as a sinus function

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

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

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

The horizontal and vertical velocities

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

## Linear wave (2)

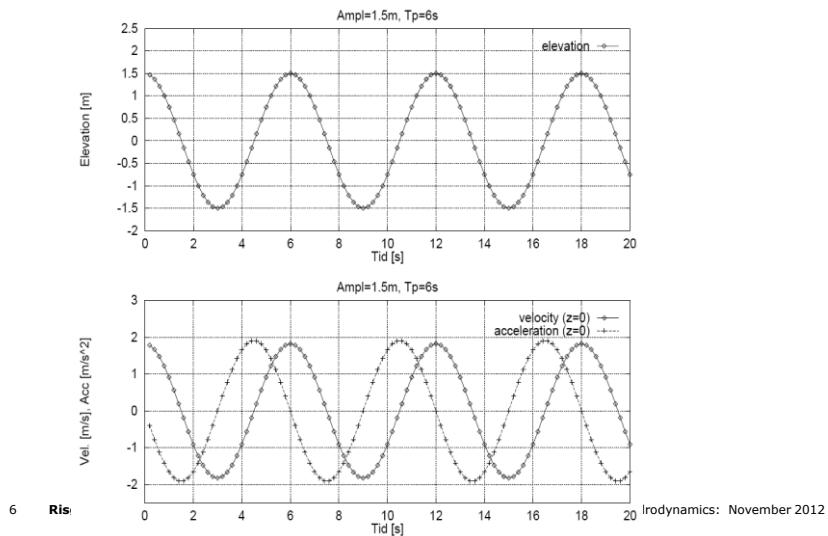
The particle accelerations

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

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.

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

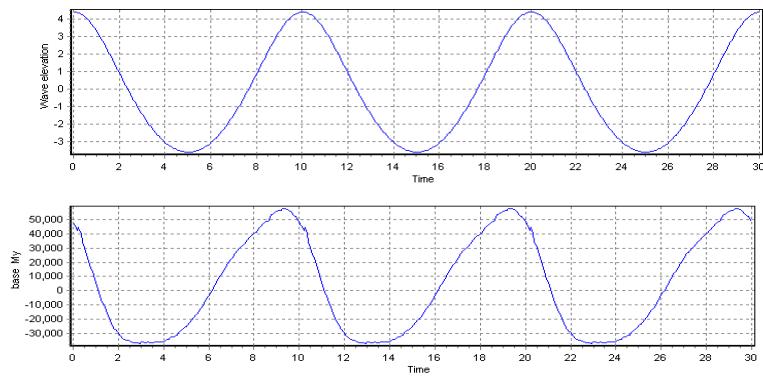
This is inserted in the calculation of velocities instead of  $z+d$ . It ensures that the velocities calculated at  $z=0$  without stretching is inserted on the structure at the elevation level.

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

## Stream function wave

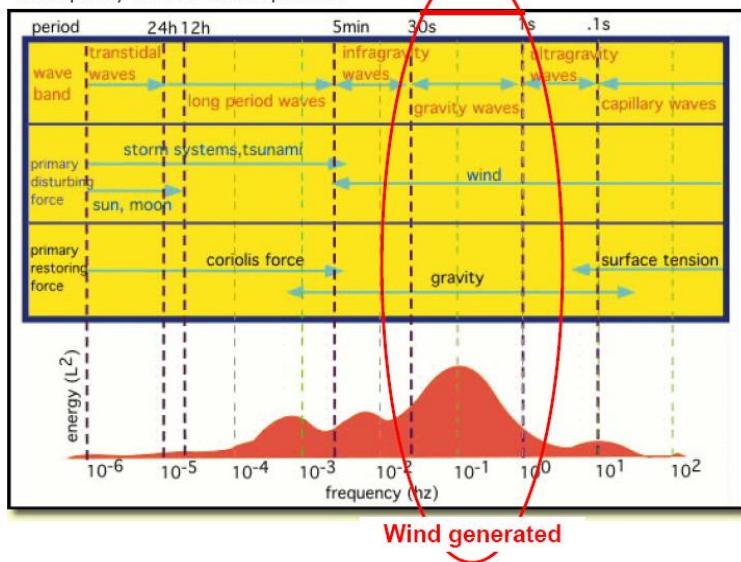
- Stream function waves now added to the wave kinematics generator. (Missing feature for a long time).
- Method by Chaplin, Southampton University
  - Dyn. pressure set to zero so far.

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



## Irregular Waves(I)

Exemplarily surface wave spectrum



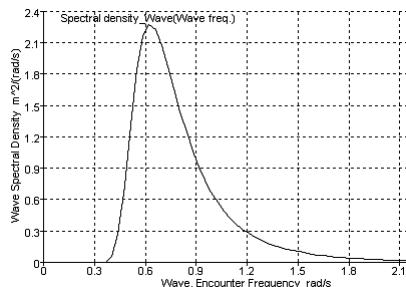
### Jonswap spectrum for irregular waves:

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

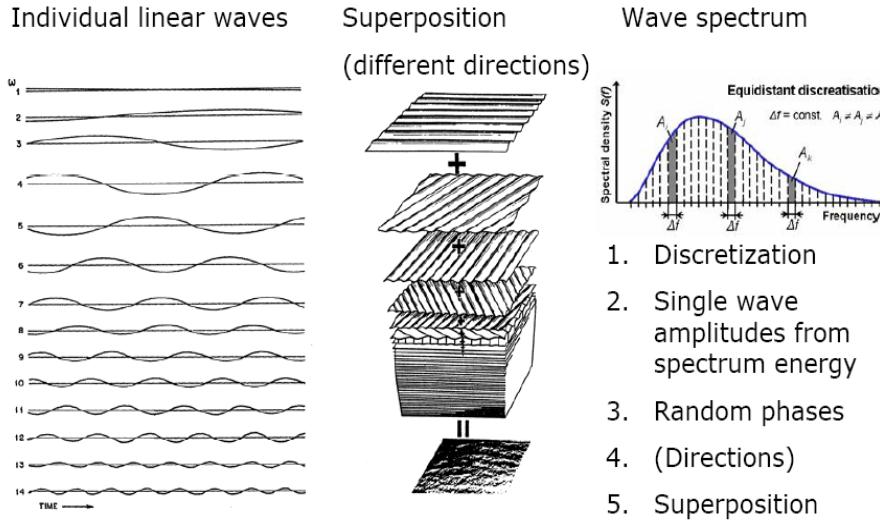
$$\alpha = \frac{5}{16} \frac{H_s^2 \omega_p^4}{g^2} (1 - 0.287 \ln(\gamma))$$

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

$\sigma=0.07$  for  $\omega \leq \omega_m$ ,  $\sigma=0.09$  for  $\omega > \omega_m$ .



## Irregular wave(III)



### Water kinematics, irregular with spreading directions



$$\alpha_i = \omega_i t - k_i x \cos \theta_i - k_i y \sin \theta_i + \varphi_i$$

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

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

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

$$u_i(t, x, y, z) = \omega_i \frac{\cosh[k_i(z + z_0)]}{\sinh[k_i z_0]} A_i \sin \alpha_i \cos \theta_i$$

$$u(t, x, y, z) = \sum_{i=1}^N u_i(t, x, y, z)$$

$$v_i(t, x, y, z) = \omega_i \frac{\cosh[k_i(z + z_0)]}{\sinh[k_i z_0]} A_i \sin \alpha_i \sin \theta_i$$

$$v(t, x, y, z) = \sum_{i=1}^N v_i(t, x, y, z)$$

$$w_i(t, x, y, z) = \omega_i \frac{\sinh[k_i(z + z_0)]}{\sinh[k_i z_0]} A_i \cos \alpha_i$$

$$w(t, x, y, z) = \sum_{i=1}^N w_i(t, x, y, z)$$

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

$$\dot{u}(t, x, y, z) = \sum_{i=1}^N \dot{u}_i(t, x, y, z)$$

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

$$\dot{v}(t, x, y, z) = \sum_{i=1}^N \dot{v}_i(t, x, y, z)$$

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

$$\dot{w}(t, x, y, z) = \sum_{i=1}^N \dot{w}_i(t, x, y, z)$$

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

$$p(t, x, y, z) = \sum_{i=1}^N p_i(t, x, y, z)$$

## Directional spreading

The directional spreading distribution function

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

$s := 2$

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

The integrated function is calculated

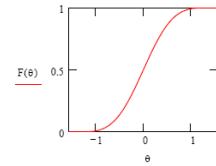
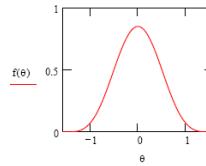
$$F(\theta) = \int_{-\frac{\pi}{2}}^{\theta} f(\theta') d\theta'$$

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

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

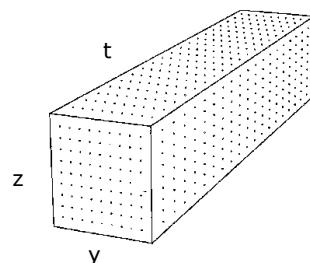
The inverse of this function is calculated so theta can be found with basis of F value.

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!

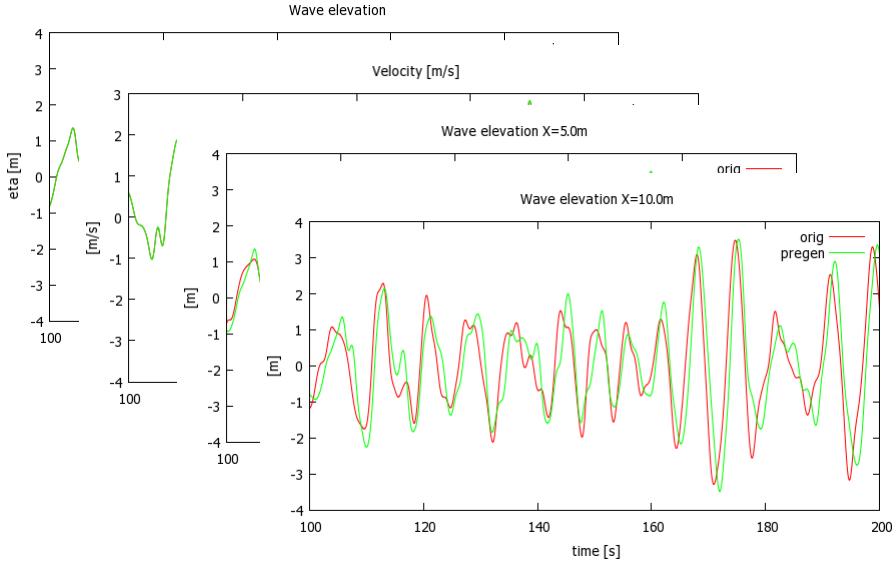


The assumed relation between time and x pos is based on the group velocity of waves.

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

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

## Comparison



15 Risø DTU, Technical University of Denmark

HAWC2 Hydrodynamics: November 2012

## Morison formula

For flexible slender structures D/L<0.2

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

Froude-Krylov force	water added mass	drag force
---------------------	------------------	------------

Which for stiff slender structures decouples to

$$dF = \rho \frac{\pi D^2}{4} C_M \dot{U} + \frac{1}{2} \rho D C_d U_{rel} |U_{rel}|, \quad C_M = 1 + C_a$$

- hydrodynamic drag ( $C_d$ ) and inertia ( $C_m = 1 + C_a$ ) 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)

CS: November 2012

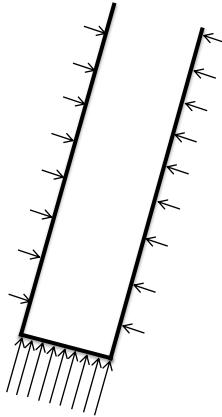
## Buoyancy model in HAWC2

Boyancy implemented as the result of integration of external pressure

- Buoyancy, distributed load contributions:

$$\vec{F}_b = -g\rho \begin{Bmatrix} A_{31}S \\ A_{32}S \\ -\frac{\partial S}{\partial z}(z - z_0) + \frac{\partial S}{\partial z} p_{dyn} \end{Bmatrix}$$

$$\vec{M}_b = -g\rho \begin{Bmatrix} -A_{32} \frac{\partial r}{\partial z} \pi r^3 \\ A_{31} \frac{\partial r}{\partial z} \pi r^3 \\ 0 \end{Bmatrix}$$



- 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

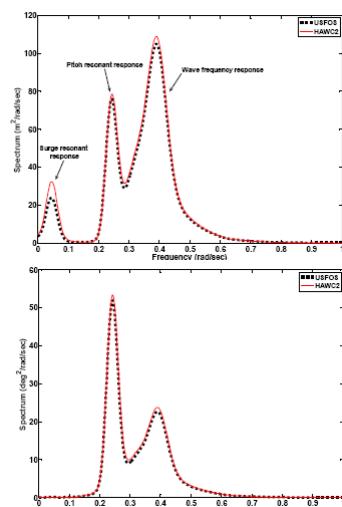
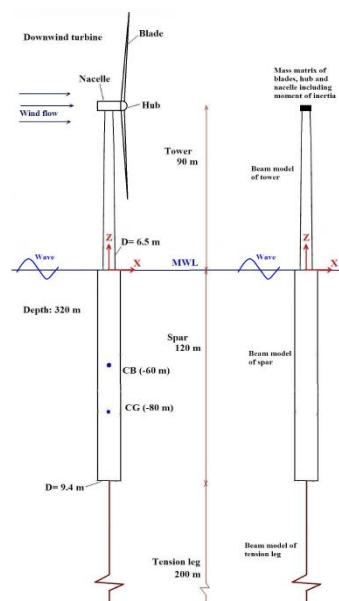
g : gravity  
P : pressure  
ρ : density

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

17 Risø DTU, Technical University of Denmark

HAWC2 Hydrodynamics: November 2012

## Validation for a spar bouy

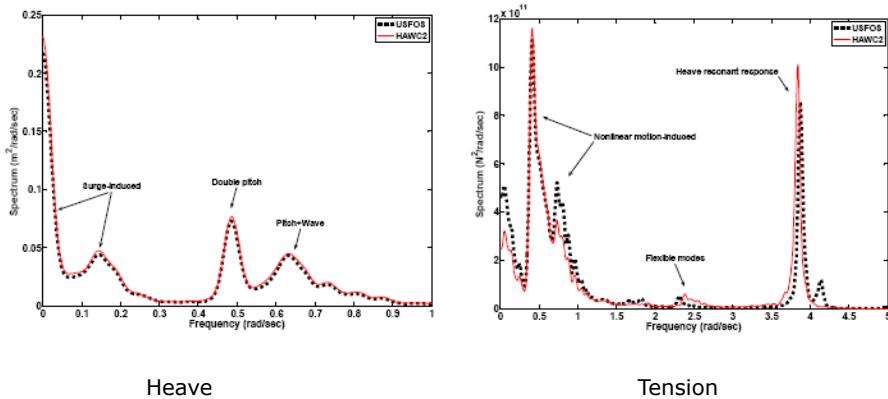


Surge

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 buoy (2)

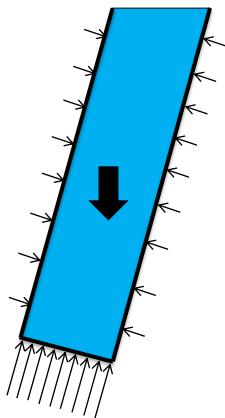


19 Risø DTU, Technical University of Denmark

HAWC2 Hydrodynamics: November 2012

## What about flooded members?

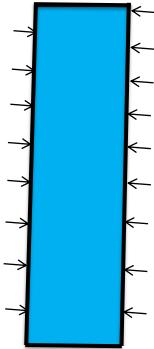
Important for piles in the jacket



- 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



$$\begin{aligned}
 F_{waves} &= m_{steel} \dot{u}_{cyl} + m_{water} \dot{u}_{cyl} \\
 \Rightarrow \rho A \dot{u} + \rho C_m A_R \dot{u}_{rel} + \frac{1}{2} \rho D C_d u_{rel} |u_{rel}| &= m_{steel} \dot{u}_{cyl} + \rho A_i \dot{u}_{cyl} \\
 \dot{u}_{rel} &= \dot{u} - \dot{u}_{cyl} \\
 \rho (A - A_i) \dot{u} + \rho (C_m A_R + A_i) \dot{u}_{rel} + \frac{1}{2} \rho D C_d u_{rel} |u_{rel}| &= m_{steel} \dot{u}_{cyl}
 \end{aligned}$$

and end up with a new kind of modified Morisons formula for flooded members

The benefit is an added mass not affected by gravity.

## Buoyancy for flooded members

- The buoyancy 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:

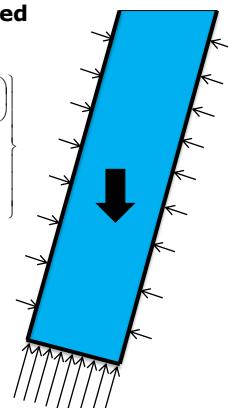
$$\vec{F}_b = -g\rho \left\{ \begin{array}{l} A_{3,1}(S - S_i) \\ A_{3,2}(S - S_i) \\ -\frac{\partial S}{\partial z}(z - z_0) + \frac{\partial S_i}{\partial z}(z - z_0) + \frac{\partial S}{\partial z} p_{dyn} \end{array} \right\} \quad \vec{M}_b = -g\rho \left\{ \begin{array}{l} -A_{3,2}\pi \left( \frac{\partial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

- Buoyancy and drag contributions at end nodes

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

Also applied at inner nodes when jump in plate thickness occurs

$g$  : gravity  
 $P$  : pressure  
 $\rho$  : 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.

$C_M$  is the section added mass matrix

$$M_A = \int_L \begin{bmatrix} AC_M^S A^T & AC_M^S [\{r_c\} \times I] & AC_M^S N_c \\ -[\{r_c\} \times I] C_M^S [\{r_c\} \times I] & [C_M^S [\{r_c\} \times I]]^T C_M^S N_c \\ N_c^T C_M^S N_c \end{bmatrix} dz$$

where  $C_M^S = T_{AS} C_M T_{AS}^T$ . Since  $A$  is part of the added mass matrix and  $A$  is time dependent, the added mass matrix also becomes time dependent, however,  $A$  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  $A$  - the remainder of the matrix is integrated only once.

23 Risø DTU, Technical University of Denmark

HAWC2 Hydrodynamics: November 2012

Position of a point on a structure

$$u = R + A(r_c + N_c q)$$

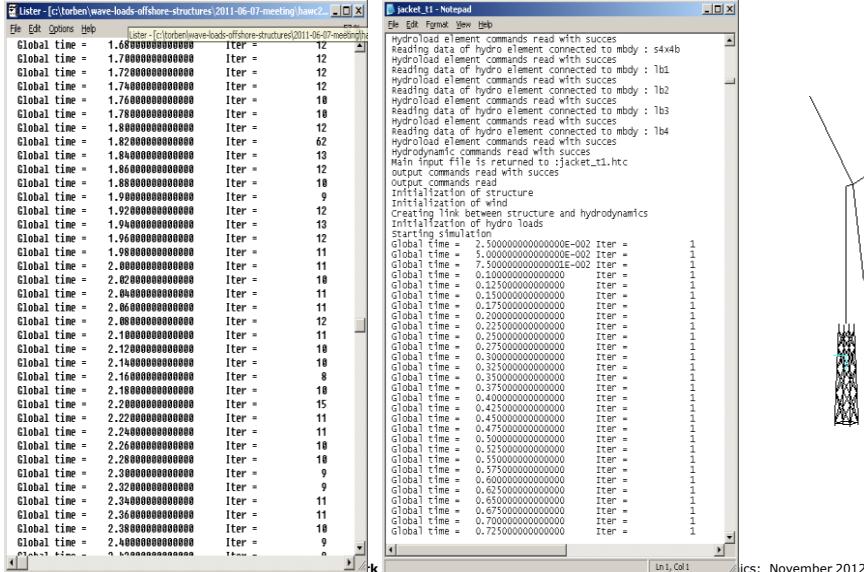
Acceleration of point

$$\ddot{u} = \ddot{R} - A [\{r_c\} \times I] \dot{\omega} + AN_c \ddot{q}$$

External force

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

## Better convergence from version 10.4



```

Lister - (C:\torben\wave-loads-offshore-structures) 2011-06-07-meeting.hawc2
File Edit Options Help
Lister - (C:\torben\wave-loads-offshore-structures) 2011-06-07-meeting.hawc2
File Edit Format New Help
jacket_11 - Notepad
hydroload element commands read with success
Reading data of hydro element connected to mbdy : s4x4b
hydroload element commands read with success
Reading data of hydro element connected to mbdy : tbd1
hydroload element commands read with success
Reading data of hydro element connected to mbdy : tbd2
hydroload element commands read with success
Reading data of hydro element connected to mbdy : tbd3
hydroload element commands read with success
Reading data of hydro element connected to mbdy : tbd4
hydroload element commands read with success
Main input file is returned to :jacket_11.htc
output commands read with success
Output command read
Input position of structure
Initialization of wind
Creating link between structure and hydrodynamics
Initialization of hydro loads
start simulation
Global time = 2.500000000000000E-002 Iter =
Global time = 5.000000000000000E-002 Iter =
Global time = 7.500000000000000E-002 Iter =
Global time = 0.100000000000000 Iter =
Global time = 0.125000000000000 Iter =
Global time = 0.150000000000000 Iter =
Global time = 0.175000000000000 Iter =
Global time = 0.200000000000000 Iter =
Global time = 0.225000000000000 Iter =
Global time = 0.250000000000000 Iter =
Global time = 0.275000000000000 Iter =
Global time = 0.300000000000000 Iter =
Global time = 0.325000000000000 Iter =
Global time = 0.350000000000000 Iter =
Global time = 0.375000000000000 Iter =
Global time = 0.400000000000000 Iter =
Global time = 0.425000000000000 Iter =
Global time = 0.450000000000000 Iter =
Global time = 0.475000000000000 Iter =
Global time = 0.500000000000000 Iter =
Global time = 0.525000000000000 Iter =
Global time = 0.550000000000000 Iter =
Global time = 0.575000000000000 Iter =
Global time = 0.600000000000000 Iter =
Global time = 0.625000000000000 Iter =
Global time = 0.650000000000000 Iter =
Global time = 0.675000000000000 Iter =
Global time = 0.700000000000000 Iter =
Global time = 0.725000000000000 Iter =

```

Hydrodynamics: November 2012

## HAWC2 example

```

begin HYDRO ;
begin WATER_PROPERTIES ;
  rho 1025 : [kg/m^3]
  gravity 9.816 : [m/s^2]
  mw1 0.000: [m]
  mudlevel 14.500 : [m]
  water_kinematics_dll ./wkin_dll.dll ./hydrohtc/reg_airy_1.htc ;
end WATER_PROPERTIES ;

;
begin HYDRO_ELEMENT ;
  mbody_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_1in 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;
;
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;
;
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;
;
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

## 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 ;
```