

CS341 SYSTEMS BIOLOGY I

LECTURE 13

NUMERICAL METHODS FOR SOLVING DIFFERENTIAL EQUATIONS (CONT)

In lectures 11 and 12, we implemented the forward Euler method and examined the effects of time step on the simulation output. We observed that if the step is smaller, the error is smaller. And if the method's output is relatively close to the true solution, that error depends linearly on the step size.

That linear dependence is due to the fact that we use information just from the beginning of the time step to predict the value at the end of the time step (for a derivation, please see Wikipedia, which will show that the local truncation error is $\mathcal{O}(\Delta t^2)$ and the global error as $\mathcal{O}(\Delta t)$, where global error is what we are measuring).

More accurate Runge-Kutta methods take into account more information within the time step. Today, we will look at a 2nd order Runge-Kutta method – a method whose error is proportional to Δt^2 .

TRAPEZOIDAL METHOD

The trapezoidal method (also called Heun's method) attempts to take into account curvature in the line. The basic idea is that the forward Euler method will either undershoot or overshoot the true solution because the solution curves. If we knew the slope at the end of the time step, then we can correct our estimate to account for at least some of the curvature. But how do we learn the slope at the end? We use Forward Euler to estimate y at the end of the time step, then use that value to estimate the slope $\frac{dy}{dt}$ at the end of the time step. Finally, we use the average of the slopes at the beginning and end of the time step to take our step. (See Figure 1 for an illustration.)

The method is

$$\widehat{y}(t + \Delta t) = \widehat{y}(t) + \Delta t \cdot \frac{f(t, \widehat{y}(t)) + f(t + \Delta t, \widehat{y}_{FE})}{2}$$

where

$$\widehat{y}_{FE}(t + \Delta t) = \widehat{y}(t) + \Delta t \cdot f(t, \widehat{y}(t))$$

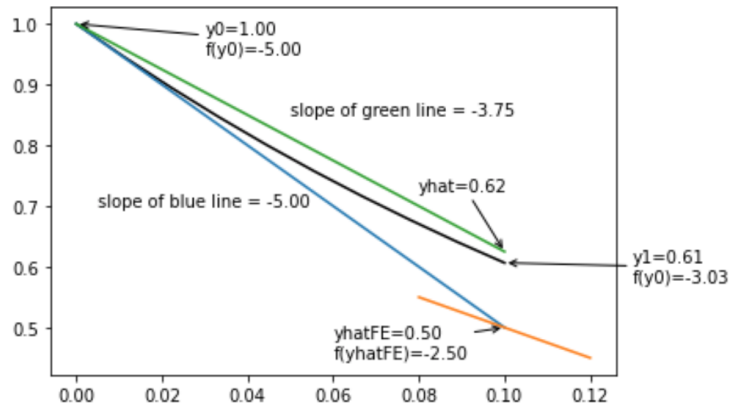


FIGURE 1. Explicit Trapezoidal method used to estimate the one time step of test equation (degradation model) with degradation rate constant 2.5, initial condition $y(0) = 1$, and a time step of 0.1. We show the true solution (y , black line) and the approximate (which should be $\hat{y}$, but I couldn't figure out how to get that notation in matplotlib, so I wrote it as "yhat") solution as estimated by the explicit trapezoidal method (green line). I use "yhatFE" for the intermediary solution from the forward Euler method (blue line).

At $t = 0$, $y=0$, $yhat=0$, and $yhatFE=0$. The forward Euler method computes the tangent line at $y(0)$ – it has a slope of -5. That slope is used to compute $yhatFE$ (blue line indicates the line followed to take the step). The tangent line at $yhatFE$ has a slope of -2.5 (indicated with the orange line). The two slopes are averaged (-3.75), and a step is taken in that direction (green line), resulting in $yhat(0.1)=0.62$, which is closer to the true solution $y(0.1)=0.61$ than the estimate by Forward Euler ($yhatFE=0.5$).

The error from explicit trapezoidal will generally be smaller than the error from forward Euler. And, in the range of reasonable step sizes, the error should be a quadratic function of the step size. For the test equation, we see the expected trends (linear dependence on step size for forward Euler, quadratic dependence on step size for Explicit Trapezoidal). (See Figure 2.)

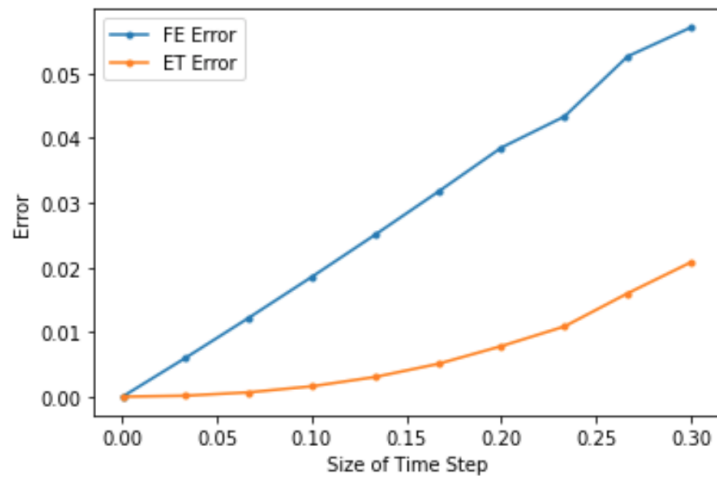


FIGURE 2. For the test equation with 2 ODE's with degradation rates of 2 and 2.5, integrated from $y_0=(1,1)$ from $t=0$ to $t=4$, we solved the equations with both the forward Euler (FE) and explicit trapezoidal (ET) methods for time steps ranging from $1e-4$ to 0.3. The error from each simulation was computed as the mean over time of the Euclidean distance between the estimated solution and the true solution at each time step.