QUICK FATIGUE TOOL FOR MATLAB®

User Settings Reference Guide

Version Information

Documentation revision: 13 [21/06/2017]

Concurrent code release: 6.11-00

Contents

1. Job file options

1.1 Overview

The job file contains a list of options which define a specific analysis. The options are separated into the following categories:

1. Job
2. Material
3. Loading
4. High frequency loadings
5. Abaqus RPT / data set file
6. Analysis
7. Surface finish / notch effects
8. Virtual strain gauges
9. Output requests
10. ODB Interface
11. BS 7608 weld definition
12. Additional material data

Job files are created either from the template located in *Project\job\template_job.m*, or from a user-defined text file. Guidance on creating and submitting jobs for analysis is provided in Section 2.4 “Configuring and running an analysis” of the *Quick Fatigue Tool User Guide*.

If the user modified the template job file and wishes to revert back to the original, a back-up template is contained in *Application_Files\default\default_job.m*.

1.2 Job file options table

The following table provides a description of each option in the job file.

Option	Description
JOB_NAME	The name of the job, also the name of the corresponding output directory. *Job file usage:* JOB_NAME = '<job_name>';
JOB_DESCRIPTION	A short description of the analysis. Note that a summary of the job settings is automatically written to the log file. *Job file usage:* JOB_DESCRIPTION = '<job_description>';
CONTINUE_FROM	The name of the job onto which field data from the current job will be superimposed. *Job file usage:* CONTINUE_FROM = '<job_name>';
DATA_CHECK	Aborts the job just before the start of the fatigue analysis. *Job file usage:* DATA_CHECK = n; n has the following definitions: 0: Do not perform Data Check analysis 1: Perform a Data Check analysis
MATERIAL	The material used for the analysis. The file extension *.mat* can be omitted. The material must exist in the following directory: *Data\material\local* *Job file usage:* MATERIAL = '<material_name>.mat';

USE_SN

Controls whether stress-life data is being used for the analysis.

Job file usage:
USE_SN = n;

n has the following definitions:
0: Derived (Sf' and b)
1: Direct (S-N data points)

(Refer to section 4 for detailed job file usage)

SN_SCALE

Scales S-N stress data points.

Job file usage:
SN_SCALE = n;

n is a real number.

SN_KNOCK_DOWN

S-N knock-down curve for S-N data points.

Job file usage:
SN_KNOCK_DOWN = n;

n is a real number.

DATASET

The file, or files, containing the stress data for the analysis.

Job file usage for simple loading:
DATASET = '<fea_definition>.rpt';

(Refer to section 3 for detailed job file usage)

HISTORY

The file, or files, containing the load history data for the analysis.

Job file usage for a simple loading:
HISTORY = '<history_data>.dat';

(Refer to section 3 for detailed job file usage)

UNITS	Units for the stress data. *Job file usage:* UNITS = n; n has the following definitions: 0: User-defined 1: Pa 2: kPa 3: MPa 4: psi 5: ksi 6: Msi

CONV	Conversion factor for user-defined units. *Job file usage:* CONV = n; n is the conversion factor between Pascal [Pa] and the user-defined unit.

LOAD_EQ	Unit of measurement to which the complete loading definition is equal. *Job file usage:* LOAD_EQ = { n, '<unit>'}; n is the number of repeats that the loading is equivalent to, expressed by the units <unit>. By default, the loading is equivalent to 1 repeat.

SCALE	Load history scale factors. *Job file usage for a single load history:* SCALE = n; n is a real number. *Job file usage for multiple load histories:* SCALE = [n_1, n_2,…, n_i]; i is the number of load histories.

OFFSET	Load history offset values. *Job file usage for a single load history:* OFFSET = n; n is a real number. *Job file usage for multiple load histories:* OFFSET = [n_1, n_2,…, n_i]; i is the number of load histories.

REPEATS	The number of repeats of the loading. *Job file usage:* REPEATS = n; n is a real number.

HF_DATASET	The file, or files, containing high frequency stress data sets. *Job file usage for simple loading:* HF_DATASET = '<hf_dataset>.rpt'; (Refer to section 3 for detailed job file usage)

HF_HISTORY	The file, or files, containing high frequency load history data. *Job file usage for a simple loading:* HF_HISTORY = '<hf_history>.dat'; (Refer to section 3 for detailed job file usage)

HF_TIME	The time period for the low and high frequency data sets. *Job file usage:* HF_TIME = {t_1, t_2}; t_1 is the time period of the low frequency data t_2 is the time period of the high frequency data

HF_SCALE	A scale factor for the entire loading. *Job file usage for a single load history:* HF_SCALE = n; n is a real number. HF_SCALE is used in the same way as SCALE

PLANE_STRESS	Specifies whether the input file reader should recognise data set files containing plane stress elements. *Job file usage:* PLANE_STRESS = n; n has the following definitions: 0: Allow data set files with 3D stress elements only. Combinations of 2D/3D elements will not be recognised 1: Allow data set files with plane stress elements. Combinations of 2D/3D elements will be recognised

OUTPUT_DATABASE	Associates an analysis with an Abaqus output database (.odb) file. If the stress data set(s) were created from an Abaqus model, Quick Fatigue Tool will attempt to write field data back to the .odb file specified. *Job file usage:* OUTPUT_DATABASE = '<odb_name>.odb'; Where <odb_name> is the full (absolute) path to the Abaqus .odb file.

PART_INSTANCE	The part instance in the Abaqus model to which field data should be written. *Job file usage:* PART_INSTANCE = '<part_instance_name>'; Multiple part instances may be specified as a cell: *Job file usage:* PART_INSTANCE = {'<instance_1>',..., '<instance_n>'};

EXPLICIT_FEA	The analysis procedure which was used to calculate the FE stresses. If the results step originates from an Abaqus/Explicit analysis, the proceeding step must also be Explicit. Quick Fatigue Tool requires this information from the user in order to write fatigue results back to the ODB file. *Job file usage for Explicit FEA:* EXPLICIT_FEA = 1.0; *Job file usage for all other FEA types:* EXPLICIT_FEA = 0.0;

STEP_NAME	The name appended to the results ODB step. This option is not compulsory, but minimizes the risk of naming clashes which would prevent Quick Fatigue Tool from successfully writing field data to the .odb file. *Job file usage:* STEP_NAME = '<step_name>';

RESULT_POSITION	Results position of the exported field data. *Job file usage:* RESULT_POSITION = 'ELEMENT NODAL' | 'UNIQUE NODAL' | 'INTEGRATION POINT' | CENTROID';

GROUP	Defines analysis groups for the use of individual properties to user-defined regions. *Job file usage:* GROUP = {'<group_1>.rpt',…, '<group_n>.rpt'};

ALGORITHM	The algorithm used for analysis. *Job file usage:* ALGORITHM = n; n has the following definitions: 0: Default 1: [Reserved for future release] 2: [Reserved for future release] 3: Uniaxial Strain-Life 4: Stress-based Brown-Miller (CP) 5: Normal Stress (CP) 6: Findley's Method (CP) 7: Stress Invariant Parameter 8: BS 7608 Fatigue of welded steel joints 9: NASALIFE 10: Uniaxial Stress-Life 11: User-defined If $n = 0$, the default algorithm from the material properties will be used.

MS_CORRECTION	The mean stress correction used for analysis. *Job file usage:* MS_CORRECTION = n; n has the following definitions: 0: Default 1: Morrow 2: Goodman 3: Soderberg 4: Walker 5: Smith-Watson-Topper 6: Gerber 7: R-ratio S-N curves 8: None If $n = 0$, the default mean stress correction from the material properties will be used. If the mean stress correction is user-defined, it is specified by its file name. *Job file usage:* MS_CORRECTION = '<mscFilename>.msc';

ITEMS	The list of items for analysis. *Job file usage:* ITEMS = [n_1, n_2,…, n_i]; Each value in ITEMS references a row number in the stress definition file, and is not necessarily the same as the position number (if the stresses originate from an Abaqus .rpt file. To analyse all items in the model: ITEMS = 'ALL'; To analyse items from hotspots: ITEMS = 'hotspots_<jobname>.dat';

DESIGN_LIFE	The life at which the component meets its fatigue design criterion. *Job file usage:* DESIGN_LIFE = n; n is the design life is repeats (N_f). To set the design life as the material's constant amplitude endurance limit: DESIGN_LIFE = 'CAEL'; If the analysis contains multiple group definitions, the constant amplitude endurance limit is taken from the material defined in the last group.

KT_DEF	Surface finish definition. A value of K_t can be specified directly, or a K_t surface finish definition file can be provided. *Job file usage:* KT_DEF = n; n is a positive, real number. To calculate K_t from a file, specify the file name: KT_DEF = '<filename>.kt'; If the option is left empty, a K_t value of 1.0 will be assumed: KT_DEF = [ ];

KT_CURVE	Used to specify the K_t curve corresponding to a particular surface treatment, or a surface roughness value, R_z: *Job file usage:* KT_CURVE = n; If the surface finish is selected from a list, n is a positive integer. The maximum value of n depends on how many curves are defined in the .kt file. If the option is left empty, the first curve in the .kt will be used: KT_CURVE = [ ]; If the surface finish is defined as a value, n is a positive real number specifying the surface roughness.

NOTCH_CONSTANT	Material constant defining the relationship between the elastic stress concentration factor and the fatigue notch factor. Depending on the value of notchSensitivityMethod, the constant is either the characteristic length or the notch sensitivity. *Job file usage:* NOTCH_CONSTANT = n; n is a real number.

NOTCH_RADIUS	Notch root radius to calibrate the notch sensitivity estimation. *Job file usage:* NOTCH_RADIUS = n; n is a real number.

RESIDUAL	Specifies an in-plane residual stress component in the loading. The stress is assumed to act uniformly in all directions. *Job file usage:* RESIDUAL = n; n is a real number. Positive values of n correspond to a tensile residual stress, while negative values are compressive.

FACTOR_OF_STRENGTH	Enable Factor of Strength (FOS) iterations. *Job file usage:* FACTOR_OF_STRENGTH = n; n has the following definitions: 1: Enable FOS iterations 0: Disable FOS iterations

FATIGUE_RESERVE_FACTOR	Material life envelope expressed for Fatigue Reserve Factor calculations. *Job file usage:* FATIGUE_RESERVE_FACTOR = n; n has the following definitions: 1: Goodman (Default) 2: Goodman B 3: Gerber If the proof stress is not defined and the Goodman B envelope is specified, the standard Goodman envelope will be used by default. For user-defined envelopes, the FRF data (.msc) file is specified. *Job file usage:* FATIGUE_RESERVE_FACTOR = '<fileName>.msc';

HOTSPOT	Save items below the design life to a separate file. This file can be used in conjunction with ITEMS to re-analyse the model only for items which failed the previous design life criterion. *Job file usage:* HOTSPOT = n; n has the following definitions: 1: Enable hotspots 0: Disable hotspots

GAUGE_LOCATION	Define virtual gauges to simulate a rectangular rosette strain gauge at selected analysis items. *Job file usage:* GAUGE_LOCATION = {'$m_1.s_1$', '$m_2.s_2$',…, '$m_n.s_n$'}; '$m_n.s_n$' is the main and sub position ID of the location of gauge n. The IDs must be separated by a decimal point. For integration point or element-nodal positions, m_n is the element number and s_n is the integration point/node number. For centroidal or unique nodal positions, m_n is the centroid/node number. s_n is always 1.0.

GAUGE_ORIENTATION	Define gauge orientations for virtual strain gauges. *Job file usage:* GAUGE_ORIENTATION = {[α_1, β_1, γ_1], [α_2, β_2, γ_2],…, [α_n, β_n, γ_n]}; α_n is the angle measured counter clockwise from the positive global (Cartesian) x-direction to rosette arm A of gauge n. β_n is the angle measured counter clockwise from rosette arm A to B of gauge n. γ_n is the angle measured counter clockwise from rosette arm B to C of gauge n. Angles are measured in degrees. Define gauge orientations using flags. *Job file usage:* GAUGE_ORIENTATION = {'RECTANGULAR ' | 'DELTA '};

OUTPUT_FIELD	Field data output request. *Job file usage:* OUTPUT = n; n has the following definitions: 1: Write field output 0: Life at worst item only

OUTPUT_HISTORY	History data output request. *Job file usage:* OUTPUT = n; n has the following definitions: 1: Write history output 0: No history output

OUTPUT_FIGURE	MATLAB figure output request. *Job file usage:* OUTPUT = n; n has the following definitions: 1: Create MATLAB figures for histories 0: No MATLAB figures

WELD_CLASS	Define the $S_r - N$ curve for the BS 7608 analysis algorithm in terms of weld class. *Job file usage:* WELD_CLASS = n; WELD_CLASS = '<Weld identifier>'; n is an integer corresponding to one of the ten BS 7608 $S_r - N$ curves: 1: B 2: C 3: D 4: E 5: F 6: F2 7: G 8: W 9: S 10: T 11: X (Axially loaded bolts) Alternatively, <Weld identifier> may be used to reference the curve designation directly as a string value. The $S_r - N$ curve may be defined as user data. *Job file usage:* WELD_CLASS = {'<userCurveFile>.sn', ['ROW' | 'COL']} If the second argument is omitted, column-wise data is assumed.

YIELD_STRENGTH	The user-specified yield stress of the weld plate material for BS 7608 analysis. *Job file usage:* YIELD_STRENGTH = n; n is a real number.

UTS	The user-specified ultimate tensile strength of the bolt section for BS 7608 analysis, when using Class X. *Job file usage:* UTS = n; n is a real number.

DEVIATIONS_BELOW_MEAN	The number of deviations below the mean S-N curve family for BS 7608 analysis, used to specify the probability of failure. *Job file usage:* DEVIATIONS_BELOW_MEAN = n; n is a positive integer.

FAILURE_MODE	The crack initiation criterion for BS 7608 analysis. *Job file usage:* FAILURE_MODE = n; FAILURE_MODE = 'NORMAL' | 'SHEAR' | 'COMBINED'; n has the following definitions: 1: Normal failure 2: Shear failure 3: Combined (normal + shear) failure

CHARACTERISTIC_LENGTH	The plate thickness for all weld classifications (or bolt diameter for axially loaded bolts), used to correct the S-N curve for BS 7608 analysis. *Job file usage:* CHARACTERISTIC_LENGTH = n; n is a positive real number. The length units are *mm*. The option should set empty if the characteristic length is not important: CHARACTERISTIC_LENGTH = [ ];

SEA_WATER	Specifies whether the welded feature is exposed to sea water for BS 7608 analysis. *Job file usage:* SEA_WATER = n; n is 1 or 0.

B2	Secondary Basquin's exponent used above a user-defined knee point. *Job file usage:* B2 = n; n is less than zero.

B2_NF	Life above which the secondary Basquin's exponent (B2) is used. *Job file usage:* B2_NF = n; n is greater than zero.

UCS	The material ultimate compressive strength. *Job file usage:* UTS = n; n is a real number.

2. Environment variables

2.1 Overview

The environment file contains a list of variables which define the general behaviour of Quick Fatigue Tool. Environment variables are applied *globally* and *locally*. Global environment variables define the behaviour of the program for all analysis jobs executed in the current Quick Fatigue Tool directory, whereas local environment variables define the behaviour of the program for a specific job.

2.2 Global environment file

The global environment variables are contained in *Application_Files\default\environment.m*. Changes made to this file will affect all jobs submitted in the current working directory. In the event that the user wishes to revert to the original environment settings, a list of environment variables with their default values can be found in *Application_Files\default\default_environment.m*.

2.3 Local environment files

Local environment variables may be defined for a particular job. This is done by making a copy of the global environment file and placing it in *Project\job*. Alternatively, the user may create a blank text file and specify only the environment variables they wish to define for that job. However, for the analysis to run properly, the global environment file must always be available.

The local environment file must obey the naming convention *<jobName>_env.m*, otherwise it will not be processed by Quick Fatigue Tool.

2.4 Processing of environment variables

At the beginning of each analysis, Quick Fatigue Tool searches *Application_Files\default* for the global environment file. It then searches *Project\job* for the local environment file. Local environment settings supersede global environment settings.

Before each analysis, it is compulsory for all environment variables to be defined. Even if the user specifies non-default values in a local environment file, the global environment file must still exist and contain all of the variables. If the global environment file is modified or lost, and the default environment file is no longer available, the fastest way to recover the settings is to re-download the application from the MATLAB File Exchange.

2.5 Environment variables table

The following table provides a description of each variable in the environment file.

Variable	Description
gateTensors	Controls whether the peak-valley analysis algorithm is used for tensors. *Environment file usage:* setappdata(0, 'gateTensors', n); n has the following definitions: 0: Off 1: Gate tensors (as % of max tensor) 2: Gate tensors (Nielsony's method)
tensorGate	Tensor gating values. *Environment file usage:* setappdata(0, 'tensorGate', n); n has the following definitions: 0: Zero-derivative method N: % of max tensor
gateHistories	Controls whether the peak-valley analysis algorithm is used for load histories. *Environment file usage:* setappdata(0, 'gateHistories', n); n has the following definitions: 0: Off 1: Pre-gate load histories (as % of max tensor) 2: Pre-gate load histories(Nielsony's method)

historyGate	Load history gating values. *Environment file usage:* setappdata(0, 'historyGate', n); n has the following definitions: 0: Zero-derivative method N: % of max tensor
noiseReduction	Noise reduction algorithm applied to load history. *Environment file usage:* setappdata(0, 'noiseReduction', n) n is 1.0 or 0.0.
numberOfWindows	Number of averaging windows used by the noise reduction algorithm. *Environment file usage:* setappdata(0, 'numberOfWindows', n) n is a positive integer.
groupDefinition	Controls how Quick Fatigue Tool should treat a group definition file. *Environment file usage:* setappdata(0, 'groupDefinition', n); n has the following definitions: 0: Program controlled 1: Always read group data as an FEA subset

modifiedGoodman	Defines the envelope for the Goodman mean stress correction. *Environment file usage:* setappdata(0, 'modifiedGoodman', n); n has the following definitions: 0: Use the standard Goodman envelope 1: Use the intersection of the Buch with the Goodman envelope
goodmanMeanStressLimit	Defines the horizontal (mean stress) axis intercept for the Goodman mean stress correction. *Environment file usage:* setappdata(0, 'goodmanMeanStressLimit', [n | 'string']); n has the following definitions: 'UTS': Use the material UTS 'PROOF': Use the material proof stress 'S-N': Use the S-N intercept stress (at 1 repeat) n: User-defined User-defined values are not supported if the modified Goodman envelope is selected.
walkerGammaSource	Controls how the Walker gamma parameter is calculated. *Environment file usage:* setappdata(0, 'walkerGammaSource', n); n has the following definitions: 1: Calculate the gamma parameter from the Walker regression fit 2: Use Dowling's approximation for steel and aluminium 3: User-defined
userWalkerGamma	User-defined value of the Walker gamma parameter if walkerGammaSource = 2.0. *Environment file usage:* setappdata(0, 'userWalkerGamma', n); n is a real number, positive number.

rainflowAlgorithm	Selects the rainflow cycle counting algorithm. *Environment file usage:* setappdata(0, 'rainflowAlgorithm', n); n has the following definitions: 1: De Morais (Legacy) 2: Vallance The second version is a more recent implementation and has been found to produce much better results. The original algorithm is available for regression and testing purposes.
rainflowMode	Specify the order of operations for the selected rainflow algorithm for two-parameter cycle counting. *Environment file usage:* setappdata(0, 'rainflowMode', n); n has the following definitions: 1: Combine parameters, count cycle 2: Count parameters, combine cycles This setting currently only applies to the Stress-based Brown-Miller algorithm.
shellLocation	Sets the default face from which to read shell element stresses. *Environment file usage:* setappdata(0, 'shellLocation', n); n has the following definitions: 1: LOC 1 (SNEG) 2: LOC 2 (SPOS)

nlMaterial	Controls which material model is used for the analysis. *Environment file usage:* setappdata(0, 'nlMaterial', n); n has the following definitions: 0: Linear (Hookean) material model 1: Nonlinear elastic (Ramberg-Osgood) material model A linear elastic material model is acceptable for most stress-based analyses. Use with caution.

cssTolerance	Precision of the iterative Ramberg-Osgood solver for the nonlinear material model. *Environment file usage:* setappdata(0, 'cssTolerance', n); n is a positive real number.

cssMaxIterations	Maximum number of iterations performed by the iterative Ramberg-Osgood solver before accepting the solution. *Environment file usage:* setappdata(0, 'cssMaxIterations', n); n is a positive integer.

nodalElimination	Controls whether the nodal elimination algorithm is being used for analysis. *Environment file usage:* setappdata(0, 'nodalElimination', n); n has the following definitions: 0: Analyse all nodes 1: Nodal elimination based on material's CAEL 2: Nodal elimination based on user design life

thresholdScaleFactor	A factor which scales the conditional stress, σ_{COND}, for the nodal elimination algorithm. *Environment file usage:* setappdata(0, 'thresholdScaleFactor', n); n is a real number. The default value is 0.8.

yieldCriterion	Perform static strength yield calculations. *Environment file usage:* setappdata(0, 'yieldCriterion', n); n has the following definitions: 0: Do not perform yield calculations 1: Perform yield calculations based on the total strain energy theory 2: Perform yield calculations based on the shear strain energy theory

stepSize	The step size for the critical plane analysis algorithm. *Environment file usage:* setappdata(0, 'stepSize', n); n must be a factor of 180. The default step size is 15 degrees.

checkLoadProportionality	Enables load proportionality checking to determine if the critical plane step size may be increased. *Environment file usage:* setappdata(0, 'checkLoadProportionality', n); n has the following definitions: 0: Use step size defined by stepSize 1: Check for load proportionality and increase the step size to 45 degrees if applicable

proportionalityTolerance	The tolerance angle for the load proportionality check. The load is considered to be proportional if the maximum deviation in the principal directions does not exceed this angle. *Environment file usage:* setappdata(0, 'proportionalityTolerance', n); n is an angle in degrees. The default tolerance angle is 1 degree.

cpSample	The re-sample rate for the critical plane analysis plots. *Environment file usage:* setappdata(0, 'cpSample', n); n is a real number. The re-sample rate only affects the smoothness of the MATLAB figure data. It does not affect the tabulated data from which the plots are created.

cpShearStress	Specify how the maximum shear stress history is determined on the critical plane. *Environment file usage:* setappdata(0, 'cpShearStress', n) n has the following definitions: 1: Maximum chord method 2: Maximum resultant shear stress

signConvention	The default sign convention for effective stress quantities. When using the Stress-Based Brown-Miller, Normal Stress, Findley’s Method or BS 7608, the effective stress quantity is the resultant shear stress on the critical plane. When using the von Mises algorithm, the effective stress quantity is the von Mises stress. *Environment file usage:* setappdata(0, 'signConvention', n); n has the following definitions: 1: Sign from the hydrostatic stress 2: Sign from the largest stress

nasalifeParameter	The effective stress parameter for the NASALIFE analysis algorithm. *Environment file usage:* setappdata(0, 'nasalifeParameter', n); n has the following definitions: 1: Manson-McKnight 2: Sines 3: Smith-Watson-Topper 4: R-Ratio Sines 5: Effective

plasticSN	Specifies which region(s) of the stress-life curve to consider for analysis when using the stress-based Brown-Miller analysis algorithm. *Environment file usage:* setappdata(0, 'plasticSN', n) n has the following definitions: 0: Use elastic region of S-N curve only (Sf' and b) 1: Include plastic region of S-N curve (Ef' and c)

findleyNormalStress	Specify how the normal stress is matched with shear cycles for Findley's Method. *Environment file usage:* setappdata(0, 'findleyNormalStress', n) n has the following definitions: 1: Use the maximum normal stress over the loading 2: Use the maximum normal stress over the maximum shear cycle interval 3: Use the average normal stress over the maximum shear cycle interval

stressInvariantParameter	Specify the damage parameter for the Stress Invariant Parameter algorithm. *Environment file usage:* setappdata(0, 'stressInvariantParameter', n) n has the following definitions: 0: Program controlled 1: von Mises 2: Principal 3: Hydrostatic (pressure) 4: Tresca
nasalifeParameter	Specify the damage parameter for the NASALIFE algorithm. *Environment file usage:* setappdata(0, 'nasalifeParameter', n) n has the following definitions: 1: Manson-McKnight 2: Sines 3: Smith-Watson-Topper 4: R-Ratio Sines 5: Effective
ndCompression	Controls whether damage is calculated for compressive cycles. *Environment file usage:* setappdata(0, 'ndCompression', n); n has the following definitions: 1: No damage in compression 0: Compressive cycles cause damage The default value of n is 1.

fatigueLimitSource	Controls how the fatigue limit is calculated. *Environment file usage:* setappdata(0, 'fatigueLimitSource', n); n has the following definitions: 1: Calculate the fatigue limit from Basquin material coefficients (Sf and b) 2: Calculate the fatigue limit from algorithm-specific equation 3: User-defined The default value of n is 1. When using Basquin coefficients, the fatigue limit is calculated from $\sigma_\infty = \sigma'_f(CAEL)^b$, where σ_∞ is the fatigue limit stress, σ'_f is the fatigue strength coefficient, $CAEL$ is the constant amplitude endurance limit ($2N_f$) and b is the fatigue strength exponent. The algorithm-specific equations only apply to the Stress-based Brown-Miller and the Findley algorithms. Stress-based Brown-Miller (without fatigue ductility): $$\sigma_\infty = \sigma'_f(1.65 \times CAEL)^b$$ Stress-based Brown-Miller (with fatigue ductility): $$\sigma_\infty = E\left(\frac{1.65\sigma'_f}{E}(CAEL)^b + 1.75\varepsilon'_f(CAEL)^c\right)$$ Findley's method: $$\sigma_\infty = \tau^*_f(CAEL)^b$$ If USE_SN = 1.0 in the job file, the fatigue limit will be calculated by interpolating the S-N data points, unless a user-defined endurance limit is specified.

userFatigueLimit	User-defined value of the endurance limit if fatigueLimitSource = 3.0. *Environment file usage:* setappdata(0, 'userFatigueLimit', n); n is a real number, positive number. Custom values of the fatigue limit are not supported by the BS 7608 analysis algorithm.

ndEndurance	Assumes no damage for cycles below the endurance limit. *Environment file usage:* setappdata(0, 'ndEndurance', n); n has the following definitions: 0: Program-controlled 1: Calculate damage for cycle below the endurance limit 2: Assume no damage for cycles below the endurance limit For a detailed explanation of the way Quick Fatigue Tool treats the endurance limit, consult Appendix I: Fatigue analysis techniques.

modifyEnduranceLimit	Reduces the endurance limit for damaging cycles. *Environment file usage:* setappdata(0, 'modifyEnduranceLimit', n); n has the following definitions: 0: Damaging cycles do not affect the endurance limit 1: Damaging cycles reduce the endurance limit to 25% of its original value

enduranceScaleFactor	The amount by which the endurance limit is scaled after a damaging cycle, if modifyEnduranceLimit=1. *Environment file usage:* setappdata(0, 'enduranceScaleFactor', n); The default value of n is 0.25.

cyclesToRecover	The number of non-damaging cycles before the endurance limit returns to its original value, if modifyEnduranceLimit=1. *Environment file usage:* setappdata(0, 'cyclesToRecover', n); n is an integer number of cycles. The default value of n is 50.

fosTarget	Target life for the FOS calculations *Environment file usage:* setappdata(0, 'fosTarget', n); n has the following definitions: 1: Perform FOS calculations for user-defined design life 2: Perform FOS calculations for infinite design life (CAEL)

fosMaxValue	The maximum reported value of the FOS. *Environment file usage:* setappdata(0, 'fosMaxValue', n); n is a real number. The default value of n is 2.

fosMaxFine	The maximum FOS value at which fine increments are performed. *Environment file usage:* setappdata(0, 'fosMaxFine', n); n is a real number. The default value of n is 1.5.

fosMinFine	The minimum FOS value at which fine increments are performed. *Environment file usage:* setappdata(0, 'fosMinFine', n); n is a real number. The default value of n is 0.8.

fosMinValue	The minimum reported value of the FOS. *Environment file usage:* setappdata(0, 'fosMinValue', n); n is a real number. The default value of n is 0.5.
fosCoarseIncrement	The step size used for FOS iterations outside the fine bands. *Environment file usage:* setappdata(0, 'fosCoarseIncrement', n); n is a real number. The default value of n is 0.1.
fosFineIncrement	The step size used for FOS iterations within the fine bands. *Environment file usage:* setappdata(0, 'fosCoarseIncrement', n); n is a real number. The default value of n is 0.01.
fosMaxCoarseIterations	The maximum number of coarse iterations for the FOS calculation. *Environment file usage:* setappdata(0, 'fosMaxCoarseIterations', n); n is a positive integer. The default value of n is 4.
fosMaxFineIterations	The maximum number of fine iterations for the FOS calculation. *Environment file usage:* setappdata(0, 'fosMaxFineIterations', n); n is a positive integer. The default value of n is 6.
fosTolerance	The percentage tolerance value for the FOS calculation. *Environment file usage:* setappdata(0, 'fosTolerance', n); n is a real number. The default value of n is 5%.

fosBreakAfterBracket	Terminates FOS iterations if the most recently calculated life crosses the target life. *Environment file usage:* setappdata(0, 'fosBreakAfterBracket', n); n has the following definitions: 1: Enable bracketing 2: Disable bracketing
fosAugment	Enables augmented FOS iterations. If the current FOS increment meets the threshold criterion set by fosAugmentThreshold, the increment is scaled by the factor fosAugmentFactor. *Environment file usage:* setappdata(0, 'fosAugment', n); n has the following definitions: 1: Enable augmented iterations 2: Disable augmented iterations
fosAugmentThreshold	Threshold value for augmented FOS iterations. *Environment file usage:* setappdata(0, 'fosAugmentThreshold' , n); n is a number between 0.0 and 1.0.
fosAugmentFactor	Scale factor for augmented FOS iterations. *Environment file usage:* setappdata(0, 'fosAugmentFactor', n); n is a real number.

fosDiagnostics	Diagnostic output for the FOS calculation. *Environment file usage:* setappdata(0, 'fosDiagnostics', n); n has the following definitions: 1: Write output 2: Do not write output
frfInterpOrder	Interpolation order for user-defined FRF data. *Environment file usage:* setappdata(0, 'frfInterpOrder', '<ARG>'); The following strings are accepted for <ARG>: 'NEAREST', 'LINEAR', 'SPLINE', 'PCHIP'. For a description of each argument, see Section 8.2.5, or type doc interp1 in the MATLAB command window.
frfNormParamMeanT	Normalization parameter for tensile mean stress user-defined FRF data. *Environment file usage:* setappdata(0, [' frfNormParamMeanT' | '<param>' | n]); '<param>' has the following definitions: 'UTS': Ultimate tensile strength 'UCS': Ultimate compressive strength 'PROOF': 0.2% Proof stress n is a user-defined normalization parameter.

frfNormParamMeanC	Normalization parameter for compressive mean stress user-defined FRF data. *Environment file usage:* setappdata(0, [' frfNormParamMeanC' | '<param>' | n]); '<param>' has the following definitions: 'UTS': Ultimate tensile strength 'UCS': Ultimate compressive strength 'PROOF': 0.2% Proof stress n is a user-defined normalization parameter.

frfNormParamAmp	Normalization parameter for stress amplitude user-defined FRF data. *Environment file usage:* setappdata(0, [' frfNormParamAmp' | '<param>' | n] '<param>' has the following definitions: 'LIMIT': Fatigue limit n is a user-defined normalization parameter.

frfTarget	Target life for the FRF calculations. *Environment file usage:* setappdata(0, 'frfTarget', n); n has the following definitions: 1: Perform FRF calculations for user-defined design life 2: Perform FRF calculations for infinite design life (CAEL)

frfMaxValue	The maximum reported value of the FRF. *Environment file usage:* setappdata(0, 'frfMaxValue', n); n is a real number. The default value of n is 10.

frfMinValue	The minimum reported value of the FRF. *Environment file usage:* setappdata(0, 'frfMinValue', n); n is a real number. The default value of n is 0.01.

frfDiagnostics	Diagnostic output for the FRF calculation when user-defined envelopes are specified. *Environment file usage:* setappdata(0, 'frfDiagnostics', [I_1, I_2, …, I_n]); n has the following definitions: [ ]: Disabled I_n: Write output for item number I_n in the data set

notchFactorEstimation	Method for approximating the fatigue notch factor based on the elastic stress concentration factor. *Environment file usage:* setappdata(0, 'notchFactorEstimation', n); n has the following definitions: 1: Peterson (default) 2: Peterson B 3: Neuber 4: Harris 5: Heywood 6: Notch sensitivity

eigensolver	Eigensolver used to calculate the principal stress history. *Environment file usage:* setappdata(0, 'eigensolver', n); n has the following definitions: 1: MATLAB (built-in) 2: Luong

numberOfBins	The number of range bins used for the rainflow cycle histogram plot, RHIST. *Environment file usage:* setappdata(0, 'numberOfBins', n); n is a positive integer.

figureFormat	File format for MATLAB figures. *Environment file usage:* setappdata(0, 'figureFormat', '<format>'); <format> specifies the file format. e.g.: 'fig': MATLAB Figure 'png': Portable Network Graphics 'jpg': JPEG

fieldFormatString	Specify output format of field output file. *Environment file usage:* setappdata(0, 'fieldFormatString', '<format>');

historyFormatString	Specify output format of history output files. *Environment file usage:* setappdata(0, 'historyFormatString', '<format>');

echoMessagesToCWIN	Echo message file contents to the MATLAB command window. *Environment file usage:* setappdata(0, 'echoMessagesToCWIN', n); n has the following definitions: 1: Enabled 0: Disabled

cleanAppData	Controls when the session application data should be cleaned. *Environment file usage:* setappdata(0, 'cleanAppData', n); n has the following definitions: 1: Before the analysis 2: After the analysis 3: Before and after the analysis 4: Never

workspaceToFile	Cache the workspace variables and associated APPDATA into a MATLAB binary (.mat) file. *Environment file usage:* setappdata(0, 'workspaceToFile', n); n has the following definitions: 0: Disabled 1: Every n analysis items 2: n evenly spaced analysis items 3: From analysis item IDs The analysis item interval/IDs are specified with workspaceToFileInterval.

workspaceToFileInterval	Set the frequency at which workspace variables and APPDATA are written. *Environment file usage:* setappdata(0, 'workspaceToFileInterval', {n, '<OPTION>'); If workspaceToFile = 1.0 | 2.0, n is the interval number. If workspaceToFile = 3.0, n is an analysis item ID list: $[ID_1, \ldots, ID_n]$. <OPTION> has the following definitions: 'OVERLAY': Overwrite the cache file. Only the cache file at the last interval/ID is retained. 'RETAIN': Create an individual cache file for each interval/ID.

autoExport_ODB	Enables automatic export of field data to an Abaqus output database (.odb) file. *Environment file usage:* setappdata(0, 'autoExport_ODB', n); n has the following definitions: 1: Enabled 0: Disabled
autoExport_stepType	Controls whether field output is written to a new ODB step, or an existing step from a previous Quick Fatigue Tool analysis. *Environment file usage:* setappdata(0, 'autoExport_stepType', n); n has the following definitions: 1: Export results to new step 2: Export results to existing step
autoExport_autoPosition	Allow Quick Fatigue Tool to determine the data position automatically, based on the format of the field data position IDs. *Environment file usage:* setappdata(0, 'autoExport_autoPosition', n); n has the following definitions: 1: Enabled 0: Disabled If the data position is already known, a value of 0 is strongly advised.
autoExport_upgradeODB	Specify whether the model ODB file should be upgraded. *Environment file usage:* setappdata(0, 'autoExport_ upgradeODB', n); n has the following definitions: 1: Upgrade the model ODB file 0: Do not upgrade the model ODB file The Abaqus version to which the model ODB file is upgraded depends on the setting of autoExport_abqCmd.

autoExport_abqCmd	Specify the Abaqus API version used by the ODB Interface. *Environment file usage:* setappdata(0, 'autoExport_ abqCmd', '<abaqus_command>'); <abaqus_command> is the name of the batch file used to launch Abaqus and is usually located in *<Abaqus_installation_directory>\Commands*. If the ODB upgrade utility is enabled with autoExport_upgradeODB, this is the Abaqus version to which the file is upgraded.
autoExport_createODBSet	Write an element or node set to the output database file containing the elements or nodes used for analysis. If the input stresses are at the unique nodal position, a node set containing these nodes is written to the .odb file. If the input stresses are at the element-nodal position, both element and node sets containing the element-nodes used for analysis are written to the .odb file. If the input stresses are at the integration point or centroidal position, an element set constituting the domain of the integration points or centroids is written to the .odb file. *Environment file usage:* setappdata(0, 'autoExport_createODBSet', 1.0);

autoExport_ODBSetName	Name of the ODB element/node set if createODBSet = 1.0. If this variable is left blank, a default name is used, composed of the part instance name and the step name.
	
	Environment file usage:
	setappdata(0, 'autoExport_ODBSetName', [ ]);
	setappdata(0, 'autoExport_ODBSetName', '<name>');

autoExport_executionMode	Controls the visibility of the Python script used by the Abaqus API. *Environment file usage:* setappdata(0, 'autoExport_executionMode', n); n has the following definitions: 1: Create ODB, discard Python script 2: Create ODB, retain Python script 3: Write Python script only

autoExport_selectionMode	Output variable selection type. *Environment file usage:* setappdata(0, 'autoExport_selectionMode', n); n has the following definitions: 1: Select the output variables from a list 2: Use preselected defaults 3: Select all available output variables If $= 1$, the output variable selection is taken from the definitions of autoExport_<field_name>. If $n = 2$, the following variables are requested by default: *BS 7608 analysis:* $LL, SMAX, WCM, WCA$ *All other analyses:* $LL, FRFR, FRFV, FRFH, FRFW, SMAX, WCM, WCA$ If the factor of strength or yield calculations are enabled, the FOS and $YIELD$ variables are automatically written to the ODB. Note that output to the ODB file is not supported by the Uniaxial Stress-Life algorithm. If $n = 3$, all available output variables are written to the ODB file.

autoExport_<field_name>	Controls which fields are written to the Abaqus .odb file if the variable selection type is set to “Select from list below” by autoExport_selectionMode. *Environment file usage:* setappdata(0, 'autoExport_<field_name >', n); n has the following definitions: 1: Export field <field_name> 0: Do not export field <field_name>

3. Material keywords

3.1 Overview

This section describes all of the keywords that are available for defining materials in a text file format in Quick Fatigue Tool. Materials are defined by lines in the text file. Three types of input lines are used in a material text file: *keyword* lines, *data* lines, and *comment* lines. A carriage return is required at the end of each line in a material text file.

- Keyword lines introduce options and often have *parameters*, which appear as words or phrases on the keyword line. Parameters can stand alone or have a value, and they may be required or optional.
- Data lines, which are used to provide numeric or alphanumeric entries, follow most keyword lines.
- Any line that begins with stars in columns 1 and 2 (**) is a comment line. Such lines can be placed anywhere in the file. They are ignored by Quick Fatigue Tool.

3.2 Keyword lines

The following rules apply when entering a keyword line:

- The first non-blank character of each keyword line must be a star (*).
- The keyword must be followed by a comma (,) if any parameters are given.
- Blanks on a keyword line are ignored.
- Keywords and parameters are not case sensitive.
- Keywords and parameters need not be spelled out completely, but there must be enough characters given to distinguish them from other keywords and parameters that begin in the same way.
- You should not use case as a method of distinguishing values. For example, Quick Fatigue Tool does not distinguish between the following definitions:

  ```
  *USER MATERIAL, NAME=STEEL
  *USER MATERIAL, NAME=Steel
  ```

3.3 Data lines

Data lines are used to provide data that are more easily given in lists than as parameters. Most options require one or more data lines; if they are required, the data lines must immediately follow the keyword line introducing the option. The following rules apply when entering a data line:

- All data lines must be separated by comas (,). An empty data field is specified by omitting data between commas.
- Empty data fields at the end of a line can be ignored.
- Floating point numbers can be given with or without an exponent. Any exponent, if input, must be preceded by E and an optional (-) or (+), as per the usual MATLAB syntax.

3.4 Material keywords table

The following table provides a description of each material keyword and their associated parameters.

*USER MATERIAL

Begin the definition of a material.

This option is used to indicate the start of a material definition.

Required parameter:

NAME

Set this parameter to a label that will be used to refer to the material with the job file option MATERIAL.

There are no data lines associated with this option.

*DESCRIPTION

Provide a description of the material.

This option is used to provide a summary for the material.

There are no parameters associated with this option.

Data line to provide a description:

First line:

1. The description.

Repeat this data line as often as necessary to define the material description.

*DEFAULT ALGORITHM

Specify the default analysis algorithm.

This option is used to specify the analysis algorithm which is used when ALGORITHM=default in the job file.

Required, mutually exclusive parameters:

UNIAXIAL STRESS | UNIAXIAL STRAIN | SBBM | NORMAL | FINDLEY | INVARIANT | NASALIFE

Uniaxial Stress-Life, Uniaxial Strain-Life, Stress-based Brown-Miller, Normal Stress, Findley's Method, Stress Invariant Parameter, and NASALIFE, respectively.

There are no data lines associated with this option.

*DEFAULT MSC

Specify the default mean stress correction.

This option is used to specify the mean stress correction which is used when MS CORRECTION=default in the job file.

Required, mutually exclusive parameters:

MORROW | GOODMAN| SODERBERG| WALKER| SWT| GERBER| RATIO | NONE

There are no data lines associated with this option.

*CAEL

Specify the constant amplitude endurance limit.

This option is used to specify the constant amplitude endurance limit of the material. This is the life below which Quick Fatigue Tool assumes the material experiences no fatigue damage.

There are no parameters associated with this option.

Data line to define the constant amplitude endurance limit:

First (and only) line:

1. The constant amplitude endurance limit ($2N_f$).
2. A flag (1 or 0) indicating if this variable is active in the material.

*REGRESSION

Specify the regression algorithm.

This option is used to specify the regression algorithm for undefined material properties.

Required, mutually exclusive parameters:

UNIFORM | UNIVERSAL | MODIFIED | 9050 | NONE

Uniform Law (Baumel & Seeger), Universal Slopes (Manson), Modified Universal Slopes (Muralidharan), 90/50 Rule, none, respectively.

There are no data lines associated with this option.

*MECHANICAL

Specify mechanical constants.

This option is used to define mechanical properties.

There are no parameters associated with this option.

Data lines to define mechanical properties:

First line:

1. Young's Modulus (E).
2. Poisson's ratio (ν).
3. Ultimate tensile strength (σ_U).
4. Yield (proof) stress (σ_Y).

Second (optional) line:

1. A flag (1 or 0) indicating if E is active in the material.
2. A flag (1 or 0) indicating if ν is active in the material.
3. A flag (1 or 0) indicating if σ_U is active in the material.
4. A flag (1 or 0) indicating if σ_Y is active in the material.

*FATIGUE

Specify fatigue properties.

This option is used to define fatigue properties. Fatigue properties can be supplied as constants for the stress-life and strain-life equations, and/or as test data defining the stress-life curve.

Required, mutually exclusive parameters:

CONSTANTS | TEST DATA

Data lines to define fatigue properties (CONSTANTS):

First line:

1. Fatigue strength coefficient ($\sigma_f{}'$).
2. Fatigue strength exponent (b).
3. Fatigue ductility coefficient ($\varepsilon_f{}'$).
4. Fatigue ductility exponent (c).

Second (optional) line:

1. A flag (1 or 0) indicating if $\sigma_f{}'$ is active in the material.
2. A flag (1 or 0) indicating if b is active in the material.
3. A flag (1 or 0) indicating if $\varepsilon_f{}'$ is active in the material.
4. A flag (1 or 0) indicating if c is active in the material.

Data lines to define fatigue properties (TEST DATA):

First line:

1. First N-value.
2. Corresponding S-value for the first load ratio.
3. Corresponding S-value for the second load ratio.

Continue this data line until the S-values are provided for all load ratios.

Second line:

1. Second N-value.
2. Corresponding S-value for the first load ratio.
3. Corresponding S-value for the second load ratio.

Repeat this data line as often as necessary to define all N-values. A minimum of two N-values is required.

*R RATIOS

Specify load ratios for S-N data.

This option is used to define a list of load ratios. It is only required if the **TEST DATA** parameter is used with ***FATIGUE** and more than one S-value is provided for each N-value.

There are no parameters associated with this option.

Data line to define load ratios:

First (and only) line:

1. First load ratio.

Continue this data line until the load ratios are provided for all S-N curves.

*CYCLIC

Specify cyclic properties.

This option is used to define cyclic hardening properties. These properties are required for the following features in Quick Fatigue Tool:

- plasticity correction;
- yield calculation; and
- damage calculation for Multiaxial Gauge Fatigue analysis.

There are no parameters associated with this option.

Data lines to define cyclic properties:

First line:

1. Cyclic strain hardening coefficient (K').
2. Cyclic strain hardening exponent (n').

Second (optional) line:

1. A flag (1 or 0) indicating if K' is active in the material.
2. A flag (1 or 0) indicating if n' is active in the material.

*NORMAL STRESS SENSITIVTY

Specify the normal stress sensitivity constant for Findley's Method.

This option is used to define the normal stress sensitivity constant, k, if ALGORITHM=findley in the job file.

Required, mutually exclusive parameters:

USER | SOCIE | GENERAL | DANGVAN | SINES | CROSSLAND

Data line to specify a user-defined value of k (USER):

First (and only) line:

1. Normal stress sensitivity constant (k).

Data line to define k based on the general formula (GENERAL):

First (and only) line:

1. Reference load ratio (R_i).
2. Tensile fatigue limit at R_i.
3. Torsional fatigue limit at $R = -1$.

Data line to define k based on the Dang Van formula (DANGVAN):

First (and only) line:

1. Tensile fatigue limit at $R = -1$.
2. Torsional fatigue limit at $R = -1$.

Data line to define k based on the Sines formula (SINES):

First (and only) line:

1. Tensile fatigue limit at $R = -1$.
2. Torsional fatigue limit at $R = -1$.
3. Ultimate tensile strength.

Data line to define k based on the Crossland formula (CROSSLAND):

First (and only) line:

1. Tensile fatigue limit at $R = -1$.
2. Torsional fatigue limit at $R = -1$.

*CLASS

Specify the material class.

This option is used to specify the material class.

If ALGORITHM=findley in the job file, the material class is used along with the normal stress sensitivity constant, k, to determine the value of the modified fatigue shear strength coefficient, ${\tau_f}^*$.

If ndEndurance = 0 in the environment file, the material class is used to determine whether damage is calculated below the material's endurance limit.

Required, mutually exclusive parameters:

WROUGHT STEEL | DUCTILE IRON| MALLEABLE IRON| WROUGHT IRON | CAST IRON | ALUMINIUM | OTHER

There are no data lines associated with this option.

*END MATERIAL

End the definition of a material.

This option is used to indicate the end of a material definition.

There are no parameters or data lines associated with this option.