

Development of a Generic Hull Stiffener Modelling Capability for Trident Modeller

Tom MacAdam, Martec Limited

Prepared By:
Martec Limited
1888 Brunswick Street, Suite 400
Halifax, Nova Scotia, B3J 3J8

Contractor's Document Number: TR-13-48
Contract Project Manager: Tom MacAdam, 902-425-5101 X239
PWGSC Contract Number: W7707-125422
CSA: John MacKay, Defence Scientist, Warship Performance, 902-426-3100 X382

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Defence Research & Development Canada

Contract Report
DRDC-RDDC-2014-C16
September 2013

This page intentionally left blank.

Abstract

This report details work undertaken to add a general hull stiffener modeling capability to the Trident Modeller software application. The capability was designed after analysis of the way hull stiffeners are typically defined in ship construction plans. The resulting implementation allowed creation of stiffeners through a number of means including arclength measurements along the hull surface, equal spacing between reference curves on the hull surface, and importing the stiffener's shape from external CAD definitions. The new capability was tested and exhibited a level of robustness on par with the existing Trident Modeller RMGScript modeling entities.

This page intentionally left blank.

Table of contents

Abstract	i
Table of contents	iii
List of figures	iv
1 Hull Stiffeners for Trident Modeller.....	1
1.1 Introduction	1
1.2 Design of Hull Stiffeners	1
1.3 Implementation of Hull Stiffeners.....	2
1.3.1 The HullStiffener RMGScript Entity	2
1.3.1.1 Specifying the objects to attach to	3
1.3.1.2 Defining the shape and orientation of the web	3
1.3.1.3 Specifying the scantlings and trim options	6
1.3.1.4 Considerations and Limitations	7
1.3.2 The HullStiffener Creation Algorithm	8
1.3.3 Testing Examples	9
1.3.3.1 Using direct HullStiffenerPoints.....	9
1.3.3.2 Using HullStiffenerPoint at distance along a curve.....	10
1.3.3.3 Using equally spaced HullStiffenerPoints	11
1.3.3.4 Using HullStiffenerPlane.....	13
1.3.3.5 Using HullStiffenerCurve.....	13
1.4 Conclusions and Future Work	13
References	14

List of figures

Figure 1: HullStiffenerPoint modeling option using the intersection of two curves.	4
Figure 2: HullStiffenerPoint modeling option using an distance along an intersection curve between two other intersection curves.	5
Figure 3: HullStiffenerPoint modeling option using equally spaced points along an intersection curve between two other intersection curves.	5
Figure 4: Two HullStiffeners created using the default web orientation following the surface normal (top) and an enforced constant web orientation (bottom).	10
Figure 5: HullStiffeners created using the HullStiffenerPoint arclength option (equal arclength distances measured from upper deck).	11
Figure 6: HullStiffeners created using the HullStiffenerPoint evenly spaced modeling option (‘n’=5; ‘i’=1..4).	12

1 Hull Stiffeners for Trident Modeller

1.1 Introduction

Trident Modeller is a generic offshoot of the SubSAS [1] submarine modeling and analysis application jointly developed by Defence Research and Development Canada – Atlantic (DRDC-A), the United Kingdom Ministry of Defence (MoD) and Martec Limited. Trident Modeller shares SubSAS' underlying geometry modelling language, RMGScript, as well as its capabilities for auto-generating finite element models from geometric models. While SubSAS focusses on submarine geometry and analysis, Trident Modeller is intended to support modelling of general structures. To date Trident Modeller has been mostly used to model ship structures.

Despite inheriting a wide ranging set of modeling objects from its sister application SubSAS, Trident Modeller lacked some important types of objects common to ship structures. In particular, the Modeller had only rudimentary support for defining stiffeners on non-planar surfaces. This made it difficult to accurately represent an important category of ship structural members, hull stiffeners, characterised by their nonlinear path across the hull and their complex web orientation based on the contour of the hull. This work focussed on adding the necessary RMGScript entities to allow modeling such stiffeners.

1.2 Design of Hull Stiffeners

As part of this work, examples of hull stiffener definitions, particularly hull longitudinal definitions, were sought out in various ship drawings. It was found that hull longitudinals were often specified on shell expansion drawings, in which case their trace curve position on the expanded shell was defined as a function of longitudinal location. From a shell expansion drawing, one could envision coming up with a list of 2D (*longitudinal location, arclength*) coordinates which could then be converted to 3D (x,y,z) coordinates by evaluating their position on the hull surface. From there a spline could be fit to the 3D coordinates in order to reproduce the trace curve of the stiffener along the hull. Using this strategy, one would have to watch closely that the fit spline was a reasonable reproduction of the actual stiffener trace curve (i.e. it accurately followed the shape of the hull), but this could be aided by increasing the number of coordinates passed to the fitting algorithm as well as evening their spacing relative to each other.

In other cases, it was found that stiffener locations on the hull could be derived relative to other objects. For instance, it was common that stiffeners would be evenly spaced on the hull surface between pairs of decks. In such cases, coordinates could be determined by dividing a hull section between two parallel objects into equally-spaced portions. Doing so at a series of longitudinal locations would yield points to which a curve could be fit in the same manner described above.

Lastly, and least commonly, it was found that some stiffeners (most often deep girders or stringers) were specified at an relative offset from a design reference (i.e. centerline or baseline). In these cases, their location on the hull surface could be determined by intersecting the hull with a plane at the given offset position. If the stiffener was at a uniform offset, then this resulting curve would be the stiffener trace curve; otherwise a series of points would have to be built up by

intersecting the hull at the given longitudinal locations and offsets and fitting a curve as described above.

In addition to the stiffener trace curve, a definition of the stiffener web orientation would be necessary to accurately reproduce the actual stiffener shape. In many cases, longitudinal hull stiffeners were oriented orthogonal to the hull surface at all points along their length. In other cases, particularly those of deep girders or stringers, their orientation was in line with a principal direction (i.e. up/down or port/starboard). In fewer cases, orientation was specifically prescribed, typically to align a stiffener with another member (e.g. another stiffener, a deck or a margin plate).

Given the above observations, a hull stiffener modeling capability was designed. In most cases, the hull stiffener would be built by fitting a curve to a series of 3D points defining its shape along the hull (unless the trace curve could be determined by intersecting the hull with a single plane). The points could be determined using a variety of methods, designed to cover the approaches observed (i.e. shell expansion coordinates, even distributions between reference objects, etc.). The orientation would default to orthogonal to the hull surface, but could be enforced at each specified 3D point. The implementation of this design is discussed in the next Section.

1.3 Implementation of Hull Stiffeners

Support for modelling hull stiffeners was added to Trident Modeller through the introduction of new RMGScript entities that could specify a definition of the stiffener trace line and orientation as well as new algorithms that could interpret that definition and turn it into geometry representing the stiffener. The new RMGScript entities could only typically capture an approximation of the stiffener trace line, and as such the resulting geometry had to be considered an approximation, but it was felt it was of sufficient accuracy to be fit for purpose (and accuracy could always be improved by increasing the resolution of the approximation).

The hull stiffeners were implemented with specific goals in mind. Firstly, they had to be general enough to apply to any surface or collection of surfaces – both non-planar as well as planar. This would ensure they were equally applicable to the curved portions of a hull as to the planar portions (e.g. flats of side or bottom). Secondly, they would have to support both high and low fidelity representations in the resulting finite element (FE) model as did other RMGScript stiffeners (e.g. PlanarWebStiffener). They would also have to support being trimmed precisely to objects at their ends, as was the case for PlanarWebStiffeners. Lastly, they would need to follow a consistent syntax for their definition, particularly of any parameters that were common to other RMGScript stiffener types (e.g. the definition of end trim objects).

With these goals in mind, new RMGScript entities were created as described in the following Section.

1.3.1 The HullStiffener RMGScript Entity

The new top-level RMGScript entity HullStiffener was added for creating hull stiffener objects. HullStiffener required three pieces of information to complete its definition: (a) which object(s) it would attach to, (b) the shape and orientation of its web, and (c) its scantlings and trim options.

All of this information was supplied through either direct attributes or child entities of the HullStiffener. Details are given in the following three Sections.

1.3.1.1 Specifying the objects to attach to

The object(s) to which the HullStiffener object would attach had to be specified in one or more AttachBase child objects. Each AttachBase child contained an ‘objref’ attribute that referred to one or more objects to which the stiffener would attach, and a ‘normside’ flag to indicate whether the stiffener attached to the normal or anti-normal side of the referenced object(s).

1.3.1.2 Defining the shape and orientation of the web

The HullStiffener web shape and orientation could be defined in three ways:

1. by intersection of the attached objects with a plane (see HullStiffenerPlane below)
2. by definition of multiple points on the attached objects (see HullStiffenerPoint below)
3. by importing or otherwise defining a RefCurve that lies on the attached objects (see HullStiffenerCurve below)

1.3.1.2.1 HullStiffenerPlane

The HullStiffenerPlane object was used to create a stiffener web by intersecting a plane with the objects specified in the AttachBase children of the HullStiffener object. It had a single attribute, ‘pl’, which specified the plane to be intersected. The web trace curve was taken to be the intersection line of the plane and the AttachBase objects. The web orientation was taken to be in the given plane.

1.3.1.2.2 HullStiffenerPoint

An ordered collection of two or more HullStiffenerPoint objects could be used to build up the shape of the stiffener web trace curve along the AttachBase objects. The HullStiffenerPoint object provided a variety of ways to define its location:

1. By simply specifying a point location in 3D space – in this case the ‘p’ attribute specified the point (explicitly or by reference/anonymous object).
2. At the intersection of two curves on the AttachBase surfaces – in this case, ‘obj1’ and ‘obj2’ specified two objects that would be intersected with the AttachBase objects to create two curves. The resulting curves were then intersected with each other, and the intersection point became the value of the HullStiffenerPoint (Figure 1). If the two curves intersected in more than one place, two options existed for resolving the ambiguity – specifying a point in the ‘rp’ attribute and the closest intersection to that point would be chosen, or specifying a vector in the ‘rd’ attribute and the point closest to the direction of that vector (relative to the global origin) would be chosen.

3. At a given length along a curve on the AttachBase surfaces – in this case ‘obj1’ specified an object that would be intersected with the AttachBase objects to get a curve. ‘obj2’ and ‘obj3’ specified two mandatory objects that would trim that intersection curve. ‘a’ was the arclength distance to travel along the intersection curve from the intersection with ‘obj2’ to get the point (Figure 2). If there was any ambiguity in determining the intersection curve segment trimmed between ‘obj2’ and ‘obj3’, two options existed for resolving the ambiguity – specifying a point in the ‘rp’ attribute that was on the desired segment, or specifying a vector in the ‘rd’ attribute that pointed towards the segment to keep (from the global origin).
4. At a given index of equal spacing along a curve on the AttachBase surfaces – in this case, ‘obj1’ specified an object that would be intersected with the AttachBase objects to get a curve. ‘obj2’ and ‘obj3’ specified two mandatory objects that would trim that intersection curve. ‘n’ specified the number of equal arclength segments to divide the curve into and ‘i’ specified the index of the segment endpoint to choose (i.e. i=’1’” picked the endpoint of the first segment; Figure 3). If there was any ambiguity in determining the intersection curve segment trimmed between ‘obj2’ and ‘obj3’, two options existed for resolving the ambiguity – specifying a point in the ‘rp’ attribute that was on the desired segment, or specifying a vector in the ‘rd’ attribute that pointed towards the segment to keep.

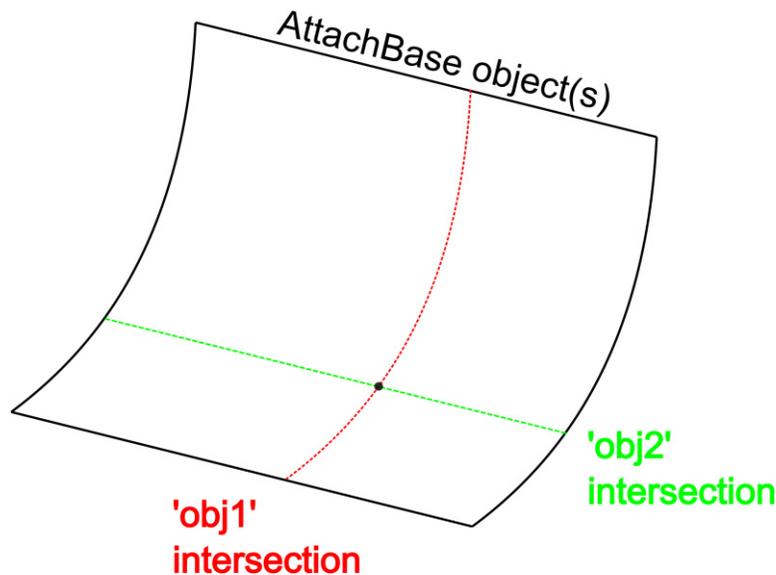


Figure 1: HullStiffenerPoint modeling option using the intersection of two curves.

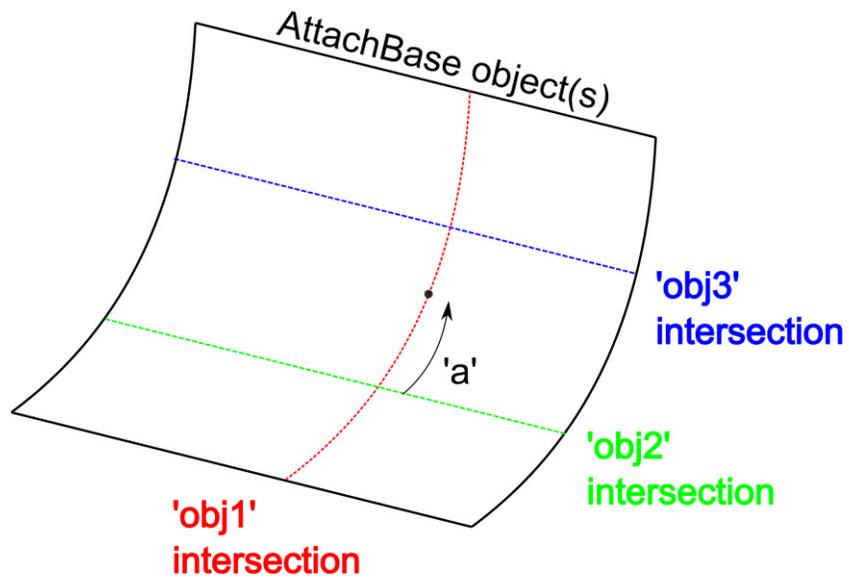


Figure 2: *HullStiffenerPoint* modeling option using an distance along an intersection curve between two other intersection curves.

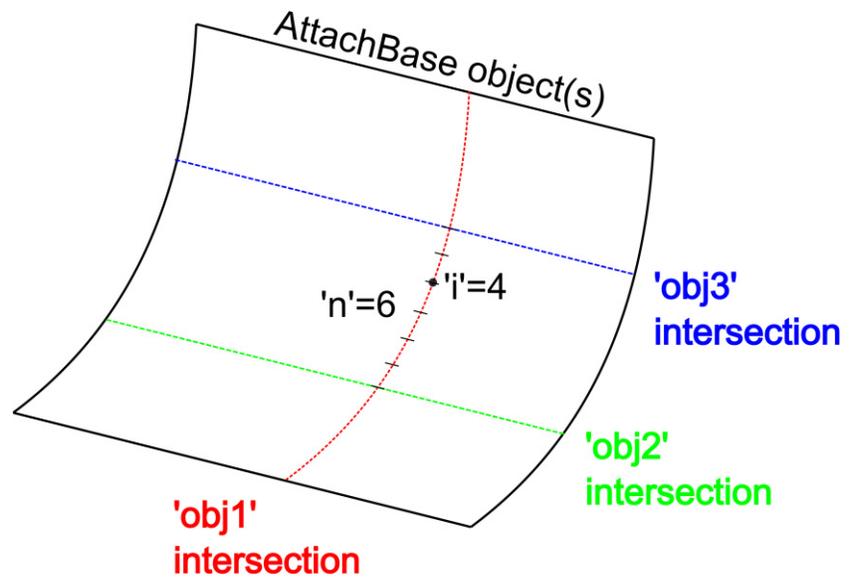


Figure 3: *HullStiffenerPoint* modeling option using equally spaced points along an intersection curve between two other intersection curves.

Along with the point value specified for each HullStiffenerPoint, an optional direction vector could also be specified in the 'd' attribute (explicitly or by reference/anonymous object). The direction vector was used to orient the stiffener web at the corresponding point. If omitted, the orientation would be taken based on the normal of the AttachBase objects' surface at the given point location. If the 'normside' attribute of the AttachBase object was set to "1" (the default), the orientation would be in the surface normal direction at the point location. If 'normside' was set to "0", the orientation would be in the opposite of the surface normal direction.

1.3.1.2.3 HullStiffenerCurve

The HullStiffenerCurve object allowed referencing one or more RefCurve entities that could be used to build up the overall trace curve. The RefCurve object could not only consist of its own analytical or approximated curve definition, but also could support importing a curve from industry-standard Computer Aided Design (CAD) interchange files (IGES) [2]. If any of the HullStiffenerCurve-referenced curves were slightly off the AttachBase objects, they would undergo an automatic "drop" operation to ensure the resulting trace curve was actually on the AttachBase objects.

The orientation for the web would, by default, be chosen as normal to the AttachBase objects, but if a vector was specified for the optional attribute 'd', the web orientation would be in the direction of that vector (along the full length of the web). In the case that multiple HullStiffenerCurve objects were specified to define the trace curve for a single HullStiffener, if any of the HullStiffenerCurve provided a value for 'd', then all had to provide the same value (i.e. 'd' had to be constant for the whole stiffener).

1.3.1.3 Specifying the scantlings and trim options

The attributes belonging to the HullStiffener object itself governed its scantlings, material and end trimming options. Specifically:

e1, e2 – specified the objects at which the stiffener ended. Normally, these objects simply trimmed the stiffener trace edge, and did not further influence the web and flange(s). If 'trim1'/'trim2' attributes were used in conjunction, however, the stiffener web and flange(s) would be trimmed precisely to the objects specified in 'e1'/'e2'

trim1, trim2 – if non-zero, indicated that the stiffener web and flange should be trimmed precisely to the respective 'e1'/'e2' object references. To achieve this, the web and flange surfaces were extrapolated beyond their defined extent to be sure they fully intersected with the 'e1'/'e2' objects. The magnitude of the extrapolation could be fine-tuned by the value specified for 'trim1'/'trim2'. A value of "1" would extrapolate a pre-defined extent which was felt to be sufficient in most cases. Specifying a value greater than "1" would increase the extrapolation distance to handle situations where the normal extrapolation distance was not sufficient. For example, a value of "2" would extrapolate twice the normal distance. It should be noted that the stiffener was extrapolated based only on its defined web trace curve shape, which could lead to

situations where the stiffener diverged from the surface it was attached to; in such cases, it was preferable to explicitly over-define the stiffener trace edge instead of extrapolating.

s – a reference to the stiffener scantling object to be used for the stiffener. Only one ‘s’ attribute was allowed, meaning the scantlings were uniform for the entire length of the stiffener.

m – a reference to the material object to be used for the stiffener

p – an optional point object used to resolve any potential ambiguities in deciding the stiffener trace curve while trimming to the ‘e1’/‘e2’ objects. In such cases, the point would be specified to touch the portion of the trace curve that was to be kept (not used when HullStiffenerPoints were used).

numApproxPts – an optional attribute to specify the number of points to use when dropping HullStiffenerCurve curves to the AttachBase object(s). The default value was 50, but specifying more points was useful if it appeared that the dropped curve still deviated from the hull (or if the stiffener failed to equivalence to the hull).

1.3.1.4 Considerations and Limitations

The following points should be considered with respect to the HullStiffener definition described above:

- HullStiffeners could not traverse discontinuities within (or between) the object(s) to which they were attached (i.e. it was assumed HullStiffeners were applied to smooth surfaces). This had to be maintained even for sections of the stiffener that might be cut away during end trimming (i.e. if the stiffener was being manually “over-defined”).
- If multiple RefCurves were specified through the HullStiffenerCurve option, or if a RefCurve referenced an IGES file containing more than one curve, it was assumed that the curves already formed an ordered, connected path; the software would not rearrange or reorient curves that did not form a proper path. The curves could not contain any gaps, nor form a closed loop.
- When HullStiffenerPoints were used to define the trace curve, the points could not form a closed loop.
- The flange orientation for non-symmetric stiffener profiles (e.g. angle stiffeners) was always chosen as the cross product of the direction of the web trace curve and the web orientation. The direction of the web trace curve was chosen differently depending on the method used to define it. If a collection of HullStiffenerPoints was used, the direction was always in the order of the points. If a HullStiffenerPlane or HullStiffenerCurve was used, the direction was relative to the e1/e2 end trims (e1 at the beginning of the trace curve and e2 at the end).
- HullStiffeners only supported “T”, angle and flatbar stiffener profiles. Closed profiles (pipes, boxes, etc.) were not supported, nor were “I” beams.

- If “advanced trimming” was being applied to only one end of the stiffener, it had to be clear which ends of the defined trace curve corresponded to “end 1” and “end 2”. This was most easily achieved by ensuring that the ends of the trace curve were each closest to only one of the ‘e1’ and ‘e2’ object references.

1.3.2 The HullStiffener Creation Algorithm

The hull stiffener geometry creation algorithm was devised to interpret RMGScript HullStiffener definitions as described in the previous Section and turn them into model geometry. The algorithm consisted of the following sequential process:

1. Beginning with a series of 3D points and associated direction vectors:
2. a series of *rail curves* would be created by first fitting a curve to the 3D points, then fitting curves to translations of the 3D points according to the direction vectors associated with each point and the cross-sectional shape of the stiffener (i.e. a “T” profile, “L” profile, etc.);
3. the rail curves would be grouped into pairs to form the longitudinal edges of faces of the resulting stiffener (i.e. the web face and any applicable flange faces);
4. if ‘trim1’ or ‘trim2’ were specified, the rail curves would be extended to accommodate accurately trimming the stiffener ends;
5. the stiffener face surfaces would be created as ruled surfaces between the pairs of rail curves; both high and low-fidelity representations of the stiffener surfaces would be created;
6. if ‘trim1’ or ‘trim2’ were specified, the stiffener face surfaces would be accurately trimmed to the objects specified at their ends (i.e. ‘e1’ and ‘e2’).

This general creation algorithm was used in all cases to create hull stiffener geometry. In the case that HullStiffenerPoints were specified, they would be gathered and directly passed into step 1 of the algorithm. In the case that HullStiffenerPlane or HullStiffenerCurve entities were used to supply curves directly, they would first be discretized into evenly-spaced points then passed into step 1 of the algorithm.

While it was somewhat odd to consider discretizing a given curve only to use the points to re-approximate that given curve, this strategy was nonetheless used for a couple of reasons. Firstly, there was no ready means to calculate the necessary additional rail curves given only the web trace curve. The curve could not have simply been translated, nor could it have been elastically offset since that operation is limited to curves in plane. As such, the curve would have to be discretized yielding points, those points translated according to the orientation vectors, and a new curve fit to the points. This, however, would yield rail curves that were only approximations of the original curve, and potentially not of identical shape (which could in turn lead to stiffeners of non-uniform depth through their length). Secondly, in order to get the direction vectors in the first place given only a curve, the curve would, again, have to be discretized and the hull orientation calculated at the discretized locations anyway. With both of these points in mind, it

seemed reasonable to simply discretize the given curve at the outset despite the fact that at first glance it appeared to be a step backwards.

1.3.3 Testing Examples

The new HullStiffener object was tested extensively on an example ship hullform. The following Sections illustrate some example RMGScript definitions taken from the testing models that make use of the various modeling options mentioned above.

1.3.3.1 Using direct HullStiffenerPoints

The first example illustrates the definition of a HullStiffener using HullStiffenerPoints where the points are directly specified by {INTPOINT} anonymous objects. Two stiffeners are created, each at a constant z-elevation on the starboard hull. The top stiffener allows the web orientation to follow the shape of the hull, which is the default behavior if the HullStiffenerPoint does not specify a 'd' attribute. The bottom stiffener specifies a constant web orientation (in the port direction) by setting the 'd' attribute. The two stiffeners are shown in Figure 4.

```
<HullStiffener name="Hull longitudinal 1"
  fidelity="[Hull Stiffener Fidelity]" p="" e1="" e2=""
  trim1="" trim2="" s="[Long_T_Side_Shell]" m="[HSS]">
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F4], 0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F5], 0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F6], 0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F7], 0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F8], 0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F9], 0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F10],0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F11],0,z2)},{STARDIR()})}" />
  <HullStiffenerPoint p="{INTPOINT([Hull Plating],{POINT([F12],0,z2)},{STARDIR()})}" />
  <AttachBase objref="[Hull Plating]" normside="1" />
</HullStiffener>

<HullStiffener name="Hull longitudinal 2"
  fidelity="[Hull Stiffener Fidelity]" p="" e1="" e2=""
  trim1="" trim2="" s="[Long_T_Side_Shell]" m="[HSS]">
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F4], 0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F5], 0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F6], 0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F7], 0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F8], 0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F9], 0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F10],0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F11],0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <HullStiffenerPoint p="{INTPOINT([stbd_hull],{POINT([F12],0,z1)},{STARDIR()})}"
    d="{PORTDIR()}" />
  <AttachBase objref="[stbd_hull]" normside="1" />
</HullStiffener>
```

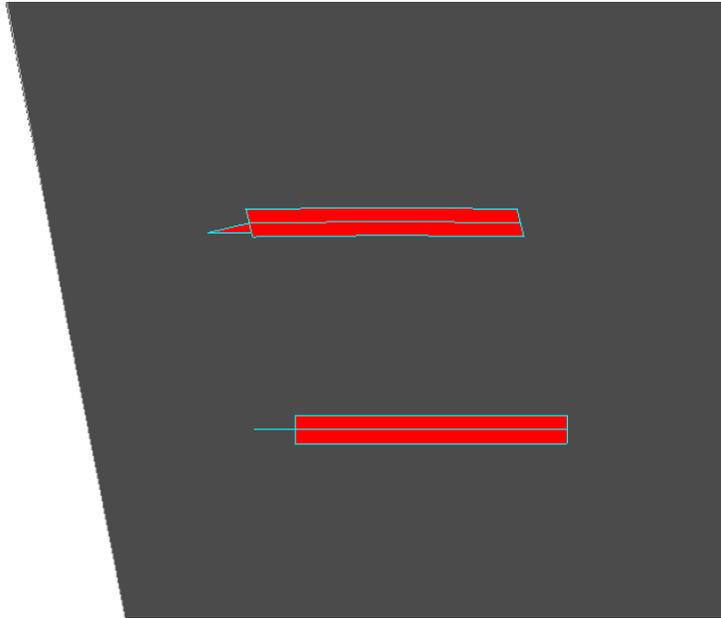


Figure 4: Two HullStiffeners created using the default web orientation following the surface normal (top) and an enforced constant web orientation (bottom).

1.3.3.2 Using HullStiffenerPoint at distance along a curve

The second example illustrates defining a HullStiffener with HullStiffenerPoints that are located at arclength distances along a hull surface. In this case, a series of transverse planar intersections are made with the hull (the 'obj1' attributes) and then trimmed to between two horizontal planes (the 'obj2' and 'obj3' attributes). Arclength distances along the resulting trimmed vertical lines are used to come up with points on the hull (in this case, all points are at the same arclength, a=500). Figure 5 shows four stiffeners modeled using this approach (spaced at multiples of 500).

```

<HullStiffener name="Hull Longitudinal D2-3 2 of 4"
  fidelity="[Hull Stiffener Fidelity]" p="" e1="" e2=""
  trim1="" trim2="" s="[Long_T_Side_Shell]" m="[HSS]">
  <HullStiffenerPoint obj1="{AFTPLANE ([F4])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F5])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F6])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F7])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F8])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F9])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F10])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F11])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <HullStiffenerPoint obj1="{AFTPLANE ([F12])}" obj2="[Deck1]" obj3="[Deck2]" a="500"/>
  <AttachBase objref="[stbd_hull]" normside="1"/>
</HullStiffener>
  
```

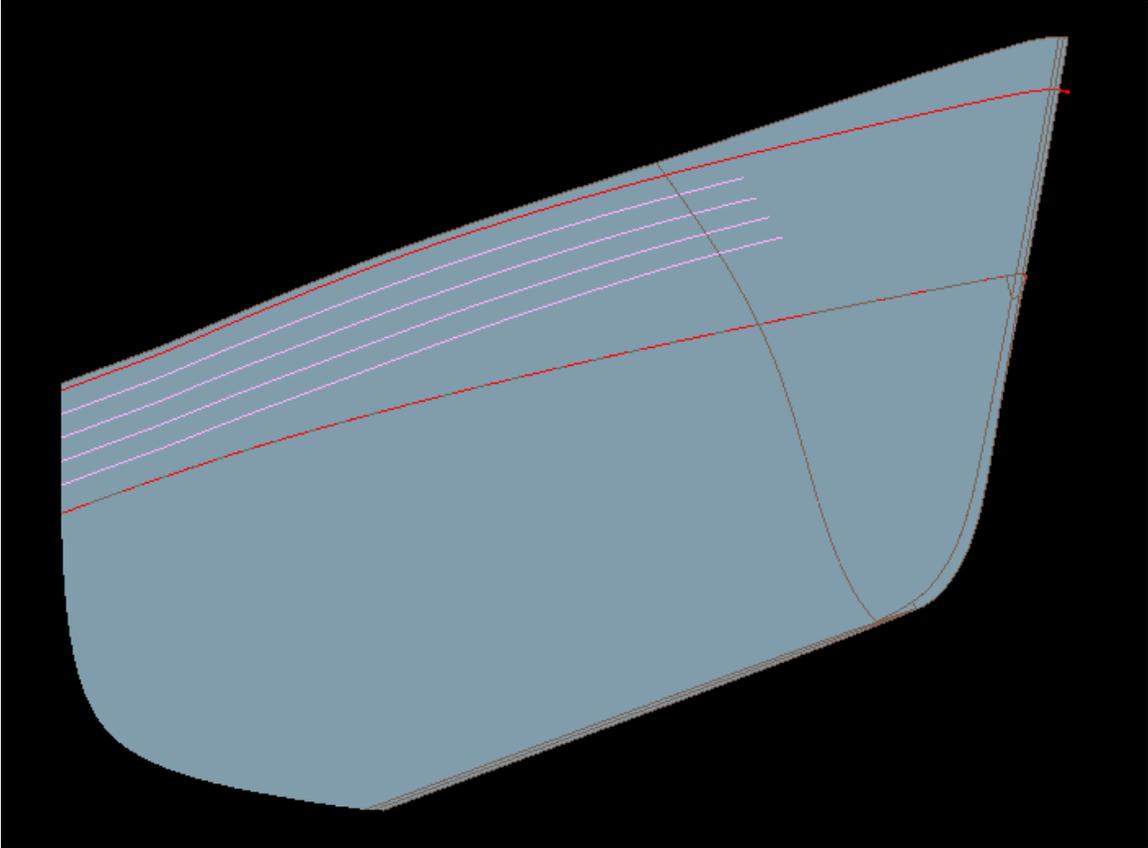


Figure 5: HullStiffeners created using the HullStiffenerPoint arclength option (equal arclength distances measured from upper deck).

1.3.3.3 Using equally spaced HullStiffenerPoints

The third example illustrates defining a HullStiffener using HullStiffenerPoints that are evenly spaced between two objects along a hull surface. In this case, a series of transverse planar intersections are made with the hull (the 'obj1' attributes) and then trimmed to between two decks (the 'obj2' and 'obj3' attributes). The parameter $n=5$ is then used to indicate that each trimmed intersection should be divided into 5 equal portions, and the point at the end of the 2nd portion (i.e. $i=2$) will be the resulting location for the HullStiffenerPoint. Figure 6 shows four equally spaced stiffeners modeled using this approach.

```

<HullStiffener name="Hull Longitudinal D2-3 2 of 4" fidelity="[Hull Stiffener
Fidelity]" p="" e1="" e2="" trim1="" trim2="" s="[Long_T_Side_Shell]" m="[HSS]">
  <HullStiffenerPoint obj1="{AFTPLANE([F4])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F5])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F6])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F7])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F8])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F9])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F10])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F11])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F12])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <HullStiffenerPoint obj1="{AFTPLANE([F13])}" obj2="[Deck1]" obj3="[Deck2]"
    i="2" n="5"/>
  <AttachBase objref="[stbd_hull]" normside="1"/>
</HullStiffener>

```

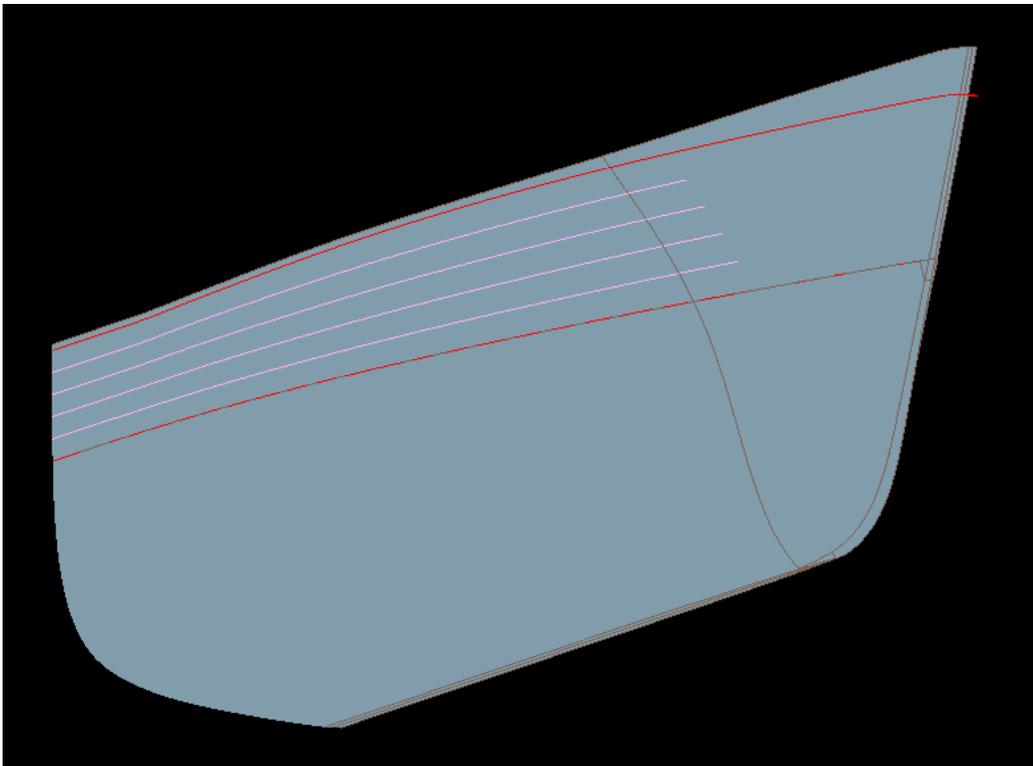


Figure 6: HullStiffeners created using the HullStiffenerPoint evenly spaced modeling option ('n'=5; 'i'=1..4).

1.3.3.4 Using HullStiffenerPlane

The fourth example illustrates defining a HullStiffener using HullStiffenerPlane. In this case, a hull frame is being created at frame location 12 on the starboard side of the hull, between the first and second deck. Because the HullStiffenerPlane will intersect the full hull, the ‘p’ attribute of the HullStiffener is specified to ensure that the curve between “Deck1” and “Deck2” on the *starboard side* is kept.

```
<HullStiffener name="Fr12 Stbd"  
  p="{INTPOINT([Hull1], {POINT([F12],0,8000)}, {STARDIR()})}"  
  e1="[\\Deck1\\]" e2="[\\Deck2\\]"  
  s="[\\Sections\\W250x8_F120x12\\]" m="[\\Materials\\Steel_GradeA\\]"  
  <HullStiffenerPlane pl="{AFTPLANE([F12])}" />  
  <AttachBase objref="[\\Hull\\]" normside="1" />  
</HullStiffener>
```

1.3.3.5 Using HullStiffenerCurve

The final example illustrates defining a HullStiffener using HullStiffenerCurve. In this case, a hull longitudinal is being created between frame location 1 and 80 on the starboard side of the hull. The trace curve for the stiffener is stored in an IGES file, which is imported into a RefCurve child of the HullStiffener and referenced in the HullStiffenerCurve entity.

```
<HullStiffener name="Hull Long 15" e1="{AFTPLANE([F1])}" e2="{AFTPLANE([F80])}"  
  s="[Hull Long]" m="[SteelA]" numApproxPts="50">  
  <AttachBase objref="[stbd_hull]" normside="1" />  
  <RefCurve name="HSC" igesfile="\\IGES_files\\Hull_Longitudinals\\Long-15.igs" />  
  <HullStiffenerCurve c="[HSC]" />  
</HullStiffener>
```

1.4 Conclusions and Future Work

It is concluded that this work has resulted in a flexible capability for modelling general hull stiffeners in Trident Modeller. Each definition scenario identified by reviewing actual drawings has been captured by a modelling option in the resulting RMGScript entities. Each option has been tested and shown to work reliably, and as such the general robustness of the new objects is felt to be in line with the remainder of RMGScript.

It is suggested that support for additional stiffener profiles, particularly bulb profiles, be added in the future. In addition, support for a “medium” fidelity finite element representation – i.e. shell elements for the stiffener web with beam elements representing the flange – should be added.

References

- [1] 2004. SubSAS: An Integrated Suite of Submarine Structural Analysis Codes User's Manual. SM-04-14. Martec Limited. Halifax, NS.
- [2] 1996. *Initial Graphics Exchange Specification: IGES 5.3*, N. Charleston, SC: U.S. Product Data Association, "Formerly an ANSI Standard September 23, 1996 – September 2006"