

APPLICATION OF MULTIVARIATE OPTIMISATION TECHNIQUES TO DETERMINATION OF OPTIMUM FLIGHT PATH TRAJECTORIES

1. NOTATION*

$c_i(X)$	constraint functions ($i = 1, \dots, n$)
$f(X)$	objective function, <i>i.e.</i> function to be minimised
H	geopotential height
j	numbers identifying independent variables
k	number of independent variables
M	Mach number
m	number of equality constraints
n	total number of constraints
t	time
V	true airspeed
V_c	calibrated airspeed (CAS)
X	vector or independent variables, ($x(1), \dots, x(j), \dots, x(k)$)
$x(j)$	independent variables ($j = 1, \dots, k$)

Subscripts

cl	denotes climb
cr	denotes cruise
des	denotes descent
i	numbers identifying constraint functions
md	denotes minimum drag
min	denotes minimum value

Superscripts

L	denotes a lower bound
U	denotes an upper bound

* This Data Item departs from the usual practice of providing a comprehensive Notation and Units Section. This is done for the following reasons.

- The only equations quoted are in general terms (*i.e.* not involving dimensional quantities) and many quantities are referred to in words rather than symbols. The exceptions are for the sake of brevity (or space available).
- Values quoted in Tables 5.1 to 5.4, Figures 1 to 3, Appendix A, Sketches A1 and A2, and Tables B7.1 and B7.4 are expressed in widely-used operational units which do not form a coherent set. The common abbreviations for these units are employed only in those cases where no confusion of meaning is likely.

Issued June 1994
With Amendment A, March 1995

2. INTRODUCTION

Data Item No. 93021 (Reference 3) provides examples of the derivation of optimum flight paths by use of a constrained multivariate optimisation program. The present Data Item deals with the principles and difficulties of formulation of a problem for solution by this method.

Constrained multivariate optimisation is used to solve the problem of optimising a function (the *objective function*) subject to the variables lying between simple upper and lower bounds and satisfying user-specified constraints (the *constraint functions*). In this Data Item, the term “*optimiser*” is used to describe any software capable of solving constrained multivariate optimisation problems. The particular constrained multivariate optimisation program used in Reference 3 is RQPMIN*.

Section 3 presents a general description of the constrained multivariate optimisation process. Appendix A discusses difficulties that might arise in formulation of the problems (based on experience of setting up the complete sortie optimisations dealt with in Reference 3) and Appendix B discusses the specific requirements of RQPMIN.

WARNING: There is no guarantee that any numerical process will locate an absolute optimum in a single search and it is always necessary to consider the use of more than one search, starting from widely different initial points, to check the validity of the result. See Section 3.3.

3. OPERATION OF OPTIMISER PROGRAMS

3.1 General Description

Mathematically, the optimisation problem to be solved can be stated as follows.

$$\text{Minimise}^{\dagger} f(X) \quad (3.1)$$

where $f(X)$ is the objective function and

X is the vector of the independent variables, $(x_1, \dots, x_k)$,

$$\text{subject to } c_i(X) = 0 \quad (i = 1, \dots, m), \quad (3.2)$$

$$c_i(X) \leq 0 \quad (i = m + 1, \dots, n), \quad (3.3)$$

$$\text{and } x_j^L \leq x_j \leq x_j^U \quad (j = 1, \dots, k), \quad (3.4)$$

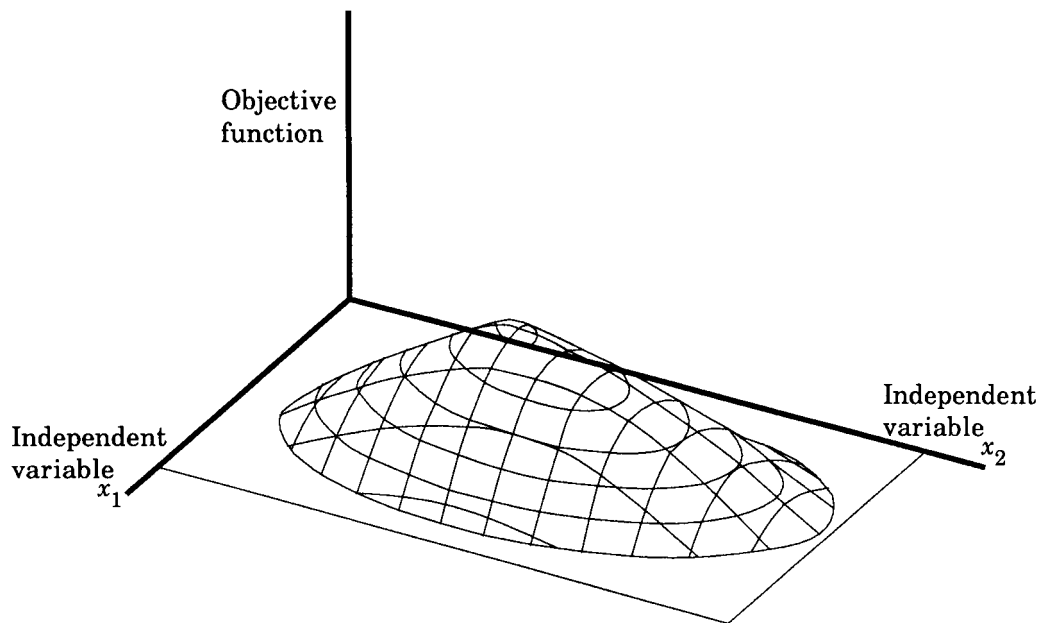
where c_i are the constraints.

As the independent variables, x_j , are varied the objective function can be considered as defining a surface in $(k + 1)$ -dimensional space. This surface is referred to as the *design surface* (shown in Sketch 3.1 for $k = 2$). The region where the constraints are satisfied is known as the *feasible region*. The periphery of the feasible region is known as the *constraint boundary*.

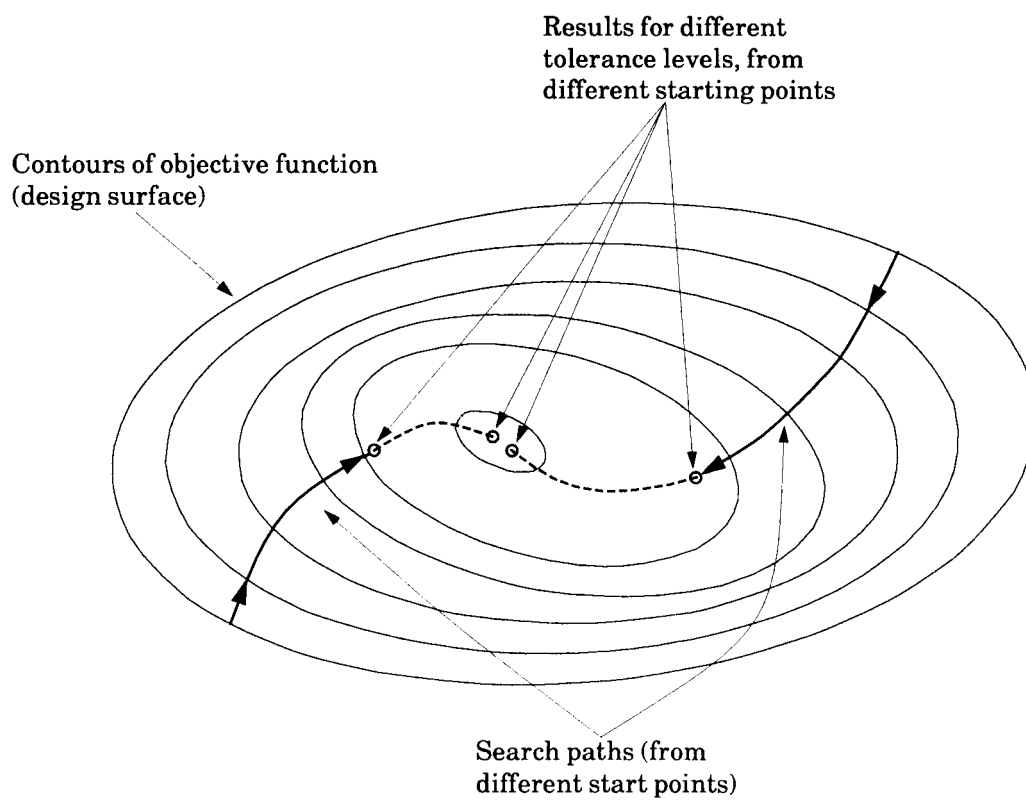
The values of the independent variables which correspond to the solution are referred to as their *optimum* values.

* RQPMIN Version 1, January 1986 (Reference 1) developed at the Royal Aircraft Establishment (now DRA), Farnborough.

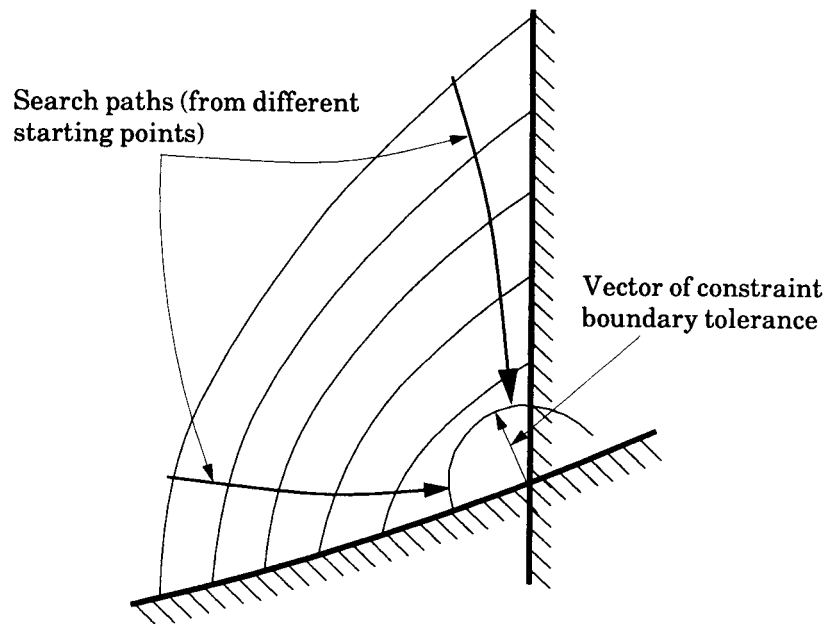
† Note that a problem requiring maximisation of $f(X)$ can be expressed in the same form, by replacing $f(X)$ by $-f(X)$.



Sketch 3.1 Design surface



Sketch 3.2 Objective function tolerance



Sketch 3.3 Constraint boundary tolerance

3.1.1 Scale factors

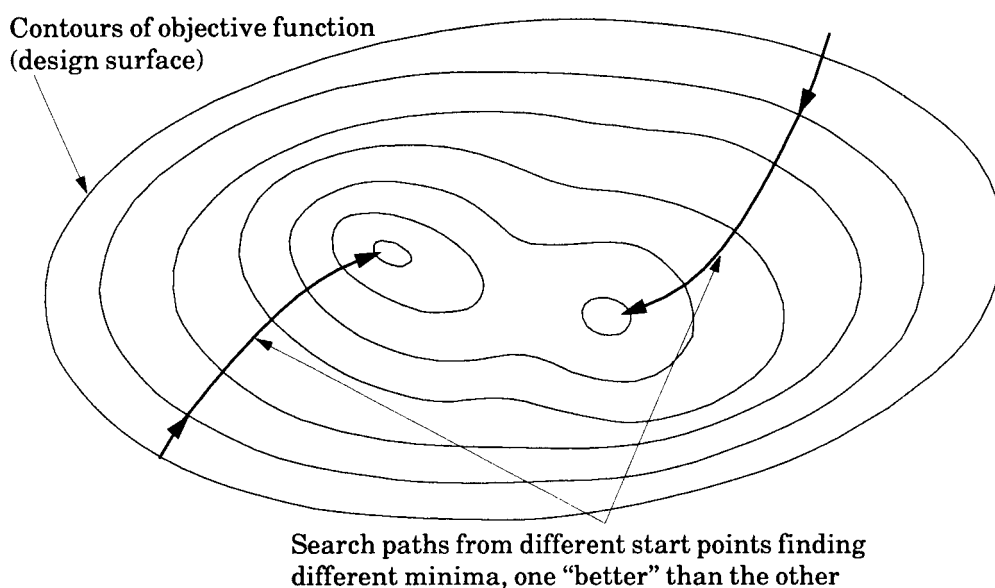
All variables, constraints and the objective function are normalised by the introduction of scale factors. In general, these scale factors are chosen so that all variables are reduced to the order of unity. Each variable then has equal influence on the design surface search. However, it is possible to choose other values for the scale factors in order to alter the relative weight of the variables in the optimisation process.

3.1.2 Tolerances

It is unlikely that any type of numerical optimisation process will find an absolute minimum point. The process always has to be judged complete when some criteria with respect to the objective function and the constraints are satisfied within chosen tolerances – see Sketches 3.2 and 3.3.

Assignment of the tolerance value is a critical aspect in the definition of the problem. If the tolerance is too fine it might not be possible to complete the search in an acceptable number of steps. On the other hand, if the tolerance is too large, the accuracy of the minimum will be suspect. Solutions obtained from widely differing starting points can indicate the effect of too large a tolerance by producing results that are significantly different (Sketch 3.2). However, it should be noted that different results might also arise from the existence of local sub-optima (Sketch 3.4).

The exact solution may lie *on the constraint boundary*, with one or more of the constraints being met exactly. In such a case, it is unlikely that a numerically generated solution will lie exactly on the constraint boundary. Another tolerance is therefore required to specify the acceptable displacement of the minimum point from the boundary. The same considerations apply to this tolerance as for the objective-function tolerance (Sketch 3.3).



Sketch 3.4 Possible effect of existence of local sub-optimum point

3.1.3 The optimisation process

Optimiser programs are of many types (see Reference 6), mostly using some form of gradient-search procedure, constraints being accounted for by the introduction of a *penalty function* added to the objective function. At various stages of the search this penalty function is “tightened” to allow the search to move closer to the constraint boundary.

The method used by RQPMIN is essentially that known as *sequential or recursive quadratic programming*. RQPMIN has its own method of ensuring that the constraints are met, which does not involve the use of a penalty function.

It should be noted that the optimum does not necessarily lie *on the constraint boundary*, unlike those constrained optimisation problems in which the objective function and the constraints are all linear functions of the independent variables and which are solvable by the process known as “linear programming”.

3.1.4 Feasible region

The search commences from an initial point specified by the user (*i.e.* initial values are required for each variable). This point need not necessarily satisfy all the constraints (*i.e.* it can be outside the feasible region) provided that the optimisation method does each of the following.

Checks for feasibility,

includes a method by which the values of the variables can be modified (*i.e.* the point can be moved) in order to satisfy the constraints.

NOTE that it is possible for a problem to be such that a feasible region does not exist.

Substantial computer time can be involved in moving the initial point into the feasible region. For this reason it is desirable to start from a point that is likely to be within the feasible region. For systematic

studies, a suitable initial point might be the output values of the variables from the previous study. Additional runs may be necessary to guard against these effects (and against the possibility of local sub-optima). Such searches for a solution should be started from several widely different points. In some studies it may be necessary to reduce tolerances in order to avoid masking small changes under investigation.

3.1.5 User-Written Subroutine(s)

The user must provide a subroutine or subroutines defining the problem to be solved. For instance, in the problems considered in Reference 3, the subroutine consists of a series of calls to other subroutines which calculate the various segments (acceleration, climb, cruise, *etc.*) of the sortie, followed by an evaluation of the block time, range, fuel, or direct operating cost which constitutes the objective function, *i.e.* the function to be optimised. The subroutine also calculates the values of the constraint functions.

3.2 Formulation of the problem

Experience shows that there are many pitfalls to avoid in the formulation of a problem to be solved by these methods. In particular, calculation of the problem functions (*i.e.* the objective function and the constraint functions) can be set up in a way that would normally suffice for a single calculation of the function. However, it is then possible that various aspects of the algorithm employed can interfere with operation of the optimiser. Appendix A discusses various points that must be considered in writing the subroutines so as to avoid these problems.

3.3 Results

3.3.1 Validity of results

It may be necessary to investigate the effect on the solution of repeating the search but starting from a different initial point (see Section 3.1.2). This can help to determine whether a solution has been affected by tolerances that are too large or by the presence of local sub-optima.

Additional searches starting from different initial points are always advisable in an isolated study. However, in a series of studies it is possible to use trends in the results to determine whether a further search is required. For instance, Tables 5.1 to 5.4 show a summary of the complete set of the optimisation runs used to produce the results presented in Table 8.1 of Reference 3. The differences between the two tables correspond to the extra runs required to reach a satisfactory conclusion, from which it is possible to illustrate the process of determining whether further runs are necessary.

Thus, in the “Minimum fuel for 3 500 n.miles range” example, the first run for Case 5 terminated with a descent speed of 287.9 knots CAS compared with about 235 knots CAS in the previous cases. Descent-related results were also markedly different (descent time, in particular). The run was repeated with initial values close to the expected final values, giving the result listed here as “Case 5a” (and as “Case 5” in Reference 3). In this result the constraint on minimum V/V_{md} is exceeded marginally (1.0483 compared with the required minimum value of 1.05). Further runs might have eliminated this but an error of 0.2% is acceptable. (See the comments on constraint tolerances in Section B6.1.)

However, the first run in Case 7 resulted in much larger violations of all three constraints on minimum speed and so required an extra run. The result of this run then failed to meet the cabin inclination constraint and yet another run was necessary. The final run, labelled “Case 7b” here (and “Case 7” in Reference 3) met the constraints.

Similar needs for extra runs arose in all four examples and can be found in Tables 5.1 to 5.4. Those values from the output which indicated (or helped to indicate) the need for further runs are shown by underlining.

Case 7 in the “Minimum time for 2 000 n.miles range” example and Case 6 in the “Minimum DOC for 2 000 n.miles range” example both terminated without having achieved the required range. In both examples, satisfactory results were achieved in subsequent runs.

3.3.2 Nature of the minimum

The form of the design surface near the minimum can be determined by calculating values* for the objective function around the minimum point, *i.e.* varying one independent variable whilst holding the rest fixed, and repeating this for each individual variable. If the variation of the objective function is then compared with the variation of appropriate constraint functions, it is possible to see how the constraints affect the minimum.

The results of this exercise for Case 5 in the “Minimum fuel for 3 500 n.miles range” example are shown in Figures 1 to 3. In this example, zero fuel mass remains constant throughout and, hence, minimum fuel corresponds to a minimum value for ramp mass. The upper parts of Figures 1, 2, and 3 show how ramp mass varies with the six sortie parameters around their individual optimum values. (Note that the ramp mass scale has to be very open in order to show the variations.) The only constraints having any influence on the minimum in this example are the minimum values for V/V_{md} . The corresponding variations of V/V_{md} with the sortie parameters are shown in the lower parts of Figures 1, 2, and 3.

Figure 1 shows the effect of variation in climb and descent speed. The optimum speed for climb is at the speed for minimum ramp mass; V/V_{md} is not affected by climb speed above about 270 knots CAS because of the constraint $V/V_{md} \geq 1.05$. On the other hand, minimum ramp mass would be at its minimum with respect to descent speed at about 210 knots CAS but, at this speed, V/V_{md} is well below the constraint at only 0.94. The optimum value for descent speed is 234 knots CAS, where V/V_{md} first reaches 1.05.

Figure 2 shows the effect of variation in climb, cruise, and descent Mach number. In every case the minimum ramp mass is at a lower Mach number than the selected optimum of the V/V_{md} constraint although for climb and descent the curves are extremely flat.

Figure 3 shows the effect of variation in cruise height. Again, at the height for minimum ramp mass (about 35 000 ft) V/V_{md} has a value (1.022) well below the constraint value. The selected optimum height is 33 672 ft, at which $V/V_{md} = 1.05$.

In summary, examination of the shape of the design surface around the optimum point shows whether the design surface is well behaved, *i.e.* without discontinuities, to what extent the optimum is affected by the constraints and to what extent it corresponds to a minimum of the objective function with respect to the variables. In the particular illustration, the optimum is limited only by the constraint $V/V_{md} \geq 1.05$ and is a minimum with respect to all other variables.

* Using only the subroutine(s) defining the objective function, not the optimiser.

4. REFERENCES

1. SKROBANSKI, J.J. User Guide for RQPMIN (a program for constrained optimisation). RAE Tech. Memor. Aero 2059, January 1986.
2. SKROBANSKI, J.J. The Method of RQPMIN (a program for constrained optimisation). RAE unpublished working paper, May 1988.
3. ESDU Examples of flight path optimisation using a multivariate, gradient-search method. Data Item No. 93021, December 1993.
4. LEE, C. DRA Farnborough. Private communication, October 1993.
5. SKROBANSKI, J.J. The MVA Consultancy. Private communication, May 1994.
6. SCALES, L.E. *Introduction to Non-linear Optimization*. Macmillan, London, 1985.

Reports quoted are not necessarily available to members of the public or commercial organisations.

5. TABLES GIVING RESULTS OF OPTIMISATION RUNS

TABLE 5.1 Minimum Fuel For 3500 n.miles Range

Case	Cruise type	Minimum V/V_{md} required	Ramp mass kg	Block fuel kg	Time mins	Range n.miles	Cost function \$	Input values for optimisation					
								V_{ccl} knots CAS	M_{cl}	M_{cr}	Cruise height (start) ft	V_{cdes} knots CAS	M_{des}
0	Cruise climb	—	114449	29828	498.3	3500		300.0	0.7000	0.7000	35000	300.0	0.7000
1	Cruise climb	1.00	114460	29839	496.2	3500		300.0	0.7000	0.7000	35000	300.0	0.7000
2	Cruise climb	1.05	114803	30105	486.1	3500		300.0	0.7000	0.7000	35000	300.0	0.7000
3	Cruise climb	1.05	114799	30100	486.1	3500		300.0	0.7000	0.7000	35000	300.0	0.7000
4	Cruise climb	1.05	114858	30115	486.3	3500		300.0	0.7000	0.7000	35000	300.0	0.7000
5	Cruise climb	1.05	115076	30273	481.8	3500		300.0	0.7000	0.7000	35000	300.0	0.7000
5a	Cruise climb	1.05	114927	30131	486.4	3500		315.0	0.7650	0.7650	33600	235.0	0.7600
6	Stepped	1.05	115938	31095	485.9	3500		315.0	0.7580	0.7580	33000	235.0	0.7280
7	Stepped	1.05	115149	30342	481.1	3500		315.0	0.7580	0.7580	37000	235.0	0.7290
7a	Stepped	1.05	116571	31697	467.9	3500		318.0	0.8000	0.8140	37000	235.0	0.0800
7b	Stepped	1.05	116531	31659	468.3	3500		315.3	0.7990	0.7990	37000	235.0	0.8090
8	Level	1.05	116251	31392	483.7	3500		315.0	0.7580	0.7580	33000	235.0	0.7280

Case	Cruise type	Output values from optimisation							Minimum V/V_{md}			Buffet margin (increment in load factor)
		V_{ccl} knots CAS	M_{cl}	M_{cr}	Cruise start	height ft end	V_{cdes} knots CAS	M_{des}	Climb	Cruise	Descent	
0	Cruise climb	315.3	0.7507	0.7514	35046	40664	211.2	0.6529	0.9968	0.9979	0.8593	0.152
1	Cruise climb	315.8	0.7523	0.7522	34973	40596	222.9	0.7482	1.0009	1.0009	1.0000	0.439
2	Cruise climb	314.8	0.7653	0.7652	33691	39396	237.3	0.7620	1.0500	1.0499	1.0502	0.548
3	Cruise climb	314.5	0.7658	0.7657	33732	39433	234.4	0.7813	1.0500	1.0500	1.0500	0.556
4	Cruise climb	315.5	0.7654	0.7654	33695	39396	234.5	0.7778	1.0500	1.0500	1.0500	0.558
5	Cruise climb	314.8	0.7650	0.7649	33613	39359	<u>287.9</u>	0.7614	1.0500	1.0500	1.0500	0.550
5a	Cruise climb	314.3	0.7653	0.7653	33672	39375	234.2	0.7620	1.0500	1.0500	1.0483	0.547
6	Stepped	321.1	0.7593	0.7670	33000	37000	235.1	0.7294	1.0500	1.0500	1.0504	0.539
7	Stepped	315.0	0.7912	0.7761	37000	41000	230.1	0.7813	<u>1.0348</u>	<u>1.0011</u>	<u>1.0284</u>	0.351
7a	Stepped	315.3	0.7988	0.7991	37000	37000	235.1	0.8089	1.0501	1.0511	1.0500	0.331
7b	Stepped	310.2	0.7987	0.7987	37000	37000	235.1	0.8196	1.0500	1.0500	1.0500	0.315
8	Level	317.7	0.7601	0.7601	33000		234.6	0.7250	1.0500	1.0500	1.0500	0.585

Case	Cruise type	Pressure differential*				Descent time mins	Cabin inclination degrees		
		Climb	Cruise		Descent		Climb	Cruise	Descent
			Start	End					
0	Cruise climb	6.73	7.47	8.29	4.08	29.86	8.39	2.24	−3.41
1	Cruise climb	6.76	7.46	8.29	4.32	28.86	8.09	2.18	−3.70
2	Cruise climb	7.12	7.24	8.13	4.95	26.20	8.38	1.86	−4.06
3	Cruise climb	7.09	7.25	8.13	4.80	26.73	8.38	1.86	−4.07
4	Cruise climb	7.11	7.24	8.13	4.80	26.70	8.37	1.92	−4.05
5	Cruise climb	7.11	7.23	8.12	<u>8.11</u>	<u>18.75</u>	8.09	1.91	−4.60
5a	Cruise climb	7.10	7.24	8.13	4.80	26.70	7.50	1.89	−2.49
6	Stepped	7.18	7.12	7.78	4.99	25.31	7.39	1.90	−2.48
7	Stepped	5.27	7.78	8.33	4.55	28.15	9.75	2.24	<u>−5.19</u>
7a	Stepped	4.87	7.78	7.78	4.55	25.32	9.73	2.05	<u>−9.54</u>
7b	Stepped	4.86	7.78	7.78	4.98	25.32	7.61	2.06	−4.13
8	Level	7.16	7.12		5.19	23.17	7.23	1.90	−0.20

* 1 lbf/in² = 6.9 kN/m².