Addendum to Chapter 8 of *Applied Ballistics for Long Range Shooting-Second Edition* by Bryan Litz

Chapter 8 of *Applied Ballistics for Long Range Shooting – Second Edition* is titled *Using Ballistics Programs,* and explains basically how ballistic models are used to predict trajectories, and discusses best practices for using ballistic programs. The final section of this chapter begins on page 121 and is titled: *Trajectory Validation, or "Truing"*. It is this section that this addendum applies, and is attached at the end of this addendum for reference.

Supersonic vs Subsonic Flight Regime

There is an issue of context regarding the range over which a trajectory is trued, specifically in relation to where the bullet becomes transonic and subsonic. With the exception of Chapter 11: *Extended Long Range Shooting,* the material in this book pertains primarily to shooting to the extent of the bullets' supersonic range. This context applies to the section in question regarding truing. *In other words, the statements made in the truing section were intended to apply to trajectories where the bullet remains supersonic.* At such ranges, it is usually possible to gather muzzle velocity and BC inputs accurate enough to produce correspondingly accurate trajectory predictions sufficient for most field shooting applications, without truing. In the event that a shooter cannot get sufficiently accurate inputs, a minor level of truing can be useful for bringing the predictions of the program into better alignment with reality. If a large amount of truing is required such that the shooter is using inputs that are obviously false (like a BC that's off by 10%+, or a MV that's off by more than 100 fps) then the shooter stands to suffer from the pitfall described in the book where the trajectory predictions may only be valid for a specific range, and off by a significant amount at other ranges[1]. *It is thereby advisable that the shooter make every effort to use accurate inputs, in order to minimize the amount of truing that's required for supersonic range.*

When a spin stabilized projectile flies thru the transonic and into the subsonic flight regimes, there are many additional variables which do not apply to the supersonic flight regime. Variables that are specific to a rifle barrel, for example, can affect the RPM's retained by the bullet at long range, thereby affecting its transonic and subsonic stability, which in turn can affect the amount of yaw induced drag experienced by the bullet. These transonic stability levels can be very subject to atmospheric variables, and can be difficult to model accurately with modern methods[2]. Because of these and other rifle specific variables, *truing is far more necessary for accurate trajectory predictions thru the transonic and subsonic flight regimes than it is for supersonic.*

---

[1] This addendum is not intended to be specific to a particular program. Different programs do different things behind the scenes to true or validate their solutions, and some are more or less sensitive to error than others.

[2] Magnus moment and spin damping coefficients, as well as pitch and yaw damping derivatives would have to be modeled to a level of accuracy currently not possible with modern aeroballistic prediction methods.

There are a host of deterministic variables which can be accounted for that will bring a prediction closer for transonic/subsonic flight like custom drag curves, Coriolis effects, optics related aiming errors, etc. But because of rifle specific variables, it is not always possible to predict acceptably accurate trajectories beyond the supersonic range of a bullet. *The power and utility of truing is that it accounts for all the variables including those unique to a specific rifle, resulting in more accurate trajectory predictions than would otherwise be possible beyond the supersonic range of the bullet.*

The final paragraph of the attached section likens truing to "falsifying" inputs. This was a poor choice of words because done correctly; the act of truing will make an input *more accurate* than some initial, unknown guess. Again, there are various ways in which truing is practiced, and some are better than others, but most are better than nothing.

Trajectory Validation, or "Truing"

Sometimes a shooter doesn't have access to an accurate chronograph, or is working with a bullet that has an uncertain BC, or has some other uncertainty about the inputs to the ballistics program, but still wants to use the program to predict a trajectory. In the case where inputs are unknown or uncertain, a ballistic solution can be *tweaked* to match some real world data. Some programs have a built in feature to automatically find the BC and/or MV that is required to match your real world drop.

For example, let's say you know your muzzle velocity is 2650 fps, but you don't know the BC for a particular bullet that you're shooting. You can simply take some long range shots at a known distance, note the point of impact, and adjust the BC in the program until the output matches the observed impact. Calibrating a solution like this will result in a prediction that's *better than nothing*, BUT ***it is highly recommended that the shooter make every effort to collect and enter accurate inputs to the ballistic solver.***

In the above example, BC was the only unknown factor. As long as all the other variables are known, in theory the shooter should be able to arrive at the correct BC provided he's testing at a far enough range, and the rifle has the accuracy to resolve drop with a high degree of certainty.

Sometimes shooters don't know their muzzle velocity or the BC but still want to use a ballistics program to get a useful trajectory prediction. This is where a calibration exercise has the potential to be problematic. If you choose the wrong variable to *tweak*, the result can be a solution that's only right for one distance, and has error at all other distances. For example, consider a trajectory based on an *actual* muzzle velocity of 2650 fps and a G7 BC of 0.240; the observed drop will be 92" at 600 yards. Now, if you didn't know the MV or BC, you can find thru trial and error that a MV of 2700 fps and a G7 BC of 0.218 produce a trajectory prediction equal to the observed 92" of drop at 600 yards. Based on this, you would have pretty high confidence in the solution that you *calibrated* based on observation. In fact, there will be very little error in this predicted trajectory over short range. However, the problem comes in when you get far away from the calibration point. For example, at 1000 yards, the true drop of the bullet will be 401", but the *calibrated* drop which is based on the wrong MV and BC would be 420", an error of 19". Figure 8.6 below shows an example of a trajectory that's *corrected* at one range and off at other ranges.

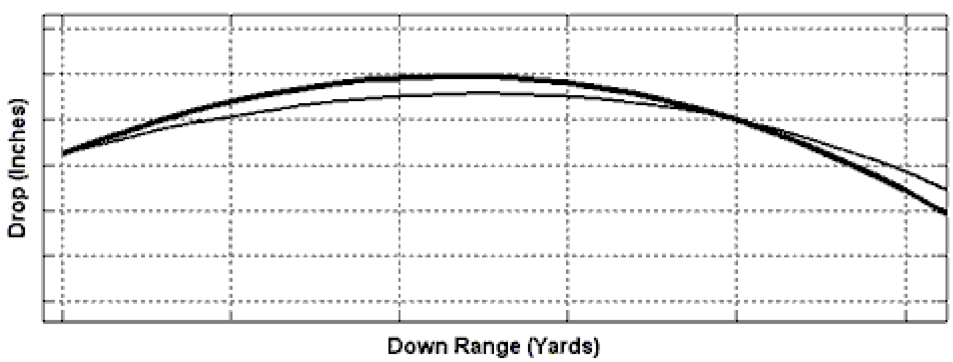

Figure 8.6. Trajectory validation can correct a trajectory for a single point, but may result in more or less error at other ranges.

The above is presented to illustrate the potential problems with trajectory calibration/validation/truing. It is possible to use this method more effectively by gathering the test points at a farther range for example, or locking down all but one of the variables.

In conclusion, the act of *falsifying* inputs to a ballistics program should be avoided if at all possible. If a scenario exists where all the inputs cannot possibly be known with a high degree of certainty (as is commonly the case in some scenarios), then using *conditioned* inputs is the only way to get a trajectory prediction. If you find yourself in a situation where you have to falsify inputs, choose the input with great care and be warned that the trajectory prediction may be inaccurate for ranges far from the distance used to calibrate the solution.