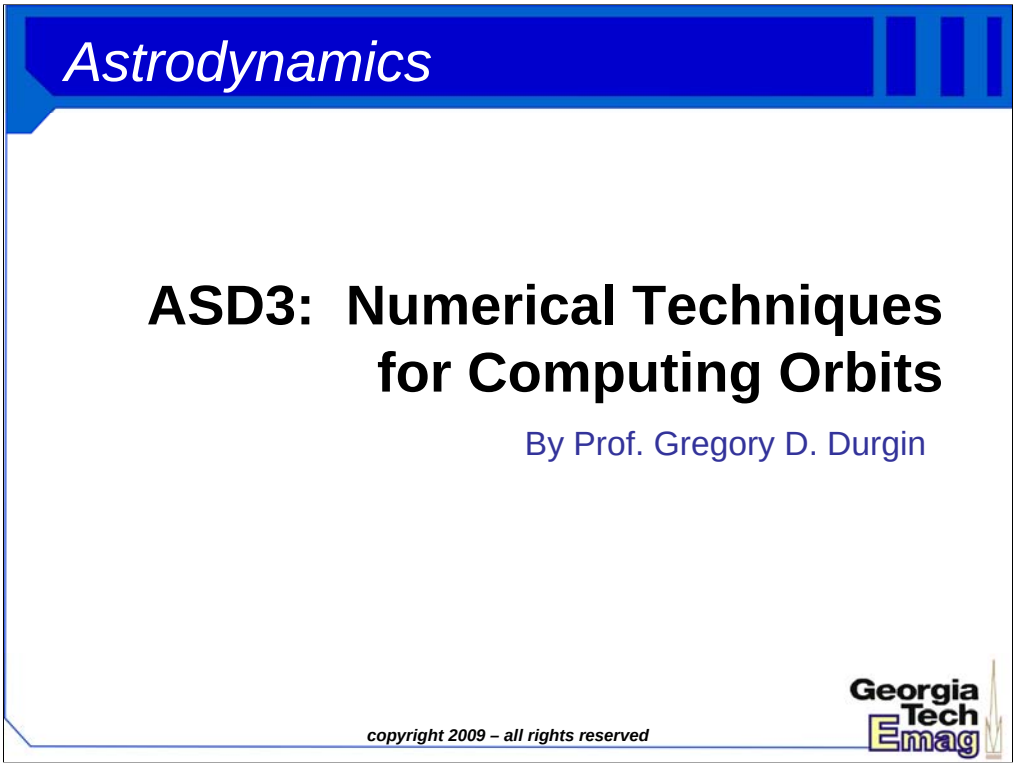
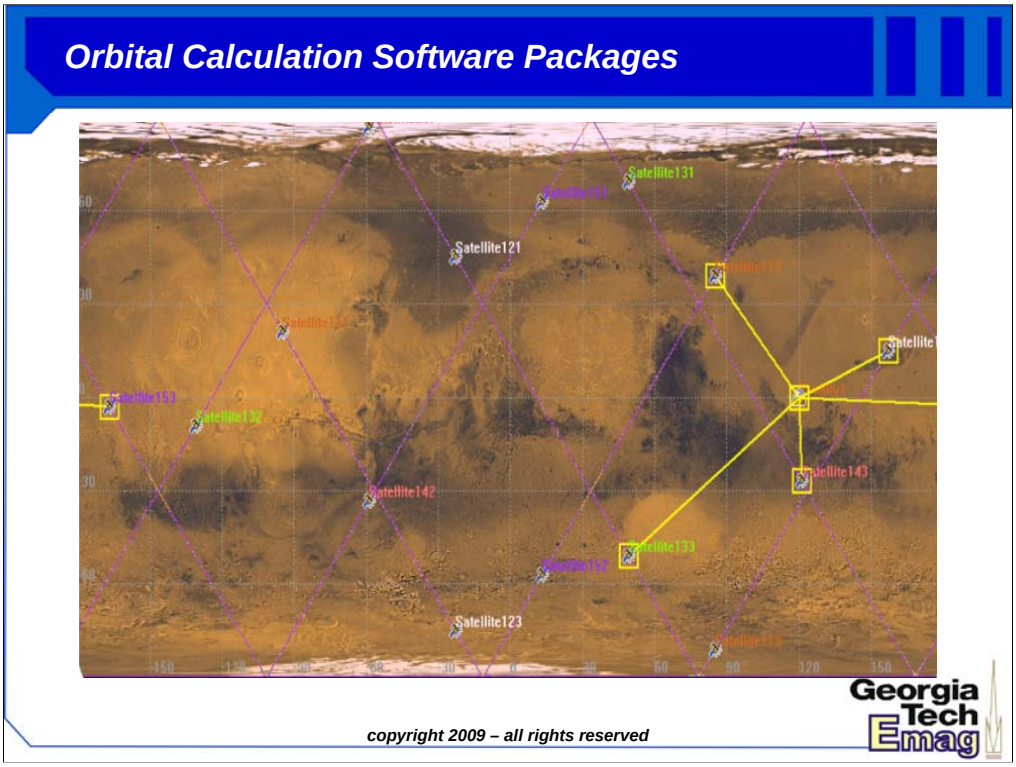


In this lecture, we present a method for computing the trajectories of satellites around planets or stars with a set of initial velocity conditions. Again, we'll be making our 2D assumption about the geometry of the orbital system of our two-body problem. There are actually some sophisticated simulation packages out there now for orbital planning, which are useful for planning that requires more precision or complexity. However, our approach in this lecture is a very generic method for turning the equations of motion into a set of discrete difference equations. The technique is actually very useful for all types of numerical simulations – not just orbital problems.

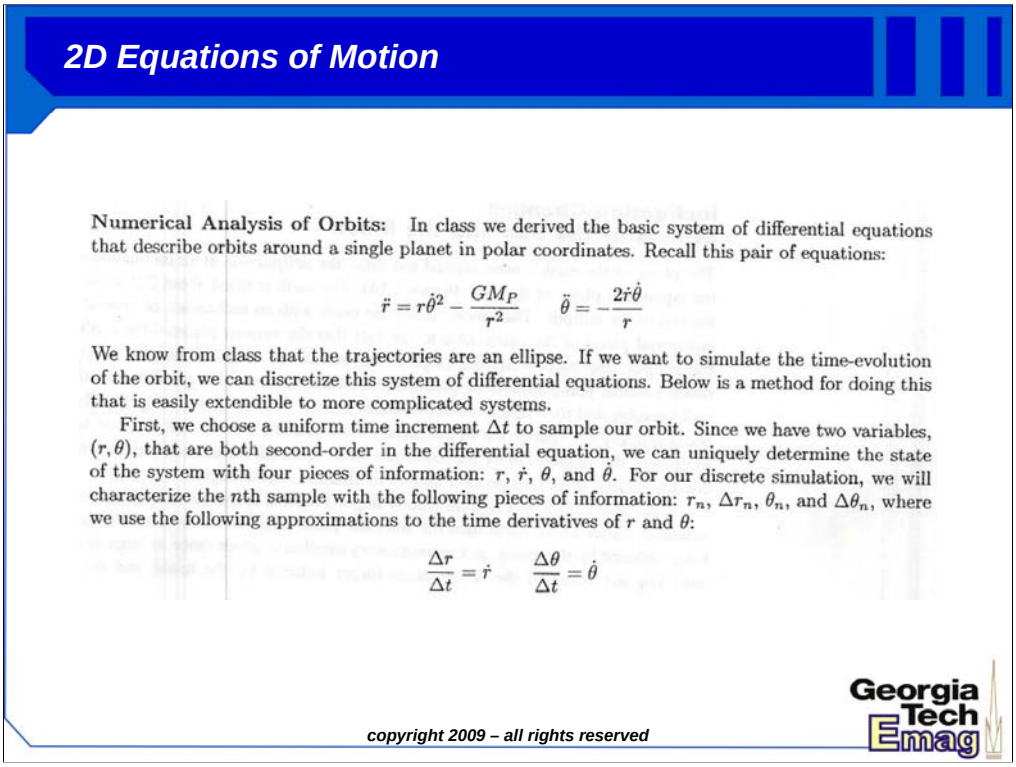
By the end of the class, the student should be able to conduct basic computer-based orbital analysis without resorting to expensive software packages. Later, we will tackle some interesting exercise problems using student-developed code.



In this day in age, there are some powerful planning packages available on the computer that were not available to early satellite designers. They are capable of plotting the time-evolving orbits of entire satellite constellations, taking into account multi-body effects and irregularities that our simple 2D models cannot. The computer software packages also allow for extraordinary visualizations of flight paths, ground tracks, and look angles.

However, there are many times when these expensive packages are overkill, or at least fail to provide a mode of analysis that may be required in a system analysis. There is also a lot of value in understanding the general mechanisms of mechanical simulation.

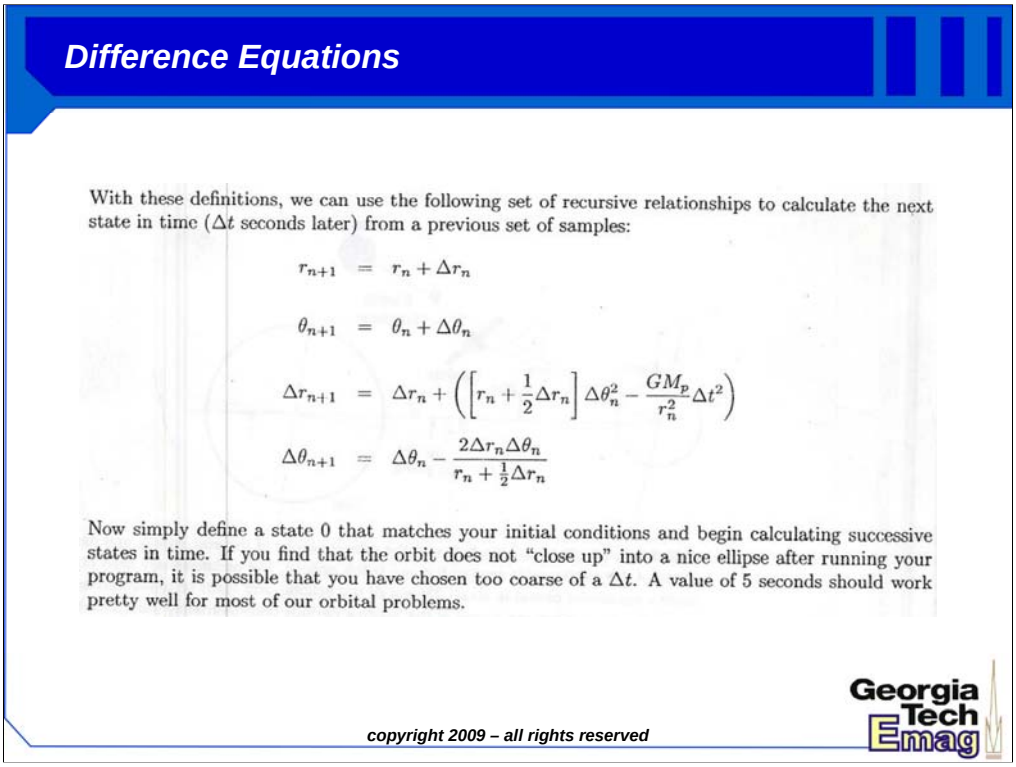
The above picture is a snapshot from the "Satellite Toolkit" by AGI, which is a popular package for orbital systems. In this picture, the tool is showing the time evolution of a constellation of satellites over the surface of Mars. There is an Mars station on the equator (right side). The tool is demonstrating which of the satellites in the constellation are capable of communicating with this station. This is an excerpt from a 2008 Satcom project where the students were charged with designing a global positioning system for Mars. Here they are checking to make sure at least 4 satellites are in view of a station at all times so that they can guarantee 3D positioning with a time-difference of arrival system.



Above are the rearranged 2D equations of motion. We do not need to spend 10s of thousands of dollars to calculate an orbit; in fact, we can code up a simulator in just a few minutes. The equations of motions involve two variables of trajectory (r,theta) and they appear in addition to their first and second time derivatives. Thus, we can consider this a second-order, two-variable system (one order each time a derivative is taken). To specify an exact solution in a single-variable, second-order system, we only require two pieces of information to serve as initial conditions. In this system, we can uniquely specify the solution to this system of differential equations with 4 pieces of information: a current trajectory (r,theta) and the velocity at that point (or r-dot, theta-dot). By keeping track of these four parameters, we can completely specify the state of the oribt.

We will need to discretize/sample our trajectory over time, so start by picking a time increment delta-t. A trajectory will be represented by a sequence or array of (r,theta) points sampled at time interval delta-t. There is an art to picking the time increment. The smaller delta-t, the finer and more accurate the simulation – but these precision gains come at the cost of computation time. Larger delta-t speeds up the computation (same number of calculated points results in a larger time span), but the coarser jumps in time result in larger errors that accumulate faster through the analysis.

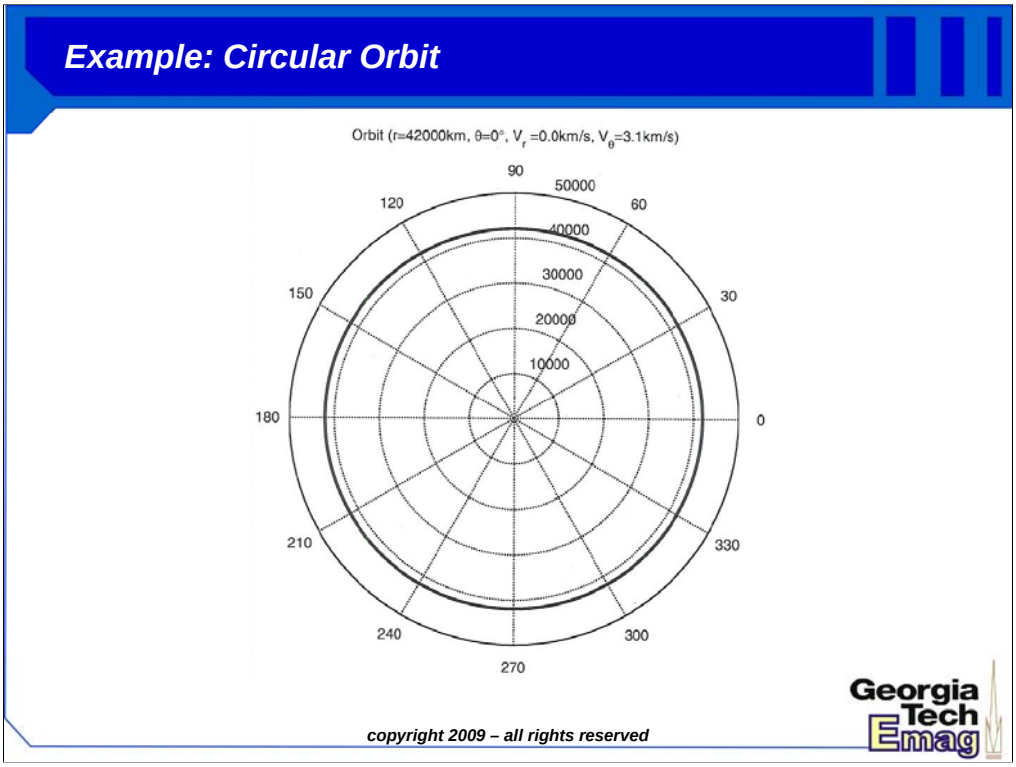
Once delta-t is picked, we can estimate velocities using the concept of a “digital derivative”. For example, the radial component of velocity (r-dot) can be estimated as the change in consecutive radius samples in time, divided by the time increment delta-t.



In the same way, we can estimate accelerations with the digital derivative of velocity. For this case, we take the difference of two adjacent digital velocity values and again divide by our time increment. This is the straightforward "change of a change" definition of acceleration/second derivative.

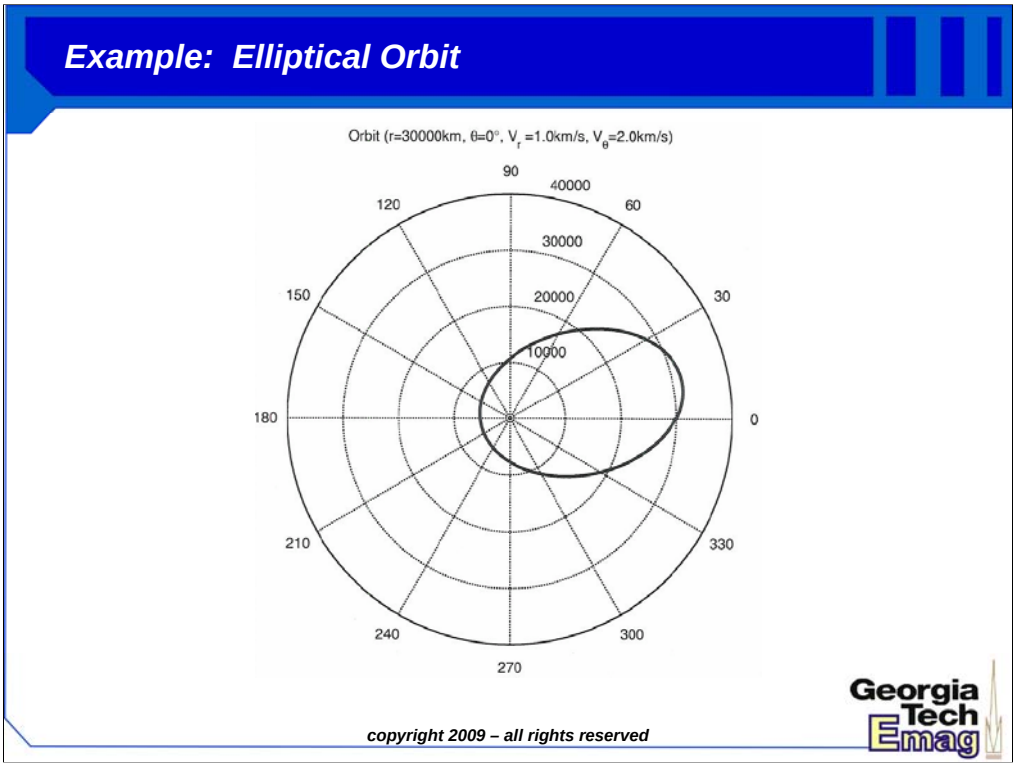
The above system of equations, after some simplifications, represents to four set of difference equations that are required to simulate this system. The four pieces of information that are tracked at each time increment to specify the system are position (r, theta) and change increments (delta r, delta theta). Note that this system is iterative. Given a set of initial conditions (say position and velocity), we must calculate the position and change increments for the n=0 initial "seed" sample. Then it's off to the races. We base the next n=1 sample, occurring delta-t seconds later than the initial condition, on a computation involving the previous set of samples. We can place this simple system of difference formulas into a program loop and calculate the trajectory for as long as we have patience and computer time!

Now, what should come out of here for a stable system is some sort of elliptical orbit path. It is often best to orient your problem to start at theta = 0 and include a stopping criterion at theta >= 2pi (always use radians) when the orbit has completed one whole period. If the computer locks up, it's likely that you picked a time increment too small. If the orbit looks sort of like a mismatched oval that doesn't close on itself, the time increment was too large – too many errors built up in your calculation.



Here's an example orbit in Matlab using the difference equation. It's a geosynchronous orbit, having an initial radius of 42 million meters from the planet center and an initial velocity of 3.1 km/s (all in the theta/tangential direction). We calculated these values in class earlier. If they were correct, then our difference equation engine should trace out a perfectly circular path in 23 hours 56 minutes (1 sidereal day).

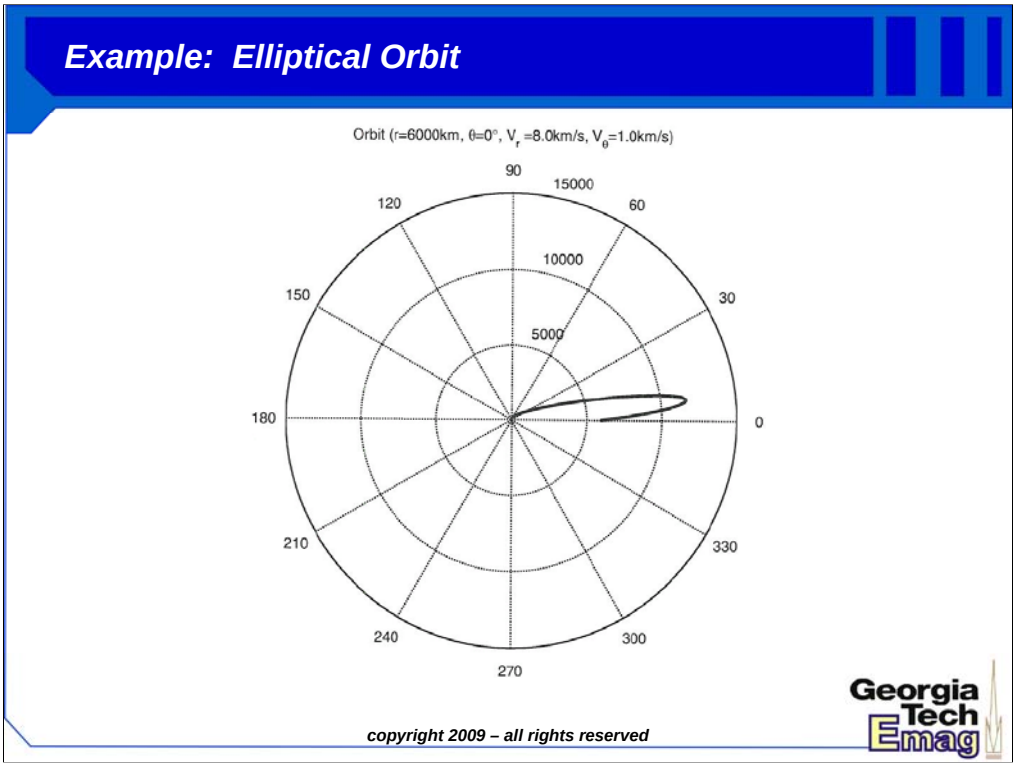
Sure enough, that is what we see. The time increment for this plot was chosen to be 5 seconds.



Here is an example of an elliptical orbit. In this case, the initial condition places the satellite at 30,000 km from the center of the earth. This time, a skewed velocity is given as the initial condition for the satellite: positive 1km/s in the radial direction and 2 km/s in the theta direction.

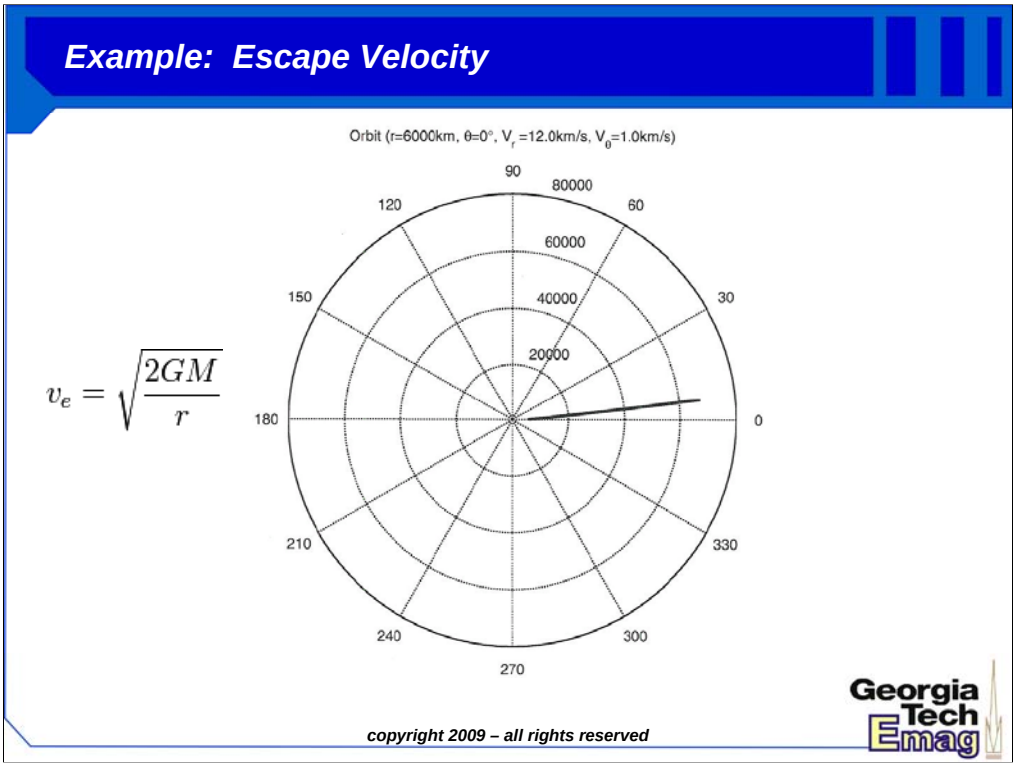
Initially the satellite continues along this direction, but because it lacks a lot of kinetic energy (less than the previous example) and is closer to the planet initially, the earth pulls the trajectory back towards itself. The satellite speeds up according to Kepler's second law, whips around the back side of the planet and returns to its starting point in orbit.

In fact, the graphs suggests an approach by the satellite that is dangerously close to the surface of the earth, possibly beneath the 6380 km altitude required to clear the planet's surface. So be careful interpreting results on graphs like this!



Here's another highly eccentric orbit that has its starting point on the surface of the earth and its velocity vector mostly in the radial direction. With essentially 8 km/sec, the rocket "orbit" would bring this satellite crashing back down to earth. Of course, a rocket cannot instantly accelerate to this velocity; it must slowly and steadily burn its fuel until a comparable amount of velocity/kinetic energy has been imparted to the spacecraft. Launch calculations are quite intricate, since most of the energy at the start of the launch is spent lifting the fuel to maintain thrust! It's not very efficient, but solid or liquid rockets are the only thing we can feasibly build right now to provide not only the energy but also the power to get objects into space.

Clearly, an additional rocket burn is required at higher altitudes to get the satellite into a stable orbit. Otherwise, the rocket still crashes back to earth!



With 12km/s, the rocket has enough kinetic energy to continue away from the earth. This trajectory will never close on itself, because the spacecraft has too much initial kinetic energy to be captured by the gravitational field. We call the minimum speed that an object must have to leave the earth completely the *escape velocity*.

As measured from the starting point on the surface of the earth, escape velocity is about 7 miles/sec (about 11 km/sec).