Sample Design File for Stitch & Glue Boat

The command file that I used to construct my boat is explained here. Text with line numbers in `Courier typeface` are lines of the command file, everything else is editorial description to help you learn its requirements.

```
  1) | This is a wherry design file
```

As illustrated in line 1, a comment is any text after the vertical bar |. Comments may occur anywhere. Blank lines are ignored.

```
  2) NumPoints = 401
  3) StitchInc = 3
  4) StitchInset = 0.3125
```

Line 2 describes the number of points used for analysis. I choose 401 points because I knew the length of my boat would be 200" and thus 401 would space analysis points along the x-axis every 0.5". Lines 3-4 specify where the markers for the stitch holes will be, that is every 3" and inset 0.3125" (5/16") from the edge of the tortured plywood panels.

```
  5) | Note - 1 mil thick epoxy is 0.094 oz/sq-ft
  6) | Observation - The estimate from West Systems seems too heavy because
  7) |   a 12 mil thick finish weights 1.13 oz/sq-ft whereas the estimate is 2.25
  8) |   for an approximate 24 mil thick finish.
  9) | From West Systems, an estimate of epoxy requirements (15% overage)
 10) |   GallonsEpoxy = A x [(Wf x 0.00085) + 0.0075]
 11) |   where A is area, Wf is weight (oz/sq-yd) fiberglass material
 12) |   note specific gravity of cured epoxy resin is 1.18
 13) | Thus weight of epoxy plus fiberglass material per sq-ft -
 14) |   Finish(oz/sq-ft) = 128 * 1.18 * [(Wf x 0.00085) + 0.0075] * 0.85  +  Wf / 9
 15) |                    = 128.4 * [(Wf x 0.00085) + 0.0075]  +  Wf / 9
 16) | For 5.85 oz C-cloth, Finish&Material = 2.25 oz/sq-ft
 17) | For 3.7 oz Deck cloth, Finish&Material = 1.78 oz/sq-ft
 18) HullPlywood=6.63, 0.15, 4mm Okoume plywood
 19) HullInFinish=1.78, .007, #1522 Deck cloth, 3.7 oz/sq-yd, 4.8 mil, epoxy resin
 20) HullExFinish=2.25, .012, #7533 C-cloth, 5.85 oz/sq-yd, 8 mil, epoxy resin
 21) DeckPlywood=6.63, 0.15, 4mm Okoume plywood
 22) DeckInFinish=1.78, .007, #1522 Deck cloth, 3.7 oz/sq-yd, 4.8 mil, epoxy resin
 23) DeckExFinish=1.78, .007, #1522 Deck cloth, 3.7 oz/sq-yd, 4.8 mil, epoxy resin
 24) BulkPlywood=6.63, 0.15, 4mm Okoume plywood
 25) BulkFinish=1.78, .007, #1522 Deck cloth, 3.7 oz/sq-yd, 4.8 mil, epoxy resin
```

Lines 5-17 are reminder comments. Lines 18-25 describe the characteristics of the materials used. Each of these takes 3 comma separated parameters. The first parameter is the weight in ounces/sq-ft, the second parameter is it’s thickness in inches, and the third parameter is a descriptive name.

```
 26) Payload=-100 @(0,4,0) Normal payload stuff

 27) | Rowing Rigg
 28) |   track about 60 inches long
 29) |   seat over mounting plate 7-1/2 inch
 30) |   oarlocks over seat from 5-1/4" to 6-3/4"
 31) |   weight = 18 lb
 32) |   estimated center of gravity, (x=30, y=6, z=0)
 33) |   mounting plate above hull bottom = 1-1/2"
 34) |   assume rowing rigg mounted starting at hull center
```

```
35) Payload=5 @(-13, 12, 0) Rowing Rig (arms only)
36) Payload=1.5  @(-36, 4, 0) Foot stretcher
37) Payload=1.5 @(3, 5, 0) Sliding seat & 32" rails
38) Payload=4.5 @(-13, 6, 0) Solid wood bulkheads for rig
39) Payload=2 @(-15, 5, 0) Wood support for seat & deck
40) Payload=1 @(-20, 1, 0) Wood runner inside at keel (36")

41) | Oars
42) |   material = carbon fiber
43) |   weight each (wild ass guess) = 4
44) |   estimated center of gravity, rigg center, 2" above gunwale
45) Payload=-8 @(-13, 14, 0) Rowing Oars (x2)

46) | density of brass  = 4.4 oz/cu-in
47) | density of brass 1/32" x 0.75" = 1.25 oz/ft
48) | density white oak = 0.44 oz/cu-in
49) | density white oak, 0.75x1.5 = 6 oz/ft
50) | density white oak, 0.5x0.75, 0.25 round corners, 0.25x0.25 slot = 0.61 oz/ft
51) | perimeter of 16 ft long boat approx = 35 ft
52) Payload=3.5 @(0, 11, 0) Deck perimeter (0.5"x0.75" rounded white oak) trim
53) Payload=1.1 @(0, 0, 0) Keel protector (0.75"x0.25" white oak)
54) Payload=1.6 @(0, 0, 0) Keel protector (0.75" wide x 1/32") brass
```

Lines 26-54 describe the weight of various items associated with the boat. These can be physically attached such as the rowing rig (line 35) or things placed in the boat like tent, food, water (line 26). Payload takes 3 groups of parameters: The first parameter is the item weight in pounds, the second parameter after the @ sign and enclosed in parentheses is a comma separated list of the @(x,y,z) position of that item with units of inches, and the third parameter is a descriptive name. If the payload weight is entered as negative (as in line 26), it flags the program to use this item only for stability calculations. So in my boat: Lines 27-40 describe row boat rigging, lines 41-45 describe the oars, and lines 46-54 describe some items attached to the boat to protect the keel and perimeter from the elements.

```
55) | Variable load - i.e. me on rowing rig
56) |   movement relative to rig pin, note movement of legs strapped in foot stretcher
57) |   x-axis center of gravity at catch       => -4 inch (relative to rig pin)
58) |   x-axis center of gravity at release => +12 inch (relative to rig pin)
59) |   rig pin located at -13 inch
60) |   my weight = 165 lb
61) |   assume my center of gravity above seat is 7.5" above seat (@ belly button)
62) Vload=165 @(-17:-1:8, 15, 0:7:1) Rower on sliding seat
```

Lines 55-62 describe the characteristics of a variable position load such as the rower moving forward and back on a rowing rig. The information is used solely for stability analysis. Syntax is similar to Payload except that the position parameter allows for a range of locations. So each x,y,z position is specified as a colon separated list of an initial location, a final location, and an increment. The syntax for each x,y,z is thus @(x0:x1:xinc,y0:y1:yinc,z0:z1:zinc). If the position doesn't change along an axis, then only the initial value need be entered. So in my boat, line 62 described me who weights 165 pounds on a rowing rig that moves along the x-axis from -17 inches to -1 inch in increments of 8 inches; has a fixed y-axis position of 15 inches; and moves along the z-axis from 0 (center-line) to 7 inches in increments of 1 inch. When stability analysis is run, it uses these range limits. Although the range along the x-axis is real because this is the range of the row rig tracks, it is hoped that I as a rower will always be balanced along the center-line but the analysis helps show what will happen if I don't.

```
63) Cockpit = 67.2, -79          | accounts for 75 degree bulkhead
```

The cockpit directive, line 63, describes the section of the boat that has no deck. By providing this information, boat statistics are more comprehensive along with its stability analysis. My boat was designed to resemble a wherry with forward and aft water tight compartments. In between is the cockpit where the rowing rig and me as rower are placed. Bulkheads at these locations finish the water tight compartments.

```
64) Bulkhead=-76, 105, Aft deck BH
65) Bulkhead= 64, 75, Fwd deck BH
66) | BulkheadAft= OR BulkheadFwd DO NOT USE
```

The Bulkhead directive, lines 64-66, takes 3 comma separated parameters. The first is its location on the x-axis at the logical bottom of the boat (y=0). The second parameter is its angle in degrees, where 90 is up & down, less than 90 tilts towards the bow, and greater than 90 tilts toward the stern. For my boat I defined the two bulkheads at either end of the cockpit. I had plans to incorporate explicit bulkheads for Aft and Fwd, but the feature was never implemented even though the command file accepts them (see line 66).

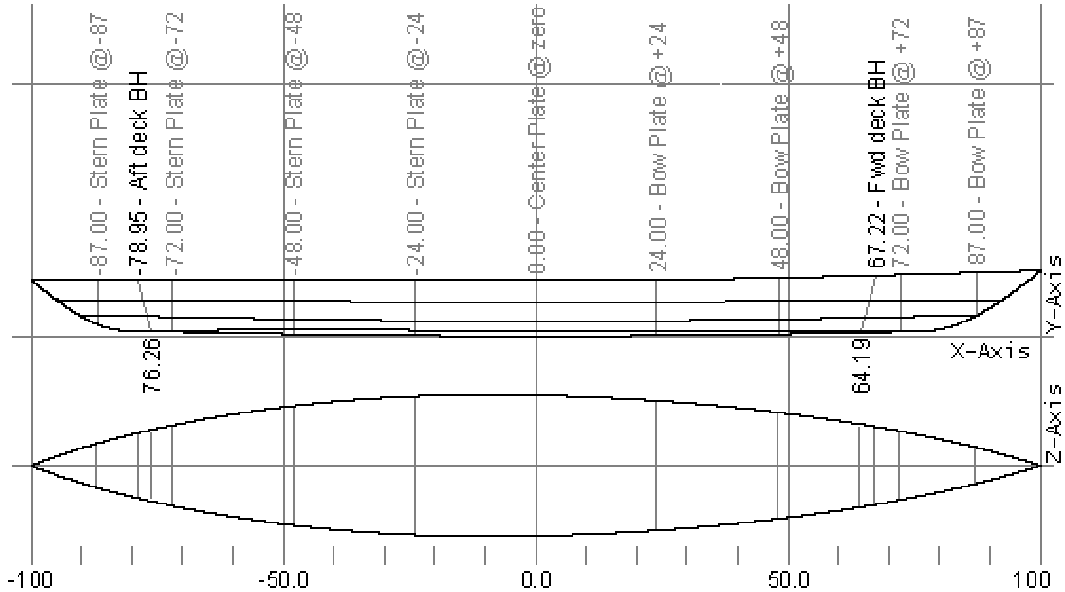
```
67) |Marker= -13, 90, Rigger Pin
68) Marker=-87, 90, Stern Plate @ -87
69) Marker=-72, 90, Stern Plate @ -72
70) Marker=-48, 90, Stern Plate @ -48
71) Marker=-24, 90, Stern Plate @ -24
72) Marker= 0, 90, Center Plate @ zero
73) Marker=24, 90, Bow Plate @ +24
74) Marker=48, 90, Bow Plate @ +48
75) Marker=72, 90, Bow Plate @ +72
76) Marker=87, 90, Bow Plate @ +87
```

Markers (lines 67-76) are essentially phantom bulkheads. These are useful to created templates plates at various positions. Boat analysis ignores them. When I built my boat I originally created templates for plates every 24 inches, but I ultimately only required 3: One in the center and the ones at +/-72 inches. When plotting templates, these markers show up as light gray lines at their position and angle.

```
77) | Template Info
78) TemplateSheet=12, 42
79) TemplateScale=1.0
80) TemplateBanner=TRUE, 0.0
81) TemplateGrid=1.0
```

The Template directives (lines 77-81) provide a means to adjust plot output for a variety of printers. I used an Epson Stylus Photo 1280 which has the ability to print up to 13 inch wide paper of essentially any length. I say essentially because this printer can print at most a sheet size of 13x42, but the sheets may be butted up against each other for continuous output. This method is called Banner mode. So I purchased 12 inch wide roll paper and set up the TemplateSheet to 12 inches wide by 42 inches long. Other related directives are TemplateScale=1.0 which plots full scale, TemplateBanner=TRUE,0.0 which enables banner mode and sets the edge offset to 0.0 inch, and sets the TemplateGrid to 1 inch.

When creating this program, I was more interested in results than making the program a pretty mouse driven windows application. So the output syntax is postscript. These postscript (*.ps) files can be printed directly or they can be converted into PDF files with the appropriate PDF writer. Note that the advantage of converting output to PDF is the file becomes portable, thus you can view it on any computer, print scaled versions on standard paper, or print a full sized version possibly at a local Kinkos store if you don’t own a compatible printer.



The graphic above is a snippet of the report showing two views of the general shape of my boat, these being side and top (bird's eye). As can be seen, my boat is 200" (16' 8") long, has a tear-drop shape that is 28" wide, is taller at bow (13") than stern (11"), and consists of 4 tortured plywood bent panels on each side. The image also identifies the positions of the plate markers at -87", 72", ..., 0.0", ... +87". Also shown are the forward and aft bulkheads, each having a 15° angle from up/down. The x-axis location of the marker and bulkhead intersects to the boat are given, and note that although the forward and aft deck bulkheads are designed to be located at +64 and -76 respectively, they intersect the keel and deck at slightly different locations.

```
 82) Contour                          | top view contour of boat
 83) | Shape[-120:-10] = 28cos(0.72618x  +  7.2618) ^ .5     | 28" wide & 19' long
 84) | Shape[ -10:120] = 28cos(0.60905x  +  6.0905) ^ .5
 85) | Shape[-120:120] = -14

 86) | Shape[-120:-10] = 42cos(0.72618x  +  7.2618) ^ .5     | 42" wide & 19' long
 87) | Shape[ -10:120] = 42cos(0.60905x  +  6.0905) ^ .5
 88) | Shape[-120:120] = -21

 89) | Shape[-110:-10] = 30cos(0.83914x  +  8.3914) ^ .5     | 30" wide & 16.67' long
 90) | Shape[ -10:110] = 30cos(0.68657x  +  6.8657) ^ .5
 91) | Shape[-110:110] = -15

 92)   Shape[-110:-10] = 28cos(0.83914x  +  8.3914) ^ .5     | 28" wide & 16.67' long
 93)   Shape[ -10:110] = 28cos(0.68657x  +  6.8657) ^ .5
 94)   Shape[-110:110] = -14

 95) | Shape[-110:-10] = 30cos(0.87817x  +  8.7817) ^ .5     | 30" wide & 16' long
 96) | Shape[ -10:110] = 30cos(0.71248x  +  7.1248) ^ .5
 97) | Shape[-110:110] = -15

 98) | Shape[-110:-10] = 28cos(0.87817x  +  8.7817) ^ .5     | 28" wide & 16' long
 99) | Shape[ -10:110] = 28cos(0.71248x  +  7.1248) ^ .5
100) | Shape[-110:110] = -14

101) | Shape[-110:-10] = 24cos(0.87817x  +  8.7817) ^ .5     | 24" wide & 16' long
102) | Shape[ -10:110] = 24cos(0.71248x  +  7.1248) ^ .5
103) | Shape[-110:110] = -12
```

Beginning with line 82 the shape of the boat is specified. The first shape that must be given is the top view contour. Lines 82-103 describe the shape of my boat with several commented out versions that I tried and rejected.

Syntax for Shape definitions is somewhat free format and is often entered in piecewise segments. For my boat these segments for the Contour were: [-110:-10] stern shape, [-10:110] bow shape, and [-110:110] entire boat shape. The program takes the sum of all these segments when analyzing a shape. The general form of a shape is:

F(x) = Ax + Bx^m + C(x+D)^N + ... + Ecos(Gx)^P + ... + Hsin(Jx+K)^r + ...

where user supplied constants are entered for A, B, C, etc. Note that shape equations allow sin( ) and cos( ) functions and also allow exponents. Units of x are inches and units of angle are degrees. So in the line 92 shape, 28cos(0.83914x+8.3914)^.5, the number 8.3914 has units of degrees, the number 0.83914 has units of degrees/inch, and the number 28 has units of inches. I raised the cosine function to the power of 0.5 which gave me a more desirable boat contour. You might want to use this as your template, but of course there are other possibilities. Also, when the cosine is analyzed at -10", the result is cos(0)=1.0, and this result matches the cosine of the line 93 shape, thus the boat contour is smooth at this position. When the contour shape function is analyzed, its value crosses the y-axis at x=-100" and then again at x=+100". The segment range [-110:110] is thus slightly larger than required.

There are 3 different forms of shape equations called quadratic, cosine and sine:

Shape[x0:x1]=C(x+A)^Exp
Shape[x0:x1]=Ccos(Dx+P)^Exp
Shape[x0:x1]=Csin(Dx+P)^Exp

where:

- [x0:x1] is valid range of shape
- C=constant multiplier, if not supplied defaults to 1
- Exp=exponent, e.g. 0.5 is sqrt or 2 is square, if blank defaults to:
  - 0 for quadratic if no ’x’, e.g. Shape[ ] = C
  - 1 for quadratic if includes ’x’, e.g. Shape[ ] = Cx
  - 1 for cos or sin, e.g. Shape[ ] = Ccos( )
- A=offset for x, if not supplied defaults to 0
- D=phase multiplier (in degrees/unit x), if not supplied defaults to 1
- P=phase offset (in degrees), if not supplied defaults to 0
- x=x-axis position delimiter
- ^=exponent delimiter

```
104) |HullProfile    | use segmented generated hull profile from HullFabric

105) HullFabric
106)   Name=Top Shape
107)   Rotation = 85
108) | note: height profile puts rig at 5-1/2" above seat, this dimension is
109) | important so as to allow oars to comfortably sit in water during the
110) | stroke and out of the water during the return.  The seat height is
111) | 8 inches above the bottom of the boat where the feet are located.
112)   Shape[-110:110]=11                    | 11" high stern and 13" bow
113)   Shape[   0:110]=-2cos(0.9000x)        | 2" rise from center to bow
114)   Shape[   0:110]=2
115) HullFabric
116)   Name=Upper Side
117)   Shape[-110:110]=0
118)   Shape[-110:110]=-0.5cos(.9x)          | 1/2" flare forward at keel
119) HullFabric
```

```
120)   Name=Lower Side
121)   Shape[-110:110]=0
122)   Shape[-110:110]=-1.5cos(.9x)          | 1" flare forward at keel
123) HullFabric
124)   Name=Keel Line
125)   ProfileUsage=50
126)   Rotation = 5
127)   Shape[-110:110]=-1.5cos(.9x)          | 1-1/2" flare
128)   Shape[-110:110]=1.5
```

Beginning with line 104 the shape of the hull is defined. Whereas the top view is a singular definition by piecewise shapes, the hull profile has multiple Hull Fabric definitions, one for each hull plate. In my boat the hull is composed of four plates on each side.

Syntax for the hull profile follows the top view contour definition. The definitions consist of one set of HullFabric directives for each plate. In my boat I chose the rotation method, another method is discussed from the help option of the makeboat (mb.exe) program if you're interested. With the rotation method each hull fabric definition consists of (i) *HullFabric* directive, (ii) *Name=* directive followed by a descriptive name, (iii) optional *Rotation=* directive followed by the desired angle in degrees of that plate, (iv) optional *ProfileUsage=* directive followed by the desired percentage for the perimeter of the hull at the x=0 axis, (v) one or more *Shape[ ]=* directives to describe what the shape of the plate is when viewed from the starboard.

When the hull fabric is generated, two orthogonal shapes are in play. The first is the shape of the hull when viewing it straight on from the bow as sliced at the x=0 axis, and here directives *Rotation* and *ProfileUsage* are used. The second uses the piecewise *Shape* directives to describe the contour of the plates as viewed straight on from the starboard.

In my boat, command lines 107, 125, and 126 are rotation specifications used to describe what the hull looks like as viewed from the bow at the x=0 axis. It was important to me that the bottom was relatively flat and wide (for stability), hence a rotation of 5 degrees covering 50% of the hull was used. Likewise it was important that the side at the gunwale had a rotation of 85 degrees. Other rotations and profile usage is left blank, and the program generated these for best fit.

The piecewise shape directives are used here in the same manner as used for the top contour, except here it is y=Shape(x). The upper half of the graphic above shows the contours for the four plates while the lower half of this graphic shows the top view contour.

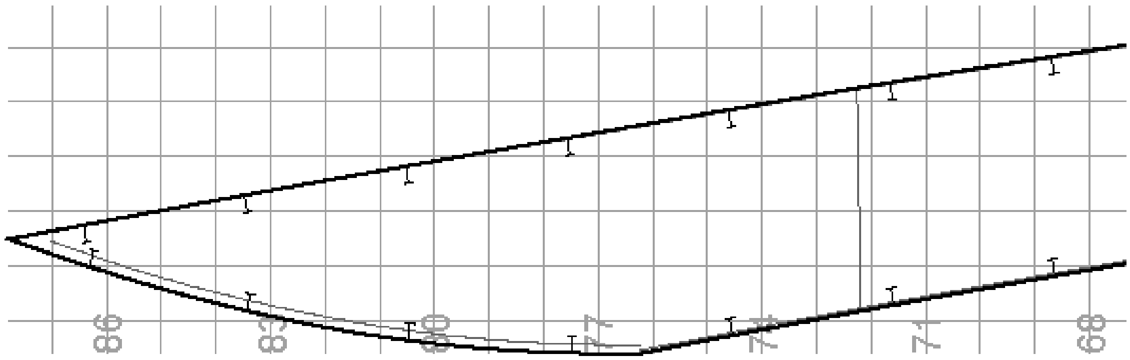
```
129) HullBowLine
130)   Taper = 50.0                          | taper to center
131) | Shape[0:1] = x
132)   Shape[0:1] = sin(90x)

133) HullSternLine
134)   Taper = 50.0                          | taper to center
135) | Shape[0:1] = x
136)   Shape[0:1] = sin(90x)
```

Command lines 129 through 136 define the shape of the bow and stern lines. Although arbitrary shape equations can be used, the use of anything be linear (or near linear) may cause a severe tortured wood problem as I encountered (refer to my build log on date 11/6/04). So even though this command file uses a Shape[0:1]=sin(90x) function, I recommend rather the Shape[0:1]=x be used.

Directives for the bow or stern lines comprise: (i) *HullBowLine* or *HullSternLine* directive, (ii) *Taper=* followed by a percentage of the boat length bow/stern line tortured wood stress is relieved, and (iii) a set of shape equations over the range [0:1].

For my boat I chose a taper of 50%, that is half the boat. The idea here is that some of the sharp bending needed at the bow/stern is uniformly transferred along the length of the hull. But this certainly didn't work adequately when I built my boat using the Shape[0:1]=sin(90x). Oddly, the stress analysis report identified this problem, but since this was my first boat and the first use of this program I was doubtful. Live and learn!



The plot above is a snippet from the stern Lower Side plate. Notable here: (i) Outline of the template, (ii) Stitch holes identified every 3" and inset by 5/16" along the plate perimeter, (iii) Bow plate marker at 72" location, (iv) 1"x1" grid labeled every 3", and (v) Outline of where adjacent plates join on the inside (aka dry or payload) side of the this plate as easily seen by the curved line at the bow edge but is also less visible along the top and bottom edges.

```
137) |DeckProfile    | use segmented generated deck profile from DeckFabric

138) DeckFabric
139)   Name=Deck

140) DeckCenter
141)   Shape[   0:110]=-2cos(0.9000x)        | 2" rise from  center to bow
142)   Shape[   0:110]=2
143)   Shape[-110:110]=11
```

Command lines 138 through 143 describe the deck fabric. The rules here follow the rules for the hull. For my boat the deck is flat and ultimately is located only at sections near the bow and stern.

```
144)   end
```

Line 144 terminates the command file.