KONGSBERG

CAESES USERS MEETING 2019

# Advanced Feature Modeling

Paulo Macedo
Design & Simulation Specialist
19/09/2019

KONGSBERG

# Presentation Structure

- Understanding *FFeatureDefinitions*
- Motivation
- Starting point
- Examples
  - Shipflow
  - NAPA arrangement generation
- Challenges
- Conclusions

KONGSBERG

# Understanding *FFeatureDefinitions*

What are they for?

**“Encapsulation of recurrent tasks”**

- Curve Engines -> Meta Surfaces
- File I/O - reading and writing
- Conditional statements
- Run external processes
- Loops
- Containers
- Functions
- **Nesting**

```
if(condition)
  echo("hello")
elseif(condition)
  echo("world")
else
  echo("!!!")
endif
```

```
switch(myIntegerValue)
  case 1
    echo("value is one")
  case 2
    echo("value is two")
  default
    echo("value is neither one nor two")
endswitch
```

```
while(condition)
  echo("do something")
endwhile
```

```
loop(count)
  echo("iteration count: " + $$i)
endloop
```

```
foreach(F3DPoint p in myObjectList)
  // set the x-coordinate for all points in the objectlist to 4.
  // Objects that are not points are skipped.
  p.setX(4)
endfor
```

FEntityGroup

FObjectList

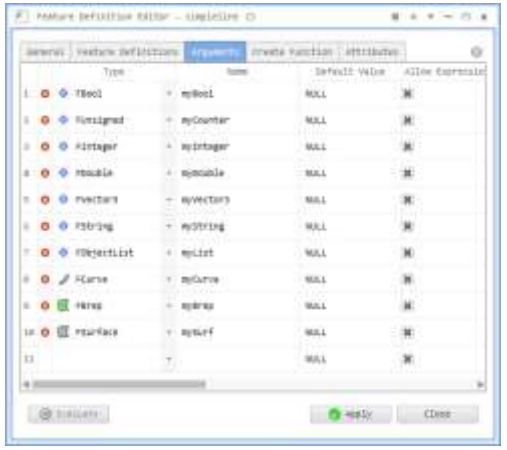
```
function myLineCreator(F3DPoint p1, F3DPoint p2) : FLine
  line l(p1, p2)
  return(l)
endfunction
```

KONGSBERG

# Understanding *FFeatureDefinitions*

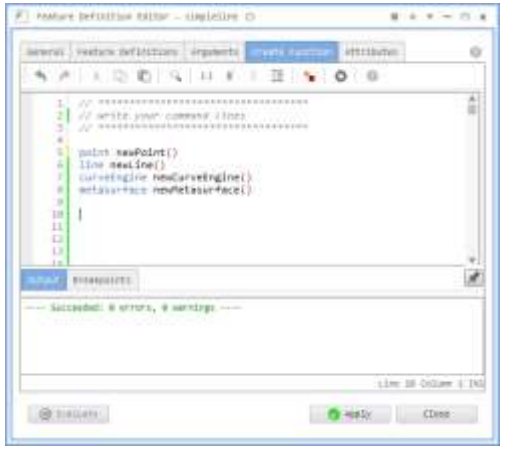
How do I use them?

1. Input arguments



understand entities

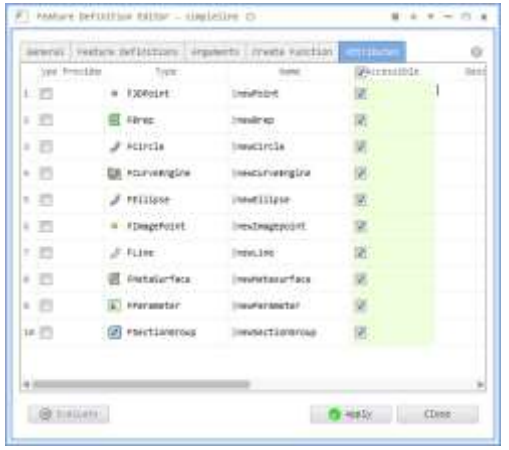

2. Command sequence



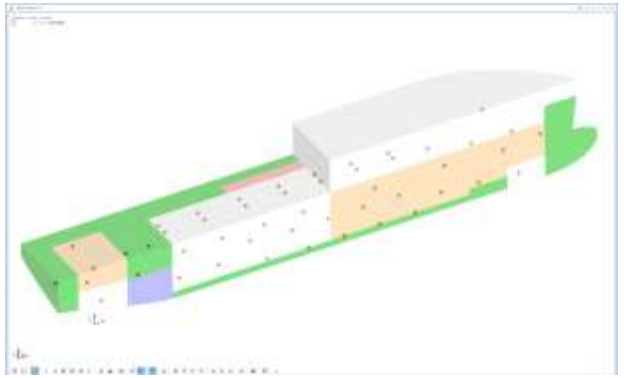
program desired actions

```
// set up string for current design directory adding the name of the feature
string workingDir(getDesignDir() + "/" + this.getName() + "/")
// create a new path
mkpath(workingDir)

// open a file for writing
FFile out(workingDir + "value.dat")
out.openWrite()
// write out the parameter value of myParam
out.writeLine("value="+myParam.toString())
out.close()
```

3. Output attributes



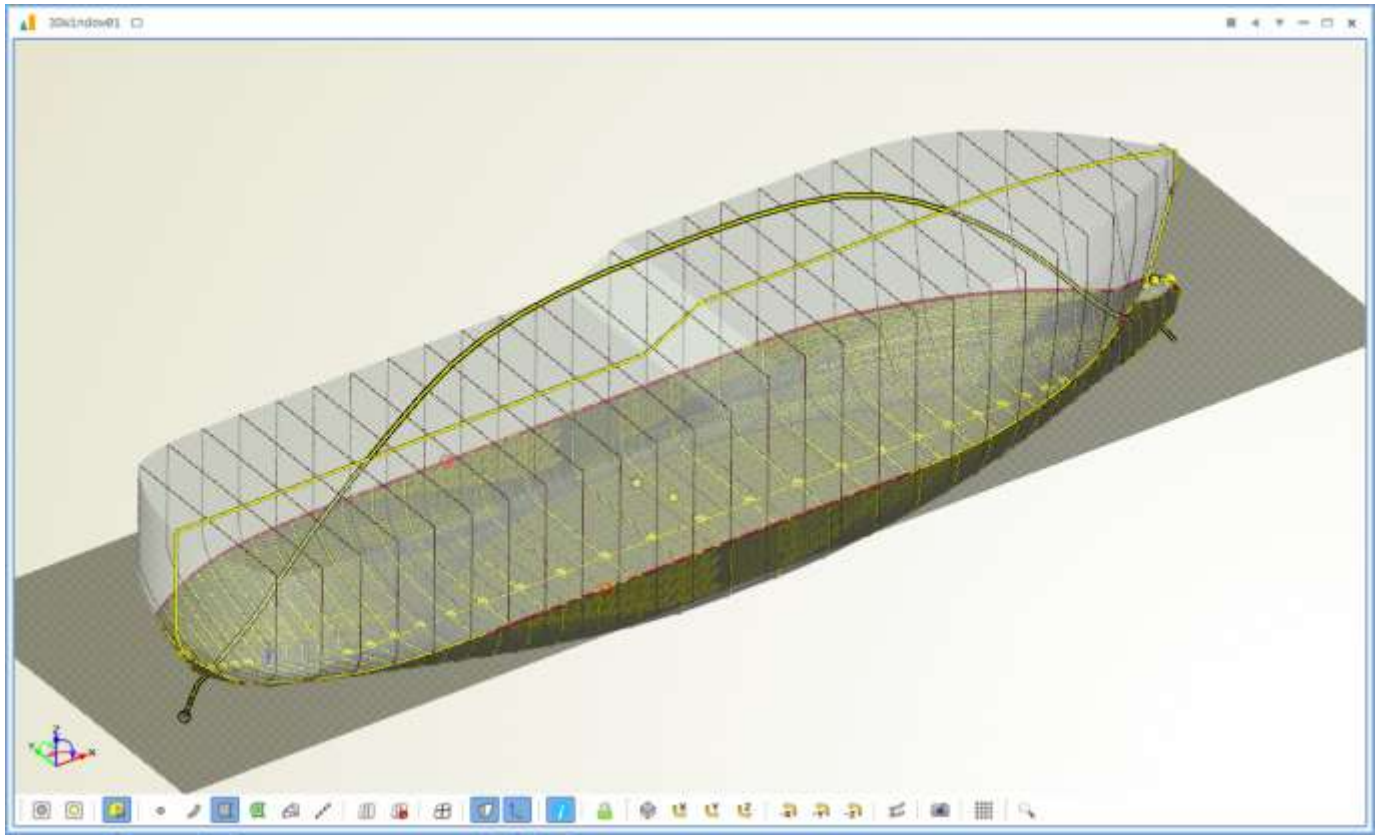
return/display results



# Motivation

## Ship Design at Kongsberg Maritime

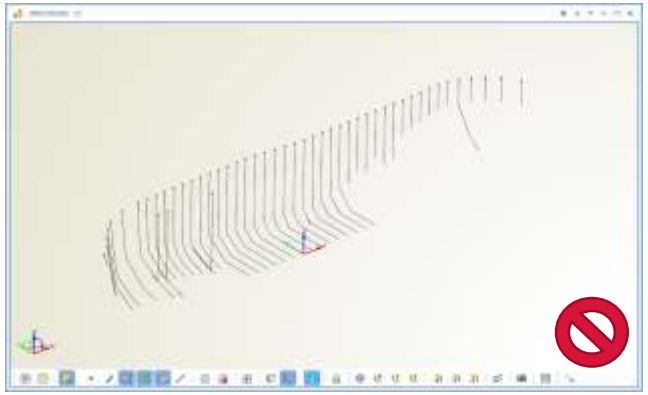
- Increase **efficiency** in early design phase
  - Hydrostatics extraction
  - Intact stability, IMO criteria
  - Crane positioning/capacity
  - Resistance estimations / initial optimization
  - Seakeeping evaluations
  - Station-keeping – future
  - Maneuvering – future
  - Simplified damage stability – future



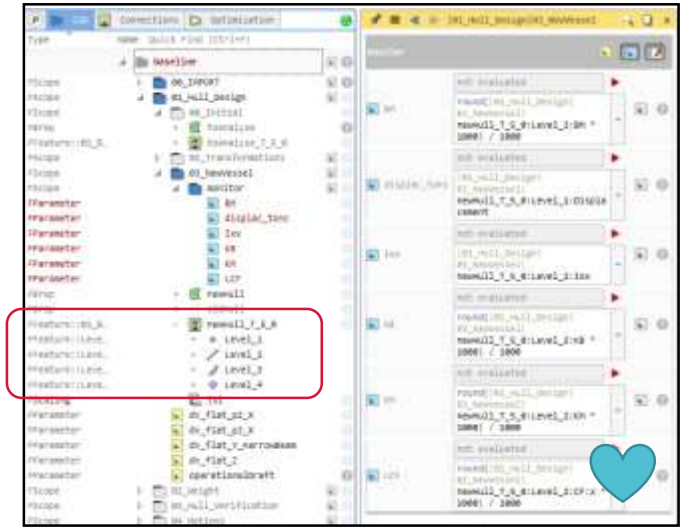
# Motivation

## Ship Design at Kongsberg Maritime

- Robust results



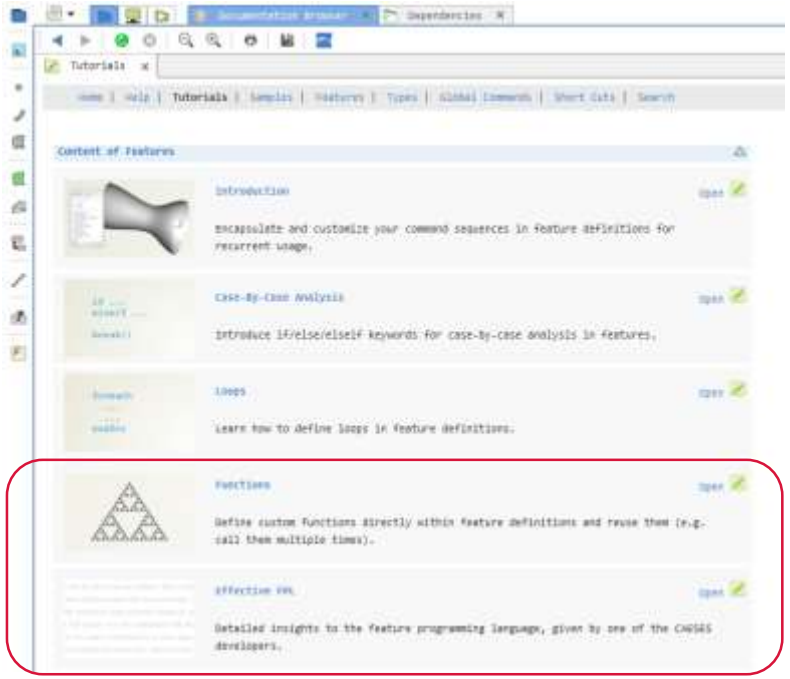
- Organization and user-friendliness

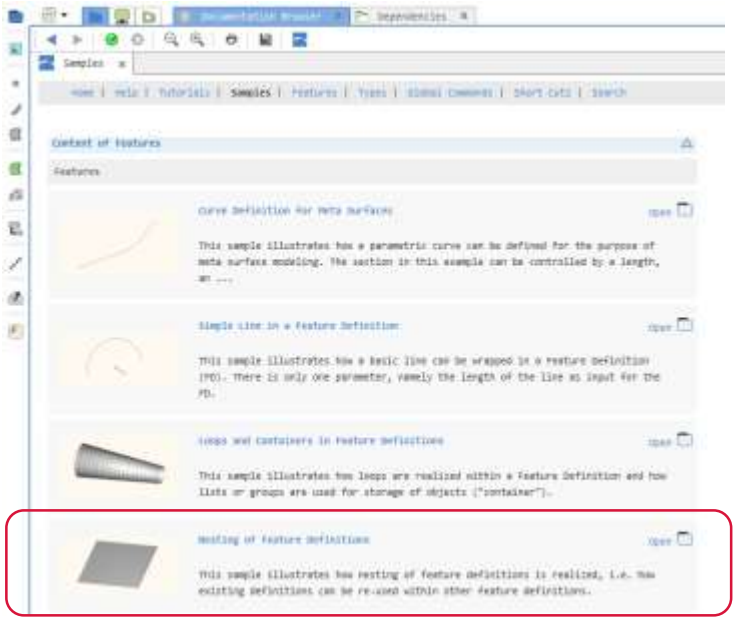


# Starting Point

## Tutorials & Samples

- Learn  Types and its available commands ( ". + ctrl + space")
- Documentation on functions and features



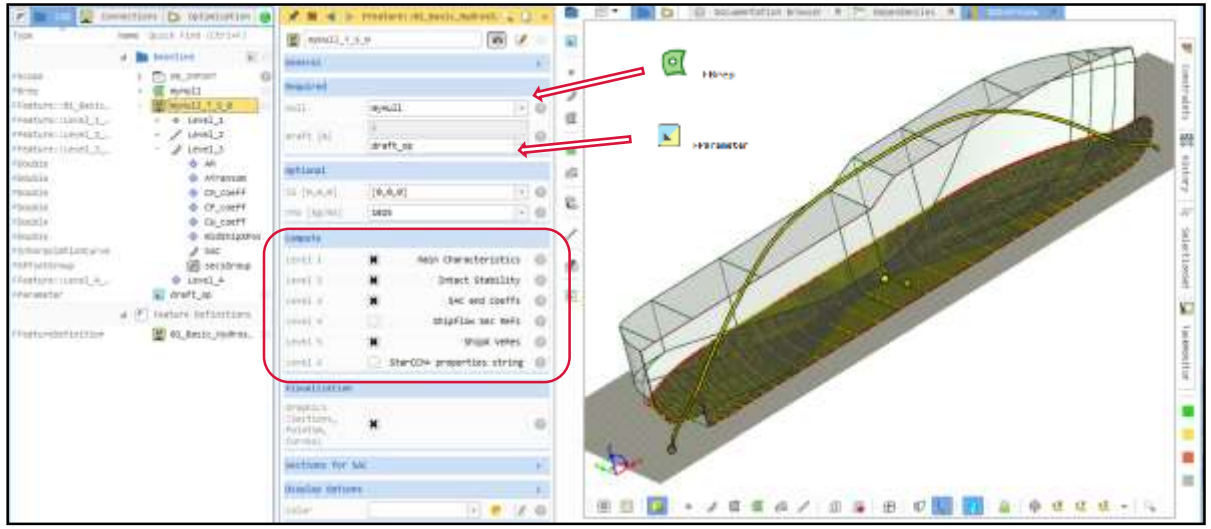



KONGSBERG

# Starting Point

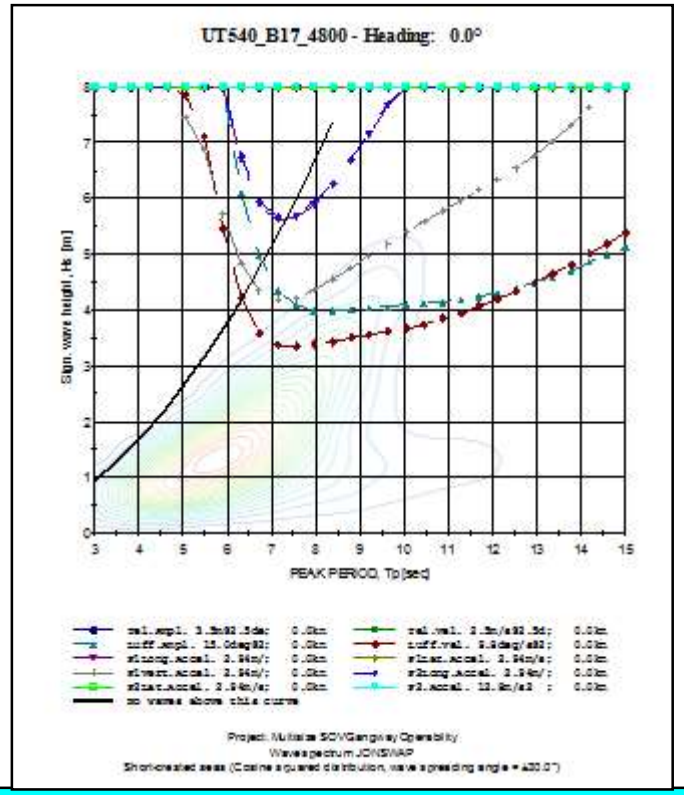
First usages

- “Basic Hydrostatics” FFeature::

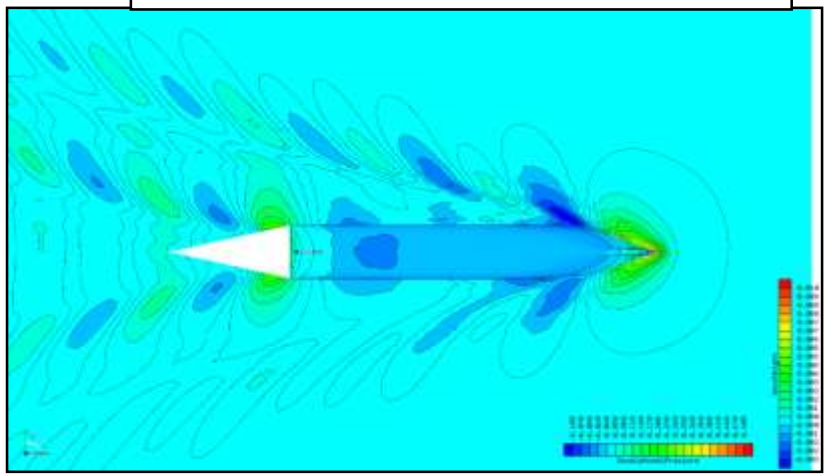


- Main characteristics
- Geometry conversions (including ASCII)
- *String* scripting

ShipX VeRes



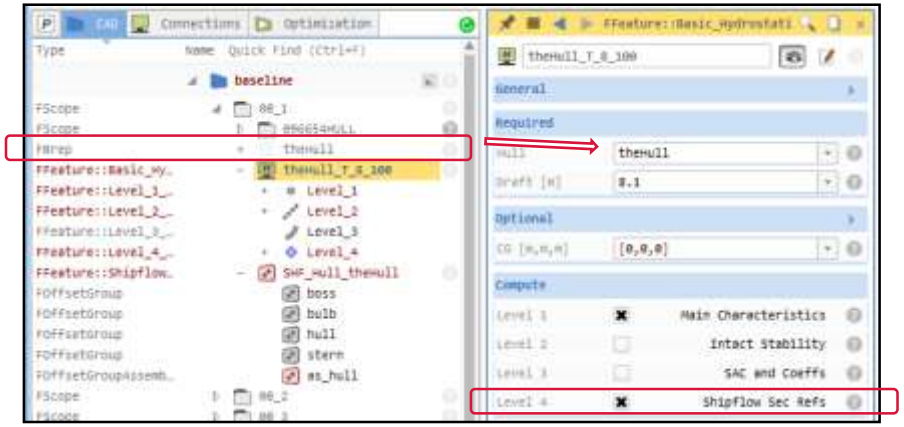

Shipflow

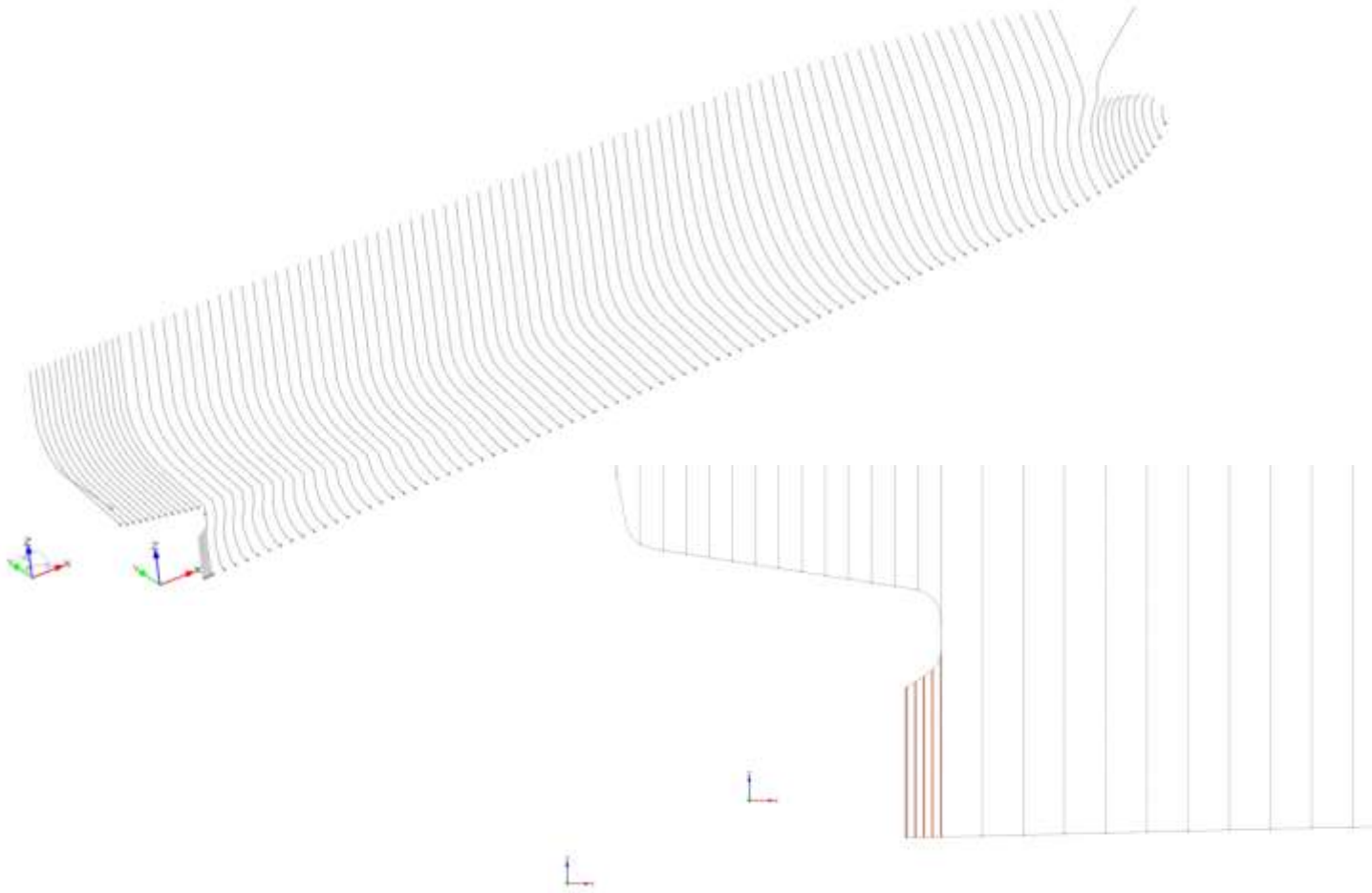


# Examples

## Shipflow

- Robust sections generation

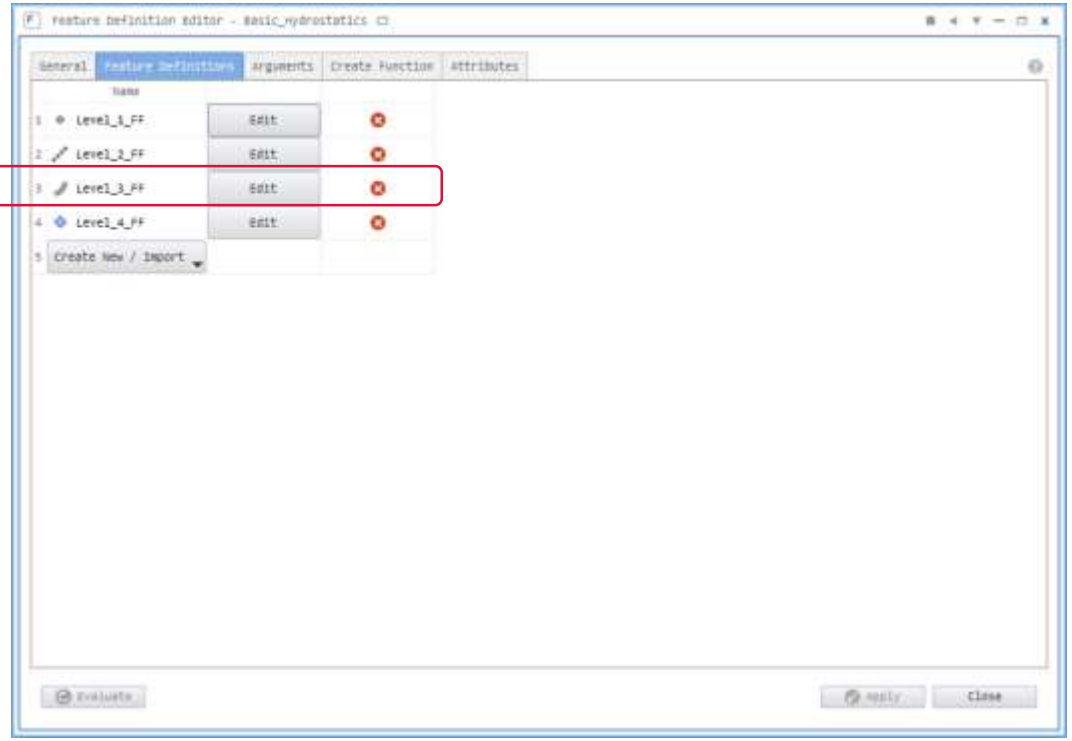


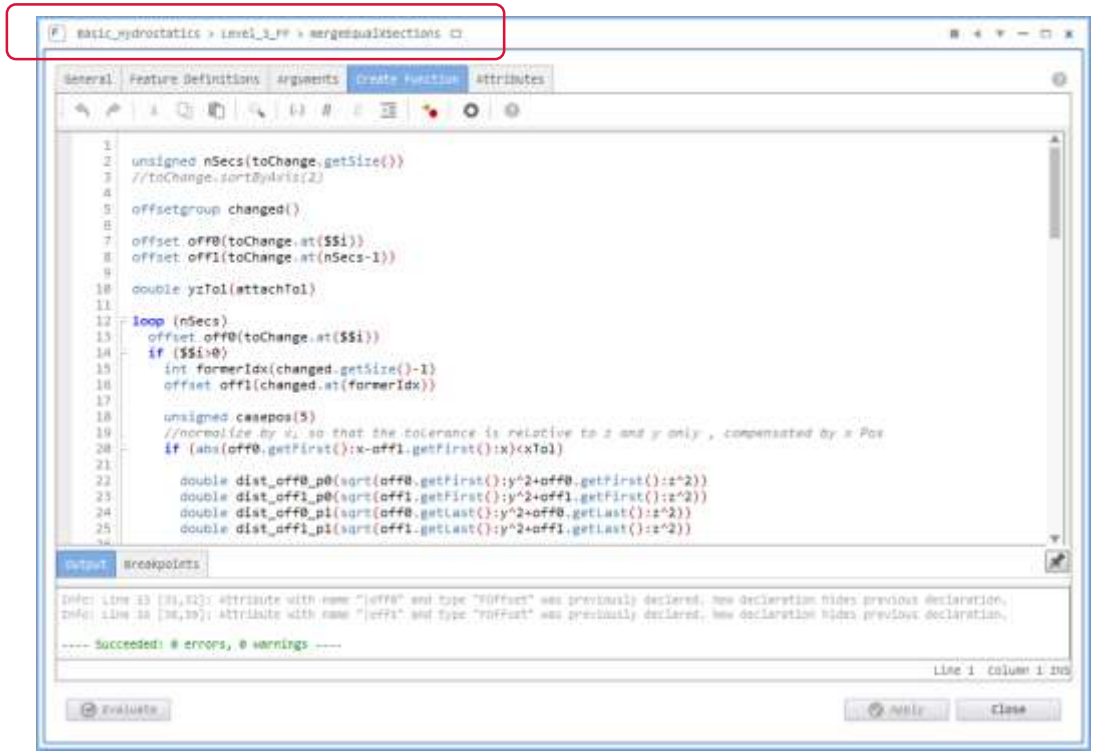



# Examples

## Shipflow – Coding

- Nesting features



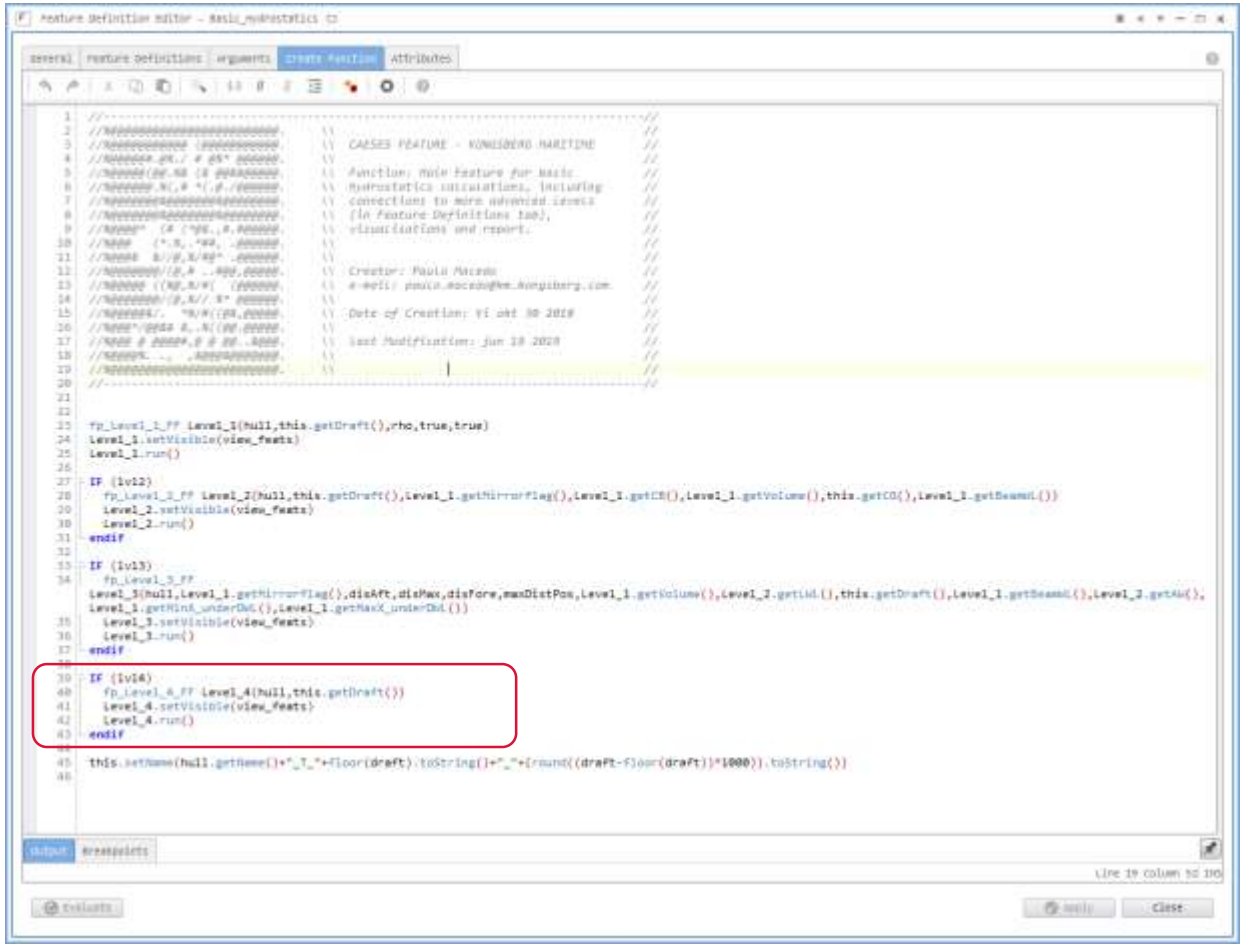
# Examples

## Shipflow – Coding

- Calling nested features
  - *"fp_FFeature_name"*
  - FFeature::name
  - (arguments) - "ctrl+space" for hints

```
39  fp_mergeEqualXSections changed(sectionsForSAC.getSections().cutMinMax(-999,9999,0,999,-100,draft),0.5,min(startDis,endDis)/2)
40  changed.run()
41  offsetGroup secsGroup(changed.getChanged())
42
```

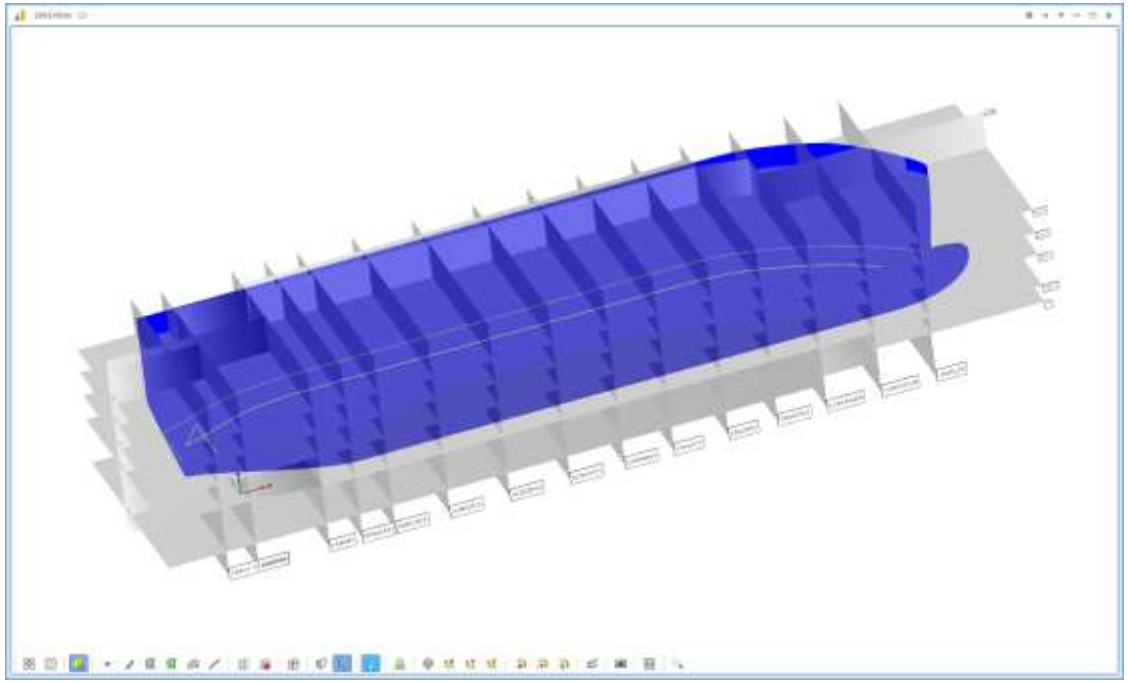
  - Run (if update only on user request)
  - Transfer results

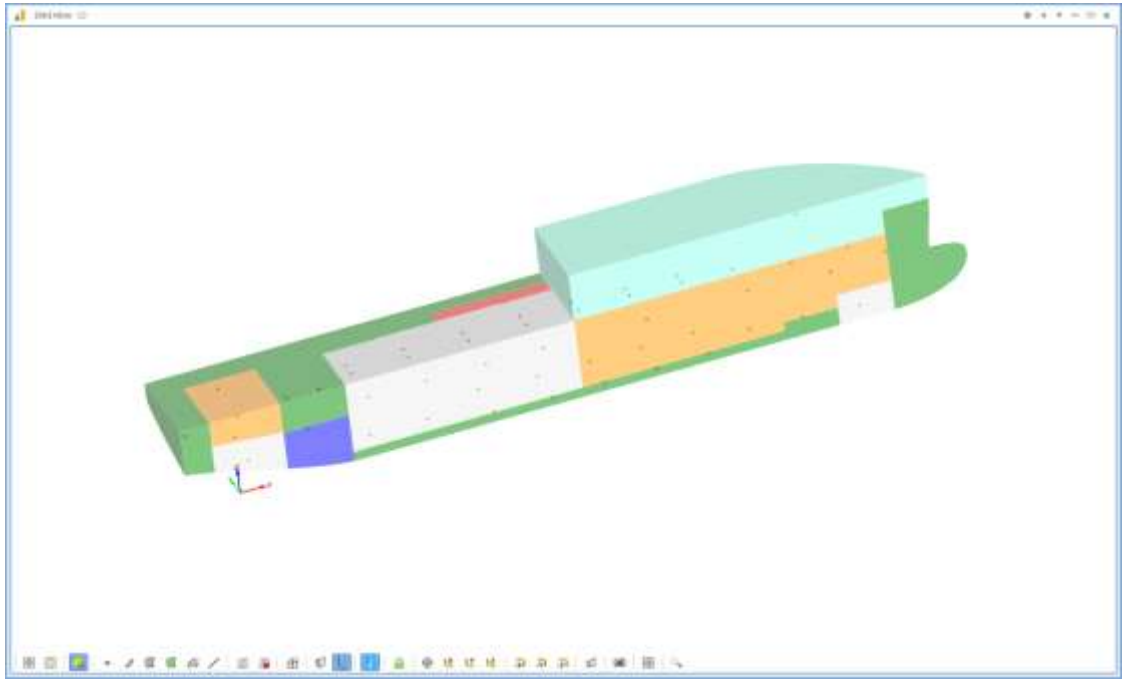


# Examples

## NAPA Damage Stability

- Flexible arrangement creation
- NAPA equivalent (user friendliness)

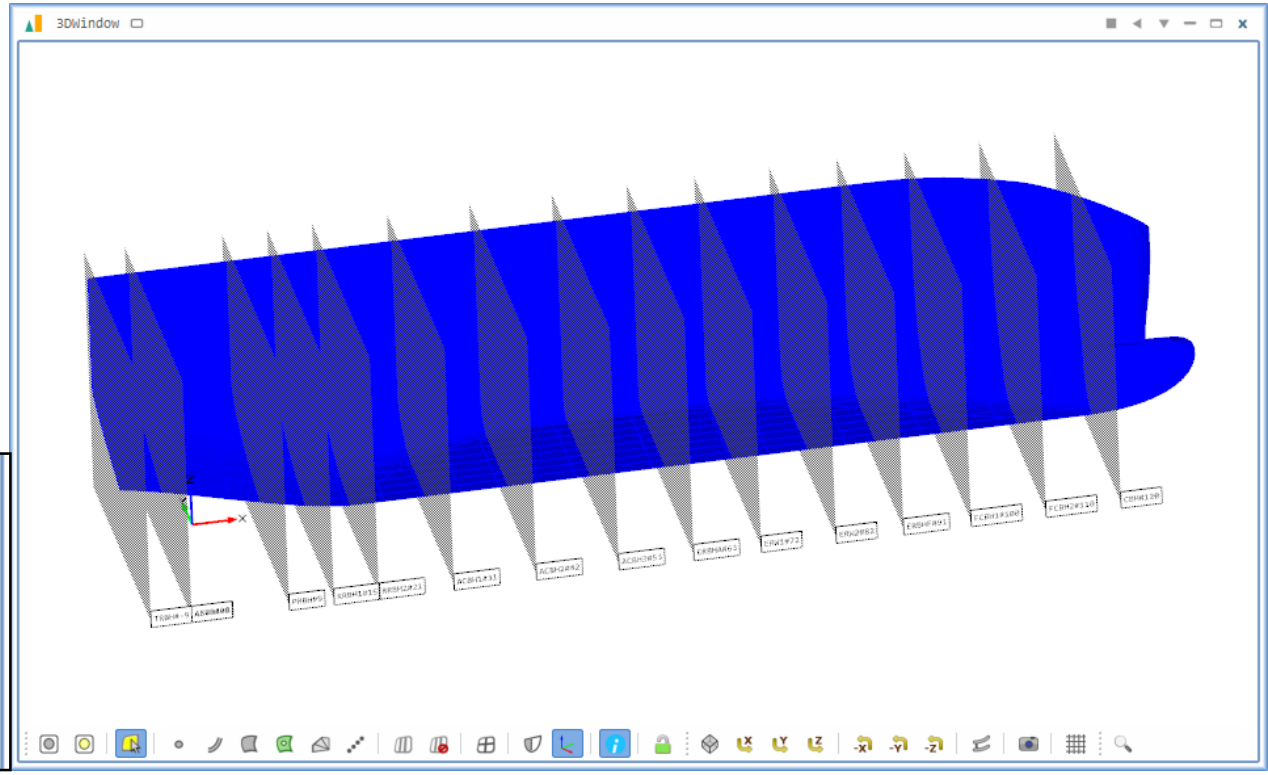




# Examples

## NAPA Damage Stability

- Flexible arrangement creation
- NAPA equivalent (user friendliness)
- Transversal bulkheads
  - *FFeature:NAPA_BHs*



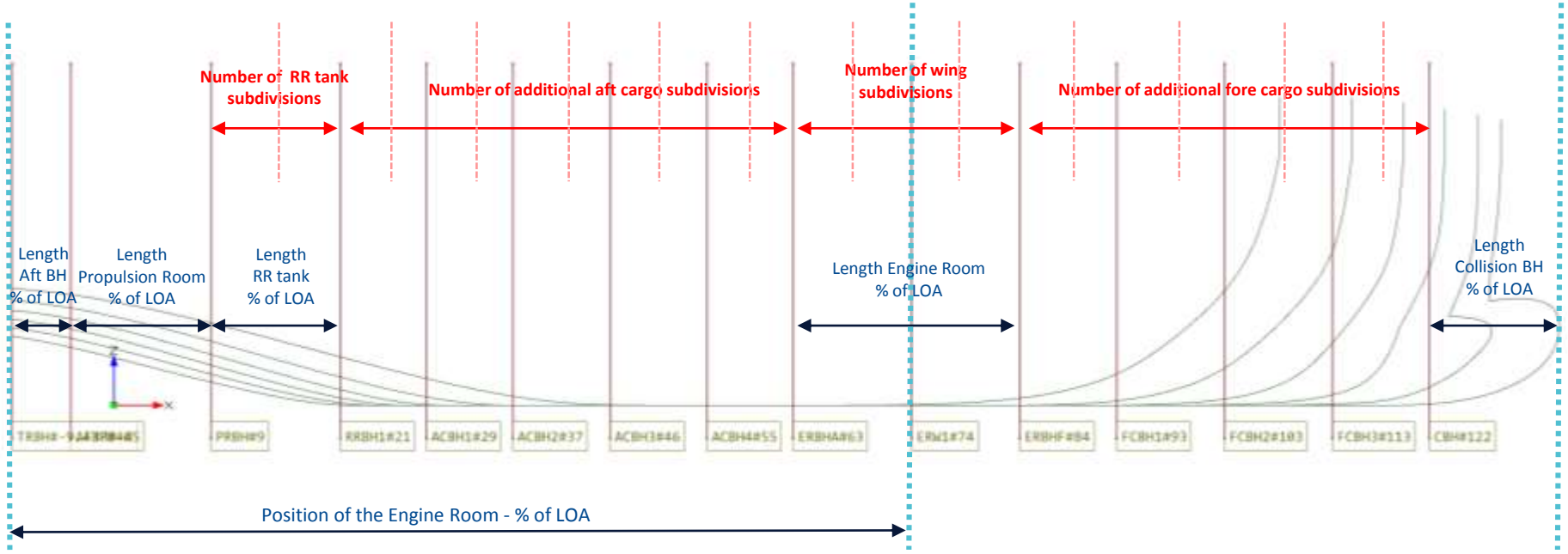

```
FFeature::NAPA_BHs  –  BHsss
FDouble                LOA
FEntityGroup           BHsOrdered
FEntityGroup           BHsSurfsOrdered
FEntityGroup           labels
FInteger               createBHsA
FInteger               createBHsF
FInteger               runFlag
FString                BHsgenStr
```

# Examples

## NAPA Damage Stability

- Transversal bulkheads
  - *FFeature:NAPA_BHs*

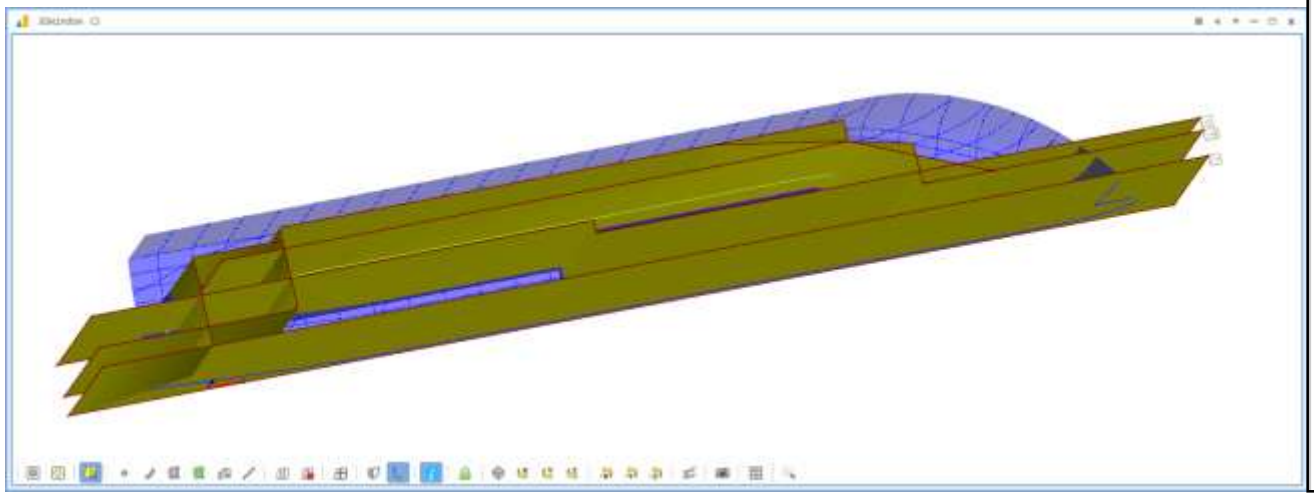
- Auto ordering, naming and numbering
- #Frames positioning system
- Custom manual BHs
- Auto restriction of BH length and/or quantity

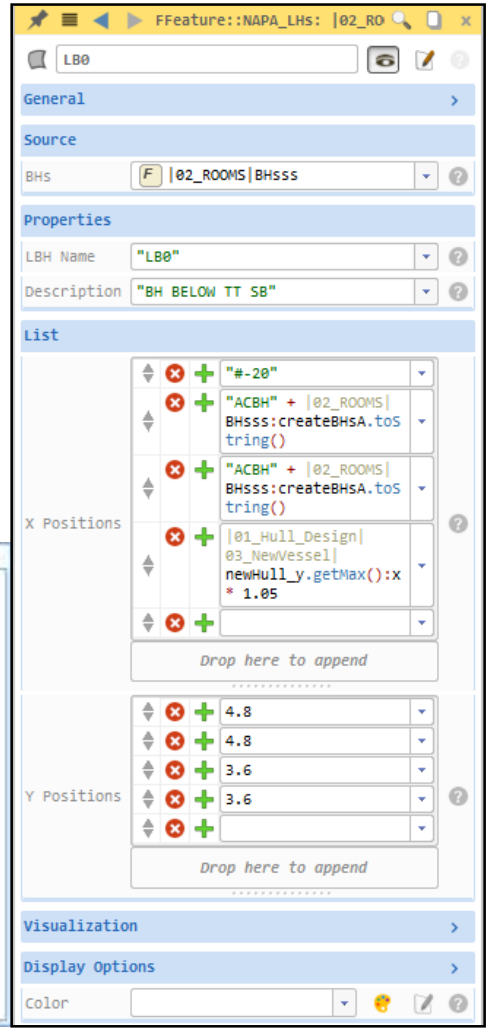



# Examples

## NAPA Damage Stability

- Flexible arrangement creation
- NAPA equivalent (user friendliness)
- Longitudinal bulkheads
  - *FFeature:NAPA_LHs*



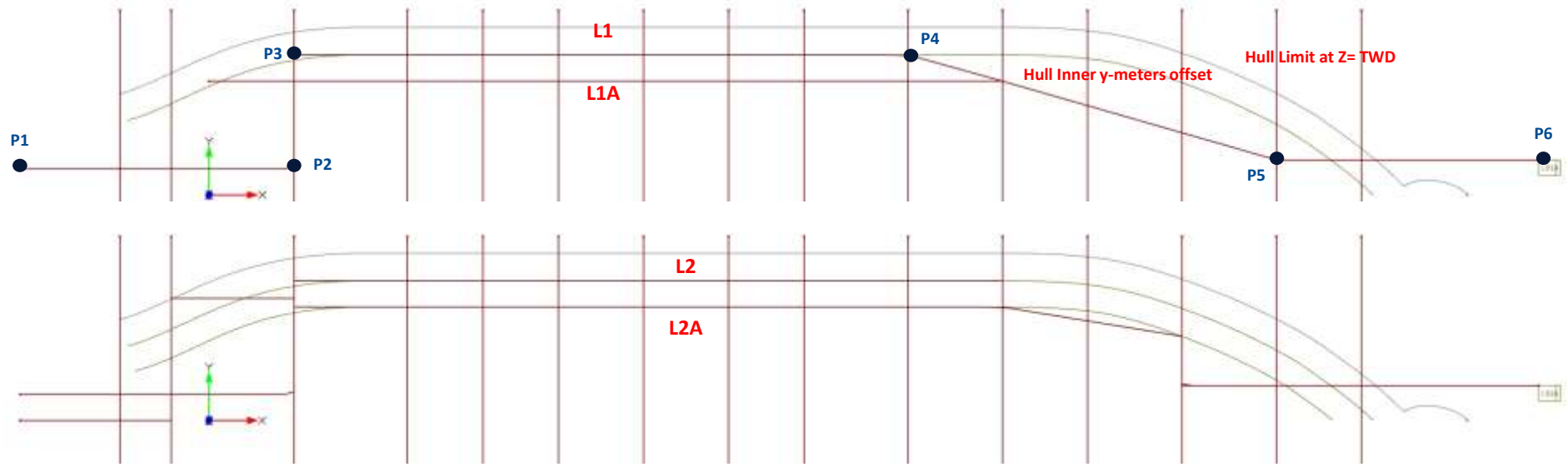


# Examples

## NAPA Damage Stability

- Longitudinal bulkheads
  - *FFeature:NAPA_LHs*

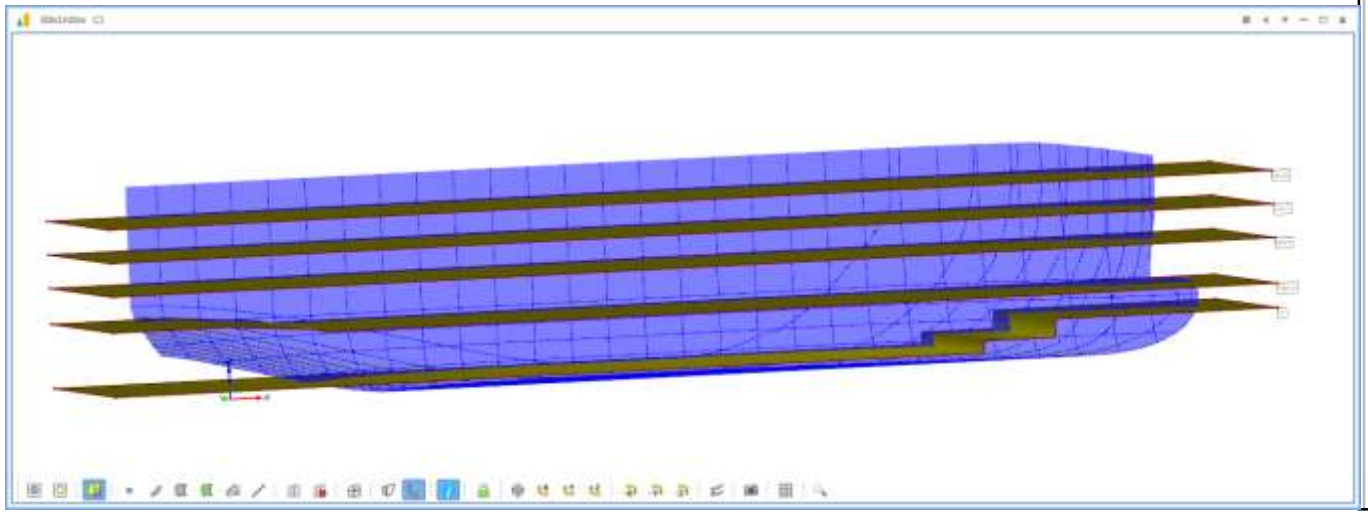
- Point based
- As many as you need
- BHs snapping
- Hull clearance/offset snapping

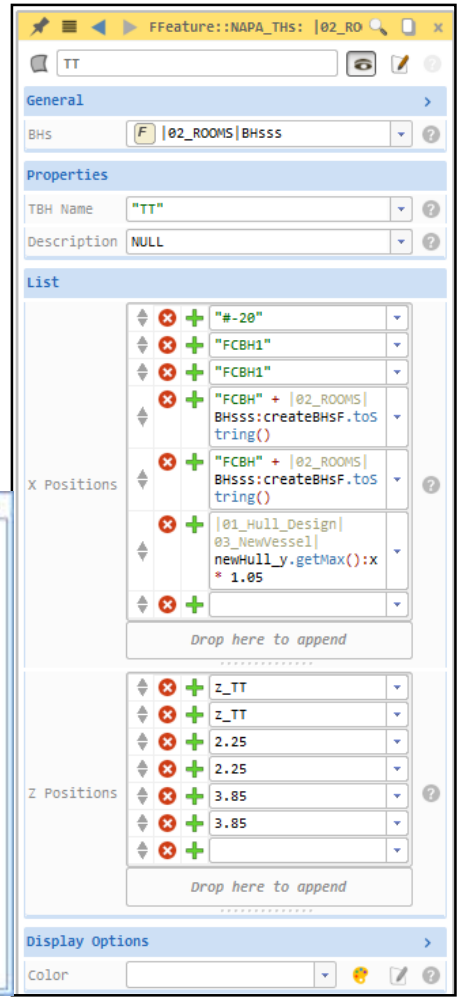


# Examples

## NAPA Damage Stability

- Flexible arrangement creation
- NAPA equivalent (user friendliness)
- Deck level bulkheads
  - *FFeature:NAPA_THs*



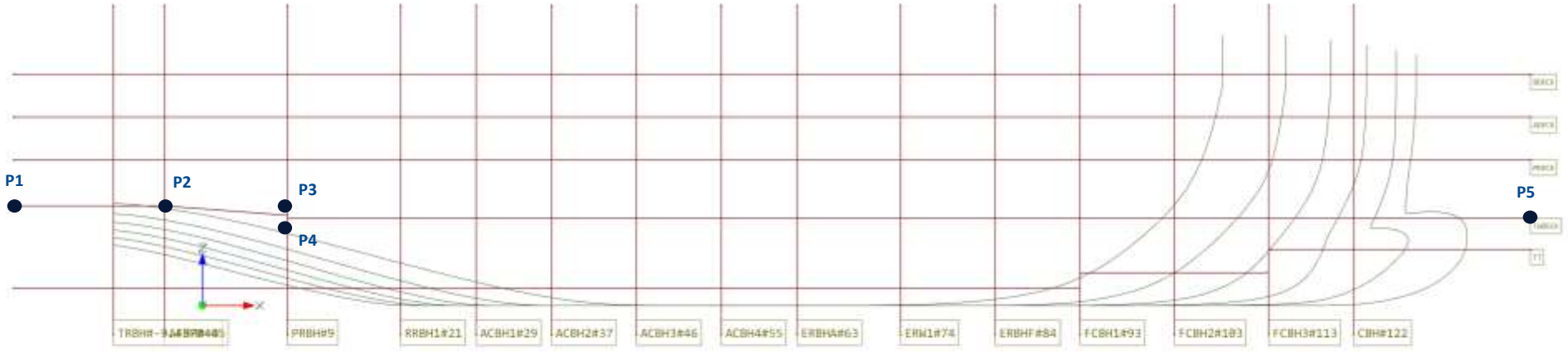


# Examples

## NAPA Damage Stability

- Deck level bulkheads
  - *FFeature:NAPA_THs*

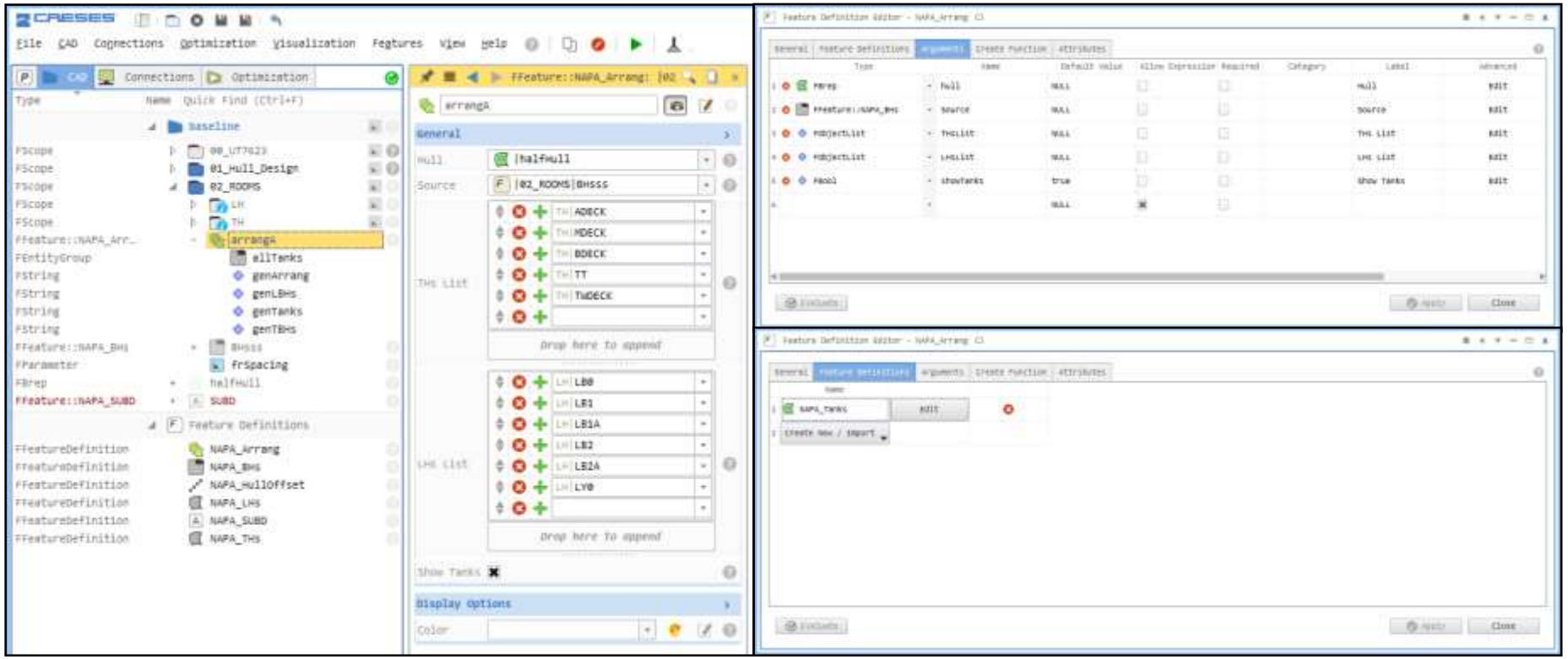
- Point based
- As many as you need
- BHs snapping



# Examples

## NAPA Damage Stability

- The arrangement
  - *FFeature::NAPA_Arrang*

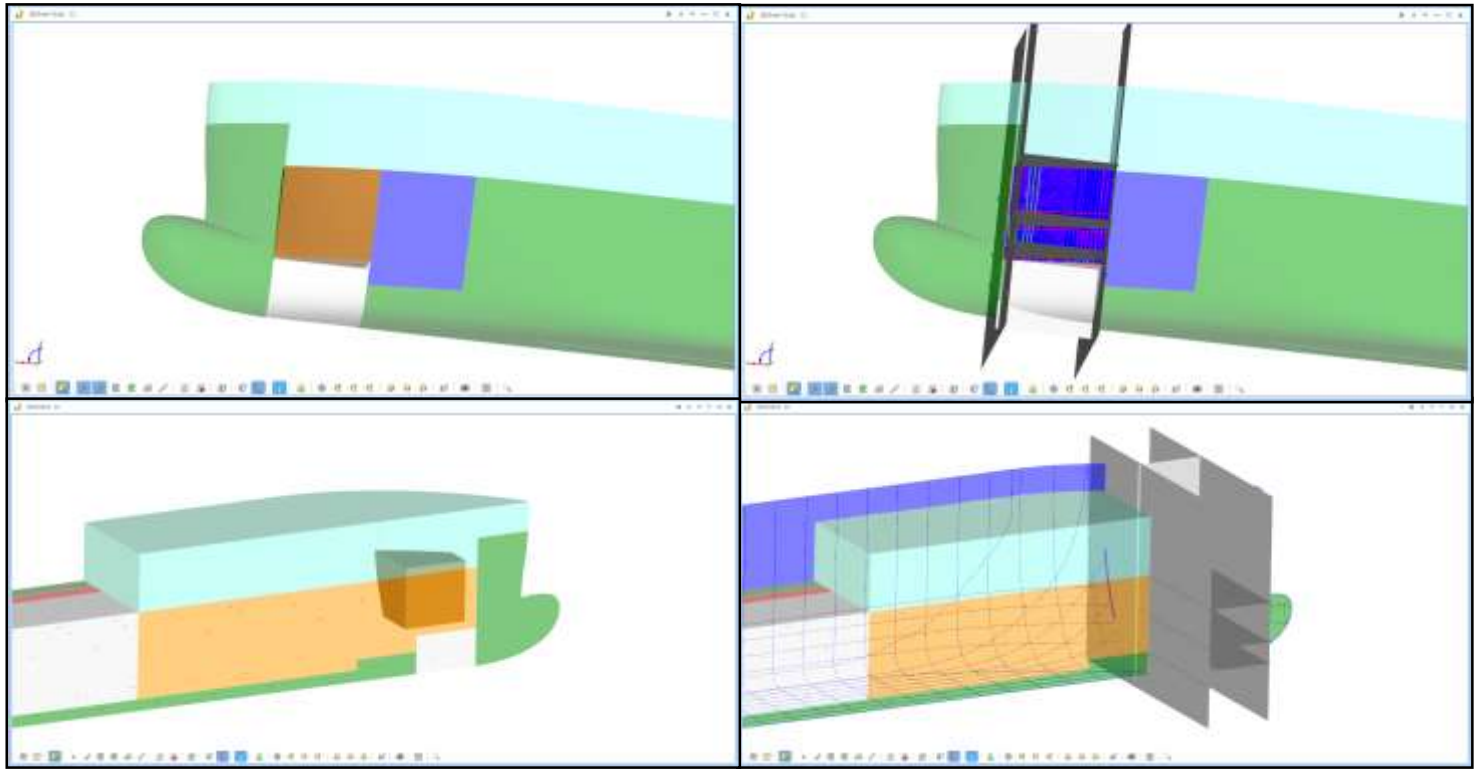


# Examples

## NAPA Damage Stability

- The arrangement
  - *FFeature::NAPA_Arrang*

- Tank intersections
- Tank merging



KONGSBERG

# Examples

## NAPA Damage Stability

- Similar to NAPA (list of limits)

```
fp_NAPA_Tanks T1_ForePeak(showTanks,hull,AllBHs,["CBH","-","LY0","HULL","-","ADECK"],1,2,idxs.at(1),true,false)
```

- Combination of tanks

```
fp_NAPA_Tanks R1_AuxMachAUX(showTanks,hull,AllBHs,["ERBHF",Source.at(findBHsID("CBH")-
IF(size>2,2,size)).getname(),"LY0","LB1","TT","TWDECK","Y<HULL"],5,6,91,true,true).setVisible(false)
fp_NAPA_Tanks R1_AuxMach(showTanks,hull,AllBHs,["ERBHF",Source.at(findBHsID("CBH")-
IF(size>2,2,size)).getname(),"LY0","LB2","TWDECK","MDECK","Y<HULL"],5,6,idxs.at(5),true,false,true,[R1_AuxMachAUX])
```

- Loop generation between n-BHs for tank generation

```
// ANTI ROLLING TANKS
size = Source.getN_RR()
loop (size)
```

```
finteger size(Source.getCreateBHsF())
loop (size)
  fp_NAPA_Tanks T2_WBbottomTT_Aut(showTanks,hull,AllBHs,
[Source.at(findBHsID("CBH")-2-$$i).getname(),Source.at(findBHsID("CBH")-$$i-1).getname(),"LY0","HULL","-","TT"],3,2,idxs.at(1),true,false)
  genTanks.append(T2_WBbottomTT_Aut.getTankDef())
  idxs.replace(idxs.at(1)+1,1)
  tankList.at(1).castto(fstring).append(findTNum(T2_WBbottomTT_Aut.getTankDef()))
  T2_WBbottomTT_Aut.setName(T2_WBbottomTT_Aut.getTankname())
  allTanks.add(T2_WBbottomTT_Aut)

  integer myID(if($$i==0,3,2))

  fp_NAPA_Tanks T_Fore_wing_Aut_Aux(showTanks,hull,AllBHs,
[Source.at(findBHsID("CBH")-2-$$i).getname(),Source.at(findBHsID("CBH")-1-$$i).getname(),"LB1","HULL","TT","TWDECK"],2,myID,idxs.at(myID-1),true,fa
ible(false)
  fp_NAPA_Tanks T_Fore_wing_Aut(showTanks,hull,AllBHs,
[Source.at(findBHsID("CBH")-2-$$i).getname(),Source.at(findBHsID("CBH")-1-$$i).getname(),"LB2","HULL","TWDECK","MDECK"],2,myID,idxs.at(myID-1),fals
e,[T_Fore_wing_Aut_Aux],[T_Fore_wing_Aut_Aux.getTankDef()], T_Fore_wing_Aut_Aux.getTankCG())
  genTanks.append(T_Fore_wing_Aut.getTankDef())
  tankList.at(myID-1).castto(fstring).append(findTNum(T_Fore_wing_Aut.getTankDef()))
  idxs.replace(idxs.at(myID-1)+2,myID-1)
  T_Fore_wing_Aut.setName(T_Fore_wing_Aut.getTankname())
  allTanks.add(T_Fore_wing_Aut)
endloop
```

- Purpose definition

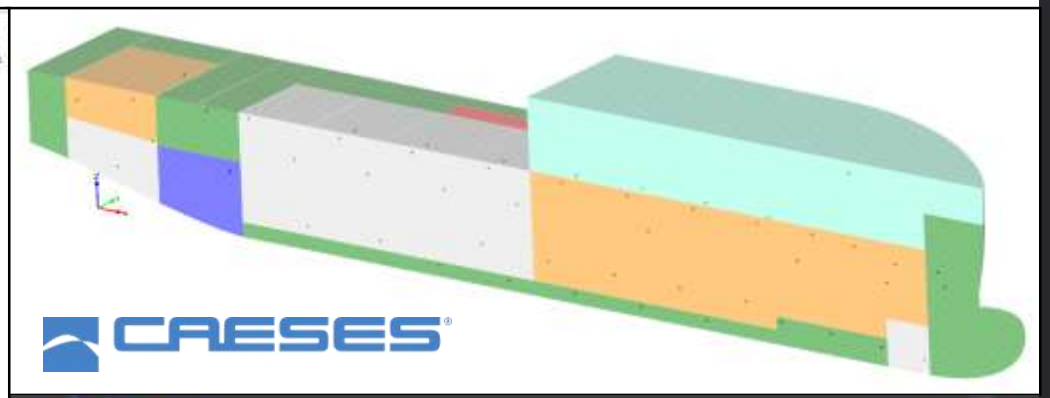
```
//purps.add([PURP,PDES,RHO,PERM,COLOUR R,COLOUR G,COLOUR B])
purps.add(["VOI","VOID SPACE","T8",1,0.95,211,211,211])      //lightsilver   1
purps.add(["WB","WATER BALLAST","T1",1.025,0.95,0,128,0])    //green         2
purps.add(["FW","FRESH WATER","T2",1,0.95,0,0,255])          //blue          3
purps.add(["FO","FUEL OIL","T3",0.9,0.95,255,0,0])           //red           4
purps.add(["MIS","MISCELLANIES","R2",1,0.95,127,255,212])    //aqua marine   5
purps.add(["ENG","MACHINERY SPACE","R1",1,0.85,255,250,250]) //snow white    6
purps.add(["CHA","CHAIN LOCKER","T6",1,0.85,255,0,255])      //fuchsia       7
```

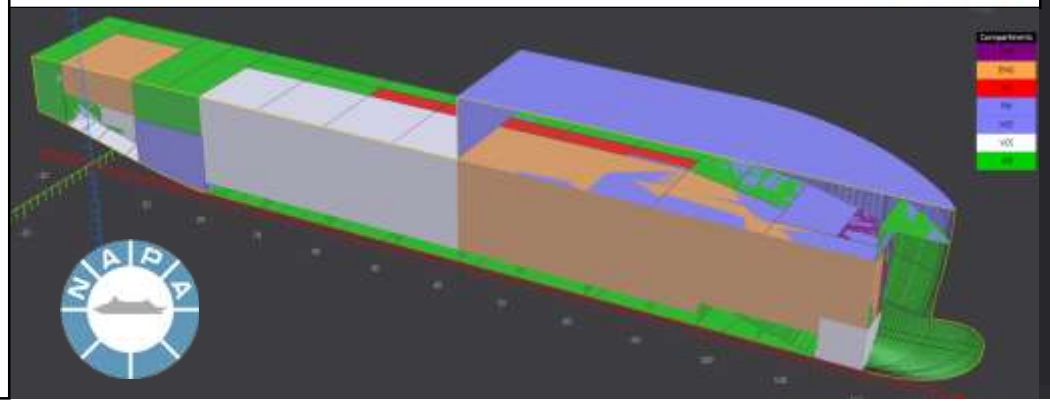
# Examples

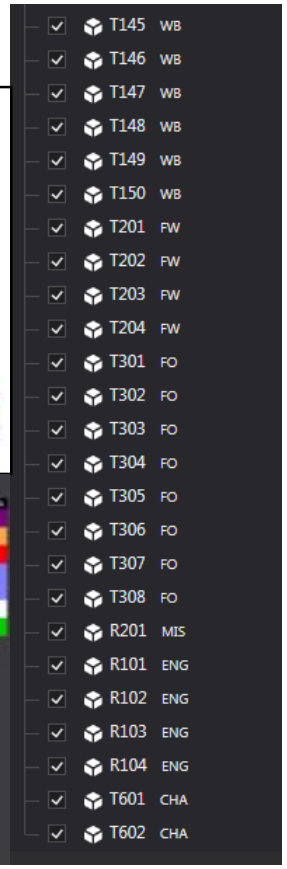
## NAPA Damage Stability

- Resulting geometry: CAESES -> NAPA (via Software connection)

```
## Note 1: 3-4 non-essential or blank lines in beginning - required for napa
## Note 2: double# is interpreted as a comment line in napa
## Note 3: commands in this script are restricted to "command window" in napa

@projnam='KLSRF_CAESES_DMG_<entry>projnam</entry>'
@descr='CAESES DMG STABILITY DEMO FOR HOLISHIP'
## array maindim containing reference dimensions (Lpp,B,T,frdist)
@maindim=arr(2)
@maindim(1)=<entry>length</entry>
@maindim(2)=<entry>beam</entry>
@maindim(3)=<entry>draught</entry>
@maindim(4)=0.65
@n=mn.deleteproject(projnam)
@n=mn.newproject(projnam,descr,'','','',maindim)

ref
coord RIGHTHANDED
end

def
from iges <entry>Exports\hull.igs</entry> HULL out-y !
ok
ROOM DAMHULL
LIM - - 0 hull - HDECK
ADD ERBEA - 0 hull HDECK BDECK
SYM
ok
<entry>genBMaPlanes</entry>
<entry>genTBHs</entry>
<entry>genLBHs</entry>
ok
<entry>RXEN_T</entry>
ok
!end

SN
NEW A
SAVE
##!HULL TYPE=R
@ADD LIST()
<entry>ADD_PUMP</entry>
REFL
REG A PERM
!end

DEF
GEN  PROFIL, ARR*A/Y=0.01 MERGE
!END
```





KONGSBERG

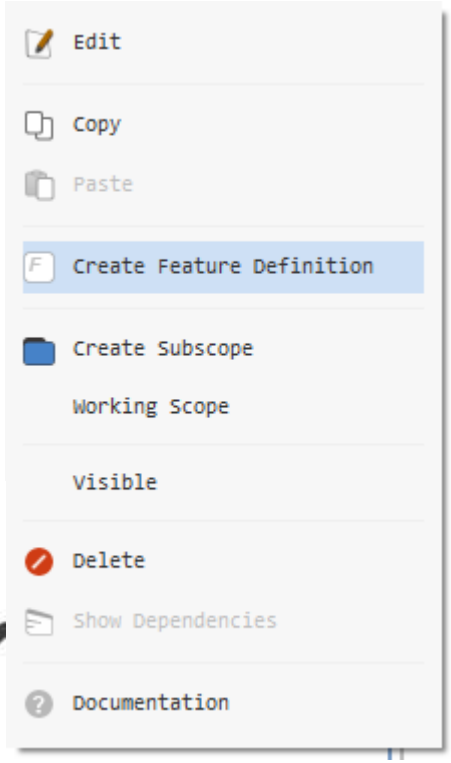
# Challenges

- Improve scripting
  - Computational efficiency
  - User-friendliness
- Reach robustness
  - Geometry quality
  - Sewing tolerances
- Increase CAESES usage within Ship Design
  - Steep learning curve
  - Resources availability

# Conclusions

- *FFeatureDefinitions* is the way forward with CAESES
  - Learn object types and its commands/possibilities
  - Tutorials, community forum and helpdesk
  - Create Features from CAD objects

- Your imagination is the limit



KONGSBERG

paulo.macedo@km.kongsberg.com