

PROFESSOR: The most frequently used ODE solver in MATLAB and Simulink is ODE45. It is based on method published by British mathematicians JR Dormand and PJ Prince in 1980. The basic method is order five. The error correction uses a companion order four method.

The slope of t_n is, first same as last left over from the previous successful step. Then there are five more slopes from function values at $1/5 h$, $3/10h$, $4/5h$, $8/9h$ and then at t_n plus 1. These six slopes, linear combinations of them, are used to produce y_n plus 1.

The function is evaluated at t_n plus 1 and y_n plus 1 to get a seventh slope. And then linear combinations of these are used to produce the error estimate.

Again, if the error estimate is less than the specified accuracy requirements the step is successful. And then that error estimate is used to get the next step size. If the error is too big, the step is unsuccessful and that error estimate is used to get the step size to do the step over again.

If we want to see the actual coefficients that are used, you can go into the code for ODE45. There's a table with the coefficients. Or you go to the Wikipedia page for the Dormand-Prince Method and there is the same coefficients.

As an aside, here is an interesting fact about higher order Runge-Kutta methods. Classical Runge-Kutta required four function evaluations per step to get order four. Dormand-Prince requires six function evaluations per step to get order five. You can't get order five with just five function evaluations. And then, if we were to try and achieve higher order, it would take even more function evaluations per step.

Let's use ODE45 to compute e to the t . y prime is equal to y . We can ask for output by supplying an argument called `tspan`. Zero and steps of 0.1 to 1. If we supply that as the input argument to solve this differential equation and get the output at those points, we get that back as the output. And now here's the approximations to the solution to that differential equation at those points.

If we plot it, here's the solution at those points. And to see how accurate it is, we see that we're actually getting this result to nine digits. ODE45 is very accurate.

Let's look at step size choice on our problem with near singularity, is a quarter. y_0 is close to

16. The differential equation is $y' = 2(a-t)y^2$. We let ODE45 choose its own step size by indicating we just want to integrate from 0 to 1. We capture the output in t and y and plot it.

Now, here, there's a lot of points here, but this is misleading because ODE45, by default, is using the refine option. It's only actually evaluating the function at every fourth one of these points and then using the interpolant to fill in in between. So we actually need a little different plot here.

This plot shows a little better what's going on. The big dots are the points that ODE45 chose to evaluate the differential equation. And the little dots are filled in with the interpolant. So the big dots are every fourth point. And the refine option says that the big dots are too far apart and we need to fill it in with the interpolant. And so this is the continuous interpolant in action.

The big dots are more closely concentrated as we have to go around the curve. And then, as we get farther away from the singularity the step size increases. So this shows the high accuracy of ODE45 and the automatic step size choice in action.

Here's an exercise. Compare ODE23 and ODE45 by using each of them to compute π . The integral $\int_0^1 \frac{4}{1+t^2} dt$ is π . You can express that as a differential equation, use each of the routines to integrate that differential equation and see how close they get to computing π .