Example Simulations

This chapter presents several examples of the use of CompuRoc in model rocketry calculations. The examples correspond to the simulation documents found in the 'Examples' folder on the CompuRoc distribution disk. These illustrative problems are intended to further familiarize beginning users with the use of CompuRoc, and also to give some ideas for the possible R&D and special project applications for which CompuRoc can be used.

• Mass Optimization Example

Often the only strategy needed for getting the highest altitude from a rocket is to reduce both the drag and the mass as much as possible. In a drag-free situation (e.g. a vacuum), less mass is always best. In general though, there exists an optimum mass for a given engine, below which the gains realized in increased thrust/weight ratio are more than offset by increased drag deceleration during coast. If this optimal mass is greater than the mass of the engine, it makes sense to try for a liftoff mass as close as possible to this value.

In this example calculation, the optimal mass is found for a rocket powered by a single C6 engine, assuming two different body tube diameters. Parameters for this example are found in the example document named 'Mass Optimization'. As can be seen in the plot of Fig.9, the smaller the body tube (that is, the lower the drag), the smaller is the optimum mass. In effect, the draggier BT-50 rocket needs more mass during coast to "plow" through the air. You may want to experiment with other drag parameters or with different thrust curves to gain more insight into optimizing rocket mass.

ig. 9 - Altitude versus Rocket Mass

• Thrust Optimization Example

Just as there exists an optimum mass for a rocket under the influence of aerodynamic drag, there is also an optimal thrust level. For a fixed total impulse, one can (in theory, anyway) choose to burn fast at a high average thrust, or to burn at a lower thrust level for a longer time. Again looking at the drag-free situation, it is best to burn as quickly and at as high a thrust as possible. In the presence of drag however, it is more efficient to plow through the air at some lower thrust, avoiding the drag penalty associated with higher speeds.

ig. 10 - Altitude versus Average Thrust (20 N-s)

Figure 10 contains the results of CompuRoc simulations of the flight of a D-powered (20 N-sec) rocket using a range of different thrust levels. The files needed to reproduce this example are found in the examples subfolder named 'Thrust Opt. Folder'. Two different initial masses are plotted, and the thrust curves used are idealized "boxcar" profiles with a constant thrust level throughout the burn.

As can be seen from the plot, lower thrust levels are generally preferred, although naturally there exists some minimum thrust value below which the performance drops because the rocket's thrust is approaching the amount needed just to balance its own weight. In the case of the heavier (170 gram) model, the optimum D-class engine would have an average thrust of about 8 Newtons.

In practice, thrust curve profiles must be selected from the available commercial engines, so the results of this idealized study may not be completely applicable. Also, the effects of wind and/or launch angle may be very significant. An interesting possible extension of this kind of study would be an examination of the different shapes of thrust curves available. For example, the relative merits of "regressive" and "progressive" thrust profiles can be evaluated in this way.

• Elevation/Temperature Effects

An interesting question which often arises is just how much do atmospheric variables like temperature and elevation affect a particular model's altitude performance? Such "what if" analyses are easy to perform with CompuRoc, as this example illustrates.

The bar graph of Fig. 11 shows the altitude reached under different circumstances by a given single stage rocket under the power of a D12 engine. The corresponding CompuRocsetup is found in the example document named 'Elevation & Temperature'.

ig. 11 - Altitude versus Atmospheric Variables

The baseline case plotted on the left side is for launch from sea level at an ambient temperature of 20° C. This case assumes no density decrease with altitude. The second case is the same except that it uses the "standard lapse" atmosphere model. The difference, for a flight to only about 300 meters, is so small as to be negligible. Not negligible however, is the difference between flying on a cold winter day, and in hot summer temperatures. As the graph indicates, the decrease in air density with increasing temperature makes for a quite noticeable effect in this model.

Even larger however, is the effect of launching from high starting elevations, as shown by the last two bars. This particular model has a quite distinct advantage when launched from the elevation of, for example, Denver and even more of an advantage for higher launch sites.
Besides the educational value of exploring the effects of atmospheric variables, this kind of calculation is potentially useful for "fine-tuning" altitude predictions for a particular launch site's conditions. While on any particular flight, other "uncontrolled" variables like engine variations may wind up having a larger effect that the atmospheric variables, knowing the atmospheric corrections accurately will help the predictions to be more accurate on the average.

• Drag Coefficient Determination

One of the most obvious applications for CompuRoc is the determination of rocket drag coefficients from experimental altitude data. In principle, the determination is quite straightforward; simply compare the measured altitude with the CompuRoc simulations for a range of trial drag coefficients. In practice however, the accuracy with which the drag coefficient can be pinned down is limited by the flight-to-flight variability in measured flight performance. The purpose of this example is to give some idea of how well constrained the drag coefficient is by typical tracking data.

ig. 12 - Altitude versus Drag Coefficient

The example calculation illustrates the case of a typical egglofter rocket, powered by two different popular D-class engines. Figure 12 plots peak altitude versus drag coefficient for C_D ranging between 0.55 and 0.80. The corresponding CompuRoc simulation files will be found in the example document named 'Drag Coeff.'. If we knew the exact altitude for vertical launch in still air and the exact thrust curve of the engine, we could read the value of C_D right off these curves. In practice though, there will be a good deal of random variation in the "fixed" parameters describing launch conditions and engine performance.

Suppose that after a half-dozen flights, we have pinned down the peak altitude to plus or minus 15 meters. For the D12 case, the drag coefficient would be uncertain by about ±.08, which is over 10%. The D8 case is somewhat less sensitive to altitude uncertainty, but still leaves C_D uncertain by ±.05.

The moral of this story is that using flight data to calibrate a rocket's drag coefficient precisely requires a very tight handle on random altitude variations. While this method will be useful for getting C_D to ten percent or so, more precise determinations will probably require wind tunnel measurements (or other methods like drop testing).

• Delayed Staging Optimization

It is generally known that two-stage rockets reach higher altitudes if the second stage ignition is delayed by some amount. The reason for this is much the same as the explanation of why moderately low-thrust engines generally outperform high-thrust engines of equal total impulse. Burning the second stage motor immediately on first stage burnout maximizes the velocity and also the drag. Delaying the ignition somewhat allows the rocket to gain coast altitude and bleed off speed, before accelerating again during the second stage. So what is the optimum staging delay for a particular rocket?

ig. 13 - Altitude versus Staging Delay

Figure 13 (and the corresponding simulation in the 'Delayed Staging' example document) examines this question for a typical two-stage flight powered by a pair of D12 engines. One variable that turns out to be fairly important in this scenario is the launch angle. If it is possible to launch the rocket precisely vertically, then the highest altitude is reached by letting the rocket coast to a fairly low speed before sustainer ignition. In this particular case, about 3 seconds is the optimal delay for a vertical launch.

It is fairly easy to imagine however, that if the launch is not vertical, a relatively long delay between stages will allow the rocket to "turn over" and ignite the second stage aiming at a low angle. This will severely reduce the peak altitude, and may even be unsafe since in an extreme case the sustainer may ignite pointing down! With this in mind, Fig. 13 also shows a plot for the same rocket launched at 10° from vertical.

The severe penalty for excessively long staging delays is evident in this plot, and the optimum delay is reduced to somewhere in the vicinity of 1.5 seconds. Considering the practical difficulties of achieving perfectly vertical flight (even without wind), this second case would probably be better to use in planning an actual flight. Interested users might want to explore wider ranges of rocket parameters with and without wind in order to get a more solid feeling for what the "best" staging delays are likely to be for realistic flight conditions.

• Underpowered Rocket Example

An all too common occurrence on the flight range is the launch of a heavy rocket that has too low of a thrust-to-weight ratio. Often the rocket is so heavy because it contains some valuable payload like a framing camera or telemetry transmitter. When asked about the rocket's flightworthiness, the owner might respond, "Oh, that's no problem -- it's powered by a composite F!". The more experienced flier will immediately ask, "Yes, but an F-what ?". The purpose of this example is to provide some of the "flight experience" needed to avoid pranging that camera (and without the expense of all those composite engines!).

ig. 14 - Permissible Launch Angle versus Liftoff Mass

A long-burning, low thrust engine like the Aerotech F10 can be a very effective powerplant for sending small to moderate payloads to high altitude. A rocket weighing several hundred grams at launch however, has the nasty habit of turning over under power with such a low thrust/weight ratio. As in the delayed staging example, this presents no problem if you can be sure of launching the rocket exactly vertically. The farther from vertical the launch though, the worse the "turnover".

Figure 14 is a plot of the calculated maximum launch angle as a function of liftoff mass. This maximum permissible angle was determined by running CompuRoc for an increasing succession of launch angles until the rocket went horizontal under thrust (the simulation documents are found in the 'Heavy' example subfolder). As can be seen from the plot, any rocket in the weight range of a pound or more is very difficult to keep from powering in on an F10. This criterion for "permissible angle" is actually pretty liberal, and a more realistic safety limit might be to keep the thrust vector within say, 30° or 45° of vertical during burn. Sometimes overweight rockets don't actually power in but are still damaged because of recovery system problems associated with their "depressed trajectory".

The example presented here is admittedly a little bit absurd, and some might argue that, "No one would ever really try to launch a 500 gram model on an F10..." (wanna bet?). Nevertheless, it is instructive to see quantitatively just how bad an idea it is, and similar simulations could be used to analyze borderline cases in which it might not be clear whether a particular thrust level was safe or not. There's lots of room here for further exploration of the question of what a "safe" flight is, including maximum safe parachute deployment speed and the like.

• Wind Compensation Example

A classic problem faced by most every flyer at one time or another is the question of how best to compensate for wind in trying for the highest altitude. Aiming too little downwind may cost altitude due to "weathercocking", while aiming too much downwind risks overcorrecting in the opposite direction and reducing the chances of successful tracking and recovery. This example is an analysis of the wind compensation problem for a C6 powered rocket given different wind speeds (the simulation documents are found in the 'Wind Compensation' example subfolder).

ig. 15 - Altitude versus Launch Angle in Wind

The cases plotted in Fig. 15 are for wind speeds of 6 and 10 meters per second (see the Appendix for conversion factor to miles per hour). With the gentler wind speed, aiming downwind by about 15° gives the best altitude performance. It is interesting to note how large the effect of an angle change of only five or ten degrees can be. In the case of somewhat stronger winds, the optimal launch angle increases significantly to around 25°, while the highest achievable altitude is slightly reduced. (Note: while CompuRoc simulations sometimes indicate the desirability of large launch angles or extreme values of other launch parameters, modelers are urged always to operate within the limits of the NAR safety code.)

Exploring a range of simulations similar to this one could potentially be very useful in optimizing altitude flights for wind conditions. One matter of some significance here (as mentioned earlier in this manual) is the question of launch rail length. In this particular example, the rocket is sufficiently light and quick to accelerate that a launcher length of one meter is reasonable. Other, slower rockets may require some "playing with" the launcher length in order to realistically simulate the early flight profile during which the rocket is orienting itself with respect to the relative wind. In the absence of detailed dynamic stability information, the modeler's own flight experience and intuition will be most valuable here. The specified value of launcher length should reflect the altitude at which the particular rocket "settles down" in an equilibrium attitude with respect to the prevailing wind. For some heavy and/or long rockets this altitude can easily be tens of meters or more.

There are many other possible combinations of variables to explore in this area, such as the effectiveness of different thrust profiles or initial masses in varying wind conditions. Application to trajectory planning problems (e.g. Spot Landing events) are also good candidates for exploration. An interesting possibility is defining a "delay" with a huge effective area and drag coefficient, to simulate the descent of a rocket on a parachute or streamer.

• Retrobraking Example

This example concerns the problem of model rocket recovery by retrorocket braking. Many flyers have wondered from time to time about the feasibility of using forward-facing engines to cancel a rocket's downward velocity and achieve a soft landing. Some years ago I witnessed some moderately spectacular failures in attempted retro-recovery, and concluded that the method wasn't practical as a stand-alone recovery method. Here, we'll look at just how hard it is to make retrobraking work.

ig. 16 - Retrobraking Altitude/Velocity History

The plot of Fig. 16 is a time history of the final few seconds of the flight defined by the documents in the example subfolder named 'Retrobraking'. This flight consists of an ordinary single stage profile with the addition of a "negative burn" which occurs about 17 seconds into the flight. As noted earlier in this manual, the negative thrust effect is achieved by specifying a "cluster factor" of -1.0 in the burn event data dialog.

There are two important factors involved in setting up the retroburn: braking engine impulse and timing. The braking engine impulse is fairly easy to determine. Since the rocket approaches the ground falling at a nearly constant terminal velocity, the total impulse needed to bring the rocket to rest is just equal to its falling momentum (mass ∞ velocity). For this example, the required impulse turns out to be approximately 6 Newton-seconds (or 6 kg-m/sec). Rather than try to match this with a commercially available engine, we have taken the easy way out in this example and defined our own "custom" 6 N-sec boxcar engine that burns for one second.

The trick now is to ignite the braking engine at just the right time to bring the velocity to zero just as the rocket reaches the ground. Ignite too soon, and the rocket free falls from some height; too late and it hits before the end of the burn. The "correct" timing shown in Fig. 16 was arrived at by several trial and error runs. But how critical is this timing? Well, in this case, a change in the ignition time of only ±0.1 sec results in a landing error of 5 meters or 5 meters/sec. (This however, doesn't even include the possibility of such random errors as variability in engine impulse.) Since "errors" much bigger than this would not be tolerable from either a damage or safety standpoint, it seems that pure retrorocket recovery is not practical without a means of reliably igniting the braking engine at a quite precise altitude.

Still, this exercise may be of more than purely academic interest if we don't have to bring the rocket to rest exactly at ground level. We can imagine that it might be desirable under some circumstances to allow a rocket to free fall from high altitude (for example, to avoid a long parachute drift) and then to retrobrake the model at some modest altitude for recovery deployment. Using conventional timing methods, it should be possible to bring the falling model to rest safely within 100 meters of the ground or less, and then drop it the rest of the way on a parachute. This kind of CompuRoc simulation could be useful for planning such a flight profile.

As a final note, some particular care is needed in using CompuRoc for simulations of this kind. It should be remembered that CompuRoc always aims the thrust vector along the instantaneous direction of motion. This means that if a braking burn completely cancels the rocket's motion and starts to move it backwards, CompuRoc will immediately swing the thrust vector around to oppose further motion. In effect, the simulation will "hover", flopping around in mid-air! In practice then, the simulation will only be valid up until the point that the fall is halted (although an actual aerodynamically stabilized retrorocket might really behave something like this). Therefore it is best to run simulations using braking engines that fall slightly short of cancelling all falling momentum so that this bizarre behavior doesn't occur.