# Read Me

This program simulates the flight of intercontinental ballistic missiles based on parameters at launch time. It was written by Josh Levinger for GlobalSecurity.org in June 2005. Original credit for the simulation goes to Dr. David Wright at MIT, who wrote it for his paper ”Depressed Trajectory SLBMS", Science and Global Security, 1992, Vol 3, p101-159.

## Installation

### Windows

Double click setup.exe, the installer will run from the disc. A shortcut will be placed in the Start Menu to open the program.

### Mac OS X

Drag Ballistic.app from the Binaries folder to your Applications folder.

### Linux

It’s a Python script. Requires wxPython and Numeric.

## The Simulation

The equations of motion for a missile trajectory lying in a plane are given by:

$$\frac{dV}{dt} = \frac{T}{m}\cos\eta - \frac{C_d\rho V^2 A}{2m} - g\sin\gamma \qquad \frac{d\gamma}{dt} = \frac{d\Psi}{dt} + \frac{T}{Vm}\sin\eta - \frac{g}{V}\cos\gamma$$

$$\frac{d\Psi}{dt} = \frac{V\cos\gamma}{R_e + h} \qquad \frac{dh}{dt} = V\sin\gamma \qquad \frac{dm}{dt} = \frac{T}{g_0 I_{sp}}$$

Where V is velocity, T is thrust, m is the combined mass of the stages and RV during boost phase and the mass of a single RV after burnout, rho is atmospheric density, A is the cross sectional area of the booster, h is the altitude, g is the gravitational acceleration at h, psi is the range angle, gamma is the angle between the velocity and the local horizontal, and eta is the angle between the thrust direction and the missile axis. For gravity turn trajectories, eta is non-zero for a short period during boost and again if the orbit is depressed. For the minimum energy trajectory, eta is always zero, gamma is 90˚ during boost, and set a each stage burnout to the optimum angle for that range:

gamma_burnout = 1/2 * tan$^{-1}$(sin(phi)/cos(phi) - 1 -  h/R_earth)

Thrust is calculated as increasing with altitude according to normalized data from the Saturn V. For the first stage h_norm is h / 160934 meters (100 miles), the percent increase = -.4339*(h_norm)$^3$+.6233*(h_norm)$^2$-.01*(h_norm)+1.004 . For subsequent stages, an increase of 19% over ideal is assumed.

Drag is calculated during burn as C_drag*area*rho*V$^2$/2 where rho decreases with altitude according to the barometric formula for heights less than 19,200 meters and according to the NASA's 1976 Standard Atmospheric model for heights between 19,200 meters and 47,000 meters. C_drag is as calculated by Dr. David Wright for the Scud-A. Drag is neglected during re-entry, due to insufficient data on the typical RV.

## The Solver

The Advanced panel contains the ability to solve for the fuel fraction of the missile, given that an approximate range is known. Using the secant form of Newton's method (where the definition of the derivative is replaced for the df/dx), the program attempts to find the correct fuel mass and dry mass that attains the range while still summing to a known stage mass. The stage mass value does not include the payload, which is added by the simulation before beginning.

Note that the solver will occasionally fail to converge on a reasonable value. This is due to the fact that Newton's method requires two known starting points, and often the user only knows one. The program assumes that 99% of the given value is still a valid number, but this may not be the case. If it is not, enter a larger value in the fuel fraction field before clicking Solve.

## Presets

You can edit the list of preset missile parameters, in preset.txt. The data are stored in a relatively simple format, as a dictionary of Python dictionaries. Note that fuelmass, drymass, Isp0, burntime and thrust are lists whose first (zeroth) entry is 0. This is because lists in python are zero-based, and it makes sense to track stage data by index than remembering this. The quotes around each key are also important, don't forget them if you add another preset. This file is read and eval'ed by Python, so be careful.

## License