



ESDU 00039

Issued November 2002
With Amendments A to C
October 2013

Aircraft Performance Program Part 9: Testing programs APP02 and APP03



Endorsed by
The Royal Aeronautical Society

AVA Documents
Document Sharing Hub

ESDU DATA ITEMS

Data Items provide validated information in engineering design and analysis for use by, or under the supervision of, professionally qualified engineers. The data are founded on an evaluation of all the relevant information, both published and unpublished, and are invariably supported by original work of ESDU staff engineers or consultants. The whole process is subject to independent review for which crucial support is provided by industrial companies, government research laboratories, universities and others from around the world through the participation of some of their leading experts on ESDU Technical Committees. This process ensures that the results of much valuable work (theoretical, experimental and operational), which may not be widely available or in a readily usable form, can be communicated concisely and accurately to the engineering community.

We are constantly striving to develop new work and review data already issued. Any comments arising out of your use of our data, or any suggestions for new topics or information that might lead to improvements, will help us to provide a better service.

THE PREPARATION OF THIS DATA ITEM

The work on this particular Data Item was monitored and guided by the Performance Committee. This Committee first met in 1946 and at the time of first issue of this Item had the following membership:

Chairman	
Mr K.J. Balkwill	— Independent
Members	
Mr P.V. Aidala*	— Northrop Grumman ESID, Bethpage, New York, USA
Mr T. Bartup	— BAE SYSTEMS, Regional Aircraft, Woodford
Mr M. Broad	— Independent
Mr G.M.J. Davis	— Defence Evaluation and Research Agency, Boscombe Down
Mr N.J. Herniman	— Airbus Industrie, Toulouse, France
Mr R.G. Humpston	— Rolls-Royce plc, Aero Division, Derby
Mr T.S.R. Jordan	— Independent
Mr R.J. Orlowski*	— Boeing Commercial Airplane Company, Seattle, Washington, USA
Dr P. Render	— University of Loughborough
Mr P. Renze*	— Raytheon Aircraft Company, Wichita, Kansas, USA
Mr D.N. Sinton	— Independent
Mr G.J.R. Skillen	— Civil Aviation Authority, Safety Regulation Group, Gatwick
Mr G.E. Smith	— Independent
Mr A. Stanbrook	— Independent
Mr R. Storey	— BAE SYSTEMS, Military Aircraft and Aerostructures, Brough
Prof. E. Torenbeek*	— Delft University of Technology, The Netherlands
Mr S. Tsach*	— Israel Aircraft Industries, Jerusalem, Israel
Mr C.J. Turner*	— Boeing, Douglas Products Group, Long Beach, Calif., USA
Mr M. Wilson	— Britten-Norman Ltd, Isle of Wight.

* Corresponding Member.

The construction and subsequent development of the Data Item was undertaken by

Mr A.J. Grundy — under contract to ESDU.

The person with overall responsibility for the work in this subject area was Mr D.J. Mitchell.

AIRCRAFT PERFORMANCE PROGRAM
PART 9: TESTING PROGRAMS APP02 AND APP03

CONTENTS

	Page
1. INTRODUCTION	1
2. PHILOSOPHY	1
3. PROGRAM CHECKING	2
3.1 Testing Data Input	3
3.2 Testing Performance Calculations	3
3.2.1 Major service routines	3
3.2.1.1 Atmosphere	4
3.2.1.2 Speed conversion	4
3.2.1.3 Tables (Reference 11, Section 3)	4
3.2.1.4 Iteration	5
3.2.1.5 Optimum search	6
3.2.1.6 Integration	6
3.2.2 Aircraft performance	6
3.2.2.1 Rough testing	7
3.2.2.2 Initial testing	7
3.2.2.3 Final testing	8
3.2.2.3.1 Point Performance	8
3.2.2.3.2 Path Performance	9
3.2.2.3.3 Utility Options	10
3.2.2.4 Other checks	10
3.3 Testing APP03	11
4. ISSUED VERSION	12
4.1 Testing Issued Version	12
5. REFERENCES	14

ESDU Copyright material. For current status contact ESDU.

AIRCRAFT PERFORMANCE PROGRAM

PART 9: TESTING PROGRAMS APP02 and APP03

1. INTRODUCTION

This Data Item outlines the test procedures implemented to design, code and validate the various components of the ESDU Aircraft Performance Program.

The programs considered are

- APP02, the Flight Performance Program,
- APP03, the Interpolation Program.

2. PHILOSOPHY

The programs are written in ANSI FORTRAN. The FORTRAN coding is organised to control the common blocks used; all common blocks are fully declared when used and a reference set is maintained.

The sections of the program are first developed in a Program Description Language (PDL) (see Reference 13) that describes the program organisation and logic. The FORTRAN coding is then derived from these descriptions. In initial testing, and later during more detailed analysis, errors in the coding may be detected and corrected. Subsequently, the PDL is revised in conjunction with the modifications made to the FORTRAN coding. The program logic can be re-examined and the program organisation refined.

Significant events, problems and solutions to problems which occur during the development and testing of the programs are recorded. A list is maintained of improvements to be made to the programs to add extra facilities, to make them more flexible or to ensure that they are more robust and error-proof; these improvements are incorporated as new versions of the program(s) are produced.

Program APP02 is designed to allow flexibility in the order of data input, the form of data input and the input and output units used – all calculations are performed in SI units. All input data and data tables are rigorously checked for syntax, limits on the input values and completeness of the data required for the chosen calculation – default values may be set to meet data input requirements. Iterative solutions are used for all the performance options to ensure that table variables are correct and form a coherent set. The standard output provides sufficient information to enable understanding and at-a-glance interpretation of the results; debugging output options are also provided to allow detailed inspection of the calculation process.

3. PROGRAM CHECKING

For all stages of development testing the programs are compiled with options to check that

- all variables are declared explicitly,
- no variable is declared with the same type more than once in the same program unit (this ensures that the same variable name is not used for two different items),
- each variable and array element in a program has been assigned a value before its contents are accessed (for example, as the right hand side of an assignment statement),
- array bounds are not exceeded,
- character variables are the correct size and their sub-strings are correct (this ensures that enough storage space has been allocated and that illegal sub-string operations are not used),
- calculation overflow does not occur,
- no zero-trip DO-loop is executed,
- subroutine and function arguments are consistent.

Note: The issued executable programs are derived from the development-tested programs **but do not incorporate these checks** (see Section 4).

In the initial stages the programs are run using simple test data. Any compiler errors detected (above) are corrected until successful operation is achieved. Hard-copy printouts of the FORTRAN code are then examined line-by-line ('walked through') using typical test case data to examine the program logic.

With the FORTRAN compiler checks included, the FORTRAN code is then executed line-by-line ('traced through') using the FORTRAN debugger to ensure that program internal markers, array contents *etc.* are correctly set. The programs are further tested with a series of data files to check

- input data options,
- error messages,
- correct expansion of permutations and combinations,
- performance calculations.

The output from these development test cases is compared initially with the results obtained from hand calculations, and makes use of the program debug facilities (*e.g.* DEBUG 5) to obtain calculated values to six significant figures. The files (input data, output data and any other files produced) are retained for future testing. Upon updating the contents of the program(s), the retained test cases are re-solved to verify that the proposed modifications have been implemented with no unintentional consequences in the program output. When the program is deemed to be functioning correctly, the modified files replace the previously-stored versions.

Hard-copy printouts of the final development test case files for each issued version of the programs are retained for future reference. These copies show the cases checked and the results of the hand calculations. ESDU Memoranda have been developed from these cases (References 14 to 16), and can be obtained upon request.

Upon implementing the necessary modifications to the relevant source files, the program versions planned for issue are compiled and executed. Each program is run using copies of the retained test case data files and the output is compared with that obtained from the corresponding development version for verification purposes (see Section 4).

3.1 Testing Data Input

Testing the data input follows a systematic order and makes use of the program debug output facilities to inspect the data. A series of test data files is used to check that all keywords are identified, that the values (or tables used) are stored in the corresponding array elements with the correct conversion to SI units and that the combination and permutation data are stored appropriately. The use of factors and increments on tables and special speeds is checked. The memory required to store table names, permutation and combination data *etc.* is compared with the relevant array capacity limits.

This testing uses unique values or table names for every item of data input, tests all available forms of data input (single value, range, list, permutation, combination, data table) and checks that the use of upper- or lower-case text in the keyword input does not interfere with the functioning of the program. The data input sequence is varied and includes multiple input of keywords and the use of comment lines. The use of table factors and increment tables (MODTAB variables), keyword synonyms (*e.g.* STARTHP, HP, XNMAIN) and the special keywords (CANCEL, DEFAULT, DEBUG, END_CASE and END_DATA) is checked.

Testing is conducted to ensure that the program can detect erroneous data and call on the appropriate error message(s).

This set of test case input and output files is retained for future testing.

3.2 Testing Performance Calculations

Testing the Flight Performance Program follows a systematic order and makes use of the debug output to output calculation values to six significant figures.

3.2.1 Major service routines

Before the accuracy of the performance calculations is confirmed the major service routines are examined. These routines are,