



NLR-TP-2001-287

## **Computer aided sequencing of loads and stresses for fatigue analysis and testing**

R. Houwink, R.P.G. Veul and H.J. ten Hoeve

This report is based on a presentation held at the ICAF 2001 symposium, dated 27-29 June 2001, Toulouse, France.

This investigation has been carried out partly under contract with the Netherlands Agency for Aerospace Programmes, project number 24914 FA.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Division:	Structures and Materials
Issued:	21 June 2001
Classification of title:	Unclassified



## Contents

1. INTRODUCTION	3
2. THE LOAD DISCRETISATION AND SEQUENCE GENERATION PROCESS	4
2.1 MISSION AND LOADS ANALYSES	4
2.2 LOAD DISCRETISATION AND SEQUENCE GENERATION	5
3. CONDITION CODES	5
4. THE AUTOMATED PROCESS	5
4.1 THE PREPROCESSOR	6
4.2 THE CONDITION CODE TABLE GENERATION PROGRAM	6
4.3 THE CONDITION CODE SEQUENCE GENERATION PROGRAM	7
4.4 THE LOAD AND STRESS SEQUENCE GENERATION PROGRAM	7
4.5 THE LOAD AND STRESS SPECTRUM GENERATION PROGRAM	7
5. CONCLUSIONS	8
6. ACKNOWLEDGMENTS	8
7. REFERENCES	8

5 Figures

(11 pages in total)



## COMPUTER AIDED SEQUENCING OF LOADS AND STRESSES FOR FATIGUE ANALYSIS AND TESTING

- \* R. Houwink (houwink@nlr.nl)
- \* R.P.G. Veul (veul@nlr.nl)
- \* H.J. ten Hoeve (hjth@nlr.nl)

**Abstract:** An automated procedure is presented for the generation of load and stress sequences or spectra for fatigue analysis and testing. The procedure starts with the generation of an initial database with all relevant input data. Different computer programs are developed that use the initial database and discretise exceedance curves, generate load and stress sequences or spectra. The programs are part of the NLR LOADPROC environment, developed to standardise various load processing procedures using a common database structure.

### 1. INTRODUCTION

Aircraft fatigue analysis or testing requires the input of a sequence of loads simulating the history of loading on the aircraft structure during its life. Because the generation of realistic load sequences often is a time-consuming process, at Fokker Aerostructures B.V. the need arose to obtain a software tool to generate these load sequences automatically. In view of the experience of NLR with load sequence generation in the past this resulted in a cooperation with FAE to develop a generic computer program which generates a sequence or spectrum of loads or stresses automatically for an arbitrary aircraft structure. The program is based on the concept of condition codes and on load sequence generation software developed at NLR and used for the Fokker 100 fatigue testing programs in the mid eighties (Ref. 1,2). The new program modules read their input from a database file and write the results generated to the same file. Input data include the build-up of one or more typical flights, each of which is described by a number of segment/loadtype combinations, their load exceedance curves and corresponding load distributions. These input data have to be supplied by the aircraft integrator after a thorough mission and loads analysis of the complete aircraft under consideration. The load sequence process is based on the concept of condition codes, each one describing a loading condition for the complete aircraft. The condition codes are determined after identifying the relevant load events in a mission and discretisation of the corresponding exceedance curves. Load and stress sequences are derived from the condition code sequence using a working factor table, unit load distributions and stress coefficients. Load and stress spectra can be generated immediately from the discretised exceedance curves.

In this paper the loads discretisation and sequence generation process is described, and a summary is presented of the available program modules.

\* National Aerospace Laboratory NLR, Loads and Fatigue Department, Amsterdam, the Netherlands.



## 2. THE LOAD DISCRETISATION AND SEQUENCE GENERATION PROCESS

The load discretisation and sequence generation process can be considered as the process preceding a fatigue analysis or test (figure 1). The goal of the process is to obtain a limited sequence of loads, which on the one hand is sufficiently representative for the loading history of an aircraft or structural component during its service life, and on the other hand allows fatigue analysis or tests at reasonable cost and acceptable accuracy. In the following the process is described in two stages: the initial part with a number of aspects, which cannot be automated, and the second part, which has been automated in the current program modules.

### 2.1 MISSION AND LOADS ANALYSES

The first part of the process consists of a mission analysis and subsequent loads analyses. In the mission analysis the expected usage of the aircraft within the total aircraft life is analysed. Goal of the mission analysis is to define a limited number of typical flights (or mission profiles), which are representative for the operations of the complete fleet and to identify all relevant load events occurring in the various flight phases.

The total aircraft life is split up in a number of equal (test) blocks of flights, all containing the same flight sequence. The highest load in one block is the highest load that will be simulated in the aircraft total life sequence. As a rule of thumb, the total life of the aircraft can be split up in about 20 equal parts. This way extreme loads are eliminated.

The missions of an average aircraft of the fleet during its life are then approximated by a mix of the typical flights, of which the utilisation (or frequency of occurrence) is determined by statistical analyses. For each of these typical flights, a detailed flight or mission profile with regard to altitude, speed, (flap-)configuration and weight is determined. Next, each mission is split up into a number of successive mission segments. In each segment the altitude, speed, configuration and weight are kept a constant. On the basis of the defined segments and load events, loads analyses need to be carried out to obtain mean load distributions and spectra of load variations about the mean condition for each phase of a flight. The spectra are mostly defined as exceedance curves in terms of a master load. A master load is a load quantity that is expected to describe best the fatigue loading of the component considered. The most important master load is called the head master load (Ref. 3).

In summary, the mission and loads analyses should provide the following information, which acts as input to the load discretisation and sequence generation process:

- the total number of flights within the block considered,
- the frequency of occurrence of the typical flights within the block,
- for each typical flight the successive segments, and the load types occurring in that segment, the mean load in each segment,
- the exceedance curves of the incremental load,
- the load distributions corresponding to the mean and deviated states.

This information provides the basic input data for the second part: the discretisation and sequencing of loads and stresses.



## 2.2 LOAD DISCRETISATION AND SEQUENCE GENERATION

On the basis of the above information spectra of occurrences of load types for each flight segment are determined. In order to simulate the different severities e.g. the effect of weather conditions, load spectra per segment may be subdivided into spectra for various severities. A process to generate such spectra is the TWIST procedure described in reference 4.

The generation of a realistic sequence of loads or stresses is a time-consuming process if it is not automated. In order to save time, for the Fokker 100 fatigue tests FORTRAN programs were written performing the generation of load and stress sequences using the concept of condition codes. This concept is the basis for the computer programs programmed in C, presented here. The concept of condition codes is explained in the next section.

## 3. CONDITION CODES

In order to be able to generate load and stress sequences in a practical way, the concept of condition code was introduced for the Fokker 100 fatigue tests by Ben de Jonge in the mid eighties. Each condition code determines a unique state of the aircraft, either a mean steady state or a deviation relative to this state due to gust, taxi, manoeuvre etc. The condition code is a sequence of strings in the form F-S-T-M-D that identifies the load case:

F	- the flight type ID	(e.g. 1=short distance, 2=long distance flight)
S	- the segment ID	(e.g. C=climb, L=landing)
T	- the load type ID	(e.g. LG=load due to lateral gust, T=taxi)
M	- the load magnitude ID	(e.g. ranging from 1 to N)
D	- the load direction ID	(e.g. 1=up 2=down, 3=left, 4=right)

The magnitude identifier of the loading condition refers to a set of working factors and corresponding unit load distributions. The condition codes can be defined after definition of the typical flight profiles (figure 2).

Once all unique, discretised load states of the aircraft are translated into condition codes, a condition code sequence is generated representing a series of successive states of the aircraft. The sequences of loads and stresses in any point of the aircraft then are derived from the condition code sequence using the working factors, the unit load distributions or the stress coefficients. This will be discussed later as part of the automated process.

## 4. THE AUTOMATED PROCESS

The discretisation and sequence generation process is automated using a number of C programs, which communicate with a database. In the database all data are stored as datasets using a standardised format. The following programs can be recognised in figure 3:

- A preprocessor that can be used to generate an initial database. The input for this program is the data obtained from the mission and loads analysis. Also required are the condition code definition and the sequence generation options.
- The program lpct that can be used to generate the Condition Code Tables. This program reads the data prepared by the preprocessor as well as the exceedance curves and generates condition code tables or occurrence spectra and a working factor table.
- The program lpccs that can be used to generate the Condition Code Sequence. This program reads the data prepared by the preprocessor and the condition code tables generated with the lpct program. It generates a condition code sequence.



- The program *lplss* that can be used to generate a Load or Stress Sequence. This program reads the condition code sequence, the working factor table and the unit load distributions and calculate the loads or stresses in a number of user defined stations or points.
- The program *lpfsg* that can be used to generate a Load and Stress Spectrum. It reads the output from the *lpcct* program and generates a load or stress spectrum.
- Finally, not shown in figure 3, the program *lpmon*, a monitor program that reads a database and provides a display of the selected data sets.

All these programs, except the monitor program, will be explained in more detail in the next sections.

#### 4.1 THE PREPROCESSOR

The preprocessor reads all data needed to start the load discretisation and sequence generation process, and creates a database with all data in standardised format (datasets). The preprocessor has to be adjusted to the way the input data are made available by the aircraft integrator. The datasets to be written into the database are the following:

- Block Build-up Table. This dataset describes the way the typical flights are sequenced in a block. It consists of a list the typical flights, the flight type ID for the condition code, and the number of flights of each type.
- Flight Build-up Table(s). This dataset describes a typical flight. It contains a list with the segment/loadtype combination occurring in the typical flight, and references to the exceedance curve table(s), the condition code table(s) (spectra), and seeds table(s). In addition it contains condition code ID's for segment, loadtype and direction. Finally information about the way the condition codes are sequenced (e.g. unrestrained, half-cycle restrained and full cycle restrained, see section 4.3) is given here.
- Exceedance Curve table(s). This dataset describes the exceedance curves per segment/loadtype combination in terms of the head master load as provided by the aircraft integrator. Also a reference to the corresponding unit load distribution is contained here.
- Unit Load Distributions. These datasets contain per loadtype the distribution of sectional loads over the aircraft or structural component.
- Seeds Tables: These datasets contain seeds for random selection of the spectrum severity for each segment/loadtype combination, and for random selection of the load levels from this spectrum. The load level corresponds to the magnitude identifier in the condition code (see section 3).
- Stress Coefficient Table. This dataset contains coefficients that determine local stresses for given sectional loads.

#### 4.2 THE CONDITION CODE TABLE GENERATION PROGRAM

The Condition Code Table generator is a program to discretise exceedance curves, to generate tables of occurrences of condition codes and to generate the working factor table. As input the block- and flight build-up tables and the exceedance curves are used.

It first determines a total exceedance curve by summing up all exceedance curves with the same head master load. For this curve the truncation level, omission level and number of load levels is specified.

Using the resulting load levels all exceedance curves are discretised and optionally saved into the database (DEXC curves) (figure 4). The total exceedance curve is subdivided into a number of exceedance curves having different severities (e.g. weather conditions) in a manner comparable to the TWIST procedure (Ref. 4). Finally the condition code tables (occurrence



spectra) and the working factor table are generated and saved into the database. The latter table contains all unique condition codes, the number of occurrences and the working factors to be applied on the unit load distributions.

#### 4.3 THE CONDITION CODE SEQUENCE GENERATION PROGRAM

Prior to generating a load or stress sequence a condition code sequence is generated. The sequence generation module reads the datasets containing the block- and flight build-up tables, the seeds tables and performs the sequence generation of the condition code for each flight in the block. Each segment starts with a mean condition, followed by all load cycles specified by the selected occurrence spectra. In the sequence generation process load spectra of various types can be mixed (e.g. vertical and lateral gust) and constrained. The following constraint options are available:

- Unrestrained: each load cycle is drawn randomly from the occurrence spectra in the segment;
- Half cycle restrained: each load cycle is followed by a load cycle of opposite direction with arbitrary magnitude,
- Full cycle restrained: each load cycle is followed by a load cycle of opposite sign and having the same magnitude.

The Condition Code Sequence generated is written as a dataset on the database.

#### 4.4 THE LOAD AND STRESS SEQUENCE GENERATION PROGRAM

For a given condition code sequence the corresponding load- and stress sequence can be generated using the load and stress sequence generation module. The module reads the condition code sequence, the working factor table and unit load distributions and calculates the load corresponding to each condition code for the section(s) specified by the user. The load distribution corresponding to a condition code is calculated as the sum of the products of working factors and unit load distributions. An example of a load sequence generated for a flight segment is given in figure 5.

A stress sequence is generated by reading the stress coefficients and multiplying them by the loads for the section specified. The load or stress sequences are written to the database as datasets.

#### 4.5 THE LOAD AND STRESS SPECTRUM GENERATION PROGRAM

Using the information generated by the Condition Code Table generation program, load and stress spectra can be generated immediately without generating a sequence first. The program *lpfsg* can be used for this operation. It reads the working factor table, and unit load distributions to generate a load spectrum for the section(s) specified by the user. A stress spectrum is generated by reading the stress coefficients and multiplying them by the loads for the section specified. The load or stress spectra are written to a separate file.



## 5. CONCLUSIONS

A cluster of computer programs has been developed for computer aided discretisation of load spectra and subsequent generation of load and stress sequences for fatigue testing and analysis. The programs form part of the NLR LOADPROC environment developed to standardise various load processing procedures using a common database structure. In the present paper the load and stress sequence generation process and the programs are described. These programs provide an efficient way to generate load and stress sequences for both analysis and testing.

## 6. ACKNOWLEDGMENTS

The authors wish to thank Mr. Ben de Jonge for his advice and support in realising the present programs.

## 7. REFERENCES

1. Jonge, J.B. de, van der Linden, H.H., ten Have, A.A. ten; "Test load specification for the damage tolerance test on the Fokker 100 wing/fuselage (TA-15), Issue 3". NLR CR 85137 C, 1985.
2. Jongebreur, A.A. Louwaard, E.P., van der Velden, R.; "Damage tolerance test program of the Fokker 100". Paper presented at 13th ICAF conference, Pisa, Italy, 1985.
3. Jonge, J.B. de, Vink, W.J.; "Determination of gust load sequences for analysis and fatigue tests". NLR CR 95387 L, 1995.
4. Jonge, J.B. de, Schütz, D., Lowak, H., Schijve, J.; "A standardized load sequence for flight simulation tests on transport aircraft wing structures". LBF - Bericht FB-106. Also: NLR TR 73029 U, 1973.

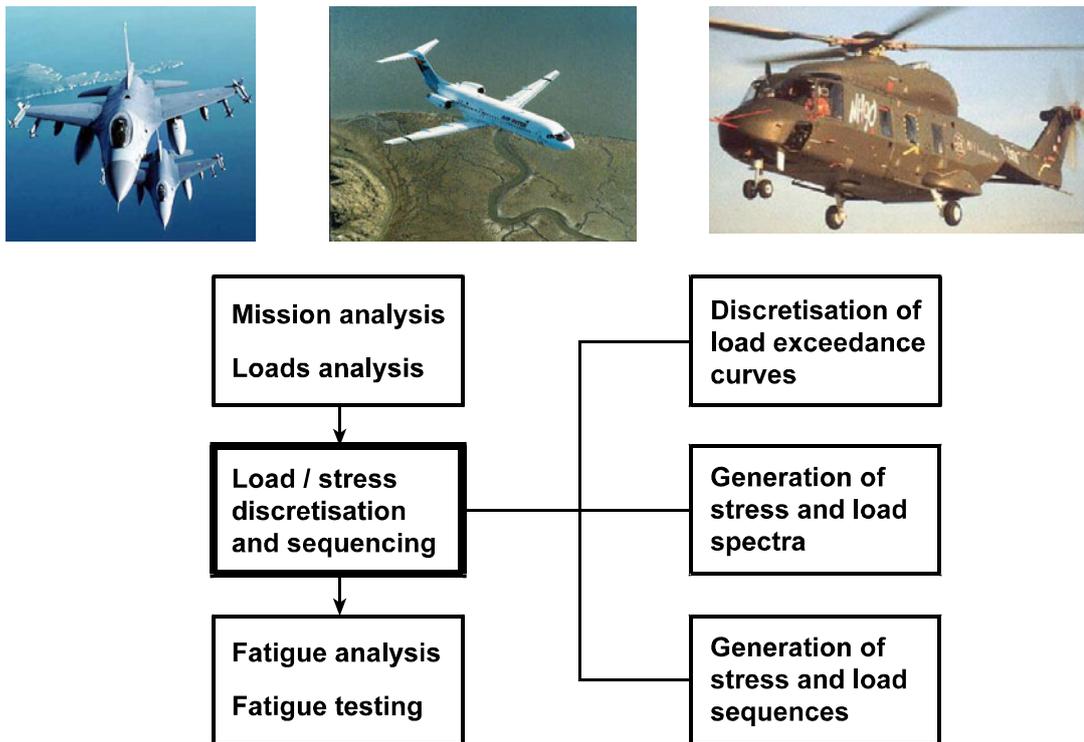


Fig. 1 Phases in the aircraft loads and fatigue analysis process

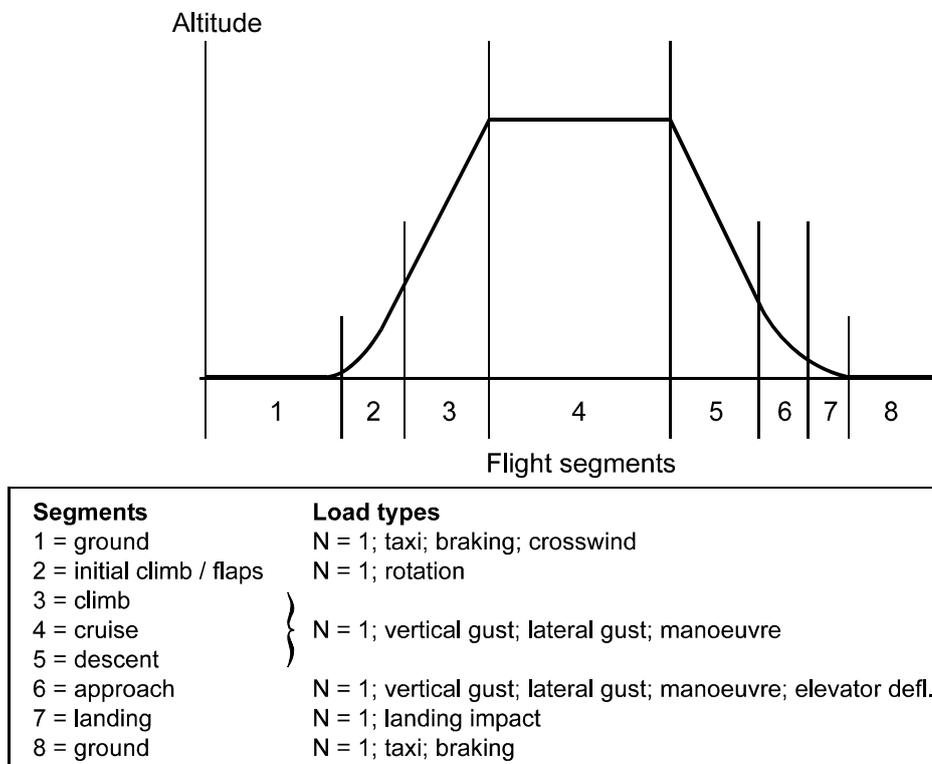


Fig. 2 Example of typical flight profile

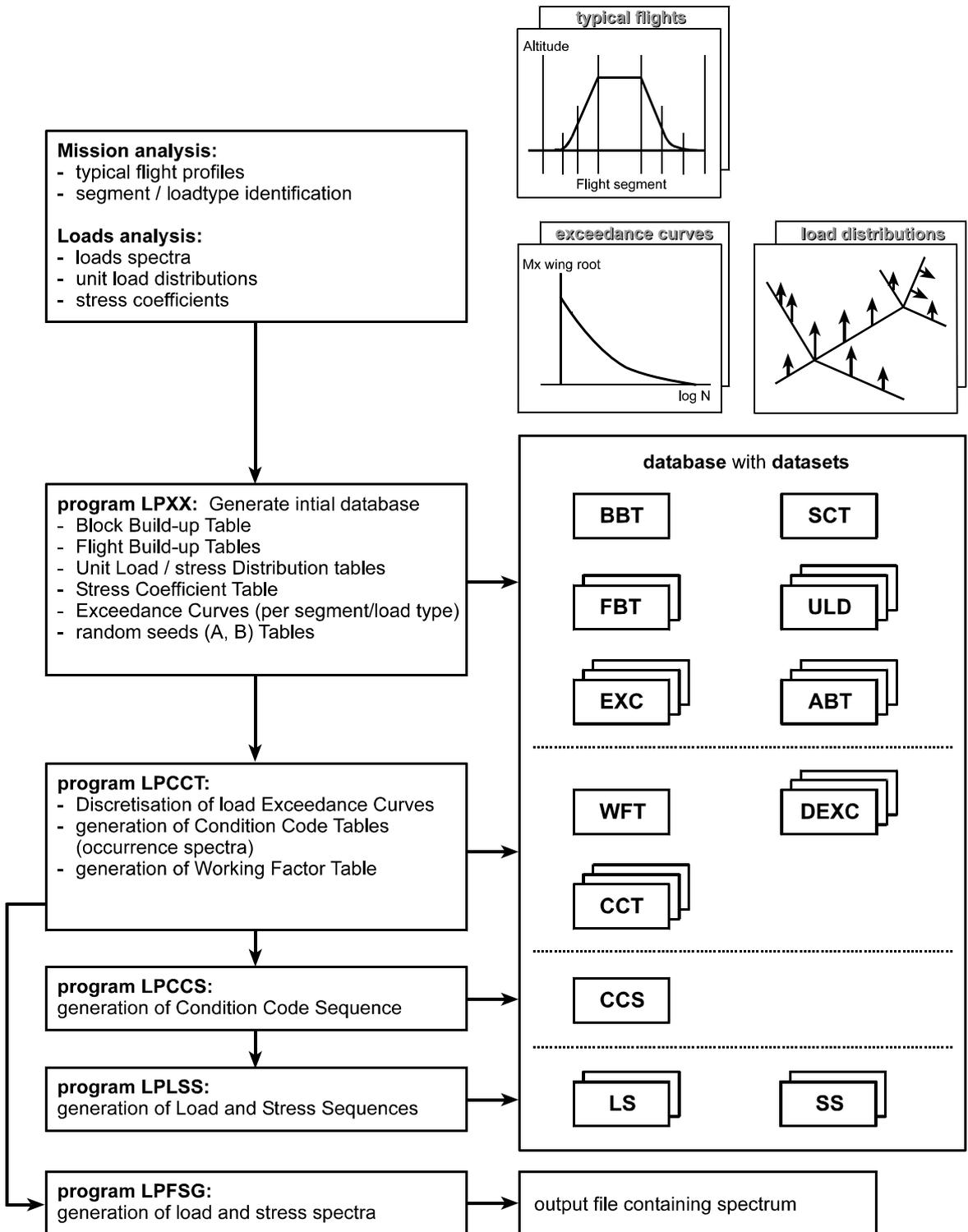


Fig. 3 Scheme of automated load sequencing process

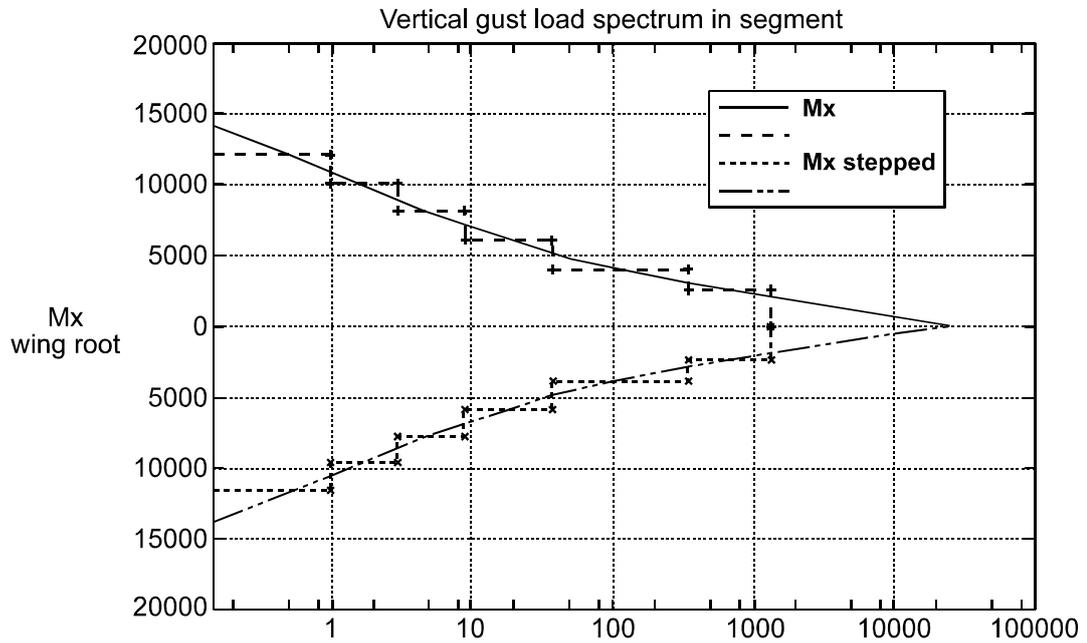


Fig. 4 Stepped exceedance curve for a segment

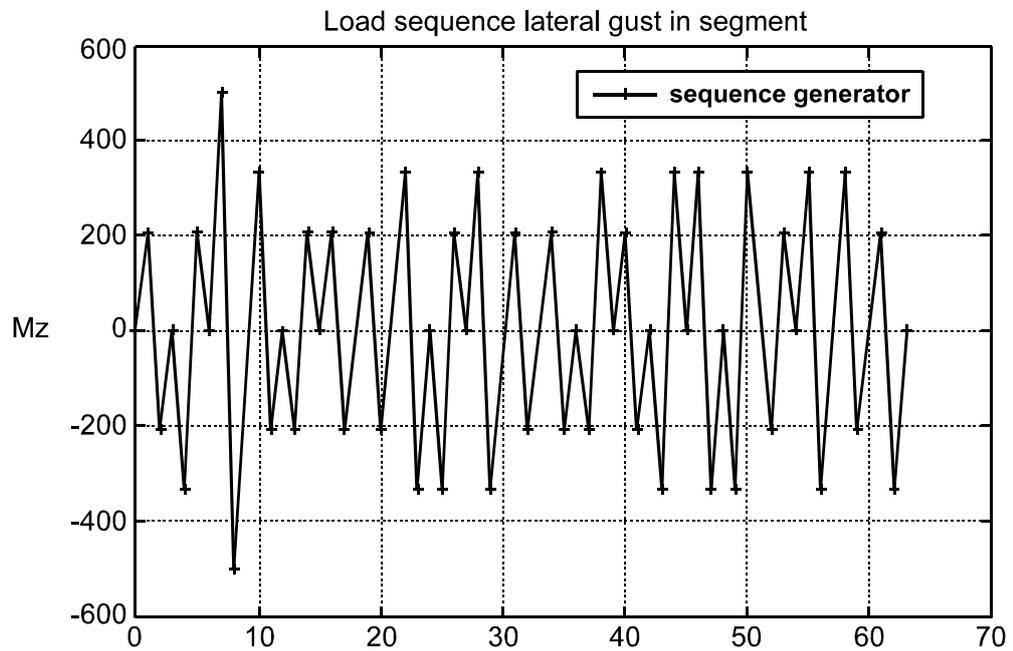


Fig. 5 Example of a load sequence for a segment