CRANFIELD UNIVERSITY

MATTHEW GREAVES

MODELLING OF WRECKAGE TRAJECTORIES FOLLOWING IN-FLIGHT BREAKUP

SCHOOL OF ENGINEERING

MSc by RESEARCH THESIS

Academic year: 2012-2013

Supervisor: Prof G. Braithwaite

April 2013

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

MSc by RESEARCH THESIS

Academic Year 2012-2013

M GREAVES

MODELLING OF WRECKAGE TRAJECTORIES FOLLOWING IN-FLIGHT BREAKUP

Supervisor: Prof. G Braithwaite

April 2013

Abstract

The study of ballistic trajectories is well-established, with work dating back to the 1940s. More recently, ballistic trajectory analysis has been used by air accident investigators in an attempt to understand events leading up to an accident, with notable examples including the investigation into the Lockerbie bombing.

Building on the history of a previous model, this thesis offers an enhanced model for the calculation of ballistic trajectories incorporating altitude dependence for wind, density and gravity. Attempts to solve the model analytically were unsuccessful, and therefore, the model was solved numerically using an implicit scheme, to deal with the inherent stiffness of the equations, and an extrapolation technique. Cubic splines were used to accurately represent the wind profile in an analytical way.

The numerical solution was verified against a simplified analytical case and results are presented for two simulated breakup cases. Four key parameters were then varied to provide information about the sensitivity of final wreckage location to variation in system parameters.

The results indicate that for simulated large aircraft breakups, low ballistic coefficient items are most heavily affected by breakup altitude, wind magnitude and wind angle whereas large ballistic coefficient items are most heavily affected by breakup velocity, although to a much lesser extent (around 15% of the distance of low ballistic coefficient).

For small aircraft breakups, wind angle and breakup altitude have the largest effect on low ballistic coefficient items, with velocity and altitude affecting high ballistic coefficient items to a larger extent (around 50% of low ballistic coefficient items).

The results will allow investigators to understand better the factors which affect items of differing ballistic coefficient in different situations. The project as a whole provides a new solution engine for the trajectory problem which can form the basis of a new software package for investigators.

Acknowledgements

I would like to thank my supervisor for his continued help, support and faith in me.

I would also like to recognise the significant help and support given by my Mum and Dad and also by Phil.

Finally, and most importantly, my love and thanks to my long-suffering wife and kids.....for everything!

"Je n'ai fait celle-ci plus longue que parce que
je n'ai pas eu le loisir de la faire plus courte".

Blaise Pascal

Contents

List of Figures

Notation

$a,b,c,d_{x,y,z}$	cubic spline coefficients in the x,y and z direction
C_B	ballistic coefficient in kg/m^2
C_D	drag coefficient
$\mathbf{F}_g$	three-dimensional gravitational force vector
$\mathbf{F}_a$	three-dimensional aerodynamic force vector
g_0	standard acceleration due to gravity, taken as $9.80665\,m/s^2$
M_0	sea level mean molar mass taken as $28.964420\,kg/kmol$
m	mass in kg
p	air pressure in Pascals
R	specific gas constant in $J/(K \bullet kg)$ taken to be constant and defined as $R = R^*/M_0$
R^*	universal gas constant taken as $8{,}314.32\,J/(K \bullet kmol)$
$\mathbf{r}$	three-dimensional position vector
r_e	nominal radius of the earth, taken as 6,356,766 m
S	frontal area in m^2
T	air temperature in Kelvin
$\mathbf{v}_{TAS}$	true airspeed velocity vector
v	velocity
v_{GND} or v^G	velocity with respect to the ground
v_{WIND} or v^W	wind speed
$W_{x,y,z}$	x, y or z component of wind speed
$*_b$	the value concerned ($*$) calculated at the bottom of the layer in question
β	temperature gradient or lapse rate K/km
ρ	air density in kg/m^3

Chapter 1 - Introduction

One of the problems faced by air accident investigators is that of aircraft suffering in-flight breakup. Such breakups can be caused by a number of mechanisms including mid-air collision, disintegration or detonation of explosives. High profile examples of this type of event include the sabotage of Pan Am 103 over Lockerbie [1] and the explosion of flight TWA 800 over the Atlantic Ocean [2]. In such events, trajectory analysis is often employed in an attempt to understand certain aspects of the accident.

In general, trajectory analysis involves attempting to understand the behaviour of unpowered aircraft parts under the effects of gravity and drag after they have departed the aircraft. This may be to predict where parts have landed, infer breakup sequence from a wreckage field or to understand the position of the aircraft prior to breakup.

For example, often in cases of in-flight breakup, questions exist about the position of the aircraft prior to breakup including altitude, speed, heading etc. Data recorders can sometimes provide some or all of these details but, as was the case with the accident at Lockerbie, the recording will often be ended by some event, such as an explosion or loss of power, which stops the recording. This leaves unanswered questions regarding the behaviour of the aircraft after the event and its debris after breakup.

An alternative case exists when aircraft are lost, as was the case with Adam Air Flight 574 [3], which disappeared from radar over Indonesia and remained unlocated for 9 days until small parts of wreckage were found. In such cases, it could be useful to know the possible search region assuming a catastrophic event occurred between radar returns, particularly if there is a long delay between returns.

A third example exists when searching for a specific piece of wreckage which will aid the investigation, such as in the uncontained fan disk failure of flight

UA 232 which crash-landed at Sioux City [4]; despite being a crucial piece of evidence, the fan was not discovered until 3 months after the accident.

Trajectory analysis is suitable for application to all of these and many other accidents. As such, many trajectory analysis tools already exist and have been used for exactly this purpose. One such tool was developed within Cranfield [5] and was used in support of the Lockerbie accident. The final report into this accident [1] says

> *"A detailed trajectory analysis was carried out by Cranfield Institute of Technology in an effort to provide a sequence for the aircraft disintegration. This analysis comprised several separate processes, including individual trajectory calculations for a limited number of key items of wreckage and mathematical modelling of trajectory paths adopted by a series of hypothetical items of wreckage encompassing the drag/weight spectrum of the actual wreckage."*

In addition, Wood and Sweginnis [6] note that,

> *"During the Pan American B-747 Lockerbie investigation , the British turned this technique [trajectory analysis] into a science. They had wreckage scattered over a large portion of Scotland. They had very good information from the FDR and ATC Radar and they also had good meteorological information on the upper air winds. They wrote a computer program that would calculate trajectories for the wreckage they had and predict where other wreckage would be found."*

The program developed by Cranfield, entitled TRAJAN (from TRAJectory ANalysis), went through a number of iterations and forms, finally ending up as a Fortran 95 program producing text output and plots intended to be laid over Ordnance Survey maps.

Whilst this was appropriate at the time, the step change that has occurred in Geographic Information Systems (GIS) in the last decade, as embodied by the ubiquitous Google Earth, means that immediate access to accurate mapping data of much of the earth is now possible. Coupled with the prevalence of GPS data, including its incorporation into digital photographs, this technology opens the possibility of a revolutionary trajectory analysis tool.

Fundamental to this goal is the ‘engine’ (the mathematical approach and the computational algorithm used for calculation) employed to predict the flight of each component. All aspects of any trajectory analysis tool will draw on this engine, making its developmental crucial. Ideally, the engine will provide an extremely quick and efficient calculation of the desired result, thereby allowing an interface to be developed that offers almost real-time response to the user.

Additional areas of interest include:

- use of the tool to establish ‘safe’ areas at airshows, or to restrict flight areas; and
- protection of sensitive targets from airborne security threats.

Therefore, the primary aim of this research is to develop an efficient, detailed engine for calculating the trajectory of aircraft debris following in-flight breakup.

The thesis will deal with the literature surrounding mid-air breakups, before moving on to the mathematics and numerical approaches around solving the problem. Verification results and application of the model to example cases will then be presented and discussed. Additional issues surrounding further implementation of the solution will be discussed along with possible approaches. Finally, conclusions will be presented.

Chapter 2 - Literature Review

Early studies of trajectory analysis appear to have been centred at the Royal Aircraft Establishment (RAE) and the Aeronautical Research Council in the UK and in Canada at the National Aeronautical Establishment.

One of the earliest references linking trajectory analysis to aircraft accident investigation dates from 1946 [7] and was informed by bomb trajectory tables. This work supplied graphs based on terminal velocity which allowed trajectories to be constructed for differing objects, altitudes and initial velocities. Despite being written more than 60 years ago, this report encompasses many of the key features of trajectory analysis such as the dependence on air density, velocity and wind. In 1949, Owen and Grinsted [8] used this work to analyze a number of airframe failures from the Second World War, with useful results regarding breakup order.

In 1956, Templin and Callan [9] noted that solving the trajectory equations for multiple items in accident investigation could be laborious. Therefore, they developed a graphical technique for easily calculating trajectories based upon the initial angle and the ratio of initial velocity to terminal velocity. Also in 1956, Braun [10] published a similar graphical technique which also included recommendations regarding confidence limits and was supported by "dropping tests". The work of Templin and Callan was then extended in 1960 by Currie [11] who employed a computer (which was 5 foot tall and weighed 300kg!) to produce a set of tables to calculate trajectory curves in Cartesian coordinates.

In 1961, Gwilt [12] produced a report which is still used by some workers in the field today. Part of this report was a detailed description of how to use Currie's tables to calculate a range of factors including altitude of breakup. However, Gwilt also suggested techniques for estimating terminal velocities, supported by measurements, including the estimation of terminal velocities for ‘tumbling’ items. As will be discussed in later sections, this tumbling behaviour presents a major difficulty in trajectory analysis.

In 1966, Waterfall [13] presented a technique for analyzing the trajectory of a ballistic missile including effects of earth rotation and variable gravity. However, the base model did not include the effect of wind although this was included as a separate model. Ultimately, it was a curve-fitting approach to the trajectory problem.

Boksenbom [14] offered a graphical technique for calculating trajectories, but this was primarily aimed at space applications, and hence has little relevance here.

Bergen-Henegouwen [17, 18] developed a computational technique for calculating the most probable initial breakup conditions given a wreckage layout. This was achieved by calculating an error function between the calculated position, based on a trajectory model, and the measured position and then minimizing that function. This approach used a fifth-order polynomial fit for the wind data.

Matteson [19] applied iterative calculation using a digital computer to analyse the dependence of wreckage patterns on a variety of parameters such as wind speed, wind direction and breakup altitude.

In 1976 Kepert [20] published work detailing the approach taken by the Australian Aeronautical Research Laboratories when analyzing wreckage trajectories. This work was limited to breakups below 10,000 ft. The model employed made no allowance for wind effects in the first instance, but it was also highlighted that other uncertainties (such as drag coefficient) were significant as well. Kepert’s work drew on data gathered during missile trials which involved destroying aircraft with a known position and initial velocity to give guidelines for calculating initial conditions before breakup.

In 1978, work began at Cranfield in participation with the Accidents Investigation Branch to develop a computerised method for calculating trajectories. This began with work by Khan [21] and was followed in 1980 by Hull [22] and in 1983 by Steele [23]. This work was developed by Taylor into the TRAJAN package which was used for a number of applications including Lockerbie and the DC-9 over Ustica [1, 5, 24, 25].

More recently, trajectory analyses have also been conducted for a number of other significant accidents including TWA 800 [26], Air India AI182 [27] and China Airlines CI611 [28]. All of these analyses used theory similar to that already in existence.

An alternative, but equally valid use of trajectory analysis tools is for airshow safety. Calculating where the wreckage would land if an accident were to happen during a display is a key part of the safety planning process for a display. There are at least two examples [15, 16] of the application of standard techniques for this process. Similarly, trajectory analysis could provide an estimation tool for security purposes, allowing three-dimensional exclusion zones to be defined around sensitive installations.

Since accident investigation is such an applied science, it is not always possible to gauge the state of the art from journal publications. Therefore, 9 major NIAs were contacted to establish their trajectory analysis capabilities. No additional capability was discovered, with most capabilities accurately represented in the literature.

In addition, it is conceivable that military solutions to this problem exist. However, assuming they exist, if such solutions are not available for testing and validation and are only available for civilian purposes in extreme cases, then their benefit is limited. A publication exists from the Canadian Forces which offers practical advice for the calculation of trajectories, including typical drag coefficients and graphical solution techniques. No significant theory is presented.

2.1 Research aims

The governing research aim is to improve the TRAJAN model thereby offering investigators an improved tool for accident investigation.

Further to this, given an improved model, it would be beneficial to identify which parameters investigators should focus on in order to improve overall accuracy, and conversely, which pieces of ground information are most useful to inform investigators about prior events.

2.2 Methodology

- The initial part of the project involved searching archived hard drives in order to rediscover, update (from a software perspective) and recompile the TRAJAN model. This was necessary in order to understand the processes and approaches adopted by the software and inform the improvement process.

- Following this, a decision was taken as to whether the software TRAJAN model should be used as a basis, or a new model developed.

- The theoretical basis for the improved model was established and implemented as a software product.

- The improved model was then verified against a simple analytical model, incorporating fewer adjustments than the improved model. It was also checked for consistency given simplified initial states.

- A practical sensitivity analysis was conducted to examine the dependence of final location on four major parameters.

- Finally, the lessons learned and thoughts for the future were documented in order to provide a starting specification for a commercial implementation of this approach.

Chapter 3 - Model Development

3.1 Introduction

Accepting that the calculation of trajectories from midair breakups is still relevant and of interest for accident investigation, it is necessary to develop a suitable approach to modelling the behaviour of a particle.

The problem is essentially one of calculating ballistic trajectories. Whilst the investigator is clearly interested in more than the path taken by a part falling to ground, if the trajectory is understood then other variables such as initial velocity, final velocity, time to fall to earth, impact velocity, aerodynamic force etc. are also directly available.

NATO and the US DOD define a ballistic trajectory as the "*trajectory traced after the propulsive force is terminated and the body is acted upon only by gravity and aerodynamic drag*." [35]. Clearly this definition is appropriate when considering wreckage created through midair breakup. Exceptions to this definition would include an aircraft which is damaged but still producing propulsive force and components which are capable of generating lift. The latter point is an important one - in the subsequent analysis, the components will be considered to be acted on by drag alone; no lift force will be included.

In addition, the 'tumbling' of parts whilst falling will also be discounted and instead replaced by a single drag coefficient. Both of these assumptions are deviations from reality. Whilst they might accurately describe the behaviour of a high mass, compact body (one with a high value of ballistic coefficient - see Section 3.5), a lighter part with a large area capable of producing lift (such as a section of fuselage skin with stringers) is clearly very likely to produce lift and tumble as it falls, such as a sheet of cardboard might do if dropped.

It may be possible to calculate the exact path of a single component computationally, but it would almost certainly require: an accurate digital model of the part; an enormous amount of computational fluid dynamics; and a precise model of the wind. Even given those data, it his highly likely that the calculations would be extremely sensitive to initial conditions and any perturbations such as collisions with other parts, disturbed airflow etc. would cast considerable doubt on any results.

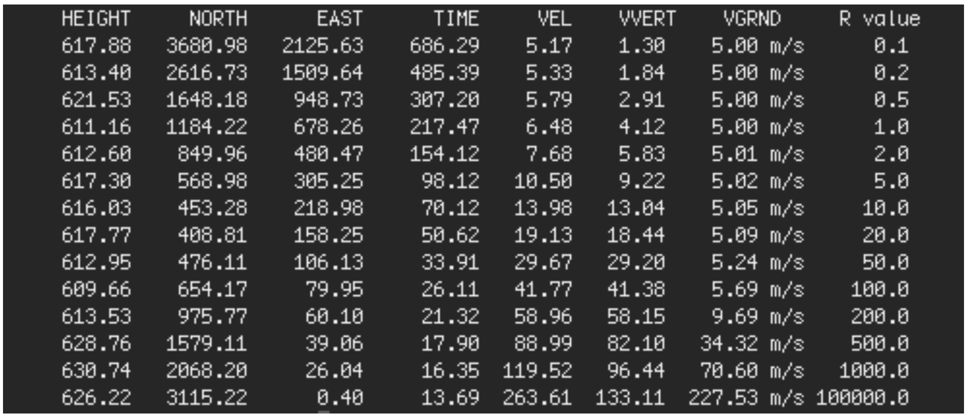
For this reason, it is appropriate to consider a large number of parts when modelling trajectories in the hope that some sort of 'averaging' will occur between components and behaviours. It may also be useful to consider different categories of components, such as differentiating between those that are little affected by the wind and those that are strongly affected - for example. If predictions of the former are accurate whilst predictions of the latter are not, then it may be prudent to reexamine the wind data being used.

Having defined the fundamentals of the modelling problem, this chapter will briefly discuss the TRAJAN model and then propose and detail an alternative approach to solving the problem.

3.2 TRAJAN

The TRAJAN model was born out of work done and reported as MSc theses by Khan [21], Hull [22] and most significantly, Steele [23]. This work was later reviewed and developed by Anker and Taylor and formed the basis of the model used for analysing the Lockerbie accident in 1988.

The original version of TRAJAN was written in FORTRAN running under the VAX/VMS operating system on VAX hardware. It was originally thought that the only surviving instance of the TRAJAN source code was paper printouts of this VAX version. However, for this research project, source code was 'unearthed' written in the more modern Fortran 95 language. However, like many legacy programs, there were numerous sets of code in various states of development with no discernible version control or commenting. Therefore, it was necessary to step through, interpret and comment the code. It was then recompiled with all subroutines and calls renamed to ensure the full dependencies were established. Figure 3.1 shows a typical output screen from the recompiled version of the TRAJAN program.



```
   HEIGHT     NORTH      EAST       TIME       VEL     VVERT     VGRND       R value
   617.88   3680.98   2125.63     686.29      5.17      1.30    5.00 m/s       0.1
   613.40   2616.73   1509.64     485.39      5.33      1.84    5.00 m/s       0.2
   621.53   1648.18    948.73     307.20      5.79      2.91    5.00 m/s       0.5
   611.16   1184.22    678.26     217.47      6.48      4.12    5.00 m/s       1.0
   612.60    849.96    480.47     154.12      7.68      5.83    5.01 m/s       2.0
   617.30    568.98    305.25      98.12     10.50      9.22    5.02 m/s       5.0
   616.03    453.28    218.98      70.12     13.98     13.04    5.05 m/s      10.0
   617.77    408.81    158.25      50.62     19.13     18.44    5.09 m/s      20.0
   612.95    476.11    106.13      33.91     29.67     29.20    5.24 m/s      50.0
   609.66    654.17     79.95      26.11     41.77     41.38    5.69 m/s     100.0
   613.53    975.77     60.10      21.32     58.96     58.15    9.69 m/s     200.0
   628.76   1579.11     39.06      17.90     88.99     82.10   34.32 m/s     500.0
   630.74   2068.20     26.04      16.35    119.52     96.44   70.60 m/s    1000.0
   626.22   3115.22      0.40      13.69    263.61    133.11  227.53 m/s  100000.0
```

Figure 3.1 - Example output of Fortran 95 version of TRAJAN

In Figure 3.1, each row represents a different ‘R’ value (the ballistic coefficient - see Section 3.5) with large R values representing heavy / low drag items. For each R value, results are given for: final height (the timestep closest to the defined ground level), North and East location, time taken, final velocity, final vertical velocity and ground velocity.

In addition to this screen output, the TRAJAN code also produces a number of text output files. An example of output taken from one of these files is shown in Figure 3.3. This Figure details the trajectory for the final R value given in the last column in Figure 3.1. As the time column suggests, TRAJAN is a time-stepping code using a constant timestep and assuming linear behaviour across the timestep. This will be discussed in greater detail in Section 3.6.

Whilst the TRAJAN model is not especially sophisticated, it does benefit from some interesting subtleties. It includes, for example, variation of density with altitude, albeit through the use of lookup tables for altitude bands rather than a value calculation at each timestep point. It also uses the concept of uncertainties to allow the user to incorporate what they know about the level of uncertainty in their data. For example, if it is clear that the wind data available is little more than a ‘best guess’ then considerable uncertainty may then be attached to it, specified in percentage terms. The model will then run, taking data from the extremes of the uncertainty range in order to give a 'zone' of possible results.

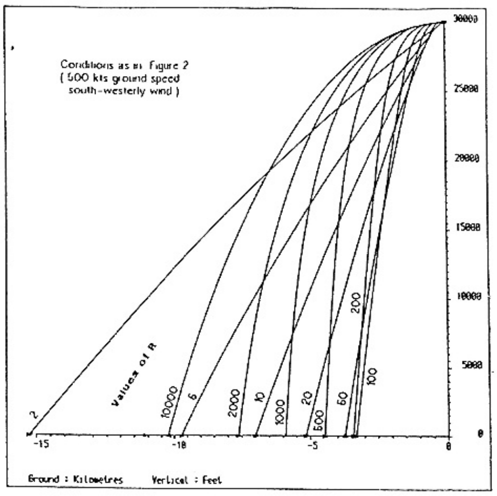
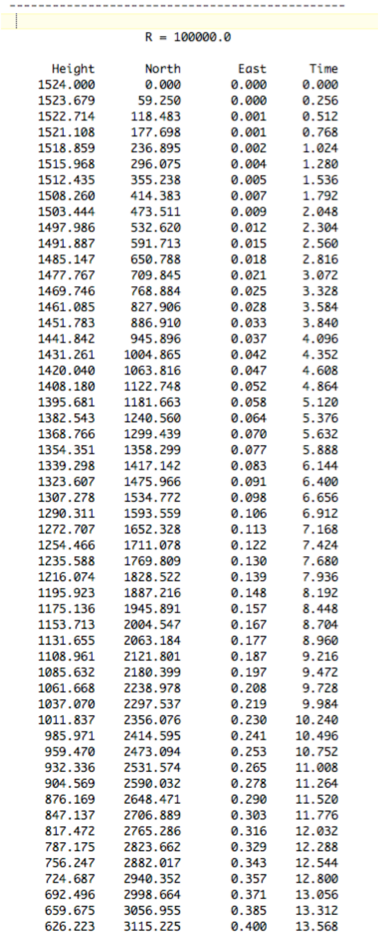


Figure 3.2 - Trajectories of falling pieces

The program provides no graphical output which is most likely a result of the era in which it was produced; in 1978 graphical output was extremely hard to produce with the first Graphical User Interface (GUI) not appearing until 1981. However, the study of trajectories is one that is often most easily understood through visualisation. Therefore, the TRAJAN output was often used to produce graphs and curves such as those show in Figure 3.2.



```
----------------------------------------------

                  R = 100000.0

      Height        North        East       Time
    1524.000        0.000       0.000      0.000
    1523.679       59.250       0.000      0.256
    1522.714      118.483       0.001      0.512
    1521.108      177.698       0.001      0.768
    1518.859      236.895       0.002      1.024
    1515.968      296.075       0.004      1.280
    1512.435      355.238       0.005      1.536
    1508.260      414.383       0.007      1.792
    1503.444      473.511       0.009      2.048
    1497.986      532.620       0.012      2.304
    1491.887      591.713       0.015      2.560
    1485.147      650.788       0.018      2.816
    1477.767      709.845       0.021      3.072
    1469.746      768.884       0.025      3.328
    1461.085      827.906       0.028      3.584
    1451.783      886.910       0.033      3.840
    1441.842      945.896       0.037      4.096
    1431.261     1004.865       0.042      4.352
    1420.040     1063.816       0.047      4.608
    1408.180     1122.748       0.052      4.864
    1395.681     1181.663       0.058      5.120
    1382.543     1240.560       0.064      5.376
    1368.766     1299.439       0.070      5.632
    1354.351     1358.299       0.077      5.888
    1339.298     1417.142       0.083      6.144
    1323.607     1475.966       0.091      6.400
    1307.278     1534.772       0.098      6.656
    1290.311     1593.559       0.106      6.912
    1272.707     1652.328       0.113      7.168
    1254.466     1711.078       0.122      7.424
    1235.588     1769.809       0.130      7.680
    1216.074     1828.522       0.139      7.936
    1195.923     1887.216       0.148      8.192
    1175.136     1945.891       0.157      8.448
    1153.713     2004.547       0.167      8.704
    1131.655     2063.184       0.177      8.960
    1108.961     2121.801       0.187      9.216
    1085.632     2180.399       0.197      9.472
    1061.668     2238.978       0.208      9.728
    1037.070     2297.537       0.219      9.984
    1011.837     2356.076       0.230     10.240
     985.971     2414.595       0.241     10.496
     959.470     2473.094       0.253     10.752
     932.336     2531.574       0.265     11.008
     904.569     2590.032       0.278     11.264
     876.169     2648.471       0.290     11.520
     847.137     2706.889       0.303     11.776
     817.472     2765.286       0.316     12.032
     787.175     2823.662       0.329     12.288
     756.247     2882.017       0.343     12.544
     724.687     2940.352       0.357     12.800
     692.496     2998.664       0.371     13.056
     659.675     3056.955       0.385     13.312
     626.223     3115.225       0.400     13.568
```

Figure 3.3 - Example output taken from TRAJAN text file

In addition, ground maps were sometimes produced, to scale, on acetate for overlay on Ordnance Survey maps. Again, this is representative of the time of development. Computing technology and the availability of digital mapping mean that this aspect should be easy to improve on in a modern implementation.

Whilst the TRAJAN model provides useful information from a valid approach, there are a number of factors which require improvement. When this is coupled with a desire to produce an application using a more modern programming language, it is clear that a ‘fresh start’ is the most appropriate route, rather than trying to modify or update the existing code.

3.3 Model formulation

When ignoring wind and air resistance, the theory behind ballistic (i.e. unpowered under the influence of gravity) trajectories of idealised projectiles is simple and well-understood. This is complicated slightly when including the effects of air resistance, although in some cases this is still easily solved.

However, an opportunity exists to develop a model which incorporates many of the more subtle effects necessary for it to be widely applicable. These include: full three-dimensional effects of wind; effects of atmospheric density changes; and the variation of gravity with height. These effects are particularly important if high altitude accidents, such as the Columbia (STS-107) or Challenger (STS-51-L) space shuttle accidents are to be analysed. At present, the rotation of the earth is not considered.

Any trajectory model must be dependent upon some estimation of the drag coefficient of the part, which is often difficult to achieve. This is compounded by the fact that many objects will ‘tumble’ whilst falling, effectively offering a variable drag coefficient. Given the unavoidable inaccuracies inherent in drag coefficient estimation, it is arguable whether a more advanced calculation technique is necessary. However, there is no reason not to minimize as many errors as possible, as long as the other inaccuracies and limitations are understood.

Most existing computational solutions use time-stepping to calculate the particle trajectory, with some using a linear approximation between steps. This approach relies on the time step being small enough to accurately ‘follow’ the curved path. However, if a fixed time step is used this can result in either unnecessary computation or lack of fidelity. Alternatively, an adaptive (variable) time step can be adopted but this also carries a computational overhead. Since one of the desired outcomes is to allow investigators to see results of 'modifications' in real-time, a quick, accurate, computationally efficient solver engine is crucial. Therefore, the approach outlined below is aimed at finding an analytical solution to the governing equations. If successful, this will provide an exact solution in an explicit form which will greatly reduce the computational overhead.

3.4 Analytical approach - one dimension

Before incorporating the more complex aspects of the model, such as variable density and variable gravity, it was considered useful to derive the analytical expressions for the one-dimensional behaviour of a ballistic particle subject to gravity and a drag force proportional to velocity squared. This is shown in Appendix 1 for reference. It is worth noting that the expressions accurately reflect the asymptotic approach to terminal velocity that would be expected.

At its most simple, assuming one-dimensional (x) vertical motion, drag proportional to velocity squared (v^2) and assuming constant mass (m) and gravity (g) and ignoring wind effects, gives the governing differential equation of

$$m\frac{dv}{dt} = mg - \frac{1}{2}\rho C_D S v^2 \qquad \text{(Eqn 3.1) where} \qquad v(t) = \frac{dx}{dt},$$

S is frontal area, C_D is drag coefficient and ρ is density, which can be solved for $v(t)$ and $x(t)$ as given in Appendix 1. This is Newton's Second Law, with the forces acting on the body on the right hand side of the equation and the resulting acceleration on the left hand side.

In order to expand this simple model to incorporate the more complex aspects, it is necessary to quantify and model the effects and variations of each of the components. Once this has been done, they will be inserted into the one-dimensional equation and an analytical solution attempted.

Variation of acceleration due to gravity

The ICAO standard atmosphere [29] facilitates the calculation of atmospheric parameters based on '*a perfect gas free from moisture and dust and based on conventional initial values of temperature, pressure and density of the air for mean sea level.*' It also provides guidance for the calculation of acceleration due to gravity. Acceleration due to gravity is a function of both altitude and also latitude. The variation of acceleration due to gravity with latitude is given by [29] as:

$$g_\varphi = 9.80616(1 - 0.0026373\cos 2\varphi + 0.0000059\cos^2\varphi)$$

Taking a latitude of $\varphi = 45^\circ 32' 33''$ gives a value of $g_0 = 9.80665\text{m/s}^2$. In the subsequent analysis, this value will be used and change in latitude during descent is not considered.

The ICAO standard uses the concept of the geopotential altitude (H) with units of geopotential metres. H is defined as the ratio of the gravity potential or geopotential of a point (Φ) to the standard acceleration due to gravity (g_0).

A simple expression for the relationship between acceleration g and geometric altitude (h) can be obtained by '*neglecting centrifugal acceleration and using only Newton's gravitation law*', giving:

$$g = g_0\left(\frac{r_e}{r_e + h}\right)^2$$

where r_e is the nominal radius of the earth, taken as 6,356,766 m. At an altitude of 60,000 m (c. 197,000 ft) this simplification gives a difference of less than 0.001% from a more accurate expression. This allows a relationship between geopotential and geometric altitude to be written explicitly:

$$H = \frac{r_e h}{r_e + h} \qquad \text{and} \qquad h = \frac{r_e H}{r_e - H} \qquad \text{(Eqn 3.2 a and b)}$$

Therefore, the variation of gravity with respect to altitude in equation (3.1) can be written as:

$$g(x) = g_0 \left(\frac{r_e}{r_e + x} \right)^2$$

Variation of air density

An assumption of constant air density is one of the more inaccurate assumptions that can be made when calculating trajectories, particularly for high altitudes. This is because the pressure, density and temperature of the air all vary significantly with altitude and will hence have a significantly varying effect on particle behaviour.

The earth's atmosphere is divided into a number of layers with differing properties, the first few of which are (moving from the earth upwards): the troposphere, the stratosphere and the mesosphere. At the top of the troposphere is the tropopause and similarly at the top of the stratosphere is the stratopause. These two transitional zones are the areas which define the change in layer since the temperature gradient (the rate of change of temperature with height) changes sign within these zones, as shown in Table 3.1.

Figure 3.4 shows some of these zones in an image taken from the International Space Station. The orange coloured layer is the troposphere, with the tropopause being the sharp-edged boundary between the orange troposphere and the blue stratosphere.

Geopotential altitude (km) ***H***	**Temperature (K)** ***T***	**Temperature gradient (K/km)** β	**Atmospheric zone**
0	288.15		
		-6.50	Troposphere
11	216.65		
		0.00	Tropopause
20	216.65		
		+1.00	Stratosphere
32	228.65		
		+2.80	
47	270.65		
		0.00	Stratopause
51	270.65		
		-2.80	Mesosphere
71	214.65		
		-2.00	
80	196.65		

Table 3.1 - Properties of atmospheric layers (adapted from ICAO [29])

Figure 3.4 - Photograph of the earth's atmosphere from the International Space Station [from NASA]

Because of the variation in temperature gradients between zero and non-zero in the different layers, it is necessary to define two separate expressions for calculating the atmospheric properties. The ICAO standard atmosphere, gives an expression for the pressure variation with geopotential altitude as

$$p = p_b \left[1 + \frac{\beta}{T_b}\left(H - H_b\right) \right]^{-g_0/\beta R} \quad \text{for } \beta \neq 0$$

and

$$p = p_b \exp\left[-\frac{g_0}{RT}\left(H - H_b\right) \right] \text{ for } \beta = 0$$

where p is the atmospheric pressure, β is the temperature gradient, T is the temperature, and R is the specific gas constant. The subscript b indicates the given value evaluated at the lower limit of the layer of concern.

From these expressions for pressure, an expression for density can be calculated by using the perfect gas relation.

$$\rho = \frac{p}{RT}\,.$$

As with the expressions for gravity, the geopotential altitude can be converted using Equation 3.2. This gives governing expressions for the variation of air density with geometric altitude for the two temperature gradient regimes as

$$\rho = \frac{p_b}{RT_b}\left[1 + \frac{\beta}{T_b}\left(\frac{r_e x}{r_e + x} - H_b \right) \right]^{-(1 + g_0/\beta R)} \quad \text{for } \beta \neq 0$$

and

$$\rho = \frac{p_b}{RT}\exp\left[-\frac{g_0}{RT}\left(\frac{r_e x}{r_e + x} - H_b \right) \right] \text{ for } \beta = 0$$

It is worth noting that the ICAO standard atmosphere is valid up to ‘only’ 262,500 ft (80km). Whilst this is many times the normal cruising altitude of commercial aircraft it does not encompass the full range of altitudes that may be seen by spacecraft operating in the atmosphere. For example Virgin Galactic plan to take SpaceShipTwo to 360,000 ft (110 km) [32]. This is not a significant limitation and can be easily adapted, but it should be noted before making high altitude predictions.

Aerodynamic drag

As described previously, drag is being assumed in the form of $0.5\rho C_D S v^2$. This is a well established approach [e.g. 33], however it is important to note that the velocity described is the square of the component's airspeed not its groundspeed. This is because the drag is created by the relative airflow; a particle travelling at precisely the windspeed in theory experiences no drag. The difference which results from using airspeed rather than groundspeed may be small if the particle velocity greatly exceeds the windspeed, but it should not be ignored.

The assumption of constant drag is a significant aerodynamic assumption. As shown generically in Figure 3.5, drag is a function of Mach number and also of Reynolds number.

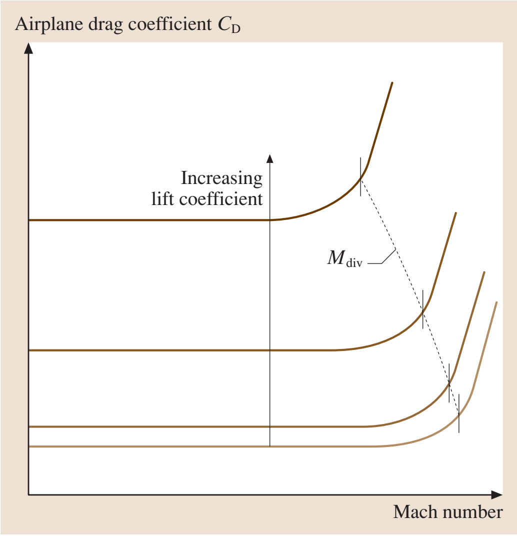


Figure 3.5 - Generic plot showing variation of drag coefficient with Mach number (from [34])

Whilst it may be possible to ignore the effects of compressibility at lower speeds, they will significantly alter the drag coefficient at higher speeds and greatly differing Reynolds numbers. However, in this situation an assumption of constant drag coefficient is warranted since it is contained within the ballistic coefficient (see Section 3.5). By calculating results for a range of ballistic coefficient, errors in drag coefficient can be compensated for, although the variation in drag coefficient with time cannot be accounted for in this way.

Wind profile

In order to provide a full one-dimensional analytical solution to the trajectory problem which incorporates wind data, it is necessary to provide some function which accurately describes the variation of wind speed (in this 1D case, acting vertically) with altitude. This subject will be returned to, but for now it is sufficient to assume that such a function exists, which shall for these purposes be denoted $W_x(x)$.

The wind appears in the drag calculation, and it is this effect that causes the particle to adopt the surrounding windspeed. Whilst aftercasts are often available, their accuracy can depend on many factors, including local meteorological resources and the global meteorological model. In some cases, predictions may be based on coarse grid sizes s large as 10km[1].

Governing equation

Combining all of these effects - variable gravity, variable density, and the expression to describe wind - allows Equation 3.1 to be modified from

[1] *see e.g. ruc.noaa.gov*

$$m\frac{dv}{dt}=mg-\frac{1}{2}\rho C_D S v^2$$

to

$$m\frac{dv_{GND}}{dt}=mg(x)-\frac{1}{2}\rho(x)C_D S v_{TAS}^2$$

where v_{TAS} is the true airspeed in the x direction and v_{GND} is the vertical speed relative to the ground. Incorporating the previously derived expressions, gives

$$m\frac{dv_{GND}}{dt}=mg_0\left(\frac{r_e}{r_e+x}\right)^2-\frac{1}{2}\frac{p_b C_D S}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e x}{r_e+x}-H_b\right)\right]^{-(1+g_0/\beta R)}v_{TAS}^2 \quad \text{for } \beta\neq 0$$

$$m\frac{dv_{GND}}{dt}=mg_0\left(\frac{r_e}{r_e+x}\right)^2-\frac{1}{2}\frac{p_b C_D S}{RT}\exp\left[-\frac{g_0}{RT}\left(\frac{r_e x}{r_e+x}-H_b\right)\right]v_{TAS}^2 \quad \text{for } \beta=0$$

and

$$v_G=v_{TAS}+W_x$$

which are the differential equations governing the one-dimensional motion of a particle falling through the atmosphere.

In an attempt to solve these expressions analytically, first rewrite the $\beta\neq 0$ case as

$$\frac{d}{dt}v_G+\frac{1}{2}\frac{p_b C_D S}{mRT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e x}{r_e+x}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(v_G-W_x\right)^2-g_0\left(\frac{r_e}{r_e+x}\right)^2=0$$

Expressing in terms of displacement gives,

$$\frac{d^2x}{dt^2}+\frac{1}{2}\frac{p_bC_DS}{mRT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_ex}{r_e+x}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(\frac{dx}{dt}-W_x\right)^2-g_0\left(\frac{r_e}{r_e+x}\right)^2-W_x=0$$

which is an example of an autonomous equation of the form

$$\frac{d^2x}{dt^2}+f(x)\left[\frac{dx}{dt}+g(x)\right]^2+h(x)=0$$

Substituting $v=\frac{dx}{dt}$ and since $\frac{d^2x}{dt^2}=\frac{d}{dt}v(x)=\frac{dv(x)}{dx}\frac{dx}{dt}=v\frac{dv}{dx}$

gives

$$v\frac{dv}{dx}+f(x)\left[v+g(x)\right]^2+h(x)=0$$

Expanding gives

$$vv'+f(x)v^2+2f(x)g(x)v+f(x)g^2(x)+h(x)=0$$

and rearranging

$$vv'=p(x)v^2+q(x)v+r(x)$$

which is an Abel differential equation of the second kind. This is an equation type for which only some specific solutions are available. Use the substitution [30],

$$v=E(x)w \text{ where } E(x)=\exp\left(\int p(x)dx\right)$$

leading to

$$ww'_x=F_1(x)w+F_0(x) \qquad \text{(Eqn 3.3)}$$

where

$$F_1(x) = \frac{q(x)}{E(x)} \qquad \text{and} \qquad F_0(x) = \frac{r(x)}{E^2(x)}$$

Taking

$$z = \int F_1(x)dx \qquad \text{and} \qquad \varphi(z) = \frac{F_0(x)}{F_1(x)} \text{ in (3.3)}$$

gives

$$ww'_z - w = \varphi(z). \quad \text{(Eqn 3.4)}$$

This is an Abel equation of the second kind written in canonical form, which until recently, would be regarded as almost insoluble; a general solution of this type of equation was not available, with only certain cases being open to solution. However, in 2006 Panayotounakos and Kravvaritis [31] proposed an explicit solution for this type of equation.

Considerable effort was expended in this project attempting to apply this general solution to Equation 3.4. If achievable, it would have offered an analytical solution for the one-dimensional trajectory (velocity and displacement) of a particle given variable gravity and density. However, unfortunately, the analytical calculations became intractable, preventing the general solution from being applied.

It was reasonable to expect that the analytical solution shown in Appendix 1 could be extended to the include variable gravity, variable density and wind. However, it appears that the approach adopted does not yield a usable answer. That is not to say that an alternative approach or a more skilled attack using the existing approach will not yield an analytical solution, and the goal of an analytical solution to the problem is still a valid one.

Clearly, since the simple case is soluble and the more involved case appears not to be (using this approach), it is likely that there is a point at which the increased complexity prevents the problem from being solved analytically. Whilst it might be possible to discover this point, this was considered to be a futile endeavour; the required complexity should define the problem to be solved rather than the available solution dictating the level of complexity.

However, whilst an analytical solution was the ideal, an alternative solution will still allow calculation and analysis. Therefore, the remainder of this chapter will describe attempts to produce a numerical solution of the governing equations.

3.5 Numerical approach - three dimensional governing equations

Moving to a three-dimensional formulation, the following conventions will be adopted:

The position of a piece of wreckage is given by $\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

x, y and z are defined relative to the final aircraft track, with x directly along the aircraft track, y orthogonal to the aircraft track (when viewed from above ,y increases to the left) and z being positive upwards.

Velocity relative to the ground, is given by $\mathbf{v}_{GND} = \frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x^G \\ v_y^G \\ v_z^G \end{bmatrix}$

Aerodynamic drag

As in the previous approach, aerodynamic drag is assumed to be of the form $0.5\rho C_D S v^2$. At this point, the concept of the ballistic coefficient will be introduced. The ballistic coefficient is a concept used in the analysis of ballistic trajectories, and is defined as:

$$C_B = \frac{m}{C_D S}$$

In earlier publications connected with TRAJAN, this coefficient was labelled as 'R' and given the slightly self-aggrandising name of the 'Cranfield Loading Coefficient'. There have also been differing definitions of the ballistic coefficient with some using mass and others using weight (i.e. differing by a factor of 'g'). In this work, the mass definition given above will be used.

The ballistic coefficient governs the aerodynamic drag behaviour of an object and offers a single parameter for classifying objects. An object with a high ballistic coefficient will have a high mass, low product of drag coefficient and frontal area, or both.

Therefore, it is possible to express the aerodynamic drag experienced by a component as

$$\mathbf{F_a} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left|\mathbf{v}_{TAS}\right|\mathbf{v}_{TAS} \qquad \beta \neq 0$$

and

$$\mathbf{F_a} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left|\mathbf{v}_{TAS}\right|\mathbf{v}_{TAS}\exp\left[-\frac{g_0}{RT}\left(\frac{r_e x}{r_e+x}-H_b\right)\right] \qquad \beta = 0$$

This expression is equivalent to a drag vector in the opposite direction to the three-dimensional airspeed vector, scaled in magnitude by the coefficients.

For simplicity, from this point only the $\beta \neq 0$ solution will be used which restricts the use to the troposphere (11,000m). However, the approach outlined is equally applicable to both scenarios and the full atmosphere is available through this solution.

Three-dimensional wind profile

As shown in Equation 3.1, the drag force acting on the particle is dependent on the relative windspeed, and hence the particle velocity and wind speed and direction. In order to incorporate this into the model, it will be assumed that either wind measurement data or 'aftercast' data are available. An aftercast is an estimation of the weather conditions at a certain time in a particular location after that time has passed; it is a forecast, made after the event, hence the name. In order to incorporate the wind data, which will be supplied at discrete altitudes, it is necessary to choose an interpolation method in order to allow data to be obtained at altitudes other than those supplied.

One option for representing the wind data is to assume constant windspeed in each band (which is equivalent to the 'zero order hold' technique of signal processing). This brings mathematical simplicity, but introduces problems at the transitions such as speed discontinuities which bring differentiation problems. This approach is also unlikely to accurately represent the true physical situation, unless the data points are closely spaced.

An alternative approach is to adopt linear interpolation between data points. This removes the discontinuity problem, although there may still be significant gradient transitions at data points. It also provides, by definition, a precise fit to the supplied data points. However, linear interpolation offers no attempt to smoothly transition through data points - intuitively wind speeds are more likely to vary smoothly with altitude than with step changes in gradient. As with the constant assumption, linear interpolation is of more benefit where the data points are closely spaced.

Ideally it would be possible to accurately fit a curve to all of the data points which can be simply described mathematically, such as a polynomial curve. However, whether this is possible depends on the data to which the curve is to be fitted.

Figure 3.6 shows measured data for an arbitrary location and time, taken from the NOAA website [33]. The complete data set is given in Appendix 2. The data presented is wind speed (ignoring direction at present) at a range of altitudes. This data is presented simply as a random sample of a wind profile to see what might be expected in terms of data, gradients etc. It is not intended to be representative of anything other than possible values.

Examining Figure 3.6, it is apparent that neither the third-order nor the sixth-order polynomial curve accurately represent the measured data.

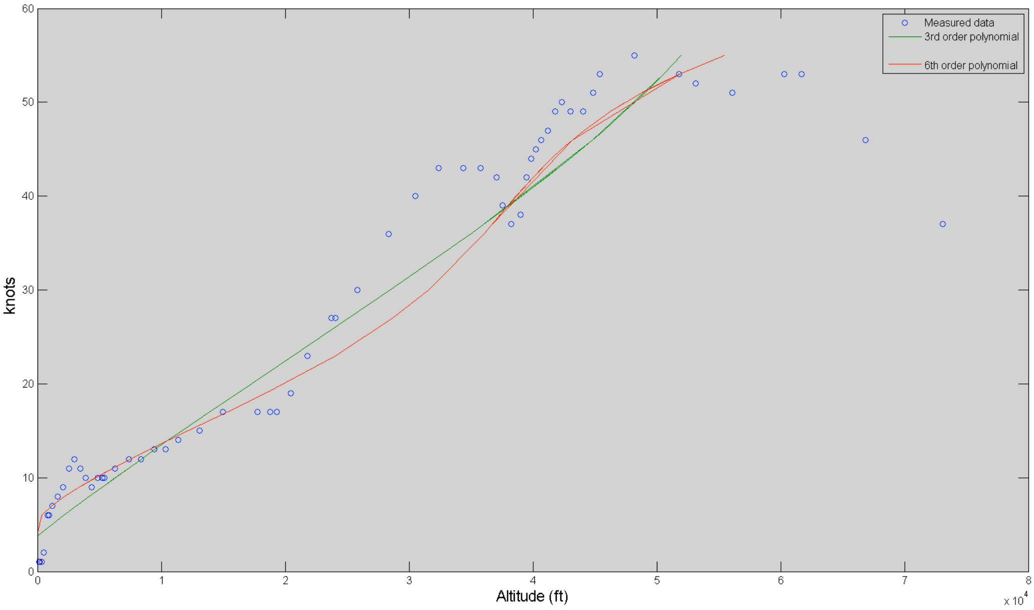


Figure 3.6 - Third order and sixth order polynomial fit to measured wind data

Because of the poor fit provided by the simple polynomials, an alternative approach is required. In this situation, a spline curve will be used.

Spline curves use the supplied data points as control points for fitting a polynomial of some degree. The difference between fitting a polynomial, such as in Figure 3.6, and using a spline is that in the latter, the curve is constructed from many polynomial curves which are pieced together (so-called piecewise polynomial curves). This means that each polynomial curve is only attempting to fit to a small number of data points, rather than the entire data set, thereby allowing a much more precise fit.

It is common to use a cubic polynomial as the basis for the spline since a cubic curve is the lowest degree of polynomial that can support an inflection. The typical form for the one-dimensional case is given as:

$$W(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

Cubic curves are also very well behaved numerically; their roots can always be found algebraically, and they are continuous to second differential. Therefore, in order to fit to the data, a cubic spline will be adopted which takes the measured data points as the reference points. In addition, the cubic spline can always be reverted to the linear interpolation or constant band assumptions described earlier, by setting constants in the expansion to zero.

Clearly, wind data is three-dimensional, although it is often assumed to be two-dimensional with any vertical component being neglected. This two-dimensional data is normally supplied as a direction (conventionally described as the 'from' direction rather than the 'to' direction) and a strength or magnitude. However, it is a matter of simple trigonometry to convert these so-called r, θ values into x, y components if required.

It is moot whether it is more appropriate to fit a spline through the x and y wind components or rather perform a fit through the wind speed and direction. The decision rests on which physical parameter is more likely to vary smoothly. Intuitively, because of the mechanisms involved in wind generation and modification such as centripetal acceleration and Coriolis force, it is tempting to imagine that the speed and direction (r, θ) description is more appropriate. Figures 3.7 and 3.8 show the cubic spline fit for both the x and y components and the r and θ component for the random wind sample.

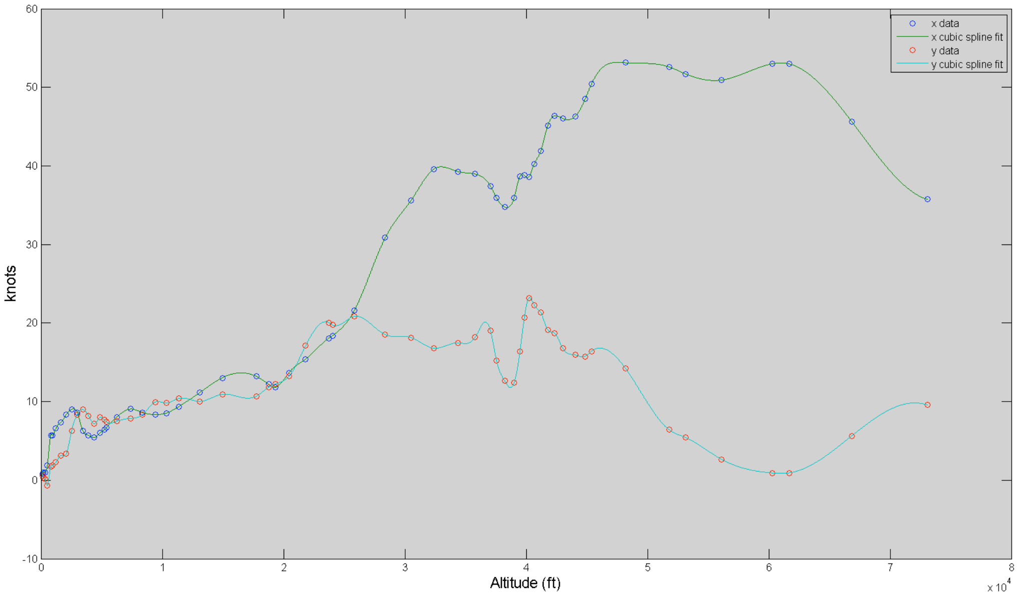


Figure 3.7 - Cubic spline fit for x and y data

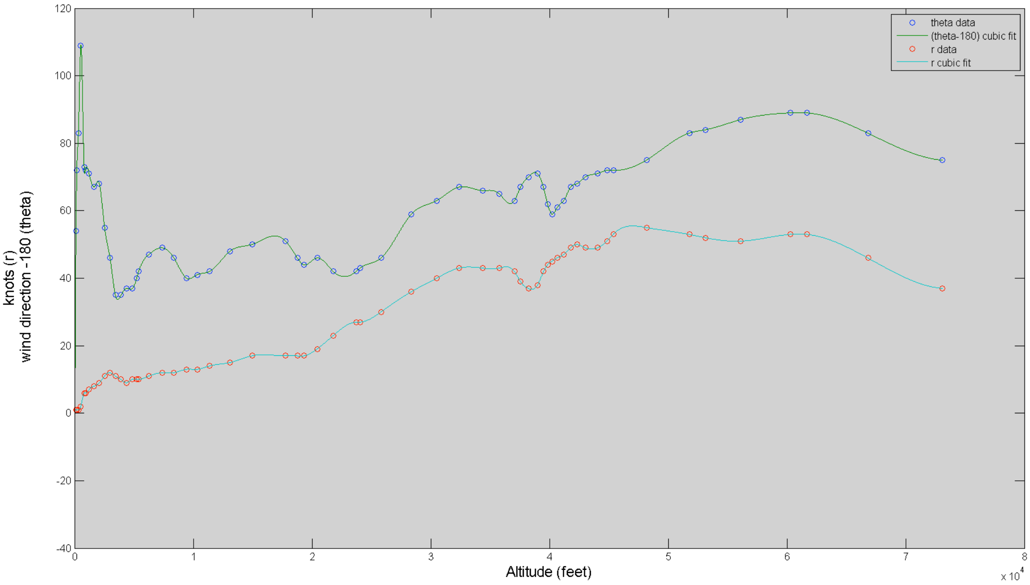


Figure 3.8 - Cubic spline fit for r and θ data

Both graphs show some difficulty in the cubic spline accurately representing smooth transitions from point to point without introducing data outside the shortest path. In reality the most suitable approach will be decided by the specific data. Therefore, it is sufficient to ensure that the curve fit does not deviate from the measured data by an amount more than is 'acceptable'. In this case, the curve fit will be conducted on the x and y components of the wind, since that provides the greatest mathematical convenience.

Because it is the windspeed vector that is of interest at each altitude, it is possible to fit a cubic spline through each orthogonal coordinate. If the assumption is made that the wind does not vary with x and y location, then the following expression, which is a function only of z can be defined.

$$\mathbf{v}_{WIND}(z) = \begin{bmatrix} v_x^W \\ v_y^W \\ v_z^W \end{bmatrix} = \begin{bmatrix} a_x + b_x z + c_x z^2 + d_x z^3 \\ a_y + b_y z + c_y z^2 + d_y z^3 \\ a_y + b_y z + c_y z^2 + d_y z^3 \end{bmatrix} = \mathbf{a} + \mathbf{b}z + \mathbf{c}z^2 + \mathbf{d}z^3$$

In situations where the wreckage travels significant distances, or where the wind profile changes rapidly, the assumption of a constant wind profile will cease to be valid. Whilst it is not difficult to incorporate values from a new profile for components which move outside the zone of validity for the initial profile, some attention should be given to the smooth variation from one zone to the next; this is merely an extension of the earlier argument about constant assumption, linear interpolation etc. but oriented in a horizontal plane. Along with the mathematical complexity this three-dimensional interpolation would add, there would also be an increased need for wind data. Arguably at least one adjacent profile should be established for all directions. For these reasons, a single profile is assumed in this case.

In order to incorporate the wind model into the three-dimensional governing equations, it is first necessary to define its relationship to airspeed. The groundspeed, airspeed and wind vector are related as

$$\mathbf{v}_{TAS} = \mathbf{v}_{GND} - \mathbf{v}_{WIND} \qquad \text{(Eqn 3.5)}$$

This simply expresses the fact that total speed relative to the ground is the sum of the true three-dimensional airspeed and the three-dimensional windspeed.

Finally, the gravitational force acting on each component, which acts only downwards, can be expressed as

$$\mathbf{F_g} = \begin{bmatrix} 0 \\ 0 \\ -mg_0\left(\dfrac{r_e}{r_e+z}\right)^2 \end{bmatrix}.$$

The governing equation is given by,

$$m\frac{d\mathbf{v}_{GND}}{dt} = \mathbf{F}_g + \mathbf{F}_a$$

and therefore, inserting all of the three-dimensional expressions gives,

$$\frac{d\mathbf{v}_{GND}}{dt} = \left(\frac{r_e}{r_e+z}\right)^2 \begin{bmatrix} 0 \\ 0 \\ -g_0 \end{bmatrix} - \frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z} - H_b\right)\right]^{-(1+g_0/\beta R)} |\mathbf{v}_{TAS}|\mathbf{v}_{TAS} \quad \text{(Eqn 3.6).}$$

3.6 Implementation of numerical solution

Equation 3.6 represents the governing, equation which when coupled with Equation 3.5 gives the governing equations for the trajectory problem. As discussed, in order to solve these equations, it is necessary to implement a numerical scheme.

Incorporating Equation 3.5 into the Equation 3.6 gives,

$$\frac{d\mathbf{v}_{GND}}{dt} = \left(\frac{r_e}{r_e+z}\right)^2 \begin{bmatrix} 0 \\ 0 \\ -g_0 \end{bmatrix} - \frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)} \left(\mathbf{v}_G-\mathbf{v}_{WIND}\right)\left|\mathbf{v}_G-\mathbf{v}_{WIND}\right|$$

Expanding this vector equation into three scalar first-order ordinary differential equations (ODEs) gives,

$$\frac{dv_x^G}{dt} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)} \left(v_x^G-v_x^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$\frac{dv_y^G}{dt} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)} \left(v_y^G-v_y^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$\frac{dv_z^G}{dt} = -g_0\left(\frac{r_e}{r_e+z}\right)^2 - \frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)} \left(v_z^G-v_z^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

In addition to these equations, the spatial coordinates must be related to velocities through

$$\frac{dx}{dt} = v_x^G \text{ , } \frac{dy}{dt} = v_y^G \text{ , and } \frac{dz}{dt} = v_z^G \text{ . } \quad \text{(Eqn 3.7)}$$

These six equations form the coupled state equations which are to be solved numerically.

There are myriad integration schemes [see e.g. 36], with popular examples including , Euler, Runge-Kutta, Richardson extrapolation (an implementation of which is Bulirsch-Stoer) and predictor-corrector or multistep methods.

The linear, time-stepping approach adopted by TRAJAN is an example of the Euler method. It is an explicit (or forward), first-order scheme meaning that the solution at a point depends only on the points prior to that and that the truncation error is of the order of the timestep. It is summarised with the expression:

$$y_{n+1} = y_n + hy'_n$$

which suggests that the value of a function at the next point in time is equal to the current value plus the gradient at that point multiplied by the timestep; this is a simple linear assumption. It is possible to increase the accuracy by decreasing the timestep, but with a corresponding increase in computational effort. In addition, the Euler method can be unstable particularly with stiff equations (see below). Press et al. suggest that Euler's method is "*not* recommended for any practical use" [36].

Stiffness

The possibility of a system becoming 'stiff' arises as soon as more than one first-order ODE is involved. A stiff system of ODEs is one in which the ratio of the greatest eigenvalue to the smallest is much greater than one [37]. Eigenvalues are representative of the solutions to the ODEs, with a large eigenvalue representing a contribution to the solution which dies away quickly. It is this property that presents the difficulty of stiffness.

Stiff systems give two connected problems - stability and accuracy. In order for some methods to remain stable, the corresponding steplength is required to be extremely small. If an inherently stable system is used, stability is no longer an issue, but for reasonable steplength, the component corresponding to largest eigenvalue will be approximated very inaccurately [37]. The rapidly decaying part of the solution requires a very small time step, and leaves behind integration instability even after it has diminished to zero.

Furthermore, inherently stable systems present the most serious stability problems for widely separated eigenvalues i.e. stiff systems [37].

Press et al. [36] suggest that the "simplest cure" to the "generic disease of stiff equations" is to employ implicit differencing. In implicit schemes, the value at a point depends not only on prior points but also on the current point. Implicit schemes can provide stability to stiff system whereas an explicit scheme does not.

In order to detect whether a system is stiff or not, requires the eigenvalues to be calculated. Lapidus [37] notes that for a nonlinear system of ODEs, the eigenvalues of the ODEs are also those of the Jacobian matrix. Therefore, by constructing the Jacobian, it will be possible to derive the eigenvalues and hence deduce the stiffness of the system. This would normally be done using the six coupled system equations given above. However, since the solution is not directly dependent upon x and y the first two equations in 3.7 can be ignored in the Jacobian, since the derivatives produced will be equal to zero.

The Jacobian of the coupled, 4-state system is defined as

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial^2 v_x^G}{\partial t \, \partial v_x^G} & \frac{\partial^2 v_x^G}{\partial t \, \partial v_y^G} & \frac{\partial^2 v_x^G}{\partial t \, \partial v_z^G} & \frac{\partial^2 v_x^G}{\partial t \, \partial z} \\ \frac{\partial^2 v_y^G}{\partial t \, \partial v_x^G} & \frac{\partial^2 v_y^G}{\partial t \, \partial v_y^G} & \frac{\partial^2 v_y^G}{\partial t \, \partial v_z^G} & \frac{\partial^2 v_y^G}{\partial t \, \partial z} \\ \frac{\partial^2 v_z^G}{\partial t \, \partial v_x^G} & \frac{\partial^2 v_z^G}{\partial t \, \partial v_y^G} & \frac{\partial^2 v_z^G}{\partial t \, \partial v_z^G} & \frac{\partial^2 v_z^G}{\partial t \, \partial z} \\ \frac{\partial^2 z}{\partial t \, \partial v_x^G} & \frac{\partial^2 z}{\partial t \, \partial v_y^G} & \frac{\partial^2 z}{\partial t \, \partial v_z^G} & \frac{\partial^2 z}{\partial t \, \partial z} \end{bmatrix}$$

and each of these derivatives must be calculated by hand which is shown in Appendix 3. The eigenvalues are then computed by calculating $|\mathbf{J} - \lambda \mathbf{I}| = 0$. This can be done algebraically, but since a repeated numerical solution is required corresponding to multiple sets of numerical parameters, this was achieved using the `eig` function within MATLAB.

The results produced 4 negative eigenvalues, indicating a stable system, but with a stiffness ratio, SR (largest eigenvalue / smallest eigenvalue) of 5 x 10^7 for a typical value set. Hall and Watt [40] suggest that a system is stiff if

SR>>0, that if SR is of the order of 10 it may be considered to be marginally stiff, and that orders of 10^6 are not uncommon. Clearly, the trajectory problem derived here needs to be recognised as a stiff system and treated appropriately.

3.7 Schema selection

The problem in hand is an initial-value problem; the values of the parameters are known at the start, and it is desired to find them at some given time point later. In addition, because a real-time implementation is of no practical interest this case, it is appropriate to use a variable or adaptive stepsize rather than a fixed timestep schema.

Press et al. [36] suggest that most problems will benefit from a higher-order scheme and therefore suggest three types of higher-order implicit methods for use with stiff systems, namely:

- Generalizations of the Runge-Kutta method such as the Kaps-Rentrop methods,
- Generalizations of Bulirsch-Stoer method, and
- Predictor-corrector methods.

They cite one example of a stiff problem in which a Kaps-Rentrop scheme is able to solve the problem in 29 steps, whilst a Runge-Kutta scheme would require 51,012! The Kaps-Renstrop scheme is simple to implement and competitive with more advanced schemes "for moderate accuracies (tolerances of $10^{-4} - 10^{-5}$) and moderate sized-systems ($N \lesssim 10$)". The Semi-Implicit Extrapolation Method (a generalization of the Bulirsch-Stoer method) provides good accuracy but is slightly more complex to implement. For this reason the Semi-Implicit Extrapolation (SIE) method will be adopted in the first instance, with the option of reverting to the Kaps-Renstrop scheme if necessary.

Press et al. provide a C++ implementation of the SIE approach [38] which will be used 'as is' for this implementation. Functions for the Jacobian and the derivatives need to be provided to this scheme, along with appropriate controlling code.

Semi-Implicit Euler

The SIE scheme is based on the semi-explicit Euler method of solution. As discussed previously, the explicit Euler method uses a simple linear assumption to calculate the function value at the next point in time, based on the gradient at that point, written as

$$y_{n+1} = y_n + hy'_n \, .$$

The most simple approach to solving stiff equations would be to modify this scheme to become explicit, such that

$$y_{n+1} = y_n + hy'_{n+1}$$

which gives better stability. Applying implicit differencing to a generalized set of equations given by

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}) \quad \text{gives} \quad \mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{f}(\mathbf{y}_{n+1}).$$

This generalized case is often extremely complex to solve (as in this case) and needs to be done iteratively at each step. Linearizing this solution, as in Newton's method gives

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left[\mathbf{f}(\mathbf{y}_n) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right|_{\mathbf{y}_n} .(\mathbf{y}_{n+1} - \mathbf{y}_n)\right].$$

which uses the Jacobian previously utilised for the eigenvalues (although employing all six equations in this case). It is this linearisation that gives the method the name of *semi-implicit* rather than implicit. Rearranging this gives

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left[\mathbf{1} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right]^{-1}.\mathbf{f}(\mathbf{y}_n) \qquad \text{(Eqn 3.8)}$$

which defines the semi-implicit Euler method. This method doesn't guarantee stability but it usually is since the local behaviour is nominally linear.

Semi-implicit Extrapolation (SIE) Method

Rewriting Equation 3.8 as

$$\left[\frac{\mathbf{1}}{h} - \frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right].(\mathbf{y}_{n+1} - \mathbf{y}_n) = \mathbf{f}(\mathbf{y}_n)$$

gives the fundamental equation of the method. However, as with Bulirsch-Stoer methods an extrapolation method is used. The method employs *Richardson's deferred approach to the limit* which considers the final solution to be an analytical function of the time step. Fitting the polynomial function is achieved by first taking a deliberately large time step (*H*), and then seeing how the function changes as successively smaller time steps are taken. Once this function is fitted then it can be evaluated at the limit of infinitely small time steps (see Figure 3.9).

A full description of this technique and all of the numerical techniques used to obtain good performance is beyond the scope of this thesis, but the details are supplied [36]. Similarly, the complete listing for this numerical solution contains around 2,000 lines of code which are not reproduced here. The code was developed using the C++ language and developed and compiled using the Apple software Xcode.

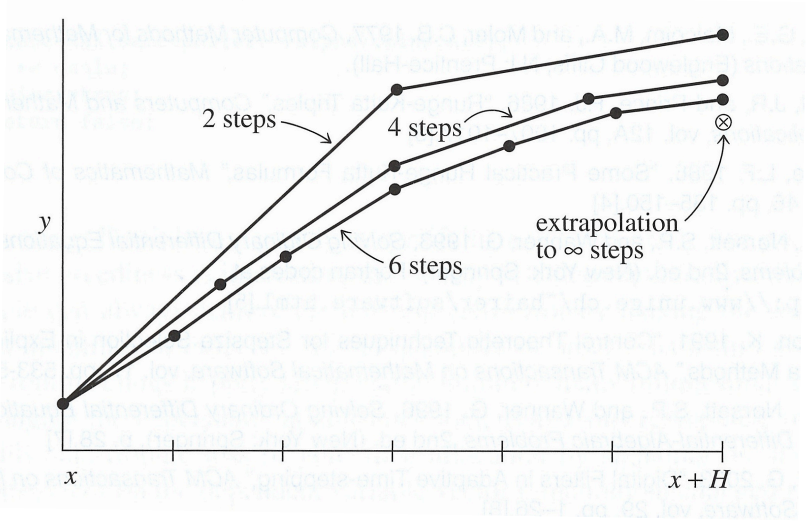


Figure 3.9 - Richardson Extrapolation (from [36])

However, a complete numerical solution to the problem employing the SIE method was successfully implemented which delivers as its output, the 6 variables defined in the governing equations, namely three-dimensional velocity and three-dimensional displacement for a particle falling through an atmosphere with altitude-dependent density and acted upon by altitude-dependent wind and gravity.

The next Chapter will detail attempts to verify the coded solution and also present results obtained from the model.

Chapter 4 - Results and Discussion

4.1 Introduction

This Chapter will discuss the results obtained from running the numerical model in different conditions.

Initially, some comparisons will be conducted with the analytical solution for the simplified case solved in Appendix 1 (see Section 3.4) in order to provide verification and confidence in the numerical solution. Further to that, some simple cases incorporating wind will be examined to provide an initial check. It is not easily possible to validate the model, due to a lack of real-world breakup data.

A practical form of sensitivity analysis will then be conducted in order to identify the magnitude of changes that can be expected in a problem as different parameters change. This will allow an investigator to focus effort on improving the accuracy of relevant parameters rather than wasting time on aspects which may become inconsequential.

4.2 Verification

In order to provide some simple verification and confidence in the numerical model, it will first be compared to the simple analytical solution given in Appendix 1. The analytical model incorporates mass, constant gravity and a drag force proportional to the square of velocity (taken as $0.5\rho C_D S v^2$) assuming a nominal constant density (taken as $1.17\,kg/m^3$). Three conditions will be assessed, corresponding to low (30m), medium (300m) and high altitude (10,000m). All conditions assume the same mass ($m = 100kg$), frontal area ($S = 1m^2$) and drag coefficient ($C_D = 1$), giving a constant ballistic coefficient ($C_B = 100kg/m^2$). There is no initial particle velocity and no wind velocity. All particle velocities are vertical, since the analytical model is one-dimensional.

Comparison with analytical model - Low altitude

Figure 4.1 shows the vertical particle velocity against time for the analytical solution of Appendix 1 and the numerical integration routine discussed in Chapter 3, for a drop of 30m. It also shows the difference between the two values as a function of time in the form of 100 times the magnitude of the difference (100*|difference|) to make it visible on the same axes. Figure 4.2 shows the altitude versus time for the same condition.

This condition has been chosen since it provides a negligible difference in atmospheric density. The two methods should show agreement and thus provides an initial verification condition.

The velocity graph in Figure 4.1 shows excellent agreement, with both velocities increasing nearly linearly for the duration of the drop. At the ground impact point there is a final velocity of approximately 22.2 m/s with a difference in velocities of 0.1 m/s (0.45%). The altitude plot in Figure 4.2 also shows excellent agreement with the two curves being almost indistinguishable. The final difference in altitude is 0.07m (0.23%).

These two plots provide good confidence that in this condition the numerical solution is working as it should be since agreement with the analytical solution is very good. The small differences may be due to slight differences in chosen density value, value of gravitational acceleration or slight numerical inaccuracies during the calculation.

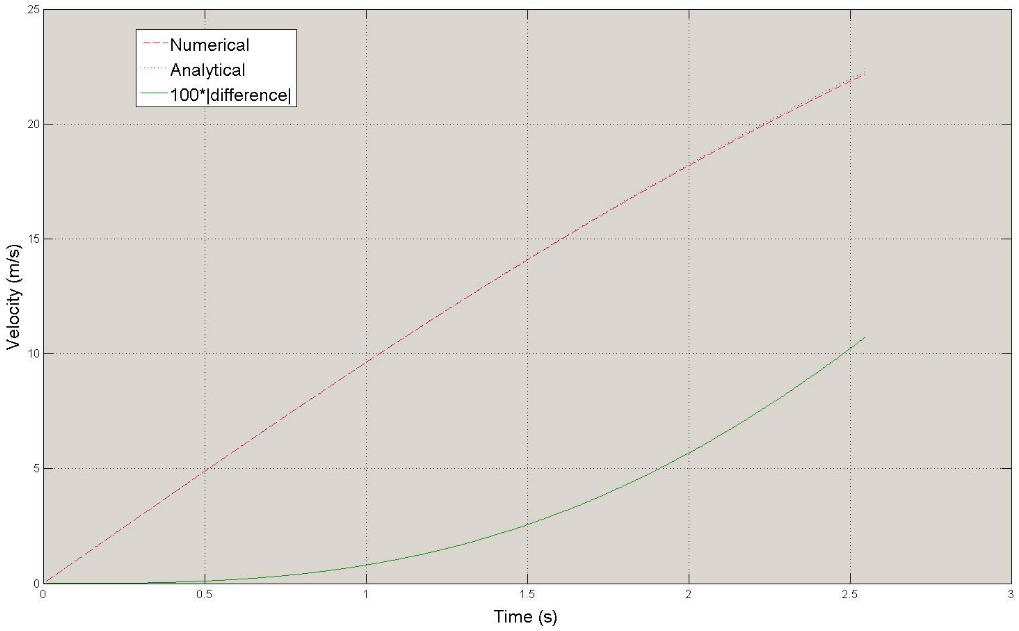


Figure 4.1 - Comparison of simple analytical solution to numerical approach for velocity versus time, ($v_0 = [0,0,0]$, $z_0 = 30m \approx 100ft$, $v_{WIND} = [0,0,0]$ *)*

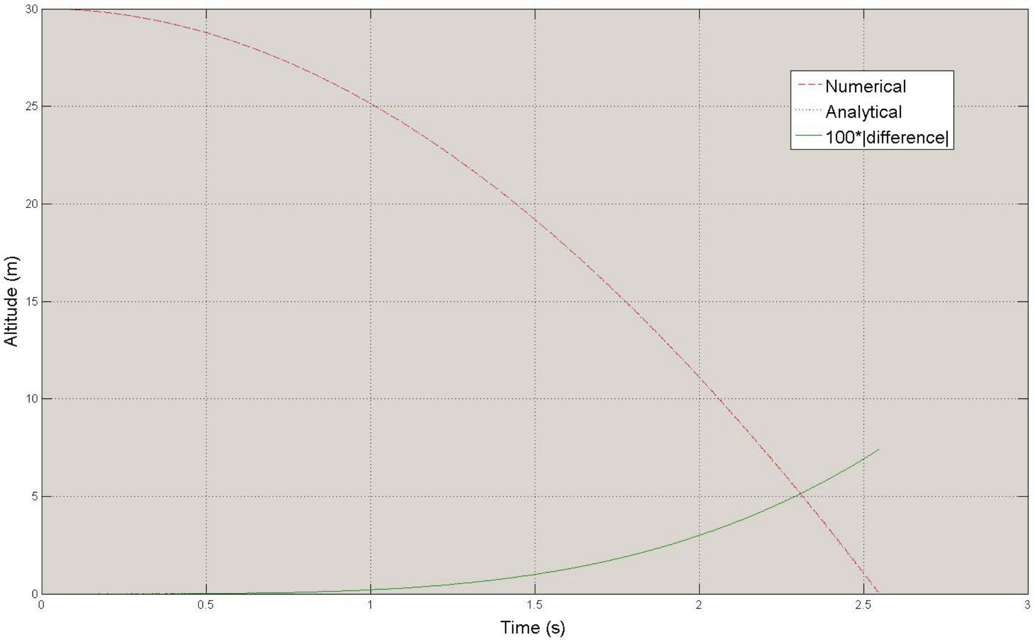


Figure 4.2 - Comparison of simple analytical solution to numerical approach for altitude versus time, ($v_0 = [0,0,0]$, $z_0 = 30m \approx 100ft$, $v_{WIND} = [0,0,0]$ *)*

Comparison with analytical model - Medium altitude

Figure 4.3 shows the particle velocity against time for the analytical solution of Appendix 1 and the numerical integration routine discussed in Chapter 3, for a particle dropped from 300m. It also shows the difference between the two values as a function of time in the form of ten times (not the 100 times of the previous case) the magnitude of the difference (10*|difference|) to make it visible on the same axes. Figure 4.4 shows the altitude versus time for the same particle.

This condition has been chosen since it provides a reasonable amount of time for the differences to build, whilst keeping the particle within a relatively small range of atmospheric density. The two methods should show quite good agreement and hence this provides a useful verification condition.

The velocity plots using the two methods show extremely good agreement until around two seconds elapsed, at which point they start to diverge. However, this divergence is relatively small with a maximum difference of 0.7 m/s (1.8 %) at ground impact with a final velocity of 39.6 m/s. The altitude plot also shows little deviation until around two seconds which is as would be expected, since the displacement is simply the integral of the velocity. The altitudes diverge reaching a maximum difference of 28.7 m (2.87 %) at ground impact during a fall of 1,000 m. The reason for the increased error seen with the altitude is that any velocity error during the fall leads to an increasing displacement error, due to the integration.

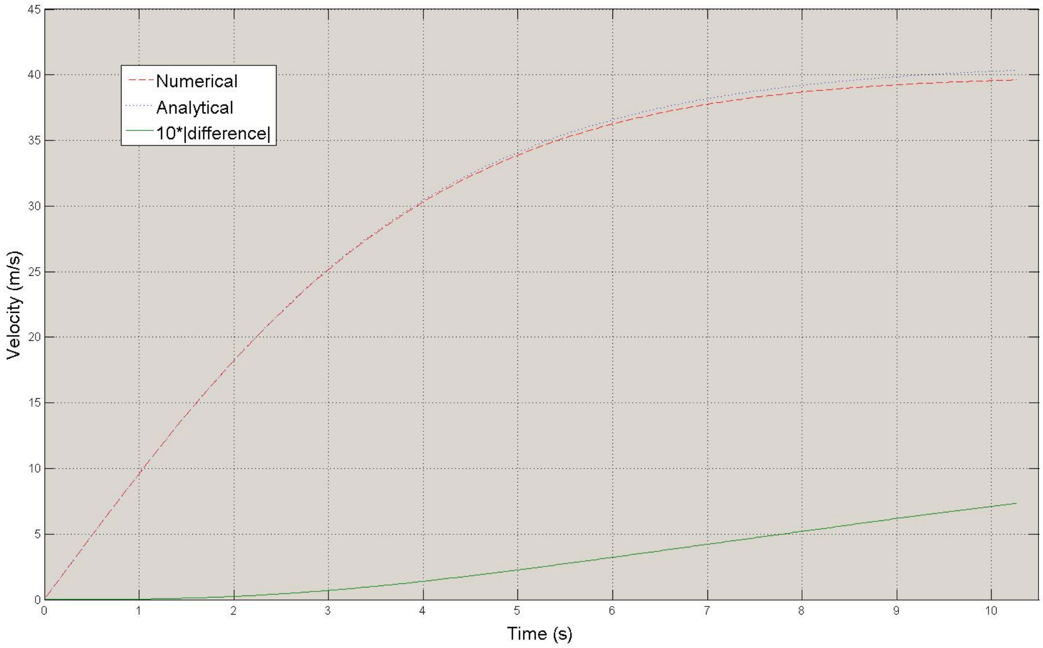


Figure 4.3 - Comparison of simple analytical solution to numerical approach for velocity versus time, ($v_0 = [0,0,0]$*,* $z_0 = 300m \approx 1000ft$ *,* $v_{WIND} = [0,0,0]$*)*

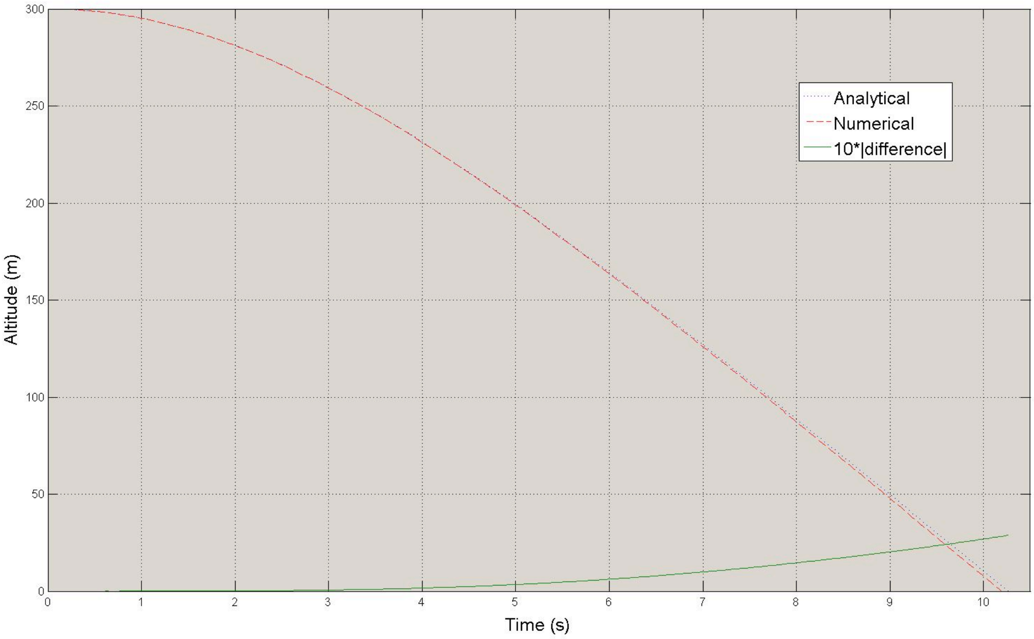


Figure 4.4 - Comparison of simple analytical solution to numerical approach for altitude versus time, ($v_0 = [0,0,0]$*,* $z_0 = 300m \approx 1000ft$ *,* $v_{WIND} = [0,0,0]$*)*

Comparison with analytical model - High Altitude

Figures 4.5 and 4.6 show the velocity and altitude plots for a 10,000m drop with zero initial velocity and zero wind. The difference plot on both Figures is no longer scaled, reflecting the increased error.

The velocity plot in Figure 4.5 shows good agreement for the first few seconds between the analytical and numerical solutions. However, the velocities quickly diverge. This is because the analytical solution does not allow for variable density or variable gravitational acceleration and hence the terminal velocity which is reached is constant through the fall. The numerical solution incorporates both of these effects, and shows the reducing terminal velocity as altitude decreases. Since the analytical solution is constant, the velocity difference reflects the shape of the numerical curve. The increase in error at the end of the fall is due to taking the magnitude of the difference, which would otherwise have changed sign at this point.

The divergence of the the altitudes shows the numerical model predicting a faster loss of altitude (due to the higher terminal velocity) resulting in an increasing difference peaking at approximately 2,000m. This altitude difference corresponds to a difference in impact time of approximately 47 seconds. The rate of increase of the difference reduces with time due to the numerically-calculated velocity converging with the constant analytical terminal velocity.

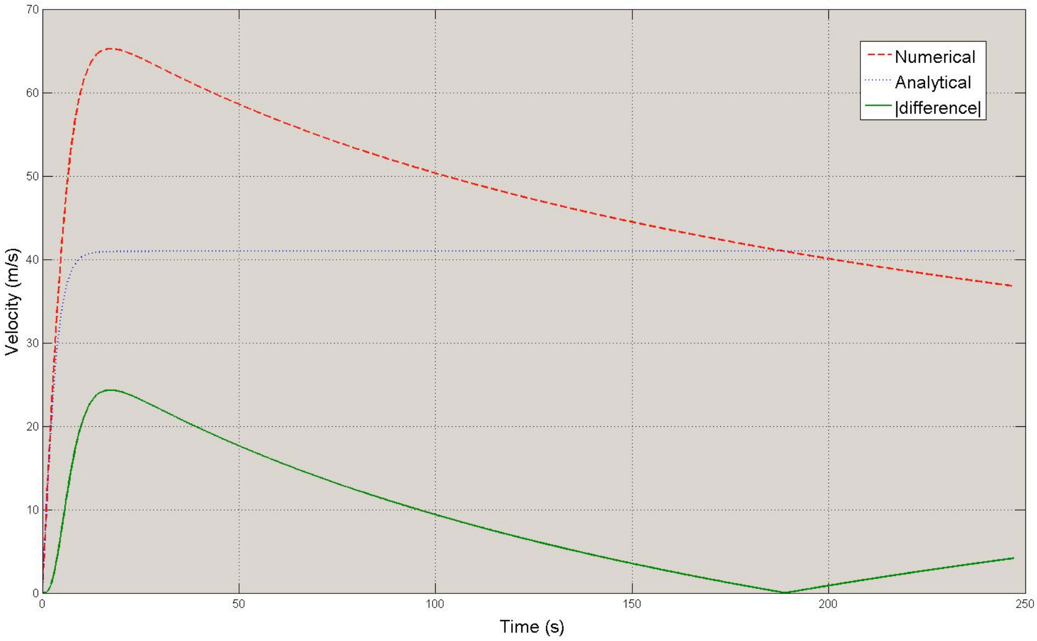


Figure 4.5 - Comparison of simple analytical solution to numerical approach for velocity versus time, ($v_0 = [0,0,0]$*,* $z_0 = 10{,}000m \approx 33{,}000 ft$ *,* $v_{WIND} = [0,0,0]$*)*

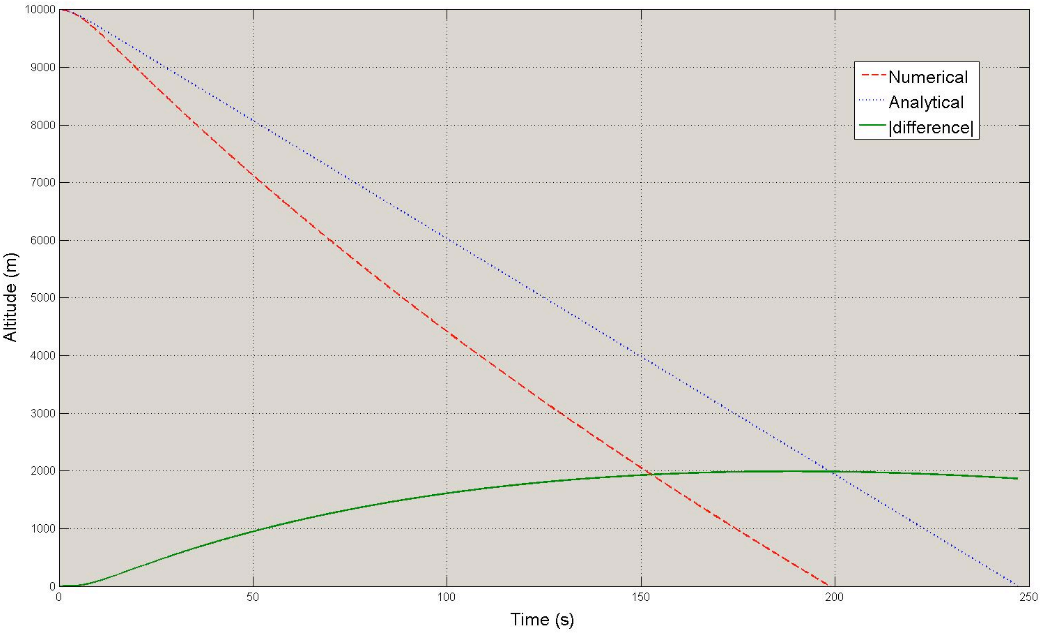


Figure 4.6 - Comparison of simple analytical solution to numerical approach for altitude versus time, ($v_0 = [0,0,0]$*,* $z_0 = 10{,}000m \approx 33{,}000 ft$ *,* $v_{WIND} = [0,0,0]$*)*

Initial velocity

Figures 4.7, 4.8 and 4.9 show the effect of introducing forward speed into the initial conditions. In these examples, an initial x-direction (aircraft track) velocity of 250 m/s (485 kts) is introduced.

Figure 4.7 shows the z-direction (vertical) velocity and displacement versus time which is similar to that obtained in Figures 4.5 and 4.6 as would be expected. Figure 4.8 shows the x-direction velocity and displacement. The velocity starts at 250 m/s, as imposed by the initial conditions, and then decreases exponentially towards zero, since the windspeed is zero. The x-displacement increases exponentially towards a final value of 700 m which is reached by around 40 s. Since the z-direction fall continues for another 70 s, it follows that the majority of the fall will continue purely vertically and this is represented in Figure 4.9. Here the two velocity components are almost independent; the x-velocity starts very large and decays very rapidly whilst the z-velocity begins at zero and then grows to a terminal velocity much smaller than the initial x-velocity. This relationship would change with a change in ballistic coefficient.

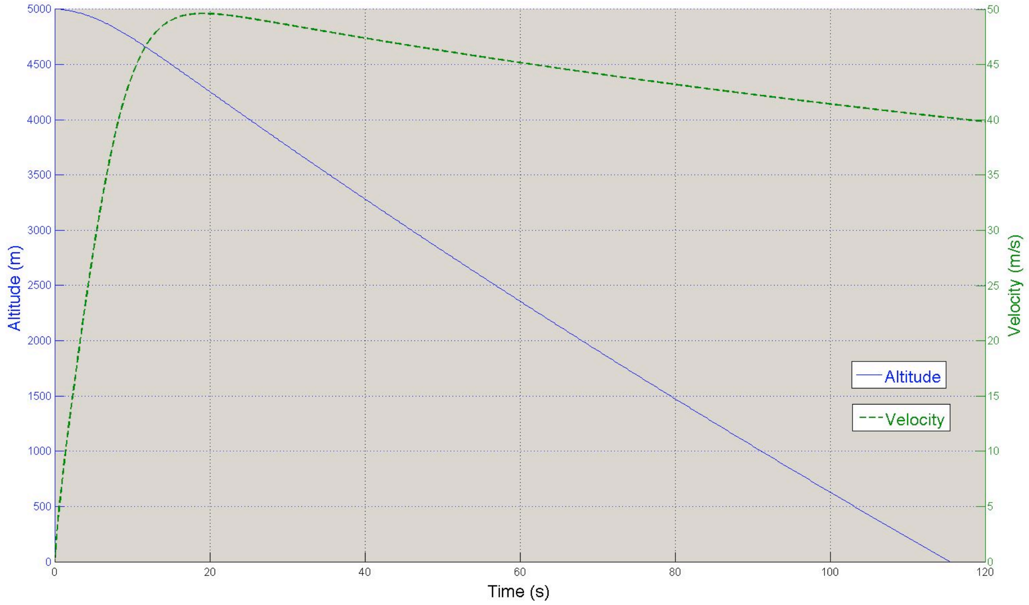


Figure 4.7 - Vertical velocity and displacement against time
$v_0 = [250,0,0]$, $z_0 = 5{,}000m \approx 16{,}400ft$, $v_{WIND} = [0,0,0]$

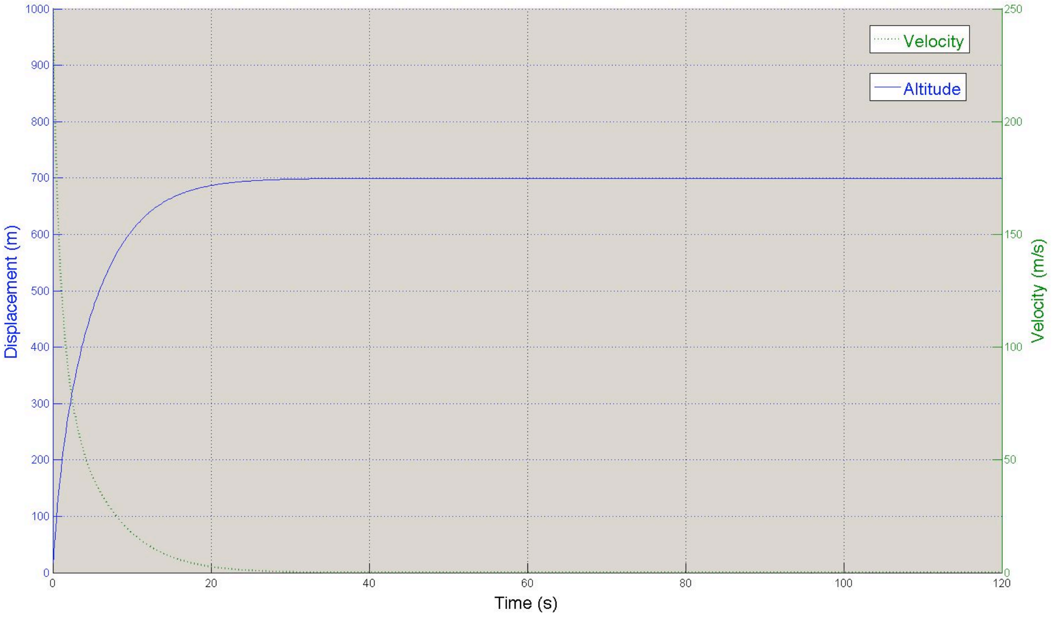


Figure 4.8 - X-direction velocity and displacement against time
$v_0 = [250,0,0]$, $z_0 = 5{,}000m \approx 16{,}400 ft$, $v_{WIND} = [0,0,0]$

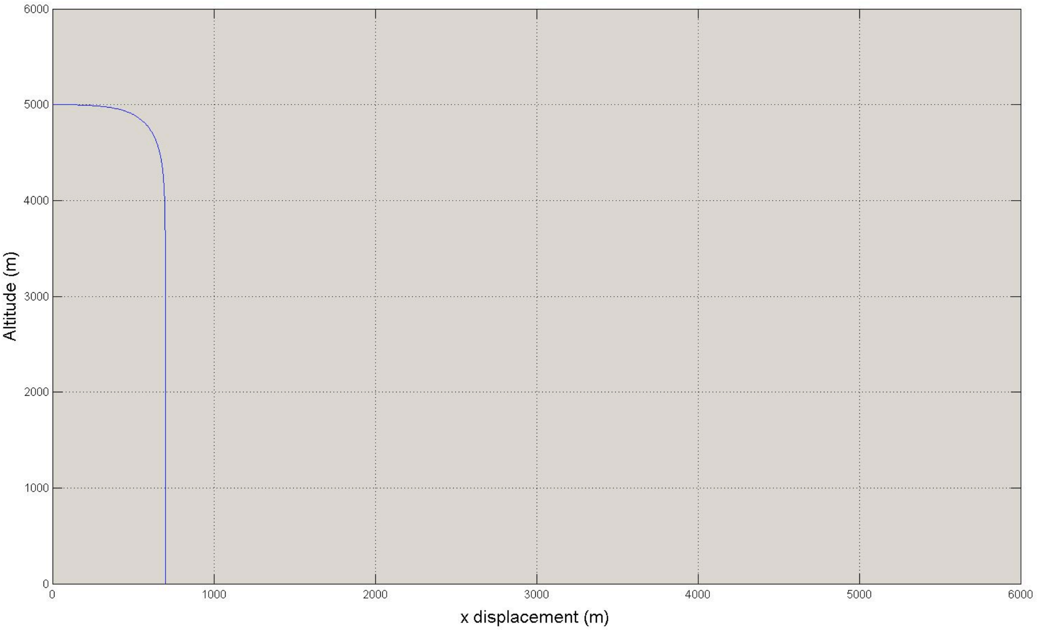


Figure 4.9 - Altitude against x-direction displacement
$v_0 = [250,0,0]$, $z_0 = 5{,}000m \approx 16{,}400 ft$, $v_{WIND} = [0,0,0]$

Initial velocity and wind

In this condition, a headwind (i.e. negative x-direction) is introduced. Figures 4.10, 4.11 and 4.12 replicate those of the previous section but with a headwind of 25 m/s (50 knots) being introduced. Figure 4.10 is identical to Figure 4.7 which is as it should be, since the introduction of an x-direction wind component should have no effect on the z-direction behaviour.

Figure 4.11 shows the x-direction velocity and displacement. The velocity starts at 250 m/s, as imposed by the initial conditions, and then decreases exponentially towards a constant speed of -25 m/s, i.e. the windspeed, which is as it should be. The x-displacement increases initially to a value of around 390m before reducing at a constant rate, given by the constant negative velocity, for the remainder of the fall. The variation in x-direction displacement with altitude is shown in Figure 4.12. When compared with Figure 4.9, the effect of a constant windspeed can be seen, providing a constant rate of displacement (in this case negative since it is a headwind).

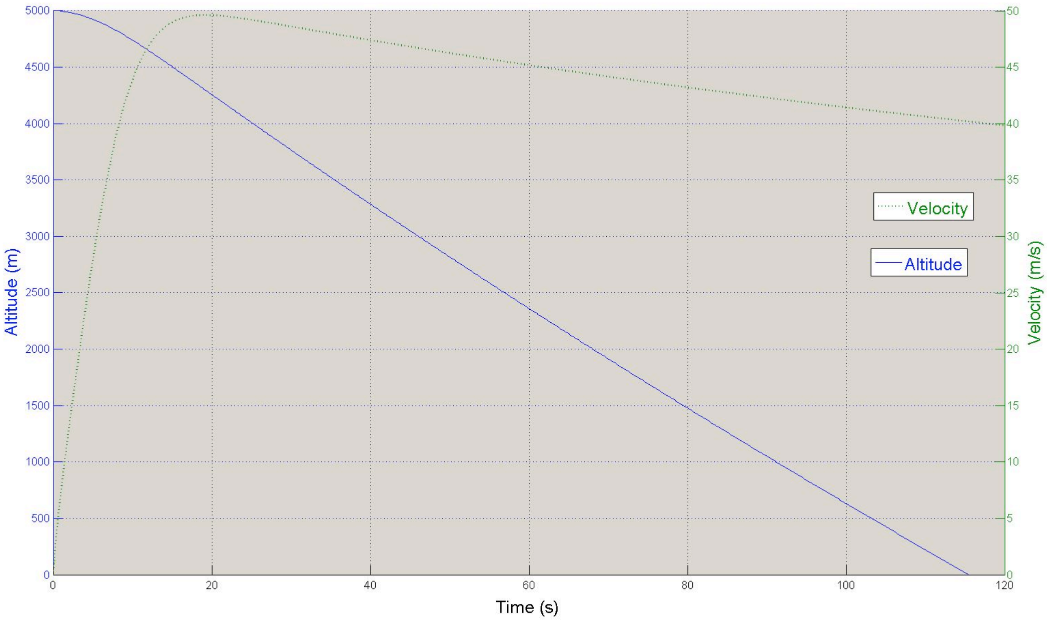


Figure 4.10 - Vertical velocity and displacement against time
$v_0 = [250,0,0]$, $z_0 = 5{,}000m \approx 16{,}400ft$, $v_{WIND} = [-25,0,0]$

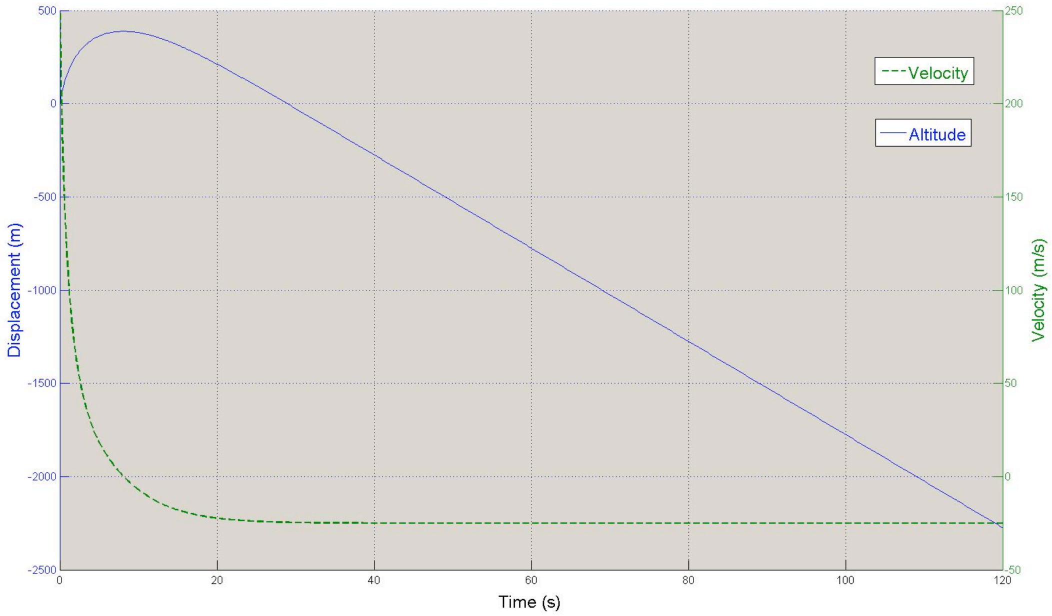


Figure 4.11 - X-direction velocity and displacement against time
$v_0 = [250,0,0]$, $z_0 = 5{,}000m \approx 16{,}400ft$, $v_{WIND} = [-25,0,0]$

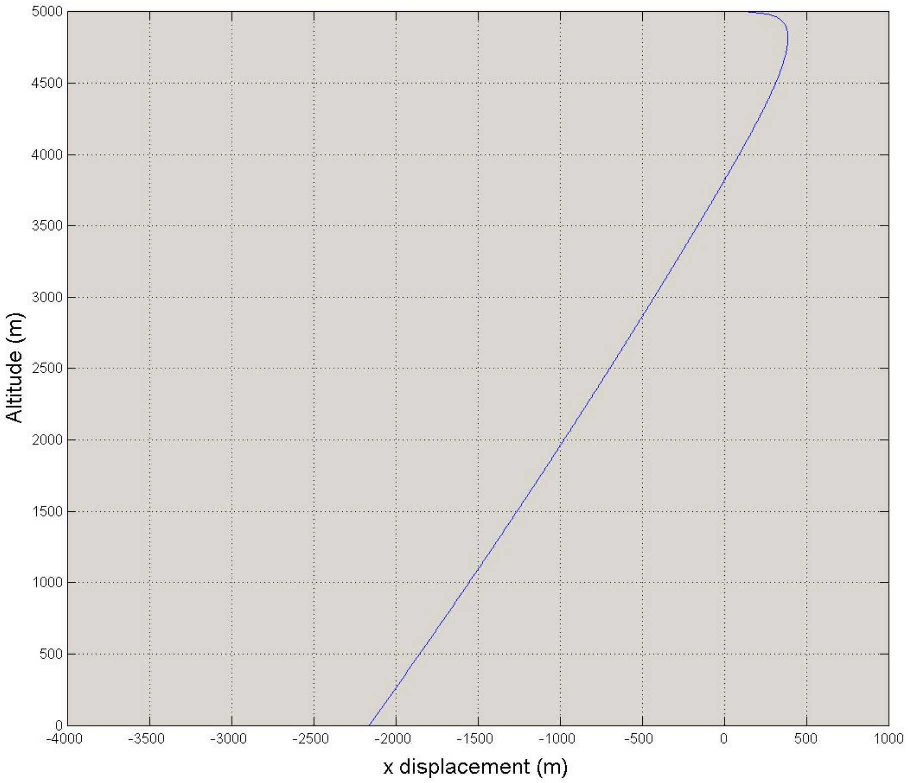


Figure 4.12 - Altitude against x-direction displacement
$v_0 = [250,0,0]$, $z_0 = 5{,}000m \approx 16{,}400ft$, $v_{WIND} = [-25,0,0]$

These simple verification and comparison tests have provided confidence in the numerical solution as it has been implemented. Therefore, in the following analyses, the model will be assumed to be correct and focus will instead switch to results obtained from the model.

4.3 Practical Sensitivity Analysis

Having verified the model, the numerical solution now provides the ability to calculate the trajectory of a particle given a set of initial conditions and wind data. Therefore, the model allows investigators to calculate potential trajectories based on the information they have available. However, whilst the model provides answers for the situation it is given, it provides no information about which factors are important in the behaviour of a given particle. Put differently, there is no 'sensitivity' information available - it is not clear whether changing, say, the initial velocity will induce a massive change in final ground impact position, or whether it will be almost inconsequential. For this reason, a form of sensitivity analysis will be conducted.

When constructing this sensitivity analysis approach, it is important to be conscious of the desired final application, i.e. the study of aircraft breakups. Whilst it might be academically interesting to study, say, the variation in time of a trajectory with changes in particle mass, this may be of little relevance to the problem of accident investigation, where normally the key information is at the point of ground impact. (An exception to this may be in trying to match radar traces to a breakup sequence). However, such analysis can always be achieved by an individual investigator running a range of specific scenarios.

This analysis will instead focus on uncertainty levels. Faced with a given scenario, there is likely to be uncertainty around a number of different variables used in the analysis including: wind speed and direction, initial position, initial velocity, drag coefficient etc. In trying to understand a particular set of events, an investigator may wish to vary certain variables through a range of values or improve the accuracy of certain parameters. However, establishing which parameters are likely to produce the greatest difference in ultimate wreckage location will be a process of trial and error.

Therefore, the following analysis will take two example initial conditions about which parameters will be varied by some given percentage. This will allow a greater understanding of the key variables which may be required.

It is worth noting that the problem in hand is highly nonlinear. This means that in practice, any results obtained for a specific problem set are valid for only that problem, and hence are not generalizable to all problems. However, many of the parameters in this problem vary smoothly and hence rapid deviations in behaviour from one problem set to another are unlikely to be encountered.

The approach adopted is not a rigorous sensitivity analysis, which has specific statistical meaning, but rather a practical approach aimed at offering general guidance. The two problem sets will be:

A simulated large aircraft breakup - a breakup at 10,000 m (c. 33,000 ft), with an initial forward velocity of 250 m/s (c. 485 kts) and a cross/tail wind of 45 m/s, decreasing with reducing altitude).

A simulated small aircraft breakup - a breakup at 1,000 m (c. 3,250 ft), with an initial forward velocity of 60 m/s (c. 120 kts) and a light cross/tail wind of around 12 m/s (decreasing with reducing altitude).

The parameters which will be varied are:

- Breakup altitude,
- Initial x-direction (aircraft track) velocity,
- Wind direction, and
- Wind magnitude.

Drag coefficient will not be studied explicitly. However, the scenarios will be calculated for a range of ballistic coefficients and hence changes in drag coefficient can be inferred by moving from one ballistic coefficient value towards the next.

The wind profile adopted is defined as:

Wind Speed (m/s)	Altitude (metres)	'To' heading (degrees)
45	10,000	45
19	6,000	45
14	3,000	45
1	0	45

A cubic polynomial is then fitted to this data which is equivalent to using one 'span' of the cubic spline. As described previously, the model is designed to use a cubic spline to describe the entire wind profile. The full implementation would simply calculate the velocities and displacements for the initial span and use those as the starting point for the next span. Clearly, in a real situation a more accurate, complex wind profile would be adopted. However, this has the potential to confuse the following analysis and hence a more easily visualised wind profile is included.

The results for each change will be presented individually, and then the differences for each parameter will be compared.

4.4 Large Aircraft Breakup

Breakup altitude variation

Figure 4.13a shows the effect of altering breakup height on the final location of wreckage. The plot shows the *x* and *y* location (with the breakup occurring at 0,0 and the aircraft travelling in the positive *x*-direction i.e. up the page) for the speed and wind conditions described above. Five values of ballistic coefficient are indicated (10,000, 1,000, 100, 10 and 1 kg/m^2), and for each case the effect of increasing or reducing the breakup altitude by 10% (=1,000 m) are shown.

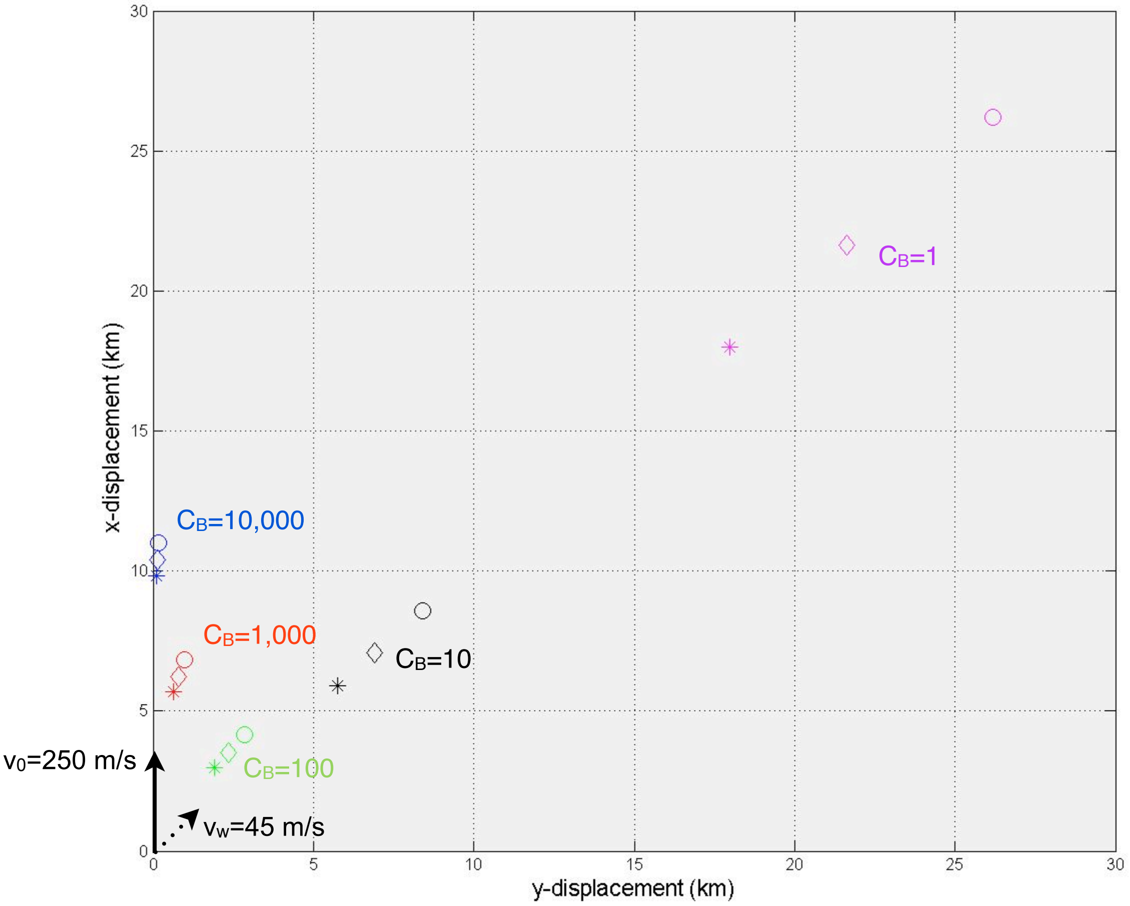


Figure 4.13a - The effect on final wreckage location of changing breakup altitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ✻ *= -10%*

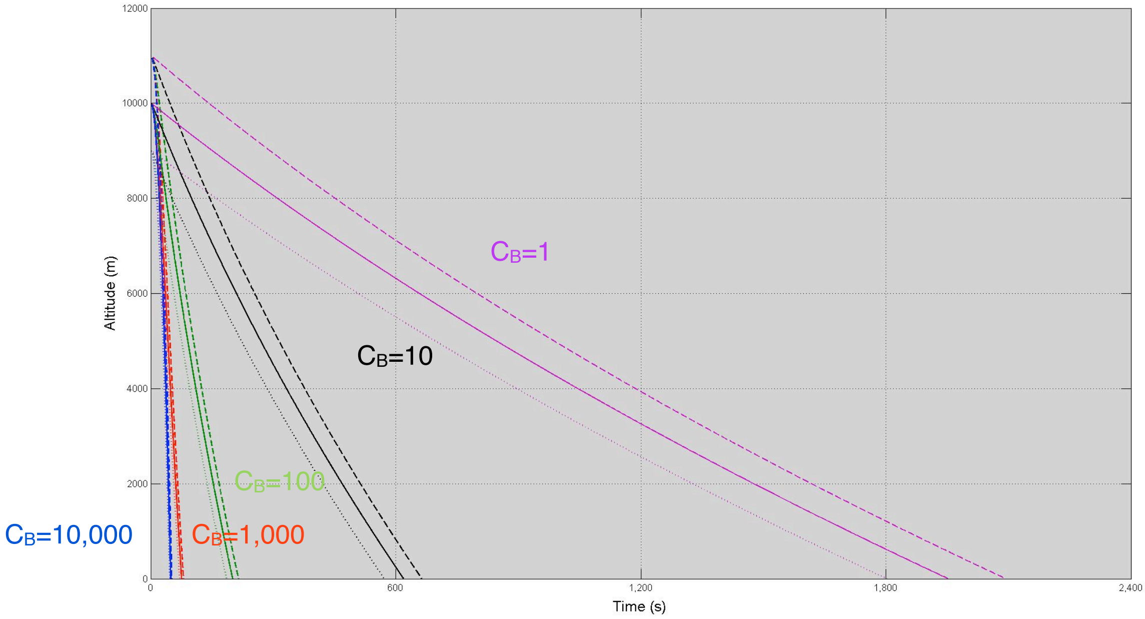


Figure 4.13b - The effect on time to fall to ground of changing breakup altitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

Figure 4.13b shows the time taken for objects with different ballistic coefficients to fall to ground under the reference condition, as shown in Figure 4.13a.

The first point of note from Figure 4.13a is the scale of the wreckage distribution. Wreckage is spread over an area of 30km x 30km, with the furthest pieces having ‘flown’ for in excess of 30 minutes as shown in Figure 4.13b. Clearly in the case of strong winds, these timescales and distances will increase.

The general shape of the wreckage distribution is a ‘hockey stick’ which has been highlighted in previous studies. This is governed by the two pairs of factors involved which are forward track and velocity, and wind speed and direction.

Figure 4.13a shows that for high C_B items (high mass / low drag) the effect of altitude increase or decrease is to slightly increase or decrease the ‘throw’ of the item. However, as C_B decreases, so the difference in ground position from the reference is increased, with a 10% increase in breakup altitude giving rise to around 6km of increased displacement. The reason for this is that the increased flight time of the low C_B particle, allows proportionally more time under the wind influence due to its low terminal velocity (≈6 m/s) compared to the high C_B particle with a high terminal velocity (>340 m/s).

As would be expected Figure 4.13b shows that the difference in time taken to fall to ground is greater for lower ballistic coefficients since those objects have a lower terminal velocity.

Initial velocity variations

Figure 4.14a shows the effect of varying initial velocity on the final location of particles. For each value of ballistic coefficient, the initial velocity is varied by 10% (=25 m/s) above and below the reference speed of 250 m/s.

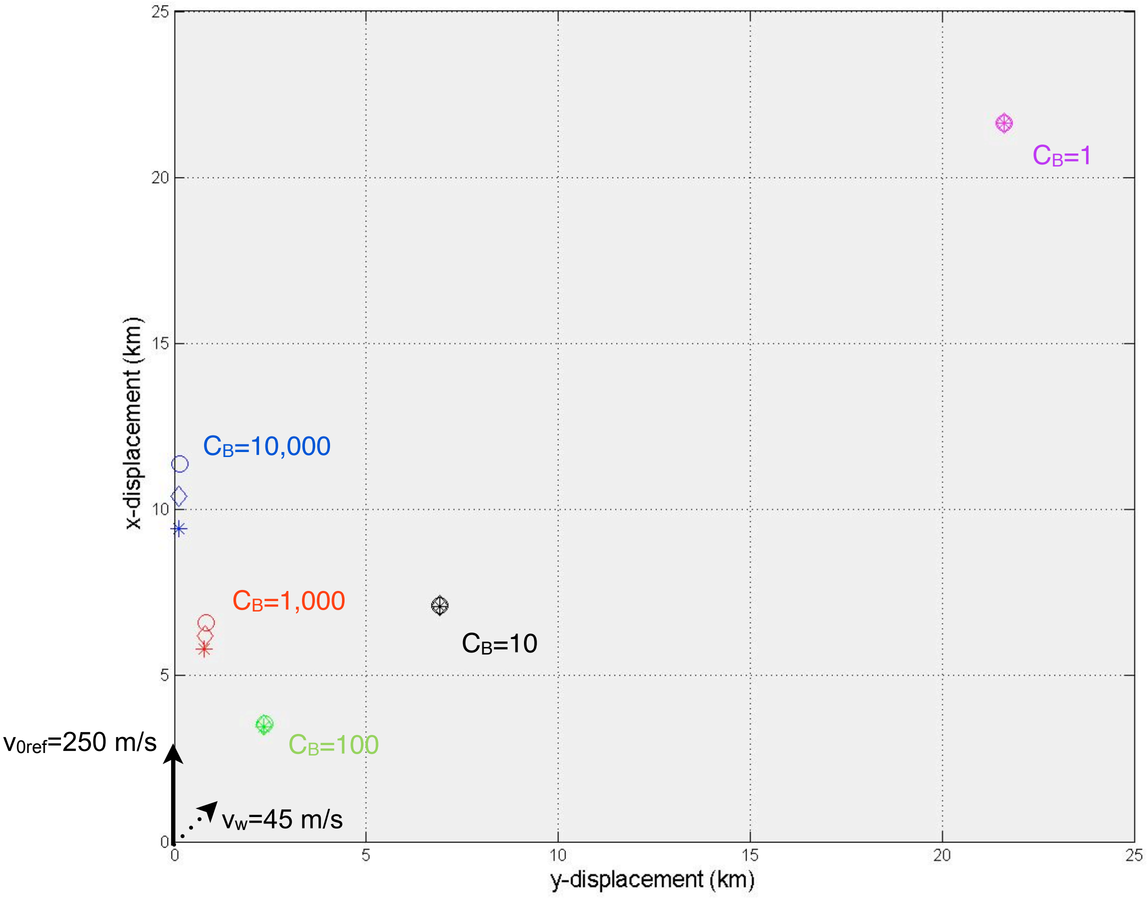


Figure 4.14a - The effect on final wreckage location of changing breakup forward velocity for a simulated large aircraft accident.
C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1
◇= *reference,* O = *+10%,* ✻ = *-10%*

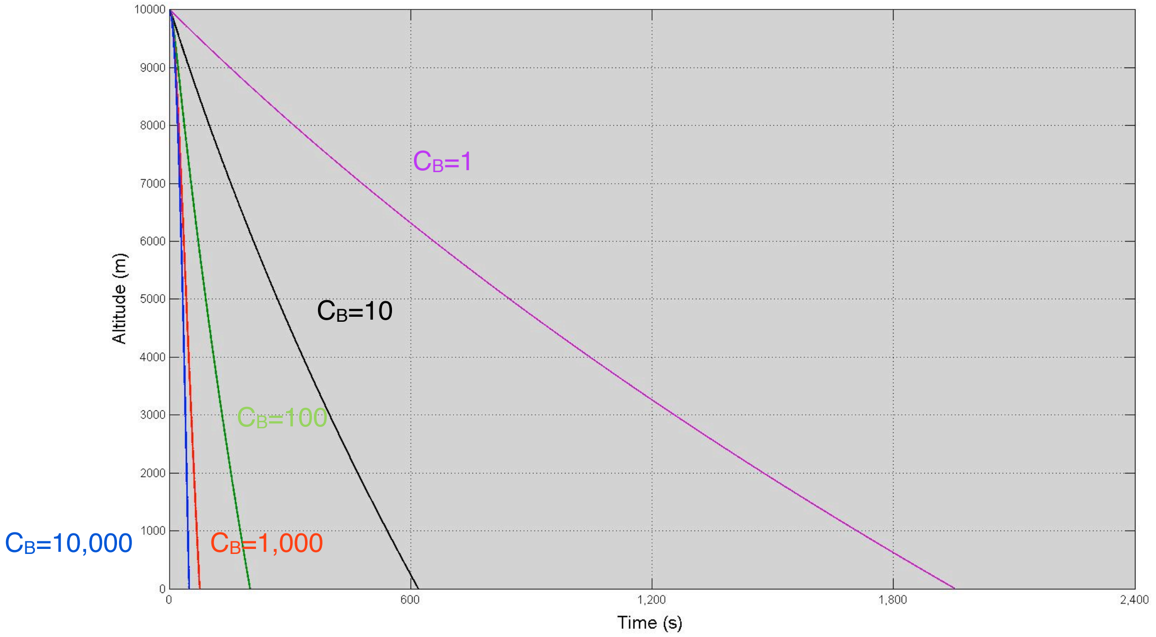


Figure 4.14b - The effect on time to fall to ground of changing breakup forward velocity for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

It is clear that the lower ballistic coefficient particles (C_B <= 100) are almost entirely unaffected by the change in initial velocity, with all showing near identical positions for all three cases. The reason for this is that low mass, high drag components decelerate extremely rapidly after release and therefore the 'modified' initial velocity has a very short period over which to influence the behaviour of the particle; it quickly adopts the surrounding windspeed.

The high ballistic coefficient particle, with high mass and low drag, are able to sustain the modified velocity for longer before finally adopting terminal velocity (if at all) and hence a larger difference is visible. A maximum difference of around 1km is noted for the case of C_B=10,000.

Figure 4.14b shows that forward velocity has no almost effect on the time taken to fall to ground; the lines for the three conditions are indistinguishable. The reason for this is that the vertical component of the fall is identical in each case i.e. the same initial vertical velocity from the same initial altitude. In an idealised situation, the equations are separable and changes to the two-dimensional, horizontal plane initial conditions or wind conditions in flight, do not affect the fall time. This can be seen for the next two cases of wind variation (Figure 4.15b and 4.16b). They are included for completeness but will not be commented upon further.

Wind magnitude variation

Figures 4.15a and 4.15b show the effect of variation in wind magnitude (but retaining a 'to' direction of 45°).

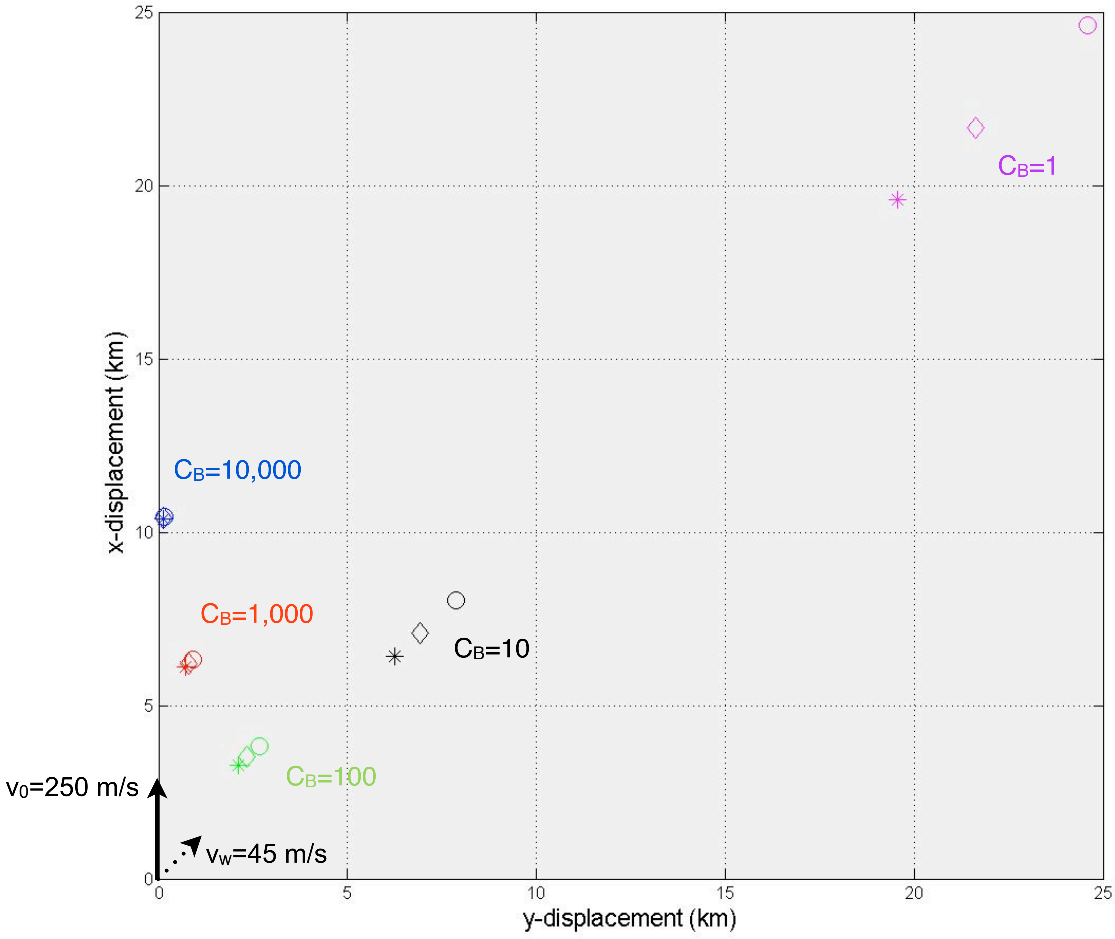


Figure 4.15a - The effect on final wreckage location of changing wind magnitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇= reference, O = +10%, ✱ = -10%

As would be expected, the lower C_B value items are more affected by the change in wind velocity than the higher value items. The effect on all items tends to extend or reduce the displacement along the ‘windline’, although the magnitude is clearly much greater for the lower C_B items.

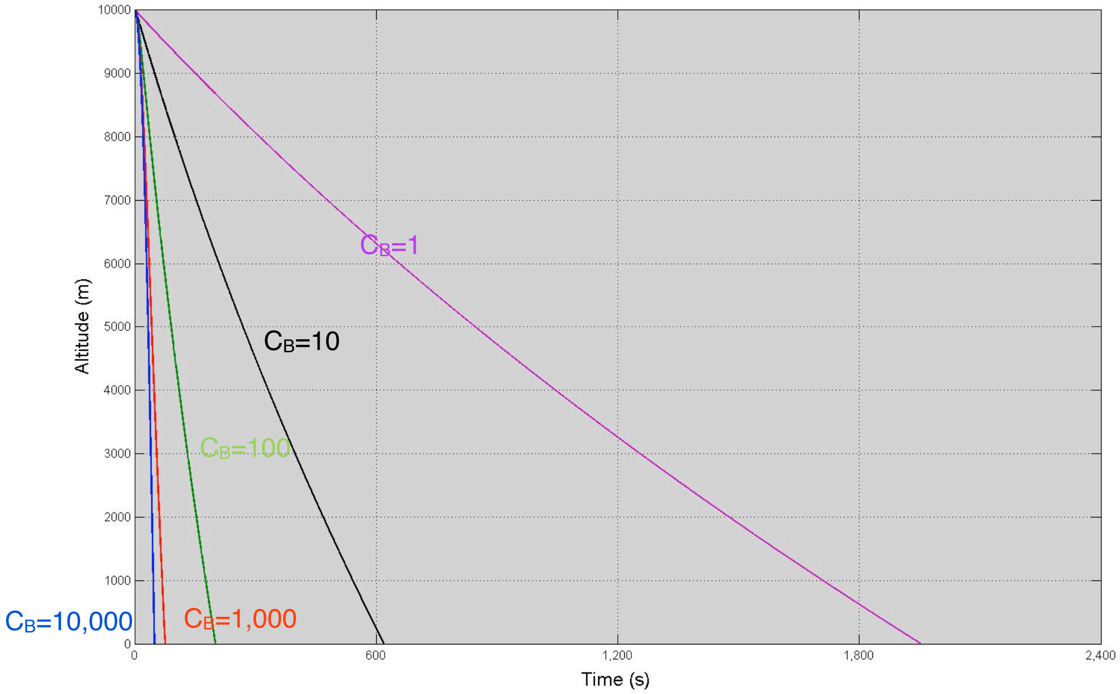


Figure 4.15b - The effect on time taken to fall to ground of changing wind magnitude for a simulated large aircraft accident..
C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1
—= reference, --- = +10%, ˙˙˙˙ = -10%

Wind angle variation

Figures 4.16a and 4.16b show the effect of varying the wind angle, where a positive change shifts the wind direction clockwise. Varying an angle by a percentage is more difficult than varying, say, a speed. Whilst 250 m/s is clearly more than 60 m/s, 250° is no ‘larger’ in physical terms than 60° and hence a variation of ±25° versus ±6° is not appropriate.

In this situation, a variation of ±9° was used since that represents 10% of a quadrant, and intuitively is the sort of error that might be expected. However, it is an arbitrary figure.

The pattern of variation with ballistic coefficient is matched to that in Figure 4.15a. However, rather than moving towards or away from the breakup point, the wreckage follows the change in windline angle. For this reason, the magnitude of movement from the reference point is related to the distance from the breakup point.

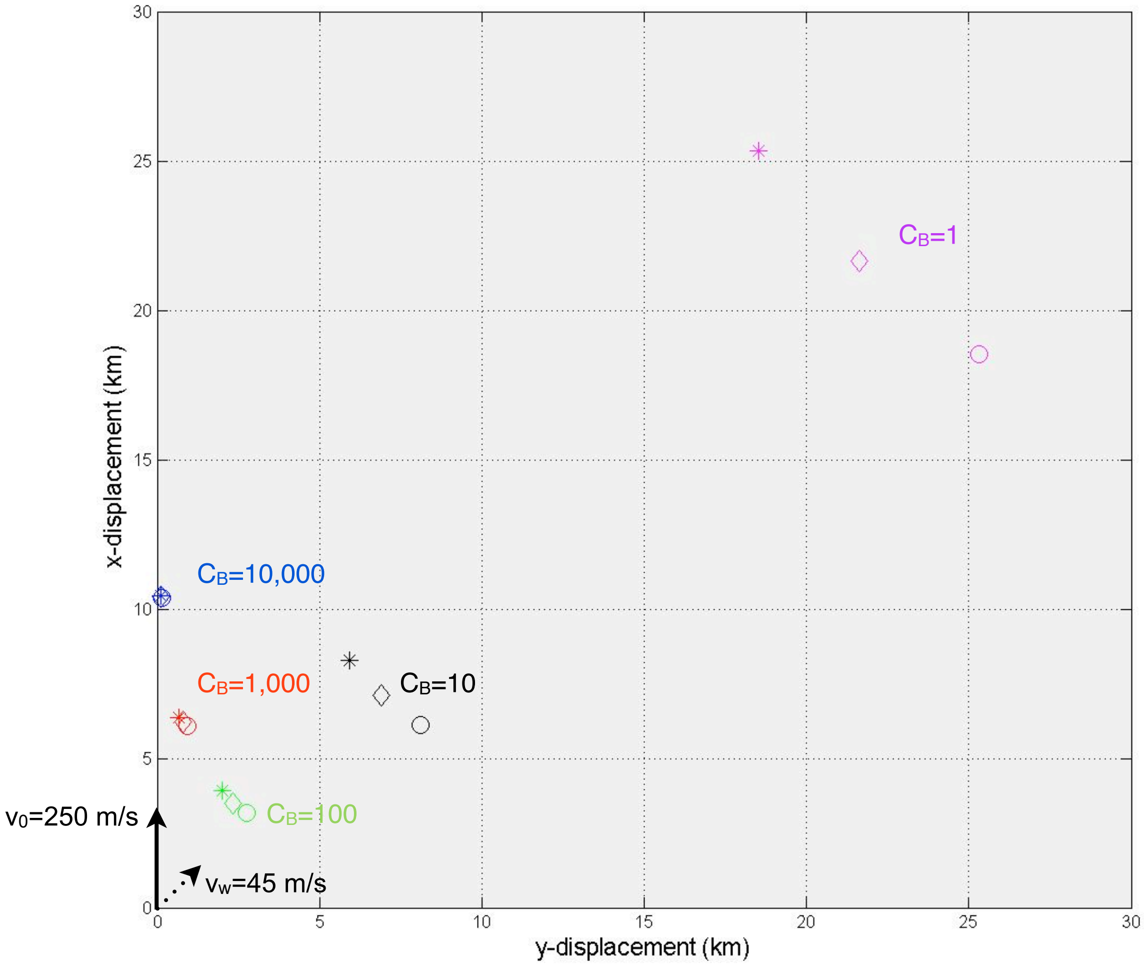


Figure 4.16a - The effect on final wreckage location of changing wind angle for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ = *reference,* O = *+10%,* ✻ = *-10%*

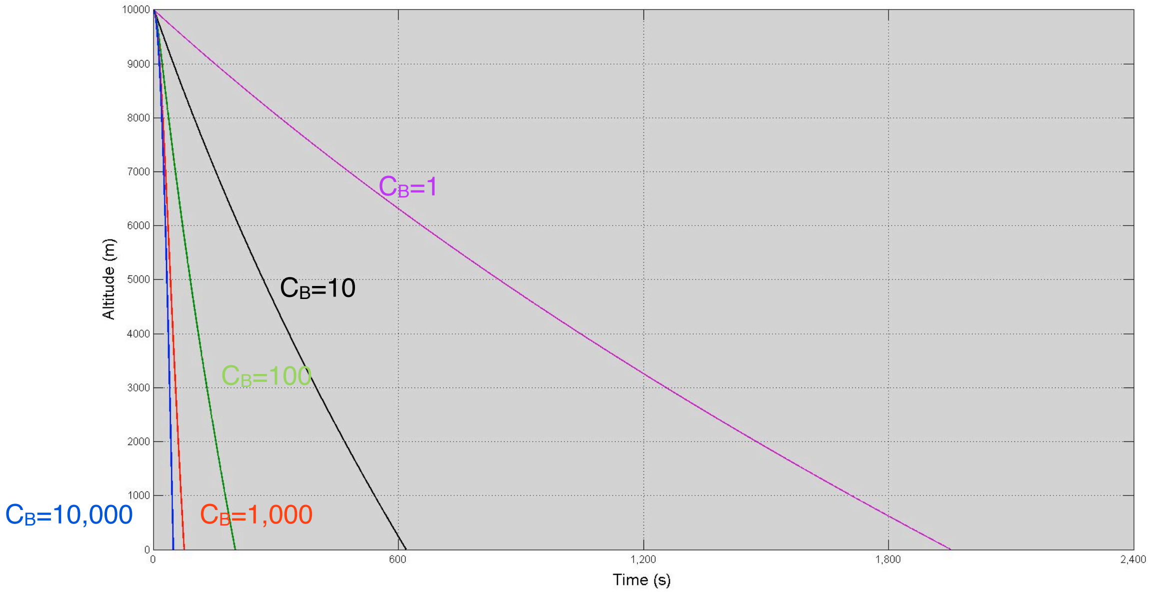


Figure 4.16b - The effect on time taken to fall to ground of changing wind angle for a simulated large aircraft accident..

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%,* ¨¨ *= -10%*

4.5 Small Aircraft Breakup

It is tempting to assume that since the same model is providing output for particles of identical C_B, that the plots for the small aircraft breakup will be very similar to those of the large aircraft breakup. Whilst it is certainly true that the effect on particles will be similar (e.g. high C_B prone to be affected by wind) the relationship from breakup-to-ground is nonlinear and hence the scaling of effects will vary considerably.

Altitude variation

Figure 4.17a shows the predicted wreckage ground impact for ballistic coefficients for a reference breakup altitude and ±10% (=100m). Comparison with Figure 4.13 shows the difference in scale that can be expected for such an accident (c. 1km x 1km rather than 30km x 30km for a large aircraft).

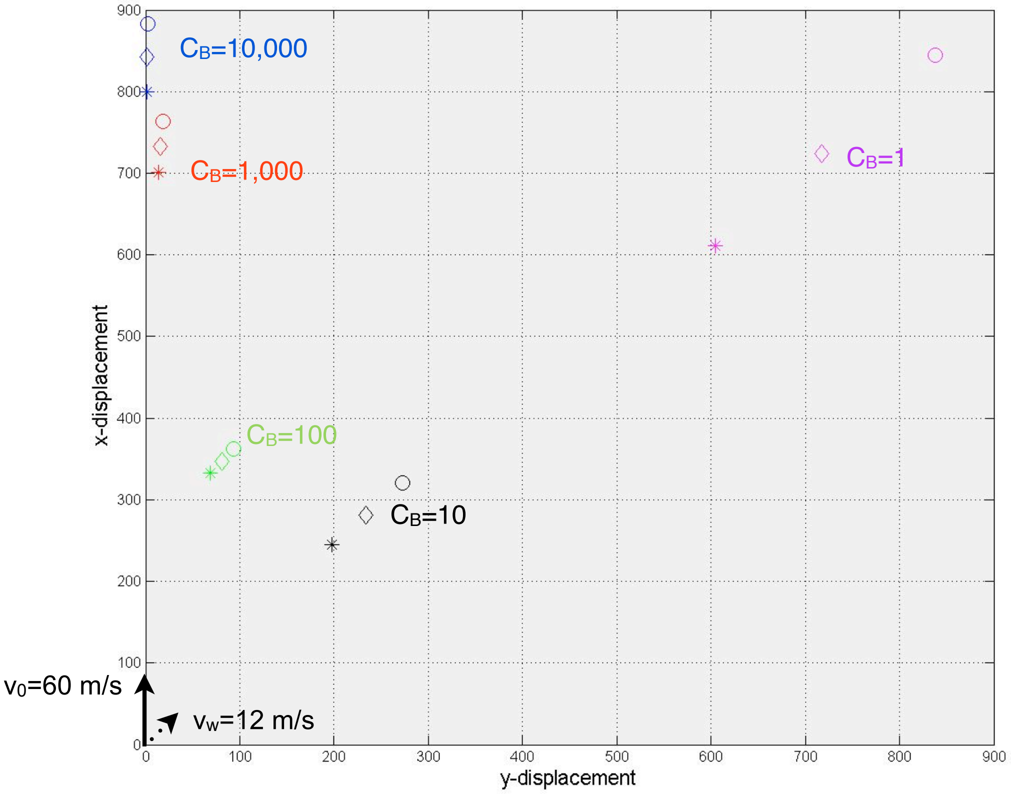


Figure 4.17a - The effect on final wreckage location of changing breakup altitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ = reference, O = +10%, ✱ = -10%

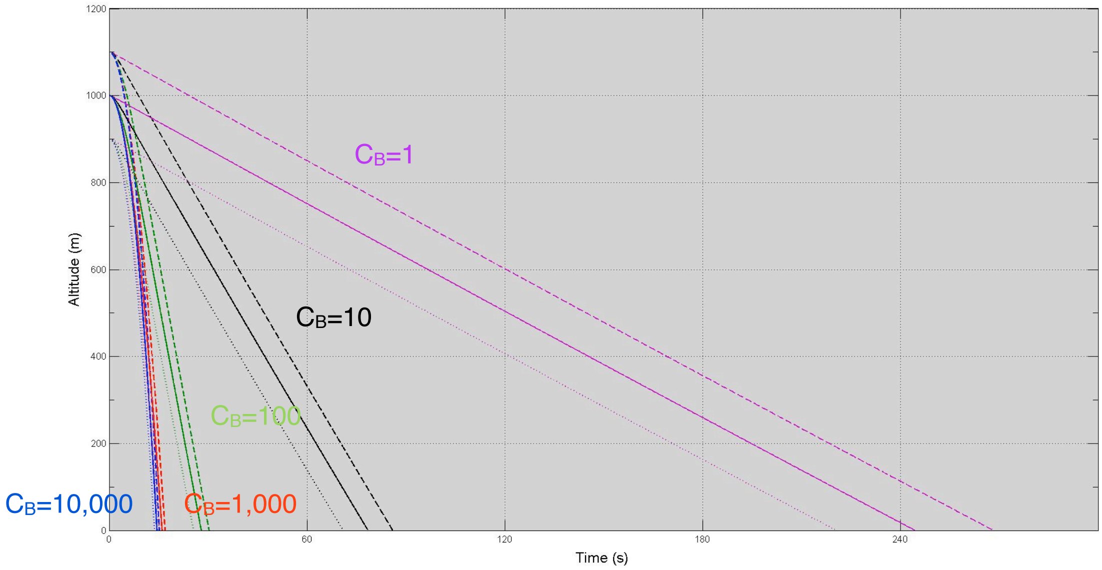


Figure 4.17b - The effect on time to fall to ground of changing breakup altitude for a simulated small aircraft accident

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%,* ···· *= -10%*

Initial Velocity Variation

Figure 4.18a shows the effect on the small aircraft breakup simulation of modifying initial forward velocity.

As with the large aircraft scenario, the high C_B items are virtually unchanged whereas the low C_B items are more greatly affected with all differences occurring along the line of aircraft track.

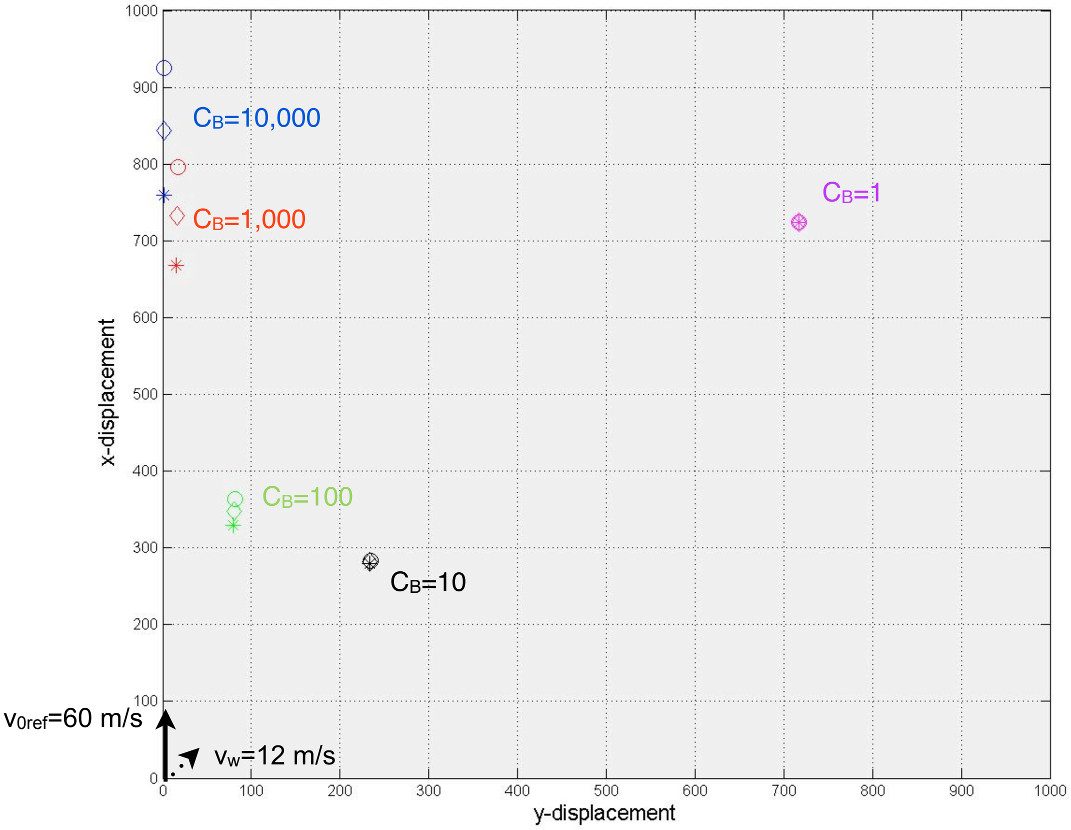


Figure 4.18a - The effect on final wreckage location of changing initial forward velocity for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ✱ *= -10%*

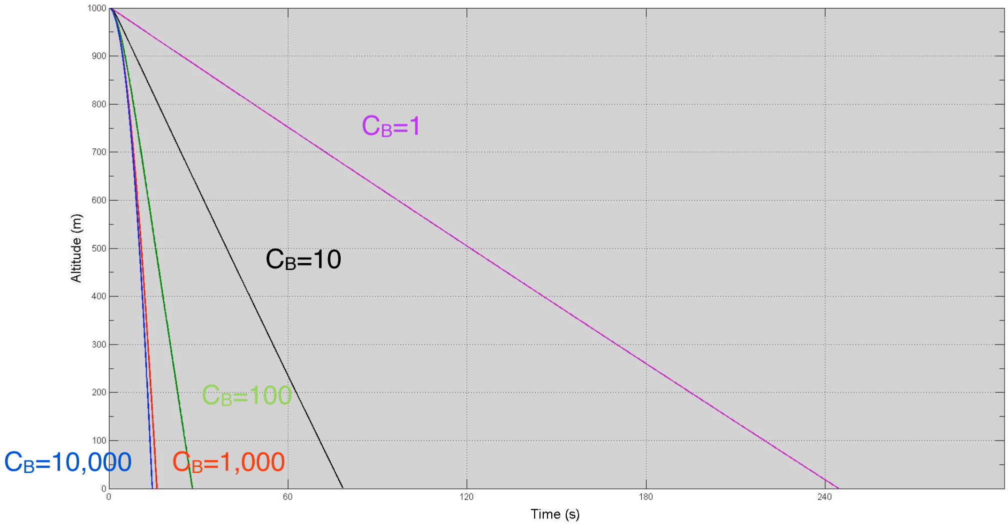


Figure 4.18b - The effect on time to fall to ground of changing initial forward velocity for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—— *= reference,* --- *= +10%,* ˙˙˙˙ *= -10%*

Wind magnitude variation

Figure 4.19a shows the effect of changing wind magnitude for the simulated small aircraft breakup.

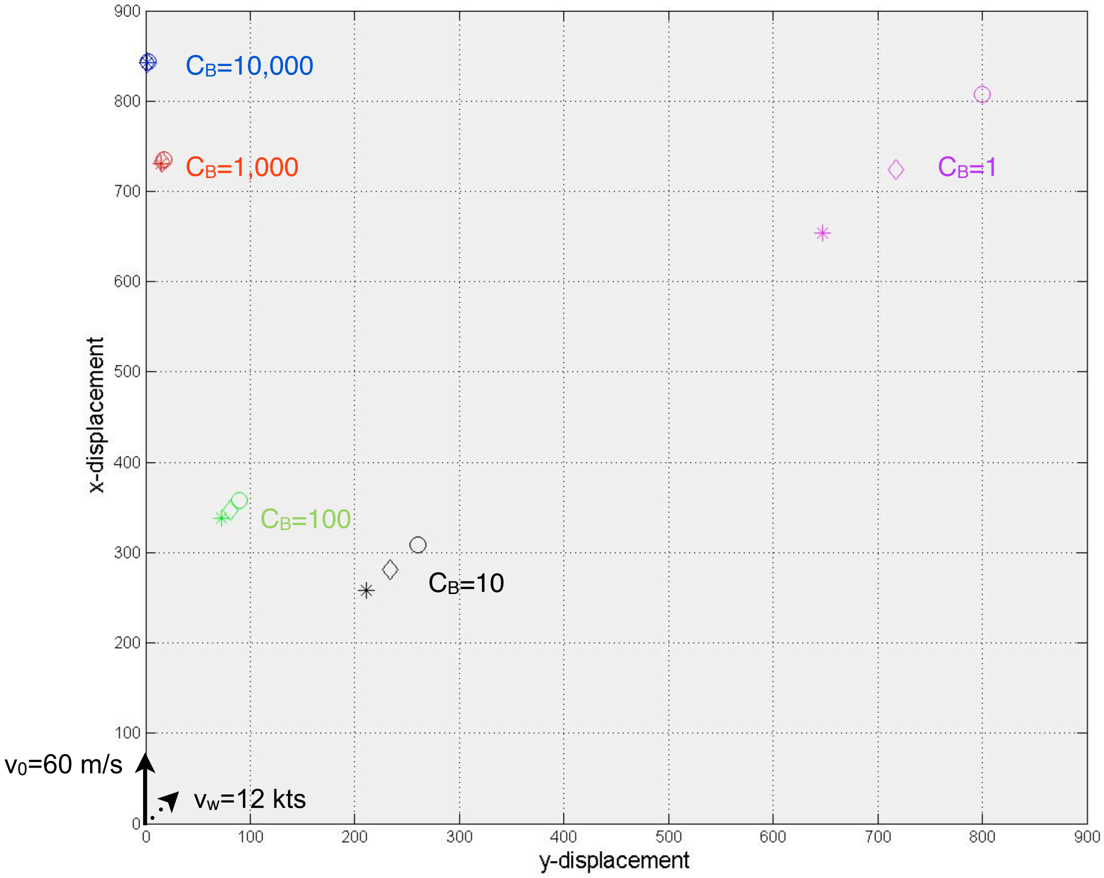


Figure 4.19a - The effect on final wreckage location of changing wind magnitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇= *reference,* O = *+10%,* ✻ = *-10%*

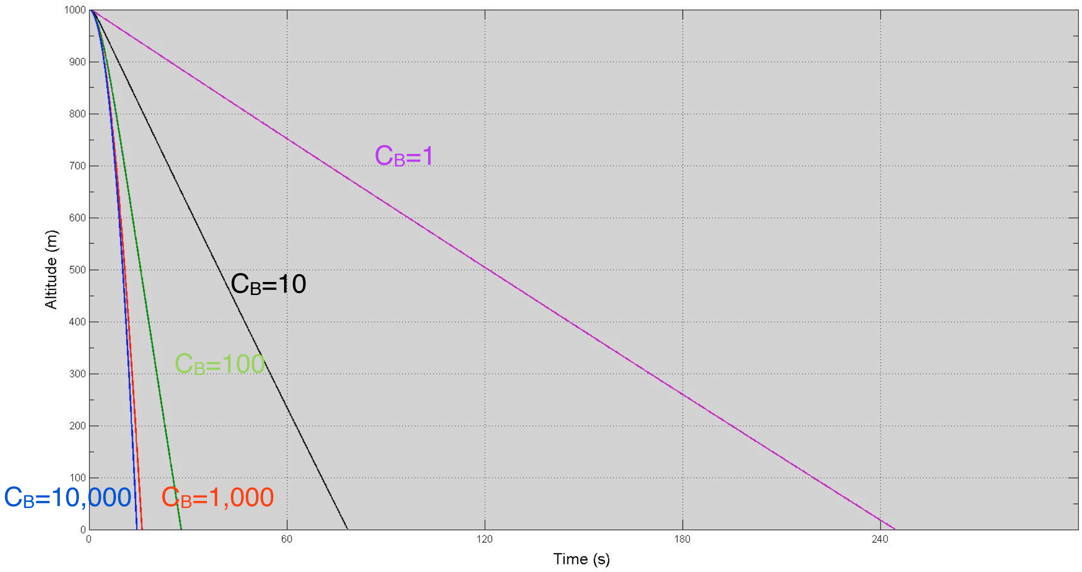


Figure 4.19b - The effect on time to fall to ground of changing wind magnitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

As with the large aircraft breakup, the low C_B items are more greatly affected than the high C_B items which are almost unchanged.

It is interesting to compare this plot to that for altitude variation. The pattern for low C_B items is similar to that seen in Figure 4.17a. However, the high C_B items are differently distributed. Therefore, for an investigator who is attempting to modify a modelled solution which has disagreement between the low C_B items and the model along the windline, it would be instructive to examine the accuracy of the high C_B items. If the fit is good, error in wind magnitude is more likely to be the cause whereas if the fit is poor, altitude inaccuracy may be responsible.

Wind angle variation

Figure 4.20a shows the effect of varying the wind angle by ±9° as discussed with respect to the large aircraft breakup.

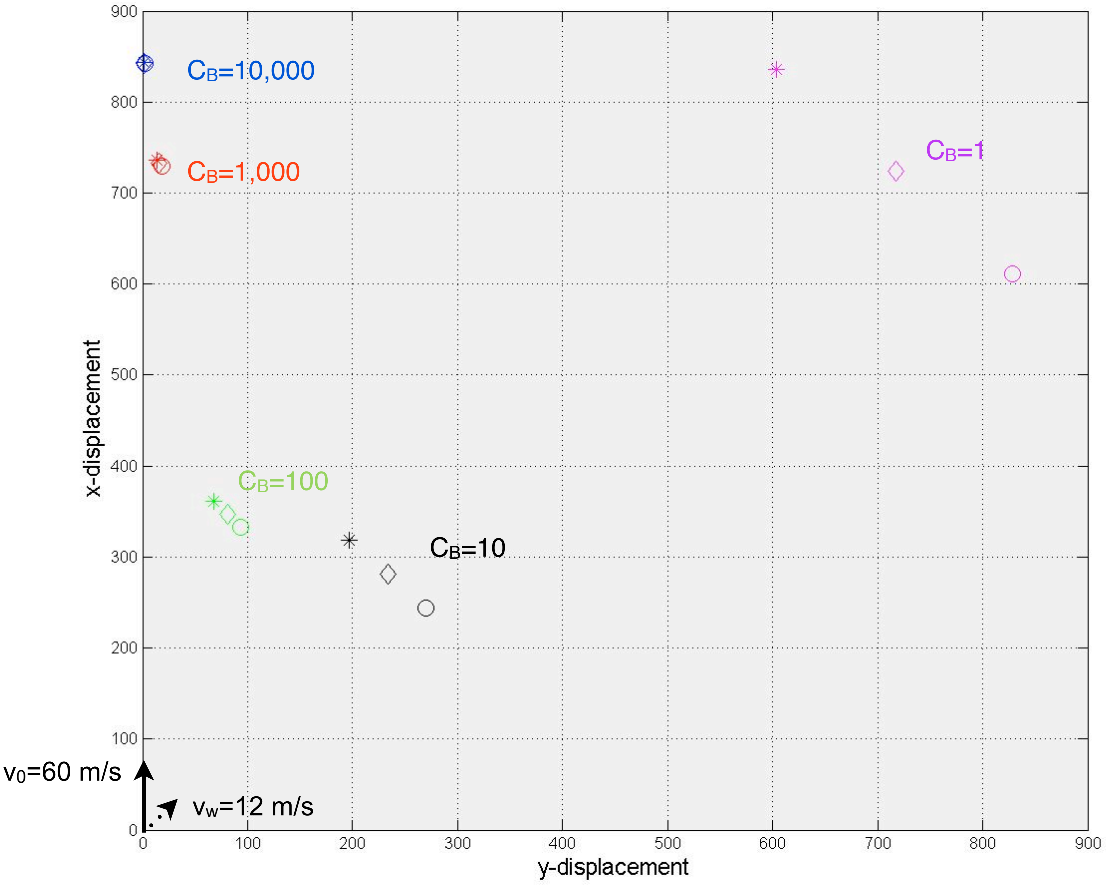


Figure 4.20a - The effect on final wreckage location of changing wind angle for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ✻ *= -10%*

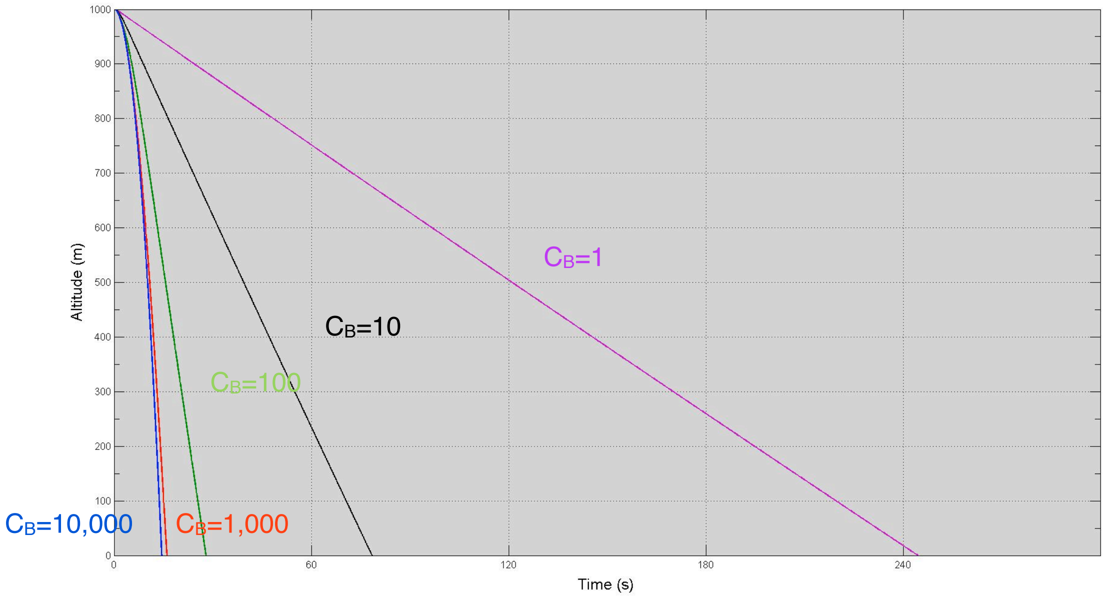


Figure 4.20b - The effect on time to fall to ground of changing wind angle for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ····· = -10%

As with earlier examples, the effect is very similar to that seen for the large aircraft simulation.

Appendices 9 and 10 show identical Figures to those given in 4.13 to 4.20 but with a wind angle of +30˚ (60˚ from aircraft heading) in Appendix 9 and -30˚ (120˚ from aircraft heading) in Appendix 10.

4.6 Difference

Large Aircraft

Figure 4.21 shows the magnitude of the ground location difference for a parameter reduction of 10% for the 5 different values of C_B and Figure 4.22 shows the same for a 10% increase.

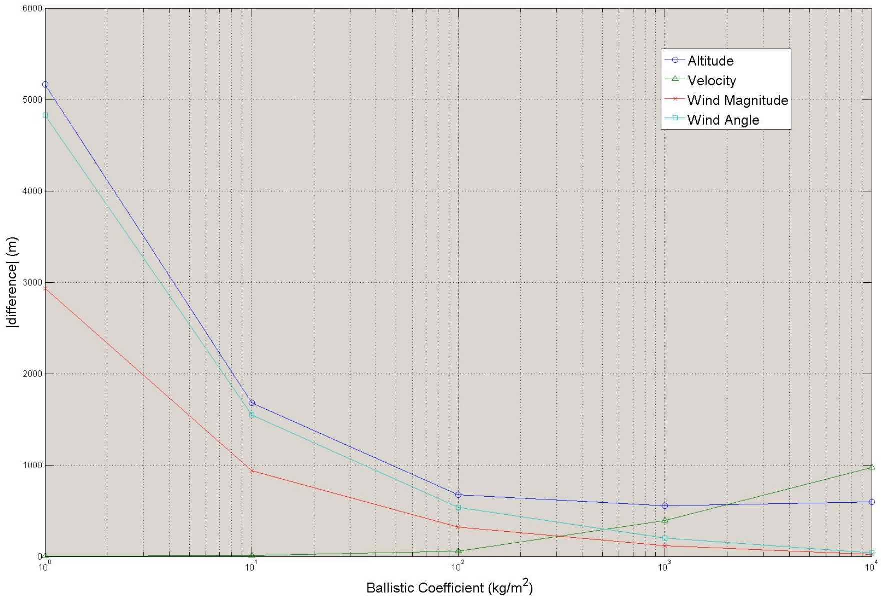


Figure 4.21 - Variation in magnitude of distance from position in reference case for reduction of various parameters and C_B

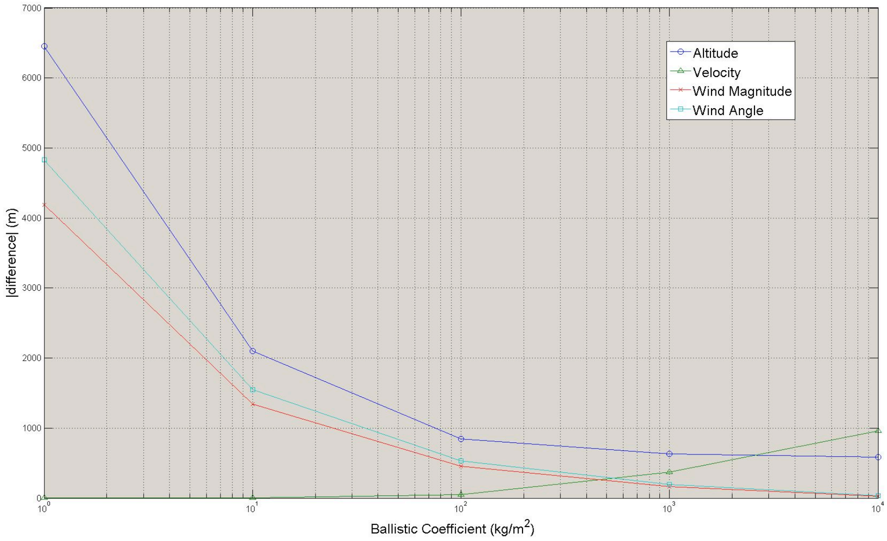


Figure 4.22 - Variation in magnitude of distance from position in reference case for increase of various parameters and C_B

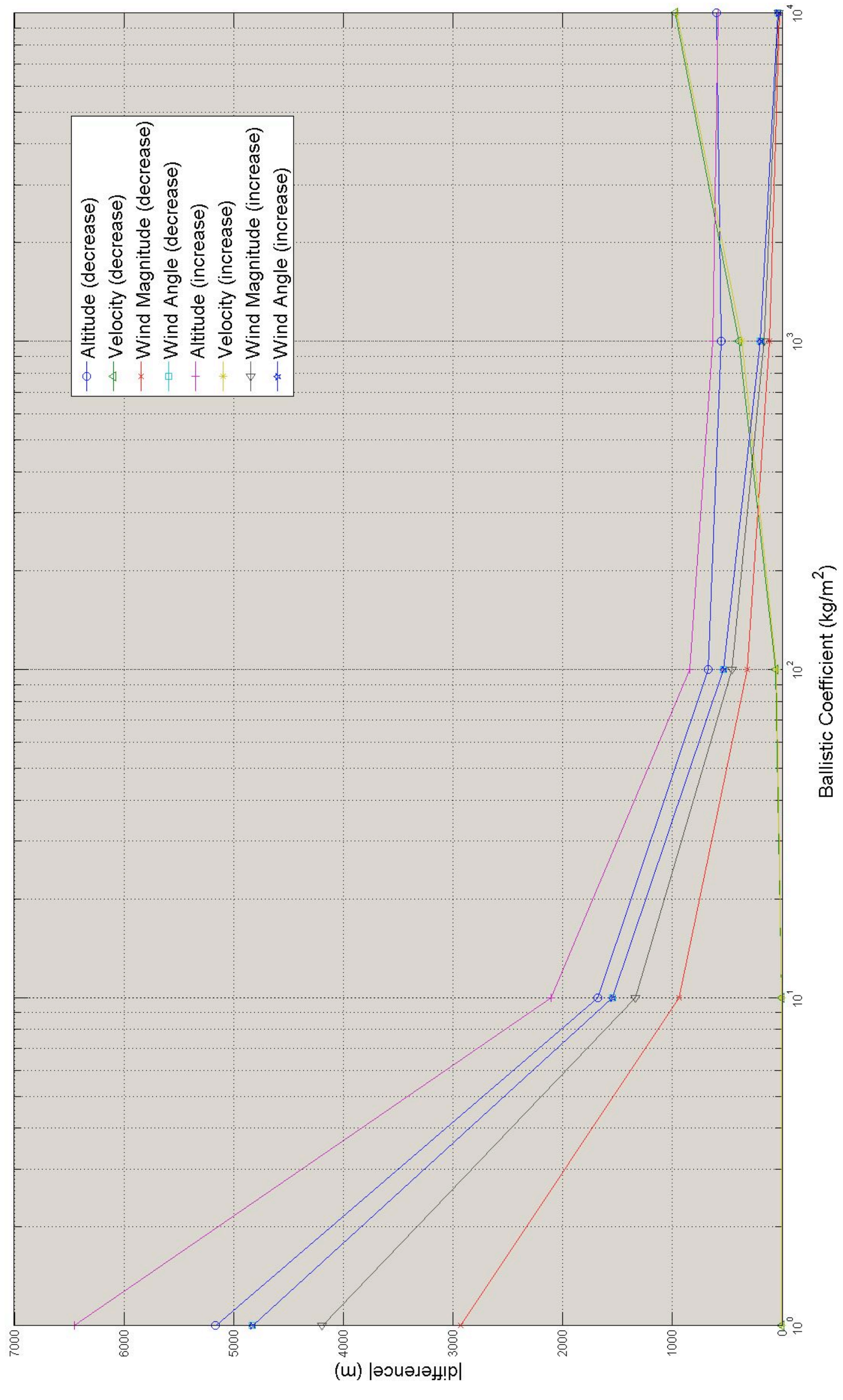


Figure 4.23 - Variation in magnitude of distance from position in reference case for of various parameters and C_B

Figure 4.23 shows the results of Figures 4.21 and 4.22 combined onto one graph. The Figures can be interpreted in a number of ways.

One option is to see which values of C_B are subject to large variation with a certain parameter. For example, in Figure 4.21, both altitude and wind angle have a significant effect on low C_B items with a 10% or 9° reduction producing a difference of around 5,000m compared to less than 1,000m for a high C_B item. This implies that to reduce errors for low C_B items, particular attention should be paid to breakup altitude and wind angle.

Alternatively, the Figure could be examined to establish which level of C_B is least affected by a particular parameter. For example, Figure 4.21 shows that low C_B items are almost independent of initial velocity changes. This implies that any low C_B discrepancies in the model fit cannot be corrected by adjusting initial velocity. Conversely, if using low C_B items to inform the modelling process, initial velocity inaccuracies will be almost completely removed and hence the other three parameters can be studied. Similarly large C_B items will tend to be independent of wind parameters.

Finally, by examining the Figure, the most appropriate 'general' parameter can be assessed. In the case of Figure 4.21 those items with a C_B between 10^2 and 10^3 will be least sensitive to errors in the parameters investigated.

Therefore, it may be most appropriate to base initial modelling estimates on 'medium' C_B values, before using the high and low values for isolating and tuning specific parameters. It is in this process that the 'interactive' approach described earlier may be most powerful. Alternatively, it may be possible to automatically optimise a parameter set through software.

Figure 4.22 showing the variation with increase, displays a very similar shape to that of Figure 4.21 albeit with slightly different amplitudes.

Examining Figure 4.23 shows that for high altitude, high initial velocity breakups, deviations in altitude and wind angle will produce the greatest effect on the ground impact location with a possible errors of more than 6,000m arising from a 10% deviation. It is components with low ballistic coefficients that will be most susceptible to these errors.

<u>*Small Aircraft*</u>

Figure 4.24 shows the magnitude of the ground location difference for a parameter reduction of 10% for the 5 different values of C_B and Figure 4.25 shows the same for a 10% increase.

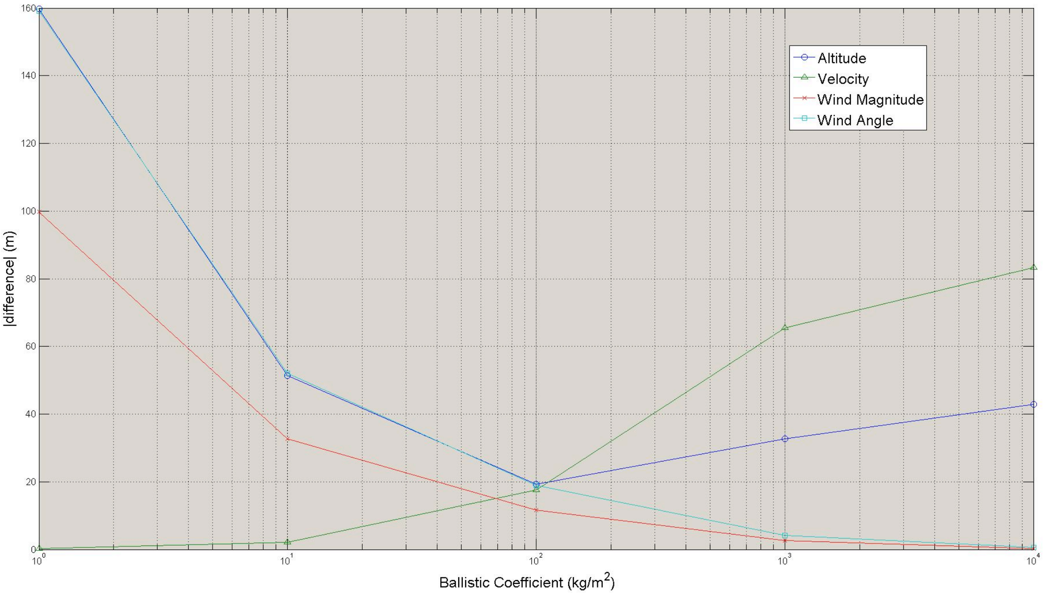


Figure 4.24 - Variation in magnitude of distance from position in reference case for reduction of various parameters and C_B

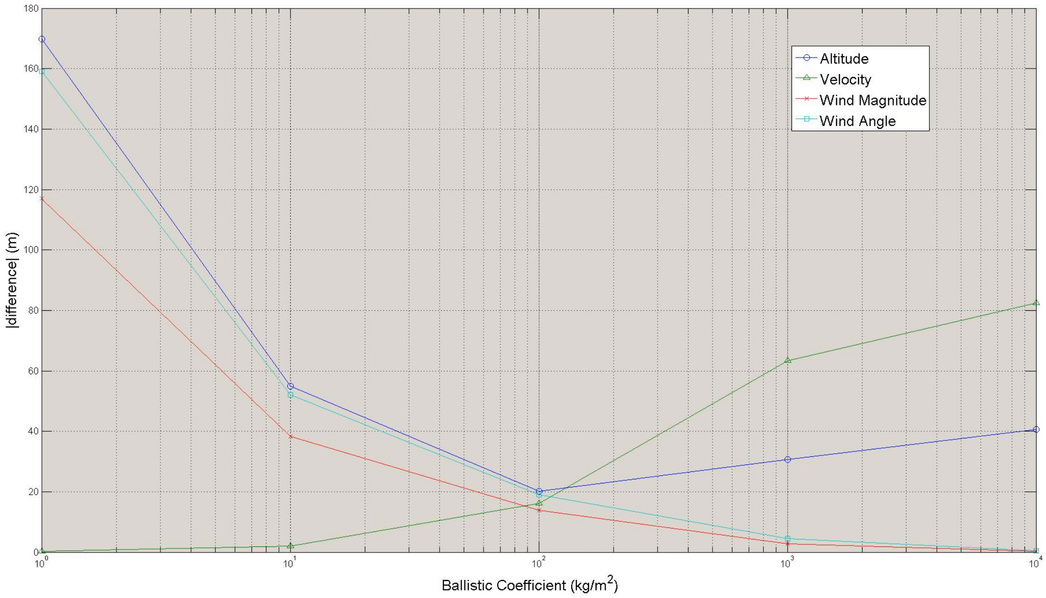


Figure 4.25 - Variation in magnitude of distance from position in reference case for increase of various parameters and C_B

The shape of Figures 4.24 and 4.25 are very similar to those of the large aircraft breakup in Figures 4.21 and 4.22. However, as with the ground location plots the magnitudes differ greatly. Maximum difference from reference are now just over 160m in comparison with 6,000m previously. This is indicative of the lower breakup altitude and shorter time of flight.

As with the large aircraft breakup, the greatest error still arises from deviations in breakup altitude and wind angle for low C_B items. However whereas for the large aircraft breakup a 10% deviation in velocity produced an error of one-sixth (1,000m) of the maximum for a high C_B item, for the small aircraft breakup this value is increased to approximately one-half of the maximum error.

Whereas for the large aircraft breakup two C_B values were appropriate as 'general' items, for the small aircraft breakup, a ballistic coefficient of 100 is least sensitive to deviations in the four parameters studied.

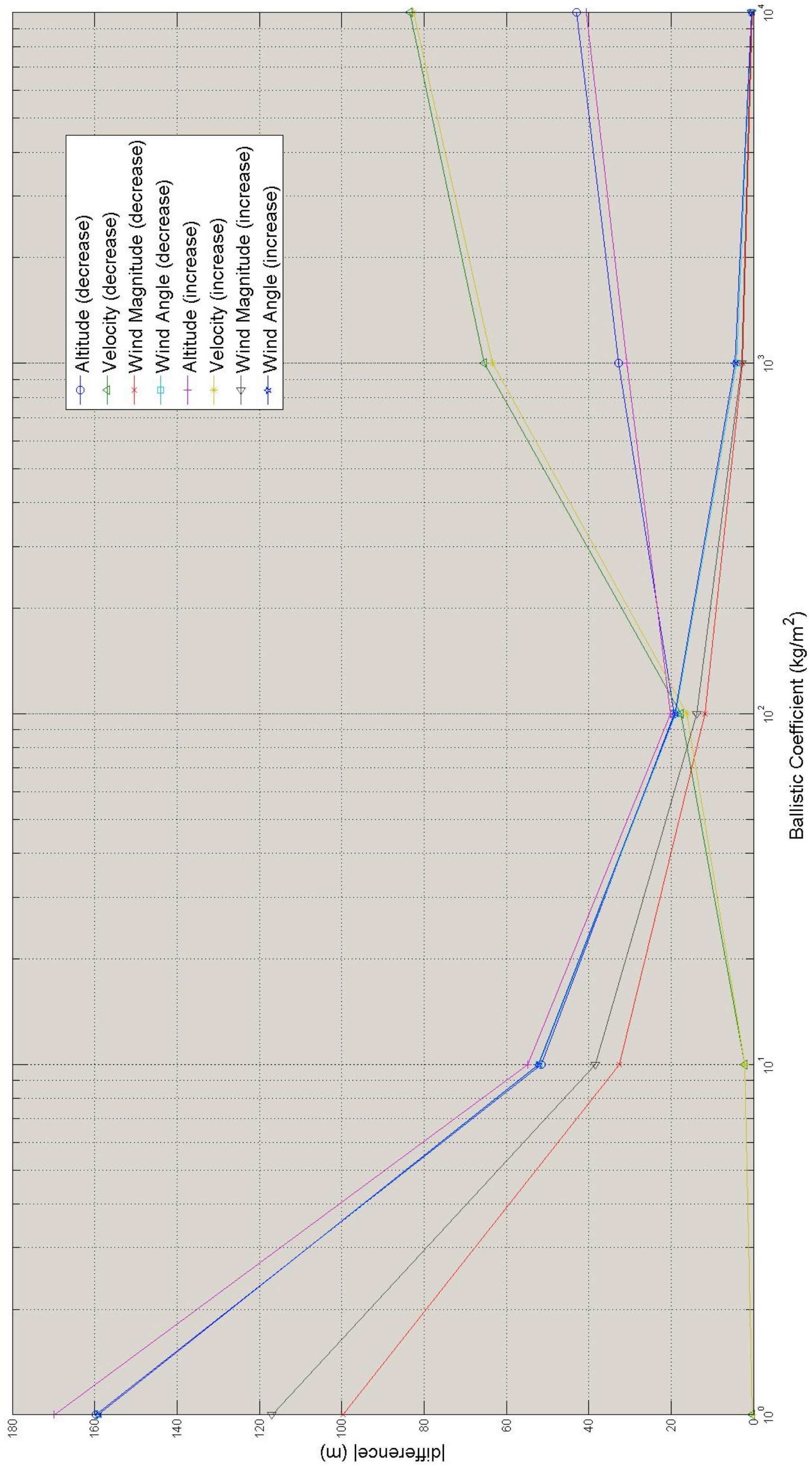


Figure 4.26 - Variation in magnitude of distance from position in reference case for combined of various parameters and C_B

Chapter 5 - Additional Aspects and Further Work

This section outlines some of the issues and possibilities that have arisen through the course of conducting this research. It serves both as a starting point for future research but also a primer for a piece of software which could be developed using the models and techniques outlined in the thesis.

5.1 Drag Estimation

The preceding analysis has relied on the use of ballistic coefficients to analyse possible drag behaviour. This is an excellent approach when dealing with large breakups since there will be a continuous spread of ballistic coefficients marking the entire trail. However, when dealing with small numbers of components, or when trying to locate a component that has departed the aircraft (as was the case with the Sioux City accident) it will be necessary to estimate the drag of a single component. Therefore, this section will examine the feasibility of improved drag estimation techniques for improving the accuracy of trajectory models.

Methods of drag estimation

There are a number of ways of ascertaining the drag coefficient for a specific particle, each with strengths and weaknesses.

One of the greatest difficulties in drag estimation is in predicting the likely orientation of an object when falling. This is affected by factors including the centre of pressure and centre of gravity.

As discussed previously, the possibility of ‘tumbling’ of a component presents real problems in trajectory analysis. Some possible approaches to this issue will be outlined later, but a detailed analysis of this problem is beyond the scope of this thesis. Therefore, it will be assumed that the behaviour of a specific component is governed by a single drag coefficient which is given by the component falling in one specific attitude; the possibility of tumbling will be ignored.

Direct measurement

The most direct way of measuring drag coefficient would be to mount a specific piece of wreckage in a wind tunnel and take measurements to infer the drag coefficient. This would need to be done for a wide range of attitudes in order to estimate the most likely attitude of the component whilst falling. Alternatively, it may be possible to 'gimbal' the part, thereby allowing it to adopt its 'preferred' attitude. However, this process is expensive and time-consuming.

A less direct way to measure drag coefficient is to measure the properties (either in a wind tunnel as above or possibly by drop testing) of a number of representative pieces of wreckage and then use them as reference points for estimating the drag coefficient, usually 'by eye'. However, this is extremely subjective and open to error.

One difficulty that affects every method of drag estimation is that of damage on landing. Even if it is assumed that a component retains the same form from the initial breakup to the final impact, the likelihood of it impacting the ground without any further damage or change in form is extremely small, unless the terminal velocity of the component is extremely small. With the 'direct' measurement techniques outlined above, the only way to find the drag coefficient of the component before impact, is to try to 'repair' the damage caused by impact and then test the component. Clearly this is almost impossible to achieve, even if the damage were fully understood. It would also be extremely time consuming since a large number of possible repairs would have to be tested to determine the sensitivity of the drag coefficient on areas of uncertain damage.

For these reasons, direct measurement may not represent the best way to evaluate drag coefficient and hence an alternative approach is outlined below.

Indirect or virtual measurement

A method of drag estimation is proposed which involves testing the component in a virtual way, rather than performing experiments directly on it. There are two overarching stages to this process:

- digitisation of the component which involves measuring the three-dimensional form of the component and also measuring its weight distribution to establish its centre of gravity, and
- analysis of the digital model which will allow the most likely attitude of descent to be established, and thereby ascertain the most likely drag coefficient and frontal area of the component.

Digitisation of the component

There are two methods which are commonly used for digitising / measuring from components, namely contact and laser measurement.

One of the most common types of measuring system is known as the coordinate measuring machine (CMM). This often consists of an articulated arm or stages which allow the location of a contact probe to be tracked (see Figures 5.1 and 5.2). By touching the probe at various points on the object and recording the location, the three-dimensional coordinates of each point are registered and a model is built up.

CMM systems come in a range of sizes, but a typical articulated arm might have a measuring range of up to 4m with an accuracy of around 50 microns.

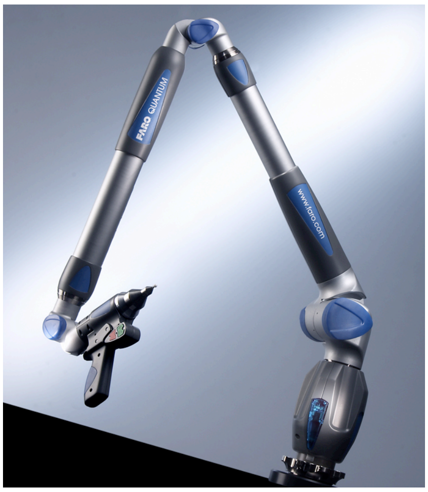


Figure 5.1 - Articulated arm type CMM

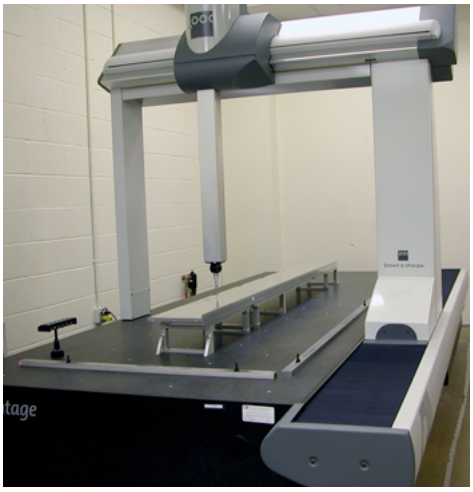


Figure 5.2 - Stage type CMM

A laser-based CMM works in a similar way, but rather than using a contact probe at the end of the articulated arm, a laser head is used instead. The laser head sends out a line of laser light consisting of a number of individual beams forming 'spots' on the surface to be measured. This line of beams is then swept over the surface of the component. By measuring the distance each beam travels to the surface, and knowing the location of the end of the arm and the angle of the head, the surface can be digitised.

An alternative process, often used in digitising larger components, is the use of rotating laser scanners. These scanners, often used for site capture, tend to sit on tripods and send out a single laser beam at a known elevation, and rotational angle (see Figure 5.3). By measuring the distance from the scanner to the component by using the reflected beam, a three-dimensional point can be measured, and by doing this repeatedly at different angles a full three-dimensional set of points can built-up around the scanner.

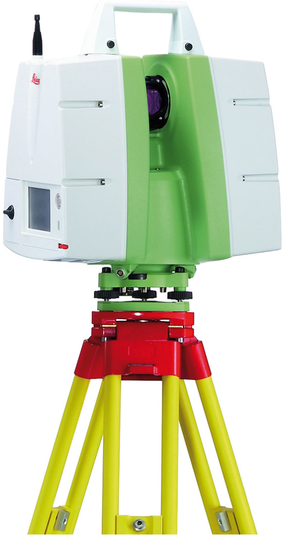
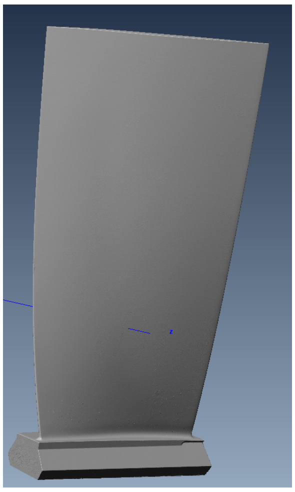


Figure 5.3 - Rotating scanner

Typical scan data sets captured as part of this research are shown in Figures 5.4 and 5.5.



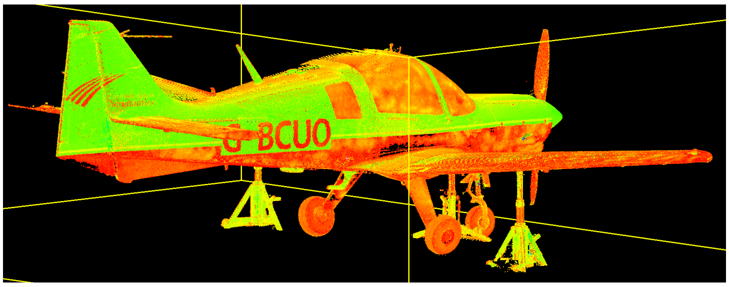


Figure 5.4 - Blade scan

Figure 5.5 - Aircraft scan

Once a set of three-dimensional points is produce (known as a point cloud), it can be edited and manipulated for the required data set. Usually, a number of point clouds will have to be combined since it is necessary either to move the scanner or rotate the part to get access to all sides of the component.

Once a single point cloud describing the component has been produced, it is necessary to join the data points together using polygons to form a surface or mesh. For large smooth surfaces this is a relatively straightforward process, but difficulties arise when dealing with sharp edges and unusual surfaces. Unfortunately these are often present on damaged components.

Some of the skill in surfacing lies in selecting the correct parameters for the surfacing tool. The surfacing program 'knows' nothing about the actual component, all decisions about which points to join are based on the proximity and form of the surrounding points. Therefore, the settings regarding what size of gap to bridge are crucial to creating a successful surface.

In order to establish the centre of gravity it may be possible to estimate the weight distribution of the panels involved. Alternatively, it may be necessary to measure the location of the centre of gravity directly- this can be done through the use of specialised equipment, or more approximately by measuring the 'balance planes' using a knife edge and calculating the intersection.

Challenges

Both of the laser techniques described above rely on the laser beam being reflected from the surface to measure the distance to the object. It follows that objects that do not effectively reflect the laser cannot be measured. In general, the types of surfaces that do not reflect are surfaces which are dark in colour. This may be through material (e.g. black rubber), through paint (e.g. dark blue livery) or by dint of the accident sequence (e.g. sooting). It is possible to spray powder onto surfaces to ensure that the laser is reflected, but covering large areas can be difficult, time-consuming and expensive.

Digital Analysis

Once the digital model is developed, it is straightforward, but time-consuming, task to enter the model into a CFD package, such as ANSYS Fluent, and to derive a graph of drag coefficient against flow angle. By examining the minimum drag condition in parallel with stability considerations, it should be possible to establish a likely drag coefficient, and also estimate the likelihood of the object to tumble (see next section).

5.2 Additional Considerations

Tumbling

It might be possible to modify the drag coefficient of a component by some amount to incorporate the possibility of tumbling. This modification could be scaled by the likelihood of a component to be subject to this type of behaviour which could be extracted from the scan models, or included 'by hand'. However, this approach will still only modify the behaviour of the component in terms of acceleration on a set trajectory; it will take no account of falling such as a stereotypical leaf moving from side to side. Incorporating this type of behaviour will require lift to be included and a form of CFD calculation to be conducted. The potential for errors to be introduced through this process is large, and so it may be more prudent to identify those components which are predisposed to tumbling and simply discount them from the calculation. At the very least, they should be identified to the user so that they can be discounted if necessary.

Mapping

In April 2010, the Ordnance Survey made some of their mapping data freely available as part of the OpenData initiative. This would provide a superb, free, starting point for the implementation of any graphical trajectory application. In addition, the use of a digital terrain map (DTM) would allow the 'flat earth' assumption to be improved upon to incorporate particles striking areas of high ground, rather than continuing 'through' them to the flat earth assumption.

GIS

Since a large portion of any advanced trajectory application would be the accurate graphical representation of data, either with or without mapping data, it might be sensible to incorporate the model as a 'layer' in a GIS system such as esri ArcGIS or erdas IMAGINE. This would remove much of the programming overhead of creating interfaces, displays, and accurately incorporating mapping. Whilst this might limit the interface flexibility and customisation options, it would be a sensible staring point; if total freedom were required it could be introduced once a successful implementation was established.

Uncertainties

As discussed previously, one useful feature of TRAJAN was the inclusion of uncertainties. Whilst this was achieved in a relatively simple way, it is a feature that is well worth retaining. Graphically this could be achieved by using differently-coloured areas to display 'exact' solutions, and then uncertain solutions.

As an extension of this, it would be very useful to be able to display, preferably in real-time, the effect of changing any parameter on the resulting wreckage distribution. This would allow users not only to modify parameters in an attempt to best fit the data, based on their understanding of the situation, but also an insight into the importance of given parameters in a specific situation. This will allow investigators to focus effort on those parameters which may be more important to the outcome, for example, the wind profile may only alter final locations by tens of metres whereas breakup point may modify them by kilometres (see Chapter 4). Clearly in this case, effort spent on the exactly locating the breakup point will yield better results than attempting to obtain a more accurate wind profile.

Validation

A key aspect of the problem will be the thorough validation of the model. Whilst trivial cases (zero wind, straight fall to ground etc.) have confirmed the general operation of the model, in order to prove its utility to the investigation community and to provide confidence to potential users, it will be necessary to validate the model. Clearly, real accident data is the ultimate test for the model, but in addition to these real-world accidents, it may be possible to use published ballistic data (e.g. missile tests) to validate the model. A significant advantage of this form of data would be the lack of tumbling, and accurately known drag coefficients that should be available. For this reason, validation through this route should be adopted before moving on to real-world breakup data.

Appendix 4 gives a list of accidents which may prove suitable for validation. The suitability of a particular accident will depend on a wide range of factors including: data reported, breakup type, privileged access to data, altitude etc. At this stage, all possible candidates are included – later detailed analysis will reveal whether they are suitable.

Inverse calculation

One aspect of trajectory analysis that has received little attention in this thesis is that of inverse or back calculation.

In many accident investigations, investigators are presented with only a wreckage trail and hence only the final location information is available. It is therefore of interest to consider whether the model can provide useful information in such a situation.

This is an example of a boundary value problem where values are specified at more than one point in time. These represent a more complex class of problem than that already presented but one that should still be solvable.

It is tempting to think that final velocity, if it were available either directly or indirectly, might assist the investigator. However, this will only be the case if the object has not hit terminal velocity. If it has reached terminal velocity, there is no way of knowing, without timings, whether it has been falling at that rate for 1 second or 100 seconds.

It is clear that the path of a single particle with a given final position is not unique; it might have been travelling, say, quickly at low altitude or slowly at high altitude. The situation is improved when multiple particles and locations are available. However, some authors (e.g. 6, 7) have suggested that by tracing the trajectory of a number of particles backwards, you can find the common starting point, but this approach assumes exact knowledge of the initial velocities which is unlikely, something acknowledged in 1949 by Owen and Grinstead [8] and in 1956 by Braun [10].

Whilst the solution through a boundary value approach should yield a solution, it may be possible to also achieve a solution through the use of the Monte Carlo technique. This would involve defining a possible set of inputs (e.g. altitudes, speeds etc.) and then running the predictive model for random combinations of those inputs. By seeing where the modelled projectiles land, more can be understood about the system. This technique is particularly useful when considerable uncertainty is present in the system, as may be the case for the investigator.

Chapter 6 - Conclusions

This thesis has described the process of producing a working numerical solver for the problem of ballistic trajectory analysis.

Ballistic trajectory analysis has been key to many large investigations and much of the science is well understood. However, there has been no package that has incorporated variable gravity, variable density and variable wind profiles into a set of differential equations and then solved them in a robust way. The techniques presented in this thesis provide the calculation basis for such a package.

The work has recognised the importance of stiffness in the governing equations and has invoked a suitable solution technique that should remain robust, accurate and efficient for the full range of appropriate inputs. Whilst this result is simple to describe, the process of arriving at it was far more complex. The time taken to achieve the numerical solution, and the space occupied in the thesis, reduced the amount of time and space available to exploit the software. However, useful results were obtained.

The numerical solution was verified against a simplified analytical case and the results for two simulated breakup cases. provide investigators with information regarding the effect on ground location for variations in four significant parameters.

The results indicate that for simulated large aircraft breakups, low ballistic coefficient items are most heavily affected by breakup altitude, wind magnitude and wind angle whereas large ballistic coefficient items are most heavily affected by breakup velocity, although to a much lesser extent (around 15% of the distance of low ballistic coefficient).

For small aircraft breakups, wind angle and breakup altitude have the largest effect on low ballistic coefficient items, with velocity and altitude affecting high ballistic coefficient items to a larger extent (around 50% of low ballistic coefficient items).

Finally, a number of issues and possible solutions emerged during this research which are documented in Chapter 5.

To be useful to investigators, the technique needs to be cheap, both financially and, possibly more importantly, computationally. The engine developed through this research provides that cheap computation, albeit with an 'unfriendly' interface at present. The model should be thoroughly validated, and then it is hoped that this work can form the basis of a trajectory analysis tool which is able to exploit the advances of the last 20 years.

References

1 Air Accidents Investigation Branch (AAIB), 1990. *Report on the accident to Boeing 747-121, N739PA at Lockerbie, Dumfrieshire, Scotland on 21 December,* Aircraft Accident Report 2/90, London: HMSO.

2 National Transportation Safety Bureau (NTSB), 2000. *In-flight Breakup Over The Atlantic Ocean Trans World Airlines Flight 800 Boeing 747-131, N93119 Near East Moriches, New York July 17, 1996*, NTSB/AAR-00/03, Washington D.C.: NTSB.

3 National Transportation Safety Committee (NTSC), 2008. *Boeing 737–4Q8 PK–KKW, Makassar strait, Sulawesi, Republic of Indonesia, 1 January 2007.* KNKT/07.01/08.01.36 Indonesia: NTSC.

4 National Transportation Safety Bureau (NTSB), 1990. *United Airlines Flight 232, McDonnell Douglas DC-10-10, Sioux Gateway Airport, Sioux City, Iowa, July 19, 1989*, NTSB/AAR-90/06, Washington D.C.: NTSB.

5 Anker, R. and Taylor, A. F., *The trajectories of falling parts following in-flight breakup*, ISASI forum magazine, (Proceeding of the 20th International Seminar) Vol 22, No.4 Dec 1989, pp. 133-140

6 Wood, R. H. and Sweginnis, R. W., 2006. *Aircraft Accident Investigation*, 2nd Edition, Casper: Endeavor

7 Royal Aircraft Establishment (RAE), 1946. *Note on estimation of the height and order of disintegration of an aircraft breaking up in flight by calculation of the trajectories of the pieces found in the wreckage trail,* Deptl. Note No. Acc. 146, London: HMSO.

8 Owen, J. B. B. and Grinsted, F., 1949. *The investigation of accidents involving air-frame failure,* Aeronautical Research Council R. & M. No. 2300, London: HMSO.

9 Templin, R. J. and Callan, M. M., 1956. *Generalized trajectory curves for bodies moving in air*, National Aeronautical Establishment, Note 13 (LR-159), Ottawa.

10 Braun, W., 1956. *A Method of Plotting Trajectories of Pieces from Disintegrating Aircraft*, Royal Aircraft Establishment, Accident Note No: Structures 281, London: HMSO.

11 Currie, M. M., 1960. *Generalized Tables for the Calculation of Trajectory Curves for Bodies Moving in the Air,* National Research Council of Canada, National Aeronautical Establishment, Aeronautical Report LR-277, NRC No. 6585, Ottawa.

12 Gwilt, S. R., 1961. *Trajectory Analysis in Aircraft Accident Investigation,* National Research Council of Canada, National Aeronautical Establishment, Aeronautical Report LR-310, Ottawa.

13 Waterfall, A. P., 1966. *A Computer Technique for the Analysis of Three Dimensional Ballistic Trajectory Data Including Aerodynamic Effects*, Royal Aircraft Establishment, Technical Report No. 66163, London: HMSO.

14 Boksenbom, A. S., *Graphical Trajectory Analysis,* NASA Technical Note D-64.

15 Cranfield Aviation Safety Centre, 1993. *Airshow Separation Distances*, Report for Civil Aviation Authority, Cranfield.

16 Oldham Jr., H. E., 1990. *Aircraft Debris Trajectory Analysis*, http://www.proairshow.com/aircraft_debris.htm, accessed 4th Feb 2009. Reproduced in Appendix 5.

17 Bergen-Henengouwen, S. G., 1970. *Wreckage Trajectory Analysis in Aircraft Accident Investigation*, MEng thesis, Carleton University, Ottawa, Ontario.

18 Bergen-Henengouwen, S. G., 1971. *Wreckage Trajectory Analysis in Aircraft Accident Investigation*, Canadian Aeronautics and Space Journal, Nov, pp. 355-361.

19 Matteson, F. H., 1974. *Analysis of Wreckage Patterns from In-Flight Disintegrations*, Journal of Safety Research, Vol 6, No. 2, pp. 60-71.

20 Kepert, J. L., 1976. *The Use of Wreckage Trajectories in Aircraft Accident Investigation,* Defence Science and Technology Organization, Aeronautical Research Laboratories, Structures Note 427, Melbourne.

21 Khan, I. J., 1978. *Study of Trajectories of Aircraft Parts Following In-flight Breakup*, MSc Thesis, Cranfield Institute of Technology.

22 Hull, S. R., 1980. *A 3-D Trajectory Analysis of Aircraft Parts Following Inflight Break Up,* MSc thesis, Cranfield Institute of Technology.

23 Steele, R. M. G., 1983. *Trajectory Plots of Aircraft Debris Following In-flight Breakup*, MSc thesis, Cranfield Institute of Technology.

24 Taylor, A. F., 1991. *The effect of wind on the wreckage trail following in-flight breakup*, Cranfield College of Aeronautics Aerogram, Vol. 6, No. 3.

25 Taylor, A. F. 1998. *The study of aircraft wreckage: the key to aircraft accident investigation*, Technology, Law and Insurance, Vol 3, pp. 129-147.

26 National Transportation Safety Bureau (NTSB), 2000. *In-flight Breakup Over The Atlantic Ocean Trans World Airlines Flight 800 Boeing 747-131, N93119 Near East Moriches, New York July 17, 1996*, Section 22A: Trajectory Study, NTSB/AAR-00/03, Washington D.C.: NTSB.

27 Kirpal, Hon'ble Mr Justice B. N., 1986 *Accident to Air India Boeing 747 Aircraft VT-EFO, "Kanishka" on 23rd June 1985.*

28 Guan, W. and Yong, K., *Ballistic Trajectory Analysis for the CI611 Accident Investigation*, Aviation Safety Council Publication

29 International Civil Aviation Organisation ICAO), 1993. *Manual of the ICAO Standard Atmosphere extended to 80 kilometres (262 500 feet)*, 3rd Edition, Doc 7488/3.

30 Polyanin, A. D., 2004. *Abel Differential Equation of the Second Kind*, from http://eqworld.ipmnet.ru/en/solutions/ode/ode0126.pdf accessed 4th Feb 2009. Reproduced in Appendix 6.

31 Panayotounakos, D. E. and Kravvaritis, D. C., 2006. *Exact analytical solutions of the Abel, Emden-Fowler and generalized Emden-Fowler nonlinear ODEs,* Nonlinear Analysis, Vol. 7, pp. 634-650

32 http://www.virgingalactic.com/assets/downloads/Virgin_Galactic_Brochure.pdf accessed Jan 16th 2012. Relevant section reproduced in Appendix 7.

33 Hoerner, S. F., 1958. Fluid-dynamics Drag,

34 *Springer Handbook of Mechanical Engineering*, Ed. Grote, K-H and Antonsson, E. K., Springer 2009

35 US Joint Military Terminology Group, *Dictionary of Military and Associated Terms*, JCS Pub 1-02, 30th Nov 1989

36 Press et al., 2007. *Numerical Recipes,* 3rd Edition, Cambridge University Press.

37 Lapidus, L. and J. H. Seinfeld, 1971. *Numerical Solution of Ordinary Differential Equations,* Academic Press.p268

38 (http://www.nr.com/webnotes/nr3web24.pdf), accessed 16th Jan 2012. Reproduced in Appendix 8.

39 Canadian Defence Force, Occurrence Investigation Techniques For The Canadian Forces, Document A-GA-135-002/AA-001

40 Hall, G. and J. M. Watt, 1976. *Modern Numerical Methods for Ordinary Differential Equations,* Clarendon Press

Appendices

Appendix 1

Dealing with one-dimensional motion, with drag proportional to v^2 and assuming constant mass and gravity gives the governing differential equation of

$$m\frac{dv}{dt} = mg - kv^2 .$$

Separation of variables gives,

$$\int \frac{dv}{mg - kv^2} = \int \frac{dt}{m} . \qquad (1)$$

To solve, use the definition

$$\int \frac{dx}{a - bx^2} = \frac{\tanh^{-1}\left(\frac{\sqrt{b}x}{\sqrt{a}}\right)}{\sqrt{a}\sqrt{b}} .$$

Therefore, the lhs of (1) becomes,

$$\begin{aligned}\int \frac{dv}{mg - kv^2} &= \frac{\tanh^{-1}\left(\frac{\sqrt{k}}{\sqrt{mg}}v\right)}{\sqrt{mg}\sqrt{k}} , \\ &= \frac{\tanh^{-1}\left(v/v_{TERM}\right)}{\sqrt{mgk}}\end{aligned}$$

where

$$v_{TERM} = \sqrt{\frac{mg}{k}}$$

and (1) becomes

$$\tanh^{-1}\left(v/v_{TERM}\right) = \sqrt{mgk}\left(\frac{t}{m} + c_1\right)$$

where c_1 is a constant of integration. Therefore, the general solution of (1) is,

$$v = v_{TERM} \tanh\left[\sqrt{mgk}\left(\frac{t}{m} + c_1\right)\right]. \qquad (2)$$

which is asymptotic towards , the terminal velocity. To solve the initial value problem,

$$m\frac{dv}{dt} = mg - kv^2 \qquad \text{where} \qquad v\big|_{t=0} = v_0$$

using (2) gives,

$$v_0 = v_{TERM} \tanh\left[c_1\sqrt{mgk}\right]$$

$$c_1 = \frac{1}{\sqrt{mgk}}\tanh^{-1}\left(\frac{v_0}{v_{TERM}}\right)$$

To solve for displacement, note that

$$x(t) = \int v(t)\,dt = v_{TERM}\int \tanh\left[\sqrt{\frac{gk}{m}}t + c_1\sqrt{mgk}\right]dt. \quad (3)$$

Using the definition

$$\int \tanh(ax+b) = \frac{\ln(\cosh(ax+b)}{a}.$$

gives,

$$x(t) = \frac{v_{TERM}}{\sqrt{gk/m}}\ln\left[\cosh\left(\sqrt{\frac{gk}{m}}t + c_1\sqrt{mgk}\right)\right] + c_2 \qquad (4)$$

where c_2 is a constant of integration. Again, solving for the initial value,

$$x\big|_{t=0} = x_0$$

(4) gives

$$x_0 = \frac{v_{TERM}}{\sqrt{gk/m}}\ln\left[\cosh\left(c_1\sqrt{mgk}\right)\right] + c_2$$

and hence,

$$c_2 = x_0 - \frac{v_{TERM}}{\sqrt{gk/m}} \ln\left[\cosh\left(c_1\sqrt{mgk}\right)\right].$$

Therefore, the governing equations for velocity and displacement for this type of motion are

$$v = v_{TERM} \tanh\left[\sqrt{\frac{gk}{m}}t + \tanh^{-1}\left(\frac{v_0}{v_{TERM}}\right)\right]$$

and

$$x(t) = \frac{v_{TERM}}{\sqrt{gk/m}} \ln\left[\cosh\left(\sqrt{\frac{gk}{m}}t + \tanh^{-1}\left(\frac{v_0}{v_{TERM}}\right)\right)\right] + c_2$$

where

$$c_2 = x_0 - \frac{v_{TERM}}{\sqrt{gk/m}} \ln\left[\cosh\left(\tanh^{-1}\left(\frac{v_0}{v_{TERM}}\right)\right)\right]$$

where

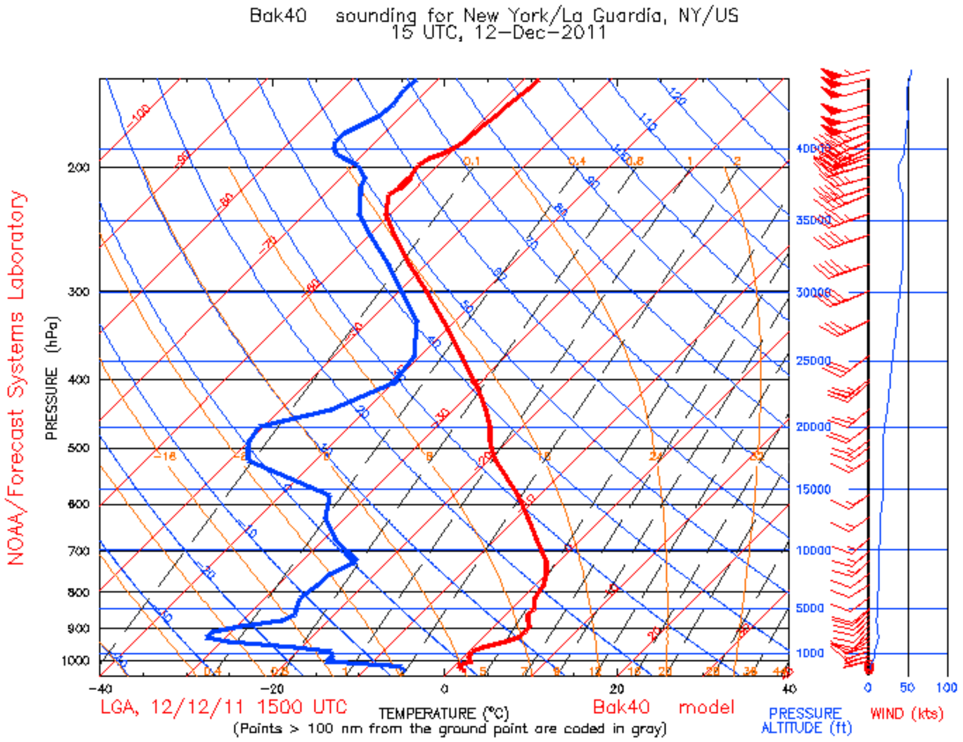
m is the object mass

g is gravity

k is the drag proportionality constant

t is time

Appendix 2



Sounding from the Bak40 analysis
data is for nearest Bak40 grid point, 9.1 nm / 308 deg from LGA
Sounding for: -3 h | -2 h | previous hour | next hour | +2 h | +3 h
FAA 604 format for SHARP <- ?

```
             (DD = 'A' means analysis)
Pressure_Alt DD Dir Spd  ---Temp--- DewPt
 (ft)   (mb)        (kts)  (F)   (C)   (C)

   124 1031.0 A 234   1  34.9   1.6  -5.4
   193 1028.0 A 252   1  34.2   1.2  -5.7
   328 1023.0 A 263   1  33.6   0.9  -6.0
   534 1015.0 A 289   2  32.5   0.3  -7.5
   810 1004.0 A 253   6  34.7   1.5 -15.4
   935 1000.0 A 252   6  34.2   1.2 -15.3
  1233  988.0 A 251   7  33.1   0.6 -15.2
  1660  972.0 A 247   8  32.5   0.3 -15.9
  2093  956.0 A 248   9  33.8   1.0 -20.8
  2536  941.0 A 235  11  37.6   3.1 -29.1
  2992  925.0 A 226  12  39.0   3.9 -32.2
  2992  925.0 A 226  12  39.0   3.9 -32.2
  3451  909.0 A 215  11  39.4   4.1 -32.1
  3923  893.0 A 215  10  39.2   4.0 -29.4
  4402  877.0 A 217   9  37.9   3.3 -25.3
  4888  861.0 A 217  10  36.5   2.5 -24.6
  5239  850.0 A 220  10  36.5   2.5 -24.9
  5403  844.0 A 222  10  36.5   2.5 -25.1
  6276  817.0 A 227  11  34.7   1.5 -25.7
  7417   782.0  A  229   12   33.6    0.9 -25.3
  8372   754.0  A  226   12   32.0    0.0 -25.1
  9452   724.0  A  220   13   29.5   -1.4 -23.5
 10334   700.0  A  221   13   26.2   -3.2 -25.8
 11381   672.0  A  222   14   22.5   -5.3 -28.6
 13116   628.0  A  228   15   16.3   -8.7 -32.1
 14954   584.0  A  230   17    9.7 -12.4 -34.3
 17785   520.0  A  231   17   -2.4 -19.1 -47.8
 18772   500.0  A  226   17   -5.8 -21.0 -49.2
 19317   488.0  A  224   17   -7.6 -22.0 -50.0
 20439   466.0  A  226   19 -10.7 -23.7 -50.4
 21811   440.0  A  222   23 -15.2 -26.2 -44.2
 23740   405.0  A  222   27 -22.7 -30.4 -39.9
 24071   400.0  A  223   27 -24.0 -31.1 -40.0
 25830   370.0  A  226   30 -31.2 -35.1 -40.7
 28385   331.0  A  239   36 -42.2 -41.2 -44.4
 30524   300.0  A  243   40 -52.2 -46.8 -49.7
 32388   275.0  A  247   43 -61.1 -51.7 -54.3
 34402   250.0  A  246   43 -70.4 -56.9 -59.8
 35790   233.0  A  245   43 -76.9 -60.5 -63.5
 37057   219.0  A  243   42 -80.1 -62.3 -65.6
 37562   214.0  A  247   39 -79.2 -61.8 -66.5
 38231   207.0  A  250   37 -79.4 -61.9 -67.1
 39005   199.0  A  251   38 -80.1 -62.3 -69.4
 39494   194.0  A  247   42 -79.4 -61.9 -71.8
 39852   191.0  A  242   44 -77.8 -61.0 -73.2
 40219   188.0  A  239   45 -76.5 -60.3 -74.1
 40662   184.0  A  241   46 -75.5 -59.7 -74.7
 41207   179.0  A  243   47 -75.1 -59.5 -74.6
 41781   174.0  A  247   49 -74.9 -59.4 -73.6
 42342   169.0  A  248   50 -74.6 -59.2 -72.7
 43038   163.0  A  250   49 -74.0 -58.9 -72.3
 44061   155.0  A  251   49 -73.1 -58.4 -72.8
 44862   150.0  A  252   51 -72.9 -58.3 -72.6
 45383   146.0  A  252   53 -72.9 -58.3 -72.4
 48195   127.0  A  255   55 -74.7 -59.3 -73.9
 51778   107.0  A  263   53 -79.2 -61.8 -81.2
 53152   100.0  A  264   52 -80.1 -62.3 -81.6
 56122    86.0  A  267   51 -82.3 -63.5 -82.5
 60318    70.0  A  269   53 -86.4 -65.8 -84.4
 61673    65.0  A  269   53 -87.9 -66.6 -85.0
 66837    50.0  A  263   46 -98.3 -72.4 -89.5
 73103    36.0  A  255   37 ***** -79.5 -94.9
```

Sounding for: -3 h | -2 h | previous hour | next hour | +2 h | +3 h

Please send questions about this page to one or more of the following people.

- ***Susan Sahm for technical questions about the java display (such as a blank screen)***
- ***Brian Jamison for questions about soundings and the SkewT display in general, and about RAOB soundings***
- ***John Brown for questions about model soundings (types of models, available forecasts, latency, etc.)***

Last modified: Mon Apr 4 21:13:00 GMT 2011

Appendix 3

Although only the 4-state system is required to calculate the eigenvalues (since the *x* and *y* displacements do not feature in the governing equations), the full 6-state system Jacobian will be solved here, since it will be required for the numerical integration scheme. The coupled equations for the 6-state system are:

$$\frac{dv_x^G}{dt} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(v_x^G-v_x^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$\frac{dv_y^G}{dt} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(v_y^G-v_y^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$\frac{dv_z^G}{dt} = -g_0\left(\frac{r_e}{r_e+z}\right)^2-\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(v_z^G-v_z^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$\frac{dx}{dt} = v_x^G$$

$$\frac{dy}{dt} = v_y^G$$

$$\frac{dz}{dt} = v_z^G$$

which gives the Jacobian as:

$$\mathbf{J}=\frac{\partial\mathbf{f}}{\partial\mathbf{y}}=\begin{bmatrix} \frac{\partial^2 v_x^G}{\partial t\,\partial v_x^G} & \frac{\partial^2 v_x^G}{\partial t\,\partial v_y^G} & \frac{\partial^2 v_x^G}{\partial t\,\partial v_z^G} & \frac{\partial^2 v_x^G}{\partial t\,\partial x} & \frac{\partial^2 v_x^G}{\partial t\,\partial y} & \frac{\partial^2 v_x^G}{\partial t\,\partial z} \\ \frac{\partial^2 v_y^G}{\partial t\,\partial v_x^G} & \frac{\partial^2 v_y^G}{\partial t\,\partial v_y^G} & \frac{\partial^2 v_y^G}{\partial t\,\partial v_z^G} & \frac{\partial^2 v_y^G}{\partial t\,\partial x} & \frac{\partial^2 v_y^G}{\partial t\,\partial y} & \frac{\partial^2 v_y^G}{\partial t\,\partial z} \\ \frac{\partial^2 v_z^G}{\partial t\,\partial v_x^G} & \frac{\partial^2 v_z^G}{\partial t\,\partial v_y^G} & \frac{\partial^2 v_z^G}{\partial t\,\partial v_z^G} & \frac{\partial^2 v_z^G}{\partial t\,\partial x} & \frac{\partial^2 v_z^G}{\partial t\,\partial y} & \frac{\partial^2 v_z^G}{\partial t\,\partial z} \\ \frac{\partial^2 x}{\partial t\,\partial v_x^G} & \frac{\partial^2 x}{\partial t\,\partial v_y^G} & \frac{\partial^2 x}{\partial t\,\partial v_z^G} & \frac{\partial^2 x}{\partial t\,\partial x} & \frac{\partial^2 x}{\partial t\,\partial y} & \frac{\partial^2 x}{\partial t\,\partial z} \\ \frac{\partial^2 y}{\partial t\,\partial v_x^G} & \frac{\partial^2 y}{\partial t\,\partial v_y^G} & \frac{\partial^2 y}{\partial t\,\partial v_z^G} & \frac{\partial^2 y}{\partial t\,\partial x} & \frac{\partial^2 y}{\partial t\,\partial y} & \frac{\partial^2 y}{\partial t\,\partial z} \\ \frac{\partial^2 z}{\partial t\,\partial v_x^G} & \frac{\partial^2 z}{\partial t\,\partial v_y^G} & \frac{\partial^2 z}{\partial t\,\partial v_z^G} & \frac{\partial^2 z}{\partial t\,\partial x} & \frac{\partial^2 z}{\partial t\,\partial y} & \frac{\partial^2 z}{\partial t\,\partial z} \end{bmatrix}$$

Due to the lack of coupling mentioned above, some of these expressions are trivial and can be evaluated by inspection, specifically (with some repetition to allow rows and columns to be followed):

$$\frac{\partial^2 v_x^G}{\partial t\,\partial x}=\frac{\partial^2 v_y^G}{\partial t\,\partial x}=\frac{\partial^2 v_z^G}{\partial t\,\partial x}=\frac{\partial^2 x}{\partial t\,\partial x}=\frac{\partial^2 y}{\partial t\,\partial x}=\frac{\partial^2 z}{\partial t\,\partial x}=0$$

$$\frac{\partial^2 v_x^G}{\partial t\,\partial y}=\frac{\partial^2 v_y^G}{\partial t\,\partial y}=\frac{\partial^2 v_z^G}{\partial t\,\partial y}=\frac{\partial^2 x}{\partial t\,\partial y}=\frac{\partial^2 y}{\partial t\,\partial y}=\frac{\partial^2 z}{\partial t\,\partial y}=0$$

$$\frac{\partial^2 x}{\partial t\,\partial v_x^G}=\frac{\partial^2 y}{\partial t\,\partial v_y^G}=\frac{\partial^2 z}{\partial t\,\partial v_z^G}=1$$

$$\frac{\partial^2 x}{\partial t\,\partial v_y^G}=\frac{\partial^2 x}{\partial t\,\partial v_z^G}=\frac{\partial^2 x}{\partial t\,\partial x}=\frac{\partial^2 x}{\partial t\,\partial y}=\frac{\partial^2 x}{\partial t\,\partial z}=0$$

$$\frac{\partial^2 y}{\partial t\,\partial v_x^G}=\frac{\partial^2 y}{\partial t\,\partial v_z^G}=\frac{\partial^2 y}{\partial t\,\partial x}=\frac{\partial^2 y}{\partial t\,\partial y}=\frac{\partial^2 y}{\partial t\,\partial z}=0$$

$$\frac{\partial^2 z}{\partial t\,\partial v_x^G}=\frac{\partial^2 z}{\partial t\,\partial v_y^G}=\frac{\partial^2 z}{\partial t\,\partial x}=\frac{\partial^2 z}{\partial t\,\partial y}=\frac{\partial^2 z}{\partial t\,\partial z}=0$$

which reduces the Jacobian to,

$$\frac{\partial \mathbf{f}}{\partial \mathbf{y}}=\begin{bmatrix} \frac{\partial^2 v_x^G}{\partial t\,\partial v_x^G} & \frac{\partial^2 v_x^G}{\partial t\,\partial v_y^G} & \frac{\partial^2 v_x^G}{\partial t\,\partial v_z^G} & 0 & 0 & \frac{\partial^2 v_x^G}{\partial t\,\partial z} \\ \frac{\partial^2 v_y^G}{\partial t\,\partial v_x^G} & \frac{\partial^2 v_y^G}{\partial t\,\partial v_y^G} & \frac{\partial^2 v_y^G}{\partial t\,\partial v_z^G} & 0 & 0 & \frac{\partial^2 v_y^G}{\partial t\,\partial z} \\ \frac{\partial^2 v_z^G}{\partial t\,\partial v_x^G} & \frac{\partial^2 v_z^G}{\partial t\,\partial v_y^G} & \frac{\partial^2 v_z^G}{\partial t\,\partial v_z^G} & 0 & 0 & \frac{\partial^2 v_z^G}{\partial t\,\partial z} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The remaining 12 expressions must be calculated by hand. This is detailed below. Indices are [row,column].

$$\begin{aligned}
\frac{\partial^2 v_x^G}{\partial t\,\partial v_x^G} &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\frac{\partial}{\partial v_y^G}\left(v_x^G-v_x^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2} \\
&= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\frac{\partial}{\partial v_y^G}A.B \\
&= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(A\frac{\partial B}{\partial v_y^G}+B\frac{\partial A}{\partial v_y^G}\right) \\
&= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\begin{pmatrix}\left(v_x^G-v_x^W\right)\frac{1}{2}\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)\right]^{-\frac{1}{2}}2\left(v_x^G-v_x^W\right)+ \\ \sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\end{pmatrix} \\
&= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\begin{pmatrix}\left(v_x^G-v_x^W\right)^2\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)\right]^{-\frac{1}{2}}+ \\ \sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\end{pmatrix}=[1,1]
\end{aligned}$$

Similarly,

$$\frac{\partial^2 v_y^G}{\partial t\,\partial v_y^G}=-\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\begin{pmatrix}\left(v_y^G-v_y^W\right)^2\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)\right]^{-\frac{1}{2}}+ \\ \sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\end{pmatrix}=[2,2]$$

and

$$\frac{\partial^2 v_z^G}{\partial t\,\partial v_z^G}=-\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\begin{pmatrix}\left(v_z^G-v_z^W\right)^2\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)\right]^{-\frac{1}{2}}+ \\ \sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\end{pmatrix}=[3,3]$$

$$\frac{\partial^2 v_y^G}{\partial t\,\partial v_x^G} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_y^G-v_y^W\right)\frac{\partial}{\partial v_x^G}\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_y^G-v_y^W\right)\left(v_x^G-v_x^W\right)\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right]^{-\frac{1}{2}} = [2,1]$$

$$\frac{\partial^2 v_z^G}{\partial t\,\partial v_x^G} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r\ +z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_z^G-v_z^W\right)\frac{\partial}{\partial v_x^G}\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_z^G-v_z^W\right)\left(v_x^G-v_x^W\right)\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right]^{-\frac{1}{2}} = [3,1]$$

$$\frac{\partial^2 v_z^G}{\partial t\,\partial v_y^G} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_z^G-v_z^W\right)\frac{\partial}{\partial v_y^G}\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_z^G-v_z^W\right)\left(v_y^G-v_y^W\right)\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right]^{-\frac{1}{2}} = [3,2]$$

$$\frac{\partial^2 v_x^G}{\partial t\,\partial v_y^G} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_x^G-v_x^W\right)\frac{\partial}{\partial v_y^G}\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_x^G-v_x^W\right)\left(v_y^G-v_y^W\right)\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right]^{-\frac{1}{2}} = [1,2]$$

$$\frac{\partial^2 v_x^G}{\partial t\,\partial v_z^G} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_x^G-v_x^W\right)\frac{\partial}{\partial v_z^G}\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_x^G-v_x^W\right)\left(v_z^G-v_z^W\right)\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right]^{-\frac{1}{2}} = [1,3]$$

$$\frac{\partial^2 v_y^G}{\partial t\,\partial v_z^G} = -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{rz}{r+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_y^G-v_y^W\right)\frac{\partial}{\partial v_z^G}\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}$$

$$= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left[1+\frac{\beta}{T_b}\left(\frac{rz}{r+z}-H_b\right)\right]^{-\left(1+g_0/\beta R\right)}\left(v_y^G-v_y^W\right)\left(v_z^G-v_z^W\right)\left[\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right]^{-\frac{1}{2}} = [2,3]$$

$$\begin{aligned}\frac{\partial^2 v_x^G}{\partial t\partial z} &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(v_x^G-v_x^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\\ &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}\left[\frac{T_b-\beta H_b}{T_b}+\frac{\beta r_e}{T_b}\frac{z}{r_e+z}\right]^{-(1+g_0/\beta R)}\left(v_x^G-v_x^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\\ &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}A.B_x.C\\ &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left(B_xC\frac{\partial A}{\partial z}+AC\frac{\partial B_x}{\partial z}+AB_x\frac{\partial C}{\partial z}\right)=[1,6]\end{aligned}$$

where

$$\begin{aligned}A &= \left[\frac{T_b-\beta H_b}{T_b}+\frac{\beta r_e}{T_b}\frac{z}{r_e+z}\right]^{-(1+g_0/\beta R)}\\ B_x &= \left(v_x^G-a_x-b_xz-c_xz^2-d_xz^3\right)\\ C &= \sqrt{\left(v_x^G-a_x-b_xz-c_xz^2-d_xz^3\right)^2+\left(v_y^G-a_y-b_yz-c_yz^2-d_yz^3\right)^2+\left(v_z^G-a_z-b_zz-c_zz^2-d_zz^3\right)^2}\end{aligned}$$

$$\begin{aligned}\frac{\partial A}{\partial z} &= -\left(1+g_0/\beta R\right)\left[\frac{T_b-\beta H_b}{T_b}+\frac{\beta r_e}{T_b}\frac{z}{r_e+z}\right]^{-(2+g_0/\beta R)}\frac{\partial}{\partial z}\left[\frac{T_b-\beta H_b}{T_b}+\frac{\beta r_e}{T_b}\frac{z}{r_e+z}\right]\\ &= -\left(1+g_0/\beta R\right)\left[\frac{T_b-\beta H_b}{T_b}+\frac{\beta r_e}{T_b}\frac{z}{r_e+z}\right]^{-(2+g_0/\beta R)}\frac{\beta r_e}{T_b}\frac{\left(r_e+z\right).1-z.1}{\left(r_e+z\right)^2}\end{aligned}$$

$$\begin{aligned}\frac{\partial B_x}{\partial z} &= \frac{\partial}{\partial z}\left(v_x^G-a_x-b_xz-c_xz^2-d_xz^3\right)\\ &= -b_x-2c_xz-3d_xz^2\end{aligned}$$

$$\begin{aligned}\frac{\partial C}{\partial z} &= \begin{Bmatrix}\left(v_x^G-a_x-b_xz-c_xz^2-d_xz^3\right)^2+\\ \left(v_y^G-a_y-b_yz-c_yz^2-d_yz^3\right)^2+\\ \left(v_z^G-a_z-b_zz-c_zz^2-d_zz^3\right)^2\end{Bmatrix}^{\frac{1}{2}}\\ &= \frac{1}{2}\left\{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right\}^{-\frac{1}{2}}\frac{\partial}{\partial z}\begin{Bmatrix}\left(v_x^G-a_x-b_xz-c_xz^2-d_xz^3\right)^2+\\ \left(v_y^G-a_y-b_yz-c_yz^2-d_yz^3\right)^2+\\ \left(v_z^G-a_z-b_zz-c_zz^2-d_zz^3\right)^2\end{Bmatrix}\\ &= \frac{1}{2}\left\{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2\right\}^{-\frac{1}{2}}\begin{Bmatrix}2\left(v_x^G-v_x^W\right)\left(-b_x-2c_xz-3d_xz^2\right)+\\ 2\left(v_y^G-v_y^W\right)\left(-b_y-2c_yz-3d_yz^2\right)+\\ 2\left(v_z^G-v_z^W\right)\left(-b_z-2c_zz-3d_zz^2\right)\end{Bmatrix}\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 v_y^G}{\partial t \partial z} &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(v_y^G-v_y^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\\ &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}\left[\frac{T_b-\beta H_b}{T_b}+\frac{\beta r_e}{T_b}\frac{z}{r_e+z}\right]^{-(1+g_0/\beta R)}\left(v_y^G-v_y^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\\ &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}A.B_y.C\\ &= -\frac{1}{2C_B}\frac{p_b}{RT_b}\left(B_y C\frac{\partial A}{\partial z}+AC\frac{\partial B_y}{\partial z}+AB_y\frac{\partial C}{\partial z}\right)=[2,6]\end{aligned}$$

with A , C , $\dfrac{\partial A}{\partial z}$ and $\dfrac{\partial C}{\partial z}$ as above, $B_y=\left(v_y^G-a_y-b_y z-c_y z^2-d_y z^3\right)$ and

$$\begin{aligned}\frac{\partial B_y}{\partial z} &= \frac{\partial}{\partial z}\left(v_y^G-a_y-b_y z-c_y z^2-d_y z^3\right)\\ &= -b_y-2c_y z-3d_y z^2\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 v_z^G}{\partial t \partial z} &= -g_0\left(\frac{r_e}{r_e+z}\right)^2-\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}\left[1+\frac{\beta}{T_b}\left(\frac{r_e z}{r_e+z}-H_b\right)\right]^{-(1+g_0/\beta R)}\left(v_z^G-v_z^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\\ &= -g_0 r_e^2\frac{\partial}{\partial z}\left(z+r_e\right)^{-2}-\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}\left[\frac{T_b-\beta H_b}{T_b}+\frac{\beta r_e}{T_b}\frac{z}{r_e+z}\right]^{-(1+g_0/\beta R)}\left(v_z^G-v_z^W\right)\sqrt{\left(v_x^G-v_x^W\right)^2+\left(v_y^G-v_y^W\right)^2+\left(v_z^G-v_z^W\right)^2}\\ &= -2g_0 r_e^2\left(z+r_e\right)^{-3}-\frac{1}{2C_B}\frac{p_b}{RT_b}\frac{\partial}{\partial z}A.B_z.C\\ &= -2g_0 r_e^2\left(z+r_e\right)^{-3}-\frac{1}{2C_B}\frac{p_b}{RT_b}\left(B_y C\frac{\partial A}{\partial z}+AC\frac{\partial B_z}{\partial z}+AB_y\frac{\partial C}{\partial z}\right)=[3,6]\end{aligned}$$

with A , C , $\dfrac{\partial A}{\partial z}$ and $\dfrac{\partial C}{\partial z}$ as above, $B_z=\left(v_z^G-a_z-b_z z-c_z z^2-d_z z^3\right)$ and

$$\begin{aligned}\frac{\partial B_z}{\partial z} &= \frac{\partial}{\partial z}\left(v_z^G-a_z-b_z z-c_z z^2-d_z z^3\right)\\ &= -b_z-2c_z z-3d_z z^2\end{aligned}$$

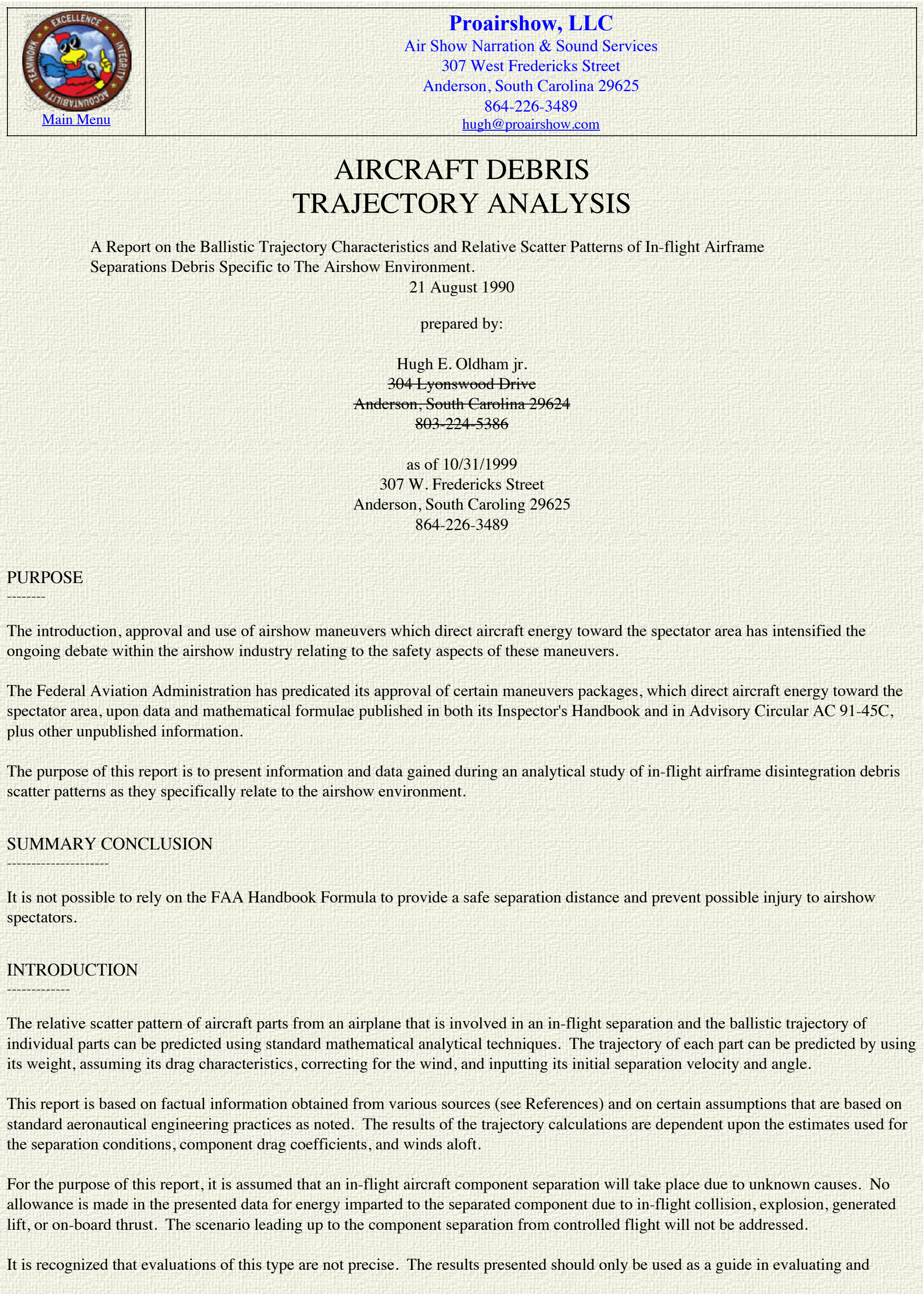
Appendix 4

Flight	Year	Notes
Comets G-ALYP and G-ALYY	1954	In-flight breakup
L-188 Lockheed Electra	1959	Wing separation
Continental 707	1962	Explosive decompression, tail and engine separation
TWA 800	1996	Explosion at 15,000 ft
Adam Air KI574	2007	Loss of control – difficulty in location
Air India AI182	1985	Bomb explosion at 31,000 ft
UA811	1989	Cargo door and structural failure
Knight Air Bandeirante, Leeds	1995	Spiral dive, mid-air breakup
China Airline CI611	2002	Mid-air breakup at FL350
Aer. Itavia DC-9, Ustica	1980	In-flight explosion
BEA Vanguard, Belgium	1971	Rupture at FL190
Nimrod XV230 Afghanistan	2006	Mid-air explosion
Convair 580, NZ	2003	Spiral dive, in-fligh breakup
L-188 OB-R-941	1971	Structural failure after lightning strike
BAC 167 Strikemaster	2006	Wing separation
Partnairs, CV-580	1989	Structural failure at FL220
New York Air Disaster	1960	Mid-air collision
Woomera missile trials		Kepert
HS-125 N40PC	1977	Structural failure
Columbia, STS-107	2003	Disintegration during reentry
Turkish Airlines Flight 981	1984	Cargo door failure, cargo door, seats and passengers separated
UA232 Sioux City	1989	Uncontained fan disk failure
Turbo Commander, Eastbourne	1984	In-flight breakup 19,000 ft
YB-49 Flying wing	1948	Structural failure
Challenger, STS-51-L	1983	Disintegration after launch

Potential accidents for future model validation

Appendix 5

Reproduction of http://www.proairshow.com/aircraft_debris.htm accessed 29th March 2013.



Main Menu

Proairshow, LLC
Air Show Narration & Sound Services
307 West Fredericks Street
Anderson, South Carolina 29625
864-226-3489
hugh@proairshow.com

AIRCRAFT DEBRIS TRAJECTORY ANALYSIS

A Report on the Ballistic Trajectory Characteristics and Relative Scatter Patterns of In-flight Airframe Separations Debris Specific to The Airshow Environment.

21 August 1990

prepared by:

Hugh E. Oldham jr.
~~304 Lyonswood Drive~~
~~Anderson, South Carolina 29624~~
~~803-224-5386~~

as of 10/31/1999
307 W. Fredericks Street
Anderson, South Caroling 29625
864-226-3489

PURPOSE

The introduction, approval and use of airshow maneuvers which direct aircraft energy toward the spectator area has intensified the ongoing debate within the airshow industry relating to the safety aspects of these maneuvers.

The Federal Aviation Administration has predicated its approval of certain maneuvers packages, which direct aircraft energy toward the spectator area, upon data and mathematical formulae published in both its Inspector's Handbook and in Advisory Circular AC 91-45C, plus other unpublished information.

The purpose of this report is to present information and data gained during an analytical study of in-flight airframe disintegration debris scatter patterns as they specifically relate to the airshow environment.

SUMMARY CONCLUSION

It is not possible to rely on the FAA Handbook Formula to provide a safe separation distance and prevent possible injury to airshow spectators.

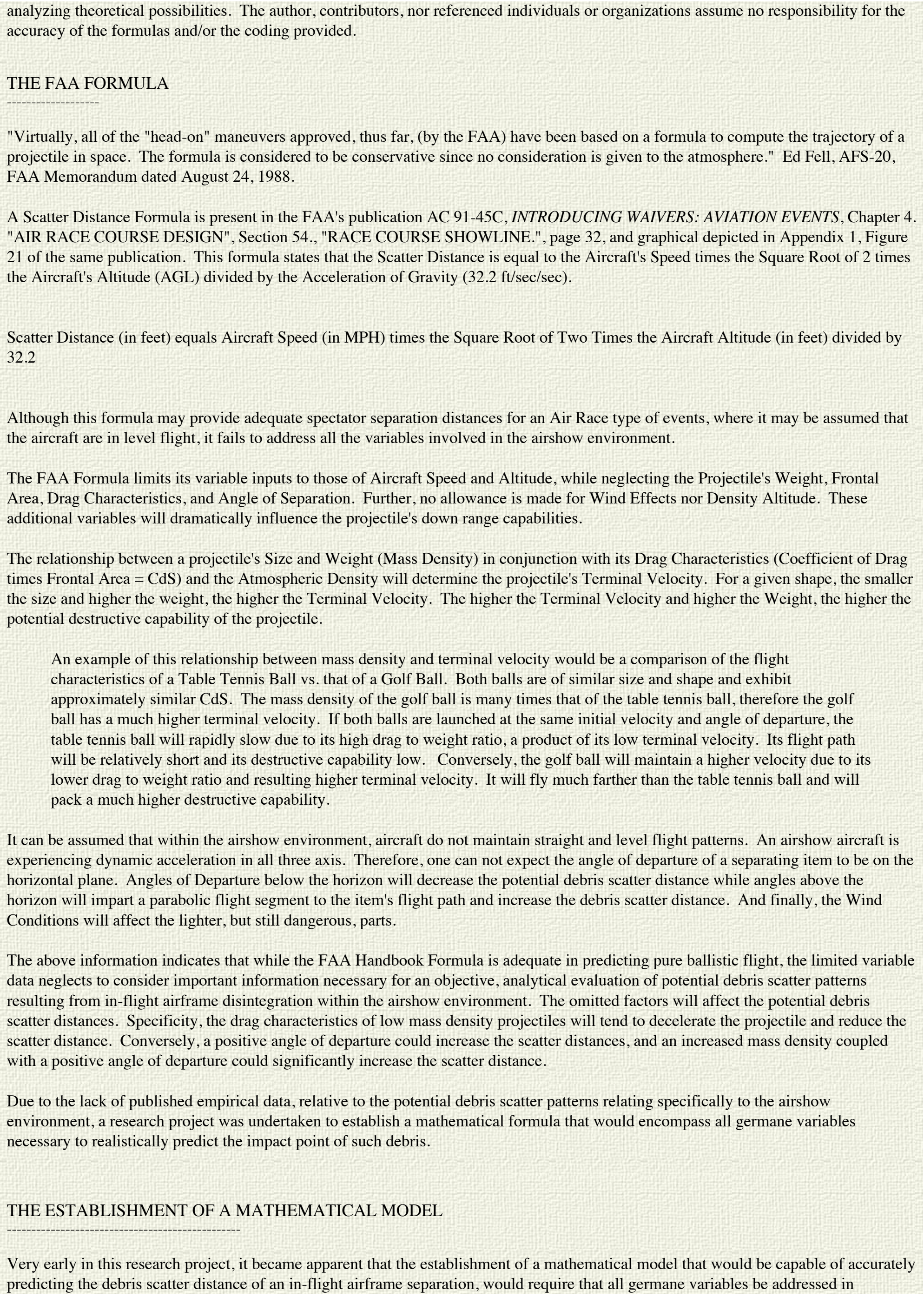
INTRODUCTION

The relative scatter pattern of aircraft parts from an airplane that is involved in an in-flight separation and the ballistic trajectory of individual parts can be predicted using standard mathematical analytical techniques. The trajectory of each part can be predicted by using its weight, assuming its drag characteristics, correcting for the wind, and inputting its initial separation velocity and angle.

This report is based on factual information obtained from various sources (see References) and on certain assumptions that are based on standard aeronautical engineering practices as noted. The results of the trajectory calculations are dependent upon the estimates used for the separation conditions, component drag coefficients, and winds aloft.

For the purpose of this report, it is assumed that an in-flight aircraft component separation will take place due to unknown causes. No allowance is made in the presented data for energy imparted to the separated component due to in-flight collision, explosion, generated lift, or on-board thrust. The scenario leading up to the component separation from controlled flight will not be addressed.

It is recognized that evaluations of this type are not precise. The results presented should only be used as a guide in evaluating and



analyzing theoretical possibilities. The author, contributors, nor referenced individuals or organizations assume no responsibility for the accuracy of the formulas and/or the coding provided.

THE FAA FORMULA

"Virtually, all of the "head-on" maneuvers approved, thus far, (by the FAA) have been based on a formula to compute the trajectory of a projectile in space. The formula is considered to be conservative since no consideration is given to the atmosphere." Ed Fell, AFS-20, FAA Memorandum dated August 24, 1988.

A Scatter Distance Formula is present in the FAA's publication AC 91-45C, *INTRODUCING WAIVERS: AVIATION EVENTS*, Chapter 4. "AIR RACE COURSE DESIGN", Section 54., "RACE COURSE SHOWLINE.", page 32, and graphical depicted in Appendix 1, Figure 21 of the same publication. This formula states that the Scatter Distance is equal to the Aircraft's Speed times the Square Root of 2 times the Aircraft's Altitude (AGL) divided by the Acceleration of Gravity (32.2 ft/sec/sec).

Scatter Distance (in feet) equals Aircraft Speed (in MPH) times the Square Root of Two Times the Aircraft Altitude (in feet) divided by 32.2

Although this formula may provide adequate spectator separation distances for an Air Race type of events, where it may be assumed that the aircraft are in level flight, it fails to address all the variables involved in the airshow environment.

The FAA Formula limits its variable inputs to those of Aircraft Speed and Altitude, while neglecting the Projectile's Weight, Frontal Area, Drag Characteristics, and Angle of Separation. Further, no allowance is made for Wind Effects nor Density Altitude. These additional variables will dramatically influence the projectile's down range capabilities.

The relationship between a projectile's Size and Weight (Mass Density) in conjunction with its Drag Characteristics (Coefficient of Drag times Frontal Area = CdS) and the Atmospheric Density will determine the projectile's Terminal Velocity. For a given shape, the smaller the size and higher the weight, the higher the Terminal Velocity. The higher the Terminal Velocity and higher the Weight, the higher the potential destructive capability of the projectile.

> An example of this relationship between mass density and terminal velocity would be a comparison of the flight characteristics of a Table Tennis Ball vs. that of a Golf Ball. Both balls are of similar size and shape and exhibit approximately similar CdS. The mass density of the golf ball is many times that of the table tennis ball, therefore the golf ball has a much higher terminal velocity. If both balls are launched at the same initial velocity and angle of departure, the table tennis ball will rapidly slow due to its high drag to weight ratio, a product of its low terminal velocity. Its flight path will be relatively short and its destructive capability low. Conversely, the golf ball will maintain a higher velocity due to its lower drag to weight ratio and resulting higher terminal velocity. It will fly much farther than the table tennis ball and will pack a much higher destructive capability.

It can be assumed that within the airshow environment, aircraft do not maintain straight and level flight patterns. An airshow aircraft is experiencing dynamic acceleration in all three axis. Therefore, one can not expect the angle of departure of a separating item to be on the horizontal plane. Angles of Departure below the horizon will decrease the potential debris scatter distance while angles above the horizon will impart a parabolic flight segment to the item's flight path and increase the debris scatter distance. And finally, the Wind Conditions will affect the lighter, but still dangerous, parts.

The above information indicates that while the FAA Handbook Formula is adequate in predicting pure ballistic flight, the limited variable data neglects to consider important information necessary for an objective, analytical evaluation of potential debris scatter patterns resulting from in-flight airframe disintegration within the airshow environment. The omitted factors will affect the potential debris scatter distances. Specificity, the drag characteristics of low mass density projectiles will tend to decelerate the projectile and reduce the scatter distance. Conversely, a positive angle of departure could increase the scatter distances, and an increased mass density coupled with a positive angle of departure could significantly increase the scatter distance.

Due to the lack of published empirical data, relative to the potential debris scatter patterns relating specifically to the airshow environment, a research project was undertaken to establish a mathematical formula that would encompass all germane variables necessary to realistically predict the impact point of such debris.

THE ESTABLISHMENT OF A MATHEMATICAL MODEL

Very early in this research project, it became apparent that the establishment of a mathematical model that would be capable of accurately predicting the debris scatter distance of an in-flight airframe separation, would require that all germane variables be addressed in nonlinear, second order equations. Such equations do not lend themselves to explicit solution, but are readily solved using interactive procedures. For this reason, computer simulation would be necessary. The first attempts to redefine the FAA Formula were attempted using Lotus 123 spreadsheets. As the formulas evolved and additional research was digested, the 123 spreadsheets became cumbersome.

The evolved formulae were then programmed in the BASIC language for solution on an IBM compatible computer. The interactive integration was performed with time increments of 0.05 seconds, displayed at one second intervals in order to achieve economy of computation. More refined methods are available. The BASIC language and MS-DOS were chosen due to their universal availability and understanding. (Copies of the program disk (5 1/4" & 3 1/2") are available at cost.)

THE COMPUTER PROGRAM

A program originally developed by the National Transportation Safety Board (NTSB) was used as a starting point. The NTSB program (Clark 1985) lacked the flexibility to incorporate the possible variables encountered in the airshow environment. It was necessary to subject the NTSB program to a process of refinement and expansion, evolving into a new program specifically tailored to the airshow environment. This new program was named "TAP" for Trajectory Analysis Program.

Further input was gained from The International Society of Air Safety Investigators and informal conversations with many aviation safety experts and aerospace engineers.

The initial TAP Input requirements were as follows:

1. Initial altitude of disintegration.
1. Initial density altitude.
2. Altitude of impact at Ground Level.
3. Wind velocity and direction.
4. Horizontal true airspeed at disintegration.
5. Rate of climb or sink at disintegration.
6. Weight of projectile.
7. Projectile Drag Coefficient.
8. Projectile frontal area.

The TAP design goal Output were as follows:

1. Horizontal distance from disintegration at impact.
2. Horizontal, vertical, and total velocities.
3. Terminal velocity.
4. Time to fall.
5. Flight-path angle at impact.
6. Ground speed of projectile at impact and x and z components of that velocity.

These initial requirements were refined as the process of developing the formulae progressed as follows:

INPUTS

Wind and Density Altitude.

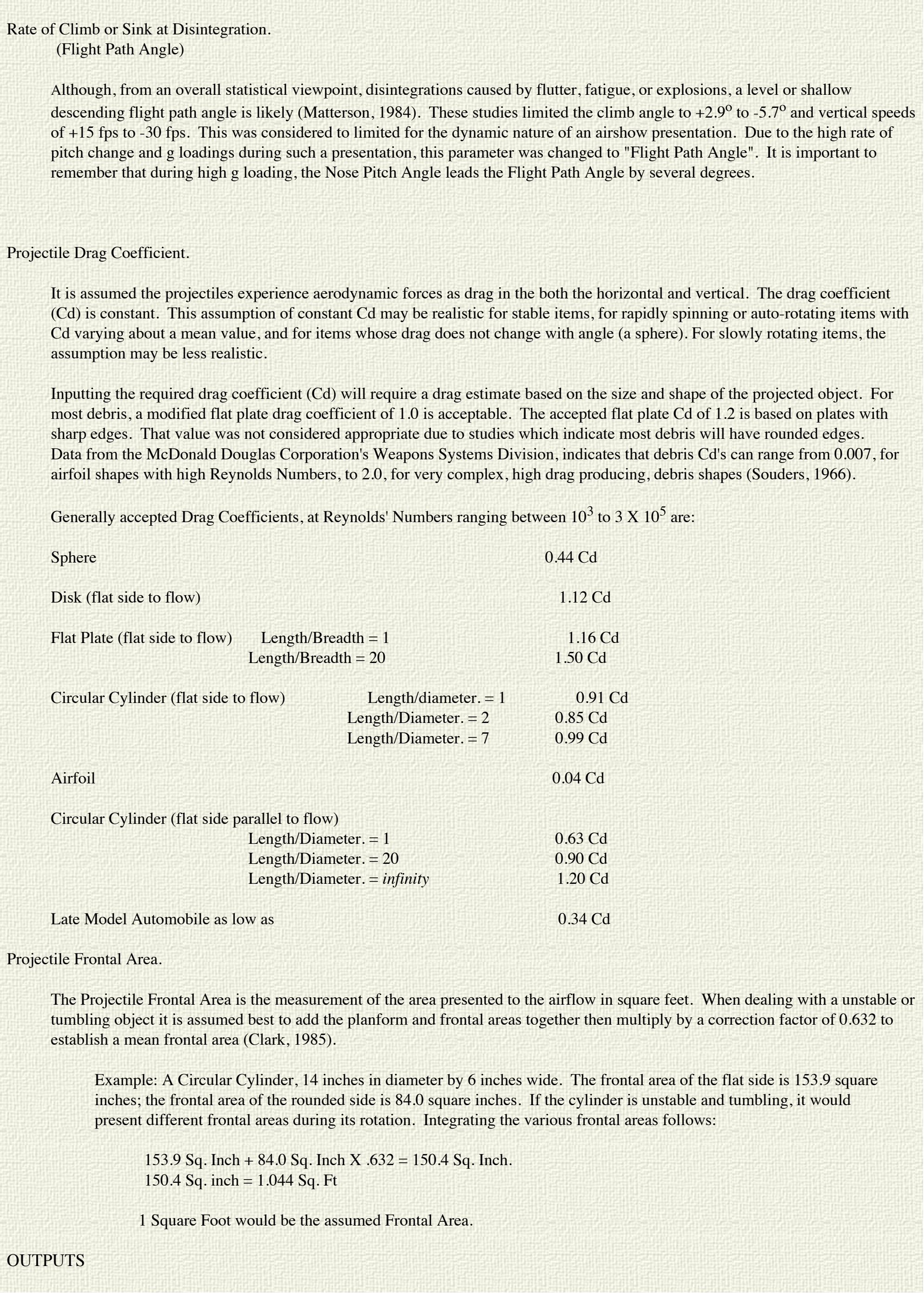
The wind conditions and atmospheric variables are limited in the airshow environment by the localized nature of the event and the limited altitude envelope. The possible wind/altitude shift is limited within the airshow altitude envelope, therefore lateral corrections for wind shift are not made. The vertical component is equal to zero. A model of the wind at various altitudes at the show site was taken from "Dynamic and Physical Meteorology," Haltiner and Martin, McGraw-Hill Book Co., NY, NY, 1957. The following equation for the wind at altitude was derived:

$$\text{Wind} = \text{Surface Wind (SW)} + \text{SW} * (\text{altitude}/30)^{.26}$$

The density altitude at disintegration altitude can be inputted as an additional variable or will default to the disintegration altitude. The atmospheric density at sea level is assumed, a standard day; with a density of .002378 lb sec^2/ft^4 (Slugs). The program adjusts the atmospheric density to the actual altitude as the projectile falls (ICAO Standard Atmosphere, NACA 1955).

Horizontal True Airspeed at Disintegration.

At the instant of disintegration, the aircraft is assumed to be in steady, unyawed, and unaccelerated flight and suddenly disintegrates into a number of parts. (Multiple or progressive disintegrations can be synthesized by superimposition of a series of sudden disintegrations using multiple computer runs.)



Rate of Climb or Sink at Disintegration.
(Flight Path Angle)

Although, from an overall statistical viewpoint, disintegrations caused by flutter, fatigue, or explosions, a level or shallow descending flight path angle is likely (Matterson, 1984). These studies limited the climb angle to $+2.9^{o}$ to -5.7^{o} and vertical speeds of +15 fps to -30 fps. This was considered to limited for the dynamic nature of an airshow presentation. Due to the high rate of pitch change and g loadings during such a presentation, this parameter was changed to "Flight Path Angle". It is important to remember that during high g loading, the Nose Pitch Angle leads the Flight Path Angle by several degrees.

Projectile Drag Coefficient.

It is assumed the projectiles experience aerodynamic forces as drag in the both the horizontal and vertical. The drag coefficient (Cd) is constant. This assumption of constant Cd may be realistic for stable items, for rapidly spinning or auto-rotating items with Cd varying about a mean value, and for items whose drag does not change with angle (a sphere). For slowly rotating items, the assumption may be less realistic.

Inputting the required drag coefficient (Cd) will require a drag estimate based on the size and shape of the projected object. For most debris, a modified flat plate drag coefficient of 1.0 is acceptable. The accepted flat plate Cd of 1.2 is based on plates with sharp edges. That value was not considered appropriate due to studies which indicate most debris will have rounded edges.
Data from the McDonald Douglas Corporation's Weapons Systems Division, indicates that debris Cd's can range from 0.007, for airfoil shapes with high Reynolds Numbers, to 2.0, for very complex, high drag producing, debris shapes (Souders, 1966).

Generally accepted Drag Coefficients, at Reynolds' Numbers ranging between 10^3 to 3 X 10^5 are:

Sphere		0.44 Cd
Disk (flat side to flow)		1.12 Cd
Flat Plate (flat side to flow)	Length/Breadth = 1	1.16 Cd
	Length/Breadth = 20	1.50 Cd
Circular Cylinder (flat side to flow)	Length/diameter. = 1	0.91 Cd
	Length/Diameter. = 2	0.85 Cd
	Length/Diameter. = 7	0.99 Cd
Airfoil		0.04 Cd
Circular Cylinder (flat side parallel to flow)		
	Length/Diameter. = 1	0.63 Cd
	Length/Diameter. = 20	0.90 Cd
	Length/Diameter. = *infinity*	1.20 Cd
Late Model Automobile as low as		0.34 Cd

Projectile Frontal Area.

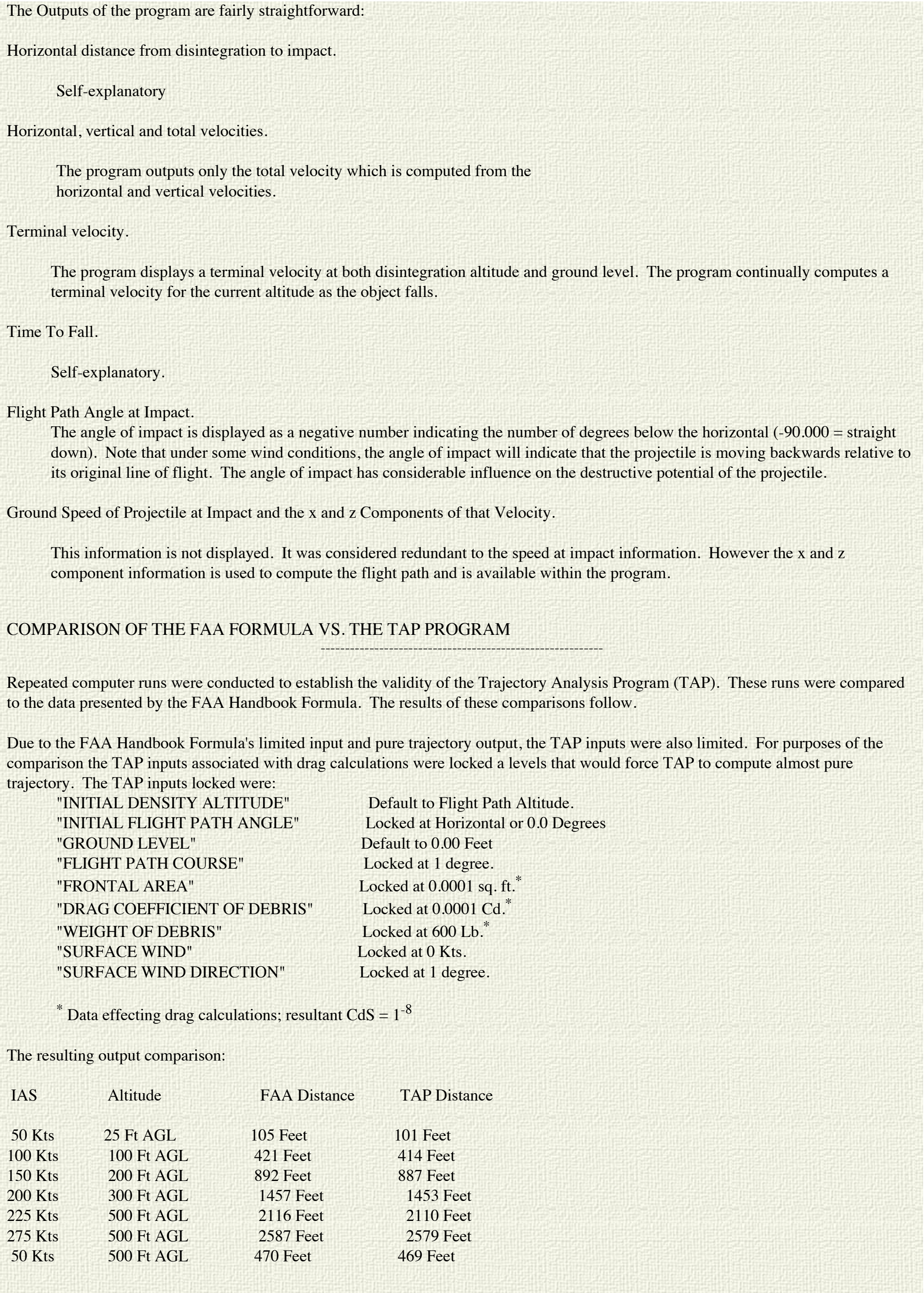
The Projectile Frontal Area is the measurement of the area presented to the airflow in square feet. When dealing with a unstable or tumbling object it is assumed best to add the planform and frontal areas together then multiply by a correction factor of 0.632 to establish a mean frontal area (Clark, 1985).

Example: A Circular Cylinder, 14 inches in diameter by 6 inches wide. The frontal area of the flat side is 153.9 square inches; the frontal area of the rounded side is 84.0 square inches. If the cylinder is unstable and tumbling, it would present different frontal areas during its rotation. Integrating the various frontal areas follows:

153.9 Sq. Inch + 84.0 Sq. Inch X .632 = 150.4 Sq. Inch.
150.4 Sq. inch = 1.044 Sq. Ft

1 Square Foot would be the assumed Frontal Area.

OUTPUTS



The Outputs of the program are fairly straightforward:

Horizontal distance from disintegration to impact.

Self-explanatory

Horizontal, vertical and total velocities.

The program outputs only the total velocity which is computed from the horizontal and vertical velocities.

Terminal velocity.

The program displays a terminal velocity at both disintegration altitude and ground level. The program continually computes a terminal velocity for the current altitude as the object falls.

Time To Fall.

Self-explanatory.

Flight Path Angle at Impact.

The angle of impact is displayed as a negative number indicating the number of degrees below the horizontal (-90.000 = straight down). Note that under some wind conditions, the angle of impact will indicate that the projectile is moving backwards relative to its original line of flight. The angle of impact has considerable influence on the destructive potential of the projectile.

Ground Speed of Projectile at Impact and the x and z Components of that Velocity.

This information is not displayed. It was considered redundant to the speed at impact information. However the x and z component information is used to compute the flight path and is available within the program.

COMPARISON OF THE FAA FORMULA VS. THE TAP PROGRAM

Repeated computer runs were conducted to establish the validity of the Trajectory Analysis Program (TAP). These runs were compared to the data presented by the FAA Handbook Formula. The results of these comparisons follow.

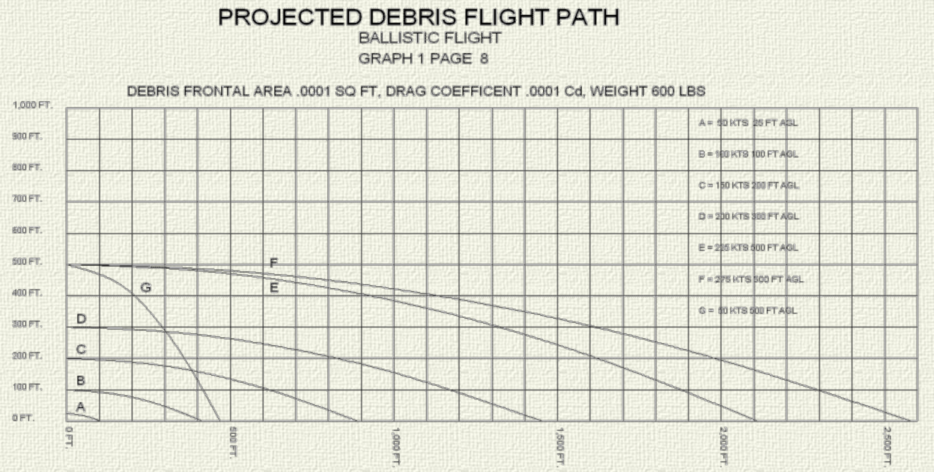
Due to the FAA Handbook Formula's limited input and pure trajectory output, the TAP inputs were also limited. For purposes of the comparison the TAP inputs associated with drag calculations were locked a levels that would force TAP to compute almost pure trajectory. The TAP inputs locked were:

"INITIAL DENSITY ALTITUDE"	Default to Flight Path Altitude.
"INITIAL FLIGHT PATH ANGLE"	Locked at Horizontal or 0.0 Degrees
"GROUND LEVEL"	Default to 0.00 Feet
"FLIGHT PATH COURSE"	Locked at 1 degree.
"FRONTAL AREA"	Locked at 0.0001 sq. ft.*
"DRAG COEFFICIENT OF DEBRIS"	Locked at 0.0001 Cd.*
"WEIGHT OF DEBRIS"	Locked at 600 Lb.*
"SURFACE WIND"	Locked at 0 Kts.
"SURFACE WIND DIRECTION"	Locked at 1 degree.

* Data effecting drag calculations; resultant CdS = 1^{-8}

The resulting output comparison:

IAS	Altitude	FAA Distance	TAP Distance
50 Kts	25 Ft AGL	105 Feet	101 Feet
100 Kts	100 Ft AGL	421 Feet	414 Feet
150 Kts	200 Ft AGL	892 Feet	887 Feet
200 Kts	300 Ft AGL	1457 Feet	1453 Feet
225 Kts	500 Ft AGL	2116 Feet	2110 Feet
275 Kts	500 Ft AGL	2587 Feet	2579 Feet
50 Kts	500 Ft AGL	470 Feet	469 Feet



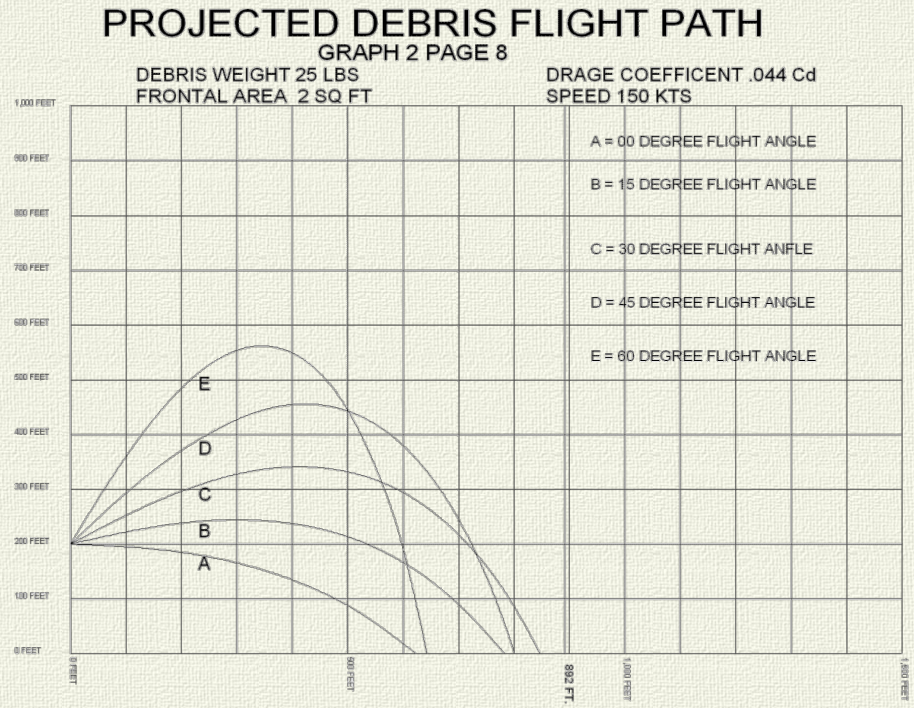
The above data indicates that it can be assumed that the FAA and TAP formulas will yield similar results when compared in the calculation of pure trajectory.

Yet, in the real world, aircraft debris will not travel in a vacuum. The debris will be subject to drag from the atmosphere and it cannot be assumed that the debris will depart on a horizontal plane.

A further comparison varies from the FAA data. The following variables were unlocked and set to simulate a projectile of moderate mass density with different Cd's and Angles of Departure.

Initial Indicated Airspeed	150 Knots.
Initial Flight Path Altitude	200 Feet AGL
Frontal Area	2 Sq. Ft
Weight of Debris	25 Lbs.

Cd	CdS	Flight Angle	TAP Distance	Terminal Velocity	Time To Impact	Speed at Impact
0.44	0.88	00.00	624 Ft	91.5 Kts	3.99 Sec	80 Kts
0.44	0.88	15.00	783 Ft	91.5 Kts	5.95 Sec	74 Kts
0.44	0.88	30.00	848 Ft	91.5 Kts	7.90 Sec	75 Kts
0.44	0.88	45.00	801 Ft	91.5 Kts	9.60 Sec	78 Kts
0.44	0.88	60.00	643 Ft	91.5 Kts	11.00 Sec	81 Kts



PROJECTED DEBRIS FLIGHT PATH

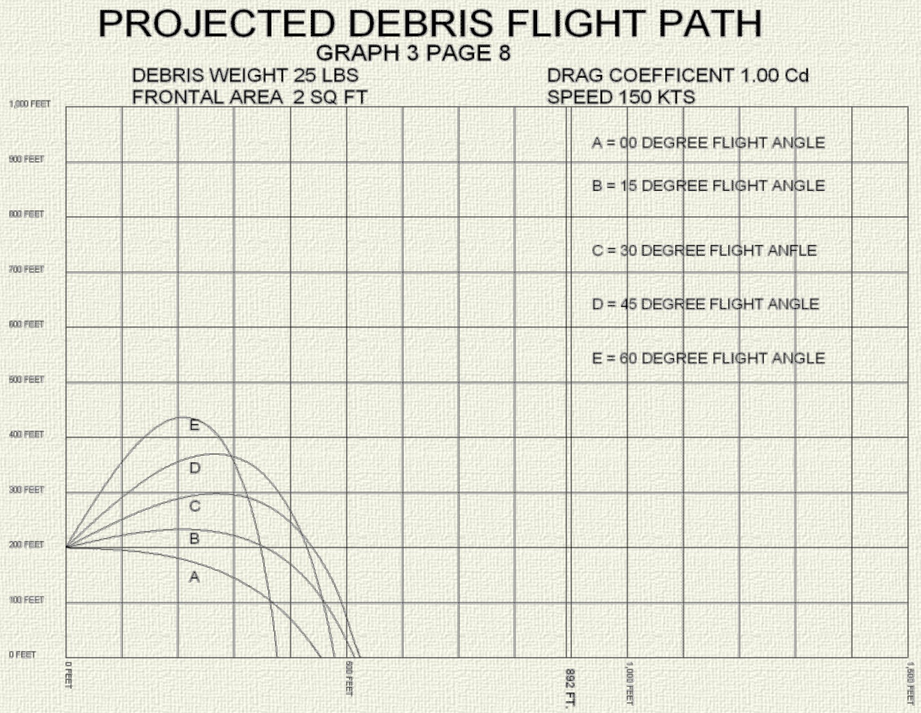
GRAPH 2 PAGE 8

DEBRIS WEIGHT 25 LBS
FRONTAL AREA 2 SQ FT

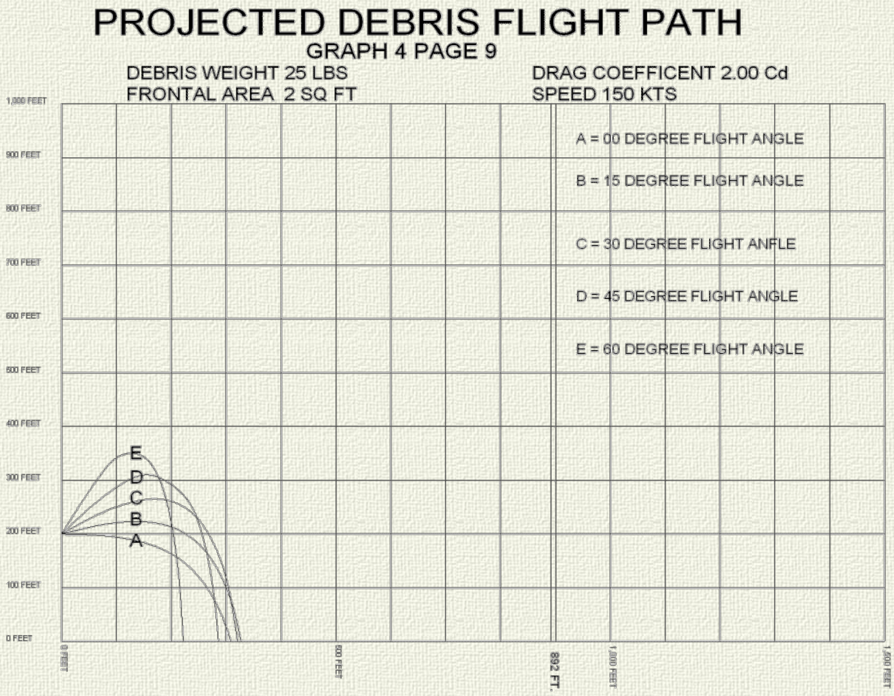
DRAGE COEFFICENT .044 Cd
SPEED 150 KTS

Cd	CdS	Flight Angle	TAP Distance	Terminal Velocity	Time To Impact	Speed at Impact
1.00	2.00	00.00	455 Ft	60.7 Kts	4.35 Sec	54 Kts
1.00	2.00	15.00	515 Ft	60.7 Kts	5.85 Sec	54 Kts
1.00	2.00	30.00	525 Ft	60.7 Kts	7.35 Sec	56 Kts
1.00	2.00	45.00	479 Ft	60.7 Kts	8.65 Sec	57 Kts
1.00	2.00	60.00	377 Ft	60.7 Kts	9.75 Sec	59 Kts

See Graph 3



Cd	CdS	Flight Angle	TAP Distance	Terminal Velocity	Time To Impact	Speed at Impact
2.00	4.00	00.00	309 Ft	42.9 Kts	4.80 Sec	41 Kts
2.00	4.00	15.00	328 Ft	42.9 Kts	5.90 Sec	41 Kts
2.00	4.00	30.00	321 Ft	42.9 Kts	7.05 Sec	42 Kts
2.00	4.00	45.00	287 Ft	42.9 Kts	8.10 Sec	42 Kts
2.00	4.00	60.00	222 Ft	42.9 Kts	8.95 Sec	43 Kts
				See Graph 4		



The relationship between weight, frontal area, and Cd results in a relative slow terminal velocity. The scatter range is well below the 892 feet predicted by the FAA formula. In this case, the FAA formula proves more than adequate.

> NOTE: The higher the "CdS", for a given weight, the lower the "Flight Path Angle" for maximum throw distance. This is due to the relationship between "Terminal Velocity" and "Time to Impact".

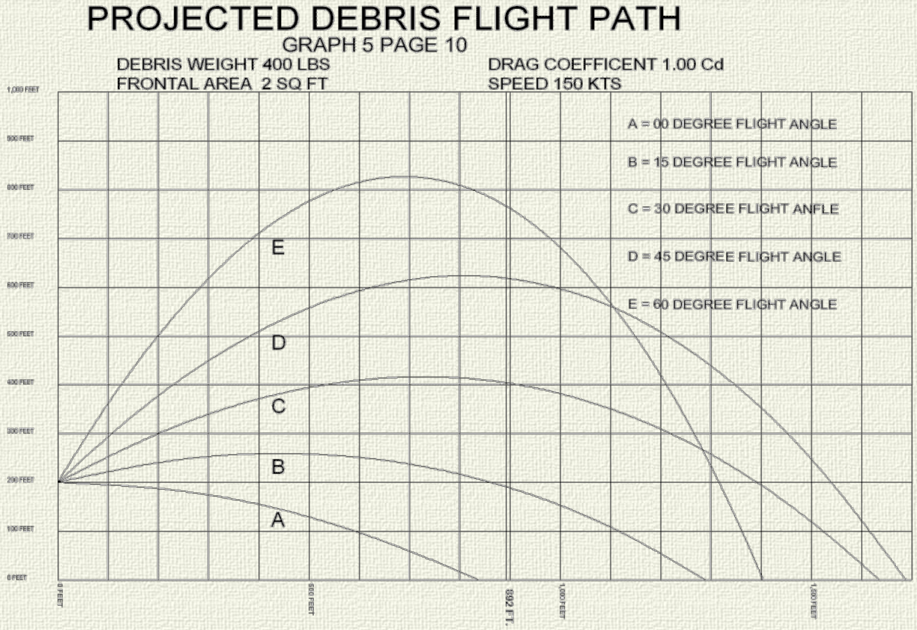
During an in-flight airframe disintegration, it can be assumed that the debris projectile mass density will vary over a broad range. The average Terminal Velocity of light plane parts has been reported to be approximately 35 fps (Logan, 1968). Parts with Terminal Velocities in this range would not pose a threat to the spectator area when using the FAA Scatter Distance Formula. It can also be assumed that many parts (castings, forgings, landing gear assemblies, wheels and brakes, engines and accessories, propeller blades and hubs, etc.) will have much greater mass densities and associated higher Terminal Velocities. These high mass density parts, like the golf ball used in the example on page 3, will have both a higher scatter distance potential and pack the greatest destructive capability. The trajectory of these parts will more closely follow the pure ballistic flight path used by the FAA Scatter Distance Formula. When a Flight Path Angle of Departure above the horizontal is computed, the Scatter Distance of such debris can exceed the FAA Scatter Distance.

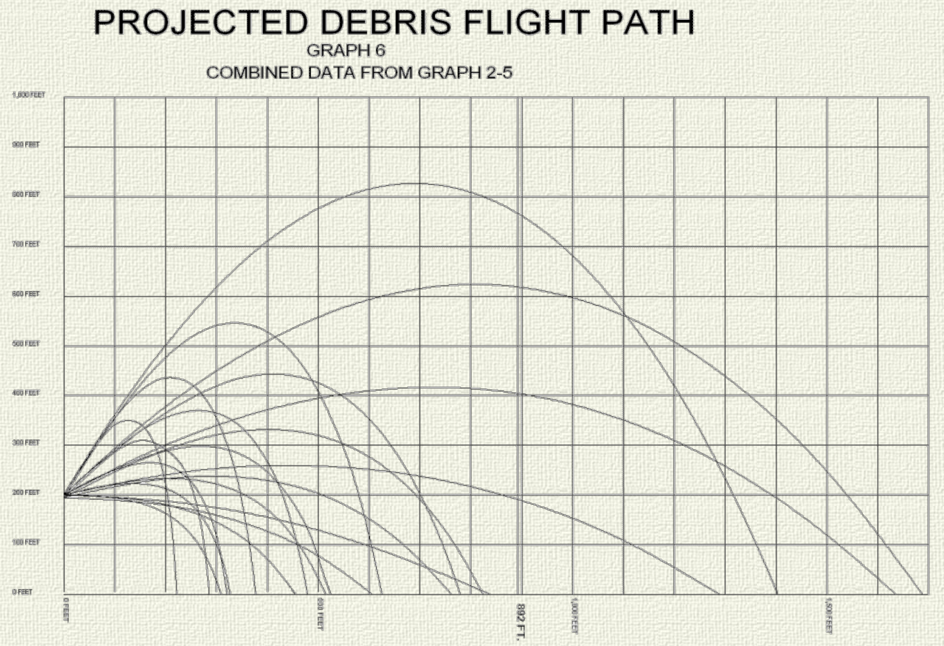
It is recognized that there is a low probability of an airshow aircraft disintegration scenario-taking place while the aircraft is directing energy toward the show's spectators. It must also be recognized that the possibility exists. The results of an disintegration incident which displaces aircraft debris into the designated spectator area would be disastrous. Such a high potential for catastrophic results exists that worst case scenarios must be addressed during an objective, analytical evaluation of any airshow maneuver.

One part of a disintegrating aircraft that has a high mass density and a great chance of intact survivability is the aircraft engine. Reprogramming the variables to simulate worst case scenario involving such an object, will result in a high terminal velocity and a long scatter range.

Initial Indicated Airspeed	150 Knots.
Initial Flight Path Altitude	200 Feet AGL
Frontal Area	2 Sq. Ft
Weight of Debris	400 Lbs

Cd	CdS	Flight Angle	TAP Distance	Terminal Velocity	Time To Impact	Speed at Impact
1.00	2.00	00.00	838 Ft	242.8 Kts	3.59 Sec	142 Kts
1.00	2.00	15.00	1299 Ft	242.8 Kts	6.05 Sec	133 Kts
1.00	2.00	30.00	1636 Ft	242.8 Kts	8.85 Sec	129 Kts
1.00	2.00	45.00	1689 Ft	242.8 Kts	11.45 Sec	131 Kts
1.00	2.00	60.00	1405 Ft	242.8 Kts	13.50 Sec	135 Kts
				See Graph 5		





The above data indicates that even with the flight path horizontal, the heavier projectile would fly to within 54 feet of the predicted impact point of the FAA formula. At impact, the projectile would be flying only 26 degrees below the horizon at 142 Knots. This low impact angle and high speed could allow the projectile to bounce, crossing the 892 foot mark.

With the Flight Path Angle only 3° above the horizon, the projectile would cross the 892 foot mark while still airborne.

The maximum throw distance would occur with the projectile at a Flight Path Angle of Departure of +40°. It would cross the FAA Scatter Distance point of 892 feet from the point of disintegration with an airborne altitude of over 500 feet, and impact the ground 1,710 feet from the point of disintegration at -53° flight angle at 130 Knots. Total in-flight time of the projectile, from disintegration to impact would be 10.65 seconds.

CONCLUSIONS

The dynamic nature of airshow maneuvers does not allow for precise, analytical predictions of aircraft debris scatter patterns. The parameters affecting the potential flight paths of objects, which may separate from controlled flight in any attitude, offer multiple variables that interactively affect the trajectory of the separated part. The data presented in this report and supporting documentation, confirms that the referenced FAA Handbook Formula is inadequate for use in an objective, analytical evaluation of airshow maneuvers directed at the spectator area and the establishment of safe spectator separation distances for these maneuvers. It is not possible to relay on the FAA Handbook Formula to provide safe separation distances and prevent possible injury to airshow spectators in the event of an in-flight airframe disintegration.

Therefore, any airshow maneuver that directs aircraft energy toward the spectator area, approved under current FAA policy, is suspect.

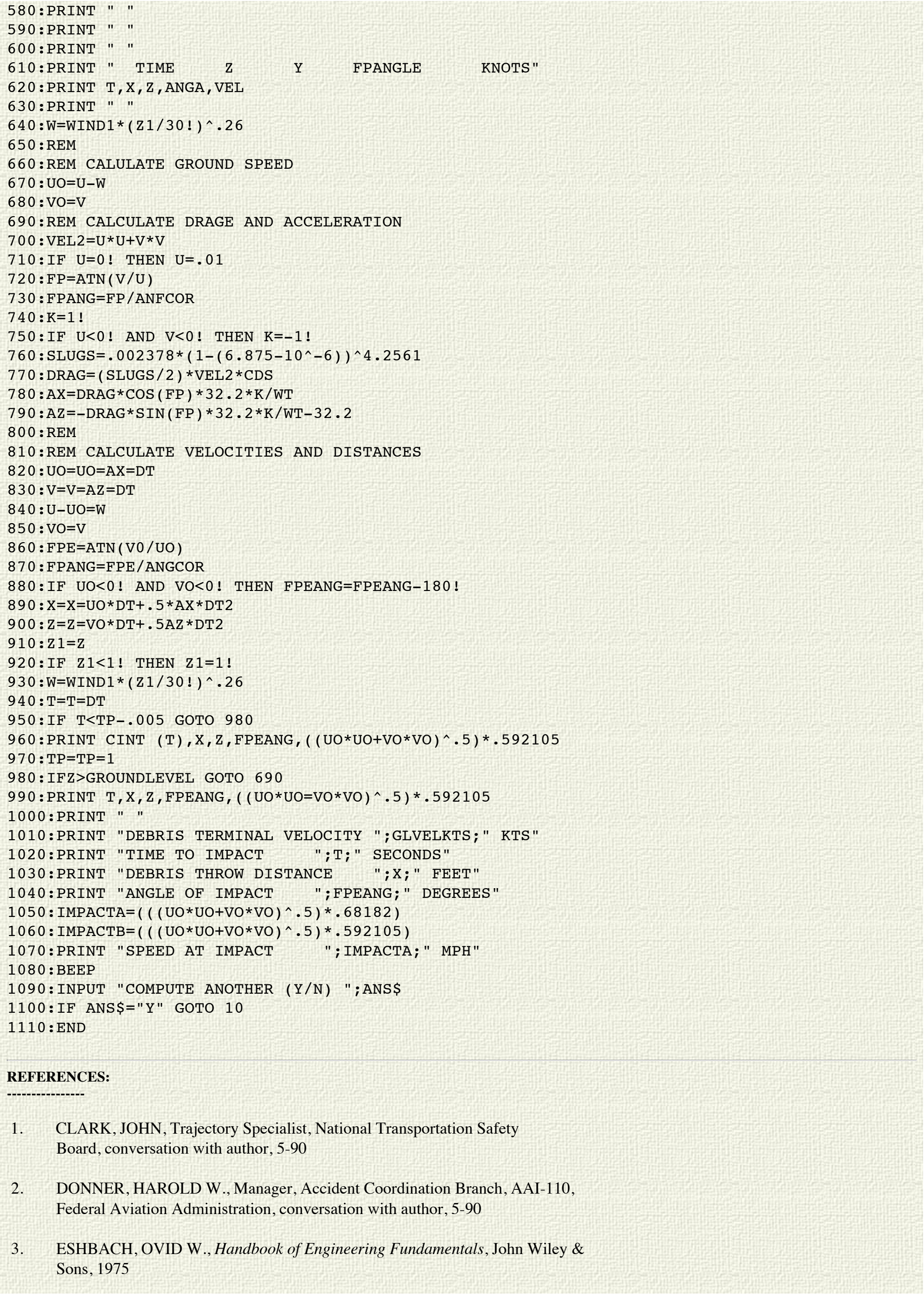
Hugh E. Oldham
21 August 1990

TRAJECTORY ANALYSIS FOR AIRCRAFT DEBRIS

COMPUTER PROGRAM

The TAP Basic Computer Program (On Screen Version) used to generate the projected debris flight path data used in The Airshow Environment Aircraft Debris Trajectory Analysis Report.

```
10:REM FULL TRAJETORY ANALYSIS PROGRAM FOR SCREEN DISPLAY
20:REM PROGRAM 1.10  8/20/90
30:REM FILE NAME "TRAJSCRN"
40:CLS
50:PRINT "TRAJECTORY ANALYSIS"
60:PRINT "FOR"
70:PRINT "AIRCRAFT DEBRIS"
80:PRINT " "
90:INPUT "INTIAL INDICATED AIR SPEED (KTS)";VEL
100:INPUT "INTIAL FLIGHT PATH ANGLE (DEG +/-)";ANGA
110:INPUT INTIAL FLIGHT PATH ALTITUDE (FEET AGL)";ALT
120:INPUT "INTIAL FLIGHT PATH DENSITY ALTITUDE IF DIFFERENT FROM
     INTIAL ALT ";DALT
130:IF DALT =O THEN DALT=ALT
140:PRINT "INTIAL DENSITY ALTITUDE ";DALT
150:INPUT "GROUND LEVEL (MSL FEET) ";GROUNDLEVEL
160:INPUT FLIGHT PATH COURSE MAG (DEG 001-360)";COURSE
170:IF COURSE <1 GOTO 160
180:IF COURSE >360 GOTO 160
190:INPUT "FRONTAL AREA OF DEBRIS (SQ FEET) ';FAREA
200:INPUT "DRAG COEFFICIENT OF DEBRIS (Cd)';CD
210:CDS=CD*FAREA
220:INPUT "WEIGHT OF DERBIS (LBS)";WT
230:INPUT "SURFACE WIND SPEED (KTS)";SWIND
240:INPUT "SURFACE WIND DIRECTION (DEG mAG 01 - 360)";DWIND
250:IF DWIND < 1 GOTO 240
260:IF DWIND > 360 GOTO 240
270:IF COURSE > DWIND THEN WINDC=COS(COURSE-DWIND)*SWIND
280:IF COURSE > DWIND THE WINDC=COS(DWIND-COURSE)*SWIND
290:IF SWIND >0 THE AWIND=WINDC+(ALT/30)^.26
300:IF SWIND=0 THE AWIND=SWIND
310:PRINT "HEAD WIND FACTOR AT FLIGHT PATH ALTITUDE ",AWIND
320:PRINT "HEAD WIND FACTOR AT SURFACE ",WINDC
330:PRINT "COMPUTE AIR MASS DENSITY AT ";DALT;" FEET MSL"
340:REM COMPUTE AIR MASS DENSITY IN SLUGS PER CUBIC FOOT
350:SLUGS=.002378*(1-(6.875*10^-6*ALT))^4.2561
360:GSLUGS=.002378*(1-(6.875*10^-6*GROUNDLEVEL))^4.2561
370:TVEL=(2*WT/(CDS*SLUGS))^.5
380:GLTEVL=(2*WT/(CDS*GSLUGS))^.5
390:PRINT "INTIAL TERMINAL VELOCITY (FPS) = ";TVEL
400:TVELKTS-TVEL*.5921052
410:GLTVELKTS=GLTVEL*.5921052
420:PRINT "INTIAL TERMINAL VELOCITY ";TVELKTS;" KTS"
430:TVELKTS=TVEL*.5921052
440:PRINT "GROUND LEVEL TERMINIAL VELOCITY ";GLTVELKTS," KTS"
450:PI=3.1416
460:TP=1!
470:DT=.05
480:WIND1=WINDC*6080/3600
490:T=0!
500:X=0!
510:Z=ALT
520:ANGCOR=PI/180!
530:DT2=DT*DT
540:ANG=ANGA*ANGCOR
550:REM CALCULATE TRUE AIRSPEED (FPS)
560:U=1.69*VEL*COS(ANG)
570:V=1.69*VEL*SIN(ANG)
```



```
580:PRINT " "
590:PRINT " "
600:PRINT " "
610:PRINT "  TIME     Z      Y     FPANGLE      KNOTS"
620:PRINT T,X,Z,ANGA,VEL
630:PRINT " "
640:W=WIND1*(Z1/30!)^.26
650:REM
660:REM CALULATE GROUND SPEED
670:UO=U-W
680:VO=V
690:REM CALCULATE DRAGE AND ACCELERATION
700:VEL2=U*U+V*V
710:IF U=0! THEN U=.01
720:FP=ATN(V/U)
730:FPANG=FP/ANFCOR
740:K=1!
750:IF U<0! AND V<0! THEN K=-1!
760:SLUGS=.002378*(1-(6.875-10^-6))^4.2561
770:DRAG=(SLUGS/2)*VEL2*CDS
780:AX=DRAG*COS(FP)*32.2*K/WT
790:AZ=-DRAG*SIN(FP)*32.2*K/WT-32.2
800:REM
810:REM CALCULATE VELOCITIES AND DISTANCES
820:UO=UO=AX=DT
830:V=V=AZ=DT
840:U-UO=W
850:VO=V
860:FPE=ATN(V0/UO)
870:FPANG=FPE/ANGCOR
880:IF UO<0! AND VO<0! THEN FPEANG=FPEANG-180!
890:X=X=UO*DT+.5*AX*DT2
900:Z=Z=VO*DT+.5AZ*DT2
910:Z1=Z
920:IF Z1<1! THEN Z1=1!
930:W=WIND1*(Z1/30!)^.26
940:T=T=DT
950:IF T<TP-.005 GOTO 980
960:PRINT CINT (T),X,Z,FPEANG,((UO*UO+VO*VO)^.5)*.592105
970:TP=TP=1
980:IFZ>GROUNDLEVEL GOTO 690
990:PRINT T,X,Z,FPEANG,((UO*UO=VO*VO)^.5)*.592105
1000:PRINT " "
1010:PRINT "DEBRIS TERMINAL VELOCITY ";GLVELKTS;" KTS"
1020:PRINT "TIME TO IMPACT     ";T;" SECONDS"
1030:PRINT "DEBRIS THROW DISTANCE    ";X;" FEET"
1040:PRINT "ANGLE OF IMPACT    ";FPEANG;" DEGREES"
1050:IMPACTA=(((UO*UO+VO*VO)^.5)*.68182)
1060:IMPACTB=(((UO*UO+VO*VO)^.5)*.592105)
1070:PRINT "SPEED AT IMPACT     ";IMPACTA;" MPH"
1080:BEEP
1090:INPUT "COMPUTE ANOTHER (Y/N) ";ANS$
1100:IF ANS$="Y" GOTO 10
1110:END
```

REFERENCES:
----

1. CLARK, JOHN, Trajectory Specialist, National Transportation Safety Board, conversation with author, 5-90

2. DONNER, HAROLD W., Manager, Accident Coordination Branch, AAI-110, Federal Aviation Administration, conversation with author, 5-90

3. ESHBACH, OVID W., *Handbook of Engineering Fundamentals*, John Wiley & Sons, 1975

4. FELL, EDGAR C., Federal Aviation Administration Memorandum, INFORMATION: Airshow Bulletin Summary of August 8-12, 1989, dated August 24, 1989.

5. FEDERAL AVAITION ADMINSTRATION, *General Aviation Operations Inspector's Handbook*, Order # 8440.5A, 3-19-75 Change 37.

6. FEDERAL AVIATION ADMINISTRATION, *Waivers: Aviation Events, Advisory Circular AC 91-45B*, 2-1-90

7. GODFREY, DAVE, Flight Test Engineering Section, McDonnell Douglas Corporation, McDonnell Aircraft Company, St. Louis, Mo.

8. HALTINER & MARTIN, *Dynamic and Physical Meteorology*, McGraw - Hill Book Co. NY, 1957.

9. LOGAN, R. M., Fundamentals of break-up analysis. Paper presented at the Failure Analysis Symposium, Ottawa, 1968

10. MATTESON, FREDERICK H., Analysis of Wreckage Patterns from In-Flight Disintegrations. *Forum, The International Society of Air Safety Investigators*, Volume 17, #2, 1984

11. NATIONAL ADVISORY COMMITTEE FOR AERONAUTICS. *Standard Atmosphere-Tables and Data for Altitudes to 65,800 feet*. Report 1235, 1955.

12. NATIONAL TRANSPORTATION SAFETY BOARD, Bureau of Technology, Washington, DC *Trajectory Study*, by John C. Clark, Aerospace Engineer, July 12, 1985.

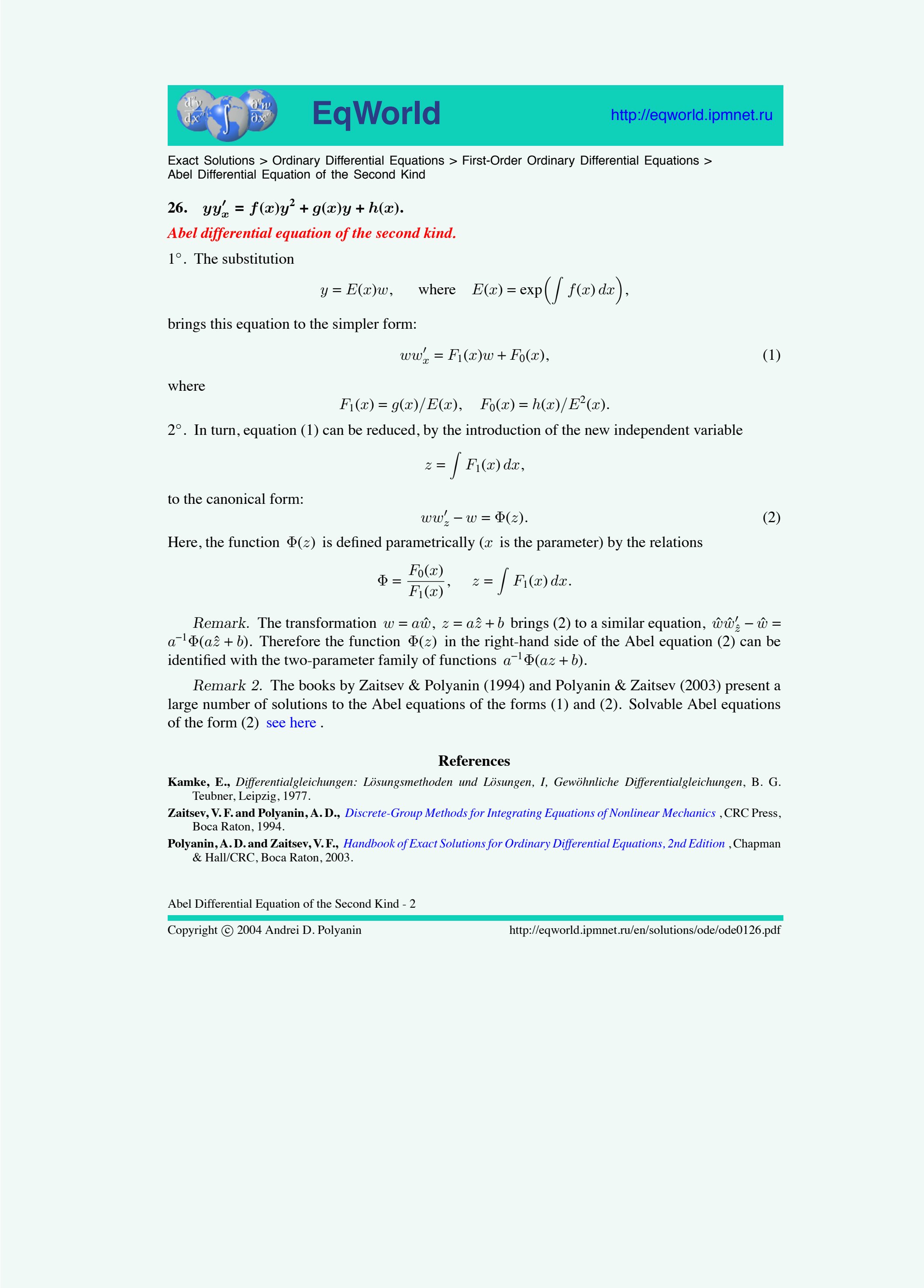
13. SOUDERS, MOTT, *The Engineer's Companion*, John Wiley and Sons, 1966

14. TEMPLIN, R. J. & CALLAN, M. M., *Generalized trajectory curves for bodies moving in air*. National Aeronautical Establishment, Note 13 (LR-159), Ottawa, 1956

Top

Appendix 6

Reproduction of http://eqworld.ipmnet.ru/en/solutions/ode/ode0126.pdf accessed 29th March 2013.



EqWorld http://eqworld.ipmnet.ru

Exact Solutions > Ordinary Differential Equations > First-Order Ordinary Differential Equations > Abel Differential Equation of the Second Kind

26. $yy'_x = f(x)y^2 + g(x)y + h(x).$

Abel differential equation of the second kind.

1°. The substitution

$$y = E(x)w, \qquad \text{where} \quad E(x) = \exp\Bigl(\int f(x)\,dx\Bigr),$$

brings this equation to the simpler form:

$$ww'_x = F_1(x)w + F_0(x), \tag{1}$$

where

$$F_1(x) = g(x)/E(x), \quad F_0(x) = h(x)/E^2(x).$$

2°. In turn, equation (1) can be reduced, by the introduction of the new independent variable

$$z = \int F_1(x)\,dx,$$

to the canonical form:

$$ww'_z - w = \Phi(z). \tag{2}$$

Here, the function $\Phi(z)$ is defined parametrically (x is the parameter) by the relations

$$\Phi = \frac{F_0(x)}{F_1(x)}, \qquad z = \int F_1(x)\,dx.$$

Remark. The transformation $w = a\hat{w}$, $z = a\hat{z} + b$ brings (2) to a similar equation, $\hat{w}\hat{w}'_{\hat{z}} - \hat{w} = a^{-1}\Phi(a\hat{z} + b)$. Therefore the function $\Phi(z)$ in the right-hand side of the Abel equation (2) can be identified with the two-parameter family of functions $a^{-1}\Phi(az + b)$.

Remark 2. The books by Zaitsev & Polyanin (1994) and Polyanin & Zaitsev (2003) present a large number of solutions to the Abel equations of the forms (1) and (2). Solvable Abel equations of the form (2) see here .

References

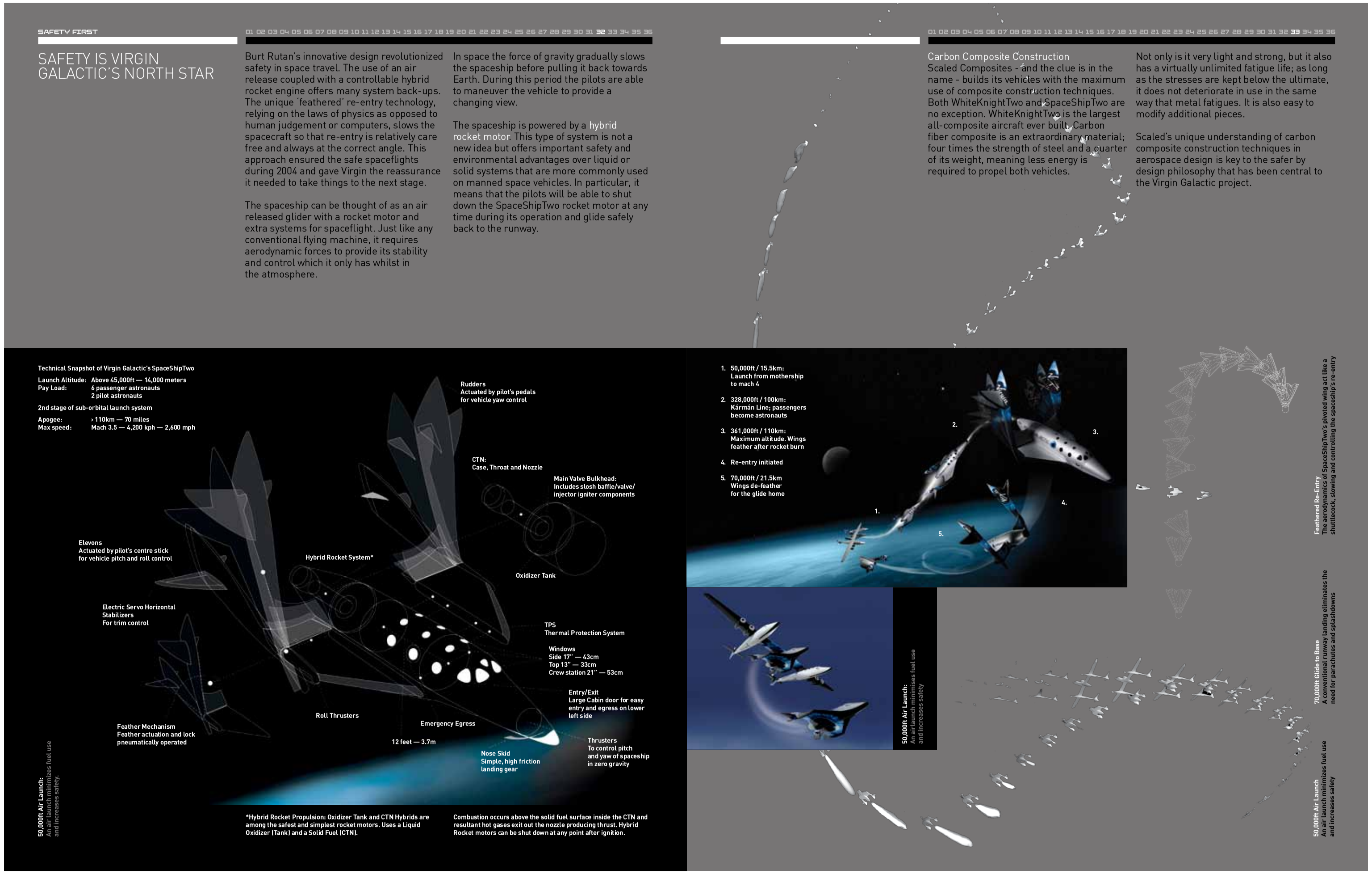
Kamke, E., *Differentialgleichungen: Lösungsmethoden und Lösungen, I, Gewöhnliche Differentialgleichungen*, B. G. Teubner, Leipzig, 1977.

Zaitsev, V. F. and Polyanin, A. D., *Discrete-Group Methods for Integrating Equations of Nonlinear Mechanics* , CRC Press, Boca Raton, 1994.

Polyanin, A. D. and Zaitsev, V. F., *Handbook of Exact Solutions for Ordinary Differential Equations, 2nd Edition* , Chapman & Hall/CRC, Boca Raton, 2003.

Appendix 7

Reproduction of http://www.virgingalactic.com/assets/downloads/ Virgin_Galactic_Brochure.pdf accessed 29th March 2013.



SAFETY IS VIRGIN GALACTIC'S NORTH STAR

Burt Rutan's innovative design revolutionized safety in space travel. The use of an air release coupled with a controllable hybrid rocket engine offers many system back-ups. The unique 'feathered' re-entry technology, relying on the laws of physics as opposed to human judgement or computers, slows the spacecraft so that re-entry is relatively care free and always at the correct angle. This approach ensured the safe spaceflights during 2004 and gave Virgin the reassurance it needed to take things to the next stage.

The spaceship can be thought of as an air released glider with a rocket motor and extra systems for spaceflight. Just like any conventional flying machine, it requires aerodynamic forces to provide its stability and control which it only has whilst in the atmosphere.

In space the force of gravity gradually slows the spaceship before pulling it back towards Earth. During this period the pilots are able to maneuver the vehicle to provide a changing view.

The spaceship is powered by a hybrid rocket motor. This type of system is not a new idea but offers important safety and environmental advantages over liquid or solid systems that are more commonly used on manned space vehicles. In particular, it means that the pilots will be able to shut down the SpaceShipTwo rocket motor at any time during its operation and glide safely back to the runway.

Carbon Composite Construction
Scaled Composites - and the clue is in the name - builds its vehicles with the maximum use of composite construction techniques. Both WhiteKnightTwo and SpaceShipTwo are no exception. WhiteKnightTwo is the largest all-composite aircraft ever built. Carbon fiber composite is an extraordinary material; four times the strength of steel and a quarter of its weight, meaning less energy is required to propel both vehicles.

Not only is it very light and strong, but it also has a virtually unlimited fatigue life; as long as the stresses are kept below the ultimate, it does not deteriorate in use in the same way that metal fatigues. It is also easy to modify additional pieces.

Scaled's unique understanding of carbon composite construction techniques in aerospace design is key to the safer by design philosophy that has been central to the Virgin Galactic project.

Technical Snapshot of Virgin Galactic's SpaceShipTwo

Launch Altitude:	Above 45,000ft — 14,000 meters
Pay Load:	6 passenger astronauts 2 pilot astronauts

2nd stage of sub-orbital launch system

Apogee:	› 110km — 70 miles
Max speed:	Mach 3.5 — 4,200 kph — 2,600 mph

*Hybrid Rocket Propulsion: Oxidizer Tank and CTN Hybrids are among the safest and simplest rocket motors. Uses a Liquid Oxidizer (Tank) and a Solid Fuel (CTN).

Combustion occurs above the solid fuel surface inside the CTN and resultant hot gases exit out the nozzle producing thrust. Hybrid Rocket motors can be shut down at any point after ignition.

50,000ft Air Launch:
An air launch minimizes fuel use and increases safety.

1. 50,000ft / 15.5km: Launch from mothership to mach 4
2. 328,000ft / 100km: Kármán Line; passengers become astronauts
3. 361,000ft / 110km: Maximum altitude. Wings feather after rocket burn
4. Re-entry initiated
5. 70,000ft / 21.5km Wings de-feather for the glide home

50,000ft Air Launch:
An airlaunch minimises fuel use and increases safety

Feathered Re-Entry
The aerodynamics of SpaceShipTwo's pivoted wing act like a shuttlecock, slowing and controlling the spaceship's re-entry

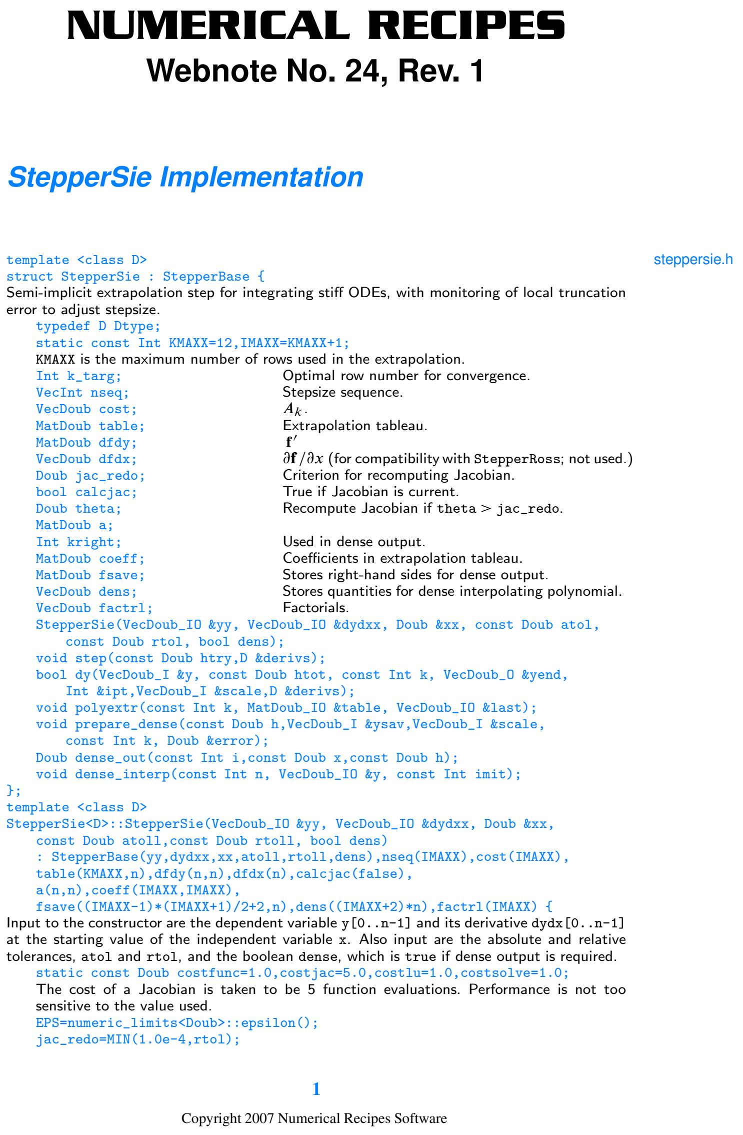
70,000ft Glide to Base
A conventional runway landing eliminates the need for parachutes and splashdowns

50,000ft Air Launch
An air launch minimizes fuel use and increases safety

Rudders
Actuated by pilot's pedals for vehicle yaw control
CTN:
Case, Throat and Nozzle
Main Valve Bulkhead:
Includes slosh baffle/valve/ injector igniter components
Elevons
Actuated by pilot's centre stick for vehicle pitch and roll control
Hybrid Rocket System*
Oxidizer Tank
Electric Servo Horizontal Stabilizers
For trim control
TPS
Thermal Protection System
Windows
Side 17" — 43cm
Top 13" — 33cm
Crew station 21" — 53cm
Entry/Exit
Large Cabin door for easy entry and egress on lower left side
Roll Thrusters
Emergency Egress
Feather Mechanism
Feather actuation and lock pneumatically operated
12 feet — 3.7m
Thrusters
To control pitch and yaw of spaceship in zero gravity
Nose Skid
Simple, high friction landing gear
1.
2.
3.
4.
5.

Appendix 8

Reproduction of http://www.nr.com/webnotes/nr3web24.pdf accessed 29th March 2013.



NUMERICAL RECIPES

Webnote No. 24, Rev. 1

StepperSie Implementation

steppersie.h

```
template <class D>
struct StepperSie : StepperBase {
```

Semi-implicit extrapolation step for integrating stiff ODEs, with monitoring of local truncation error to adjust stepsize.

```
    typedef D Dtype;
    static const Int KMAXX=12,IMAXX=KMAXX+1;
```

KMAXX is the maximum number of rows used in the extrapolation.

```
    Int k_targ;                     Optimal row number for convergence.
    VecInt nseq;                    Stepsize sequence.
    VecDoub cost;                   A_k.
    MatDoub table;                  Extrapolation tableau.
    MatDoub dfdy;                   f′
    VecDoub dfdx;                   ∂f/∂x (for compatibility with StepperRoss; not used.)
    Doub jac_redo;                  Criterion for recomputing Jacobian.
    bool calcjac;                   True if Jacobian is current.
    Doub theta;                     Recompute Jacobian if theta > jac_redo.
    MatDoub a;
    Int kright;                     Used in dense output.
    MatDoub coeff;                  Coefficients in extrapolation tableau.
    MatDoub fsave;                  Stores right-hand sides for dense output.
    VecDoub dens;                   Stores quantities for dense interpolating polynomial.
    VecDoub factrl;                 Factorials.
    StepperSie(VecDoub_IO &yy, VecDoub_IO &dydxx, Doub &xx, const Doub atol,
        const Doub rtol, bool dens);
    void step(const Doub htry,D &derivs);
    bool dy(VecDoub_I &y, const Doub htot, const Int k, VecDoub_O &yend,
        Int &ipt,VecDoub_I &scale,D &derivs);
    void polyextr(const Int k, MatDoub_IO &table, VecDoub_IO &last);
    void prepare_dense(const Doub h,VecDoub_I &ysav,VecDoub_I &scale,
        const Int k, Doub &error);
    Doub dense_out(const Int i,const Doub x,const Doub h);
    void dense_interp(const Int n, VecDoub_IO &y, const Int imit);
};
template <class D>
StepperSie<D>::StepperSie(VecDoub_IO &yy, VecDoub_IO &dydxx, Doub &xx,
    const Doub atoll,const Doub rtoll, bool dens)
    : StepperBase(yy,dydxx,xx,atoll,rtoll,dens),nseq(IMAXX),cost(IMAXX),
    table(KMAXX,n),dfdy(n,n),dfdx(n),calcjac(false),
    a(n,n),coeff(IMAXX,IMAXX),
    fsave((IMAXX-1)*(IMAXX+1)/2+2,n),dens((IMAXX+2)*n),factrl(IMAXX) {
```

Input to the constructor are the dependent variable y[0..n-1] and its derivative dydx[0..n-1] at the starting value of the independent variable x. Also input are the absolute and relative tolerances, atol and rtol, and the boolean dense, which is true if dense output is required.

```
    static const Doub costfunc=1.0,costjac=5.0,costlu=1.0,costsolve=1.0;
```

The cost of a Jacobian is taken to be 5 function evaluations. Performance is not too sensitive to the value used.

```
    EPS=numeric_limits<Doub>::epsilon();
    jac_redo=MIN(1.0e-4,rtol);
    theta=2.0*jac_redo;                         Make sure Jacobian is computed on first step.
    nseq[0]=2;                                  Sequence is different from StepperBS.
    nseq[1]=3;
    for (Int i=2;i<IMAXX;i++)
        nseq[i]=2*nseq[i-2];
    cost[0]=costjac+costlu+nseq[0]*(costfunc+costsolve);
    for (Int k=0;k<KMAXX;k++)
        cost[k+1]=cost[k]+(nseq[k+1]-1)*(costfunc+costsolve)+costlu;
    hnext=-1.0e99;                              Impossible value.
    Doub logfact=-log10(rtol+atol)*0.6+0.5;
    k_targ=MAX(1,MIN(KMAXX-1,Int(logfact)));    Initial estimate of optimal k.
    for (Int k=0; k<IMAXX; k++) {               Coefficients in equation (17.3.8), but ratio
        for (Int l=0; l<k; l++) {                   not squared.
            Doub ratio=Doub(nseq[k])/nseq[l];
            coeff[k][l]=1.0/(ratio-1.0);
        }
    }
    factrl[0]=1.0;
    for (Int k=0; k<IMAXX-1; k++)
        factrl[k+1]=(k+1)*factrl[k];
}
template <class D>
void StepperSie<D>::step(const Doub htry,D &derivs) {
```

Attempts a step with stepsize htry. On output, y and x are replaced by their new values, hdid is the stepsize that was actually accomplished, and hnext is the estimated next stepsize.

```
    const Doub STEPFAC1=0.6,STEPFAC2=0.93,STEPFAC3=0.1,STEPFAC4=4.0,
        STEPFAC5=0.5,KFAC1=0.7,KFAC2=0.9;
```

Stepsize and order control parameters are different from StepperBS.

```
    static bool first_step=true,last_step=false;
    static bool forward,reject=false,prev_reject=false;
    static Doub errold;
    Int i,k;
    Doub fac,h,hnew,err;
    bool firstk;
    VecDoub hopt(IMAXX),work(IMAXX);
    VecDoub ysav(n),yseq(n);
    VecDoub ymid(n),scale(n);
    work[0]=1.e30;
    h=htry;
    forward = h>0 ? true : false;
    for (i=0;i<n;i++) ysav[i]=y[i];             Save the starting values.
    if (h != hnext && !first_step) {            h gets reset in Odeint for the last step.
        last_step=true;
    }
    if (reject) {                               Previous step was rejected.
        prev_reject=true;
        last_step=false;
        theta=2.0*jac_redo;                     Make sure Jacobian gets recomputed.
    }
    for (i=0;i<n;i++)                           Initial scaling.
        scale[i]=atol+rtol*abs(y[i]);
    reject=false;
    firstk=true;
    hnew=abs(h);
    compute_jac:                                Restart here if Jacobian error too big.
    if (theta > jac_redo && !calcjac) {         Evaluate Jacobian.
        derivs.jacobian(x,y,dfdx,dfdy);
        calcjac=true;
    }
    while (firstk || reject) {                  Loop until step accepted.
        h = forward ? hnew : -hnew;
        firstk=false;
        reject=false;
        if (abs(h) <= abs(x)*EPS)
    throw("step size underflow in StepperSie");
Int ipt=-1;                       Initialize counter for saving stuff.
for (k=0; k<=k_targ+1;k++) {      The sequence of semi-implicit Euler steps.
    bool success=dy(ysav,h,k,yseq,ipt,scale,derivs);
    if (!success) {               Stability problems, reduce stepsize.
        reject=true;
        hnew=abs(h)*STEPFAC5;
        break;
    }
    if (k == 0)
         y=yseq;
    else                          Store result in tableau.
        for (i=0;i<n;i++)
            table[k-1][i]=yseq[i];
    if (k != 0) {
        polyextr(k,table,y);      Perform extrapolation.
        err=0.0;                  Compute normalized error estimate err_k.
        for (i=0;i<n;i++) {
            scale[i]=atol+rtol*abs(ysav[i]);
            err+=SQR((y[i]-table[0][i])/scale[i]);
        }
        err=sqrt(err/n);
        if (err > 1.0/EPS || (k > 1 && err >= errold)) {
            reject=true;          Stability problems, reduce stepsize.
            hnew=abs(h)*STEPFAC5;
            break;
        }
        errold=max(4.0*err,1.0);
        Doub expo=1.0/(k+1);
        Compute optimal stepsize for this order. Note k instead of 2k in exponent.
        Doub facmin=pow(STEPFAC3,expo);
        if (err == 0.0)
            fac=1.0/facmin;
        else {
            fac=STEPFAC2/pow(err/STEPFAC1,expo);
            fac=MAX(facmin/STEPFAC4,MIN(1.0/facmin,fac));
        }
        hopt[k]=abs(h*fac);
        work[k]=cost[k]/hopt[k];  Work per unit step (17.3.13).
        if ((first_step || last_step) && err <= 1.0)
            break;
        if (k == k_targ-1 && !prev_reject && !first_step && !last_step) {
            if (err <= 1.0)       Converged within order window.
                break;
            else if (err>nseq[k_targ]*nseq[k_targ+1]*4.0) {
                reject=true;      No convergence expected by k_targ+1.
                k_targ=k;
                if (k_targ>1 && work[k-1]<KFAC1*work[k])
                    k_targ--;
                hnew=hopt[k_targ];
                break;
            }
        }
        if (k == k_targ) {
            if (err <= 1.0)       Converged within order window.
                break;
            else if (err>nseq[k+1]*2.0) {
                reject=true;      No convergence expected by k_targ+1.
                if (k_targ>1 && work[k-1]<KFAC1*work[k])
                    k_targ--;
                hnew=hopt[k_targ];
                break;
            }
        }
```

```
                if (k == k_targ+1) {
                    if (err > 1.0) {
                        reject=true;
                        if (k_targ>1 && work[k_targ-1]<KFAC1*work[k_targ])
                            k_targ--;
                        hnew=hopt[k_targ];
                    }
                    break;
                }
            }
        }                                    Go back and try next k.
        if (reject) {                        Arrive here from any break in for loop.
            prev_reject=true;
            if (!calcjac) {
                theta=2.0*jac_redo;
                goto compute_jac;
            }
        }
    }                                        Go back if step was rejected.
    calcjac=false;                           Successful step. Allow Jacobian to be
    if (dense)                                   recomputed if theta too big.
        prepare_dense(h,ysav,scale,k,err);
    xold=x;                                  Used by dense output.
    x+=h;
    hdid=h;
    first_step=false;
    Int kopt;                                Determine optimal order for next step.
    if (k == 1)
        kopt=2;
    else if (k <= k_targ) {
        kopt=k;
        if (work[k-1] < KFAC1*work[k])
            kopt=k-1;
        else if (work[k] < KFAC2*work[k-1])
            kopt=MIN(k+1,KMAXX-1);
    } else {
        kopt=k-1;
        if (k > 2 && work[k-2] < KFAC1*work[k-1])
            kopt=k-2;
        if (work[k] < KFAC2*work[kopt])
            kopt=MIN(k,KMAXX-1);
    }
    if (prev_reject) {                       After a rejected step neither order nor
        k_targ=MIN(kopt,k);                      stepsize should increase.
        hnew=MIN(abs(h),hopt[k_targ]);
        prev_reject=false;
    }
    else {                                   Stepsize control for next step.
        if (kopt <= k)
            hnew=hopt[kopt];
        else {
            if (k<k_targ && work[k]<KFAC2*work[k-1])
                hnew=hopt[k]*cost[kopt+1]/cost[k];
            else
                hnew=hopt[k]*cost[kopt]/cost[k];
        }
        k_targ=kopt;
    }
    if (forward)
        hnext=hnew;
    else
        hnext=-hnew;
}
template <class D>
```

```
bool StepperSie<D>::dy(VecDoub_I &y,const Doub htot,const Int k,VecDoub_O &yend,
    Int &ipt,VecDoub_I &scale,D &derivs) {
```

Semi-implicit Euler step. Inputs are y, H, k and scale[0..n-1]. The output is returned as yend[0..n-1]. The counter ipt keeps track of saving the right-hand sides in the correct locations for dense output.

```
    VecDoub del(n),ytemp(n),dytemp(n);
    Int nstep=nseq[k];
    Doub h=htot/nstep;                      Stepsize this trip.
    for (Int i=0;i<n;i++) {                 Set up the matrix 1/h - f'.
        for (Int j=0;j<n;j++) a[i][j] = -dfdy[i][j];
        a[i][i] += 1.0/h;
    }
    LUdcmp alu(a);                          LU decomposition of the matrix.
    Doub xnew=x+h;                          Special step for nonautonomous system.
    derivs(xnew,y,del);
    for (Int i=0;i<n;i++)
        ytemp[i]=y[i];
    alu.solve(del,del);
    if (dense && nstep==k+1) {
        ipt++;
        for (Int i=0;i<n;i++)
            fsave[ipt][i]=del[i];
    }
    for (Int nn=1;nn<nstep;nn++) {          General step.
        for (Int i=0;i<n;i++)
            ytemp[i] += del[i];
        xnew += h;
        derivs(xnew,ytemp,yend);
        if (nn ==1 && k<=1) {               Stability test and test for recomputing Jacobian.
            Doub del1=0.0;
            for (Int i=0;i<n;i++)
                del1 += SQR(del[i]/scale[i]);
            del1=sqrt(del1);
            derivs(x+h,ytemp,dytemp);
            for (Int i=0;i<n;i++)
                del[i]=dytemp[i]-del[i]/h;
            alu.solve(del,del);
            Doub del2=0.0;
            for (Int i=0;i<n;i++)
                del2 += SQR(del[i]/scale[i]);
            del2=sqrt(del2);
            theta=del2/MAX(1.0,del1);
            if (theta > 1.0)
                return false;
        }
        alu.solve(yend,del);
        if (dense && nn >= nstep-k-1) {
            ipt++;
            for (Int i=0;i<n;i++)
                fsave[ipt][i]=del[i];
        }
    }
    for (Int i=0;i<n;i++)                   Last step.
        yend[i]=ytemp[i]+del[i];
    return true;
}
template <class D>
void StepperSie<D>::polyextr(const Int k,MatDoub_IO &table,VecDoub_IO &last) {
```

Use polynomial extrapolation to evaluate l functions at $h = 0$. This routine is identical to the routine in StepperBS.

```
    Int l=last.size();
    for (Int j=k-1; j>0; j--)
        for (Int i=0; i<l; i++)
            table[j-1][i]=table[j][i]+coeff[k][j]*(table[j][i]-table[j-1][i]);
    for (Int i=0; i<l; i++)
        last[i]=table[0][i]+coeff[k][0]*(table[0][i]-last[i]);
}
template <class D>
void StepperSie<D>::prepare_dense(const Doub h,VecDoub_I &ysav,VecDoub_I &scale,
    const Int k,Doub &error) {
```

Store coefficients of interpolating polynomial for dense output in dens array. Input stepsize h, function at beginning of interval ysav[0..n-1], scale factor atol $+ |y|$rtol in scale[0..n-1], and column k in which convergence was achieved. Output interpolation error in error.

```
    kright=k;
    for (Int i=0; i<n; i++) {
        dens[i]=ysav[i];
        dens[n+i]=y[i];
    }
    for (Int klr=0; klr < kright; klr++) {     Compute differences.
        if (klr >= 1) {
            for (Int kk=klr; kk<=k; kk++) {
                Int lbeg=((kk+3)*kk)/2;
                Int lend=lbeg-kk+1;
                for (Int l=lbeg; l>=lend; l--)
                    for (Int i=0; i<n; i++)
                        fsave[l][i]=fsave[l][i]-fsave[l-1][i];
            }
        }
        for (Int kk=klr; kk<=k; kk++) {     Compute derivatives at right end.
            Doub facnj=nseq[kk];
            facnj=pow(facnj,klr+1)/factrl[klr+1];
            Int ipt=((kk+3)*kk)/2;
                Int krn=(kk+2)*n;
            for (Int i=0; i<n; i++) {
                dens[krn+i]=fsave[ipt][i]*facnj;
            }
        }
        for (Int j=klr+1; j<=k; j++) {
            Doub dblenj=nseq[j];
            for (Int l=j; l>=klr+1; l--) {
                Doub factor=dblenj/nseq[l-1]-1.0;
                for (Int i=0; i<n; i++) {
                    Int krn=(l+2)*n+i;
                    dens[krn-n]=dens[krn]+(dens[krn]-dens[krn-n])/factor;
                }
            }
        }
    }
    for (Int in=0; in<n; in++) {             Compute coefficients of the interpolation poly-
        for (Int j=1; j<=kright+1; j++) {        nomial.
            Int ii=n*j+in;
            dens[ii]=dens[ii]-dens[ii-n];
        }
    }
}
template <class D>
Doub StepperSie<D>::dense_out(const Int i,const Doub x,const Doub h) {
```

Evaluate interpolating polynomial for y[i] at location x, where xold $\le$ x $\le$ xold $+$ h.

```
    Doub theta=(x-xold)/h;
    Int k=kright;
    Doub yinterp=dens[(k+1)*n+i];
    for (Int j=1; j<=k; j++)
        yinterp=dens[(k+1-j)*n+i]+yinterp*(theta-1.0);
    return dens[i]+yinterp*theta;
}
```

Appendix 9

This Appendix shows results for the same conditions as given in Chapter 4, but with the wind changed to an angle of +30° from the horizontal (60° measured clockwise from aircraft heading). This results in a similar cross/ tail wind, but with a larger crosswind component than the figures given in Chapter 4.

Large Aircraft Accident - Altitude Variation

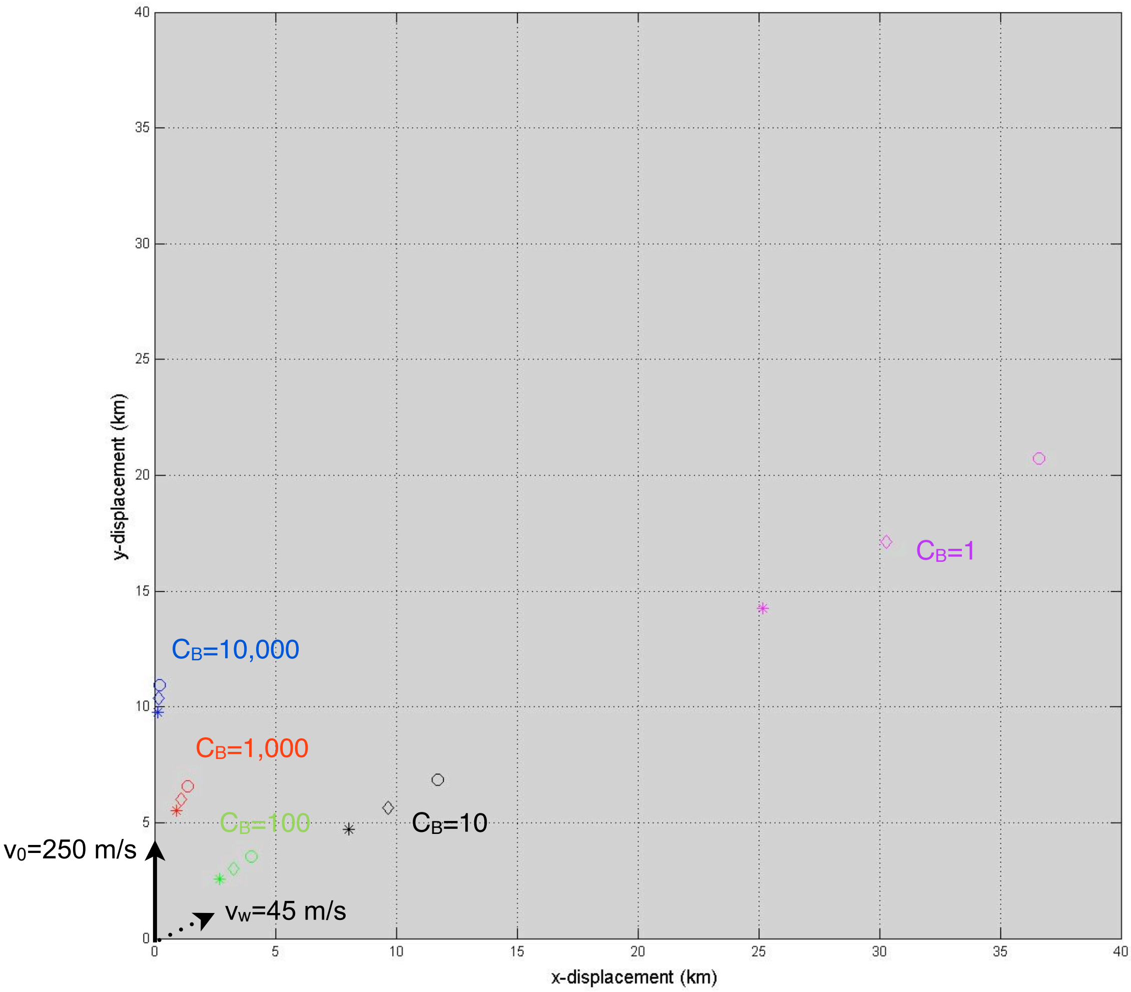


Figure 9.1a - The effect on final wreckage location of changing breakup altitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ = reference, O = +10%, ✱ = -10%

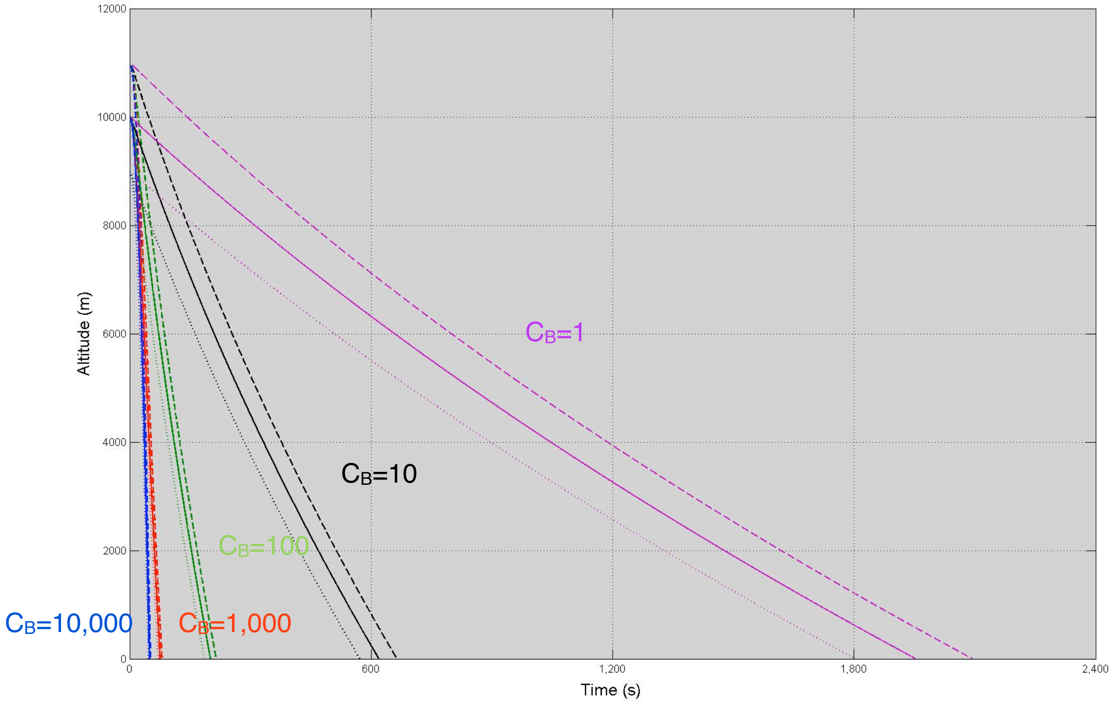


Figure 9.1b - The effect on time to fall to ground of changing breakup altitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ¨¨ = -10%

Initial velocity variation

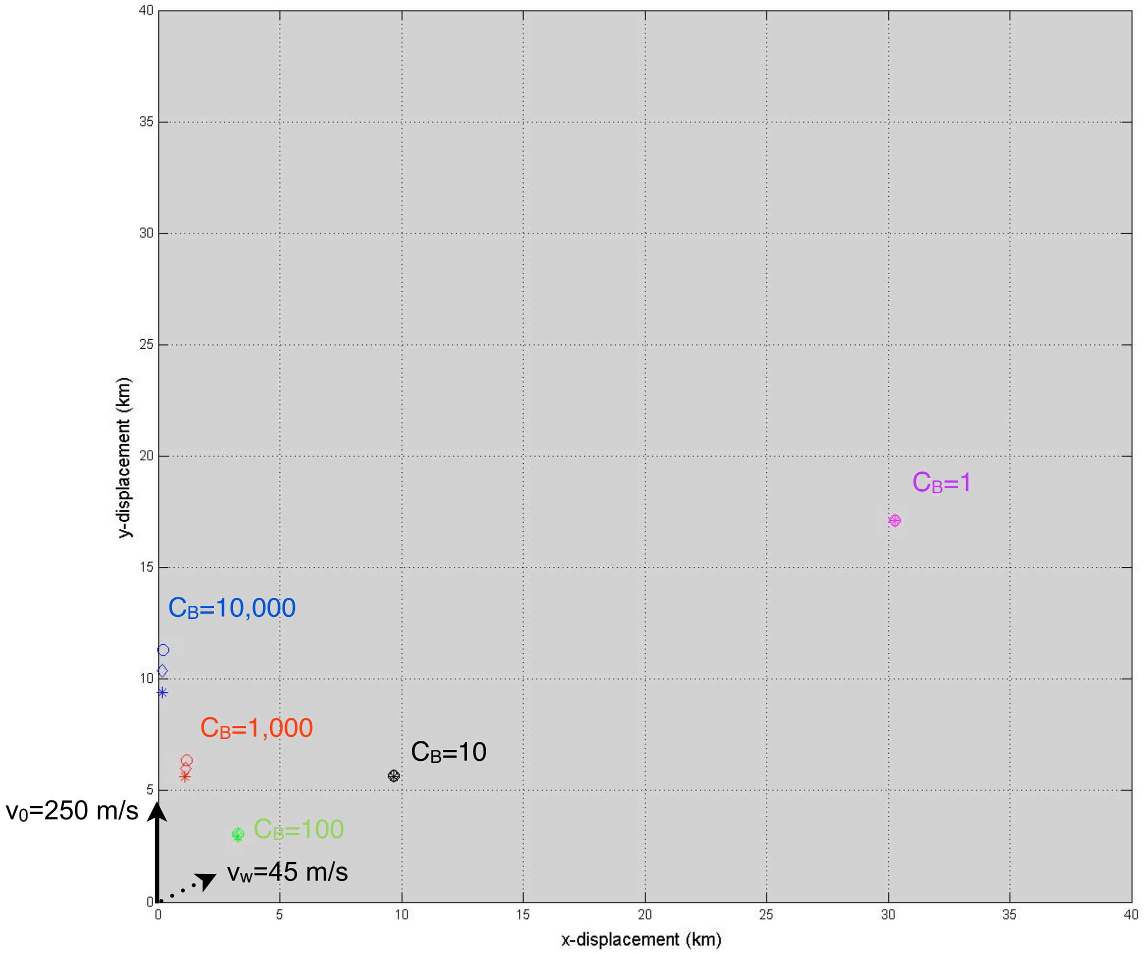


Figure 9.2a - The effect on final wreckage location of changing breakup forward velocity for a simulated large aircraft accident.

C$_B$=10,000 ; C$_B$=1,000, C$_B$=100; C$_B$=10; C$_B$=1

◇ *= reference,* O *= +10%,* ❋ *= -10%*

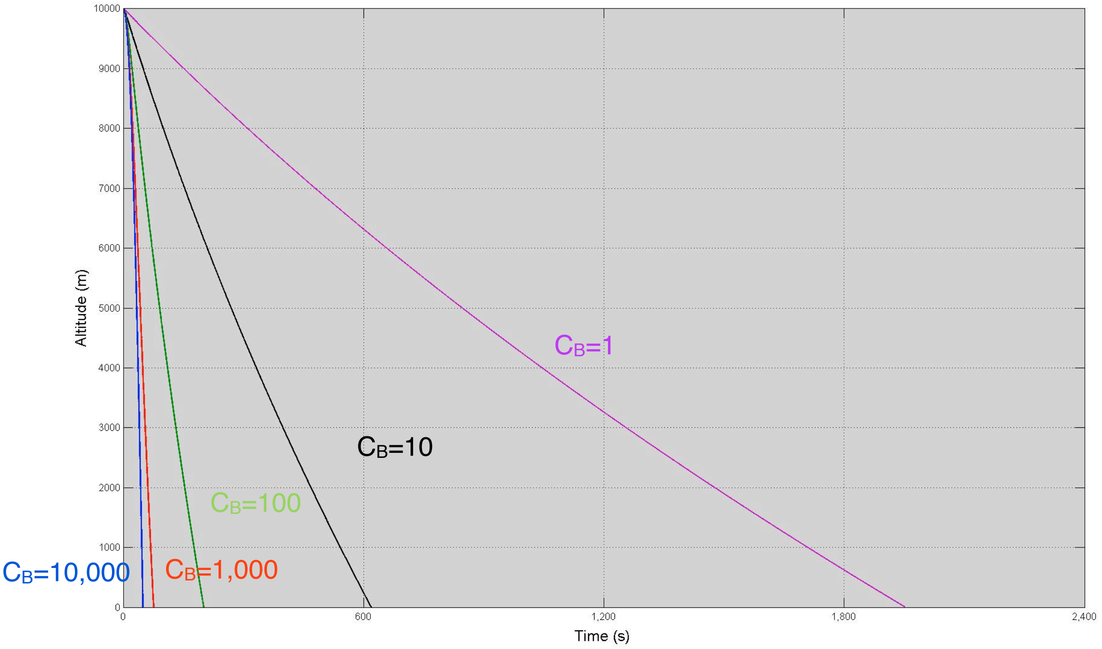


Figure 9.2b - The effect on time to fall to ground of changing breakup forward velocity for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

Wind magnitude variation

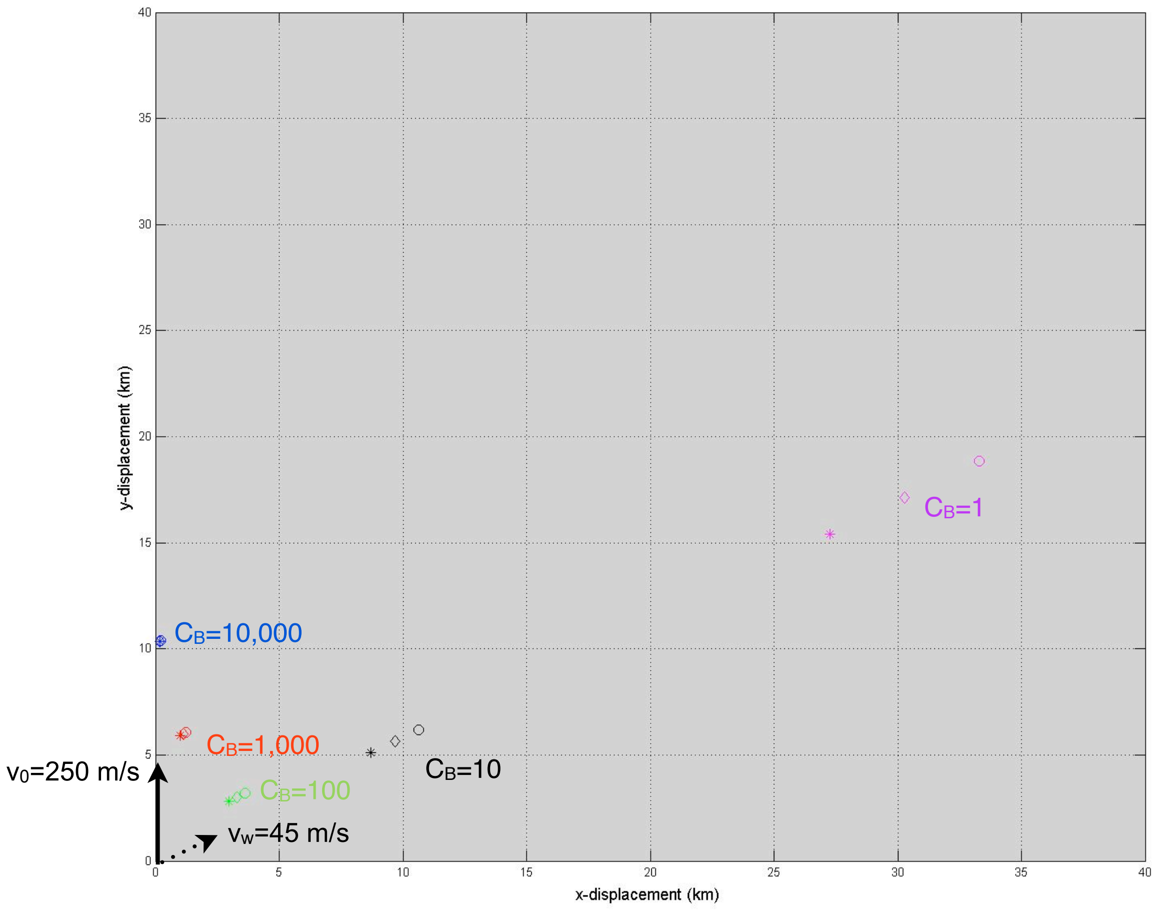


Figure 9.3a - The effect on final wreckage location of changing wind magnitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ = *reference,* O = *+10%,* ✻ = *-10%*

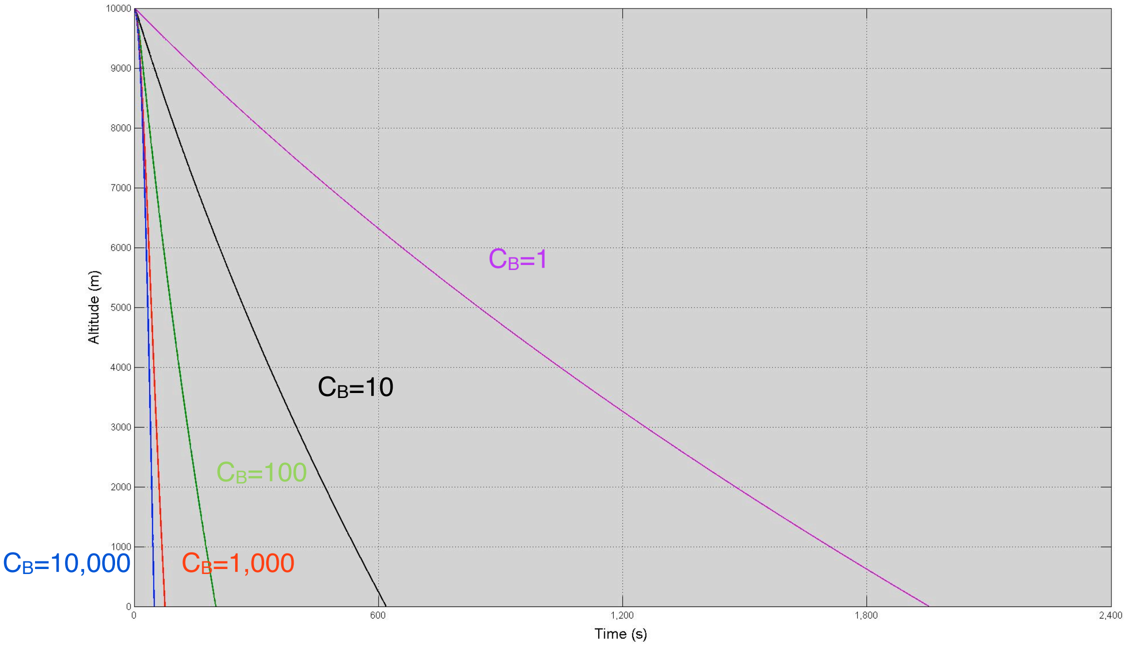


Figure 9.3b - The effect on time taken to fall to ground of changing wind magnitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%, ¨¨ = -10%*

Wind angle variation

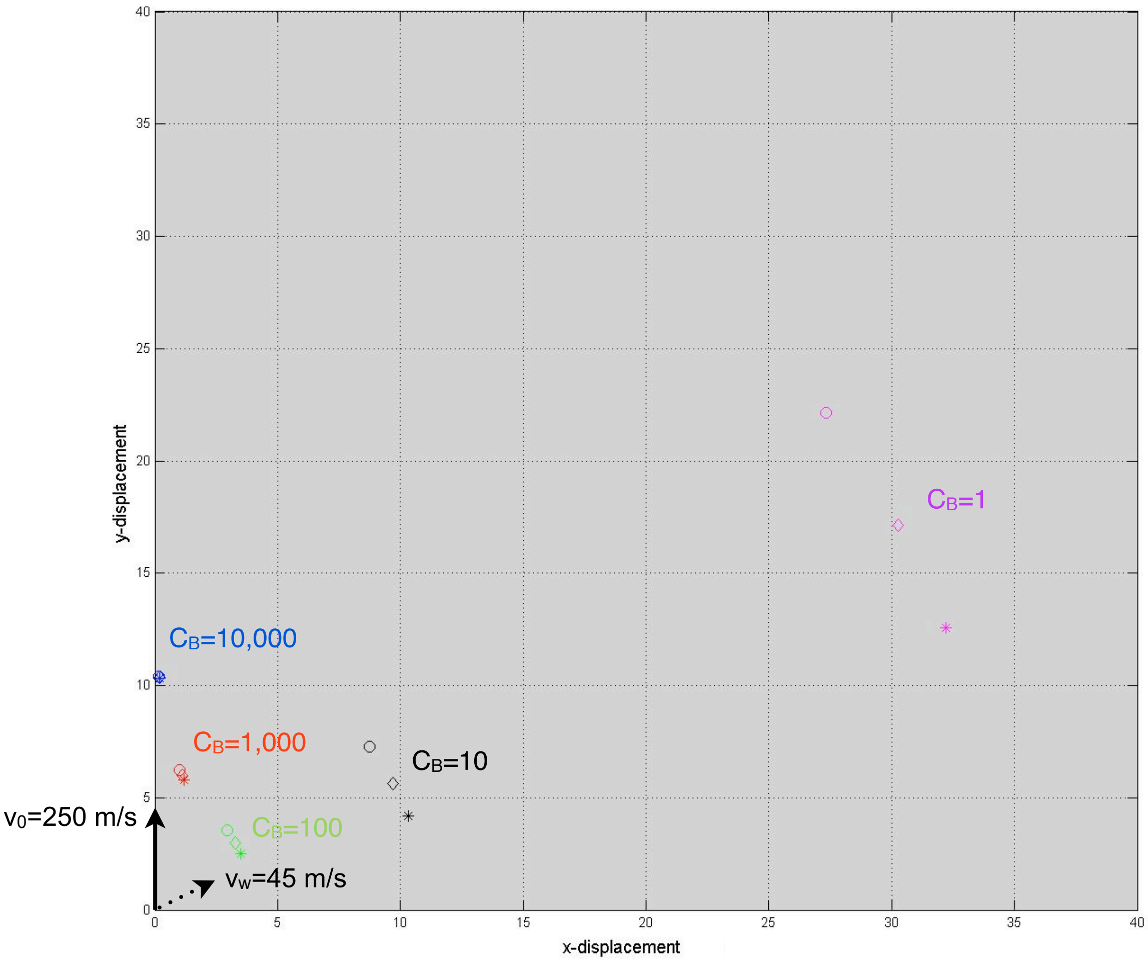


Figure 9.4a - The effect on final wreckage location of changing wind angle for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ❋ *= -10%*

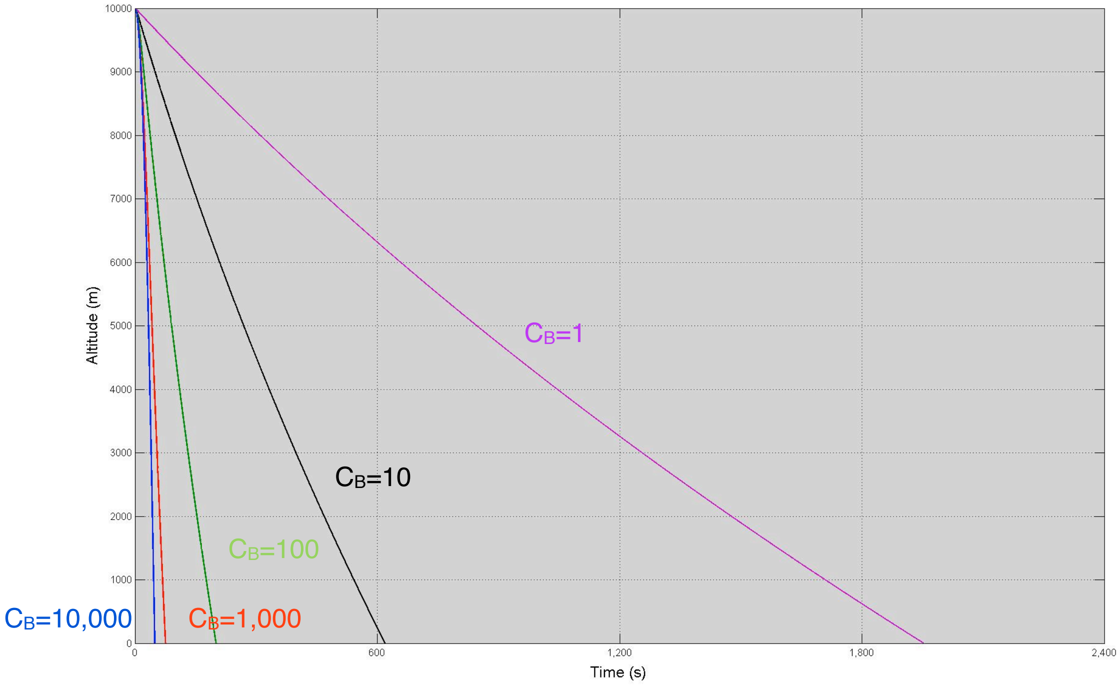


Figure 9.4b - The effect on time taken to fall to ground of changing wind angle for a simulated large aircraft accident..

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ¨¨ = -10%

Small Aircraft Accident - Altitude Variation

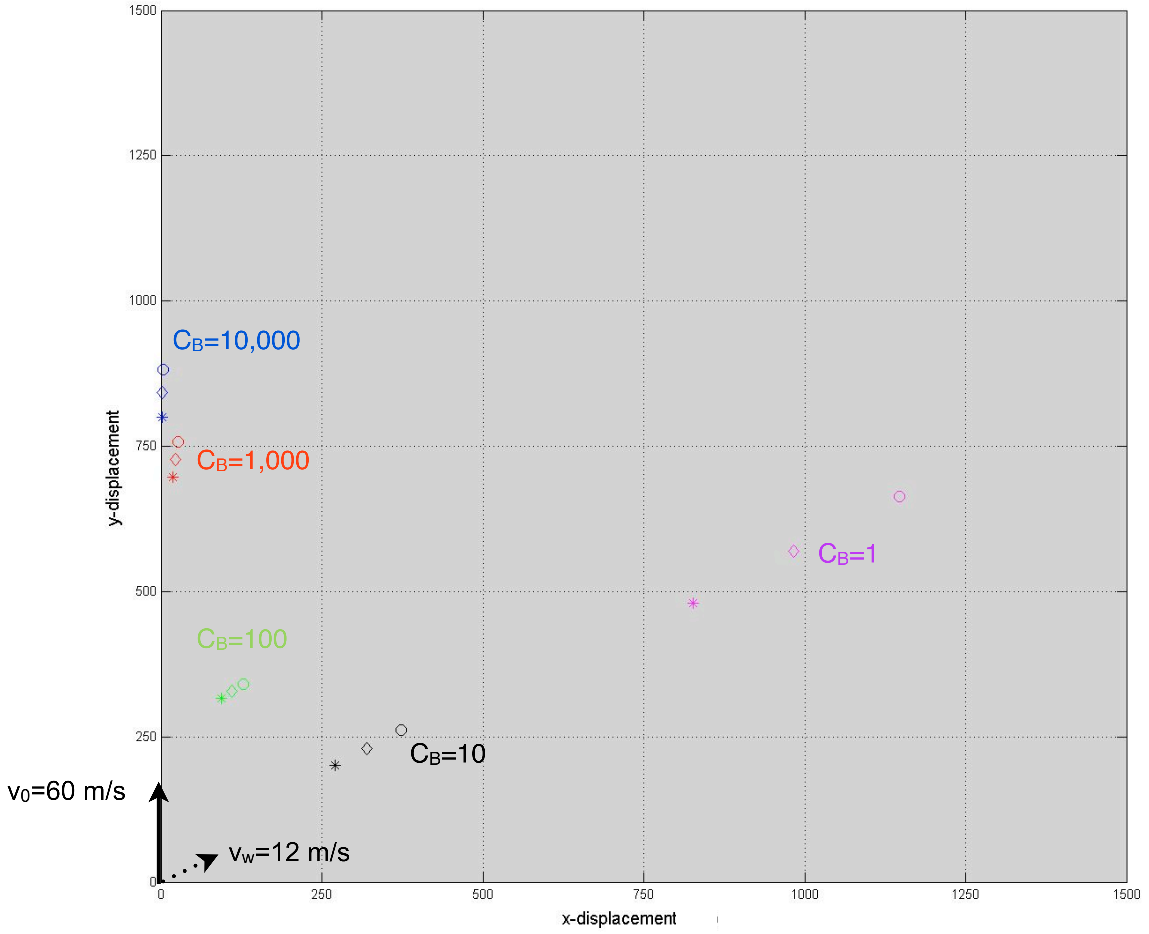


Figure 9.5a - The effect on final wreckage location of changing breakup altitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ✱ *= -10%*

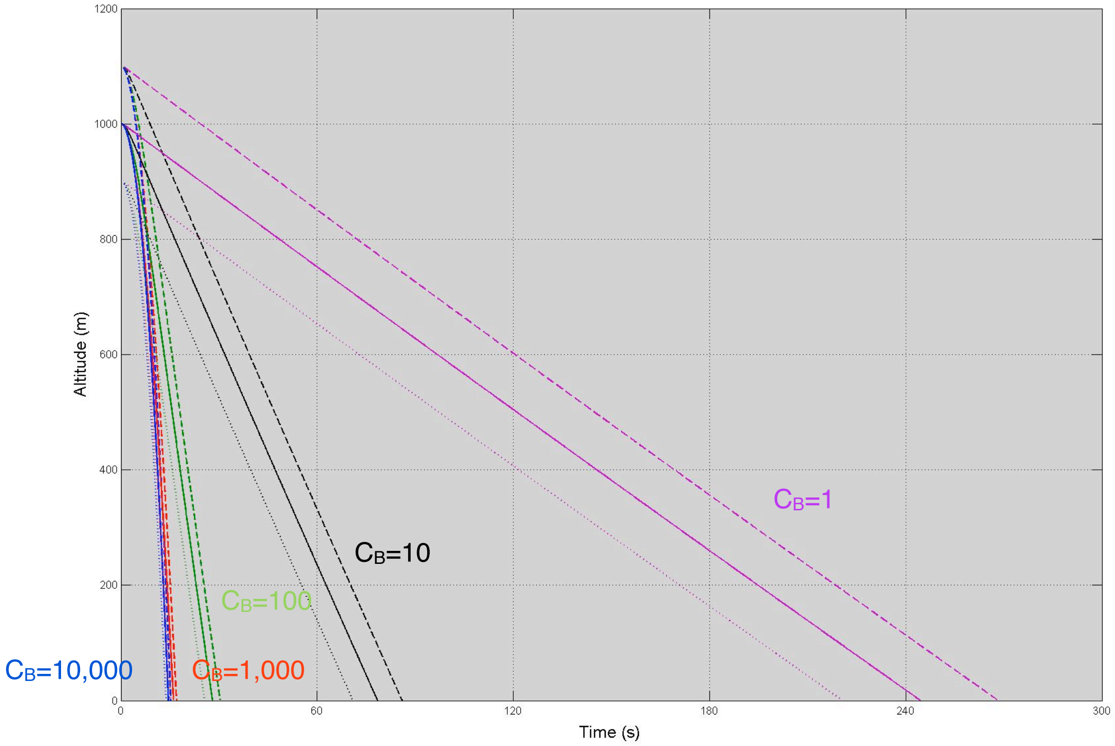


Figure 9.5b - The effect on time to fall to ground of changing breakup altitude for a simulated small aircraft accident

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

Initial velocity variation

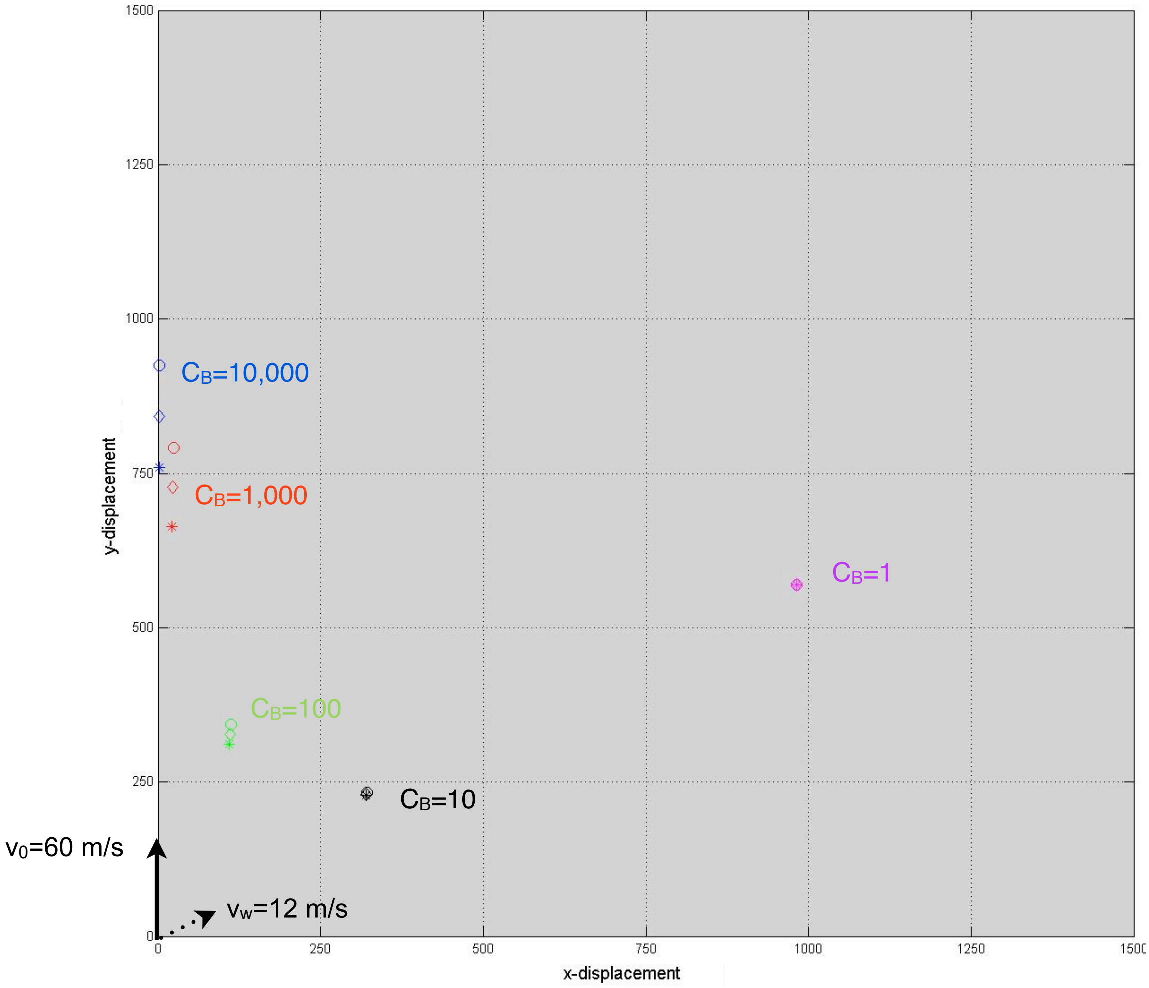


Figure 9.6a - The effect on final wreckage location of changing initial forward velocity for a simulated small aircraft accident.
C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1
◇ = reference, O = +10%, ✻ = -10%

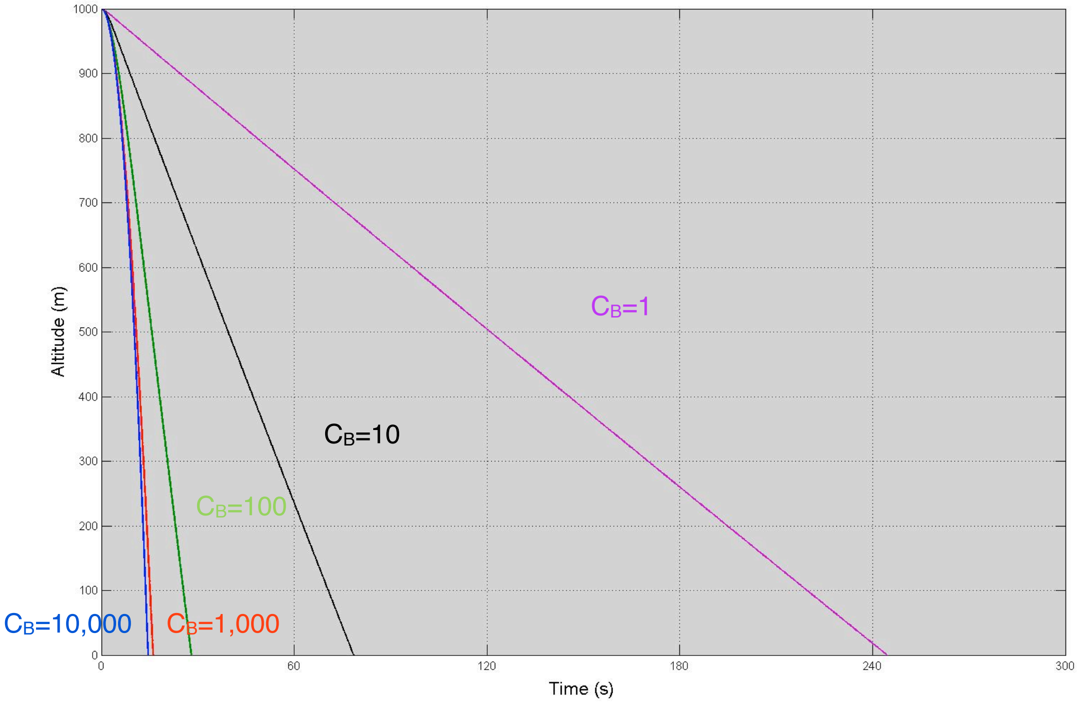


Figure 9.6b - The effect on time to fall to ground of changing breakup forward velocity for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

Wind magnitude variation

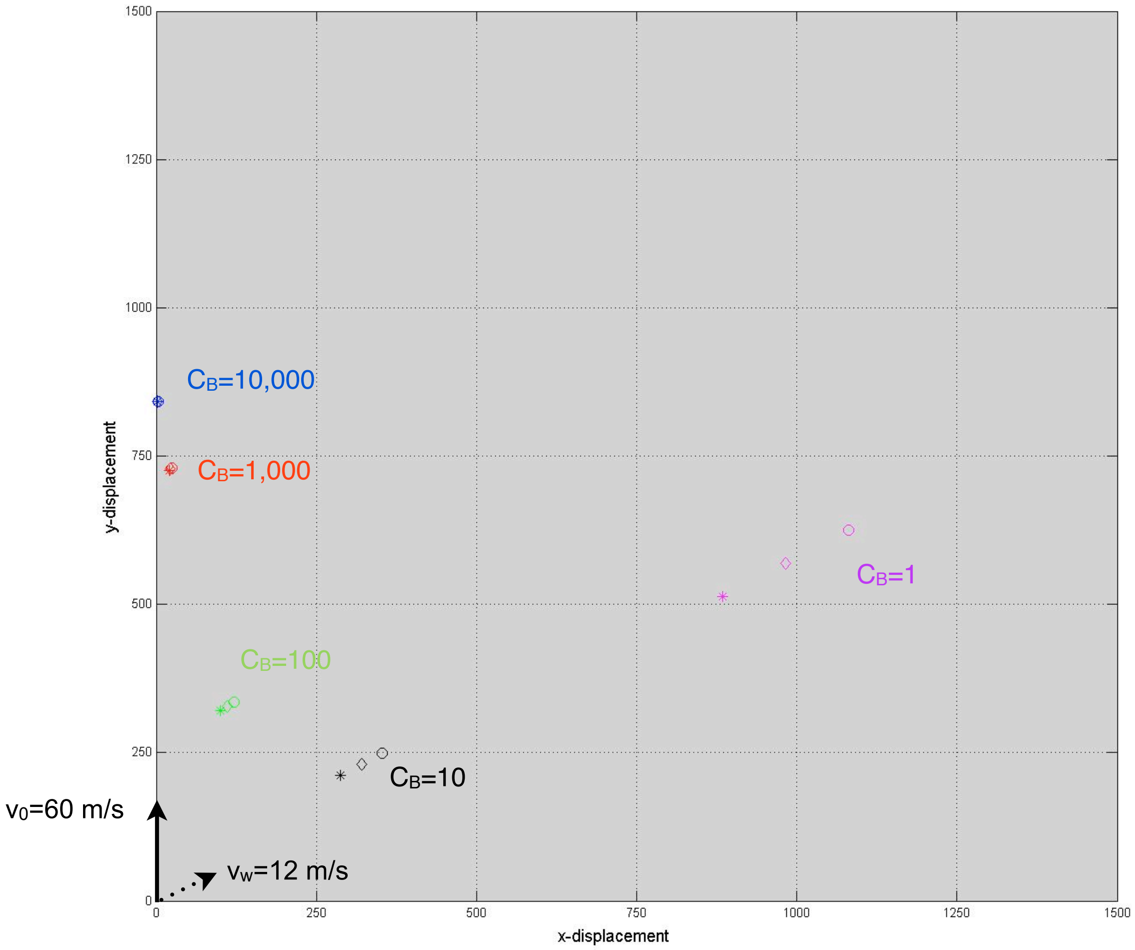


Figure 9.7a - The effect on final wreckage location of changing wind magnitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇= reference, O = +10%, ✻ = -10%

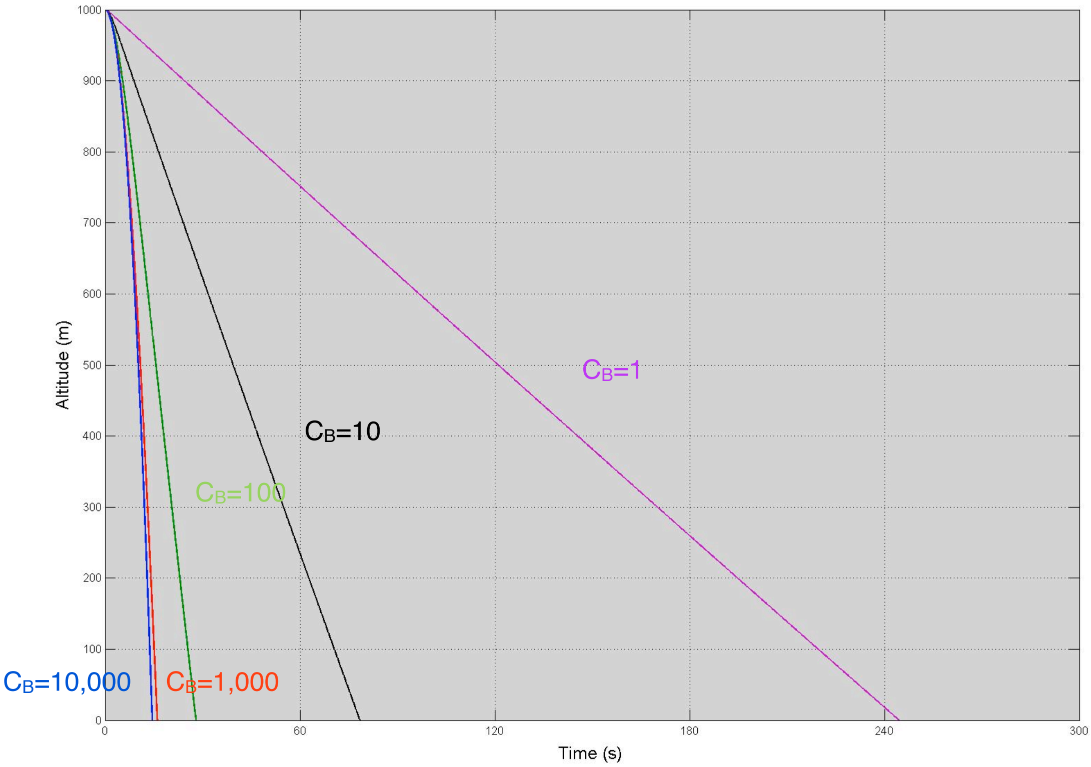


Figure 9.7b - The effect on time to fall to ground of changing wind magnitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%,* ···· *= -10%*

Wind angle variation

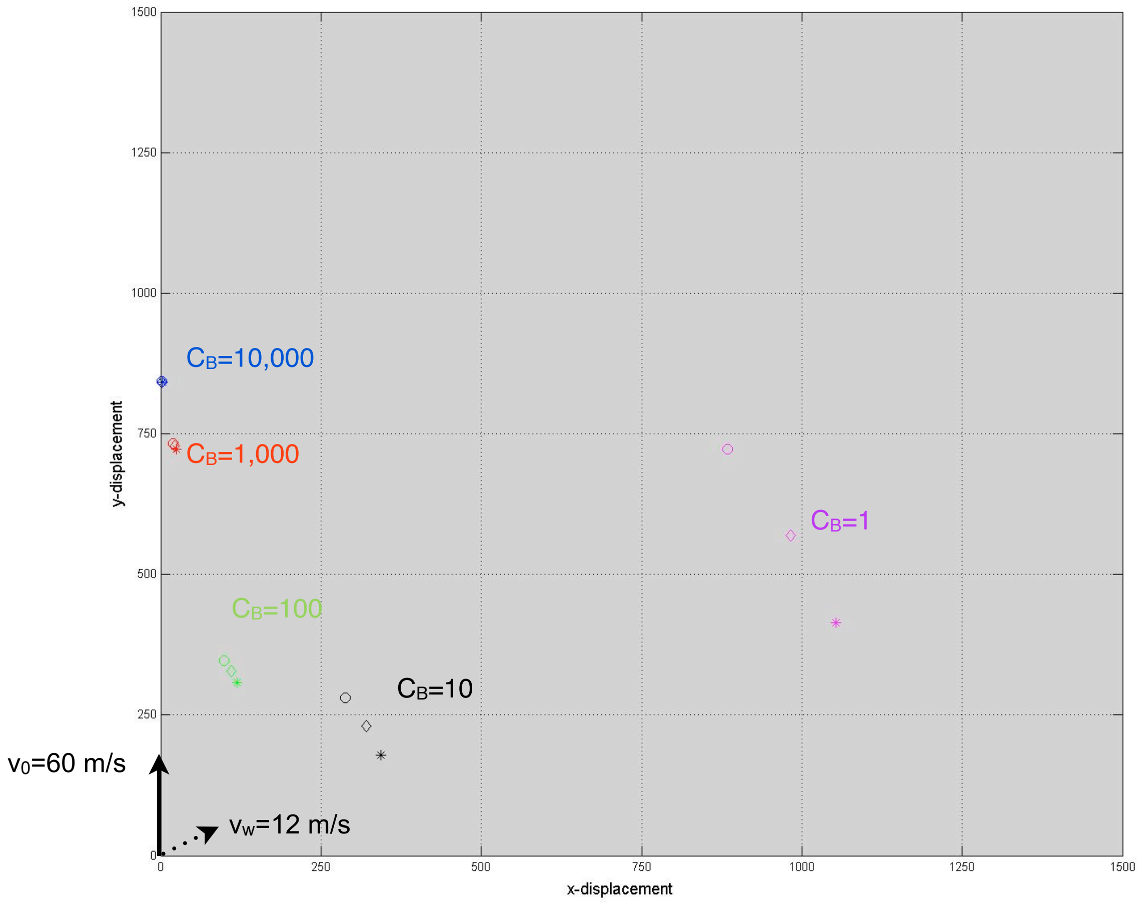


Figure 9.8a - The effect on final wreckage location of changing wind angle for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ❋ *= -10%*

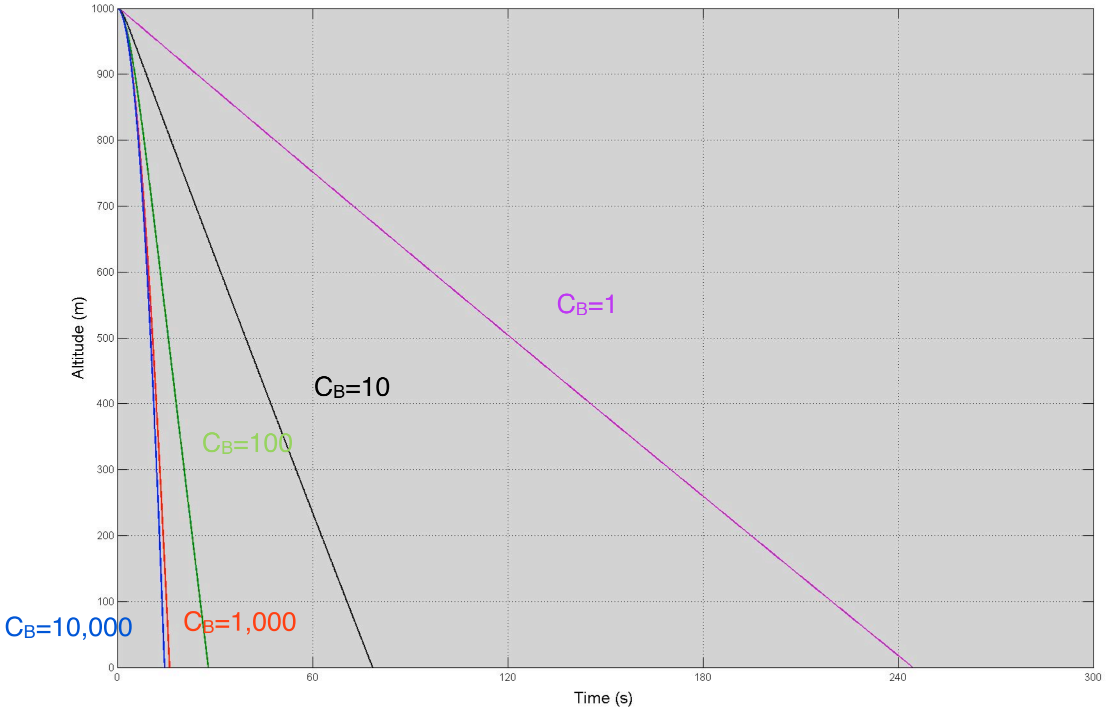


Figure 9.8b - The effect on time to fall to ground of changing wind angle for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ····· = -10%

Appendix 10

This Appendix shows results for the same conditions as given in Chapter 4, but with the wind changed to an angle of -30° from the horizontal (120° measured clockwise from aircraft heading). This results in a cross/head wind.

Large Aircraft Accident - Altitude Variation

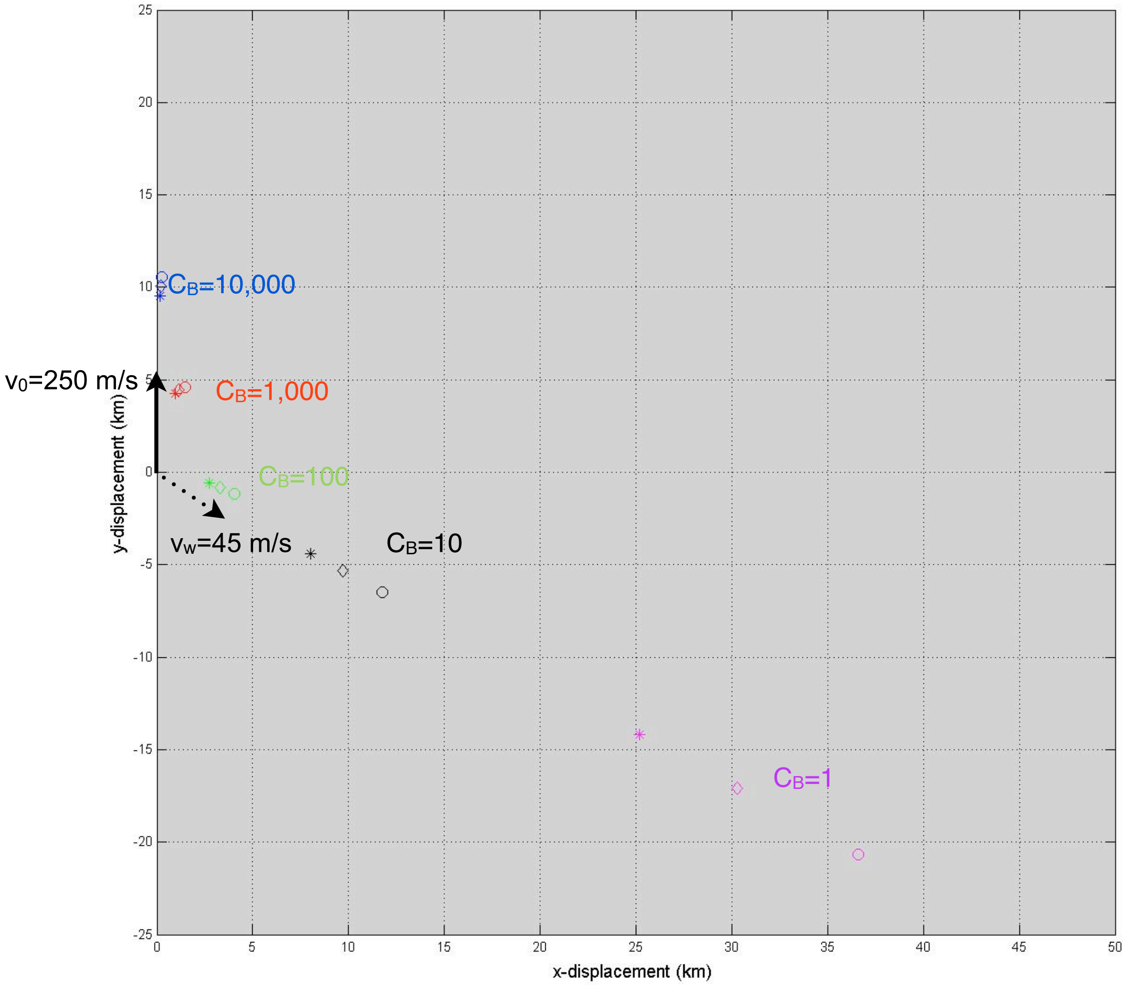


Figure 10.1a - The effect on final wreckage location of changing breakup altitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇= reference, O = +10%, ✻ = -10%

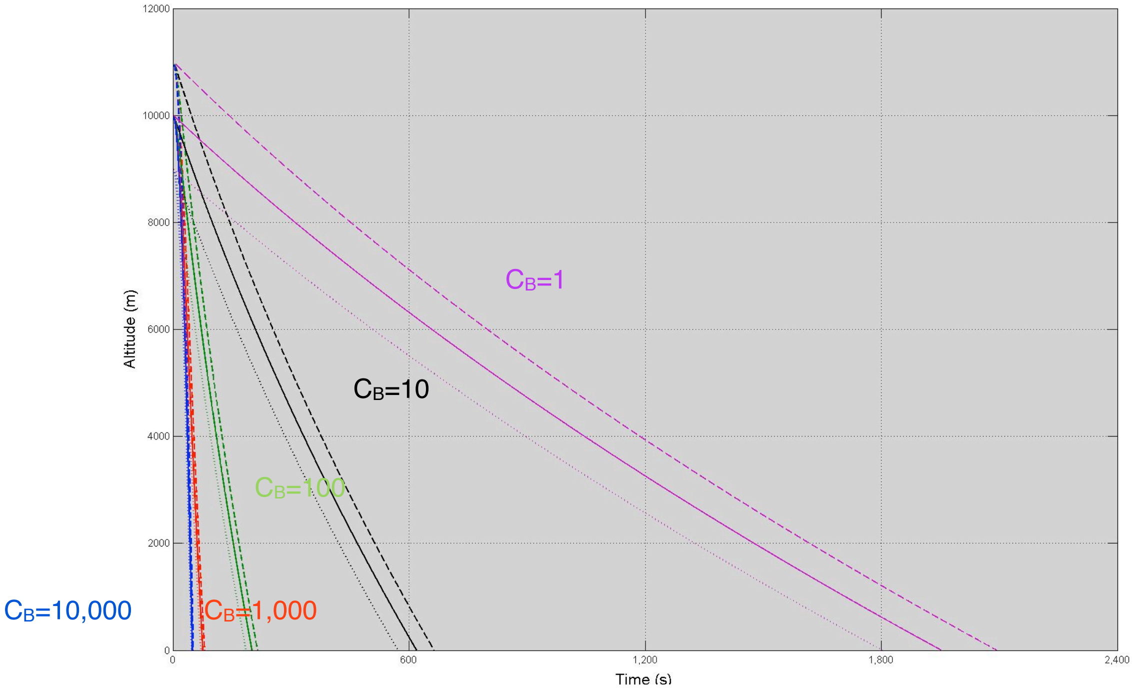


Figure 10.1b - The effect on time to fall to ground of changing breakup altitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ¨¨ = -10%

Initial velocity variation

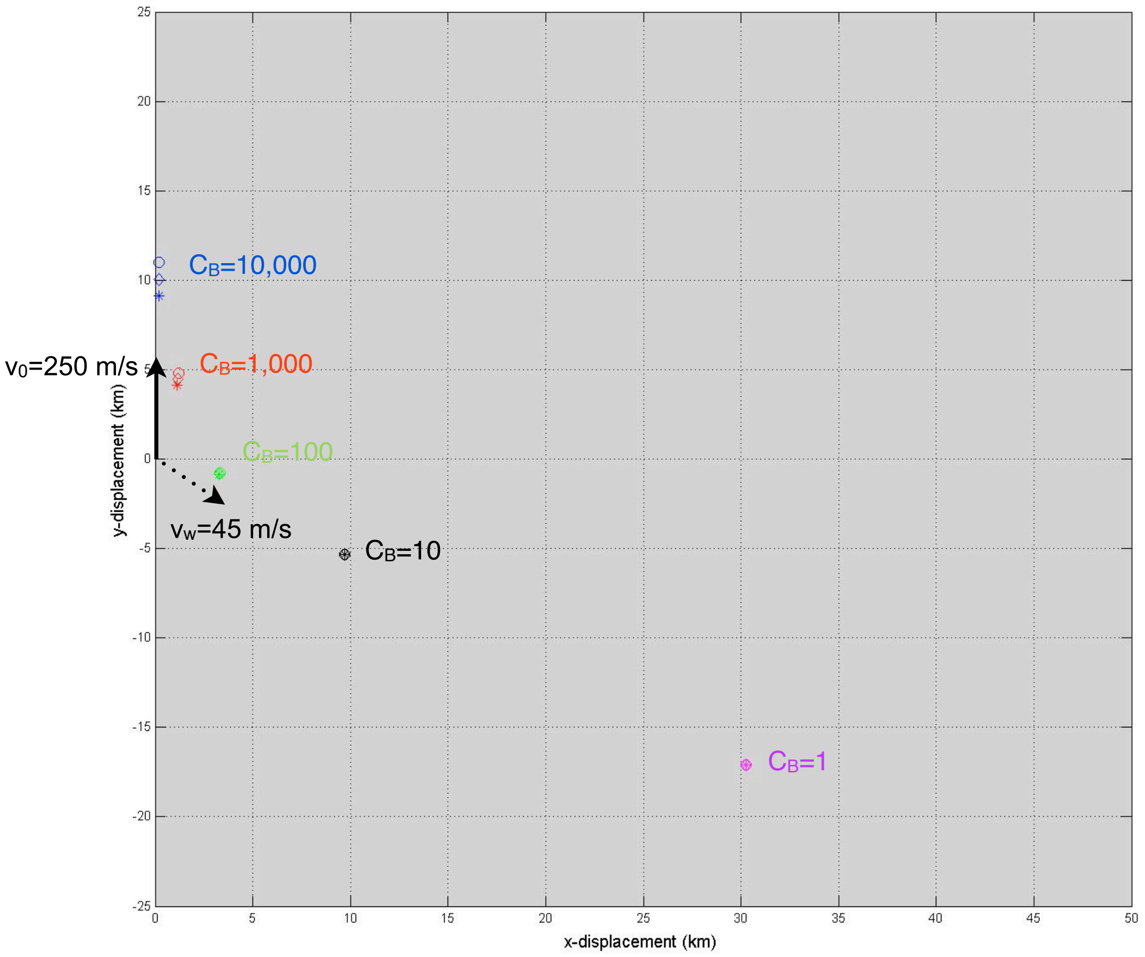


Figure 10.2a - The effect on final wreckage location of changing breakup forward velocity for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇= reference, O = +10%, ✻ = -10%

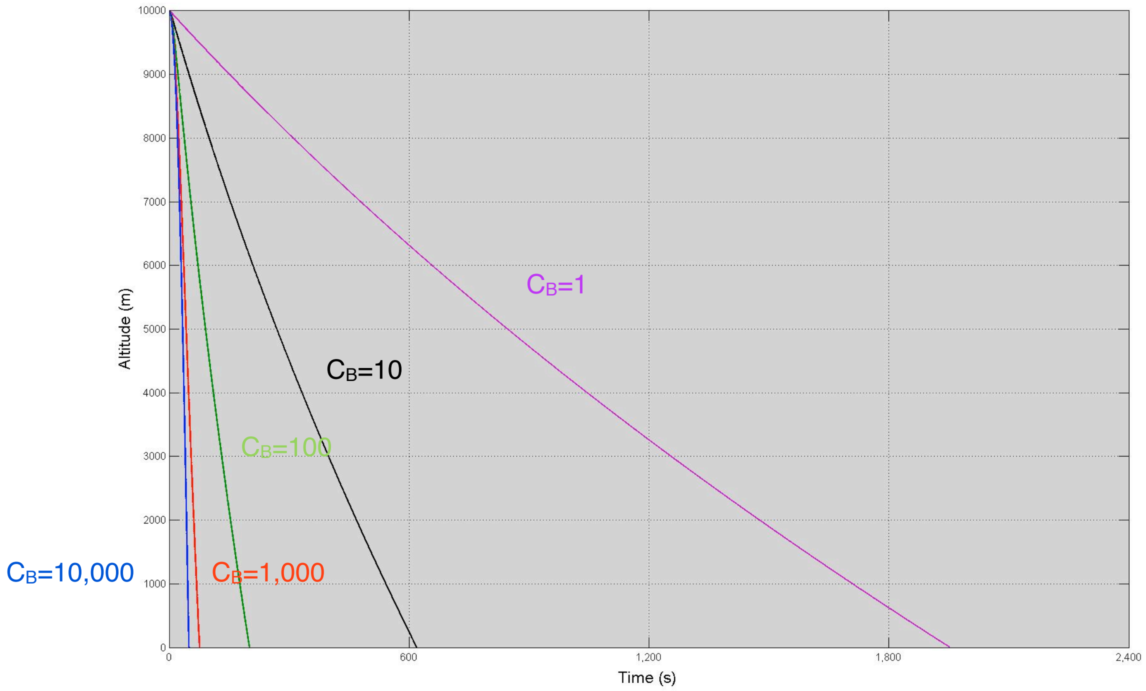


Figure 10.2b - The effect on time to fall to ground of changing breakup forward velocity for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

Wind magnitude variation

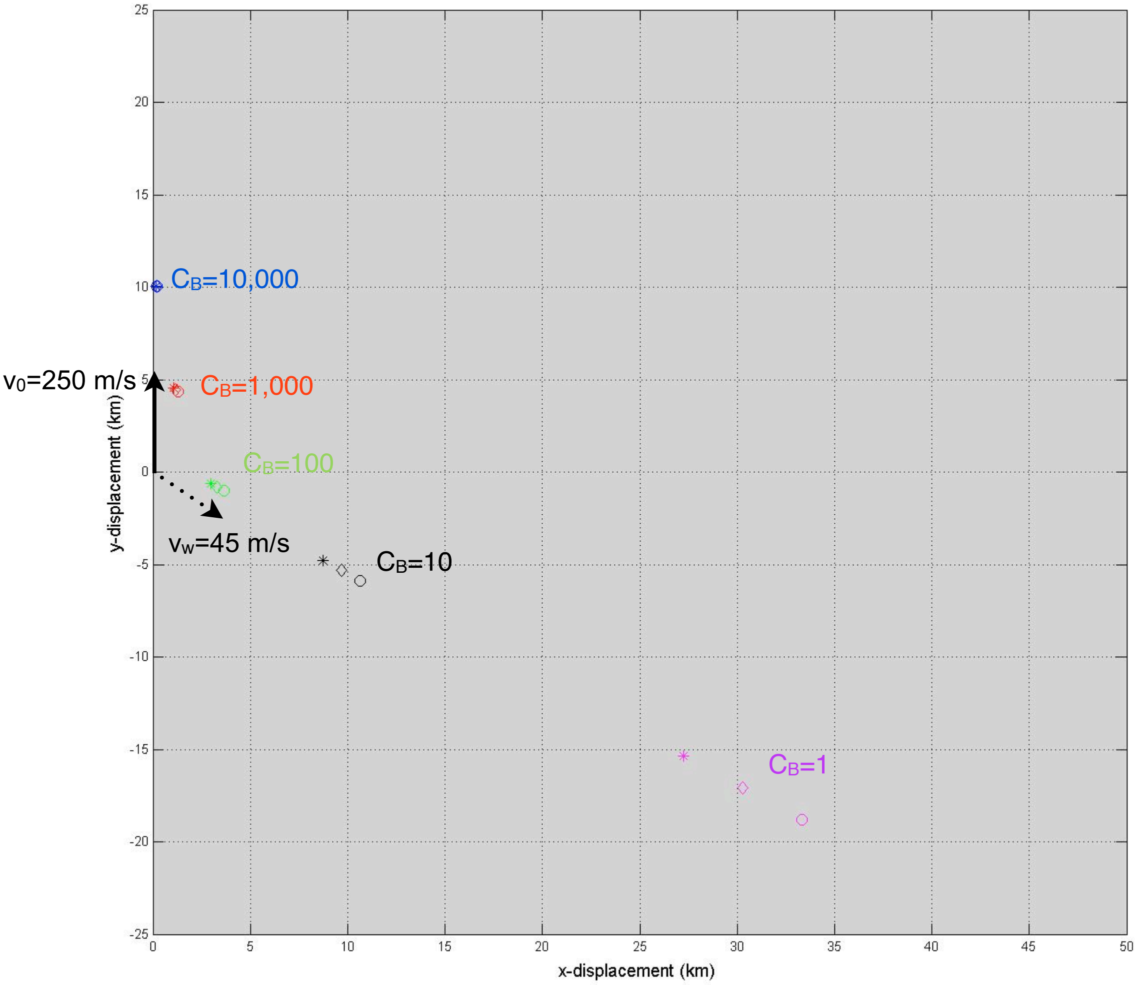


Figure 10.3a - The effect on final wreckage location of changing wind magnitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ = *reference,* O = *+10%,* ✱ = *-10%*

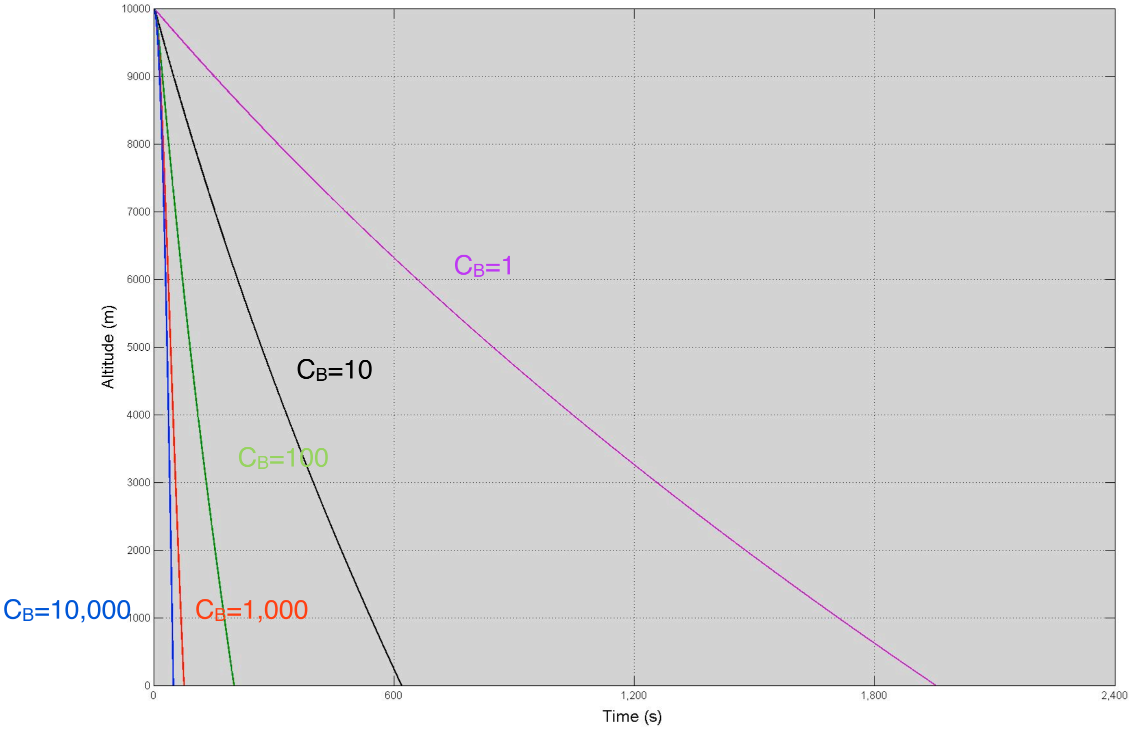


Figure 10.3b - The effect on time taken to fall to ground of changing wind magnitude for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%,* ···· *= -10%*

Wind angle variation

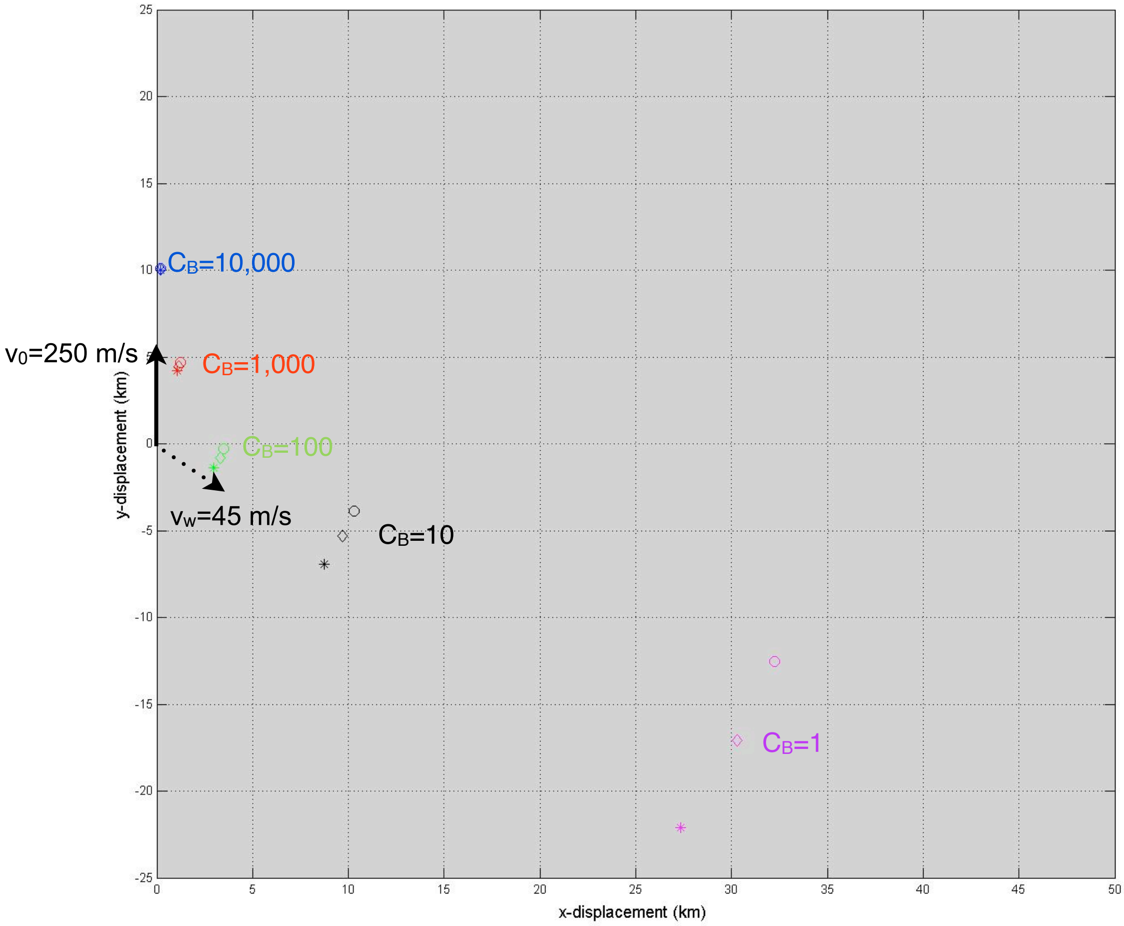


Figure 10.4a - The effect on final wreckage location of changing wind angle for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ✻ *= -10%*

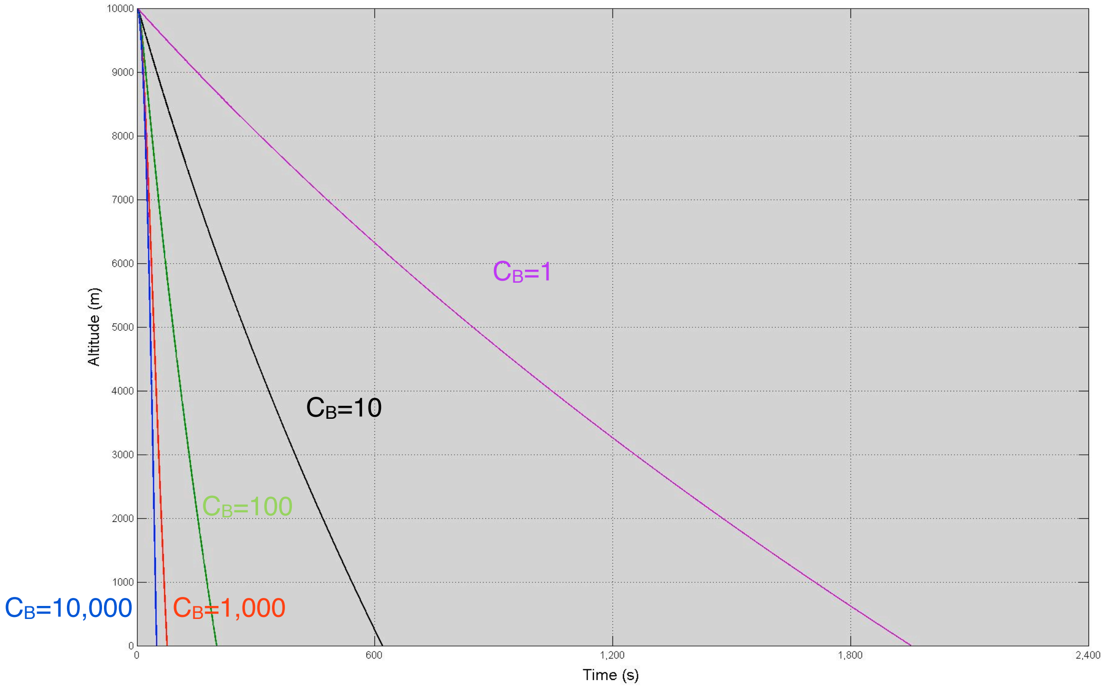


Figure 10.4b - The effect on time taken to fall to ground of changing wind angle for a simulated large aircraft accident..

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

Small Aircraft Accident - Altitude Variation

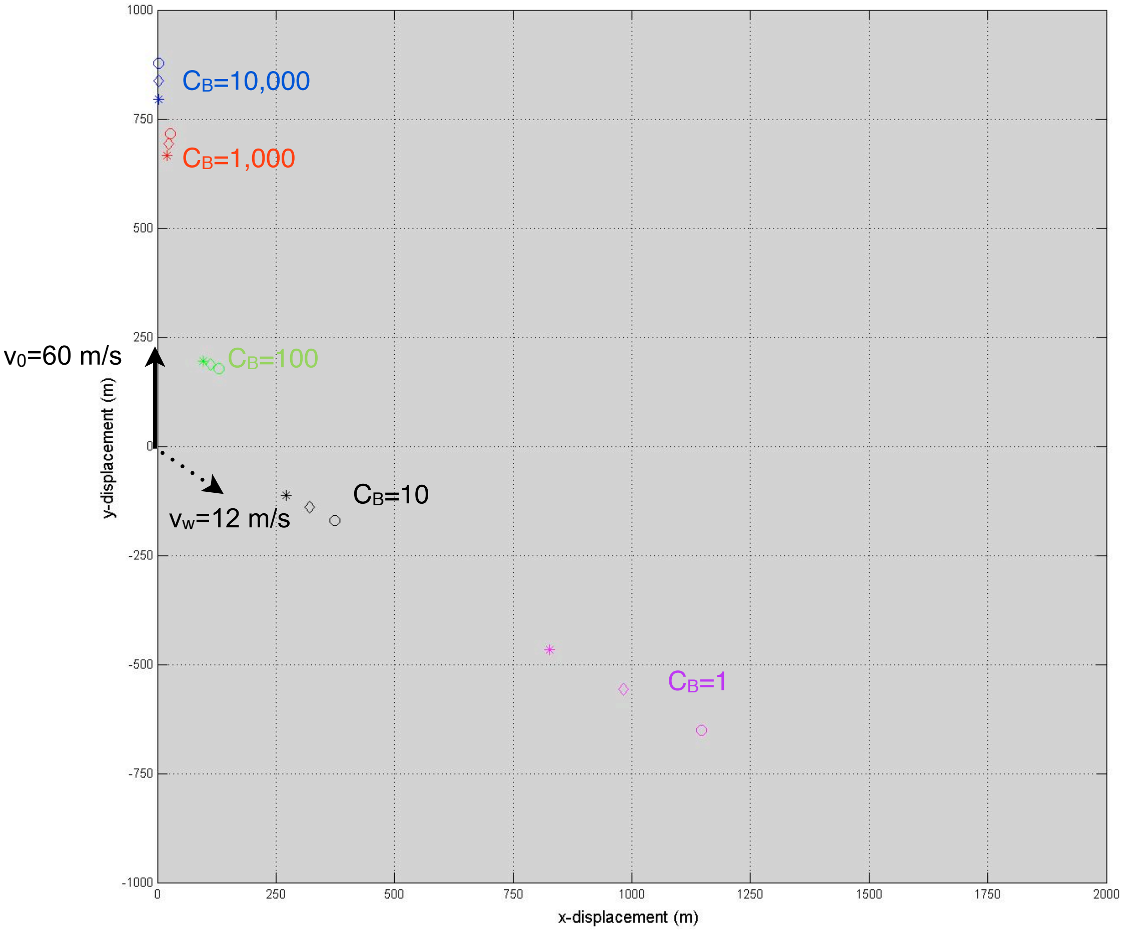


Figure 10.5a - The effect on final wreckage location of changing breakup altitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ❋ *= -10%*

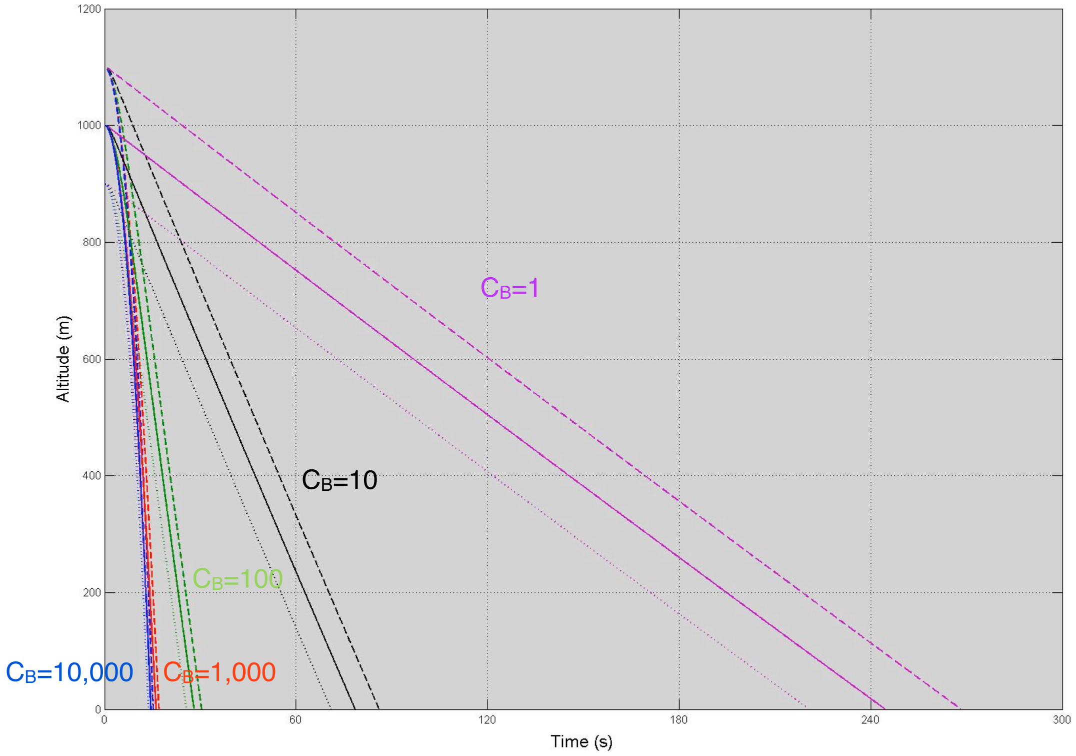


Figure 10.5b - The effect on time to fall to ground of changing breakup altitude for a simulated small aircraft accident

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, --- = +10%, ···· = -10%

Initial velocity variation

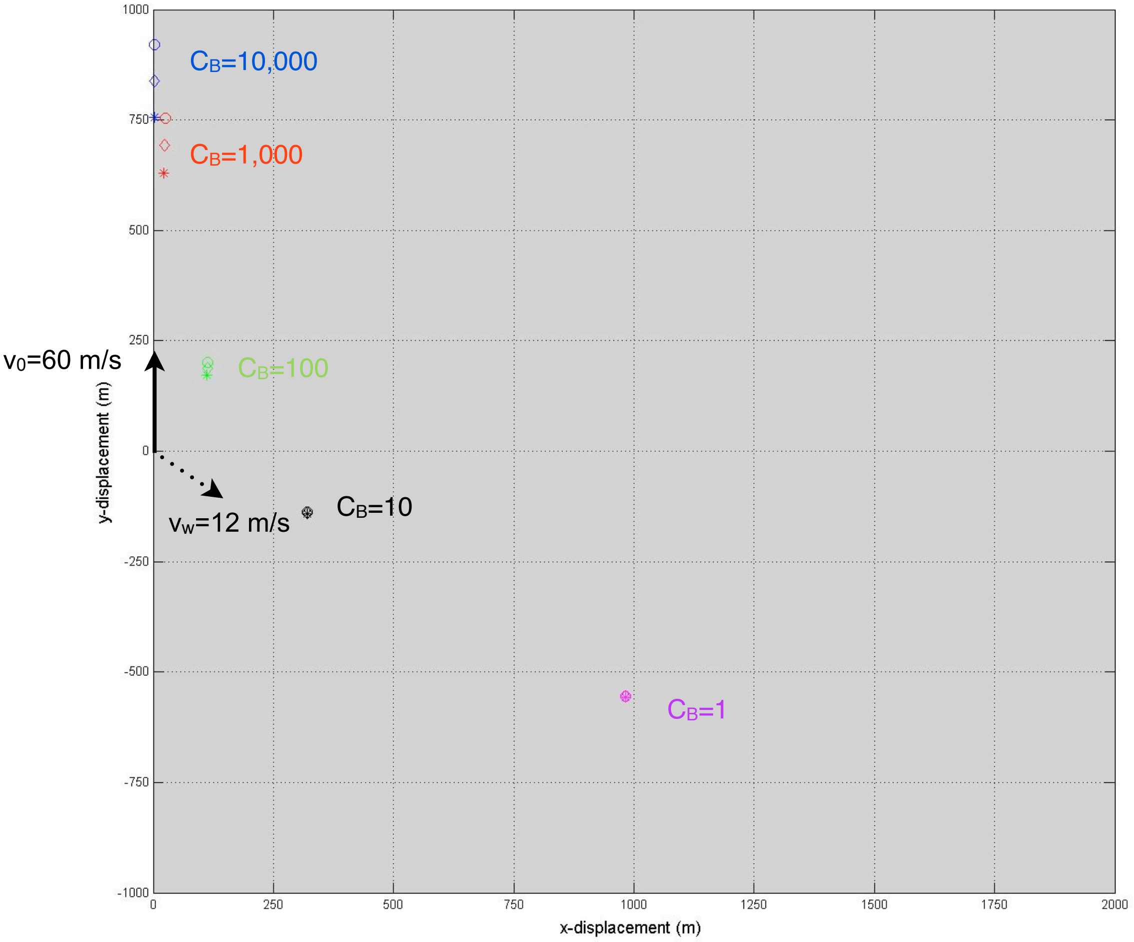


Figure 10.6a - The effect on final wreckage location of changing initial forward velocity for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ *= reference,* O *= +10%,* ❋ *= -10%*

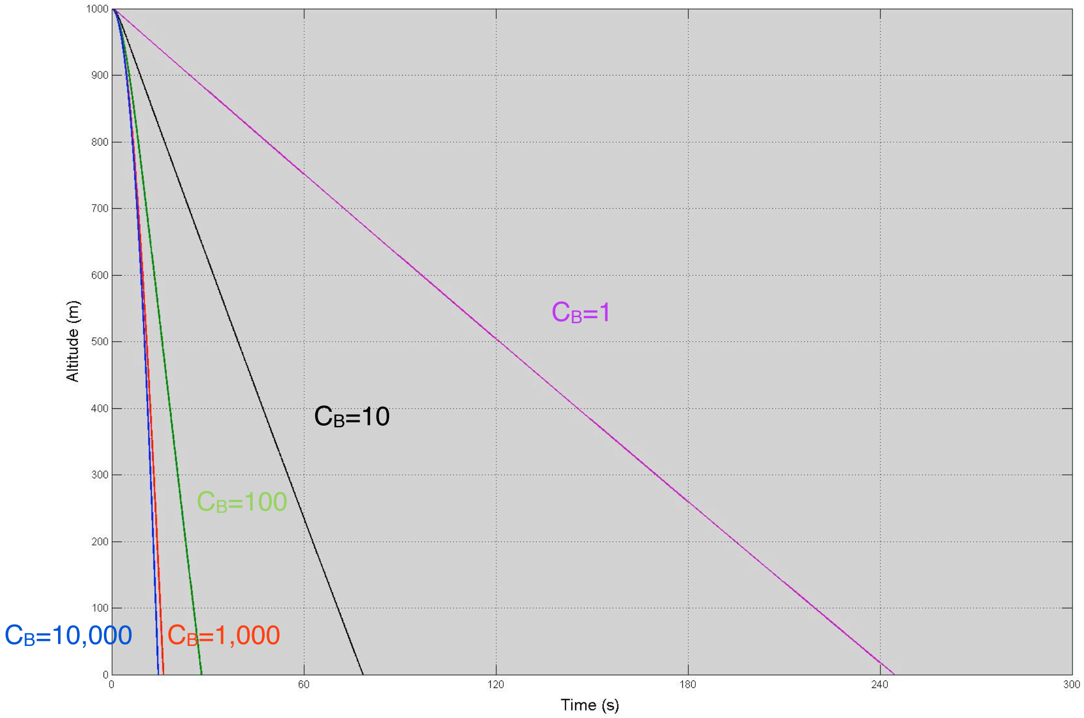


Figure 10.6b - The effect on time to fall to ground of changing breakup forward velocity for a simulated large aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%, ˙˙˙˙ = -10%*

Wind magnitude variation

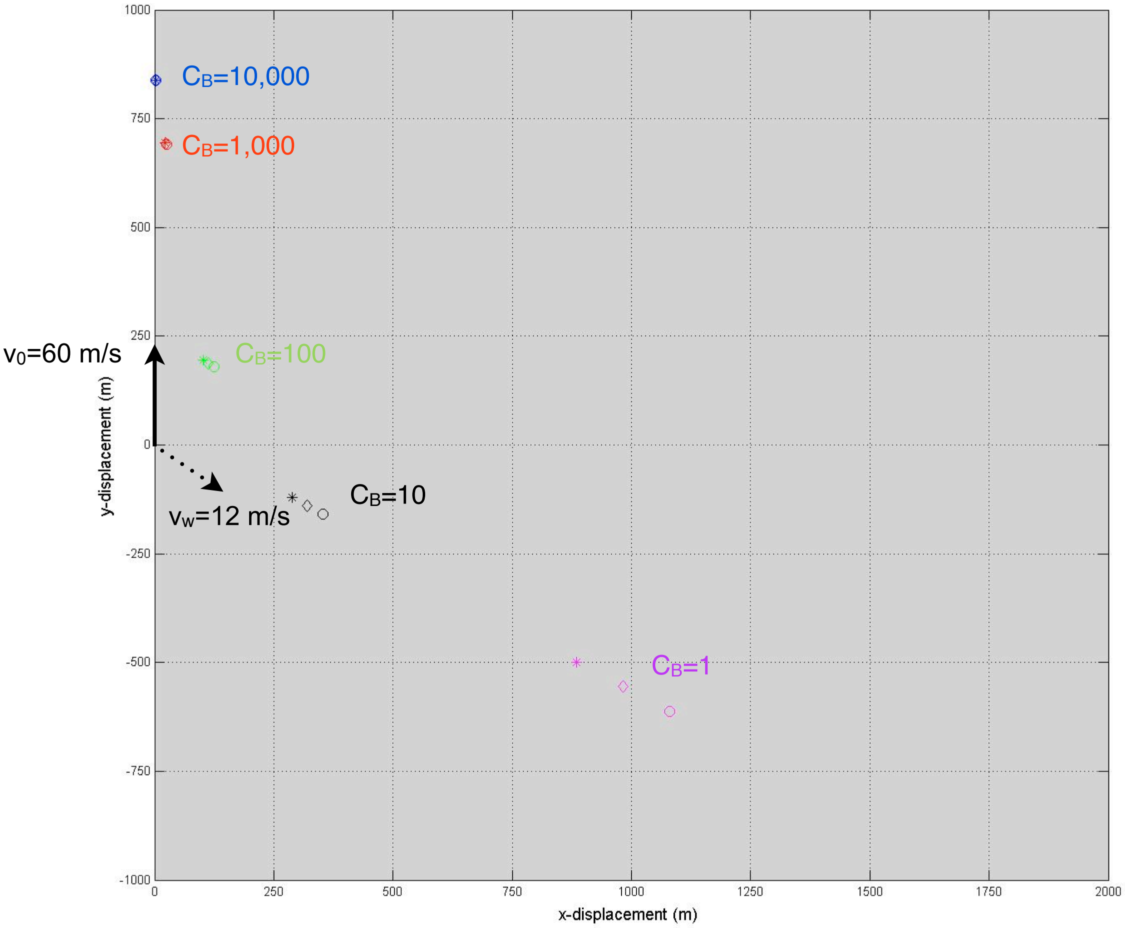


Figure 10.7a - The effect on final wreckage location of changing wind magnitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇= reference, O = +10%, ✻ = -10%

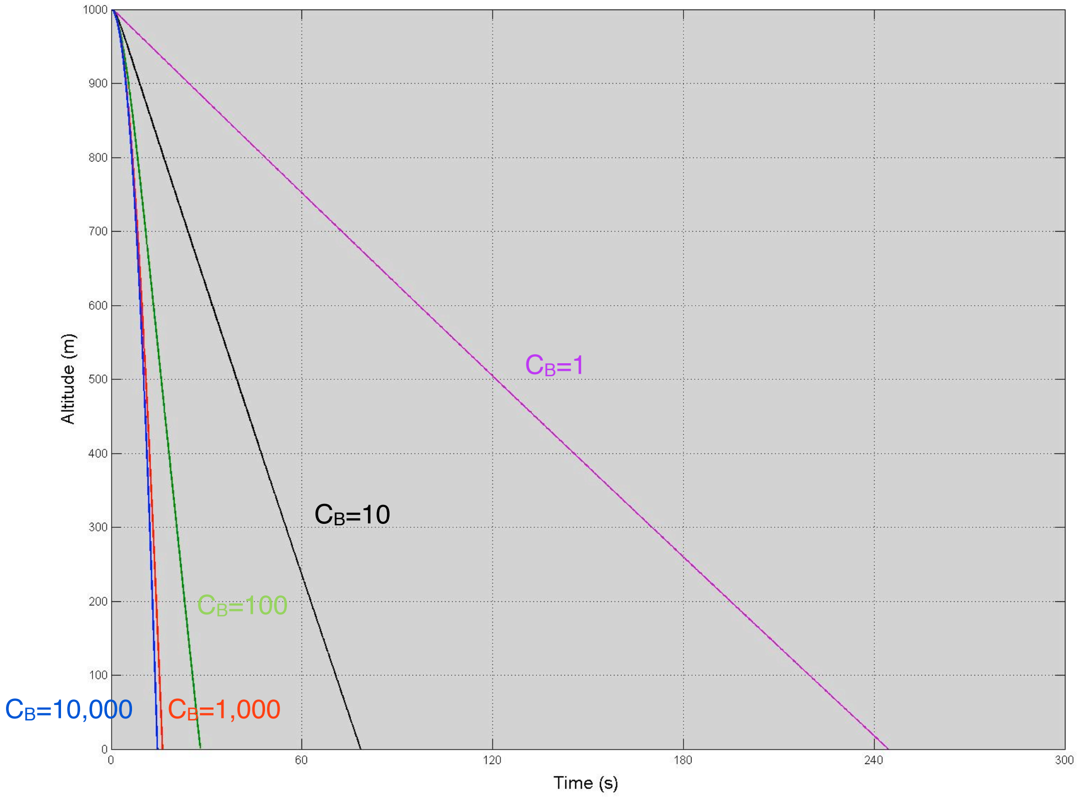


Figure 10.7b - The effect on time to fall to ground of changing wind magnitude for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%, ···· = -10%*

Wind angle variation

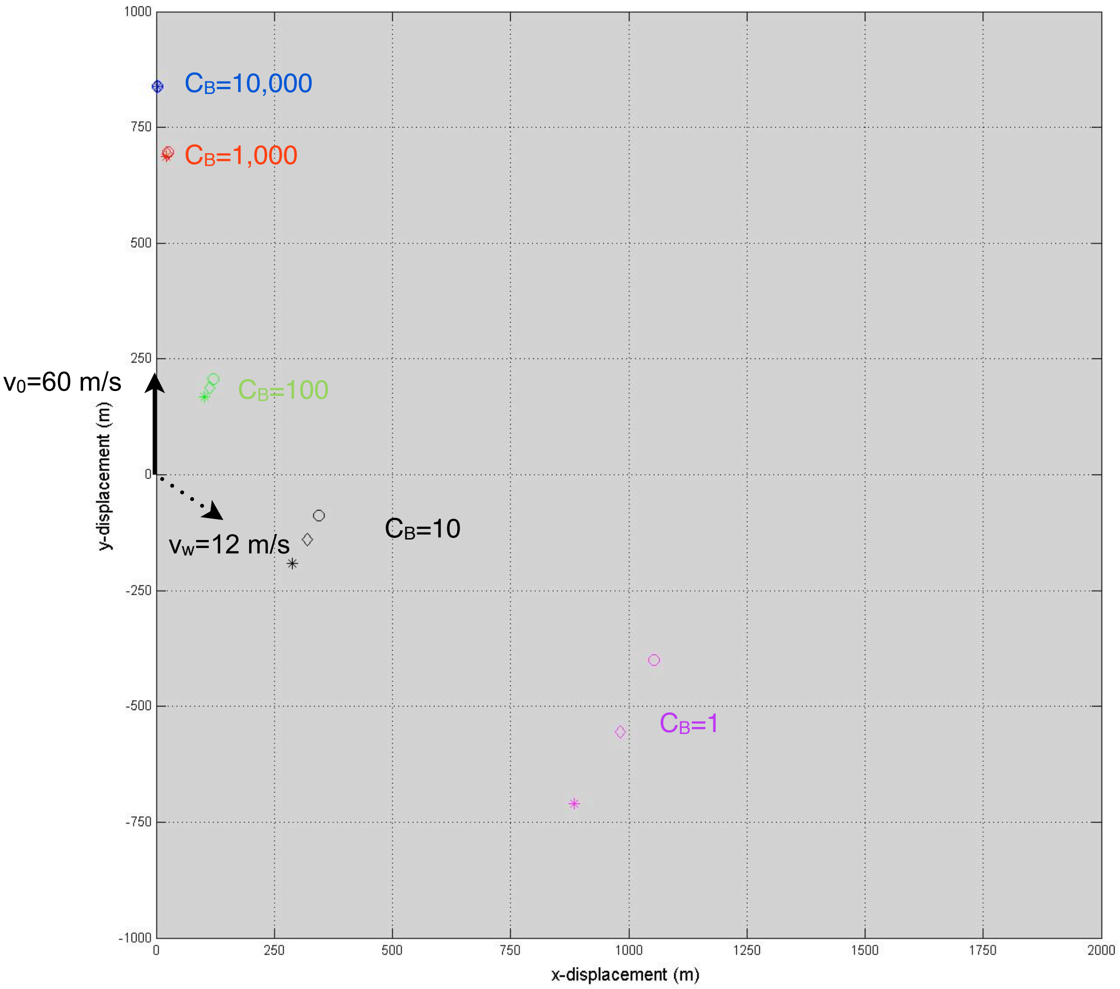


Figure 10.8a - The effect on final wreckage location of changing wind angle for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

◇ = reference, O = +10%, ❋ = -10%

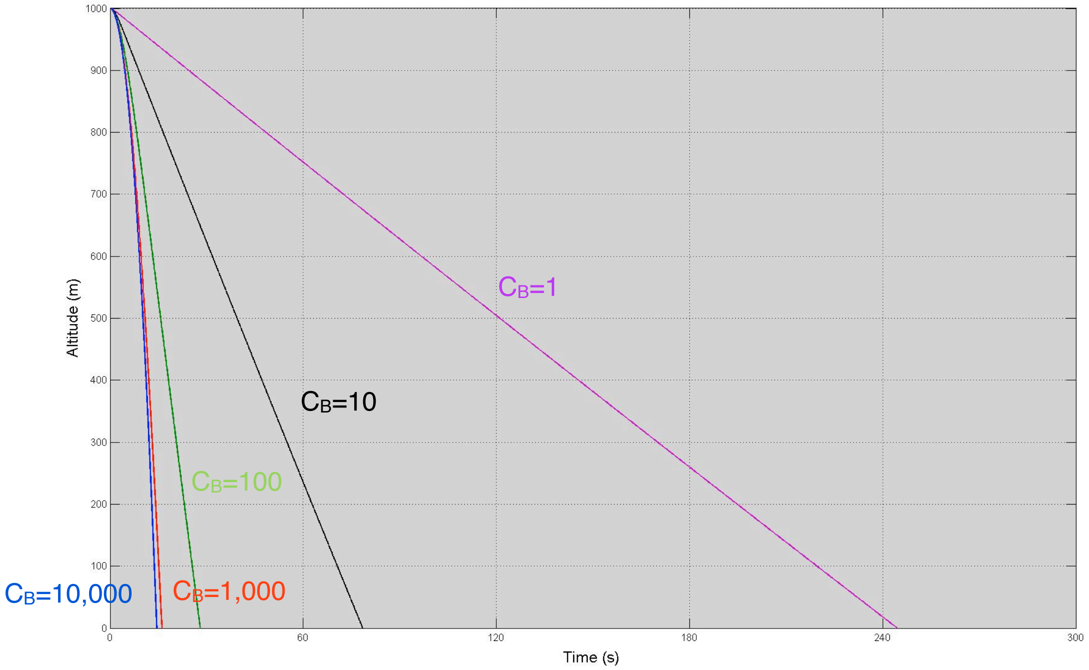


Figure 10.8b - The effect on time to fall to ground of changing wind angle for a simulated small aircraft accident.

C_B=10,000 ; C_B=1,000, C_B=100; C_B=10; C_B=1

—= reference, **---** *= +10%, ···· = -10%*