GlyphWorks Fatigue Theory Guide

About this Document

Copyright Notice

Warranty Notice

Liability Notice

Feedback

We value your feedback. If you have any documentation questions, comments, or concerns, please e-mail us at: techdocs@hbmncode.com

Customer Support

For issues involving the installation or use of your software, please e-mail us at: ncodesupport@hbmprenscia.com

or use the nCode website, www.ncode.com, to find support information for your country.

Contents

Figures

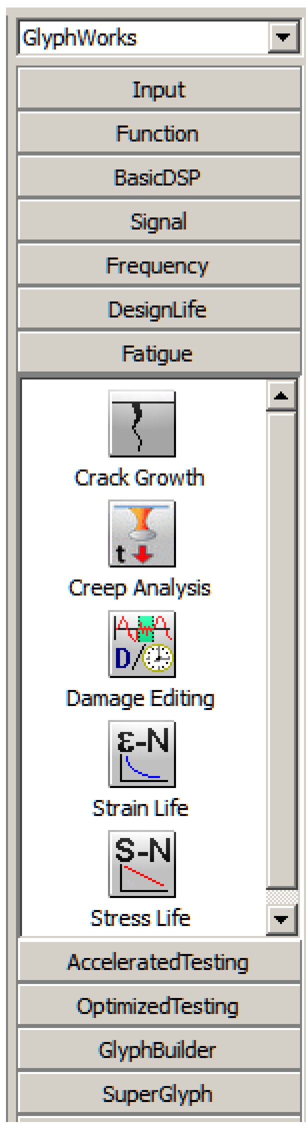
Tables

1 Introduction

This document describes the methods used within the GlyphWorks fatigue glyphs to make fatigue calculations from (typically) test-based data. Fatigue glyphs can also be used with hand-calculated stress results, or with data generated using finite element results, but these are typically performed using DesignLife. See the DesignLife Theory Guide, which is tailored to the way DesignLife does its calculations. Given the correct settings and equivalent inputs, DesignLife and GlyphWorks will give the same results, because the underlying calculations are performed by the same code.

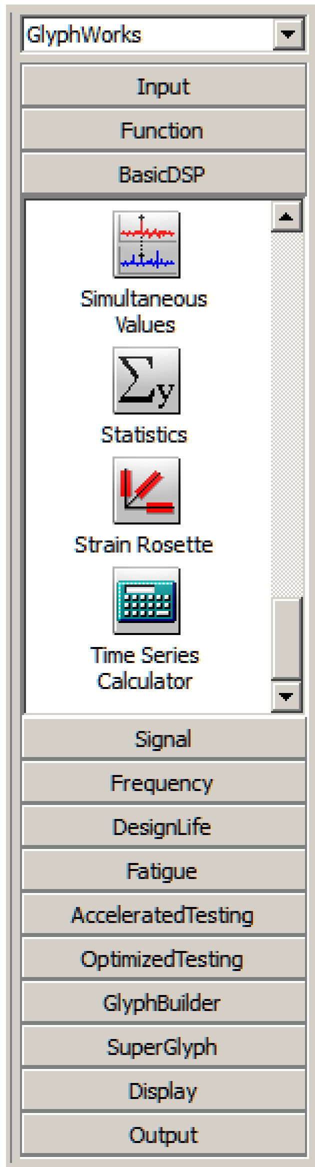
Fatigue glyphs are found on the GlyphWorks Fatigue palette, as shown in Figure 1-1.

Fig. 1-1 The Fatigue palette in standard GlyphWorks



In addition, this document describes the theory behind, and the use of, the Strain Gauge Rosette glyph. This glyph is often used to pre-process rosette data prior to a fatigue calculation. This glyph is available on the BasicDSP palette, as shown in Figure 1-2.

Fig. 1-2 The BasicDSP palette in standard GlyphWorks

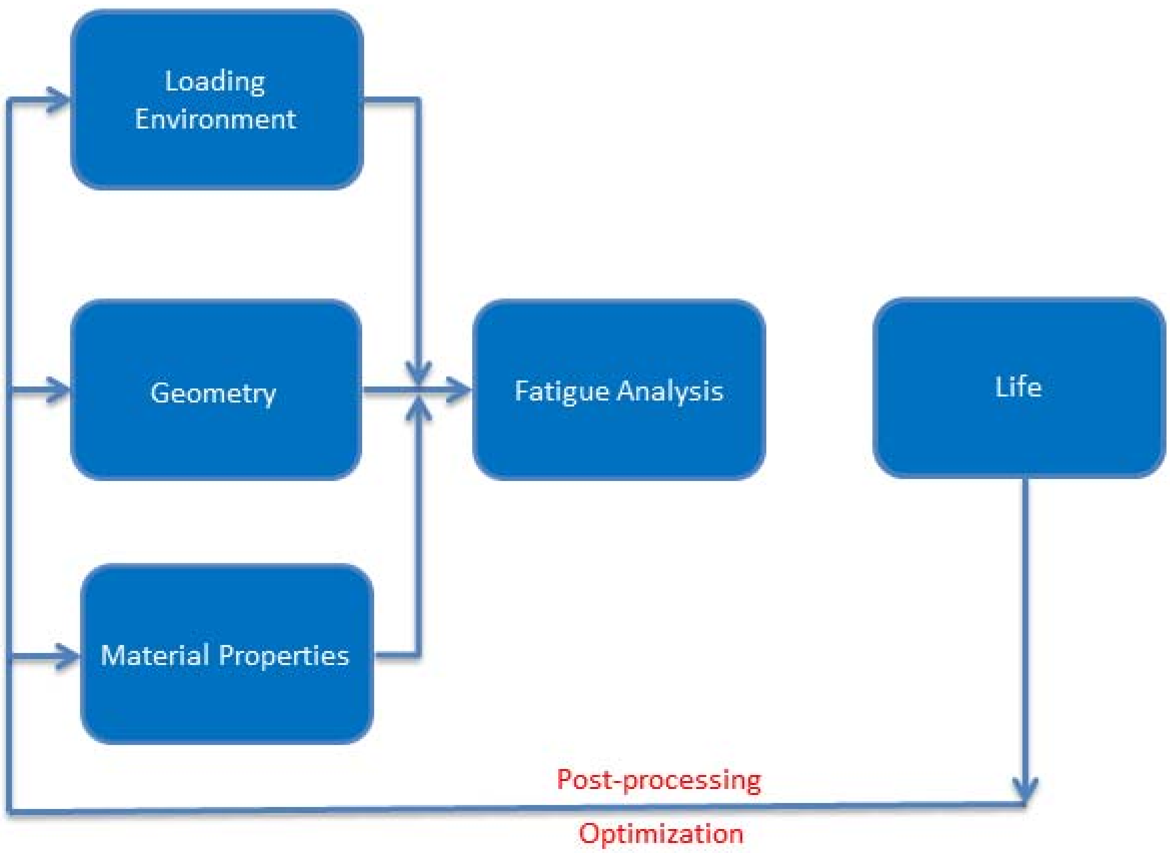


The GlyphWorks User Guide contains all the information needed to use fatigue glyphs within the GlyphWorks framework. The guide explains how those calculations are performed, but also references the input pipes and property settings that are used to effect a particular calculation.

1.1 Basics of Fatigue Calculations

Fundamentally, all fatigue calculations require information about the service loading, the geometry of the component, and the properties of the material. These can be shown in a simple five-box diagram:

Fig. 1-3 5 Box Diagram



The level of complication in each of these boxes varies from the simple to the highly complex – from calculations that can be done by hand or calculator, up to calculations that take hours to complete in a parallel-processed computing environment.

GlyphWorks offers an environment for fatigue calculations that allows complex calculations to be done using a simple user interface, one which in fact mimics closely the five-box diagram above. Some calculations have many possible options and properties that refine the calculation; these are given sensible defaults so that a reasonable answer can be given quickly and easily. However, with any fatigue calculation it is important that someone, often the person who configures GlyphWorks for others to use, understands these options so that the best estimate of fatigue damage can be obtained.

We strongly recommend that GlyphWorks users attend a theoretical fatigue training course. The tutor can explain any of the technical details and also describe the circumstances under which the individual methods and options are used in particular industries.

1.2 Further Reading

This document provides background in the methods and algorithms used in DesignLife. We can also recommend that users use other background material to assist in understanding, and in particular we recommend the following publication. Where our methods coincide with the book, we have referenced the appropriate page or section:

Metal Fatigue Analysis Handbook: practical problem-solving techniques for computer aided engineering / Yung-Li Lee, Mark E. Barkey, Hong-Tae Kang ISBN 978-0-12-385204-5.

2 Stress-Life Fatigue

The StressLife glyph is typically used for component tests in the medium to high cycle regimes and, unlike the Strain-Life method, does not correct for plasticity.

Strain gage data with units of strain, not stress, are valid inputs, as stresses can be calculated from strains via material properties.

The loading input stresses can be in the form of a time history, a list of cycles, or a rainflow histogram. The method by which the stress cycles information needed for the fatigue calculation is obtained using each of these loading types is outlined in "Loading" on page 14.

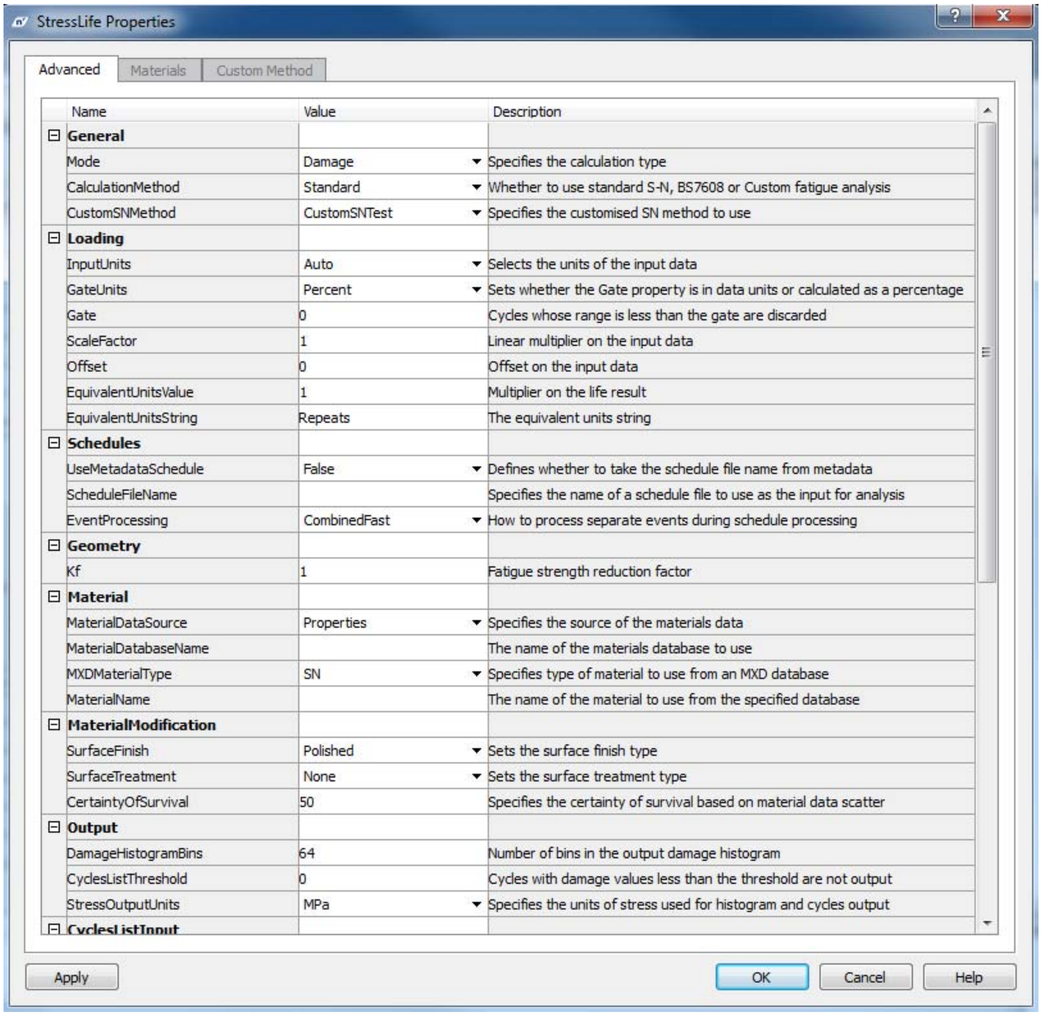
Stress time histories exported from DesignLife can be used directly in the StressLife glyph.

Note that stress life analysis is sometimes referred to as SN where S is stress and N is the life as a number of cycles to failure.

The materials data needed for stress life calculations comes typically from materials tests done under load control. The properties derived from those tests describe the relationship between the applied cyclic stress (calculated from applied load and geometry) and life as a number of cycles to failure. Static test properties, such as the Ultimate Tensile Strength, are also required as they are used by some of the correction algorithms. Full details of the materials properties are given in the Materials Data section.

Additional properties control corrections for geometry, surface effects, mean stress and other factors affecting fatigue life. These properties are found on the property form for the glyph as shown in Figure 2-1.

Fig. 2-1 Property sheet for the StressLife glyph

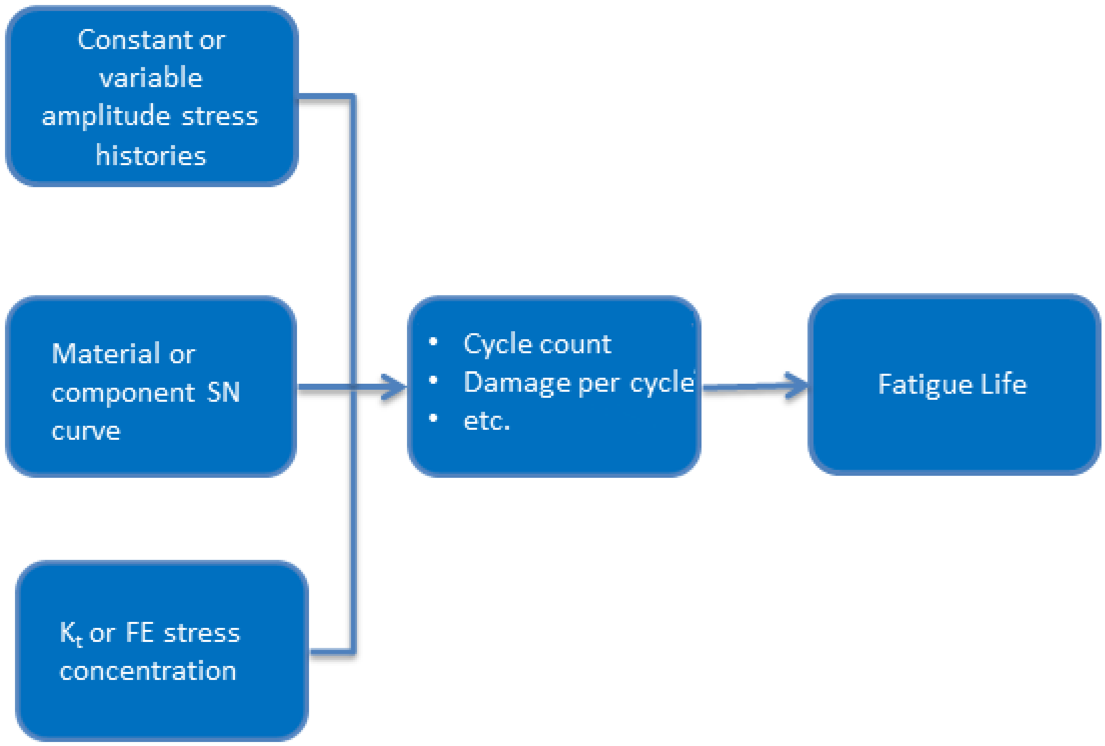


The basic process in stress-life calculations is as follows:

1. Make any corrections necessary to the materials curve to account for surface effects, geometry, etc. This usually occurs at the start of the calculation but may be done during the calculation on a cycle-by-cycle basis
2. Determine the rainflow cycles from the input loading that drive the fatigue process, converting to stress from strain if required.
3. Calculate and accumulate damage from the stress-life curve information, using linear damage accumulation (Miner's Rule).

The five-box diagram for stress life looks like this:

Fig. 2-2 5 Box Diagram for Stress



For further background on the S-N method, refer to page 116 in the Introduction section of the Stress-Life chapter in “Metal Fatigue Analysis Handbook”[1].

2.1 Loading

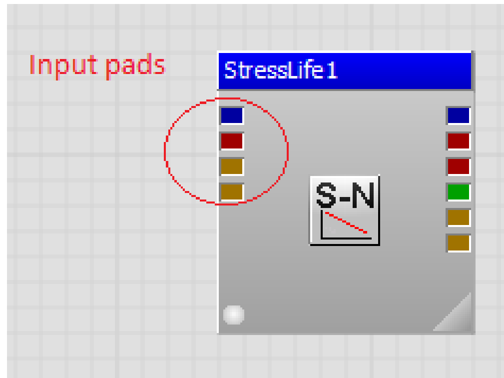
Loading in the StressLife glyph comes in any one of four forms; three are controlled by passing data in on input pipes, and the fourth by specifying a schedule (or duty cycle) via property settings. Figure 2-3 shows the input pads that are

---

1. Metal Fatigue Analysis Handbook: practical problem-solving techniques for computer aided engineering / Yung-Li Lee, Mark E. Barkey, Hong-Tae Kang ISBN 978-0-12-385204-5.

used for time series (blue), histogram (red) and multi-column data (the first of the two brown pads).

Fig. 2-3 Input pads on the StressLife glyph



Note The fourth pad brings in multi-curve materials data (see the "Materials Data" section).

2.1.1 Time Series Input

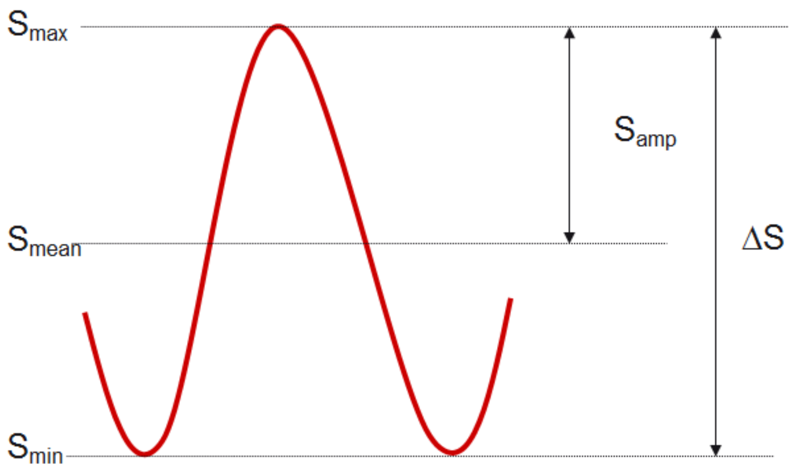
The Time Series input pad brings in multiple channels of data in the form of (typically) stress or strain versus time, with constant time intervals between the samples. Conversion of the units from the input units to MPa takes place by selecting the input unit on the property form:

⊟ **Loading**		
InputUnits	Auto	Selects the units of the input data

The Auto option attempts to recognize the input units from the metadata of the input data. This must be a unit of stress, or of strain, and be defined within nCode's units system. If the data is in strain units, it is converted to stress using Young's Modulus from the material properties and assumes both linearity and a uniaxial stress state.

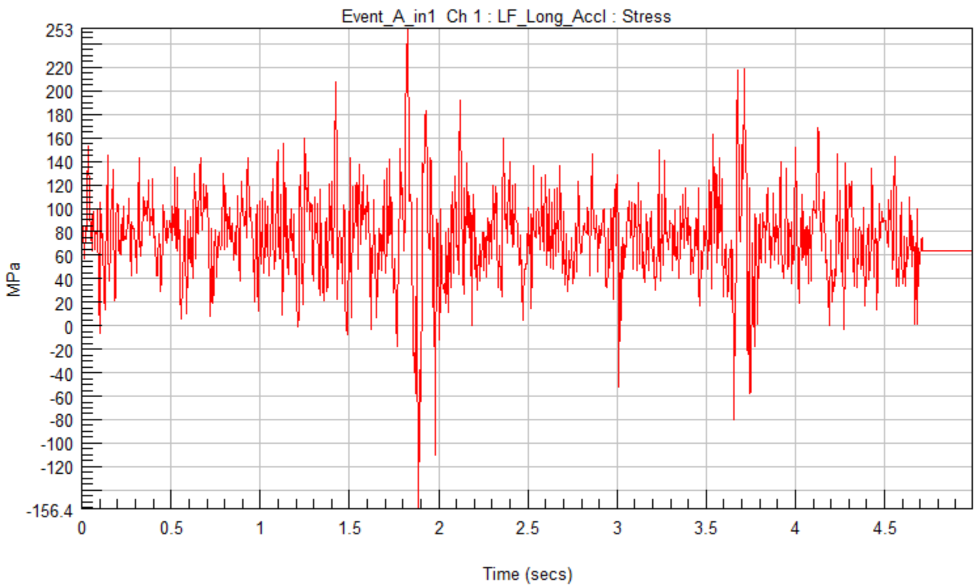
The simplest of loadings is the constant amplitude type, often but not necessarily sinusoidal. The shape of the time history, and the frequency content, has no effect in this type of fatigue calculation but the amplitude of the cycles is critical. The mean stress is also important, as will be explained later. The diagram shows a typical stress cycle, and shows the definitions of maximum stress, minimum stress, mean stress, stress amplitude, and stress range. These terms will all be used later.

Fig. 2-4 Typical stress cycle



If the time history is randomly loaded, rainflow counting then takes place to determine the cycles that drive fatigue damage. Here is an example of a “variable amplitude” stress loading history.

Fig. 2-5 Variable Amplitude stress loading history



Often time histories appear to be random but aren't. However, the process of rainflow cycle counting doesn't need to know this, and will calculate the cycles simply from the time history data it is given.

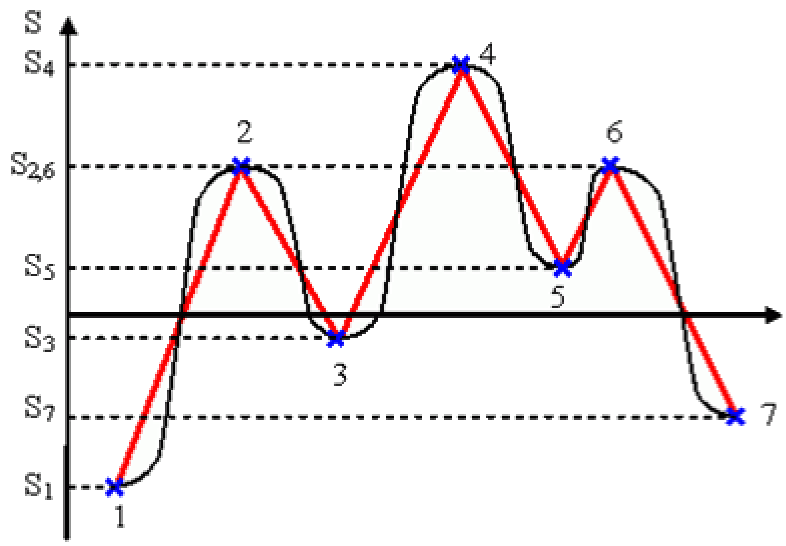
Rainflow cycle counting for the SN method is the same as for the basic counting method, as used in the Rainflow glyph, without the additional requirements that the strain-life method needs [i.e., to know the total (elastic-plastic) strain and stress for each cycle].

This means the method can start at the beginning of the data and work through to the end, without knowing the absolute maximum value in the whole history.

The process is as follows:

1. Reduce the stress history to a peak-valley sequence.

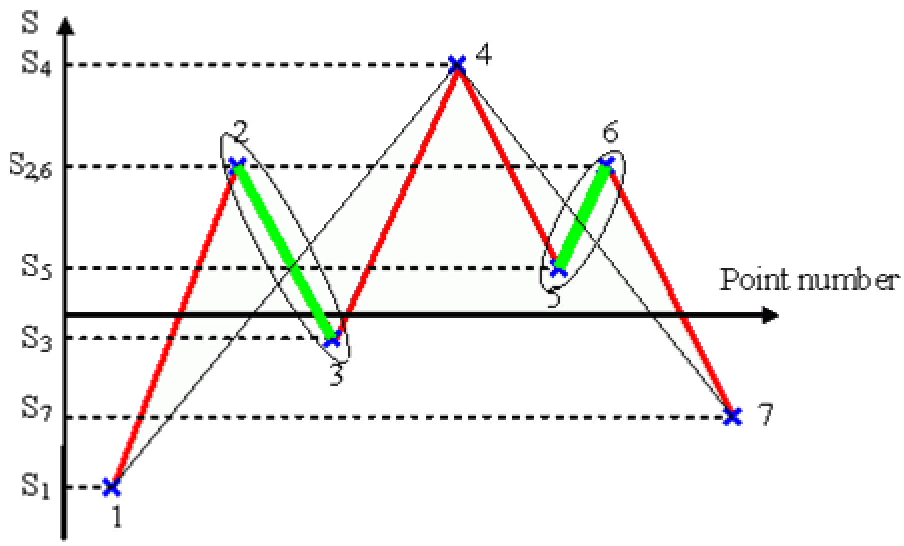
Fig. 2-6 Reduction of stress history to turning points



2. Cycles are identified by considering four points at a time. The logic is as follows:
   - If point n is a maximum, IF Sn ≥ Sn-2 AND Sn-1 ≥ Sn-3 then Sn-2 and Sn-1 make a cycle that can be extracted from the sequence.
   - If point n is a minimum, IF Sn ≤ Sn-2 AND Sn-1 ≤ Sn-3 then Sn-2 and Sn-1 make a cycle that can be extracted from the sequence.

After this logic has been applied to the whole sequence, any remaining points are known as the residual. This is illustrated below:

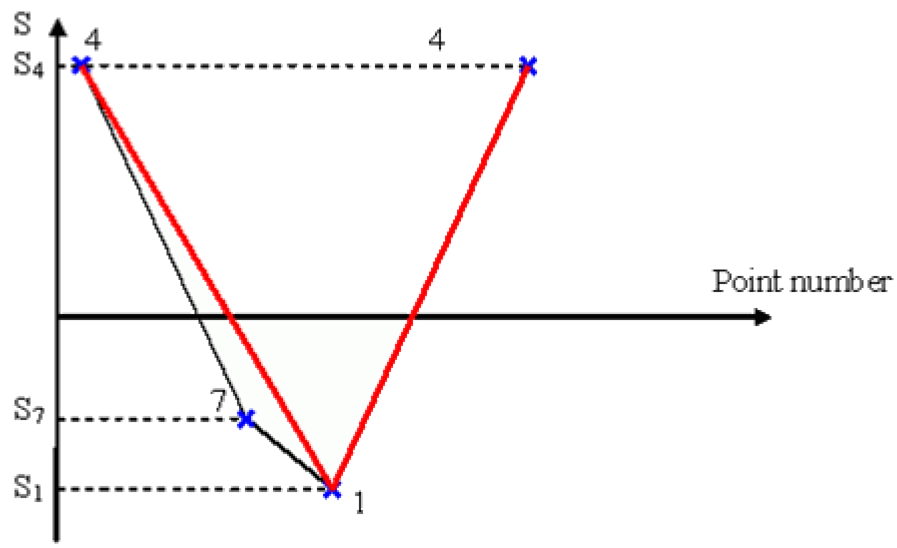
Fig. 2-7 Extraction of rainflow cycles



Two cycles are extracted from the sequence, represented by points 2 and 3 and by points 5 and 6. The remaining turning points, 1, 4, and 7, form the residual. (In the trivial case where the history contains only two or three points, there will be only one obvious cycle and no residual.)

3. Finally, the residual may be closed down. This is done by re-ordering the points to start from the point with the largest absolute value, and repeating this point at the end. Steps 1 and 2 are then repeated. In this case, point 7 is no longer a turning point, so we are left with the trivial case with only three points.

Fig. 2-8 Closing the residual



Note that if you start cycle counting a stress time history starting from the absolute maximum value, all cycles will close; that is, there will be no residual.

This is the four-point counting technique as described in "Metal Fatigue Analysis Handbook"[2], page 102.

The rainflow method is based on the hysteresis loops used to track stress and strain in the E-N method (see Figure 3-6), but the plasticity correction and the strains are not needed or used.

At this stage we have a "cycles list" with all the information needed to calculate the damage for each cycle and then the total damage. This is the same calculation whether the loading data is from time history, histogram, multi-column, or schedule files, and therefore this is described below in the section "Calculating Damage".

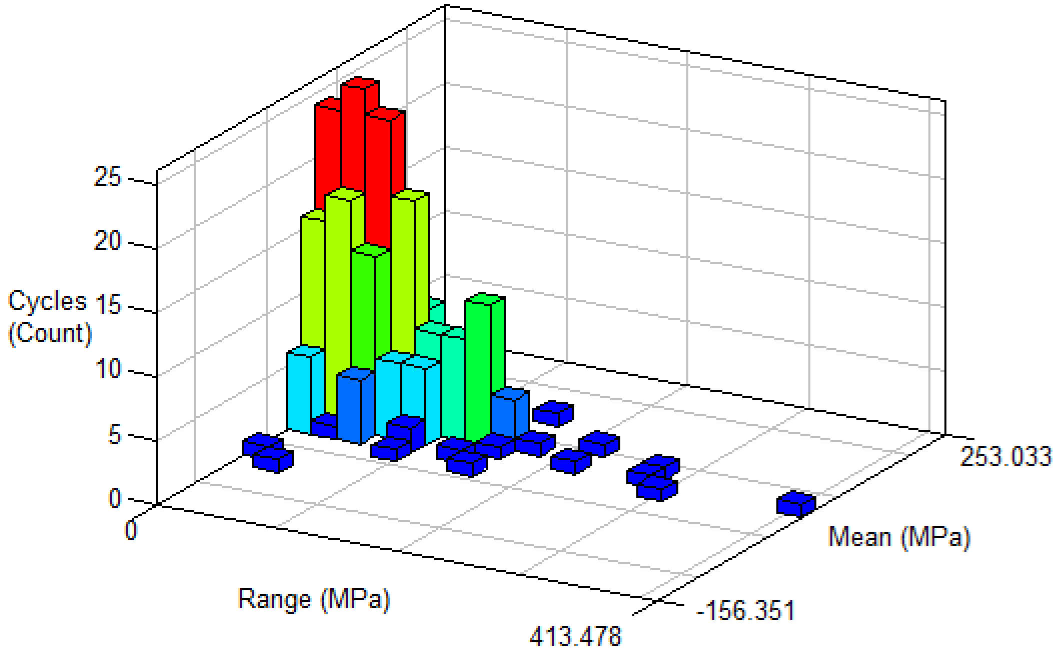
2.1.2 Histogram Input

The histogram input pad can bring in a matrix of pre-counted cycles (usually, but not necessarily, based on rainflow counting) that have been put into even or non-evenly spaced bins within one of four types of matrix, that characterize cycles using the mean, range, maximum and minimum (see Figure 2-4).

Range-Mean—One axis has bins split by range of the cycle (max-min) and the other axis is the mean of the cycle ((max+min)/2).

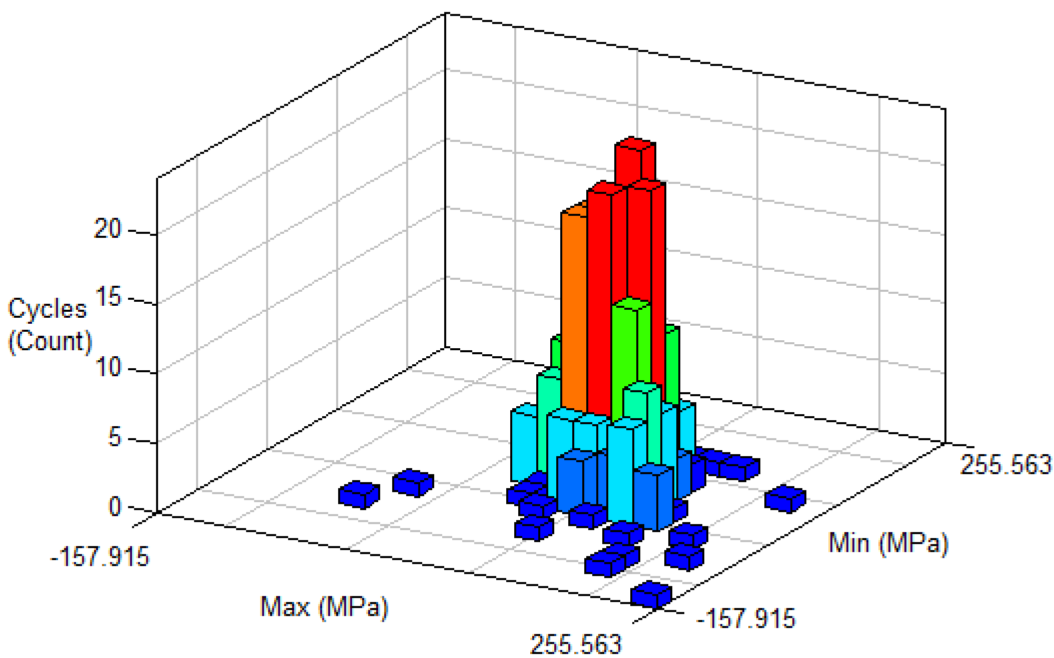
2. Metal Fatigue Analysis Handbook: practical problem-solving techniques for computer aided engineering / Yung-Li Lee, Mark E. Barkey, Hong-Tae Kang ISBN 978-0-12-385204-5.

Fig. 2-9 3d Histogram: Cycles - Range - Mean



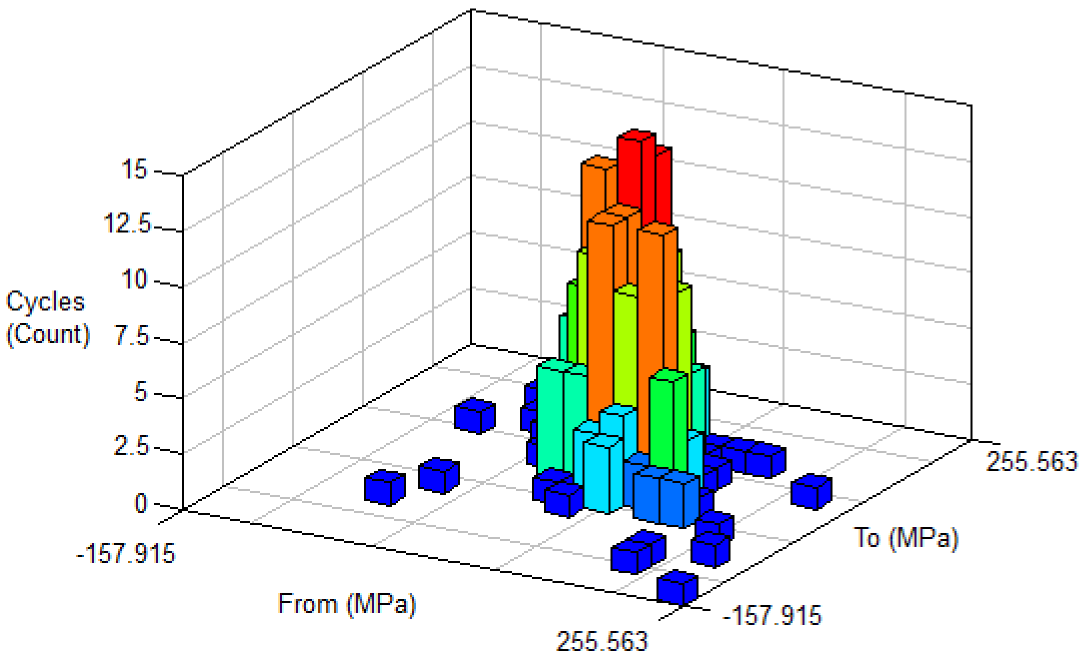
Max-Min—One axis is the maximum value of the cycle, the other the minimum.

Fig. 2-10 3D Histogram: Cycles - Maximum - Minimum



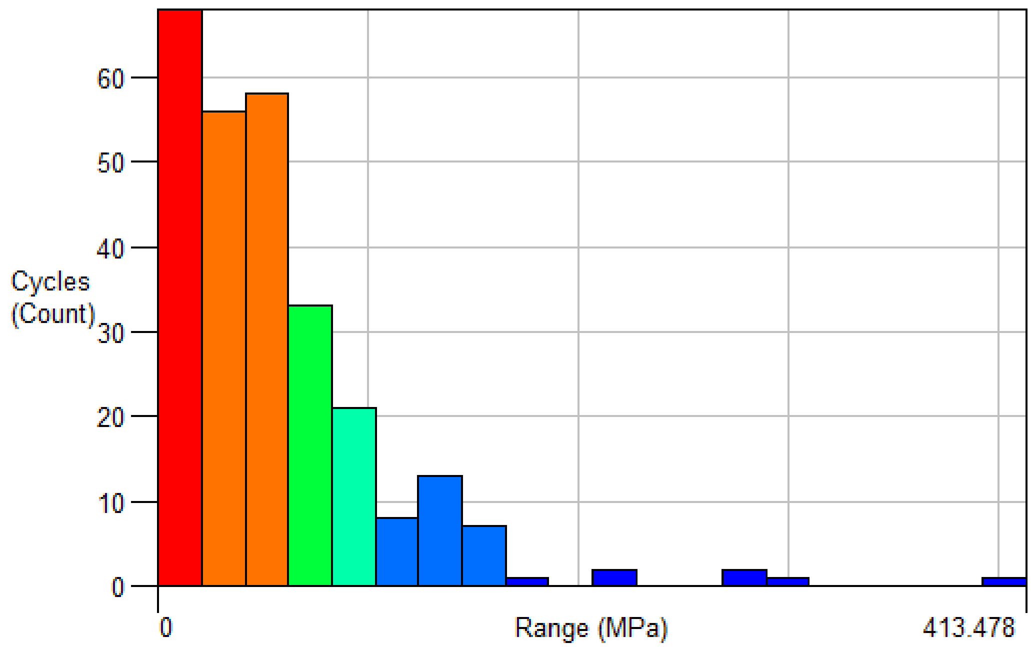
From-To—This is similar to a max-min but takes into account the sequence of the points. One axis contains the binned values for the "From" values–the first in the sequence, with the second axis being the "To" values, the second in the sequence

Fig. 2-11 **3D Histogram: Cycles - From - To**



Range Only—This is a single-axis histogram that uses just the cycle range.

Fig. 2-12 2D Histogram:



The input histogram is checked to understand what format it is—that is, the algorithm is smart enough to understand the difference between Range-Mean and From-To.

The Stress-Life method does not need to take account of cycle sequence, so the process of generating a cycle list from the histogram is straightforward. The glyph goes through each histogram bin and uses the centre point of the bin to calculate the cycle maximum and minimum.

Only bins with a non-zero count value are used to populate the cycle lists and the number of counts is also carried within the list as this will be used to multiply the damage calculated for a single cycle.

At this stage we have a "cycles list" with all the information needed to calculate the damage for each cycle and then the total damage. This is the same calculation whether the loading data is from time history, histogram, multi-column or schedule files, and therefore this is described below in the section "Calculating Damage".

2.1.3 Multi-column Input

The multiple column input to the StressLife glyph effectively defines the cycles list directly, at least in terms of its input stress components. Each row defines a cycle, or a set of exactly equivalent cycles (defined by a RepeatCount column). The cycles can be defined in different ways, using the CyclesListInputType property.

CyclesListInputType	MaxMin	Specifies whether the input is Range-Mean, Amplitude-Mean, consecutive points or Max-Min data.

The types available are MaxMin and RangeMean. In each case, a separate column is used to define Max and Min or Range and Mean respectively.

If the cycles columns are not labeled, and/or they are not defined as the first columns in the multi-column data, the glyph allows the user to specify the columns for the data and for the repeat count.

CyclesListInput		
CyclesListInputRepeatCountColumn		The identifier of the column that contains the repeat count for each cycle
CyclesListInputColumns		Specifies the column numbers or keywords that have the cycles data in them

2.1.4 Schedule Files

Schedule files can be used to combine events that are stored in separate data files. Schedule file inputs can improve processing speed by taking advantage of the Palmgren-Miner Rule, which states that damage increases linearly as cycles are repeated. This means that damage for multiple repeats of an event can be calculated simply by multiplying the damage for a single repeat by the number of repeats. Further, total schedule damage can be found simply by summing each event's subtotal damage. This schedule damage summation can have huge performance benefits over working with very long concatenated time series inputs.

In the StressLife glyph, a schedule file can be specified using two methods. The first is to set the name in a property:

ScheduleFileName		Specifies the name of a schedule file to use as the input for analysis

In this case, the time history input pad is ignored and can be disconnected.

Alternatively, it can be specified by passing the schedule filename in on the time history input pad with the UseMetadataSchedule property set to True. The Schedule name is read from metadata item Schedule.ScheduleName.

Schedules		
UseMetadataSchedule	False	Defines whether to take the schedule file name from metadata

Using the schedule file name (with no data on the input pad) is the most efficient way to process schedules. A large schedule brought in on a pipe will still slow the process down because the software is still transferring data in memory and will be performing statistics calculations.

Schedules of histograms and multi-column, which can be created within the software, are not supported in the glyph. These have to be processed by combining the data together in the input glyph.

A schedule, typically created with the ScheduleCreate module, is defined as a series of events and a repeat count. For example, a schedule could be defined as:

```
<TimeSeriesSchedule Version="2.0" Channels = "1" PadValue="0"
PadMissingChannels="None"><Join Time="1.0000" Type="None"/>
<Taper Time="1.0000" Type="None"/>
<WrapEnds Wrap="0"/>
<Events>
<Event Name="LT1_r01" Active="True" Testname="LT1_r01_1chan"
RepeatCount="154088"/>
<Event Name="LT1_r02" Active="True" Testname="LT1_r02_1chan"
RepeatCount="7722"/>
<Event Name="LT1_r03" Active="True" Testname="LT1_r03_1chan"
RepeatCount="23917"/>
<Event Name="LT1_r04" Active="True" Testname="LT1_r04_1chan"
RepeatCount="2500000"/>
</Events>
</TimeSeriesSchedule>
```

This defines four events. Note the very high number of repeat counts. If this schedule were combined into a single time series before the glyph, the processing time could be very long. Note that the same time history file can be re-used later in the schedule, as the sequence simply takes one event after another. For example, this would be a valid schedule:

```
<TimeSeriesSchedule Version="2.0" Channels = "1" PadValue="0"
PadMissingChannels="None"><Join Time="1.0000" Type="None"/>
<Taper Time="1.0000" Type="None"/>
<WrapEnds Wrap="0"/>
<Events>
<Event Name="LT1_r01" Active="True" Testname="LT1_r01_1chan"
RepeatCount="154088"/>
<Event Name="LT1_r02" Active="True" Testname="LT1_r02_1chan"
RepeatCount="7722"/>
<Event Name="LT1_r03" Active="True" Testname="LT1_r03_1chan"
RepeatCount="23917"/>
<Event Name="LT1_r04" Active="True" Testname="LT1_r04_1chan"
RepeatCount="2500000"/>
<Event Name="LT1_r02a" Active="True" Testname="LT1_r02_1chan"
RepeatCount="15000"/>
<Event Name="LT1_r04a" Active="True" Testname="LT1_r04_1chan"
RepeatCount="10000"/>
</Events>
</TimeSeriesSchedule>
```

The advantage gained in processing the schedule within the glyph rather than combining the time series together on input is primarily in performance. If a large number of repeat counts is defined for the events within the schedule, the TimeSeries input glyph will create very long time histories and a lot of processing will be required that isn't strictly necessary.

Once the glyph has access to the schedule directly, it can offer two types of calculation, "Independent" and "CombinedFast".

EventProcessing	CombinedFast	How to process separate events during schedule processing

Independent mode calculates the damage for each event separately from the others and then sums those damages (including repeat counts) to get a total damage for the whole schedule. This will clearly be quicker than processing the whole schedule if there are significant numbers of repeats. However, cycles that cross events will not be counted. These cycles can produce significant damage and should ideally be included in the calculation.

Setting the EventProcessing property to CombinedFast will take account of cross-event cycles and give results that are very close or identical to using the full combined time history. However, each event is processed only once, so the performance advantage of using the schedule is retained.

2.1.5 Applying Corrections to the Input Loading

A scale factor and offset can be applied to the input loading.

ScaleFactor	1	Linear multiplier on the input data
Offset	0	Offset on the input data

This is done as a $y=mx + c$ linear calibration, where m is the scale factor and c is the offset. In a time series or schedule loading, this is applied to each point in turn.

For histogram loadings, the factor and offset are applied to the histogram limits (XMin, XMax, YMin, YMax). This calculation is slightly different for Range-Mean, Min-Max, From-To, and Range-Only histogram types.

In a multi-column input, the factor and offset are applied to the cycle maximum and minimum values.

A "gate" can be applied to the cycles that are counted that only allows through larger cycles. These are typically more damaging than smaller ones, although mean stress effects can change that. Usually, a relatively small value is set to exclude very small cycles that cause very little damage and where the mean effect is insignificant even at high mean levels.

This is set using the properties GateUnits and Gate.

GateUnits	Data	Sets whether the Gate property is in data units or calculated as a percentage
Gate	0	Cycles whose range is less than the gate are discarded

The gate unit options are “Data”, where the gate is specified as a value in the input data units, and “Percent”, where the glyph calculates the gate to use as a percentage of the total range of the input data.

For example, if the input data is in MPa, the GateUnits are set to Data and the Gate is set to 100, then any cycles whose range (max-min) is less than or equal to 100 MPa will have a damage value of zero assigned to them.

If the input data goes from -200 to 300 MPa, the GateUnits property is set to Percent and the Gate property is set to 10, cycles with a range less than or equal to 50 MPa will be assigned a damage of 0.

The gate applies to all the loading types: time history, histogram, multi-column and schedule file. In the case of a schedule file, the percentage gate option is applied differently depending whether the schedule mode is Independent or CombinedFast. In CombinedFast mode, a single percentage value is calculated across the whole range of the signal input, even if the maximum and minimum are in different events. In Independent mode, the percentage gate value is calculated and applied on a “per event” basis.

Because the scale factor is applied after the gate is checked, cycles are gated based on their original size, not on the scaled size.

2.2 Materials Data

The basic requirement for a stress-life analysis is access to materials data in the form of a stress-life (SN) curve, which may be parameterized or in the form of X-Y pairs (typically a family of curves based on mean stress, R-ratio or temperature). In addition, the Ultimate Tensile Strength (UTS) is used to determine static failure conditions due to the calculated stresses being too large. See “Interpreting the Results Generated by the StressLife Glyph” for more details on how the glyph handles the case where the UTS is exceeded.

The data can come from a variety of sources. These are set by the properties MaterialDataSource and MaterialDatabaseName.

⊟ **Material**		
MaterialDataSource	Properties	Specifies the source of the materials data
MaterialDatabaseName		The name of the materials database to use
MXDMaterialType	SN	Specifies type of material to use from an MXD database
MaterialName		The name of the material to use from the specified database

Data source options are as follows:

Properties—The basic parametric materials data can be entered directly into the glyph property sheet. This does not support multiple curves.

MaterialData		
MaterialStressUnits	MPa	Selects the units of the stress type material properties
Material_UTS		Ultimate Tensile Strength
Material_E		Youngs Modulus
Material_SRI1		Stress Range Intercept
Material_b1		Main S-N slope
Material_NC1	1E25	Transition point
Material_b2	0	Second S-N slope
Material_StandardError	0	Standard error of log(life)
Material_RRatio	-1	R-Ratio

MDM database—Basic properties can be extracted from the MDM database format. This is an obsolete format used in the legacy products nSoft and FE-Fatigue. A valid material name corresponding to a database entry must be specified in the MaterialName property. Multiple curve data types are not supported in this file format.

MXD database—This is the database format supported in GlyphWorks and DesignLife; data can be entered into the database using the nCode MaterialsManager. A valid material name corresponding to a database entry must be specified in the MaterialName property.

For both database types, GlyphWorks also provides a selection dialog that allows the database to be specified, and the material name to be selected from a list of available materials of the appropriate type.

Input Pipe—The lower of the two multi-column input pipes allows multi-curve data to be passed in to the glyph from an external source. Standard parametric data sets are not supported using this method.

2.2.1 Materials Data Types

Standard Parametric Data

The standard stress-life material properties, which can be found in the MXD material database, consist of a set of generic properties (which can be inherited from a generic parent) and a set of specific properties that define the SN curve.

The generic properties are as follows:

Parameter Name	Description
MaterialType	Numeric code defining the type of material. The material type is required for correct application of surface finish corrections. For a complete list of codes, see "Material Type Codes" on page 193.
UTS	Ultimate tensile strength. This is required for the correct definition of the upper part of the S-N curve, and to apply the static failure criterion. It is also required for surface corrections.
YS	Yield stress. Optional (not currently used).
E	Modulus of elasticity. Required for S-N when the input is elastic strain.
Me	Elastic Poisson ratio (defaults to 0.3 if undefined)
Mp	Plastic Poisson ratio (defaults to 0.5 if undefined)
Comments	
References	

The specific S-N properties are as follows. M1 to M4 are required only if the FKM mean stress correction method is being used, and even then they can often be estimated based on the material type:

Parameter Name	Description
SRI1	Stress range intercept
b1	First fatigue strength exponent
Nc1	Transition life
b2	Second fatigue strength exponent
SE	Standard error of log10(N). This is used to calculate the life adjusted to a certain probability of failure/survival.
RR	R-ratio (ratio of minimum to maximum load) of the tests used to define the S-N curve
Nfc	Numerical fatigue cutoff life. Beyond this life, damage will be assumed to be zero.
M1	Mean stress sensitivity when $R > 1$ for FKM mean stress correction method
M2	Mean stress sensitivity when $-\infty < R < 0$ for FKM mean stress correction method
M3	Mean stress sensitivity when $0 \leq R < 0.5$ for FKM mean stress correction method

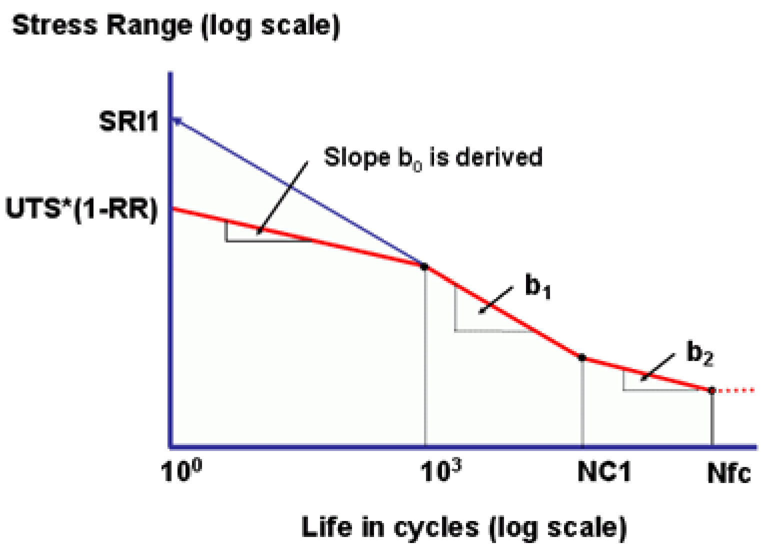
Parameter Name	Description
M4	Mean stress sensitivity when 0.5 ≤ R < 1 for FKM mean stress correction method

The standard S-N curve consists of 3 linear segments on a log-log plot. The central section has the formula

$$\Delta\sigma = SRI1\left(N_f\right)^{b1}$$

where Nf is the number of cycles to failure.

Fig. 2-13 Stress Range v Life in cycles (log scales)



NC1 defines the point on the curve where it transitions to the second slope b2. If b2 is set to zero, this acts as a fatigue limit. At lives of less than 1000 cycles, the slope of the curve may be modified to take into account the limitation in fatigue strength imposed by the static strength of the material, so that, at a life of 1 cycle, the maximum stress is equal to the UTS.

The fatigue cutoff is a numerical limit, normally set at around 1E30 cycles. It has no physical interpretation.

RR is the R-ratio or load ratio (σmin/σmax) of the tests used to determine the S-N curve. It is important when a mean stress correction is to be applied.

SE, the standard error of log10(N), is used to adjust the life/damage predicted to any given probability of survival. Fatigue life always includes some scatter, and at any given level of stress range, the distribution of fatigue lives is assumed to be a log-normal distribution, that is, a Normal or Gauss distribution of the logarithm of the fatigue life.

The Gauss distribution defines the probability density function of a value x as:

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

and the cumulative probability of x as:

$$\Phi(x) = \frac{1}{2}\left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$$

When applying this to fatigue calculations:

x is replaced by log10(N) where N is the fatigue life in cycles.

μ is replaced by log10(N50) where N50 is the number of cycles at which 50% of tested specimens are predicted to fail.

σ is the standard error of log10(N) associated with the S-N curve (obtained from linear regression of the original test data —SE in the material database).

In practice, if we want to make a life or damage prediction based on a particular percentage probability of survival, we use a lookup table to determine the deviation from the mean (50%) life in terms of the number of standard errors.

In the glyph, this is selected using the CertaintyOfSurvival property:

CertaintyOfSurvival	50	Specifies the certainty of survival based on material data scatter

Note that the value set for this property will have no effect if the standard error value in the materials data is set to zero.

The certainty of survival is converted to a standard deviation using the following values. Linear interpolation is used for values not in the table.

Table 2-1 List of Probabilities vs. Uncertainty

%Certainty of Survival	Standard Deviations from mean
99.9	-3
99.4	-2.5
97.7	-2
93	-1.5
84	-1
69	-0.5
50	0
31	0.5

Table 2-1 List of Probabilities vs. Uncertainty (Continued)

%Certainty of Survival	Standard Deviations from mean
16	1
7	1.5
2.3	2
0.6	2.5
0.1	3

For example, if the standard error is 0.1, creating a calculation with 97.7% certainty of survival corresponds to -2 standard errors. Therefore:

$$\log_{10} N = \log_{10} N_{50} - 0.2$$

$$N = N_{50} \cdot 10^{-0.2}$$

A few companies, notably in the aerospace industry, do not like to rely on the generic Goodman, Gerber, and FKM methods for modeling the effect of mean stress, but prefer to take mean effects into account by interpolation between curves obtained at different mean or load ratios. The next two options are for these companies.

Multi-curve Data—Multiple Mean

When MultiMeanCurve is selected, the S-N analysis engine expects the S-N data in the form of a family of S-N curves representing the fatigue strength of the material at different mean stress levels (nCode SN Mean Stress Curves). This type of data is found only in the MXD database (not supported in the nSoft MDB database). The data consists of a set of generic data with a number of child datasets storing the individual curves. The generic data consists of the following parameters:

Parameter Name	Description
MaterialType	Numeric code defining the type of material. The material type is required for correct application of surface finish corrections. For a complete list of codes, see "Material Type Codes" on page 193.
UTS	Ultimate tensile strength. This is required for the static failure criterion.
E	Modulus of elasticity. Required for S-N when the input is elastic strain.
Nfc	Numerical fatigue cutoff life. Beyond this life, damage will be assumed to be zero. In cycles.

Parameter Name	Description
Ne	Endurance limit. This is a specified life in cycles. The main function of this is to define the point on the S-N curves where surface finish corrections are applied.
SEIs	Standard error of $\log_{10}$(stress)
StressType	Type of stress used to define each cycle—Range, Amplitude, or Maximum
Comments	
References	

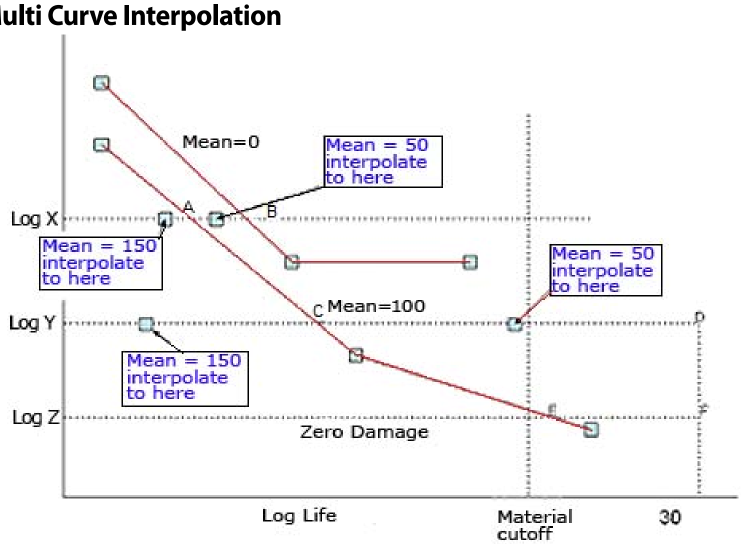
Each child dataset (one for each mean stress value) defines a lookup curve of stress vs. life and has the following parameters:

Parameter Name	Description
MeanStress	Value of mean stress
StressValues	Comma-separated list of stress values
LifeValues	Comma-separated list of life values

MultiCurveInterpolation	BetweenPoints	Specifies the way in which the interpolation is performed

The normal way to use such a curve set is to interpolate between the curves to determine the life and corresponding damage for each cycle. This is used when MultiCurveInterpolation is set to “Between Points”. The process is illustrated in the following diagram:

Fig. 2-14 Multi Curve Interpolation



The material S-N behavior is described in this case by two S-N curves representing the material at mean stress levels of 0 and 100.

Consider a cycle with Range X. First we must identify which two curves to use for interpolation by finding the pair with mean values either side of that of our cycle.

If the cycle has a mean of 50, this lies between our two curves. In this case, we look up the log(Life) values on the two curves corresponding to LogX, and locate points A and B. If necessary, we extrapolate the curves beyond their end points. We then linearly interpolate between these two log(Life) values based on the mean stress of our cycle to determine the log(Life) of our cycle.

If the cycle has a mean less than or equal to the minimum mean of all the curves (in this case 0), the life is determined using the curve with the minimum mean value. The life corresponds to Point B.

If the cycle has a mean greater than the maximum mean in the curve set, in this case 100, either the two curves with the highest mean values can be used to determine the log(life) by extrapolation, or the curve with the highest mean may be used, i.e., in this case the life corresponds to point A. This is controlled by the InterpolationLimit property.

InterpolationLimit	UseMaxCurve ▾	Multicurve material interpolation limit

Consider a cycle with range Y. This corresponds to point C on the Mean = 100 curve, but it does not intersect the mean = 0 curve, or that intersection point is beyond the Fatigue Cutoff Life. In this case, the second point D used for interpolation or extrapolation is set to 30 (corresponding to a life of 1E30).

Consider now a cycle with Range Z. This cycle does not intersect any of the S-N curves at a value less than the material cutoff, so the damage value will be set to zero. Any cycle for which the resulting life is beyond the material cutoff will have its damage set to zero.

The variability in material strength is taken into account by the standard error of log10Stress (SEls). For example, if SEls is 0.1 and a calculation is to be made with 97.7% certainty of survival, the fatigue strength (all the stress values in each curve) would be reduced by 2 standard errors:

log(Fatigue Strength) = log(Mean Fatigue Strength) – 2*0.1

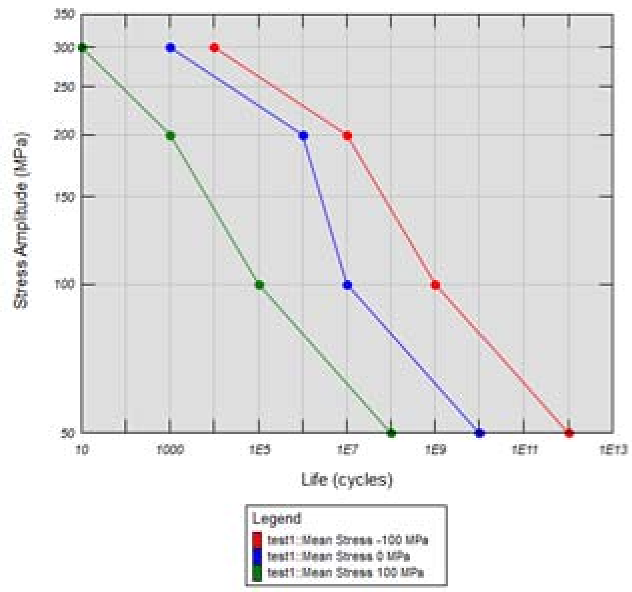
Fatigue Strength = (Mean Fatigue Strength)x$10^{-0.2}$

MultiCurveInterpolation	WholeCurve ▾	Specifies the way in which the interpolation is performed

If MultiCurveInterpolation is set to “WholeCurve” the procedure is different.

When the interpolation method is set to “WholeCurve”, each curve must have the same number of X-Y points. Figure 2-15 below shows an example of a set of curves that would be valid for this method:

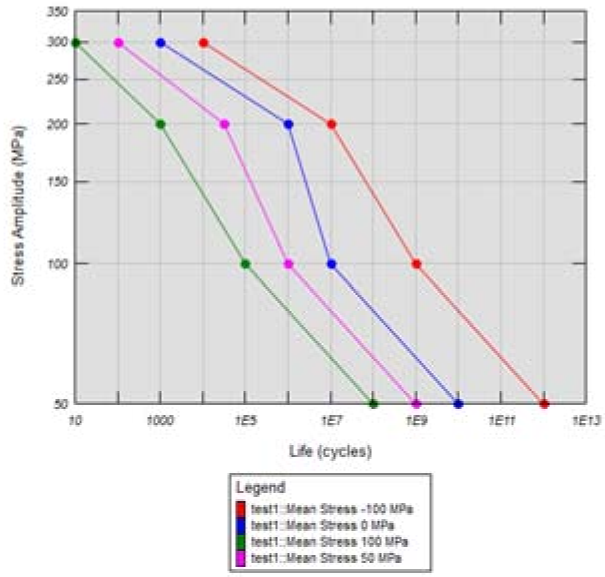
Fig. 2-15 Multi-mean curves with the same number of points



In this case, the interpolation is done point-by-point on the whole curve, creating a new curve at the required mean stress that can then be used for the stress-life lookup.

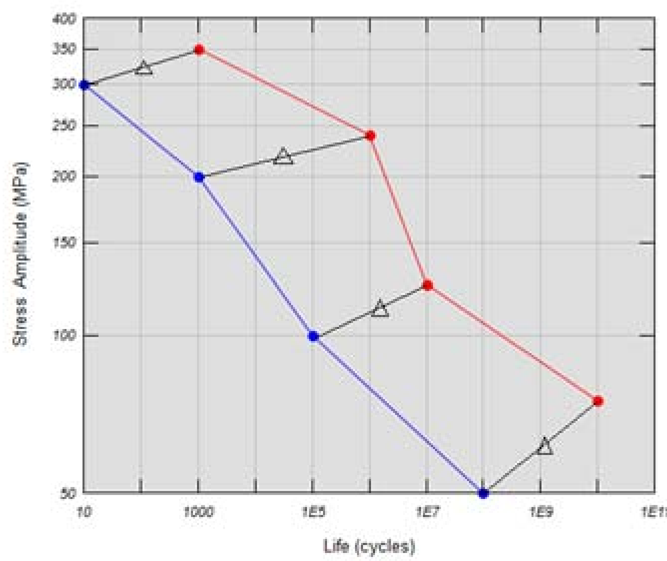
In the example above, for a mean stress of 50 we have to interpolate (log-log) halfway between the curves for 0 and 100. This creates the temporary new purple curve as shown below (second from the left):

Fig. 2-16 Stresses at same level



This is a simple case because the stresses are at the same level. If the stresses are not at the same level, the point interpolation is between the points, as shown in the following example:

Fig. 2-17 Stresses not at same level



For a mean stress halfway between the curves, the interpolated points for the new curve lie approximately where the triangles are—50% of the way between the two points on a straight line in log-log space.

Treatment of Large Stress Cycles

Where the low-cycle end of the material curve is exactly a flat line (e.g. two life values for the same stress) and a cycle stress range exceeds this flat line stress value, it is not possible to extrapolate a valid damage. If a cycle stress range exceeds this flat line section, the damage value for this cycle will correspond to the lower life value.

For example, for a flat line section:

Life = 0.01 cycles, Stress = 500 MPa

Life = 10 cycles, Stress = 500 MPa

If the stress cycle is 600 MPa, the corresponding life will be the lower life value of 0.01 cycles, and the returned damage will be 100 (1/0.01). In this way, the user can control the returned damage value by setting an appropriate minimum life value.

If this large stress range exceeds the UTS static failure check, this is correctly flagged as a static failure.

Multi-curve Data—Multiple R-Ratio

The handling of MultiRRatioCurve data is very similar to that of MultiMeanCurve data. See Multi-curve Data—Multiple Mean for more information.

When MultiRRatioCurve data is selected, the S-N analysis engine expects the S-N data in the form of a family of S-N curves representing the fatigue strength of the material as tested at different R-ratios. This type of data is found only in the MXD database (not supported in the nSoft MDB database) under the category nCode SN R-Ratio Curves. The data consists of a set of generic data and then a number of children representing the individual S-N curves.

The generic data consists of the following parameters:

Parameter Name	Description
MaterialType	Numeric code defining the type of material. The material type is required for correct application of surface finish corrections. For a complete list of codes, see “Material Type Codes” on page 193.
UTS	Ultimate tensile strength. This is required for the static failure criterion.
E	Modulus of elasticity. Required for S-N when the input is elastic strain.

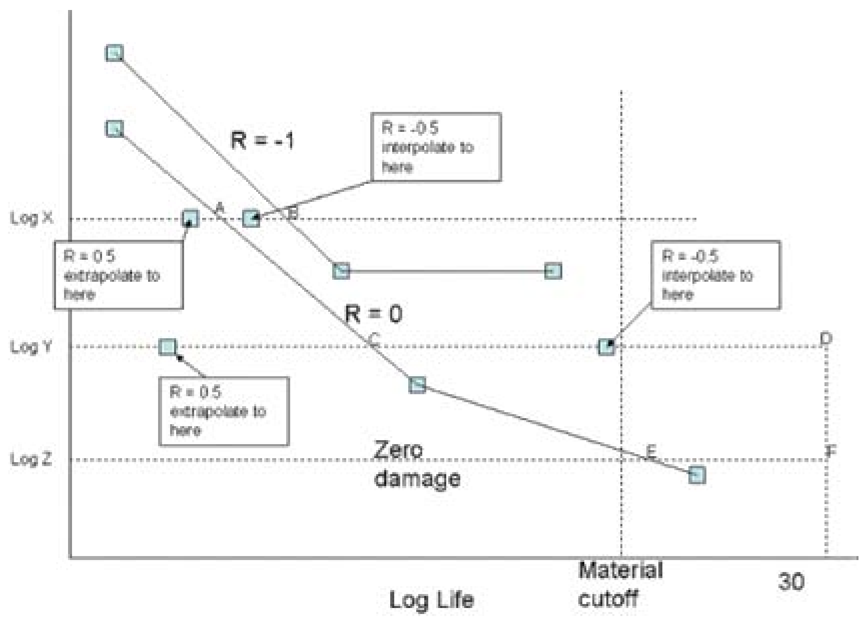
Parameter Name	Description
Nfc	Numerical fatigue cutoff life. Beyond this life, damage will be assumed to be zero. In cycles.
Ne	Endurance limit. This is a specified life in cycles. The main function of this is to define the point on the S-N curves where surface finish corrections are applied.
SEIs	Standard error of $\log_{10}$(stress)
StressType	Type of stress used to define each cycle—Range, Amplitude, or Maximum
Comments	
References	

Each child dataset (S-N curve) defines a lookup curve of stress vs. life and has the following parameters:

Parameter Name	Description
R-Ratio	R-ratio for this stress-life curve
StressValues	Comma-separated list of stress values
LifeValues	Comma-separated list of life values

These curves are normally used by interpolating between the curves for each cycle depending on its R-ratio stress. This is illustrated with the following diagram:

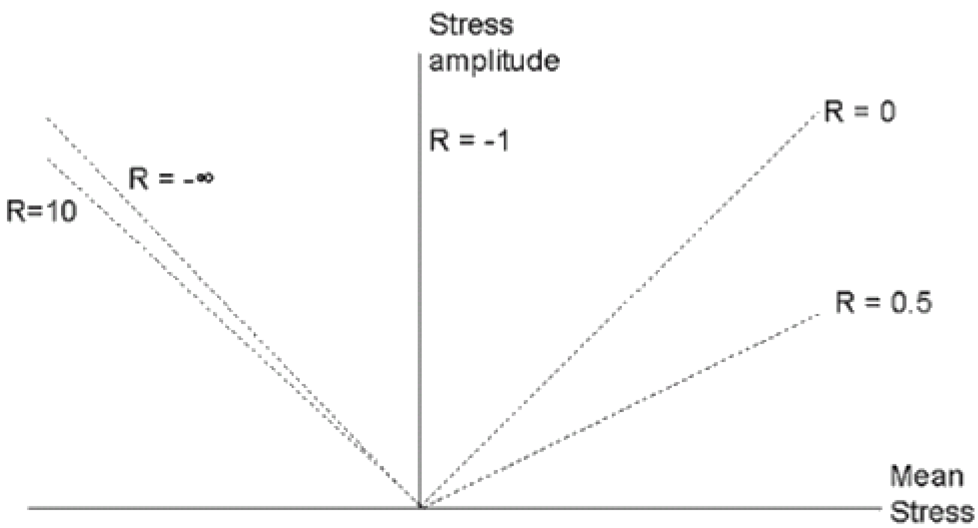
Fig. 2-18 Interpolation scheme for MultiRRatioCurve data



The material S-N behavior is described in this case by two S-N curves representing the material at R-ratios of -1 and 0.

Consider a cycle with range X. First we must identify which two curves to use for interpolation by finding the pair with mean values either side of that of our cycle. We have to be a little careful because the R-ratio has two distinct regimes bounded by the condition where maximum stress = 0, at which point R is undefined.

Fig. 2-19 Lines of constant R-ratio



If the cycle has an R-ratio of -0.5, this lies between our two curves. In this case, we look up the log(Life) values on the two curves corresponding to LogX, and locate points A and B. If necessary, we extrapolate the curves beyond their end points. We then calculate the mean stress corresponding to points A and B and linearly interpolate between these two log(Life) values based on the mean stress of our cycle to determine the log(Life) of our cycle. We don't use the R-ratio for interpolation because of the non-linear and discontinuous behavior illustrated in Figure 2-19. Assuming that there are no curves present with R > 1 (i.e., totally in the compressive regime, which is unlikely):

- If the cycle has an R-ratio less than or equal to the minimum curve R-ratio, or which is greater than 1, the life is determined using the curve with the minimum R-ratio value.
- If the cycle has an R-ratio greater than the maximum R-ratio in the curve set, but less than 1, either the two curves with the highest R-ratio values can be used to determine the log(life) by extrapolation, or the curve with the highest R-ratio value may be used, i.e., in this case, the life would correspond to point A. This is controlled by the InterpolationLimit property.

Consider a cycle with Range Y. This corresponds to point C on the R = 0 curve in Figure 2-19, but it does not intersect the R = -1 curve, or that intersection point is beyond the Fatigue Cutoff Life. In this case, the second point D used for interpolation or extrapolation is set to 30 (corresponding to a life of 1E30).

Consider now a cycle with Range Z. This cycle does not intersect any of the S-N curves at a value less than the material cutoff, so the damage value will be set to zero.

Any cycle for which the resulting life is beyond the material cutoff will have its damage set to zero.

See “Treatment of Large Stress Cycles” on page 37.

The variability in material strength is taken into account by the standard error of log10Stress (SEls). For example, if SEls is 0.1 and a calculation is to be made with 97.7% certainty of survival, the fatigue strength (all the stress values in each curve) is reduced by two standard errors:

log(Fatigue Strength) = log(Mean Fatigue Strength) – 2*0.1

Fatigue Strength = (Mean Fatigue Strength)x$10^{-0.2}$

If MultiCurveInterpolation is set to “WholeCurve”, the procedure is different.

MultiCurveInterpolation	WholeCurve	Specifies the way in which the interpolation is performed

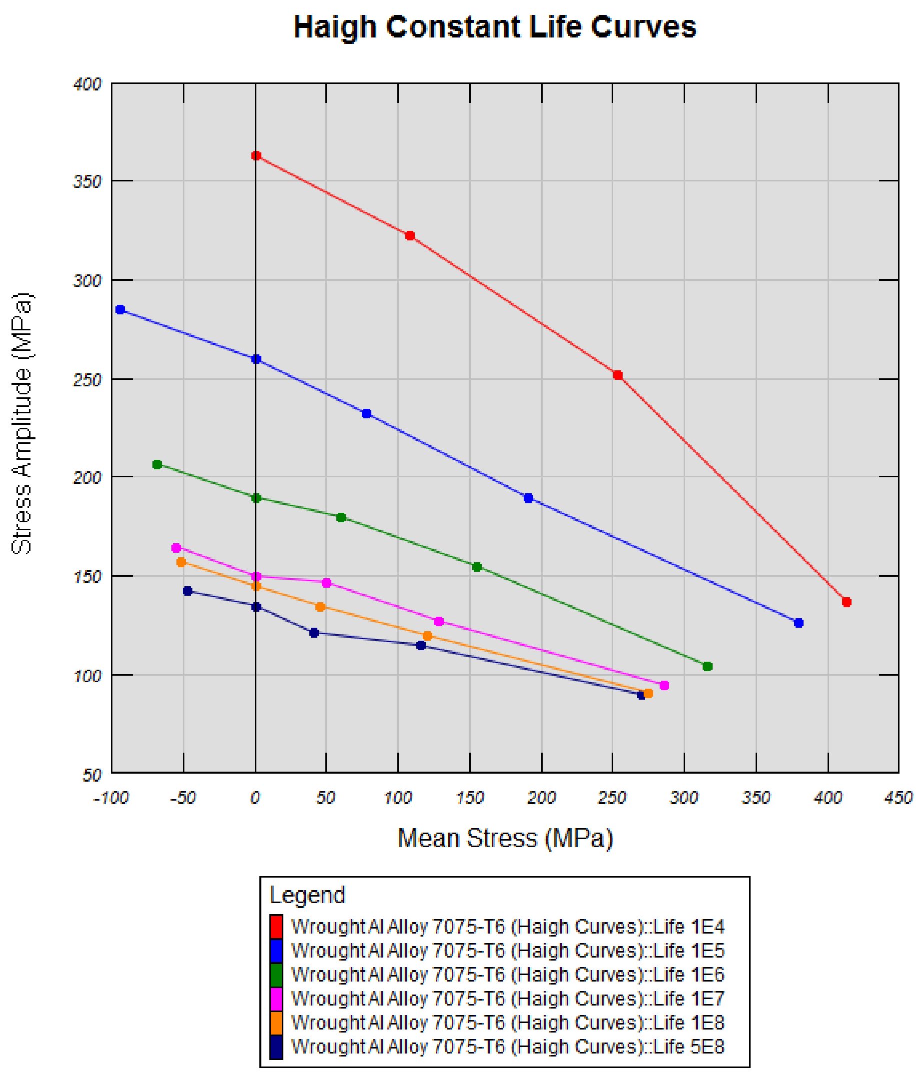
See the section “Multi-curve Data—Multiple Mean” on page 32 for details on the WholeCurve method of interpolation.

Multi-curve Data—Haigh Diagrams

Stress-life data can also be defined in terms of a constant life or Haigh diagram.

The material behavior is described by a number of curves on a plot of stress amplitude vs. mean stress, each of which represents test failures with the same fatigue life.

Fig. 2-20 Haigh constant life curves diagram



Once again, the dataset consists of a set of generic parameters, and a number of children containing the constant life curves.

Parameter Name	Description
UTS	Ultimate tensile strength. This is required to apply the static failure criterion.
E	Modulus of elasticity. Required for S-N when the FE results are elastic strain.
Nfc	Numerical fatigue cutoff life. Beyond this life, damage will be assumed to be zero. In cycles.
Ne	Endurance limit. This is a specified life in cycles. The main function of this is to define the point on the S-N curves where surface finish corrections are applied.
SEIs	Standard error of log10(stress)
StressType	Type of stress used to define each cycle-Range, Amplitude or Maximum
Comments	
References	

Each child constant life curve has the following properties:

Parameter Name	Description
Life	Life in cycles for constant life curve
MeanStressValues	Comma-separated list of mean stress values
StressValues	Corresponding comma-separated list of stress values (Range, Amplitude or Maximum)

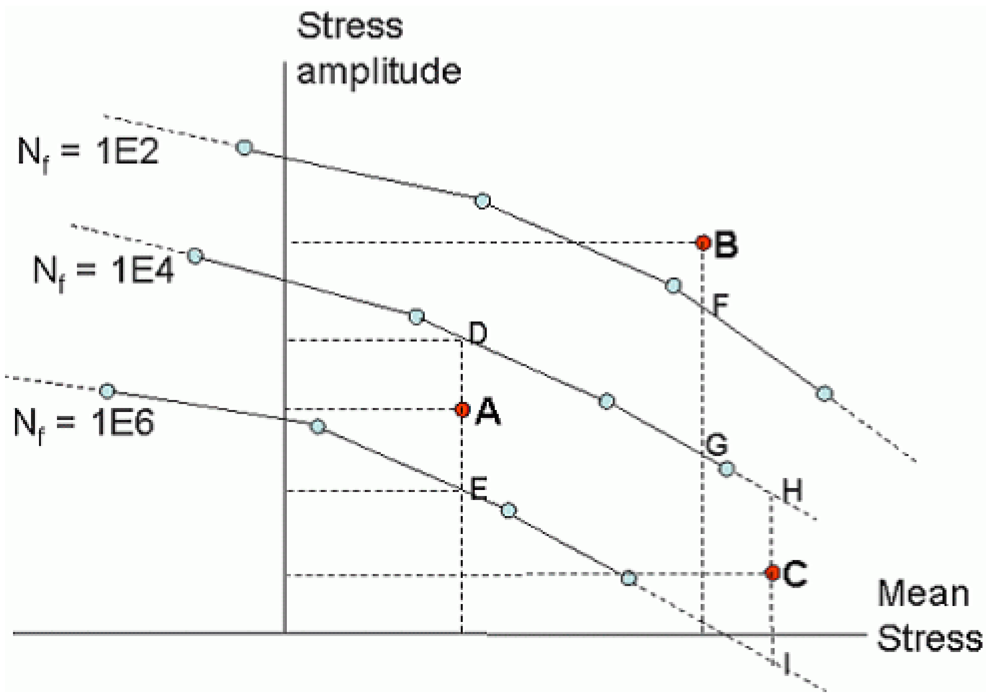
The life for a given cycle is determined by interpolation between the curves of the Haigh diagram. For example, in Figure 2-21 below, consider the cycle represented by point A. This lies between the curves for Nf = 1E4 and Nf = 1E6. Based on the mean stress of this cycle, we look up the stress amplitudes corresponding to points D and E. The life for this cycle is determined by linear interpolation of the log(Life) between these points. For example, if A lies halfway between D and E, the predicted life for that cycle will be 1E5 (damage 1E-5).

The stress amplitude points for interpolation may be determined by extrapolation if necessary, as for point C.

For a cycle that lies above or below all the curves such as that represented by point B, the life and damage is determined by extrapolation (subject to limitations imposed by the other limiting properties).

Results based on cycles that use extrapolation should be treated with some caution.

Fig. 2-21 Fatigue life by interpolation using the Haigh diagram

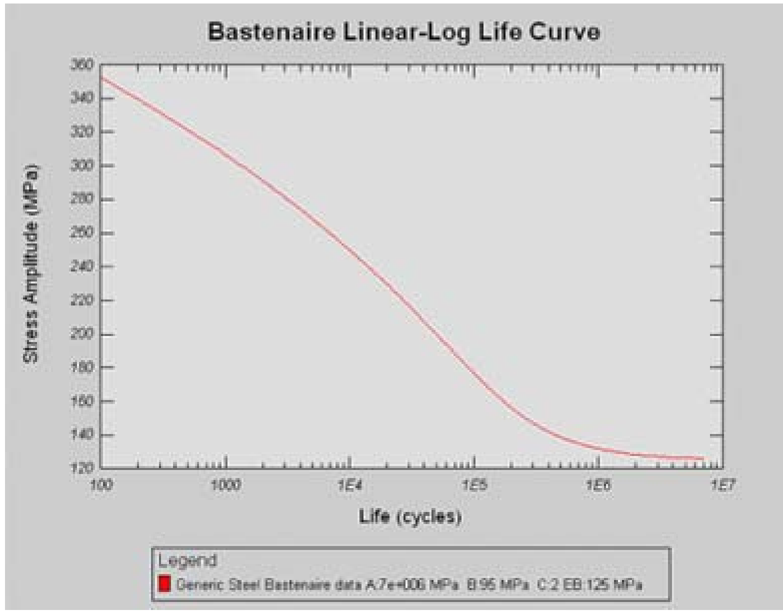


Note that the InterpolationLimit property is not used in the Haigh method.

The stress amplitude values are adjusted to consider the certainty of survival in the same way as for the multi-curve formulation. That is, the log of the fatigue strength is reduced by an appropriate number of standard errors based on the lookup table.

Bastenaire

Fig. 2-22 Bastenaire S-N curve



The material dataset for the Bastenaire method is as follows:

Property Name	Description
MaterialType	Numeric code defining the type of material. The material type is required for correct application of surface finish corrections. For a complete list of codes, see “Material Type Codes” on page 193.
UTS	Ultimate tensile strength. This is required for the static failure criterion.
E	Modulus of elasticity. Required for S-N when the input is elastic strain.
A	Bastenaire coefficient—a parameter positioning the curve along the life axis
B	Scale factor parameter
C	Bastenaire exponent
EB	Bastenaire fatigue limit
RR	R-ratio of test
StressType	Amplitude, Range or Maximum
Sd	Bastenaire scatter factor
Comments	

Property Name	Description
References	

Based on the analysis of thousands of tests carried out on different steels, Bastenaire proposed in 1974 a general formulation of the Stress-Life curve:

$$N = \frac{A}{S - E_B} \exp\left[-\left(\frac{S - E_B}{B} \right)^C \right]$$

The four parameters A, B, C and EB are derived from material raw data. If C > 1, the SN curve has an inflexion point. If C = 0, the model simplifies to the Stromeyer formulation N = A/(S-E) (with a factor of e).

The Bastenaire formula aims to correctly describe the whole endurance domain and the parameter calculation takes into account all tested specimens, including the run-outs.

Bastenaire curves can also be modified to calculate lives at any certainty of survival, using a scatter factor and the normal distribution law (the assumption is made that the stresses are normally distributed for a specified life):

$$N_{P\%} = \frac{A}{(S \pm m * sd - E_B)} \exp\left[-\left(\frac{S \pm m * sd - E_B}{B} \right)^C \right]$$

sd: scatter factor, estimation of the standard deviation on stress

m: number of standard deviations from the mean value, defines the required probability and is given by the normal law tables (see Table 2-1 on page 31).

An example formula for 84% certainty of survival is given below:

$$N_{84\%} = \frac{A}{(S + sd - E_B)} \exp\left[-\left(\frac{S + sd - E_B}{B} \right)^C \right]$$

As usual, linear damage accumulation applies, and damage per cycle is assumed to be 1/N.

Note This data type is available in the database but can only be supported in the software using the "Custom" coding method.

Custom Materials Data

Users can create their own datasets and properties for S-N curves using MaterialsManager and then use the custom analysis option in the StressLife glyph to perform the calculations using the Python programming language. For more details, consult the Glyph Reference Guide.

General		
Mode	Damage	Specifies the calculation type
CalculationMethod	Custom	Whether to use standard S-N, BS7608 or Custom fatigue analysis
CustomSNMethod	CustomSNTest	Specifies the customised SN method to use

Using BS7608:1993 Design Curves

BS7608:1993 is a procedure laid down by the British Standards Institution for the formalization of the fatigue analysis of welded, and in some cases non-welded, components. Weld classes are represented by specific S-N damage curves, which can be used to take into account non-linear damage accumulation.

To use a BS7608:1993 design curve, select it as the CalculationMethod on the property sheet. The following describe the modifications to the BS7608:1993 design curve according to BS7608:1993.

CalculationMethod	BS7608	Whether to use standard S-N, BS7608 or Custom fatigue analysis

Specific S-N damage curves for weld classes are CLASSB, CLASSC, CLASSD, CLASSE, CLASSF, CLASSF2, CLASSS, and CLASSW.

No specific account is taken of mean stress because the S-N curves themselves are considered to have significant mean stress levels as a result of the welding process itself. Each curve has a fatigue transition point, at which the slope changes, at 10 million cycles. BS7608:1993 states that if no cycle has a stress range greater than the stress at 10 million cycles, then no damage is accumulated and the loading is said to be below the fatigue limit.

However, if even 1 cycle has a stress greater than the 10 millionth, then ALL other cycles will cause damage.

Those used for non-weld classes are CLASSB, CLASSC, CLASSD, CLASSE, and CLASSG. Note that there is an overlap and that some curves are used for both welded and non-welded components. For non-welded components, BS7608:1993 reduces the mean compressive stress amplitudes by 60% of their measured values.

To select the curve and whether the joint is welded or not, use the following properties:

BS7608		
BS7608WeldClass	B	Defines the type of weld
BS7608WeldedJoint	Yes	Whether the joint is welded in the BS7608 analysis

The stress-life curves for welded components may be corrected for thickness of weld. If it is 16mm or less, there is no effect. If it is greater than 16mm, a correction factor is calculated as follows:

$T = (16.0 / \text{Thickness})^{0.25}$

The material is modified by multiplying the stress range intercept by this value.

The thickness is set using the BS7608Thickness property. By default there is no thickness correction.

BS7608Thickness		The joint thickness in the BS7608 analysis, in millimetres

Corrosion can also be taken into account. If the joint is corroded, the damage for each cycle is multiplied by 2.0. This can be set using the BS7608Corroded property.

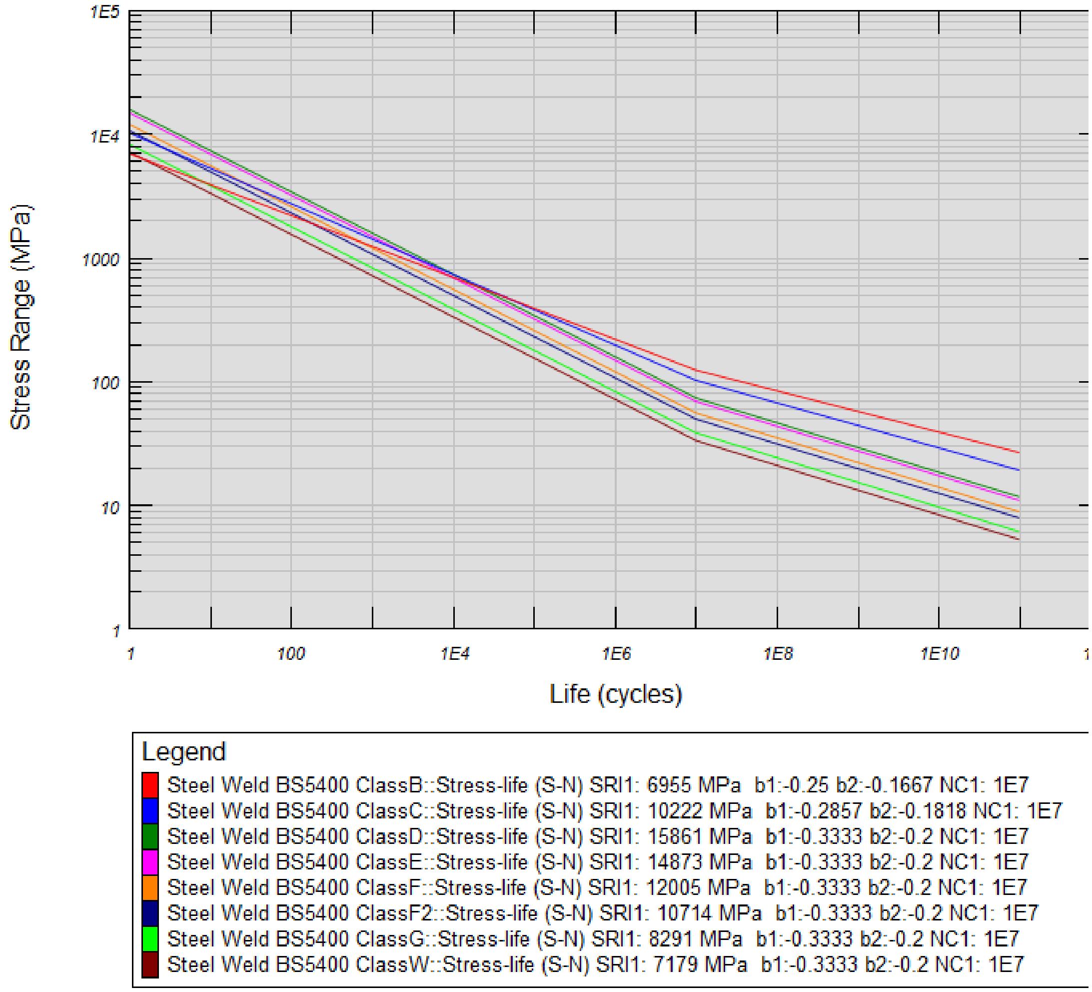
BS7608Corroded	No	Whether the joint is corroded in the BS7608 analysis

Properties that are ignored when the BS7608:1993 method is used include Kf, MeanStressCorrection, SurfaceFinish, SurfaceTreatment, and SmallCycleCorrection.

The materials data for the curves is hard-coded into the algorithm, so no selection from the materials database is required.

The curves are available in the database and can be used in the “Standard” method. However, no special BS7608:1993 modifications are made in this case. These database curves are mean curves. The BS7608:1993 design curves are -2 standard deviations from the mean curves. To convert these to design curves set Certainty of Survival to 97.7, from Table 3-1 on page 102 Certainty of Survival Conversion to Standard Deviation.

Fig. 2-23 BS7608:1993 curves



2.2.2 Defining and Using Surface Finish and Surface Treatment in Stress-Life

Surface finish and treatment can have a significant effect on fatigue behavior. Rough surface finishes, e.g., due to machining marks, will in general reduce the fatigue strength, whereas surface treatments are often applied to increase the fatigue strength.

In GlyphWorks, surface finish and treatment effects are modeled in the S-N engine by means of a single Surface Factor Ksur. This works in a different way from the scale factor and a slightly different way to the fatigue concentration factor value Kf.

GlyphWorks offers pick lists for specific surface finish and treatment types.

SurfaceFinish	Polished	Sets the surface finish type
SurfaceTreatment	None	Sets the surface treatment type

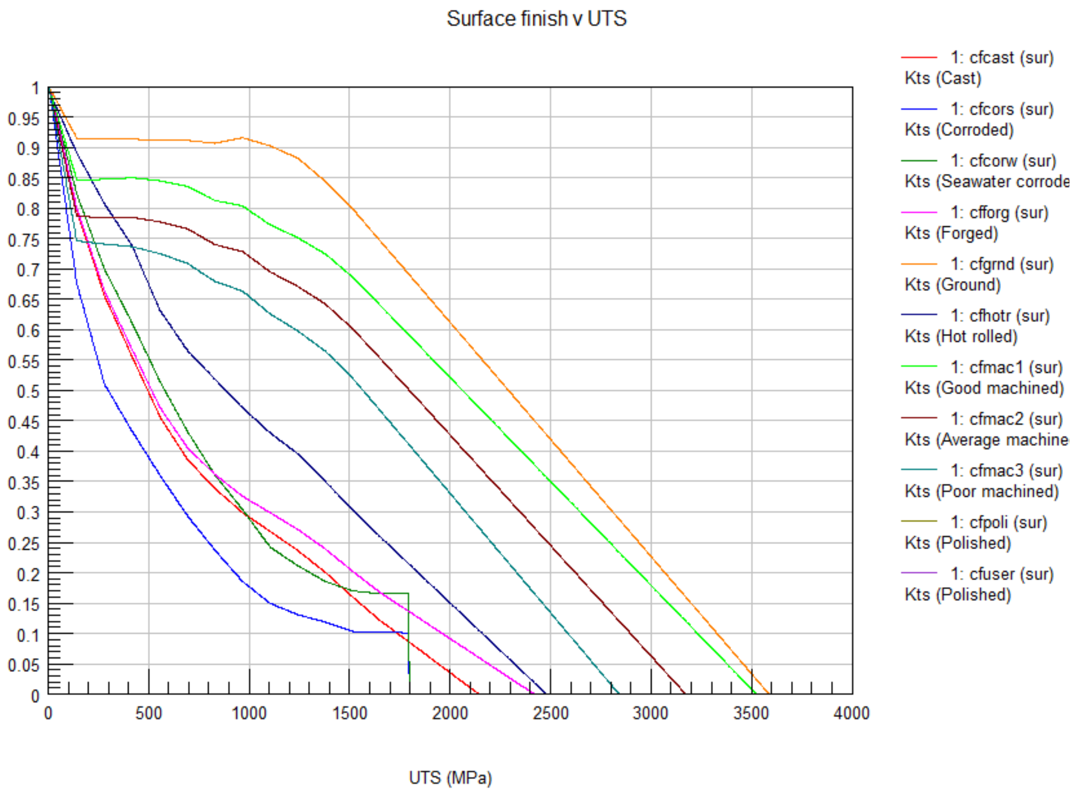
The default values, shown above, will both calculate a factor of 1. Ksur is calculated by multiplying the factors for finish and treatment together, so the net result of setting the default is a Ksur of 1, which leaves the materials data unchanged. Note that this assumes that the materials data is from a material in a polished, untreated condition. If the data comes from a material in any other condition, the values should be left as defaults.

The Surface Factor is used to adjust the material curve. The application is slightly different for the S-N and E-N methods, but the basic principal is the same—the surface factor is applied to the fatigue strength of the material in the high cycle (long-life) regime, but the effect is diminished in the low cycle (short-life) regime.

GlyphWorks offers a set of standard surface finish and treatment values that can be used to estimate the effect of surface changes as a relative calculation. The surface finish values are obtained from a set of files in the materials directory of the GlyphWorks installation that are in the nCode DAC format, and are in the form of Ksur vs. UTS. The value of UTS is obtained for the material and the value of Ksur is then obtained by looking up the curve. One of the options is "User"; if the file `cfuser.sur` is replaced by the user's own data file, then this will be used for the User option.

The plot below shows all the surface finish curves together.

Fig. 2-24 Surface Finish curves



Surface treatments are calculated using the following table, which also requires knowledge of the surface finish. If the surface finish is not one of those listed, the treatment has no effect (+0%).

Finish	Shot Peened (%)	Cold Rolled (%)	Nitrided (%)
Polished	+15	+50	+100
Ground	+20	+0	+100
Machined	+30	+70	+100
Hot Rolled	+40	+0	+100
Cast	+40	+0	+100
Forged	+100	+0	+100

*Example calculation: if the value extracted from the surface finish *.SUR file happens to be 0.8, and the surface treatment is cast and shot peened (40% from the table) then GlyphWorks adds 40% of 0.8 (.32) to 0.8, and the composite correction factor is 1.12.*

These curves and the treatment data are based on limited data and were generated over 50 years ago. They are also valid only for steels. For more accurate information, we strongly recommend that specific material testing be performed using the actual alloy in its raw and polished conditions. This will determine a more accurate value for Ksur that can be used in the software. Alternatively, a separate material data set can be created for the raw and polished conditions.

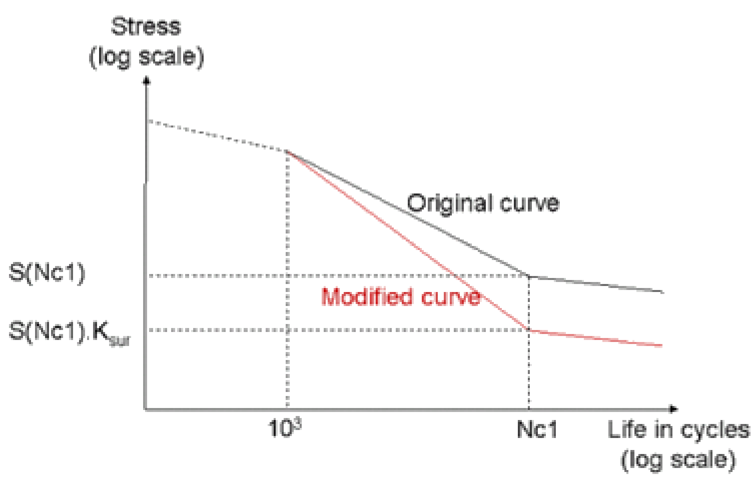
The effect of surface condition in the StressLife glyph is modeled by means of a surface factor Ksur which, in the case of a rough surface finish, reduces the fatigue strength.

The exact mode of application depends on the type of materials data in use.

Standard Materials Data

The effect of the surface condition is modeled by changing the slope of the elastic part of the strain-life curve so that the fatigue strength at the endurance limit Ne is reduced by the surface factor.

Fig. 2-25 Application of surface factors in the S-N method



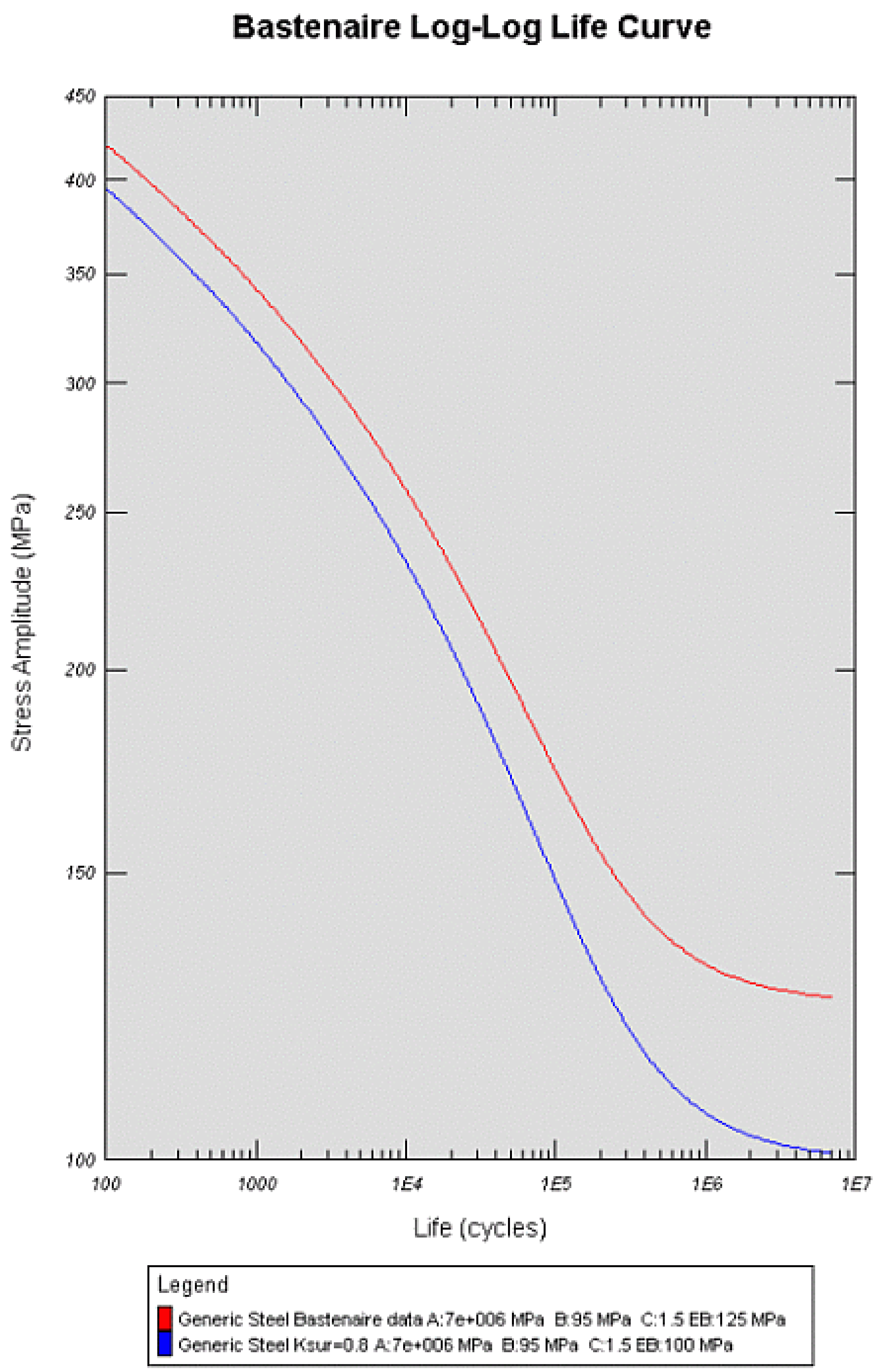
Standard S-N curves are modified by changing the slope and intercept of the central portion of the S-N curve so that the fatigue strength is reduced by a factor Ksur at life ≥ Nc1 (the transition life), but remains the same for life ≤ 1000 cycles. See Figure 2-25.

Note that if Ne is not defined, the fatigue transition point Nc will be used instead.

Bastenaire

Bastenaire data is modified by multiplying the Bastenaire fatigue limit parameter EB by Ksur resulting in shifting the curve as illustrated in Figure 2-26:

Fig. 2-26 Effect of surface finish correction on Bastenaire S-N curves



Multi-curve Options

Surface correction is not supported for Multi-Curve data.

2.2.3 Fatigue Strength Reduction Factor

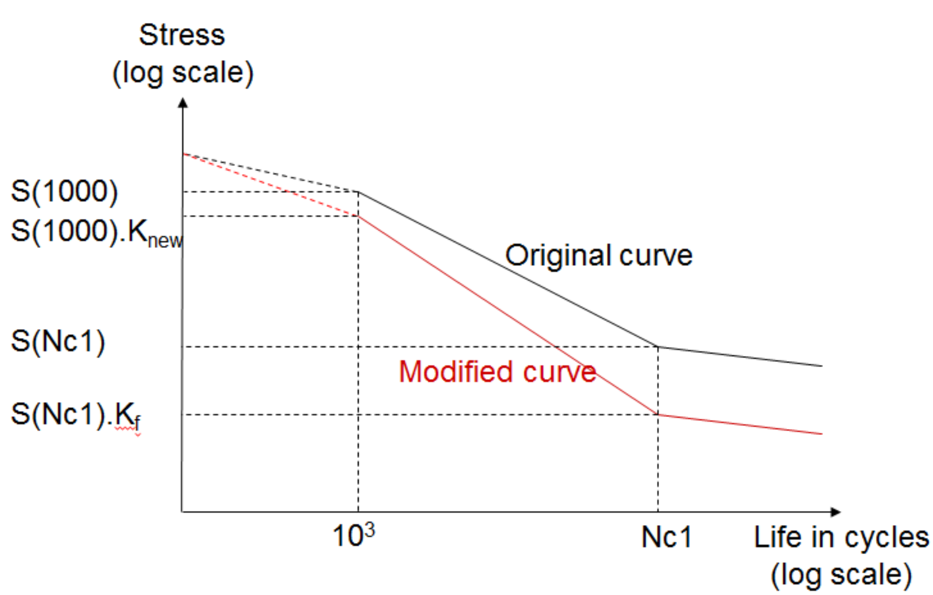
In the StressLife glyph, Kf is applied as a fatigue strength reduction factor, similar to, but not the same as the surface effects.

⊟ Geometry		
Kf	1	Fatigue strength reduction factor

If the fatigue strength reduction factor is not known, then the stress concentration factor, Kt, is a conservative approximation.

With a Kf value, the value of stress at the transition life NC1 is divided by Kf. The value of stress at 1000 cycles, which is unchanged for surface effects, is altered by a value related to Kf and also to a notch sensitivity factor related to fatigue strength and material type. So the slope b1 is different when a Kf is applied as a fatigue strength reduction factor from when a factor derived for surface finish is applied, even if that number is the same. The second slope is unchanged. The following diagram illustrates the difference.

Fig. 2-27 Curve modified by Kf



The factor for ferrous materials is calculated from the lookup table:

UTS	Factor
335	0
445	0.0628
555	0.1258
665	0.1888
775	0.2432
885	0.2957
995	0.3482
1105	0.4
1215	0.4425
1325	0.4845
1435	0.5265
1545	0.5685
1655	0.607
1765	0.6358
1875	0.6644
1985	0.6931
2095	0.7217
2205	0.7503
2315	0.7789
2425	0.8075

Values are interpolated between the values. UTS values that are less than, or equal to, 335MPa result in an unchanged curve. No extrapolation is allowed—if UTS is beyond the end of the table, the last value is used.

For aluminium/titanium alloys, the table is:

UTS	Factor
110	0
146	0.0577

UTS	Factor
182	0.1183
218	0.1789
254	0.2336
290	0.2852
326	0.3367
362	0.3883
398	0.4318
434	0.4731
470	0.5143
506	0.5555
542	0.5967
578	0.6343
614	0.655
650	0.6836
686	0.7123
722	0.7409
758	0.7695
794	0.8

The new Kf at 1000 cycles is calculated from

Kf_{new} = Val*(Kf-1)+1.0

“Val” is the value from the lookup table. The stress at 1000 cycles is divided by Kfnew as shown in the diagram above.

Kf can be applied in addition to a surface correction. The correction factors are both applied as multipliers at Nc1.

Multi-curve Options

Fatigue Strength Reduction Factor (Kf) is not supported for Multi-Curve data.

2.2.4 Standard Materials Data Sets

Materials data from the BS7608:1993 and Eurocode 3 standards are supplied in the central database that comes with GlyphWorks. However, for BS7608:1993 we recommend that these not be used along with a “Standard” calculation, but that the BS7608:1993 method be applied instead, as this not only includes the weld class materials data but also applies other corrections recommended in the standard.

The Eurocode 3 materials data sets are supplied to be used with the “Standard” method.

2.3 Calculating Damage

2.3.1 Damage Summation—Miner's Rule

A stress-based, constant amplitude, fatigue damage curve represents a set of tests at constant stress amplitude together with associated lives. See “Standard Parametric Data” on page 28 for more details.

Operation at a stress amplitude σ1 will result in failure in, say, N1 cycles. Operation at the same stress amplitude for a number of cycles less than N1, Nj say, will result in a smaller fraction of damage, dj, which is often referred to as a partial damage. Operation over a spectrum of different stress levels results in a partial damage contribution di from each stress cycle, σi.

Failure is then predicted when the sum of these partial damage fractions reaches unity so that:

d1 + d2 + ... + di-1 + d i = 1

The Palmgren-Miner rule (or Miner's Rule) asserts that the partial damage at any strain amplitude σ_i, is linearly proportional to the ratio of number of cycles of operation, ni, to the total number of cycles that would produce failure at that strain level, Ni, i.e.:

di = n i / N i

Failure is predicted then if:

n1 / N1 + n2 / N2 + ... + ni-1 / Ni-1 + n i / N i >= 1

The above equations represent a statement of the linear damage rules used by all the GlyphWorks fatigue analyzers. Experience shows that linear damage summation is somewhat of an oversimplification of reality. The most important shortcoming is that no account is made of the sequence in which strain levels are experienced, and damage is assumed to accumulate at the same rate for a given stress level regardless of prehistory. In particular, it appears that large stress amplitudes that precede smaller ones cause the smaller cycles to become more damaging than expected and vice versa. The net result is that in the first instance, the Miners' sum is measured to be less than 1, and in the latter case it becomes greater than 1. Since most service environments involve quasi-random loading sequences, the use of the Palmgren-Miner linear damage rule summing to a constant of 1 is mostly satisfactory.

In Stress-Life analysis, a couple of methods are available to account for the effect of small cycles following a larger, damaging cycle.

SmallCycleCorrection

SmallCycleCorrection	None	For a standard calculation, adjusts materials data in the high cycle regime.

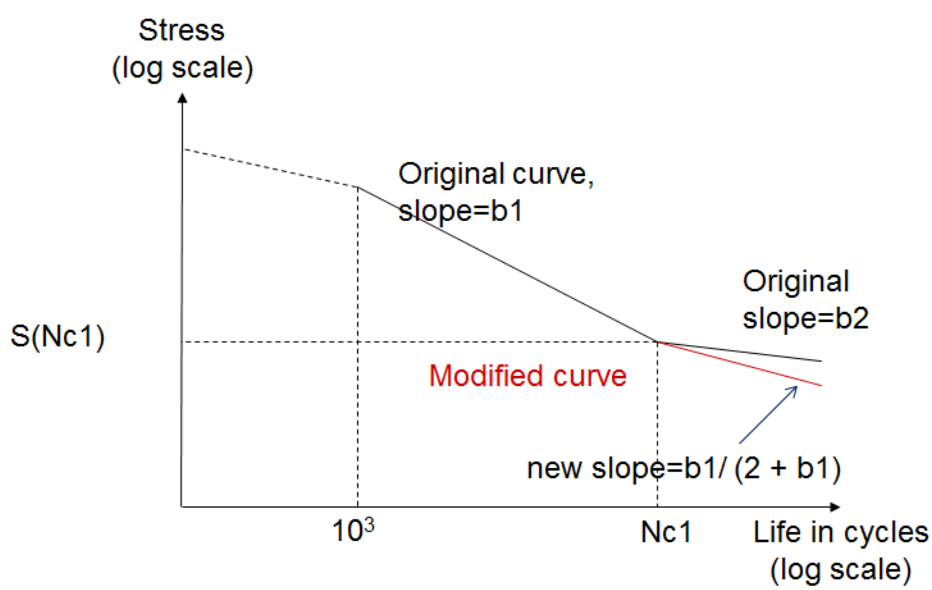
The options are *None*, *Haibach*, *Extrapolate*, and *BS7608:1993*.

If the option is *None*, then no correction is made to the materials data for small cycles.

In the *Haibach* option, the largest cycle in the history is examined. If it is larger than the stress level at the transition life Nc1, then the slope b2 is changed.

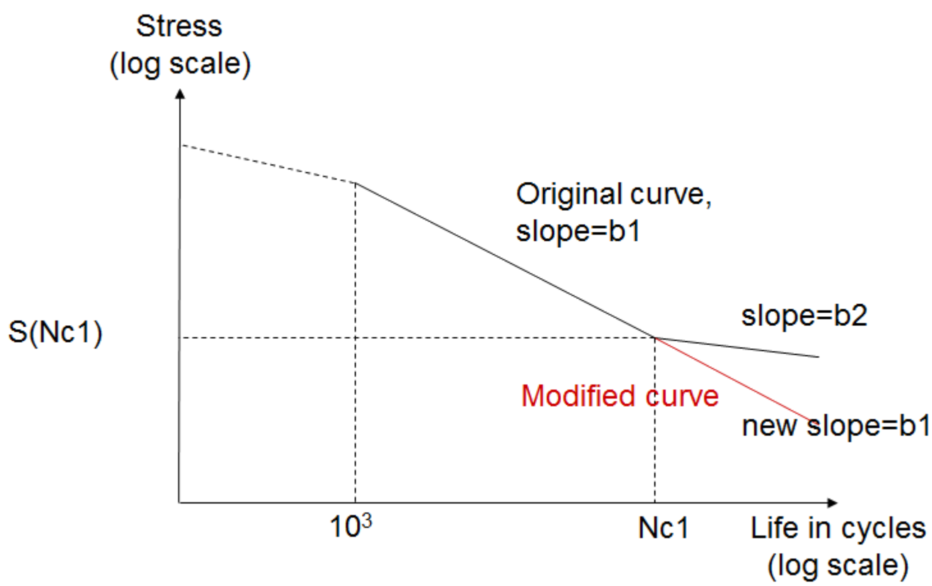
In the *Haibach* correction, there is a slightly less conservative modification to b2. If the largest cycle exceeds S(Nc1), as shown in Figure 2-28, then b2 is modified to b1/(2+b1). If it is smaller than S(Nc1), it is set to 0, and therefore all cycles will have a damage of 0.

Fig. 2-28 Modified slope b2 in "Haibach" small cycle correction



In the *Extrapolate* option, the second slope is set to the same as the slope b1 (the slope between 1E3 and NC1 cycles).

Fig. 2-29 Modified slope b2 in "Extrapolate" small cycle correction



See "Using BS7608:1993 Design Curves" on page 47 for more on BS7608:1993.

Note that BS7608:1993 has its own *method* and this property is ignored if a BS7608:1993 analysis has been requested.

2.3.2 Calculating Damage from the Cycles List

The cycles list generated from any of the input types has to contain the following information for each cycle to be calculated:

- Max stress
- Min stress
- Number of cycles

This is sufficient information to calculate the damage for a cycle of this size.

Without mean stress correction, it is simply a matter of using the appropriate equation, or interpolation, to get the life for a given stress range or amplitude.

The damage for each cycle is

$d = 1/(N_f)$

Any cycle for which the resulting life is beyond the endurance limit will have its damage set to zero.

This damage is simply multiplied by the number of cycles and added to the total damage. When all the damage is calculated and summed, the total life can be calculated (see “Damage Summation—Miner's Rule” above).

2.3.3 Mean Stress Corrections

The main feature of a stress cycle that affects fatigue damage is its range. Fatigue damage is also influenced by the mean stress of each cycle. Mean stress correction methods allow the effect of mean stress to be modeled and taken into account in the life prediction. This section describes the different methods supported.

Mean stress corrections in Stress-Life take two forms. The first group of corrections interpolate directly from a family of curves. These are typically based on curves with different mean stress levels or different R-ratio values. Interpolation mode is set using the MeanStressCorrection property:

MeanStressCorrection	Interpolate	Specifies mean stress correction damage model

This option can be used only on a material with multiple curves.

The second group of corrections changes the value of each cycle to give an “effective” stress range that can be used with curves generated for zero mean stress. These methods can be used on multi-curve datasets, provided there is a curve for mean=0 (or R-ratio=-1). See the following table.

Table 2-2 Allowable S-NMethod/MeanStressCorrection Combinations

	SNMethod				
Mean Stress Correction	**Standard**	**MultiMean Curve**	**MultiRRatio Curve**	**Bastenaire**	**Haigh**
None	✓	✓*	✓*	✓	×
Goodman	✓	✓*	✓*	✓	×
Gerber	✓	✓*	✓*	✓	×
Interpolate	×	✓	✓	×	✓
FKM	✓	✓*	✓*	✓	×
GoodmanTensionOnly	✓	✓*	✓*	✓	×
GerberTensionOnly	✓	✓*	✓*	✓	×

✓ = Allowed
× = Not allowed
✓* = Allowed but a curve for R = -1 or zero mean stress must be present

NoCorrection

Mean stress is not taken into account. This can work with all S-N data types, but if it is to function with MultiMeanCurve or MultiRRatioCurve data, the material dataset must include datasets corresponding to zero mean or R = -1, otherwise an error message is issued. If a standard SN curve is used, no account is taken of the RR value for that curve; it is used as-is.

Goodman

The Goodman mean stress correction calculates an effective stress amplitude based on the mean stress and UTS of each cycle. Again, this can work with all S-N data types, but if it is to function with MultiMeanCurve or MultiRRatioCurve data, the material dataset must include datasets corresponding to zero mean or R = -1, otherwise an error message is issued. In its original form, it is used to calculate an effective stress Se that can be compared to an R = -1 S-N curve, based on the stress amplitude Sa, mean stress Sm and the material UTS:

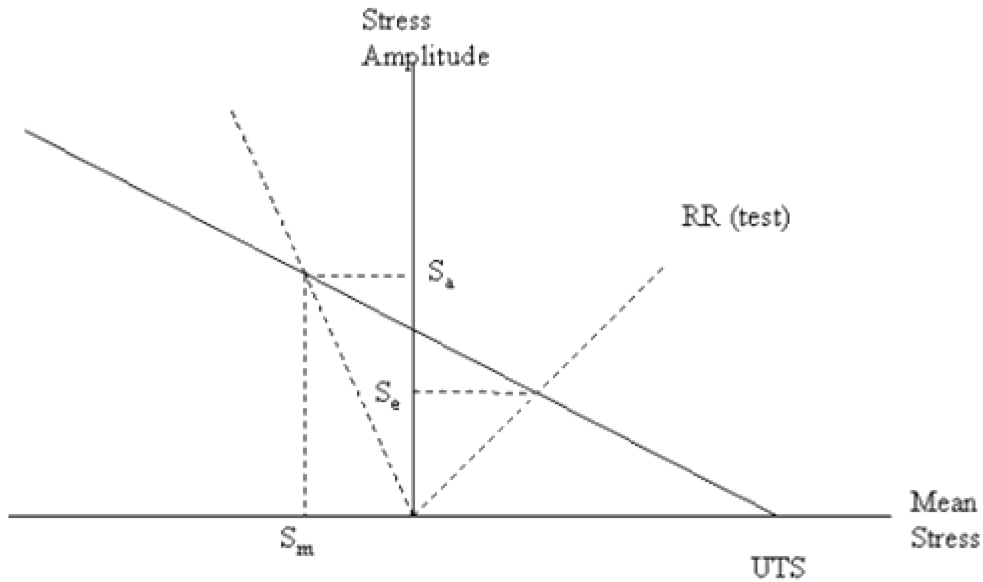
$$\frac{S_a}{S_e(R=-1)} + \frac{S_m}{UTS} = 1$$

In GlyphWorks, this has been extended to allow the equivalent stress to be determined for any R-ratio:

$$S_e(RR) = S_a \frac{UTS}{UTS - S_m + S_a(1+RR)/(1-RR)}$$

This is illustrated below:

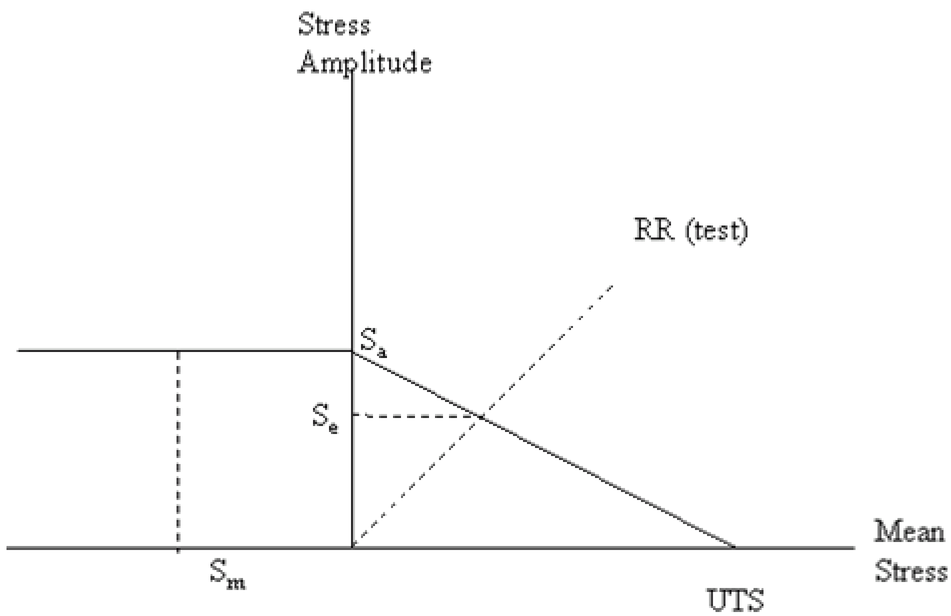
Fig. 2-30 Graphical interpretation of Goodman correction



GoodmanTensionOnly

The Goodman correction in the form described above can be rather non-conservative for cycles with compressive mean stresses. GoodmanTensionOnly addresses this by flattening off the constant life curve when the mean stress is compressive, as illustrated below:

Fig. 2-31 Graphical representation of GoodmanTensionOnly



Gerber

The Gerber correction in its original form is similar to Goodman, except that the second term is squared:

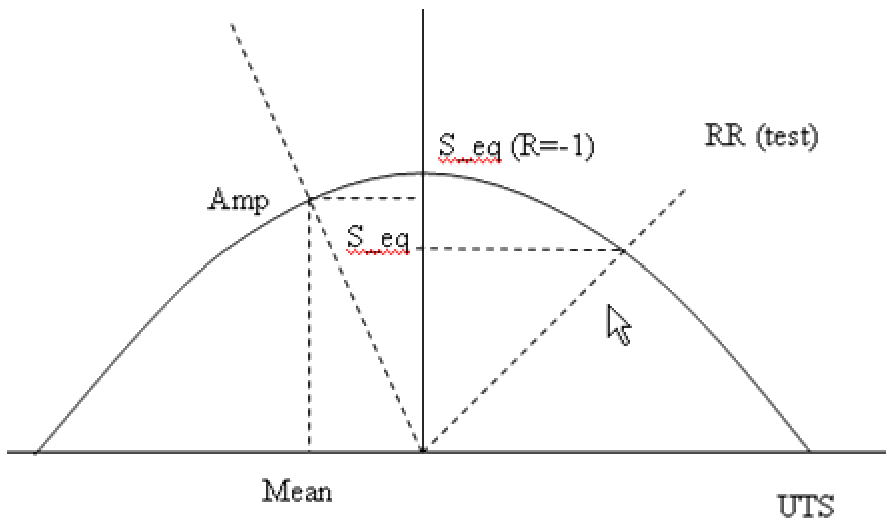
$$\frac{S_a}{S_e(R=-1)}+\left(\frac{S_m}{UTS}\right)^2=1$$

We can then calculate the equivalent stress for any other R-ratio RR:

$$S_e(RR)=\left(\sqrt{\left(1+\frac{4\cdot S_e(R=-1)\cdot(1+RR)^2}{(1-RR)^2\cdot UTS^2}\right)}-1\right)\cdot\frac{(1-RR)^2 UTS^2}{2\cdot S_e(R=-1)\cdot(1+RR)^2}$$

Graphically, this looks like:

Fig. 2-32 Graphical interpretation of the Gerber correction in its original form

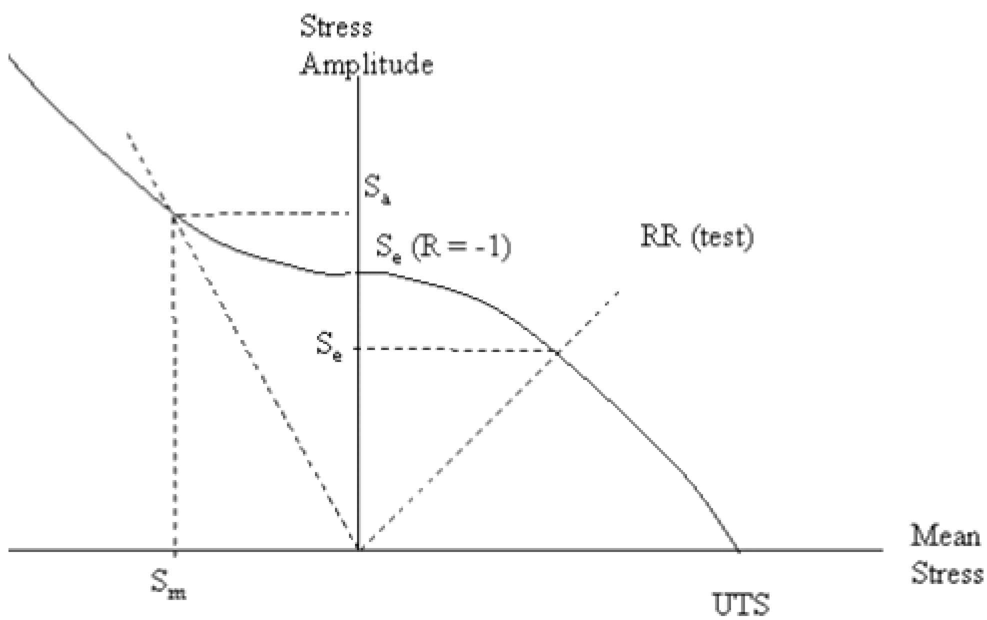


In practice this is not very realistic, because the reduction in fatigue strength under compressive loading is the same as it is under tensile loading. The Gerber correction in its original form will in general be rather non-conservative for tensile mean stresses and rather pessimistic for compressive mean stresses. This is particularly unrealistic, and so the method as implemented in GlyphWorks is modified so that for compressive loadings:

$$\frac{S_a}{S_e(R=-1)} - \left(\frac{S_m}{UTS}\right)^2 = 1$$

Nevertheless, this method (illustrated in Figure 2-33) is not highly recommended. Also (although this is not likely to be an issue in practice), the user can imagine that this method may have problems finding a unique solution for the equivalent stress, or any solution at all, when RR (the R-ratio of the test data) lies deep in the compressive region.

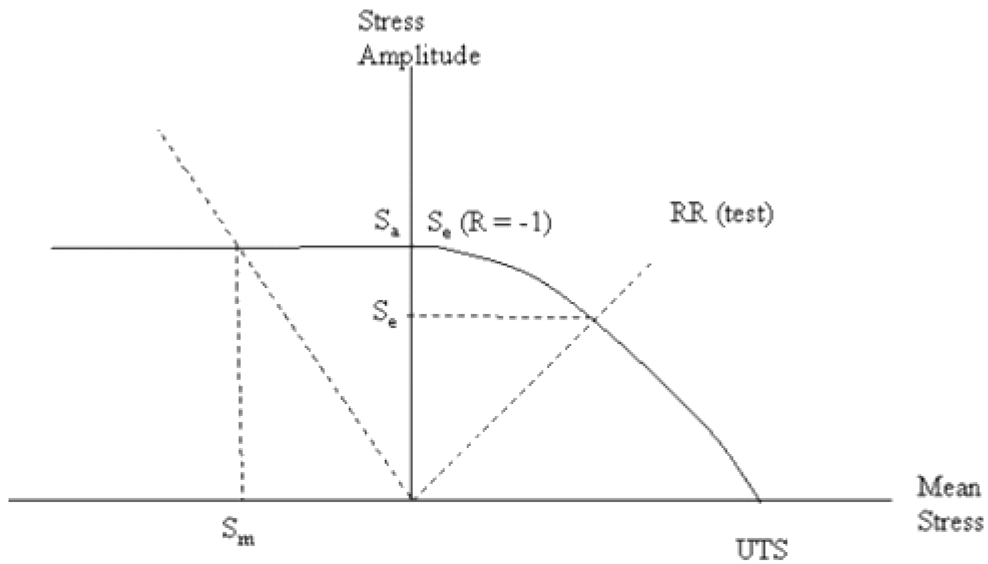
Fig. 2-33 Modified Gerber mean stress correction



GerberTensionOnly

This option provides a better solution by flattening off the constant life diagram in the compressive region in the same way as for GoodmanTensionOnly.

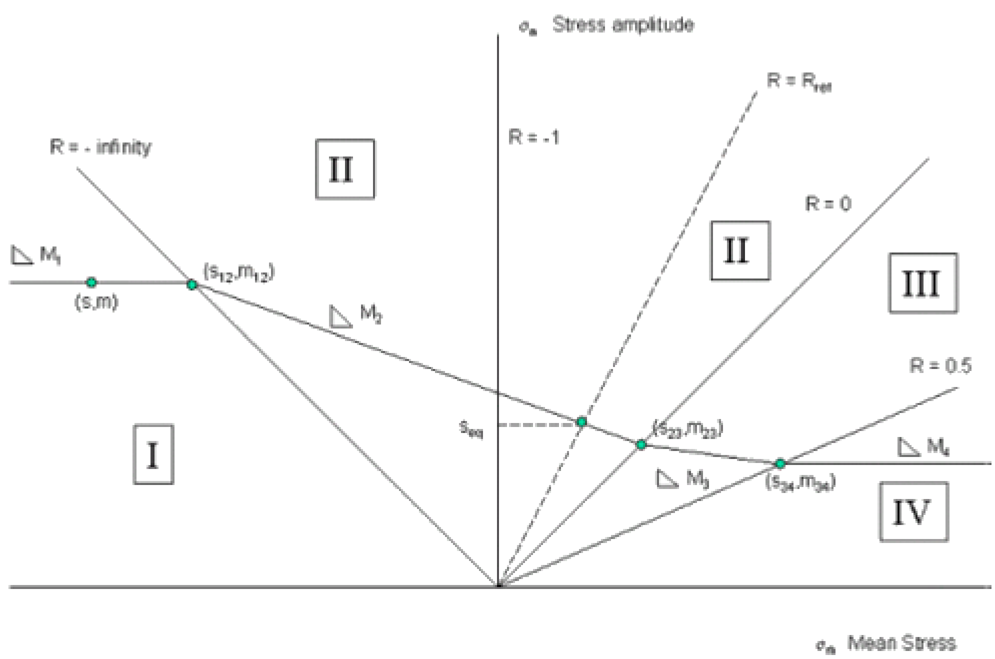
Fig. 2-34 GerberTensionOnly



FKM

The FKM method as implemented here is based on the method described in the FKM Guideline "Analytical Strength Assessment of Components in Mechanical Engineering", Tr. E. Haibach. 2003. In essence it uses 4 factors M1-4 which define the sensitivity to mean stress in 4 regimes: 1. R>1 2. –infinity <= R < 0 3. 0 <= R < 0.5 4. 0.5 <= R < 1 where R is the stress ratio (min/max) of the loading cycle. The method allows us to determine the equivalent stress amplitude Seq at a particular material R-ratio, Rref. The method is illustrated in the form of a constant life or Haigh diagram in Figure 2-35:

Fig. 2-35 Graphical representation of the FKM mean stress correction



The values of M1-4 can be determined from material tests or estimated as follows:

M1 = 0

M2 = -Mσ

M3 = -Mσ /3

M4 = 0

Where the value of Mσ is estimated as follows for the supported material types:

Mσ = a_M*10-3*R_m + b_M where a_M and b_M are constants and R_m is the UTS in MPa.

Values of a_M and b_M for the different supported material classes are as follows:

Table 2-3 FKM Mean Stress Correction Parameters

Material Type	Steel	GS (cast steel)	GGG (nodular cast iron)	GT (malleable cast iron)	GG (cast iron with lamellar graphite)	Wrought Al alloy	Cast Al alloy
nCode Material Type No.	13,14,16-22-25,26-99	9-12,15	5-8	2-4	1	100-105	106
a_M	.35	.35	.35	.35	0	1.0	1.0
b_M	-0.1	0.05	0.08	0.13	0.5	-0.04	0.2

In the software, if M1-4 are undefined, and the material type is one of those listed, all the parameters will be estimated using these rules. If only M2 is defined, then M1 and M4 will be set to zero and M3 to M2/3.

Zero Compressive Damage

Although not strictly a mean stress correction method, another property that is affected by the position of the cycle is ZeroCompressiveDamage.

If this option is switched to On, any cycle that has a maximum stress less than or equal to zero will be assigned a damage of zero, regardless of the mean stress correction being applied.

For more information on mean stress corrections in S-N analysis, see “Metal Fatigue Analysis Handbook”[3], page 151.

2.3.4 User-defined Mean Stress Corrections or Materials Data

If the option for mean stress correction that the user requires is not available in GlyphWorks, it can be added by using a custom Python script. For details, see the Glyph Reference Guide.

2.3.5 “Back” Calculations

When the glyph’s calculation mode is set to ScaleFactor or Kf, the software uses an iterative calculation to determine the value of the scale factor or Kf (fatigue

3. Metal Fatigue Analysis Handbook: practical problem-solving techniques for computer aided engineering / Yung-Li Lee, Mark E. Barkey, Hong-Tae Kang ISBN 978-0-12-385204-5.

concentration factor) that is required to achieve a specified target life. The two properties that are exposed, and have to be set, to achieve this calculation are:

⊟ **BackCalculation**		
TargetLife	1	Target life for back calculations
BackCalculationAccuracy	1	Specifies the percentage tolerance for back calculations

The target life is specified in equivalent units, which are set as part of the loading information. This normally defaults to "Repeats"; i.e., repeats of the input loading history/histogram/multi-column data, and is used to specify the life result.

See the section "Interpreting the Results Generated by the StressLife Glyph" below for more details on how life is calculated and reported.

The back calculation accuracy, *a*, is specified as a percentage value of the target life. The algorithm iterates using an interval halving technique until the calculated life lies within *a*% of the target life. Therefore, the smaller the value of *a*, the more accurate the estimate of Kf or scale factor will be.

As with any iterative algorithm, there is a potential for it to not converge on an answer. However, this tends to happen only if the materials data is very badly formed. The main error that can arise from a back calculation is caused by setting lives that are too short or too long for the damage calculation to work—i.e., when static failure is an issue or where the lives calculated are beyond the cut-off value.

Note that when using schedule files in "Independent" mode, only one scale factor or Kf is calculated for the whole data, not one per event.

When mode=Kf, the property "Kf" is not used, but its existing value is retained. Similarly for the ScaleFactor property when Mode=ScaleFactor.

2.4 Interpreting the Results Generated by the StressLife Glyph

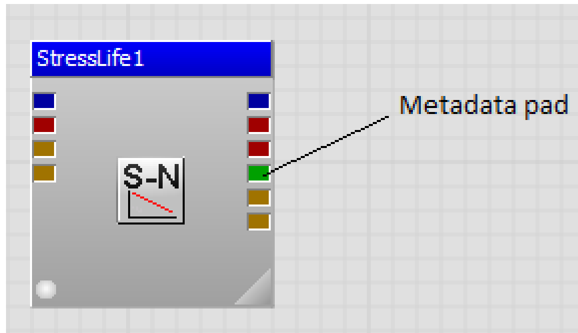
2.4.1 General Comments

Fatigue is a statistical process and calculating damage, and life gives us an estimate of likely failure, not an absolute prediction. In any fatigue calculation, we are making assumptions about loading, material behavior, and geometry and in each case we are using sample data that does not necessarily correspond exactly to a particular component specimen in service.

Therefore, it is important to see fatigue analysis software as a tool that helps make engineering decisions. The more reliable the data that goes in, the more confidence an engineer will have in the estimate that is given by the software. In all cases it is wise to allow safety margins on estimates, and to examine the effects any small changes in loading, materials behavior or other factors like surface condition may have on an answer.

2.4.2 Metadata Results

The key life and damage results are output on a metadata pad.



Typically, two metadata results sets are created. One contains the property settings on the glyph that created the result (called <glyph_name>_Properties) and the other has the results (<glyph_name>_Results). These metadata results are also sent to the time series, histogram, and multi-column output pads, if those data are created.

The content of the results set varies depending on the input loading type. The common results are:

TotalDamage—The total damage calculated from 1 repeat of the input loading. If a static failure condition has occurred (stress maximum is greater than UTS), then the damage value is reported but the DamageStatus result will be set to 1.

Life—The fatigue life, reported in equivalent units. This is usually repeats of the loading history (or histogram or multi-column data or repeats of the whole schedule if the input is a schedule). The properties EquivalentLifeString and EquivalentLifeValue can be set so that a more meaningful description of the loading sequence can be defined.

EquivalentUnitsValue	5	Multiplier on the life result
EquivalentUnitsString	Laps	The equivalent units string

The life is first calculated from the inverse of the total damage for one repeat. This is then multiplied by the equivalent units value to get the life that is reported. If the damage is 0.25 and the equivalent units are set to 5 laps, the life will be 5 * (1/0.25) = “20 laps”. The life result is always reported as a string. If the total damage is zero, the life cannot be calculated so the string is set to “Beyond cut-off”. If the maximum local stress has exceeded UTS, this is reported as “Static failure”.

DamageStatus—An integer, set to 0 for a successful calculation and 1 for a failed calculation. Examples of failed calculations include “Static failure”, when the UTS is exceeded, or a failure to converge a back calculation.

NumCyclesCounted—The total number of cycles counted and used in the analysis. Cycles that are removed by the gate are not included in this number.

Duration—This is set to the actual length of the time history or schedule that has been processed. If the input loading is a histogram, this may be passed through from its metadata.

ScaleFactor—The result of a back calculation on scale factor

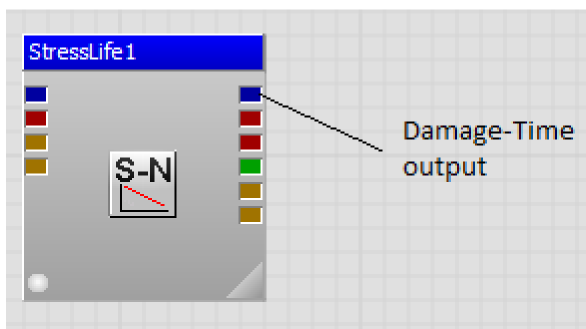
Kf—The result of a back calculation on Kf

DamagePerHour—This is output if the Duration metadata item is available and is calculated as 3600.0*TotalDamage/Duration. This assumes the duration is in seconds.

LifeInHours—Life is converted to hours using the Duration metadata item, if it is available.

2.4.3 Interpreting the Damage-Time History

A damage-time history is exported when the input is a time history. It is not created if the input loading is from a rainflow matrix or multi-column. Connecting to this pad is optional.

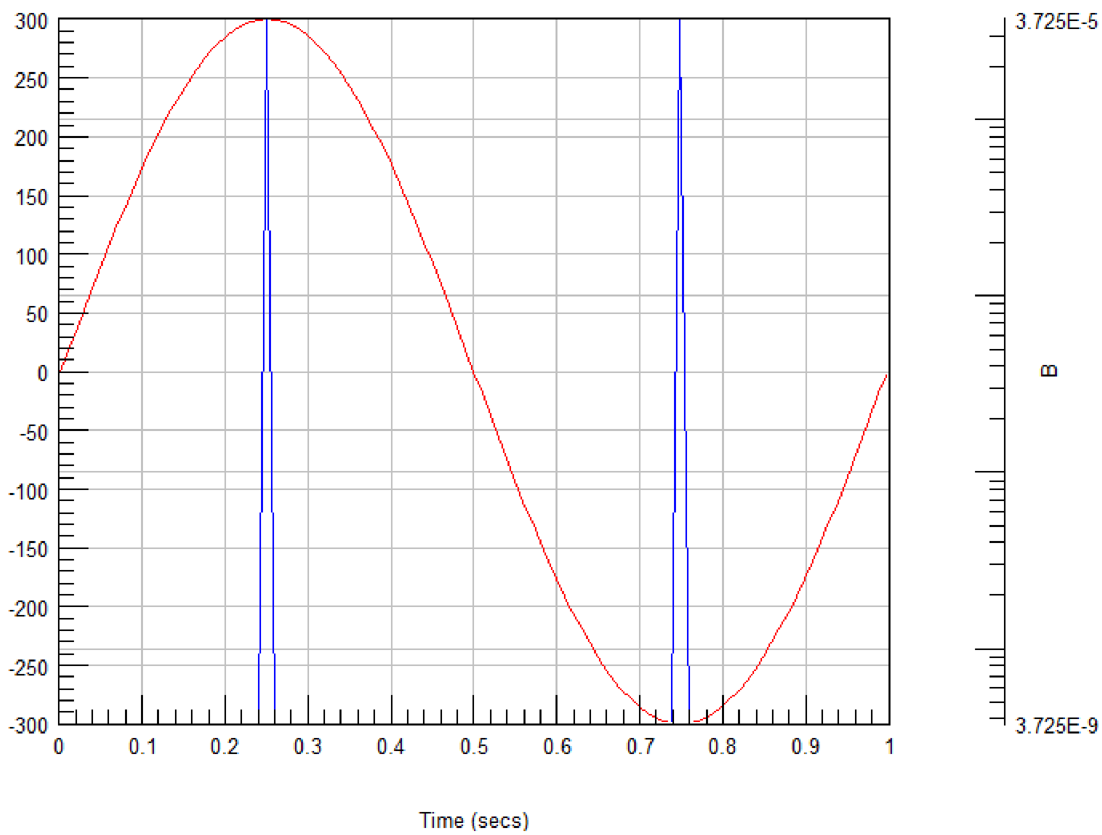


The output is dependent on the DamageTimeAccumulation property. The options are Standard and Holford.

Standard Damage Accumulation Method

For each input channel, an output time series channel is created with the same sample rate and length. Each point therefore corresponds to a point in the original strain time series. During the calculation, rainflow cycles are counted from the turning points in the data. When the cycles are counted, the position of the maximum and minimum of the cycle is remembered in the cycles list. From this, the glyph allocated half of the damage for each cycle to the maximum and minimum points of the cycle and this is what is shown in the damage-time series. A very simple example is a sine wave:

Fig. 2-36 Maxima and minima in a sine wave



Here there is one cycle, with the maximum and minimum at the extremes of the wave. The total damage in this case is 1.2504E-4 and half of this is allocated to the maximum and half to the minimum. The resulting damage-time history is shown overlaid in blue on the sine wave. The allocation of damage in this way is intended as a guide to the user as to which events within a time history are causing damage (or not causing it). The damage-time history is used by the DamageEditing glyph to remove sections of data that do not contribute to damage. This helps to reduce testing time.

Holford Damage Accumulation

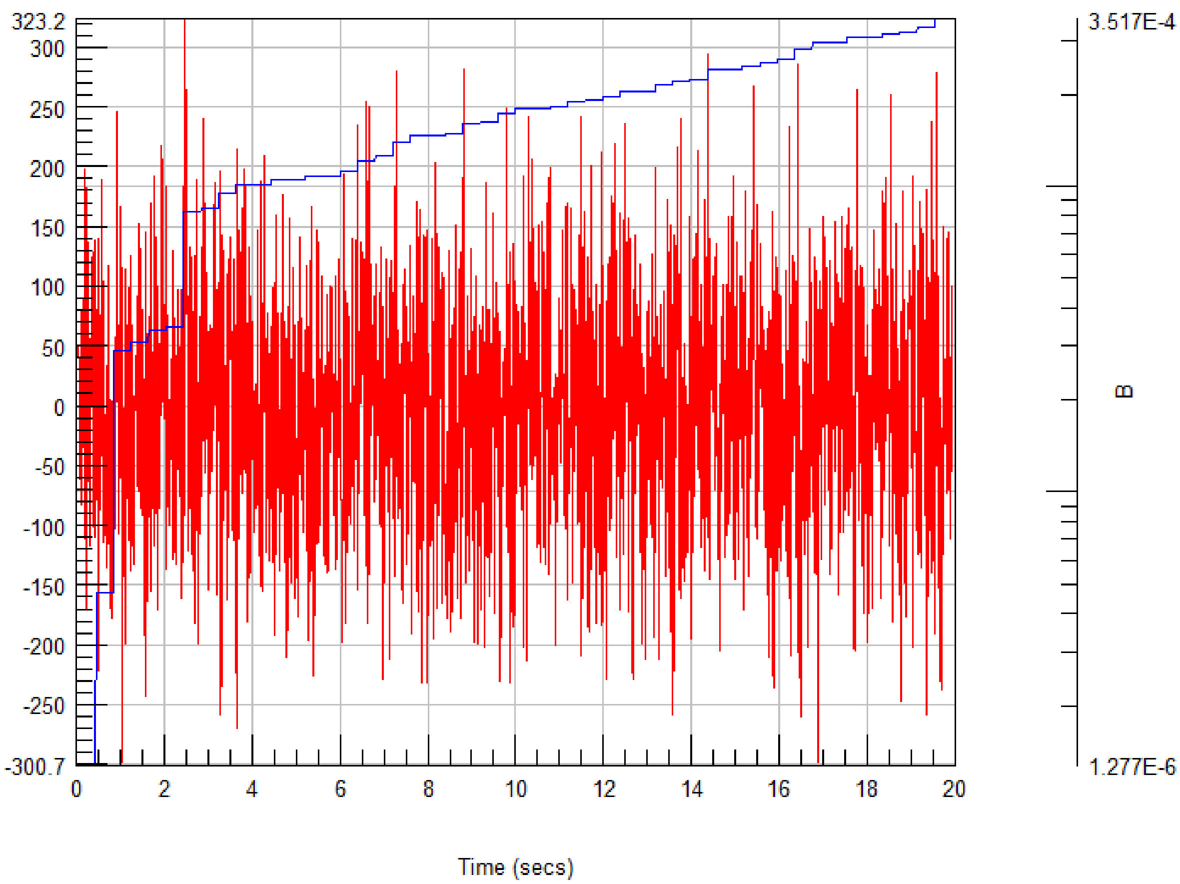
The Holford method breaks the time history into buffers that it treats independently for damage assignment. The damage is assigned to the whole buffer, not on a point-by-point basis. The HolfordOutputType property determines whether only the damage actually accumulated in the buffer is counted, or whether future damage partially caused by data in the buffer is included as an "estimate".

HolfordOutputType	Estimated	Specifies whether output includes the estimated damage per buffer

The options are "Estimated" and "Actual". The damage-time history looks different from the standard method because the damage is allocated per buffer, and is also cumulative.

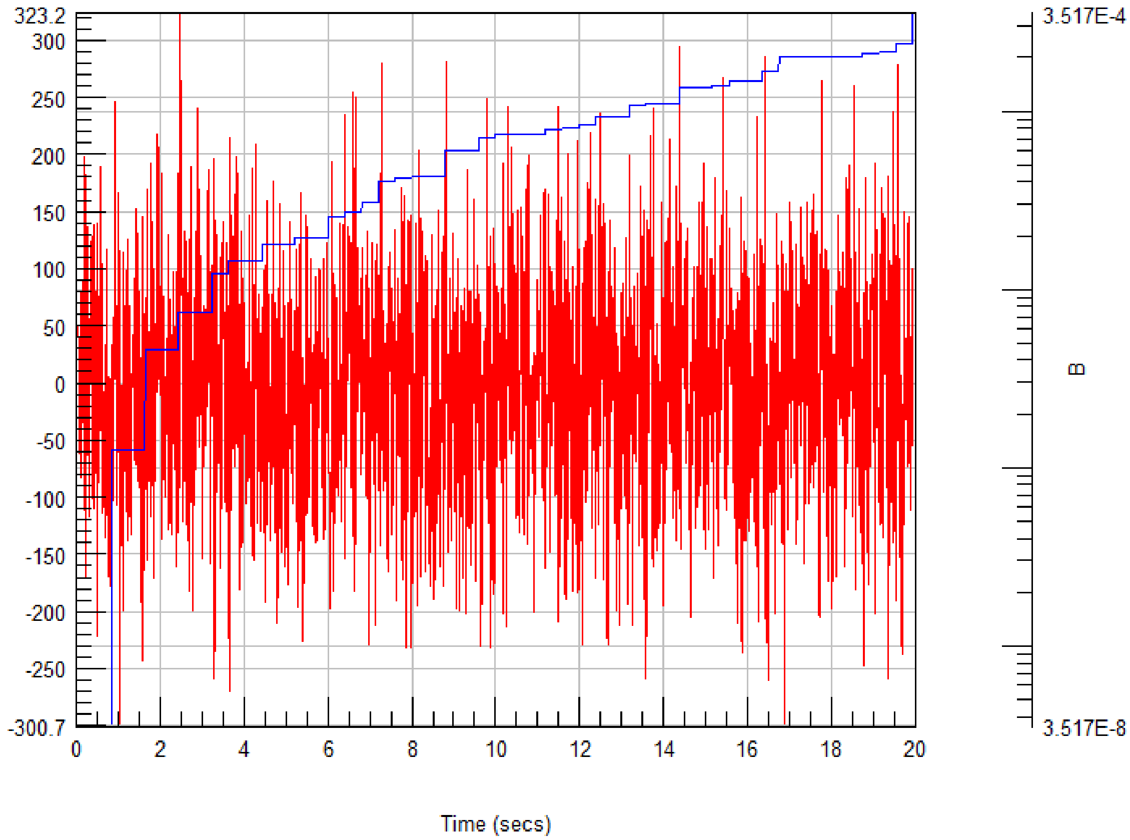
The example below shows the estimated damage in a random loading example:

Fig. 2-37 Estimated damage with random loading



The actual damage option accumulates differently, but results in the same total damage.

Fig. 2-38 Actual damage with random loading

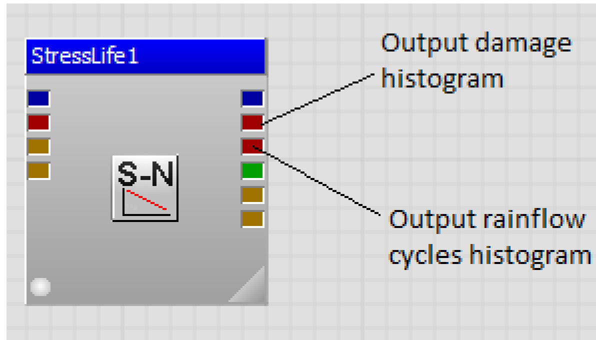


The glyph expects the buffer size (in seconds) to be entered. This must be at least 10 samples long and less than half the length of the input data.

HolfordBufferSize		Sets the buffer size for the Holford method in seconds

2.4.4 Histogram Outputs

When the input is a time history or a rainflow matrix, two output histograms are created, one containing the rainflow cycles that have been counted and one containing the damage.

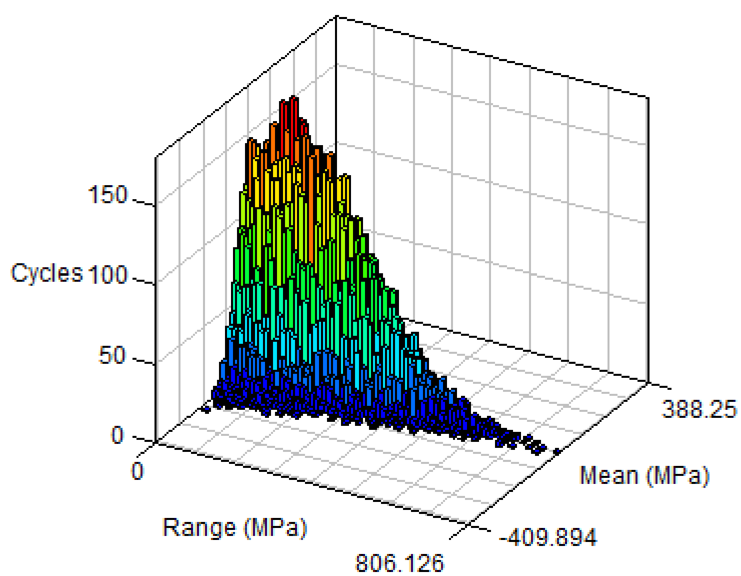


The number of bins in the output histograms is controlled via a property:

DamageHistogramBins	64	Number of bins in the output damage histogram

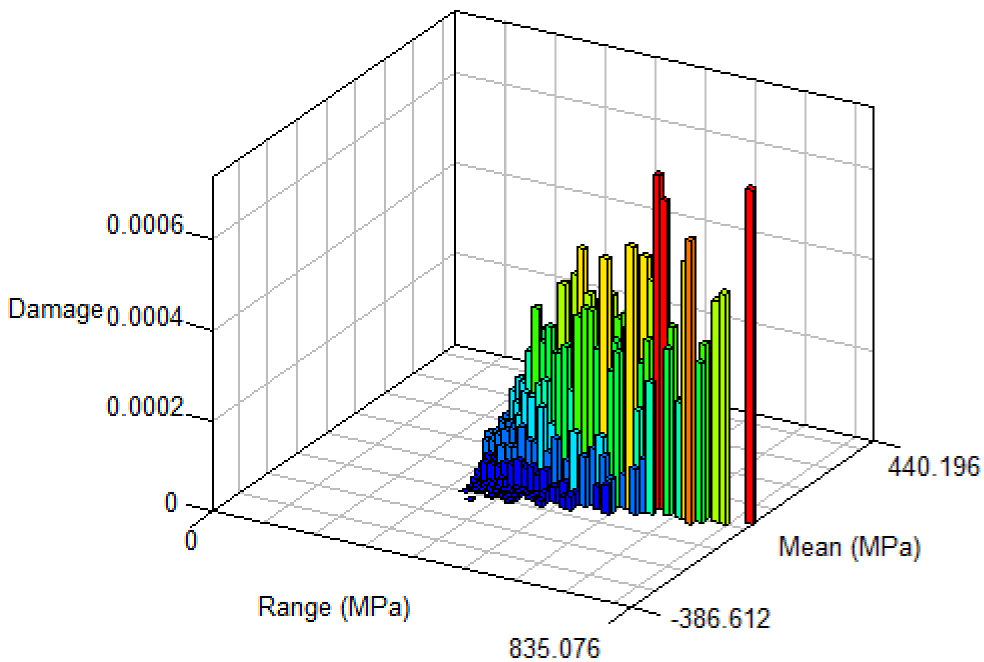
If the input is a time series, the rainflow cycles are counted and put into a range-mean matrix and the number of bins on each axis determined by the DamageHistogramBins property. A typical rainflow matrix looks like this:

Fig. 2-39 Rainflow matrix



The same bin sizes are used for the damage histogram. For the example shown above, the damage histogram looks like this:

Fig. 2-40 Damage histogram



Note that only the larger cycles are generating damage.

If using time series inputs, the rainflow histogram is used only for visualization. The unbinned list of cycles is used for damage. Binning is done only for viewing purposes with the rainflow histogram.

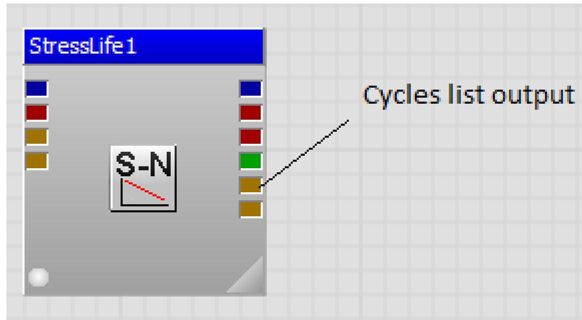
If the input loading is a rainflow matrix, then the DamageHistogramBins property has no effect. The output histograms will have the same axes and number of bins as the input histogram.

The units of the output histogram can be changed. This is controlled by the property:

StressOutputUnits	MPa	Specifies the units of stress used for histogram and cycles output

2.4.5 Cycles List Output

A full list of cycles used in the analysis is exported on a multi-column output pad.



The data in the columns is dependent on the input loading type. For a long time history analysis, many cycles can be generated and the data set can become very large. All the information on the size of the cycle, nominal and local stresses and strains, the direction of loading, and the damage associated with each cycle is in the table of results.

In order to limit the size of the cycles list, a threshold can be applied that prevents cycles being exported to the pipe if their damage is less than the specified value.

CyclesListThreshold	0	Cycles with damage values less than the threshold are not output

Because multiple cycles are binned together in a histogram, the cycle list has a "Number of cycles" column in each row. The reported damage is the damage for all those cycles, not for a single cycle of that size. So if the damage reported is 0.5 and there are 5 cycles, the damage for each cycle is 0.1. If the number of cycles in a bin is zero, then there will be no output from that bin.

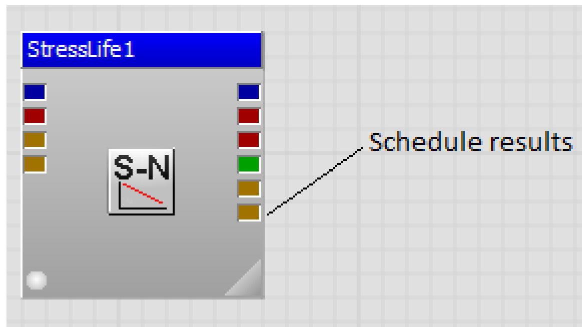
If the input is a multi-column loading, then the cycles list output will match the input (one row of input equals one row of output). Again there can be a number of cycles defined for the same size, and the damage reported per row is for all those cycles, if it is for a histogram and not the damage for a single cycle of that size.

The units of the output cycles list can be changed. This is controlled by the property:

StressOutputUnits	MPa	Specifies the units of stress used for histogram and cycles output

2.4.6 Schedule Results

If a schedule file is used for input loading, extra results are output onto a second multi-column pad.



These results show the damage for each channel, per event, and summed across events.

1	2	3	4	5	6	7
Channel number	Channel title	Event name	Damage	Percent damage	Damage with status	Repeat count
1	LF_Long_Accl	Event_A	0.002211	8.064	0.002211	1
1	LF_Long_Accl	Event_B	0.02521	91.94	0.02521	4
1	LF_Long_Accl	ALL	0.02742	100	0.02742	1
2	LF_Lat_Accl	Event_A	0	0	Beyond cutoff	1
2	LF_Lat_Accl	Event_B	0	0	Beyond cutoff	4
2	LF_Lat_Accl	ALL	0	0	Beyond cutoff	1
3	LF_Vert_Accl	Event_A	0.007676	32.96	0.007676	1
3	LF_Vert_Accl	Event_B	0.01561	67.04	0.01561	4
3	LF_Vert_Accl	ALL	0.02329	100	0.02329	1

The percent damage column shows the percentage of damage per event on the channel.

2.5 The Stress-Life Method: A Summary

The stress-life method relates stress cycles to fatigue life through a material's S-N curve.

- The stress-life method is ideally suited for cases where there is minimal plasticity. In can be used in low cycle regimes, but the assumption of linearity makes the estimate of life less reliable.
- The material parameters for the S-N curve are determined experimentally.
- Tensile mean stresses are detrimental to life while compressive mean stresses are beneficial.

- The S-N curve can be used for whole components, including complex geometries, welds, surface, and other effects.
- These methods are valid only for uniaxial (or near uniaxial) loading conditions.

3 Strain-Life Fatigue

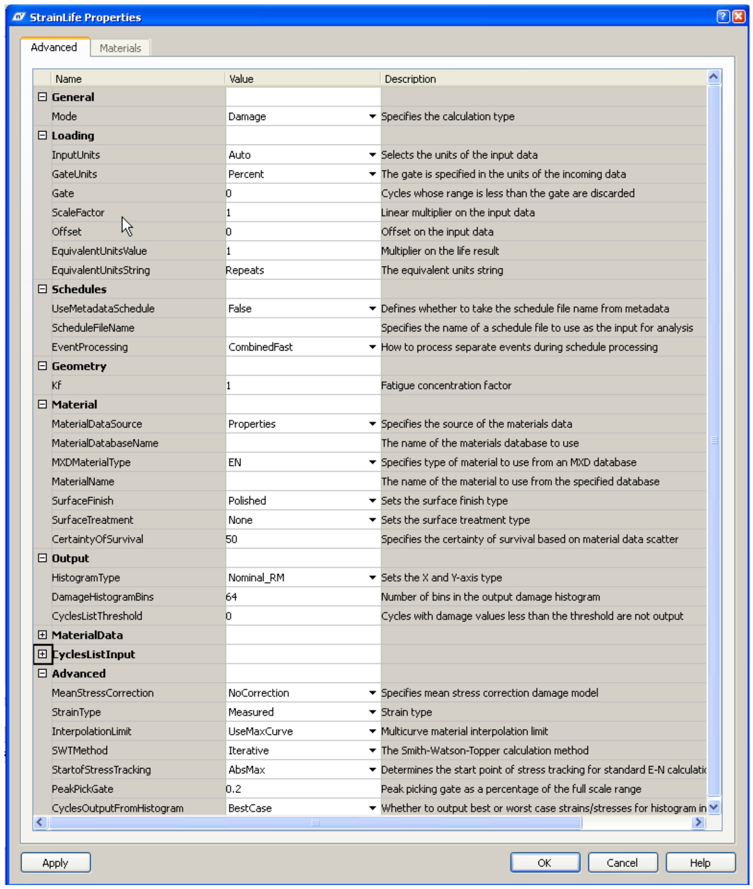
The StrainLife glyph is typically used when the inputs to the fatigue calculation are in measured strain, usually from a strain gauge. In this case the strains are potentially already elastic-plastic, but the glyph can also be used for fully elastic strains from, for example, a linear finite element (FE) analysis. The loading input strains can be in the form of a time history, a list of cycles, or a rainflow histogram. The method by which the local strain and stress information needed for the fatigue calculation is obtained using each of these loading types is outlined below.

Note that strain life analysis is sometimes referred to as EN where E (or more correctly ε) is strain and N is the life as a number of cycles to failure.

The materials data needed for strain life calculations comes typically from materials tests done under local strain control. The properties derived from those tests describe the cyclic stress-strain relationship and also the strain-life relationship. Static test properties, such as the Ultimate Tensile Strength (UTS), are also required as they are used by some of the correction algorithms. Full details of the materials properties are given in the Materials Data section below.

Additional properties control corrections for geometry, surface effects, mean stress, and other factors affecting fatigue life. These properties are found on the property form for the glyph, as shown in Figure 3-1.

Fig. 3-1 Property sheet for the StrainLife glyph

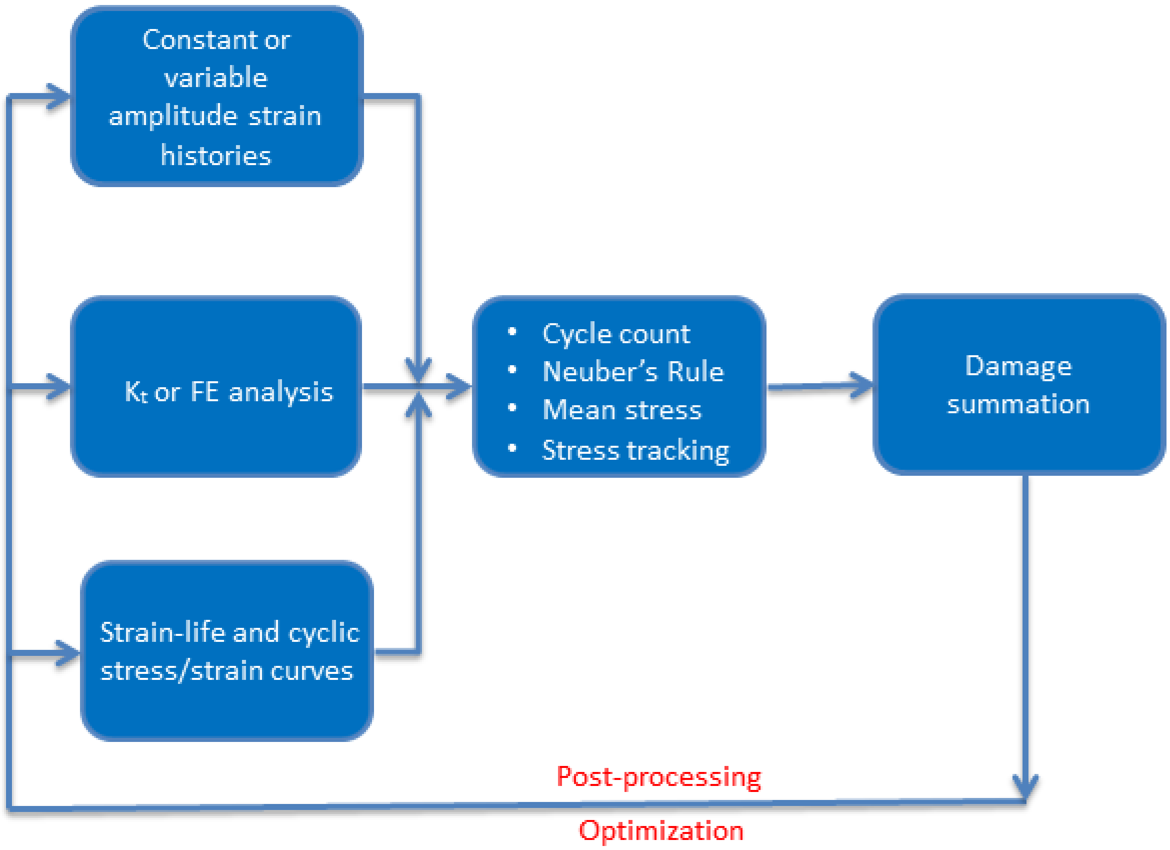


The basic process for performing strain-life calculations is as follows:

1. Make any corrections necessary to the materials curve to account for surface effects, geometry, etc. This usually occurs at the start of the calculation but may be done during the calculation on a cycle-by-cycle basis.
2. Determine the rainflow cycles from the input loading that drive the fatigue process, converting to strain if required. At the same time, estimate the total local strain (elastic and plastic) using a notch rule (e.g., Neuber's Rule) and track or estimate the shape and position of each hysteresis loop. In addition, local stresses are calculated to enable mean stress to be taken into account.
3. Calculate and accumulate damage from the strain-life curve information, using linear damage accumulation (Miner's Rule).

The five-box diagram for strain life looks like this:

Fig. 3-2 5 Box Diagram - strain life



For further background on the E-N method, refer to page 215 of the Introduction section of the uniaxial Strain-Life chapter in "Metal Fatigue Analysis Handbook"[4].

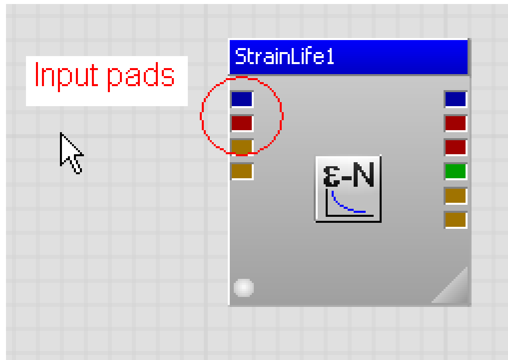
3.1 Loading

Loading in the StrainLife glyph comes in any one of four forms; three are controlled by passing data in on input pipes and the fourth by specifying a schedule (or duty cycle) via property settings. Figure 3-3 shows the input pads that are

4. Metal Fatigue Analysis Handbook: practical problem-solving techniques for computer aided engineering / Yung-Li Lee, Mark E. Barkey, Hong-Tae Kang ISBN 978-0-12-385204-5.

used for time series (blue), histogram (red) and multi-column data (the first of the two brown pads).

Fig. 3-3 Input pads on the StrainLife glyph



3.1.1 Time Series Input

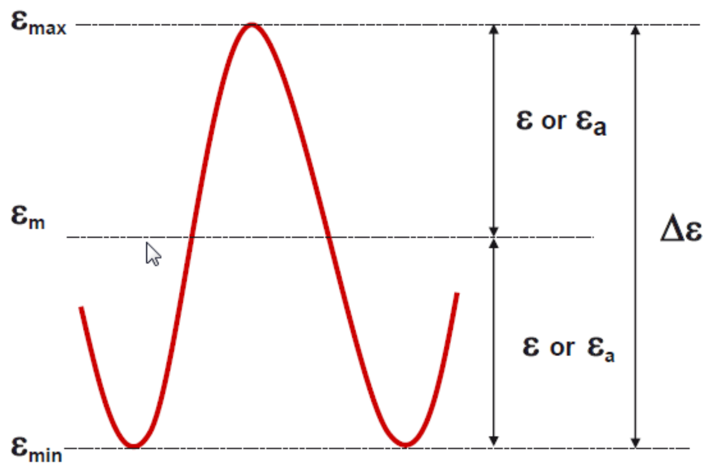
The Time Series input pad brings in multiple channels of data in the form of (typically) strain versus time, with constant time intervals between the samples. Conversion of the units from the input units to microstrain (or at the lowest level, strain) takes place by selecting the input unit on the property form:

⊟ Loading		
InputUnits	Auto ▾	Selects the units of the input data

The Auto option attempts to recognize the input units from the metadata of the input data. This must be a unit of strain, or of stress, and be defined within nCode's units system. If the data is in stress units, it is converted to strain using Young's Modulus from the material properties and assumes both linearity and a uniaxial stress state.

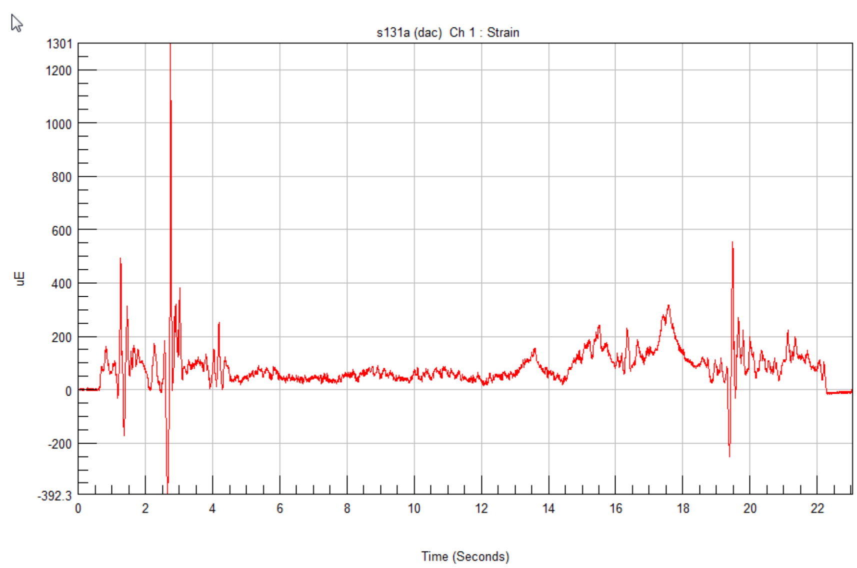
The simplest of loadings is the constant amplitude type, often but not necessarily sinusoidal. The shape of the time history, and the frequency content, has no effect in this type of fatigue calculation, but the amplitude of the cycles is critical. The mean stress is also important, as will be explained later. The diagram shows a typical strain cycle, and shows the definitions of maximum strain, minimum strain, mean strain, strain amplitude and strain range. These terms will all be used later.

Fig. 3-4 Typical strain cycle



If the time history is randomly loaded, rainflow counting then takes place to determine the cycles that drive fatigue damage. Here is an example of a “variable amplitude” loading history.

Fig. 3-5 Variable amplitude loading history

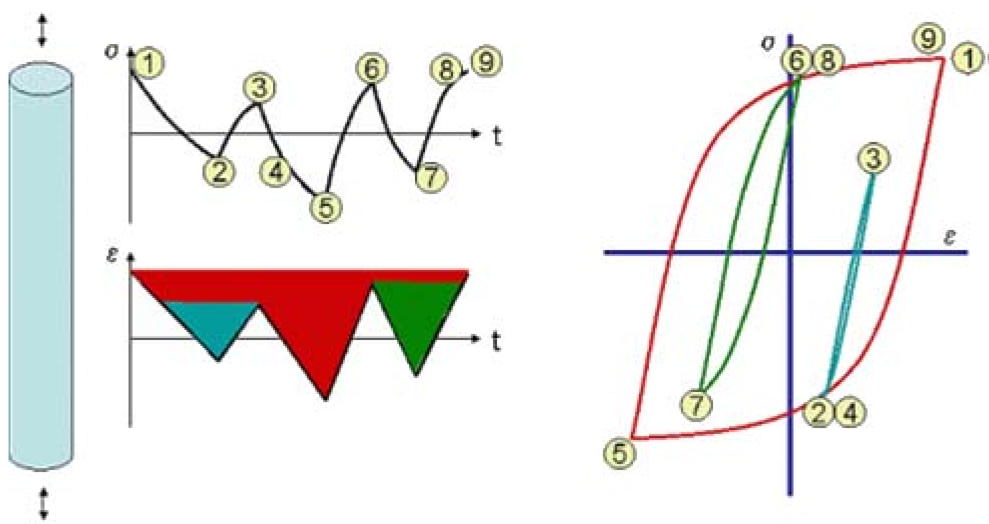


Often time histories appear to be random but aren't. However, the process of rainflow cycle counting doesn't need to know this, and will calculate the cycles simply from the time history data it is given.

Assuming that the strain-time history describes the local elastic-plastic strain, without any corrections for geometry, then the following procedure describes the counting and tracking.

Rainflow cycle counting for the local strain approach is based on the same principles as for the basic counting method, as used in the Rainflow glyph and the stress-life calculations, but there are additional requirements in that we need to know the total (elastic-plastic) strain range for each cycle, together with its mean or maximum stress. Consider the case of a simple bar, subjected to variable amplitude loading under uniaxial conditions, and suppose we can determine the total strain and stress histories. If we plot stress against strain, the stress-strain history will appear as a number of nested hysteresis loops, each corresponding to a rainflow cycle, as illustrated in Figure 3-6.

Fig. 3-6 Hysteresis loop formation



Each excursion takes the shape of part of the hysteresis curve. Starting at point 1, we follow the hysteresis curve as far as point 2. At this point the strain direction reverses, so we start again from point 2 up to 3, and from point 3 we reset to the beginning of the hysteresis curve again until point 4. The sequence 2-3-4 represents a closed rainflow cycle. At point 4, the well-known material memory effect has to be considered. The material appears to “remember” that it was interrupted after starting from point 1, so when it gets to point 4 it resumes along the hysteresis curve starting from 1 until it reaches the next turning point at 5. By continuing in this way, we will find that the entire history can be tracked and reduced to a number of hysteresis loops that can be identified with rainflow cycles. For each cycle, we can determine the important parameters we need for a fatigue calculation, namely the strain range and mean or maximum stress.

The identification of these cycles uses the cyclic stress-strain curve defined as part of the materials data. This is described in more detail in "Materials Data" on page 97.

Note that FE-based E-N fatigue calculations are very often based on elastic FE results, so that the equivalent strain history generated by the E-N analysis engine based on the load provider is a pseudo-elastic strain history. In reality, there may be plasticity, especially at critical locations, and this needs to be estimated if realistic life predictions are to be obtained. To tell the software that the data is linear elastic, set the StrainType property to FullyElastic. This is normally set to Measured, meaning it is measured strain gauge data.

In addition, a geometric fatigue concentration factor can be applied to account for increased strain and stress in the root of a notch.

In these circumstances, the tracking rainflow cycle counting procedure has to be modified to include the Neuber notch correction procedure, to estimate the total strain range of each cycle and to allow the resulting hysteresis loops to be correctly positioned.

The Neuber method is described later. For the basic time history calculation, we assume we have access to the entire time history segment, and that the segment repeats a significant number of times thereby negating the fact that it takes a number of cycles for the material to stabilize; over tens or hundreds of repeats this small difference will be lost. A method that tracks from the beginning is described later, but for the standard method the steps in the calculation are as follows:

1. Reduce the strain history to a peak-valley sequence, and re-order it to start from the Abs Max value, with the first point repeated at the end.
2. Position this point in stress-strain space using the cyclic stress strain curve and using the Neuber notch correction method to estimate the elastic-plastic stress and strain.
3. From this point, calculate the next excursion using the hysteresis curve and the Neuber correction again.
4. Repeat this process. When a cycle or cycles close, remove these points from the sequence, noting the calculated total strain range, mean, and maximum stress for the cycle, and reset the starting point for the next strain excursion to the last remaining point.

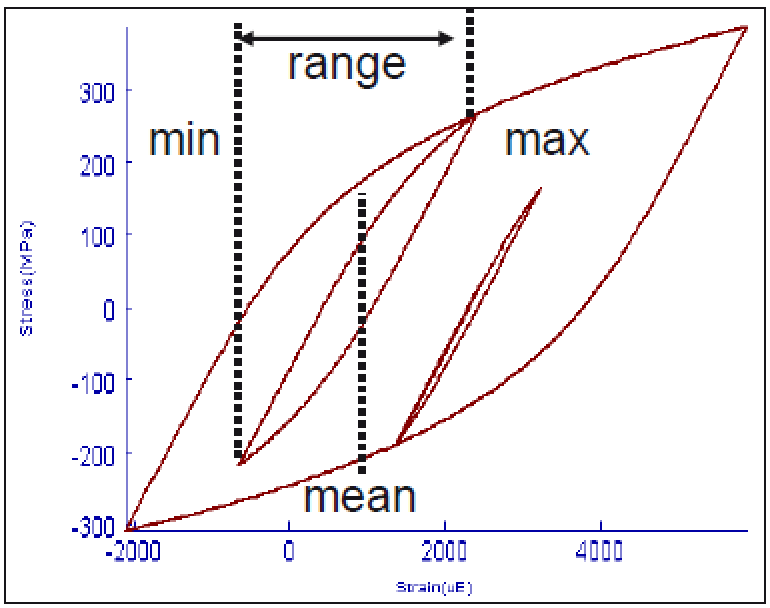
5. Continue until the end of the history is reached and all cycles have closed.

At this stage we have a "cycles list" with all the information needed to calculate the damage for each cycle and then the total damage. This is the same calculation whether the loading data is from time history, histogram, multi-column or schedule files, and therefore this is described below in the section "Calculating Damage" on page 121.

3.1.2 Histogram Input

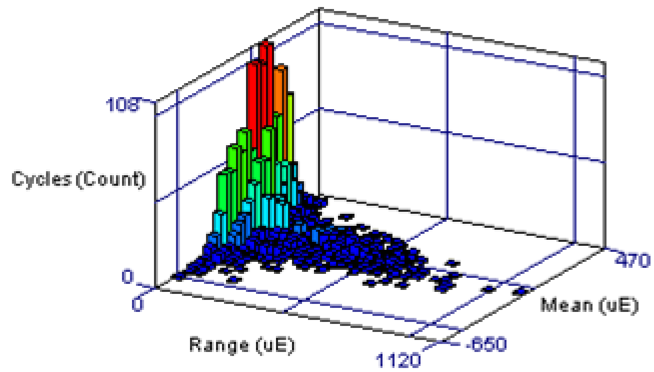
The histogram input pad can bring in a matrix of pre-counted rainflow cycles that have been put into even or non-evenly spaced bins within one of four types of matrix, that characterize cycles using the mean, range, max and min as shown below:

Fig. 3-7 Hysteresis curves: mean, range, max, min



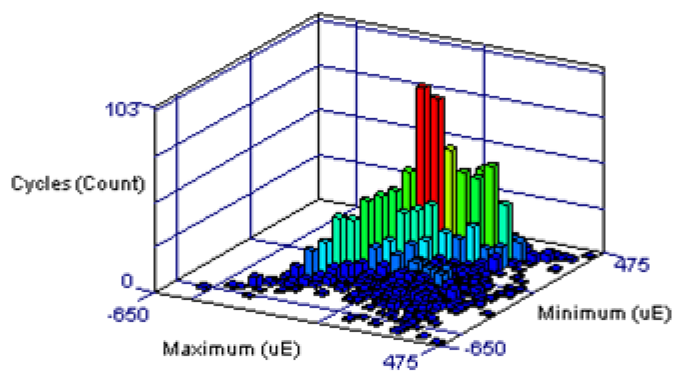
Range-Mean—One axis has bins split by range of the cycle (max-min) and the other axis is the mean of the cycle ((max+min)/2).

Fig. 3-8 Range-mean 3D histogram



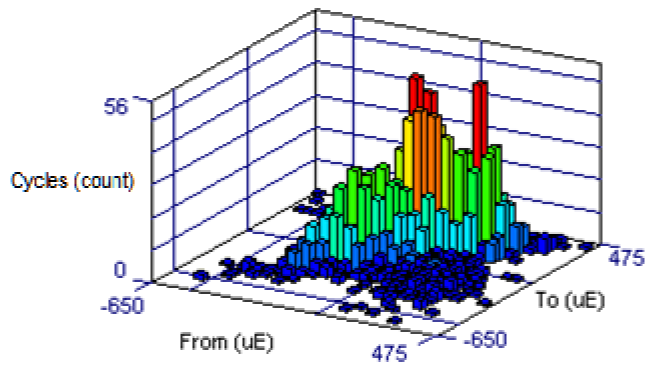
Max-Min—One axis is the maximum value of the cycle, the other the minimum.

Fig. 3-9 Max-min 3D histogram



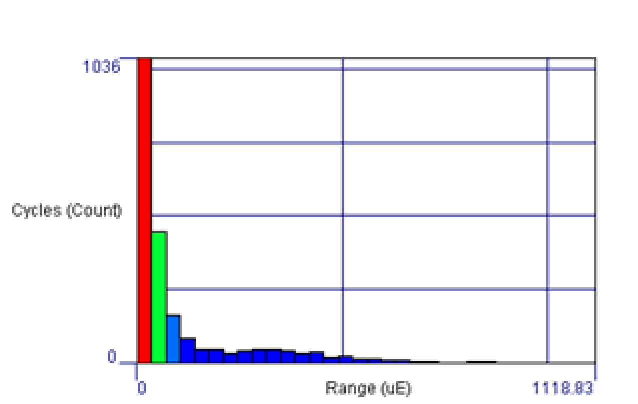
From-To—This is similar to a Max-Min but takes into account the sequence of the points. One axis contains the binned values for the “From” values, the first in the sequence, with the second axis being the “To” value, the second in the sequence.

Fig. 3-10 From-to 3D histogram



Range Only—This is a single-axis histogram that uses just the cycle range.

Fig. 3-11 Range only 2D histogram



The input histogram is checked to understand what format it is, i.e., the algorithm is smart enough to understand the difference between Range-Mean and From-To.

There are two issues in using histogram information for strain-life fatigue calculations. The first is that the sequence of the cycles has been lost and so true "tracking" of the hysteresis loops to determine the stress values is not possible. The second is that the data has been discretized into bins and as such the exact value of each cycle is not known.

To get around this, and produce reasonable estimates, the StrainLife glyph calculates two results to bracket the estimate you would get if the full cycle information were available.

To get the “best case” or “minimum damage” value, the software uses the location within each bin that gives the lowest cycle range and the lowest mean value. For the “worst case” or “maximum damage” value, the software uses the location within each bin that gives the highest cycle range and the highest mean value. These locations vary according to the type of histogram.

Only bins with a non-zero count value are used to populate the cycle lists and the number of counts is also carried within the list, because this will be used to multiply the damage calculated for a single cycle.

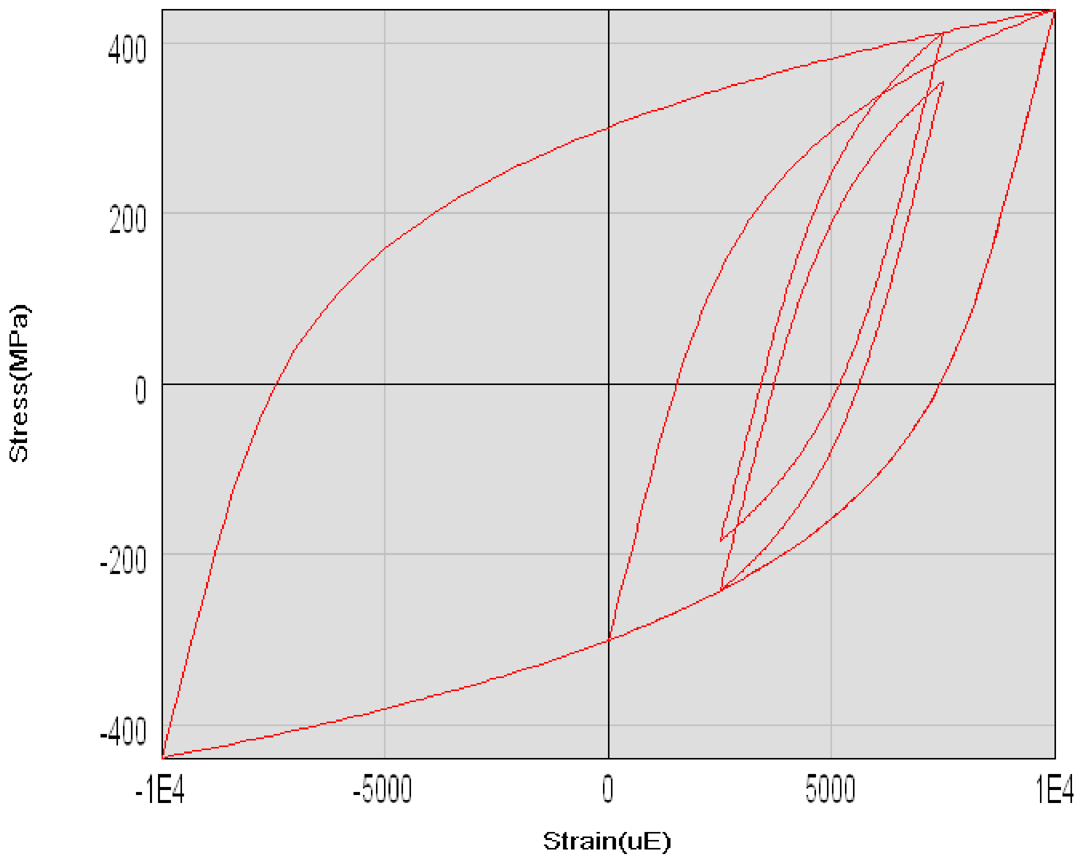
This generates two lists of cycles in terms of their maximum and minimum nominal strains. They now have to be converted to local strains and stresses in order to generate full cycle lists that can be used to calculate damage.

First of all, the “outside loop” must be determined. This is found by looking within the histogram for the cycle with the largest range. Occasionally this can find two with the same range but different mean values, and the one with the largest mean is typically chosen.

Once the basic outside loop coordinates are determined, Neuber's Rule is used to calculate the full outside loop (Neuber is explained in more detail below). This is used to account for linear elastic strain inputs or the use of the fatigue concentration factor Kf to account for notches.

The inner loops are more difficult to determine. For example, take a time series sequence 0,10000,2500,7500,-10000,7500,2500,10000. This generates cycles of 10000:-10000, one of 2500:7500 and another of 7500:2500 (these are given in their from:to variant). The hysteresis loops from the counted time series are shown in Figure 3-12:

Fig. 3-12 Hysteresis loops from a simple time series



If these are binned into a Range-Mean, Max-Min or Range-Only matrix, the two smaller cycles will appear in the same bin and be indistinguishable. To take account of the differing mean stresses, the “best case” cycle list assumes that all the inner cycles “stand” from the lower limb of the outer loop and the “worst case” list assumes that all the inner cycles “hang” from the outer loop. This simple example shows that both answers (best and worst) will be incorrect but will give damage values that bracket the answer that would be calculated from knowledge of the full history.

Although the From-To matrix has more information, and this can be used in the simple example above to hang and stand the two inner cycles correctly, this still cannot be fully accurate because it does not handle loops that are positioned inside inner loops.

Note that some of the cycles recovered from the matrix may not fit correctly inside the outer loop due to either rounding issues in the binning algorithm, or a badly formed matrix. Where possible, the algorithm adjusts for these situations, but in some cases it may not be able to process the histogram. This can happen if two histograms with significantly different means are summed together; if they were added as time histories, an extra cycle would be counted that spanned the two sections of time history missing from the summed histograms. This would be the outer loop, but is missing so the outer loop that is found may not contain all the other loops, leading to problems with the calculation. The warning message that is output by the software reads “The rainflow matrix contained invalid cycles. One or more inner loop cycles lie totally outside the calculated outside loop. Damage accumulated from these cycles may not be valid”.

Neuber's Rule is used on the inner loops as well as the outer loops to get the full coordinates of stress and strain for each cycle.

At this stage, we have two “cycles lists” with all the information needed to calculate the damage for each cycle and then the total damage for the best and worst cases. This is the same calculation whether the loading data is from time history, histogram, multi-column or schedule files, and therefore this is described in the section “Calculating Damage”.

Once we have two damages, these can be used to calculate the best and worst lives. The average of the lives is the main result provided by the software, but the best and worst lives are also provided. Note that the average of the lives is not the same as the average of the damages. The results are written to metadata as described in “Metadata Results” on page 134.

For example, if best case damage is 0.1 (life=10 repeats) and worst case is 0.5 (life=2 repeats), the average life is 6 repeats. The average damage is 0.3, which would give a life of 3.33 repeats. When using histograms, understanding the spread of the best and worst estimates is important in order to interpret the validity of the estimate.

Note Because the histogram generates two output cycles lists, one for the best case and one for the worst case, an option that selects which of these is sent to the output pads is provided. This is explained below.

3.1.3 Multi-Column Input

The multiple column input to the StrainLife glyph effectively defines the cycles list directly, at least in terms of its input strain components. Each row defines a cycle or a set of exactly equivalent cycles (defined by a RepeatCount column). The cycles can be defined in different ways, using the CyclesListInputType property.

The types available are MaxMin, RangeMean, AmplitudeMean and Point1Point2. In each case, a separate column is used to define Max and Min, Range and Mean, Amplitude and Mean, Point 1 and Point 2 respectively. Point1Point2 allows the user to identify the sequence in which the points in the cycle occur. The time sequence 10000, 2500, 7500, -1000, 7500, 2500, 10000 generates 3 cycles. The cycles, in Max Min terms, are

10000,-10000

7500, 2500

7500, 2500

or

10000,-10000,1

7500,2500,2

(if you have a third column defining the RepeatCount). Note that an input file in CSV format could be created in exactly this way and the glyph would recognize these as cycles without any extra header information.

The Point1Point2 output would look like this:

10000,-10000

2500,7500

7500,2500

The Point1Point2 type is processed differently if the CyclesListMethod is set to CombinedFast.

The default method is Independent, which means that each row is processed independently. In this case, each cycle is treated as if it were its own constant amplitude time history, and there is no connection whatsoever with any of the other cycles.

If this is too much of an oversimplification, and there are many cases where summing the damages from independent cycles would give a radically different result to processing them as a connected block of cycles, the CombinedFast method can be used. In this method, a rainflow tracking cycle counter is used (see the time history section for a more detailed explanation of this) and the cycle points (max and min, or point1 then point2) are passed into it in order so that cross-cycle effects and sequence effect is taken into account. For the max and min input type, the max value is always inserted first into the generated time history sequence, followed by the minimum value. For the “Point 1 - Point2” type, they are inserted with Point 1 first and Point 2 second. The type is determined by the column headings in the input file.

The cycles that are counted from this method will be put into the output cycles list. This may or may not match the input cycles list when CombinedFast is set to True, but will match it if Independent mode is used.

If the cycles columns are not labeled, and/or they are not defined as the first columns in the multi-column data, the glyph allows the user to specify the columns for the data and for the repeat count.

CyclesListInputColumns		Specifies the column numbers or keywords that have the cycles data in them.
CyclesListInputRepeatCountColumn		The column identifier that has the repeat count for the cycle defined in that row.

3.1.4 Schedule Files

Schedule files can be used to combine events that are stored in separate data files. Schedule file inputs can improve processing speed by taking advantage of the Palmgren-Miner Rule, which states that damage increases linearly as cycles are repeated. This means that damage for multiple repeats of an event can be calculated simply by multiplying the damage for a single repeat by the number of repeats. Further, total schedule damage can be found simply by summing each event's subtotal damage. This schedule damage summation can have huge performance benefits over working with very long concatenated time series inputs.

In the StrainLife glyph, a schedule file can be specified using two methods. The first is to set the name in a property.

ScheduleFileName		Specifies the name of a schedule file to use as the input for analysis

In this case, the time history input pad is ignored and can be disconnected.

Alternatively, it can be specified by passing the schedule filename in on the time history input pad with the UseMetadataSchedule property set to True. The Schedule name is read from metadata item Schedule.ScheduleName.

⊟ **Schedules**		
UseMetadataSchedule	False	Defines whether to take the schedule file name from metadata

Using the schedule file name (with no data on the input pad) is the most efficient way to process schedules. A large schedule brought in on a pipe will still slow the process down because the software is still transferring data in memory and will be performing statistics calculations.

Schedules of histograms and multi-column, which can be created within the software, are not supported in the glyph. These must be processed by combining the data together in the input glyph.

A schedule, typically created with the ScheduleCreate module, is defined as a series of events and a repeat count. For example, a schedule could be defined as

```
<TimeSeriesSchedule Version="2.0" Channels = "1" PadValue="0"
PadMissingChannels="None"><Join Time="1.0000" Type="None"/>
<Taper Time="1.0000" Type="None"/>
<WrapEnds Wrap="0"/>
<Events>
<Event Name="LT1_r01" Active="True" Testname="LT1_r01_1chan"
RepeatCount="154088"/>
<Event Name="LT1_r02" Active="True" Testname="LT1_r02_1chan"
RepeatCount="7722"/>
<Event Name="LT1_r03" Active="True" Testname="LT1_r03_1chan"
RepeatCount="23917"/>
<Event Name="LT1_r04" Active="True" Testname="LT1_r04_1chan"
RepeatCount="2500000"/>
</Events>
</TimeSeriesSchedule>
```

This defines four events. Note the very high number of repeat counts. If this schedule were combined into a single time series before the glyph, the processing time could be very long. Note that the same time history file can be re-used later in the schedule, as the sequence simply takes one event after another. For example, this would be a valid schedule:

```
 <TimeSeriesSchedule Version="2.0" Channels = "1" PadValue="0"
PadMissingChannels="None"><Join Time="1.0000" Type="None"/>
<Taper Time="1.0000" Type="None"/>
<WrapEnds Wrap="0"/>
<Events>
<Event Name="LT1_r01" Active="True" Testname="LT1_r01_1chan"
RepeatCount="154088"/>
<Event Name="LT1_r02" Active="True" Testname="LT1_r02_1chan"
RepeatCount="7722"/>
<Event Name="LT1_r03" Active="True" Testname="LT1_r03_1chan"
RepeatCount="23917"/>
<Event Name="LT1_r04" Active="True" Testname="LT1_r04_1chan"
RepeatCount="2500000"/>
<Event Name="LT1_r02a" Active="True" Testname="LT1_r02_1chan"
RepeatCount="15000"/>
<Event Name="LT1_r04a" Active="True" Testname="LT1_r04_1chan"
RepeatCount="10000"/>
</Events>
</TimeSeriesSchedule>
```

The advantage gained in processing the schedule within the glyph rather than combining the time series together on input is primarily in performance. If a large number of repeat counts is defined for the events within the schedule, the Time Series input glyph will create very long time histories and a lot of processing will be performed that isn't strictly necessary.

Once the glyph has access to the schedule directly, it can offer two types of calculation: “Independent” and “CombinedFast”.

EventProcessing	CombinedFast	How to process separate events during schedule processing

Independent mode calculates the damage for each event separately from the others and then sums those damages (including repeat counts) to get a total damage for the whole schedule. This will clearly be quicker than processing the whole schedule if there are significant numbers of repeats. However, cycles that cross events will not be counted. These cycles can produce significant damage and should ideally be included in the calculation.

Setting the EventProcessing property to CombinedFast will take account of cross-event cycles and give results that are very close or identical to using the full combined time history. However, each event is processed only once, so the performance advantage of using the schedule is retained.

3.1.5 Applying Corrections to the Input Loading

A scale factor and offset can be applied to the input loading.

ScaleFactor	1	Linear multiplier on the input data
Offset	0	Offset on the input data

This is done as a y=mx + c linear calibration, where m is the scale factor and c is the offset. In a time series or schedule loading, this is applied to each point in turn.

For histogram loadings, the factor and offset are applied to the histogram limits (XMin, XMax, YMin, YMax). This calculation is slightly different for Range-Mean, Min-Max, From-To, and Range-Only histogram types.

In a multi-column input, the factor and offset are applied to the cycle maximum and minimum values.

A “gate” can be applied to the cycles that are counted that allows only larger cycles through. These are typically more damaging than smaller ones, although mean stress effects can change that. Usually, a relatively small value is set to exclude very small cycles that cause very little damage and where the mean effect is insignificant even at high mean levels.

This is set using the properties GateUnits and Gate.

GateUnits	Data	The gate is specified in the units of the incoming data
Gate	0	Cycles whose range is less than the gate are discarded

The gate unit options are “Data”, where the gate is specified as a value in the input data units, and “Percent”, where the glyph calculates the gate to use as a percentage of the total range of the input data.

For example, if the input data is in microstrain, the GateUnits are set to Data and the Gate is set to 100, then any cycles whose range (max-min) is less than or equal to 100 microstrain will have a damage value of zero assigned to them.

If the input data goes from -2000 to 3000 microstrain, the GateUnits property is set to Percent and the Gate property is set to 10, then cycles with a range less than or equal to 500 microstrain will be assigned a damage of 0.

The gate applies to all the loading types, time history, histogram, multi-column and schedule file. In the case of a schedule file, the percentage gate option is applied differently depending on whether the schedule mode is Independent or CombinedFast. In CombinedFast mode, a single percentage value is calculated across the whole range of the signal input, even if the maximum and minimum are in different events. In Independent mode, the percentage gate value is calculated and applied on a “per event” basis.

Because the scale factor is applied after the gate is checked, cycles are gated based on their original size, not on the scaled size.

3.2 Materials Data

The basic requirement for a strain-life analysis is access to materials data in the form of a parameterized cyclic-stress strain curve and a strain-life curve, which may be parameterized or in the form of X-Y pairs (typically a family of curves based on mean stress, R-ratio or temperature). In addition, the UTS is used to determine static failure conditions due to the calculated stresses being too large, and in some mean stress corrections. See “Interpreting the Results Generated by the StrainLife Glyph” for more details on how the glyph handles the case where the UTS is exceeded.

The data can come from a variety of sources. These are set by the properties MaterialDataSource and MaterialDatabaseName.

Material		
MaterialDataSource	Properties	Specifies the source of the materials data
MaterialDatabaseName		The name of the materials database to use
MXDMaterialType	EN	Specifies type of material to use from an MXD database
MaterialName		The name of the material to use from the specified database
SurfaceFinish	Polished	Sets the surface finish type
SurfaceTreatment	None	Sets the surface treatment type
CertaintyOfSurvival	50	Specifies the certainty of survival based on material data scatter

Data source options are as follows:

Properties—The basic parametric materials data can be entered directly into the glyph property sheet. This does not support multiple curves or gray iron data.

MaterialData		
MaterialStressUnits	MPa	Selects the units of the stress type material properties
Material_UTS		Ultimate Tensile Strength
Material_E		Youngs Modulus
Material_Sfp		Fatigue strength coefficient
Material_b		Fatigue strength exponent
Material_efp		Fatigue ductility coefficient
Material_c		Fatigue ductility exponent
Material_np		Cyclic strain hardening exponent
Material_Kp		Cyclic strength coefficient
Material_Nc	2e+008	Fatigue cut-off
Material_StandardErrorElastic	0	Standard error of log(e) - elastic
Material_StandardErrorPlastic	0	Standard error of log(e) - plastic
Material_StandardErrorCyclic	0	Standard error of log(e) - cyclic

MDM database —Basic properties can be extracted from the MDM database format. This is an obsolete format used in the legacy products nSoft and FE-Fatigue. A valid material name corresponding to a database entry must be specified in the MaterialName property. Multiple curve and gray iron data are not supported in this format.

MXD database —This is the database format supported in GlyphWorks and DesignLife; data can be entered into the database using the nCode MaterialsManager. A valid material name corresponding to a database entry must be specified in the MaterialName property.

For both database types, GlyphWorks also provides a selection dialog that allows the database to be specified, and the material name to be selected from a list of available materials of the appropriate type.

Note	If you choose either database option, you can select which database and which material using the Materials tab.

Input Pipe—The lower of the two multi-column input pipes allows multi-curve data to be passed in to the glyph from an external source. Standard parametric and gray iron data are not supported using this method.

3.2.1 Materials Data Types

Standard Parametric Data

The standard strain-life material properties, which can be found in the MXD material database, consist of a set of generic properties (which can be inherited from a generic parent) and a set of specific properties that define the shapes of the stress-strain and strain-life curves used by the E-N analysis engine.

The generic properties are as follows:

Parameter Name	Description
MaterialType	Numeric code defining the type of material. The material type is required for correct application of surface finish. For a complete list of codes, see "Material Type Codes" on page 193.
UTS	Ultimate tensile strength. This is required to apply the static failure criterion.
YS	Yield stress. Optional (not currently used).
E	Modulus of elasticity. Required to convert from linear stress to strain and as part of the stress-strain curve.
Me	Elastic Poisson ratio (not used in GlyphWorks)
Mp	Plastic Poisson ratio (not used in GlyphWorks)
Comments	
References	

The specific E-N properties are as follows:

Parameter Name	Description
Sf’	Fatigue strength coefficient, σ_f'
b	Fatigue strength exponent
c	Fatigue ductility exponent
Ef’	Fatigue ductility coefficient, ε_f'
n’	Cyclic hardening exponent
K’	Cyclic strength coefficient
n’90	Cyclic hardening exponent for 90-degree out-of-phase multiaxial loading. This is not used and therefore not required for standard EN calculations.
K’90	Cyclic hardening coefficient for 90-degree out-of-phase multiaxial loading. This is not used and therefore not required for standard EN calculations.
Nc	Fatigue cutoff. Damage is set to zero beyond this point. It is set in REVERSALS. 2 reversals = 1 cycle
SEe	Standard error of log(plastic strain)
SEp	Standard error of log(elastic strain)
SEc	Standard error of log(cyclic strain)
Ne	Endurance limit – in reversals, used only for material corrections
FSN	Fatemi-Socie parameter (not currently used)
S	Wang-Brown parameter. This is not used and therefore not required for standard EN calculations.

The properties are used to define two basic relationships, as follows.

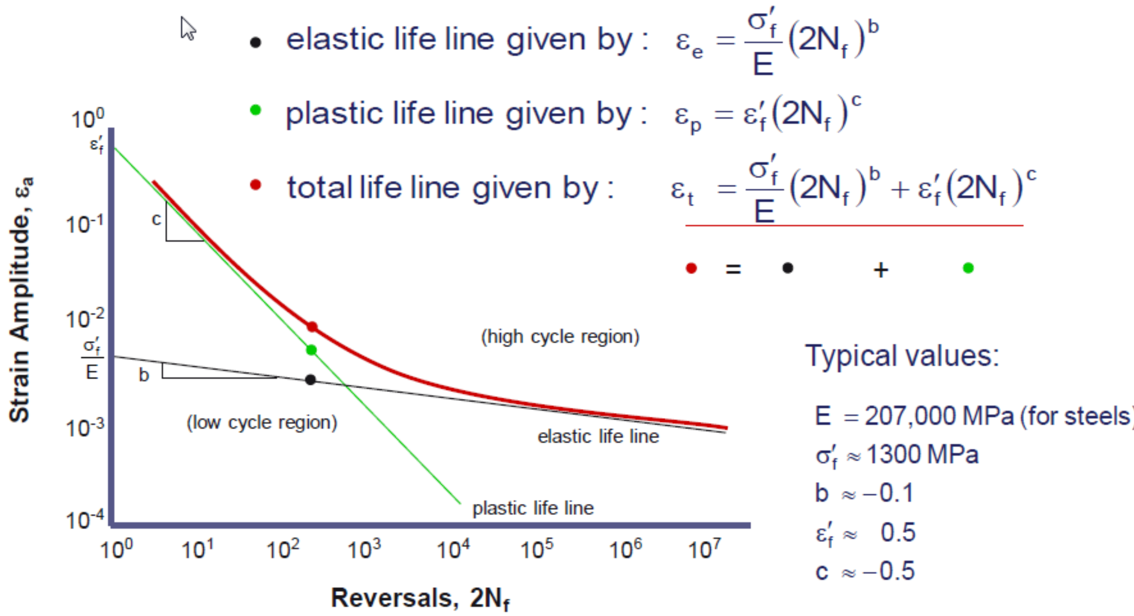
Strain-life Relationship

This is the Coffin-Manson-Basquin formula, defining the relationship between strain amplitude ε_a and the number of cycles to failure N_f:

$$\varepsilon_a = \frac{\sigma'_f}{E}(2N_f)^b + \varepsilon_f'(2N_f)^c$$

Graphically, it looks like this:

Fig. 3-13 Coffin-Manson-Basquin strain-life relationship



Fatigue damage is predicted in the same way as for the S-N method—the damage due to an individual cycle is calculated by looking up the strain amplitude of that cycle on the curve to find the number of reversals to failure 2Nf. The damage assigned to that cycle is then 1/Nf.

If the life N_f corresponding to a particular cycle is greater than the cutoff Nc, zero damage is predicted. Ne is an endurance limit and is used in an entirely different way than Nc. The main application for Ne is in the consideration of surface finish.

Surface finish and treatment corrections are detailed in the section "Defining and Using Surface Finish and Surface Treatment in Strain-Life".

The standard error parameters are used to determine the life associated with a particular design criterion (certainty of survival).

CertaintyOfSurvival	50	Specifies the certainty of survival based on material data scatter

Note that the value set for this property will have no effect if the standard error values in the materials data is set to zero.

The certainty of survival is converted to a standard deviation using the following values. Linear interpolation is used for values not in the table.

Table 3-1 Certainty of Survival Conversion to Standard Deviation

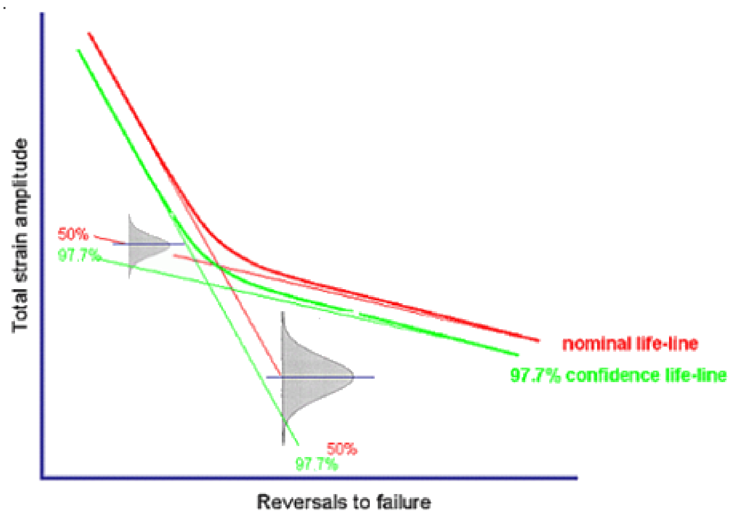
%Certainty of Survival	Standard Deviations from mean (Z)
99.9	-3
99.4	-2.5
97.7	-2
93	-1.5
84	-1
69	-0.5
50	0
31	0.5
16	1
7	1.5
2.3	2
0.6	2.5
0.1	3

The strain-life curve is then shifted downwards, as follows:

$$\varepsilon_t = \frac{\sigma'_f}{E}(2N_f)^b \bullet 10^{zs_e} + \varepsilon_f'(2N_f)^c \bullet 10^{zs_p}$$

where Z is the number of standard deviations from the mean and s_e and s_p are the standard errors of log elastic and log plastic strain respectively.

Fig. 3-14 Adjustment of strain-life curve to account for design criterion



Stress-Strain Relationship

The local strain approach also requires that the cyclic stress-strain behavior be modeled. This requires that we know the relationship between stress amplitude and strain amplitude (cyclic stress-strain curve), and how to predict the shape of hysteresis loops. The uniaxial cyclic stress-strain curve is modeled using a Ramberg-Osgood relationship.

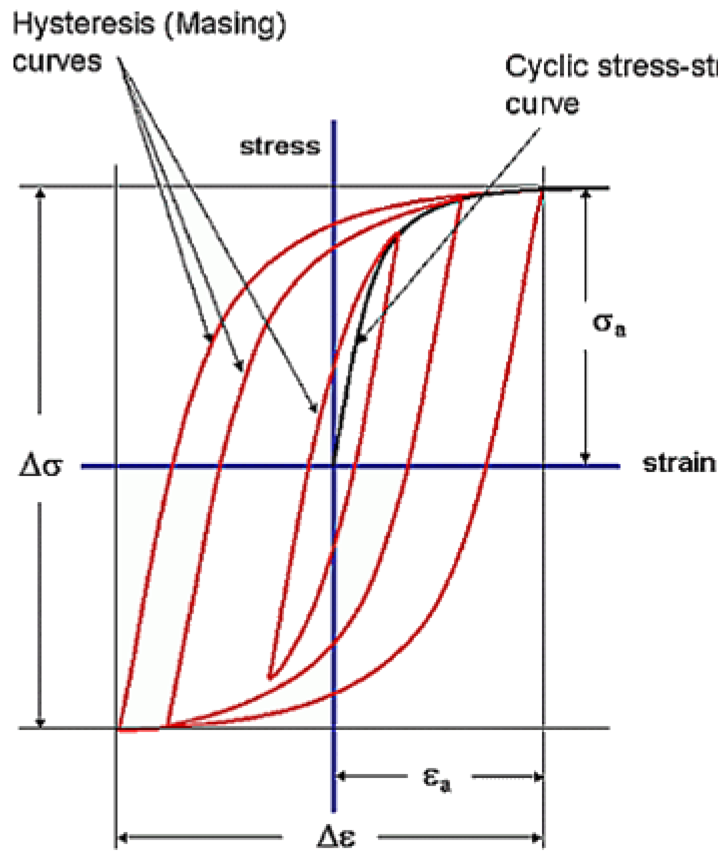
$$\varepsilon_a = \frac{\sigma_a}{E} + \left(\frac{\sigma_a}{K'}\right)^{\frac{1}{n'}}$$

Masing's hypothesis is used to predict the shape of each hysteresis loop by doubling the size of the cyclic stress-strain curve.

$$\Delta\varepsilon = \frac{\Delta\sigma}{E} + 2\left(\frac{\Delta\sigma}{2K'}\right)^{\frac{1}{n'}}$$

The relationship between the two curves is illustrated in Figure 3-15 below.

Fig. 3-15 Cyclic stress strain and hysteresis curves



If SEc > 0, then the design criterion (certainty of survival) will also adjust the stress strain behavior as follows, based on the number of standard errors from the mean:

$$\Delta\varepsilon = \frac{\Delta\sigma}{E} + 2\left(\frac{\Delta\sigma}{2K'}\right)^{\frac{1}{n'}} \cdot 10^{-z \cdot SEc}$$

The effects of this adjustment can be counter-intuitive, particularly when the Smith-Watson-Topper method is being used. If unsure, it is probably best to leave SEc set to 0.

The equation used for the stress-strain relationship applies for the standard Strain-Life data set and the multi-curve options but not for gray irons—these are detailed in "Gray Iron Material Behavior" on page 108.

3.2.2 Multi-curve Material Data

A few companies, notably in the aerospace industry, do not like to rely on the generic Morrow and SWT methods for modeling the effect of mean/max stress, but prefer to take mean effects into account by interpolation between curves obtained at different mean or load ratios. The next two options are for these companies.

Multi-curve Data—Multiple Mean

When MultiMeanCurve is selected, the E-N analysis engine expects the E-N data in the form of a family of E-N curves representing the fatigue strength of the material at different mean stress levels (nCode EN Mean Stress Curves). This type of data is found only in the MXD database (not supported in the nSoft MDB database). The data consists of a set of generic data with a number of child data sets storing the individual curves. The generic data consists of the following parameters:

Parameter Name	Description
MaterialType	Numeric code defining the type of material. The material type is required for correct application of surface finish corrections.
UTS	Ultimate tensile strength. This is required to apply the static failure criterion.
E	Modulus of elasticity
n’	Cyclic hardening exponent
K’	Cyclic hardening coefficient
me	Elastic Poisson ratio
SEc	Standard Error of Cyclic Strain
Ne	Endurance limit
Nfc	Fatigue cutoff
Strain Type	Range or Amplitude
Comments	
References	

Each child data set (E-N curve) defines a lookup curve of strain vs. life and has the following parameters:

Parameter Name	Description
MeanStress	Mean Stress for this strain-life curve
StrainValues	Comma-separated list of strain values

LifeValues	Comma-separated list of life values (in reversals)

These curves are normally used by interpolating between the curves for each cycle depending on its mean stress. The interpolation scheme is the same as for MultiMeanCurves in the S-N glyph. Please see the appropriate section for details.

Note that the effect of design criterion and surface condition are not currently supported for this data type.

3.2.3 MultiRRatioCurve

The handling of MultiRRatioCurve data is very similar to that of MultiMeanCurve data. See "Multiple Mean Curves" for more information.

When MultiRRatioCurve is selected, the E-N analysis engine expects the E-N data in the form of a family of E-N curves representing the fatigue strength of the material at different R-ratio levels (nCode EN R-ratio Curves). R-ratio is defined as the ratio min/max of a cycle. A fully reversed cycle therefore has R-ratio = -1. This type of data is found only in the MXD database (not supported in the nSoft MDB database). The data consists of a set of generic data with a number of child data sets storing the individual curves.

The generic data consists of the following parameters:

Parameter Name	Description
MaterialType	Numeric code defining the type of material. The material type is required for correct application of surface finish corrections.
UTS	Ultimate tensile strength. This is required to apply the static failure criterion.
E	Modulus of elasticity
n'	Cyclic hardening exponent
K'	Cyclic hardening coefficient
me	Elastic Poisson ratio
SEc	Standard Error of Cyclic Strain
Ne	Endurance limit
Nfc	Fatigue cutoff
StrainType	Range or Amplitude
Comments	
References	

Each child data set (E-N curve) defines a lookup curve of strain vs. life and has the following parameters:

Parameter Name	Description
R-Ratio	R-ratio for this strain-life curve
StrainValues	Comma-separated list of strain values
LifeValues	Comma-separated list of life values

These curves are normally used by interpolating between the curves for each cycle depending on its R-ratio stress.

Note that the effect of design criterion and surface condition are not currently supported for this data type.

3.2.4 Gray Iron Data

The gray iron data set consists of the following properties:

Parameter Name	Description
UTS	Ultimate tensile strength. This is required to apply the static failure criterion.
E0	Tangent modulus at zero
nt	Tensile strain hardening exponent
Kt	Tensile strain hardening coefficient
Nc	Compressive strain hardening exponent
Kc	Compressive strain hardening coefficient
mc	Compressive secant slope
mu	Unloading secant slope
Ks	Damage curve co-efficient
ns	Damage curve exponent
RC	Damage curve cutoff
SD	Standard error of log life
mt	Tensile secant slope
Comments	
References	

This data set is typically a child of a generic data set. See "Standard Parametric Data" on page 99.

3.2.5 Gray Iron Material Behavior

Gray irons have significantly different stress-strain behavior to most other alloys, due to the weakness of the graphite flakes in tension. In order to determine the properties, standard tests are performed at a variety of strain levels and life is noted.

The tensile cyclic stress strain curve is defined as

$$\varepsilon_t = \frac{\sigma}{E_0 + m_t\sigma} + \left(\frac{\sigma}{K_t}\right)^{1/n_T}$$

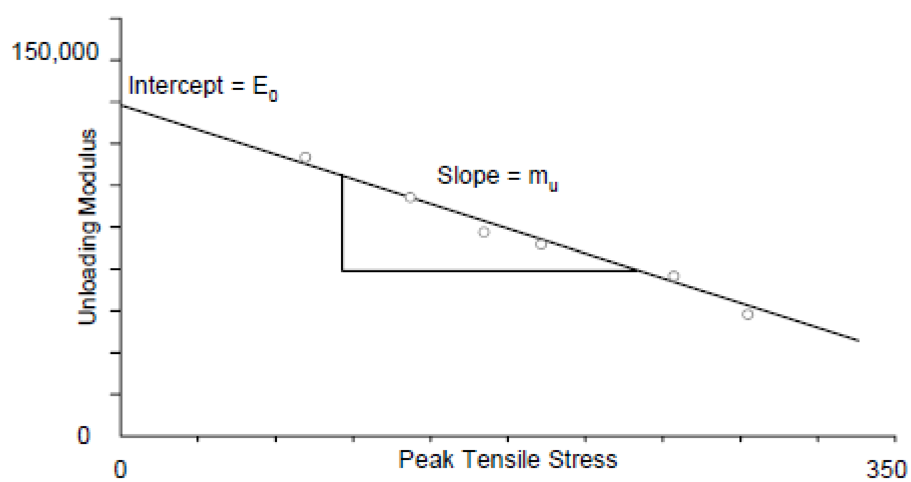
Total strain amplitude is the sum of the secant strain and the residual non-elastic strain.

The unloading modulus Eu is related to maximum tensile stress according to

$E_u = m_u S_{max} + E_0$

For each strain amplitude, plot unloading modulus vs. peak tensile stress. Linear regression will provide a slope, mu, and intercept Eo.

Fig. 3-16 Grey iron unloading modulus v peak tensile stress

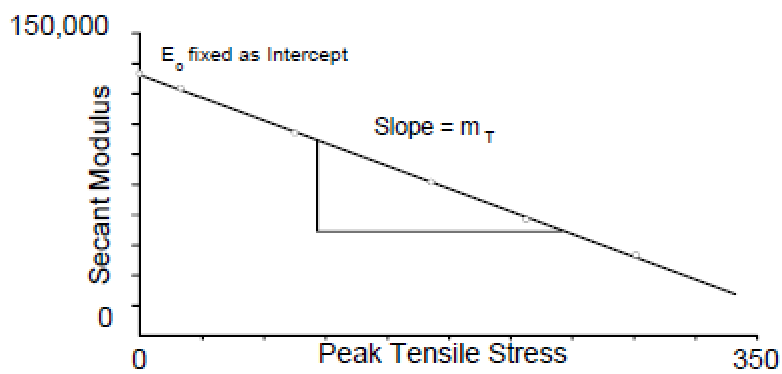


The secant modulus is related to maximum tensile stress according to

$$E_s = \frac{\sigma_{max}}{\varepsilon_a} = m_t\sigma_{max} + E_0$$

For each strain amplitude, divide the initial peak tensile stress by the strain amplitude to get the secant modulus and then plot these moduli against the peak stress. Regression analysis through the linear portion of the plot, including the value of E_o calculated above, will give a slope of m_t, the tensile secant slope.

Fig. 3-17 Secant modulus v peak tensile stress



Secant strain is calculated from

$$\varepsilon_s = \frac{\sigma}{(E_0 + m_t\sigma)}$$

and the remaining non-elastic strain from

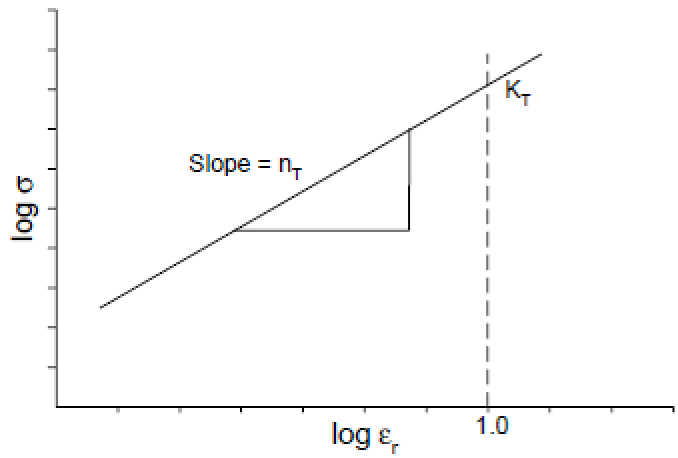
$$\varepsilon_T = \varepsilon_t - \varepsilon_s$$

K_t and n_t are related to residual non-elastic strain

$$\varepsilon_T = \left(\frac{\sigma}{K_t}\right)^{1/n_t}$$

Log-log regression of remaining non-elastic strain vs. peak stress will provide values for K_t and n_t.

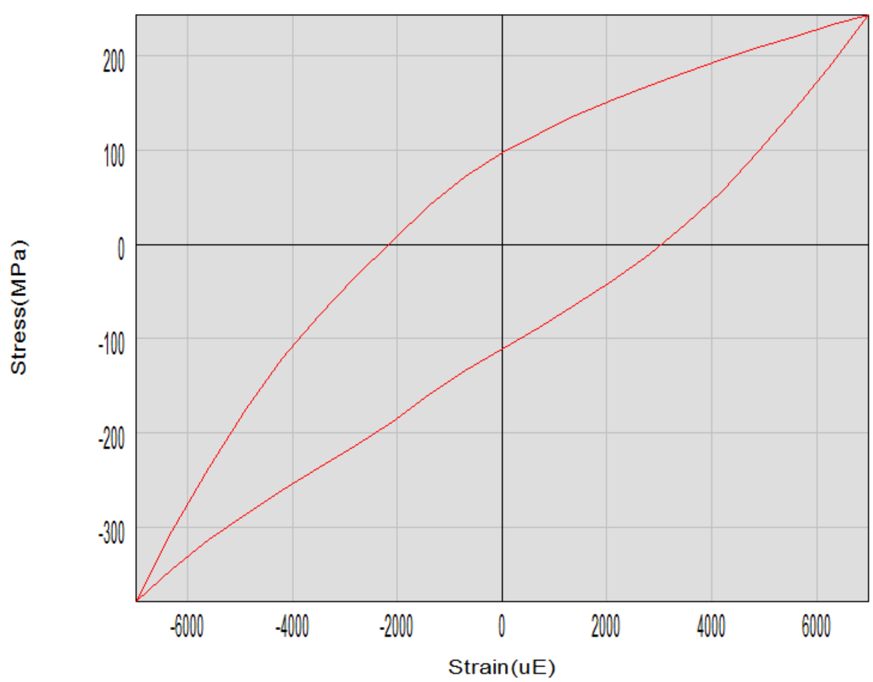
Fig. 3-18 Log-log regression of remaining non-elastic strain vs. peak stress



The compressive cyclic stress strain properties m_C, K_C, and n_C are determined in the same way as for the tensile equivalents.

A typical hysteresis loop for gray iron using these equations looks like this:

Fig. 3-19 Hysteresis loop for gray iron



Note that the curve shape changes in loading and unloading and in tension and compression.

The damage curve is determined by log-log regression of the SWT parameter against life to failure. The equation is

$$\sigma_{max}\varepsilon_a = A(N_f)^b$$

This gives the values of A (Ks in the data set) and b (ns in the data set).

3.2.6 Defining and Using Surface Finish and Surface Treatment in Strain-Life

Surface finish and treatment can have a significant effect on fatigue behavior. Rough surface finishes, e.g., due to machining marks, will in general reduce the fatigue strength, whereas surface treatments are often applied to increase the fatigue strength.

In GlyphWorks, surface finish and treatment effects are modeled in the E-N engine by means of a single Surface Factor Ksur. This works in a different way from the scale factor and the Kf, with which it should not be confused.

GlyphWorks offers pick lists for specific surface finish and treatment types.

SurfaceFinish	Polished	Sets the surface finish type
SurfaceTreatment	None	Sets the surface treatment type

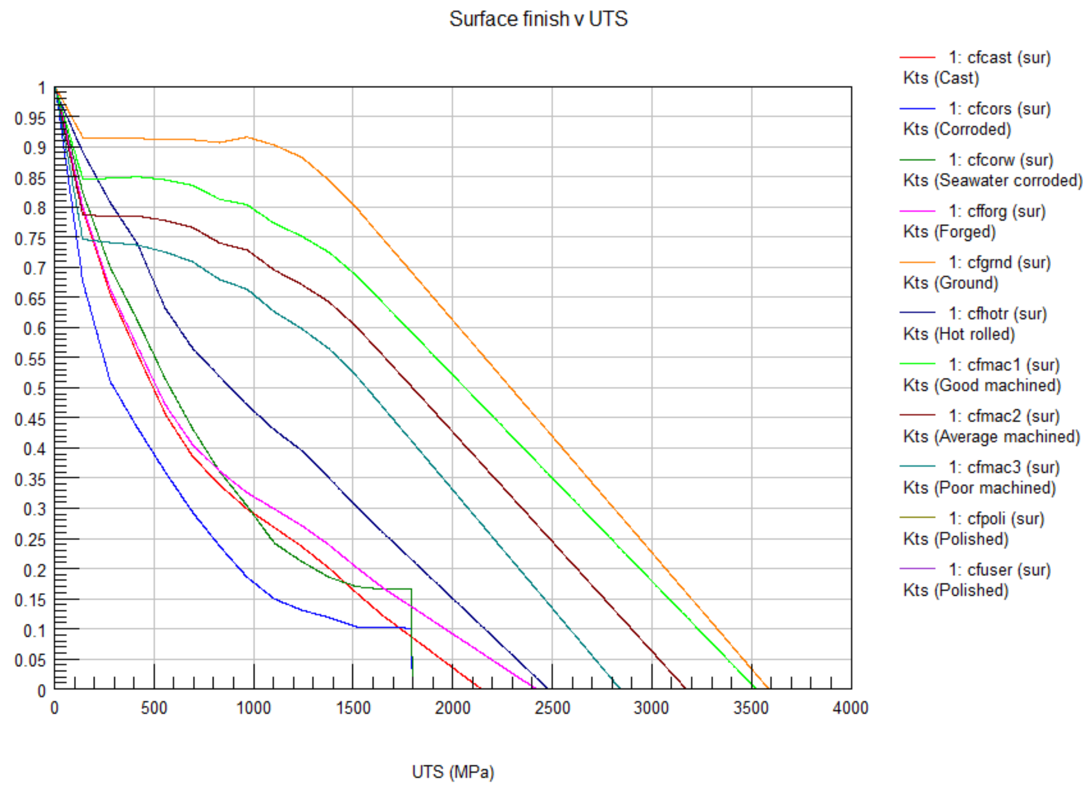
The default values, shown above, will both calculate a factor of 1. Ksur is calculated by multiplying the factors for finish and treatment together, so the net result of setting the default is a Ksur of 1, which leaves the materials data unchanged. Note that this assumes that the materials data is from a material in a polished, untreated condition. If the data comes from a material in any other condition, the values should be left as defaults.

The Surface Factor is used to adjust the material curve. The application is slightly different for the S-N and E-N methods, but the basic principal is the same—the surface factor is applied to the fatigue strength of the material in the high cycle (long-life) regime, but the effect is diminished in the low cycle (short-life) regime.

GlyphWorks offers a set of standard surface finish and treatment values that can be used to estimate the effect of surface changes. The surface finish values are obtained from a set of files in the materials directory of the GlyphWorks installation that are in the nCode DAC format, and are in the form of Ksur v UTS. The value of UTS is obtained for the material and the value of Ksur is then obtained by looking up the curve. One of the options is “User”—if the file `cfuser.sur` is replaced by the user's own data file, then this will be used for the User option.

The plot below shows all the surface finish curves together.

Fig. 3-20 Surface finish curves



Surface treatments are calculated using the following table, which also requires knowledge of the surface finish. If the surface finish is not one of those listed, the treatment has no effect (+0%).

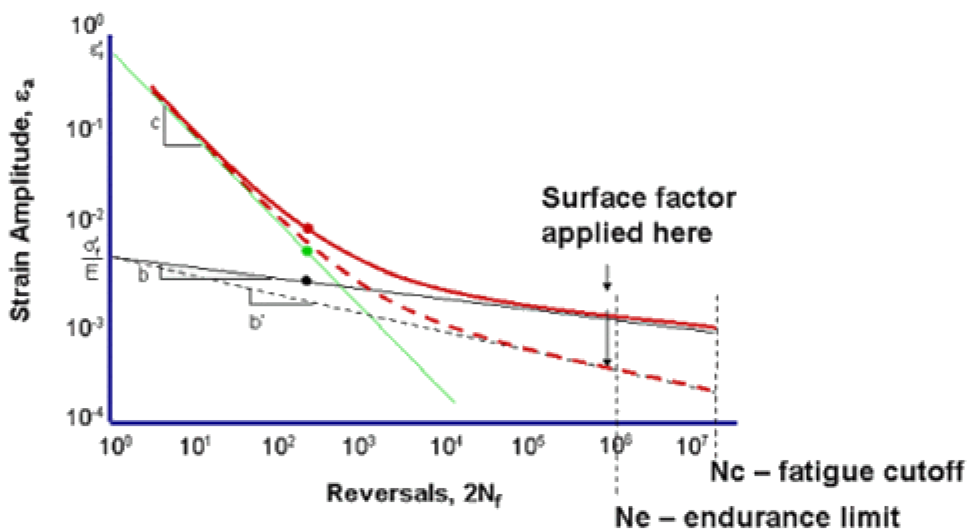
Finish	Shot Peened (%)	Cold Rolled (%)	Nitrided (%)
Polished	+15	+50	+100
Ground	+20	+0	+100
Machined	+30	+70	+100
Hot Rolled	+40	+0	+100
Cast	+40	+0	+100
Forged	+100	+0	+100

*Example calculation: Assuming a cast finish, where the value extracted from the surface finish *.SUR file happens to be 0.8, and the surface treatment is cast and* *shot peened (40% from the table for cast finish.) then GlyphWorks adds 40% of 0.8 (.32) to 0.8, and the composite correction factor is 1.12.*

These curves and the treatment data are based on limited data and were generated over 50 years ago. They are also valid only for steels. For more accurate information, we strongly recommend that specific material testing be performed using the actual alloy in its polished and as-manufactured surface condition. This will allow a more accurate value of Ksur to be determined, which can be used in the software. Alternatively, a separate material data set can be created for the raw and polished conditions.

The effect of surface condition in the StrainLife glyph is modeled by means of a surface factor Ksur which, in the case of a rough surface finish, reduces the fatigue strength. The derivation of surface factors is described in the "Defining and Using Surface Finish and Surface Treatment in Strain-Life" section. In the case of the E-N method, the effect of the surface condition is modeled by changing the slope of the elastic part of the strain-life curve so that the fatigue strength at the endurance limit Ne is reduced by the surface factor.

Fig. 3-21 Application of surface factors in the local strain approach



If the surface factor is denoted K_{sur}, the modified slope b' is calculated from:

$$b' = b - \frac{\log(K_{sur})}{\log(Ne)}$$

Note that if Ne is not defined, the fatigue cut-off Nc will be used instead.

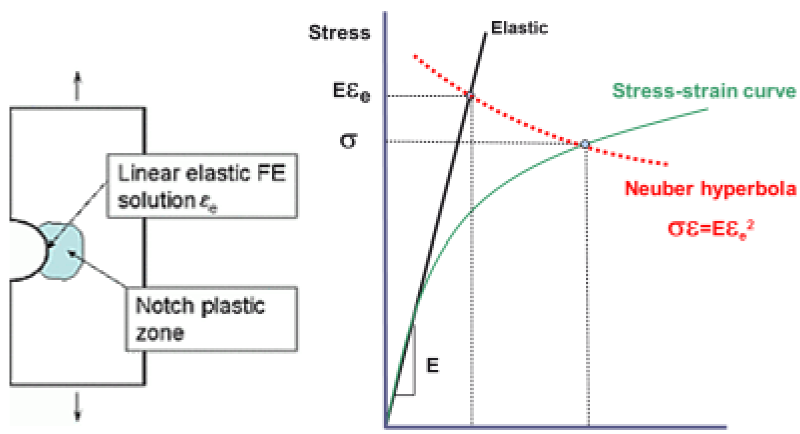
3.3 Notch Correction—Neuber's Rule

The Neuber method provides a simple way of estimating the total elastic-plastic strain and stress at an “average” stress concentration, based on the local elastic stress/strain. Consider a simple notched specimen subjected to a uniaxial loading. Because of its formulation, it can be used effectively for strains calculated using FE solutions, or as a way of correcting elastic-plastic strains for notches using the fatigue concentration factor Kf. See Figure 3-22.

As long as the yield stress is not exceeded, elastic FE analysis gives (assuming a good model) a reasonably accurate estimate of the strain and stress at the root of the notch. Similarly, using measured strains and a fatigue concentration factor that has been verified for the particular local geometry also gives a reasonable estimate.

However, once the yield stress is exceeded, a purely elastic solution becomes increasingly unrealistic. In practice, as yielding occurs, there will be a redistribution of stress and strain around the notch, so that the real strain will be greater than the elastic value and the real stress less than the value from elastic analysis. The true solution must lie somewhere on the material stress-strain curve. To get a reasonably accurate estimate of the way this stress and strain is redistributed, we could carry out an elastic-plastic FE solution, taking into account the geometry of the specimen, but this could be rather time consuming, especially if we have to consider many loading cycles.

Fig. 3-22 Neuber method: estimate elastic plastic strain and stress at notch



The Neuber method provides a simple alternative that provides a rough estimate for how the stress and strain might redistribute, without reference to the real geometry. The Neuber method assumes that the product of stress and strain before and after redistribution is constant. This is represented by the Neuber hyperbola, where the product of stress and strain is constant and equal to the elastic stress x elastic strain.

The Neuber method can be applied to monotonic or cyclic loading. In Glyph-Works, we apply it to cyclic loading, in two ways:

We position the outside hysteresis loop by applying Neuber to the Abs Max value of the elastic strain in the strain history, together with the cyclic stress-strain curve. When the input is linear elastic, this is achieved by solving the pair of equations

$$E\left(\varepsilon_{e,AbsMax}\right)^2 = \sigma_{\max}\varepsilon_{\max}$$

$$\varepsilon_{\max} = \frac{\sigma_{\max}}{E} + \left(\frac{\sigma_{\max}}{K'}\right)^{\frac{1}{n'}}$$

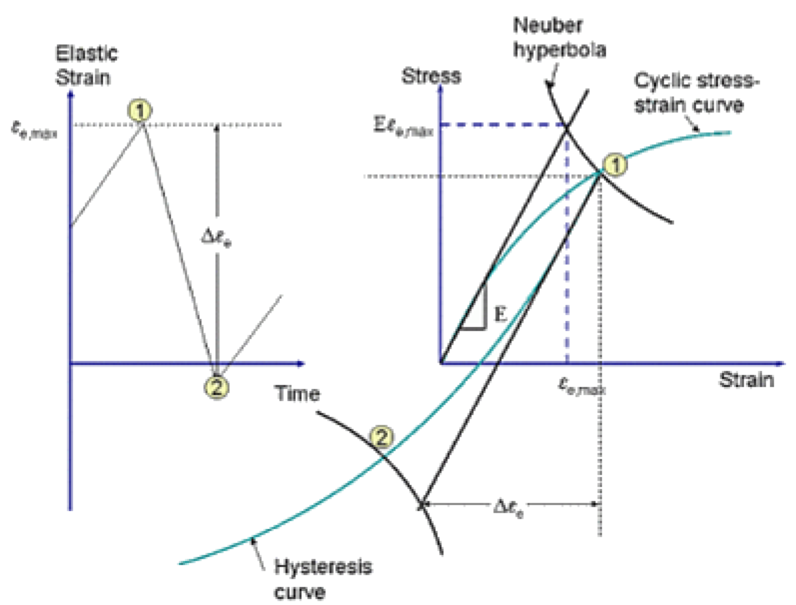
To calculate a subsequent strain excursion, by applying Neuber to the elastic strain range of the excursion, together with the hysteresis curve, i.e., by solving the pair of equations

$$E\left(\Delta\varepsilon_e\right)^2 = \Delta\sigma\Delta\varepsilon$$

$$\Delta\varepsilon = \frac{\Delta\sigma}{E} + 2\left(\frac{\Delta\sigma}{2K'}\right)^{\frac{1}{n'}}$$

These two ways of using Neuber are illustrated in Figure 3-23. First, the position of the Absolute Maximum value is estimated using the Neuber method at Point 1. All hysteresis loops are positioned relative to this point. The next strain excursion, to Point 2, is then corrected by applying the Neuber correction to the range of the excursion, using the hysteresis curve.

Fig. 3-23 Application of Neuber correction to cyclic loading



If the data is measured, it is often already past the yield point on the cyclic stress-strain curve, even when measured away from the notch root. Neuber's formulation, however, means that the same process can be applied, but this time the fatigue concentration factor, Kf, can be used to convert nominal strains to local strains. The equation

$K_f^2 e_n.s_n = e_l.s_l$

can be used for the outside loop and

$K_f^2 \Delta e_n.\Delta s_n = \Delta e_l.\ \Delta s_l$

for subsequent excursions.

Note that the Neuber method as implemented in GlyphWorks assumes that the strain is under uniaxial loading conditions. In reality, the stress state is sometimes not uniaxial, but to solve this requires more complex modelling that is not available in the basic StrainLife glyph.

If the data comes from a strain gauge rosette, the level of biaxiality can be assessed using the Strain Rosette glyph, which can also used to derive a single strain channel that can be used in a uniaxial solver - this is often the absolute maximum principal strain.

3.3.1 The “Online” Method—Tracking from the Beginning

In the advanced properties the user can set whether the stress tracking starts at the absolute maximum value of the data or at the beginning. This is applicable only for a time history loading input and cannot be used with gray iron data.

StartofStressTracking	AbsMax	Determines the start point of stress tracking for standard E-N calculation

The standard method of rainflow counting within the StrainLife glyph, and the stress-strain tracking that goes with it, relies on starting at the absolute maximum of the data, and this is the default setting.

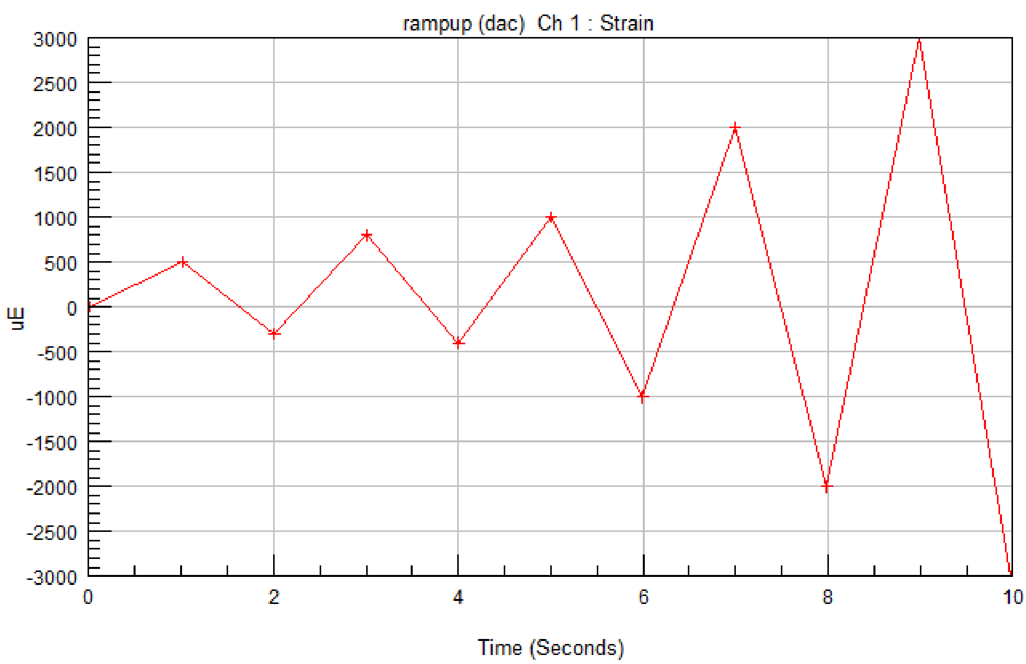
This method assumes that the data will be repeated many times before failure, and that any differences in the reversals and cycles that are counted before the absolute maximum is encountered in the first repeat will be insignificant compared to the many other times those cycles will be counted in the second and subsequent repeats.

However, if the time history describes the whole of the life of a component, and a more accurate representation of a single pass of the time history is required, GlyphWorks allows the tracking to start at the beginning of the time history. If the property StartOfStressTracking is set to “First”, the glyph will perform this alternative counting and tracking algorithm.

The main difference in this algorithm, which still uses the cyclic and twice cyclic stress-strain relationships, is that reversals can be counted. These are “half-cycles” and the damage allocated to them is half the damage of a full cycle. In addition, the glyph will report a life value, but the life value assumes that the damage is repeated and in this case we are assuming a non-repeating sequence so the life should be used for information purposes only.

How this algorithm works is best seen using a time series that ramps upwards; for example, like this:

Fig. 3-24 Ramping time series



When the cycle counting starts at the absolute maximum, the following cycles are counted:

1	2	12	13
Maximum nominal	Minimum nominal s	Number of cycles	Damage
uE	uE		
500	-300	1	0
800	-400	1	0
1000	-1000	1	0
2000	-2000	1	1.196e-05
3000	-3000	1	7.617e-05

However, when counting from the first point, we get all reversals, which are shown in the cycles list output as “Number Of Cycles = 0.5". The number of reversals is also shown in the metadata output (NumReversalsCounted).

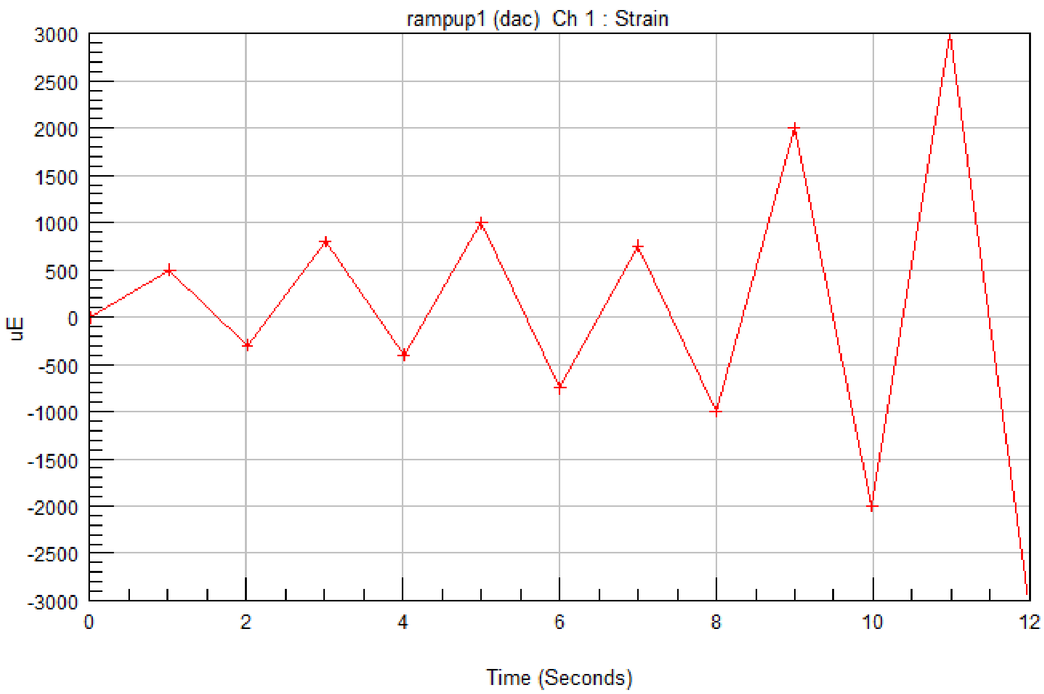
1	2	12	13
Maximum nominal	Minimum nominal s	Number of cycles	Damage
uE	uE		
500	0	0.5	0
500	-300	0.5	0
800	-300	0.5	0
800	-400	0.5	0
1000	-400	0.5	0
1000	-1000	0.5	0
2000	-1000	0.5	1.016e-06
2000	-2000	0.5	5.948e-06
3000	-2000	0.5	1.767e-05
3000	-3000	0.5	3.785e-05

The counting algorithm, when starting from the first point, takes four consecutive points and tries to close a cycle, at the same time tracking stress and strain hysteresis loops.

If the second and third points lie within the first and fourth points, they form a cycle that can be closed. In this example, the third point lies outside the first and fourth. At this point, the algorithm counts points 1 and 2 as a reversal (half cycle) and removes them, and the stresses and strains they mapped in the hysteresis loops, and starts again using point 3 as the first point. Because the signal ramps up, this counting sequence continues until the end of the data, as at no point do the conditions apply to close a cycle.

In a less extreme example, cycles and reversals can be counted together.

Fig. 3-25 3rd point lying outside the 1st and 4th

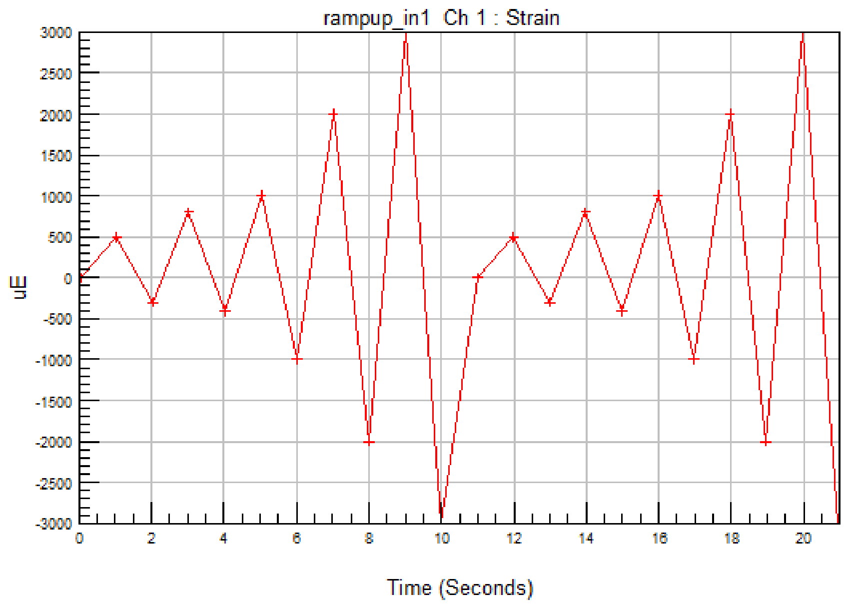


In this case, a cycle has been inserted at the 7th and 8th points and this appears in the cycle list:

1	2	12	13
Maximum nominal	Minimum nominal s	Number of cycles	Damage
uE	uE		
500	0	0.5	0
500	-300	0.5	0
800	-300	0.5	0
800	-400	0.5	0
1000	-400	0.5	0
750	-750	1	0
1000	-1000	0.5	0
2000	-1000	0.5	1.016e-06
2000	-2000	0.5	5.948e-06
3000	-2000	0.5	1.767e-05
3000	-3000	0.5	3.785e-05

Note that at the end of counting, any reversals that are in the rainflow buffer are not closed down. If we take the first example and repeat it once, it looks like this:

Fig. 3-26 3rd point lying outside the 1st and 4th



The results of counting are as follows:

1	2	12	13
Maximum nominal	Minimum nominal s	Number of cycles	Damage
uE	uE		
500	0	0.5	0
500	-300	0.5	0
800	-300	0.5	0
800	-400	0.5	0
1000	-400	0.5	0
1000	-1000	0.5	0
2000	-1000	0.5	1.016e-06
2000	-2000	0.5	5.948e-06
3000	-2000	0.5	1.767e-05
500	-300	1	0
800	-400	1	0
1000	-1000	1	0
2000	-2000	1	1.19e-05
3000	-3000	1	7.569e-05
3000	-3000	0.5	3.785e-05

The smaller excursions now lie within the "outer loop" of 3000 to -3000 and can therefore be closed as cycles on the second pass.

Having counted a 3000 to -3000 cycle when the 2nd 3000 value appears, the rainflow counting buffer will have a 3000 and -3000 left in there. This is treated as a final reversal.

If the signal is re-ordered to start and finish at the absolute maximum value, the cycle count will be identical. However, there may be small differences in the stresses and strains calculated as the algorithm incrementally steps along the stress-strain curves rather than being able to jump in one step to the next turning point, as it can when starting at the absolute maximum.

It is not common for the difference between the two cycle counting methods to be significant, but in the case where the time history represents the whole life of the component, the option "StartofStressTracking=First" may be worth considering.

3.4 Calculating Damage

3.4.1 Damage Summation—Miner's Rule

A strain-based, constant amplitude, fatigue damage curve represents a set of tests at constant strain amplitude (ε_q) or SWT parameter ($\sigma_{max}\ \varepsilon_a$) together with associated lives. See "Standard Parametric Data" on page 99 for more details.

Operation at a strain amplitude ε_1 will result in failure in, say, N_1 cycles.

Operation at the same strain amplitude for a number of cycles less than N_1, N_j say, will result in a smaller fraction of damage, d_j, which is often referred to as a partial damage. Operation over a spectrum of different strain levels results in a partial damage contribution d_i from each strain cycle, ε_1. Failure is then predicted when the sum of these partial damage fractions reaches unity so that:

$d_1 + d_2 + ... + di_{-1} + d_i = 1$

The Palmgren-Miner rule (or Miner's Rule) asserts that the partial damage at any strain amplitude ε_i , or SWT level, is linearly proportional to the ratio of number of cycles of operation, ni, to the total number of cycles that would produce failure at that strain level, Ni, i.e.:

$di = n\ i\ /\ N\ i$

Failure is predicted then if:

$n_1 / N_1 + n_2 / N2 + ... + n_{i-1} / N_{i-1} + n_{\ i} / N_{\ i} >= 1$

The above equations represent a statement of the linear damage rules used by all the GlyphWorks fatigue analyzers. Experience shows that linear damage summation is somewhat of an oversimplification of reality. The most important shortcoming is that no account is made of the sequence in which strain levels are experienced, and damage is assumed to accumulate at the same rate for a given strain level regardless of prehistory. In particular, it appears that large strain amplitudes that precede smaller ones cause the smaller cycles to become more damaging than expected and vice versa. The net result is that in the first instance, the Miners' sum is measured to be less than 1, and in the latter case it becomes greater than 1. Since most service environments involve quasi-random loading sequences, the use of the Palmgren-Miner linear damage rule summing to a constant of 1 is mostly satisfactory.

The cycles list generated from any of the input types has to contain the following information for each cycle to be calculated:

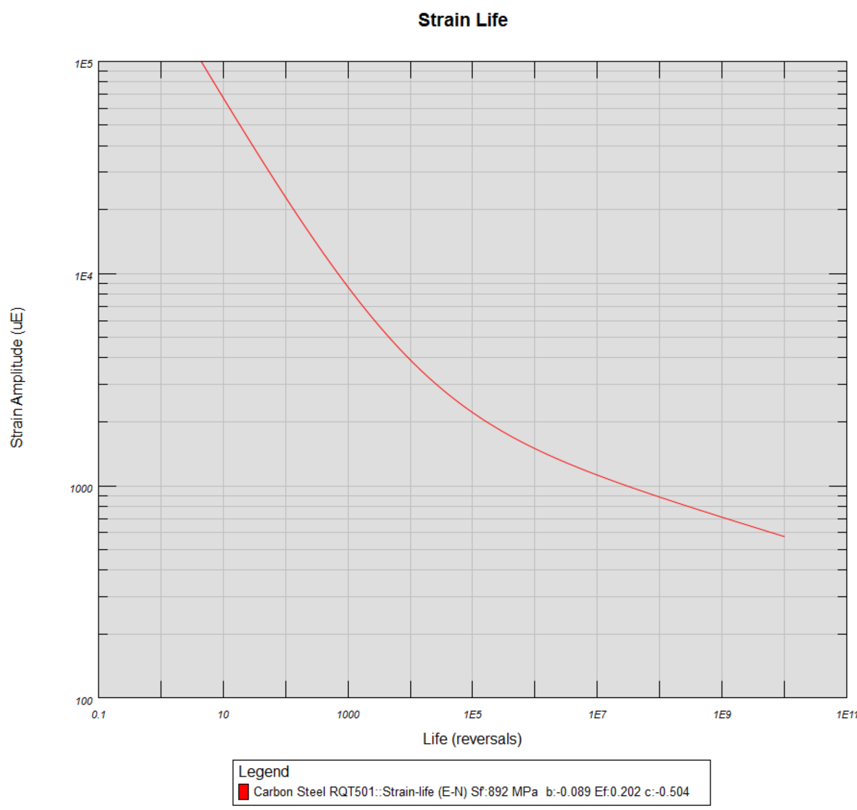
- Max strain
- Min strain
- Max stress
- Min stress
- Number of cycles

This is sufficient information to calculate the damage for a cycle of this size and position in stress-strain space. It has been shown experimentally that damage is not only determined by the range of the cycle, but also the mean (or max) stress level associated with the cycle. A higher positive mean stress is generally more damaging. The damage calculation either includes this mean stress effect directly or as a correction. For each of the different material types, in combination with mean stress correction, there are slightly different methods for the damage of the cycle to be calculated.

3.4.2 Standard Parametric Data Set

An example of the curve follows:

Fig. 3-27 Carbon steel parametric data curve



This curve is created using the equation

$$\varepsilon a = \frac{\sigma_f{}'}{E}(2N_f)^b + \varepsilon_f{}'(2N_f)^c$$

2Nf is the life in reversals (i.e., half cycles) and if no mean stress correction is applied it is simply a matter of calculating the strain amplitude (which is (max-min)/2) and then using a Newton-Ralphson iterative solution to get a value of 2Nf. The damage for each cycle is

d = 1/Nf

Any cycle for which the resulting life is beyond the material cutoff will have its damage set to zero.

This damage is simply multiplied by the number of cycles and added to the total damage. When all the damage is calculated and summed, the life can be calculated (see "Damage Summation—Miner's Rule" on page 121).

If the Morrow mean stress correction is to be applied, the mean stress for each cycle is calculated

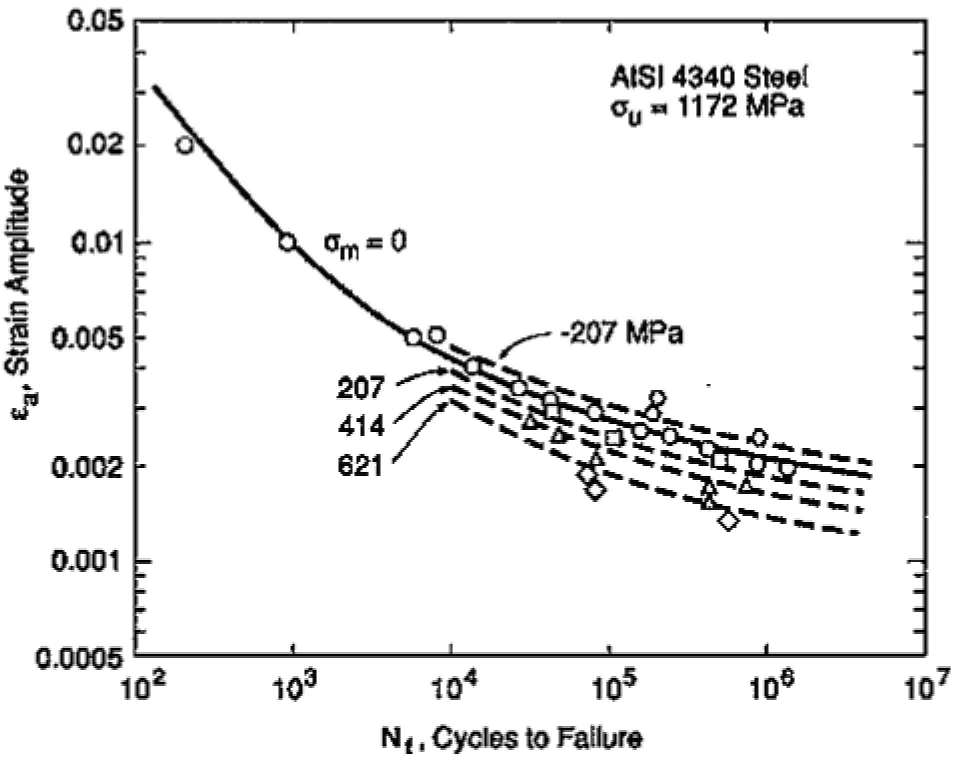
$\sigma_m = (\sigma_{max}+\sigma_{min})/2$

The Morrow mean stress correction adjusts the value of the intercept of the elastic part of the strain life curve before looking up the life/damage, on a cycle-by-cycle basis as follows:

$$\varepsilon_a = \frac{(\sigma_f' - \sigma_m)}{E}(2N_f)^b + \varepsilon_f'(2N_f)^c$$

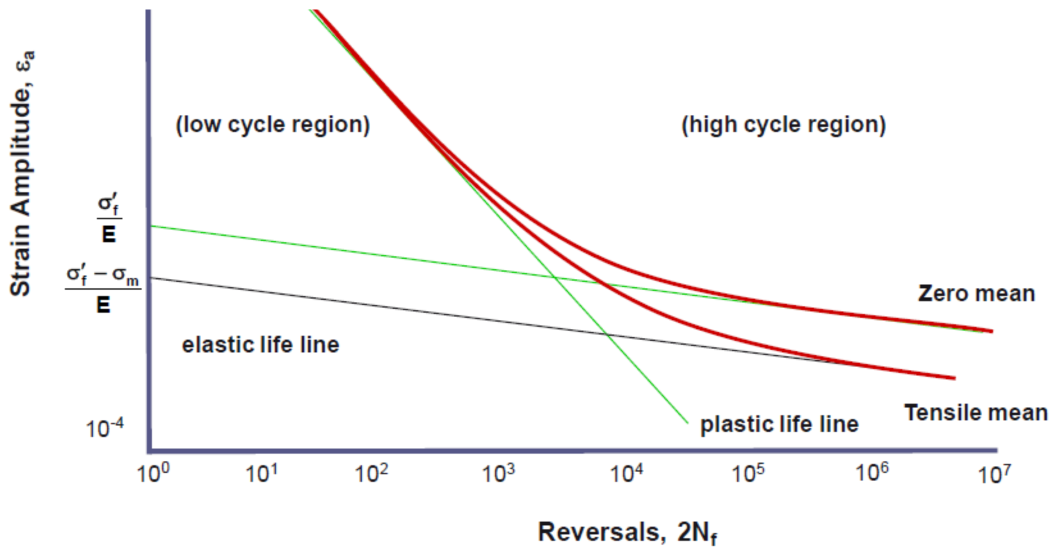
where σ_m is the mean stress of each cycle. The effect of this mean stress correction, correlated with test data, is illustrated in Figure 3-28:

Fig. 3-28 Morrow mean stress correction



This affects the strain-life curve as shown in Figure 3-29:

Fig. 3-29 Morrow equation applied to strain-life curve



Smith Watson-Topper (SWT) requires the maximum stress value only.

While loosely termed a mean stress correction method, the SWT method actually defines a new damage parameter based on the product of the strain amplitude and the maximum stress of each cycle.

$$P_{SWT} = \varepsilon_a \sigma_{\max}$$

In practice, there are two ways this can be applied. These are set using the SWT-Method property.

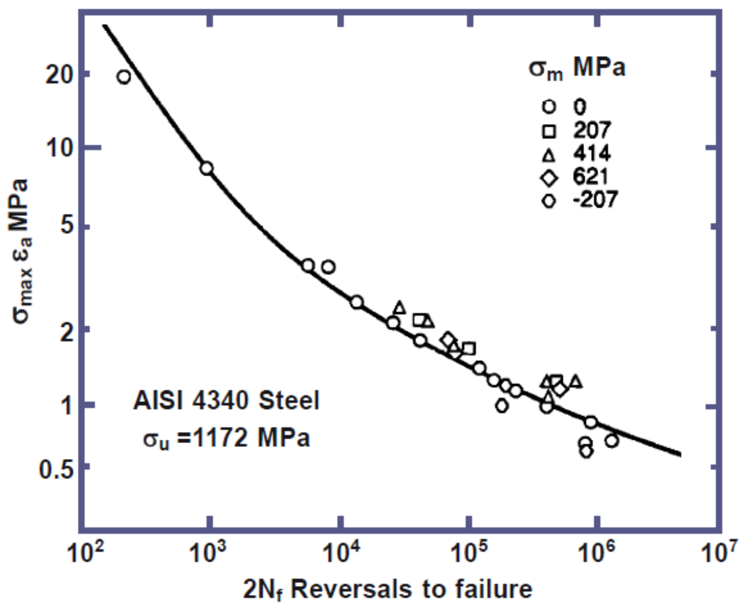
SWTMethod	Iterative	The Smith-Watson-Topper calculation method

In the Formula method, the following equation is solved for life/damage.

$$\varepsilon_a \sigma_{\max} = \frac{(\sigma'_f)^2}{E}(2N_f)^{2b} + \varepsilon_f{}'\sigma'_f(2N_f)^{b+c}$$

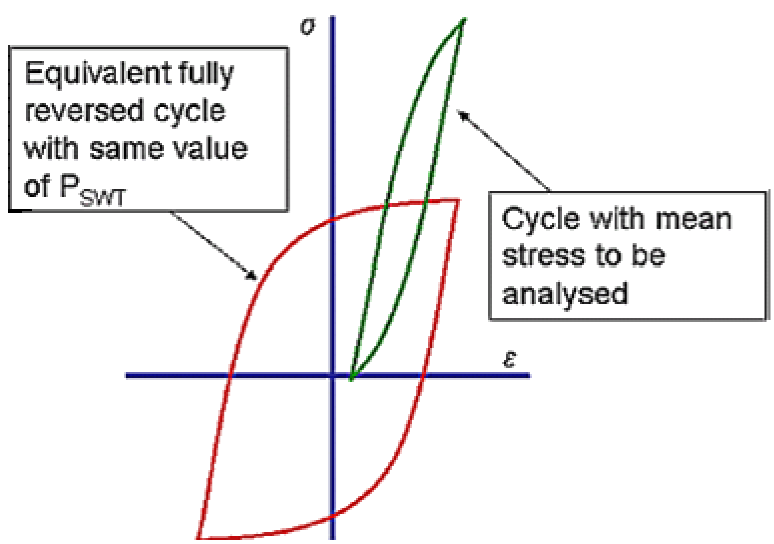
This formula works only for cycles whose maximum is greater than zero. If the cycle is fully compressive, there is no damage using this model.

Fig. 3-30 Smith-Watson-Topper mean stress correction



In the Iterative method, a fully reversed cycle that has the same value of P_{SWT} as the cycle being analyzed is sought.

Fig. 3-31 Iterative Smith-Watson-Topper method



The strain amplitude of the fully reversed cycle defines an equivalent strain amplitude that can be looked up on the standard strain-life curve. Because the strain amplitude and max stress of the equivalent fully reversed cycle are related by the cyclic stress-strain curve, the equations to be solved are:

$$P_{SWT} = \varepsilon_a \sigma_{\max} = \varepsilon_{a,equiv} \sigma_{\max,equiv}$$

$$\varepsilon_{a,equiv} = \frac{\sigma_{\max,equiv}}{E} + \left(\frac{\sigma_{\max,equiv}}{K'}\right)^{\frac{1}{n'}}$$

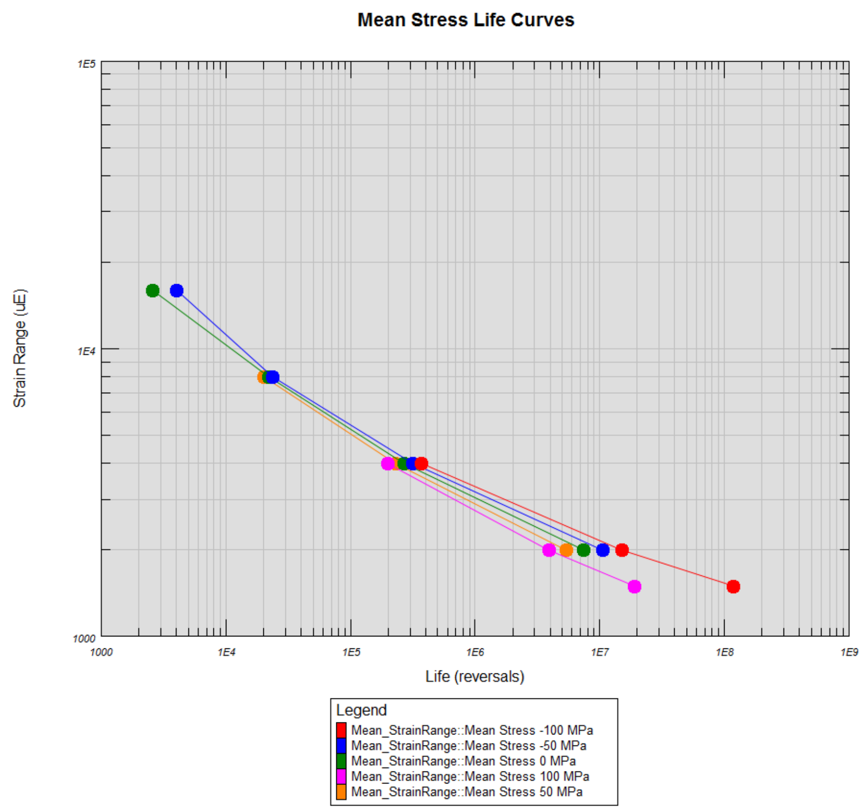
The equivalent stress amplitude is then looked up on the standard strain-life curve as normal.

Generally, the Iterative method is considered a more correct approach.

3.4.3 Multiple Mean Curves

An example of this kind of data is shown below:

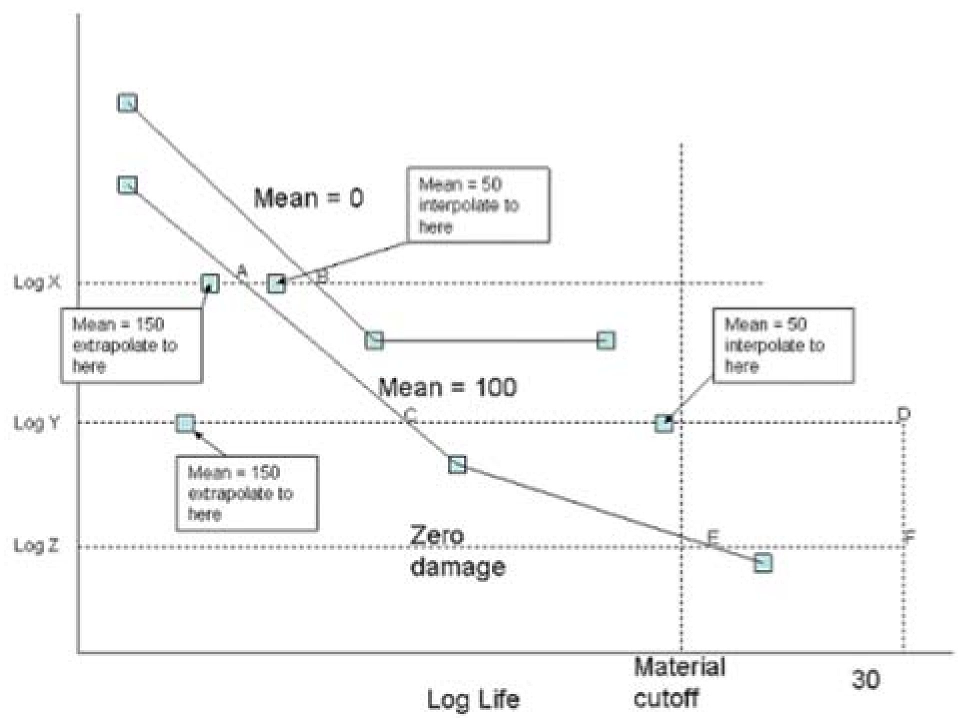
Fig. 3-32 Multiple mean curves



The MeanStressCorrection property for this type of data is usually set to Interpolate: the interpolation is based on mean stress.

The normal way to use such a curve set is to interpolate between the curves to determine the life and corresponding damage for each cycle. This process is illustrated in the following diagram:

Fig. 3-33 Interpolation scheme for multiple mean data



The material behavior is described in this case by two S-N curves representing the material at mean stress levels of 0 and 100.

Consider a cycle with Range X. First we must identify which two curves to use for interpolation by finding the pair with mean values either side of that of our cycle.

If the cycle has a mean of 50, this lies between our two curves. In this case, we look up the log(Life) values on the two curves corresponding to LogX, and locate points A and B. If necessary, we extrapolate the curves beyond their end points. We then linearly interpolate between these two log(Life) values based on the mean stress of our cycle to determine the log(Life) of our cycle.

If the cycle has a mean less than or equal to the minimum mean of all the curves (in this case 0), the life is determined using the curve with the minimum mean value. The life corresponds to Point B.

If the cycle has a mean greater than the maximum mean in the curve set, in this case 100, either the two curves with the highest mean values can be used to determine the log(life) by extrapolation, or the curve with the highest mean may be used, i.e., in this case the life corresponds to point A. This is controlled by the InterpolationLimit property.

Consider a cycle with range Y. This corresponds to point C on the Mean = 100 curve, but it does not intersect the mean = 0 curve, or that intersection point is beyond the Fatigue Cutoff Life. In this case, the second point D used for interpolation or extrapolation is set to 30 (corresponding to life of 1E30).

Consider now a cycle with Range Z. This cycle does not intersect any of the S-N curves at a value less than the material cutoff, so the damage value will be set to zero.

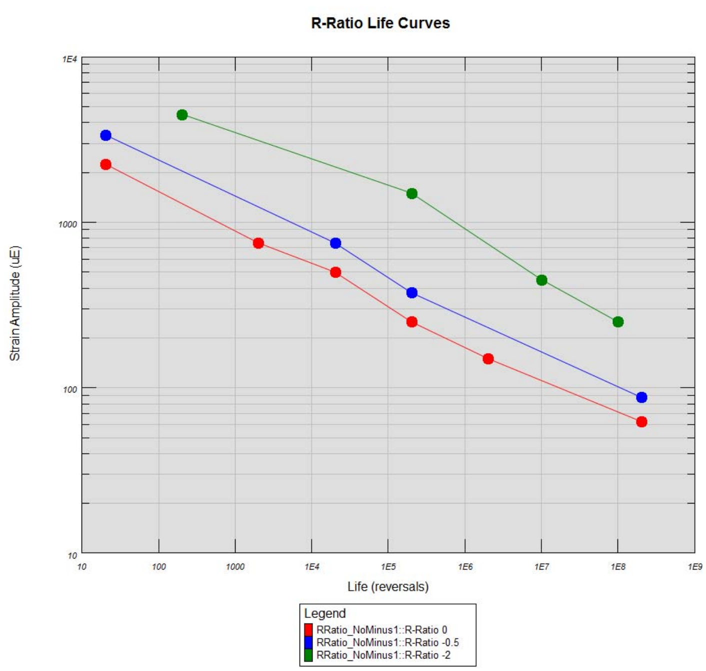
Any cycle for which the resulting life is beyond the material cutoff will have its damage set to zero.

The InterpolationLimit property applies only when the ENMethod is set to MultiMeanCurve or MultiRRatioCurve. It controls whether or not extrapolation will be carried out if the mean or R-ratio of a cycle lies outside the range defined by the curve set.

3.4.4 Multiple R-ratio Curves

The R-ratio is defined as the cycle minimum divided by the cycle maximum. An example of this type of data is shown below:

Fig. 3-34 R-ratio Life Curves



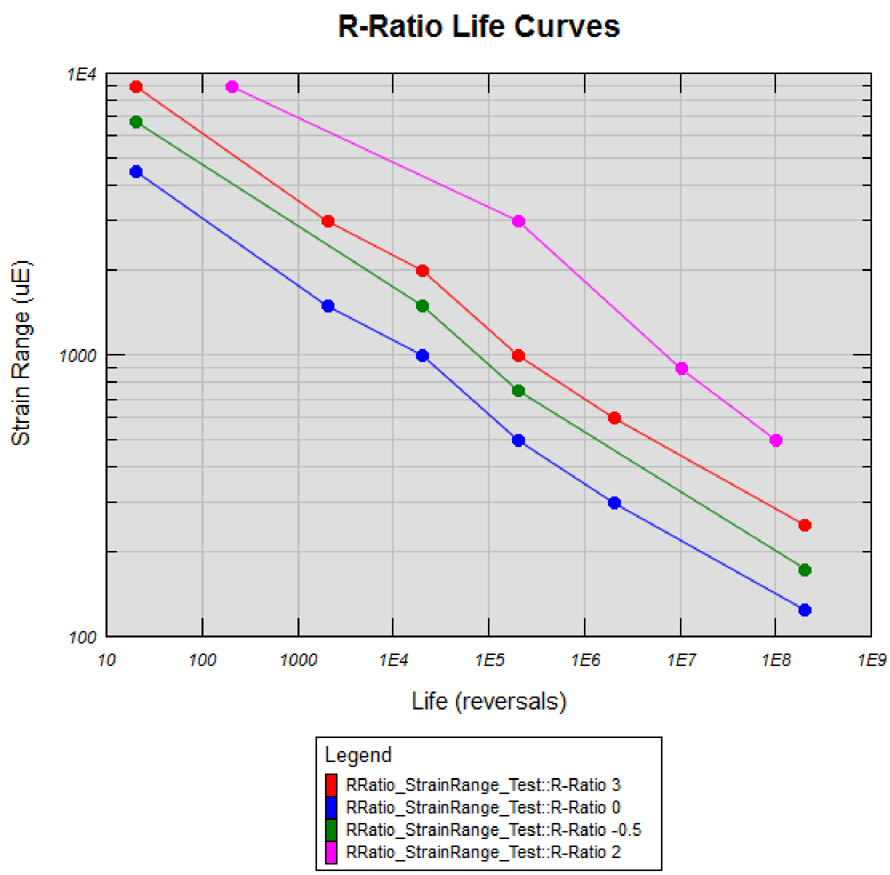
One issue with the extrapolation of R-ratio curves occurs because of the discontinuity that occurs when R=1 (R=1 is not possible as min cannot equal max in a cycle). R-ratios greater than 1 have negative mean value; R-ratios between -infinity and 1 have positive means, as is shown in the table below for a cycle of range 6 with a gradually reducing mean.

Max	**Min**	**R-Ratio**
10	4	0.40
9	3	0.33
8	2	0.25
7	1	0.14
6	0	0.00
5	-1	-0.20
4	-2	-0.50
3	-3	-1.00
2	-4	-2.00
1	-5	-5.00
0	-6	-infinity
-1	-7	7.00
-2	-8	4.00
-3	-9	3.00
-4	-10	2.50
-5	-11	2.20
-6	-12	2.00
-7	-13	1.86
-8	-14	1.75
-9	-15	1.67
-10	-16	1.60
-11	-17	1.55
-12	-18	1.50

Therefore, if a family of curves has some curves with R>1 and some with R<1, then they are split into two families of curves and a cycle can only be interpolated or extrapolated amongst the sub-family of curves.

This is illustrated by the data set displayed graphically below:

Fig. 3-35 R-Ratio Life Curves

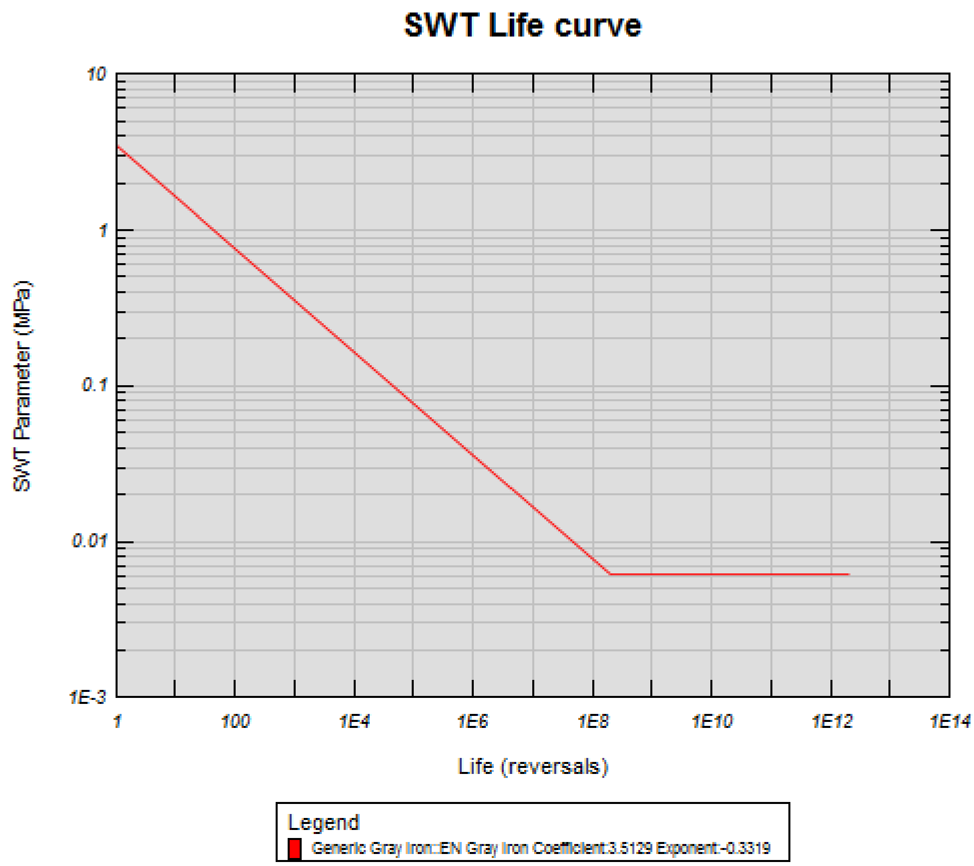


The curves give longer lives for the same strain range as the curves go from left to right. However, the R-ratio goes in the sequence 0, -0.5, 3, and 2. Because the R-ratios are not monotonically sequenced, it is unsafe to interpolate between them, so a separation is made at R=1.

3.4.5 Gray Iron

The gray iron solver uses a very simple linear Smith-Watson-Topper relationship to calculate damage per cycle.

Fig. 3-36 SWT curve for gray iron



The equation is

$$SWT = K_s.(2Nf)^{Ns}$$

The properties Ks and Ns can be found in the materials data set for gray iron, as can the cut-off value (in cycles). The cut-off defines the life beyond which testing shows insignificant or no damage, and is shown by the flat part of the curve in the plot. See Figure 3-36.

A standard error (based on log life) can also be provided and the SWT equation is modified using this value (SD) and the number of standard deviations from the mean Z (calculated from the certainty of survival—see Table 3-1 Certainty of Survival Conversion to Standard Deviation.

$$SWT = 10^{Z.SD}.K_s.(2Nf)^{Ns}$$

3.4.6 “Back” Calculations

When the glyph's calculation mode is set to ScaleFactor or Kf, the software uses an iterative calculation to determine the value of the scale factor or Kf that is required to achieve a specified target life. The two properties that are exposed, and have to be set, to achieve this calculation are:

BackCalculation		
TargetLife	1	Target life for back calculations
BackCalculationAccuracy	1	Specifies the percentage tolerance for back calculations

The target life is specified in equivalent units, which are set as part of the loading information. This normally defaults to “Repeats” (i.e., repeats of the input loading history/histogram/multi-column data) and is used to specify the life result.

See “Interpreting the Results Generated by the StrainLife Glyph” below for more details on how life is calculated and reported.

The back calculation accuracy, *a*, is specified as a percentage value of the target life. The algorithm iterates using an interval halving technique until the calculated life lies within *a%* of the target life. Therefore, the smaller the value of *a*, the more accurate the estimate of Kf or scale factor will be.

As with any iterative algorithm, there is a potential for it to not converge on an answer. However, this tends to happen only if the materials data is very badly formed. The main error that can arise from a back calculation is caused by setting lives that are too short or too long for the damage calculation to work, i.e., when static failure is an issue or where the lives calculated are beyond the cut-off value.

Note that when using schedule files in “Independent” mode, only one scale factor or Kf is calculated for the whole data, not one per event.

When mode=Kf the value of kf is calculated to achieve the target life. The property “Kf” is therefore not used, but its existing value is retained. Similarly for the ScaleFactor property when Mode=ScaleFactor.

3.5 Interpreting the Results Generated by the StrainLife Glyph

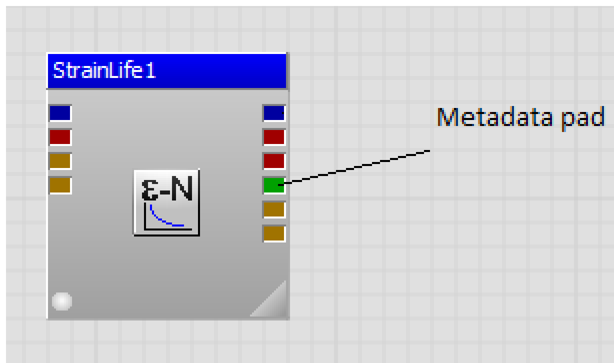
3.5.1 General Comments

Fatigue is a statistical process and calculating damage, and life, gives us an estimate of likely failure, not an absolute prediction. In any fatigue calculation, we are making assumptions about loading, material behavior and geometry and in each case we are using sample data that does not necessarily correspond exactly to a particular component specimen in service.

Therefore, it is important to see fatigue analysis software as a tool that helps make engineering decisions. The more reliable the data that goes in, the more confidence an engineer will have in the estimate that is given by the software. In all cases it is wise to allow safety margins on estimates, and to examine the effects any small changes in loading, materials behavior or other factors like surface condition may have on an answer.

3.5.2 Metadata Results

The key life and damage results are output on a metadata pad.



Typically, two metadata results sets are created. One contains the property settings on the glyph that created the result (called <glyph_name>_Properties) and the other has the results (<glyph_name>_Results). These metadata results are also sent to the time series, histogram and multi-column output pads, if those data are created.

The content of the results set varies depending on the input loading type. The common results are:

TotalDamage—The total damage calculated from 1 repeat of the input loading. If a static failure condition has occurred (local stress maximum is greater than UTS) then the damage value is reported but the DamageStatus result will be set to 1.

Life—The fatigue life, reported in equivalent units. This is usually repeats of the loading history (or histogram or multi-column data, or repeats of the whole schedule if the input is a schedule). The properties EquivalentLifeString and EquivalentLifeValue can be set so that a more meaningful description of the loading sequence can be defined.

EquivalentUnitsValue	5	Multiplier on the life result
EquivalentUnitsString	Laps	The equivalent units string

The life is first calculated from the inverse of the total damage for one repeat. This is then multiplied by the equivalent units value to get the life that is reported. If the damage is 0.25 and the equivalent units are set to 5 laps, the life will be 5 * (1/0.25) = “20 laps”. The life result is always reported as a string. If the total damage is zero, the life cannot be calculated, so the string is set to “Beyond cut-off”. If the maximum local stress has exceeded UTS, this is reported as “Static failure”.

DamageStatus—An integer, set to 0 for a successful calculation and 1 for a failed calculation. Examples of failed calculations include “Static failure”, when the UTS is exceeded, or a failure to converge a back calculation.

NumCyclesCounted—The total number of cycles counted and used in the analysis. Cycles that are removed by the gate are not included in this number.

Duration—This is set to the actual length of the time history or schedule that has been processed. If the input loading is a histogram, this may be passed through from its metadata. Units are always in seconds.

ScaleFactor—The result of a back calculation on scale factor

Kf—The result of a back calculation on Kf

MaxLife and MinLife —The extreme values of life calculated when the input is a rainflow matrix. These are in addition to the Life result.

MaxTotalDamage and MinTotalDamage—The extreme values of damage calculated when the input is a rainflow matrix. These are in addition to the TotalDamage result.

MaxStrainNominal and MaxStrainLocal—The extreme strain values before and after notch correction is applied

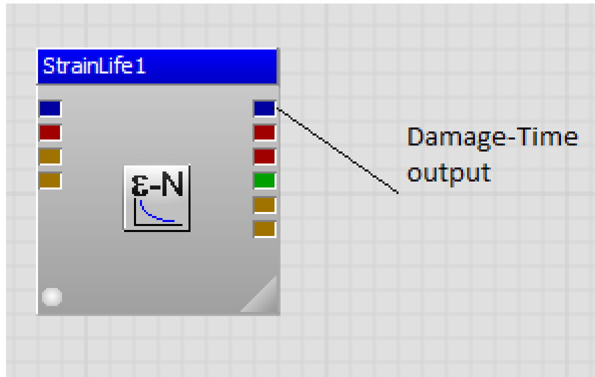
MaxStressNominal and MaxStressLocal—The extreme stress values before and after notch correction is applied

DamagePerHour—This is output if the Duration metadata item is available and is calculated as 3600.0*TotalDamage/Duration. This assumes the duration is in seconds.

LifeInHours—Life is converted to hours using the Duration metadata item, if it is available.

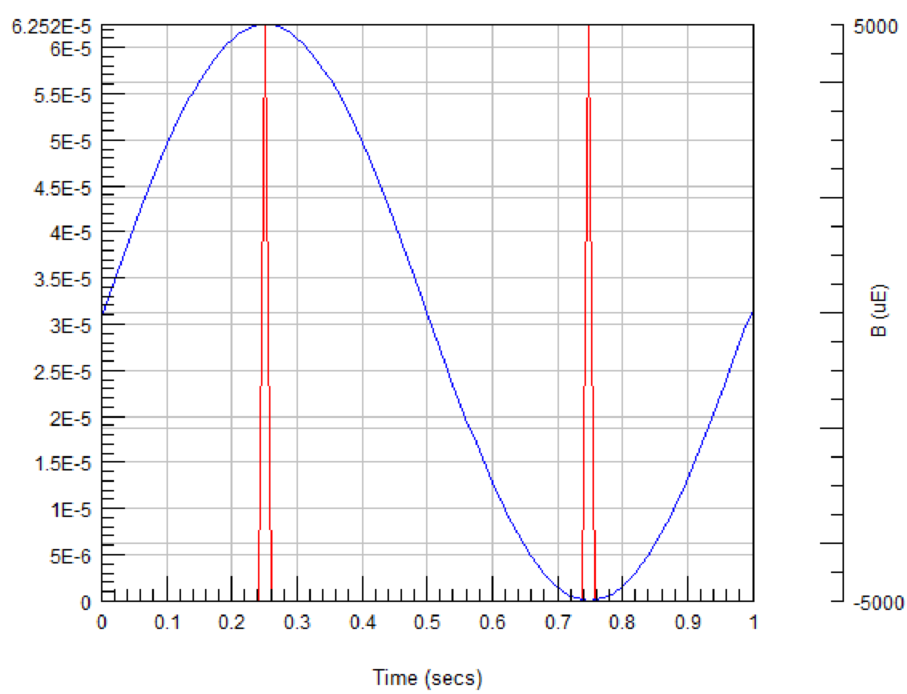
3.5.3 Interpreting the Damage-time History

A damage-time history is exported when the input is a time history. It is not created if the input loading is from a rainflow matrix or multi-column. Connecting to this pad is optional.



For each input channel, an output time series channel is created with the same sample rate and length. Each point therefore corresponds to a point in the original strain time series. During the calculation, rainflow cycles are counted from the turning points in the data. When the cycles are counted, the position of the maximum and minimum of the cycle is remembered in the cycles list. From this, the glyph allocates half of the damage for each cycle to the maximum and minimum points of the cycle, and this is what is shown in the damage-time series. A very simple example is a sine wave:

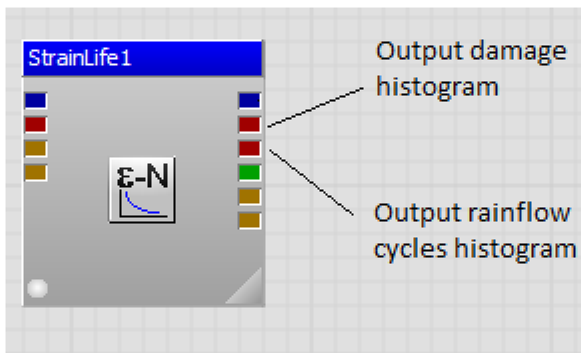
Fig. 3-37 Sine wave - damage



Here there is one cycle, with the maximum and minimum at the extremes of the wave. The total damage in this case is 1.2504E-4 and half of this is allocated to the maximum and half to the minimum. The resulting damage-time history is shown overlaid in red on the sine wave. The allocation of damage in this way is intended as a guide to the user as to which events within a time history are causing damage—or not causing it. The damage-time history is used by the Damage Editing glyph to remove sections of data that do not contribute to damage; this helps to reduce testing time.

3.5.4 Histogram Outputs

When the input is a time history or a rainflow matrix, two output histograms are created, one containing the rainflow cycles that have been counted and one containing the damage.



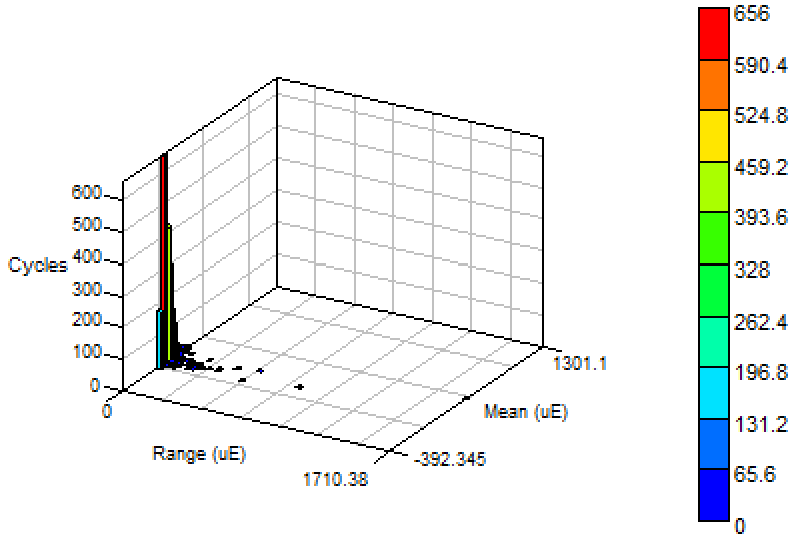
The options available in the glyph to control the output histograms are:

HistogramType	Nominal_RM	Sets the X and Y-axis type
DamageHistogramBins	64	Number of bins in the output damage histogram

If using time series inputs, the rainflow histogram is used only for visualization. The unbinned list of cycles is used for damage. Binning is done only for viewing purposes with the rainflow histogram. The histogram axes are determined by the HistogramType property and the number of bins on each axis determined by the DamageHistogramBins property.

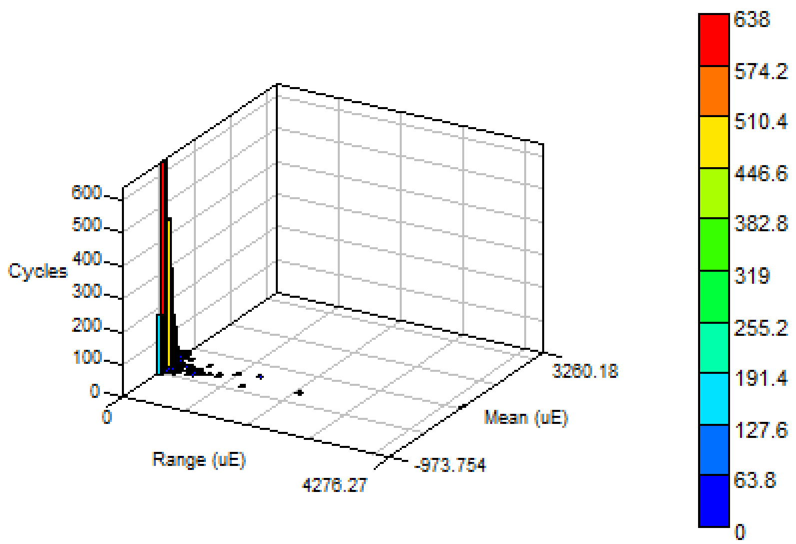
A typical range-mean rainflow matrix, with nominal strains (Histogram-Type=Nominal_RM), looks like this:

Fig. 3-38 Rainflow matrix with nominal strains



If the HistogramType property is set to Local_RM (local strain, range-mean) for the same analysis, where the fatigue concentration factor is set to 2.5, the output looks like this:

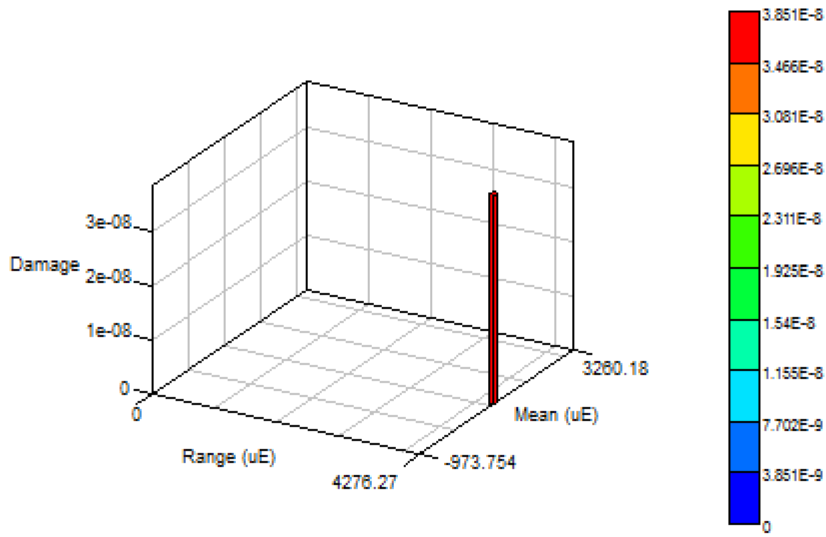
Fig. 3-39 Fatigue concentration factor = 2.5



The bin sizes and limits have changed, and the count in the bins is also different. If Kf=1, and the data is from a measured strain gauge, these histograms will be the same.

The same bin sizes are used for the damage histogram. For the example shown above, the damage histogram looks like this:

Fig. 3-40 Damage histogram

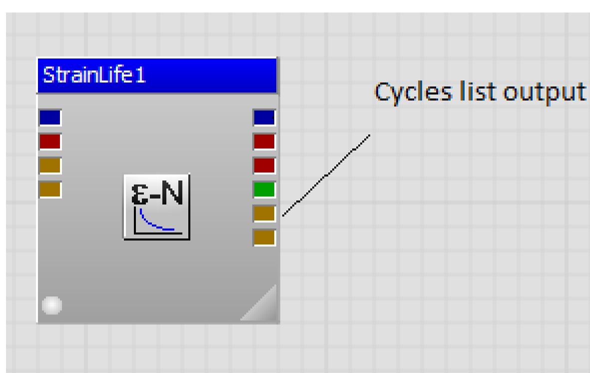


Note that only the largest cycle is generating damage.

If the input loading is a rainflow matrix, then the properties HistogramType and DamageHistogramBins have no effect. The output histograms will have the same axes and number of bins as the input histogram, even if a Kf is applied.

3.5.5 Cycles List Output

A full list of cycles used in the analysis is exported on a multi-column output pad.



The data in the columns is dependent on the input loading type. For a long time history analysis, many cycles can be generated and the data set can become very large. All the information on the size of the cycle, nominal and local stresses and strains, the direction of loading, and the damage associated with each cycle is in the table of results.

If the input loading is a histogram, two calculations are performed, best case and worst case. Only one cycle list is created, and this is controlled by the property CyclesOutputFromHistogram.

CyclesOutputFromHistogram	BestCase	Whether to output best or worst case strains/stresses for histogram input.

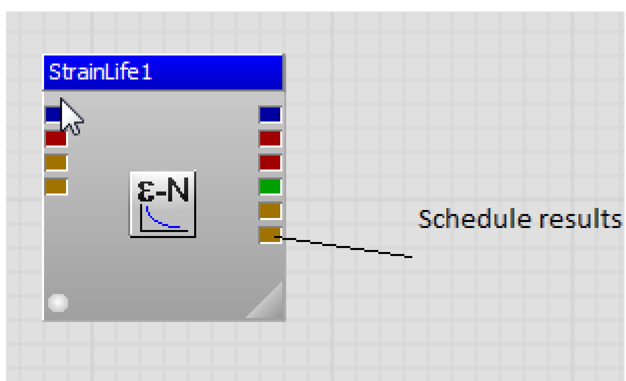
The options are BestCase and WorstCase.

Because multiple cycles are binned together in a histogram, the cycle list has a "Number of cycles" column in each row. The reported damage is the damage for all those cycles, not for a single cycle of that size. So, if the damage reported is 0.5 and there are 5 cycles, the damage for each cycle is 0.1. If the number of cycles in a bin is zero, then there will be no output from that bin.

If the input is a multi-column loading, then the cycles list output will match the input: one row of input equals one row of output. Again there can be a number of cycles defined for the same size, and the damage reported per row is for all those cycles, at it is for a histogram and not the damage for a single cycle of that size.

3.5.6 Schedule Results

If a schedule file is used for input loading, extra results are output onto a second multi-column pad.



These results show the damage for each channel, per event and summed across events.

1	2	3	4	5	6	7
Channel number	Channel title	Event name	Damage	Percent damage	Damage with status	Repeat count
1	LF_Long_Accl	Event_A	0.002211	8.064	0.002211	1
1	LF_Long_Accl	Event_B	0.02521	91.94	0.02521	4
1	LF_Long_Accl	ALL	0.02742	100	0.02742	1
2	LF_Lat_Accl	Event_A	0	0	Beyond cutoff	1
2	LF_Lat_Accl	Event_B	0	0	Beyond cutoff	4
2	LF_Lat_Accl	ALL	0	0	Beyond cutoff	1
3	LF_Vert_Accl	Event_A	0.007676	32.96	0.007676	1
3	LF_Vert_Accl	Event_B	0.01561	67.04	0.01561	4
3	LF_Vert_Accl	ALL	0.02329	100	0.02329	1

The percent damage column shows the percentage of damage per event on the channel.

3.6 The Strain-Life Method: A Summary

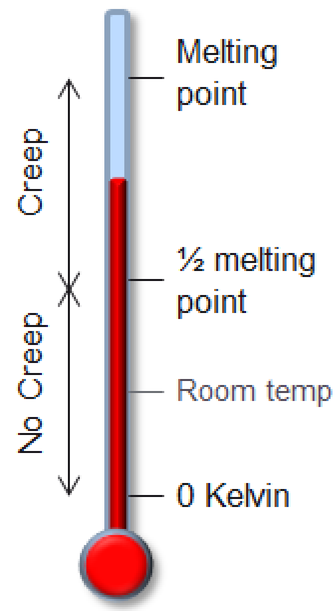
The strain-life method relates strain cycles to fatigue life through a material's E-N curve.

- The strain-life method is valid for both low- and high-cycle fatigue regimes.
- The strain-life method tracks stress-strain response even after yielding and therefore makes theoretical sense when the applied loading is likely to induce significant plasticity.
- The material parameters s'f, e'f, b, c, n' and K' are determined experimentally.
- Tensile mean stresses are detrimental to life while compressive mean stresses are beneficial.
- At high strain amplitudes, in the low cycle region, the effect of mean stress and other modifying factors is minimized.
- Neuber's Rule can be used to model local stress-strain response in fatigue-sensitive areas like stress concentrations.
- These methods are valid only for uniaxial (or near uniaxial) loading conditions.

4 Creep Analysis

Creep is a damage mechanism in materials that occurs when a material is subjected to a constant stress at temperatures close to the melting point. The level at which this occurs is material dependent and often begins at around half the melting point (in Kelvin).

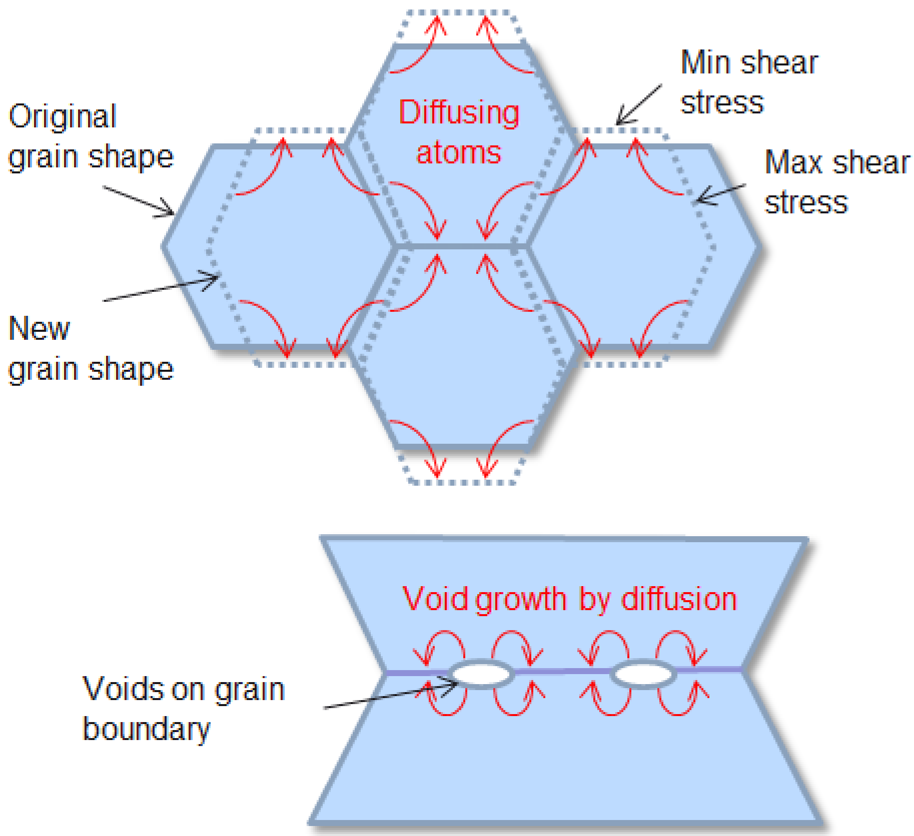
Fig. 4-1 The creep zone



Unlike fatigue, the stress does not need to be "cycled" for creep damage to occur. There are a number of different mechanisms of creep depending on temperature and stress. The one illustrated below is Nabarro-Herring creep.

As temperature increases, the atoms within the material are excited and diffuse from the stressed face to the unstressed face, causing microscopic gaps to occur that gradually, over time, grow larger.

Fig. 4-2 Creep mechanism (diffusion and void growth)



The rate of deformation within the structure is a function of the material properties, exposure time, exposure temperature and the applied structural load. Creep is usually of concern to engineers and metallurgists when evaluating components that operate under high stresses or high temperatures. Eventually creep can cause the material to fail, or "rupture".

Several models have been proposed that estimate the time to rupture based on a constant temperature and stress. Two of these methods are implemented in GlyphWorks: Larson-Miller and Chaboche. The starting point for these calculations is the materials data. To obtain this data, creep tests must be performed. A set of creep tests consists of taking a specimen of known geometry, hanging a weight from the specimen and heating the specimen to a known, constant temperature. Eventually the specimen will rupture and fail. The time to rupture is recorded and so a table of values for stress, temperature and time to rupture is obtained. For the different methods, these test results are used in different ways to obtain the curves or equations that describe the materials behavior.

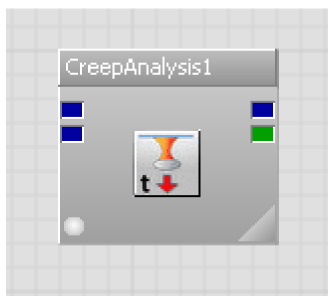
If either or both of the temperature and stress are varying with time, it is possible to calculate partial creep damage values for small increments of time and then sum them together to calculate damage against time. When the total damage reaches 1, rupture is estimated to have occurred.

In the Larson-Miller method, damage is summed linearly, whereas Chaboche includes a non-linear damage summation model.

Creep damage can be summed with fatigue damage to get a total damage, on the assumption that there is no interaction between the creep and fatigue modes.

4.1 Creep Analysis Glyph

GlyphWorks provides a glyph for creep analysis offering options for both the Larson-Miller and Chaboche methods.



The method is selected depending on the type of materials data used by the glyph.

4.2 Time History Inputs

Two optional input pipes are provided. The top one takes a stress or strain input. This is converted to MPa units based on an automatic recognition of its units, or by the user forcing one of the supported unit types in the nCode system. This is done using the InputUnits property.

Loading		
InputUnits	Auto	Selects the units of the input data.

If no time series of stress is supplied, the glyph will take a constant value of stress from the property ConstantStress, which also has the ability to be in different units specified by the ConstantStressUnits property.

ConstantValue		
ConstantStress		Constant stress value if no stress time history is supplied.
ConstantStressUnits	MPa	Selects the units of the constant stress property.

If no time history of temperature is supplied, the glyph will take a constant value of temperature from the property ConstantTemperature, which is in units of Celsius.

ConstantTemperature	600	Constant temperature (in Celsius) if no temperature time history is supplied.

If either of the constant values is specified, they take priority over the connected time series input and a warning is issued that the time history will be ignored.

Scaling the Stresses

A scale factor and offset can be defined to calibrate or adjust the stresses. The scale factor and offset are applied only to the stress history as a y=mx+c equation. The units of the offset must match the input units—if the input is strain, then the offset must be specified in MPa.

ScaleFactor	1	Linear multiplier on the input stress data.
Offset	0	Offset on the input stress data.

Units Conversions

All internal calculations in the glyph are in MPa, degrees Celsius and hours. At various points inputs can be in different units to these and the glyph will ask for those units and perform the necessary conversion using the nCode units system.

Typically the input time histories, if used, will be in seconds. The x-axis unit is interrogated by the glyph and a conversion to hours performed. If the unit is not recognized, seconds are assumed. If it is recognized as a type which is not "time", then an error will be generated.

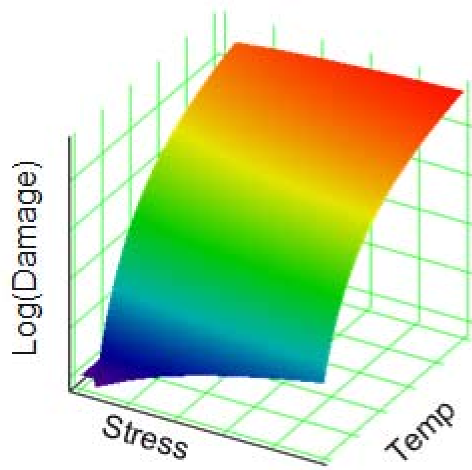
4.3 Larson-Miller Method

The Larson-Miller method collapses the results from the creep tests onto a single curve that is independent of temperature.

$$P(T, t_r) = T(C + \log(t_r))$$

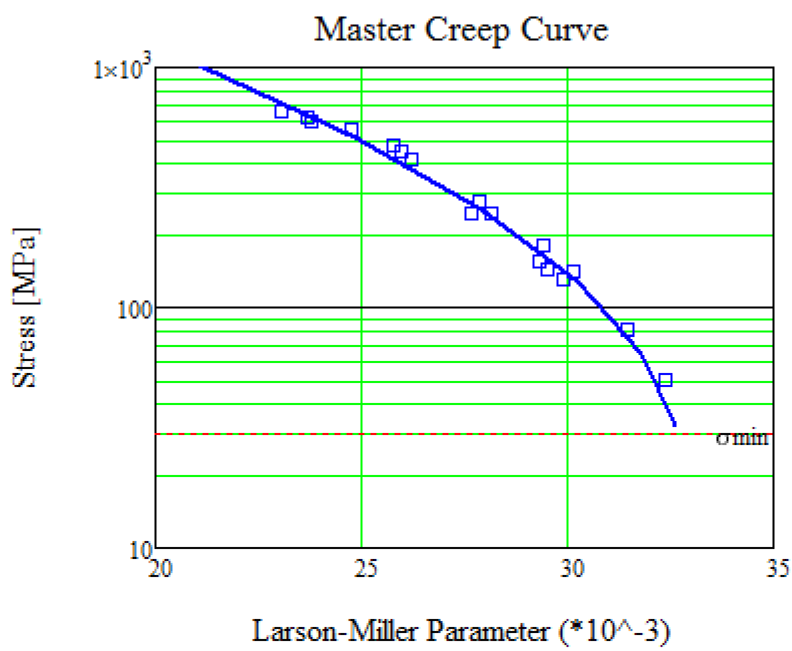
Where P is the Larson-Miller parameter, T is the temperature in Kelvin, C is the Larson-Miller constant, and t_r is the time to rupture. C is a parameter defined during the curve fitting process that, for steels, is around 20. The following figure shows a general example of this function.

Fig. 4-3 Larson-Miller curve



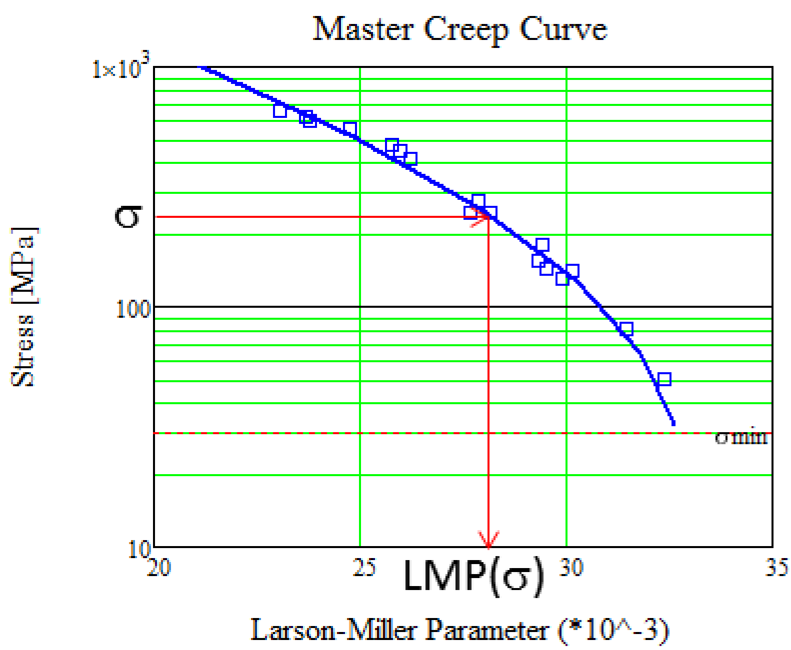
A master creep curve is generated from the test data for Stress v. Larson-Miller parameter. The curve looks like this:

Fig. 4-4 Master creep curve



In this case, a polynomial is fitted through the data—this is one of two methods allowed in the software to represent the master creep curve. The other is a piece-wise X-Y curve—these are described in "Larson-Miller Materials Data" on page 151. In both cases, for a given value of stress, the curve is used to obtain a value for the Larson-Miller parameter. Polynomial materials definition is currently in terms of stress, in MPa.

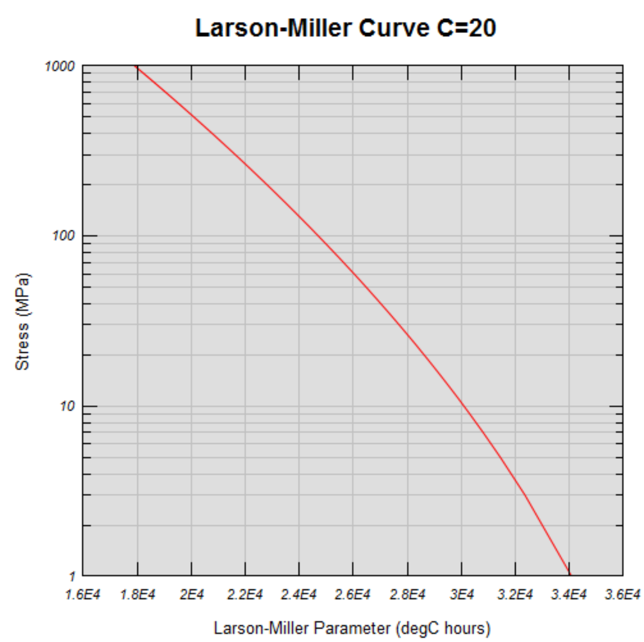
Fig. 4-5 Derive Larson-Miller Parameter (LMP)



If this is a polynomial, the Larson-Miller parameter can be calculated directly as a function of σ.

	LMp Test Data 2	Description
MaximumStress	800	Maximum Allowable Stress (MPa)
MinimumStress	30	Creep Stress Threshold (MPa)
MaximumTemperature	1050	Maximum Temperature (deg C)
MinimumTemperature	500	Creep Temperature Threshold (deg C)
C	20	Larson-Miller Constant
a4	0	Multiplier on the 4th order of the polynomial.
a3	0	Multiplier on the 3rd order of the polynomial
a2	-705	Multiplier on the 2nd order of the polynomial
a1	-3280	Multiplier on the linear term of the polynomial
b	3.41E4	Constant term of the polynomial
Comments		Comments
References		References

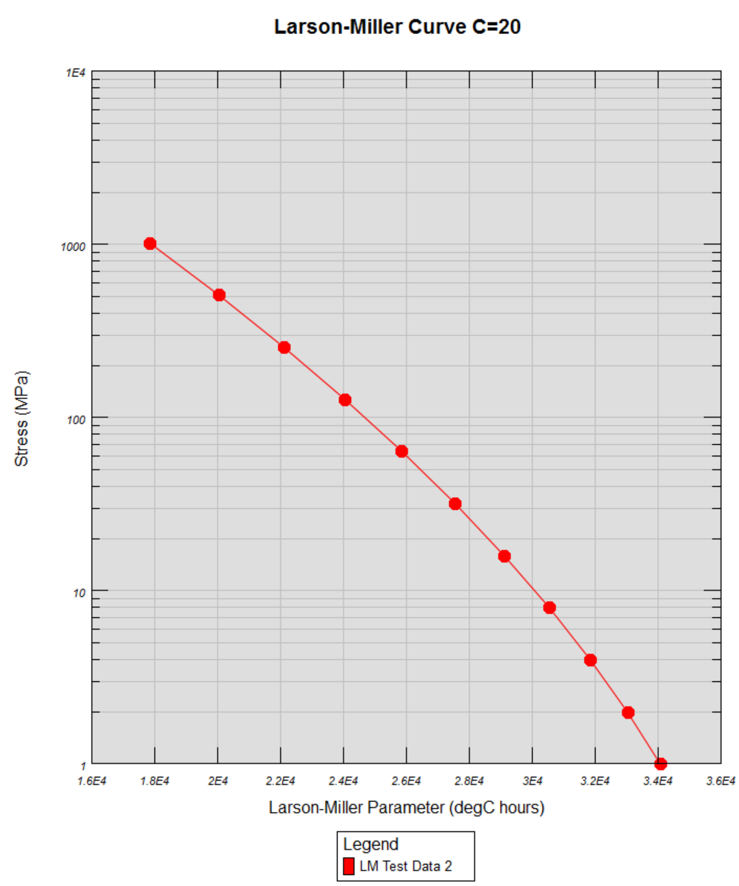
Fig. 4-6 LM Curve, constant = 20



For a piecewise curve, linear interpolation is used to get the Larson-Miller parameter from stress. Note that the data is plotted, and the interpolation performed with log Stress against linear Larson-Miller parameter.

	LM Test Data 2	Description
MaximumStress	800	Maximum Allowable Stress (MPa)
MinimumStress	30	Creep Stress Threshold (MPa)
MaximumTemperature	1100	Maximum Temperature (deg C)
MinimumTemperature	500	Creep Temperature Threshold (deg C)
C	20	Larson-Miller Constant
StressValues	1,2,4,8,16,32,64,128,256,512,1024	Stress values (MPa)
PLMValues	3.408E4,3.303E4,3.185E4,3.055E4,2.911E4,2.755E4,2.5...	Larson-Miller parameter values (degC hours)
Comments		Comments
References		References

Fig. 4-7 Log stress v LMP



Note The Larson-Miller method assumes that compressive stresses contribute no creep damage.

From the Larson-Miller parameter and the temperature, the time to rupture is defined as

$$t_r = 10^{\frac{PLM(\sigma)}{T}}$$

So, in the simple case where we have a static temperature and a static stress, the lookup from the master creep curve to get the Larson-Miller parameter from stress and the temperature can be used to calculate the life (in hours) directly.

However, if either the stress or temperature, or both, varies with time, then the incremental damage is calculated per sample point. This is derived by calculating the time to rupture based on the stress S_n at time t_n and the corresponding value of T, T_n , at the nth time step. This value is called tr_n. The incremental damage is defined as

$d_n = \Delta t / tr_n$

where Δt is the time increment in hours between samples.

The total damage, D, is the sum of all the incremental damages for time steps 1 to N.

$$D = \sum_{n=1}^{n=N} d_n$$

The number of repeats of the sequence to failure (rupture) is

$$R = \frac{1}{D}$$

The time to failure (rupture) in hours is obtained from the length of the input data, L (hours):

$$T_f = \frac{L}{D}$$

4.3.1 Larson-Miller Materials Data

Two forms of the Larson-Miller master creep curve are supported in Materials-Manager. For both types of data, the value of C must be defined, Young's modulus is required and there are limit values on maximum and minimum stress and temperature.

The first of the two data sets defines a polynomial. (P_{LM} = Larson-Miller parameter.)

$P_{LM} = b + a_1(\sigma) + a_2(\sigma)^2 + a_3(\sigma)^3 + a_4(\sigma)^4$

Property	Description
MaximumStress	The maximum value of stress allowed in the calculation
MinimumStress	The minimum value of stress allowed in the calculation
MaximumTemperature	The maximum value of temperature allowed in the calculation
MinimumTemperature	This is effectively the creep temperature limit—below this temperature no creep damage is accumulated.
C	Larson-Miller constant
b	The constant term of the polynomial
a1	The multiplier of the linear term in the polynomial
a2	The multiplier of the quadratic term in the polynomial
a3	The multiplier of the cubic term in the polynomial
a4	The multiplier of the 4th order term in the polynomial
Comments	
References	

The second type is a piecewise curve defined by pairs of points for S and the Larson-Miller parameter. There must be the same number of points defined for each parameter.

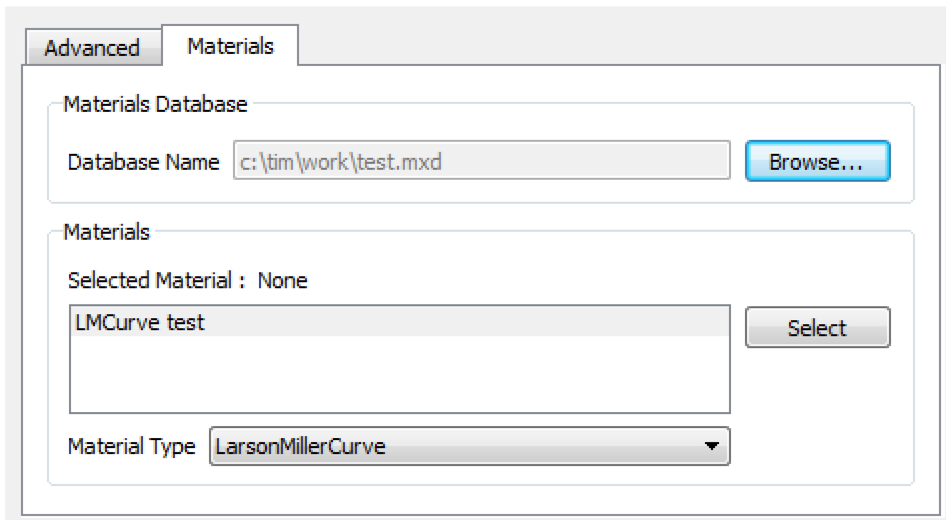
Property	Description
MaximumStress	The maximum value of stress allowed in the calculation
MinimumStress	The minimum value of stress allowed in the calculation
MaximumTemperature	The maximum value of temperature allowed in the calculation
MinimumTemperature	The minimum value of temperature allowed in the calculation
C	Larson-Miller constant
StressValues	The values of stress in the stress-P(LM) pairs
PLMValues	The corresponding values of P (Larson-Miller)
Comments	
References	

To select the materials data in the glyph, the name of the database, the type of Larson-Miller dataset, and the name of the material can be typed in as properties:

Material		
MaterialDataSource	MXD_database	Specifies where the material properties are found.
MaterialDatabaseName		The name of the materials database to use.
MaterialName		
MXDMaterialType	LarsonMillerCurve	Specifies type of material to use from an MXD database.

The MaterialDataSource property has only one option at present, the nCode MXD materials database.

Alternatively, the database and the material type and name can be selected from the Materials dialog.



4.4 Chaboche Method

Based on the work of Rabotnov (1958) and Kachanov (1969), Chaboche and Lemaitre (1978, 1981) proposed a creep damage evolution equation which is expressed as Equation 1:

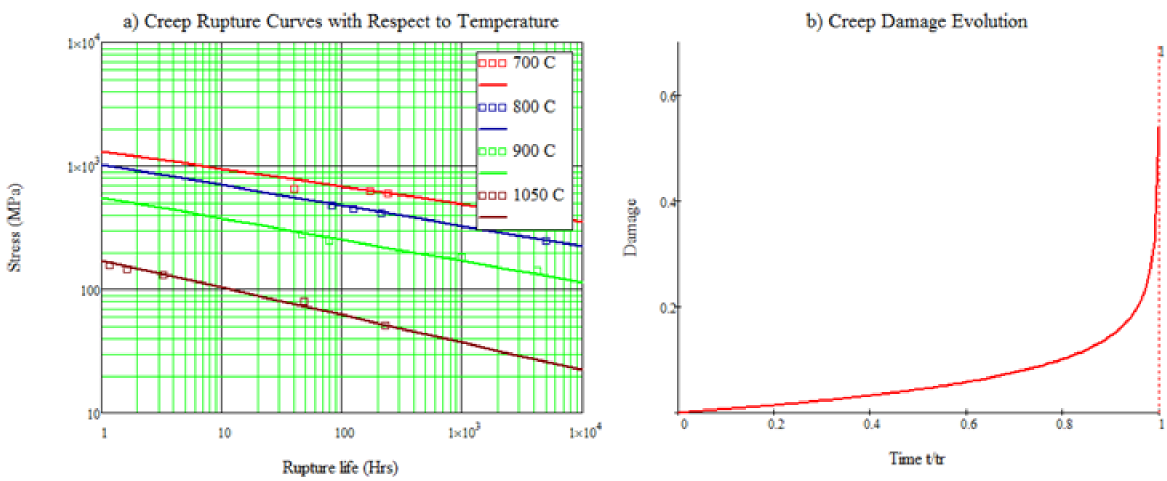
Equation 1

$$dD = \left(\frac{\sigma}{A}\right)^{r} (1 - D)^{-k} dt$$

where D is the damage variable ranging between 0, for the virgin condition, and 1 for the failed condition. A, r and k are temperature-dependent material parameters and are determined from creep rupture tests at several stress levels, as illustrated in “Creep Rupture Curves with Respect to Temperature” in Figure 4-8.

The creep rupture curve is idealized as a straight line in log-log space. Parameter A represents the intercept of the curve with the stress axis while the exponent r represents the slope. k is used to describe the non-linear damage evolution as observed from damage measurements. See "Creep Damage Evolution" in Figure 4-8. A value of k = 0 implies linear damage accumulation as per Robinson's rule (1938). k has been found larger than r in general and is a function of increasing stress. However, constant k is preferred for creep-fatigue life calculations and the method used in GlyphWorks assumes that k is constant with respect to both stress and temperature. Some variations of the above model can be found in the work of Chaboche (1976, 1977). In this document, the focus is on life calculation for different stress and temperature cycles.

Fig. 4-8 Creep rupture and creep damage evolution curves



If both the stress and temperature are constant, a straightforward equation for the time to rupture is derived by integrating Equation 1. This comes out as:

$$t_r = \frac{1}{k+1}\left(\frac{\sigma}{A}\right)^{-r}$$

A and r are the values at the constant temperature; if this temperature is not one of the values for which tests have been performed, an interpolation algorithm is used to calculate the values for that temperature. A method that uses logarithmic interpolation for both A and r is used in GlyphWorks and this seems to give reasonable results when compared to test data. Note that because of the logarithmic nature of the interpolation it is done with temperatures in units of Kelvin. Extrapolation of A and r is allowed, as long as the temperature is within the maximum and minimum material temperatures.

When processing time histories we can calculate the incremental damage for a time series sample.

If the inter-sample time is Δt, and this is reasonably small, the incremental damage for stress σ is

$$\Delta D = \Delta t.\left(\frac{\sigma}{A(T_i)}\right)^{r(T_i)}.(1-D)^{-k}$$

where A(Ti) and r(Ti) are the material properties at the temperature over which incremental damage is being calculated, D is the total damage so far, and k is the non-linear summation factor.

The total damage D=D+ ΔD and this process is then repeated continuously, looping around the time series as many times as required, until D=1 at rupture.

The problem with this method is that it is very inefficient and an optimized algorithm is required for long durations. The solution relies on the non-linear damage evolution constant k being assumed invariant with temperature and that the time history is repeated a number of times before rupture occurs.

In this case we can remove the term involving k from the calculation of ΔD to reduce it to:

$$\Delta D = \Delta t.\left(\frac{\sigma}{A(T_i)}\right)^{r(T_i)}$$

The total damage D is calculated D=D+ ΔD and at the end of one repeat of the time series we have the damage for the first repeat.

To calculate the number of repeats to rupture:

$$N_f = \frac{1}{(k+1).D}$$

And the time to rupture, in hours, is

$$t_r = N_f.Duration$$

where Duration is the length of the signal in hours, assuming that the signal is repeating; this means the equation is

$$Duration = \Delta t.N$$

where N is the number of samples. This differs from the normal calculation of duration in GlyphWorks, which uses N-1. In long time histories, this difference is not really noticeable, but can be seen on short sequences.

The algorithm accounts for rare cases of compressive healing in materials. The healing coefficient ‘h’ is specified in the range $-1 \leq h \leq 1$ and applies to the incremental damage resulting from compressive stresses. The formula was derived by Chaboche and Lin (2006) and the total creep damage is obtained from Equation 2.

Equation 2

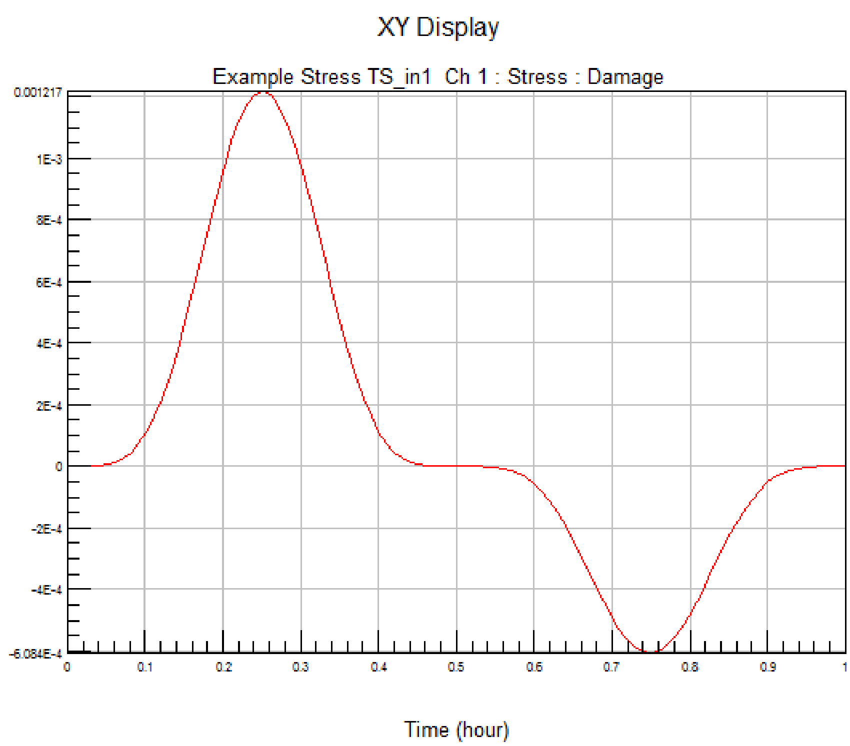
$$D_{total} = D(\sigma_{tension}) + h \cdot D(|\sigma_{compression}|)$$

where D_{total} = total creep damage, $D(\sigma_{tension})$ is damage associated with tensile stress and $D(|\sigma_{compression}|)$ is damage associated with compressive stress.

A value of h = 0 implies no creep damage arising from compressive stresses. This is the most commonly used assumption for creep-fatigue analysis. A value of h > 0 implies positive damage arising from compressive stresses. A value of h = 1 implies equal damage from compressive and tensile stresses. A value of h < 0 implies a proportion of compressive healing. The algorithm prevents the damage sum going negative.

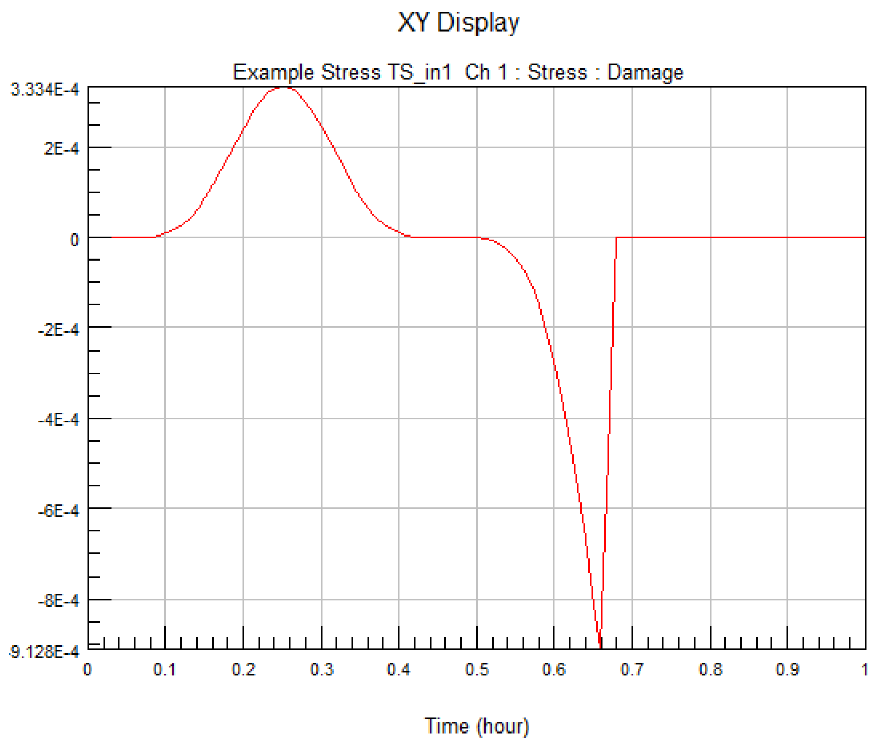
An example of how this affects the incremental damage is shown below, where the input stress history is a sine wave at a constant temperature. In this case h=-0.5 so the healing effect is half of the damaging effect.

Fig. 4-9 **Input stress = sine wave at constant temperature**



In the next example, the data has a negative mean, so the compressive stresses have a larger magnitude than the tensile stresses. This shows the point at which healing takes the total damage back to zero. All subsequent compressive stresses have an incremental damage of zero.

Fig. 4-10 Data with negative mean



Note that to use healing, the parameter h must be set in the materials data, and the minimum stress must be set to a negative value to allow the compressive stresses to be used in the algorithm.

4.4.1 References

Chaboche, J. L. (1981) Nuclear Engng Design 64, 233.

Chaboche, J. L., Policella, H., and Savalle, S. (1976) Application of the Continuous Damage Approach to the Prediction of High Temperature Low-Cycle Fatigue. ONERA.

Chaboche, J. L., Policella, H., and Kazmarek, H. (1977) Applicability of the SRP Method and Creep-Fatigue Damage Approach to the LCHTF Life Prediction of IN-100 Alloy. ONERA.

Kachanov, L. M. (1958) Lzv. Akad. Nauk. SSR, Otd Tekh. Nauk, No. 8, pp. 26-31.

Lemaitre, J., and Chaboche, J. L. (1978) J. de Meca. App. 2, 317.

Rabatnov, Y. N. (1969) North Hollan Publishing Company, Amsterdam. London.

Robinson, E. L., (1938) Trans. ASME 160, 253-259.

4.4.2 Chaboche Creep Materials Data

The ChabocheCreep materials dataset is defined as a parent dataset, with each temperature curve defined as a child set.

The parent properties are:

Property	Description
MaterialType	Numeric code defining the type of material. The material type is not currently used for creep calculations and is used purely for reference. See “Material Type Codes” for details.
MaximumStress	The maximum allowable stress independent of temperature
MinimumStress	The stress limit independent of temperature—below this value the damage is always set to 0.
MaximumTemperature	The maximum temperature allowed in the calculation for this material
MinimumTemperature	This is effectively the creep temperature limit—below this temperature no creep damage is accumulated.
k	The non-linear damage summation parameter
h	Compressive healing factor
Comments	
References	

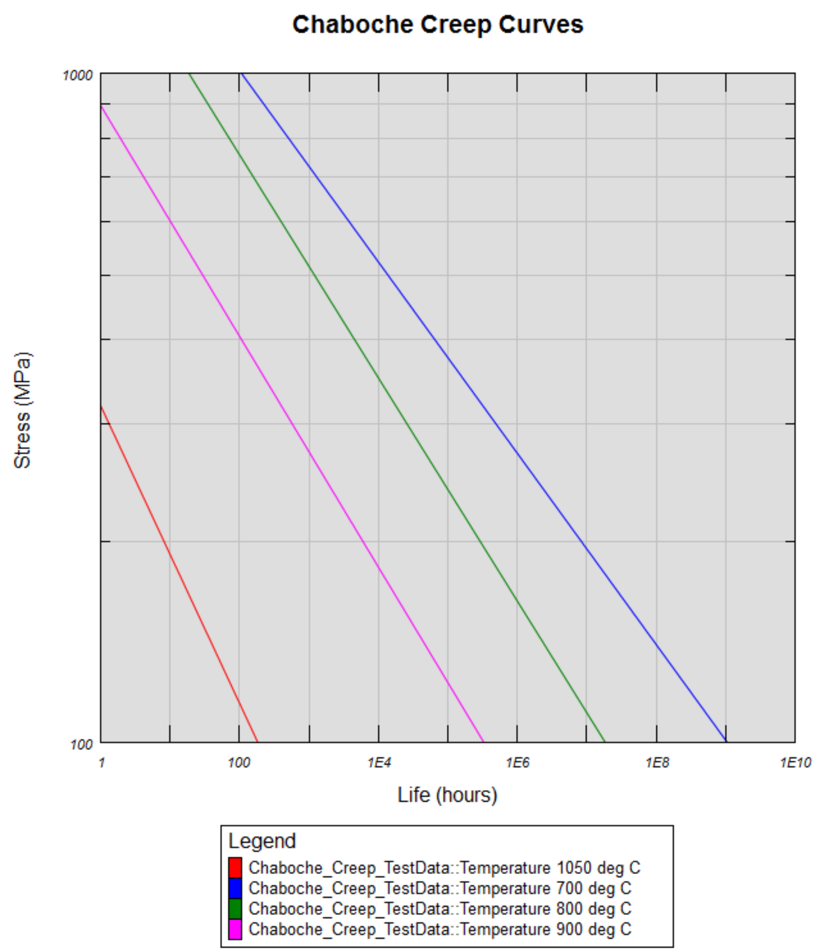
The child properties are:

Property	Description
Temperature	The temperature value for this curve
A	The co-efficient of the Chaboche Creep Stress-Life curve
r	The exponent of the Chaboche Creep stress-life curve

Parameters k, MaximumStress and MinimumStress are shown in this dataset but are inherited from the parent and cannot be modified.

Visually, the curves look like this:

Fig. 4-11 Chaboche creep curves



To use the Chaboche method, select ChabocheCreep as the MXDMaterialType and pick a dataset from an MXD database that contains Chaboche properties.

4.5 Stress Limits

For both methods, the solution is bounded for both low values and high values of stress. The **minimum allowable stress** effectively sets a creep limit below which no creep damage is accrued. This is particularly important when using a polynomial based Larson-Miller materials dataset as the equations can become unstable in areas where the data is not fitted to the curve.

The **maximum allowable stress** is the largest stress for which the calculation is performed. This could be defined as the UTS, or at least the UTS for a fixed temperature. However, because the Larson-Miller parameter is defined for a range of temperatures, the creator of the dataset should define the “maximum allowable stress”. If the calculation exceeds this value, “static failure” is reported and the damage is set to 1.

4.6 Temperature Limits

For both methods, the solution is bounded for both low values and high values of temperature. The **minimum allowable temperature** is the temperature below which no creep, or insignificant creep, occurs. This is typically around 50% of the melting point in Kelvin, but again may be noted based on materials tests.

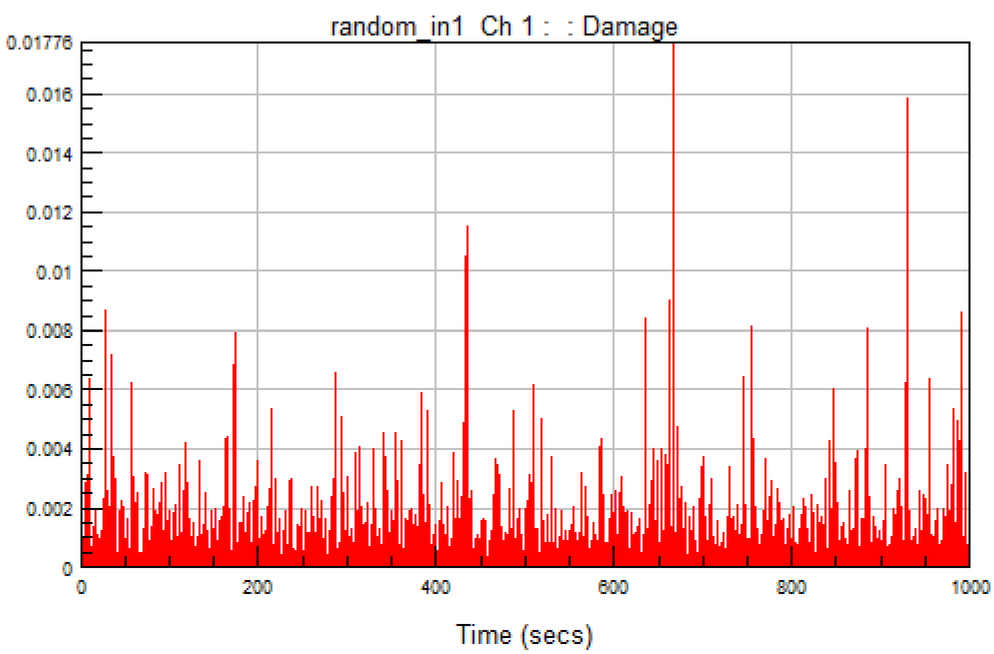
The **maximum allowable temperature** is set to prevent temperatures close to or above the melting point being used in the analysis. If it is exceeded, damage is set to 1.

4.7 Interpreting the Results of a Creep Analysis

Two output pipes deliver the results of the analysis.

The top pipe is a time series of incremental damage against time.

Fig. 4-12 Creep damage vs time



If both the stress and temperature are constant, this output is not produced.

The second pipe carries the results of the analysis as metadata. The metadata also has a data set that audits the properties used in the analysis.

	Duration	LifeInHours	MaximumStressExceeded	MaximumTemperatureExceeded	Life	TotalDamage
1	1000	1.977E6	False	False	7.116E6 repeats	1.405366561E-7

The results are tabulated below.

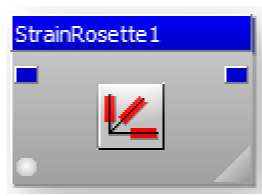
Metadata item	**Description**
LifeInHours	The time to rupture, in hours
Life	The number of repeats of the data before rupture occurs
TotalDamage	The total incremental damage for one repeat
MaximumStressExceeded	True if the maximum allowable stress is exceeded
MaximumTemperatureExceeded	True if the maximum allowable temperature is exceeded
Duration	The length of the data in seconds

The Duration and Life results are applicable only if the calculation is time series based. If both stress and temperature are constant, these items are not written out.

5 Strain Gauge Rosette Analysis

Strain gauges are used in testing to measure the strain in a component. Strain is a normalized measure of deformation representing the displacement in a body relative to a reference length. Strains are dimensionless, and in general are a tensor quantity, although since scalar strains are often required for analysis, for example in uniaxial fatigue calculations, there is a requirement to calculate scalar strain from tensor strains. This glyph performs that function, and also calculates other parameters that show how much the strain changes in direction with time, as this can be critical to the validity of uniaxial fatigue calculations.

5.1 Strain Rosette Glyph



GlyphWorks provides a glyph that takes data from a variety of rosette types (include virtual rosettes from DesignLife) and generates output scalar strains and stresses, plus other useful parameters, for use in fatigue glyphs and for display.

5.2 Types of Strain Gauge

The StrainRosette glyph supports three types of rosette configuration: rectangular, delta and tee.

Tee rosettes consist of two mutually perpendicular grids. They are used when the direction of the principal strain is known, so that it can be aligned to that direction. Cylindrical pressure vessels and shafts in torsion are two classical examples where this type of gauge is used.

Rectangular rosettes have three grids, with the second and third grids angularly displaced from the first grid by 45 and 90 degrees, respectively.

Delta rosettes have three grids, with the second and third grids 60 and 120 degrees away, respectively, from the first grid.

In both cases, these can determine principal values and directions without prior knowledge.

If three input signals are provided, the user must specify which of the three leg rosette types is in use.

If two channels are supplied, the configuration is assumed to be Tee.

Provided below are geometrically different, but functionality equivalent configurations of rectangular and delta rosettes:

Fig. 5-1 Rectangular rosette

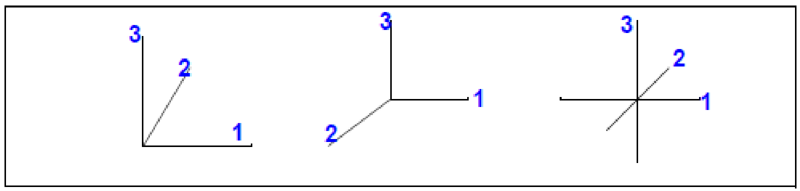
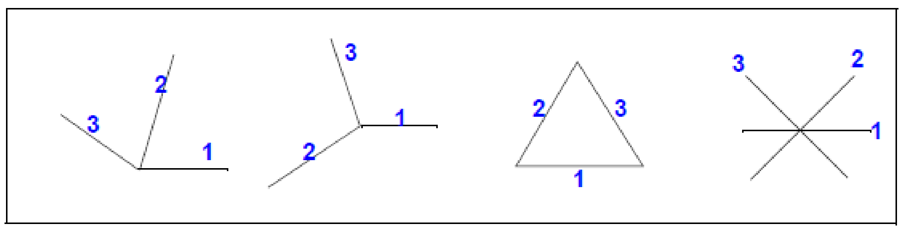


Fig. 5-2 Delta rosette



By convention, gauge numbering, 1,2,3 is counter-clockwise.

Rectangular rosette:

ε_1 First strain gauge
ε_2 Strain gauge at 45 to ε_1
ε_3 Strain gauge at 90 to ε_1

Delta rosette:

ε_1 First strain gauge
ε_2 Strain gauge at 60 to ε_1
ε_3 Strain gauge at 120 to ε_1

5.3 Time History Input

The glyph either takes in a multiple channel time history where the channels can include strain gauge data, or it can take in two or three separate inputs where each strain gauge leg is on a separate input.

This is configured using the NumTSInputPads property.

⊟ **Number of Input Test**		
NumTSInputPads	Single	Specifies the number of input tests

⊟ **Number of Input Test**		
NumTSInputPads	Multiple	Specifies the number of input tests

If only a single input stream is provided, the user must supply a list of rosette definitions, using the channel numbers or channel titles for each leg in the correct order. One way in which this can be done is by using the “Channels” property.

⊟ **Channel Options**		
Channels		Specifies the channels for a number of rosettes

The format of the string is to use curly brackets to delimit the rosettes and have the channel numbers as a comma-separated list inside them. For example:

{1,2},{4,5,6},{12,13,14}

This will generate results for three rosettes, where the first is a tee rosette and the second two are either rectangular or delta depending on the RosetteType property.

Channel titles can also be used. See the Glyph Reference Guide for more details on how to specify titles.

The other way to specify rosettes is to use a Group File.

GroupFile		Specifies a filename containing information about which channels are grouped as rosettes

This can be used in addition to or instead of the Channels property. The format is specified in the Glyph Reference Guide.

A number of rosettes may be defined in the file or channel list, but not all of these may actually be present in the data, for example if a gauge is broken. The glyph can either error or continue, depending on the IgnoreMissingRosettes property.

IgnoreMissingRosettes	False	Specifies whether or not channels referred to in a group definition must exist

5.4 Strain Gauge Configuration

Strain gauges can be configured as stacked (layered on top of each other) or planar (next to each other in the same plane). Planar rosettes are normally preferred when strain gradients are not severe but they occupy more surface area and when space is limited a stacked configuration may be preferred.

All the gauges in a stacked rosette have the same gauge factor and transverse sensitivity while the grids of a planar rosette will have slightly differing values of these properties. In the latter case, the transverse sensitivities of grids 1 and 3 are generally the same and grid 2 is different.

The rosette manufacturer's data sheet should indicate the type of rosette that the analysis is being set up for.

⊟ Rosette Construction		
RosetteConstruction	Stacked	Specifies the configuration of a three leg rosette

5.5 Transverse Sensitivity

Transverse sensitivity in a strain gauge refers to the behavior of the gauge in responding to strains that are perpendicular to the primary sensing axis of the gauge. Ideally, it would be preferable if strain gauges were completely insensitive to transverse strains. In practice, most gauges exhibit some degree of transverse sensitivity; but the effect is usually quite small, of the order of a few percent of the axial sensitivity.

The transverse sensitivity coefficient, Kt, is defined as the ratio of the transverse and axial gauge factors and is used to correct measured strains for the effects of transverse Poisson's strains. It should not be confused with the elastic stress concentration factor of the same name.

A maximum of three transverse sensitivity coefficients can be entered—the number required depends on the configuration.

The number of values to be entered is:

One—Stacked Rosette Configuration

Two—Planar Rosette Configuration

Three—User-defined Rosette Configuration

Enter the value of transverse sensitivity coefficient as a percentage. Please refer to the rosette manufacturer's data sheet for the value(s) of this factor.

The default value for transverse sensitivity is zero, which means no correction will take place.

TransverseSensitivity1	0	Specifies the first transverse sensitivity coefficient
TransverseSensitivity2	0	Specifies the second transverse sensitivity coefficient
TransverseSensitivity3	0	Specifies the third transverse sensitivity coefficient
GaugePoissonRatio	0.285	Specifies the Poisson ratio for the gauge material

The Poisson ratio for the gauge is also required to calculate the transverse sensitivity compensation. This is also available from the manufacturer.

The equations used for the transverse sensitivity correction are as follows; note that in the equation the K values are fractional—the percentage values entered by the user are divided by 100.

In these equations the following nomenclature applies:

ε_1' – Gauge (1) strain corrected for transverse sensitivity

ε_2' – Gauge (2) strain corrected for transverse sensitivity

ε_3' – Gauge (3) strain corrected for transverse sensitivity

K_t – transverse sensitivity factor

K_{t1} – transverse sensitivity factor for gauge (1)

K_{t2} – transverse sensitivity factor for gauge (2)

K_{t3} – transverse sensitivity factor for gauge (3)

υ_0 – Gauge Poisson ratio

Stacked Tee Rosette

$$\varepsilon_1' = \frac{(1 - \upsilon_0 K_t)(\varepsilon_1 - K_t \varepsilon_2)}{1 - K_t^2}$$

$$\varepsilon_2' = \frac{(1 - \upsilon_0 K_t)(\varepsilon_2 - K_t \varepsilon_1)}{1 - K_t^2}$$

Planar Tee Rosette

$$\varepsilon_1' = \frac{\varepsilon_1(1 - \upsilon_0 K_{t1}) - K_{t1}\varepsilon_2(1 - \upsilon_0 K_{t2})}{1 - K_{t1}K_{t2}}$$

$$\varepsilon_2' = \frac{\varepsilon_1(1 - \upsilon_0 K_{t2}) - K_{t2}\varepsilon_1(1 - \upsilon_0 K_{t1})}{1 - K_{t1}K_{t2}}$$

Stacked Rectangular Rosette

$$K_{t1} = K_{t2} = K_{t3} = K_t$$

$$\varepsilon_1' = \frac{(1 - \upsilon_0 K_t)(\varepsilon_1 - K_t \varepsilon_3)}{1 - K_t^2}$$

$$\varepsilon_2' = \frac{(1-\upsilon_0 K_t)(\varepsilon_2 - K_t(\varepsilon_1+\varepsilon_3-\varepsilon_2))}{1-K_t^2}$$

$$\varepsilon_3' = \frac{(1-\upsilon_0 K_t)(\varepsilon_3 - K_t\varepsilon_1)}{1-K_t^2}$$

Planar Rectangular Rosette

When the transverse sensitivities of the orthogonal gauges (1) and (3) are nominally the same:

$$K_{t1} = K_{t3} = K_{t13}$$

$$\varepsilon_1' = \frac{(1-\upsilon_0 K_{t13})(\varepsilon_1 - K_{t13}\varepsilon_3)}{1-K_{t13}^2}$$

$$\varepsilon_2' = \frac{(1-\upsilon_0 K_{t13})(1+K_{t13})\varepsilon_2 - K_{t13}(1-\upsilon_0 K_{t13})(\varepsilon_1+\varepsilon_3)}{(1+K_{t13})(1-K_{t2})}$$

$$\varepsilon_3' = \frac{(1-\upsilon_0 K_{t13})(\varepsilon_3 - K_{t13}\varepsilon_1)}{1-K_{t13}^2}$$

In the case where all three transverse sensitivities are dissimilar:

$$\varepsilon_1' = \frac{\varepsilon_1(1-\upsilon_0 K_{t1}) - K_{t1}\varepsilon_3\,(1-\upsilon_0 K_{t3})}{1-K_{t1}K_{t3}}$$

$$\varepsilon_2' = \frac{\varepsilon_2(1-\upsilon_0 K_{t2})}{1-K_{t2}} - \frac{K_{t2}\varepsilon_1(1-\upsilon_0 K_{t1})(1-K_{t3}) + \varepsilon_3(1-\upsilon_0 K_{t3})(1-K_{t1})}{(1-K_{t1}K_{t3})(1-K_{t2})}$$

$$\varepsilon_3' = \frac{\varepsilon_3(1-\upsilon_0 K_{t3}) - K_{t3}\varepsilon_1(1-\upsilon_0 K_{t1})}{1-K_{t1}K_{t3}}$$

Stacked Delta Rosette

$$K_{t1} = K_{t2} = K_{t3} = K_t$$

$$\varepsilon_1' = \frac{(1 - \upsilon_0 K_t)}{1 - K_t^2}\left[\left(1 - \frac{K_t}{3}\right)\varepsilon_1 + \frac{2}{3}K_t(\varepsilon_2 + \varepsilon_3)\right]$$

$$\varepsilon_2' = \frac{(1 - \upsilon_0 K_t)}{1 - K_t^2}\left[\left(1 - \frac{K_t}{3}\right)\varepsilon_2 + \frac{2}{3}K_t(\varepsilon_3 + \varepsilon_1)\right]$$

$$\varepsilon_3' = \frac{(1 - \upsilon_0 K_t)}{1 - K_t^2}\left[\left(1 - \frac{K_t}{3}\right)\varepsilon_3 + \frac{2}{3}K_t(\varepsilon_1 + \varepsilon_2)\right]$$

Planar Delta Rosette

When the transverse sensitivity of two of the gauges, (1) and (3) are nominally the same:

$$K_{t1} = K_{t3} = K_{t13}$$

Then the strain ε_2' can be corrected using the simplified equation

$$\varepsilon_2' = \frac{\varepsilon_2(1 - \upsilon_0 K_{t2})(3 + K_{t13}) - 2K_{t2}[(\varepsilon_1 + \varepsilon_3)(1 - \upsilon_0 K_{t13})]}{K_{t13} - 3K_{t13}K_{t2} - K_{t2} + 3}$$

The other strains ε_1' and ε_3' can use the full equations below (for the dissimilar case).

In the case where all three transverse sensitivities are dissimilar:

$$\varepsilon_1' = \frac{\varepsilon_1(1 - \upsilon_0 K_{t1})(3 - K_{t2} - K_{t3} - K_{t2}K_{t3}) - 2K_{t1}[\varepsilon_2(1 - \upsilon_0 K_{t2})(1 - K_{t3}) + \varepsilon_3(1 - \upsilon_0 K_{t3})(1 - K_{t2})]}{3K_{t1}K_{t2}K_{t3} - K_{t1}K_{t2} - K_{t2}K_{t3} - K_{t1}K_{t3} - K_{t1} - K_{t2} - K_{t3} + 3}$$

$$\varepsilon_2' = \frac{\varepsilon_2(1 - \upsilon_0 K_{t2})(3 - K_{t3} - K_{t1} - K_{t3}K_{t1}) - 2K_{t2}[\varepsilon_1(1 - \upsilon_0 K_{t3})(1 - K_{t1}) + \varepsilon_3(1 - \upsilon_0 K_{t1})(1 - K_{t3})]}{3K_{t1}K_{t2}K_{t3} - K_{t1}K_{t2} - K_{t2}K_{t3} - K_{t1}K_{t3} - K_{t1} - K_{t2} - K_{t3} + 3}$$

$$\varepsilon_3' = \frac{\varepsilon_3(1 - \upsilon_0 K_{t3})(3 - K_{t1} - K_{t2} - K_{t1}K_{t2}) - 2K_{t3}[\varepsilon_1(1 - \upsilon_0 K_{t1})(1 - K_{t2}) + \varepsilon_2(1 - \upsilon_0 K_{t2})(1 - K_{t1})]}{3K_{t1}K_{t2}K_{t3} - K_{t1}K_{t2} - K_{t2}K_{t3} - K_{t1}K_{t3} - K_{t1} - K_{t2} - K_{t3} + 3}$$

5.6 Strain Outputs

Based on the rosette input, the glyph calculates the following scalar strain quantities per rosette:

5.6.1 Principal Strain

These equations give the values for maximum principal if you add the square root term, and the minimum principal if you subtract it. The absolute maximum principal strain is the selection from min and max that has the largest absolute value. For example, if the values were 1000 uE and -500uE, then the absolute maximum would be 1000uE. If the values were 500uE and -1000uE, then the value would be -1000uE.

Rectangular Rosette

$$\varepsilon_{pq} = \frac{\varepsilon_1 + \varepsilon_3}{2} \pm \frac{1}{\sqrt{2}} \sqrt{(\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2}$$

Delta Rosette

$$\varepsilon_{pq} = \frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3}{3} \pm \frac{\sqrt{2}}{3} \sqrt{(\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_3 - \varepsilon_1)^2}$$

Tee Rosette

$$\varepsilon_p = \varepsilon_1$$

$$\varepsilon_q = \varepsilon_2$$

5.6.2 Shear Strain

The maximum shear strain is calculated as

$$\gamma_{max} = \varepsilon_p - \varepsilon_q$$

To get the “signed” shear strain, the sign is taken from the sign of the absolute largest of the principals.

5.6.3 Von Mises Strain

To calculate Von Mises strain, it is necessary to calculate the tensor strains, and to do this requires a value of Poisson’s ratio, υ. This can be entered as a property

PoissonRatio	0.3

Note that this is the material Poisson ratio and is different from that used for transverse sensitivity corrections (the gauge Poisson ratio).

Rectangular Rosette

$$\varepsilon_x = \varepsilon_1$$

$$\varepsilon_y = \varepsilon_3$$

Delta Rosette

$$\varepsilon_x = \varepsilon_1$$

$$\varepsilon_y = \frac{-\varepsilon_1 + 2\varepsilon_2 + 2\varepsilon_3}{3}$$

Tee Rosette

$$\varepsilon_x = \varepsilon_1$$

$$\varepsilon_y = \varepsilon_2$$

Then for all types

$$\varepsilon_z = \frac{-\upsilon(\varepsilon_x + \varepsilon_y)}{1 - \upsilon}$$

And

$$\varepsilon_{VM} = \frac{\sqrt{(\varepsilon_p - \varepsilon_q)^2 + (\varepsilon_p - \varepsilon_z)^2 + (\varepsilon_q - \varepsilon_z)^2}}{\sqrt{2}\,(1 + \upsilon)}$$

To get the “signed” Von Mises strain, the sign is taken from the sign of the absolute largest of the principals.

5.6.4 Strain at a Specified Angle

Rectangular Rosette

$\varepsilon_x = \varepsilon_1$

$\varepsilon_y = \varepsilon_3$

$\gamma_{xy} = 2\varepsilon_2 - (\varepsilon_1 + \varepsilon_3)$

Delta Rosette

$\varepsilon_x = \varepsilon_1$

$$\varepsilon_y = \frac{-\varepsilon_1 + 2\varepsilon_2 + 2\varepsilon_3}{3}$$

$$\gamma_{xy} = \frac{2(\varepsilon_2 - \varepsilon_3)}{\sqrt{3}}$$

Tee Rosette

$\varepsilon_x = \varepsilon_1$

$\varepsilon_y = \varepsilon_2$

$\gamma_{xy} = 0$

For an angle, θ, the strain resolved to that angle is:

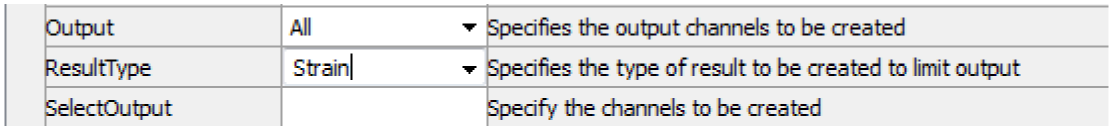
$$\varepsilon_\theta = \frac{\varepsilon_x + \varepsilon_y}{2} + \frac{(\varepsilon_x - \varepsilon_y)\cos(2\theta)}{2} + \frac{\gamma_{xy}\sin(2\theta)}{2}$$

5.6.5 Critical Plane Strains

Critical plane stresses are output at 10-degree increments from grid 1 from 0 degrees to 170 degrees using the formula specified in "Strain at a Specified Angle" above.

5.6.6 Selecting the Strain Values to Output

The glyph offers a user interface to help with the selection—see Figure 5-3 below. This GUI sets, and has its defaults set by, the following properties:



If All results are output, or ResolveToAngle is selected, then the following angle properties are also used:

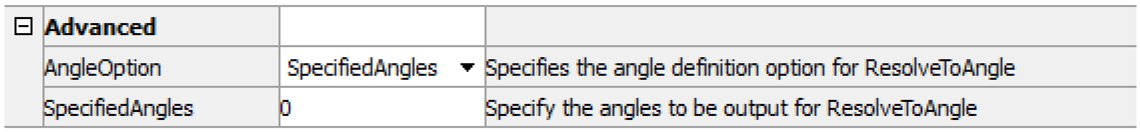
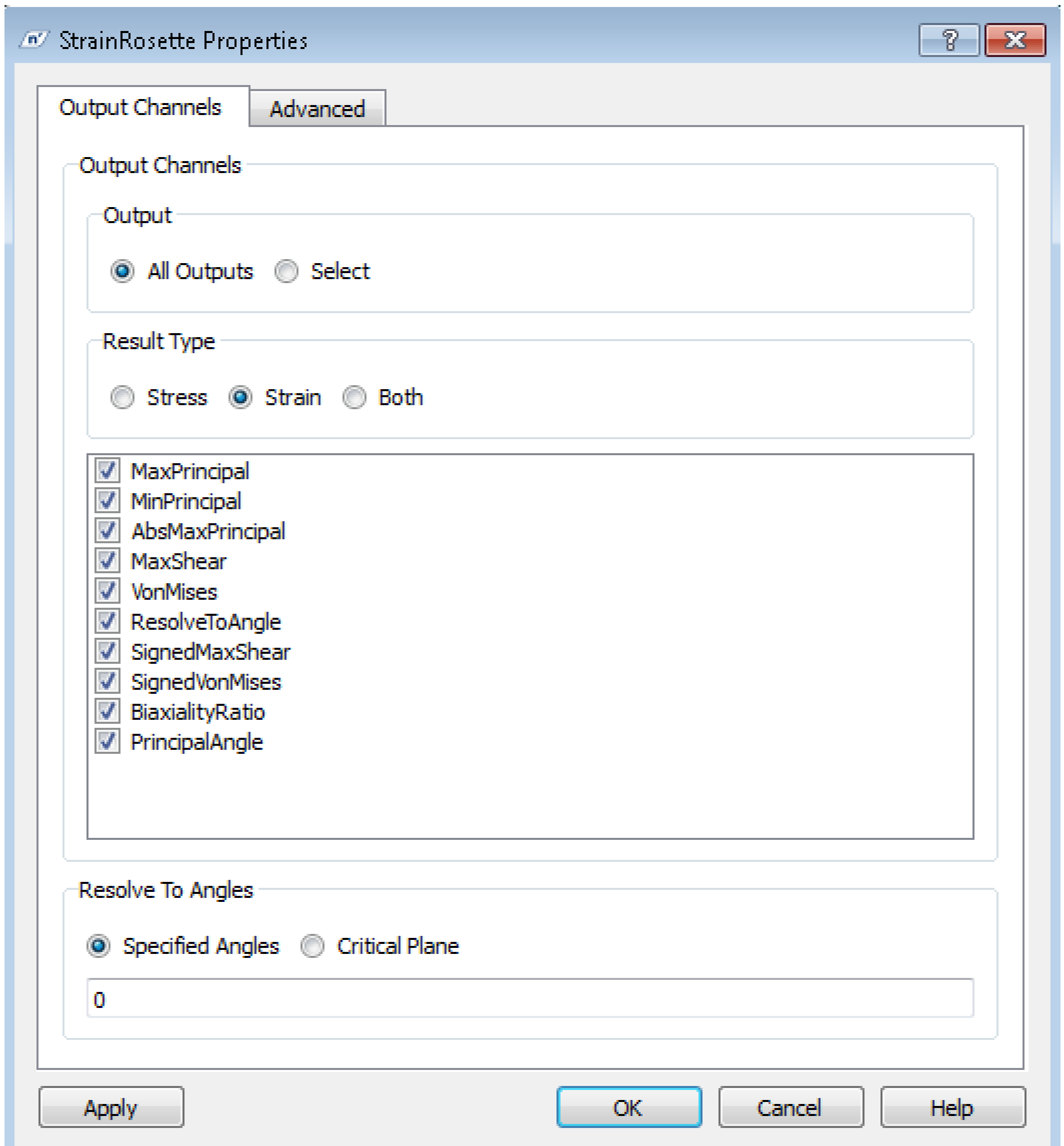


Fig. 5-3 GUI for selection of output channels



See the Glyph Reference Guide for details on these property settings.

5.7 Stress Outputs

Based on the rosette input, the glyph calculates the following scalar stress quantities per rosette:

5.7.1 Principal Stress

These equations give the values for maximum principal if you add the square root term, and minimum principal if you subtract it. The absolute maximum principal stress is the selection from min and max that has the largest absolute value. For example, if the values were 1000MPa and -500MPa, then the absolute maximum would be 1000MPa. If the values were 500uMPa and -1000MPa, then the value would be -1000MPa.

The values for E and υ are required and can be set using the following properties.

Stress Conversion	
StressUnits	MPa
YoungsModulus	2.05E5
PoissonRatio	0.3

The stress units are the units in which E is entered, and therefore the units in which the output stresses are written.

Rectangular Rosette

$$\sigma_{pq}= \frac{E}{2}\left(\frac{\varepsilon_1 + \varepsilon_3}{1-\upsilon} \pm \frac{1}{1+\upsilon}\sqrt{(\varepsilon_1-\varepsilon_2)^2+(\varepsilon_2-\varepsilon_3)^2}\right)$$

Delta Rosette

$$\sigma_{pq}= \frac{E}{3}\left(\frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3}{1-\upsilon} \pm \frac{\sqrt{2}}{1+\upsilon}\sqrt{(\varepsilon_1-\varepsilon_2)^2+(\varepsilon_2-\varepsilon_3)^2+(\varepsilon_3-\varepsilon_1)^2}\right)$$

Tee Rosette

$$\sigma_p = \frac{E}{1-\upsilon^2}(\varepsilon_1 + \upsilon\varepsilon_2)$$

$$\sigma_q = \frac{E}{1-\upsilon^2}(\varepsilon_2 + \upsilon\varepsilon_1)$$

5.7.2 Shear Stress

$$\tau_{max} = \frac{\sigma_p - \sigma_q}{2}$$

To get the “signed” shear stress, the sign is taken from the sign of the absolute largest of the principals.

5.7.3 Von Mises Stress

The Von Mises stress, under plane stress conditions, is:

$$\sigma_{VM} = \sqrt{\sigma_p{}^2 - \sigma_p\sigma_q + \sigma_q{}^2}$$

To get the “signed” Von Mises stress, the sign is taken from the sign of the absolute largest of the principals.

5.7.4 Stress at a Specified Angle

First of all, the strains, ε_x, ε_y, ε_z and γ_{xy} are calculated using these equations:

Rectangular Rosette

$$\varepsilon_x = \varepsilon_1$$

$$\varepsilon_y = \varepsilon_3$$

$$\gamma_{xy} = 2\varepsilon_2 - (\varepsilon_1 + \varepsilon_3)$$

Delta Rosette

$$\varepsilon_x = \varepsilon_1$$

$$\varepsilon_y = \frac{-\varepsilon_1 + 2\varepsilon_2 + 2\varepsilon_3}{3}$$

$$\gamma_{xy} = \frac{2(s_2 - s_3)}{\sqrt{3}}$$

Tee Rosette

$\varepsilon_x = \varepsilon_1$

$\varepsilon_y = \varepsilon_2$

$\gamma_{xy} = 0$

Then for all types

$$\varepsilon_z = \frac{-\upsilon(\varepsilon_x + \varepsilon_y)}{1 - \upsilon}$$

For an angle, θ, the strains on x and y resolved to that angle are:

$$\varepsilon_{\theta x} = \frac{\varepsilon_x + \varepsilon_y}{2} + \frac{(\varepsilon_x - \varepsilon_y)\cos(2\theta)}{2} + \frac{\gamma_{xy}\cos(2\theta)}{2}$$

$$\varepsilon_{\theta y} = \frac{\varepsilon_x + \varepsilon_y}{2} - \frac{(\varepsilon_x - \varepsilon_y)\cos(2\theta)}{2} + \frac{\gamma_{xy}\sin(2\theta)}{2}$$

And the stress in the x direction at that angle is:

$$\sigma_\theta = \left(\frac{\upsilon E}{(1+\upsilon)(1-2\upsilon)}\right)(\varepsilon_{\theta x} + \varepsilon_{\theta y} + \varepsilon_z) + \left(\frac{E\varepsilon_{\theta x}}{1+\upsilon}\right)$$

Note that if strains are in microstrain, the value must be divided by 1e6.

5.7.5 Critical Plane Stresses

Critical plane stresses are output at 10-degree increments from grid 1 from 0 degrees to 170 degrees using the formula specified in "Stress at a Specified Angle" above.

5.7.6 Selecting Stress Values to Output

Stress values are set in the same way that strain values are set. This is described in "Selecting the Strain Values to Output" above.

It is possible to output sets of both stress and strain time histories for each rosette, or just strains, or just stresses.

5.8 Biaxiality Ratio

The stress biaxiality ratio is the ratio of the minimum principal stress to the maximum principal stress. It can take on values which range from -1 (pure shear), through zero (uniaxial), to +1 (equibiaxial) loading.

5.9 Angle of Principal Stress/Strain

Angle between Grid 1 and the Maximum Principal

Rectangular Rosette

$$\phi_{pq} = \frac{1}{2} tan^{-1}\left[\frac{2\varepsilon_2 - \varepsilon_1 - \varepsilon_3}{\varepsilon_1 - \varepsilon_3}\right]$$

Delta Rosette

$$\phi_{pq} = \frac{1}{2} tan^{-1}\left[\frac{\sqrt{3}(\varepsilon_2 - \varepsilon_3)}{2\varepsilon_1 - \varepsilon_2 - \varepsilon_3}\right]$$

The angle between grid 1 and the maximum principal strain of a Tee rosette is either 0 or 90 degrees depending upon whether the maximum is e1 or e2.

The user can also choose to output the angle to the absolute maximum principal.

OutputAngleType	MaxPrincipal	Specifies the output angle type

OutputAngleType	AbsMaxPrin	Specifies the output angle type

The angle to the absolute maximum is either the angle to the maximum, as calculated above, or the angle to the minimum, which is 90 degrees from the maximum.

The variation of this angle with time through a loading history provides a valuable insight into the nature of the loading environment. It is sometimes useful to carry out a time at level analysis of the angle file in order to gauge angular stability.

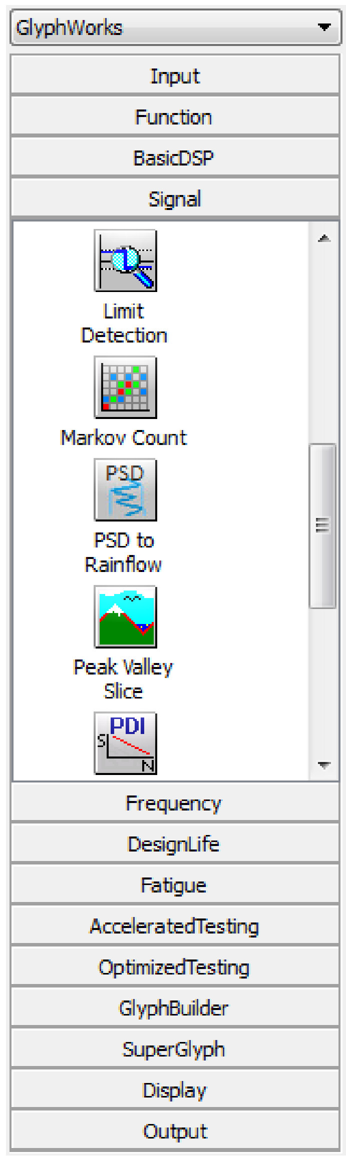
6 PSD to Rainflow

In order to calculate fatigue life from frequency information, a method of converting spectral information into rainflow counts is required. The PSDToRainflow glyph is provided within GlyphWorks to achieve this. This section provides details of the theory behind the four methods that the glyph offers, together with an explanation of the glyph properties.

6.1 PSD To Rainflow Glyph

The glyph can be found on the Signal palette.

Fig. 6-1 Signal palette



6.2 Input

The glyph accepts a histogram that must be either specifically a PSD, or a generic 2D histogram containing PSD data. In either case, the bins must be evenly spaced.

See the Methods section for details on the constraints that are placed on the input PSD to enable the theory to be sound.

6.3 Outputs

The glyph generates a 2D Range-Only rainflow matrix and also outputs a PSD that is the result of scaling and transforming the input PSD.

6.4 Rainflow Output

The rainflow output is controlled by the following properties:

NumberOfBins	64	Sets the number of bins in the output histogram

The number of bins in the matrix affects the accuracy of any fatigue calculation performed on the histogram data. The more bins there are, the more accurate the result typically is.

ClippingLimit	0	Maximum allowed level in the rainflow histogram

The Clipping limit defines the maximum range value in the output histogram. If this is set to 0, the glyph will automatically determine the maximum level based on the largest-sized cycle that would statistically occur in a 10,000 year period.

Gate	0	Cycles whose range is less than the gate are not included in the output.

The Gate property is used to exclude small cycles. The bin that the gate value falls in is calculated and this bin, together with all those below it, has its count set to zero. Any fatigue calculation will therefore have no damage accumulated in these bins.

DurationSource	MetaData	Specifies where to find the duration of the input data.

The DurationSource property is used to determine the length of time, in seconds, that the output histogram represents. The matrix is a statistical representation of the cycles and their relative likelihood of occurrence. The count of cycles in each bin is linearly related to the length of time that the matrix represents. The matrix that represents 10 minutes has 10 times the counts for a matrix representing 1 minute.

This doesn't affect the fatigue life calculation—the damage for one repeat changes, the number of repeats to failure changes, but the time to failure remains constant.

If the source is "Metadata", then the duration of the data from which the PSD was generated is obtained from the metadata of the input PSD.

If the source is "Custom", then the user can provide the time using the following properties:

CustomDuration		
DurationSeconds	1	Exposure duration in seconds
DurationHours	0	Exposure duration in hours
DurationDays	0	Exposure duration in days
DurationYears	0	Exposure duration in years

The total time in seconds is obtained by adding these values together:

$$\begin{aligned} Time &= DurationSeconds + (3600 * DurationHours) + (86400 * DurationDays) \\ &+ (31536000 * DurationYears) \end{aligned}$$

A year is approximated to 365 days.

In the absence of any transformations of the PSD (see Scaling and Transforming the PSD), the units of the X axis of the rainflow matrix are derived from the units of the Y axis on the input PSD by removing the ^2.Hz-1 – hence “mm^2.Hz-1” becomes “mm”. The same applies if the ZUnits property is used to override the units string of the PSD.

6.5 Scaling and Transforming the PSD

The output PSD is derived from the input PSD and is changed by the following properties:

PSDScaling	1	Scaling value applied to the input PSD

This value allows the user to scale the input PSD. This is a linear scaling – the glyph will apply the “square” of this number to the PSD automatically. Hence to convert a PSD from being based on ”m/s^2” to “g”, the value of 9.81 can be put in the property.

Integration	None	Integration or differentiation option

If the input data is acceleration, but the fatigue calculation is based on, say, displacement, or strain derived from displacement, then the data must be integrated (twice) to transform it correctly.

The options are integration, double integration, differentiation and double differentiation. The transformations are achieved by multiplying/dividing by omega^2:

For a frequency f:

Integration

$$PSD_f = \frac{PSD_f}{(2\pi f)^2}$$

Double Integration

$$PSD_f = \frac{PSD_f}{(2\pi f)^4}$$

Differentiation

$$PSD_f = PSD_f(2\pi f)^2$$

Double Differentiation

$$PSD_f = PSD_f(2\pi f)^4$$

Note that the PSDScaling is applied before the integration/differentiation.

⊟ Geometry		
Kf	1	Fatigue concentration factor

The properties ZTitle and ZUnits can be used to override the default titles on the output PSD. The default is to copy the input title and units, or attempt to convert the units if one of the integration or differentiation options are used. However, if a scaling factor is used (e.g., to convert from “m/s^2” to “g”), the program cannot determine this automatically so the user gets the opportunity to change the text. Note that in a multi-channel input, all the channels will be changed.

If PSDScaling=1, Integration=None, and ZTitle and ZUnits are blank, then the output data will be identical to the input data.

6.6 Methods

RainflowMethod	Lalanne	Specifies how the rainflow count is obtained from the PSD

This section describes four approaches for computing rainflow cycle ranges directly from a PSD of stress as opposed to a time signal. For more background information, see Bishop and Sherratt [1], Lalanne [2] and Halfpenny [3],[4].

6.6.1 Narrow-band Approach (Bendat and Rice)

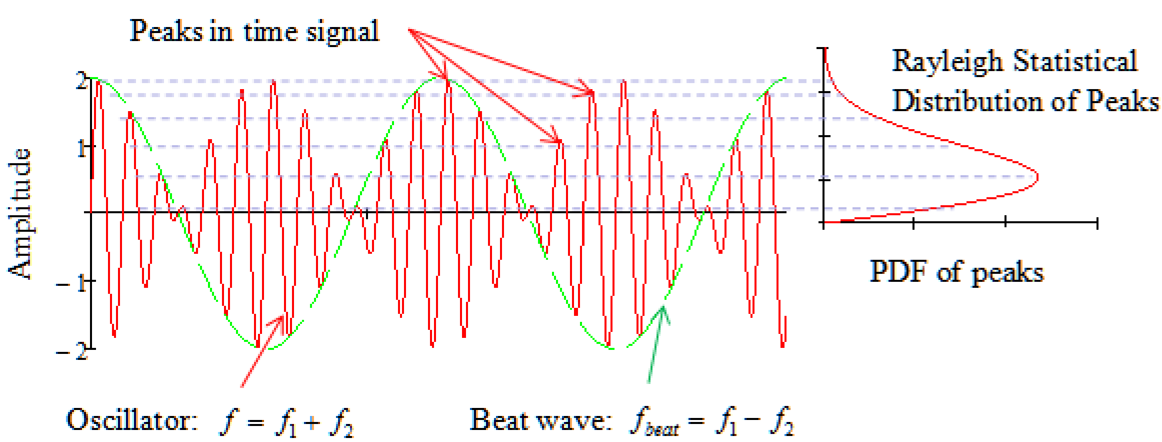
In 1964 Bendat [5] proposed the first significant step towards a method for determining rainflow cycle ranges and hence fatigue life from PSD. Earlier, in 1954, Rice [6] showed that the Probability Density Function (PDF) of peaks for a narrow-band signal tended towards a Rayleigh distribution as the bandwidth reduced. Figure 6-2 illustrates the effect by showing the sum of two sinusoidal waves where the frequencies (f_1 and f_2) are close together (i.e., narrow frequency range). The resulting time signal shows a high-frequency oscillation ($f_1 + f_2$) attenuated by a low frequency beat wave (f_1 - f_2). The oscillation and beat frequencies are clearly seen through classical trigonometry in equation (1).

Equation 1

$$\sin(2\pi f_1 t) + \sin(2\pi f_2 t) = \underbrace{2\sin\left(2\pi \frac{f_1 + f_2}{2} t\right)}_{\text{Oscillator}} \underbrace{\cos\left(2\pi \frac{f_1 - f_2}{2} t\right)}_{\text{Beat wave}}$$

Using this assumption, Bendat reasoned that the PDF of rainflow range (i.e., stress range) would also tend to a Rayleigh distribution. Rainflow range is defined as 2x peak amplitude. Figure 6-2 shows that for narrow-band time signals, each peak is typically followed by a valley of approximately the same amplitude. Therefore, the PDF of rainflow range should be twice the PDF of peaks.

Fig. 6-2 Narrow-band time signal showing "beat" wave and Rayleigh PDF of peaks



To complete his solution, Bendat used a series of equations derived by Rice [6] to estimate the expected number of cycles occurring per second of exposure. Using moments of area under the PSD, Rice derived formulae to estimate the number of peaks per second (or peak rate) $E[P]$, and the estimated number of zero up-crossings per second $E[0]$. His formulae are given in equation (2).

Equation 2

$$E[0] = \sqrt{\frac{m_2}{m_0}} \qquad E[P] = \sqrt{\frac{m_4}{m_2}}$$

m_0, m_2 and m_4 are the 0^{th}, 2^{nd} and 4^{th} moments of area of the PSD about the zero *Hz* axis. The n^{th} moment of area is defined by equation (3).

Note	m_0 is equal to the area under the PSD which also represents the "mean square" value of the time signal or the square of the RMS [Root Mean Square].

Equation 3

$$m_n = \int_0^\infty f^n \cdot G(f)df$$

G(f) is the single-sided PSD of stress amplitude at frequency f Hz.

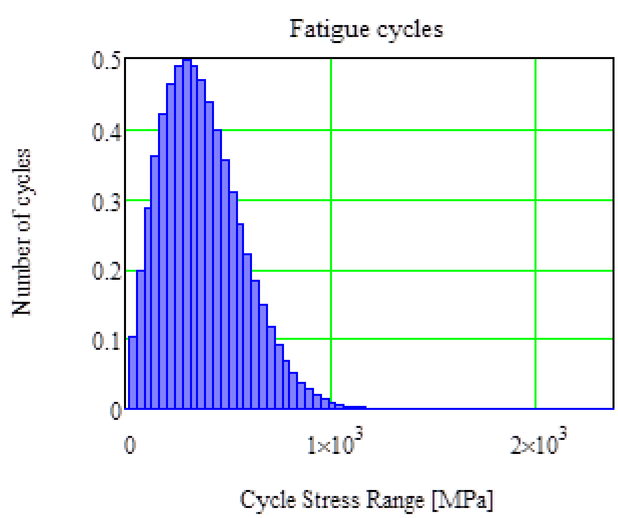
Bendat's narrow-band solution for the rainflow range histogram is therefore determined from the PDF expressed in equation (4).

Equation 4

$$\mathrm{N(S) = E[P] \cdot T} \cdot \left\{ \frac{\mathrm{S}}{4\mathrm{m_0}} \cdot \mathrm{e}^{\frac{-S^2}{8\mathrm{m_0}}} \right\}$$

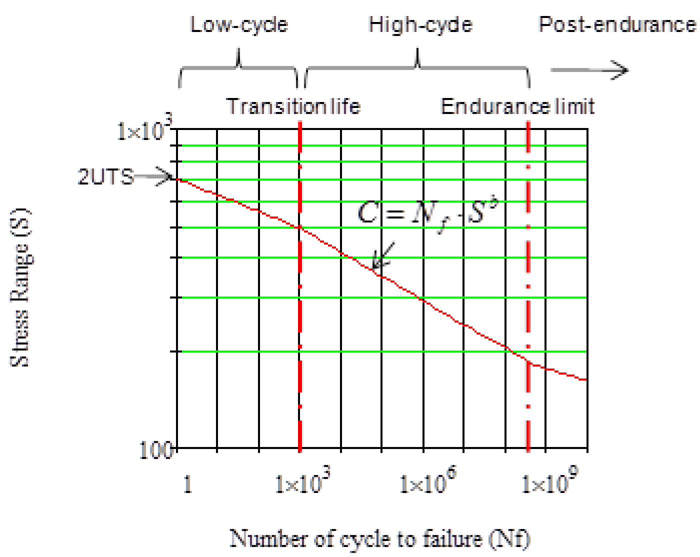
The term in brackets in equation (4) is the Raleigh probability distribution. *N* is the expected number of cycles of stress range *S* occurring in *T* seconds. *E*[*P*] is the expected number of peaks obtained by equation (2). An illustration of a typical rainflow distribution using Bendat's narrow-band approach is shown in Figure 6-3.

Fig. 6-3 Rainflow cycle histogram derived from PSD using narrow-band approach



The fatigue damage is determined in exactly the same way as with traditional time-domain rainflow cycle counting. The damage from each cycle is determined from the fatigue life curve (SN curve) and the damage is then summed linearly using Palmgren-Minor [7],[8] damage accumulation laws. A typical SN fatigue curve for aluminium alloy 6082 in the T6 condition is illustrated in Figure 6-4 using a 3 segment SN curve. Each segment is represented as a straight line in log space as described by the Basquin power-law relationship given in equation (5). The PSDToRainflow glyph is designed to interface directly with the GlyphWorks Stress Life fatigue glyph to perform damage analysis. Simple relative damage analysis, using a single slope SN curve, is incorporated within the PSDToRainflow glyph.

Fig. 6-4 SN curve for 6082-T6 aluminium alloy



Equation 5

$$C = N_f \cdot S^b$$

S is the stress range in *MPa*, *Nf* is the number of rainflow cycles to failure, *C* is the Basquin coefficient (intercept of the SN curve with the Stress axis) and *b* is the Basquin exponent (gradient of the SN curve in log space). Fatigue damage (and hence fatigue life) is determined by equation (6).

Equation 6

$$D = \frac{1}{N_f} = \frac{1}{C}\int_0^{2UTS} N(S) \cdot S^b ds$$

D is the fatigue damage ratio. If $D \geq 1$, then the component is likely to fail within the test duration T. If $D<1$, then the fatigue life can be determined as T/D in seconds.

The rainflow cycle histogram derived by Bendat is an infinite exponential distribution. It therefore contains stress ranges which can exceed UTS (Ultimate Tensile Stress). These stresses may have a very low probability of occurrence; however, the engineer must check to ensure that they do not compromise the accuracy of the fatigue calculation.

Fatigue damage is calculated in a conventional manner [as per equation (6)] for stresses less than UTS. When the stress exceeds UTS, then we assume that failure is unavoidable and so the damage $\left(\frac{S^b}{C}\right)$ is set to 1.0. Therefore, for stresses greater than UTS, the rainflow histogram represents the probability of failure by exceeding UTS. This approach is analogous to the statistical "hazard" function. The expected failure rate due to an exceedence of UTS is estimated by equation (7).

Equation 7

$$D = \int_{2UTS}^{\infty} N(S)\, ds$$

By merging these two principles, we obtain an estimate of the expected reliable life considering failure from both fatigue and exceeding UTS as per equation (8). The Stress Life fatigue glyph is designed to work in conjunction with the output from the PSDToRainflow glyph and damage is accumulated using equation (8).

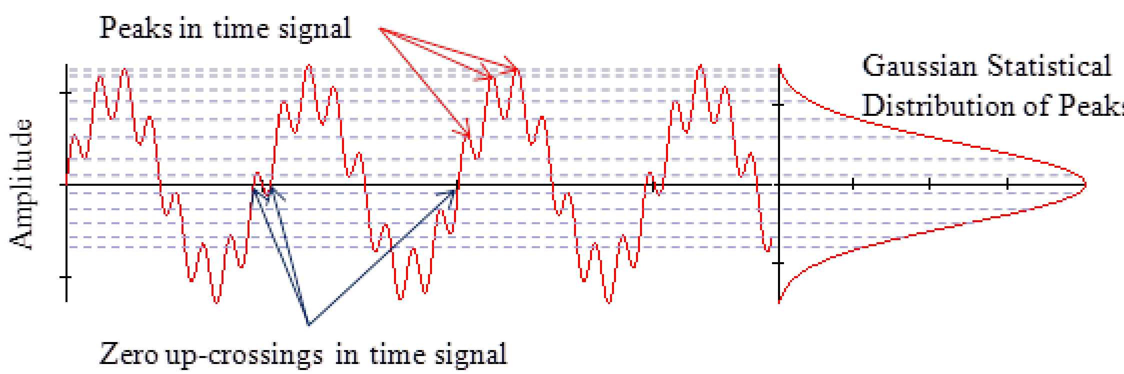
Equation 8

$$D = \frac{1}{C}\int_{0}^{2UTS} N(S) \cdot S^b dS + \int_{2UTS}^{\infty} N(S)\, dS$$

6.6.2 Broad-band Approach (Steinberg)

Bendat's narrow-band solution tends to be conservative for broad-band signals. Figure 6-5 illustrates a typical broad-band signal represented by a high-frequency sinusoidal wave superimposed on a low-frequency carrier. As the bandwidth increases, then the PDF of peaks becomes symmetric and tends to a Gaussian distribution, as illustrated in the figure. Rice [6] therefore concludes that the actual PDF of peaks depends on the bandwidth of the signal. Narrow-band signals tend to a Rayleigh distribution, whereas broad-band signals tend to a Gaussian distribution.

Fig. 6-5 Broad-band time signal and Gaussian PDF of peaks



A reliable measure of bandwidth was offered by Rice [6] as a ratio of the number of zero up-crossings in a time signal, to the number of peaks. This ratio is often known as the "irregularity factor γ" and is given by equation (9). For narrow-band signals, the irregularity factor tends to unity (all peaks occur above the mean), whereas broad-band signals progressively tend to zero (peaks occur symmetrically about the mean).

Equation 9

$$\gamma = \frac{E[0]}{E[P]} = \frac{m_2}{\sqrt{m_0 \cdot m_4}}$$

Steinberg [9] assumed that for broad-band signals the PDF of rainflow range would also tend to a Gaussian distribution and proposed a solution based on discrete multiples of the RMS (Root Mean Square) amplitude. This approach is used extensively in the field of electronics testing and the Steinberg equation for rainflow ranges is shown in equation (10). The expected number of cycles, *N*(*S*), are calculated at *S*=2RMS, *S*=4RMS and *S*=6RMS amplitudes.

Equation 10

$$N\begin{bmatrix} 2\sqrt{m_0} \\ 4\sqrt{m_0} \\ 6\sqrt{m_0} \end{bmatrix} = E[0] \cdot T \cdot \begin{bmatrix} 0.683 \\ 0.271 \\ 0.043 \end{bmatrix}$$

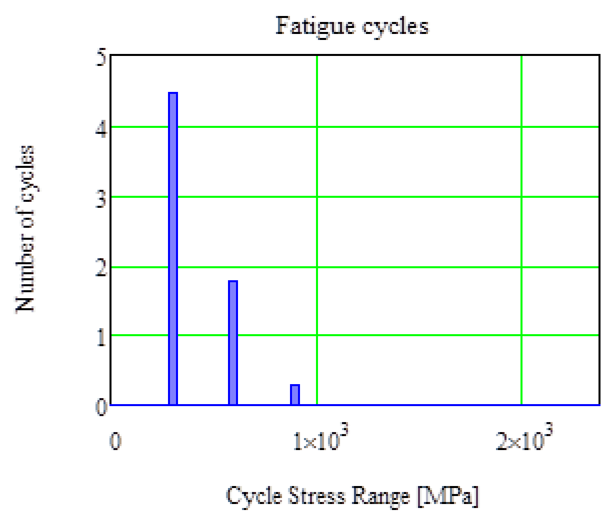
Steinberg does not give a continuous PDF of rainflow range like the other methods. It is defined only at the first 3 multiples of the RMS, as illustrated in Figure 6-6. This simplification makes Steinberg ideal for hand calculation but renders the approach unsatisfactory for anything other than single slope SN curves because the discrete rainflow ranges tend to fall arbitrarily on different segments of the SN curve.

Fatigue damage (and hence life) can be obtained by substituting equation (10) in equation (6) and simplifying. The result is given in equation (11).

Equation 11

$$D = \frac{1}{N_f} = \frac{E[0] \cdot T}{C} \cdot \left\{ 0.683 \times \left(2\sqrt{m_0}\right)^b + 0.271 \times \left(4\sqrt{m_0}\right)^b + 0.043 \times \left(6\sqrt{m_0}\right)^b \right\}$$

Fig. 6-6 Rainflow cycle histogram derived from PSD using Steinberg approach



6.6.3 General Approach for All Bandwidths (Dirlik)

Rice [6] concluded that for a signal of arbitrary bandwidth, the PDF of peaks could be obtained from the weighted sum of the Rayleigh and Gaussian distributions. However, Dirlik [10] reasoned that the PDF of peaks is not the same as the PDF of rainflow range. In 1985, he proposed an empirical solution to estimate the PDF of rainflow range following extensive computer simulations using the Monte Carlo technique. The Dirlik formulation is given in equations (12) and (13).

Equation 12

$$N(S) = E[P] \cdot T \cdot p(S)$$

Where $N(S)$ is the number of rainflow cycles of stress range S *MPa*, expected in time T *sec*. $E[P]$ is the expected number of peaks obtained by equation (2).

Equation 13

$$p(S) = \frac{\frac{D_1}{Q} \cdot e^{\frac{-Z}{Q}} + \frac{D_2 \cdot Z}{R^2} \cdot e^{\frac{-Z^2}{2R^2}} + D_3 \cdot Z \cdot e^{\frac{-Z^2}{2}}}{2\sqrt{m_0}}$$

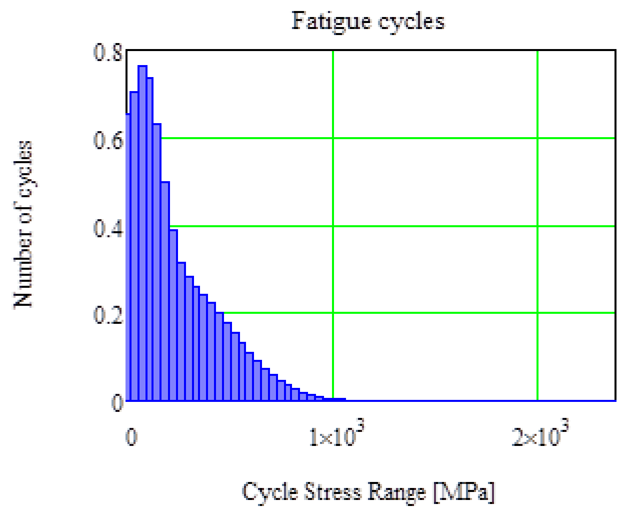
$$D_1 = \frac{2(x_m - \gamma^2)}{1+\gamma^2} \qquad D_2 = \frac{1-\gamma-D_1+D_1{}^2}{1-R} \qquad D_3 = 1 - D_1 - D_2$$

$$Z = \frac{S}{2\sqrt{m_0}} \qquad Q = \frac{1.25(\gamma - D_3 - D_2 \cdot R)}{D_1} \qquad R = \frac{\gamma - x_m - D_1{}^2}{1-\gamma-D_1+D_1{}^2}$$

$$\gamma = \frac{m_2}{\sqrt{m_0 \cdot m_4}} \qquad x_m = \frac{m_1}{m_0} \cdot \sqrt{\frac{m_2}{m_4}}$$

The Dirlik equation is based on the weighted sum of the Rayleigh, Gaussian and exponential probability distributions. In terms of accuracy, Dirlik's empirical formula for rainflow ranges has been shown to be far superior to the earlier methods—see Bishop[11]. Dirlik's Rainflow histogram is shown in Figure 6-7 and the fatigue damage (and hence life) is derived by substituting equation (12) into equation (8) and solving numerically.

Fig. 6-7 Rainflow cycle histogram derived from PSD using the Dirlik approach



6.6.4 Lalanne/Rice Approach

In contradiction to Dirlik, Lalanne [2] reasoned that over a sufficiently long period of time, the PDF of rainflow range would tend to the PDF of peaks, and thereby demonstrated that Rice's original formula (based on a simple weighted sum of the Rayleigh and Gaussian distributions) would also suffice for rainflow ranges. The Lalanne/Rice formula is given in equations (14) and (15).

Equation 14

$$N(S) = E[P] \cdot T \cdot p(S)$$

Where *N*(*S*) is the number of stress cycles of range *S MPa*, expected in time *T* sec. *E*[*P*] is the expected number of peaks obtained by equation (2).

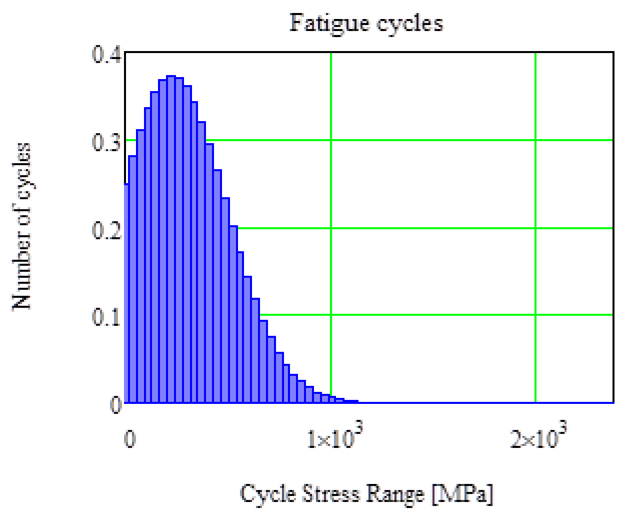
Equation 15

$$p(S) = \frac{1}{2\sqrt{m_0}}\left\{\frac{\sqrt{1-\gamma^2}}{\sqrt{2\pi}}e^{\frac{-S^2}{8m_0(1-\gamma^2)}} + \frac{S\cdot\gamma}{4\sqrt{m_0}}e^{\frac{-S^2}{8m_0}}\left[1+erf\left(\frac{S\cdot\gamma}{\sqrt{8m_0(1-\gamma^2)}}\right)\right]\right\}$$

Where γ is the irregularity factor determined from equation (9), and *erf*(*x*) is the Gauss error function defined by: $erf(x) = \frac{2}{\sqrt{\pi}} \cdot \int_0^x e^{-t^2} dt$.

The Lalanne/Rice approach is equally robust as the Dirlik method. It gives similar results in most cases and offers the advantage of being less empirical. A typical rainflow histogram obtained using the Lalanne/Rice approach is illustrated in Figure 6-8 and the fatigue damage (and hence life) is derived by substituting equation (14) into equation (8) and solving numerically.

Fig. 6-8 Rainflow cycle histogram derived from PSD using the Lalanne/Rice approach



6.6.5 Conclusion

Four methods for PSD-based rainflow cycle extraction have been reviewed. For a detailed comparison between time signal and PSD cycle counting techniques, please refer to Halfpenny [12]. This paper demonstrates excellent correlation for the Dirlik and Lalanne methods. The Narrow-band method is generally too conservative for broad-band signals but demonstrates excellent correlation for narrow-band signals. The Steinberg method is very popular in the electronics industry and demonstrates good correlation for broad-band signals but is excessively non-conservative for narrow-band signals and shallow SN slopes.

The PSD-based methods discussed here are not appropriate for transient loads or systems with a non-linear dynamic response. The approach assumes that loading can be represented by a PSD. The PSD describes amplitude and frequency content of a signal but does not provide information on phase content. The approach assumes that phase is random. This implies that the underlying process is “Ergodic stationary Gaussian and random” (for more information, see Halfpenny [13]). Where PSD techniques are appropriate, then these techniques are found to offer a more robust and efficient solution than the equivalent reconstructed time signals.

6.6.6 References

[1] NWM Bishop and F Sherratt, Fatigue life prediction from power spectral density data. Environmental Engineering, vol. 2, 1989

[2] C Lalanne, Mechanical vibration and shock, Vol 4, Hermes Penton Science, London, 2009

[3] A Halfpenny and NWM Bishop, Vibration Fatigue, HBM-nCode, UK, 1997

[4] A. Halfpenny, Rainflow Cycle Counting and Fatigue Analysis from PSD, Proceedings of the ASTELAB conference, France, 2007

[5] JS Bendat, Probability functions for random responses, NASA report on contract NAS-5-4590, 1964.

[6] SO Rice, Mathematical analysis of random noise, Selected papers on noise and stochastic processes, Dover, New York, 1954

[7] A Palmgren, Die lebensdauer von kugellagern, zeitschrift des vereinesdeutscher ingenierure, Vol. 68, No. 14, 1924, pp. 339-341, 1924

[8] MA Miner, Cumulative damage in fatigue, J. Applied Mechanics, Vol. 67, 1945, pp. A159-A164, 1945

[9] DS Steinberg, Vibration analysis for electronic equipment, Wiley-Interscience, ISBN 978-0471376859, 2000

[10] T Dirlik, Application of computers to fatigue analysis, PhD Thesis, Warwick University, 1985

[11] NWM Bishop et al, Methods for the rapid evaluation of fatigue damage on the Howden HWP330 wind turbine. Proceedings of 13th British Wind Energy Conference, Swansea, UK, pp 317-321. 1991

[12] A Halfpenny, Rainflow Cycle Counting and Acoustic Fatigue Analysis Techniques for Random Loading, 10th International Conference, RASD 2010, UK

[13] A Halfpenny, What is the Frequency Domain and How Do We Use A PSD?

6.7 Simple Relative Damage Calculation

The glyph includes a simple relative damage calculation that uses the following properties:

RelativeDamage		
SNSlope	-3	The slope of the relative damage S-N curve
SNIntercept	1	The intercept of the relative damage S-N curve

If A is the intercept and b is the slope, the equation for the damage of a cycle of range R is

$$d_R = AR^b$$

The simple calculation uses the output histogram rather than the closed form solutions mentioned in Methods—the range of each bin is calculated using its midpoint and the damage for this range is calculated. The damage is then multiplied by the bin count (note that this is a real number and may contain fractional cycle counts) and added to the total damage.

6.8 Metadata Results

The results of all the intermediate statistics used in the calculation, together with the damage result from the Simple Relative Damage Calculation, are output to the metadata of both the output rainflow matrix and the PSD.

A Material Type Codes

0000	Type undefined
0001	Flake cast iron
0002	Ferritic cast iron with compacted graphite
0003	Pearlitic cast iron with compacted graphite
0004	Bainitic cast iron with compacted graphite
0005	Ferritic cast iron with spheroidal graphite
0006	Ferrite/pearlite cast iron with spheroidal graphite
0007	Pearlitic cast iron with spheroidal graphite
0008	Bainitic cast iron with spheroidal graphite
0009	Cast steel with less than 0.2% carbon
0010	Normalized cast steel with 0.2-0.4% carbon
0011	Quenched & tempered cast steel with 0.2-0.4% carbon
0012	Normalized cast steel with 0.4-0.7% carbon
0013	Plain carbon wrought steel with < 0.2% carbon
0014	Hot rolled/normalized plain carbon wrought steel, 0.2-0.4% carbon
0015	Quenched & tempered cast steel with 0.4-0.7% carbon
0016	Quenched & tempered plain carbon wrought steel, 0.2-0.4% carbon
0017	Hot rolled/normalized plain carbon wrought steel, 0.4-0.7% carbon
0018	Quenched & tempered plain carbon wrought steel, 0.4-0.7% carbon
0019	Normalized low alloy wrought steel
0020	Quenched & tempered low alloy wrought steel
0021	Normalized Ni/Cr/Mo wrought steel
0022	Quenched & tempered Ni/Cr/Mo wrought steel
0023	Austenitic stainless steel
0024	Ferritic stainless steel
0025	Martensitic stainless steel
0026	Annealed plain carbon wrought steel, 0.2-0.4% carbon
0027	Annealed plain carbon wrought steel, 0.4-0.7% carbon
0028	Normalized carbon/manganese steel
0029	Quenched and tempered carbon/manganese steel
0030	Hardened chromium steel
0031	Quenched and tempered chromium steel

0099	Steel of unknown heat treatment
0100	Wrought aluminium
0101	Wrought aluminium-copper alloy
0102	Wrought aluminium-manganese alloy
0103	Wrought aluminium-magnesium alloy
0104	Wrought aluminium-magnesium-silicon alloy
0105	Wrought aluminium-zinc alloy
0106	Cast aluminium alloy
0200	Wrought copper
0201	Wrought brass
0202	Wrought aluminium bronze
0203	Cupronickel
0204	Nickel silver
0205	Wrought phosphor bronze
0206	Wrought copper beryllium
0207	Cast copper alloys
0300	Titanium alloy
0400	Wrought magnesium alloys
0401	Cast magnesium alloys
0500	Fusible alloys, solders
0600	Cast zinc alloys
0700	Wrought nickel alloys
0701	Cast nickel alloys
0800	Precious metals
0900	Clad materials
1000	Thermoplastics
1001	Thermosetting plastics
1200	Epoxy adhesive

HBM - nCode

26555 Evergreen Road - Suite 700 -
Southfield, MI 48076 - USA
Tel. +1 248 350 8300 - Fax: +1 248 350 1678
Email: info@hbmncode.com - www.hbm.com/ncode

measure and predict with confidence

HBM