Probabilistic Evaluation of Tankship Damage in Collision Events

by

Kurtis Wayne Crake
B.S., Physics (1987)
University of Michigan

Submitted to the Department of Ocean Engineering
in Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer

and

Master of Engineering in Ocean Engineering
Joint Program in Marine Environmental Systems
at the

Massachusetts Institute of Technology

and the

Woods Hole Oceanographic Institute

June 1998

Signature of Author ....
Department of Ocean Engineering
May 8, 1998

Certified by ....
Alan J. Brown
Senior Lecturer in Naval Architecture and Marine Engineering
Thesis Supervisor

Accepted by..
J. Kim VanDiver
Chairman, Department Committee on Graduate Students

Accepted by..
Michael S. Triantafyllou
Chairman, Joint Committee for Applied Ocean Science and Engineering

PROBABILISTIC ANALYSIS OF TANKSHIP DAMAGE IN COLLISION EVENTS

by

KURTIS WAYNE CRAKE

Submitted to the Department of Ocean Engineering
on May 8, 1998 in partial fulfillment of the
requirements for the Degree of Naval Engineer and
Master of Science in Ocean Engineering

ABSTRACT

An analytical method for predicting the extent of damage in ship collisions is developed. The method calculates both the longitudinal and transverse extents of damage for the struck ship as a function of collision scenario parameters, such as ship speeds, relative courses, and collision impact point, as well as considering structural details of the ship.

This prediction method, or collision model, is used in a Monte Carlo analysis with probability density functions defining the specific collision scenario parameters for each "case". The Monte Carlo analysis generates a statistically significant number of collision events and results which are applied directly to calculate oil outflow, and from which resultant pdf's for longitudinal and transverse extent of damage are calculated to compare structural concepts.

The collision model scenario inputs are initially "calibrated" using a MARPOL single-hull model by minimizing the difference between the model result damage pdf's and the pdf's specified by the IMO. A quantitative comparison is made between structural models for an intermediate oil-tight deck (or "mid-deck") tanker, and a series of double-hull tankers based on calculated oil outflow parameters.

Of the three ship designs studied, the double-hull series shows the best performance, followed closely by the mid-deck tanker. The single-hull ship results predict both more frequent and larger spills.

Thesis Supervisor: Alan Brown

Title: Senior Lecturer in Naval Architecture and Marine Engineering

Table of Contents

Acknowledgements

I could not have completed this thesis, or many other things in the last three years, without the help of many others. I would like to express my thanks to:

- Chris Levesque, for providing conversation and a sounding board during those long drives to and from Nashua.

- My other fellow 13A students: Allan Andrew, Julie Chalfant,Casey Moton, Tom Laverghetta, Tom Trapp, for their help and insight on problem sets and group projects. I look forward to working with all of you in the future.

- Professor Alan Brown, for guiding me through the "forest" of this thesis, and keeping me from running into most of the "trees".

- CDR Mark Welsh, for pretty much having the answer to any question or problem that came up anytime over the last three years.

Last, but not least, thanks to Beth for remaining supportive through many long, trying weeks, and to Colin for teaching me that life's important lessons can all be learned from Sesame Street.

1. Introduction

1.1. Motivation

The majority of the world's petroleum travels by sea before reaching the ultimate consumer. While this oil is seaborne, it presents a risk to the marine ecosystem and shoreline communities. Occasionally, this risk is highlighted by spectacular and highly publicized events such as the grounding of Exxon Valdez. More often, it is demonstrated through less sensational incidents which result in smaller releases of oil.

In an effort to mitigate this risk, the International Maritime Organization developed regulations and associated guidelines which use a probabilistic method to evaluate the performance of tankships in groundings and collisions. Alternatives to double-hull ship designs must demonstrate performance at least equivalent to double hull "reference" designs included in the regulation. This regulation provides an objective method of assessing the suitability of potential ship designs, but it has some weaknesses:

- The current regulation does not consider structural detail. The only design attribute important to the analysis is subdivision. It is clear that some structural detail beyond this is important in determining the response of the structure in these accidents. An improved regulation would account for differences in structural detail important to structural response.

- The current method of analysis assumes that the transverse and longitudinal extents of damage are independent. This is an inappropriate assumption in the case of collision. An improved regulation would eliminate this assumption or provide a method to incorporate the coupling between longitudinal and transverse damage extents.

- The current regulation is based on collision data from single hull ships. The current regulation uses probability density functions for damage extent to calculate values of extreme and mean outflow, and the probability of zero outflow. These pdf's were developed from data collected from actual ship collisions, but the data only includes collisions involving single hull ships. Applying a pdf developed from single-hull ships to a ship with substantially different characteristics may unfairly penalize the new ship design by not accounting for improved crash resistance. An improved regulation would use pdf's that are developed specifically for the design types under consideration, or eliminate the use of damage extent pdf's altogether.

- The current regulation only includes cases where the hull envelope was breached. Undoubtedly, there have been instances of collision where no hull failure has occurred, and including these cases in the damage extent pdf's may have important effects on the probability of zero outflow and value of mean outflow. An improved regulation would include the effects of cases where the collision did not result in hull failure.

- The current regulation uses pdf's generated from a small sample size. The pdf's in the current regulation were developed from only 52 collisions. Creating a probability distribution from such a limited data set is an arbitrary process because there are many distribution functions that could fit the data equally well. The final pdf's chosen may not accurately reflect the actual probabilistic distributions. An improved regulation would base pdf's on a larger sample size, or not use pdf's at all.

- The current regulation assumes bigger ships suffer more extensive damage. Because of the way the regulation uses pdf's normalized by the struck ship's length and beam, the resulting damage extents are larger when applied to bigger ships. The extent of damage should be related to the energy absorbed in the collision, not the size of the ship suffering damage.

To improve the current regulation, a method must be developed which can rationally predict the performance of a specific ship design in a given collision scenario. Once this is done, the method should be used to predict the performance of the ship design in a large number of scenarios. The specifics of each scenario should be determined randomly. Analysis of a large number of collision scenarios provides pdf's for extent of damage and oil outflow that directly incorporate both the probability that a given collision will result in no outflow, and the coupled nature of longitudinal and transverse damage. This also properly credits designs with enhanced resistance to collision. This thesis develops such a method, and applies it to several ship designs to demonstrate how the current regulations might be improved.

In a larger view, the proper assessment of risk requires a total system approach [7] where the risk is evaluated by separately assessing both the likelihood of an adverse event (such as collision, grounding, accidental operational discharge, etc.) and the consequence of such events. This thesis attempts to help fill in a missing piece of the overall risk picture by providing a rational and quantitative method of assessing the probable collision damage to a ship and resulting oil outflow, given the appropriate input distributions derived from the specific waterway characteristics. This application of the method is not shown here, but would be a straightforward application of the model developed.

1.2. Review of Collision Analysis Methods

Previous studies of this subject have taken four different general approaches: finite element analyses, model tests, analyses based on the "first principles" of structural mechanics, and empirical studies of actual collisions. Each approach has strengths and weaknesses that make it suitable for some applications and unsuitable for others. Each approach is reviewed briefly to assess suitability for the use at hand.

Finite element analyses use sophisticated computer programs to construct models of the ships involved in a collision. By dividing the model structure into many small elements, each of which has behavior dictated by its material properties, the structural response to given loads can be calculated. The entire collision process can be modeled with high precision, including as much structural detail as desired. The drawback to these methods is that development of the finite element models is time consuming, and evaluating the collision process is computationally intensive.

"Model" tests consist of the construction of scale or full size models, usually of only part of a ship's structure. A collision test is conducted using either a wedge-shaped object or another model of a ship's bow as the striking piece. These tests allow carefully controlled initial conditions and extensive post-collision analysis. It is also possible to instrument the model hulls with strain gauges or accelerometers to collect time series data during the collision process. The drawbacks to this method are that only one collision scenario can be studied per model, and that constructing models is both time consuming and expensive compared to analytical methods. Also, if more economical scale models are used, issues related to scaling effects arise. These effects range from the scale effects of added mass to the scale effects of grain size in the material used for model construction.

Progress is being made toward calculating collision effects from the "first principles" of mechanics. Examples of this approach include works by McDermott, et al. [15], Reckling [16], and most recently by Wierzbicki [6]. These approaches emphasize closed form analytical solutions to a variety of sub-problems within the global perspective of ship collision. For example, solutions have been proposed for plate tearing, plate fracture, and failure and resistance of transverse members. These methods and solutions are promising, and predict deformation/energy absorption characteristics well in laboratory tests. They are difficult to apply in the analysis of a "real-world" collision because of the complex interrelationships between modes of failure of each structural member.

V.U. Minorsky conducted the first and best known of the empirical collision studies [2]. Since then others have re-validated Minorsky's original analysis [3] and extended it to the analysis of other ship types [4]. The method consists of relating the energy dissipated in a collision event to the volume of damaged structure. Actual collisions in which the ship speeds, collision angles, and extent of damage are known are used to empirically determine a proportionality constant. This constant relates damage volume to energy dissipation. In the original analysis the collision is assumed to be totally inelastic, and motion is limited to a single degree of freedom. Under these assumptions, a closed form solution for damaged volume can be obtained, and using the known structural details of the ships, extent of damage can be calculated.

Table 1 summarizes the strengths and weaknesses of these four methods.

Method	Advantages	Disadvantages	Suitable?
Finite Element	Precision	Complex Time consuming model construction Computationally intensive Accuracy strongly dependent on appropriateness of modeling	No
Model Tests	Control of conditions Extensive analysis	Expensive One collision per model Scaling problems	No
First Principles	Generality of approach Basis in physical law	Still in development	No
Empirical	Simplicity	Oversimplifies collision process	Yes

Table 1: Summary of collision analysis methods

A thorough review of current research and methodologies was conducted by the Ship Structures Committee [1]. After evaluating each of the analysis methods described above, they concluded that the most promising possibility was to extend Minorsky's original analysis of high-energy collisions by including consideration of shell membrane energy absorption. This is the approach taken here.

1.3. Overview of this Analysis

This section presents an overview of the major elements of this analysis. Each is described in more detail in later sections.

1.3.1. Collision Analysis Method

Of the four collision analysis methods discussed above, only one is suitable for use here. The objectives of this work are to develop an analysis method that includes the effect of important structural detail, yet is simple enough to be readily applied to a large number of collision cases and a number of different ship designs in a relatively short time, so that a statistically valid distribution of results is obtained.

The finite element method of analysis is unsuitable for this application because it requires a substantial amount of time to develop the finite element model. Finite element analyses are computationally intensive, and require a substantial amount of time to complete the calculations for even a single collision. Developing a statistically valid distribution requires thousands of cases. Conducting thousands of finite element studies for each proposed ship design is not a practical approach. Also, the intent of the IMO regulations is to provide an objective means of approving ship designs before they are built. Finite element models could require a potential shipbuilder to expend substantial effort in design development (in order to get the level of detail needed for a finite element model) only to find that the design is not adequate.

Experimental analyses are critical in the development and validation of analytic models, but are not suitable for this application. The time and effort involved in constructing hundreds or thousands of scale models, all correct in structural detail and scale, then subjecting them to collision tests, is prohibitive.

The "first principles" approach is ideal for this application, and is used in the analysis of grounding events with excellent results [8]. The grounding analysis is facilitated by the existence of a well-developed computer program that performs the grounding damage assessment for a given set of inputs. Unfortunately, the application of this kind of analysis to the structural elements of ship side shells, bilges, bottoms and inner bottoms has only just begun[6], and no such computer program currently exists. Development of such a program is beyond the scope of this work.

The empirical analysis method is best suited for this application for now. The empirical method is simple enough to adapt to different ship designs quickly, but is sensitive to major structural details. Calculation of results for a single collision can be done rapidly. Calculation of many collision cases can be done in a reasonable amount of time and support development of a statistically valid sample population. This is the method utilized in this analysis. In particular, the empirical method of Minorsky is used, but the method is extended to provide better prediction of damage in low-energy collisions, and is generalized to allow for three degrees of freedom of motion for each ship.

1.3.2. Scenario Inputs

Each collision scenario is defined by a set of parameters that establishes the initial conditions of the collision event; the collision model then determines the results of the collision. The parameters defining the initial conditions are called "scenario inputs". They are determined by randomly selecting each input using a pdf that describes the range and variability of that parameter. The scenario inputs are:

- speed (independently chosen for each ship)
- collision angle
- bow entrance angle (for the striking ship)
- Minorsky energy coefficient
- initial collision contact point
- striking ship mass

The pdf's describing the range and variation for these inputs are developed through a combination of data collected from actual collisions and expert opinion, and then modified by a calibration process (described more fully in Section 3.1) to produce reasonable results. Figure 1 shows how these inputs describe the developing collision scenario. The struck ship's course angle (w1 in Figure 1) is zero at the start of every collision.

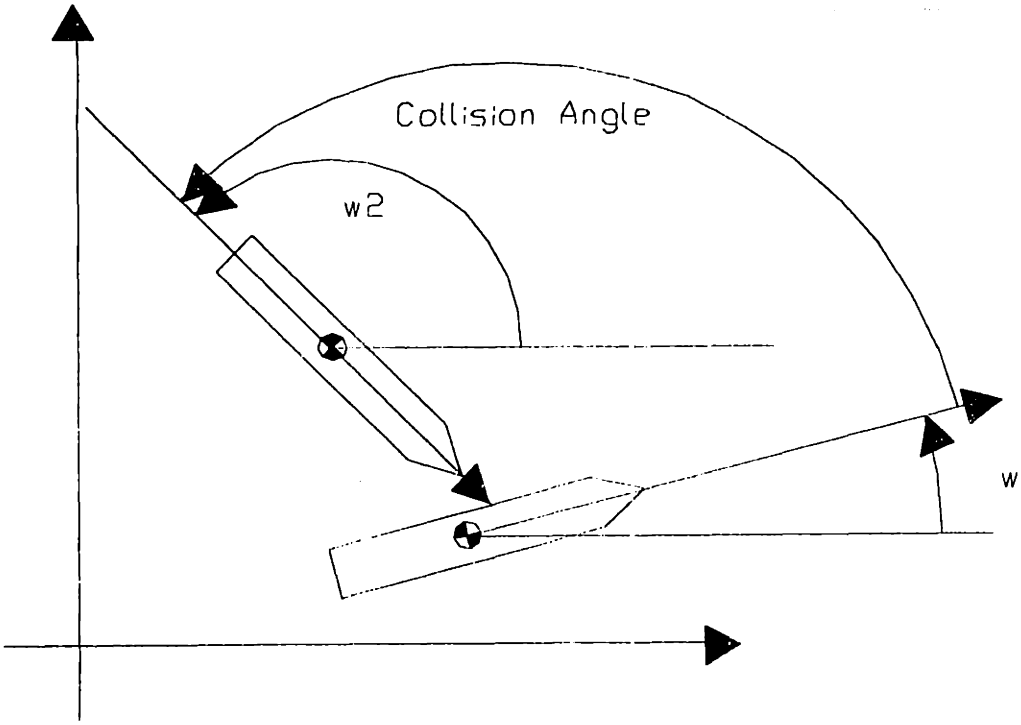


Figure 1: Collision scenario definitions

1.3.3. Collision Kinematics and Simulation

Minorsky's empirical analysis [2] is generalized to allow for freedom of motion in the x-y plane, and rotation about the z-axis for each ship. One unfortunate result of this generalization is that a closed-form analytical solution is no longer obtained. Minorsky assumed that the only energy important to the collision was that from striking ship motion perpendicular to the struck ship. He also assumed that the collision was totally inelastic, so that the ships remained joined after the collision. In the generalized model all of the kinetic energy is considered and the ships are both free to rotate. This allows energy to go into rotational motion as well as deformation of struck ship structure. These additional "unknowns" prevent solution of a set of simultaneous equations, and require a time-domain collision simulation to calculate the final state variables (linear and rotational velocity) and damage extents. The collision simulation is further discussed in Section III.

1.3.4. Calibration of Input pdf's

A "calibration" process is conducted to complete the definition of scenario inputs. The concept underlying the calibration process is that if the collision simulation model is accurate, use of scenario inputs drawn from pdf's that describe the "real world" range and variation of these inputs should produce result pdf's that match "real world" damage data. Although several shortcomings of the IMO regulation pdf's have been identified, they *were* developed from data collected from actual collisions, and the raw data that forms the basis of the pdf's is also available [12]. This makes them a good source of "real world" statistics for comparison. Two of the previously described limitations – considering only single-hull ship collisions and only considering cases where the outer shell is ruptured – are compensated for by using similar restrictions in the calibration process. The input pdf's are calibrated to minimize the difference between the damage extent pdf's produced by the model and the IMO guideline pdf's. This process is described in detail in Section VI.

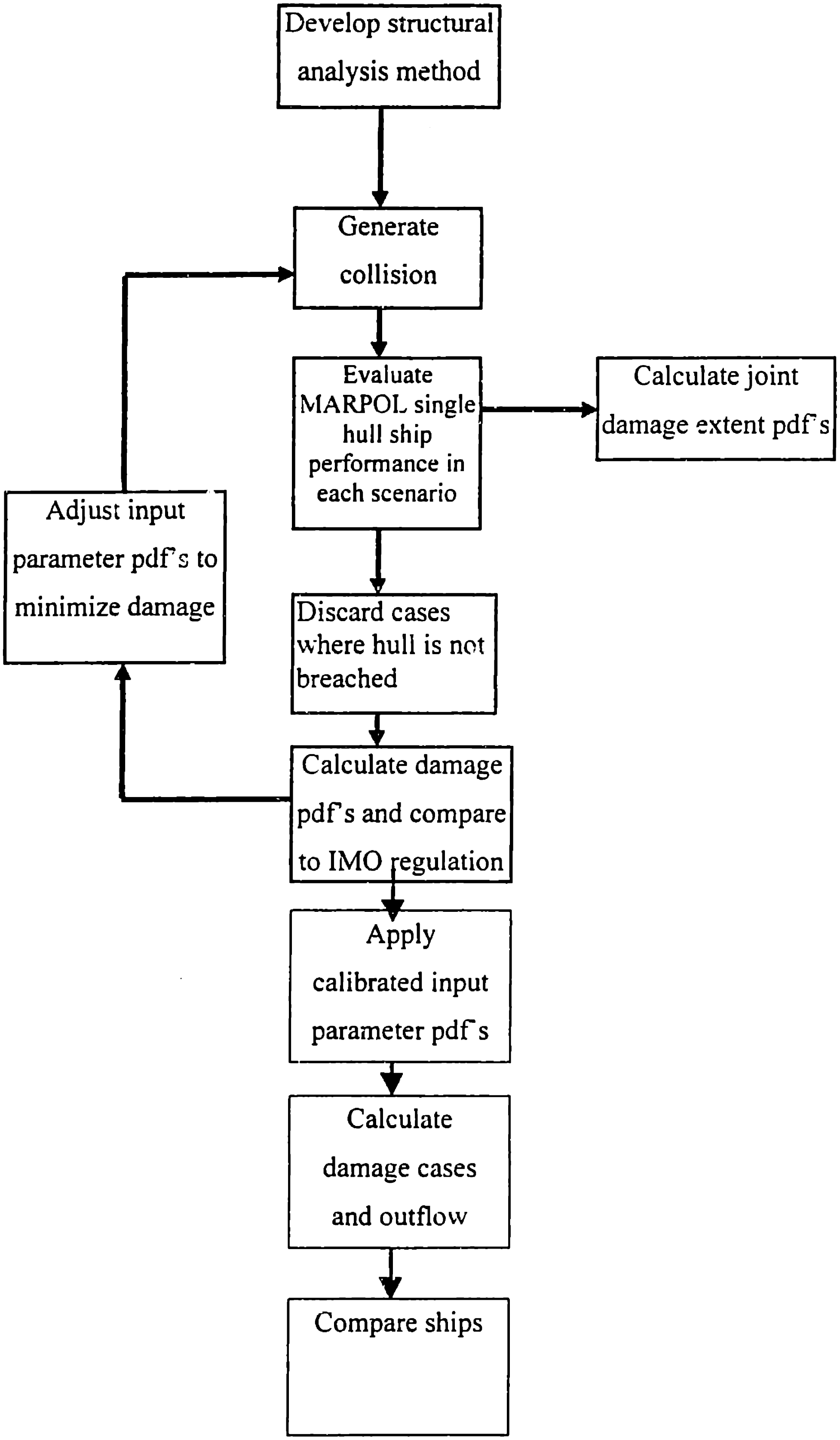
1.3.5. Calculation of Damage Extent and Oil Outflow pdf's

After the scenario input pdf's are calibrated against the current regulatory damage extent pdf's, the collision simulation model is run using several different ship designs. These ship designs were developed using an American Bureau of Shipping (ABS) – proprietary software product called "SafeHull". Each design was verified to meet ABS requirements for section modulus. The ships include a MARPOL 73/78 single-hull design (used in the calibration process), a tanker with an intermediate oil-tight deck (or "mid-deck" design), and several variations on a double-hull design. The individual ship designs are described in Section IV.

The collision simulation model produces pdf's describing oil outflow and extent of damage for each of the ship designs. The oil outflow for each case is calculated directly in the simulation rather than from the damage pdf's. The oil outflow pdf's are used to determine the probability of zero outflow and mean outflow for each ship type. The pdf's describing extent of damage are used only to assess the coupling between transverse and longitudinal extents of damage, and to compare the crashworthiness of the different designs. The following process description and Figure 2 show the overall approach taken for this investigation, and how the calibration process fits into the overall evaluation scheme.

Overview of the Selected Method

1. Choose a means of modeling the structural response.
2. Make initial estimates for input parameter pdf's.
3. Evaluate a number of collision events for a MARPOL single-hull ship.
4. Extract cases resulting in breach of the outer hull, and compare the resulting damage extent pdf's to those found in the IMO regulations.
5. Modify the input parameter pdf's to provide the best match to the given IMO pdf's and underlying data.
6. Repeat steps 3 - 6 until the match is satisfactory, or no further improvement is noted.
7. Fix the input pdf's.
8. Evaluate a statistically significant number of collision events for the ship design types of interest, i.e., single-hull, double-hull and mid-deck tankers.
9. Analyze the results.


Develop structural analysis method
Generate collision
Evaluate MARPOL single hull ship performance in each scenario
Calculate joint damage extent pdf's
Adjust input parameter pdf's to minimize damage
Discard cases where hull is not breached
Calculate damage pdf's and compare to IMO regulation
Apply calibrated input parameter pdf's
Calculate damage cases and outflow
Compare ships

2. Collision Model Development

This section describes the time-domain simulation, and the assumptions and approximations used to model the force mechanisms.

2.1. Requirements and Generalization of the Minorsky Method

Previous work in the area of this study recommended that Minorsky's analysis be extended to include the effects of hull envelope on energy absorbtion [1], and outlined a method to generalize Minorsky's analysis to allow for ship motion in more than one direction [4]. Both of these concepts are incorporated in this collision model. The considerations that guided the development of the model are allowances for:

- forward (surge) motion of each ship
- lateral (sway) motion of each ship
- rotation of each ship about it's own vertical axis (yaw)
- energy absorption of hull membrane prior to fracture or rupture
- energy absorption of interior longitudinal bulkheads prior to fracture or rupture
- calculation of oil outflow resulting from each collision
- calculation of the longitudinal and transverse extent of the damaged region at each moment of time during the collision process

The first three considerations result in the need to calculate and include the effects of hydrodynamic added mass. The fourth and fifth considerations require a model of energy absorption for shell plating and bulkheads. The last two considerations require that the final collision process model be capable of tracking the extent of damage throughout the collision, thereby producing a final damage plan. Each of these requirements is addressed.

2.2. Assumptions

Implicit and explicit assumptions include:

1. The presence of a free surface is neglected. No wave interactions or effect of free surface on hydrodynamic behavior is modeled.

2. The striking ship's bow is modeled as a rigid triangular structure. None of the collision energy is absorbed by the striking ship's structure. This assumption is conservative, in that it causes the model to overestimate damage to the struck ship.

3. The collision is assumed to occur with little to no warning, so no collision avoidance actions are taken. This is implicitly included in the model by

    - setting initial yaw rate for both ships to zero, and

    - neglecting any effects of backing down on ship velocities

Assuming that no collision avoidance actions are taken prior to or during the collision is also conservative.

4. The analysis is not a survivability calculation. No calculation of hull girder residual strength or damaged stability is conducted. The assumption is that the struck ship remains intact and continues to float.

5. Two force mechanisms are considered. One, herein called the "Minorsky force", results from plastic deformation of structure internal to the struck ship. The other, referred to as the "membrane force", results from energy absorbed by plastic deformation of the struck ship's hull or internal longitudinal bulkheads prior to fracture or rupture. Each of these forces has an associated direction of action and reaction.

- The Minorsky force acts in a direction parallel to the direction of relative motion between the ships at the point of impact, i.e., in a direction to oppose relative motion.

- The membrane force acts in a direction perpendicular to the axis of the struck ship, including any rotation that occurs during the collision process.

2.3. Calculation of Added Mass

As the ship moves through the water, it interacts with the fluid medium. The effect of the surrounding water must be accounted for when calculating the result of external forces applied to the ship. This is done by including a hydrodynamic "added mass". Added mass is a tensor, the components of which depend on the geometry of the ship hull, and the direction of movement with respect to the ship axes. The form of the tensor is:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

where the subscripts indicate the principal body-fixed axes of the ship. The subscript "1" corresponds to the principal longitudinal axis and motions in surge. The subscript "2" corresponds to the transverse axis and motions in sway, and the subscript "3" to the vertical axis and motions in heave. In the absence of detailed information about the hull forms and appendages, it is assumed that the ships are symmetric across all three planes. This eliminates coupling between the cross-terms in the added mass tensor, which can then be written:

$$A = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}$$

Finally, since this analysis does not consider motion in the z-axis direction, none of the elements of the tensor relating to heave motion are retained. This leaves the final form:

$$A = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}$$

The added mass in surge is calculated by approximating the added mass as equal to that of a flat plate having the same area as the midship section of the ship. The added mass of a flat plate for motion normal to the plate is equal to the displaced mass of the circumscribed circle [5]. Therefore,

$$a_{11} = \frac{\frac{4}{3}\left[\frac{BT}{\pi}\right]^{\frac{3}{2}}}{\Delta}$$

where the midship section coefficient is assumed to be equal to one. For the ships examined in this study, this formulation results in an added mass in surge of approximately 9% of the ship's displacement.

A strip theory approach is used to calculate added mass in sway. The coefficient of added mass for a two-dimensional rectangular section with side length of 2a under lateral acceleration is [5]:

$$m_{22}(x) = 4.754\rho a^2$$

Neglecting any effects due to the free surface, and setting the calculated draft equal to 2a yields:

$$m_{22}(x) = 1.189\rho T^2$$

Using strip theory to obtain the added mass coefficient for the three-dimensional ship:

$$a_{22}\Delta = \int_0^L m_{22}(x)dx$$

or

$$a_{22} = \frac{1.189\rho T^2 L}{\Delta}$$

This results in an added mass in sway of about 42.5% of the ship displacement. This is likely greater than what would be measured for a real ship. This is because the derivation treats the ship as a long rectangular structure, neglects the reduction in area at the bow and stern, and also neglects the effect of the free surface. As a comparison, Minorsky's original analysis assumed a value of 40% [2]. Experiments have measured this quantity at 40% for short collisions [17], but noted that the value is strongly dependent on the collision duration. The formulation outlined above is retained because there is no better means of calculating added mass in sway without more detailed knowledge of the hull and appendage geometry. A more accurate approach requires substantially more information about hull geometry than exists for the particular ship designs under consideration. It would probably not appreciably change the results of the model. Using the larger value for the added mass coefficient is also conservative because the struck ship acquires less translational velocity during the collision, and therefore more energy goes into structural deformation.

2.4. Rotational Inertia

In addition to linear translation (in the x- and y- axis directions), rotational degrees of freedom must be considered for each ship. This requires a calculation of both physical mass moment of inertia and added mass moment of inertia about the z-axis for each ship. The physical mass moment of inertia is calculated by the familiar formula:

$$I_{66} = \int r^2 dm(r)$$

where the integration is performed over the entire mass of the body. The notation "dm(r)" indicates that the mass distribution is a function of the distance "r" from the origin. For this calculation, the mass of the ship is assumed to be evenly distributed along the length of the ship. This is also a conservative estimate, in that the actual mass distribution will have relatively less mass at the extreme ends of the ship, and therefore have a lower moment of inertia. The model will therefore have less energy going into rotational motion, and more into structural deformation.

The added mass moment of inertia is calculated in a similar manner, using the two-dimensional added mass in sway for each incremental section about the axis of rotation:

$$I_{66A} = \int r^2 dm_{22}(r)$$

The virtual mass moment of inertia for the ship is then the sum of the physical and added mass moments of inertia.

$$I_{66V} = I_{66} + I_{66A}$$

The virtual mass moment of inertia is used to calculate the rotational accelerations and velocities resulting from the forces developed during the collision.

2.5. Overall Energy Balance, and Elapsed Time during collision

In order to provide an additional check on the final results of the collision process model, an overall energy balance is developed. By making simplifying assumptions similar to those made by Minorsky, an upper bound is calculated for the energy absorbed in structural deformation. The specific assumptions for this energy balance are:

- each collision is totally inelastic, so that after the collision the ships translate as a single body, rotating about a common center of mass
- neither ship has any initial rotational velocity
- the final physical arrangement of the two ships is such that the striking ship is embedded to a depth of one-half the beam into the struck ship, with an angle between the two ship's longitudinal axes equal to the initial collision angle. This sets the final mass distribution so the virtual mass moment of inertia can be calculated for the two-ship system.

Making these assumptions, and using conservation of linear and rotational momentum yields an equation for the energy absorbed by the ship structure:

$$E_a = \frac{1}{2}\left(\{[M_{v1}]\mathbf{V}_1\} \bullet \mathbf{V}_1 + \{[M_{v2}]\mathbf{V}_2\} \bullet \mathbf{V}_2 - \{[M_{v1} + M_{v2}]\mathbf{V}_f\} \bullet \mathbf{V}_f - [I_{66vf}]\varpi_f \bullet \varpi_f\right.$$

where

$M_{v\#}$ = the virtual mass (physical plus added) of ship #

$V_{\#}$ = the initial velocity of ship #

V_f = the final velocity of the two ship system

I_{66f} = the virtual mass moment of inertial of the two ship system

ϖ_f = the final rotational velocity of the two ship system

This energy is compared to the energy absorbed by the structure in the collision model. The energy calculated in the energy balance represents an upper bound on the energy that could be absorbed by the structure, because the energy balance permits less energy to go into other degrees of freedom.

As previously mentioned, the collision process model used is a time domain simulation of the collision process. In order to execute such a simulation, an appropriate time step must be selected. Hutchison [4] analyzed the effects of time step size on solution accuracy for a similar time domain collision analysis. The solution accuracy was judged by how well energy was conserved through the collision process by comparing the sum of energy absorbed and final kinetic energy to the initial kinetic energy of the system. The results of that analysis indicated that the accuracy of the simulation converged rapidly as time step size was reduced to a value of T/400, where T is the total time that elapses during the collision process. Rather than conducting a similar analysis for this simulation process, a smaller value of T/1000 is used. T is calculated as demonstrated by Hutchison:

$$T = \frac{\pi}{2}\sqrt{\left(\frac{1}{300000t \cdot \tan(\alpha)}\right) \cdot \left(\frac{M_{v1} \cdot M_{v2}}{(M_{v1} + M_{v2})}\right)}$$

where

t = the aggregate structural thickness, in inches

α = the striking ship bow half-entrance angle

$M_{v\#}$ = the virtual mass (physical plus added) of ship #

Again, the energy balance equation is used only as a check on the results of the collision process model, and plays no role in the process model itself. The calculation of collision duration is only used to help select an appropriate time step for the simulation process. This shortens the computation time for a given collision scenario by matching the time step with the duration of that collision. An equally valid but less elegant approach is to choose a conservatively small time step and use that for all collision scenarios.

2.6. Energy Absorption

Development of this collision model is guided by previous work done by the Ship Structures Committee [1]. They categorized the relative importance of possible energy dissipation mechanisms as "primary (or significant), secondary (or not very important), and tertiary (or negligible)"[1]. The various mechanisms ranked by category were:

Primary:

1. Membrane tension in plating, deck and stiffeners
2. Plastic bending in plating, deck and stiffeners
3. Plastic energy in shearing deformation of web frames
4. Rigid body motion (translation and rotation)

Secondary:

5. Elastic bending
6. Elastic vibration

Tertiary:

7. Thermal

In this collision process model the secondary and tertiary mechanisms are assumed to be negligible. The primary mechanisms are combined into two categories. The first category describes the energy absorbed by membrane tension in hull and internal longitudinal bulkhead plating. This is referred to as "membrane energy" along with a

---

[1] The quote comes directly from Volume 1 of Reference [1]. The original phrasing is kept to convey the panel's intent to portray even secondary mechanisms as "not very important".

corresponding "membrane force". The second category covers all other primary structural interactions and deformations. This is accomplished by relating the amount of energy absorbed to a volume of structure damaged via an empirically determined coefficient. This is referred to as "Minorsky energy" and the "Minorsky force".

2.6.1. Membrane Energy

Minorsky's original analysis of ship collisions forms the basis of the collision process model developed here, but Minorsky concentrated on "high-energy" – collisions where the ships involved are large and have high initial relative velocities. This focus produced a good linear relationship between volume of damaged structure and absorbed energy for high-energy collisions, but did not produce good results for low energy collisions.

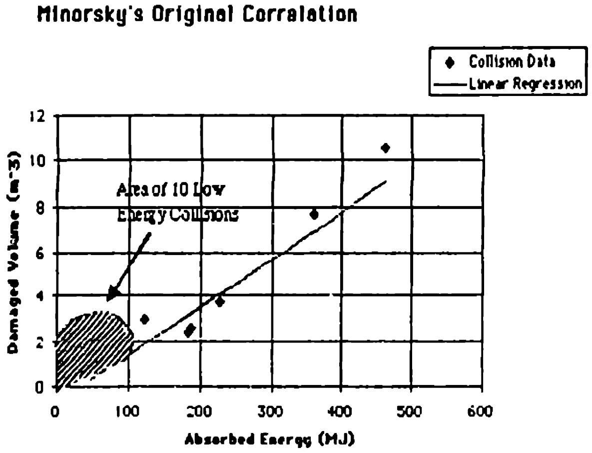


Figure 2: Original Minorsky Correlation[2]

---

[2] Adapted from reference [2].

Noting the area of poorly fit data points in the low energy regime of

Figure 2, the Ship Structures Committee recommended that Minorsky's method be extended into the low-energy regime by adding some consideration of the energy absorption capacity of the hull envelope [1]. Reardon and Sprung accomplish this by using a theoretical model of a wedge cutting a plate [3, 13, and 14]. The revalidated relationship shows almost exactly the same numerical value for the Minorsky coefficient, but improves the fit in the low energy regime. See Figure 3.

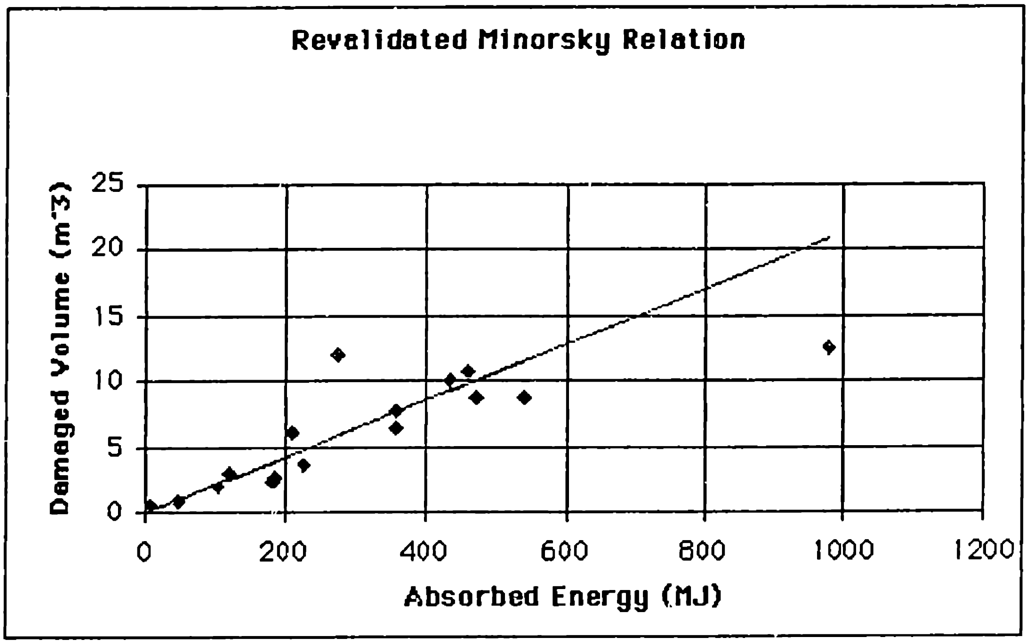


Figure 3: Revalidated and extended Minorsky correlation[3]

In Appendix C of Reference [1], a different method to do this is outlined by Jones, and further elaborated by Van Mater. This method treats the panels of the hull envelope as thin, broad "beams", with "pinned-pinned" end boundary conditions. The ends of the beam correspond to the panel attachments to internal ship framing.

---

[3] Adapted from reference [3].

Used together with Reardon and Sprung's results, this method allows:

- calculation of the energy used in deforming the beam
- arbitrary location of the application force between frames
- an approximation of the deflection prior to plate fracture or rupture

For this study it is also important to consider any inner hull membranes (as in the case of double-hull and mid-deck tankers) and internal longitudinal bulkheads. This is important not only from an energy absorption and crashworthiness point of view, but also when determining whether or not a given cargo bulkhead has ruptured and allowed oil outflow. Because of these issues, the extension proposed by Jones and Van Mater is applied to all of the vertical longitudinal plate structures.

The specific formulas used in these calculations are from [1]:

$$E_{mem} = \frac{\sigma_y t B w^2}{L} \cdot \frac{a}{b}$$

$$w_{\text{limit}} = 0.452a$$

where

E_{mem} = energy absorbed by the deformed membrane
σ_y = yield stress of steel
t = thickness of plate
B = effective breadth of plate
w = deflection of plate
L = length of plate between clamped ends
a & b = shorter and longer distance from point of force application to clamped ends

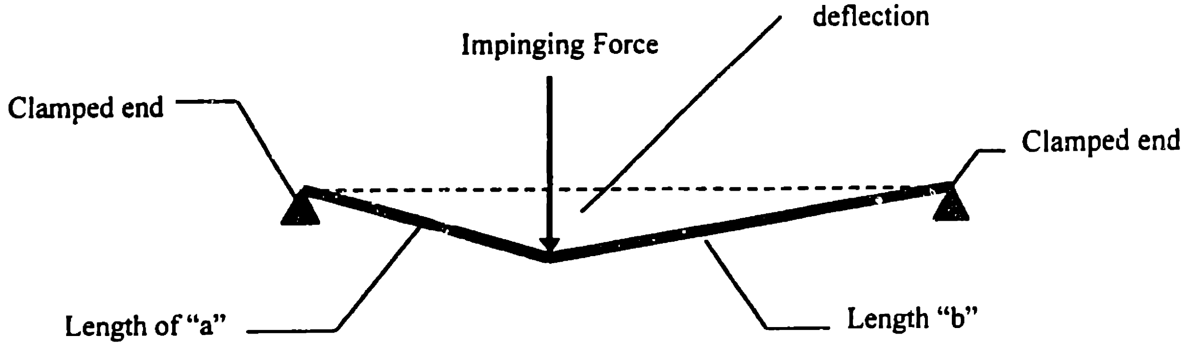


Figure 4: Membrane force mechanism concept

Figure 4 shows how the deflection of the plate is calculated from geometrical considerations.

The plate breadth (B) is not the vertical extent of the plate, but an effective breadth. This effective breadth is calculated by requiring the expectation value of energy absorption to match the shell energy absorption value of 28.4 MJ calculated by Reardon and Sprung in [3]:

$$B = \frac{1}{a} \cdot \frac{28.4MJ \cdot L}{\sigma_y \cdot t \cdot \int_0^{\frac{L}{2}} \frac{(L-a)}{(0.452a)^2 \cdot a} da}$$

The expectation value is calculated over the possible range of the short leg length, a. This provides an effective breadth that will result in a expectation value for E_{mem} of 28.4 MJ, as calculated in [3].

In each time step, the plate deflection is calculated from the geometry established by the bow shape and ship positions. Then there are two possibilities:

If the deflection is less than the deflection limit (w_{limit}) given above, the energy absorbed by the plate is calculated. Dividing this work done in the time step by the incremental deflection in the time step gives the magnitude of the force due to the plate deflection.

The direction of the force is assumed to be perpendicular to the longitudinal axis of the struck ship, and forms an action/reaction force pair, which acts on both ships.

If the deflection limit is exceeded, the energy absorbed by the membrane is set to zero, and the calculated force is therefore also zero. This is also used to define the rupture status of this membrane. In the simulation code, this triggers a transition to a different subroutine that no longer calculates plate deflection or energy absorption until and if a new plate or membrane structure is encountered.

2.6.2. Minorsky Energy

The other structural forces are combined into a single mechanism based on Minorsky's relationship between energy absorption and volume of damaged structure.

Figure 5: Minorsky mechanism concept

This is implemented by calculating the volume of newly damaged structure at each time step. Since one of the initial assumptions is that the striking ship's bow is a rigid triangle, this becomes a relatively simple problem in trigonometry. See Figure 5.

To calculate the newly damaged structure, the area covered by the triangular bow's incursion into the struck ship's side is determined, and then the area covered during the previous time step is subtracted. Finally, the newly damaged area due to longitudinal relative motion is calculated and added to give the newly damaged area during each time step. This area is multiplied by the Minorsky coefficient to give the energy absorbed during the time step. This is converted to a force by dividing the energy by the distance of relative travel during the time step. The force is modeled as an action/reaction force pair directed to oppose the relative motion.

3. Collision Scenarios

3.1. Selection and Calibration of Input pdf's

The collision simulation model is a deterministic process model, i.e., if the simulation is run repeatedly with the same initial input parameters, the same output will be produced. In order to explore the probabilistic nature of the damage extents resulting from collision, a stochastic process is employed to choose the parameters that define the initial state variables of the system. These parameters that define the initial conditions of an individual collision scenario are called "input parameters". They are drawn randomly from a set of probability density functions, each of which describes the relative likelihood of occurrence of any particular range of values for that parameter.

3.2. Ship Speeds

The ship speeds were selected from a pdf defined by a bimodal distribution made up of two normal distributions with mean values of 5 and 10 knots, each with variance of 1 knot2.

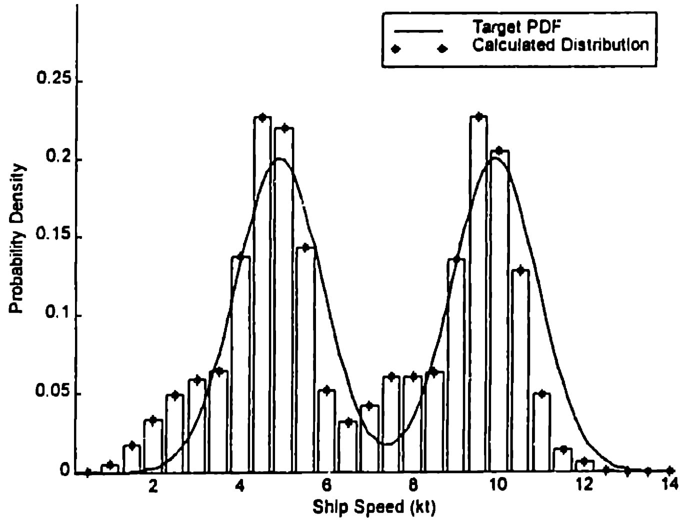


Figure 6: Ship speed histogram and pdf

This distribution was selected to match that used by Rawson [8], and is based on rational argument and expert opinion. The assumption is that tankers spend the majority of their time operating in one of two modes:

- Transit mode – represented by the 10 knot mode of the distribution
- Maneuvering mode – represented by the 5 knot mode of the distribution

The variance of each mode is an estimate based on the opinion of the author and several professional mariners. Figure 6 is a histogram of collision speeds selected for one of the ships (based on 5000 collision cases) compared to the speed probability density function.

3.3. Collision Angle

Collision angle is defined as the angle of incidence between the ships at the moment of impact. This input parameter is based on a uniform distribution. The choice to use this distribution is based on rational argument. Different distributions could be postulated (c.f. [9]) but would depend on a particular route and waterway. As an example, a route that includes a long straight channel would result in a collision angle pdf with higher density near zero degrees and 180 degrees.

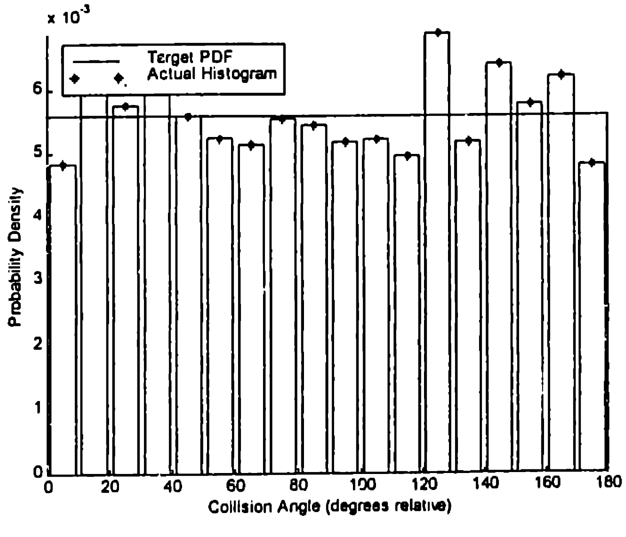


Figure 7: Collision angle probability density function

A route that includes an extended port approach with many turns and blind spots would result in a collision angle pdf with higher density near 90 degrees. The proposed uniform pdf represents a compromise for a generic tanker in worldwide trade. Collisions occurring at a relative angle of zero degrees are constrained to have an initial impact point at the bow of the struck ship. Collisions occurring at a relative angle of 180 degrees are constrained to have an impact point at the stern, and are only allowed if the striking ship's speed exceeds the struck vessel's speed. Figure 7 shows a histogram of selected collision angles (based on 5000 collisions) compared to the specified pdf from which they are drawn.

3.4. Impact Point

The point along the struck ship where the striking ship's bow initially makes contact is called the impact point. This point is allowed to vary with equal probability along the entire length of the ship. The selection of this pdf is based on rational argument. The pdf for impact point is simply a linear function with density equal to one along the length of the ship. Figure 8 shows a histogram of impact points based on 5000 collision cases compared to the pdf from which the impact points were drawn.

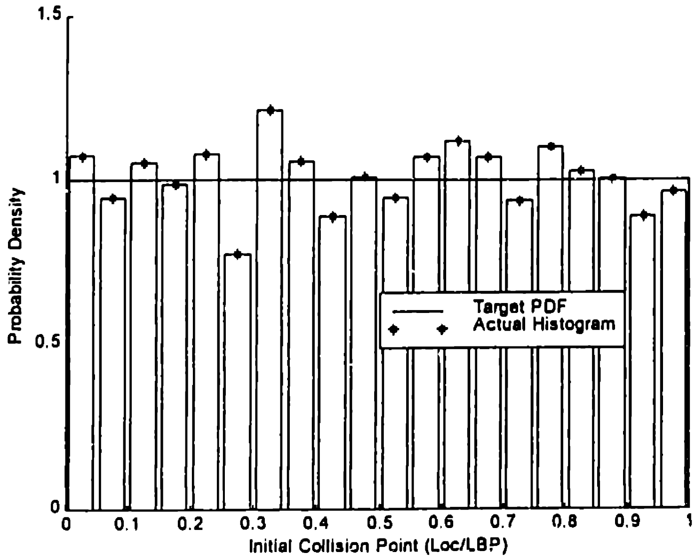


Figure 8: Initial impact point probability density function

The impact point includes parts of the ship forward of the forward-most cargo bulkhead, and aft of the aft-most cargo bulkhead. The collision dynamics account for the effect of forces and moments applied at these locations. The extent of damage is recorded and compared to the actual cargo tank boundaries in order to calculate oil outflow. This maximizes the "realism" of the collision simulation, rather than constraining the collision to begin within the cargo block.

3.5. Minorsky Coefficient

For each collision scenario generated, a particular value is selected for use in the Minorsky relationship between energy absorption and volume of structure damaged. The pdf used to select this value is a normal distribution with mean equal to 47.1 MJ/m3 and standard deviation of 8.8 MJ/m3. This distribution is based on a validation of Minorsky's original work done by Reardon and Sprung [3], including the addition of new data points from collisions that have occurred since Minorsky's work in 1959. Figure 9 shows a 5000 case histogram of this parameter, along with the pdf from which the selections were drawn.

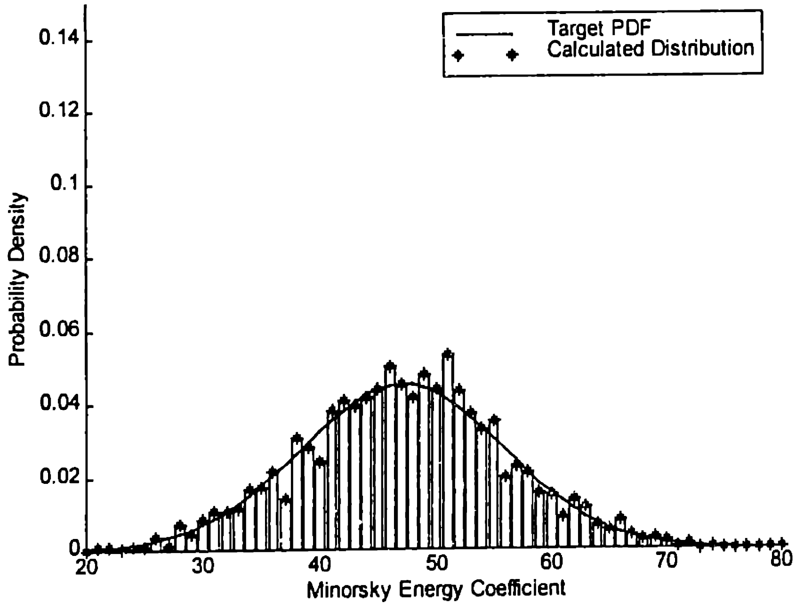


Figure 9: Minorsky constant pdf

3.6. Bow Entrance Angle

The shape of the bow of the striking ship is important because it determines the volume of structure subject to damage during the collision.

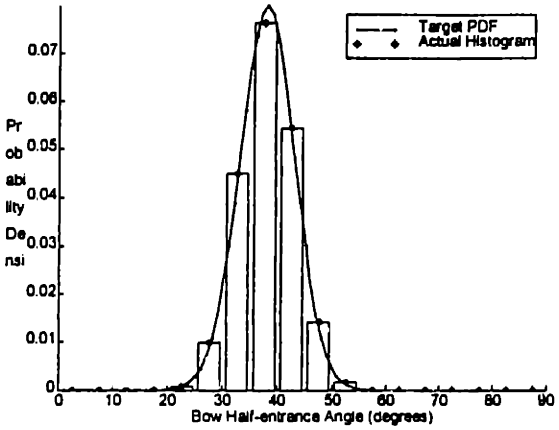


Figure 10: Bow half-entrance angle probability density function

In this analysis, the shape of the striking ship's bow is idealized as a triangle, with no rake. For each collision scenario, the bow half-entrance angle is selected using a pdf with a normal distribution, with a mean value of 38 degrees and standard deviation of 5 degrees. This distribution is based on data presented in [3] and [9] showing representative bow half-entrance angles for a range of ship displacements, and adjustments made during pdf calibration.

Figure 10 shows a histogram of selected bow half-entrance angles (based on 5000 collisions) compared to the specified pdf from which they are drawn.

3.7. Striking Ship Displacement

For each collision scenario, the particulars describing the struck ship are given and constant. The striking ship is not known, and its characteristics must be chosen using a distribution as for other scenario input parameters. A common approach to this problem is to assume that the striking ship and the struck ship are identical in all respects. This is based on the assumption that "like ships" travel the same waterways (being engaged in the same trade), and are therefore more likely to have collisions with other similar ships. This approach is not satisfying here, because the amount of energy the striking ship imparts to the collision process (and therefore the extent of damage) is strongly dependent on the mass of the striking vessel. The mass of the striking vessel should be chosen from a distribution that, like collision angle, reflects the waterway environment in which the ship is or will be operating. For this study, which is not specific to any particular waterway, the striking ship's displacement was selected from a normal distribution with a mean of 150,000 metric tons, and a standard deviation of 30,000 metric tons. This choice for this distribution was based on data from [9], and validated by the calibration process. The distribution is shown in Figure 11.

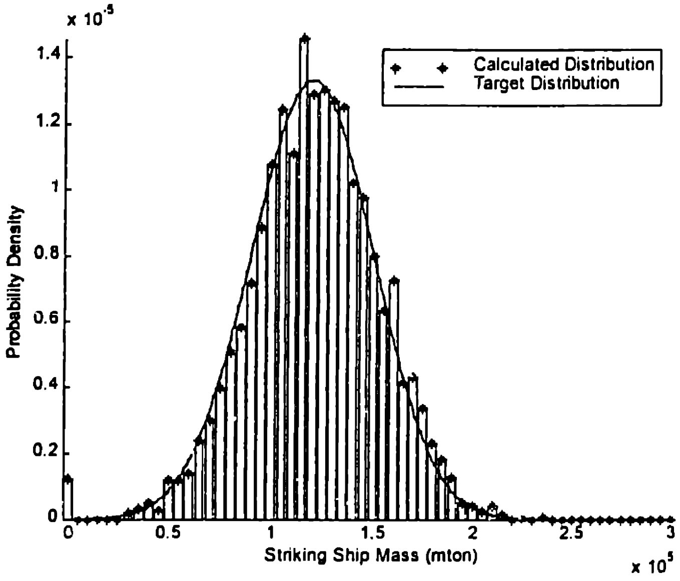


Figure 11: Striking ship mass histogram and pdf

3.8. Calibration of Input Parameter pdf's

To ensure that the results of the simulation are reasonable, the damage extent pdf's resulting from the collision simulation model and scenario inputs were compared to the equivalent pdf's in Regulation 13F. The pdf's in the Regulation are based on data collected from real collisions involving single hull vessels during the period 1980 – 1990.

They only include collisions in which the hull envelope was breached. In order to base this comparison on similar data sets, the collision simulation model was run using a MARPOL single hull ship as the struck vessel, and the output data was discarded for collisions where the outer hull is not ruptured. This initial comparison was favorable, and provided a basis for confidence in the validity of the model. Input pdf's required only minor adjustment. The end result of the calibration process was a set of input parameter pdf's that are verified to give reasonable results in this model, and may be used by others conducting similar analyses.

Following the initial comparison, a calibration process was undertaken to improve the correlation between the pdf's in the Regulations and those produced by the model. The correlation is measured by constructing a "goodness of fit" parameter similar to the "R-squared" parameter used to quantify correlation in a linear regression method (c.f., [11], especially Chapter 9). After the collision model was run and inappropriate cases discarded, the damage extent pdf's were calculated. The simulation pdf values (over discrete ranges) were compared to the IMO pdf over that same range. The difference is squared, and the sum taken over all the discrete ranges in the pdf, then divided by the number of ranges.

In equation form:

$$R^2 = \frac{\sum (S_{pdf} - R_{pdf})^2}{n}$$

where

Spdf = the value of the simulation pdf over the ith range

Rpdf = the value of the IMO pdf calculated over the ith range

n = the number of ranges taken

and the sum is taken over all n ranges. This value was calculated for each of the pdf's in the Regulation: longitudinal extent of damage, extent of transverse penetration, and longitudinal location of the center of the damaged area. An overall goodness-of-fit (hereafter "fit") was calculated by summing all squared differences and dividing by the total number of sample ranges.

Some of the input parameter pdf's were not considered for modification. The characteristics of the ship speed pdf's are considered fixed, since they are based on work done concurrently by Rawson [8] and provide good results in a grounding analysis. The pdf describing initial point of contact for the collision is also fixed, since only a uniform random distribution reproduces the regulatory pdf describing the location of the center of the damaged area. The parameters left to vary were those describing the striking ship's bow half-entrance angle, and the striking ship displacement. The medians of both these distributions were varied over reasonable ranges in an attempt to maximize the fit parameter. The results are shown in Figure 12.

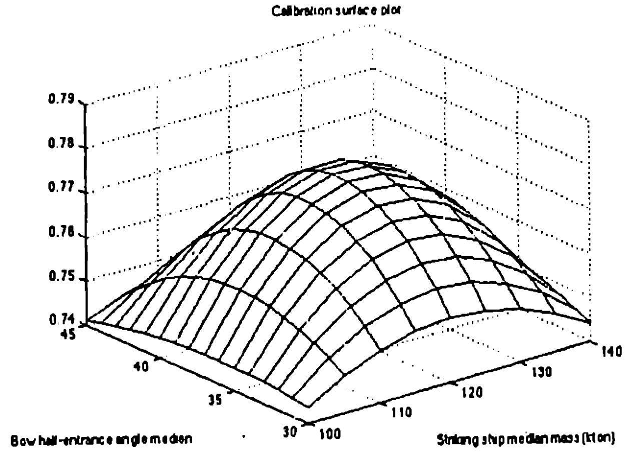


Figure 12: Calibration sensitivity plot

This calibration surface was generated by conducting nine simulation runs with the parameters varied over the range of interest. The data was then fit using a cubic interpolation scheme. It is apparent from the data collected that the local maximum for goodness of fit lies somewhere in the middle of the range explored. The median values for these distributions were selected by this process, and are as described previously. The scale used in Figure 12 makes the difference in fit over the explored range seem large.

In fact, the best fit parameter obtained was 0.7754, and the worst fit was 0.7411, a difference of only about 5%. This means that the results of the collision simulation model are relatively insensitive to variations in these two parameters within a reasonable range.

3.9. Single Hull Results Compared to IMO pdf's

3.9.1. Center of Damaged Extent

The pdf describing the location of the center of the damaged section corresponds fairly well with the pdf specified in the Regulation. There is some random variation around the uniform density level of one. Superimposed on this random variation is a bias toward the stern of the ship. This bias is caused by the motion of the stricken ship, which is assumed to be in the forward direction.

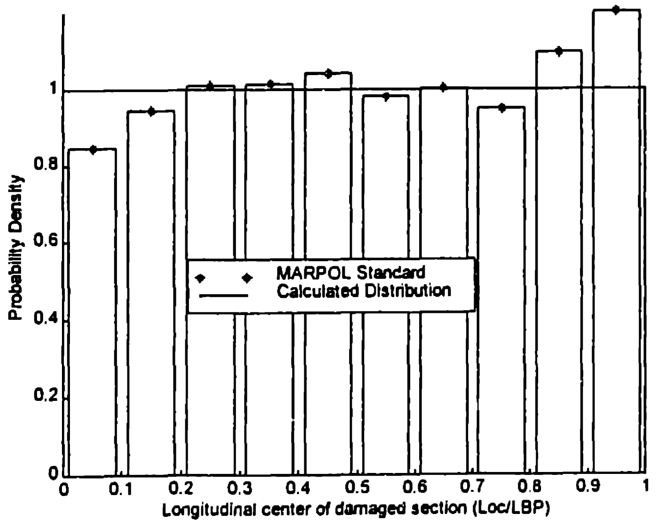


Figure 13: Location of center of damaged section for single hull tanker

3.9.2. Longitudinal Extent of Damage

The longitudinal extent of damage predicted by the collision simulation model matches the Regulation very closely. One difference is that the model predicts some damage that exceeds 0.3L, which is the upper limit of the Regulation pdf. The frequency of these cases is low. The transverse extent of damage associated with these cases is also low (this can be seen from Figure 23.) These damage patterns result from high energy collisions with low angles of incidence.

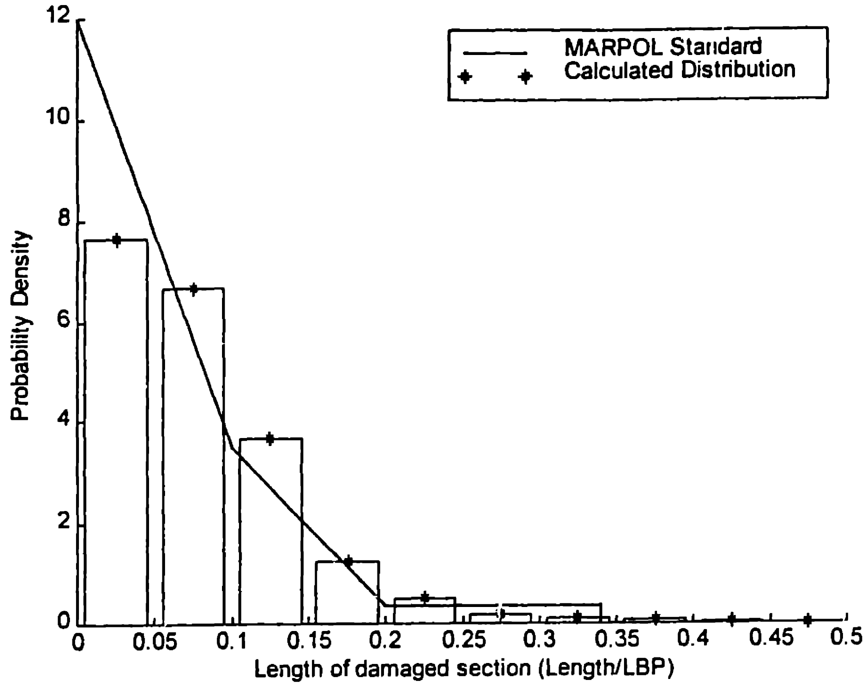


Figure 14: Longitudinal Extent of Damage for Single hull Tanker

Another difference is that extremely short longitudinal extents do not appear as frequently as the IMO Regulation would predict.

3.9.3. Transverse Extent of Damage

The pdf describing transverse extent of damage differs substantially from the regulation pdf. The most notable characteristic is the collection of damage cases with transverse extent around 0.2B. The collision simulation model produces this result because of the longitudinal bulkhead located at 0.1875B. The additional resistance presented by this bulkhead stops many collisions, and reduces the relative velocity so that many more are stopped in the next few meters. The model also predicts a small probability of transverse damage exceeding 0.3B, whereas the Regulation never predicts this.

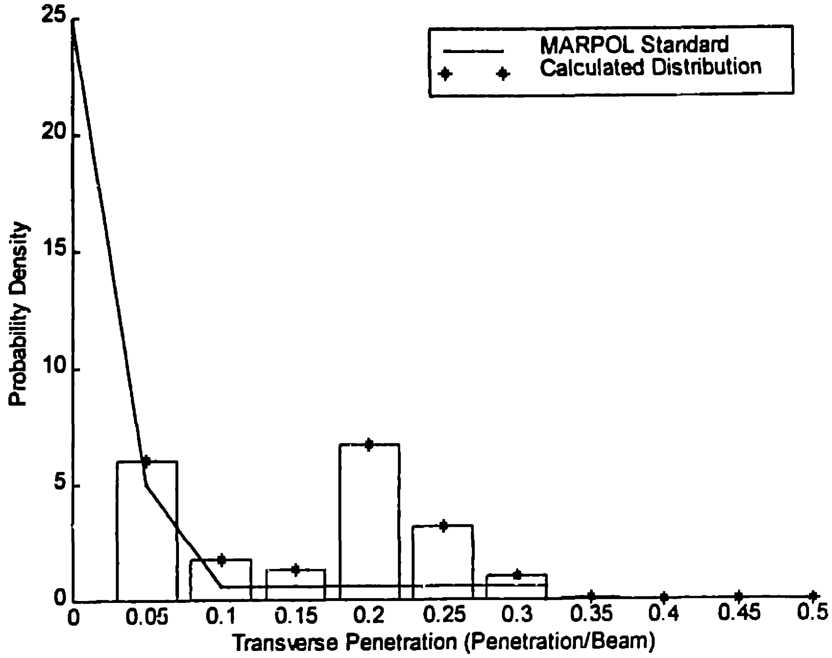


Figure 15: Transverse Extent of Damage for Single hull Tanker

This particular pdf causes the most difficulty in the calibration process. No variation of input parameters will eliminate this "spike" from the model output. This is the limiting factor in improving correspondence with the IMO pdf's.

4. Ship Designs

4.1. General Specifications

A family of representative tankers was designed by Rawson [10] for calibrating the input scenario pdf's and estimating the effect of structural enhancements on crashworthiness. The tankers include a MARPOL single hull tanker, five double hull tanker variants and an intermediate oil-tight deck (mid-deck) tanker, all of Suezmax (150,000 dwt) dimensions. The single hull tanker is designed consistent in material and configuration with vessels in service between 1980 and 1990, the period included in the data compiled by the classification societies to generate the current IMO damage pdf's. This design is used to calibrate the scenario probability density functions by matching the calculated damage extent density functions to the density functions provided in the Guidelines. The double hull and mid-deck configurations are designed using current shipbuilding practices and used for comparisons between design alternatives.

All designs have the same principal dimensions listed in Table 2, with bulkheads located to maintain equal cargo capacities and compliance with MARPOL Regulations for protective location of segregated ballast tanks, maximum tank volumes and double hull requirements. Scantlings are the minimum allowed by current classification societies standards, as determined by the American Bureau of Shipping's SafeHull system. The effect of structural enhancements on crashworthiness is studied using five separate double hull variants. Each variant is a derivative of the original baseline double hull model, with either the plating thickness, stiffener sizes, stiffener spacing or frame spacing modified. For each new variant design, the remaining structural parameters are re-examined using SafeHull to ensure optimum (i.e., minimum) compliance with classification scantling requirements.

Table 2 and Figure 16 provide the general specifications to which each ship configuration is designed. Details are provided in Figure 17 through Figure 19 and Table 3 through Table 10. In each figure, a plan view and transverse section are shown. Scantlings are listed in the tables. The plate thickness listed is the average of the plate thickness over the breadth of the respective bottom, side, deck or bulkhead. Frame spacing is 4.7 m for all ships except the final double-hull variant, where frame spacing is 3.7 m.

	Single-Hull	Double-Hull	IOTD
Length Between Perpendiculars	264 m	264 m	264 m
Beam	48 m	48 m	48 m
Draft	16.8 m	16.8 m	16.8 m
Depth	24 m	24 m	24 m
Double-Bottom Depth	N/A	2.4 m	N/A
Wing Ballast Tank Width	0 m	2 m	5.5 m
Displacement	178,411 mton	178,411 mton	178,411 mton
Deadweight Tonnage	~ 150k	~ 150k	~ 150k
Plating material	MS24	HT36	HT36
Cargo Tank Arrangement	5 x 3 plus 2 slop tanks	6 x 3 plus 2 slop tanks	6 x 2 over 6 x 1 plus 2 slop tanks

Table 2: General Specifications of Ship Designs

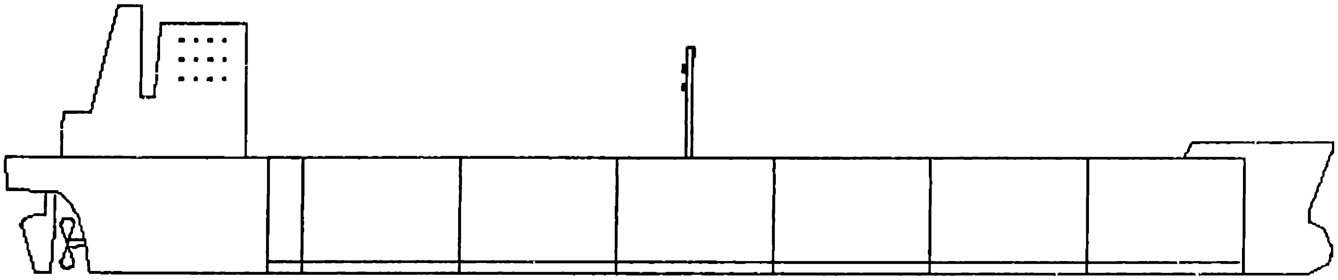


Figure 16: Typical tankship profile

4.2. Single Hull

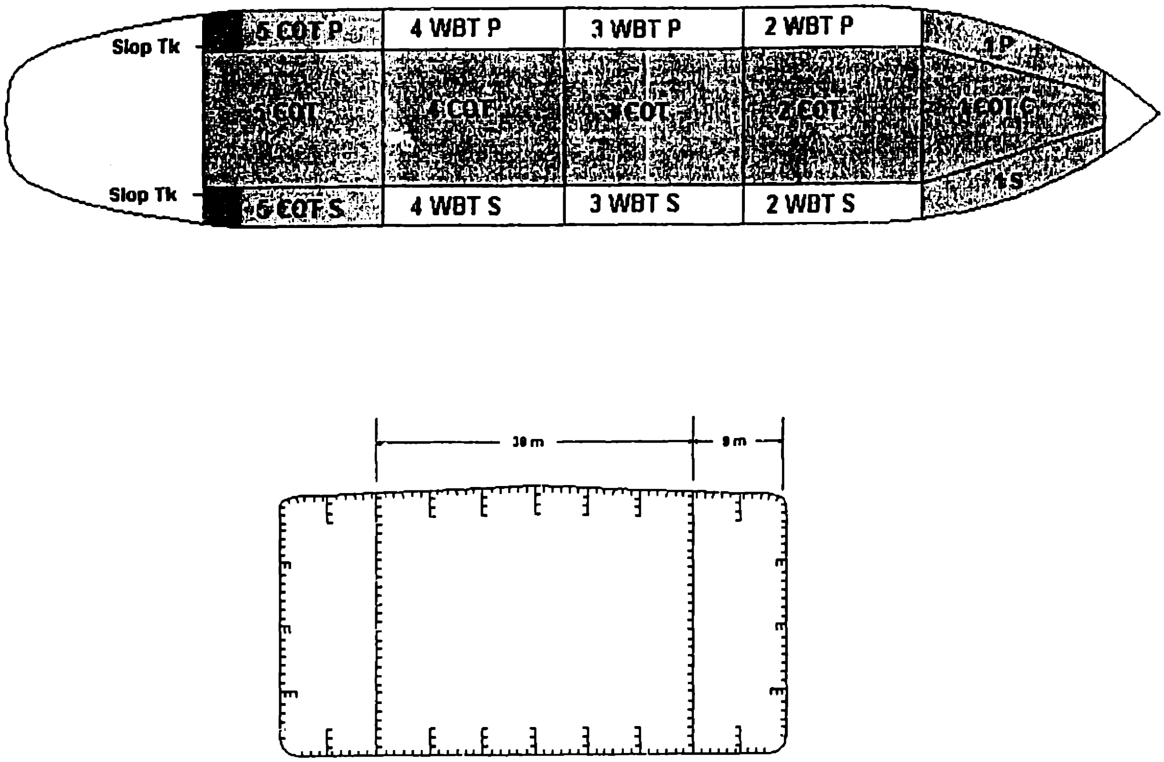


Figure 17: Single Hull Plan and Section

	Slop Tank	Cargo #5	Cargo #4 or Ballast	Cargo #3 or Ballast	Cargo #2 or Ballast	Cargo #1
Port	2195	6520	8878	8878	8712	5975
Center	N/A	29052	29592	29592	29041	19918
Starboard	2195	6520	8878	8878	8712	5975

Table 3: Cargo Tank volumes for Single-hull ship design (m3)

The scantling dimensions used in the Minorsky and membrane force calculations for the single-hull ship are:

Component	Thickness
Side shell plate thickness	2.576 cm
Bottom plate thickness	2.181 cm
Upper Deck plate thickness	2.2 cm
Aggregate deck thickness	4.381 cm
Internal longitudinal bulkhead plate thickness	1.952 cm

Table 4: Single Hull Scantlings for collision analysis

4.3. Double Hull Ships

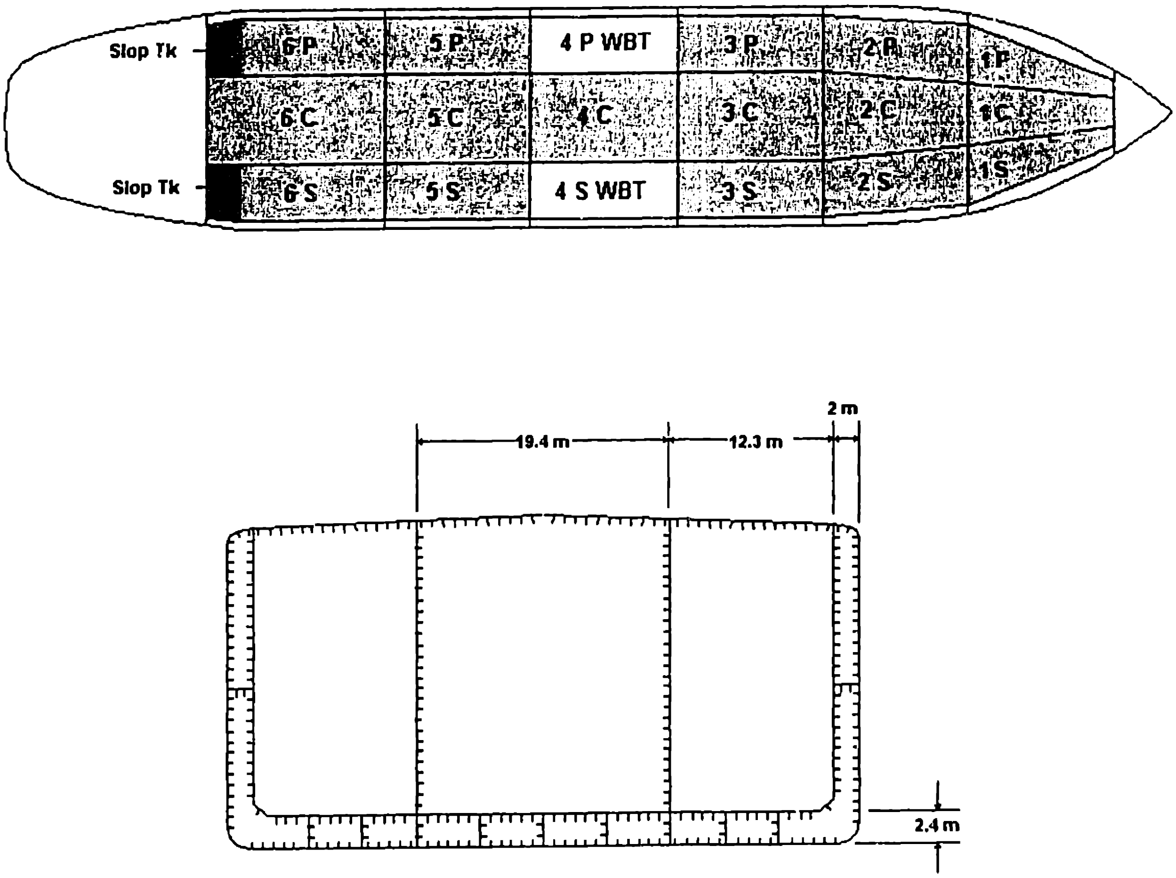


Figure 18: Double hull Plan and Section

	Slop Tank	Cargo #6	Cargo #5	Cargo #4 or Ballast	Cargo #3	Cargo #2 or Ballast	Cargo #1
Port	1953	8767	8767	8767	8767	8348	5399
Center	N/A	16908	13828	13828	13828	13167	8516
Starboard	1953	8767	8767	8767	8767	8348	5399

Table 5: Cargo Tank volumes for Double-hull ship designs (m3)

4.3.1. DH - Baseline

Rawson developed five separate versions of the double-hull configuration. The baseline DH design represents an "optimized" ship that uses the minimum weight of steel to meet ABS requirements. The scantling dimensions used in the Minorsky and membrane force calculations for the baseline double-hull (DH) ship are listed in Table 6:

Component	Thickness
Side shell plate thickness	1.8 cm
Bottom plate thickness	1.881 cm
Inner Bottom plate thickness	1.771 cm
Upper Deck plate thickness	2.1 cm
Aggregate deck thickness	5.752 cm
Inner skin plate thickness	1.842 cm
Internal longitudinal bulkhead plate thickness	1.725 cm

Table 6: DH Scantlings for collision analysis

4.3.2. DH1

DH variant #1 (DH1) is derived from the baseline by increasing plate thickness to 150% of its original value. The

Component	Thickness
Side shell plate thickness	2.7 cm
Bottom plate thickness	2.822 cm
Inner Bottom plate thickness	2.656 cm
Upper Deck plate thickness	3.0 cm
Aggregate deck thickness	8.478 cm
Inner skin plate thickness	2.763 cm
Internal longitudinal bulkhead plate thickness	2.587 cm

Table 7: DH1 scantlings for collision analysis

4.3.3. DH2

DH variant #2 (DH2) is derived from the baseline double-hull by increasing the scantlings of all stiffeners so that their contribution to total section modulus is 150% of the original design. Because the collision model used here does not consider the impact of individual stiffeners, and the plate thickness is not changed from the original design, there is no difference in collision performance between the baseline ship and DH2 in this model. This variant is not discussed further.

4.3.4. DH3

DH variant #3 (DH3) is derived from the original double-hull design by reducing the stiffener spacing to 75% of that used in the original design.

Component	Thickness
Side shell plate thickness	1.614 cm
Bottom plate thickness	1.405 cm
Inner Bottom plate thickness	1.4 cm
Upper Deck plate thickness	1.855 cm
Aggregate deck thickness	4.66 cm
Inner skin plate thickness	1.694 cm
Internal longitudinal bulkhead plate thickness	1.256 cm

Table 8: DH3 scantlings for collision analysis

Plate thickness is reduced to the maximum extent possible while still meeting ABS requirements for section modulus. The dimensions used in the collision analysis for DH3 are shown in Table 8.

4.3.5. DH4

DH variant #4 uses the same scantlings as the baseline ship, but the frame spacing is reduced from 4.7 m to 3.7 m. In the collision simulation model frame spacing is used to determine the dimensions of plating for calculating membrane energy and force.

4.4. Intermediate Oil-Tight Deck Ship

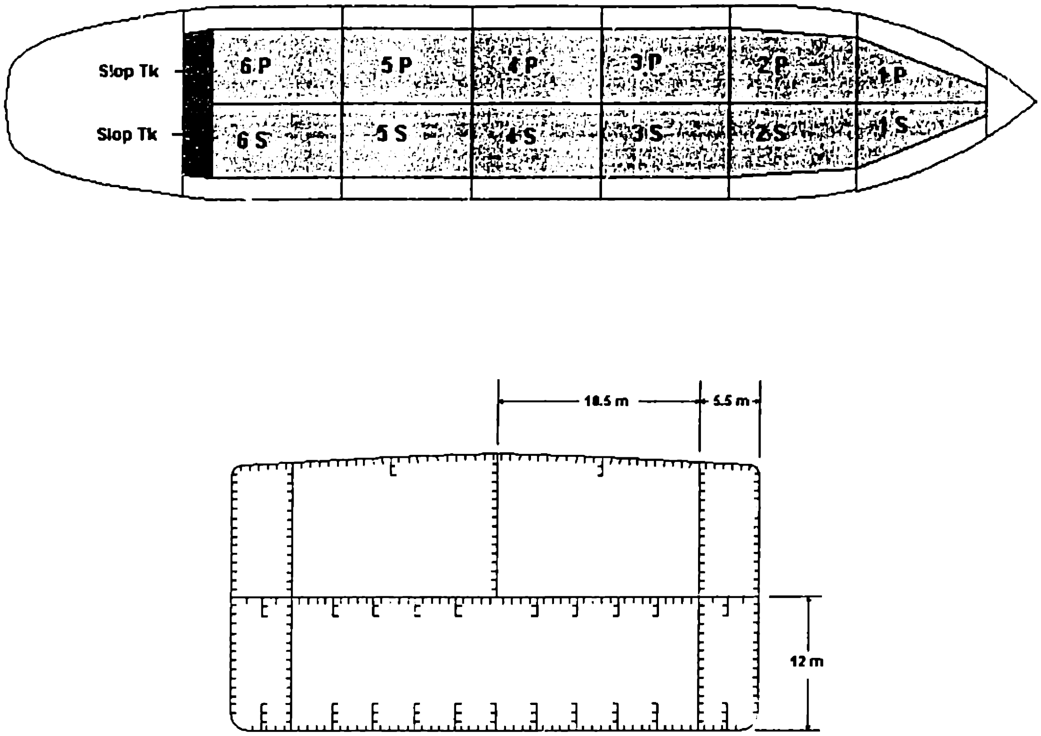


Figure 19: Mid Deck Plan and Section

	Slop Tank	Cargo #6	Cargo #5	Cargo #4/ Ballast	Cargo #3/ Ballast	Cargo #2/ Ballast	Cargo #1
Port	1626	7326	7326	7326	7326	6909	3979
Starboard	1626	7326	7326	7326	7326	6909	3979
Lower		17903	14652	14652	14652	13819	7959

Table 9: Cargo Tank volumes for IOTD ship design (m3)

The scantling dimensions used in the Minorsky and membrane force calculations for the IOTD ship are:

Component	Thickness
Side shell plate thickness	1.8 cm
Bottom plate thickness	1.8 cm
Inner Bottom plate thickness	1.56 cm
Upper Deck plate thickness	2.32 cm
Aggregate deck thickness	5.68 cm
Inner Skin plate thickness	1.843 cm
Centerline Bulkhead plate thickness	1.622 cm

Table 10: Mid Deck scantlings for collision analysis

The mid-deck tanker represents an alternative design under the Regulation. This ship has shell scantlings similar to the MARPOL single hull tanker, but the decks and bottom are reduced because of the presence of the internal horizontal mid-deck. This ship also has a centerline bulkhead which extends from the upper deck to the mid-deck, but not to the inner bottom. Another distinctive feature of this design is the rather wide ballast tankage outboard (double sides). This functions as protectively located ballast and provides a measure of protection in collisions.

5. Results

5.1. Mean Outflow and Probability of Zero Outflow

Mean oil outflow and the probability of zero outflow are calculated for each ship design. The results are shown in Table 11.

The probability of zero outflow calculated in this analysis is not the same as the probability of zero outflow as used in the Regulation. In the Regulation the calculated value is a conditional probability that is more properly described as "the probability of zero outflow given a collision that results in hull rupture." The value calculated in this study is also a conditional probability. It is "the probability of zero outflow given a collision".

The value of mean outflow is non-dimensionalized by dividing the outflow volume by the total cargo volume of the ship.

Ship Design	Probability of Zero Outflow	Mean Outflow
Single Hull	0.50	0.08
Mid-Deck	0.62	0.10
Double Hull (baseline)	0.47	0.06
Double Hull (enhanced plate)	0.49	0.05
Double Hull (reduced stiffener spacing)	0.45	0.06
Double Hull (reduced frame spacing)	0.45	0.05

Table 11: Probability of zero outflow and mean outflow results

Referring to Table 11:

The mid-deck design has the highest probability of zero outflow, but also the highest mean outflow. The high mean outflow is related to the subdivision scheme chosen by the designer. Once the cargo boundary is breached, 75% of the oil in that cargo section is lost. This ship would have substantially improved performance if an intermediate oil-tight deck were combined with a more typical "three tank across" arrangement or a center line bulkhead in the lower tank. These arrangements would also reduce potential intact stability problems associated with free surface effects during loading and unloading. To explore this, the simulation was run again with a lower centerline bulkhead added to the design. P_0 remained constant at 62%,but mean outflow dropped from 10% to 8%.

The double hull designs all show roughly similar performance. The design with enhanced plate thickness shows the best performance within the double-hull group. This is expected from the way the Minorsky method relies on in-plane elements for energy absorption. The double hull designs have a lower P_0 in collision, due to the relatively small protective layer of the double-side, but have the lowest mean outflow because of their greater subdivision

The single hull ship shows P_0 second only to the mid-deck, and mean outflow between mid-deck and double hull. The mean outflow of the single hull is also adversely impacted by the chosen subdivision.

5.2. Extent of Damage pdf's

The extent of damage pdf's presented are not conditional on rupture of the ship's hull. Because of the difference in conditionality, these pdf's should not be compared to the IMO pdf's, only between ship designs analyzed here.

5.2.1. Longitudinal Extent of Damage

The pdf describing longitudinal center of damage is roughly uniform, as expected. There is a slight bias toward the stern of the ship. This is a result of the forward motion of the struck ship. The initial point of impact was selected from a uniform pdf. This sets one end of the damaged extent. The location of the other end depends on the relative speeds of the two ships. The only way the "end" of the damaged section can be forward of the initial collision point is for the striking ship to have more velocity in the x-direction that the struck ship. This tends to shift the center of damage aft.

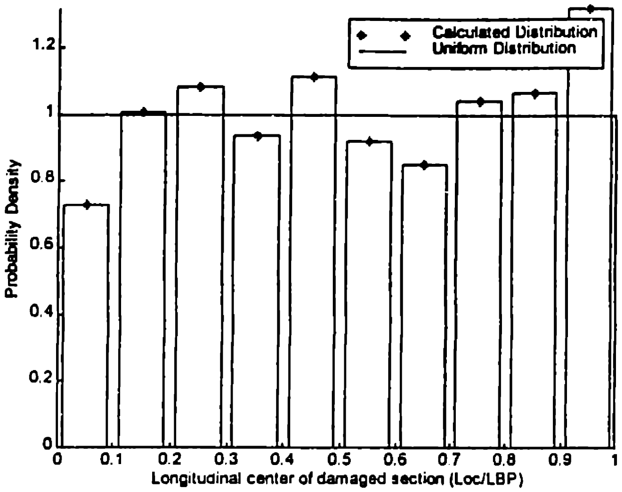


Figure 20: Probability density function for center of damage

The effect is not dramatic, but it is seen consistently through all of the simulations. A plot of the pdf from the Regulations and the pdf resulting from the simulation for the single hull ship is shown in Figure 20. The results for the other ships are indistinguishable from the single hull case. Since the oil outflow is calculated from the damage plan for each individual collision, this pdf does not play any part in assessing the performance of the ship designs. Pdf's for the other ship designs are not presented because they are essentially identical.

5.2.2. Longitudinal Extent of Damage

The pdf's describing longitudinal extents of damage are shown in Figure 21.

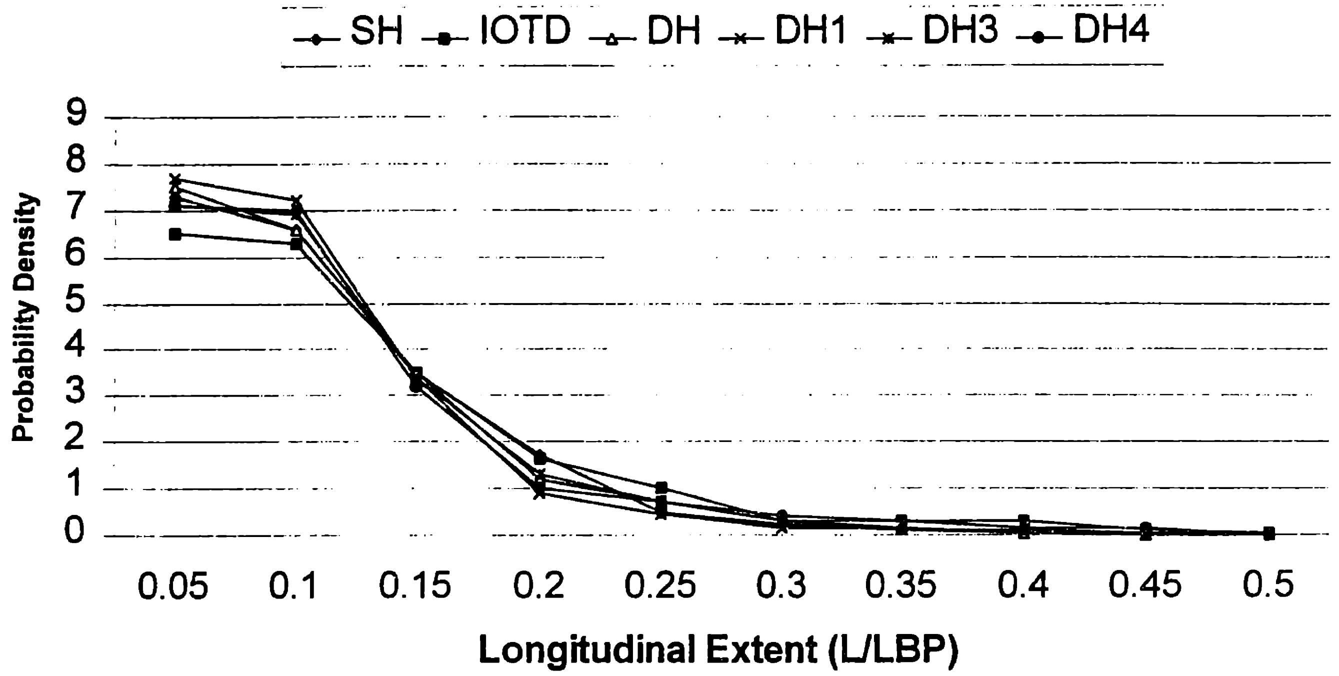


Figure 21: Longitudinal extent of damage pdf's

These pdf's are similar within the group because all the ships have similar side protection systems. The double hull actually has the least protection in side collision. The single hull has substantial protectively located ballast tankage, and the mid-deck has even more side protection.

Note that longitudinal damage extents exceeding 0.3L are predicted for all ship designs. The Regulation does not predict any damage beyond this length. There are three explanations for this:

- The small number of collision cases that form the basis of the Regulation pdf's did not show damages of this extent. It is possible that a larger sample size would have shown this. The pdf's developed in this analysis are based on 5000 collisions, and the pdf's show that damage exceeding 0.3L is uncommon.

- The assumptions that are included in the simulation model are intentionally conservative. This produces greater extents of damage than a model without these assumptions. The conservative assumptions that cause this include the treatment of added mass and the rigid striking bow assumption.

- The Regulation pdf's are based on collisions that result in rupture of the hull. Some of the collisions that result in damage extents exceeding 0.3L do not result in hull rupture. They can be characterized as "glancing blow" collisions. Collisions of this type are excluded from the IMO pdf's.

Overall, correspondence with the "real world" data is good, and shows that the collision simulation model produces reasonable results.

5.2.3. Transverse Extent of Damage

The pdf's describing transverse extent of damage for each ship design are shown in

Figure 22.

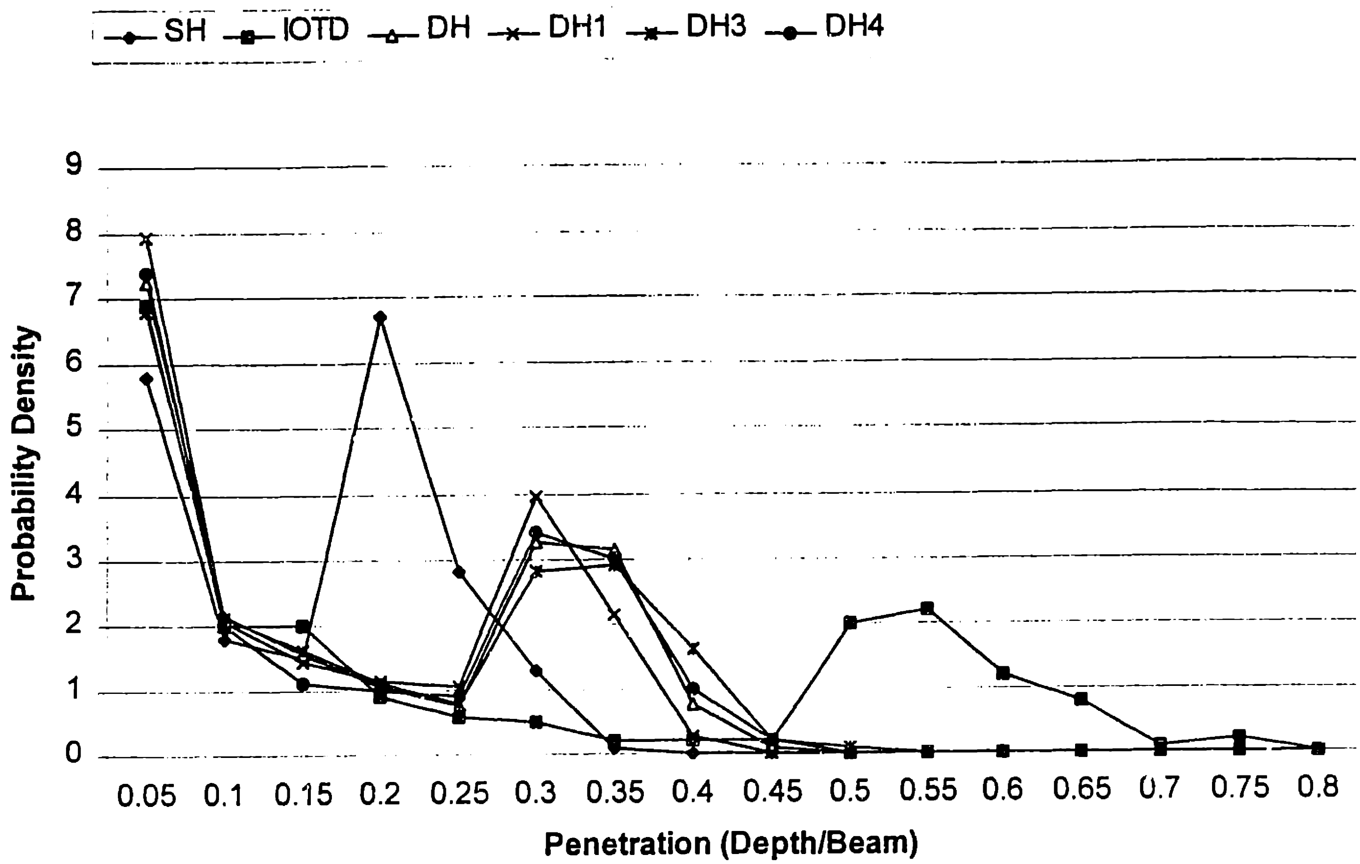


Figure 22: Transverse penetration pdf's

The general trend shows two main concentrations of transverse penetration. There is a cluster of collisions that are halted or nearly halted by the shell of the ship. There is another cluster of cases that are stopped by the next internal membrane. This can be seen in all ship designs.

- The single hull ship shows this second cluster around 0.2B, which corresponds to the longitudinal bulkhead at 0.1875B.
- The double hull ships show this second cluster around 0.3B, which corresponds to the internal longitudinal bulkhead at 0.2975B. It is not possible to see a similar grouping between the inner and outer hull because they are so closely spaced. Two meters is a difference of 0.042B. This is too fine to be resolved by the bin sizes of 0.05B.

- The mid-deck ship has an internal cargo bulkhead at 0.1145B so the second cluster is easier to identify. The mid-deck ship also has another cluster at 0.5B, which corresponds to the centerline bulkhead.

The mid-deck and double hull ships generally show greater transverse penetration. This is because they have less in-plane structure (deck thickness) to absorb energy via the Minorsky mechanism. All the ships were designed to satisfy ABS requirements for section modulus, but the extra material used in constructing the double hull or internal oil-tight deck allowed thinner plate to be used in the bottoms and decks.

DH1 shows transverse penetration comparable, but still greater than the single hull ship. This is seen in spite of having greater aggregate deck thickness (5.572 cm vs 4.381 cm for the single hull). This shows that shell plating is a significant energy absorber. The significant energy absorption by shell plating is what causes the data scatter in the low energy regime of Minorsky's original analysis.

Thinking about collision resistance in terms of the Minorsky interaction, it is clear that the more efficiently a ship is designed (assuming traditional structural designs), the less collision resistance the ship will have. This is similar to the argument that improvements in engineering knowledge have resulted in decreased "safety room" as the design margins have been whittled away over time by improved knowledge of structural response.

5.2.4. Joint Longitudinal/Transverse Damage pdf's

Figure 23 through Figure 28 are joint probability density functions showing the distribution of coupled transverse and longitudinal extents of damage. The damage results of each collision scenario are recorded and analyzed as a set so that the transverse and longitudinal extents can be plotted as a dependent pair. Each ship shows slightly different results, mostly in transverse extent of damage. The differences result from the same factors previously discussed relating to transverse extent of damage. It is clear that there is coupling between transverse and longitudinal extent of damage. One can see how the pdf for one extent (say transverse extent) depends on the other extent of damage by imagining a two-dimensional "slice" of the joint pdf taken at a representative value of the other extent (longitudinal, in this case). As one moves the "slice" along the selected axis, the shape of the slice changes. It is impossible to capture this effect with the methods in the current Regulation without a joint pdf plot. The method used in this analysis captures the effect completely. This coupling effect may or may not be important. The only way to tell is to use the outflow characteristics developed by this analysis to calculate a "pollution prevention index" for each of these ships, and then compare to the pollution prevention index calculated via the IMO Regulation.

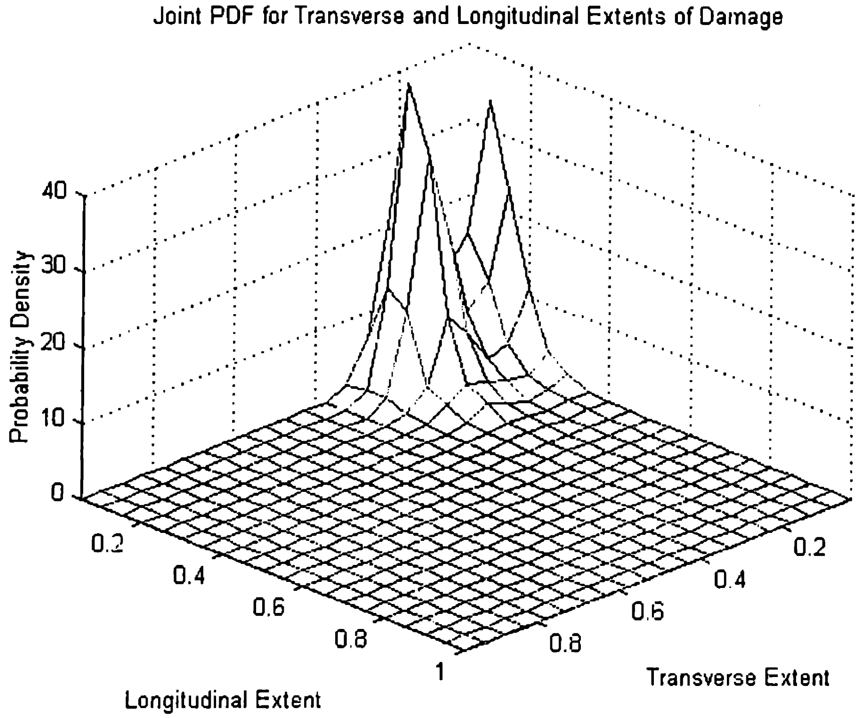


Figure 23: Joint damage pdf for single hull

The single hull joint pdf shows the impact of the internal longitudinal bulkhead at 0.1875B. As a ship strikes, the first resistance encountered is from the shell. In this model, the direction of the force developed by the shell tends to "turn away" the incoming ship, and directs collision damage into longitudinal extent. If a ship "punches through" the exterior shell, damage tends to be transverse until a second longitudinal membrane is contacted. The same turning force is exerted, and damage is directed longitudinally.

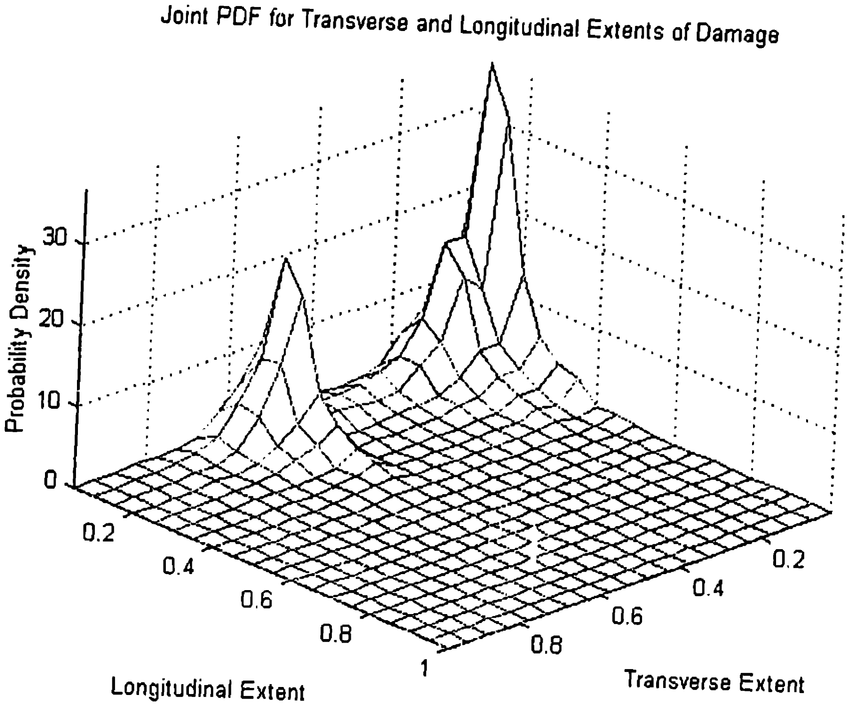


Figure 24: Joint damage pdf for mid-deck tanker

Figure 24 shows the same effects, including the protective bulkhead at 0.1142B. This is seen as a smaller ridge inboard and parallel to the ridge created by the shell membrane. Of the cases where a striking ship proceeds past this bulkhead, many proceed all the way to the center line bulkhead. This ship design has the most transverse penetration of those analyzed.

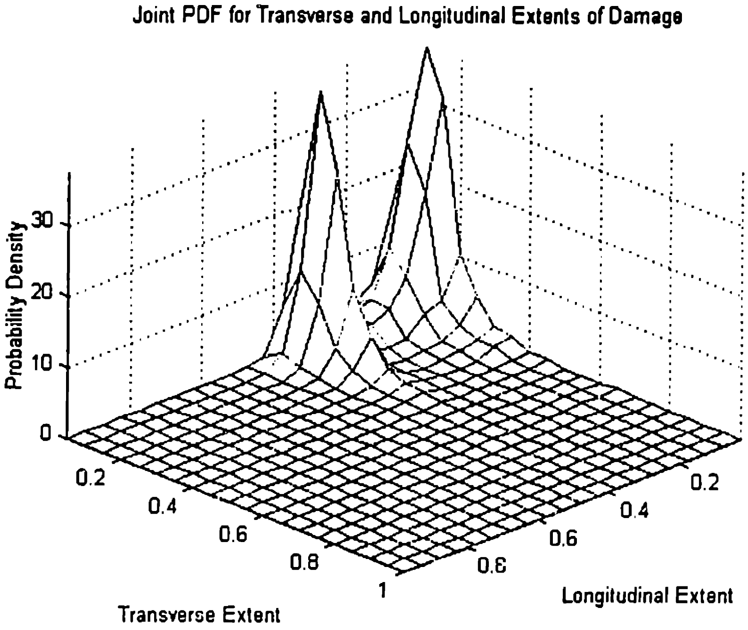


Figure 25: Joint damage pdf for baseline double hull

Figure 25 through Figure 28 show the joint damage pdf's for the double hull series. They have the same characteristics because of their similarity in design. They show similar behavior to the other ships with respect to the longitudinal membranes. The double hull series does not show the effect of the inner hull very clearly because of the grid size used in the joint probability plots. This is a trade-off. A fine mesh size would provide the resolution needed to see the effect of the closely spaced hulls. A larger mesh size prevents random variation in the bin populations from obscuring the trends. The mesh size used here is the smallest that prevents random variation from becoming problematic. Using a finer mesh requires more than 5000 cases.

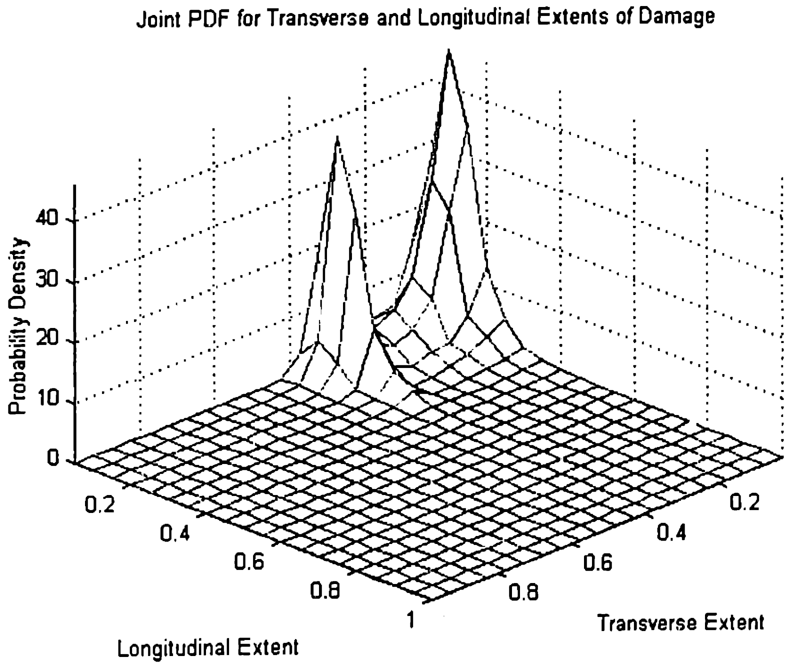


Figure 26: Joint pdf for double hull DH1

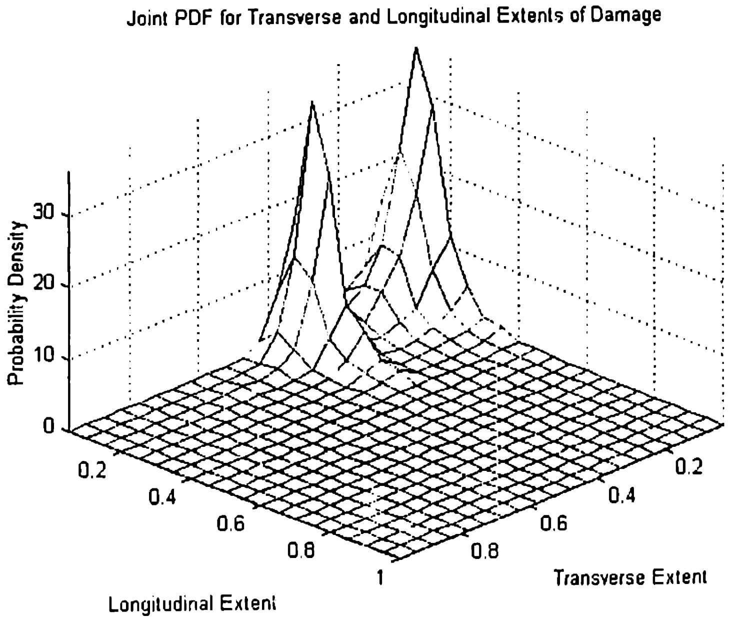


Figure 27: Joint damage pdf for double hull DH3

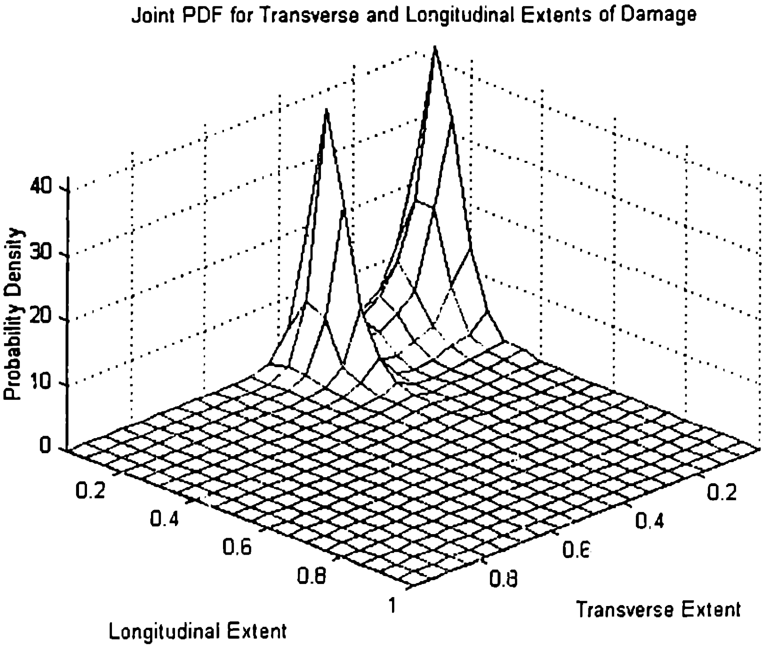


Figure 28: Joint damage pdf for double hull DH4

5.3. Oil Outflow pdf's

The oil outflow pdf's for all ships are shown in Figure 29 through Figure 34.

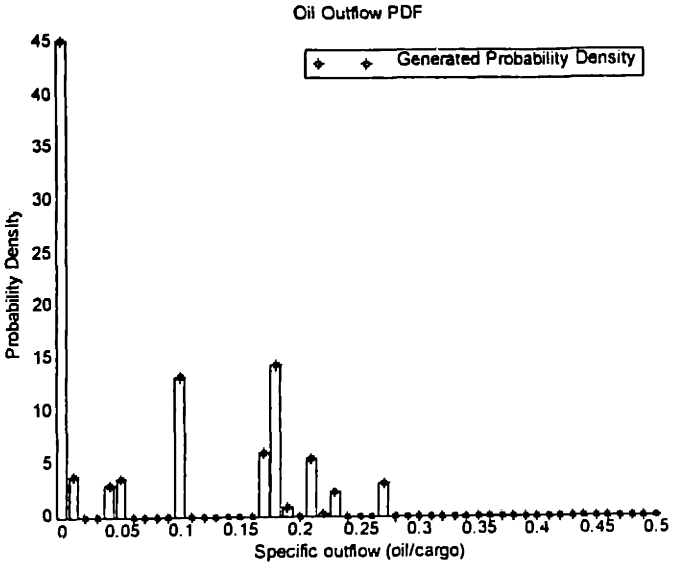


Figure 29: Oil outflow pdf for single hull

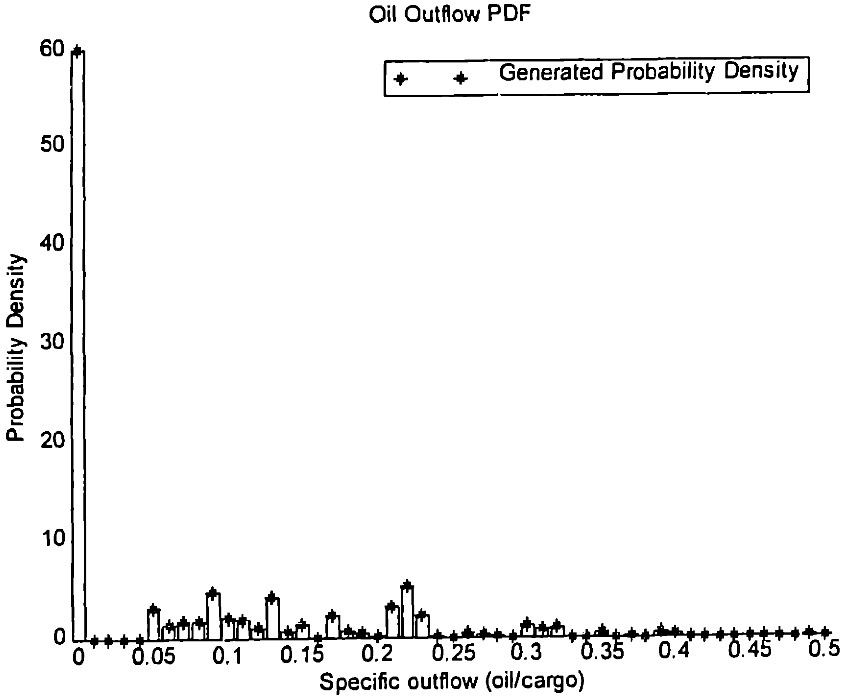


Figure 30: Oil outflow pdf for mid-deck

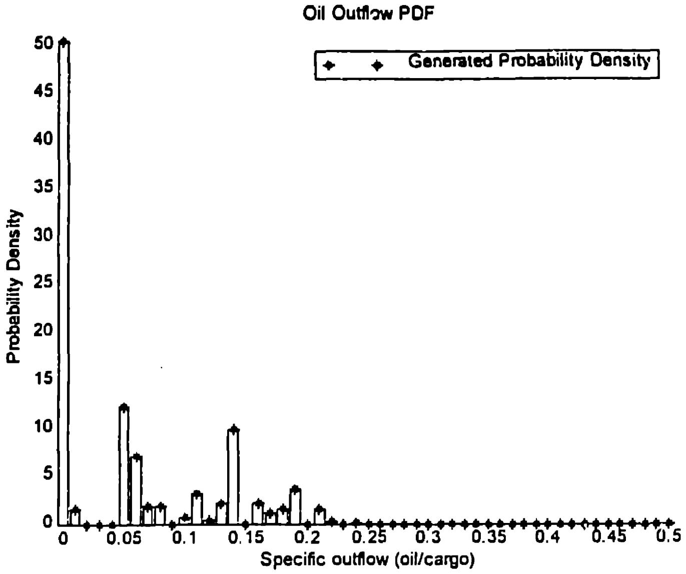


Figure 31: Oil outflow pdf for baseline double hull

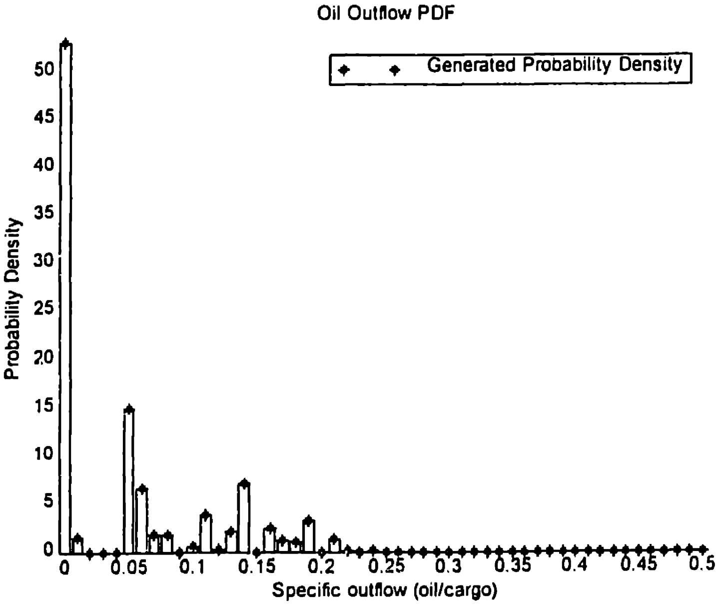


Figure 32: Oil outflow pdf for DH1

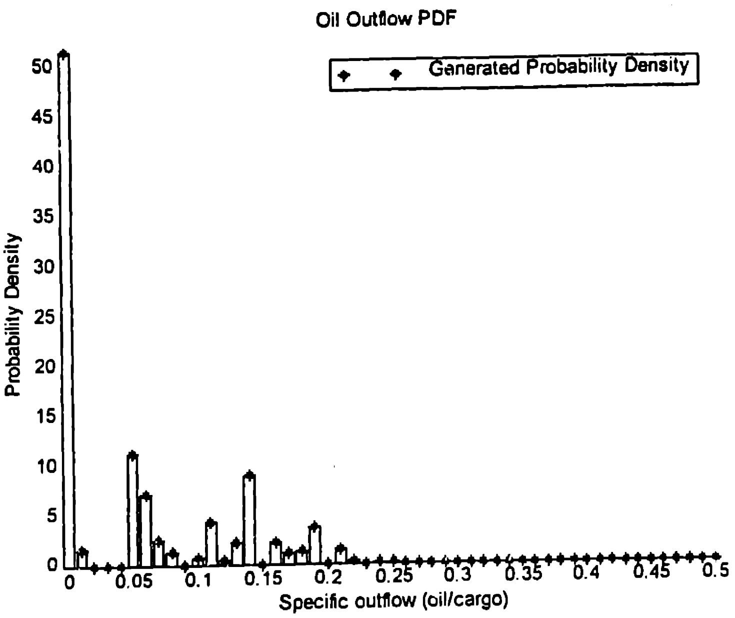


Figure 33: Oil outflow pdf for DH3

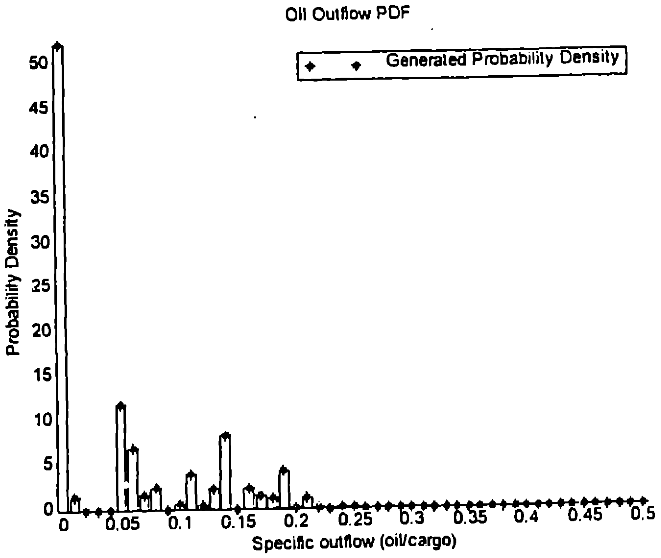


Figure 34: Oil outflow pdf for DH4

Examination of these pdf's reveals three items of interest:

- The first column on the left-hand side of the graph corresponds to the probability of zero outflow. The way the pdf graphs are constructed, this probability can be approximated from the value of the "zero" column, although some cases of very small outflow are included in this column.
- The height of the remaining columns corresponds to the likelihood of an oil outflow of a given size.
- The discrete nature of the oil outflow shows that there are only a few values of oil outflow possible. The particular values depend on the ship design, and are a result of assuming that once a cargo tank is ruptured all the oil in that tank is lost.

These figures are interesting because they depict the full range of possible results given that a collision has occurred. The probability of zero outflow includes cases where the hull is not ruptured as well as those cases where the hull fails. The remainder of the graph shows the range and probability of all outflows. The current regulation does not require such a plot, although one could be constructed using the Regulation methodology.

6. Conclusions

A rational method of calculating the probabilities of oil outflow has been demonstrated. The method:

- Considers structural detail
- Treats ships appropriately based on size
- Considers the coupled nature of longitudinal and transverse extents of damage
- Is tailored to the particular ship design
- Has a statistically significant basis for prediction

This method is rapidly adaptable to other ship designs. It is fast, simple, and treats structural differences in a rational manner. It is an improvement over the current IMO Regulation methodology.

With regard to the particular ship designs considered here, the following conclusions are drawn:

- The double-hull provides the best performance of the designs considered
- The mid-deck design is superior to the double-hull in terms of providing maximum chance of preventing all outflow in a collision, but has higher mean outflow.
- The single hull design shows larger spills than the double-hull and more frequent oil spills than the mid-deck.
- Subdivision is critical in limiting oil outflow. The mid-deck design could be comparable to the double-hull if an improved subdivision scheme were implemented. The performance of the single hull ship would also be improved if more subdivision were used.

7. Future work

There are several areas where this analysis could be improved. Most of the improvements lie in the elimination or refinement of assumptions made in this work. Efforts that would provide the greatest benefit include:

- Elimination of the rigid bow assumption. Damage surveys following collisions typically reveal that a substantial amount of damage is also done to the bow of the striking ship. This analysis has assumed that the striking bow is impervious to damage. This causes more energy to be absorbed by the struck ship, and therefore produces more extensive damage.

- Include resistance mechanisms for major transverse elements. This analysis does not explicitly consider the transverse elements. Finding a way to include the effects of transverse webs and bulkheads would improve the responsiveness of the model to structural detail, and eliminate another conservative assumption that tends to over-predict extents of damage.

- Collect and implement better statistics for input parameters. The pdf's for input parameters used here are the result of expert opinion and the calibration process described in Section VI.. This process was consciously undertaken with the goal of producing damage extent pdf's similar to those in the Regulation. A better approach would be to collect enough data from ship operations to base the input pdf's on real-world statistical data. A comparison should then be made to real-world statistical data on actual collisions. The collision data should not be "scrubbed" to include only the collisions that result in hull failure. The collision data should be sorted by ship type.

- The model should be run using actual collision data to see how well it predicts the results of a single case where the inputs and results are well known. This would guide further improvement efforts.
- The model's use of the Minorsky mechanism should be eliminated. This mechanism represents a range of failure modes aggregated into a single constant. When first-principles methods are mature enough to represent the majority of these failure modes, the individual failure mode calculations should replace the broad-brush Minorsky method.

8. References

[1] Ship Structures Committee, Critical Evaluation of Low-Energy Ship Collision - Damage Theories and Design Methodologies (Two Volumes), Report SSC-284, 1979

[2] Minorsky, V.U., "An Analysis of Ship Collisions With Reference to Protection of Nuclear Power Plants," Journal of Ship Research, Vol. 3, No. 1, 1959

[3] Reardon, P. and Sprung, J. "Validation of Minorsky's Ship Collision Model and Use of the Model to Estimate the Probability of Damaging a Radioactive Material Transportation Cask during a Ship Collision," Proceedings of the SNAME/SNAJ Joint Conference, 1996

[4] Hutchison, B., Principal Investigator for The Glosten Associates, "Barge Collisions, Rammings and Groundings – An Engineering Assessment of the Potential for Damage to Radioactive Material Transport Casks," Contractor Report to Sandia National Laboratories, January 1986

[5] Newman. J.N., Marine Hydrodynamics, The MIT Press, Cambridge, MA, 1992.

[6] Wierzbicki, T., and Simonsen, B., Global Structural Model of Bow Indentation into Ship Side, Report 2 in Rupture Analysis of Oil Tankers in a Side Collision, September 1996

[7] Amrozowicz, M. and Brown, A. "Tanker Environmental Risk – Putting the Pieces Together" Proceedings of the Joint SNAME/SNAJ Conference, 1996

[8] Rawson, C., "A Probabilistic Evaluation of Tankship Damage in Grounding Events", Master's Thesis, Massachusetts Institute of Technology, 1998

[9] Keith, V., Heid, R., and Vann, R., "Technical Support for Risk Analysis and Shipment of Plutonium by Sea," ECO, Inc., Annapolis, MD, November 1995.

[10] Rawson, C., "Suezmax Tanker Designs for Evaluating Damage", Special Project Report for MIT Course 13.414, (Prof. A.J. Brown) February, 1998

[11] Hogg, R. and Ledolter, J., Applied Statistics for Engineers and Physical Scientists, MacMillan Publishing, New York 1987

[12] International Maritime Organization, "Report on IMO Comparative Study on Oil Tanker Design", February 1992

[13] Lu, G. and Calladine, C. R., "On the Cutting of a Plate by a Wedge," Int. J. Mech. Sci., **32**, 293 (1990)

[14] Wierzbicki, T. and Thomas, P., "Closed-Form Solution forWedge Cutting Force Through Thin Metal Sheets," Int. J. Mech. Sci., **35**, 209 (1993)

[15] McDermott, J. F., et al., "Tanker Structural Analysis for Minor Collisions," SNAME Transactions, Volume 82, 1974

[16] Reckling, K. "Mechanics of Minor Ship Collisions", Int J. of Impact Engineering, Vol. 1, No. 3, pp. 281-299, Pergamon Press, Ltd., 1983

[17] Akita, Y., et al., "Studies on Collision-Protective Structures in Nuclear Powered Ships," Nuclear Engineering and Design 19, pp. 365-401, North-Holland Publishing Company, 1972

9. Appendix

All of the computer code used in this analysis is included. All of the code is in the form of MATLAB® script files. MATLAB® Version 5 was used throughout this analysis

The script files are presented in the order used. The first script, calculate.m, calls the following scripts as required to complete the calculations and produce the output.

To run the code, all files need to be in a common directory in the MATLAB path. Typing "calculate" at the MATLAB prompt starts the execution. The analysis is guided by user input prompted by a series of questions.

In order to analyze different double-hull ships, changes need to be made in the "calculate.m" script for new deck thicknesses, shell plating thickness, and internal bulkhead thickness.

In order to change the frame spacing, a change needs to be made in "constants.m" in the variable "spacing". This is marked with comments in the code.

% calculate.m

```
% the control script to run each subroutine in order
% Date created:  10/15/97
% Last revision:   3/10/98

% Inputs:      Number of desired runs, "n"

% Output:
%    Results of momentum and energy balance for final
velocities, collision energy, and time step
%    "dt" - an estimate of the appropriate time step to be
used in the time domain simulation
clf
clear
%n=1;
n = input('How many runs, please?    ');
type = input('Press 1 for single hull, 2 for double-hull, or
3 for IOTD.....   ');


% Set up storage arrays
E =zeros(n,1); % pre-allocated memory for EA
Eshell = zeros(n,1);  %pre-allocated memory for Emem
V1=zeros(n,1); % pre-allocated memory for Vel1
V2=zeros(n,1); % pre-allocated memory for Vel2
bow_alpha = zeros(n,1); % pre-allocated memory for alpha
Len = zeros(n,1);   % pre-allocated memory for L
R = zeros(n,1);% pre-allocated memory for R - a flag for
outer shell rupture
CA = zeros(n,1);    % pre-allocated memory for collision
angle (phi)
P = zeros(n,1);% pre-allocated memory for penetration depth
Cen = zeros(n,1);   % pre-allocated memory for center of
damaged section
ICP = zeros(n,1);  % pre-allocated memory for initial
contact point
FCP = zeros(n,1); % pre-allocated memory for final contact
point
Minorsky = zeros(n,1); % pre-allocated memory for the
Minorsky constant
Mass = zeros(n,1); % preallocated memory for striking ship
mass
Time = zeros(n,1); % pre-allocated memory to track collision
time
Outflow = zeros(n,1);    % pre-allocated memory for oil
outflow
test = 0;
if type == 1
```

```
     test = input('Press 1 if this is a calibration run,
zero if not....       ');
end
constants
j=zeros(n,1);
for i=1:n;

     vargen
     energy
%    if Ea <0
%         Ea = 0;
%    end

     if type == 1
          SBH = [13,54.1,95.2,136.3,177.4,204.7,218.5,265];
          single1
          if rupture > 0
               single2
               single3
               if rupture2 > 0
                    single4
               end
          end
          oil = (Tank1P + Tank1S) * 15935 + Tank2 * 29041 +
(Tank3 + Tank4) * 29592 + Tank5P * 6520 + Tank5 * 29052 +
SlopTk * 2195;
          oil = oil/166577;
     end
     if type == 2
          sigma_y = 3.6e+08; % yield stress of HT36 steel,
in Pa
          dKE = dKE*(36/24); % correction for HT36 steel
vice MS24 per Daidola & Pet
          t = .0466;    % aggregate deck thickness for DH1
          t_plate = .01614;  % shell thickness for DH1
          DBH = [13,46,79,112,145,178,211,218.5,265];
          double1
          if rupture > 0
               t_plate = .01694;   % inner skin thickness
               double2
               double3
               if rupture2 > 0
                    t_plate = .01256;  % centerline bulkhead
thickness
                    double4
                    double5
                         if rupture3 > 0
                              double6
                         end
               end
          end
          oil = (Tank1P + Tank1S) * 9658 + Tank2P * 8348 +
Tank2C * 13167 + (Tank3P + Tank5P + Tank6P) * 8767 +
```

```
(Tank3C+ Tank4C + Tank5C) * 13828 + Tank6C * 16908 +
SlopTk*1953;
          oil = oil/164079;
     end
     if type == 3
          sigma_y = 3.6e+08;  %yield stress of HT36 steel,
in Pa
          dKE = dKE*(36/24);  % correction for HT36 steel
vice MS24 per Daidola & Pet
          t = .0568;          % sets aggregate deck
thickness for the IOTD design
          t_plate = .018;          % shell thickness for
IOTD
          IBH = [13,46,79,112,145,178,211,218.5,265];
          iotd1
          if rupture > 0
               t_plate = .01843;   % inner skin  thickness,
in meters
               iotd2
               iotd3
               if rupture2 > 0
                    t_plate = .01662;   % centerline
bulkhead thickness, in meters
                    iotd4
                    iotd5
                         if rupture3 > 0
                              iotd6
                         end
               end
          end
          oil = (Tank1P + Tank1S) * 3979 + Tank1BOT * 7959 +
(Tank2P + Tank2S) * 6909 + Tank2BOT * 13819 + (Tank3P +
Tank3S + Tank4P + Tank4S + Tank5P + Tank5S + Tank6P +
Tank6S) * 7326 + (Tank3BOT + Tank4BOT + Tank5BOT) * 14652 +
Tank6BOT * 17903 + SlopTk * 1626;
          oil = oil/167273;
     end
write
end
% Remove cases that fail the integration error test from the
data set
howmany
% zero out the cases that fail the test
for i = 1:n
     if Time(i) == 0
          P(i) = 0;
          Len(i) = 0;
          Cen(i) = 0;
          bow_alpha(i) = 0;
          CA(i) = 0;
          ICP(i) =0;
   end
   if P(i) > Beam
      Time(i) = 0;
```

```
          P(i) = 0;
          Len(i) = 0;
          Cen(i) = 0;
          bow_alpha(i) = 0;
          CA(i) = 0;
          ICP(i) =0;
     end

end

%    Now, remove those cases from the population using the
"nonzeros" function

     Len = nonzeros(Len);
     P = nonzeros(P);
     Cen = nonzeros(Cen);
     bow_alpha = nonzeros(bow_alpha);
     CA = nonzeros(CA);
     ICP = nonzeros(ICP);

% Outflow is not removed here because applying the
'nonzeros'
% function would also take out all the cases where the run
was
% acceptable but no outflow occurred.  These zeros will not
affect
% the calculated mean as long as the sum is divided by the
proper
% population size.


% Non-dimensionalize the output vectors

%Len = Len/LBP;
P = P/Beam;
%Cen = Cen/LBP;

% Plot a summary figure

figure(1)
clf
bin=[0.1:0.1:1];
subplot(3,1,1)
hold on
hist(Cen,bin)
xlabel('Longitudinal center of damaged section')
ylabel('number of occurrences')
bin=[0.05:0.1:1];
subplot(3,1,2)
hold on
hist(Len,bin)
xlabel('Longitudinal Extent of damaged section')
ylabel('number of occurrences')
subplot(3,1,3)
```

```
bin=[0.1:0.05:1];
hold on
hist(P,bin)
xlabel('Transverse Penetration')
ylabel('number of occurrences')

% Produce other output plots, and goodness-of-fit data if
this is a calibration run
if test == 1
     calibrate
else
     output
end

% Plot the 3D joint pdf
joint
rotate3d
```

% vargen.m

```
% Routine to generate random variates for use in the
collision script routine

% Input: the number, n, of variates to produce.  Currently
entered in script as 1000

% Output: Generated variates
%  Variable label   Variable description
%         U1                    Struck ship speed (kt)
%         U2                    Striking ship speed (kt)
%         alpha                 Striking ship half-entrance
angle (degrees)
%         phi                   Collision angle (degrees
relative, from struck ship)
%         L                     Location of collision point on
struck ship (meters from FP)
%         Lnd                   The collision point, non-
dimensionalized by LBP
%         dKE                   The value for the energy
absorption coeffiecient in Minorsky's equation
%         m2                    The mass of the striking ship
% Date created: 9/12/97
% Last updated: 4/23/98

% reset the random number generators to new states.

rand('state',sum(100*clock));
randn('state',sum(100*clock));

% Generate variates in proper distributions and ranges

% Input for speed generation section

himu = 10;
lomu = 5;       % mean velocity peaks in knots
hisigma = 1;    % and standard deviations
losigma = 1;    % for high and low speed
alphamu = 38;   % striking ship half-entrance angle
alphasigma = 5;


     if rand > 0.5
          mu = himu;
          sigma = hisigma;
     else
          mu = lomu;
          sigma = losigma;
     end
```

```
     U1 = mu + randn*sigma;

   rand('state',sum(100*clock));
   randn('state',sum(100*clock));

   if rand > 0.5
          mu = himu;
          sigma = hisigma;
     else
          mu = lomu;
          sigma = losigma;
     end
     U2 = mu + randn*sigma;

% set impacting bow half-entrance angle
     alpha = alphamu + randn*alphasigma;  % Normally
distributed about alphamu

   % set collision angle (phi)
     check =1;
while check ==1

     phi = 180*rand;

%    phi = 90 + randn*45;      % Normally distributed on [0-
180] degrees

%  This version is for a two mode normal dist for phi

%    himu = 135;
%    lomu = 45;
%    sigma = 30;
%    if rand > 0.25
%         mu = himu;
%    else
%         mu = lomu;
%    end
%    phi = mu + randn*sigma;

     check = or(phi<1,phi>180);
end

% set collision location point L, in meters
Lo = LBP*rand;
Lnd = Lo/LBP;    % Lnd is the impact point non-
dimensionalized by LBP
   if phi >179      % Sets collision point for end-on
collisions....
      Lo = LBP;
   end
   if phi < 1
      Lo = 1;
   end
% select value for the Minorsky resistance function:
```

```
     dKE = 47.1 + randn*8.8;

% Choose a mass for the striking ship in lton, convert to
kg, and build
% a physical mass array - this replaces the value assigned
in constants.m

m1 = 120000 + randn*30000;
m1 = m1*2240/2.2046;
M1 = [m1,0;0,m1];

% Calculate the principal dimensions of the striking ship by
scaling by the cube root of
%  the mass ratio.

LBP2 = ((m1/m2)^(1/3))*LBP;
Beam2 = ((m1/m2)^(1/3))*Beam;
Draft2 = ((m1/m2)^(1/3))*Draft;

% By maintaining dimensional similitude, the added mass
coefficients remain constant.....
```

% energy.m

```
% This script calculates the initial energy and momentum of
the two ships,
% the final momentum and velocity of the two-ship system
(assumes that they
% travel together in the end state), and then the final
energy of the system.

% The difference between the initial energy of the system
and the final energy of the
% system is used in the next script, which will determine
how the energy is
% expended in deformation of the struck ship's structure.

%    Date created:  10/7/97
%    Last revision:   4/23/98

% Inputs:      Parameter variables from "vargen.m"
%              Ship masses - from "constants.m"
%              Added mass tensor for each ship

% Output:      Ea - the energy that must be expended in
structural deformation
%              dt - an estimate of the appropriate time step
to be used in the time domain simulation


% Calculate added mass tensor for each ship
     % theta(1 or 2) is the angle between the ship's
principal axes and the ship's velocity vector.  Prior to
          % the collision, this is zero for both ships
          theta1 = 0;
          theta2 = 0;

     % Calculate components of added mass tensor for x & y
motion

     A1 = m1*[(a11*sin(theta1*pi/180)^2 +
a22*cos(theta1*pi/180)^2), ((a11-
a22)*sin(theta1*pi/180)*cos(theta1*pi/180)); ((a11-
a22)*sin(theta1*pi/180)*cos(theta1*pi/180)),
(a11*sin(theta1*pi/180)^2 + a22*cos(theta1*pi/180)^2)];
     A2 = m2*[(a11*sin(theta2*pi/180)^2 +
a22*cos(theta2*pi/180)^2), ((a11-
a22)*sin(theta2*pi/180)*cos(theta2*pi/180)); ((a11-
a22)*sin(theta2*pi/180)*cos(theta2*pi/180)),
(a11*sin(theta2*pi/180)^2 + a22*cos(theta2*pi/180)^2)];
```

```
% Combine physical and added mass matrices for total
"virtual" mass matrix

     VM1 = M1 + A1;
     VM2 = M2 + A2;

% Construct velocity vectors for each ship (in units of
meters/sec)
% so that V1 = [x-velocity, y-velocity], etc

Vel1 = 1.688*(12/39)*[U1;0];
Vel2 = 1.688*(12/39)*[-U2*cos(phi*pi/180); -
U2*sin(phi*pi/180)];

% Construct momentum matrices for each ship (units of
kg*m/sec)
     P1 = VM1*Vel1;
     P2 = VM2*Vel2;

% Total momentum is then:
     PT = P1 + P2;

% with magnitude
     PF = sqrt(PT(1)^2 + PT(2)^2);

% at angle to global coordinate system of:
     chi = atan2(PT(2),PT(1));
     chi_deg = chi*180/pi;

% With standard "small changes in mass distribution"
assumption outlined in thesis, the final virtual mass tensor
is:
     VMF = VM1 + VM2;

% and the final translational velocity is then:
     VF = VMF\PT;

%  Rotational Energy Calculation

% Find center of mass relative to struck ship origin
     L = Lo;
     x = m2*(((LBP2/2)-L)+(LBP/2)*cos(phi*pi/180))/(m1+m2);
     y = (m2*(LBP/2)*sin(phi*pi/180))/(m1+m2);

     rf = sqrt(x^2 + y^2);

% at angle to global origin of:

     beta = atan2(y,x);
     beta_degree = beta*180/pi;

% Calculate physical mass moment of inertia for both ships
     J661 = m1*(LBP^2)/12;
     J662 = m2*(LBP2^2)/12;
```

```
% Calculate added mass moment of inertia for both ships
(assumes pure sway motion arises
% from rotation about the final center of mass, so this is
the same as added mass in sway)
     J66A1 = (2.378*rho*Draft^2*LBP^3)/24;
     J66A2 =  (2.378*rho*Draft2^2*LBP2^3)/24;

% Combine to get virtual mass moment of inertia
     J66V1 = J661 + J66A1;
     J66V2 = J662 + J66A2;

% Use parallel axis theorem to calculate the virtual mass
moment of inertia about the
% system center of mass for the striking ship.  Assuming
again, that the motion of striking ship is
%  entirely in sway
% I-new = Icg + Mr^2

     J66V2C = J66V2 + m2*(1+a22)*(((LBP/2)-L-x)^2 +
(((LBP2/2)*cos(phi*pi/180))-y)^2);

% Use parallel axis theorem to calculate the virtual mass
moment of inertia about the
% system center of mass for the struck ship.  Assuming, as
before that the motion of the struck ship is
%  entirely in sway
% I-new = Icg + Mr^2

     J66V1C = J66V1 + m1*(1+a22)*((x)^2 + (y)^2);

% Combine the virtual mass moment of inertia of Ship 1 about
the system center of mass with the
% virtual mass moment of inertia of Ship 2 about the system
center of mass to obtain the
% virtual mass moment of inertia of the two-ship system
about the system center of mass

     J66F = J66V2C + J66V1C;

% Using this to solve for the final rotational velocity
gives:

     r2 = (LBP/2) - L;  % the "arm" through which the
striking vessel's linear momentum acts

     wf = (1/J66F)*(r2*(sqrt((P2(1)^2 + P2(2)^2)))*sin(phi)
- rf*PF*sin(beta-chi));
     wf_deg_per_sec = wf*180/pi;

% Now, using these quantities, calculate the difference in
initial and final energy states of
% the system.  This is an approximation of the energy that
must be absorbed by the structure.
```

```
% It is an approximation because the final virtual mass
moment of inertia is calculated by
% assuming that the final mass distribution is the same as
at moment of impact and under 90
% degrees collision angle.

      Ea = 1/2*((dot(P1,Vel1)) + (dot(P2,Vel2)) -
(dot(PT,VF)) - ((J66F*wf)*wf));

% These quantities calculated to assist in troubleshooting
code - remove for faster execution
%    KE1 = 1/2*(dot(P1,Vel1));
%    KE2 = 1/2*(dot(P2,Vel2));
%    KEf = 1/2*(dot(PT,VF));
%    KEr = J66F*wf*wf;

% and an approximation of the total time elapsed during the
collision is
     T =
(pi/2)*sqrt((1/(t*300000*tan(alpha*pi/180)))*((VM1(1,1)*VM2(
1,1)/(VM1(1,1)+VM2(1,1)))));

% and deviating slightly from Hutchison's work, the time-
step for simulating this collision is:

step = T/200;

% (Hutchison used T/100)
```

% single1.m

```
% Script to perform time-domain analysis for single-hulled
tanker collision.
% The "1" indicates that this script is for phase 1 of the
collision, which is
% from the time of impact until the shell membrane ruptures.

% Input:  dt          from energy.m
%         V1          from energy.m
%         V2          from energy.m
%         VM1         from energy.m
%         VM2         from energy.m
%         alpha       from vargen.m


% Output: Generated variates and values

% Date created: 11/3/97
% Last updated: 4/23/98

% Reset flags for tank breaching to zero:
Tank1P = 0;
Tank1S = 0;
Tank2 = 0;
Tank3 = 0;
Tank4 = 0;
Tank5 = 0;
Tank5P = 0;
SlopTk = 0;
% Determine nearest transverse structures, and distance to
each for use in applying
% Van Mater's extension to Jones method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
a = BH(j) - L;
b = L - BH(j-1);
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;
```

```
% Initialize new variables (Subscript 1 = struck ship,
subscript 2 = striking ship)
time = 0;
X1=0;
Y1=0;
X2= (LBP/2)-L+(LBP/2)*cos(phi*pi/180);
Y2= (Beam/2)+(LBP/2)*sin(phi*pi/180);
omega1=0;
omega_dot1 = 0;
omega2=(phi+180)*(pi/180);
omega_dot2 = 0;
defl = 0;
T1 = Vel1;
T2 = Vel2;
rupture = 0;
rupture2 = 0;
s=0;
ddepth = 0;
depth = 0;
Fmin = 0;
Emin = 0;
Eabs = 0;
Emem_last = 0;
dL = 0;
Pen = 0;
dPen = 0;
Pmax = 0;
relvel = 1;  % a temporary value to get through the first
cycle of the time-step routine.
%%%%%%%% Begin time-step routine  %%%%%%%%
while abs(defl) < abs(defl_lim)
if relvel < endvel
     break
end
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
     if L < 0
          L=0;
          break
```

```
      end
      if Pen > Beam
           Pen = Beam;
           break
    end
    if Pen < 0
      break
    end

% Calculate relative translation, penetration, and change in
impact point in this time step
      S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
      S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
      reltrans = (S2-S1)*step;

% Calculate direction of relative translation
      zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
      dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
      dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

% Calculate the membrane deflection
      defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
      if defl < 0
           Emem = sigma_y*t_plate*breadth*defl^2/spacing;
           Fmem = Emem/abs(defl);
      else
           break
      end
      if defl < defl_lim
           defl = defl_lim;
           Emem = sigma_y*t_plate*breadth*defl^2/spacing;
           Fmem = Emem/abs(defl);
           rupture = 1;
      end
%  Calculate the force resulting from the "Minorsky
mechanism"
      % Rtt is the total "resistance factor"
      % dRt is the differential resistance factor for this
time step
      % depth is the distance of penetration
      % ddepth is the differential distance of penetration
      % t is the aggregate in-plane structure thickness
      % alpha is the bow half-entrance angle
      % This formula accounts for the triangular bow wedge
geometry
      % and dynamic collision angle, but places no
```

```
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2)))) +
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.

% Ship 1 (Struck ship)
%    Calculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = pi/2;         % the membrane force is
always normal to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;

%    The acceleration of translation in the global X
coordinate is:
          aX1mem = Fmem*sin(omega1)/VM1(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
```

```
     % so the angular acceleration is:
          omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

% Now calculate accelerations due to the Minorsky
interaction;
% The Minorsky force is assumed to act in the direction
opposite of relative motion.
% Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:

%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
```

```
%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);
%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);
%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-zeta)/J66V2
% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;
%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;
%    Determine which tanks have been breached
     singlecargo
end
```

% single2.m

```
% Script to perform time-domain analysis for single-hulled
tanker collision.
% The "2" indicates that this script is for phase 2 of the
collision, which is
% from the time the outer shell membrane ruptures until the
inner longitudinal cargo
% bulkhead is contacted.

% Input:  all the dynamic variables from single1.m
%         dt         from write.m
%         V1         from write.m
%         V2         from write.m
%         VM1        from
%         VM2        from
%         alpha      from write.m


% Output: Generated variates and values

% Date created: 11/3/97
% Last updated: 4/23/98

%%%%%%%% Begin time-step routine  %%%%%%%%%
while Pen < SS1
if relvel < endvel
     break
end
Eabs = Eabs + Emin;
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
          break
```

```
     end
     if Pen > Beam
          Pen = Beam;
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;
```

```
      % Calculate the delta-KE from the Minorsky relation;
      Emin = dKE*10^6*dRt;

      % The corresponding force is
      Fmin = Emin/abs(ddepth);

 % Ship 1 (Struck ship)

 %    The acceleration of translation from membrane force in
 the global X coordinate is:
           aX1mem = 0;

 %    The acceleration of translation from membrane force in
 the global Y coordinate is:
           aY1mem = 0;

 %    Calculate the angular acceleration about the ship c.g.:
      % the current contact point is:
           CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
      % so the arm that the force acts through is:
           arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
      % so the angular acceleration due to membrane force is:
           omega_dotdot1mem = 0;

% Calculate accelerations due to the Minorsky interaction;
% The Minorsky force is assumed to act in the direction
opposite of relative motion.
% Since this force is in a different direction we must
recalculate the virtual mass
%          First, calculate the angle of resultant force
compared to the ship principal axis
           zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%          Now, calculate the added mass matrix based on this
angle
           A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%          Combine with physical mass to get the virtual mass
matrix
           VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
```

```
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2

%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;

%    The acceleration of translation due to membrane force
in the global Y coordinate is:
          aY2mem = 0;

%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
```

```
          omega_dot2 = omega_dot2 + omega2_dotdot*step;
          relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
          Eabs = Eabs + Emin + Emem - Emem_last;
          Emem_last = Emem;

%         Determine which tanks have been breached
          singlecargo


end
```

% single3.m

```
% Script to perform time-domain analysis for single-hulled
tanker collision.
% The "3" indicates that this script is for phase 3 of the
collision, which is
% from the time of impact on the inner longitudinal
bulkhead, or cargo boundary until that
% membrane ruptures.

% Input:  all the dynamic variables from single2.m
%         dt         from energy.m
%         V1         from energy.m
%         V2         from energy.m
%         VM1        from energy.m
%         VM2        from energy.m
%         alpha      from vargen.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

% Determine nearest transverse structures on the inner
shell, and distance to
% each for use in applying Van Mater's extension to Jones
method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
a = BH(j) - L;
if j > 1
     b = L - BH(j-1);
     else
          b = 0;
end
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;
% Reset deflection counter to zero for this new membrane:
```

```
defl = 0;

% Set new plate thickness for interior bulkhead:
t_plate = .01952;

%%%%%%%% Begin time-step routine  %%%%%%%%
while abs(defl) < abs(defl_lim)
if relvel < endvel
     break
end
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
     if L < 0
          L=0;
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
   end
   if Pen < 0
     break
   end

% Calculate relative translation, penetration, and change in
impact point in this time step
     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
```

```
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

% Calculate the membrane deflection
     defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
     if defl < 0
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
     else
          break
     end
     if defl < defl_lim
          defl = defl_lim;
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
          rupture2 = 1;
     end
%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt =(depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.
```

```
% Ship 1 (Struck ship)
%    Calculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = pi/2;          % the membrane force is
always normal to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;

%    The acceleration of translation in the global X
coordinate is:
          aX1mem = Fmem*sin(omega1)/VM1(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration is:
          omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;         % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
```

```
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:

%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];

%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;

%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);

%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
```

```
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%    Determine which tanks have been breached
     singlecargo

end
```

% single4.m

```
% Script to perform time-domain analysis for single-hulled
tanker collision.
% The "4" indicates that this script is for phase 4 of the
collision, which is
% from the time the inner cargo bulkhead ruptures until the
collision ends.

% Input:  all the dynamic variables from single3.m
%         dt        from write.m
%         V1        from write.m
%         V2        from write.m
%         VM1       from
%         VM2       from
%         alpha     from write.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

%%%%%%%% Begin time-step routine  %%%%%%%%%
while relvel < endvel

Eabs = Eabs + Emin;
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
          break
     end
     if Pen > Beam
          Pen = Beam;
```

```
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;
```

```
     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Ship 1 (Struck ship)

%    The acceleration of translation from membrane force in
the global X coordinate is:
          aX1mem = 0;

%    The acceleration of translation from membrane force in
the global Y coordinate is:
          aY1mem = 0;

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration due to membrane force is:
          omega_dotdot1mem = 0;

%  Calculate accelerations due to the Minorsky interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;       % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2
```

```
%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;
% The acceleration of translation due to membrane force in
the global Y coordinate is:
          aY2mem = 0;
%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;
%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;
% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;
%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%    Determine which tanks have been breached
     singlecargo
```

```
end
```

```
%    singlecargo.m

% the script to determine which cargo bulkheads have been
breached during
% each time step in the simulation for the single hull
tanker

% Date created:  3/10/98
% Last revision:   3/17/98

% Inputs:   L and Pen from the collision phase scripts

% Output:
%    Flags corresponding to the cargo compartments that will
release oil


j = 1;
     while L > SBH(j)
          j = j+1;
     end   % at this end, BH(j) is the first bulkhead aft of
the damage location

     if j == 2;
          if rupture > 0
               Tank1P = 1;
               if Pen >=24
                    Tank1S =1;
               end
          end
     end
     if j == 3;
          if rupture2 > 0
               Tank2 = 1;
          end
     end
     if j == 4;
          if rupture2 > 0
               Tank3 = 1;
          end
     end
     if j == 5;
          if rupture2 > 0
               Tank4 = 1;
          end
     end
     if j == 6;
          if rupture > 0
               Tank5P = 1;
               if rupture2 > 0
                    Tank5 = 1;
               end
          end
     end
```

```
if j == 7;
     if rupture > 0
          SlopTk = 1;
          if rupture2 > 0
               Tank5 = 1;
          end
     end
end
```

% double1.m

```
% Script to perform time-domain analysis for double-hulled
tanker collision.
% The "1" indicates that this script is for phase 1 of the
collision, which is
% from the time of impact until the shell membrane ruptures.

% Input:  dt        from energy.m
%         V1        from energy.m
%         V2        from energy.m
%         VM1       from
%         VM2       from
%         alpha     from vargen.m


% Output: Generated variates and values

% Date created: 12/20/97
% Last updated: 4/23/98

% Reset flags for tank breaching to zero:
Tank1P = 0;
Tank1S = 0;
Tank2P = 0;
Tank2C = 0;
Tank3P = 0;
Tank3C = 0;
Tank4C = 0;
Tank5P = 0;
Tank5C = 0;
Tank5P = 0;
Tank6P = 0;
Tank6C = 0;
SlopTk = 0;

% Determine nearest transverse frame structures, and
distance to each for use in applying
% Van Mater's extension to Jones method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
a = BH(j) - L;
b = L - BH(j-1);
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
```

```
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;

% Initialize new variables (Subscript 1 = struck ship,
subscript 2 = striking ship)
time = 0;
X1=0;
Y1=0;
X2= (LBP/2)-L+(LBP2/2)*cos(phi*pi/180);
Y2= (Beam/2)+(LBP2/2)*sin(phi*pi/180);
omega1=0;
omega_dot1 = 0;
omega2=(phi+180)*(pi/180);
omega_dot2 = 0;
defl = 0;
T1 = Vel1;
T2 = Vel2;
rupture = 0;
rupture2=0;
rupture3=0;
s=0;
ddepth = 0;
depth = 0;
Fmin = 0;
Emin = 0;
Eabs = 0;
Emem_last = 0;
dL = 0;
Pen = 0;
dPen = 0;
Pmax = 0;
%%%%%%%% Begin time-step routine  %%%%%%%%%
while abs(defl) < abs(defl_lim)
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
```

```
     if L < 0
          L=0;
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
   end
   if Pen < 0
     break
   end

% Calculate relative translation, penetration, and change in
impact point in this time step
     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

% Calculate the membrane deflection
     defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
     if defl < 0
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
     else
          break
     end
     if defl < defl_lim
          defl = defl_lim;
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
          rupture = 1;
     end

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
```

```
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.

% Ship 1 (Struck ship)
%    Calculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = pi/2;        % the membrane force is
always normal to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;

%    The acceleration of translation in the global X
coordinate is:
          aX1mem = Fmem*sin(omega1)/VM1(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
```

```
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration is:
          omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;       % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:

%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
```

```
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];

%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;

%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);

%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;       % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
```

```
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%  Determine which tanks have been breached
     doublecargo


end
```

% double2.m

```
% Script to perform time-domain analysis for double-hulled
tanker collision.
% The "2" indicates that this script is for phase 2 of the
collision, which is
% from the time the outer shell membrane ruptures until the
inner bulkhead is contacted.

% Input:  all the dynamic variables from double1.m
%         dt         from write.m
%         V1         from write.m
%         V2         from write.m
%         VM1        from
%         VM2        from
%         alpha      from write.m


% Output: Generated variates and values

% Date created: 11/3/97
% Last updated: 4/23/98

%%%%%%%% Begin time-step routine  %%%%%%%%%
while Pen < DS1
if relvel < endvel
     break
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           Ship 1
%-----------------------------------------------------
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
```

```
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;
```

```
     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Ship 1 (Struck ship)

%    The acceleration of translation from membrane force in
the global X coordinate is:
          aX1mem = 0;

%    The acceleration of translation from membrane force in
the global Y coordinate is:
          aY1mem = 0;

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration due to membrane force is:
          omega_dotdot1mem = 0;

%  Calculate accelerations due to the Minorsky interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
```

```
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2

%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;

%    The acceleration of translation due to membrane force
in the global Y coordinate is:
          aY2mem = 0;

%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
```

```
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%  Determine which tanks have been breached
     doublecargo

end
```

% double3.m

```
% Script to perform time-domain analysis for single-hulled
tanker collision.
% The "3" indicates that this script is for phase 3 of the
collision, which is
% from the time of impact on the inner shell, or cargo
boundary until that
% membrane ruptures.

% Input:   all the dynamic variables from double2.m
%          dt         from energy.m
%          V1         from energy.m
%          V2         from energy.m
%          VM1        from energy.m
%          VM2        from energy.m
%          alpha      from vargen.m


% Output: Generated variates and values

% Date created: 12/30/97
% Last updated: 4/23/98

% Determine nearest transverse structures on the inner
shell, and distance to
% each for use in applying Van Mater's extension to Jones
method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
a = BH(j) - L;
if j > 1
     b = L - BH(j-1);
     else
          b = 0;
end
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;
```

```
% Reset deflection counter to zero for this new membrane:
defl = 0;

%%%%%%%% Begin time-step routine  %%%%%%%%%
while abs(defl) < abs(defl_lim)
if relvel < endvel
     break
end
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
     if L < 0
          L=0;
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
   end
   if Pen < 0
     break
   end

% Calculate relative translation, penetration, and change in
impact point in this time step
     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);
```

```
% Calculate the membrane deflection
     defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
     if defl < 0
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
     else
          break
     end
     if defl < defl_lim
          defl = defl_lim;
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
          rupture2 = 1;
     end
%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((phi+omega1-
omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((phi+omega1-
omega2)*pi/180)^2))))+ abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.

% Ship 1 (Struck ship)
```

```
%    Calculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = pi/2;        % the membrane force is
always normal to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;

%    The acceleration of translation in the global X
coordinate is:
          aX1mem = Fmem*sin(omega1)/VM1(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration is:
          omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;       % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
```

```
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:

%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];

%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;

%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);

%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
```

```
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%  Determine which tanks have been breached
     doublecargo

end
```

% double4.m

```
% Script to perform time-domain analysis for double-hulled
tanker collision.
% The "4" indicates that this script is for phase 4 of the
collision, which is
% from the time the inner shell membrane ruptures until the
inner cargo bulkhead is
% contacted.

% Input:  all the dynamic variables from double3.m
%         dt         from write.m
%         V1         from write.m
%         V2         from write.m
%         VM1        from
%         VM2        from
%         alpha      from write.m


% Output: Generated variates and values

% Date created: 1/5/98
% Last updated: 4/23/98

%%%%%%% Begin time-step routine  %%%%%%%%
while Pen < DS2
if relvel < endvel
     break
end
Eabs = Eabs + Emin;
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
          break
```

```
     end
     if Pen > Beam
          Pen = Beam;
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;
```

```
          % Calculate the delta-KE from the Minorsky relation;
          Emin = dKE*10^6*dRt;

          % The corresponding force is
          Fmin = Emin/abs(ddepth);

 % Ship 1 (Struck ship)

 %    The acceleration of translation from membrane force in
 the global X coordinate is:
               aX1mem = 0;

 %    The acceleration of translation from membrane force in
 the global Y coordinate is:
               aY1mem = 0;

 %    Calculate the angular acceleration about the ship c.g.:
      % the current contact point is:
               CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
 L)*sin(omega1)];
      % so the arm that the force acts through is:
               arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
      % so the angular acceleration due to membrane force is:
               omega_dotdot1mem = 0;

 %  Calculate accelerations due to the Minorsky interaction;
 %  The Minorsky force is assumed to act in the direction
 opposite of relative motion.
 %  Since this force is in a different direction we must
 recalculate the virtual mass
 %          First, calculate the angle of resultant force
 compared to the ship principal axis
            zeta1 = zeta - omega1;         % the angle of
 Minorsky force to the struck ship.....
 %          Now, calculate the added mass matrix based on this
 angle
            A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
 ((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
 a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
 a22*cos(zeta1)^2)];
 %          Combine with physical mass to get the virtual mass
 matrix
            VM1 = M1 + A1;
 % Now calculate the accelerations due to the Minorsky force:
      aX1min = Fmin*cos(zeta)/VM1(1,1);
      aY1min = Fmin*sin(zeta)/VM1(2,2);
      omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
 omega1)/J66V1;

 % Sum the accelerations due to membrane and Minorsky for the
 total accleration due
 % due to relative motion and interaction for this time step
      aX1 =  aX1mem + aX1min;
      aY1 =  aY1mem + aY1min;
```

```
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2

%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;

%    The acceleration of translation due to membrane force
in the global Y coordinate is:
          aY2mem = 0;

%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;       % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
```

```
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%  Determine which tanks have been breached
     doublecargo

end
```

% double5.m

```
% Script to perform time-domain analysis for single-hulled
tanker collision.
% The "5" indicates that this script is for phase 5 of the
collision, which is
% from the time of impact on the inner cargo bulkhead,
until that
% membrane ruptures.

% Input:  all the dynamic variables from double4.m
%         dt        from energy.m
%         V1        from energy.m
%         V2        from energy.m
%         VM1       from energy.m
%         VM2       from energy.m
%         alpha     from vargen.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

% Determine nearest transverse structures on the shell, and
distance to
% each for use in applying Van Mater's extension to Jones
method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
a = BH(j) - L;
if j > 1
     b = L - BH(j-1);
     else
          b = 0;
end
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;
```

```
% Reset deflection counter to zero for this new membrane:
defl = 0;

%%%%%%%% Begin time-step routine  %%%%%%%%
while abs(defl) < abs(defl_lim)
if relvel < endvel
     break
end
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
     if L < 0
          L=0;
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
   end
   if Pen < 0
     break
   end

% Calculate relative translation, penetration, and change in
impact point in this time step
     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);
```

```
% Calculate the membrane deflection
     defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
     if defl < 0
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
     else
          break
     end
     if defl < defl_lim
          defl = defl_lim;
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
          rupture3 = 1;
     end
%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.

% Ship 1 (Struck ship)
%    Calculate the virtual mass
```

```
%          First, calculate the angle of resultant force
compared to the ship principal axis
           zeta1 = pi/2;        % the membrane force is
always normal to the struck ship.....
%          Now, calculate the added mass matrix based on this
angle
           A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%          Combine with physical mass to get the virtual mass
matrix
           VM1 = M1 + A1;

%    The acceleration of translation in the global X
coordinate is:
           aX1mem = Fmem*sin(omega1)/VM1(1,1);

%    The acceleration of translation in the global Y
coordinate is:
           aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
           CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
           arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration is:
           omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%          First, calculate the angle of resultant force
compared to the ship principal axis
           zeta1 = zeta - omega1;       % the angle of
Minorsky force to the struck ship.....
%          Now, calculate the added mass matrix based on this
angle
           A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%          Combine with physical mass to get the virtual mass
matrix
           VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
```

```
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:

%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];

%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;

%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);

%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
```

```
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%        Combine with physical mass to get the virtual mass
matrix
         VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%  Determine which tanks have been breached
     doublecargo


end
```

% double6.m

```
% Script to perform time-domain analysis for double-hulled
tanker collision.
% The "6" indicates that this script is for phase 6 of the
collision, which is
% from the time the inner cargo bulkhead ruptures until the
collision ends.

% Input:  all the dynamic variables from double5.m
%         dt        from write.m
%         V1        from write.m
%         V2        from write.m
%         VM1       from
%         VM2       from
%         alpha     from write.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

%%%%%%%% Begin time-step routine  %%%%%%%%%
while relvel < endvel

Eabs = Eabs + Emin;
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
          break
     end
     if Pen > Beam
          Pen = Beam;
```

```
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;
```

```
     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Ship 1 (Struck ship)

%    The acceleration of translation from membrane force in
the global X coordinate is:
          aX1mem = 0;

%    The acceleration of translation from membrane force in
the global Y coordinate is:
          aY1mem = 0;

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration due to membrane force is:
          omega_dotdot1mem = 0;

%  Calculate accelerations due to the Minorsky interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;       % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2
```

```
%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;
%    The acceleration of translation due to membrane force
in the global Y coordinate is:
          aY2mem = 0;
%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;
%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;
%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;
%  Determine which tanks have been breached
     doublecargo
end
```

%   doublecargo.m

```
% the script to determine which cargo bulkheads have been
breached during
% each time step in the simulation for the double hull
tanker

% Date created:  3/10/98
% Last revision:   3/17/98

% Inputs:   L and Pen from the collision phase scripts

% Output:
%    Flags corresponding to the cargo compartments that will
release oil


j = 1;
     while L > DBH(j)
          j = j+1;
     end   % at this end, BH(j) is the first bulkhead aft of
the damage location

     if j == 2;
          if rupture2 > 0
               Tank1P = 1;
               if Pen >=24
                    Tank1S =1;
               end
          end
     end
     if j == 3;
          if rupture2 > 0
          Tank2P = 1;
               if rupture3 > 0
                    Tank2C = 1;
               end
          end
     end
     if j == 4;
          if rupture2 > 0
          Tank3P = 1;
               if rupture3 > 0
                    Tank3C = 1;
               end
          end
     end
     if j == 5;
          if rupture3 > 0
               Tank4C = 1;
```

```
        end
end
if j == 6;
        if rupture2 > 0
                Tank5P = 1;
                if rupture3 > 0
                        Tank5C = 1;
                end
        end
end
if j == 7;
        if rupture2 > 0
                Tank6P = 1;
                if rupture3 > 0
                        Tank6C = 1;
                end
        end
end
if j == 8;
        if rupture2 > 0
                SlopTk = 1;
                if rupture3 > 0
                        Tank6C = 1;
                end
        end
end
```

% iotd1.m

```
% Script to perform time-domain analysis for intermediate
oil-tight deck tanker collision.
% The "1" indicates that this script is for phase 1 of the
collision, which is
% from the time of impact until the shell membrane ruptures.

% Input:  dt         from energy.m
%         V1         from energy.m
%         V2         from energy.m
%         VM1        from energy.m
%         VM2        from energy.m
%         alpha      from vargen.m


% Output: Generated variates and values

% Date created: 3/2/98
% Last updated: 4/23/98

% Reset flags for tank breaching to zero:
Tank1P = 0;
Tank1S = 0;
Tank1BOT = 0;
Tank2P = 0;
Tank2S = 0;
Tank2BOT = 0;
Tank3P = 0;
Tank3S = 0;
Tank3BOT = 0;
Tank4P = 0;
Tank4S = 0;
Tank4BOT = 0;
Tank5P = 0;
Tank5S = 0;
Tank5BOT = 0;
Tank6P = 0;
Tank6S = 0;
Tank6BOT = 0;
SlopTk = 0;

% Determine nearest transverse structures, and distance to
each for use in applying
% Van Mater's extension to Jones method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
```

```
a = BH(j) - L;
if j > 1
     b = L - BH(j-1);
     else
          b = 0;
end
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;

% Initialize new variables (Subscript 1 = struck ship,
subscript 2 = striking ship)
time = 0;
X1=0;
Y1=0;
X2= (LBP/2)-L+(LBP2/2)*cos(phi*pi/180);
Y2= (Beam/2)+(LBP2/2)*sin(phi*pi/180);
omega1=0;
omega_dot1 = 0;
omega2=(phi+180)*(pi/180);
omega_dot2 = 0;
defl = 0;
T1 = Vel1;
T2 = Vel2;
rupture = 0;
rupture2 = 0;
rupture3 = 0;
s=0;
ddepth = 0;
depth = 0;
Fmin = 0;
Emin = 0;
Eabs = 0;
Emem_last = 0;
dL = 0;
Pen = 0;
dPen = 0;
Pmax = 0;
%%%%%%%% Begin time-step routine  %%%%%%%%%
while abs(defl) < abs(defl_lim)
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
```

```
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
     if L < 0
          L=0;
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
   end
   if Pen < 0
     break
   end

% Calculate relative translation, penetration, and change in
impact point in this time step
     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

% Calculate the membrane deflection
     defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
     if defl < 0
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
     else
          break
     end
     if defl < defl_lim
          defl = defl_lim;
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
          rupture = 1;
```

```
     end

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =   (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.

% Ship 1 (Struck ship)
%    Calculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = pi/2;        % the membrane force is
always normal to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
```

```
%    The acceleration of translation in the global X
coordinate is:
          aX1mem = Fmem*sin(omega1)/VM1(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration is:
          omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:
```

```
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];

%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;

%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);

%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

% Now calculate accelerations due to the Minorsky
interaction;
% The Minorsky force is assumed to act in the direction
opposite of relative motion.
% Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;         % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
```

```
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%    Determine which tanks have been breached
     iotdcargo

end
```

% iotd2.m

```
% Script to perform time-domain analysis for IOTD tanker
collision.
% The "2" indicates that this script is for phase 2 of the
collision, which is
% from the time the outer shell membrane ruptures until the
inner bulkhead is contacted.

% Input:  all the dynamic variables from iotd1.m
%         dt          from write.m
%         V1          from write.m
%         V2          from write.m
%         VM1         from
%         VM2         from
%         alpha       from write.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

%%%%%%%% Begin time-step routine  %%%%%%%%
while Pen < IOTDS1
if relvel < endvel
     break
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            Ship 1
%----------------------------------------------------
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
```

```
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;
```

```
     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Ship 1 (Struck ship)

%    The acceleration of translation from membrane force in
the global X coordinate is:
          aX1mem = 0;

%    The acceleration of translation from membrane force in
the global Y coordinate is:
          aY1mem = 0;

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration due to membrane force is:
          omega_dotdot1mem = 0;

%  Calculate accelerations due to the Minorsky interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
```

```
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2

%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;

%    The acceleration of translation due to membrane force
in the global Y coordinate is:
          aY2mem = 0;

%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
```

```
    omega_dot1 = omega_dot1 + omega1_dotdot*step;
    omega_dot2 = omega_dot2 + omega2_dotdot*step;
    relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
    Eabs = Eabs + Emin + Emem - Emem_last;
    Emem_last = Emem;

%   Determine which tanks have been breached
    iotdcargo

end
```

% iotd3.m

```
% Script to perform time-domain analysis for IOTD tanker
collision.
% The "3" indicates that this script is for phase 3 of the
collision, which is
% from the time of impact on the inner shell, or cargo
boundary until that
% membrane ruptures.

% Input:  all the dynamic variables from iotd2.m
%         dt        from energy.m
%         V1        from energy.m
%         V2        from energy.m
%         VM1       from energy.m
%         VM2       from energy.m
%         alpha     from vargen.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

% Determine nearest transverse structures on the inner
shell, and distance to
% each for use in applying Van Mater's extension to Jones
method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
a = BH(j) - L;
if j > 1
     b = L - BH(j-1);
     else
          b = 0;
end
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;
```

```
% Reset deflection counter to zero for this new membrane:
defl = 0;

%%%%%%%% Begin time-step routine  %%%%%%%%%
while abs(defl) < abs(defl_lim)
if relvel < endvel
     break
end
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
     if L < 0
          L=0;
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
   end
   if Pen < 0
     break
   end

% Calculate relative translation, penetration, and change in
impact point in this time step
     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);
```

```
% Calculate the membrane deflection
     defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
     if defl < 0
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
     else
          break
     end
     if defl < defl_lim
          defl = defl_lim;
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
          rupture2 = 1;
     end
%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.

% Ship 1 (Struck ship)
%    Calculate the virtual mass
```

```
%          First, calculate the angle of resultant force
compared to the ship principal axis
           zeta1 = pi/2;        % the membrane force is
always normal to the struck ship.....
%          Now, calculate the added mass matrix based on this
angle
           A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%          Combine with physical mass to get the virtual mass
matrix
           VM1 = M1 + A1;

%     The acceleration of translation in the global X
coordinate is:
           aX1mem = Fmem*sin(omega1)/VM1(1,1);

%     The acceleration of translation in the global Y
coordinate is:
           aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%     Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
           CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
           arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration is:
           omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%          First, calculate the angle of resultant force
compared to the ship principal axis
           zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%          Now, calculate the added mass matrix based on this
angle
           A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%          Combine with physical mass to get the virtual mass
matrix
           VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
```

```
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:

%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];

%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;

%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);

%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;         % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
```

```
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%        Combine with physical mass to get the virtual mass
matrix
         VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%    Determine which tanks have been breached
     iotdcargo


end
```

% iotd4.m

```
% Script to perform time-domain analysis for IOTD tanker
collision.
% The "4" indicates that this script is for phase 4 of the
collision, which is
% from the time the inner shell membrane ruptures until the
inner cargo bulkhead is
% contacted.

% Input:  all the dynamic variables from iotd3.m
%         dt         from write.m
%         V1         from write.m
%         V2         from write.m
%         VM1        from
%         VM2        from
%         alpha      from write.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

%%%%%%%% Begin time-step routine  %%%%%%%%
while Pen < IOTDS2
if relvel < endvel
     break
end
Eabs = Eabs + Emin;
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
          break
```

```
     end
     if Pen > Beam
          Pen = Beam;
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;
```

```
     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Ship 1 (Struck ship)

%    The acceleration of translation from membrane force in
the global X coordinate is:
          aX1mem = 0;

%    The acceleration of translation from membrane force in
the global Y coordinate is:
          aY1mem = 0;

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration due to membrane force is:
          omega_dotdot1mem = 0;

%  Calculate accelerations due to the Minorsky interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
```

```
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2

%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;

%    The acceleration of translation due to membrane force
in the global Y coordinate is:
          aY2mem = 0;

%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;       % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
```

```
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%    Determine which tanks have been breached
     iotdcargo

end
```

% iotd5.m

```
% Script to perform time-domain analysis for IOTD tanker
collision.
% The "5" indicates that this script is for phase 5 of the
collision, which is
% from the time of impact on the centerline bulkhead,  until
that
% membrane ruptures.

% Input:  all the dynamic variables from iotd4.m
%         dt        from energy.m
%         V1        from energy.m
%         V2        from energy.m
%         VM1       from energy.m
%         VM2       from energy.m
%         alpha     from vargen.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

% Determine nearest transverse structures on the shell, and
distance to
% each for use in applying Van Mater's extension to Jones
method....

j = 1;
     while BH(j) < L
          j = j+1;
     end
a = BH(j) - L;
if j > 1
     b = L - BH(j-1);
     else
          b = 0;
end
% Ensure variable "a" represents the "short leg" of the
strained plate
if   a>b
     c=a;
     a=b;
     b=c;
end
% Calculate deflection at which plate fails, per Van Mater's
extension to Jones
     defl_lim= -0.452*a;
```

```
% Reset deflection counter to zero for this new membrane:
defl = 0;

%%%%%%%% Begin time-step routine  %%%%%%%%%
while abs(defl) < abs(defl_lim)
if relvel < endvel
     break
end
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
  if Pen > Pmax
   Pmax = Pen;
  end
     if L > LBP
          L=LBP;
          break
     end
     if L < 0
          L=0;
          break
     end
     if Pen > Beam
          Pen = Beam;
          break
   end
   if Pen < 0
     break
   end

% Calculate relative translation, penetration, and change in
impact point in this time step
     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);
```

```
% Calculate the membrane deflection
     defl = defl + reltrans(2)+((LBP/2)-L)*sin(omega1);

% Calculate the resistance force of the membrane
     if defl < 0
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
     else
          break
     end
     if defl < defl_lim
          defl = defl_lim;
          Emem = sigma_y*t_plate*breadth*defl^2/spacing;
          Fmem = Emem/abs(defl);
          rupture3 = 1;
     end
%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  Should be modified so
that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;

     % The corresponding force is
     Fmin = Emin/abs(ddepth);

% Calculate resulting accelerations from the membrane force
% For Phase I, the membrane force is assumed to act
perpendicularly to the hull surface
%  of the struck ship, and the Minorsky force acts to oppose
the direction of relative motion.

% Ship 1 (Struck ship)
%    Calculate the virtual mass
```

```
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = pi/2;        % the membrane force is
always normal to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;

%    The acceleration of translation in the global X
coordinate is:
          aX1mem = Fmem*sin(omega1)/VM1(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY1mem = -Fmem*cos(omega1)/VM1(2,2);

%    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration is:
          omega_dotdot1mem = -((0.5-Lnd)/abs(0.5-
Lnd))*Fmem*arm/J66V1;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
```

```
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;
% Ship 2
%    Calculate the virtual mass for the membrane force
acceleration:

%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = omega2 - (omega1 + pi/2);

%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];

%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;

%    The acceleration of translation in the global X
coordinate is:
          aX2mem = -Fmem*sin(omega1)/VM2(1,1);

%    The acceleration of translation in the global Y
coordinate is:
          aY2mem = Fmem*cos(omega1)/VM2(2,2);

%    The angular acceleration about the ship c.g. is:
          omega2_dotdotmem = (-Fmem*sin(omega1-omega2-
pi/2)*(LBP2/2))/J66V2;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
```

```
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
     Emem_last = Emem;

%    Determine which tanks have been breached
     iotdcargo

end
```

% iotd6.m

```
% Script to perform time-domain analysis for IOTD tanker
collision.
% The "6" indicates that this script is for phase 6 of the
collision, which is
% from the time the centerline bulkhead ruptures until the
collision ends.

% Input:  all the dynamic variables from double5.m
%         dt        from write.m
%         V1        from write.m
%         V2        from write.m
%         VM1       from
%         VM2       from
%         alpha     from write.m


% Output: Generated variates and values

% Date created: 3/10/98
% Last updated: 4/23/98

%%%%%%%% Begin time-step routine  %%%%%%%%%
while relvel < endvel

Eabs = Eabs + Emin;
% Calculate new postions and rotations at end of time step
time = time + step;
X1 = X1 + T1(1)*step;
Y1 = Y1 + T1(2)*step;
X2 = X2 + T2(1)*step;
Y2 = Y2 + T2(2)*step;
omega1 = omega1 + omega_dot1*step;
omega2 = omega2 + omega_dot2*step;
L = L + dL;
Pen = Pen + dPen;
     if Pen > Pmax
          Pmax = Pen;
     end
     if L < 0
               L=0;
               break
     end
     if L > LBP
          L=LBP;
          break
     end
     if Pen > Beam
          Pen = Beam;
```

```
          break
     end
     if Pen < 0
               break
     end

% Calculate relative translation, penetration, and change in
impact point in this time step
% S1 is the total velocity (from linear and rotational
motion) of the impact point on Ship 1.
% S2 is the same velocity for Ship 2.

     S1 = [T1(1)+((LBP/2)-L)*omega_dot1*sin(omega1),
T1(2)+((LBP/2)-L)*omega_dot1*cos(omega1)];
     S2 = [T2(1) - (LBP2/2)*omega_dot2*sin(omega2),
T2(2)+(LBP2/2)*omega_dot2*cos(omega2)];
     reltrans = (S2-S1)*step;

% Calculate direction of relative translation
     zeta = atan2(reltrans(2),reltrans(1));

% Calculate penetration and change in location
     dPen = sqrt((reltrans(1))^2 +
(reltrans(2))^2)*cos(omega1 + (3*pi/2) - zeta);
     dL = sqrt((reltrans(1))^2 + (reltrans(2))^2)*sin(omega1
+ (3*pi/2) - zeta);

%  Calculate the force resulting from the "Minorsky
mechanism"
     % Rtt is the total "resistance factor"
     % dRt is the differential resistance factor for this
time step
     % depth is the distance of penetration
     % ddepth is the differential distance of penetration
     % t is the aggregate in-plane structure thickness
     % alpha is the bow half-entrance angle
     % This formula accounts for the triangular bow wedge
geometry
     % and dynamic collision angle, but places no
     % limits on striking ship beam.  In the future, could
be modified so that if width exceeds beam,
     % remaining area is rectangular....

     ddepth = sqrt (reltrans(1)^2 + reltrans(2)^2);
     Rtt = (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))));
     depth = depth + ddepth;
     dRt =  (depth^2)*t*tan(alpha*pi/180)/(1-
((tan(alpha*pi/180))^2/((tan((omega1-omega2)*pi/180)^2))))+
abs(dL)*Pen - Rtt;

     % Calculate the delta-KE from the Minorsky relation;
     Emin = dKE*10^6*dRt;
```

```
     % The corresponding force is
     Fmin = Emin/abs(ddepth);

 % Ship 1 (Struck ship)

 %    The acceleration of translation from membrane force in
 the global X coordinate is:
          aX1mem = 0;

 %    The acceleration of translation from membrane force in
 the global Y coordinate is:
          aY1mem = 0;

 %    Calculate the angular acceleration about the ship c.g.:
     % the current contact point is:
          CP = [X2 - ((LBP/2)*cos(omega2)), ((LBP/2)-
L)*sin(omega1)];
     % so the arm that the force acts through is:
          arm = sqrt((X1-CP(1))^2 + (Y1 - CP(2))^2);
     % so the angular acceleration due to membrane force is:
          omega_dotdot1mem = 0;

%  Calculate accelerations due to the Minorsky interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta1 = zeta - omega1;        % the angle of
Minorsky force to the struck ship.....
%         Now, calculate the added mass matrix based on this
angle
          A1 = m1*[(a11*sin(zeta1)^2 + a22*cos(zeta1)^2),
((a11-a22)*sin(zeta1)*cos(zeta1)); ((a11-
a22)*sin(zeta1)*cos(zeta1)), (a11*sin(zeta1)^2 +
a22*cos(zeta1)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM1 = M1 + A1;
% Now calculate the accelerations due to the Minorsky force:
     aX1min = Fmin*cos(zeta)/VM1(1,1);
     aY1min = Fmin*sin(zeta)/VM1(2,2);
     omega1_dotdotmin = -((LBP/2)-L)*Fmin*sin(zeta - pi -
omega1)/J66V1;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX1 =  aX1mem + aX1min;
     aY1 =  aY1mem + aY1min;
     omega1_dotdot = omega_dotdot1mem + omega1_dotdotmin;

% Ship 2
```

```
%    The acceleration of translation due to membrane force
in the global X coordinate is:
          aX2mem = 0;

%    The acceleration of translation due to membrane force
in the global Y coordinate is:
          aY2mem = 0;

%    The angular acceleration about the ship c.g. due to
membrane force is:
          omega2_dotdotmem = 0;

%  Now calculate accelerations due to the Minorsky
interaction;
%  The Minorsky force is assumed to act in the direction
opposite of relative motion.
%  Since this force is in a different direction we must
recalculate the virtual mass
%         First, calculate the angle of resultant force
compared to the ship principal axis
          zeta2 = pi - omega2 + zeta;        % the angle of
Minorsky force to the striking ship.....
%         Now, calculate the added mass matrix based on this
angle
          A2 = m2*[(a11*sin(zeta2)^2 + a22*cos(zeta2)^2),
((a11-a22)*sin(zeta2)*cos(zeta2)); ((a11-
a22)*sin(zeta2)*cos(zeta2)), (a11*sin(zeta2)^2 +
a22*cos(zeta2)^2)];
%         Combine with physical mass to get the virtual mass
matrix
          VM2 = M2 + A2;
% Now calculate the accelerations due to the Minorsky force:
     aX2min = Fmin*cos(zeta-pi)/VM2(1,1);
     aY2min = Fmin*sin(zeta-pi)/VM2(2,2);
     omega2_dotdotmin = (LBP2/2)*Fmin*sin(omega2-
zeta)/J66V2;

% Sum the accelerations due to membrane and Minorsky for the
total accleration due
% due to relative motion and interaction for this time step
     aX2 =  aX2mem + aX2min;
     aY2 =  aY2mem + aY2min;
     omega2_dotdot = omega2_dotdotmem + omega2_dotdotmin;

%    Calculate new velocities
     T1(1) = T1(1) + aX1*step;
     T1(2) = T1(2) + aY1*step;
     T2(1) = T2(1) + aX2*step;
     T2(2) = T2(2) + aY2*step;
     omega_dot1 = omega_dot1 + omega1_dotdot*step;
     omega_dot2 = omega_dot2 + omega2_dotdot*step;
     relvel = sqrt(reltrans(1)^2 + reltrans(2)^2)/step;
     Eabs = Eabs + Emin + Emem - Emem_last;
```

```
     Emem_last = Emem;

%    Determine which tanks have been breached
     iotdcargo

end
```

% iotdcargo.m

```
% the script to determine which cargo bulkheads have been
breached during
% each time step in the simulation for the mid-deck tanker

% Date created:  3/10/98
% Last revision:   3/17/98

% Inputs:   L and Pen from the collision phase scripts

% Output:
%    Flags corresponding to the cargo compartments that will
release oil


j = 1;
     while L > IBH(j)
          j = j+1;
     end   % at this end, BH(j) is the first bulkhead aft of
the damage location

     if j == 2;
          if rupture2 > 0
               Tank1P = 1;
               Tank1BOT = 1;
               if Pen >=24
                    Tank1S =1;
               end
          end
     end
     if j == 3;
          if rupture2 > 0
               Tank2P = 1;
               Tank2BOT = 1;
               if Pen >=24
                    Tank2S =1;
               end
          end
     end
     if j == 4;
          if rupture2 > 0
               Tank3P = 1;
               Tank3BOT = 1;
               if Pen >=24
                    Tank3S =1;
               end
          end
     end
     if j == 5;
```

```
          if rupture2 > 0
               Tank4P = 1;
               Tank4BOT = 1;
               if Pen >=24
                    Tank4S =1;
               end
          end
     end
     if j == 6;
          if rupture2 > 0
               Tank5P = 1;
               Tank5BOT = 1;
               if Pen >=24
                    Tank5S =1;
               end
          end
     end
     if j == 7;
          if rupture2 > 0
               Tank6P = 1;
               Tank6BOT = 1;
               if Pen >=24
                    Tank6S =1;
               end
          end
     end
     if j == 8;
          if rupture2 > 0
               SlopTk = 1;
               Tank6BOT = 1;
               if Pen >=24
                    SlopTk = 2;
               end
          end
     end
```

% howmany.m

```
% script to check for integration error cases, and remove
them from the population
numberzeroed = 0;
for q = 1:n;
     if Time(q) < 0.01
        numberzeroed = numberzeroed +1;
     end
end
numberzeroed
```

% calibrate.m

```
%    This script is used in the input pdf calibration
process.
%    The first step is to remove cases where no outflow
occurs from the sample
%    population.  Next an error function based on a "least
squares" analysis is calculated.

%    Date created:  2/22/98
%    Last revision:   3/27/98

%    Inputs:   Result vectors from write.m
%    Output:   Result vectors with non-rupture cases removed
%              Value of error function for this set of
simulations
%              Plots of output with zero outflow cases
removed

%    Remove non-rupture cases from sample population

%         First, zero out the penetration, length of damage
and center of damage
%         elements from the cases where no hull rupture
occurred:

for i = 1:n
     if R(i) == 0
          P(i) = 0;
          Len(i) = 0;
          Cen(i) = 0;
          bow_alpha(i) = 0;
          CA(i) = 0;
          ICP(i) =0;
     end
end

%    Now, remove those cases from the population using the
"nonzeros" function

     Len = nonzeros(Len);
     P = nonzeros(P);
     Cen = nonzeros(Cen);
     bow_alpha = nonzeros(bow_alpha);
     CA = nonzeros(CA);
     ICP = nonzeros(ICP);
%    Record the length of the resulting vectors for future
use in the error function

     s = min(length(Len), length(P));
```

% write.m

```
% Routine to save variates and calculated results for later
analysis.
% Input: none

% Output: Generated variates and values
%  Variable label   Variable description
%        V1                   Struck ship speed (kt) [x-vel,
y-vel]
%        V2                   Striking ship speed (kt) [x-
vel, y-vel]
%        alpha                Striking ship half-entrance
angle (degrees)
%        theta                Collision angle (degrees
relative, from struck ship)
%        L         Location of collision point on struck
ship(meters from FP)
%        E                    Energy absorbed by structure
%        T                    Time for collision to occur
%        dt                   time step for time domain
analysis
%        R                    flag set to "1" if hull
rupture occurs
%        Pen                  depth of penetration during
collision
%        Len                  Length of collision damage

% Date created: 11/3/97
% Last updated: 2/16/98
 if time <= 45
     E(i)=Ea;
     Eshell(i)=Emem;
     V1(i)=U1;
     V2(i)=U2;
     bow_alpha(i) = alpha;
     if phi < 90
          LOC(1)=max(0,Lo-((Beam*cos(alpha')/2)));
          else LOC(1)=min(LBP,Lo+((Beam*cos(alpha)/2)));
     end
     LOC(2) = L;
     ICP(i) = Lo/LBP;
     FCP(i) = L/LBP;
     Len(i) = abs((LOC(1)/LBP)-FCP(i));
     R(i) = rupture;  % flag set to 1 if shell rupture
occurs
     CA(i) = phi;
     P(i) = Pmax;
     Cen(i) = (ICP(i)+FCP(i))/2;
     Minorsky(i) = dKE;
```

```
     Time(i) = time;
   Outflow(i) = oil;
   Mass(i) = m1;
end
```

% howmany.m

```
% script to check for integration error cases, and remove
them from the population
numberzeroed = 0;
for q = 1:n;
     if Time(q) < 0.01
        numberzeroed = numberzeroed +1;
     end
end
numberzeroed
```

% calibrate.m

```
%    This script is used in the input pdf calibration
process.
%    The first step is to remove cases where no outflow
occurs from the sample
%    population.  Next an error function based on a "least
squares" analysis is calculated.

%    Date created:  2/22/98
%    Last revision:   3/27/98

%    Inputs:   Result vectors from write.m
%    Output:   Result vectors with non-rupture cases removed
%              Value of error function for this set of
simulations
%              Plots of output with zero outflow cases
removed

%    Remove non-rupture cases from sample population

%         First, zero out the penetration, length of damage
and center of damage
%         elements from the cases where no hull rupture
occurred:

for i = 1:n
     if R(i) == 0
          P(i) = 0;
          Len(i) = 0;
          Cen(i) = 0;
          bow_alpha(i) = 0;
          CA(i) = 0;
          ICP(i) =0;
     end
end

%    Now, remove those cases from the population using the
"nonzeros" function

     Len = nonzeros(Len);
     P = nonzeros(P);
     Cen = nonzeros(Cen);
     bow_alpha = nonzeros(bow_alpha);
     CA = nonzeros(CA);
     ICP = nonzeros(ICP);
%    Record the length of the resulting vectors for future
use in the error function

     s = min(length(Len), length(P));
```

```
%    Record the simulation population in each bin

     % set the number of bins
     nbp = [0.05:0.05:1];
     nbl = [0.05:0.05:1];
     nbc =  [0.05:0.1:1];
     nba = [5:10:175];
     nbba = [0:5:90];
     nbout = [0:0.01:0.5];

     % record the populations
     [p,x] = hist(P,nbp);
     [l,q] = hist(Len,nbl);
     [c,z] = hist(Cen,nbc);
     [collang,d] = hist(CA,nba);
     [bowang,e] = hist(bow_alpha,nbba);
     [collpt,f] = hist(ICP,nbl);
     [outflw,g] = hist(Outflow,nbout);

%    Convert the bin populations to probability density
functions

     p = p/(s*0.05);
     l = l/(s*0.05);
     c = c/(s*0.1);
     collang = collang/(s*10);
     bowang = bowang/(s*5);
     collpt = collpt/(s*0.05);
     outflw = outflw/(s*0.01);

%    Plot the resulting pdf's
      figure(2)
      clf
      colormap(white)
      hold on
      bar(x,p)
     x=[0,.05,.1,.15,.2,.25,.3,.3];
     y=[24.96,5,0.56,0.56,0.56,0.56,0.56,0];
     plot(x,y,'r.-')
     plot(nbp,p,'k*')
     axis([0 .4 0 30])
     legend('MARPOL Standard','Calculated Distribution',1)
     ylabel('Probability Density')
      xlabel('Transverse Penetration (Penetration/Beam)')
      title('Penetration of Transverse Damage PDF')

      figure(3)
      clf
      colormap(white)
     hold on
     bar(q,l)
     x=[0,.1,.2,.34,.34];
     y=[11.95,3.5,0.35,0.35,0];
```

```
     plot(x,y,'r.-')
     plot(nbl,l,'k*')
     axis([0 .4 0 20])
     ylabel('Probability Density')
     xlabel('Length of damaged section (Length/LBP)')
     legend('MARPOL Standard','Calculated Distribution',1)
     title('Longitudinal Length of Damage PDF')

      figure(4)
      clf
      colormap(white)
     hold on
     bar(z,c)
     x=[0,1.0,1.0];
     y=[1,1,0];
     plot(x,y,'r.-')
     plot(nbc,c,'k*')
     axis([0 1.06 0 1.5])
     xlabel('Longitudinal center of damaged section
(Loc/LBP)')
     ylabel('Probability Density')
     legend('MARPOL Standard','Calculated Distribution',1)
     title('Longitudinal Center of Damage PDF')
% This part is to plot the input pdf's and the generated
histograms....

     figure(5)
     colormap(white)
     clf
     x=[0,180,180];
     dist = [0.0056,0.0056,0];
     %sigma=30;
     %for i = 1:181
     %    dist1(i)= (1/sqrt(2*pi*sigma^2))*exp((-(i-
46)^2)/(2*sigma^2));
     %    dist2(i) = 3*(1/sqrt(2*pi*sigma^2))*exp((-(i-
136)^2)/(2*sigma^2));
     %    dist(i) = (dist1(i) + dist2(i))/4;
     %end
     hold on
     bar(d,collang)
     plot(x,dist, 'r.-')
     plot(nba,collang,'k*')
     axis tight
     ylabel('Probability Density')
     xlabel('Collision Angle (degrees relative)')
     legend('Target PDF','Actual Histogram',2)
     title('Collision Angle PDF')

     figure(6)
     colormap(white)
     clf
     dist = 0;
     x=[0:1:90];
```

```
      sigma=5;
      for i = 1:91
           dist(i)= (1/sqrt(2*pi*sigma^2))*exp((-(i-
 39)^2)/(2*sigma^2));
      end
      hold on
      bar(e,bowang)
      plot(x,dist, 'r.-')
      plot(nbba,bowang,'k*')
      axis tight
      ylabel('Probability Density')
      xlabel('Bow Half-entrance Angle (degrees)')
      legend('Target PDF','Actual Histogram',1)
      title('Striking Ship Bow Half-Entrance Angle PDF')

      figure(7)
      colormap(white)
      clf
      hold on
      bar(f,collpt)
      x=[0,1.02,1.02];
      dist = [1,1,0];
      plot(x,y,'r.-')
      axis([0 1.04 0 2])
      plot(nbl,collpt,'k*')
      ylabel('Probability Density')
      xlabel('Initial Collision Point (Loc/LBP)')
      legend('Target PDF','Actual Histogram',1)
      title('Initial Point of Contact in Collision PDF')

      figure(8)
      colormap(white)
      clf
      hold on
      bar(g,outflw)
      %axis([0 0.5 0 50])
      plot(nbout,outflw,'k*')
      ylabel('Probability Density')
      xlabel('Specific outflow (oil/cargo)')
      legend('Generated Histogram',1)
      title('Oil Outflow PDF')

%     Calculate the error functions

      EFP = (p(1)-14.98)^2 + (p(2)-2.78)^2 + (p(3) - 0.56)^2
+  (p(4) - 0.56)^2 + p(5)^2 + p(6)^2 + p(7)^2 + p(8)^2 +
p(9)^2 + p(10)^2;
      EFP = EFP + p(11)^2 +p(12)^2 +p(13)^2 +p(14)^2 +p(15)^2
+p(16)^2 +p(17)^2 +p(18)^2 +p(19)^2 +p(20)^2;
      EFP = 1-sqrt(EFP)/20;

      EFL = (l(1)-9.8375)^2 + (l(2)-5.6125)^2 + (l(3) -
2.7125)^2 +  (l(4)-1.1375)^2 + (l(5)-.35)^2 + (l(6)-0.35)^2
+ l(7)^2 + l(8)^2 + l(9)^2 + l(10)^2 + l(11)^2 + l(12)^2 +
```

```
l(13)^2 + l(14)^2 + l(15)^2 + l(16)^2 + l(17)^2 + l(18)^2 +
l(19)^2 + l(20)^2;
     EFL = 1-sqrt(EFL)/20;

     EFC = (c(1)-1)^2 + (c(2)-1)^2 + (c(3) - 1)^2 +  (c(4) -
1)^2  +  (c(5) - 1)^2 + (c(6) - 1)^2 + (c(7) - 1)^2 + (c(8)
- 1)^2 + (c(9) - 1)^2 + (c(10) - 1)^2;
     EFC = 1-sqrt(EFC)/10;

     AVE = (EFP+EFL+EFC)/3;
     Product = EFP*EFL;

     EFP, EFC, EFL, AVE, Product

% Calculate probability of zero outflow, and mean outflow
Pzero = 0;
for j=1:n
     if Outflow(j) == 0
          Pzero = Pzero +1;
     end
end
Pzero=Pzero/s
Mean = sum(Outflow)/s
```

% output.m

```
%    A script to generate all output graphics from a run

%    Date created:  2/28/98
%    Last revision:   3/27/98

%    Inputs:   Result vectors from write.m
%    Output:   Plots of input and result pdf's for this set
of simulations

%    Record the length of the resulting vectors for future
use

s = min(length(P), length(Len));


%    Record the simulation population in each bin

     % set the number of bins
     nbp = [0.05:0.05:1];      % number of bins for
penetration
     nbl = [0.025:0.05:.975];       % number of bins for
length
     nbc =  [0.05:0.1:1];      % number of bins for center of
damage
     nba = [5:10:175];         % number of bins for collision
angle
     nbba = [2.5:5:87.5];      % number of bins for bow angle
     nbout = [0:0.01:0.5];     % number of bins for oil
outflow

     % record the populations
     [p,x] = hist(P,nbp);
     [l,q] = hist(Len,nbl);
     [c,z] = hist(Cen,nbc);
     [collang,d] = hist(CA,nba);
     [bowang,e] = hist(bow_alpha,nbba);
     [collpt,f] = hist(ICP,nbl);
     [outflw,g] = hist(Outflow,nbout);

%    Convert the bin populations to probability density
functions

     p = p/(s*0.05);
     l = l/(s*0.05);
     c = c/(s*0.1);
     collang = collang/(s*10);
     bowang = bowang/(s*5);
     collpt = collpt/(s*0.05);
     outflw = outflw/(s*0.01);

%    Plot the resulting pdf's
```

```
      figure(2)
      clf
      colormap(white)
      hold on
      bar(x,p)
     x=[0,.05,.1,.15,.2,.25,.32,.32];
     y=[24.96,5,0.56,0.56,0.56,0.56,0.56,0];
     plot(x,y,'r.-')
     plot(nbp,p,'k*')
     axis([0 0.5 0 25])
     legend('MARPOL Standard','Calculated Distribution')
     ylabel('Probability Density')
      xlabel('Transverse Penetration (Penetration/Beam)')

     figure(3)
     clf
     colormap(white)
     hold on
     bar(q,l)
     x=[0,.1,.2,.3,.3];
     y=[11.95,3.5,0.3,0.3,0];
     plot(x,y,'r.-')
     plot(nbl,l,'k*')
     axis([0 .5 0 12])
     ylabel('Probability Density')
     xlabel('Length of damaged section (Length/LBP)')
     legend('MARPOL Standard','Calculated Distribution',1)

     figure(4)
     clf
     colormap(white)
     hold on
     bar(z,c)
     x=[0,1.0,1.0];
     y=[1,1,0];
     plot(nbc,c,'k*')
     plot(x,y,'r.-')
     axis([0 1.02 0 1.5])
     axis tight
     xlabel('Longitudinal center of damaged section
(Loc/LBP)')
     ylabel('Probability Density')
     legend('Calculated Distribution','MARPOL Standard',1)

%  This part is to plot the input pdf's and the generated
histograms....

     figure(5)
     colormap(white)
     clf
     x=[0,180,180];
     dist = [0.0056,0.0056,0];
     %for i = 1:181
```

```
    %    dist1(i)= (1/sqrt(2*pi*sigma^2))*exp((-(i-
46)^2)/(2*sigma^2));
    %    dist2(i) = 3*(1/sqrt(2*pi*sigma^2))*exp((-(i-
136)^2)/(2*sigma^2));
    %    dist(i) = (dist1(i) + dist2(i))/4;
    %end
    hold on
    bar(d,collang)
    plot(x,dist, 'r.-')
    plot(nba,collang,'k*')
    axis tight
    ylabel('Probability Density')
    xlabel('Collision Angle (degrees relative)')
    legend('Target PDF','Actual Histogram',2)

    figure(6)
    colormap(white)
    clf
    dist = 0;
    x=[0:1:90];
    sigma=5;
    for i = 1:91
        dist(i)= (1/sqrt(2*pi*sigma^2))*exp((-(i-
39)^2)/(2*sigma^2));
    end
    hold on
    bar(e,bowang)
    plot(x,dist, 'r.-')
    plot(nbba,bowang,'k*')
    axis tight
    ylabel('Probability Density')
    xlabel('Bow Half-entrance Angle (degrees)')
    legend('Target PDF','Actual Histogram',1)

    figure(7)
    colormap(white)
    clf
    hold on
    bar(f,collpt)
    x=[0,1.0,1.0];
    dist = [1,1,0];
    plot(x,y,'r.-')
    axis([0.0 1.0 0 1.5])
    axis tight
    plot(nbl,collpt,'k*')
    ylabel('Probability Density')
    xlabel('Initial Collision Point (Loc/LBP)')
    legend('Target PDF','Actual Histogram',0)

    figure(8)
    colormap(white)
    clf
    hold on
    bar(g,outflw)
```

```
     %axis([0 0.5 0 50])
     plot(nbout,outflw,'k*')
     axis([-0.01 .5 0 20])
     axis tight
     ylabel('Probability Density')
     xlabel('Specific outflow (oil/cargo)')
     legend('Generated Probability Density',1)
     title('Oil Outflow PDF')

% Calculate probability of zero outflow, and mean outflow
Pzero = 0;
for j=1:n
     if Outflow(j) == 0
          Pzero = Pzero +1;
     end
end
Pzero=Pzero/s
Mean = sum(Outflow)/s
```

% joint.m

```
%    A script to generate joint probability function
graphics from a run

%    Date created:  3/27/98
%    Last revision:   3/27/98

%    Inputs:   Result vectors from write.m
%    Output:   Plots joint pdf's for this set of simulations
Z = zeros(20,20);   % sets up space for the joint pdf matrix
values

% for all of the length and penetration pairs
for i = 1:s

% Start at the first bin, and find the appropriate length
and penetration bins
     k = 1;
     m = 1;
     while Len(i) > nbl(k)
          k = k+1;
     end
     while P(i) > nbp(m)
          m = m + 1;
     end

     % Add one to the current bin population
     Z(k,m) = Z(k,m) + 1;
end

% Reset bin values to produce a "sensible" plot
%for m =1:20
%    nbp(m) = nbp(m)-0.05;
%    nbl(m) = nbl(m)-0.05;
%end

% convert to a pdf
Z=Z/(s*0.05*0.05);

% and plot
figure(9)
colormap(jet)
clf
meshgrid(nbl,nbp);
mesh(nbl,nbp,Z)
view([1,1,1])
axis([0 .8 0 .8 0 120])
axis tight
ylabel('Longitudinal Extent')
```

```
zlabel('Probability Density')
xlabel('Transverse Extent')
title('Joint PDF for Transverse and Longitudinal Extents of
Damage')
colormap(lines)
brighten(-0.3)
```