MIT WHOLESHIP CONSTRUCTOR CODE

J. Chalfant

MITSG 13-16

Sea Grant College Program
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

NOAA Grant No. NA06OAR4170019 Project No. 2008-ESRDC-01-LEV

MIT WholeShip Constructor Code

By Julie Chalfant
October 31, 2013
MITSG 13-16

A00_wholeship.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MIT WholeShip_v1 Version 1.0 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Last modified: September 5, 2013 by Julie Chalfant
%
% MIT WholeShip_v1 automatically and rapidly constructs a surface warship
% in Paramarine using information from S3D, and produces information
% required in S3D for surface ship design.  This code accomplishes the following:
% • Sets up a standard format for storing the ship design, using U.S.
%   Navy-specific terminology.
% • Using a small set of parameters to define the size of the ship,
%   constructs a ship hull using the “quickhull” capability within Paramarine.
% • Inserts bulkheads and subdivides the hull into watertight compartments.
% • Inserts decks.
% • Constructs a deckhouse.
% • Inserts weight data categorized by ESWBS.
% • Establishes an audit system.
% • Analyzes intact and damaged stability.
%
% This code is designed to be used in conjunction with GRC Paramarine; see
% http://www.grc.qinetiq.com/products/paramarine/Pages/default.aspx
%
% ========================================================================
% MIT WholeShip_v1 constructs a surface warship in Paramarine.
% Copyright (C) 2013 Julie Chalfant
%
%    This program is free software: you can redistribute it and/or modify
%    it under the terms of the GNU General Public License as published by
%    the Free Software Foundation, either version 3 of the License, or
%    (at your option) any later version.
%
%    This program is distributed in the hope that it will be useful,
%    but WITHOUT ANY WARRANTY; without even the implied warranty of
%    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
%    GNU General Public License for more details.
%
%    You should have received a copy of the GNU General Public License
%    along with this program.  If not, see <http://www.gnu.org/licenses/>.
%
% ========================================================================

clc
clear all

%%%%%%%%%%%%%%%%%%%%%%
% Select  Data Files %
%%%%%%%%%%%%%%%%%%%%%%
```

```
shipfilename = 'ship_data.xlsx';
bhdfilename  = 'bhd.xlsx';
dkfilename   = 'decks.xlsx';
ssfilename   = 'superstructure.xlsx';
wtfilename   = 'weight.xlsx';

shipswitch = 1;  % 1 = quickhull_frigate, 2 = quickhull_carrier, 3 = quickhull_mcmv
HullCut = 1;  %1 for yes, 0 for no; if yes, input cut locations in deckhouse data file

%%%%%%%%%%%%%%%%%%
% Default Values %
%%%%%%%%%%%%%%%%%%
Stability_heel_range_degrees = 50; % degrees to each side (+- this value)
Stability_heel_increment_degrees = 5;
bhd_thickness_m = .003;  % used for damage length calc

deckhouse_add_m = 0.15; %to ensure overlap of deckhouse and ship sections
deckhouse_permeability = 55;  %percent
hull_permeability = 55;   %percent

%damage criteria
switch 2
   case 1; damagesw = 1;  % percent damage criteria
      PercentLength = 0.15;
   case 2; damagesw = 2;  % multi compartment damage criteria
      NumDamagedCompt = 3;  %number of consecutive compartments damaged for each scenario
end

SWBSNames = {'Group_1_Structures', 'Group_2_Propulsion', 'Group_3_Electrical',
'Group_4_Control_&_Communication', 'Group_5_Auxiliary_systems', 'Group_6_Outfit_&_Furnishings',
'Group_7_Armament', 'Group_8_Variable_Load'};

sea_water_density_te_per_m3 = 1.025;
fresh_water_density_te_per_m3 = 1.000;
full_tank_level = 0.95;
low_tank_level = 0.05;

%%
%%%%%%%%%%%%%%%%%%
%Initiate KCL File%
%%%%%%%%%%%%%%%%%%
KCLPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research Corporation\Application
Profiles\Paramarine\FilePaths', 'KCLPath');
output_file = strcat(KCLPath,'\KCLReference.kcl');
fileID = fopen(output_file, 'wt+');

fprintf(fileID, 'kcl_version 6\n');
fprintf(fileID, 'deselect all\n');

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SET UP PARAMARINE FORMAT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%
% Insert Primary Files %
%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, 'new placeholder Ship_Data\n');
fprintf(fileID, 'new concept_placeholder Reference\n');
fprintf(fileID, 'new concept_placeholder Supporting_Geometry\n');
fprintf(fileID, 'new concept_placeholder Design\n');
fprintf(fileID, 'new concept_placeholder Audit\n');
fprintf(fileID, 'new concept_placeholder Analysis\n');

%%%%%%%%%%%%%%%%%%%%%
% Insert Ship Data %
%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Ship_Data} new placeholder performance\n');
fprintf(fileID, '{Ship_Data.performance} new var Range\n');
fprintf(fileID, '{Ship_Data.performance.Range} set_units_category (length_major)\n');
fprintf(fileID, '{Ship_Data.performance} new var Endurance_Speed\n');
fprintf(fileID, '{Ship_Data.performance.Endurance_Speed} set_units_category (linear_velocity)\n');
fprintf(fileID, '{Ship_Data.performance} new var Max_Speed\n');
fprintf(fileID, '{Ship_Data.performance.Max_Speed} set_units_category (linear_velocity)\n');

fprintf(fileID, '{Ship_Data} new placeholder hull\n');
fprintf(fileID, '{Ship_Data.hull} new var Displacement\n');
fprintf(fileID, '{Ship_Data.hull.Displacement} set_units_category (mass_default)\n');
fprintf(fileID, '{Ship_Data.hull} new var LBP\n');
fprintf(fileID, '{Ship_Data.hull.LBP} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull} new var Beam\n');
fprintf(fileID, '{Ship_Data.hull.Beam} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull} new var Draft\n');
fprintf(fileID, '{Ship_Data.hull.Draft} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull} new var D0\n');
fprintf(fileID, '{Ship_Data.hull.D0} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull} new var D10\n');
fprintf(fileID, '{Ship_Data.hull.D10} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull} new var D20\n');
fprintf(fileID, '{Ship_Data.hull.D20} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull} new var HOA\n');
fprintf(fileID, '{Ship_Data.hull.HOA} set_units_category (length_default)\n');

fprintf(fileID, '{Ship_Data.hull} new placeholder hullform_generation_data\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var Cp\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var Cm\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var transom_draft\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.transom_draft} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var transom_beam\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.transom_beam} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var skeg_width\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.skeg_width} set_units_category (length_default)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var bow_angle\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.bow_angle} set_units_category (angle)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var transom_angle\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.transom_angle} set_units_category (angle)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var aft_cut_up_dist_fm_AP\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.aft_cut_up_dist_fm_AP} set_units_category
(length_default)\n');
```

```
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var pmb_dist_fwd_of_midships\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.pmb_dist_fwd_of_midships} set_units_category
(length_default)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var pmb_dist_aft_of_midships\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.pmb_dist_aft_of_midships} set_units_category
(length_default)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var entry_coeff\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var run_coeff\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var skeg_area_coeff\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var transom_area_coeff\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var bow_area_coeff\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var LCB_fm_midship_as_percent_LBP\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.LCB_fm_midship_as_percent_LBP} set_units_category
(proportion)\n');
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data} new var CSA_iterations\n');

fprintf(fileID, '{Ship_Data} new placeholder deckhouse\n');
fprintf(fileID, '{Ship_Data.deckhouse} new var beam_offset_for_deckhouse\n');
fprintf(fileID, '{Ship_Data.deckhouse.beam_offset_for_deckhouse} set_units_category (length_default)\n');

fprintf(fileID, '{Ship_Data} new placeholder decks\n');
fprintf(fileID, '{Ship_Data.decks} new placeholder heights\n');
fprintf(fileID, '{Ship_Data.decks} new geom_placeholder planes\n');

fprintf(fileID, '{Ship_Data} new placeholder transverse_bulkheads\n');
fprintf(fileID, '{Ship_Data.transverse_bulkheads} new placeholder locations\n');
fprintf(fileID, '{Ship_Data.transverse_bulkheads} new geom_placeholder planes\n');

%%%%%%%%%%%%%%%%%%%%
% Insert Reference %
%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Reference} new concept_placeholder classification_systems\n');

for iter = 1:length(SWBSNames);
   fprintf(fileID, '{Reference.classification_systems} new classification %s\n',SWBSNames{iter});
end;

fprintf(fileID, '{Reference} new condition_container ship_conditions\n');
fprintf(fileID, '{Reference.ship_conditions} new condition lightship_condition\n');
fprintf(fileID, '{Reference.ship_conditions} new condition min_op_condition\n');
fprintf(fileID, '{Reference.ship_conditions} new condition full_load_condition\n');

fprintf(fileID, '{Reference} new consumables_container densities\n');
fprintf(fileID, '{Reference.densities} new density sea_water\n');
fprintf(fileID, '{Reference.densities.sea_water} =%8.4f[te/m3]\n',sea_water_density_te_per_m3);
fprintf(fileID, '{Reference.densities} new density fresh_water\n');
fprintf(fileID, '{Reference.densities.fresh_water} =%8.4f[te/m3]\n',fresh_water_density_te_per_m3);

fprintf(fileID, '{Reference} new concept_placeholder tank_levels\n');
fprintf(fileID, '{Reference.tank_levels} new var full_tanks\n');
fprintf(fileID, '{Reference.tank_levels.full_tanks} =%d\n',full_tank_level);
fprintf(fileID, '{Reference.tank_levels} new var low_tanks\n');
fprintf(fileID, '{Reference.tank_levels.low_tanks} =%d\n',low_tank_level);
```

```
fprintf(fileID, '{Reference} new service_container service_specs\n');
fprintf(fileID, '{Reference} new concept_placeholder service_line_specifications\n');

%%%%%%%%%%%%%%%%%%%%%%
% Supporting Geometry %
%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%
%      Hull          %
%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Supporting_Geometry} new geom_placeholder hull_generation\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation} new geom_placeholder hull_data\n');

fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data} new CSA_param csa_param\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.dispt_volume}
=Ship_Data.hull.Displacement / Reference.densities.sea_water\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.waterline_aft_x} = -
Ship_Data.hull.LBP\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.pmb_aft_x} = -
Ship_Data.hull.LBP / 2.000 - Ship_Data.hull.hullform_generation_data.pmb_dist_aft_of_midships\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.pmb_fwd_x} = -
Ship_Data.hull.LBP / 2.000 + Ship_Data.hull.hullform_generation_data.pmb_dist_fwd_of_midships\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.waterline_fwd_x} =0.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.Cp}
=Ship_Data.hull.hullform_generation_data.Cp\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.Cm}
=Ship_Data.hull.hullform_generation_data.Cm\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.entry_coeff}
=Ship_Data.hull.hullform_generation_data.entry_coeff\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.run_coeff}
=Ship_Data.hull.hullform_generation_data.run_coeff\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.skeg_area_coeff}
=Ship_Data.hull.hullform_generation_data.skeg_area_coeff\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.transom_area_coeff}
=Ship_Data.hull.hullform_generation_data.transom_area_coeff\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.bow_area_coeff}
=Ship_Data.hull.hullform_generation_data.bow_area_coeff\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.aft_cut_up_x}
=Ship_Data.hull.hullform_generation_data.aft_cut_up_dist_fm_AP - Ship_Data.hull.LBP\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.csa_param.input.LCB}
=Ship_Data.hull.hullform_generation_data.LCB_fm_midship_as_percent_LBP * Ship_Data.hull.LBP -
Ship_Data.hull.LBP/2\n');

fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data} new quickhull0 control_hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data} new quickhull1 ship_hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.target_CSA} -
>Supporting_Geometry.hull_generation.hull_data.csa_param\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.waterplane.z} =1\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.waterplane.d} =Ship_Data.hull.Draft\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.num_CSA_iterations}
=Ship_Data.hull.hullform_generation_data.CSA_iterations\n');

fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.guide_curves.transom} -
>Supporting_Geometry.hull_generation.hull_data.control_hull\n');
```

```
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.guide_curves.midships} -
>Supporting_Geometry.hull_generation.hull_data.control_hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.guide_curves.bow} -
>Supporting_Geometry.hull_generation.hull_data.control_hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.guide_curves.keel_aft} -
>Supporting_Geometry.hull_generation.hull_data.control_hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.guide_curves.deck_aft} -
>Supporting_Geometry.hull_generation.hull_data.control_hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.guide_curves.keel_fwd} -
>Supporting_Geometry.hull_generation.hull_data.control_hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.guide_curves.deck_fwd} -
>Supporting_Geometry.hull_generation.hull_data.control_hull\n');

fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data} new hull_surfaces2 hull_surface\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.hull_surface.port_hull_surface} -
>Supporting_Geometry.hull_generation.hull_data.ship_hull.output_surface\n');

fprintf(fileID, '{Supporting_Geometry.hull_generation} new geom_placeholder hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull} new solid_body hull\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull.hull} from_bounds
(Supporting_Geometry.hull_generation.hull_data.hull_surface.bounds )\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sample Deckhouse Compartment %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Supporting_Geometry} new geom_placeholder deckhouse_generation\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation} new geom_placeholder deckhouse_1\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1} new geom_placeholder
supporting_geometry\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new var
deckhouse_fwd_angle\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_fwd_angle}
set_units_category (angle)\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_fwd_angle} =10.000
[deg]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new var
deckhouse_aft_angle\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_aft_angle}
set_units_category (angle)\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_aft_angle} =-10.000
[deg]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new var
deckhouse_port_angle\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_port_angle}
set_units_category (angle)\n');
```

```
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_port_angle} =10.000
[deg]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new var
deckhouse_stbd_angle\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_stbd_angle}
set_units_category (angle)\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_stbd_angle} =-10.000
[deg]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new point
fwd_port_point\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.x}
= -20.00[m]\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.y}
=Ship_Data.hull.Beam / 2.000 - Ship_Data.deckhouse.beam_offset_for_deckhouse\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.z}
=Ship_Data.hull.D10\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new point
aft_stbd_point\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.x}
= 50.00 [m]\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.y}
=Ship_Data.deckhouse.beam_offset_for_deckhouse - Ship_Data.hull.Beam / 2.000\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.z}
=Ship_Data.hull.D10\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new plane
fwd\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd.x} =cos
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_fwd_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd.y} =0.000\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd.z} =sin
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_fwd_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd.d} =sqrt
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.x *
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.x / 1.000 [m2] +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.z *
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.z / 1.000 [m2]) *
sin (atan (Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.x /
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.z) +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_fwd_angle) [m]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new plane aft\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft.x} =cos
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_aft_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft.y} =0.000\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft.z} =sin
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_aft_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft.d} =sqrt
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.x *
```

```
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.x / 1.000 [m2] +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.z *
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.z / 1.000 [m2]) *
sin (atan (Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.x /
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.z) +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_aft_angle) [m]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new plane
port\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.port.x} =0.000\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.port.y} =cos
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_port_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.port.z} =sin
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_port_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.port.d} =sqrt
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.y *
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.y / 1.000 [m2] +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.z *
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.z / 1.000 [m2]) *
sin (atan (Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.y /
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd_port_point.z) +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_port_angle) [m]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new plane
stbd\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.stbd.x} =0.000\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.stbd.y} =cos
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_stbd_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.stbd.z} =sin
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_stbd_angle)\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.stbd.d} =sqrt
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.y *
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.y / 1.000 [m2] +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.z *
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.z / 1.000 [m2]) *
sin (atan (Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.y /
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft_stbd_point.z) +
Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.deckhouse_stbd_angle) [m]\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry} new
body_bounds body_bounds\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order}
new bound_pointer fwd\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order.fwd
} ->Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.fwd\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order}
new bound_pointer stbd\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order.stbd
} ->Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.stbd\n');
```

```
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order}
new bound_pointer aft\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order.aft}
->Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.aft\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order}
new bound_pointer port\n');
fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds.sides_in_order.port
} ->Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.port\n');

fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1} new solid_body deckhouse_1\n');
fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1.deckhouse_1} from_bounds
(Supporting_Geometry.deckhouse_generation.deckhouse_1.supporting_geometry.body_bounds )\n');

%%%%%%%%%%%%%%%%
% Insert Origin %
%%%%%%%%%%%%%%%%

fprintf(fileID, '{Supporting_Geometry} new geom_placeholder origin\n');
fprintf(fileID, '{Supporting_Geometry.origin} new point origin\n');

%%%%%%%%%%%%%%%%%%%%%%%%%
% Insert Profile Sheets %
%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf(fileID, '{Supporting_Geometry} new geom_placeholder profile_sheet\n');

fprintf(fileID, '{Supporting_Geometry.profile_sheet} new point fwd_high\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.fwd_high.x} =50.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.fwd_high.y} =0.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.fwd_high.z} = Ship_Data.hull.HOA + 20.000 [m]\n');

fprintf(fileID, '{Supporting_Geometry.profile_sheet} new point aft_low\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.aft_low.x} = -Ship_Data.hull.LBP - 50.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.aft_low.y} = 0.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.aft_low.z} = -20.000 [m]\n');

fprintf(fileID, '{Supporting_Geometry.profile_sheet} new sheet profile_sheet_reverse\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.profile_sheet_reverse} rectangle
(Supporting_Geometry.profile_sheet.fwd_high, Supporting_Geometry.profile_sheet.aft_low )\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.profile_sheet_reverse} copy\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet} paste\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet.profile_sheet_reverse_1} rename profile_sheet\n');

fprintf(fileID, '{Supporting_Geometry.profile_sheet} copy\n');
fprintf(fileID, '{Supporting_Geometry} paste\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet_1} rename transverse_sheet\n');
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.profile_sheet} rename transverse_sheet\n');
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.profile_sheet_reverse} rename transverse_sheet_reverse\n');

fprintf(fileID, '{Supporting_Geometry.transverse_sheet.fwd_high} rename port_high\n');
```

```
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.port_high.x} = -Ship_Data.hull.LBP/2\n');
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.port_high.y} = Ship_Data.hull.Beam*2\n');
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.port_high.z} = Ship_Data.hull.HOA + 20.000 [m]\n');

fprintf(fileID, '{Supporting_Geometry.transverse_sheet.aft_low} rename stbd_low\n');
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.stbd_low.x} = -Ship_Data.hull.LBP/2\n');
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.stbd_low.y} = -Ship_Data.hull.Beam*2\n');
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.stbd_low.z} = -20.000 [m]\n');

fprintf(fileID, '{Supporting_Geometry.profile_sheet} copy\n');
fprintf(fileID, '{Supporting_Geometry} paste\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet_1} rename plan_sheet\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.profile_sheet} rename plan_sheet\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.profile_sheet_reverse} rename plan_sheet_reverse\n');

fprintf(fileID, '{Supporting_Geometry.plan_sheet.fwd_high} rename port_fwd\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.port_fwd.x} = 50.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.port_fwd.y} = Ship_Data.hull.Beam*2\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.port_fwd.z} = Ship_Data.hull.Draft\n');

fprintf(fileID, '{Supporting_Geometry.plan_sheet.aft_low} rename stbd_aft\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.stbd_aft.x} = -Ship_Data.hull.LBP - 50.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.stbd_aft.y} = -Ship_Data.hull.Beam*2\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.stbd_aft.z} = Ship_Data.hull.Draft\n');

fprintf(fileID, '{Supporting_Geometry.profile_sheet} copy\n');
fprintf(fileID, '{Supporting_Geometry} paste\n');
fprintf(fileID, '{Supporting_Geometry.profile_sheet_1} rename margin_sheet\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.profile_sheet} rename margin_sheet\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.profile_sheet_reverse} rename margin_sheet_reverse\n');

fprintf(fileID, '{Supporting_Geometry.margin_sheet.fwd_high} rename port_fwd\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.port_fwd.x} = 50.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.port_fwd.y} = Ship_Data.hull.Beam*2\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.port_fwd.z} = Ship_Data.hull.D10 - .075[m]\n');

fprintf(fileID, '{Supporting_Geometry.margin_sheet.aft_low} rename stbd_aft\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.stbd_aft.x} = -Ship_Data.hull.LBP - 50.000 [m]\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.stbd_aft.y} = -Ship_Data.hull.Beam*2\n');
fprintf(fileID, '{Supporting_Geometry.margin_sheet.stbd_aft.z} = Ship_Data.hull.D10 - .075[m]\n');

%%%%%%%%%%%%%%%%%%%%%%%%
%       Design         %
%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Design} new geom_placeholder envelope\n');
fprintf(fileID, '{Design.envelope} new geom_placeholder hull\n');
fprintf(fileID, '{Design.envelope} new geom_placeholder deckhouse\n');

fprintf(fileID, '{Design} new building_block Design\n');
fprintf(fileID, '{Design.Design.attributes.use_sub_blocks_ignore_this}\n');

fprintf(fileID, 'deselect all\n');

%%
```

```
%%%%%%%%%%%%%%%%%%%%%%
%     Input Hull      %
%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%
%  Input Control Hull %
%%%%%%%%%%%%%%%%%%%%%%
switch shipswitch
   case 1; quickhull_frigate
   case 2; quickhull_carrier
   case 3; quickhull_mcmv
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Read Specific Ship Data %
%%%%%%%%%%%%%%%%%%%%%%%%%%
if strfind(shipfilename,'.csv');
   num = csvread(shipfilename,0,1);
elseif strfind(shipfilename,'.xl')
   [num,~] = xlsread(shipfilename);
else
   disp('ERROR:  superstructure filename must be .csv or Excel file');
   return;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%
% performance parameters %
%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Ship_Data.performance.Range} =%d[nm]\n',num(1));
fprintf(fileID, '{Ship_Data.performance.Endurance_Speed} =%d[kt]\n',num(2));
fprintf(fileID, '{Ship_Data.performance.Max_Speed} =%d[kt]\n',num(3));

%%%%%%%%%%%%%%%%%%%%%%%%%%
% principal hull parameters %
%%%%%%%%%%%%%%%%%%%%%%%%%%
LBP_m = num(5);
D10_m = num(9);

fprintf(fileID, '{Ship_Data.hull.Displacement} = %8.4f[te]\n',num(4));
fprintf(fileID, '{Ship_Data.hull.LBP} = %8.4f[m]\n',LBP_m);
fprintf(fileID, '{Ship_Data.hull.Beam} = %8.4f[m]\n',num(6));
fprintf(fileID, '{Ship_Data.hull.Draft} = %8.4f[m]\n',num(7));
fprintf(fileID, '{Ship_Data.hull.D0} = %8.4f[m]\n',num(8));
fprintf(fileID, '{Ship_Data.hull.D10} = %8.4f[m]\n',D10_m);
fprintf(fileID, '{Ship_Data.hull.D20} = %8.4f[m]\n',num(10));

%%%%%%%%%%%%%%%%%%%%%%%%%%
% hullform generation data %
%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.Cp} =%8.4f\n', num(11));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.Cm} =%8.4f\n', num(12));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.transom_draft} =%8.4f[m]\n', num(13));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.transom_beam} =%8.4f[m]\n', num(14));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.skeg_width} =%8.4f[m]\n', num(15));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.bow_angle} =%8.4f[deg]\n', num(16));
```

```
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.transom_angle} =%8.4f[deg]\n', num(17));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.aft_cut_up_dist_fm_AP} =%8.4f[m]\n', num(18));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.pmb_dist_fwd_of_midships} =%8.4f[m]\n', num(19));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.pmb_dist_aft_of_midships} =%8.4f[m]\n', num(20));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.entry_coeff} =%8.4f\n', num(21));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.run_coeff} =%8.4f\n', num(22));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.skeg_area_coeff} =%8.4f\n', num(23));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.transom_area_coeff} =%8.4f\n', num(24));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.bow_area_coeff} =%8.4f\n', num(25));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.LCB_fm_midship_as_percent_LBP} =%8.4f[%%]\n',
num(26));
fprintf(fileID, '{Ship_Data.hull.hullform_generation_data.CSA_iterations} =%8.4f\n', num(27));

clear filename num txt shipsw

%%%%%%%%%%%%%%%%%%%%%%
%  Quickhull 1 Data   %
%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.transom_btm.x} =
(Ship_Data.hull.hullform_generation_data.transom_draft)*tan(Ship_Data.hull.hullform_generation_data.transom_a
ngle)-Ship_Data.hull.LBP\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.transom_btm.y} =
Ship_Data.hull.hullform_generation_data.skeg_width \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.transom_btm.z} =
Ship_Data.hull.Draft-Ship_Data.hull.hullform_generation_data.transom_draft \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.transom_top.x} =
(Ship_Data.hull.Draft - Ship_Data.hull.D20)*tan(Ship_Data.hull.hullform_generation_data.transom_angle)-
Ship_Data.hull.LBP\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.transom_top.y} =
Ship_Data.hull.hullform_generation_data.transom_beam/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.transom_top.z} =
Ship_Data.hull.D10 \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.aft_cut_up.x}  =
Ship_Data.hull.hullform_generation_data.aft_cut_up_dist_fm_AP - Ship_Data.hull.LBP\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.aft_cut_up.y}  =
Ship_Data.hull.hullform_generation_data.skeg_width \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.aft_cut_up.z}  = 0.00 [m]\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_aft_btm.x} = -
Ship_Data.hull.hullform_generation_data.pmb_dist_aft_of_midships- Ship_Data.hull.LBP/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_aft_btm.y} =
Ship_Data.hull.hullform_generation_data.skeg_width \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_aft_btm.z} = 0.00
[m]\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_aft_top.x} = -
Ship_Data.hull.hullform_generation_data.pmb_dist_aft_of_midships- Ship_Data.hull.LBP/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_aft_top.y} =
Ship_Data.hull.Beam/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_aft_top.z} =
Ship_Data.hull.D10 \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_fwd_btm.x} =
Ship_Data.hull.hullform_generation_data.pmb_dist_fwd_of_midships- Ship_Data.hull.LBP/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_fwd_btm.y} =
Ship_Data.hull.hullform_generation_data.skeg_width \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_fwd_btm.z} = 0.00
[m]\n');
```

```
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_fwd_top.x} =
Ship_Data.hull.hullform_generation_data.pmb_dist_fwd_of_midships- Ship_Data.hull.LBP/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_fwd_top.y} =
Ship_Data.hull.Beam/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.pmb_fwd_top.z} =
Ship_Data.hull.D10 \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.bow_bottom.x}  = -
Ship_Data.hull.Draft*tan(Ship_Data.hull.hullform_generation_data.bow_angle)\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.bow_bottom.y}  =
Ship_Data.hull.hullform_generation_data.skeg_width/2\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.bow_bottom.z}  = 0.00
[m]\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.bow_top.x}     =
(Ship_Data.hull.D0 - Ship_Data.hull.Draft)*tan(Ship_Data.hull.hullform_generation_data.bow_angle) \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.bow_top.y}     =
Ship_Data.hull.hullform_generation_data.skeg_width\n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.ship_hull.key_points.bow_top.z}     =
Ship_Data.hull.D10 \n');

fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.hull_surface.deck_camber}        =
Ship_Data.hull.D0 - Ship_Data.hull.D10 \n');
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.hull_surface.x_for_deck_camber}   =
Ship_Data.hull.hullform_generation_data.pmb_dist_fwd_of_midships- Ship_Data.hull.LBP/2 \n');

%%%%%%%%%%%%%%%%%%%%%%%%%
% Insert hull in Design %
%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Supporting_Geometry.hull_generation.hull.hull} copy\n');
fprintf(fileID, '{Design.envelope.hull} paste_as_copy_of\n');
fprintf(fileID, '{Design.envelope.hull.copy_of_hull} rename hull\n');



%%
%%%%%%%%%%%%%%%%%%%%
% Insert Bulkheads %
%%%%%%%%%%%%%%%%%%%%
if strfind(bhdfilename,'.csv');
   num = csvread(bhdfilename,0,1);
elseif strfind(bhdfilename,'.xl')
   [num,txt] = xlsread(bhdfilename);
else
   disp('ERROR:  bulkhead filename must be .csv or Excel file');
   return;
end;

BhdLoc_m = sort(-1*num,'descend');
NumBhd = length(BhdLoc_m);

clear num txt bhdfilename

for iter = 1:NumBhd;
   fprintf(fileID, '{Ship_Data.transverse_bulkheads.locations} new var Bhd_%d\n', iter);
   fprintf(fileID, '{Ship_Data.transverse_bulkheads.locations.Bhd_%d} set_units_category (length_default)\n', iter);
   fprintf(fileID, '{Ship_Data.transverse_bulkheads.locations.Bhd_%d} =%4.2f[m]\n', iter, BhdLoc_m(iter));
```

```
end;

fprintf(fileID, '{Ship_Data.transverse_bulkheads} new geom_placeholder planes\n');
for iter = 1:NumBhd;
   fprintf(fileID, '{Ship_Data.transverse_bulkheads.planes} new plane Bhd_%d\n', iter);
   fprintf(fileID, '{Ship_Data.transverse_bulkheads.planes.Bhd_%d.x} =1\n', iter);
   fprintf(fileID, '{Ship_Data.transverse_bulkheads.planes.Bhd_%d.d}
=Ship_Data.transverse_bulkheads.locations.Bhd_%d\n', iter, iter);
end;

%%
%%%%%%%%%%%%%%%%%%
% Subdivide Hull %
%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Design.envelope.hull.hull} subdivide (Ship_Data.transverse_bulkheads.planes.Bhd_1)\n');
for iter = 2:NumBhd;
   fprintf(fileID, '{Design.envelope.hull.hull.sub_body_%d} subdivide
(Ship_Data.transverse_bulkheads.planes.Bhd_%d)\n', 2*iter-3, iter);
end

%renumber sections
for iter = 1:NumBhd;
   fprintf(fileID, '{Design.envelope.hull.hull.sub_body_%d} rename compt_%d\n',2*iter, iter-1);
   fprintf(fileID, '{Design.envelope.hull.hull.compt_%d.attributes.flooded_permeability} =%d[%%]\n', iter-1,
hull_permeability);
   fprintf(fileID, '{Design.envelope.hull.hull.compt_%d.attributes.tank_permeability} =%d[%%]\n', iter-1,
hull_permeability);
end
fprintf(fileID, '{Design.envelope.hull.hull.sub_body_%d} rename compt_%d\n',2*iter-1, iter);

%%
%%%%%%%%%%%%%%%%%%
%  Insert Decks  %
%%%%%%%%%%%%%%%%%%
if strfind(dkfilename,'.csv')
   num = csvread(dkfilename,0,1);
elseif strfind(dkfilename,'.xl')
   [num,txt] = xlsread(dkfilename);
else
   disp('ERROR:  deck filename must be .csv or Excel file');
   return;
end;

DeckHeight_m = num;
NumDeck = length(DeckHeight_m);
[~, main_deck] = min(abs(DeckHeight_m - D10_m));
clear num txt dkfilename

fprintf(fileID, '{Ship_Data.hull.HOA} = %8.4f [m]\n',max(DeckHeight_m));

for iter = 1:NumDeck;
   fprintf(fileID, '{Ship_Data.decks.heights} new var Deck_%d\n', iter);
   fprintf(fileID, '{Ship_Data.decks.heights.Deck_%d} set_units_category (length_default)\n', iter);
   fprintf(fileID, '{Ship_Data.decks.heights.Deck_%d} =%4.2f[m]\n', iter, DeckHeight_m(iter));
```

```
end;

fprintf(fileID, '{Ship_Data.decks} new geom_placeholder planes\n');
for iter = 1:NumDeck;
   fprintf(fileID, '{Ship_Data.decks.planes} new plane Deck_%d\n', iter);
   fprintf(fileID, '{Ship_Data.decks.planes.Deck_%d.z} =1\n', iter);
   fprintf(fileID, '{Ship_Data.decks.planes.Deck_%d.d} =Ship_Data.decks.heights.Deck_%d\n', iter, iter);
end;

%%
%%%%%%%%%%%%%%%%%%%%%%%%
% Read Deckhouse Data %
%%%%%%%%%%%%%%%%%%%%%%%%
if strfind(ssfilename,'.csv');
   num = csvread(ssfilename,1,1);
elseif strfind(ssfilename,'.xl')
   [num,txt] = xlsread(ssfilename);
else
   disp('ERROR:  superstructure filename must be .csv or Excel file');
   return;
end;

beam_offset_for_deckhouse_m = num(9,1);

deckhouse_fwd_bhd =    num(1,1:end);
deckhouse_aft_bhd =    num(2,1:end);
deckhouse_lower_deck = num(3,1:end);
deckhouse_upper_deck = num(4,1:end);
deckhouse_fwd_angle =  num(5,1:end);
deckhouse_aft_angle =  num(6,1:end);
deckhouse_port_angle = num(7,1:end);
deckhouse_stbd_angle = num(8,1:end);

NumDeckhouse = size(num,2);
if HullCut;
   CutBhd = num(10,1);
   CutDeck = num(11,1);
end;

clear num txt ssfilename

%%%%%%%%%%%%%%%%%%%%%%%%
% Insert Hull Cut %
%%%%%%%%%%%%%%%%%%%%%%%%
if HullCut;
   fprintf(fileID, '{Supporting_Geometry.deckhouse_generation} copy\n');
   fprintf(fileID, '{Supporting_Geometry} paste\n');
   fprintf(fileID, '{Supporting_Geometry.deckhouse_generation_1} rename stepped_deck\n');
   fprintf(fileID, '{Supporting_Geometry.stepped_deck.deckhouse_1} rename cut_box\n');
   fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.deckhouse_1} rename cut_box\n');

   fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.deckhouse_fwd_angle} = 0
[deg]\n');
```

```
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.deckhouse_aft_angle} = 0
[deg]\n');
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.deckhouse_port_angle} = 0
[deg]\n');
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.deckhouse_stbd_angle} = 0
[deg]\n');

    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.fwd_port_point.x} =
Ship_Data.transverse_bulkheads.locations.Bhd_%d\n', CutBhd);
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.fwd_port_point.y} =
Ship_Data.hull.Beam\n');
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.fwd_port_point.z} = 0[m]\n');
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.aft_stbd_point.x} = -
Ship_Data.hull.LBP*2\n');
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.aft_stbd_point.y} = -
(Ship_Data.hull.Beam)\n');
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.aft_stbd_point.z} = 0[m]\n');
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.body_bounds.top} -
>Ship_Data.decks.planes.Deck_%d\n', NumDeck);
    fprintf(fileID, '{Supporting_Geometry.stepped_deck.cut_box.supporting_geometry.body_bounds.bottom} -
>Ship_Data.decks.planes.Deck_%d\n', CutDeck);

    fprintf(fileID, '{Design.envelope.hull.hull} subtract (Supporting_Geometry.stepped_deck.cut_box.cut_box )\n');
end;



%%%%%%%%%%%%%%%%%%%%%
%  Insert Deckhouse  %
%%%%%%%%%%%%%%%%%%%%%

fprintf(fileID, '{Ship_Data.deckhouse.beam_offset_for_deckhouse} =%8.4f [m]\n', beam_offset_for_deckhouse_m);

for iter = 2:NumDeckhouse;
    fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_1} copy\n');
    fprintf(fileID, '{Supporting_Geometry.deckhouse_generation} paste\n');
    fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_%d.deckhouse_1} rename
deckhouse_%d\n', iter, iter);
end;

for iter = 1:NumDeckhouse;
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.deckhouse_fwd_angle} =
%8.4f [deg]\n',  iter,  deckhouse_fwd_angle(iter));
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.deckhouse_aft_angle} = %8.4f
[deg]\n',  iter,  deckhouse_aft_angle(iter));
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.deckhouse_port_angle} =
%8.4f [deg]\n', iter, deckhouse_port_angle(iter));
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.deckhouse_stbd_angle} =
%8.4f [deg]\n', iter, deckhouse_stbd_angle(iter));
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.fwd_port_point.x} =
```

```
Ship_Data.transverse_bulkheads.locations.Bhd_%d + %8.4f[m]\n', iter, deckhouse_fwd_bhd(iter),
deckhouse_add_m);
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.fwd_port_point.y} =
Ship_Data.hull.Beam/2 - Ship_Data.deckhouse.beam_offset_for_deckhouse\n', iter);
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.fwd_port_point.z} =
Ship_Data.decks.heights.Deck_%d\n', iter, main_deck);
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.aft_stbd_point.x} =
Ship_Data.transverse_bulkheads.locations.Bhd_%d - %8.4f[m] \n', iter, deckhouse_aft_bhd(iter),
deckhouse_add_m);
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.aft_stbd_point.y} = -
(Ship_Data.hull.Beam/2 - Ship_Data.deckhouse.beam_offset_for_deckhouse)\n', iter);
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.aft_stbd_point.z} =
Ship_Data.decks.heights.Deck_%d\n', iter, main_deck);
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.body_bounds.top} -
>Ship_Data.decks.planes.Deck_%d\n', iter, deckhouse_upper_deck(iter));
    fprintf(fileID,
'{Supporting_Geometry.deckhouse_generation.deckhouse_%d.supporting_geometry.body_bounds.bottom} -
>Ship_Data.decks.planes.Deck_%d\n', iter, deckhouse_lower_deck(iter));
end;

for iter = 1:NumDeckhouse;
    fprintf(fileID, '{Supporting_Geometry.deckhouse_generation.deckhouse_%d.deckhouse_%d} copy\n', iter, iter);
    fprintf(fileID, '{Design.envelope.deckhouse} paste_as_copy_of\n');
    fprintf(fileID, '{Design.envelope.deckhouse.copy_of_deckhouse_%d} rename deckhouse_%d\n', iter, iter);
    fprintf(fileID, '{Design.envelope.deckhouse.deckhouse_%d} translate (0.000000[m],0.000000[m],-%8.4f[m])\n',
iter, deckhouse_add_m);
    fprintf(fileID, '{Design.envelope.deckhouse.deckhouse_%d.attributes.flooded_permeability} =%d[%%]\n', iter,
deckhouse_permeability);
    fprintf(fileID, '{Design.envelope.deckhouse.deckhouse_%d.attributes.tank_permeability} =%d[%%]\n', iter,
deckhouse_permeability);
end;



%%%%%%%%%%%%%%%%%%%%%%%%
% Update Profie Sheets %
%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, 'deselect all\n');
fprintf(fileID, 'select Supporting_Geometry.profile_sheet.profile_sheet_reverse\n');
fprintf(fileID, 'subtract (Design.envelope.hull.hull )\n');
for iter = 1:NumDeckhouse;
    fprintf(fileID, 'subtract (Design.envelope.deckhouse.deckhouse_%d )\n', iter);
end;
fprintf(fileID, '{Supporting_Geometry.profile_sheet.profile_sheet} subtract
(Supporting_Geometry.profile_sheet.profile_sheet_reverse)\n');

fprintf(fileID, 'deselect all\n');
fprintf(fileID, 'select Supporting_Geometry.transverse_sheet.transverse_sheet_reverse\n');
fprintf(fileID, 'subtract (Design.envelope.hull.hull )\n');
for iter = 1:NumDeckhouse;
```

```
    fprintf(fileID, 'subtract (Design.envelope.deckhouse.deckhouse_%d )\n', iter);
end;
fprintf(fileID, '{Supporting_Geometry.transverse_sheet.transverse_sheet} subtract
(Supporting_Geometry.transverse_sheet.transverse_sheet_reverse)\n');

fprintf(fileID, 'deselect all\n');
fprintf(fileID, 'select Supporting_Geometry.plan_sheet.plan_sheet_reverse\n');
fprintf(fileID, 'subtract (Design.envelope.hull.hull )\n');
fprintf(fileID, '{Supporting_Geometry.plan_sheet.plan_sheet} subtract
(Supporting_Geometry.plan_sheet.plan_sheet_reverse)\n');

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% read unallocated weight data %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if strfind(wtfilename,'.csv');
    num = csvread(wtfilename,0,1);
elseif strfind(wtfilename,'.xl')
    [num,txt] = xlsread(wtfilename);
else
    disp('ERROR:  weight filename must be .csv or Excel file');
    return;
end;

Weight_te = num(1:end,2);
LCG_m = num(1:end,3);
VCG_m = num(1:end,4);

%%%%%%%%%%%%%%%%
% Input Weights %
%%%%%%%%%%%%%%%%
fprintf(fileID, '{Design} new building_block design\n');

fprintf(fileID, '{Design.design.attributes.use_sub_blocks_ignore_this}\n');

for iter = 1:size(num,1);
    fprintf(fileID, '{Design.design} new building_block %s\n', SWBSNames{iter});
    fprintf(fileID, '{Design.design.%s.attributes.characteristics} new char_weight weight\n', SWBSNames{iter});
    fprintf(fileID, '{Design.design.%s.attributes.characteristics.weight.weight} =%6.2f[te]\n',  SWBSNames{iter},
Weight_te(iter));
    fprintf(fileID, '{Design.design.%s.attributes.characteristics.weight.centroid.x_offset} =%6.2f[m]\n',
SWBSNames{iter}, -LCG_m(iter));
    fprintf(fileID, '{Design.design.%s.attributes.characteristics.weight.centroid.z_offset} =%6.2f[m]\n',
SWBSNames{iter}, VCG_m(iter));
    fprintf(fileID, '{Design.design.%s.attributes.characteristics.weight.classification} -
>Reference.classification_systems.%s\n', SWBSNames{iter}, SWBSNames{iter});
end;

%%
%%%%%%%%%%
%  Audit  %
%%%%%%%%%%

fprintf(fileID, '{Audit} new block_summary Total_Ship\n');
```

```
fprintf(fileID, '{Audit.Total_Ship.top_level_block} ->Design.Design\n');

fprintf(fileID, '{Audit} new block_definition block_definition\n');
fprintf(fileID, '{Audit.block_definition.block_summary} ->Audit.Total_Ship\n');

fprintf(fileID, '{Audit} new design_audit Weight\n');
fprintf(fileID, '{Audit.Weight.block_definition} ->Audit.block_definition\n');
fprintf(fileID, '{Audit.Weight.characteristic.weight}\n');
fprintf(fileID, '{Audit.Weight.hierarchy_level.by_level}\n');
fprintf(fileID, '{Audit.Weight.hierarchy_level.level}=2\n');

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% set up stability analysis %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(fileID, '{Analysis} new stab_placeholder Stability\n');
fprintf(fileID, '{Analysis.Stability} new stab_settings stab_settings\n');

%hull envelope
fprintf(fileID, '{Analysis.Stability} new hull_envelope hull_envelope\n');
fprintf(fileID, '{Analysis.Stability.hull_envelope.enable_checking.yes}\n');

fprintf(fileID, '{Analysis.Stability.hull_envelope.buoyant_bodies} new body_pointer hull\n');
fprintf(fileID, '{Analysis.Stability.hull_envelope.buoyant_bodies.hull} ->Design.envelope.hull.hull\n');

for iter = 1:NumDeckhouse;
    fprintf(fileID, '{Analysis.Stability.hull_envelope.buoyant_bodies} new body_pointer deckhouse_%d\n', iter);
    fprintf(fileID, '{Analysis.Stability.hull_envelope.buoyant_bodies.deckhouse_%d} -
>Design.envelope.deckhouse.deckhouse_%d\n', iter, iter);
end

fprintf(fileID, '{Analysis.Stability.hull_envelope.AP_baseline.x} =-Ship_Data.hull.LBP\n');
fprintf(fileID, '{Analysis.Stability.hull_envelope.AM_datum.x} =-Ship_Data.hull.LBP\n');
fprintf(fileID, '{Analysis.Stability.hull_envelope.wind_profile} -
>Supporting_Geometry.profile_sheet.profile_sheet\n');
fprintf(fileID, '{Analysis.Stability.hull_envelope.margin_sheet} -
>Supporting_Geometry.margin_sheet.margin_sheet\n');

%basic ship
fprintf(fileID, '{Audit} %%tree_level (4,Audit)\n');

fprintf(fileID, '{Analysis.Stability} new basic_ship basic_ship\n');
fprintf(fileID, '{Analysis.Stability.basic_ship.stability_settings} ->Analysis.Stability.stab_settings\n');
fprintf(fileID, '{Analysis.Stability.basic_ship.hull_envelope} ->Analysis.Stability.hull_envelope\n');
fprintf(fileID, '{Analysis.Stability.basic_ship.water_density} ->Reference.densities.sea_water\n');
fprintf(fileID, '{Analysis.Stability.basic_ship.datum.weight} =Audit.Weight.results.1_Total.weight\n');
fprintf(fileID, '{Analysis.Stability.basic_ship.datum.centroid.x} =Audit.Weight.results.1_Total.centroid_x\n');
fprintf(fileID, '{Analysis.Stability.basic_ship.datum.centroid.y} =Audit.Weight.results.1_Total.centroid_y\n');
fprintf(fileID, '{Analysis.Stability.basic_ship.datum.centroid.z} =Audit.Weight.results.1_Total.centroid_z\n');

fprintf(fileID, '{Audit} %%tree_level (2,Audit)\n');

%loading conditions
fprintf(fileID, '{Analysis.Stability} new stab_placeholder loading_conditions\n');
```

```
fprintf(fileID, '{Analysis.Stability.loading_conditions} new loading_condition lightship\n');
fprintf(fileID, '{Analysis.Stability.loading_conditions.lightship.basic_ship} ->Analysis.Stability.basic_ship\n');
fprintf(fileID, '{Analysis.Stability.loading_conditions.lightship.water_density} ->Reference.densities.sea_water\n');


%damage scenarios

switch damagesw;
   case 1       %15 percent criteria
      Hold = abs(BhdLoc_m);
      if Hold(1) < 0;  Hold = [Hold(1)-.001; Hold];
      else Hold = [0; Hold]; end;
      if Hold(end) > LBP_m; Hold(end+1) = Hold(end) + .001;
      else Hold(end+1) = LBP_m; end;
      Compt_Percent = (Hold(2:end) - Hold(1:end-1))/LBP_m;
      clear Hold

      DamageList = zeros(NumBhd,2);
      FirstCompt = 1;

      while FirstCompt < NumBhd+1;
         LastCompt = FirstCompt+1;
         Meas = Compt_Percent(LastCompt) + bhd_thickness_m/LBP_m;
         while Meas <= PercentLength && LastCompt <= NumBhd;
            LastCompt = LastCompt+1;
            Meas = Meas + Compt_Percent(LastCompt);
         end
         DamageList(FirstCompt,:) = [FirstCompt, LastCompt];
         FirstCompt = FirstCompt+1;
      end
      qq = find(DamageList(:,2) == NumBhd+1,1,'first');
      DamageList(qq+1:end,:) = [];
   case 2     %multi compartment criteria
      DamageList = zeros(NumBhd-NumDamagedCompt+2,2);
      for FirstCompt = 1:NumBhd-NumDamagedCompt+2;
         DamageList(FirstCompt,:) = [FirstCompt, FirstCompt+NumDamagedCompt-1];
      end
end;
clear FirstCompt LastCompt NumDamagedCompt

fprintf(fileID, '{Analysis.Stability} new stab_placeholder damage_scenarios\n');

for iter = 1:size(DamageList,1);
   fprintf(fileID, '{Analysis.Stability.damage_scenarios} new damage_summary damage_summary_%d\n',iter);
   for itera = DamageList(iter,1):DamageList(iter,2);
      fprintf(fileID, '{Analysis.Stability.damage_scenarios.damage_summary_%d} new damage compt_%d\n',iter,
itera-1);
      fprintf(fileID, '{Analysis.Stability.damage_scenarios.damage_summary_%d.compt_%d} -
>Design.envelope.hull.hull.compt_%d\n',iter, itera-1, itera-1);
   end
end

%runs
fprintf(fileID, '{Analysis.Stability} new stab_placeholder runs\n');
```

```
fprintf(fileID, '{Analysis.Stability.runs} new stab_placeholder intact_light\n');
fprintf(fileID, '{Analysis.Stability.runs.intact_light} new GZ GZ_intact_light\n');
fprintf(fileID, '{Analysis.Stability.runs.intact_light.GZ_intact_light.stability_settings} -
>Analysis.Stability.stab_settings\n');
fprintf(fileID, '{Analysis.Stability.runs.intact_light.GZ_intact_light.load_condition} -
>Analysis.Stability.loading_conditions.lightship\n');
fprintf(fileID, '{Analysis.Stability.runs.intact_light.GZ_intact_light.heel_range} new for_next range\n');
fprintf(fileID, '{Analysis.Stability.runs.intact_light.GZ_intact_light.heel_range.range.by_increment}\n');
fprintf(fileID, '{Analysis.Stability.runs.intact_light.GZ_intact_light.heel_range.range.start} =-
%d[deg]\n',Stability_heel_range_degrees);
fprintf(fileID, '{Analysis.Stability.runs.intact_light.GZ_intact_light.heel_range.range.stop}
=%d[deg]\n',Stability_heel_range_degrees);
fprintf(fileID, '{Analysis.Stability.runs.intact_light.GZ_intact_light.heel_range.range.increment}
=%d[deg]\n',Stability_heel_increment_degrees);
fprintf(fileID, '{Analysis.Stability.runs.intact_light} new stab_ass_USN_wind USN_wind\n');
fprintf(fileID, '{Analysis.Stability.runs.intact_light.USN_wind.GZ_curve} -
>Analysis.Stability.runs.intact_light.GZ_intact_light\n');

for iter = 1:size(DamageList,1);
    fprintf(fileID, '{Analysis.Stability.runs.intact_light} copy\n');
    fprintf(fileID, '{Analysis.Stability.runs} paste\n');
    fprintf(fileID, '{Analysis.Stability.runs.intact_light_1} rename damage_%d\n', iter);
    fprintf(fileID, '{Analysis.Stability.runs.damage_%d.GZ_intact_light} rename GZ_damage_%d\n', iter, iter);
    fprintf(fileID, '{Analysis.Stability.runs.damage_%d.GZ_damage_%d.damage_case} -
>Analysis.Stability.damage_scenarios.damage_summary_%d\n', iter, iter, iter);
end;



fprintf(fileID, '{Analysis.Stability.runs} %%tree_level (2,Analysis.Stability.runs)\n');
fprintf(fileID, '{Design.envelope} %%tree_level (2,Design.envelope)\n');
fprintf(fileID, '{Design.envelope.hull} %%tree_level (2,Design.envelope.hull)\n');
%fprintf(fileID, '{Audit} %%tree_level (0,Audit)\n');

%%
%%%%%%%%%%%%%%%%%%%%%%%%
% Pass to Paramarine  %
%%%%%%%%%%%%%%%%%%%%%%%%

fclose(fileID);
MessengerPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research
Corporation\Paramarine7.0', 'bin');
exportKCLcommand = ['"', MessengerPath, '\ParaMessenger.exe" \d ', output_file];
system(exportKCLcommand);
```

A06_power.m

```
%%%%%%%%%%%%%%%%%%
% Powering Values %
%%%%%%%%%%%%%%%%%%

BowBulb = 0;  %1 if present, 0 if not
eta_rr = 1; %relative rotative efficiency
thrust_ded = 0; %thrust deduction factor as a percent
taylor_wake = 0; %Taylor wake fraction as a percent

appendages_percent_of_bare_hull = 10;  %appendage resistance estimate as percent of bare hull

%%
%%%%%%%%%%%%%%%%%%%%
% Initiate KCL File %
%%%%%%%%%%%%%%%%%%%%
KCLPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research Corporation\Application
Profiles\Paramarine\FilePaths', 'KCLPath');
output_file = strcat(KCLPath,'\KCLpower.kcl');
fileID = fopen(output_file, 'wt+');

fprintf(fileID, 'kcl_version 6\n');
fprintf(fileID, '\n');

%%%%%%%%%%
% Powering %
%%%%%%%%%%
fprintf(fileID, 'new concept_placeholder Analysis\n');
fprintf(fileID, '{Analysis} new powering_placeholder Power\n');

%environment
fprintf(fileID, '{Analysis.Power} new eff_power_environment environment\n');
fprintf(fileID, '{Analysis.Power.environment.water_density} ->Reference.densities.sea_water\n');
fprintf(fileID, '{Analysis.Power} new powering_data_from_geom powering_data_from_geom\n');

%geometry
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.naked_hull_body} ->Design.envelope.hull.hull\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.AP_baseline.x} =-Ship_Data.hull.LBP\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.transom_x} =-Ship_Data.hull.LBP\n');
if ~BowBulb;  fprintf(fileID, '{Analysis.Power.powering_data_from_geom.bow_bulb_root_x}
=Ship_Data.hull.LBP\n'); end;
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.waterplane.x}
=Analysis.Stability.loading_conditions.lightship.waterplane.x\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.waterplane.y}
=Analysis.Stability.loading_conditions.lightship.waterplane.y\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.waterplane.z}
=Analysis.Stability.loading_conditions.lightship.waterplane.z\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.waterplane.d}
=Analysis.Stability.loading_conditions.lightship.waterplane.d\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.plan_sheet} -
>Supporting_Geometry.plan_sheet.plan_sheet\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.profile_sheet} -
>Supporting_Geometry.profile_sheet.profile_sheet\n');
```

```
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.transverse_sheet} -
>Supporting_Geometry.transverse_sheet.transverse_sheet\n');
%check these
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.forebody_form.V_shaped}\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.afterbody_form.U_shaped}\n');
fprintf(fileID, '{Analysis.Power.powering_data_from_geom.prop_type.twin_strut}\n');

%hull interaction
fprintf(fileID, '{Analysis.Power} new hull_interaction_user_defined hull_interaction_user_defined\n');
fprintf(fileID, '{Analysis.Power.hull_interaction_user_defined.relative_rotative_efficiency} = %6.4f\n',eta_rr);
fprintf(fileID, '{Analysis.Power.hull_interaction_user_defined.thrust_deduction_factor}
=%6.4f[%%]\n',thrust_ded);
fprintf(fileID, '{Analysis.Power.hull_interaction_user_defined.taylor_wake_fraction} =%6.4f
[%%]\n',taylor_wake);

%propeller limits
fprintf(fileID, '{Analysis.Power} new propeller_limits propeller_limits\n');
fprintf(fileID, '{Analysis.Power.propeller_limits.max_tip_speed} =pi()*4.2[m] *150/1[min]*1.5\n');
fprintf(fileID, '{Analysis.Power.propeller_limits.max_diameter} =5[m]\n');
fprintf(fileID, '{Analysis.Power.propeller_limits.max_BAR} =5\n');
fprintf(fileID, '{Analysis.Power.propeller_limits.max_pressure_coeff} =25[GPa]\n');
fprintf(fileID, '{Analysis.Power.propeller_limits.shaft_speed.max} =500[RPM]\n');

%%%%%%%%%%
% Holtrop %
%%%%%%%%%%
fprintf(fileID, '{Analysis.Power} new powering_placeholder Holtrop\n');

%effective power
fprintf(fileID, '{Analysis.Power.Holtrop} new eff_power_Holtrop eff_power_Holtrop\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_Holtrop.define_from_geometry}\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_Holtrop.geometry} -
>Analysis.Power.powering_data_from_geom\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_Holtrop.correlation_allowance} =.001\n');

fprintf(fileID, '{Analysis.Power.Holtrop} new eff_power_appendages eff_power_appendages\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_appendages.approximate}\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_appendages.naked_hull_eff_pow} -
>Analysis.Power.Holtrop.eff_power_Holtrop\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_appendages.proportion_of_naked_hull}
=%d[%%]\n',appendages_percent_of_bare_hull);

fprintf(fileID, '{Analysis.Power.Holtrop} new eff_power eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_total.environmental_data} ->Analysis.Power.environment\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_total.resistance_elements} new eff_power_pointer Holtrop\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_total.resistance_elements.Holtrop} -
>Analysis.Power.Holtrop.eff_power_Holtrop\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_total.resistance_elements} new eff_power_pointer
Appendages\n');
fprintf(fileID, '{Analysis.Power.Holtrop.eff_power_total.resistance_elements.Appendages} -
>Analysis.Power.Holtrop.eff_power_appendages\n');

fprintf(fileID, '{Analysis.Power.Holtrop} new propeller_finder propeller_finder\n');
```

```
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.effective_power} -
>Analysis.Power.Holtrop.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.hull_interaction} -
>Analysis.Power.hull_interaction_user_defined\n');
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.limits} ->Analysis.Power.propeller_limits\n');
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.propeller_series.Wageningen_B}\n');
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.search_method.neither}\n');
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.design_speed} =Ship_Data.performance.Max_Speed\n');
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.num_blades} =5\n');
fprintf(fileID, '{Analysis.Power.Holtrop.propeller_finder.num_propellers} =2\n');

fprintf(fileID, '{Analysis.Power.Holtrop} new shaft_power shaft_power\n');
fprintf(fileID, '{Analysis.Power.Holtrop.shaft_power.effective_power} -
>Analysis.Power.Holtrop.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Holtrop.shaft_power.hull_interaction} -
>Analysis.Power.hull_interaction_user_defined\n');
fprintf(fileID, '{Analysis.Power.Holtrop.shaft_power.propulsor} -
>Analysis.Power.Holtrop.propeller_finder.propeller\n');

%fix these
fprintf(fileID, '{Analysis.Power.Holtrop} new placeholder checks\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks} new var BT_3_to_4\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks.BT_3_to_4} =Ship_Data.hull.Beam / Ship_Data.hull.Draft\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks} new var LB_6_to_9_5\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks.LB_6_to_9_5} =Ship_Data.hull.LBP / Ship_Data.hull.Beam\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks} new var Cp_55_to_67\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks.Cp_55_to_67} =Ship_Data.hull.hullform_generation_data.Cp\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks} new var Fn_lessthan_pt3\n');
fprintf(fileID, '{Analysis.Power.Holtrop.checks.Fn_lessthan_pt3} =Ship_Data.performance.Max_Speed / sqrt
(9.800 * Ship_Data.hull.LBP / 1.000 [m]) / 1.000 [m/s]\n');

%%%%%%%%
% Fung %
%%%%%%%%

fprintf(fileID, '{Analysis.Power.Holtrop} copy\n');
fprintf(fileID, '{Analysis.Power} paste\n');
fprintf(fileID, '{Analysis.Power.Holtrop_1} rename Fung\n');


fprintf(fileID, '{Analysis.Power.Fung} new eff_power_Fung eff_power_Fung\n');

fprintf(fileID, '{Analysis.Power.Fung.eff_power_Fung.define_from_geometry}\n');
fprintf(fileID, '{Analysis.Power.Fung.eff_power_Fung.geometry} -
>Analysis.Power.powering_data_from_geom\n');

fprintf(fileID, '{Analysis.Power.Fung.eff_power_Fung.bow_csa_over_midships_csa} =
Supporting_Geometry.hull_generation.hull_data.ship_hull.output_CSA.output.CSA_table.area.v50/Supporting_Geo
metry.hull_generation.hull_data.ship_hull.output_CSA.output.CSA_table.area.v26\n');
fprintf(fileID, '{Analysis.Power.Fung.eff_power_Fung.lcg_aft_midships} =(-Ship_Data.hull.LBP / 2.000 [m] -
Audit.Total_Ship.totals.weight.1_Lightship_1.centroid_x / 1.000 [m]) / Ship_Data.hull.LBP * 1.000 [m]\n');
fprintf(fileID, '{Analysis.Power.Fung.eff_power_Fung.transom_area_over_midships_csa}
=Supporting_Geometry.hull_generation.hull_data.ship_hull.output_CSA.output.CSA_table.area.v2/Supporting_Geo
metry.hull_generation.hull_data.ship_hull.output_CSA.output.CSA_table.area.v26\n');
```

```
fprintf(fileID, '{Analysis.Power.Fung.eff_power_Fung.transom_beam_over_waterline_beam}
=Ship_Data.hull.hullform_generation_data.transom_beam / Ship_Data.hull.Beam\n');
fprintf(fileID, '{Analysis.Power.Fung.eff_power_Fung.transom_draught_over_mean_draught}
=Ship_Data.hull.hullform_generation_data.transom_draft / Ship_Data.hull.Draft\n');

fprintf(fileID, '{Analysis.Power.Fung.eff_power_appendages.naked_hull_eff_pow} -
>Analysis.Power.Fung.eff_power_Fung\n');
fprintf(fileID, '{Analysis.Power.Fung.eff_power_appendages.proportion_of_naked_hull}
=%d[%%]\n',appendages_percent_of_bare_hull);

fprintf(fileID, '{Analysis.Power.Fung.eff_power_total.resistance_elements.Holtrop} rename Fung\n');
fprintf(fileID, '{Analysis.Power.Fung.eff_power_total.resistance_elements.Fung} -
>Analysis.Power.Fung.eff_power_Fung\n');
fprintf(fileID, '{Analysis.Power.Fung.eff_power_total.resistance_elements.appendages} -
>Analysis.Power.Fung.eff_power_appendages\n');

fprintf(fileID, '{Analysis.Power.Fung.propeller_finder.effective_power} -
>Analysis.Power.Fung.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Fung.shaft_power.effective_power} ->Analysis.Power.Fung.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Fung.shaft_power.propulsor} ->Analysis.Power.Fung.propeller_finder.propeller\n');

%checks
fprintf(fileID, '{Analysis.Power.Fung.checks.BT_3_to_4} rename DT\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.DT} =Analysis.Power.Fung.eff_power_Fung.displacement_volume *
Reference.densities.sea_water * 0.984 / 1.000 [te]\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.LB_6_to_9_5} rename DL_20_to_250\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.DL_20_to_250} =Analysis.Power.Fung.checks.DT /
Ship_Data.hull.LBP / Ship_Data.hull.LBP / Ship_Data.hull.LBP / (3.281) ^ 3.000 * 1000000.000 [m3]\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.Cp_55_to_67} rename BT_2_2_to_5_2\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.BT_2_2_to_5_2}
=Analysis.Power.Fung.eff_power_Fung.waterline_beam / Analysis.Power.Fung.eff_power_Fung.mean_draught\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.Fn_lessthan_pt3} rename Cp_52_to_70\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.Cp_52_to_70} =Ship_Data.hull.hullform_generation_data.Cp\n');
fprintf(fileID, '{Analysis.Power.Fung.checks} new var Cm_62_to_100\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.Cm_62_to_100} =Ship_Data.hull.hullform_generation_data.Cm\n');
fprintf(fileID, '{Analysis.Power.Fung.checks} new var A20Ax_0_to_40\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.A20Ax_0_to_40}
=Analysis.Power.Fung.eff_power_Fung.transom_area_over_midships_csa\n');
fprintf(fileID, '{Analysis.Power.Fung.checks} new var B20Bx_0_to_85\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.B20Bx_0_to_85}
=Analysis.Power.Fung.eff_power_Fung.transom_beam_over_waterline_beam\n');
fprintf(fileID, '{Analysis.Power.Fung.checks} new var T20Tx_0_to_42\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.T20Tx_0_to_42}
=Analysis.Power.Fung.eff_power_Fung.transom_draught_over_mean_draught\n');
fprintf(fileID, '{Analysis.Power.Fung.checks} new var B0Bx_0_to_10\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.B0Bx_0_to_10}
=Analysis.Power.Fung.eff_power_Fung.bow_csa_over_midships_csa\n');
fprintf(fileID, '{Analysis.Power.Fung.checks} new var CWS_14_9_to_16_2\n');
fprintf(fileID, '{Analysis.Power.Fung.checks.CWS_14_9_to_16_2}
=Analysis.Power.Fung.eff_power_Fung.wetted_surface_area * (3.281) ^ 2.000 / 1.000 [m2] / sqrt
(Analysis.Power.Fung.eff_power_Fung.waterline_length * Analysis.Power.Fung.checks.DT / 1.000 [m])\n');

fprintf(fileID, '{Analysis.Power.Fung.eff_power_Holtrop} delete\n');

%%%%%%%%%%
```

```
% Taylor %
%%%%%%%%%%

fprintf(fileID, '{Analysis.Power.Holtrop} copy\n');
fprintf(fileID, '{Analysis.Power} paste\n');
fprintf(fileID, '{Analysis.Power.Holtrop_1} rename Taylor\n');

fprintf(fileID, '{Analysis.Power.Taylor} new eff_power_Taylor eff_power_Taylor\n');

fprintf(fileID, '{Analysis.Power.Taylor.eff_power_Taylor.define_from_geometry}\n');
fprintf(fileID, '{Analysis.Power.Taylor.eff_power_Taylor.geometry} -
>Analysis.Power.powering_data_from_geom\n');
fprintf(fileID, '{Analysis.Power.Taylor.eff_power_Taylor.method.calibrated}\n');
fprintf(fileID, '{Analysis.Power.Taylor.eff_power_Taylor.wetted_surface_area.calculated_by_method}\n');

fprintf(fileID, '{Analysis.Power.Taylor.eff_power_appendages.naked_hull_eff_pow} -
>Analysis.Power.Taylor.eff_power_Taylor\n');
fprintf(fileID, '{Analysis.Power.Taylor.eff_power_appendages.proportion_of_naked_hull}
=%d[%%]\n',appendages_percent_of_bare_hull);

fprintf(fileID, '{Analysis.Power.Taylor.eff_power_total.resistance_elements.Holtrop} rename Taylor\n');
fprintf(fileID, '{Analysis.Power.Taylor.eff_power_total.resistance_elements.Taylor} -
>Analysis.Power.Taylor.eff_power_Taylor\n');
fprintf(fileID, '{Analysis.Power.Taylor.eff_power_total.resistance_elements.appendages} -
>Analysis.Power.Taylor.eff_power_appendages\n');

fprintf(fileID, '{Analysis.Power.Taylor.propeller_finder.effective_power} -
>Analysis.Power.Taylor.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Taylor.shaft_power.effective_power} ->Analysis.Power.Taylor.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Taylor.shaft_power.propulsor} -
>Analysis.Power.Taylor.propeller_finder.propeller\n');

fprintf(fileID, '{Analysis.Power.Taylor.eff_power_Holtrop} delete\n');

%%%%%%%%%%%
% Series64 %
%%%%%%%%%%%

fprintf(fileID, '{Analysis.Power.Holtrop} copy\n');
fprintf(fileID, '{Analysis.Power} paste\n');
fprintf(fileID, '{Analysis.Power.Holtrop_1} rename Series64\n');

fprintf(fileID, '{Analysis.Power.Series64} new eff_power_Series64 eff_power_Series64\n');

fprintf(fileID, '{Analysis.Power.Series64.eff_power_Series64.define_from_geometry}\n');
fprintf(fileID, '{Analysis.Power.Series64.eff_power_Series64.geometry} -
>Analysis.Power.powering_data_from_geom\n');
fprintf(fileID, '{Analysis.Power.Series64.eff_power_Series64.correlation_allowance} =.001\n');

fprintf(fileID, '{Analysis.Power.Series64.eff_power_appendages.naked_hull_eff_pow} -
>Analysis.Power.Series64.eff_power_Series64\n');
fprintf(fileID, '{Analysis.Power.Series64.eff_power_appendages.proportion_of_naked_hull}
=%d[%%]\n',appendages_percent_of_bare_hull);
```

```
fprintf(fileID, '{Analysis.Power.Series64.eff_power_total.resistance_elements.Holtrop} rename Series64\n');
fprintf(fileID, '{Analysis.Power.Series64.eff_power_total.resistance_elements.Series64} -
>Analysis.Power.Series64.eff_power_Series64\n');
fprintf(fileID, '{Analysis.Power.Series64.eff_power_total.resistance_elements.appendages} -
>Analysis.Power.Series64.eff_power_appendages\n');

fprintf(fileID, '{Analysis.Power.Series64.propeller_finder.effective_power} -
>Analysis.Power.Series64.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Series64.shaft_power.effective_power} -
>Analysis.Power.Series64.eff_power_total\n');
fprintf(fileID, '{Analysis.Power.Series64.shaft_power.propulsor} -
>Analysis.Power.Series64.propeller_finder.propeller\n');

%checks
fprintf(fileID, '{Analysis.Power.Series64.checks.Cp_55_to_67} rename CB_35_to_55\n');
fprintf(fileID, '{Analysis.Power.Series64.checks.CB_35_to_55}
=Analysis.Power.powering_data_from_geom.output.Cb\n');
fprintf(fileID, '{Analysis.Power.Series64.checks.LB_6_to_9_5} rename BT_2_to_4\n');
fprintf(fileID, '{Analysis.Power.Series64.checks.BT_2_to_4}
=Analysis.Power.powering_data_from_geom.output.waterline_beam /
((Analysis.Power.powering_data_from_geom.output.draught_at_AP +
Analysis.Power.powering_data_from_geom.output.draught_at_FP) / 2.000)\n');
fprintf(fileID, '{Analysis.Power.Series64.checks.BT_3_to_4} rename DL_if_cb_lt_45_20_to_35\n');
fprintf(fileID, '{Analysis.Power.Series64.checks.DL_if_cb_lt_45_20_to_35}
=Analysis.Power.powering_data_from_geom.output.displacement_volume * Reference.densities.sea_water * 0.984
/ 1.000 [te] / (0.010 * Analysis.Power.powering_data_from_geom.output.waterline_length * 3.281 / 1.000 [m]) ^
3.000\n');
fprintf(fileID, '{Analysis.Power.Series64.checks.Fn_lessthan_pt3} rename DL_if_cb_gt_45_25_to_45\n');
fprintf(fileID, '{Analysis.Power.Series64.checks.DL_if_cb_gt_45_25_to_45}
=Analysis.Power.powering_data_from_geom.output.displacement_volume * Reference.densities.sea_water * 0.984
/ 1.000 [te] / (0.010 * Analysis.Power.powering_data_from_geom.output.waterline_length * 3.281 / 1.000 [m]) ^
3.000\n');

fprintf(fileID, '{Analysis.Power.Series64.eff_power_Holtrop} delete\n');



%%%%%%%%%%%%%%%%%%%%%%%%%%
% Speed Power Visualizers %
%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf(fileID, '{Analysis.Power} new powering_placeholder speed_power\n');

fprintf(fileID, '{Analysis.Power.speed_power} new speed_power_visualiser Holtrop\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop.source} ->Analysis.Power.Holtrop.shaft_power\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop.speeds} new for_next speeds\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop.speeds.speeds.start} =15[kt]\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop.speeds.speeds.stop} =Ship_Data.performance.Max_Speed\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop.speeds.speeds.increment} =1[kt]\n');

fprintf(fileID, '{Analysis.Power.speed_power.Holtrop} copy\n');
fprintf(fileID, '{Analysis.Power.speed_power} paste\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop_1} rename Fung\n');
fprintf(fileID, '{Analysis.Power.speed_power.Fung.source} ->Analysis.Power.Fung.shaft_power\n');
```

```
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop} copy\n');
fprintf(fileID, '{Analysis.Power.speed_power} paste\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop_1} rename Taylor\n');
fprintf(fileID, '{Analysis.Power.speed_power.Taylor.source} ->Analysis.Power.Taylor.shaft_power\n');

fprintf(fileID, '{Analysis.Power.speed_power.Holtrop} copy\n');
fprintf(fileID, '{Analysis.Power.speed_power} paste\n');
fprintf(fileID, '{Analysis.Power.speed_power.Holtrop_1} rename Series64\n');
fprintf(fileID, '{Analysis.Power.speed_power.Series64.source} ->Analysis.Power.Series64.shaft_power\n');

%%%%%%%%%%%%%%%%%%%%
% Pass to Paramarine %
%%%%%%%%%%%%%%%%%%%%

fclose(fileID);
MessengerPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research
Corporation\Paramarine7.0', 'bin');
exportKCLcommand = ['"', MessengerPath, '\ParaMessenger.exe" \d ', output_file];
system(exportKCLcommand);
```

A07_insertequipment.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% read data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%input filename with output from S3D.
eqfilename = 'output.csv';
if strfind(eqfilename,'.csv');
   DataIn = importdata(eqfilename,',',1);
   num = DataIn.data;
   txt = DataIn.textdata;
else if strfind(eqfilename,'.xl');
      [num,txt] = xlsread(eqfilename);
   else
      disp('ERROR:  equipment filename must be .csv or Excel file');
      return;
   end
end



%VariableList is the list of variable names to be used in matlab.
%ColumnHeaderList is the list of column headers in the excel file.
%The names in each list must correspond to one another.
VariableList = genvarname({'Dim_X_m',    'Dim_Y_m',   'Dim_Z_m',    'Loc_X_m',       'Loc_Y_m',
'Loc_Z_m',       'Weight_kg'})';
ColumnHeaderList =   char({'Length (m)', 'Width (m)', 'Height (m)', 'Location X (m)', 'Location Y (m)', 'Location Z
(m)', 'Weight (kg)'});

if ~strcmp(txt{1,1},'Name');
   disp(['First column in ',filename,' must be "Name"']);
   return
elseif ~(strcmp(txt{1,2},'Id') || strcmp(txt{1,2},'ID'));
   disp(['Second column in ',filename,' must be "Id"']);
   return
end;

Name = genvarname(txt(2:end,1));
ID = txt(2:end,2);

for count = 1:size(num,2);
   for counta = 1:length(VariableList);
      if strcmp(txt{1,count+2},strcat(ColumnHeaderList(counta,:)));
         eval([VariableList{counta},' = num(1:end,count);']);
         break
      end;
   end;
end;

clear count counta num txt ColumnHeaderList filename

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% check input
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%

NumItems = length(Name);
SWBS = 100*ones(NumItems,1);

for count = 1:length(VariableList);
   if exist(VariableList{count},'var');
      eval([VariableList{count},'(isnan(', VariableList{count},')) = 0;']);
   else
      eval([VariableList{count},' = zeros(NumItems,1);']);
   end
end

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% input to paramarine
%%%%%%%%%%%%%%%%%%%%%%%%%%%
KCLPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research Corporation\Application
Profiles\Paramarine\FilePaths', 'KCLPath');
output_file = strcat(KCLPath,'\KCLS3D.kcl');
fileID = fopen(output_file, 'wt+');

%Set Up SubBlock
fprintf(fileID, 'kcl_version 6\n');
fprintf(fileID, 'deselect all\n');


fprintf(fileID, '{Design.Design} new building_block S3D_equipment\n');
fprintf(fileID, '{Design.Design.S3D_equipment.attributes.use_sub_blocks_ignore_this}\n');

for i = 1:NumItems
   fprintf(fileID, '{Design.Design.S3D_equipment} new building_block %s\n', Name{i});
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.initial_geometry.cuboid}\n', Name{i});
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.initial_geometry.x_extent}=%d [m]\n', Name{i},
Dim_X_m(i));
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.initial_geometry.y_extent}=%d [m]\n', Name{i},
Dim_Y_m(i));
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.initial_geometry.z_extent}=%d [m]\n', Name{i},
Dim_Z_m(i));
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.datum_point.x}=%d [m]\n', Name{i},
Loc_X_m(i)*-1);
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.datum_point.y}=%d [m]\n', Name{i},
Loc_Y_m(i));
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.datum_point.z}=%d [m]\n', Name{i}, Loc_Z_m(i));
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.characteristics} new char_weight weight\n',
Name{i});
   fprintf(fileID, '{Design.Design.S3D_equipment.%s.attributes.characteristics.weight.weight}=%d [kg]\n',
Name{i}, Weight_kg(i));
   fprintf(fileID, 'select Design.Design.S3D_equipment.%s.attributes.characteristics.weight.classification\n');
   fprintf(fileID, '->%s\n', SWBS(i));
end;

fclose(fileID);
```

```
MessengerPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research
Corporation\Paramarine7.0', 'bin');
exportKCLcommand = ['"', MessengerPath, '\ParaMessenger.exe" \d ', output_file];
system(exportKCLcommand);
```

A111_update_bhd.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Select Bulkhead Data File %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
bhdfilename = 'bhd.xlsx';

%%
%%%%%%%%%%%%%%%%%%%%
% Initiate KCL File %
%%%%%%%%%%%%%%%%%%%%
KCLPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research Corporation\Application
Profiles\Paramarine\FilePaths', 'KCLPath');
output_file = strcat(KCLPath,'\KCLsubdiv.kcl');
fileID = fopen(output_file, 'wt+');

fprintf(fileID, 'kcl_version 6\n');
fprintf(fileID, 'deselect all\n');

%%%%%%%%%%%%%%%%%%%%
% Insert Bulkheads %
%%%%%%%%%%%%%%%%%%%%
if strfind(bhdfilename,'.csv');
   num = csvread(bhdfilename,0,1);
elseif strfind(bhdfilename,'.xl')
   [num,txt] = xlsread(bhdfilename);
else
   disp('ERROR:  bulkhead filename must be .csv or Excel file');
   return;
end;

BhdLoc_x = sort(-1*num,'descend');
NumBhd = length(BhdLoc_x);

clear num txt filename

for iter = 1:NumBhd;
   fprintf(fileID, '{Ship_Data.transverse_bulkheads.locations.Bhd_%d} =%4.2f[m]\n', iter, BhdLoc_x(iter));
end;

%%%%%%%%%%%%%%%%%%%%%%
% Pass to Paramarine %
%%%%%%%%%%%%%%%%%%%%%%

fclose(fileID);
MessengerPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research
Corporation\Paramarine7.0', 'bin');
exportKCLcommand = ['"', MessengerPath, '\ParaMessenger.exe" \d ', output_file];
system(exportKCLcommand);
```

A112_update_decks.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Select Bulkhead Data File %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dkfilename = 'decks.xlsx';

%%
%%%%%%%%%%%%%%%%%%%%
% Initiate KCL File %
%%%%%%%%%%%%%%%%%%%%
KCLPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research Corporation\Application
Profiles\Paramarine\FilePaths', 'KCLPath');
output_file = strcat(KCLPath,'\KCLsubdiv.kcl');
fileID = fopen(output_file, 'wt+');

fprintf(fileID, 'kcl_version 6\n');
fprintf(fileID, 'deselect all\n');

%%%%%%%%%%%%%%%%%%
%  Insert Decks  %
%%%%%%%%%%%%%%%%%%
if strfind(dkfilename,'.csv')
   num = csvread(dkfilename,0,1);
elseif strfind(dkfilename,'.xl')
   [num,txt] = xlsread(dkfilename);
else
   disp('ERROR:  deck filename must be .csv or Excel file');
   return;
end;

DeckHeight_m = num;
NumDeck = length(DeckHeight_m);
clear num txt filename

fprintf(fileID, '{Ship_Data.hull.HOA} = %8.4f [m]\n',max(DeckHeight_m));

for iter = 1:NumDeck;
   fprintf(fileID, '{Ship_Data.decks.heights.Deck_%d} =%4.2f[m]\n', iter, DeckHeight_m(iter));
end;

%%%%%%%%%%%%%%%%%%%%
% Pass to Paramarine %
%%%%%%%%%%%%%%%%%%%%

fclose(fileID);
MessengerPath = winqueryreg('HKEY_CURRENT_USER', 'Software\Graphics Research
Corporation\Paramarine7.0', 'bin');
exportKCLcommand = ['"', MessengerPath, '\ParaMessenger.exe" \d ', output_file];
system(exportKCLcommand);
```

quickhull_carrier.m

```
key_point_names = {'transom_btm';
   'transom_top';
   'aft_cut_up';
   'pmb_aft_btm';
   'pmb_aft_top';
   'pmb_fwd_btm';
   'pmb_fwd_top';
   'bow_bottom';
   'bow_top'};

transom_btm = [-158.496,0.609,9.229];
transom_top = [-158.496,15.374,30.535];
aft_cut_up = [-65.964,-1.272,-0.159];
pmb_aft_btm = [-7.903,16.721,0];
pmb_aft_top = [-7.903,20.364,30.539];
pmb_fwd_btm = [-7.903,16.721,0];
pmb_fwd_top = [-7.903,20.364,30.539];
bow_bottom = [160.243,0.05,0.465];
bow_top = [170.688,5,30.48];

for iter =1:length(key_point_names);
   fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.x} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(1)']));
   fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.y} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(2)']));
   fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.z} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(3)']));
end;

guide_curve_names = {'transom'; 'midships'; 'bow'; 'keel_aft'; 'deck_aft'; 'keel_fwd'; 'deck_fwd'};

transom = [-158.496,0.609,9.229;
-158.496,2.829,9.699;
-158.496,6.209,10.717;
-158.496,12.989,12.53;
-158.496,15.533,16.191;
-158.496,15.51,24.44;
-158.496,15.407,28.428;
-158.496,15.374,30.535];
midships = [-7.903,16.721,0;
-7.903,19.625,0.137;
-7.903,20.297,2.787;
-7.903,20.536,4.925;
-7.903,20.364,17.447;
-7.903,20.364,23.962;
-7.903,20.364,28.362;
-7.903,20.364,30.539];
bow = [160.243,0.05,0.465;
160.011,0.05,3.136;
156.454,0.05,6.757;
155.474,0.05,15.005;
160.954,1.13,20.764;
166.787,2.5,25.042;
```

```
170.688,3.5,27.347;
170.688,5,30.48];
keel_aft = [-158.496,0.609,9.229;
-138.141,0.609,6.487;
-99.694,0.609,0.49;
-68.824,0.609,0;
-68.132,0.609,0;
-43.651,15.403,0;
-17.422,16.721,0;
-7.903,16.721,0];
deck_aft = [-158.496,15.374,30.535;
-148.458,16.427,30.535;
-128.341,18.408,30.535;
-98.047,19.112,30.535;
-67.761,19.601,30.535;
-37.476,20.435,30.535;
-17.279,20.364,30.535;
-7.903,20.364,30.535];
keel_fwd = [-7.903,16.721,0;
3.533,16.721,0;
26.01,14.771,0;
59.518,10.342,0;
92.843,4.505,0;
126.515,1.323,0;
149.079,0.765,0;
160.243,0.05,0.465];
deck_fwd = [-7.903,20.364,30.535;
5.281,20.364,30.535;
31.258,20.493,30.535;
68.407,19.323,30.535;
107.824,17.979,30.535;
170.689,12.19,30.535;
170.001,11.573,30.535;
170.688,5,30.535;
170.688, 5.000,  30.535];

for iter =1:length(guide_curve_names);
   for itera = 1:8;
      fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.x} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,1)']));
      fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.y} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,2)']));
      fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.z} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,3)']));
   end;
end;
```

quickhull_frigate.m

```
key_point_names = {'transom_btm';
   'transom_top';
   'aft_cut_up';
```

```
    'pmb_aft_btm';
    'pmb_aft_top';
    'pmb_fwd_btm';
    'pmb_fwd_top';
    'bow_bottom';
    'bow_top'};

transom_btm = [-62.219,0.102,3.877];
transom_top = [-65.585,6.051,9.164];
aft_cut_up = [-38.301,0.249,-0.192];
pmb_aft_btm = [-5.04,0.259,-0.031];
pmb_aft_top = [-5.04,8.19,9.164];
pmb_fwd_btm = [-5.04,0.259,-0.031];
pmb_fwd_top = [-5.04,8.19,9.164];
bow_bottom = [61.984,0.026,0.545];
bow_top = [70.183,0.326,9.163];


for iter =1:length(key_point_names);
    fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.x} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(1)']));
    fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.y} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(2)']));
    fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.z} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(3)']));
end;

guide_curve_names = {'transom'; 'midships'; 'bow'; 'keel_aft'; 'deck_aft'; 'keel_fwd'; 'deck_fwd'};

transom = [-62.219,0.102,3.877;
    -62.366,0.404,3.877;
    -62.336,2.02,3.962;
    -62.739,3.737,4.532;
    -63.569,5.591,6.083;
    -64.525,6.07,7.749;
    -65.353,6.05,8.935;
    -65.585,6.051,9.164];
midships = [-5.04,0.259,-0.031;
    -5.04,0.196,0.014;
    -5.04,3.03,0.136;
    -5.04,5.744,0.115;
    -5.04,7.503,3.49;
    -5.04,8.015,6.36;
    -5.04,8.156,8.207;
    -5.04,8.19,9.164];
bow = [61.984,0.001,0.545;
    62.183,0.04,0.279;
    63.049,0.091,2.817;
    64.312,0.106,4.768;
    65.955,0.182,6.463;
    67.888,0.263,7.873;
    69.424,0.303,8.82;
    70.183,0.326,9.164];
keel_aft = [-62.219,0.102,3.877;
    -54.107,0.111,3.708;
```

```
    -45.831,0.152,2.833;
    -38.306,0.249,-0.192;
    -38.301,0.249,-0.192;
    -38.296,0.249,-0.192;
    -21.23,0.259,-0.031;
    -5.04,0.259,-0.031];
deck_aft = [-65.585,6.051,9.164;
    -61.698,6.304,9.164;
    -53.929,6.94,9.164;
    -42.257,7.581,9.164;
    -30.576,8.081,9.164;
    -18.885,8.206,9.164;
    -11.091,8.19,9.164;
    -5.04,8.19,9.164];
keel_fwd = [-5.04,0.259,-0.031;
    0.275,0.259,-0.031;
    9.257,0.259,0.134;
    22.525,0.259,0.206;
    36.06,0.259,0.364;
    49.147,0.259,0.412;
    58.354,0.121,0.545;
    61.984,0.026,0.545];
deck_fwd = [-5.04,8.19,9.164;
    -0.969,8.19,9.164;
    8.483,8.191,9.164;
    24.153,7.621,9.164;
    39.779,7.247,9.164;
    55.36,5.575,9.164;
    65.349,2.225,9.164;
    70.183,0.326,9.164];

for iter =1:length(guide_curve_names);
    for itera = 1:8;
        fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.x} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,1)']));
        fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.y} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,2)']));
        fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.z} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,3)']));
    end;
end;
```

quickhull_mcmv.m

```
key_point_names = {'transom_btm';
    'transom_top';
    'aft_cut_up';
    'pmb_aft_btm';
    'pmb_aft_top';
    'pmb_fwd_btm';
    'pmb_fwd_top';
    'bow_bottom';
```

```
    'bow_top'};

transom_btm = [-24.063,0.066,1.959];
transom_top = [-25.037,4.635,6.8];
aft_cut_up =  [-17.038,0.281,0];
pmb_aft_btm = [-2.761,0.277,0];
pmb_aft_top = [-2.761,4.635,6.8];
pmb_fwd_btm = [-2.761,0.277,0];
pmb_fwd_top = [-2.761,4.635,6.8];
bow_bottom =  [21.976,0.073,0];
bow_top =     [25.459,0.473,6.8];

for iter =1:length(key_point_names);
    fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.x} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(1)']));
    fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.y} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(2)']));
    fprintf(fileID, '{Supporting_Geometry.hull_generation.hull_data.control_hull.key_points.%s.z} = %8.4f
[m]\n',key_point_names{iter}, eval([key_point_names{iter}, '(3)']));
end;

guide_curve_names = {'transom'; 'midships'; 'bow'; 'keel_aft'; 'deck_aft'; 'keel_fwd'; 'deck_fwd'};

transom =[-24.063,0.066,1.959 ;
-24.171,0.605,2.02 ;
-24.325,1.588,2.145 ;
-24.43,3.269,2.479 ;
-24.584,4.5,3.46 ;
-24.741,4.605,5.292 ;
-24.887,4.649,6.267 ;
-25.037,4.635,6.8 ];
midships =[-2.761,0.277,0 ;
-2.761,0.886,0.053 ;
-2.761,2.205,0.11 ;
-2.761,3.807,1.194 ;
-2.761,4.5,3.044 ;
-2.761,4.649,4.878 ;
-2.761,4.626,6.179 ;
-2.761,4.635,6.8] ;
bow =[21.976,0.073,0 ;
22.488,0.079,0.001 ;
23.644,0.09,0.353 ;
24.382,0.175,1.957 ;
24.7,0.257,3.597 ;
25.046,0.348,5.201 ;
25.3,0.43,6.272 ;
25.459,0.473,6.8 ];
keel_aft =[-24.063,0.066,1.959 ;
-21.635,0.185,1.657 ;
-19.021,0.274,1.601 ;
-17.043,0.281,0 ;
-17.038,0.281,0 ;
-17.033,0.281,0 ;
-10.015,0.277,0 ;
-2.761,0.277,0 ];
```

```
deck_aft =[-25.037,4.635,6.8 ;
-23.546,4.635,6.8 ;
-20.559,4.635,6.8 ;
-16.065,4.635,6.8 ;
-11.558,4.635,6.8 ;
-7.079,4.635,6.8 ;
-4.088,4.635,6.8 ;
-2.761,4.635,6.8 ];
keel_fwd =[-2.761,0.277,0 ;
-1.29,0.277,0 ;
2.077,0.282,0 ;
7.078,0.281,0 ;
12.075,0.269,0 ;
17.086,0.217,0 ;
20.383,0.127,0 ;
21.976,0.073,0 ];
deck_fwd =[-2.761,4.635,6.8 ;
-0.706,4.635,6.8 ;
3.21,4.664,6.8 ;
9.083,4.589,6.8 ;
14.882,4.654,6.8 ;
20.521,3.624,6.8 ;
23.988,1.596,6.8 ;
25.459,0.473,6.8 ];

for iter =1:length(guide_curve_names);
    for itera = 1:8;
        fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.x} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,1)']));
        fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.y} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,2)']));
        fprintf(fileID,
'{Supporting_Geometry.hull_generation.hull_data.control_hull.guide_curves.%s.control_points.iu%d.z} = %8.4f
[m]\n',guide_curve_names{iter}, itera, eval([guide_curve_names{iter}, '(itera,3)']));
    end;
end;
```

ship_data.xlsx

Range_nm	3000
Endurance_Speed_kt	16
Max_Speed_kt	32
Displacement_mt	10000
LBP_m	144.8300
Beam_m	17.6500
Draft_m	6.7400

D0_m	13.3500
D10_m	13.3500
D20_m	13.3500
prismatic_coefficient	0.5720
midships_coefficient	0.8540
transom_draft_m	1.5000
transom_beam_m	14
skeg_width_m	0.2500
bow_angle_deg	45
transom_angle_deg	30
aft_cut_up_dist_fm_AP_m	22
pmb_dist_fwd_of_midships_m	0
pmb_dist_aft_of_midships_m	0
entry_coeff	1
run_coeff	0
skeg_area_coeff	0
transom_area_coeff	0
bow_area_coeff	0
LCB_fm_midship_as_percent_LBP	-1.4000
CSA_iterations	40

bhd.csv

Bulkhead 01	7.2
Bulkhead 02	16.8
Bulkhead 03	26.3
Bulkhead 04	35.9
Bulkhead 05	46.7
Bulkhead 06	60.86
Bulkhead 07	69.53999
Bulkhead 08	84.12
Bulkhead 09	101.4
Bulkhead 10	111.725
Bulkhead 11	123.1
Bulkhead 12	132
Bulkhead 12	138

decks.csv

Deck 1	1.2
Deck 2	4.1
Deck 3	7.6
Deck 4	11
Deck 5	13.35
Deck 6	15.7
Deck 7	18.7
Deck 8	21.7

superstructure.csv

	Compt 1	Compt 2	Compt 3
fwd bhd	4	4	8
aft bhd	7	5	10
lower deck	5	7	5
upper deck	7	8	7
fwd angle	10	10	10
aft angle	-10	-10	-10
port angle	10	10	10
stbd angle	-10	-10	-10
beam offset (m)	1.5		
cut bhd	12		
cut deck	4		