PROFESSOR: A Very important property of a numerical method is its order. The accuracy of the method is proportional to a power of the step size. And that power is called the order.

If h is the step size and p is the order, then the error made in one step is proportional to h to the p plus 1. And the error made in traversing an entire interval is proportional to H to the p.

So this means, if you're using a method of order p, and cut the step size in half, you can expect the overall error to be reduced by a factor of 2 to the p. The order of a numerical method is determined by analysis involving Taylor series during the derivation of the method. But we can also do an experiment to determine the order.

That's what this program does. The input is the name of an ODE solver. And then it's going to do a numerical integration of an ordinary differential equation, just involving t. So the result is the value of an integral. The integral of 1 over 1 plus t squared, from 0 to 1.

We know the exact answer is 1/2. So we integrate that differential equation twice, once with a step size of 0.1, and then with a step size of one half that. We integrate the differential equation, take the final value of y for each of those two integrations, compare those values with the exact answer, take the ratio of those two values. That shows how much the error is decreased when we cut the step size in half.

The log base 2 of that ratio is the order. It should be an integer so we can round it to the nearest integer, and return that value as the value in this function. Let's run our experiment first on ODE1.

We step size of 0.1, this method gets the integral as 0.5389, not very accurate. Cut the step size in half, it gets 0.5191. The ratio of those is two. Logarithm base 2 is 1. So ODE1 has order 1.

Now ODE2-- step size 0.1. 0.499. cut it in half, 0.4998. The ratio of those is close to 4. And so ODE2, we find with this experiment, is getting order 2.

Now let's try classical Runge-Kutta. This is why it's so popular. It's very accurate. We step size of 0.1, we get close to 1/2. Cut the step size in half, we get even closer. The ratio of these two is close to 16. To the log base 2 is 4. So ODE4 has, experimentally, order 4.

So we found that, at least with this single experiment, the ODE solvers 1, 2, and 4, have orders 1, 2, and 4. So as you probably realized, this is why we named them ODE 1, 2, and 4.

This brings us to the naming conventions in the functions in the MATLAB ODE suite. All the functions have names that are variations on the theme ODEpq. That means that the method ODEpq uses methods of order p and q So we've been getting a glimpse of that with our names, ODE 1, 2, and 4.

Here's an exercise. Modify order x to do further experiments involving the order of our ODE solvers. Change it to do other integrals. And check out the order of ODE 1, 2, and 4.