FATIGUE LIFE OF BULK CARRIER SIDE SHELL FRAME LOWER TOES AS A FUNCTION OF SHIP LENGTH AND LOADING CONDITION

BRIAN JAMES CHEATER

Fatigue Life of Bulk Carrier Side Shell Frame Lower Toes as a Function of Ship Length and Loading Condition

By

©Brian James Cheater, B.Eng

A thesis submitted to the School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland
February 2000

St. John's Newfoundland Canada

Abstract

New methods and tools have led to optimized ship structure, but poorly designed details persist, leading to fatigue failure. The cost of optimizing all details is prohibitive. A way is needed to parametrically assess fatigue life. Fatigue is a function of many variables ranging from design and fabrication to operational issues and little work has been done to quantify the relative effects. This work focuses on a narrow problem: fatigue life of bulk carrier side shell frame lower bracket toes. The frame was reduced to a simple two dimensional beam, subject to heave induced inertial and buoyant loads. The ship was assumed to be operating in an ITTC wave spectrum. It was assumed that the speed of advance is zero; the ship was flexible and the seas were from ahead. Boundary conditions were modeled using spring supports and fixity parameters. Fatigue analysis was performed using a hotspot SN approach and Miner's Rule. Results show a rapid decrease in fatigue life for ships greater than 100m.

A loading parameter was defined to model the difference between homogeneous and alternate hold loading. Results indicate homogeneous hold loading is preferable for ships less than approximately 290m in length and alternate hold loading is preferable for ships greater than 290m in length. This is thought to be because the heave natural frequency tends to be higher than the wave spectrum peak for small ships. Increasing length and unit loading causes a lowering of the heave natural frquency to a value closer to the forcing frequency of the waves for larger ships.

Acknowledgements

I am grateful to Dr Haddara and Dr. Bose for their help and guidance in preparing this thesis.

I am also grateful to Lloyd's Register of Shipping for their financial support during this process and for allowing me to take pictures and extracts from their Technical Association Paper "On Bulk Carrier Safety" (Ferguson *et al.*, 1992).

Table of Contents

List of Figures

List of Tables

Nomenclature

B	ship breadth
C_b	block coefficient
C_{LOAD}	cargo loading coefficient (1.0 = homogeneous, 2.0= alternate hold loading)
D	ship depth
DB	elevation of double bottom above keel
DWT	ship deadweight
F_B	buoyant wave force acting on bottom
F_I	inertial wave force
$F_{HI\_TOE\_FIXITY}$	restraint coefficient for upper toe
$F_{LO\_TOE\_FIXITY}$	restraint coefficient for lower toe
Hop	elevation of hopper tank crown
H_s	significant wave height
I_c	calculate moment of inertia
I_R	Rule required moment of inertia
K	stress intensity factor
L	ship length
LWL	load waterline
N	number of cycles to failure
N_{BOTTOM_FLARE}	number of elements in bottom flare
N_{BOTTOM_TOE}	number of elements in bottom toe
N_{MID}	number of elements in midspan
N_{TOP_FLARE}	number of elements in top flare
N_{TOP_TOE}	number of elements in top toe
R	stress ratio
RAO_x	response amplitude operator for absolute ship movement
RAO_z	response amplitude operator for relative ship movement
S	stress range
$S(\omega)$	wave spectra
T	ship draft
Top	height of topside tank
$Topside$	width of topside tank

a	crack length
$a(\omega)$	amplitude of wave of frequency ω
a_{22}	added mass coefficients
b_{22}	damping coefficients
b_f	breadth of flange
d_w	depth of web

g	gravitational accleration
h	depth
y	weighted wave amplitude
k	stiffness or wave number
l_{ah}	length along hopper of lower bracket
l_{at}	length along topside of upper bracket
l_{vh}	length along shell of lower bracket
l_{vt}	length along shell of upper bracket
$m(x_s)$	m_{sx} + added mass
m_{sx}	sectional mass of ship and cargo
p	hydrostatic pressure
p_D	dynamically modified pressure
r_1, r_2	ratios of frame dimensions
t_f	thickness of flange
t_w	thickness of web
x	absolute vertical movement of ship
x_s	coordinate along length of ship
y_c	centroid of beam
y_{cf}	centroid of flange
y_{cw}	centroid of web
z	section modulus, relative vertical movement of ship
ε	strain
η	damage ratio
ρ	density of seawater
σ	stress
ω	circular frequency
ω_m	modal wave frequency
ω_n	structural natural frequency
ω_F	wave confidence interval weighting factor
ω_{Hs}	wave probability of occurence

1. Introduction

1.1 Problem Definition

Traditional ship design was based on experience of what had worked in the past – handed down from individual shipwright to apprentice as a craft. Insurance losses dictated a more uniform approach to shipping. Classification societies originated to categorize the insurance risk associated with existing vessels. They then moved on to develop their own Rules covering ship design in one of the first attempts to rationalize design. These Rules developed to be a mix of first principles analysis modified by statistically derived correction factors and were holistic in approach. The interactions between different design variables where not understood and not explicitly accounted for in first principles, were implicit in the statistical correction factor. By designing to the Rules a reasonable structure, though not necessarily the lightest or most efficient, could be achieved.

As Hughes noted, "We are at present in the midst of a gradual but profound change in the philosophy and practice of ship structural design.. toward rationally based design which may be defined as design which is directly and entirely based on structural theory and computer based methods of structural analysis and optimization, and which achieves optimum structure on the basis of designer-selected measures of merit"

Hughes identified two distinct levels of structural design: preliminary design covering the principal scantlings and design of details (*eg.* brackets). The engineering effort was focused on the principal scantlings and the detail design, a secondary matter, left to suit production. "In the past, during the ship design and production process, the definition of structural details has too often been left to draftsmen who had no in-depth

knowledge of fatigue design principles, and production personnel have been permitted to modify the details provided to them by the designers – generally to solve real or perceived problems." (CMS, 1997,p25) The effect was often a combination of optimized panels and poor detail.

Associated with this lack of attention to detail design was an increase in the size of ships. "In the 1960s and 1970s, the size of US built tankers increased by a factor of about six in the relatively short period of 10 years" (CMS,1997,p87). Concurrent with this, was the use of high tensile steel to allow lighter scantlings and a reduction in profit margins as fuel costs escalated.

The net effect: large, poorly maintained ships with highly loaded flexible panels supported by mediocre details in a corrosive dynamic environment. It was a recipe for trouble which struck when the first generation of optimized tankers and bulk carriers built in the early 1970s hit 15-20 years of age. The combined effects of corrosion and fatigue became apparent in the early 1990s when a series of well publicized bulk carrier casualties were suffered. Ferguson *et al.* (1992) summarize bulk carrier and OBO causalities between Jan/90 and Sept/92 (Table 1.1).

Age	Type	cargo	Date	Details
15	Bulk	grain	Jan-90	missing
17	Bulk	coal	Jan-90	side shell lost No. 1 Hold
19	OBO	iron pellets	Jan-90	heavy weather damage
9	Bulk	iorn ore	Feb-90	No. 8 side shell lost
24	Bulk	phosphate	Feb-90	Damage in No. 2 hold, flooded
20	Bulk	ballast	Feb-90	Fracture No. 1 hold
23	Ore	iron ore	Mar-90	foundered
21	Ore	iron ore	Mar-90	ballast tank leak
22	Bulk	barytes	Apr-90	2m fracture in No. 6 hold
19	Bulk	-	May-90	hull damage holds flooded
13	Bulk	iron ore	May-90	hull damage flooded
12	Bulk	iron ore	May-90	Fractures in holds 2 & 3
23	Bulk	iron ore	Jul-90	No. 3 hold flooded
18	Bulk	cement	Jul-90	bow lost and keel fractured
9	Bulk	coal	Aug-90	side shell damage
17	OBO	iron ore	Sep-90	missing
24	Bulk	iron ore	Oct-90	fractured side shell
19	OBO	iron ore	Oct-90	presumed to have foundered
17	Bulk	iron ore	Oct-90	wasted framing in No 3 hold
21	Bulk	bauxite	Nov-90	fractures in holds 2,3 & 6
19	Bulk	ballast	Nov-90	12m fracture in hold No. 5
18	Bulk	-	Dec-90	bulkhead frames loosened
17	Bulk	potash	Dec-90	fractures in No. 2 hold
18	Bulk	iron ore	Jan-91	damage to frames in No. 1 hold
24	Bulk	iron ore	Jan-91	No. 2 & 4 hold flooded
19	Bulk	-	Jan-91	fractures & detached framing
24	Bulk	iron ore	Jan-91	fracture in No. 5 hold, flooded
21	OBO	iron ore	Feb-91	fracture in No. 1 hold
14	Bulk	ballast	Feb-91	fracture in No. 3 WB hold
24	Bulk	pig iron	Feb-91	took water after striking object
17	Bulk	-	Mar-91	frames detached, No. 6 hold
24	Bulk	iron ore	Apr-91	No. 1 hold flooded
21	Bulk	iron ore	Apr-91	fracture in No. 4 hold, flooded
9	Bulk	iron ore	Apr-91	missing
16	Bulk	iron ore	May-91	fracture in No. 1 hold
16	OBO	iron ore	May-91	fracture in hull below waterline
15	Bulk	iron ore	Jul-91	No. 3 hold flooded
21	Bulk	steel	Aug-91	grounded No. 4 hold flooded
21	Bulk	iron ore	Aug-91	severe crack No. 7 hold
16	Bulk	iron ore	Aug-91	flooding No.1 & 2 hold
12	Bulk	-	Aug-91	shell & frame fractures No. 3
14	Bulk	-	Aug-91	heavy weather frames cracked
23	Bulk	-	Aug-91	disabled, towed to safety
24	Bulk	-	Aug-91	No. 5 hold frames fractured
18	OBO	oil	Aug-91	hull fractures
18	OBO	oil	Sep-91	shell fracture,oil leakage
23	Bulk	phosphate	Oct-91	sank in Mediterranian
18	Bulk	to load coal	Oct-91	fractures in deck
14	Bulk	-	Nov-91	extensive corrosion of shell plating
25	Bulk	iron ore	Nov-91	Leakage in Heavy Weather
18	Ore	steel machinery	Nov-91	No.2 hold flooded
18	Bulk	iron ore	Nov-91	cracks in side shell, most holds
20	Bulk	iron ore	Nov-91	side shell fracture
26	Bulk	-	Jan-92	hull leaks on voyage to breakers
25	Bulk	calcium nitrate	Jan-92	leakage to forepeak and db
21	Bulk	coal	Jan-92	detained
18	Bulk	-	Feb-92	upper deck fracture, heavy weather
23	OBO	iron ore	Mar-92	engine room flooded
24	Bulk	iron ore	May-92	side shell fracture and flooding
20	Bulk	iron ore	Sep-92	bhd No. 8/9 holds collapsed

Table 1.1: Casualty Statistics(Ferguson *et al.1992)*

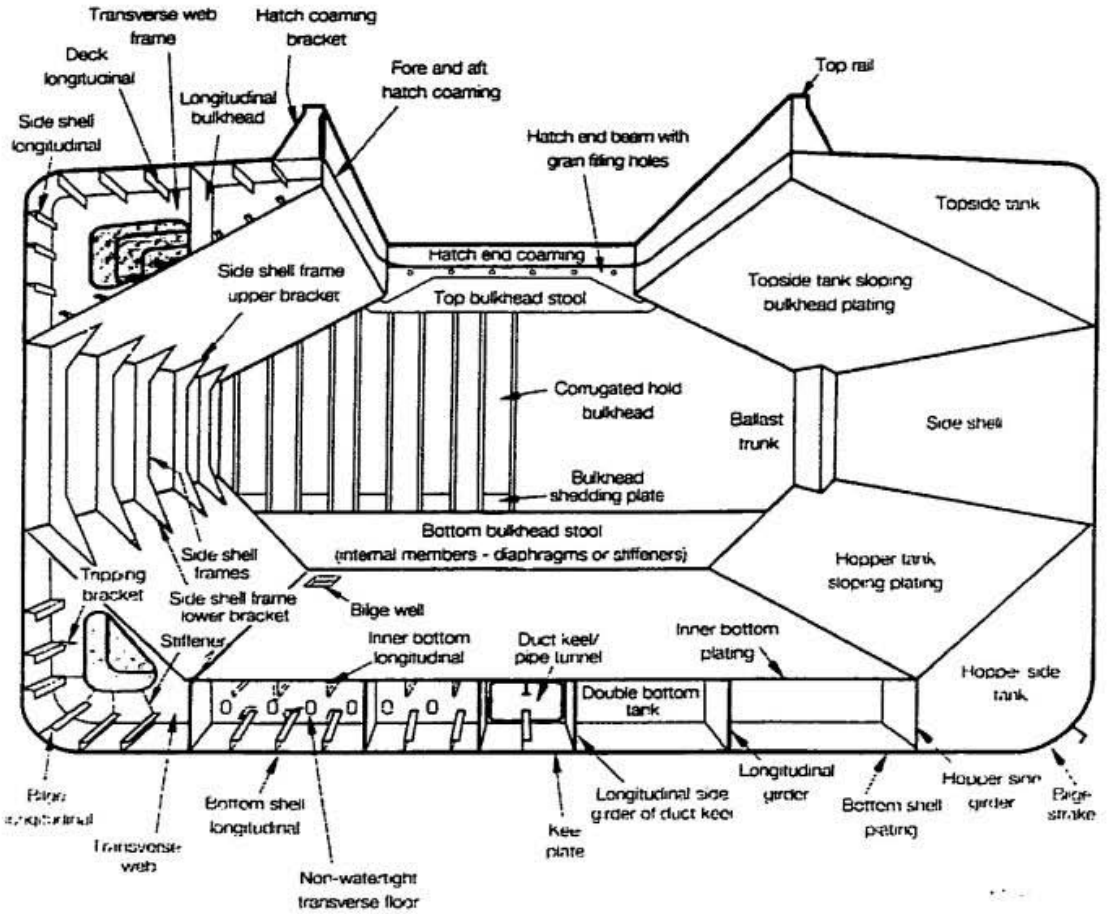
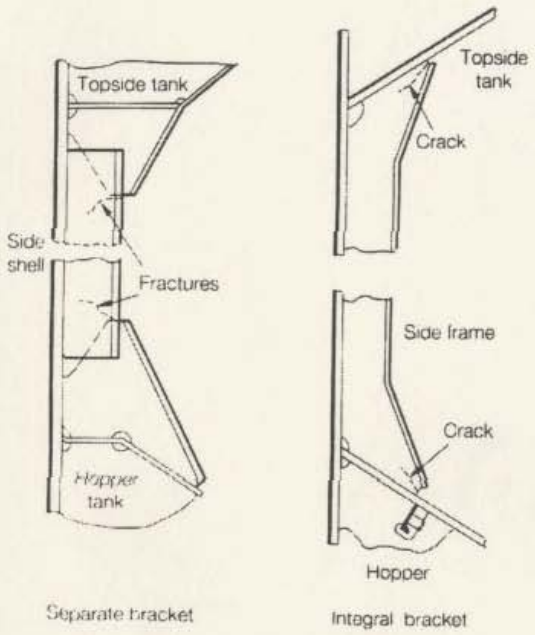


Figure 1.1: Typical Bulk Carrier(Ferguson et al.1992)



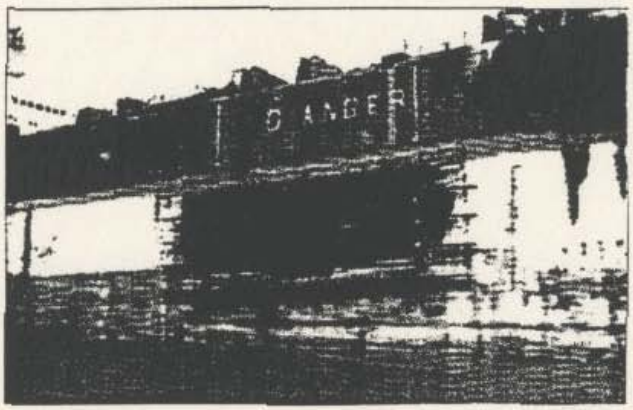


Figure 1.2: Frame Cracks(Ferguson et al,1992.) ***Figure 1.3: Shell Punch Out (Ferguson* et al.,*1992)***

As can be seen from Table 1.1, a significant number of failures involve flooding due to progressive failure of cargo hold side shell frames (see Figure 1.1 for a typical bulk carrier structure).

Typically, the upper/lower end brackets of a side shell frame fail due to fatigue(Figure 1.2). Any load these frames carried is shed to the adjacent frames which in turn accelerates their failure. This chain reaction proceeds leaving relatively large portions of the side shell unsupported and thus susceptible to accelerated bending fatigue or wave punch out (see Figure 1.3).

Upon failure of the side shell, the hold is flooded. If the transverse bulkheads are weakened, wave action can cause collapse of the bulkheads separating the holds and cause the flooding to proceed from hold to hold, quickly leading to total loss.

All ships can experience fatigue problems. Ushirokawa (1981) makes the following observations based on fatigue testing of representative tanker structures:

1) Fatigue cracks start around the point of maximum measured stress/strain.

2) The position of crack initiation is influenced by extremely localized stress concentrations such as those produced by the shape of the bead in fillet welds and surface flaws present on members.

3) Fatigue cracks initiate, not at one but simultaneously at several points

4) While depending on the form of the structure represented and the conditions of loading, cracks generally initiate at the toe or root of fillet welds

5) In many cases, the cracks initiate at wrap-around fillets surrounding brackets or stiffener extremities, or at parts of similar configuration, and propagate in the direction of the seam axis or else towards the plate member.

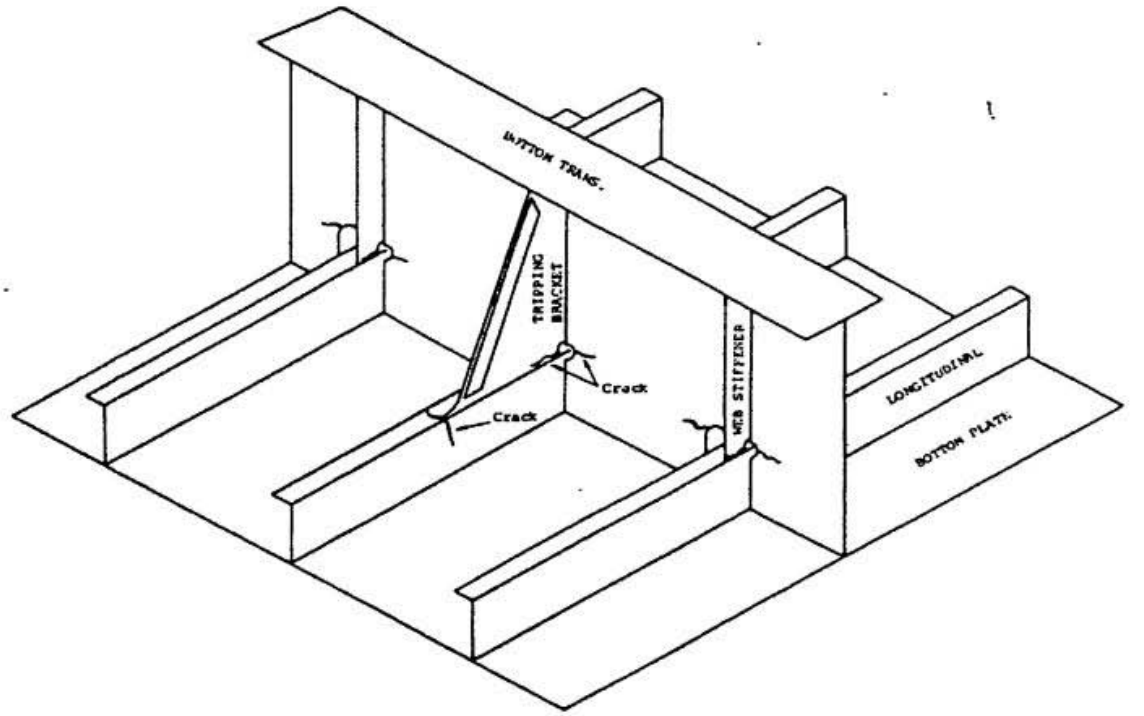


Figure 1.4: Typical Fatigue Damage in Ships

The management of these fatigue cracks in operation is difficult, due in part to the following:

-the cracks may not always be detected because of dirt or because they are in members which are in compression at the time of survey;

-repairs are expensive, not only the cost of doing the repair but also the lost charter (e.g. $10 000 to $33 000 per day for a typical ship) and repair jobs running from a day to weeks depending on availability of approved steel and the scope of the repair;

-repair facilities may not be available – *eg.* if a crack opens up in the main deck, mid-ocean, there is not much the crew can do except drill the ends of the crack and hope;

-the liability to the owner due to an oil spill from a crack in the shell can be enormous.

Given the above difficulties, it would appear that in association with optimized primary structure, detail design must also be optimized for its environment. The nature of this optimization being for example: softening of bracket noses and crowns; use of notch toughened steels; local redesign and/or use of over-sized sections. A typical example of such a recommendation for a repair is shown in the Figure 1.5.

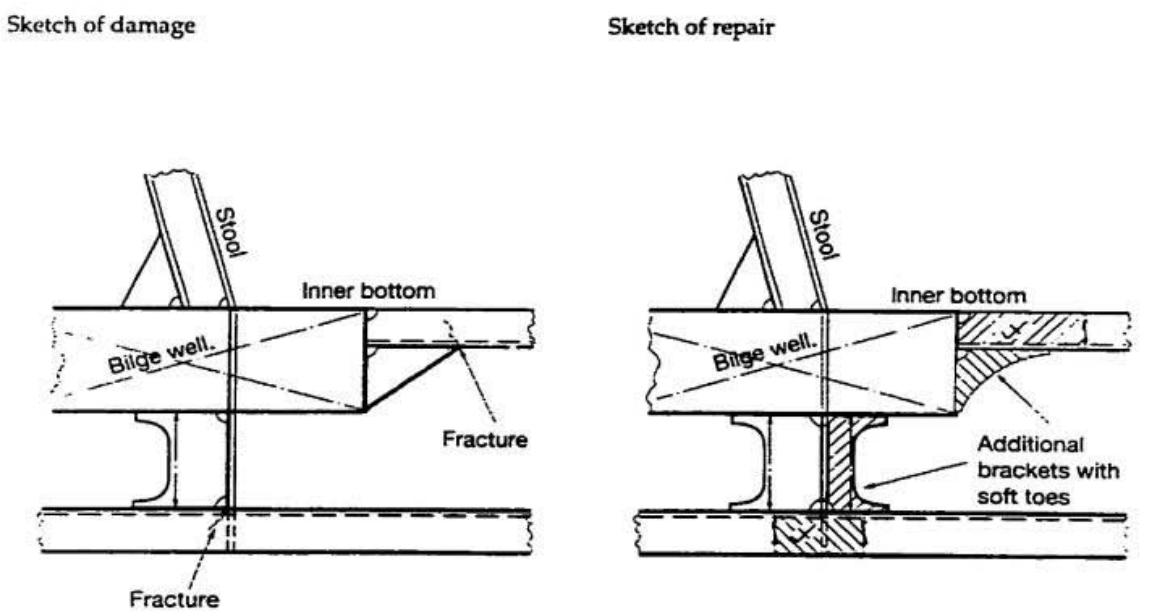


Figure 1.5: Typical Repair Details

Damage tolerant methods of fatigue crack management have been developed by the aerospace industry and are well established in those industries (CMS,1997; Harris,1997). Ma *et al.*(1995) describe a computer based repair management system for ships. These methods however are not yet widely accepted in the marine industry. In specific instances, these methods are appropriate but generally, considering the difficulty and cost of inspection, repair and liability, it seems preferable to concentrate on preventing fatigue cracking in the first instance through design.

Methods used to design structures for fatigue strength are now well established and codified in various rules and standards. There is however a cost associated with this. The question must be raised: do all details need to be optimized? If not, is there a way to identify which details must be optimized and which may be left to suit production? These are the questions that will be addressed herein.

1.2 Scope of Work

The present work focuses on the problem of bulk carrier side shell frame fatigue introduced in the previous section.

This focus is chosen because the overall problem is too broad. One purpose of this work is to review some of the issues involved in fatigue analysis and demonstrate the breadth of the problem. This breadth of scope may in fact preclude any simple means of general fatigue analysis. Despite this inherent uncertainty, it remains that some criteria for categorizing particular details as sensitive or insensitive to fatigue is desirable and in certain situations, such as the one developed here, are possible.

Consequently, this thesis:

1) reviews basic fatigue concepts;
2) puts these concepts into a shipboard framework;
3) develops and analyzes a simplified model of a bulk carrier frame;
4) examines fatigue life of this frame as a function of ship length and loading condition.

To date, there is no known published work specifically examining the fatigue life of bulk carrier side shell frame lower end brackets as a function of ship length and loading condition.

1.3 Methodology

The method used to perform the fatigue analysis is as described below. This thesis focused on the fatigue life of a typical bulk carrier lower end bracket. A "typical" bracket was defined by first performing a parametric study of a series of bulk carriers to establish basic dimensions as a function of length. Given these basic dimensions, the "Rules and Regulations for the Classification of Ships" (Lloyd's Register, 1999) were applied to determine scantlings.

The frame was assumed to be in the middle of a very long cargo hold of a flexible ship. The effect of transverse bulkheads in supporting double bottom sag due to cargo weight was ignored. The bottom hold was assumed to sag uniformly in the region of the frame leading to a downward rotation of the transverse floors which was carried up to the side shell frames as some unknown bending moment. Variations about this load, responsible for fatigue, arise from passage of the vessel through a seaway. For simplicity, the ship was assumed to have zero speed of advance in a head sea. A standard spectrum (ITTC) was used to describe the distribution of wave energy and hence height as a function of frequency. The response of the vessel was modeled as a one degree of freedom system thus yielding response amplitude operators as functions of frequency for instantaneous acceleration and draft.

Given an acceleration and a draft, simple expressions were developed for loading. These loads were applied to a finite element model of the frame. The finite element model assumed the frame was attached to the hopper and topside tank by spring supports which account for structural flexibility and model the local reaction between the frame

and hopper tanks. The stress in the toe weld as a function of frequency was simply determined by dividing this reaction by the weld area.

The results from the finite element model were then applied to a hot-spot stress fatigue analysis procedure. Using category 21S S-N data from the Ship Structure's Committee (Stambaugh, 1992), the toe stress was used to calculate the damage ratio for each component of the spectrum using Miner's Rule. The damage ratio per component was calculated by assuming the ship is fully utilized so the number of stress cycles experienced would be the number of seconds in a day divided by the period of that component. The total damage per day was obtained by numerically integrating the damage ratio over the entire range of wave frequencies. The number of days to failure was calculated by dividing the failure ratio by the daily damage. A failure ratio of 0.3 was used as recommended by Hughes (1983).

2.Background

2.1 Definitions

Fatigue is the gradual accumulation of cyclic strain damage leading to crack initiation and propagation. The cyclic strain response depends on design, loading and dynamic response if there is some overlap between the range of forcing frequencies and structural natural frequencies.

A body subjected to a load will develop a stress field. The stress field will vary depending on the load bearing area and stress concentrations. Areas of high local stress are "hotspots".

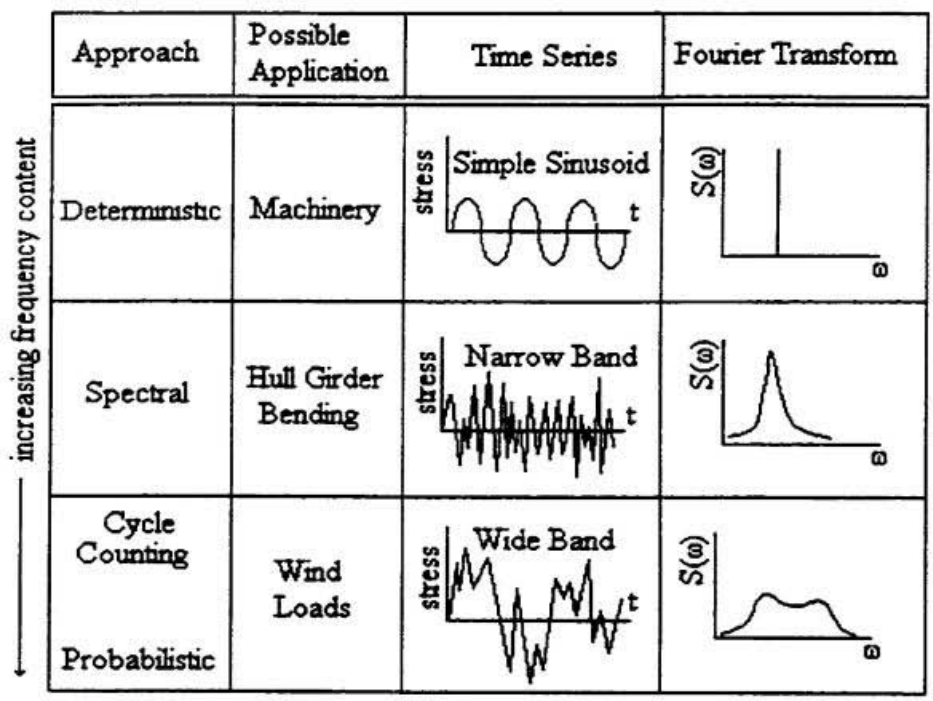


Figure 2.1: Load Types

Load patterns may be categorized as static, periodic or random. Static loads are constant. Periodic loads cycle through a constant range of values at a characteristic rate called the frequency. Random loads are stochastic time series with no deterministic interpretation. The various moments of the distribution (*eg.* mean) may be known but the instantaneous values of the series (particularly the peak values responsible for the majority of fatigue damage accumulation) cannot be known beforehand.

The stress history produced is determined by the load pattern (see Figure 2.1).

If the loads are periodic, the induced stress field will be periodic. If the period of the load approaches that of the body, dynamic effects can increase the induced stress field further.

Different situations will fall into different categories. Reciprocating machinery or rotating shafts result in approximately sinusoidal stress histories with unit impulse Fourier transforms. Simple interference loads can produce narrow banded beating patterns with discrete Fourier transforms. Continuous structures can have narrow banded stress spectra or wide banded irregular stress histories depending on the frequency content of the excitation and its frequency response function.

Generally, one of the first steps in fatigue analysis is to determine the stress history category.

2.2 Sinusoidal Stress Histories

The simplest and most important stress history is the simple sinusoid (see Figure2.2) where the Fourier transform is a single discrete spike at the forcing/natural frequency.

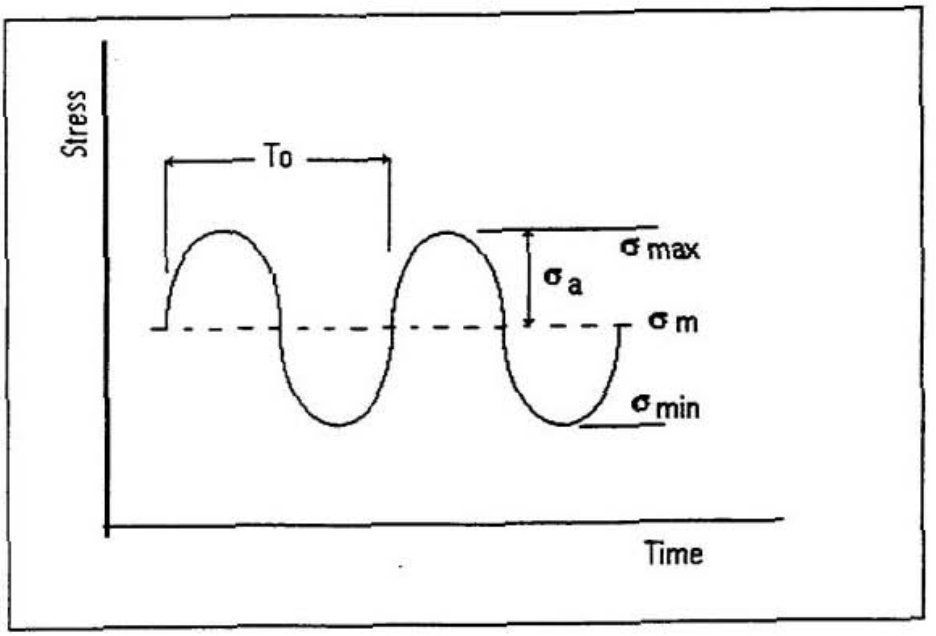


Figure 2.2: Schematic of Simple Sinusoid Stress History

The following relationships are important in fatigue analysis.

stress range: $\Delta\sigma = \sigma_{max} - \sigma_{min}$ (1)

stress amplitude: $\sigma_a = \frac{\sigma_{max} - \sigma_{min}}{2}$ (2)

mean stress : $\sigma_m = \frac{\sigma_{max} + \sigma_{min}}{2}$ (3)

stress ratio: $R = \frac{\sigma_{min}}{\sigma_{max}}$ (4)

We note that the period is constant , T_o , and that for each period there is only one maxima and one minima, consequently, the number of peaks and cycles for a given record is well defined.

Construction of Stress Histograms for a Simple Sinusoidal Process

The stress histogram for a sinusoidal process (see figure 2.3 below) would consist of one entry whose amplitude is the stress range and whose count n_i is the total time in the history divided by T_{CYCLE} .

$$n_i = \frac{T_{TOTAL}}{T_{CYCLE}} = \frac{\omega T_{TOTAL}}{2\pi} \tag{5}$$

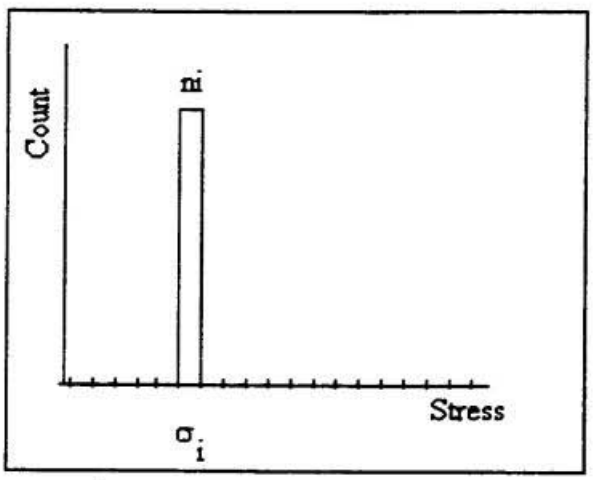


Figure 2.3: Stress Histogram for Sinusoidal Process

2.3 Spectral Methods

As the frequency content of the stress history increases, the waveform becomes irregular and so the certainty of a constant amplitude and period drops as the stress history slips towards randomness (see Figure 2.4 below)

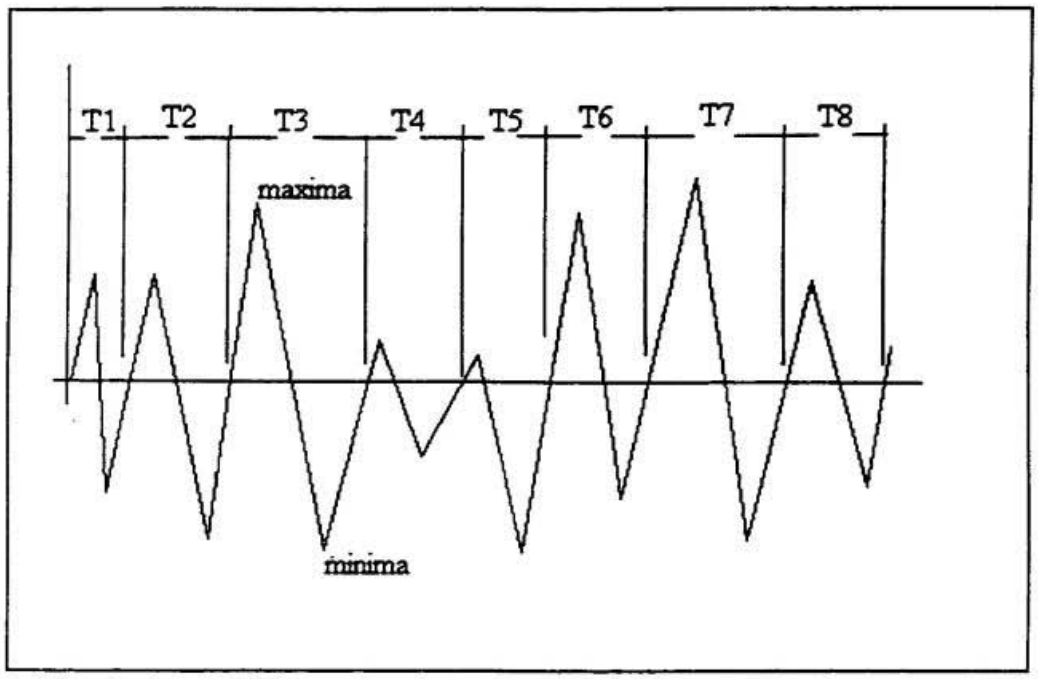


Figure 2.4: Narrow Banded Process

This slip is gradual though and when the range of frequency content is small, the history may be considered narrow banded. An essential characteristic of a narrow banded process is that there are unique cycles in the record – the amplitude and period may change but there is the constant pattern of zero:maxima:zero:minima:zero. Expressions can be developed for the number of cycles based on the properties of the distribution using probabilistic theory.

The process is treated as a random process and the instantaneous value of the process at any time t is a random variable X with a probability density function $f(x)$. Moments of the distribution are defined as follows:

$$m_n = \int_{-\infty}^{\infty} x^n f(x)dx \tag{6}$$

Using this, various properties can be defined for a process possessing a probability density function $f(x)$.

Expected value:

$$E[X] = m_1 = \int_{-\infty}^{\infty} xf(x)dx \tag{7}$$

Expected number of peaks (for narrow banded processes):

$$E[P] = \sqrt{\frac{m_4}{m_2}} \tag{8}$$

Expected number of zeroes (for narrow banded processes):

$$E[0] = \sqrt{\frac{m_2}{m_0}} \tag{9}$$

An irregularity factor which gives a quantitative indication of the frequency spread is:

$$\gamma = \frac{E[0]}{E[P]} \tag{10}$$

which varies between 0 (wide banded) and 1.0 (narrow banded)

If the process is narrow banded and Gaussian, the population of peaks can be shown to follow a Rayleigh distribution:

$$p_{\bar{x}}(\bar{x}) = \frac{\bar{x}}{\sigma_x^2} \exp\left[\frac{-\bar{x}^2}{2\sigma_x^2}\right] \tag{11}$$

If the process is further restricted to be stationary (*i.e.* moments are invariant under time transformations), then spectral methods can be used. This is analogous to Fourier analysis in which a complex signal is broken into a weighted sum of component sinusoids. This can be interpreted graphically as a linear transformation from the time domain to a vector space whose basis is the set of sinusoids of frequency ω_i and with coordinate vector G_i.

The Fourier transform pair are defined as :

$$g(t) = \int_{-\infty}^{\infty} G(\omega) e^{i\omega t} d\omega \tag{12}$$

$$G(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} g(t) e^{-i\omega t} dt \tag{13}$$

Where G is called the Fourier transform of g.

Construction of Stress Histograms using Spectral Methods

Given this set of basic sinusoids (*i.e.* stress spectrum), the stress histogram is created by superimposing the stress histograms from the basic sinusoids as obtained in Section 2.2.

In the spectral method, a correction factor (usually based on correlation to cycle counting results) must be introduced when the process departs from the narrow band assumption to wide band status.

2.4 Cycle Counting Methods

As the frequency content increases still further, the ordering of maxima and minima is lost and the pattern is random (see figure 2.5).

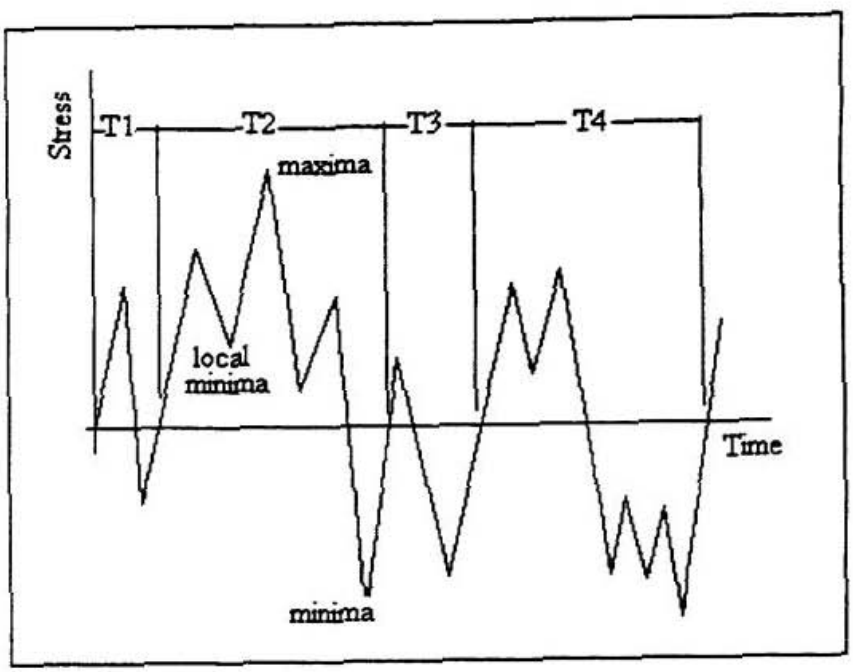


Figure 2.5 : Random Stress History

In this case, two approaches exist to sift out meaningful cyclic load information:

1) Cycle Counting (dealt with here)
2) Stochastic Modeling (dealt with in the next section)

The first is a cycle counting technique which works directly with the history Several techniques exist: level crossing, peak counting, simple range counting, rainflow counting. Only level crossing and rainflow counting algorithms will be described here.

Procedure for Level Crossing Counting

A reference stress is chosen to zero the series. The stress axis is divided into increments above and below this reference stress. Starting on the first positive slope in the series, a count is made whenever (Bannantine *et al.*,1990):

1) a positive slope crosses the reference
2) a positive slope crosses a new increment above reference
3) a negative slope crossed a new increment below reference

An example of the procedure is below (figure 2.6). Here the reference stress is zero and the level counts (marked with x's) were obtained using the previous rules. Given the level counts, distinct cycles are obtained by forming progressively smaller cycles as the level counts are used up.

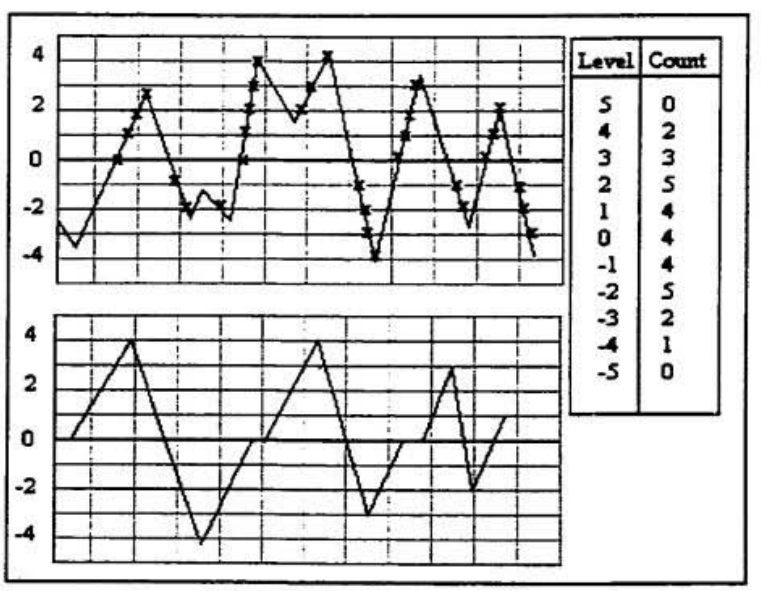


Figure 2.6 Level Crossing

Procedure for Rainflow Counting

Bannantine *et al.* (1990) reports rainflow counting was first introduced by Matsusihi and Endo. This procedure has the advantage of being able to count closed hysterisis loops. Bannantine *et al.*(1990) report the following ASTM recommended algorithm for performing rainflow counting.

1) The history is truncated so it starts with either a maxima or a minima. X = the range under consideration and Y the previous range adjacent to X.
2) Read the next peak or valley. If out of data stop
3) If there are less than three points go to step 1. Form ranges X and Y using the three most recent peaks and valleys that have not been discarded.
4) Compare the absolute values of ranges X and Y
    a. if $X<Y$, go to step 1
    b. if $X>=Y$, go to step 4
5) Count range Y as one cycle; discard the peak and valley of Y and go to step 2

Construction of Stress Histograms from a Cycle Counting Procedure

Cycle counting procedures (*eg.* rainflow or level counting) work directly with a measured portion (T_o) of the total stress history (T). Consequently, care must be taken that the history used is representative of the total history and some consideration given to the projection of a short term measured history for use as a long term stress history.

Given a representative block of stress T_o and a set of cycles $1...m$ of stress range $a_i... a_m$ derived from one of the cycle counting previously described, the number of cycles of stress a_i is T/T_o (see Figure 2.7 below).

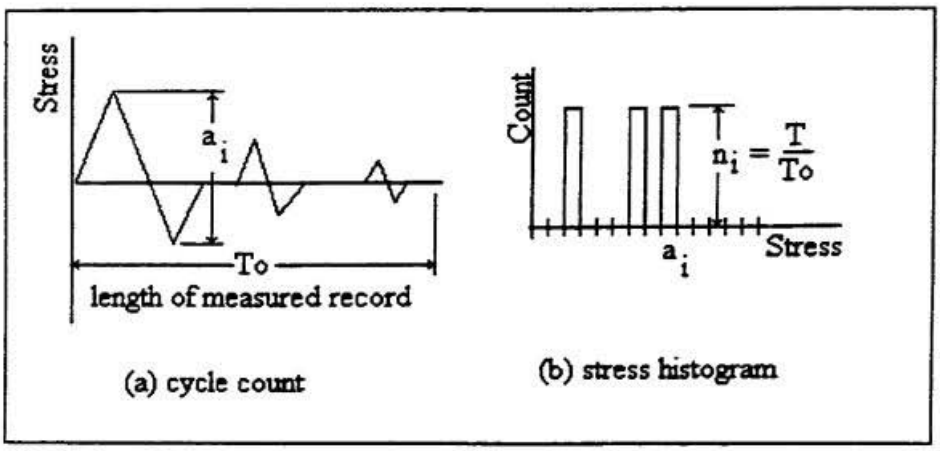


Figure 2.7: Stress HistogramConstruction using Cycle Counting

2.5 Probabilistic Methods

Another approach to obtaining cyclic loads, given a wide band, history is to assume the peak data follows a probability distribution which is in turn used to establish the distribution of stresses.

The probability distribution is chosen based on similarities to the expected distribution of stress cycles in the parent data. The following are commonly assumed probability density functions found in any standard text (Devore,1987).

Normal

$$p_x(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{14}$$

The mean(μ) is the 50% quartile and the standard deviation (σ) a measure of spread of the bell curve. The central limit theorem provides some justification for assuming the data is normally distributed as a first approximation(Devore,1987).

Rayleigh

Given a normally distributed general population, the population of peaks can be shown to follow the Rayleigh distribution shown below:

$$p_{\bar{x}}(\bar{x}) = \frac{\bar{x}}{\sigma_x^2} \exp\left[\frac{-\bar{x}^2}{2\sigma_x^2}\right] \tag{15}$$

Again, σ_X is a measure of spread of the population.

Lognormal

The lognormal probability distribution function is given below:

$$p_x(x) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left[\frac{(\ln x-\mu)^2}{2\sigma_x^{\,2}}\right] \tag{16}$$

This is a commonly used specialization of the normal distribution where it is assumed that ln(x) is normally distributed. This distribution is skewed. σ_X is a measure of dispersion.

Weibull

The Weibull probability distribution function is given below:

$$p_x(x) = \frac{\alpha}{\beta} x^{\alpha-1} \exp\left[-\left(\frac{x}{\beta}\right)^{\alpha}\right] \qquad (17)$$

α is a shape factor inversely proportional to the skew. β is a scale parameter.

Other probability distributions exist and the one chosen is what most accurately reflects the data. The relevant parameters would be determined by curve fitting to any relevant measurement data the analyst may have available. For example, Chen *et al.*(1986) describe work that was done on the Sea-Land class of container ships where strain gauge measurements were used to calibrate a gamma distribution.

Given the probability distribution function, the cycle count is determined in one of the following ways:

1) an equivalent RMS value can be formulated, but as discussed by Vaughan (1982) this has the disadvantage of neglecting the very peak values which are of interest.
2) probability exceedance charts can be created from which return periods and expected number of cycles in a given span can be calculated.

2.6 Material Properties

Steel is formed from refined iron ore with controlled additions of alloying agents. Impurities are always present in the melt. On solidification, metallic crystals form around nucleation points. These crystals entrap some of the impurities/alloying agents as interstitial or substitutional atoms in a solid solution and push the excess to the side creating a complex multi-phase microstructure. Dislocations (edge, screw, mixed) are formed as "errors" in the laying down of atoms during solidification. Dislocations are areas in a crystal where either an extra half-plane of atoms are present (edge dislocation), a half plane is shifted relative to each other (screw) or some combination of the two(mixed) (Shackleford, 1988). These dislocations are areas of high potential energy as the crystal structure is deformed around them. Solid state diffusion of the entrapped impurities occurs so the total potential energy of the system falls to a local minima referred to as the Cotrell atmosphere.

As Xu(1997) notes, "Because atoms seek the lowest energy configuration, the dislocations are effectively locked in place until an applied stress large enough to overcome the Cotrell atmosphere moves the dislocation. Once it has been moved, much smaller loads can continue the dislocation movement which is responsible for local plastic strains." This effectively establishes a threshold stress which must be exceeded for the accumulation of plastic strain or fatigue damage.

The response of steel to a monotonically applied stress is summarized on the stress-strain diagram (figure 2.8) below.

On application of stress from zero, the atomic bonds are stretched until the yield stress is reached where the Cotrell atmosphere is overcome and dislocation movement begins. Stress raises to an ultimate limit where the material will rupture.

Figure 2.8: Monotonic and Cyclic Stress-Strain Curves

The initial stretching of bonds (slope = elastic modulus *E*) is linear, elastic and recoverable upon unloading. The movement of dislocations is non-linear and non-recoverable; it represents permanent plastic damage.

The response of steel to cyclic stress shows a hysterisis effect as energy is absorbed in doing plastic strain and work hardening/softening results. The energy absorbed is proportional to the area under the curve.

Bannantine et al. (1990) reports, "Generally, transient behavior (strain hardening or softening) occurs only during the early cycles. After this the material achieves a cyclically stable condition after approximately 20-40% of the fatigue life. Consequently, fatigue properties are usually specified at the half-life when the material response is stabilized. The cyclic stress-strain curve is obtained by connecting the vertices of successive hysterisis loops.

2.7 Fatigue Process

A complete analytical description of fatigue is not available but it may be described qualitatively. Fatigue cracks begin at the surface or internal flaws where twin slip faults are formed at a stress concentration. Crystals favorably oriented toward slip are forced to yield back and forth in each cycle leading to a surface intrusion.

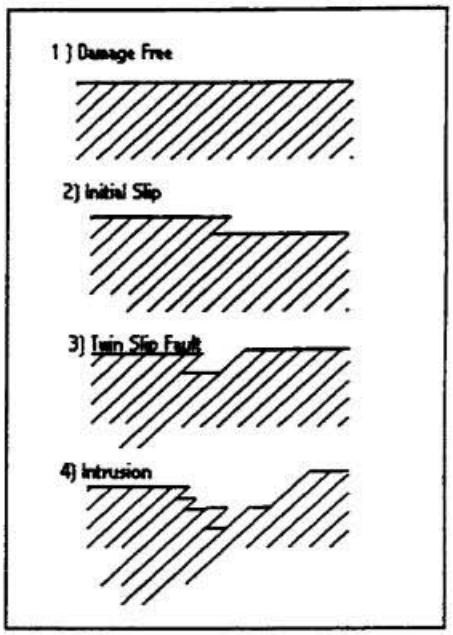


Figure 2.9: Schematic of Surface Intrusion Formation

Ellyin (1997), identifies these slip zones as persistent slip bands (PSB) and reports cracks tend to initiate at the interface between the PSB and surrounding matrix when a localized saturation of dislocation movement has occurred (*i.e.* when the material has absorbed all the plastic damage it can). Lampman (1996) reports initiation of fatigue cracks has been observed to occur along slip bands, in grain boundaries, in second phase particles and in inclusion or second-phase interfaces with the matrix phase.

Just initiated fatigue cracks are of the order of 0.1 microns (Lampman,1996, p67). The growth of cracks this size (Stage I) is heavily dependant on local micro-structure and is a very complex metallurgical problem beyond the current scope. Some/many micro-crack(s) will emerge as dominant and grow by plastic deformation along the slip planes. When the cracks reach a certain size (typically 3 to 4 grains), metallurgical randomness loses prominence and their growth may be modeled using fracture mechanics and this is termed stage II. According to fracture mechanics, a crack will propagate when the strain energy released around the tip by crack extension is greater than the surface energy absorbed by enlargement of the fracture surface. For cyclic loading this energy is supplied during the peak of each cycle. Stable crack growth occurs until the critical crack length is reached, whereupon the section will rupture.

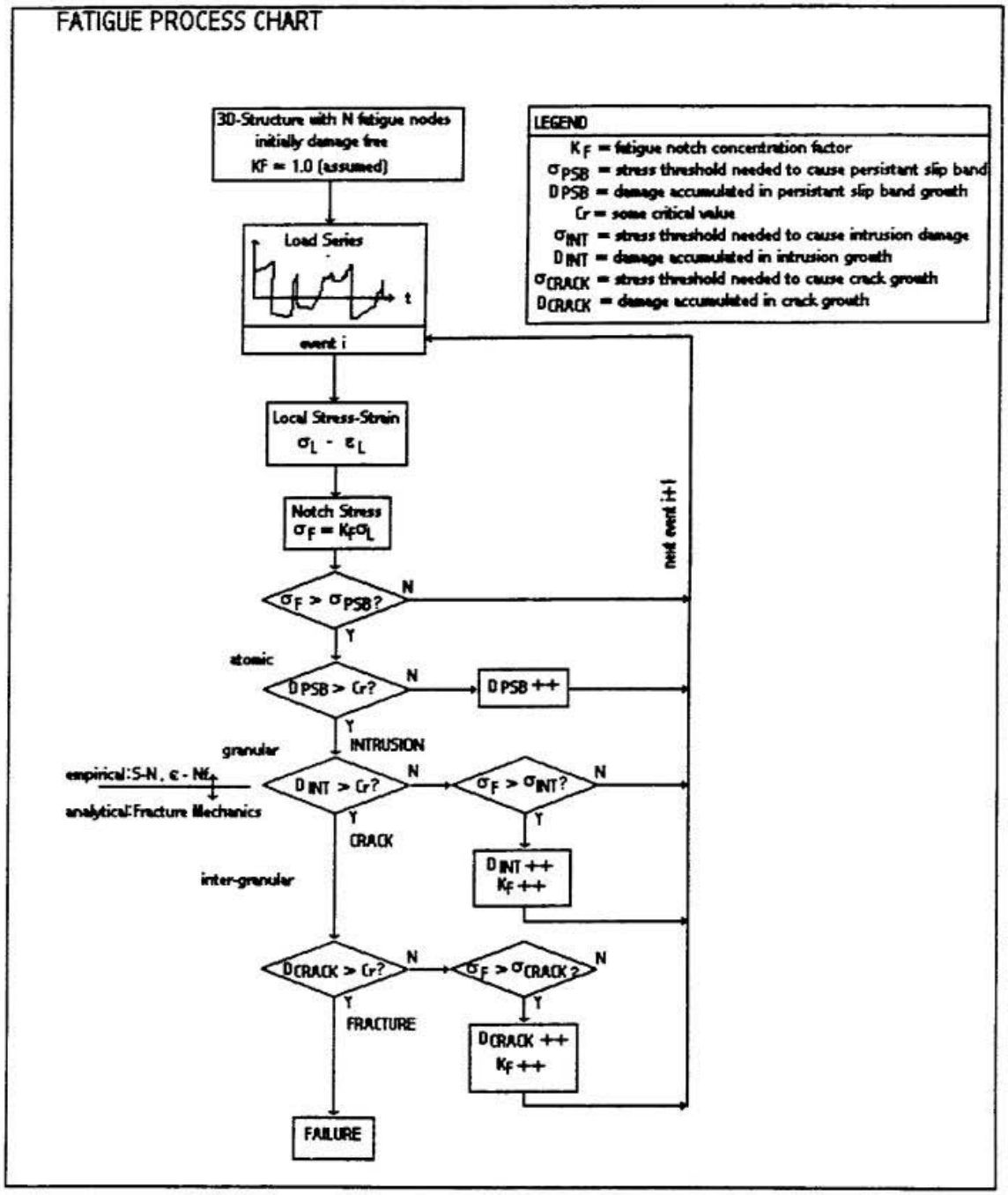


Figure 2.10: Fatigue Process Chart

2.8 Test Data

Bannantine *et al.*(1990) report, work on fatigue has been ongoing for approximately 150 years. Early work was done by Wohler (circa 1860) on rotating tests of railway axle samples to establish a safe alternating stress below which failure would not occur. He was the first to use the now familiar S-N diagram. The importance of cyclic straining was first noted by Jenkin in 1923 but ignored until the 1950's when Coffin and Manson established quantitative relationships between strain and fatigue life. Work on the propagation of cracks was published by Griffith in the 1920's. It was based on energy considerations in brittle materials. In the 1940's, Irwin extended the theory to ductile materials by quantifying plastic strain energy. In the 1950's Irwin published an asymptotic expansion for local stresses near the crack tip leading to linear elastic fracture mechanics (LEFM). LEFM is used as a basis for crack growth and rupture models given an assumed crack size.

Crack behavior is adequately modeled for macro-cracks by fracture mechanics but no such model exists for initiation and coalescence of micro-cracks. Empirical test data of the form "stress/strain vs. cycles to failure at load amplitude A and constant frequency f are used to predict crack initiation. The most common is the stress vs. number of cycles (S-N) curve. These approaches are summarized in Table 2.1

Model	*Design Goal*	*Material Model*
S-N	infinite life	elastic
ε-N	infinite life	elastic-plastic
da/dN	finite life	elastic-plastic
hybrid ε-da/dN	finite life	elastic-plastic

Table 2.1: Fatigue Models

Fatigue is a function of material properties and applied stress-strain history Analysis is based on applying the results of test samples to simplified representations of real structures. Fundamental to this is an understanding of the conditions under which these test results were obtained and consequently the limitations they impose on realistic application. For the fatigue assessment of a structure to be valid, the experimental data used must somehow "represent" it.

Fatigue crack propagation is adequately described using fracture mechanics. Fatigue crack initiation is due to the accumulation of strain damage for which no analytical method exists to predict it. Instead empirical results are used to predict crack initiation.

Three types of empirical data exist depending on the controlled variable. Stress-life data exists for controlled load tests and strain-life data exists for controlled displacement tests. The two methods are related as will be shown. The third method is a plot of crack length a versus number of cycles N for use in the fracture mechanics approach.

The stress-life, S-N method was the first approach used.. Ideal test specimens were subjected to a fully reversed mechanical deformation (e.g. rotating bend test) where it experiences a uniform load/moment each cycle. The number of cycles to fracture the specimen was counted.

The nominal stress is calculated based on the applied load and specimen geometry using simple elastic formulae (e.g. $\sigma = My/I$). Note, the elastic formulae were used even when the stresses were outside the elastic limit. All plasticity is ignored.

The test is repeated for multiple loads and results plotted on an S-N diagram (alternating stress vs. cycles to failure), an example of which is shown below in Figure 2.6

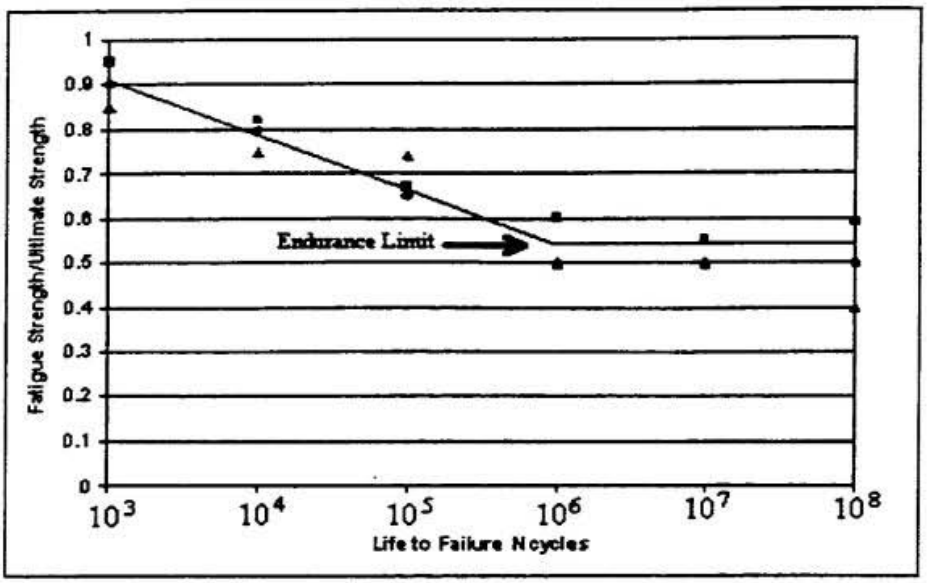


Figure 2.11: Schematic of Typical Raw S-N Data

As can be seen from the S-N curve, the curve flattens out at some value around 10^6 cycles known as the fatigue limit. Below this limit, fatigue cracking will not start in a similar experimental set-up. This can be related to the observation in Figure 2.5 that a certain value of stress is necessary to start the accumulation of fatigue damage by overcoming the Cotrell environment. In reality, extraneous factors (*eg.* damage, corrosion or material properties) often rule out the existence of an endurance limit.

The S-N curve is approximated using a power curve:

$$N(\Delta S)^m = A \qquad (18)$$

On a log-log scale, this is linearized. Bannantine *et al.*(1990) reports this first done by Basquin in 1910:

$$\log(\Delta S) = -\frac{1}{m}\log(N) + \log(A^{\frac{1}{m}}) \tag{19}$$

or

$$\log(N) = \log(A) - m\log(\Delta S) \tag{20}$$

Linear regression can be performed on the logarithmically transformed ΔS-N pairs to get a line of best fit with slope $-1/m$ and y-intercept $\log A = \log \Delta Sa$. An example of analyzed data is given in figure 2.12 below.

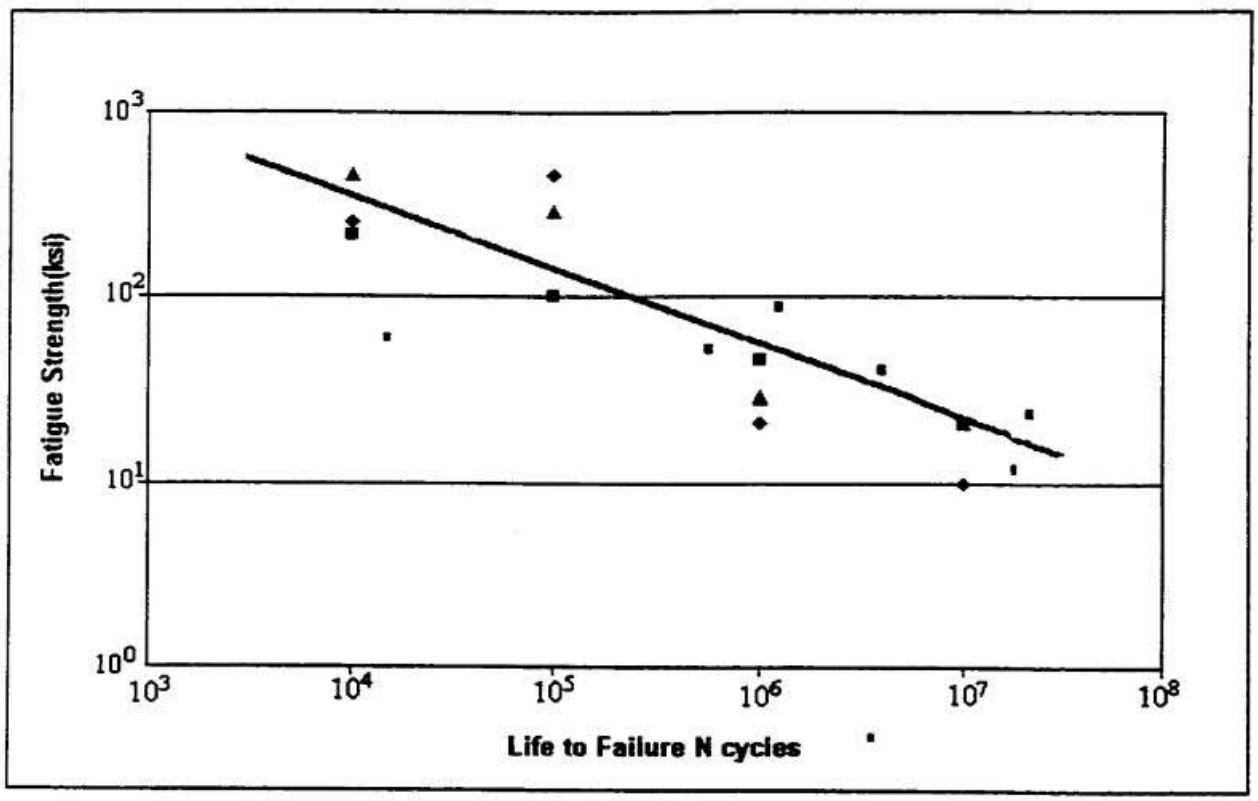


Figure 2.12: Schematic of Typical Converted S-N data

Now given ΔS_a, and m from analysis of the linearized ΔS-N data , we can reverse the logarithmic transformation:

$$N = \frac{\Delta S_a}{(\Delta S)^m} \tag{21}$$

Failure occurs in the test specimen when:

$$N > \frac{\Delta S_a}{(\Delta S)^m} \tag{22}$$

or equivalently, in terms of fatigue damage ratio (η):

$$\eta = N\frac{\Delta S_a}{(\Delta S)^m} = \begin{Bmatrix} > 1 \Rightarrow failure \\ = 1 \Rightarrow critical \\ < 1 \Rightarrow safe \end{Bmatrix} \tag{23}$$

This is the stress-life evaluation statistic for machined test specimens subjected to a uniform fully reversed load/moment with zero mean.

Two types of S-N analysis exist (Moan and Berge,1997): nominal stress approach and hot-spot stress approach.

In the nominal stress approach, the structure or component under consideration must be similar to the test piece in size, shape and material. Given, this correspondence, the fatigue stress is calculated using simple elastic formulae as above.

In the hotspot stress approach, the requirement for geometric similarity is relaxed. Test data is generated for classes of standard structural details (see figure 2.13 below for a sample). The class most closely corresponding to the one in question is used. The stress is calculated to match that reported – usually it is stress immediately adjacent to the detail (*i.e.* hot-spot stress). While the requirement for complete similarity has been relaxed, it is still necessary to maintain similarity of : material, welding effects, surface condition and notch effects.

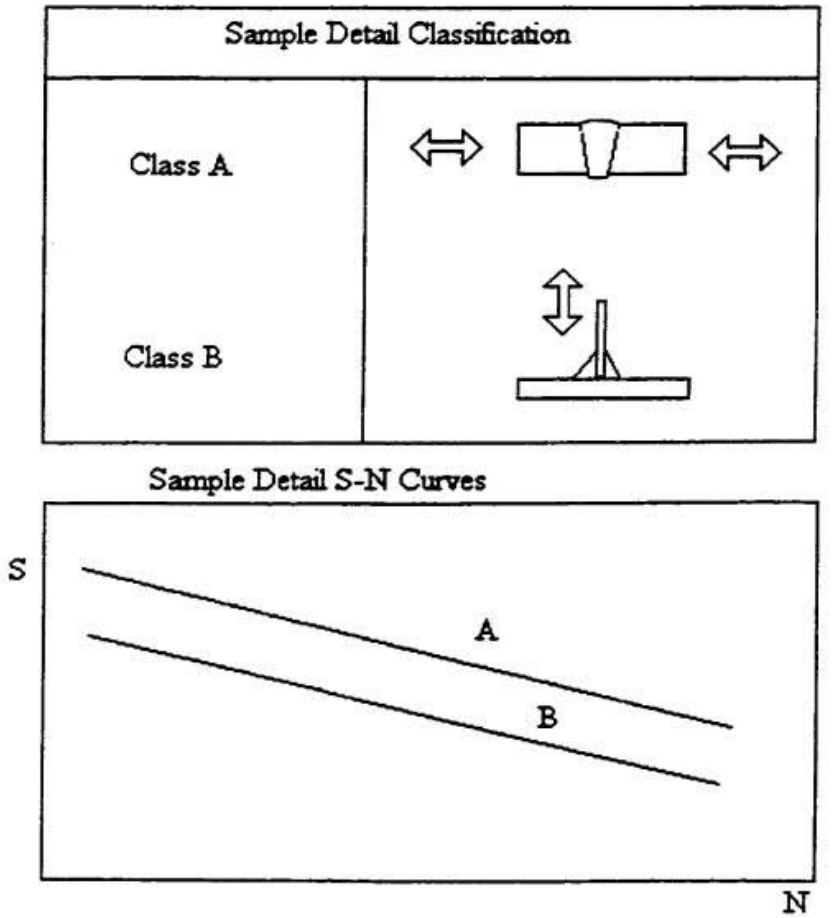


Figure 2.13: Schematic of S-N Curve Categories

This hotspot stress approach is the most commonly used in codes. The readers are referred to any of the many available codes or standards. Some examples of these are the "Structural Welding Code-Steel" (AWS,1998) and "Rules and Regulations for the Classification of a Floating Offshore Installation at a Fixed Location" (Lloyd's Register, 1999).

The stress life approach allowed an adequate means of controlling fatigue in design – keep the stresses below the endurance limit by enlarging the dimensions. This approach was not always practical in some industries such as the aerospace industry where weight constraint forced higher utilization of the material. This meant the materials

could be operating above yield in the plastic zone which was ignored by the stress-life approach Consequently, a field of study was initiated under the heading of 'low-cycle' fatigue. Research in the area of low-cycle fatigue accelerated with the commercial availability of closed loop loading systems in the mid-fifties (Ellyin,1997).
Coffin and Manson working independently in the 50's found that plastic strain-life data (*e-N*) could also be linearized on a log-log scale (Bannantine *et al.* ,1990).

Again the plastic strain life curve is approximated by a power curve

$$N(\Delta\varepsilon)^c = \varepsilon_f \tag{24}$$

which is logarithmically transformed. Linear regression is performed on the set of experimental strain – cycles data to determine the exponent *c* and y-intercept ε_f in the same way as above.

This method of applying known strains more accurately models the plasticity of fatigue cracking. It is not as well suited to elastic fatigue but recall stress and strain are related by the material's stress-strain diagram which implies:

$$\varepsilon = \frac{\sigma}{E} \tag{25}$$

The total strain is made up of a recoverable linear elastic strain (ε_e) for stresses less than yield and a non-recoverable non-linear plastic strain (ε_p) for stresses greater than yield.

$$\varepsilon = \varepsilon_e + \varepsilon_p. \tag{26}$$

Substituting from the power curve approximations for the stress-life and strain-life curves, and using the elastic equation to solve for strain ε, we get:

$$\Delta\varepsilon = \frac{1}{E}(\frac{\Delta S_a}{N})^{\frac{1}{m}} + (\frac{\varepsilon_f}{N})^{\frac{1}{c}} \tag{27}$$

This equation, called the strain life relation, combines the low cycle/high load plastic data and high cycle/low load elastic data. Given ΔS_a, m, ε_f and c from experiments and a known strain range, it is possible to determine the cycles to failure (N) numerically.

The strain-life relation curve may be plotted as shown in Figure 2.14 below

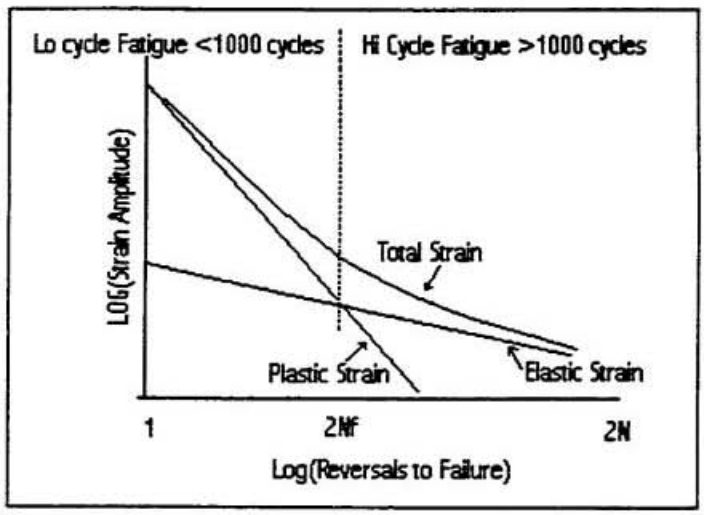


Figure 2.14 Schematic of Strain-Life Relation

As can be seen, the straight S-N approach is overly conservative for low-cycle, high load fatigue but quite accurate for lower elastic stresses. This means, the S-N approach may be used for elastic fatigue analysis whereas the strain-life relation is required at higher stresses where plasticity is important.

A distinct advantage of the strain life method is its ability to deal with variable amplitude loading through improved cumulative "damage" assessment. Cyclic plasticity responses are accounted for, and load sequence effects are reflected in the analysis and results.

Application of the strain life method in its simplest form is to compare the total strain amplitude at a detail to the part of the ε-N curve having the necessary mean stress effects included. The assumption here is that the detail on the part, perhaps in a high constraint area will respond identically to a specimen that is geometrically a smooth bar in plane stress, albeit at the same strain level (Lampman, 1996, p22).

The S-N and ε-N approaches are suitable for analysis of crack free components. Analysis of a cracked member demands a different approach based on fracture mechanics because the singularity introduced by the crack causes concepts such as stress to breakdown at the crack tip.

The third type of basic empirical data is based on fracture mechanics.

Fracture mechanics was first developed by Griffith in the 1920's based on energy considerations. Linear elastic fracture mechanics was developed in the late 1940's and 1950's.

Linear elastic fracture mechanics(LEFM) is used to study the behavior of cracks. It is formulated around the core idea that a crack will propagate by itself if the energy transfer from loss of strain energy outside the crack region to energy absorbed in crack tip plastic deformation and crack surface formation is positive.

Irwin [1957] showed the local stresses near the tip of a crack may be expressed in terms of an asymptotic expansion in polar coordinates centered on the crack tip.

$$\sigma_{ij} = \frac{K}{\sqrt{2\pi r}} f_{ij}(\theta) + \ldots\ldots \quad (28)$$

K is the stress intensity function which defines the magnitude of the local stresses around the crack tip. Irwin related these stresses to energy (based on linear stress strain relationship) and showed cracks would propagate when a critical rate of strain energy release (or equivalently a critical level of stress intensity *K*) was reached. The stress intensity factor may be generally expressed by:

$$K = f(g)\sigma\sqrt{\pi a} \quad (29)$$

Where σ = the nominal stress in the section obtained from external loading;
a = crack length, assumed or given;
$f(g)$ = geometrical correction factor, calculated from tables.

Every steel specimen has a maximum value of stress intensity it can withstand. This is a material property and is determined through crack tip opening displacement (CTOD) tests. Two types of fracture toughness are specified (Barnantine *et al.* ,1990):

K_c is the plane stress fracture toughness for thin sections

K_{Ic} is the plane strain fracture toughness for thicker sections with triaxial stresses

When a component containing a crack is subjected to cyclic loading, the crack length (a) increases with the number of fatigue cycles(N). Load amplitude(ΔP), load ratio R and cyclic frequency ν are held constant.

Crack propagation data obtained experimentally can be plotted as crack length a vs. number of cycles N of stress range $\Delta\sigma$ at which the crack was measured (Figure 2.15). $\Delta\sigma$ can be related to ΔK using equation (29) and the crack growth rate da/dN obtained by numerically taking the derivative. The results may then be plotted as shown in Figure 2.15

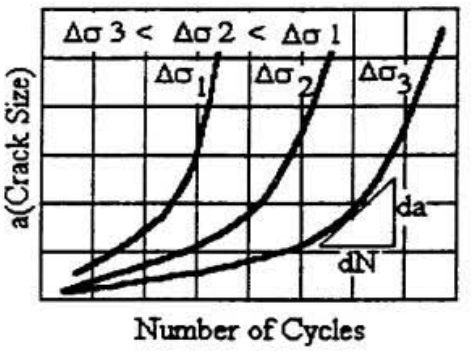


Figure 2.15: Crack Growth vs Stress Range(Bannantine et al.,1990)

Three regions of crack growth can be observed from the fatigue crack growth rate data as shown in Figure 2.16 below.

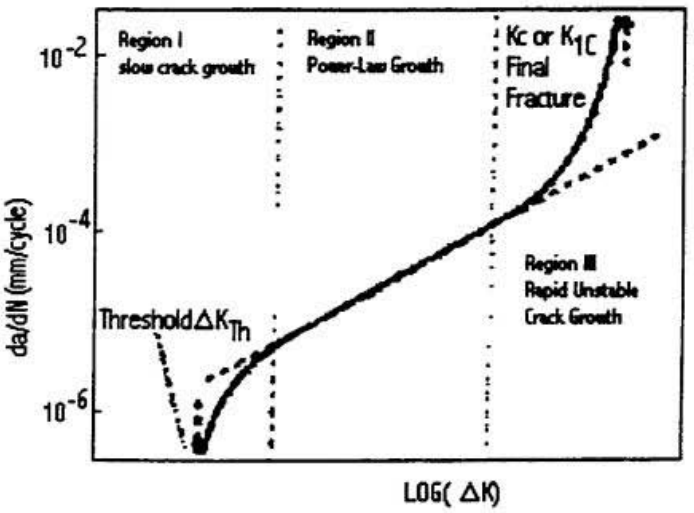


Figure 2.16: Crack Growth Regimes(Bannantine et al., 1990)

Based on analysis of such data in 1963, Paris and Erdogan proposed the following crack propagation model for Stage II crack growth (Bannantine *et al.*,1990):

$$\frac{da}{dN} = C(\Delta K)^n \tag{30}$$

Where C and n are determined empirically from the data. K is the stress intensity factor.

2.9 Factors Affecting Fatigue

Fatigue test data is generally obtained using machined specimens under laboratory conditions subjected to ideal sinusoidal stress histories. This makes the test data well suited to the analysis of machine components.

Real structures are welded assemblies which approximate a design and are subjected to random loadings. The differences in loading, surface condition, residual stresses and welding make the straight application of test data difficult.

2.9.1 Mean and Residual Stresses

Compressive residual stress will prevent crack development and improve fatigue life by opposing tensile loads tending to open cracks.

Conversely, tensile mean stresses when super-imposed on an alternating stress will lower the fatigue life. This follows from the effect of the mean stress in raising strain energy, facilitating the accumulation of plastic strain damage.

A typical contour plot was developed in this work to show the pattern of fatigue life variation with mean and alternating stress and is shown in the figure below.

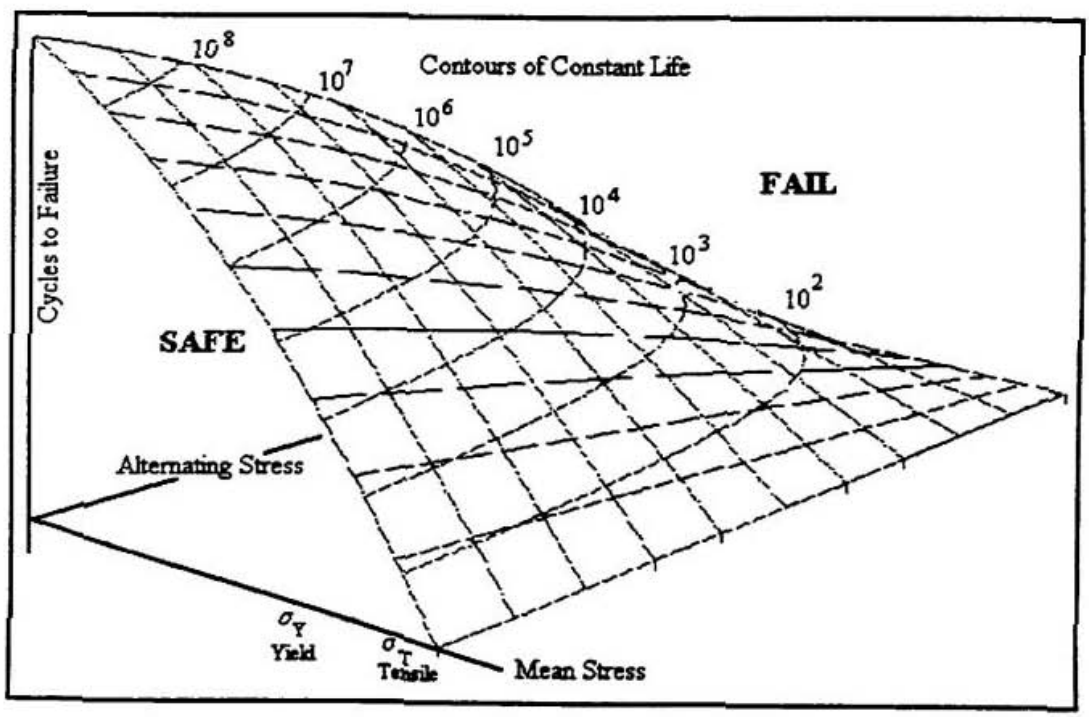


Figure 2.17: Variation of Fatigue Life with Mean and Alternating Stress

Approximations to the SAFE zone for a chosen number of cycles are made using interaction formulae; two common ones being: Soderberg and Goodman (Bannantine *et al.*, 1990):

$$\text{Soderberg: } \frac{\sigma_a}{\sigma_e} + \frac{\sigma_m}{\sigma_Y} = 1 \tag{31}$$

$$\text{Goodman: } \frac{\sigma_a}{\sigma_e} + \frac{\sigma_m}{\sigma_T} = 1 \tag{32}$$

Where: σ_Y, σ_T are the yield and tensile strengths respectively, σ_e is the endurance limit and σ_m, σ_a are the mean and alternating stresses respectively.

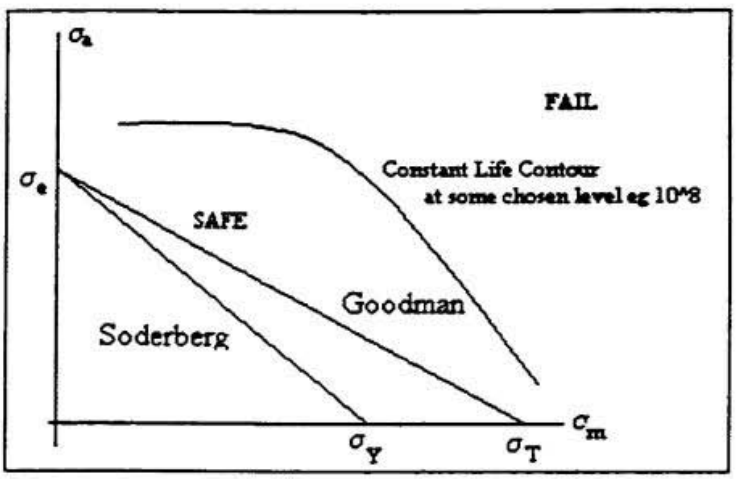


Figure 2.18: Schematic of Fatigue Interaction Formulae

2.9.2 Load Sequence Effects

Fatigue failure is a progressive process. At any time in the specimen's life, a threshold value of stress-strain must be reached to induce a packet of fatigue damage.

By examination of the fatigue process chart (Figure 2.10) the following is noted:

1) A certain stress is required to cause damage

2) The nominal hotspot stress is multiplied by a fatigue notch damage factor(K_f).

3) K_f is initially assumed to be one and is subsequently incremented by an amount proportional to the damage caused.

It is noted the stress is weighted by the fatigue notch damage factor. Given a high K_f, all stresses are significant in causing fatigue; given a low K_f, only the highest stresses are significant.

Consequently, load sequences are crucial. For example, consider two sequences, identical in the size and number of stress cycles but with one having a peak early in its life. The one with the early maxima will have a shorter life than the other because K_f is prematurely incremented causing a greater portion of the stress history to be active.

This affects the ratio of crack initiation period: crack propagation period. In structures with randomly occurring high loads, crack initiation could be caused much early than in an equivalent machine specimen.

2.9.3 Crack Closure

Local plasticity occurs at the tip of a propagating crack resulting in a slightly compressive residual stress, which retards crack growth (Xu,1997).

This is applicable only to the fracture mechanics fatigue model, which is not being used here and will not be expanded on.

2.9.4 Welding

Welding is a very complex process leading to a multitude of complications.

Welding may be defined as the fusing of metals. Many processes exist. The American Welding Society identifies approximately 50 methods of welding (Cary, 1994) but the most common use an electric arc in a variety of settings.

A weld consists of three distinct zones: parent metal, heat affected zone and weld metal. No weld is perfect. Defects can be categorized as :

non-metallic inclusions (slag/tungsten);

voids (isolated, cluster, wormhole, craters);

lack of side wall fusion (linear);

cracks (linear, chevron);

profile (overlap, undercut, incomplete penetration)

Any of these serve as stress raisers and can be the source of a fatigue crack. Even if the weld is perfect, during the thermal melting/cooling cycle, residual stress can be locked in. Many researchers report these residual stresses can be of the same order as the yield strength of the base metal (Riggs,1979).Also, due to shrinkage of the weld deposit, further residual stresses can be locked in which can exaggerate fatigue failure.

Weld effects are normally accounted for empirically by using fatigue data from welded test pieces.

2.9.5 Surface Condition

Because most fatigue cracks occur at or near the surface, there is a high sensitivity to surface condition.

Case hardening can help improve fatigue life because most case hardening procedures involve the high temperature solid state diffusion of nitrogen or carbon into the surface layers. This causes a slight increase in the volume of the surface layer which puts the surface into compression and increases fatigue life (Bannantine *et al.* 1990).

2.9.6 Corrosion

Corrosion is the oxidization of exposed surfaces of steel. The oxidization process is accelerated in the presence of seawater. Iron is oxidized over the surface to form iron oxide. The oxide is porous allowing water to penetrate and cause deeper degradation. This oxide is brittle and will crack or flake off when subjected to strain In hotspots subjected to high cyclic straining, the oxide is repeatedly cracked off leading to fresh exposure of steel. This steel is then oxided and the process begins again with the next cycle. This process leads to stress corrosion cracking which is of great importance but unfortunately extremely complex.

No analytical means of adequately predicting stress corrosion cracking exists (Lampman,1997). Cook reports as much as a 50% drop in the fatigue life of an immersed specimen over that of a specimen in air. Hughes suggests a failure ratio of 0.3 for locations that are maintainable and 0.1 for locations where maintenance and inspection are impossible.

2.9.7 Load Shedding

As a crack progresses, the stress profile across the cross section can be found using fracture mechanics. Integrating the stress over the cross sectional area yields the net load carried by the member. Results show a decreasing capacity to carry load.

In non-redundant structures, this results in an acceleration of the failure process. In redundant structures, adjacent members will take up some of the load thereby accelerating their failure and slowing the failure of the original.

2.10 Damage Accumulation Models

Given a stress history, it is then analyzed by one of the methods in section 2.1 to yield equivalent blocks of uniform cyclic loading comparable to experimental set-ups. These must be combined to predict actual damage.

Figure 2.19, developed in this work, summarizes the damage accumulation approach. A stress histogram is obtained through deterministic or semi-probabilistic methods. The number of cycles at each stress interval is determined based on the distribution of the loads. The number of cycles to failure at each stress interval is determined from the S-N curve. The damage is then summed up.

2.10.1 Summation Methods

Linear Methods – Miner's Rule

The accumulation of damage is assumed linear and independent and identically distributed. In other words, no consideration is given to load sequence effects.

Failure is predicted to occur when damage is greater than the failure ratio η :

$$\Sigma \frac{n_i}{N_i} \geq \eta \qquad (33)$$

Where : n_i = number of stress cycles at a given stress σ_i;
N_i = number of cycles to cause failure at stress σ_i.

The choice of η is somewhat arbitrary but following the lead of Hughes (1983) a value of 0.3 is used.

This is by far the most common accumulation model and is widely accepted in most codes and standards. " For most situations where there is a pseudo-random load history, Miner's rule is adequate for predicting fatigue life"(Banantine *et al.*, 1990 ,p182).

There are drawbacks however. It does not account for load sequence events and applies equal weighting to all loads. For elastic S-N analysis, this is acceptable for high cycle fatigue but can be optimistic for low cycle fatigue where the accumulated damage can be much greater than the cycle ratio n_i/N_i. As can be seen from Figure 2.14, for high cycle – low stress fatigue, the elastic strain curve merges with the total strain curve indicating elastic analysis is sufficient. For low cycle – high stress fatigue, the elastic curve falls below the total strain curve indicating that a large component of the accumulated damage (ie. the plastic component) is being neglected in the elastic analysis thus leading to an over-estimate of the life.

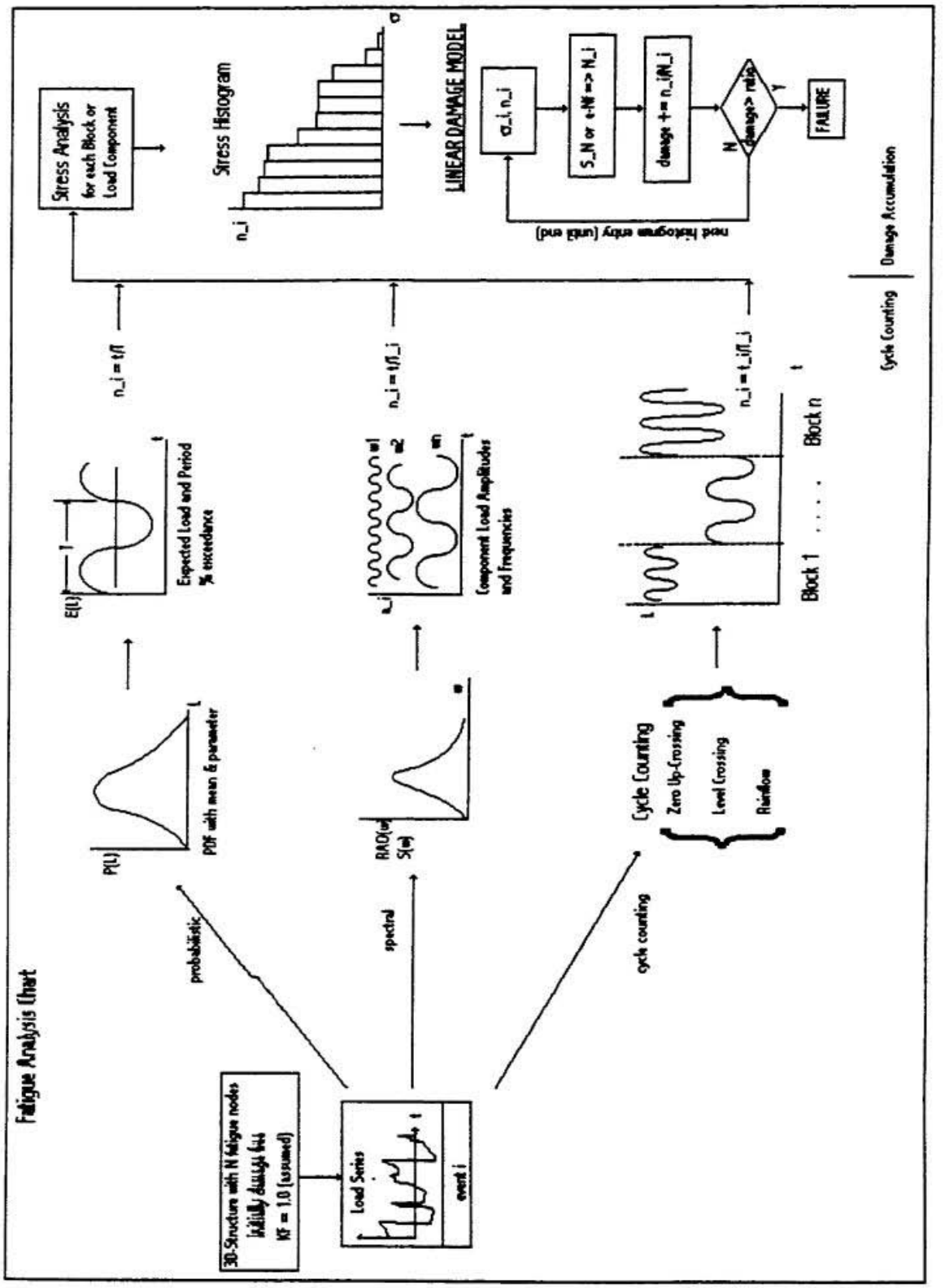


Figure 2.19 Fatigue Analysis Chart

Non-Linear Methods

Bannantine *et al.* (1990) reports several non-linear damage accumulation models have been developed which give better predictions than Miner's Rule but report that in general they suffer drawbacks:

1) they require large amounts of empirical data for material and shaping constants and accounting of load sequencing effects becomes a problem in complex stress histories
2) these methods give better predictions in lab set-ups with simplistic two-step stress-histories but a two step history does not resemble a stochastic history and so no guarantee of their accuracy can be made,

Bannantine *et al.* (1990) presents the following approach which is repeated here for reference purposes:

The damage D per cycle ratio n_i/N_i is assumed proportional to some power P of the cycle ratio:

$$D_i = \left(\frac{n_i}{N_i}\right)^P \tag{34}$$

Other non-linear models exist (Sobczyk and Spencer,1992) but these suffer similar drawbacks and as Bannantine reports, under general loading conditions offer no better results than the simple linear Miner's Rule. Chen and Shin (CMS,1997,p368) report several studies confirming that non-linear effects can be neglected. They advocate the acceptance of linear theory so linear spectral theory can be used.

Consequently, non-linear methods will not be considered here.

2.10.2 Stochastic Methods

The second approach is to recognize the inherent uncertainty in the fatigue factors and treat them as random variables. Ximenes(1991) reports sources of uncertainty in the context of tension leg platforms as:

1) environmental conditions such as sea scatter, wave spectrum and wave energy spreading
2) wave load modeling
3) structural modeling
4) methods of stress analysis
5) stress concentration factors
6) fatigue damage model
7) fatigue strength

Similar uncertainties exist for ships, meaning the occurrence of a packet of fatigue damage must be considered a random variable.

The most simplistic approach is to determine the root mean square (RMS) stress-strain, treat this as a simple sinusoidal history and apply Miner's Rule. This can under estimate the life due to neglect of peak values. Vaughan (1982) described a modified RMS approach where the RMS value is calculated not for the entire record but for sorted blocks of decreasing stress ranges

Another approach is to treat the stress range in Miner's Rule as a random variable. Recalling the basic S-N relationship:

$$NS_N^m = C \tag{35}$$

and Miner's Rule:

$$\eta = \Sigma \frac{n_i}{N_i} \tag{36}$$

Where:

S = stress range;

m, C = S-N empirical constants;

N= number of cycles to failure at S;

η = failure ratio.

We assume a maximum number of stress events (N_s) the occurrence of stress event s_i follows a certain probability density function $p(s_i)$, from which the number of stress events at s_i can be found:

$$\eta = \Sigma \frac{N_s p(s_i) \Delta s}{C / S_i^m} \tag{37}$$

taking the limit as $\Delta s \to 0$ yields:

$$\eta = \frac{N_s}{C} E(S^m) \tag{38}$$

where the expected stress range S is:

$$E(S^m) = \int_0^\infty s^m p(s) ds \tag{39}$$

Failure occurs when η reaches some value. η is nominally one but as reported by Hughes (1988), a value of 0.3 to 0.7 is not unreasonable depending on the risk and uncertainty in the analysis.

The probability density function *p(s)* is commonly assumed to be a Rayleigh or Weibull distribution whose parameters are chosen to give correlation to measured histories. As an example, assume the long term stress distribution follows a Weibull probability distribution then the expected stress can be shown to be:

$$E(S^m) = S_c^m (\ln N)^{-m/k} \Gamma\left(1+\frac{m}{k}\right) \qquad (40)$$

Again where S_c is some characteristic value of stress range, k is the Weibull shape function, m is the slope of the S-N curve, N is the expected number of cycles and Γ is the gamma function.

The difficulty with the application of statistical methods is determination of appropriate values of the statistical parameters. Ma and Bea (1995) describe a reversed fatigue analysis procedure where the lives of failed details is used to determine these parameters in the context of a repair management program.

Hughes (1988) gives long term stress exceedance plots for several tankers and containers ships. He concludes from these that a k factor of 0.7 to 1.0 is suitable for large tankers and bulk carriers and that for container ships and general cargo ships, a value of 1.0 to 1.3 is suitable. He also presents stress ranges (hull girder bending) at an exceedance probability of 10^{-8} per year for a bulk carrier of 29.5 ksi(206 MPa).

An alternative approach is to assume the stress distribution follows the loading distribution. Since for ships, the primary cyclic load source is wave action, which can be shown to follow a Rayleigh distribution, the stress distribution is assumed to follow a

Rayleigh distribution, the parameters of which are found from wave statistics. This approach is discussed in Hatlestad *et al.* (1979)

Reliability methods can be used to quantify the risk of fatigue failure. These are not pursued here and the interested reader is referred to Sobczyk and Spencer(1992) or the article "Reliability in Fatigue and Fracture Analysis of Ship Structure" by Wirsching and Mansour (CMS,1997).

3.Fatigue Analysis of Side Shell Frame

3.1 General

Generally, a ship of a given hullform carrying a cargo with a given speed, heading and route in a seasate will experience dynamic loads and stresses. The design stresses are exaggerated by local fabrication defects and degradation, leading to the high local stresses responsible for fatigue (see figure 3.1 below).

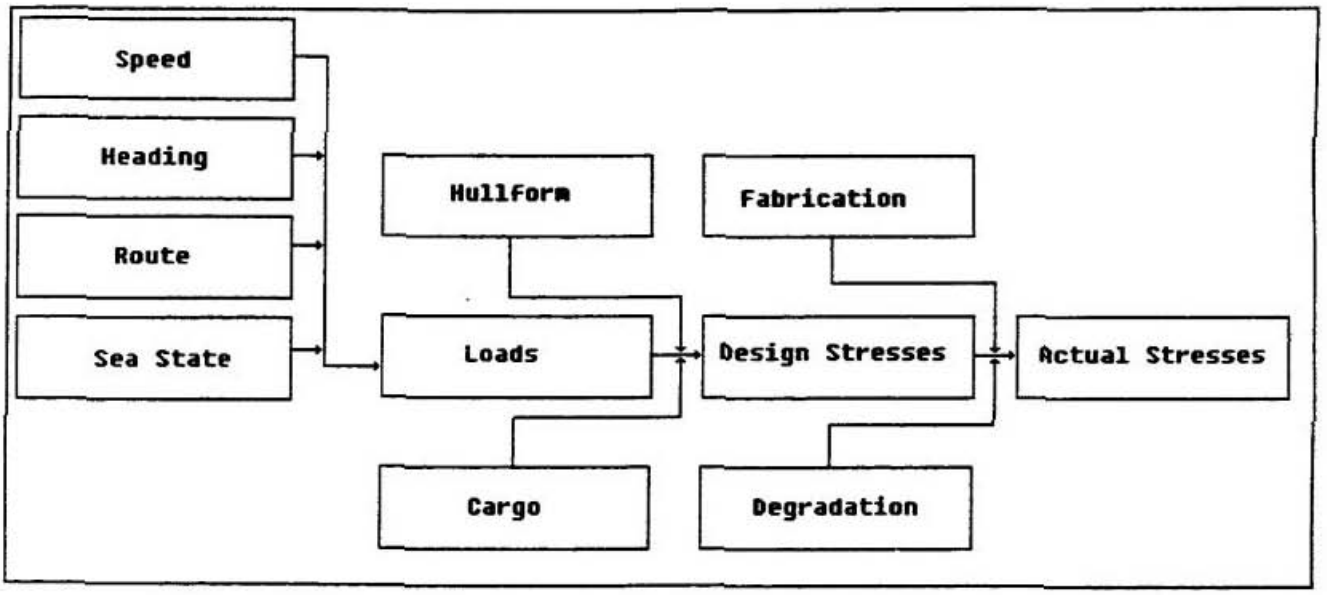


Figure 3.1: Ship Stress Schematic

Speed, Heading, Route and Seastate

The speed and heading are operational parameters and are closely tied to the route and the expected sea-state along a certain route. Route planning is normally done by the owner in preliminary design. Essentially, different areas of the world have different

characteristic wave spectra and the motions of the ship are determined by the speed of advance and heading of the vessel through a sea. This can be customized for specific routes or composites built for travel through successive areas, if known beforehand. This is a complex problem and specific to each chosen route. General approaches involve choosing a wave spectrum and varying the speed and heading of the ship through the waves; the approach used here will be to choose a wave spectrum and then chose a speed and heading which simplifies the analysis. It will be the role of future work to expand on this.

Hullform

The hullform is set during the first stages of design subject to owner requirements, navigation constraints, resistance or stability requirements. The hullform is assumed to be a given here.

The hullform in many ways governs the design of the ship. It sets the hull envelope within which all structure must fit. It defines the resistance characteristics and consequently power and fuel requirements. It controls to some extent the weight distribution.

All ships solve the same set of problems (more or less) and so are similar to each other. Consequently, it is possible to do a study of similar ships and determine average ratios for length/breadth, length/depth as required. This allows determination of parametric formulae for basic ship dimensions as a function of say length. Length is chosen because it is the most convenient measure of the size of a ship.

Cargo

The cargo is determined by the trade. Important parameters are density, total weight and unit load. The total weight of the cargo is the deadweight (DWT) measured in tonnes. It is stored over the length of the cargo hold (L_{HOLD}) leading to a sectional unit load of

$$W_X = (C_{LOAD}/L_{HOLD})DWT \qquad (41)$$

where C_{LOAD} is a parameter depending on the loading pattern.

Two alternative loading patterns exist for bulk cargoes (see figure 3.2 below):

1) alternate hold loading where every second hold is loaded

   for which C_{LOAD} would approach 2.

2) homogeneous hold loading where cargo is evenly distributed.

   for which $C_{LOAD} = 1$

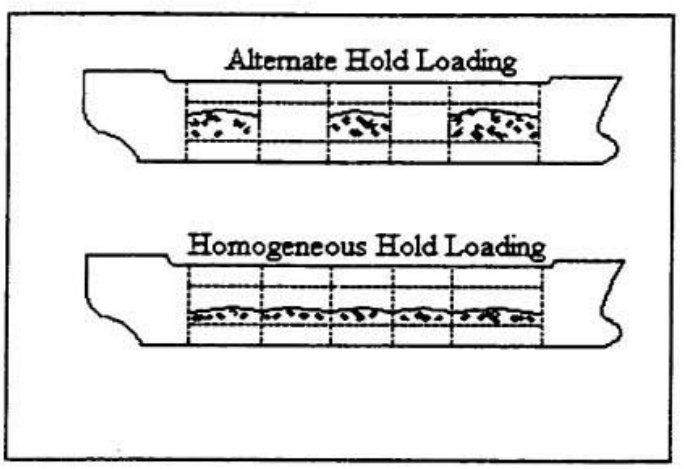


Figure 3.2 : Schematic of Bulk Cargo Loading Patterns

It is anticipated that cargo weight could be significant and will be considered separately in the current study. For the sake of simplicity, it will be assumed the ship is always loaded in the same way and, to be conservative, that it is fully utilized.

The cargo is assumed to be heaped onto the double bottom floor of the hold. It's weight is carried by inner bottom longitudinals to the transverse floors as shear. The floors sag toward the center of the hold. The hopper tanks rotate into the sag carrying the shear directly into the side shell through the turn of bilge strakes and into the side shell frames as a bottom end moment. Towards the end of the holds, near the transverse bulkheads, the inner bottom longitudinals are stiffened by the presence of the stool plate structure and the transverse bulkhead which preferentially absorb the shear load instead of the transverse floors. In this area, there is less sag, resulting in lighter bottom end loads on the side shell frames. It will be assumed that the hold is very long and all the shear loads from the cargo weight are carried up through the side shell frame as a bottom end moment.

Loads

The loads on a ship may be roughly classified as still water and wave effect loads.

Still water loads arise from sinkage of the hullform until the net upward force (buoyancy) equals the net downward weight. In general, sectional weight and buoyancy differ leading to net sectional loads inducing global bending, shear and torsion.

Wave effects are direct modifications of hydrostatic pressure, redistribution of buoyancy due to the wave profile, second order wave diffraction and radiation effects and

induced ship motions. The ship motions cause accelerations of the ship's mass causing inertial loads.

Wave loads are superimposed on the hydrostatic loads. Wave loads cause :

1) instantaneous buoyancy redistribution

2) instantaneous dynamic pressure modification

3) ship motions induced as a result of 1) and 2)

Wave elevation is a stochastic process meaning the load series is stochastic and must be dealt with probabilistically.

Fabrication

The local response (stress-strain) is calculated using design data. In reality fabrication defects may exist as given below:

1) mis-alignments

2) weld undercut

3) notching of brackets during outfitting

4) notching during burning/cutting operations

5) weld defects (inclusions, cracks, lack of fusion, lack of penetration)

These particular fabrication effects are detrimental. The following fabrication effects are beneficial:

1) peening

2) selective grinding of profiles

Welding effects will be implicitly accounted for by using S-N data obtained from welded test pieces, but otherwise, fabrication effects will not be explicitly considered here: it will be assumed the structure is built as designed.

Degradation

Degradation of the structure may occur due to :

1) corrosion

2) damage

These can be very significant. For example as previously discussed, corrosion in association with high stress can lead to stress corrosion cracking. Another example is damage caused by grabs, bulldozers or shakers during unloading operations which can cause notches or plastic damage.

It is very difficult to account for these effects. These are a function of operation, maintenance, inspection and are essentially random damage events. Similar to fabrication defects, these are recognized as important but currently, indefinable. More work is needed.

To account for degradation effects, a lower value of the fatigue failure criteria will be used. It will be assumed here that fatigue failure occurs when the damage ratio η defined in Section 2.8 is greater than 0.3 as recommended by Hughes (1983).

3.2 Physical Description

A typical general arrangement of a bulk carrier showing main dimensions is given in figure 3.3

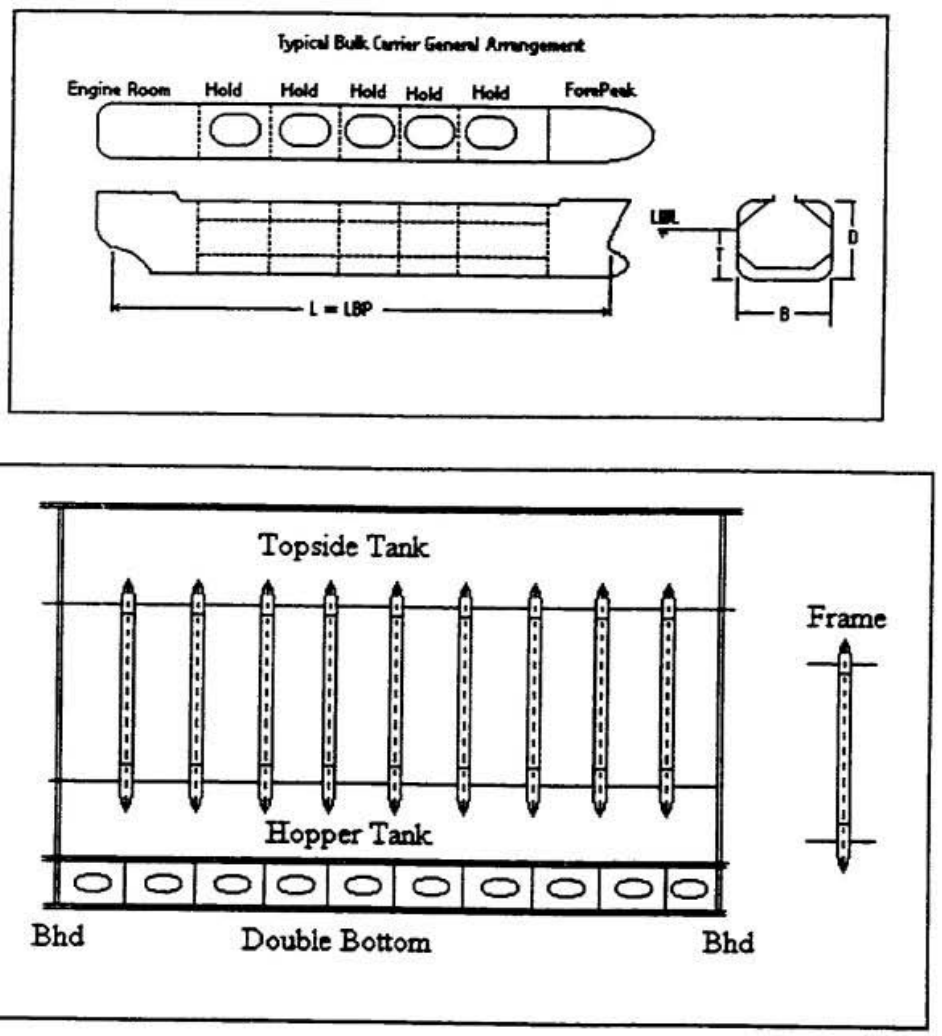


Figure 3.3 : Typical General Arrangement and Side Structure

Generally, a bulk carrier consists of one or more cargo holds amidships. There is a forecastle forward and deckhouse with engineroom aft of the cargo area. The cargo holds are of single-skin, double bottom construction with a sloping hopper tank bottom and sloping topside tank top. The double bottom, hopper and topside tanks double as ballast tanks. Cargo holds are separated by corrugated bulkheads mounted on stool plates.

The side shell is supported by vertical side shell frames. A profile view of the side shell of a hold, looking outboard , is shown in figure 3.3. These frames support the side shell against sea pressure outside and help to carry shear loads from the double bottom to the side shell.

The design of a bulk carrier is fairly uniform despite changes in size. A parametric study was completed using 78 existing similar ship designs (Appendix 1). This parameteric study was intended to create a model of a typical bulk carrier as function of length.

Based on the parametric study, overall dimensions of the ship and frame as a function of length between perpendiculars(LBP) were found as follows:

$$B = 0.152\,L(m) \tag{42}$$

$$D = 0.085\,L(m) \tag{43}$$

$$T = 0.062\,L(m) \tag{44}$$

$$DWT = 0.006\,L^3(tonnes) \tag{45}$$

The frame is developed using the following idealized midship section as shown in figure 3.4.

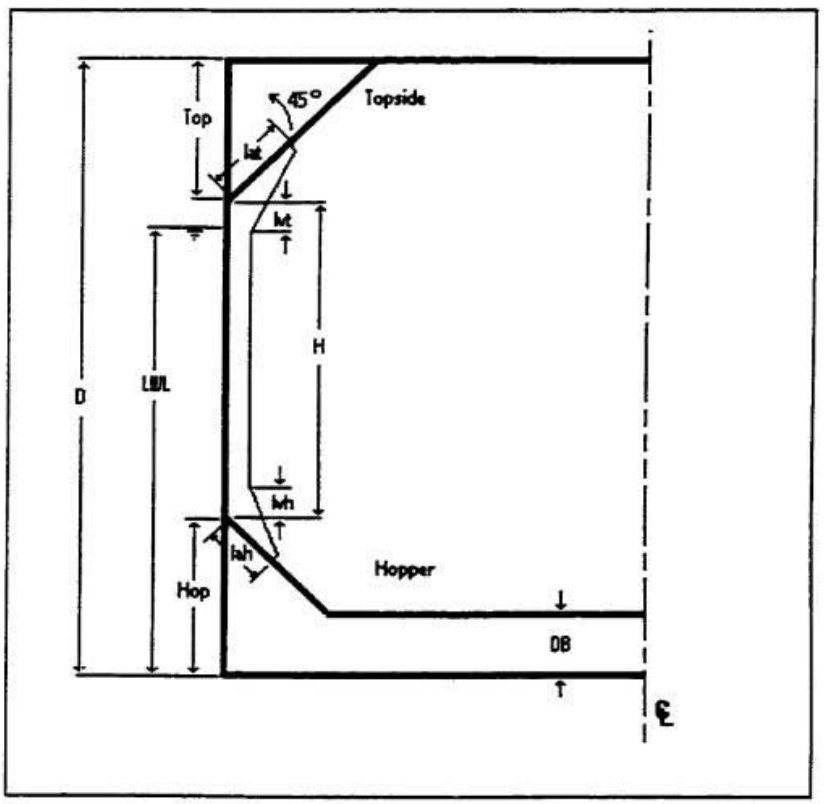


Figure: 3.4 Midship section

All dimensions are measured with respect to the baseline located at the keel. From the parametric study in Appendix 1, the following relationships hold for a "typical" bulk carrier.

$$Hop = 0.0296L(m) \tag{46}$$

$$Top = 0.0615L(m) \tag{47}$$

$$H = D - Top - Hop = 0.0319L \tag{48}$$

Given these basic dimensions, the scantlings for the frame were found by application of Lloyd's Register's Rules for the Classification of Ships (Lloyd's Register,1999) as outlined below and detailed in Appendix 2:

$$l_{ah} = 32.43\sqrt{z}(mm) \tag{49}$$

$$l_{vh} = 27.6\sqrt{z}(mm) \tag{50}$$

$$l_{at} = 30\sqrt{z}(mm) \tag{51}$$

$$l_{vt} = 26.8\sqrt{z}(mm) \tag{52}$$

Where z is the Rule section modulus.

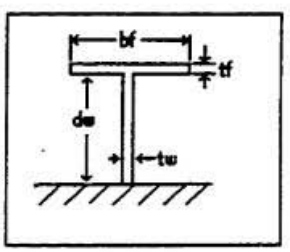


Figure 3.5 Beam profile

The ratios of flange breadth/web depth and flange thickness/web thickness are left as parameters r_1 and r_2.

$$r_1 = \frac{b_f}{d_w} \tag{53}$$

$$r_2 = \frac{t_f}{t_w} \tag{54}$$

$$t_w = \sqrt{L} \tag{55}$$

t_w is the thickness of the web in millimeters.

The following relationship can be found for the moment of inertia of the frame (I_c) :

$$I_c = \left[\frac{1}{12}r_1r_2^3L^{3/2} + \frac{1}{4}\frac{r_1r_2^3L^{3/2}}{1+r_1r_2} + \frac{1}{4}r_1^2r_2^4L^{3/2} + \frac{1}{2}r_1r_2L\right]d_w + \left[\frac{1}{2}\frac{r_1r_2^2L}{1+r_1r_2}\right]d_w^{\,2} + \left[\frac{1}{4}\frac{r_1r_2\sqrt{L}}{1+r_1r_2} + \frac{\sqrt{L}}{3}\right]d_w^{\,3}$$

$$= C_1d_w + C_2d_w^2 + C_3d_w^3 \qquad (56)$$

Where C_1, C_2, and C_3 are coefficients dependant only on the parameters r_1, r_2 and L.

For Rule based design, the frame moment of inertia must be equal to the Rule moment of inertia:

$$I_c = I_R \qquad (57)$$

This can be rewritten as

$$C_1d_w + C_2d_w^2 + C_3d_w^3 - I_R = R(dw, L) = 0 \qquad (58)$$

This was solved for d_w thus completely describing the frame as a function of ship length and chosen web depth.

3.3 Long Term Sea-State Prediction

The bulk carrier was assumed to be operating in an irregular seaway as defined by the ITTC wave spectrum. The spectral density is described by the following formula (Hughes, 1983, p 150)

$$S(\omega|H_s) = \frac{5}{16}\left(\frac{\omega_m}{\omega}\right)^4 \frac{H_s^2}{\omega} e^{-1.25(\omega_m/\omega)^4} \qquad (59)$$

A graph of $S(\omega)$ for a 3m significant wave height over all the modal frequency confidence intervals as defined in Table 3.1 is shown in figure 3.6

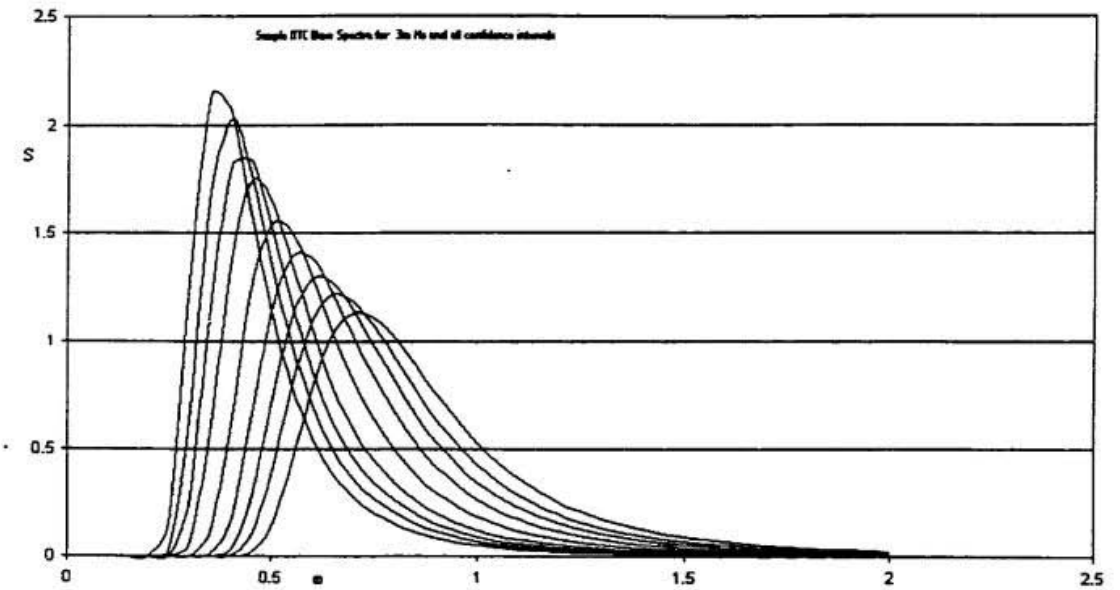


Figure 3.6: ITTC Wave Spectrum

This irregular seastate can be idealised by a histogram of sinusoidal waves with component amplitudes given by:

$$a(\omega) = \sqrt{2S(\omega)\Delta\omega} \qquad (60)$$

with corresponding period

$$T = \frac{2\pi}{\omega} \tag{61}$$

The component waves are simple sinusoidal waves with a time profile given by:

$$h_i(t) = a(\omega)\sin(\omega_i t + \theta) \tag{62}$$

where θ is some random phase angle..

The total seaway is a summation over the entire frequency range:

$$h(t) = \sum_{i=1}^{n} a(\omega_i)\sin(\omega_i t + \theta_i) \tag{63}$$

The ITTC formula gives a family of short term spectra. As Moan and Berge (1997, p329) write, " the long term stress spectrum and associated fatigue damage is calculated by weighting each short term spectrum or corresponding damage with its probability of occurrence". To get the long term distribution of sea-states, the value of *S(ω)* used is a weighted average over all possible significant wave heights and confidence intervals of expected modal frequency:

$$S(\omega) = \sum_{i=1}^{19}(\sum_{j=1}^{9} S(\omega|H_{Si}|\omega_{mj})w_{Fj})w_{H,i} \tag{64}$$

where the parameters are as per the Table 4.2.1 and 4.2.2.

This method of simple weighting by probability of occurrence is based on the assumption that the fatigue damage process is narrow banded. While widely held, this assumption is not strictly valid. Xu (1997) writes "this is generally valid for rigid ships..... but springing induced stresses and wave induced stresses generate a wide banded process"

The loading process and hence the damage process is wide-banded. to some degree. To account for this, work has been done on correlating cycle predictions made using narrow banded assumptions to cycles distilled from actual measurements using rainflow counting procedures. The most well known of these correction factors was first proposed by Wirsching and Light in 1980 (Chen at al, 1996) as:

$$\lambda(\varepsilon, m) = a(m) + [1 - a(m)](1 - \varepsilon)^{b(m)} \tag{65}$$

where,

$a(m) = 0.926 - 0.033m$

$b(m) = 1.5787m - 2.323$

m = slope of S-N curve

ε =spectral bandwidth parameter

$$\varepsilon = \sqrt{1 - \frac{m_2^2}{m_0 m_4}}$$

where m_n is the nth moment of the distribution.

The fatigue damage incurred each cycle is multiplied by the appropriate value of this rainflow correction. This effectively reduces the net damage incurred by each cycle, making the narrow-banded approach more conservative. Consequently, to be conservative, the narrow-banded assumption is used here and a rainflow correction factor not applied.

i	Confidence Interval		ω_{mi}	ω_{Fi}
1	lower ω_m	0.95	0.048(8.75-ln Hs)	0.05
2		0.85	0.054(8.44-ln Hs)	0.05
3		0.75	0.061(8.07-ln Hs)	0.0875
4		0.5	0.069(7.77-ln Hs)	0.1875
5	most probable		0.079(7.63-ln Hs)	0.25
6		0.5	0.099(6.87-ln Hs)	0.1875
7		0.75	0.111(6.67-ln Hs)	0.0875
8		0.85	0.119(6.65-ln Hs)	0.05
9	upper ω_m	0.95	0.134(6.41-ln Hs)	0.05

Table 3.1: ITTC Modal Frequencies and weights(Hughes,1983, p152)

j	H_{sj}	W_{Hsj}	j	H_{sj}	W_{Hsj}
1	<1	0.0503	10	9-10	0.07900
2	1-2	0.2665	11	10-11	0.00540
3	2-3	0.2603	12	11-12	0.00290
4	3-4	0.1757	13	12-13	0.00160
5	4-5	0.1014	14	13-14	0.00074
6	5-6	0.0589	15	14-15	0.00045
7	6-7	0.0346	16	15-16	0.00020
8	7-8	0.0209	17	16-17	0.00012
9	8-9	0.0120	18	>17	0.00009

Table 3.2: Significant Wave Height Distribution(Hughes ,1983, p157)

3.4 Motion Study

3.4.1 Typical Ship Response

A ship responds hydroelastically to the seaway with a combination of rigid body motions and deflection.

Support is provided by buoyant pressure on the ship's bottom distributed over panels supported by stiffeners which carry the pressure to webs as shear. The webs in turn accumulate shear from multiple stiffeners and carry it to the side shell or longitudinal bulkheads. Differential shear with respect to length leads to globally induced bending moments, more or less as predicted by simple beam theory.

Waves complicate the situation by dynamic pressure modifications, inertial effects and redistribution of buoyancy leading to superimposed wave moments and forces.

Full analysis of the preceding is difficult and some degree of approximation is necessary in the analysis and design of ships.

The first order approximation is to recognize resistance to global bending moment in a long wave as the most critical limit state and use empirical formulae to approximate the global bending moment and global torsion. Many such formulae exist and the reader is directed to Hughes (1983) for some examples. These values are often used as characteristic values to scale probabilistic fatigue models. This approach tends to be conservative.

The second order approximation is to analyze the structure at three levels of abstraction. The first is the hull girder as above, based on strip theory and simple beam theory. The second is hull module using the finite element method and results from the hull girder analysis for boundary conditions. The third level is that of local structure using a combination of mechanics and finite element analysis to obtain detailed local hot spot stress distributions. This is the "global-local" approach and serves as the basis for commercial fatigue analysis (CMS, 1997). Xu (1997) reports a dependence on mesh size and recommends the mesh size be on the order of the plate thickness in regions of high stress gradients.

Refinements on the above would involve direct coupling of structural and fluid responses where the hydroelastic behavior of the ship is directly considered. This may involve a coupled finite element/finite element where the fluid is modeled as an acoustic medium. An acoustic medium is compressible meaning the wave equation can be applied. Another approach is to use a coupled boundary element/ finite element approach where the effects of the fluid is reduced to a surface integral using Green's functions.

For this work, the ship was treated as a beam subject to dynamic motions. A full three dimensional hydroelastic analysis was not attempted here due to time constraints and because this is a parametric evaluation of fatigue strength requiring multiple runs which would be very time consuming if the "global-local" or coupled approach were used.

As a beam, the ship responds in three primary modes: bending, shear and torsion. Primary structure is defined here as any structure contributing to the resistance of the beam modes. Secondary structure supports the primary structure and resists local loads

only. Tertiary structure supports secondary structure. Members with multiple load sources default to the classification of the highest load. For example, bottom shell plating resists local hydrostatic pressure but would be classed as primary because it also resists global bending. This classification is primarily intended to distinguish between global and local members. Examples of the application of this classification are given in Table 3.3 below.

Primary	Secondary	Tertiary
shell plating (bending, pressure)	side shell frames	tripping brackets
bottom shell, side shell and deck longitudinals (bending)	short sections of double bottom floors	end brackets
longitudinal bulkheads if continuous (bending)	transverse bulkhead stiffeners	
double bottom plating if continuous (bending)	breast hooks and stringers	
transverse bulkhead plating (shear,torsion)	free standing tanks	
	machinery foundations	

Table 3.3: Structural Classifications

The loads on a member depend on its position in the structural hierarchy. Primary structure is subjected to both local and global loads. For example, bottom shell plating can carry both global bending moments and hydrostatic pressure loads. Secondary structure carries only local loads. For example, a machine foundation supports local machine loads only. Tertiary structure supports secondary structure. For example an end bracket can provide rotational and shear support to a stiffener.

In fatigue analysis we are concerned with local stresses and so it is necessary to establish where the member stands in the ships load paths and determine what loads are relevant and how they influence the member. Of particular importance to fatigue analysis is determination of what Chen and Shin (CMS,1997) term the "dominant" response. Many response modes may be present simultaneously but not all contribute to a particular fatigue problem.

Consider as an example, the cargo hatch corners of a bulk carrier. Obviously, as part of the main deck structure, bending moment is the dominant load. As an other example, the centerline keel brackets in a single skin tanker where vertical shear and bending moment dominate over local hydrostatic pressure.

Other examples are provided by Chen and Shin (CMS,1997) for closed deck structures, open deck structures, side shell longitidunals and bottom shell.

3.4.2 Isolation of Side Shell Frame Response

The problem of interest here is the fatigue strength of lower end brackets in a bulk carrier as a parameter of ship length and cargo loading. It was assumed that the hold was filled with some fixed amount of cargo causing the double bottom to sag toward the middle of the hold. As a result, the hopper tank will tend to sag on moving toward the center of the hold away from the transverse bulkheads and that the bracket is loaded as a result to some unknown static value (see figure 3.7 below).

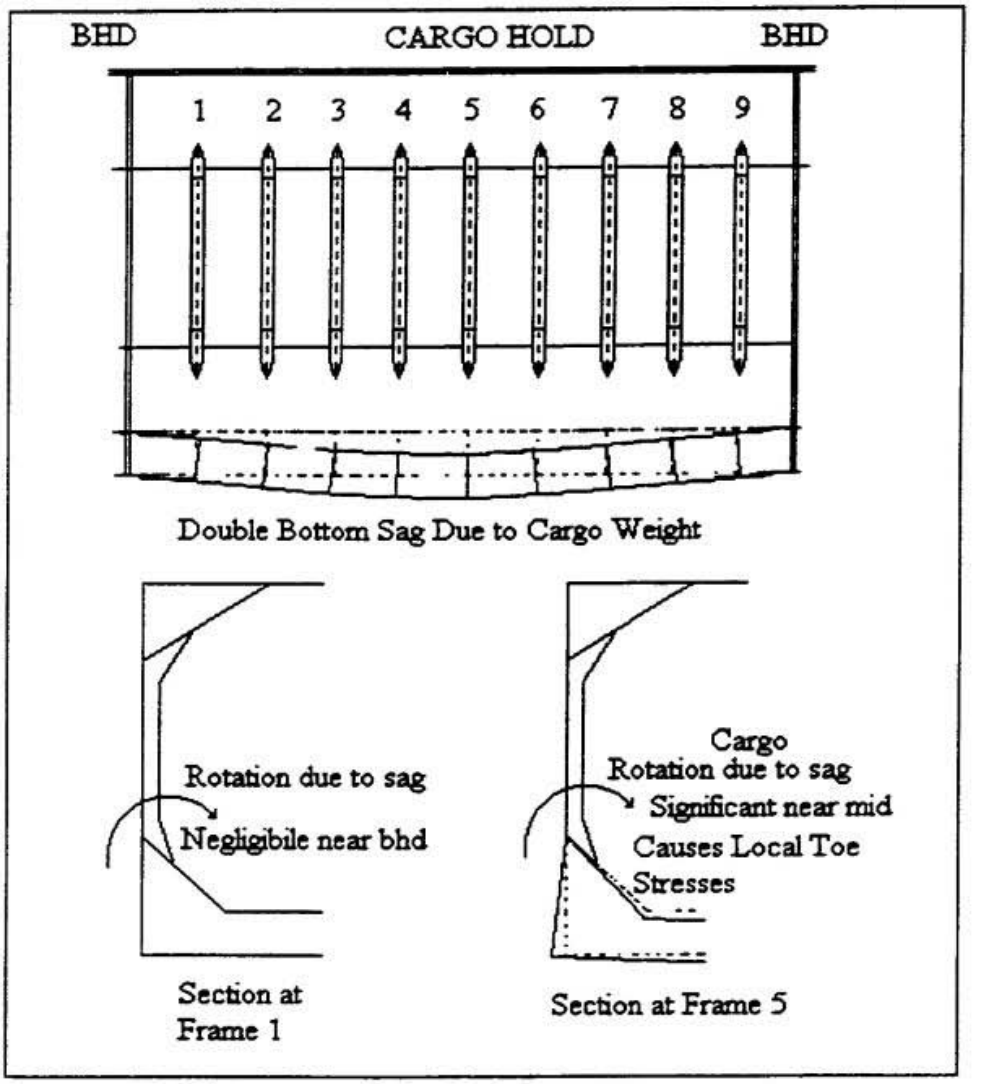


Figure 3.7: Cargo Hold Bottom Sag Due to Cargo Weight

To simplify the above, it was assumed that the particular frame in question was in the middle of a very long hold which was assumed to sag freely. In other words, the effects of the bulkheads or other transverse structure in carrying a part of the cargo weight wa neglected. In reality this is conservative since the bulkheads always have some effect but here it allows the problem to be reduced to two dimensions.

The cyclic loading induced on the bracket as a result of the sea is :

1) wetting of the side shell due to rolling and heaving
2) cargo inertial loads due to roll, pitch and heave accelerations
3) net local changes of buoyancy due to instantaneous heave above/below still water draft.

The wetting of the side shell due to rolling is assumed here to be a second order effect and insignificant compared to the other loads. It will be the role of future work to examine this.

It was assumed that the cargo is stacked in the middle of the hold so inertial loads due to rolling will be minimized because the radial distance between the center of gravity of the cargo and the roll center is minimized. Similarly, assuming the frame in question is amidships, near or at the longitudinal center of floatation, the pitch induced inertial loading is minimized by the same argument. We note, that for frames at either end of the ship, pitch cannot be neglected.

The only remaining effects are heave related for which a transfer function was developed. The present study was restricted to a head sea with zero speed of advance and the ship was allowed to be flexible.

3.4.3 Development of Transfer Functions

The following model (Figure 3.8) of a bulk carrier section was used to develop a heave response transfer function.

It will be assumed, the ship is initially at the still water line (LWL) and subject to an incident wave of amplitude y(t).

The wave induces an instantaneous net positive or negative buoyancy acting on the sectional mass of the ship causing the ship to move relative to the wave. The relative motion is described by z(t). The location of the sectional mass $m(x_s)$, relative to the still water line, is described by the coordinate x(t). This is analogous to the way the movement of a base acts through a spring to induce a movement of a mass in a one degree of freedom system.

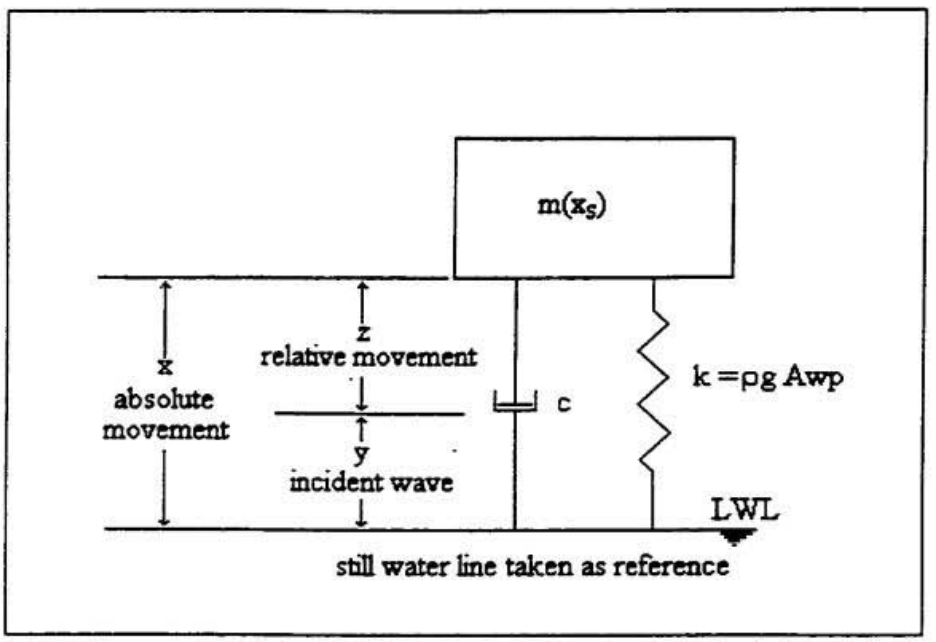


Figure 3.8: Heave Response Schematic

The governing differential differential equation from basic vibrations is :

$$m\frac{d^2x}{dt^2}+c(\frac{dx}{dt}-\frac{dy}{dt})+k(x-y)=0 \qquad (66)$$

where x_s = coordinate of section along length of ship
m = total mass of section at x_s including added mass
c = damping coefficient in kg/sec
k = restoring force in N/m .

Letting:

$$z = x - y \ , \tag{67}$$

then the differential equation can be rewritten in terms of relative coordinates as follows:

$$m\frac{d^2z}{dt^2} + c\frac{dz}{dt} + kz = m\frac{d^2y}{dt^2} = -m\omega^2 a_i \sin\omega_i t \tag{68}$$

Where the right hand side is the wave excitation force. In this derivation, it has been assumed that incident wave elevation effects dominate over diffraction and radiation effects (*i.e.* the Froude-Krylov hypothesis). This is a standard vibration problem (system excited by support motion) with the following well known solution:

$$z(t) = z_h(t) + z_p(t) \tag{69}$$

$z_h(t)$ is the solution of the homogeneous equation ($m\ddot{z} + b\dot{z} + kz = 0$) and is one of three possible cases: underdamped, critically damped or overdamped. All three of these decay exponentially and may be neglected in the steady state.

The particular (forced) solution is given by:

$$z_p(t) = \frac{m\omega^2}{\sqrt{(k - m\omega^2)^2 + (c\omega)^2}} a_i \sin(\omega t - \phi) = RAO_z a_i \sin(\omega t - \phi) \tag{70}$$

where the phase angle ϕ is given by:

$$\phi = \tan^{-1}\left(\frac{c\omega}{k - m\omega^2}\right) \tag{71}$$

Recalling , $z = x\text{-}y$, then the absolute movement x of the ship from the still water level due to heave is given by $x = z + y$, we can solve for x(t) as follows:

$$x(t) = \sqrt{\frac{k^2 + (\omega c)^2}{(k - m\omega^2)^2 + (c\omega)^2}} a \sin(\omega t - \psi) \tag{72}$$

where

$$1)\ \tan\psi = \frac{mc\omega^3}{k(k-m\omega^2)+(\omega c)^2} \tag{73}$$

$m(x_s)$ = mass of the cross sectional area + added mass
$= m_{sx} + m_a$.

$$m_{sx} = 9.47L^2C_{LOAD}\delta L \quad \text{from before} \tag{74}$$

The added mass and damping coefficients are obtained from curves given in Newman (1977) and shown in Figure 3.9 and 3.10 respectively

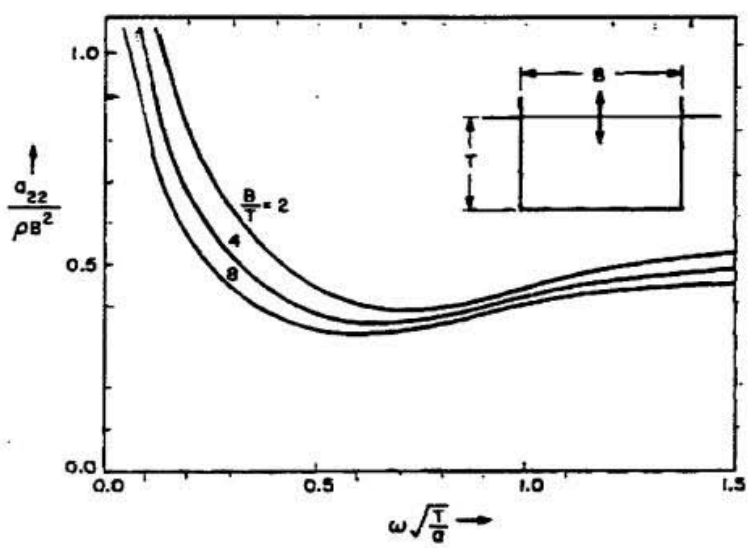


Figure 3.9: Heaving Right Cylinder Added Mass Coefficient(Newman,1977,p299)

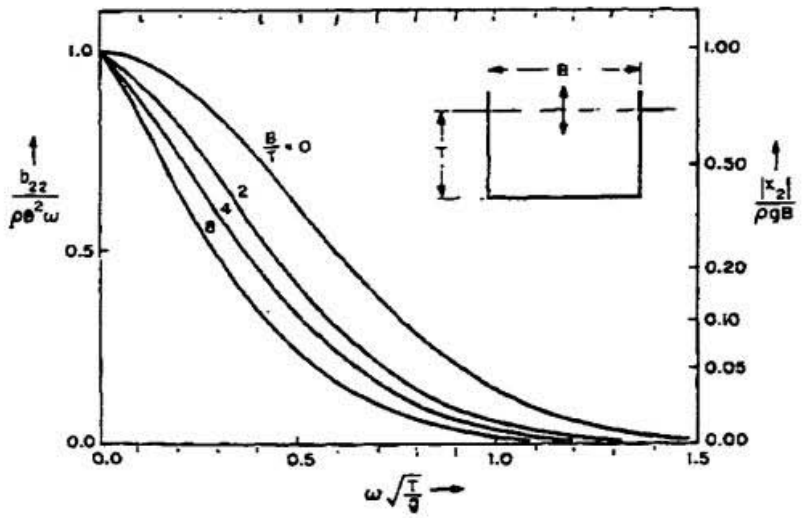


Figure 3.10: Heaving Right Cylinder Damping Coefficients(Newman,1977, p299)

The added mass and damping coefficients presented are for a partially submerged right cylinder. The cross section of which is similar to the assumed midship section.

The curves are given for different values of *B/T*. From the parametric study we can determine *B/T* is fixed at 2.45 regardless of length. There is no curve corresponding directly to 2.45 so the values were interpolated between $B/T = 2$ and $B/T = 4$.

4) k is the restoring force which is proportional to the waterplane area A_{wp}.

$$k = \rho g A_{WP} = (1025)(9.81)(0.152L)(0.7) = 1069.879L \qquad (75)$$

The response amplitude operators are seen to be

$$RAO_x = \sqrt{\frac{k^2 + (\omega c)^2}{(k - m\omega^2)^2 + (c\omega)^2}} \tag{76}$$

$$RAO_z = \frac{m\omega^2}{\sqrt{(k - m\omega^2)^2 + (c\omega)^2}} \tag{77}$$

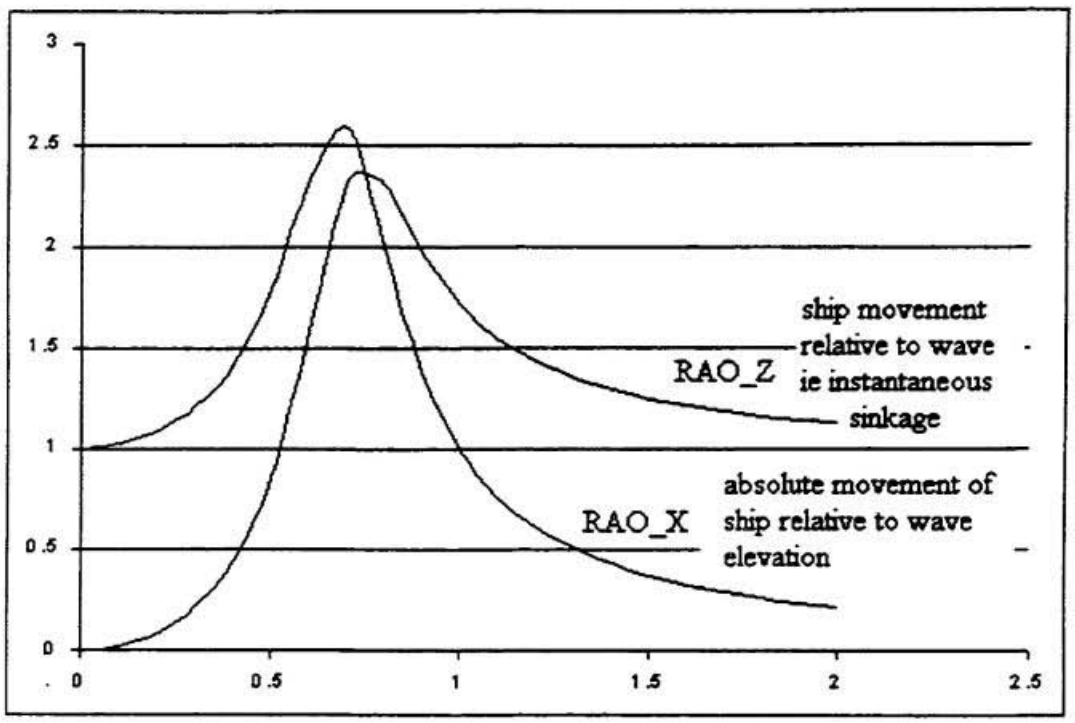


Figure 3.11: Heave Response Amplitude Operators

By examining the figure above we see that at low frequencies, the ship moves with the wave (RAO_X=1.0) leading to zero relative movement (*i.e.* difference between still water load line which is taken as the origin and the wave amplitude is zero). At higher frequencies, the ship does not respond to the wave excitation leading to zero absolute movement and a change in buoyancy directly proportional to the change in wave amplitude as RAO_Z approaches 1.0.

3.5 Formulation of Local Loads

Given the response amplitude operators from the motion study and recalling that the problem can be reduced to a two dimensional frame, the local loads were formulated as follows (schematic in Figure 3.12 below):

1) side shell pressure

2) buoyancy

3) cargo inertial loads.

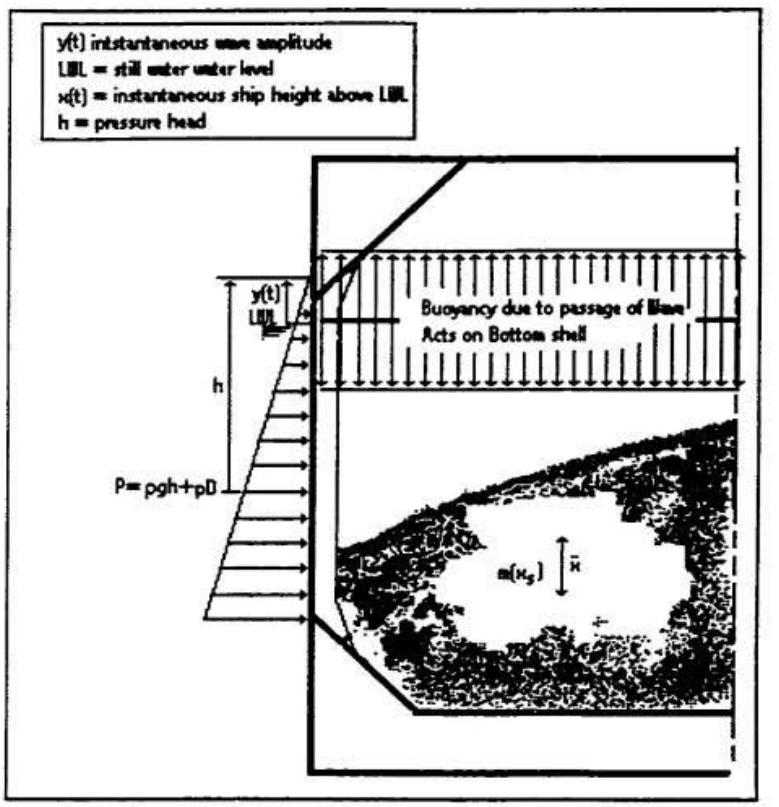


Figure 3.12: Load Schematic

Side Shell Pressure

Side shell pressure pushes directly against the side of the ship. In still water this is simply hydrostatic pressure but in the presence of waves, includes the Smith effect.

Hydrostatic pressure p in N/m^2 was calculated using:

$$p = \rho g h - \rho g a_i e^{-kh} \sin(\omega t) \tag{78}$$

where

ρ = density of salt water = 1025 kg/m^3
g = gravitational acceleration = 9.81 m/sec^2
h = depth in meters from still water line
e^{-kh} = Smith effect correction(Reddy *et al.*,1991)
k = wave number: $k = \frac{4\pi^2}{gT_P^2}$
T_P = wave period (secs)

The pressure distribution in the wave does vary sinusoidally in phase with the wave elevation. To be conservative however, only the maximum value was considered.

Bottom Shell Pressure

In still water it is assumed, since each frame is independent, that the net bottom shell pressure exactly balances the sectional weight, consequently, there is no net load. In a wave, the head changes with the passage of the wave leading to net gains or losses of buoyancy. Applying the Froude-Krylov hypothesis (Reddy,1991), the pressure in the

fluid is calculated as if the ship were not present. Given bottom shell pressure, the buoyant force is simply calculated as pressure multiplied by area:

$$F_B(h,t) = \rho g A_{WP}[h(t) - a_i e^{-kh(t)} \sin(\omega t)] \quad (79)$$

The depth *h(t)* is corrected for heave as per the motion study.

Inertial Loads

The sectional mass will be accelerated due to the heave causing inertial loads proportional to the absolute acceleration. These were calculated as follows:

$$F_I(t) = (m_{ss} + m_a)\ddot{x} = (m_{ss} + m_a) RAO_z \omega_i^2 a_i \sin(\omega_i t - \psi) \quad (80)$$

Net Bottom Load

It was assumed that F_I and F_B are in phase and acting together as a conservative measure. The net force in Newtons is then:

$$F_{NET} = \delta F_B + F_I \quad (81)$$

Net Bottom End Moment

The net moment about the heel of the frame due to this load was obtained by assuming that the cargo distribution followed a triangular profile (similar to Figure 3.12). The center of mass of this pile would then be two thirds of the half breadth. Recalling from the parametric study that $B = 0.152L$ we can write:

$$M_{NET} = \frac{2}{3}(\frac{0.152L}{2})(F_{NET}) = 0.05066L(F_{NET}) \tag{82}$$

3.6 Finite Element Formulation

The model based on the following schematic(Figure 3.13)

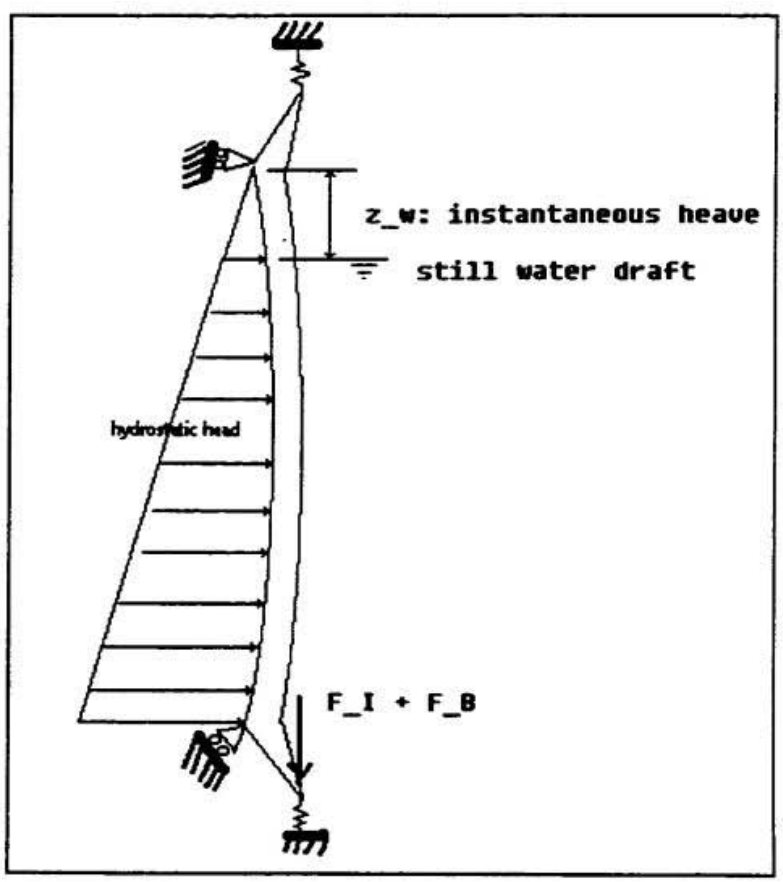


Figure 3.13 Schematic Overview of Frame Model

Given mid-span web depth from section 3.2 and the location of the toe from the assumed section (*i.e.* at end of arm plus 1" nose) , the web depth at any location along the frame can be solved for by linear interpolation. A program was written in C++ to automatically generate the frame for different ship lengths.

The program generates a model (schematic of Generated Model below in Figure 3.14) based on the values of several parameters:

N_{TOP_TOE} = number of elements in top toe

N_{TOP_FLARE} = number of elements in top flare

N_{MID} = number of elements in mid span

N_{BOTTOM_FLARE} = number of elements in bottom flare

N_{BOTTOM_TOE} = number of elements in bottom toe

The frame brackets are supported by welding which was modeled by spring elements distributed over the bracket toes.

Net bottom loading was distributed equally among the bottom bracket nodes.

Net bottom bending moment was applied to the heel node.

The heels of the upper and lower flares, at the point of intersection of the hopper/topside tanks with the side shell were assumed pinned.

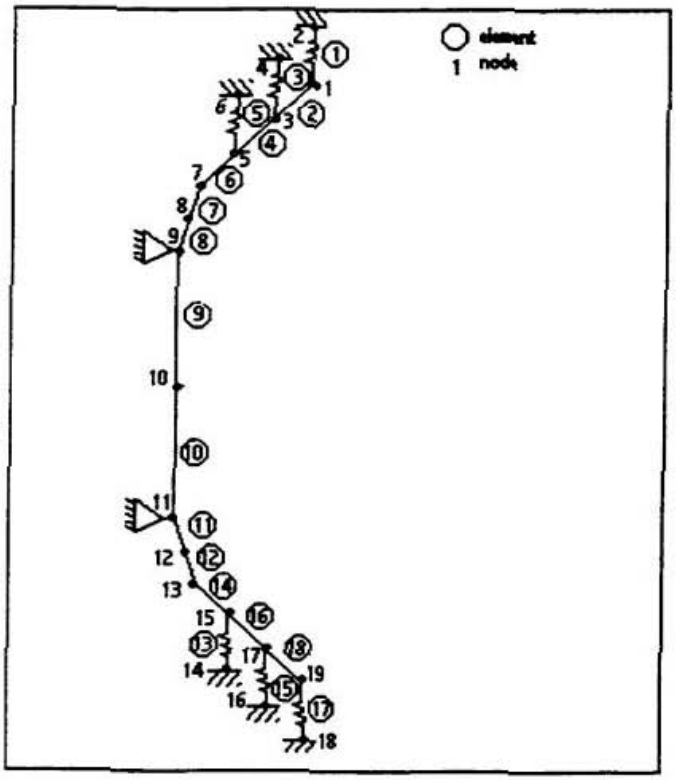


Figure 3.14: Generated Model Schematic (not to scale)

The topside and hopper tanks were assumed rigid. Flexibility of these structures was accounted for by multiplying the stiffness of the weld spring elements by fixity parameters $F_{LO\_TOE\_FIXITY}$ and $F_{HI\_TOE\_FIXITY}$ respectively.

The finite element method was used to solve the above model for displacements and element forces.

We note from Hughes(1983, p281), " the important requirement in modeling beams and panels is to provide an adequate representation of their interaction." Two modes of loading and response are involved:

1) lateral loads and the corresponding beam bending response

2) axial loads in the beam and in-plane loads in the plating which cause corresponding beam bending response."

Hughes (1983) then proposes the use of eccentric or hybrid elements for use in these situations but notes that beam bending and in-plane response are sufficiently independent to be dealt with separately.

In this case, the loading is a combination of lateral pressure and bottom loadings applied to the bracket nodes resulting in an end moment equivalent to hopper tank rotation. These were assumed to dominate over the in-plane loads in the plate and so the frame was modeled as a generalized beam element as shown in Figure 3.15 below.

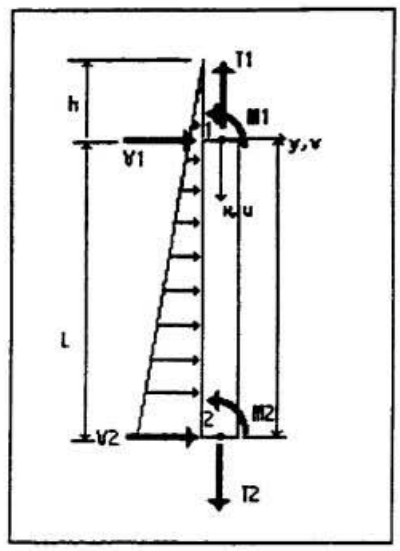


Figure 3.15 Beam Element Definitions

The local element equations were found by direct integration.

$$\begin{Bmatrix} T_1 \\ V_1 \\ M_1 \\ T_2 \\ V_2 \\ M_2 \end{Bmatrix} = \begin{bmatrix} k_1 & & & -k_1 & & \\ & k_2 & k_3 & & -k_2 & k_3 \\ & k_3 & k_4 & & -k_3 & k_5 \\ -k_1 & & & k_1 & & \\ & -k_2 & -k_3 & & k_2 & -k_3 \\ & k_3 & k_5 & & -k_3 & k_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \phi_1 \\ u_2 \\ v_2 \\ \phi_2 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 3/20\rho gL^2 s + 1/2\rho ghLs \\ 1/30\rho gL^3 s + 1/12\rho ghL^2 s \\ 0 \\ 7/20\rho gL^2 s + 1/2\rho ghLs \\ -1/20\rho gL^3 s - 5/12\rho ghL^2 s \end{Bmatrix} \quad (83)$$

where:

$$k_1 = \frac{EA}{L}$$

$$k_2 = \frac{12EI}{L^3}$$

$$k_3 = \frac{6EI}{L^2}$$

$$k_4 = \frac{4EI}{L}$$

$$k_5 = \frac{2EI}{L}$$

s = frame spacing

h, L as per figure

The elastic supports were modeled using simple truss elements as shown in figure 3.16 below:

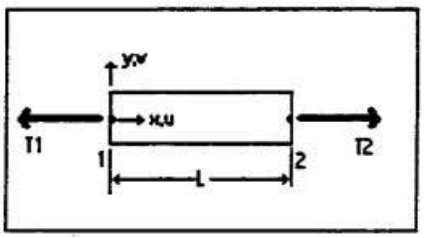


Figure 3.16 : Truss Element

The element equation (in the local coordinate system) is given as:

$$\begin{Bmatrix} T_1 \\ T_2 \end{Bmatrix} = \frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \tag{84}$$

A program was written in C++ to perform the finite element analysis on the model. It is included as Appendix 3. A custom program was written to facilitate automatic frame generation and allow multple runs over a variety of parameters.

3.7 Fatigue Analysis

Fatigue analysis was carried out using the procedure recommended by the US Coast Guard, Ship Structure's Committee (Stambaugh *et al.*, 1992).

This is a "hotspot" S-N approach with a linear Miner's Rule damage accumulation model. It was assumed that the bracket toe most closely corresponds to joint type 21s (Figure 3.17).

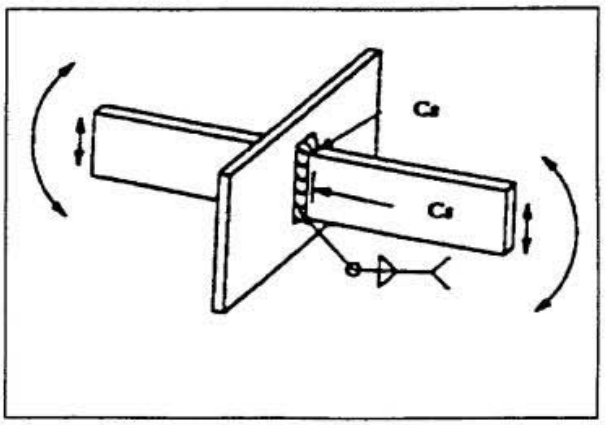


Figure 3.17: Fatigue Category Chosen (Stambaugh et al.,1992)

It should be noted that stress here is defined as shear stress on the weld (*i.e.* immediately adjacent to the weld or hotspot stress).

3.7.1 Toe Stress

The toe stress is the load in the toe support divided by its area. The area is calculated below as the area of welding corresponding to the node of the toe support.

The toe load is taken as the axial tension (T) in the spring support modeling the weld obtained from the finite element analysis.

The area of the spring support was found based on the figure 3.18.

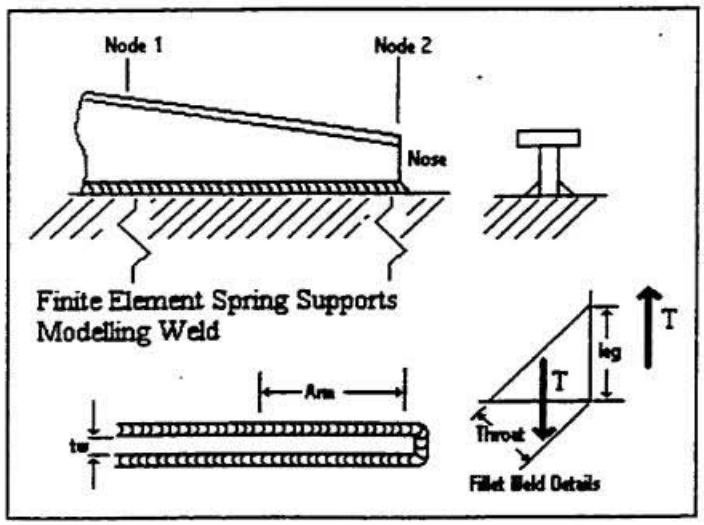


Figure 3.18: Weld Toe Details

The minimum area of the weld is found in way of the throat. Lloyd's Rules for Ships (Lloyd's Register, 1999) , Part 3 Chapter 10 require the throat to be at least *0.34tp*. Here *tp* (the plate thickness of the member being welded) is the thickness of the frame web *tw*.

$$throat = 0.34t_p = 0.34t_w = 0.34\sqrt{L} \tag{85}$$

From which the length of the leg is found to be:

$$leg = \sqrt{2}throat = 0.34\sqrt{2}\sqrt{L} = 0.4808\sqrt{L} \tag{86}$$

The arm length is one half the length of the element

$$arm = \frac{1}{2}\frac{lah}{N_{Bottom\_Toe}} \tag{87}$$

Therefore the area of the weld face attached to the bracket is:

$$A = 2 \times leg \times arm = 2\left(0.4808\sqrt{L}\right)\left(\frac{lah}{2N_{Bottom\_Toe}}\right)$$

$$= 0.4808\left(\frac{lah\sqrt{L}}{N_{Bottom\_Toe}}\right) \tag{88}$$

Finally the required shear stress is :

$$S = \frac{T}{A} \tag{89}$$

3.7.2 Fatigue Analysis

The S-N curve (Figure 3.19) for detail 21s is defined by (Stambaugh *et al.* 1992):

$$\log C = \log \Delta S + m \log N \tag{90}$$

where

C=338.7
m=0.1743

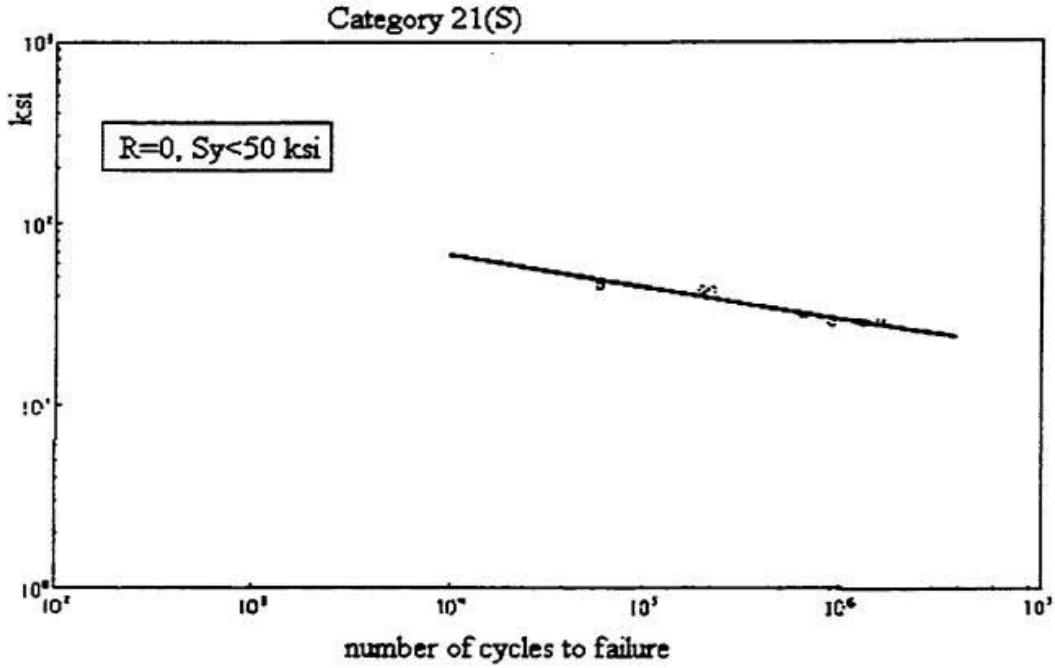


Figure 3.19: S-N Curve for Joint Category 21(s) (Stambaugh et al. 1992)

Therefore the number of cycles N at stress range ΔS wa given by:

$$N = \left(\frac{C}{\Delta S}\right)^{1/m} \tag{91}$$

This is valid for stress ranges with $R = \frac{\sigma_{min}}{\sigma_{max}} = 0$.

$\Delta S_{R\neq 0}$ is related to $\Delta S_{R=0}$ by the following (Stambaugh *et al.* 1992):

$$\Delta S_R = \frac{1+(2N)^b}{1+\frac{1+R}{1-R}(2N)^b}\Delta S_{R=0} = \frac{1+(2N)^b}{1+\frac{1+R}{1-R}(2N)^b}\left(\frac{C}{N^m}\right) \tag{92}$$

where

$$\Delta S_R = \sigma_{\max} - \sigma_{\min}$$

$$b = -\frac{1}{6}\log_2\left(1+\frac{50}{1.5S_u}\right)$$

This equation is of the form $\frac{1}{N^x} - C = 0$

which implies, $\lim_{n\to\infty} = -C$ and it is monotonic implying there is only one zero.

Consequently, a simple bisection method was used to solve for the number of cycles to failure.

4.Results and Discussion

4.1 Finite Element Convergence

Initial runs were performed to establish the rate of convergence of the model as the number of toe elements was increased. As seen from the graph below (Figure 4.1), fatigue life drops quickly and levels out at around 16 elements. Hence 16 was be taken as the required number of toe elements.

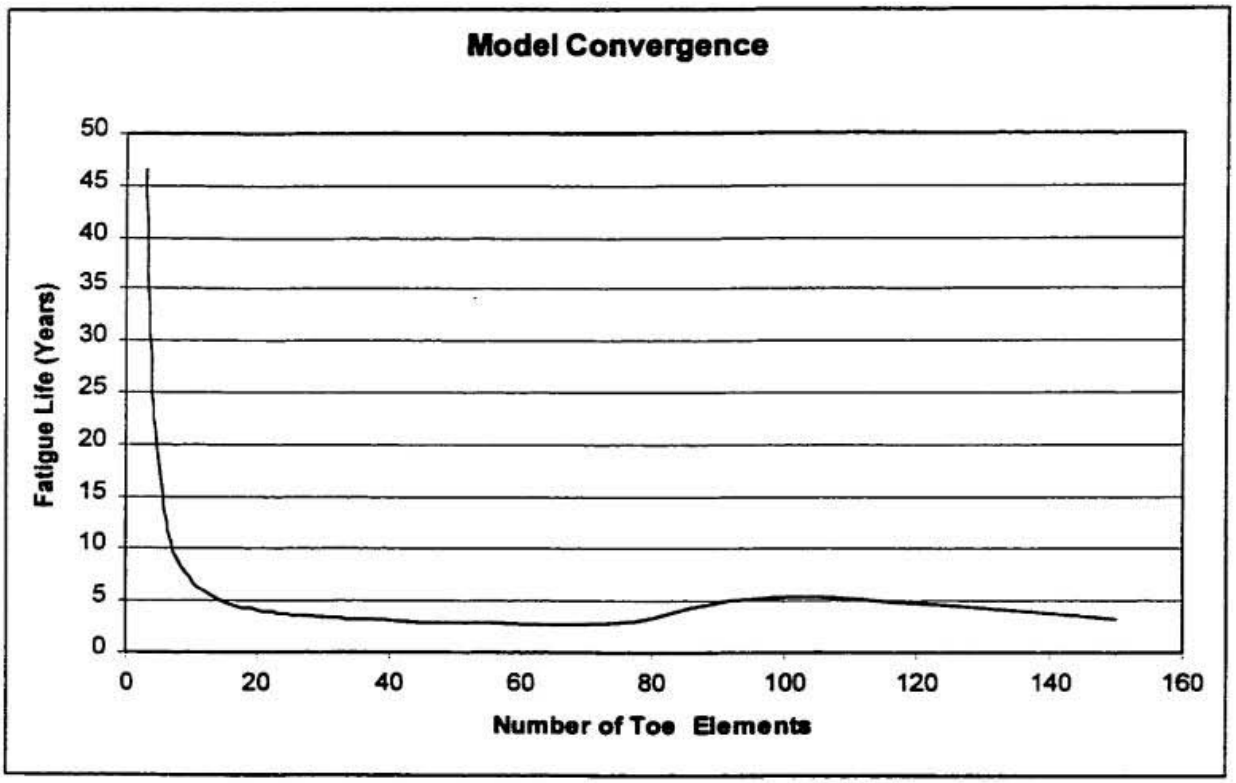


Figure 4.1: Convergence with increasing N_{BOTTOM_TOE}

4.2 End Fixity Effects

The topside and hopper tanks provide some degree of support to the frame but how much is not known and cannot be found by trivial means. As Faulkner and Snyder(1971) write, " One of the difficulties arising in confining the boundaries of the structure to one which is convenient and reasonably simple to analyze is in knowing the boundary constraints and how they may be represented mathematically." The two extremes of boundary constraint are clamped ends and simply supported ends. End fixity is a common measure of where the structure lies between these extremes: end fixity equal to zero implies simply supported; end fixity equal to one implies fixed. To study the sensitivity of fatigue life to end fixity, all parameters were fixed as follows:

1) length between perpendiculars: LBP=100m;

2) loading pattern set to homogeneous: C_{LOAD}=1.0;

3) number of elements in toe of frame: N_{BOTTOM_TOE}=16.

$F_{HI\_TOE\_FIXITY}$ was set and $F_{LO\_TOE\_FIXITY}$ varied 0.4 to 1 (Figure 4.2).

And in turn, $F_{LO\_TOE\_FIXITY}$ was fixed at 0.6 and $F_{HI\_TOE\_FIXITY}$ varied 0 to 1 and results shown on the same figure.

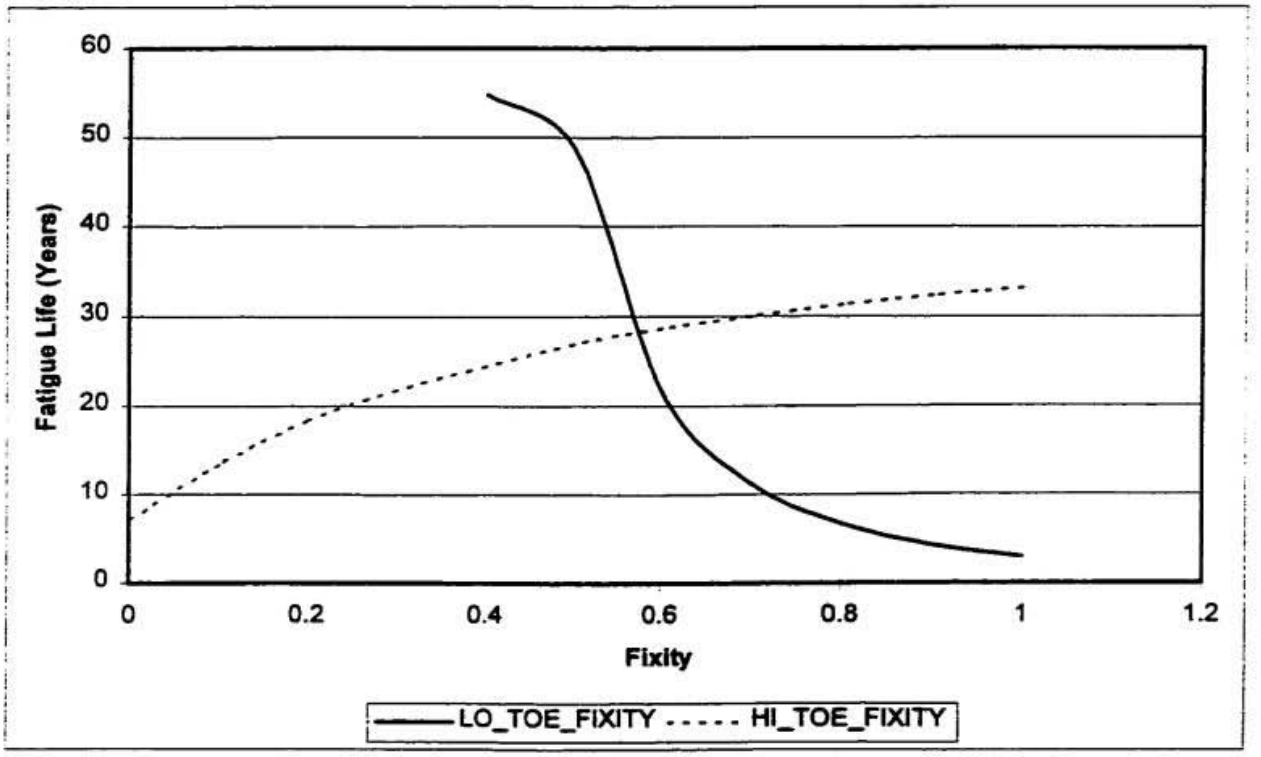


Figure 4.2: Fatigue Life with Variation of FIXITY

End fixity would appear to be important, particularly hopper tank fixity.

The torsional stiffness of all connected members lying in planes perpendicular to the plane of bending was neglected as done by Faulkner and Snyder (1971). The end fixity for the side shell frame is provided by the internal transverse stiffeners of the topside and hopper tank which are assumed to be approximately similar to each other so the degree of rotational stiffness provided to the upper and lower ends of the frame are the same (ie. $R_{AB} = R_{BA} = R$) . It will be assumed that the rotational stiffness R is 6 which Faulkner and Snyder (1971) state is typical for ships' structure.

Faulkner and Snyder (1971) present the following formula for end fixity (f) of a beam A-B with end rotational stiffness (R) subjected to an end moment (M):

$$f = \frac{M_{AB}}{M_{eA}} = \frac{R_{AB}}{N_{AB}}(4 + R_{AB}) \tag{93}$$

where:

M_{AB} = the actual end moment

M_{eA} = the resultant end moment assuming clamped ends

R_{AB} = a non-dimensional rotational stiffness parameter

N_{AB} = a recurring algebraic term, given by Faulkner and Snyder(1971) as

$= 12 + 4R_{AB} + 4R_{BA} + R_{AB}R_{BA}$.

Under the above assumptions, the end fixity is estimated at 0.625 which we note is approximately the point of intersection of the two "variation of fixity" curves.

End fixity here was taken as 0.6 for $F_{LO\_TOE\_FIXITY}$ and $F_{HI\_TOE\_FIXITY}$ corresponding to the point of intersection where the stiffness effects are equal top and bottom.

This is a point for future work as the actual value of end fixity can only be determined by a full three dimensional model using plate elements to model the sag of the entire hold including hopper and topside tanks.

4.3 Fatigue Life Variation with Length

Runs were performed for multiple lengths with N_{BOTTOM_TOE}= 16; C_{LOAD}=1.0; $F_{HI\_TOE\_FIXITY}$ and $F_{LO\_TOE\_FIXITY}$ fixed at 0.6.

Results are summarized in Figure 4.3 below.

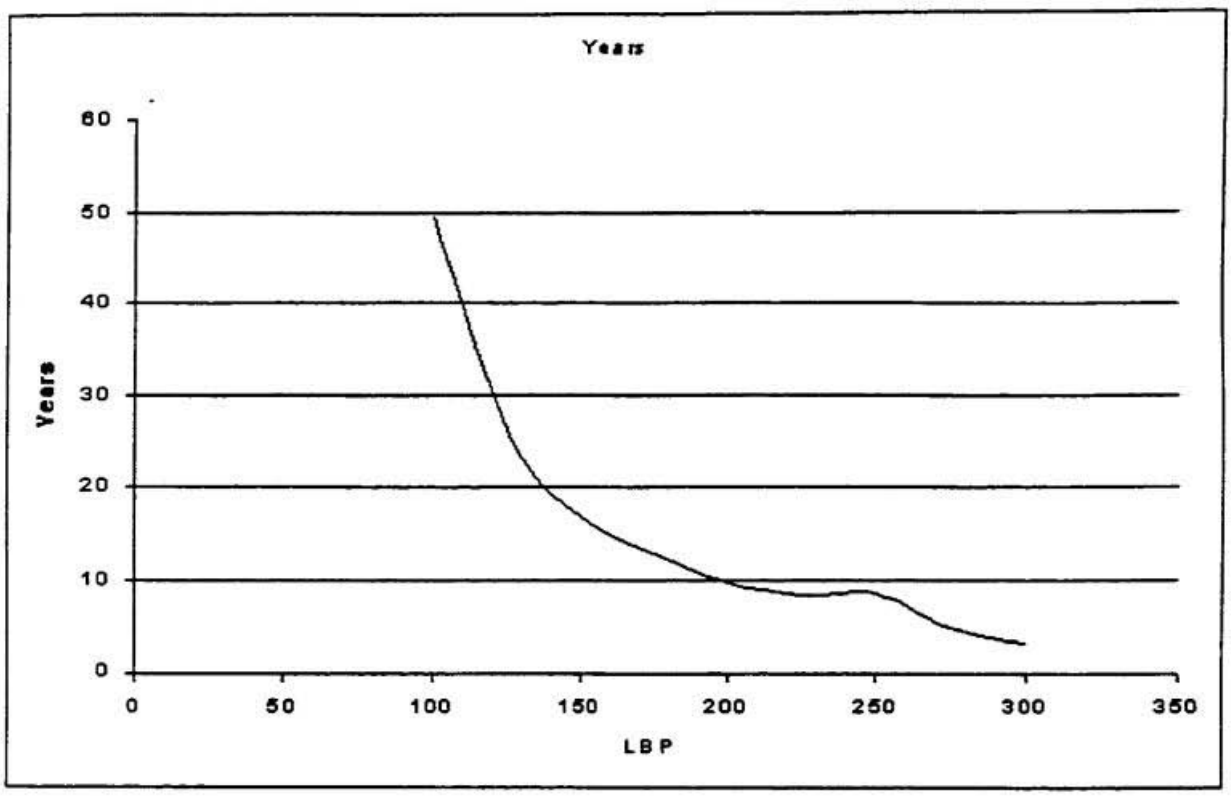


Figure 4.3: Life vs. Ship Length

These results show a quick drop in fatigue life beyond a ship length of 100m, for the toe welds of the lower end brackets, which may appear dramatic.

Life drops to less than 20 years at about 140m which roughly corresponds to the 120m length at which classification societies start to apply stricter construction and surveying requirements on bulk carriers (Lloyd's Register, 1999).

It has long been recognized within the bulk carrier industry that these brackets are a problem and, in the author's experience, these were widely regarded as "nuisance" cracks which did not immediately endanger the ship and which would re-appear after repair. Such an approach is not acceptable today because the progressive nature of the failure leading to disaster is recognized, but it does show that these details do have low fatigue lives.

Also it must be borne in mind, that this is a conservative approach. The end loading is developed on the assumption that the cargo loads are carried entirely by the side shell frames. In addition, the frames were automatically generated in the finite element model using a straight line interpolation from full depth to a 1" nose. In reality the transition would be a concave curve leading to a soft taper; ignoring the taper increases the stress concentration at the nose.

4.4 Fatigue Life Variation with Loading Pattern

Runs were performed for multiple lengths and loading coefficients. End fixities and N_{BOTTOM_TOE} were fixed.

Results are summarized in Figure 4.4 below.

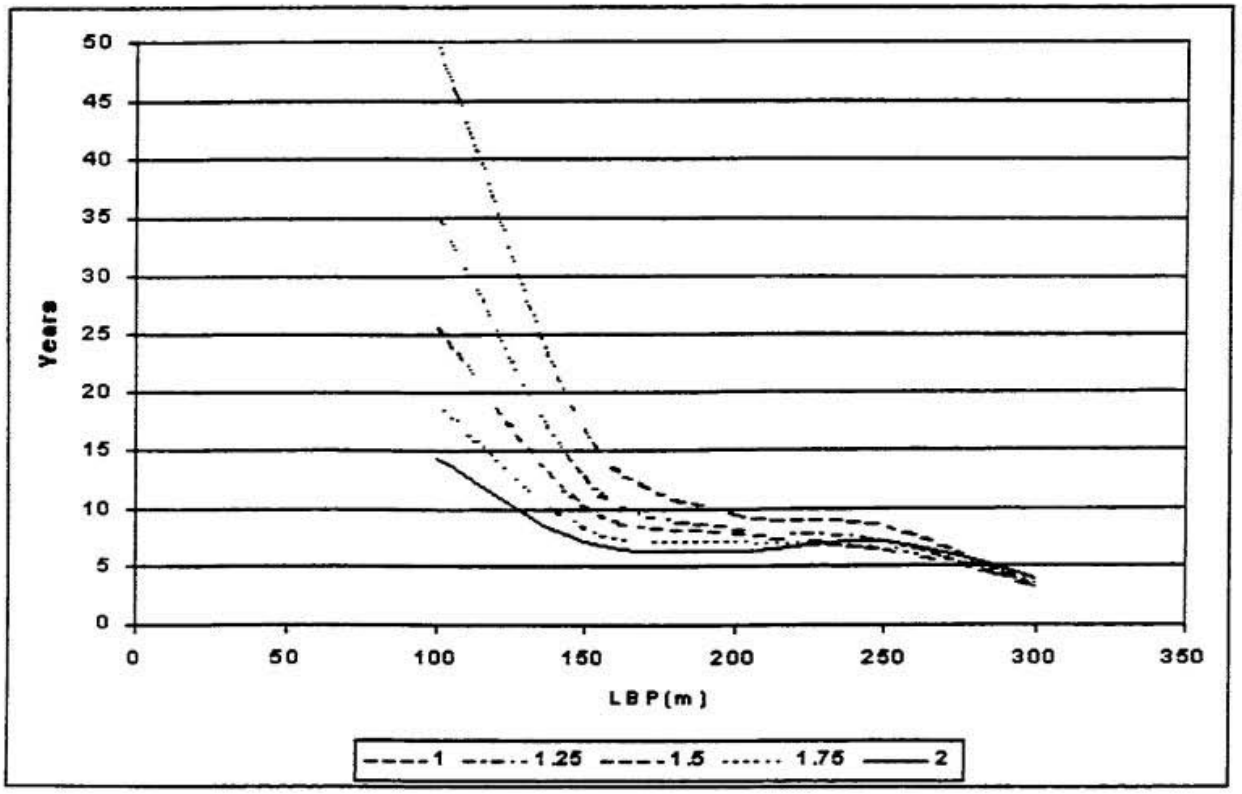


Figure 4.4: Fatigue Life vs. Length for Various Load Coefficients

As can be seen from this graph, increasing the unit cargo load for homogeneous hold loading (C_{LOAD}=1.0) to the maximum of alternate hold loading (C_{LOAD}= 2.0) decreases fatigue life for small ships and very slightly increases fatigue life for very large ships.

For small *LBP* and C_{LOAD}, the heave natural frequency is above the frequency of the most energetic waves in the spectrum. It is noted here that increasing *LBP* and C_{LOAD} have the effect of shifting the heave natural frequency down towards the frequency of the wave spectrum peak leading to increased dynamic resonance effects and loss of life span (Figure 4.5). As *LBP* increases beyond 175m to 250m and C_{LOAD} increases above 1.75 they combine to force the natural frequency below the frequency of the wave spectrum

peak for long ships and alternate hold loading. This leads to a reduction of dynamic resonance effects with a plateau effect for homogeneous loading and a slight increase in fatigue life for highly loaded ships. Further increases in *LBP* beyond 250m show decreasing fatigue life regardless of C_{LOAD}. In this range, the natural heave frequency is pushed well below the wave spectrum peak and the ship stops responding to the seaway. In this case, the full effect of changes in cyclic buoyant loading which dominates over cargo inertial loads is felt with the resultant drop in fatigue life beyond 290m. It should be noted that in this range, alternate hold loading provides a higher fatigue life for the lower bracket toe.

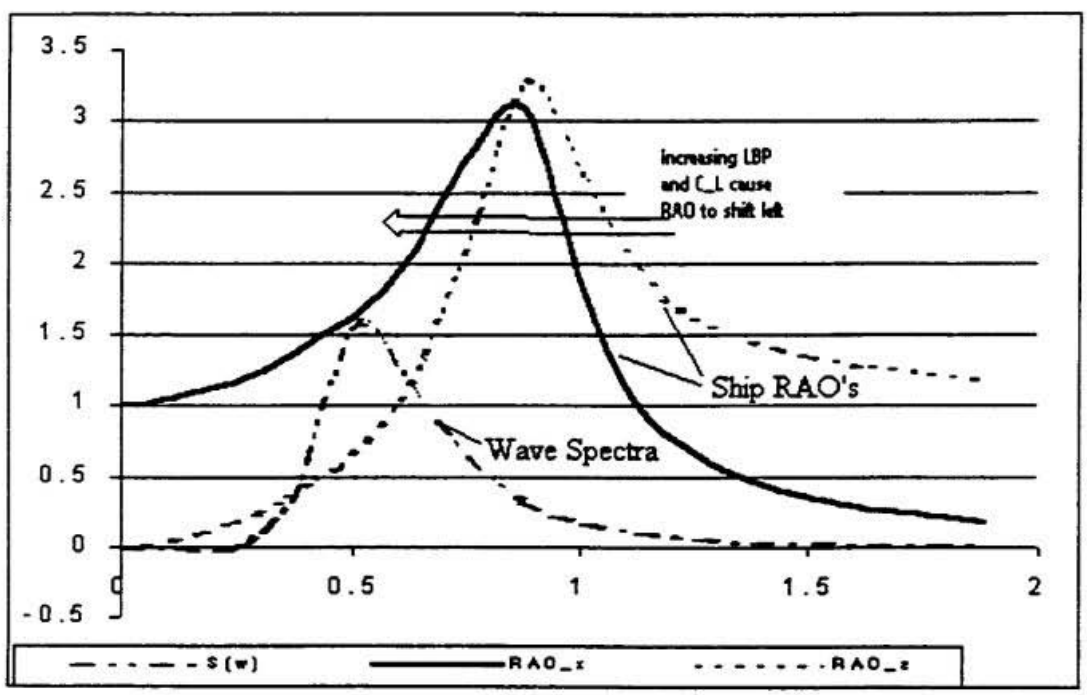


Figure 4.5: Shift of Natural Frequency Relative to Spectrum

5.Conclusions and Recommendations

Conclusions

This paper, by focusing on a narrow problem (the fatigue life of bulk carrier side shell frame lower end bracket toes) was able to identify potentially useful relationships as a function of ship length and highlight some other important parameters for further study.

These frames were modelled as simple two dimensional beams subjected to end loads and moments due to cargo inertia and variations in bottom shell pressure due to the passage of a wave.

End fixity of the frame was seen to have a significant effect on the calculated fatigue lives, but is difficult to determine. A value of 0.6 was chosen to balance the opposing effects of topside and hopper tank fixity and was close to values determined using an approximate formula presented by Faulkner and Snyder (1971).

There was a steady decrease in fatigue life for ship lengths greater than 100m and less than 175m. This was due to a reduction of the heave natural frequency towards the frequency of the wave spectrum peak. The latter is essentially the same as the forcing frequency of the dominant waves and the shift in heave natural frequency leads to a resonant response. For ship lengths between 175m and 250m, and alternate hold loading, there was a slight increase in fatigue life. For ships in this length range with homogeneous loading there was no increase in fatigue life. For ship lengths greater than 250m fatigue life again showed a steady drop with increasing length regardless of loading pattern.

Fatigue life for homogeneous hold loading was higher than for alternate hold loading below approximately 290m above which, alternate hold loading provided a better fatigue life. This implies that alternate hold loading is preferable for ships over 290m in length and homogeneous hold loading is preferable for ships less than 290m in length.

The calculated fatigue lives obtained in this study were low but not unreasonable or immediately catastrophic. It had been widely accepted in industry that these were problem areas prone to what were previously considered "nuisance cracks". The progressive nature of the failure from toe crack to side shell punch-out as described was not widely known and the finding and repairing of these cracks was routine. This implies that the calculated lives are not unreasonable or out of line with actual experience.

It is seen from the discussions on the parameters significant to the shipboard fatigue problem that this is a very complex problem ranging from metallurgy to structural analysis, hydrodynamics and statistics. The difficulty in drawing generalizations about fatigue strength of ship details is clear. The multitude of designs, fabrication and load histories and variations in material properties and degradation make such generalizations difficult at best.

This work, by concentrating on a narrow, but very important problem, has been able to highlight some parametric relationships between basic ship parameters and fatigue life which are interesting and may be useful. Consequently, it would appear that despite the difficulties, further work would be very worthwhile.

Future Work

It must be borne in mind that this is a very simple model. More work is suggested on the dynamics, an advancement being a strip theory model to determine dynamic response to combinations of seastate and speeds of advance. Additional work would be useful to determine a more accurate model of the bottom shell pressure distribution than the simple Smith effect model used here.

Also of great importance is the three dimensional nature of the hold's bottom sag which was conservatively neglected here. A refinement would be to develop a full three dimensional finite element model of a hold and determine more precise values for the end loads on the frames and end fixity effects. Further work could also include actual shipboard strain gauge measurements to validate these models. The added complexity of future work , though, must be balanced against the clarity of a simple model.

The next step would be further validation of the chosen model. One method would be through correlation of damage and repair statistics versus size of ship and loading pattern. This is difficult however because damage statistics are not always available and if available may be kept private for commercial reasons.

References

AWS,1998. Committee on Structural Welding. Structural Welding Code-Steel ANSI/AWS D1.1-98. American Welding Society, Miami Florida.

Bannantine J.A., Comer J.J. and Handrock J L, 1990. Fundamentals of Metal Fatigue Analysis Prentice Hall Inc,Englewood Cliffs, NJ.

Bishop R.E.D. & Price,W.G.,1979. Hydroelasticity of Ships Cambridge University Press, Cambridge, UK.

Cary, Howard,1994. Modern Welding Technology. Regents/Prentice Hall: Englewood Cliffs, New Jersey.

CMS (Committee on Marine Structures),1997. Prevention of Fracture in Ship Structure Ship Structure Committee, US Coast Guard Report No. PFSS95.

Cai G.Q., Yu J.S. and Lin Y.K. 1995 "Fatigue Life and Reliability of Randomly Excited Structures", Journal of Ship Research, Vol 39,No.1,62-69.

Chen Y.K., Chiou J.W., Thayamballi A.K.,1986. "Validation of Fatigue Life Prediction Using Containership Hatch-Corner Strain Measurements", SNAME Transactions, Vol. 94,255-282.

Connors JJ and Brebbia C.A.,1976. Finite Element Techniques for Fluid Flow Newnes-Butterworths, London.

Cook R.D. and Young W.C.,1985. Advanced Mechanics of Materials Macmillan Publishing Company, New York.

Devore J.L., 1987. Probability and Statistics for Engineering and the Sciences Brooks/Cole Publishing Company, Belmont, California.

Donea J(editor),1980. Advanced Structural Dynamics Applied Science Publishers, London .

Dong Yan Qiu and Weixue Lin, 1992. "Hydroelasticity and Wave Loads for a Full-Form Ship with Shallow Draft" Journal of Ship Research, Vol. 36, No. 3, 280-285.

El-Gammal, M.M.,1987. "Proposed Reduction of Thickness Factor with Relation to Fatigue", International Shipbuilding Progress, Vol. 28, No. 327, 268-274.

Ellyin, Fernand,1997. Fatigue Damage, Crack Growth and Life Prediction Chapman and Hall.

Faltinsen, O.M.,1990. Sea Loads on Ships and Offshore Structures Cambridge University Press.

Faulkner D and Snyder G.J.,1971. "A New Approach to Local Frame Analysis", Massachusetts Institute of Technology, Report No. 71-9.

Ferguson J.M. m Cheng Y.F. and Purtle B, 1992-1993 "On Bulk Carrier Safety" Lloyd's Register Technical Association. Paper No. 5 Session,1992-93.

Fine M.E., Chung Y, McCormick R.R .,1996. "Fatigue Failure in Metals", ASM, vol. 19.

Gibstein M.B.1979 , "Fatigue failure of T-joints" Norwegian Maritime Research, No. 4,29-38.

Gran S.,1983. "Fatigue in Offshore Cranes", Norwegian Maritime Research, No. 4, 2-12

Hatlestad B and Kavlie D.,1979. "Fatigue Evaluation of Steel Decks in Fixed Offshore Platforms" Norwegian Maritime Research, No. 2.

Helle H.P.E. and Hageman L.A.S.,1980. "Fatigue Damage Accumulation in Marine Propellers", International Shipbuilding Progress, Vol. 27,193-210.

Hoffman D, Hsiung C.C. and Zielinski J.E.,1975. "Wave load Distributions on Large Ships" presented at the First Ship Technology and Research (STAR) Symposium, The Society of Naval Architects and Marine Engineers.

Hughes, O.F., 1983. Ship Structural Design A Rationally-Based, Computer -Aided, Optimization Approach, John Wiley and Sons, New York, New York.

Hurty W.C., Rubinstein M.F.,1964. Dynamics of Structures Prentice Hall Inc, Englewood Cliffs, NJ.

IACS,1994. Bulk Carriers Guidelines for Survey, Assessment and Repair of Hull Structure International Association of Classification Societies, London, UK.

Kardestuncer H.,1987. Finite Element Handbook McGraw-Hill Inc. New York, New York.

Lampman S. R. (editor),1996. "ASM Handbook Volume 19: Fatigue and Fracture", ASM International. Materials Park, Ohio.

Logan D.L.,1992.A First Course in the Finite Element Method PWS Kent Publishing Company, Boston, Massachusetts.

Lohne P.W.,1979. "A fracture mechanics approach to fatigue analysis of welded joints in offshore structures", Norwegian Maritime Research No.3,12-20.

Lloyd's Register,1999. Rules and Regulations for the Classification of a Floating Offshore Installations at a Fixed Location. Lloyd's Register of Shipping. London.

Lloyd's Register,1999. Rules and Regulations for the Classification of Ships Lloyd's Register of Shipping. London.

Ma Kai-Tung and Bea R.G,1995.. "A Repair Management System for Fatigue Cracks in Ships", SNAME Transactions, Vol 103, 343-369.

Moan T and Berge S.(editor),1997. Proceedings of the 13th International Ship and Offshore Structures Congress Volume 1 Elsevier Science Ltd.

Newman, J.M.,1977. Marine Hydrodynamics The MIT Press, Cambridge Massachussetts.

Press W.H.,1992. at al. Numerical Recipes in C The Art of Scientific Computing 2nd edition Cambridge University Press.

Reddy D.V. and Arockiasamy M (editors),1991. Offshore Structures Krieger Publishing Company, Malabar, Florida.

Riggs, R.,1979. "Fatigue Considerations for Semi-Submersible Structures", Marine Technology, Vol 16,No. 1, 49-62.

Stambaugh K.A. at al.,1992. Reduction of S-N Curves for Ship Structural Details Ship Structure Committee, US Coast Guard, Report No. SSC-369.

Thomson W.T.,1993. Theory of Vibration with Applications- 4th edition Prentice-Hall, Englewoods, NJ.

Ushirokawa O.,1981. "Fatigue Strength of Ship Structures: Practical Application for Design" Transactions of The Royal Institute of Naval Architects.

Vaughan H.,1982. "Fatigue and Fracture of Structural Elements under Random Loads", The Royal Institute of Naval Architects.

Xu Tao,1997. "Fatigue of Ship Structural Details - Technical Development and Problems", Journal of Ship Research, Vol. 41, No. 4, 318-331.

Ximenes M.C.,1991. "Fatigue Reliability and Inspection of TLP Tendon System", Marine Technology, Vol. 28,No. 2, 99-110.

Appendix 1: Parametric Study

The following parametric study was based on ship data obtained from Lloyd's Register of Shipping.

Basic data on length, breadth, depth, deadweight, double bottom height, hopper tank height and topside tank dimensions was obtained for 78 ships.

A summary of basic bulk carrier dimensions sorted by length is presented in the following table. From this basic data, ratios of L/B, L/D and L^3/DWT were formed and averaged to yield expected values.

L	B	D	T	DWT	DB	Hop	Top	No. Hold	L/B	L/D	L/T	L^3/DWT
68	11.2	4.12	3.31	1408	0.35			2	6.071	16.504	20.543	223.3
71.8	13.1	6	4.95	2630	0.3			2	5.480	11.966	14.505	140.7
77.5	13.6	6.42	5.34	3020	0.38			2	5.698	12.071	14.513	154.1
84.2	14.5	7.75	5.806	4300	0.4			2	5.806	10.864	14.502	138.8
86	15	8	6.4	4400	0.3	0.5	0.5	2	5.733	10.75	13.43	144.5
95	15.55	8.75	6.93	5662	0.26		0.5	3	6.109	10.857	13.708	151.4
97.5	16	9	7.03	6570	0.3		0.4	3	6.093	10.833	13.869	141.0
115	17.3	10.1	7.47	8000	0.3	0.8	0.6	3	6.647	11.386	15.394	190.1
145.53	21	13.2	10.14	18000	0.2	0.8	0.5	4	6.930	11.03	14.352	171.2
154.33	22.8	12.5	9.14	18860	0.2	0.5	0.6	4	6.768	12.34	16.883	194.8
150	22	11.75	8.61	17100	0.2	0.59	0.49	5	6.818	12.765	17.421	197.3
134	21.4	12.2	8.9	15700	0.3	0.7	0.5	5	6.261	10.983	15.056	153.2
150	26	14	10	24850	0.2	0.7	0.6	4	5.769	10.714	15	135
168	23.1	14.5	10.5	28000	0.175	0.5	0.4	5	7.272	11.586	16	169.3
167.2	23.1	14.75	10.57	27200	0.19	0.6	0.4	5	7.238	11.335	15.818	171.8
171.67	22.4	14.2	10.5	27000	0.2	0.5	0.42	7	7.664	12.089	16.350	187.4
165	26	14.5	10	28500	0.2	0.75	0.6	5	6.346	11.379	16.5	157.6
170	23.1	14.5	10.65	29500	0.1	0.55	0.5	6	7.354	11.724	15.962	166.5
178	22.8	14.6	10.65	30000	0.175	0.5	0.4	7	7.804	12.191	16.713	187.9
162	25.4	14.5	10.43	29000	0.2	0.55	0.5	5	6.377	11.172	15.532	146.6
170	23.08	14.5	10.64	29500	0.19	0.5	0.5	6	7.365	11.724	15.977	166.5
182.94	22.94	14.33	10.41	26700	0.16	0.6	0.4	5	7.974	12.766	17.573	229.3
167	22.9	14.5	10.4	26500	0.2	0.5	0.4	5	7.292	11.517	16.057	175.7
163.02	25.45	14.02	9.95	26500	0.2	0.45	0.5	5	6.405	11.627	16.382	163.4
170.77	22.7	14.35	10.4	26350	0.15	0.4	0.4	5	7.522	11.900	16.420	188.9
172	22.8	14.15	10.21	23980	0.2	0.6	0.5	7	7.543	12.155	16.846	212.1
152	23	13.9	9.83	22500	0.2	0.4	0.4	4	6.608	10.935	15.462	156.0
148	22.8	13.5	9.743	21100	0.2	0.6	0.5	5	6.491	10.962	15.190	153.6
170	27.6	14.1	9.96	30350	0.2	0.52	0.5	5	6.159	12.056	17.068	161.8
181	23.1	14.5	10.65	30850	0.2	0.6	0.6	6	7.835	12.482	16.995	192.2
185	25.3	15.11	10.6	32500	0.2	0.7	0.5	5	7.312	12.243	17.45	194.8
185	24.4	15	11	33500	0.2	0.7	0.4	7	7.581	12.333	16.818	189.0
173	28.5	15.1	11	34600	0.2	0.75	0.4	7	6.070	11.456	15.727	149.6
185	24.2	15.2	11.11	34800	0.15	0.55	0.4	5	7.644	12.171	16.651	181.9
182.9	27.13	15.1	10.78	35000	0.2	0.6	0.5	5	6.741	12.112	16.966	174.8
204.4	23.1	14.5	10.6	35400	0.2	0.57	0.5	7	8.848	14.096	19.283	241.2
175	28.4	14.9	10.72	35000	0.2	0.6	0.5		6.161	11.744	16.324	153.1
173	28.5	15.1	11	35500	0.2	0.75	0.5	7	6.070	11.456	15.727	145.8
186	26.5	15.4	11.36	37000	0.15	0.42	0.5	7	7.018	12.077	16.360	173.9
180	28.4	15.5	11	37592	0.2	0.5	0.5	5	6.338	11.612	16.363	155.1
187.7	27.8	15.6	11.18	38000	0.2	0.6	0.4	7	6.751	12.032	16.788	174.0
179	30.6	16	11	40000	0.2	0.65	0.5	5	5.849	11.187	16.272	143.3
175	32.23	15	10.67	40000	0.2	0.6	0.5	5	5.429	11.666	16.401	133.9

continued on next page

continued from previous

L	B	D	T	DWT	DB	Hop	Topside	n Holds	L/B	L/D	L/T	L^3/DWT
183	30	15.4	10.75	40800	0.175	0.6	0.6	5	6.1	11.883	17.023	150.2
178	32.2	17.2	11.43	45000	0.2	0.75	0.45	5	5.550	10.348	15.573	125.3
175	28.95	16.3	11.89	40900	0.2				6.044	10.736	14.712	131.0
189	29	16	11.65	44000	0.2	0.6	0.4	5	6.517	11.812	16.223	153.4
215	32	15.16	10.5	46400	0.15	0.55	0.4	6	6.718	14.182	20.476	214.1
184.03	32.25	17	12.4	49700	0.2	0.7		7	5.706	10.825	14.841	125.4
191	32.2	17.1	12	50000	0.19	0.6	0.5	7	5.931	11.169	15.916	139.3
205	32.2	17	12.35	55000	0.2	0.6	0.4	7	6.366	12.058	16.599	156.6
212.7	32.2	17.6	12.82	61000	0.2	0.7	0.5	7	6.605	12.085	16.591	157.7
215.64	32.2	18	13.1	63700	0.125	0.6	0.4	7	6.696	11.98	16.461	157.4
224.9	32.2	17.8	12.9	64300	0.2	0.6	0.5	7	6.984	12.634	17.434	176.9
213.75	32.24	18	13.1	64100	0.2	0.6	0.4	7	6.629	11.875	16.316	152.3
215	32.2	17.8	12.93	63900	0.2	0.65	0.6	7	6.677	12.078	16.627	155.5
235	32.25	18	13	65000	0.2	0.6	0.5	9	7.286	13.055	18.076	199.6
220	32.25	18.3	13.3	66500	0.2	0.8	0.5	7	6.821	12.021	16.541	160.1
215	32.23	18.85	13.2	66000	0.175	0.6	0.48	7	6.670	11.405	16.287	150.5
215.4	32.26	17.8	12.9	65300	0.2	0.65	0.5	7	6.676	12.101	16.697	153.0
214	32.2	17.7	12.84	65150	0.18	0.7	0.4	7	6.645	12.090	16.666	150.4
215	32.2	18.3	13.2	68100	0.2	0.5	0.5	7	6.677	11.748	16.287	145.9
219.5	32.23	20.15	14.86	77000	0.18	0.55	0.5	7	6.810	10.893	14.771	137.3
225	32.25	19.9	13.72	70400	0.15	0.7	0.6	7	6.976	11.306	16.399	161.7
248	32.25	19.2	14.02	81400	0.2	0.6	0.4	7	7.689	12.916	17.689	187.3
250	40.7	22	16.2	119500	0.19	0.7	0.5	9	6.142	11.363	15.432	130.7
250	42	22.7	16.67	127000	0.125	0.75	0.49	9	5.952	11.013	14.997	123.0
260	43	23.8	16.4	132700	0.2	0.7	0.55	9	6.046	10.927	15.853	132.4
258	42	22.9	16.77	137000	0.2	0.8	0.5	9	6.142	11.266	15.384	125.3
259.38	42.97	23.77	16.58	138000	0.2	0.5	0.5	9	6.036	10.912	15.644	126.4
260	43	23.8	17.5	144000	0.2	0.8	0.6	9	6.046	10.924	14.857	122.0
260	43	23.4	17.1	144700	0.2	0.65	0.55	9	6.046	11.111	15.204	121.4
280	43	23.2	17	151000	0.2	0.8	0.5	9	6.511	12.068	16.470	145.3
279	46	24	17.1	161000	0.2	0.725	0.6	9	6.065	11.625	16.315	134.8
276.6	46	24	17.8	172000	0.2	0.7	0.6	9	6.013	11.525	15.539	123.0
285	50	24.6	18.2	190000	0.2	0.6	0.5	11	5.7	11.585	15.659	121.8
306	54.2	25	18.7	223000	0.2	0.7	0.5	11	5.645	12.24	16.363	128.4
305	50	27	20	232000	0.2	0.7	0.5	11	6.1	11.296	15.25	122.2
								AVG	6.570	11.768	16.158	159.4
								STD	0.687	0.893	1.234	27.12
								MIN	5.429	10.348	13.43	121.4
								MAX	8.848	16.504	20.543	241.2

where
L = length of ship
B = breadth of ship
D = depth of ship
T = loaded draft of ship
DWT = deadweight of ship
DB = height of double bottom
Hop = height of hopper tasnk
Topside = depth of topside tank

Appendix 2: Rule-Based Frame Scantlings

Details of the development of the rule based frame scantlings are given here.

The frame is developed using the following idealized midship section

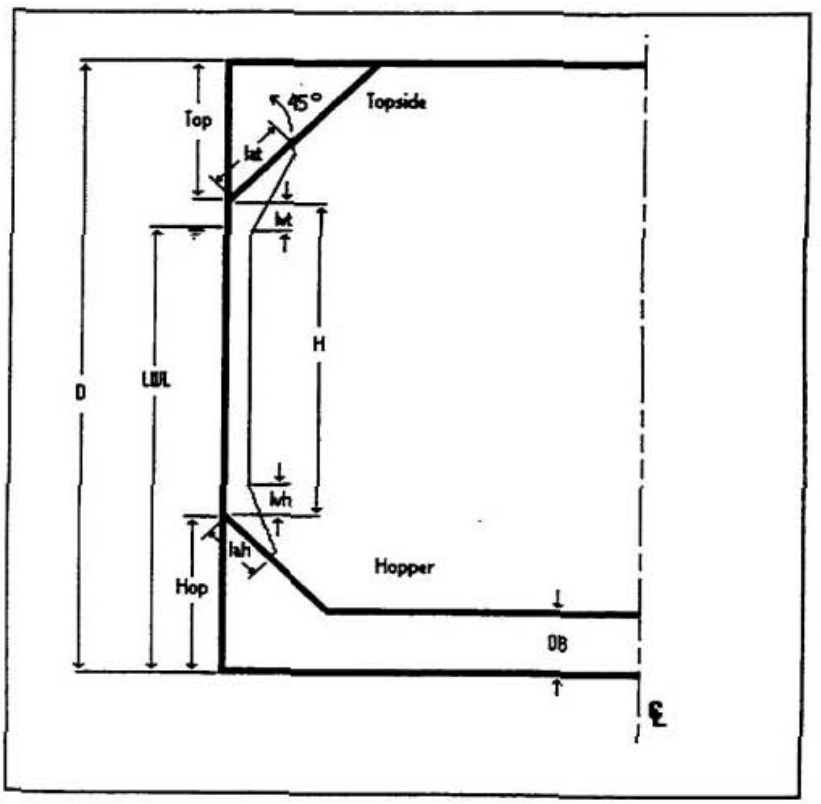


Figure: A1 Midship section

All dimensions measure with respect to the baseline at the keel.

DB = elevation of double bottom (m)
Hop = elevation of hopper tank crown (m)
Topside = vertical heigth of topside (m_
Top = elevation of topside tank toe (m)
LWL = elevation of load waterline (m)

From the parametric study in Appendix 1, the following relationships hold for a "typical" bulk carrier.

$$Hop = 0.0296L(m)$$
$$Top = 0.0615L(m)$$
$$H = Top - Hop = 0.0364L$$

Given , basic dimensions, the scantlings for the frame can be found by application of Lloyd's Register's Rules for the Classification of Ships.

Part 4, Chapter 7, Section 6.2 gives the required section modulus as:

$$z = 3.5skh_T H^2 \times 10^{-3}(cm^3)$$

where

s = frame spacing is given as 700mm or

$$s \geq (470 + \frac{L}{0.6})$$

Assume s =700mm, this will be at the design envelope for larger ships (>140m) and conservative for small ships (<140m).

k = material factor, taken as 1.0 for mild steel

$$h_T = (h_6 + C_w(1 - \frac{h_6}{2T}))$$

$$h_6 = T - \frac{h_{FRAME}}{2} = T - \frac{TOP + HOP}{2} = 0.0165L(m)$$

Cw = wave head $= 7.71 \times 10^{-2} Le^{-0.0044L}$

H = frame span = TOP-HOP = 0.0319L (m)

Upon substituting the factors back in to the section modulus formula, it can be rewritten,

$$z = 2.493L^3(0.0165 + 0.00668e^{-0.0044L}) \times 10^{-3}(cm^3)$$

Given section section modulus the Rule dimensions of the bracket flare and toes are:

$$l_{ah} = 32.43\sqrt{z}(mm)$$
$$l_{vh} = 27.6\sqrt{z}(mm)$$
$$l_{at} = 30\sqrt{z}(mm)$$
$$l_{vt} = 26.8\sqrt{z}(mm)$$

The required moment of inertia I is:

$$I_R = \frac{3.2HZ}{k} Z = 2.544L^4(0.0165 + 0.00668e^{-0.0044L})(mm^4)$$

A web profile was chosen to be slightly conservative and expedient.

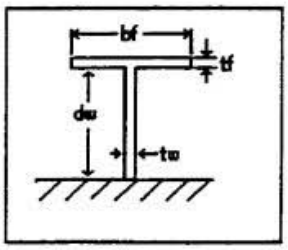


Figure A2 Beam profile

bf = breadth of flange (mm)
tf = thickness of flange (mm)
dw = depth of web (mm)
tw = thickness of web (mm)

The ratios of flange breadth/web depth and flange thk/web thk are left as parameters r1 and r2.

$$r_1 = \frac{b_f}{d_w}$$

$$r_2 = \frac{t_f}{t_w}$$

Rule requirement for thickness of the frame web is:
$t_w = \min((7 + 0.03L), 13)(mm)$

We note, $7 + 0.03L \sim \sqrt{L}$ for the length range under consideration (ie 90-300m). Also, applying the design rule of thumb that the thickness of the stiffener must be less than or equal to the thickness of the plate it's supporting, we will assume
$t_w = \sqrt{L}$

Consequently, for a given length L and set parameters r1 and r2, the following sectional properties can be defined as a function of dw:

Area of Flange: $A_f = b_f t_f = r_1 d_w r_2 t_w = r_1 r_2 d_w \sqrt{L}(mm^2)$
Area of Web: $A_w = d_w \sqrt{L}(mm^2)$
Area of Beam: $A_b = A_w + A_f = (1 + r_1 r_2) d_w \sqrt{L}(mm^2)$

Note: it was decided to maintain stiffness of the beam up to the toe by maintaining the breadth of the flange (even though in reality, the flange is tapered to a soft nose) in order to give us a worst case scenario.

Consequently, outside the midspan region, bf/dw not equal to r1 and the area of the beam must be calculated as Af+Aw.

Centroid of flange: $y_{cf} = dw + \frac{1}{2}t_f = dw + 0.5r_2\sqrt{L}(mm)$

Centroid of Web: $y_{cw} = 0.5d_w(mm)$

Centroid of Beam: $y_c = \frac{A_f y_{cf} + A_w y_{cw}}{A_b} = \frac{(0.5 + r_1 r_2)d_w^2\sqrt{L}}{(1 + r_1 r_2)d_w\sqrt{L}}(mm)$

Finally, the moment of inertia of the frame can be expressed as:

$$I_c = \frac{1}{12}b_f t_f^3 + (y_{cf} - y_c)^2 A_f + \frac{1}{12}t_w d_w^3 + (y_c - y_{cw})^2 A_w(mm^4)$$

Expressing in terms of r1,r2 and L and collecting about dw terms we get:

$$I_c = \left[\frac{1}{12}r_1 r_2^3 L^{3/2} + \frac{1}{4}\frac{r_1 r_2^3 L^{3/2}}{1 + r_1 r_2} + \frac{1}{4}r_1^2 r_2^4 L^{3/2} + \frac{1}{2}r_1 r_2 L\right]dw + \left[\frac{1}{2}\frac{r_1 r_2^2 L}{1 + r_1 r_2}\right]dw^2 + \left[\frac{1}{4}\frac{r_1 r_2\sqrt{L}}{1 + r_1 r_2} + \frac{\sqrt{L}}{3}\right]dw^3$$

$$= C_1 d_w + C_2 d_w^2 + C_3 d_w^3(mm^4)$$

For Rule based design, we require

$I_c = I_R$

$C_1 d_w + C_2 d_w^2 + C_3 d_w^3(mm^4) - I_R = R(dw, L) = 0$

where we force the residual R(dw,L) to zero.
Note, C1,C2,C3 are all positive montonically increasing functions, meaning the residual function (see figure below) is well behaved for L>0.

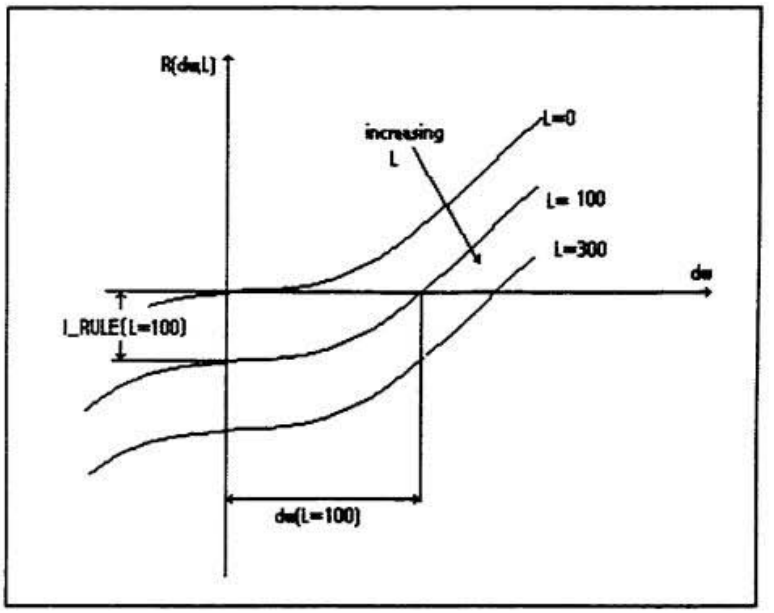


Figure A3: Schematic of Residual Function

Therefore, we can use a simple bisection method to solve for the web depth dw required to satisfy Rule Requirements at a given ship length L.

Appendix 3: Analysis Programs

The programs used to generate the frame, perform the fatigue analysis and finite element analysis are given here as developed using Symantec C++ for Windows

1) DOC_BLKR_FRAME.H : basic definitions for frame generation modules
2) DOC_BLKR_FRAME.CPP: frame generation code
3) DOC_FATIGUE.H : basic definitions for fatigue module
4) DOC_FATIGUE.CPP : fatigue analysis code
5) DOC_FEM.H : basic definitions for finite element modules
6) DOC_FEM.CPP : finite element modules

FILENAME: DOC_BLKR_FRAME.H

```
// this is the header file for the frame generation routines
#define PI 3.14159265
#define g 9.81
#define pg 10055.25
#define COS45 0.707
#define del_L 0.70
#define n_b 16
#define n_a 16
#define NOSE 17.9578  // =COS45 *25.4

//declared global so parameter window allowed access...
int N_TOP_TOE,N_TOP_FLARE,N_MID,N_BOTTOM_FLARE,N_BOTTOM_TOE;
int HI_TOE_FLEX,LO_TOE_FLEX,HOPPER_ROTATION,HOPPER_END_MOMENT;
float C_L;
int SHOW_DISPLACEMENTS,D_SCALE;

float r1,r2;
float HI_TOE_FIXITY;
float LO_TOE_FIXITY;
//end global definitions...

int FRAME_AUTOGEN;

class blkr_frame
{
public:
float RAOx,RAOz,F_I,F_B,M_I,M_B,M_NET,F_NET,z_w,x_w;
float a_11_x, b_22_x,b_x,a_x,m_x,k_x;

private:
//control variables
//int N_TOP_TOE,N_TOP_FLARE,N_MID,N_BOTTOM_FLARE,N_BOTTOM_TOE;
//int HI_TOE_FLEX,LO_TOE_FLEX;
//int C_L;

float HI_TOE_L,HI_TOE_E,HI_TOE_A;
float LO_TOE_L,LO_TOE_E,LO_TOE_A;
int dof;

float L;
float I,Af,Aw,Ab,ycf,ycw,yc;
float tw,dw,dwo,bfo,tf,bf;
float y_coord;// y coord of the centroid corrected for tank offset
float HOA,H,Ivt,Iat,Ivh,Iah,Top,Bottom;
float I_RULE,Z_RULE;
void calc_dynamic_load(float L, float A, float T);

public:
blkr_frame(FEM_model_data& model);
void init(FEM_model_data& model,float L_SHIP);
void fill_blkr_frame(FEM_model_data& model,float L_SHIP);
void define_load(FEM_model_data& model,float A,float T);
void define_bc(FEM_model_data& model);

private:
void get_section(float x);
float I_SECTION();
float A_SECTION();
float find_dwo();
};
```

FILENAME: DOC_BLKR_FRAME.CPP

```
//doc_blkr_frame.cpp  bc Sept/99
// 1) calc geometry and sectional properties of bulk carrier frame
//    and automatically generate spring toe/crown supports if set
// 2) calc loading ( dynamic and hydrostatic)
// 3) define boundary conditions

// usage:  include header after doc_fem.h
//         include code anywhere


//-------------------------------------------------------------//
blkr_frame::blkr_frame(FEM_model_data& model)
{

HI_TOE_FLEX=TRUE;
LO_TOE_FLEX=TRUE;

//add beam elements
model.n_elements=N_TOP_TOE+N_TOP_FLARE+N_MID+N_BOTTOM_FLARE+N_BOTTOM_TOE;
//add elastic foundation elements ( ie springs), if reqd
if(HI_TOE_FLEX) model.n_elements+=N_TOP_TOE;
if(LO_TOE_FLEX) model.n_elements+=N_BOTTOM_TOE;

model.n_nodes= model.n_elements+1;

//calculate number of degrees of freedom
//first the beam elements...
dof =(N_TOP_TOE+N_TOP_FLARE+N_MID+N_BOTTOM_FLARE+N_BOTTOM_TOE+1)*3;
//now for the spring elements
if(HI_TOE_FLEX) dof += N_TOP_TOE*2;
if(LO_TOE_FLEX) dof +=N_BOTTOM_TOE*2;

model.n_bc=dof;   // set to max possible value
model.n_loads=dof;
model.g_dof=dof;

// need to allow for multiple runs with change of parameters, so check
// if matrices previously allocated and if so delete and reallocate
// with new sizes
if( model.coord != NULL) delete model.coord;
if( model.el_def != NULL) delete model.el_def;
if( model.load != NULL) delete model.load;
if( model.bc != NULL) delete model.bc;

model.coord = (_matrix<float>*) new _matrix<float>(model.n_nodes  ,5);
model.el_def= (_matrix<float>*) new _matrix<float>(model.n_elements,8);
model.load  = (_matrix<float>*) new _matrix<float>(model.n_loads  ,3);
model.bc    = (_matrix<float>*) new _matrix<float>(model.n_bc     ,3);

}

//given: wave amplitude A (m)and wave period T calculate loadings and
//distribute amoung model nodes in load matrix...
void blkr_frame::define_load(FEM_model_data& model,float A, float T)
{
int n,N1,N2;

float x1,x2;
float Head,pD,L_el;
float U10,U20,V10,V20,M10,M20;


// calculate dynamic response
calc_dynamic_load(L,A,T);

float LWL_coord = -1*(16.45*L +z_w*1000);

//cycle thru elements and check if "wet"
```

```
for (int i=1;i<=model.n_elements;i++)
{
N1= *model.el_def(i,Node1); N2=*model.el_def(i,Node2);
x1= *model.coord(N1,1);    x2=*model.coord(N2,1);
Head = (x1-LWL_coord)/1000;// must convert to m (eventhough everything
L_el=(x2-x1)/1000;        // everything else in N,mm) because formulas
                                                        // for equivalent nodal loadings in terms of
                                                        // N,m. the m's drop out leaving only N so
                                                        // no conversion needed except for moments
                                                        //which must be divided by 1000 to give N-m

if (Head >0 && x1>=-1*H/2-1 && x2<=H/2+1)
            {
            pD=pg*fabs(A)*exp(-1*pow(2*PI,2)/(g*pow(T,2)*Head));

            U10=0;
            V10= 0.710*(3*pg/20*pow(L_el,2)+pg/2*Head*L_el+pD);
            M10= 0.710*(pg/30*pow(L_el,3)+pg/12*Head*L_el*L_el)/1000;
            U20=0;
            V20= 0.710*(7*pg/20*pow(L_el,2)+pg/2*Head*L_el+pD);
            M20= 0.710*( -pg/20*pow(L_el,3)-5*pg/12*Head*L_el*L_el)/1000;


            n=*model.coord(N1,global_ndx);
            *model.load(n,1)=N1;          *model.load(n+1,1)=N1;        *model.load(n+2,1)=N1;
            *model.load(n,2)=1;   *model.load(n+1,2)=2;         *model.load(n+2,2)=3;
            *model.load(n,3)=U10;  *model.load(n+1,3)=V10;     *model.load(n+2,3)=M10;

            n=*model.coord(N2,global_ndx);
            *model.load(n,1)=N2;          *model.load(n+1,1)=N2;        *model.load(n+2,1)=N2;
            *model.load(n,2)=1;   *model.load(n+1,2)=2;         *model.load(n+2,2)=3;
            *model.load(n,3)=U20;  *model.load(n+1,3)=V20;     *model.load(n+2,3)=M20;
            }
} //end wet elements


if(HOPPER_END_MOMENT !=0)
{
// apply dynamic loads as end moment

int load_node,load_dof;

load_node = model.n_nodes-N_BOTTOM_TOE*2;
load_dof= *model.coord(load_node,global_ndx);

*model.load(load_dof,1) = load_node;
*model.load(load_dof,2) =3;
*model.load(load_dof,3) -= M_NET;
  char buf[80];
  sprintf (buf,"%i %f", load_node,M_NET);
  AfxMessageBox(buf);


}

if(HOPPER_ROTATION ==0 && HOPPER_END_MOMENT==0)
{
// apply dynamic loads directly to model nodes instead of rotation
int load_node,load_dof;

for(i=model.n_nodes;i>model.n_nodes-N_BOTTOM_TOE*2;i-=2)
            {
            load_node=i; load_dof= *model.coord(load_node,global_ndx);
            *model.load(load_dof,1) = load_node;
            *model.load(load_dof,2) =1;
            *model.load(load_dof,3) += ((F_I+F_B)/(N_BOTTOM_TOE+1));
            }
}

}
```

```
void blkr_frame::calc_dynamic_load(float L, float A, float T)
{


float wn = 2*PI/T; // [rad/sec]
float wnc = wn*0.0795*sqrt(L); // [rad/sec]

float m_s_x = 6.629*L*L*C_L;   // unit mass [kg]
k_x = 1069.879*L;       // stiffness coefficient   [N/m]
// calculate damping coefficient b
// based on curves for a partially submerged right cylinder (Newman,p299)

float b_wnc_ndx[n_b+1], b_val[n_b+1];

b_wnc_ndx[1]= 0.0001; b_val[1]=1.0000;  b_wnc_ndx[9]= 0.8; b_val[9]= 0.1327;
b_wnc_ndx[2]= 0.1;   b_val[2]=0.8938;  b_wnc_ndx[10]=0.9; b_val[10]=0.0885;
b_wnc_ndx[3]= 0.2;   b_val[3]=0.7787;  b_wnc_ndx[11]=1.0; b_val[11]=0.0531;
b_wnc_ndx[4]= 0.3;   b_val[4]=0.6283;  b_wnc_ndx[12]=1.1; b_val[12]=0.0354;
b_wnc_ndx[5]= 0.4;   b_val[5]=0.4955;  b_wnc_ndx[13]=1.2; b_val[13]=0.0177;
b_wnc_ndx[6]= 0.5;   b_val[6]=0.3805;  b_wnc_ndx[14]=1.3; b_val[14]=0.0089;
b_wnc_ndx[7]= 0.6;   b_val[7]=0.2743;  b_wnc_ndx[15]=1.4; b_val[15]=0 ;
b_wnc_ndx[8]= 0.7;   b_val[8]=0.1858;  b_wnc_ndx[16]=1.5; b_val[16]=0;

int i=1; BOOL FOUND=FALSE;
// check wn ranges...
if (wnc>1.5) { b_x=0; FOUND=TRUE;}
if (wnc<0.001) { b_x=1; FOUND=TRUE;}

// otherwise, if in range interpolate...
while (!FOUND && i<n_b)
   {
   if (wnc== b_wnc_ndx[i])  { b_x = b_val[i];  FOUND=TRUE;}
   if (wnc== b_wnc_ndx[i+1]){ b_x = b_val[i+1]; FOUND=TRUE;}
   if ( wnc>b_wnc_ndx[i] && wnc<b_wnc_ndx[i+1])
      { // then wn is bracket, interpolate b_x....
      b_x = b_val[i] + (wnc-b_wnc_ndx[i])/(b_wnc_ndx[i+1]-b_wnc_ndx[i])*
            (b_val[i+1]-b_val[i]);
      FOUND=TRUE;
      }
      if(!FOUND) i++; //increment ndx and loop...
   }

if (!FOUND) AfxMessageBox("ERROR COULD NOT FIND DAMPING COEFFICIENT");

else
   b_22_x = b_x*23.6816*wn*L*L; // [kg/m-sec]


//calculate added mass coeffient a
// based on curves for a partially submerged right cylinder (Newman,p299)
float a_wnc_ndx[n_a+1], a_val[n_a+1];
a_wnc_ndx[1]= 0.0001; a_val[1]=2.5437;  a_wnc_ndx[9]= 0.8; a_val[9]= 0.3895;
a_wnc_ndx[2]= 0.1;   a_val[2]=1.2719;  a_wnc_ndx[10]=0.9; a_val[10]=0.4133;
a_wnc_ndx[3]= 0.2;   a_val[3]=0.779;   a_wnc_ndx[11]=1.0; a_val[11]=0.4452;
a_wnc_ndx[4]= 0.3;   a_val[4]=0.5882;  a_wnc_ndx[12]=1.1; a_val[12]=0.461 ;
a_wnc_ndx[5]= 0.4;   a_val[5]=0.4769;  a_wnc_ndx[13]=1.2; a_val[13]=0.4849;
a_wnc_ndx[6]= 0.5;   a_val[6]=0.4054;  a_wnc_ndx[14]=1.3; a_val[14]=0.5001;
a_wnc_ndx[7]= 0.6;   a_val[7]=0.3815;  a_wnc_ndx[15]=1.4; a_val[15]=0.5087;
a_wnc_ndx[8]= 0.7;   a_val[8]=0.3736;  a_wnc_ndx[16]=1.5; a_val[16]=0.5246;

i=1; FOUND=FALSE;
// check wn ranges...
if (wnc>1.5) { a_x=0.5246; FOUND=TRUE;}
if (wnc<0.001) { a_x=2.5437; FOUND=TRUE;}
// otherwise, if in range interpolate...

while (!FOUND && i<n_a)
   {
   if (wnc== a_wnc_ndx[i])  { a_x = a_val[i];  FOUND=TRUE;}
```

```
  if (wnc== a_wnc_ndx[i+1]){ a_x = a_val[i+1]; FOUND=TRUE;}
  if (wnc>a_wnc_ndx[i] && wnc< a_wnc_ndx[i+1])
     { // then wn is bracket, interpolate b_x....
     a_x = a_val[i] + (wnc-a_wnc_ndx[i])/(a_wnc_ndx[i+1]-a_wnc_ndx[i])*
           (a_val[i+1]-a_val[i]);
     FOUND=TRUE;}
  else // default is not found, so go on
     i++;
  }

if (!FOUND) AfxMessageBox("ERROR COULD NOT FIND ADDED-MASS COEFFICIENT");
else
   a_11_x = a_x*23.6816*L*L; // [kg]

m_x = m_s_x+a_11_x; // mass = sectional + added

// calculate RAO
RAOx= sqrt((k_x*k_x+pow(b_22_x*wn,2))/
        (pow(k_x-m_x*wn*wn,2)+pow(b_22_x*wn,2)) );
x_w = RAOx * A;     //amplitude of heave response  [m]
F_I = m_s_x*wn*wn*x_w; // inertial force [N]

RAOz= m_x*wn*wn/sqrt(pow(k_x-m_x*wn*wn,2)+pow(b_22_x*wn,2));
z_w = RAOz * A; // amplitude of relative heave response [m]

//calculate Smith effect
float Head = (0.062*L+z_w);
float pD=pg*A*exp((-4.0*PI*PI)/(g*T*T)*Head);

F_B = k_x*z_w- pD*0.1064*L;  // buoyant force [N]


F_NET = F_B + F_I; // net force [N]

M_I = 0.05066*L*F_I; //inertial moment [N-m]
M_B = 0.038*L*F_B; // inertial moment [N-m]

M_NET = M_B + M_I;



}

void blkr_frame::define_bc(FEM_model_data& model)
{
int i,bc_dof,bc_node;
//set spring ends=0
if(HI_TOE_FLEX) //fix upper toe topside tank ends
        for (i=2;i<=N_TOP_TOE*2;i+=2)
        {
        bc_node = i; bc_dof=*model.coord(bc_node,global_ndx);
        //bc_dof=4+(i-1)*5;
        *model.bc(bc_dof,1)=bc_node; *model.bc(bc_dof+1,1)=bc_node;
        *model.bc(bc_dof,2)=1;      *model.bc(bc_dof+1,2)=2;
        *model.bc(bc_dof,3)=0;      *model.bc(bc_dof+1,3)=0;
        };
if(LO_TOE_FLEX) //fix upper toe hopper tanks ends
        for (i=1;i<=N_BOTTOM_TOE*2;i+=2)
        {
        bc_node=model.n_nodes-i; bc_dof= *model.coord(bc_node,global_ndx);
        *model.bc(bc_dof,1)=bc_node; *model.bc(bc_dof+1,1)=bc_node;
        *model.bc(bc_dof,2)=1;      *model.bc(bc_dof+1,2)=2;
        *model.bc(bc_dof,3)=0;      *model.bc(bc_dof+1,3)=0;
        };

// calc dof of upper pivot taken at bottom of topside tank
if(HI_TOE_FLEX) {bc_dof = N_TOP_TOE*5+1; bc_node=N_TOP_TOE*2+1;}
else {bc_dof = N_TOP_TOE*3+1;bc_node=N_TOP_TOE+1;}
*model.bc(bc_dof,1)=bc_node;  *model.bc(bc_dof,2)=1;  *model.bc(bc_dof,3)=0;
*model.bc(bc_dof+1,1)=bc_node;*model.bc(bc_dof+1,2)=2;*model.bc(bc_dof+1,3)=0;
```

```
//calc dof of lower pivot taken at top of hooper tank
if(LO_TOE_FLEX) {bc_dof=model.g_dof-N_BOTTOM_TOE*5-1;
          bc_node=model.n_nodes-N_BOTTOM_TOE*2;}
else       {bc_dof=model.g_dof-N_BOTTOM_TOE*3-1;
          bc_node=model.n_nodes-N_BOTTOM_TOE;}
*model.bc(bc_dof,1)=bc_node; *model.bc(bc_dof,2)=1; *model.bc(bc_dof,3)=0;
*model.bc(bc_dof+1,1)=bc_node;*model.bc(bc_dof+1,2)=2;*model.bc(bc_dof+1,3)=0;


if (HOPPER_ROTATION != 0)
  //apply rotation to the lower bracket to account for dynamic moments
  {
  float M = M_NET*1000; //convert N-m to N-mm
  //distance between spring supports
  float dl = 0.707*(lah/N_BOTTOM_TOE);
  // support stifnnes, all assumed equal
  float k = LO_TOE_E*LO_TOE_A*LO_TOE_FIXITY/LO_TOE_L;
  //calc sumI2
  for(float SUMI2=0,i=1;i<=N_BOTTOM_TOE;i++) SUMI2+=i*i;
  //angle of rotation of lower hopper
  float theta = -M/(k*dl*dl*SUMI2);
  int r_ndx=0;

  if(LO_TOE_FLEX) //fix upper toe hopper tanks ends
          for (i=1;i<=N_BOTTOM_TOE*2;i+=2)
          {
          bc_node=model.n_nodes-i; bc_dof= *model.coord(bc_node,global_ndx);
          *model.bc(bc_dof,1)=bc_node;          *model.bc(bc_dof+1,1)=bc_node;
          *model.bc(bc_dof,2)=1;                *model.bc(bc_dof+1,2)=2;
          *model.bc(bc_dof,3)=(N_BOTTOM_TOE-r_ndx)*dl*theta;*model.bc(bc_dof+1,3)=0;
          r_ndx++;

          };

}



}

void blkr_frame::init(FEM_model_data& model,float L_SHIP)
{
L = L_SHIP;
H = 31.9*L;
I_RULE =2.544000*pow(L,4)*(.0165+.0668*exp(-0.0044*L));
Z_RULE =0.002493*pow(L,3)*(.0165+.0668*exp(-0.0044*L));
lah = 32.43*sqrt(Z_RULE);
lvh = 27.60*sqrt(Z_RULE);
lat = 30.00*sqrt(Z_RULE);
lvt = 26.80*sqrt(Z_RULE);

HOA   = H  +COS45*(lat+lah);
Top   =-H/2-COS45*lat;
Bottom= H/2+COS45*lah;

r1=0.6666;r2=1;

tw=sqrt(L);     tf=r2*tw;
dwo=find_dwo(); bf=r1*dwo;


HI_TOE_L=tw*10; //scale up the length so is more visible
HI_TOE_E=200E3; //scale down E by same amount to keep stiffness
HI_TOE_A=0.34*tw*lat/N_TOP_TOE;

LO_TOE_L=tw*10; // scale up the length so is more visible
LO_TOE_E=200E3/10; // scale down by same amount to keep stiffness
LO_TOE_A=0.34*tw*lah/N_BOTTOM_TOE;
```

```
}

void blkr_frame::fill_blkr_frame(FEM_model_data& model, float L_SHIP)
{
float x,intrvl,x1,x2,y1,y2,err;
int i,j,ndx,el_ndx,coord_ndx;
BOOL PAST_TOP_FLARE=FALSE;
BOOL PAST_MID=FALSE;
BOOL PAST_BOT_FLARE=FALSE;
BOOL PAST_BOT_TOE=FALSE;
BOOL SPRING_ADDED = FALSE;

init(model,L_SHIP);

intrvl = COS45*lat/N_TOP_TOE;
ndx=1; coord_ndx=1;x=Top;

while (x<Bottom-1)
        {
        x1=x;x2=x+intrvl;
        if( x2 >Bottom) x2=Bottom;

        get_section(x1); // get sectional properties && coords at node 1

        //set coords for node 1
        *model.coord(coord_ndx,1)=x1;
        *model.coord(coord_ndx,2)=y_coord;
        *model.coord(coord_ndx,node_dof)=max(3,*model.coord(coord_ndx,node_dof));

        if(x2<-H/2+1 && HI_TOE_FLEX && !PAST_TOP_FLARE)
                {// add extra node and element for spring
                *model.coord(coord_ndx+1,1)=x1-HI_TOE_L;
                *model.coord(coord_ndx+1,2)=y_coord;
                *model.coord(coord_ndx+1,node_dof)=max(2,*model.coord(coord_ndx+1,4));

                //set element_table
                *model.el_def(ndx,1)=truss_2D;
                *model.el_def(ndx,2)=coord_ndx;  //node 1 = this ndx
                *model.el_def(ndx,3)=coord_ndx+1;
                *model.el_def(ndx,4)=HI_TOE_E*HI_TOE_FIXITY;
                *model.el_def(ndx,5)=HI_TOE_A;
                ndx++; SPRING_ADDED=TRUE;
                }
        if(x2>H/2+1 && LO_TOE_FLEX && PAST_BOT_FLARE)
                {//add extra node and element for spring
                get_section(x2); // get sectional properties at node 2
                *model.coord(coord_ndx+1,1)=x2+LO_TOE_L;
                *model.coord(coord_ndx+1,2)=y_coord;
                //get_section(x1); //return to x1 temporarially
                *model.coord(coord_ndx+1,node_dof)=max(2,*model.coord(coord_ndx+1,4));
                //set element_table
                *model.el_def(ndx,1)=truss_2D;
                *model.el_def(ndx,2)=coord_ndx+1;  //node 1 = second node
                *model.el_def(ndx,3)=coord_ndx+2; //increment coord_ndx and set
                *model.el_def(ndx,4)=LO_TOE_E*LO_TOE_FIXITY;
                *model.el_def(ndx,5)=LO_TOE_A;
                ndx++; SPRING_ADDED=TRUE;
                }


        // get midspan sectional properties(yc,y_coord,I,Ab)
        get_section((x1+x2)/2);

        //set element_table
        *model.el_def(ndx,1)=beam_2D;
        *model.el_def(ndx,2)=coord_ndx;  //node 1 = this ndx
        coord_ndx++; // increment coordinate index
```

```
        if(SPRING_ADDED) {coord_ndx++; SPRING_ADDED=FALSE;} // skip added spring node

        *model.el_def(ndx,3)=coord_ndx; // node 2 = next ndx
        *model.el_def(ndx,4)=200E3; //E=210 000 N/mm^2

        *model.el_def(ndx,6)=I;
        *model.el_def(ndx,5)=Ab;
        ndx++;

        //get sectional properties && coords for node 2
        get_section(x2);
        //set coords for node 2
        *model.coord(coord_ndx,1)=x2;
        *model.coord(coord_ndx,2)=y_coord;
        *model.coord(coord_ndx,node_dof)=max(3,*model.coord(coord_ndx,node_dof));

        //coord_ndx++;
        // adjust intrvl if necessary
        err=1;
        if(x2>=-H/2-err && !PAST_TOP_FLARE)
                {
                intrvl=lvt/N_TOP_FLARE;
                x2=-H/2;
                PAST_TOP_FLARE=TRUE;
                }
        if(x2>=(-H/2+lvt-err)&& !PAST_MID)
                {
                intrvl=(H-lvt-lvh)/N_MID;
                x2=-H/2+lvt;
                PAST_MID=TRUE;
                }
        if(x2>=(H/2-lvh-err) && !PAST_BOT_FLARE)
                {
                intrvl=lvh/N_BOTTOM_FLARE;
                x2=H/2-lvh;
                PAST_BOT_FLARE=TRUE;
                }
        if(x2>=H/2-err && !PAST_BOT_TOE)
                {
                intrvl= COS45*lah/N_BOTTOM_TOE;
                x2=H/2;
                PAST_BOT_TOE=TRUE;
                }

        // increment ndx and x
        x=x2;
        }


//now that the model has been defined...
//set global index for each node, start index at 1
for(*model.coord(1,global_ndx)=1,i=2;i<=model.n_nodes;i++)
        *model.coord(i,global_ndx)=*model.coord(i-1,global_ndx)+
                                                        *model.coord(i-1,node_dof);

}


//claculates the depth of the web in way of the upper/lower end brackets
// based on the assumption the arms comply with Lloyd's Rules and
// they rest on 45 degree hopper tanks top and bottom.
void blkr_frame::get_section(float x)
{
float y,tank;
if (x<=-H/2) // in the upper bracket toe
        {y=(0.707*lat-dwo+NOSE)/(0.707*lat+lvt-NOSE)*(lvt-x-H/2)+dwo;
         tank =(-H/2-x); }
if (x>=-H/2 && x<=(-H/2+lvt)) // in the flare of the upper bkt
        {y=(0.707*lat-dwo+NOSE)/(0.707*lat+lvt-NOSE)*(lvt-x-H/2)+dwo;
         tank = 0;}
```

```
if( x>(-H/2+lvt) && x<(H/2-lvh)) // in mid span region
            {y= dwo;
             tank=0;}
if (x>=(H/2-lvh)&& x<=H/2) /// in the flare of the lower bkt
            {y=(0.707*lah-dwo+NOSE)/(0.707*lah+lvh-NOSE)*(x+lvh-H/2)+dwo;
             tank=0;}
if (x>(H/2)) // in lower toe
            {y=(0.707*lah-dwo+NOSE)/(0.707*lah+lvh-NOSE)*(x+lvh-H/2)+dwo;
             tank=(x-H/2);}

dw = y - tank;
bf = r1* dwo; tf= r2*tw;
Af = bf*tf; Aw = tw*dw; Ab = Aw+Af;
ycf = dw + 0.5*tf;
ycw = 0.5*dw;
yc = (Af*ycf + Aw*ycw)/Ab;
y_coord=yc+tank;

I = I_SECTION();
}

//returns moment of inertia in mm^4
inline float blkr_frame::I_SECTION()
{ return( bf*tf*tf*tf/12 + (ycf-yc)*(ycf-yc)*Af + tw*dw*dw*dw/12
        + (yc-ycw)*(yc-ycw)*Aw); }

//return cross sectional area in mm^2
inline float blkr_frame::A_SECTION()
{ return ((1+r1*r2)*dw*sqrt(L)); }

//based on crude bisection method but OK because residual function
// well behaved over the interval in question ( is on the upward
// arm of a cubic function and is monotonically increasing from a
// negative residual at x=0 to a positive residual at x=1000.
//returns depth of web in mm
float blkr_frame::find_dwo()
{
int count = 0;
float x_max=1000;
float x_min=0;
double x = (x_min+x_max)/2;
double R=1;

while (fabs(R)>0.0001 && count<100 && fabs(x_max-x_min)>.01)
//physical measurements in mm, anything less than 0.01mm is
//meaningless in a physical sense so jump out when max-min<.01
            {
            if ( R<0) x_min=x; else x_max=x;
            x = (x_min+x_max)/2; dwo=x;
            get_section(0); // init frame model, 0=midspan
            R = I-I_RULE;
            count++;
            }
return x;
}
```

FILENAME: DOC_FATIGUE.H

```
// doc_fatigue.h
#define max_n_histogram 10
#define material_Su 30
#define S_N_constant  338.7
#define S_N_slope 0.1743
#define w_min 0.2
#define w_max 1.5
#define del_w 0.1
#define n_w 14    // note n_w = w_max-w_min/del_w +1 recalc if any changed
#define FATIGUE_FAILURE 0.3


enum {w_i=1,block_damage};
enum {H_S=1,occurance};


class fatigue_model
{
private:
float s_max,s_min,RAOx,RAOz,F_I,F_B,M_NET,z_w,x_w,A_i,a_22,b_22,m_x,k_x,LBP,wn;
float life,damage;

public:

_matrix<float>* cycle_ratio;

fatigue_model();

void dump(int N);
float calc_life(float L_SHIP);
void get_stress_range(float L_SHIP,float a_i, float T);
float get_cycle_damage(float s_max, float s_min, float ni);
float residual(float R, float N, float b, float del_S);

};


class wave_model
{
private:
float w,wm[10][5], wh[19][3];

public:
wave_model();
float S(float w,float Hs);
float component_amplitude( float w);
};
```

FILENAME: DOC_FATIGUE.CPP

```
// doc_fatigue.cpp
void fatigue_model::dump(int N)
{
CStdioFile out;
char buf[120];
int i,j;
if(!out.Open("dump.txt",
   CFile::modeCreate|CFile::modeNoTruncate|CFile::modeWrite)) return;

if (N==1)
{
sprintf(buf,"L=,%f,  C_LOAD=,%f ",LBP,C_L);
out.WriteString(buf);

/*
sprintf(buf,"N_TOP_FLARE=,%i,  N_TOP_TOE=,%i\n",N_TOP_FLARE,N_TOP_TOE);
out.WriteString(buf);

sprintf(buf,"N_MID=,%i\n",N_MID);
out.WriteString(buf);

sprintf(buf,"N_BOT_FLARE=,%i, N_BOT_TOE=,%i\n",N_BOTTOM_FLARE,N_BOTTOM_TOE);
out.WriteString(buf);
*/
sprintf(buf,"HI_TOE_FIXITY=,%f, LO_TOE_FIXITY=,%f \n",HI_TOE_FIXITY,LO_TOE_FIXITY);
out.WriteString(buf);

//sprintf(buf," w, A_i,a_22,b_22,m_x,RAOz,RAOx,x,z,F_I,F_B,M\n");
//out.WriteString(buf);

}

if (N==2)
{
sprintf(buf,
      "%8.3f,%8.3f,%8.3f,%8.3f,%8.3f,%8.3f,%8.3f,%8.3f,%8.3f,%8.3f,%8.3f,%8.3f\n"
      ,wn,A_i,a_22,b_22,m_x,RAOz,RAOx,x_w,z_w,F_I,F_B,M_NET);
out.WriteString(buf);

}


if(N==3)
{

sprintf(buf,"w, damage, s_max, s_min \n  ");
out.WriteString(buf);

for(i=1;i<=n_w;i++)
   {
   sprintf(buf,"%f,%f,%f,%f\n", *cycle_ratio(i,w_i),*cycle_ratio(i,block_damage),
         *cycle_ratio(i,3),*cycle_ratio(i,4));
   out.WriteString(buf);
   }

sprintf(buf,"Fatigue_life= %f ,years  Accumulated Damage=%f \n ",life,damage);
out.WriteString(buf);

}
if(N==4)
{

//sprintf(buf,"Fatigue_life= %f ,years,  Accumulated Damage=%f \n ",life,damage);
sprintf(buf,"Life= %f,%f\n",life,damage);
out.WriteString(buf);

}
```

```
out.Close();
return;

}


fatigue_model::fatigue_model()
{ cycle_ratio= (_matrix<float>*) new _matrix<float>(n_w,11);}

float fatigue_model::calc_life(float L_SHIP)
{
float a_i,T,w,ni, max_T;
wave_model ITTC;
int j=1;
LBP=L_SHIP;

HOPPER_END_MOMENT=1;

dump(1);
//a load block is of duration = maximum period.
//the block damage for each component is the damage accumulated
//by ni cycles of component in max_T
max_T = 60*60*24; //choose one day as block unit...


for (L_SHIP=90;L_SHIP<=300;L_SHIP+=10)
{
for(w= w_min; w<=w_max+0.001;w+=del_w)
   {
   a_i =ITTC.component_amplitude(w);
   T = 2* PI/w;
   get_stress_range(L_SHIP,a_i,T); // sets s_min,s_max
   ni = max_T/T;

   *cycle_ratio(j,w_i)=w;
   *cycle_ratio(j,block_damage)= get_cycle_damage(s_max,s_min,ni);

   *cycle_ratio(j,3)=s_max;
   *cycle_ratio(j,4)=s_min;
   *cycle_ratio(j,5)=T;
   *cycle_ratio(j,6)=a_i;
   *cycle_ratio(j,7)=RAOz;
   *cycle_ratio(j,8)=z_w;
   *cycle_ratio(j,9)=F_I;
   *cycle_ratio(j,10)=F_B;
   *cycle_ratio(j,11)=x_w;

   A_i=a_i;wn=w;dump(2);

  j++;
   }

damage=0;
int block_count=0;


block_count = 0.3/damage; // 0.3 = failure damage

if (block_count>2E4) AfxMessageBox("MAX BLOCK COUNT. POSSIBLE INFINITE LIFE");

life = block_count * max_T/(60*60*24*365.25);  // in seconds


//dump(3);
dump(4);

}

char buf[80];
sprintf(buf," Fatigue Life= %f years damage=%f", life,damage);
```

```
AfxMessageBox(buf);


return life;    // in years

}


// will run finite element and blkr_frame modules to do parametric stress calc
// final stress will be a weighted average obtained by running
// at differenct confidence intervals of the ITTC wave spectra...

void fatigue_model::get_stress_range(float L_SHIP,float a_i, float T)
{
int L,N;
float lo_toe_stress;
FRAME_AUTOGEN=1;

blkr_frame fr(theApp.h.model);
fr.fill_blkr_frame(theApp.h.model,L_SHIP);
theApp.h.initialize();
theApp.h.build_K();
fr.define_load(theApp.h.model,a_i,T);
fr.define_bc(theApp.h.model);

RAOx = fr.RAOx;
RAOz = fr.RAOz;
F_I = fr.F_I;
F_B = fr.F_B;
M_NET=fr.M_NET;
z_w = fr.z_w;
x_w = fr.x_w;
a_22=fr.a_11_x;
b_22=fr.b_22_x;
m_x=fr.m_x;
k_x=fr.k_x;
theApp.h.build_Q();
theApp.h.build_U();
theApp.h.solve();

// low toe stress is taken as the tensile force in the spring element
// corresponding to the low toe divided by its area

N = theApp.h.model.n_elements - 1;
float A = *theApp.h.model.el_def(N,Area);
float To =*theApp.h.T(N,1);

s_max=To/(A);

fr.define_load(theApp.h.model,-1*a_i,T);
fr.define_bc(theApp.h.model);

theApp.h.build_Q();
theApp.h.build_U();
theApp.h.solve();

// low toe stress is taken as the tensile force in the spring element
// corresponding to the low toe divided by its area

N = theApp.h.model.n_elements - 1;
A = *theApp.h.model.el_def(N,Area);
To =*theApp.h.T(N,1);

s_min = To/(A);


SHOW_DISPLACEMENTS=1;
}
```

```
//given stress range, and cycle count, it returns the damage accumulated
// based on S-N curves from the SSC
float fatigue_model::get_cycle_damage(float s_max,float s_min,float ni)
{
// everything else is N,mm. SSC fatigue curves are in ksi.
// therefore must convert N/mm^->ksi by dividing by 6.896
s_max/=6.896;
s_min/=6.896;

float del_S = s_max - s_min;
float R = s_min/s_max;
float C = S_N_constant;
float m = S_N_slope;
float b = -1/6*log10(2+100/(1.5*material_Su));

// root bisection to find N
// only works with a monotonically decreasing function which
// passes through zero and for which f(x_min)>0, f(x_max)<0

float x_min = 1;
float x_max = 1E30;
float x = (x_min+x_max)/2;
int count = 0;

while (fabs(residual (R,x,b,del_S))>0.0001 && count <10000)
{
if (residual(R,x,b,del_S)>0) x_min = x;
if (residual(R,x,b,del_S)<0) x_max = x;
x = (x_min+x_max)/2;
count+=1;
}
// calculate cycle ratio and return
return ni/x;

}

float fatigue_model::residual(float R, float N, float b, float del_S)
{
float rr= (1+pow(2*N,b)) / (1+(1+R)/(1-R)*pow(2*N,b));
rr*=S_N_constant/pow(N,S_N_slope);
rr-=del_S;
return rr;
}

wave_model::wave_model()
{
// init wave data...
// confidence interval, wmi param 1, wmi param 2, weighting factor wfi
wm[1][1]=0.95; wm[1][2]=0.048; wm[1][3]=8.75;wm[1][4]=0.05;
wm[2][1]=0.85; wm[2][2]=0.054; wm[2][3]=8.44;wm[2][4]=0.05;
wm[3][1]=0.75; wm[3][2]=0.061; wm[3][3]=8.07;wm[3][4]=0.0875;
wm[4][1]=0.5;  wm[4][2]=0.069; wm[4][3]=7.77;wm[4][4]=0.1875;
wm[5][1]=0;    wm[5][2]=0.079; wm[5][3]=7.63;wm[5][4]=0.25;
wm[6][1]=0.5;  wm[6][2]=0.099; wm[6][3]=6.87;wm[6][4]=0.1875;
wm[7][1]=0.75; wm[7][2]=0.111; wm[7][3]=6.67;wm[7][4]=0.0875;
wm[8][1]=0.85; wm[8][2]=0.119; wm[8][3]=6.65;wm[8][4]=0.05;
wm[9][1]=0.95; wm[9][2]=0.134; wm[9][3]=6.41;wm[9][4]=0.05;

//wave height distribution and percentage occurance
wh[1][1]=0.5; wh[1][2]=0.0503;  wh[10][1]= 9.5; wh[10][2]=0.0079;
wh[2][1]=1.5; wh[2][2]=0.2665;  wh[11][1]=10.5; wh[11][2]=0.0054;
wh[3][1]=2.5; wh[3][2]=0.2603;  wh[12][1]=11.5; wh[12][2]=0.0029;
wh[4][1]=3.5; wh[4][2]=0.1757;  wh[13][1]=12.5; wh[13][2]=0.0016;
wh[5][1]=4.5; wh[5][2]=0.1014;  wh[14][1]=13.5; wh[14][2]=0.00074;
wh[6][1]=5.5; wh[6][2]=0.0589;  wh[15][1]=14.5; wh[15][2]=0.00045;
wh[7][1]=6.5; wh[7][2]=0.0346;  wh[16][1]=15.5; wh[16][2]=0.00020;
wh[8][1]=7.5; wh[8][2]=0.0209;  wh[17][1]=16.5; wh[17][2]=0.00012;
wh[9][1]=8.5; wh[9][2]=0.0120;  wh[18][1]=17.5; wh[18][2]=0.00009;
```

```
}

//calculates weighted average value of ITTC wave spectra...
float wave_model::S(float w, float Hs)
{
float wm_i,S_w;
S_w = 0;
for ( int i = 1; i<=9;i++) // final S_w is weighted average of
   {
   wm_i = wm[i][2]*(wm[i][3]-log(Hs)); // note log = ln in Symantec C++
   S_w += 0.3125*pow(wm_i/w,4)*Hs*Hs/w*exp(-1.25*pow(wm_i/w,4)) * wm[i][4];
   }
return S_w;
}

//calculates weighted average wave amplitude at each component frequency w
float wave_model::component_amplitude(float w)
{
float a_i =0;
for(int i= 1;i<=18;i++) a_i+=sqrt(2*S(w,wh[i][H_S])*del_w)*wh[i][occurance];
return a_i;
}
```

FILENAME: DOC_FEM.H

```
// header file for finite element routines

enum{Elcment_type=1,Node1,Node2,Elasticity,Area,
        Inertia,node_count,element_node_dof,I_y,I_z,G_ndx,J_ndx};
enum{node_x=1,node_y,node_z,node_dof,global_ndx};

enum{unknown=0,known=1};
enum{truss_2D=1,truss_3D,beam_2D,beam_3D,plate_2D};

//-----------------------------------------------------------------
// base model storage class
class FEM_model_data
{
public:
int n_dim;
int n_nodes, n_elements,n_loads,n_bc;
int g_dof,i_dof,j_dof;
int el_dof,el_n_nodes,el_nodal_dof;
_matrix<float>* coord;
_matrix<float>* el_def;
_matrix<float>* load;
_matrix<float>* bc;

void initialize();

};

//-----------------------------------------------------------------//
// FEM model manager class

class FEM_model
{
public:

FEM_model_data model;

_matrix<float>* K;
_matrix<float>* Q;
_matrix<float>* U;
_matrix<float>* SOLN;
_matrix<float>* T;

FEM_model();
~FEM_model();
void initialize();
int build_K();
void build_Q();
void build_U();
int solve();
int dump();
};


//-----------------------------------------------------------------//
//2D_rotation matrix
// valid only for two noded elements eg truss and beam
//usage: initialize by supplying four coors or two nodes
//      after init, the return matrix by flashing
// it is a utility only and must be supplied all parameters from model

class T_2D
{
public:
_matrix<float>* t;
T_2D();
void init(float x1, float y1, float x2, float y2);
};
```

```
class T_2D_Beam
{
public:
_matrix<float>* t;
T_2D_Beam();
void init(float x1, float y1, float x2, float y2);
};

class k_Tr_2D
{
public:
_matrix<float>* k;
k_Tr_2D();
void init(float E, float A, float L);
};


class k_Beam_2D
{
public:
_matrix<float>* k;
k_Beam_2D();
void init(float E, float A, float I,float L);
};

class k_Beam_3D
{
public:
_matrix<float>* k;

k_Beam_3D();
void init(float E, float A, float Iy, float Iz, float G, float J,float L);
};

//-------------------------------------------------------------//
//creates global stiffness matrix for local element, ready to assemble into global
class K_Tr_2D
{
public:
_matrix<float>* K;
int e_dof, n_dof, N[3];

K_Tr_2D();
void init(float X1,float Y1,float X2,float Y2,float E, float A);
void init(FEM_model_data& model, int i);
};

class K_Beam_2D
{
public:
_matrix<float>* K;
int e_dof,n_dof,N[3];

K_Beam_2D();
void init(float X1,float Y1,float X2,float Y2,float E, float A,float I);
void init(FEM_model_data& model, int i);
};
```

FILENAME: DOC_FEM.CPP

```
//*********************************************************************
//FEM_model
FEM_model::FEM_model(){}

FEM_model::~FEM_model() {}

int FEM_model::solve()
{int i,j;
BOOL zero;

_matrix<float> K_tmp = *K;


//parse system of equations
// check for zero rows in the global stiffness matrix...

for(i=1;i<=model.g_dof;i++)
   {
   for(zero=TRUE,j=1;j<=model.g_dof;j++) if (*K(i,j)!=0) zero=FALSE;
   if(zero) {*U(i,1)=known;} //if any found set as dummy...
   }

for (i=1;i<=model.g_dof;i++) *SOLN(i,1)=*Q(i,2);

for (i=1;i<=model.g_dof;i++)
{
   if( *U(i,1)==known)
            {
            if(*U(i,2)==0) *SOLN(i,1)=1; //set force vector component=dummy
   else // account for fixed displacement as known force F = k * x
                        for(j=1;j<=model.g_dof;j++) *SOLN(j,1) -= *K_tmp(j,i) * *U(i,2);

            //set as diagonal entry in stiffness matrix
            //first blank row and column
            for(j=1;j<=model.g_dof;j++){ *K_tmp(i,j)=0; *K_tmp(j,i)=0; }
            *K_tmp(i,i)=1;
            } // end if *U known
}// end dofs, end parse.

//solve global system for unknown displacements
K_tmp.GJ_elim(*SOLN);

//replace dummies with actual displacements and build displacement matrix U
for(i=1;i<=model.g_dof;i++) if(*U(i,1)==unknown) *U(i,2)= *SOLN(i,1);

// calculate nodal forces
 *Q = *K * *U;

//calculate tension
for ( i=1;i<=model.n_elements;i++)
{
int  N1=*model.el_def(i,Node1); int  N2=*model.el_def(i,Node2);
float X1=*model.coord(N1,1);    float X2=*model.coord(N2,1);
float Y1=*model.coord(N1,2);    float Y2=*model.coord(N2,2);
float Z1=*model.coord(N1,3);    float Z2=*model.coord(N2,3);
float DX = X2-X1; float DY = Y2-Y1; float DZ=Z2-Z1;

float L = pow(DX*DX+DY*DY+DZ*DZ,0.5);
float E = *model.el_def(i,Elasticity);
float A = *model.el_def(i,Area);
float I = *model.el_def(i,Inertia);
float Iy= *model.el_def(i,I_y);
float Iz= *model.el_def(i,I_z);
float G = *model.el_def(i,G_ndx);
float J = *model.el_def(i,J_ndx);

if(model.el_def->x[i][Element_type]==truss_2D)
```

```
            {
   //create element and transformation matrices
   k_Tr_2D k_el; k_el.init(E,A,L);
   T_2D T2D; T2D.init(X1,Y1,X2,Y2);
   //build local displacement table
   _matrix<float> Ul(4,1);
   int base= *model.coord(N1,global_ndx);
   *Ul(1,1)=*U(base,2); *Ul(2,1)=*U(base+1,2);
   base= *model.coord(N2,global_ndx);
   *Ul(3,1)=*U(base,2); *Ul(4,1)=*U(base+1,2);
   //calculate local forces
   _matrix<float> Tl(2,1); Tl = *k_el.k * (*T2D.t * Ul);
   //copy to global tension vector
   *T(i,1)= *Tl(1,1); *T(i,4)= *Tl(2,1);
            }//end truss_2D
if(model.el_def->x[i][Element_type]==beam_2D)
   {
   //create element and transformation matrices
   k_Beam_2D k_el; k_el.init(E,A,I,L);
   T_2D_Beam T2D; T2D.init(X1,Y1,X2,Y2);
   //build local displacement table
   _matrix<float> Ul(6,1);
   int base= *model.coord(N1,global_ndx);
   *Ul(1,1)=*U(base,2); *Ul(2,1)=*U(base+1,2); *Ul(3,1)=*U(base+2,2);
   base= *model.coord(N2,global_ndx);
   *Ul(4,1)=*U(base,2); *Ul(5,1)=*U(base+1,2); *Ul(6,1)=*U(base+2,2);
   //calculate local forces
   _matrix<float> Tl(6,1); Tl = *k_el.k * (*T2D.t * Ul);
   //copy to global tension vector
   *T(i,1)= *Tl(1,1); *T(i,2)= *Tl(2,1);*T(i,3)= *Tl(3,1);
   *T(i,4)= *Tl(4,1); *T(i,5)= *Tl(5,1);*T(i,6)= *Tl(6,1);
   } // end beam_2D

} // end elements

return 1;
}

void FEM_model::build_U()
{
int n,bc_node,bc_dof;

for(int i=1;i<=model.n_bc;i++)
   {
   bc_node=*model.bc(i,1); bc_dof= *model.bc(i,2);
   if(bc_node !=0)
      {
      n=*model.coord(bc_node,global_ndx) + bc_dof-1;
      *U(n,2)=*model.bc(i,3);  *U(n,1)=known;
      *Q(n,1)=unknown;
      } // end if
   } //end elements
}

void FEM_model::build_Q()
{
int i,n,load_node,load_dof;

for(i=1;i<=model.n_loads;i++)
            {
            load_node=*model.load(i,1); load_dof =*model.load(i,2);
            if(load_node !=0)
               {
               n =*model.coord(load_node,global_ndx)+load_dof-1;
      *Q(n,1)=known;  *Q(n,2)=*model.load(i,3);
      *U(n,1)=unknown;
      } // end if
            }// end elements
}
```

```
int FEM_model::build_K()
{
int i,j,k,l,m,node_count,n_dof;
int N1,N2,G1,G2,L1,L2,I[3];

for (i=1; i<=model.n_elements;i++)
            {
            switch ((int) *model.el_def(i,Element_type))
            {
            case truss_2D:
               {
               K_Tr_2D k_e; k_e.init(model,i);
               K_E=k_e.K; node_count = k_e.e_dof; n_dof = k_e.n_dof;
               break;
               } //end case truss_2D

   case beam_2D:
            {
            K_Beam_2D k_e; k_e.init(model,i);
            K_E = k_e.K; node_count = k_e.e_dof; n_dof = k_e.n_dof;
            break;
      } // end case beam_2D
}//end switch

   N1 = *model.el_def(i,Node1);  //get node numbers
   N2 = *model.el_def(i,Node2);
   I[1] = *model.coord(N1,global_ndx)-1; //get global indices
   I[2] = *model.coord(N2,global_ndx)-1;

   // distribute globalized element into global stiffness matrix...
   for (j=1;j<=node_count;j++) // for each row of nodes
      for (k=1;k<=node_count;k++) // and for each col of nodes
         {L1 = (j-1)*n_dof;L2 = (k-1)*n_dof;
          for (l=1;l<=n_dof;l++)
            for(m=1;m<=n_dof;m++) *K(I[j]+l,I[k]+m) += *K_E(L1+l,L2+m);
         }
   } //next element
return 1;
}

void FEM_model_data::initialize()
{

CStdioFile in;
char buf[80];
int i,j,n1,n2;

// file readin
CFileDialog FileDialog(TRUE,"TXT","*.TXT",OFN_HIDEREADONLY,
   "Text Files (*.TXT)|*.TXT|*.TXT|All Files(*.*|*.*|");

if(FileDialog.DoModal()!=IDOK) {TRACE0("File Error"); return;}

if(!in.Open(FileDialog.GetPathName(),
   CFile::modeRead|CFile::modeReadWrite)) return;


in.ReadString(buf,70);
sscanf(buf,"%i",&n_dim);

in.ReadString(buf,70);
sscanf(buf,"%i,%i,%i,%i",&n_nodes,&n_elements,&n_loads,&n_bc);

//now that sizes are known, dynamical allocate:
coord  = (_matrix<float>*) new _matrix<float>(  n_nodes,5);
el_def = (_matrix<float>*) new _matrix<float>(n_elements,12);
load   = (_matrix<float>*) new _matrix<float>(  n_loads,3);
bc     = (_matrix<float>*) new _matrix<float>(     n_bc,3);
```

```
for(i=1;i<=n_nodes;i++)
            {
            in.ReadString(buf,79);
            if(n_dim==2) sscanf(buf,"%g,%g",&coord->x[i][1],&coord->x[i][2]);
            else
            sscanf(buf,"%g,%g,%g",&coord->x[i][1],&coord->x[i][2],&coord->x[i][3]);
            }

for(i=1;i<=n_elements;i++)
            {
            in.ReadString(buf,79);
            sscanf(buf,"%g,%g,%g,%g,%g,%g,%g,%g,%g,%g",&el_def->x[i][1],&el_def->x[i][2],
            &el_def->x[i][3],&el_def->x[i][4],&el_def->x[i][5],&el_def->x[i][6],
            &el_def->x[i][9],&el_def->x[i][10],&el_def->x[i][11],&el_def->x[i][12]);
  // define number of nodes per element and dof per node...
            if(el_def->x[i][1]==truss_2D) {el_def->x[i][7]=2;el_def->x[i][8]=2;}
            if(el_def->x[i][1]==truss_3D) {el_def->x[i][7]=2;el_def->x[i][8]=3;}
            if(el_def->x[i][1]==beam_2D)  {el_def->x[i][7]=2;el_def->x[i][8]=3;}
            if(el_def->x[i][1]==beam_3D)  {el_def->x[i][7]=2;el_def->x[i][8]=6;}
  }

for(i=1;i<=n_loads;i++)
            {
            in.ReadString(buf,79);
            sscanf(buf,"%g,%g,%g",&load->x[i][1],&load->x[i][2],&load->x[i][3]);
            }

for(i=1;i<=n_bc;i++)
            {
            in.ReadString(buf,79);
            sscanf(buf,"%g,%g,%g",&bc->x[i][1],&bc->x[i][2],&bc->x[i][3]);
            }

//make sure dof count for each node is maximised....
for(i=1;i<=n_elements;i++)
            {
            n1= *el_def(i,Node1); n2=*el_def(i,Node2);
            switch ((int) *el_def(i,Element_type))
            {
            case truss_2D:
               *coord(n1,node_dof)=max(2,*coord(n1,node_dof));
               *coord(n2,node_dof)=max(2,*coord(n2,node_dof));
               break;
            case truss_3D:
               *coord(n1,node_dof)=max(3,*coord(n1,node_dof));
               *coord(n2,node_dof)=max(3,*coord(n2,node_dof));
               break;
            case  beam_2D:
               *coord(n1,node_dof)=max(3,*coord(n1,node_dof));
               *coord(n2,node_dof)=max(3,*coord(n2,node_dof));
               break;
            default: TRACE0("INVALID ELEMENT TYPE ON INIT");
            } // end switch
  } // end nodes

// calculate global number of degrees of freedom
for(g_dof=0,i=1;i<=n_nodes;i++) g_dof += *coord(i,node_dof);

//set global index for each node, start index at one
for(*coord(1,global_ndx)=1,i=2;i<=n_nodes;i++)
      *coord(i,global_ndx)= *coord(i-1,global_ndx)+ *coord(i-1,node_dof);


in.Close();
return;
}


void FEM_model::initialize()
{
```

```
// get model data...
if (FRAME_AUTOGEN==1)
   {
   //blkr_frame frame(model);
   //frame.init(model,theApp.LBP);

   }
else
   model.initialize();


// now that model_data is initialized and
// now that size is known, allocate global K,Q and U matrices

// need to allow for multiple runs with change of parameters, so check
// if matrices previously allocated and if so delete and reallocate
// with new sizes
if( K != NULL) delete K;
if( Q!= NULL) delete Q;
if( U != NULL) delete U;
if( SOLN != NULL) delete SOLN;
if( T != NULL) delete T;


K  =(_matrix<float>*) new _matrix<float>(model.g_dof,model.g_dof);
Q  =(_matrix<float>*) new _matrix<float>(model.g_dof,2);
U  =(_matrix<float>*) new _matrix<float>(model.g_dof,2);
SOLN=(_matrix<float>*) new _matrix<float>(model.g_dof,1);
T  =(_matrix<float>*) new _matrix<float>(model.n_elements,6);

for(int i=1;i<=model.g_dof;i++) {Q->x[i][1]=known; U->x[i][1]=unknown;}

}

int FEM_model::dump()
{
CStdioFile out;
char buf[80];
int i,j;
// file dump
CFileDialog FileDialog(FALSE,"TXT","*.TXT",OFN_HIDEREADONLY,
   "Text Files (*.TXT)|*.TXT|*.TXT|All Files(*.*|*.*|");

if(FileDialog.DoModal()!=IDOK) {TRACE0("File Error"); return 1;}

if(!out.Open(FileDialog.GetPathName(),
   CFile::modeCreate|CFile::modeWrite)) return 1;


sprintf(buf,"Global Stiffness Matrix\n");
out.WriteString(buf);

for(i=1;i<=model.g_dof;i++)
   {
   for (j=1;j<=model.g_dof;j++){sprintf(buf,"%i %i =%12f\n",i,j,K->x[i][j]);
                                                    out.WriteString(buf);}
   }

sprintf(buf,"Global Load Vector\n");
out.WriteString(buf);

for(i=1;i<=model.g_dof;i++)
   {
   for (j=1;j<=2;j++){ sprintf(buf,"%i %i = %12f\n",i,j,Q->x[i][j]);
                                                    out.WriteString(buf);}
   }


sprintf(buf,"Ship Length: %i \n",theApp.LBP);
```

```
out.WriteString(buf);
sprintf(buf,"Number of nodes: %i  \n",model.n_nodes);
out.WriteString(buf);
sprintf(buf,"Number of elements:%i  \n",model.n_elements);
out.WriteString(buf);
sprintf(buf,"Number of boundary conditions:%i  \n",model.n_bc);
out.WriteString(buf);
sprintf(buf,"Number of nodal loads:%i  \n",model.n_loads);
out.WriteString(buf);
sprintf(buf,"Node   x   y    z\n");
out.WriteString(buf);
for(i=1;i<=model.n_nodes;i++)
            {
            sprintf(buf,"%i  %12g  %12g %12g\n",i,model.coord->x[i][1],model.coord->x[i][2],
                        model.coord->x[i][3]);
            out.WriteString(buf);
            }
sprintf(buf,"\nElement Type  Node_i  Node_j  E    A    I\n");
out.WriteString(buf);
for(i=1;i<=model.n_elements;i++)
            {
            sprintf(buf,"%i,%9g,%9g,%9g,%9g,%9g,%9g\n",i,
                                                model.el_def->x[i][1],model.el_def->x[i][2],
                                                model.el_def->x[i][3],model.el_def->x[i][4],
                                                model.el_def->x[i][5],model.el_def->x[i][6]);
            out.WriteString(buf);
            }


sprintf(buf,"\nNode   DOF   Load \n");
out.WriteString(buf);
for(i=1;i<=model.n_loads;i++)
  {
            if(model.load->x[i][1]!=0 && model.load->x[i][2]!=0)
            {
            sprintf(buf,"%g,%g,%g\n",model.load->x[i][1],
                  model.load->x[i][2],model.load->x[i][3]);
            out.WriteString(buf);
            }
  }


sprintf(buf,"\nBoundary Conditions: Node  DOF  VALUE \n");
out.WriteString(buf);
for(i=1;i<=model.n_bc;i++)
 {
            if( model.bc->x[i][1]!=0 && model.bc->x[i][2]!=0)
            {
            sprintf(buf,"%g,%g,%g\n",model.bc->x[i][1],model.bc->x[i][2],model.bc->x[i][3]);
            out.WriteString(buf);
            }
 }


sprintf(buf,"Internal Reactions \nElement T1 V1  M1  T2 V2  M2\n");
out.WriteString(buf);

for(i=1;i<=model.n_elements;i++)
   {
   sprintf(buf,"%i: %9g %9g %9g %9g %9g %9g\n", i,T->x[i][1],T->x[i][2],
            T->x[i][3],T->x[i][4],T->x[i][5],T->x[i][6]);
   out.WriteString(buf);
   }


sprintf(buf,"Node:    T    V    M\n");
out.WriteString(buf);

for(i=1;i<=model.g_dof;i+=3)
   {
```

```
   sprintf(buf,"%3i: %12g %12g %12g\n"
                                                    ,i/3+1,Q->x[i][2],Q->x[i+1][2],Q->x[i+2][2]);
   out.WriteString(buf);
   }

sprintf(buf,"Node:  u  v  phi\n");
out.WriteString(buf);

for(i=1;i<=model.g_dof;i+=3)
   {
   sprintf(buf,"%3i: %12g %12g %12g\n",
                                                    i/3+1,U->x[i][2],U->x[i+1][2],U->x[i+2][2]);
   out.WriteString(buf);
   }

sprintf(buf," \n ");
out.WriteString(buf);
out.Close();
return 1;
}

//---------------------------------------------------------
//K_Tr_2D - globalized element stiffness matrix
K_Tr_2D::K_Tr_2D()
{e_dof=2; n_dof=2; K = (_matrix<float>*) new _matrix<float>(4,4); }

void K_Tr_2D::init(FEM_model_data& model, int i)
{
N[1]=*model.el_def(i,Node1);     N[2]=*model.el_def(i,Node2);
float X1=*model.coord(N[1],1);   float X2=*model.coord(N[2],1);
float Y1=*model.coord(N[1],2);   float Y2=*model.coord(N[2],2);
float Z1=*model.coord(N[1],3);   float Z2=*model.coord(N[2],3);
float DX=X2-X1; float DY=Y2-Y1; float DZ=Z2-Z1;
float L=pow(DX*DX+DY*DY+DZ*DZ,0.5);

float E =*model.el_def(i,Elasticity);
float A =*model.el_def(i,Area);

k_Tr_2D k_el;    k_el.init(E,A,L);
T_2D   T;        T.init(X1,Y1,X2,Y2);

*K = T.t->flash_T() * (*k_el.k * *T.t);
}

//---------------------------------------------------------
//K_Tr_3D - globalized element stiffness matrix

K_Tr_3D::K_Tr_3D()
{e_dof=2; n_dof=3; K = (_matrix<float>*) new _matrix<float>(6,6); }

void K_Tr_3D::init(FEM_model_data& model, int i)
{
N[1]=*model.el_def(i,Node1);     N[2]=*model.el_def(i,Node2);
float X1=*model.coord(N[1],1);   float X2=*model.coord(N[2],1);
float Y1=*model.coord(N[1],2);   float Y2=*model.coord(N[2],2);
float Z1=*model.coord(N[1],3);   float Z2=*model.coord(N[2],3);
float DX=X2-X1; float DY=Y2-Y1; float DZ=Z2-Z1;
float L=pow(DX*DX+DY*DY+DZ*DZ,0.5);

float E =*model.el_def(i,Elasticity);
float A =*model.el_def(i,Area);

k_Tr_2D   k_el;   k_el.init(E,A,L);
T_3D_Truss   T;     T.init(X1,Y1,Z1,X2,Y2,Z2);

*K = T.t->flash_T() * (*k_el.k * *T.t);
}

//K_Beam_2D - globalized element stiffness matrix
```

```
K_Beam_2D::K_Beam_2D()
{e_dof=2; n_dof=3; K = (_matrix<float>*) new _matrix<float>(6,6); }

void K_Beam_2D::init(FEM_model_data& model, int i)
{
N[1]=*model.el_def(i,Node1);   N[2]=*model.el_def(i,Node2);
float X1=*model.coord(N[1],1);  float X2=*model.coord(N[2],1);
float Y1=*model.coord(N[1],2);  float Y2=*model.coord(N[2],2);
float DX=X2-X1; float DY=Y2-Y1; float L=pow(DX*DX+DY*DY,0.5);

float E =*model.el_def(i,Elasticity);
float A =*model.el_def(i,Area);
float I =*model.el_def(i,Inertia);

k_Beam_2D k_el;      k_el.init(E,A,I,L);
T_2D_Beam   T;       T.init(X1,Y1,X2,Y2);

*K = T.t->flash_T() * (*k_el.k * *T.t);
}


///*****************************************************************
//k_Tr_2D is the local stiffness matrix for truss elements
k_Tr_2D::k_Tr_2D(){ k =(_matrix<float>*) new _matrix<float>(2,2);}

void k_Tr_2D::init(float E, float A, float L)
{
*k(1,1) =  E*A/L;      *k(1,2)=-E*A/L;
*k(2,1)= -E*A/L;       *k(2,2)=  E*A/L;
}
///*****************************************************************
//k_Beam_2D is the local stiffness matrix for 2D beam elements
k_Beam_2D::k_Beam_2D(){ k =(_matrix<float>*) new _matrix<float>(6,6);}

void k_Beam_2D::init(float E, float A, float I,float L)
{
*k(1,1)=E*A/L; *k(1,4)=-E*A/L;
*k(2,2)=12*E*I/(L*L*L);*k(2,3)=6*E*I/(L*L);*k(2,5)=- *k(2,2);*k(2,6)=*k(2,3);
*k(3,2)=*k(2,3); *k(3,3)=4*E*I/L;*k(3,5)=-1* *k(2,6); *k(3,6)=2*E*I/L;
*k(4,1)=*k(1,4); *k(4,4)= *k(1,1);
*k(5,2)=*k(2,5); *k(5,3)=*k(3,5); *k(5,5)=*k(2,2); *k(5,6)=*k(3,5);
*k(6,2)=*k(2,6); *k(6,3)=*k(3,6); *k(6,5)=*k(5,6); *k(6,6)=*k(3,3);
}



///*****************************************************************
//T_2D only valid for 2D truss elements
T_2D::T_2D() { t = (_matrix<float>*) new _matrix<float>(2,4);}

void T_2D::init(float x1, float y1, float x2, float y2)
{
float DX=x2-x1; float DY=y2-y1;
float L=pow(DX*DX+DY*DY,0.5);
float C=DX/L; float S=DY/L;

*t(1,1)=C;     *t(1,2)=S;
                    t->x[2][3]=C;     t->x[2][4]=S;
}

T_2D_Beam::T_2D_Beam() { t = (_matrix<float>*) new _matrix<float>(6,6);}

void T_2D_Beam::init(float x1, float y1, float x2, float y2)
{
float DX,DY,L,C,S;
DX=x2-x1;
DY=y2-y1;
L=pow(DX*DX+DY*DY,0.5);
```

```
C=DX/L;
S=DY/L;

*t(),1)=C;      *t(1,2)=S;
*t(2,1)=-1*S;   *t(2,2)=C;
                *t(3,3)=1;
                        *t(4,4)=C;      *t(4,5)=S;
                        *t(5,4)=-1*S;   *t(5,5)=C;
                                        *t(6,6)=1;
}
```