18.034 Honors Differential Equations

Spring 2009

Higher Order Taylor Methods

Marcelo Julio Alvisio & Lisa Marie Danz

May 16, 2007

1 Introduction

Differential equations are one of the building blocks in science or engineering. Scientists aim to obtain numerical solutions to differential equations whenever explicit solutions do not exist or when they are too hard to find. These numerical solutions are approximated though a variety of methods, some of which we set out to explore in this project.

We require two conditions when computing differential equations numerically. First, we require that the solution is continuous with initial value. Otherwise, numerical error introduced in the representation of the number in computer systems would produce results very far from the actual solution. Second, we require that the solution changes continuously with respect to the differential equation itself. Otherwise, we cannot expect the method that approximates the differential equation to give accurate results.

The most common methods for computing differential equations numerically include Euler's method, Higher Order Taylor method and Runge-Kutta methods. In this project, we concentrate on the "Higher Order Taylor Method." This method employs the Taylor polynomial of the solution to the equation. It approximates the zeroth order term by using the previous step's value (which is the initial condition for the first step), and the subsequent terms of the Taylor expansion by using the differential equation. We call it Higher Order Taylor Method, the "lower" order method being Euler's Method.

Under certain conditions, the Higher Order Taylor Method limits the error to $O(h^n)$, where n is the order used. We will present several examples to test this idea. We will look into two main parameters as a measure of the effectiveness of the method, namely accuracy and efficiency.

2 Theory of the Higher Order Taylor Method

Definition 2.1 *Consider the differential equation given by $y'(t) = f(t, y)$, $y(a) = c$. Then for $b > a$, the nth order Taylor approximation to $y(b)$ with K steps is given by y_K, where $\{y_i\}$ is defined recursively as:*

$$t_0 = a$$

$$y_0 = y(a) = c$$

$$t_{i+1} = t_i + h$$

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2}\frac{\partial f}{\partial t}(t_i, y_i) + ... + \frac{h^n}{n!}\frac{\partial^{n-1} f}{\partial t^{n-1}}(t_i, y_i)$$

with $h = (b - a)/K$.

It makes sense to formulate such a definition in view of the Taylor series expansion that is used when $y(t)$ is known explicitly. All we have done is use $f(t, y)$ for $y'(t)$, $f_t(t, y)$ for $y''(t)$, and so forth. The next task is to estimate the error that this approximation introduces.

We know by Taylor's Theorem that, for any solution that admits a Taylor expansion at the point t_i, we have

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + ... + \frac{h^n}{n!}y^{(n)}(t_i) + \frac{h^{(n+1)}}{(n+1)!}y^{(n+1)}(\sigma)$$

where σ is between t_i and t_{i+1}

Using $y' = f(t, y)$, this translates to

$$y(t_{i+1}) = y(t_i) + hf(t_i, y_i) + \frac{h^2}{2}\frac{\partial f}{\partial t}(t_i, y_i) + ... + \frac{h^n}{n!}\frac{\partial^{(n-1)} f}{\partial t^{(n-1)}}(t_i, y_i) + \frac{h^{(n+1)}}{(n+1)!}\frac{\partial^{(n)} f}{\partial t^{(n)}}(\sigma, y(\sigma))$$

Therefore, the *local error*, that is to say, the error introduced at each step if the values calculated previously were exact, is given by:

$$E_i = (h^{n+1})\frac{1}{(n+1)!}\frac{\partial^{(n)} f}{\partial t^n}(\sigma, y(\sigma))$$

which means that

$$E_i \leq \max_{\sigma \in [a,b]}(h^{n+1})\frac{1}{(n+1)!}\frac{\partial^{(n)} f}{\partial t^n}(\sigma, y(\sigma))$$

We can say $E_i = O(h^{n+1})$. Now, since the number of steps from a to b is proportional to $1/h$, we multiply the error per step by the number of steps to find a total error

$$E = O(h^n).$$

3 In Practice: Examples

We will consider differential equations that we can solve explicitly to obtain an equation for $y(t)$ such that $y'(t) = f(t, y)$. This way, we can calculate the actual error by subtracting the exact value for $y(b)$ from the value that the Higher Order Taylor method predicts for it. To approximate values in the following examples, the derivatives of $f(t, y)$ were computed by hand. MATLAB then performed the iteration and arrived at the approximation.

Notice that the definitions given in the previous section could also have been adapted for varying step size h. However, for ease of computation we have kept the step size constant. In our computations, we have chosen step size of $(b-a)/2^k$, which resulted in $K = 2^k$ evenly spaced points in the interval.

Example 3.1 *We consider the differential equation*

$$y'(t) = f(t, y) = \frac{1+t}{1+y}$$

with initial condition $y(1) = 2$. *It is clear that* $y(t) = \sqrt{t^2 + 2t + 6} - 1$ *solves this equation.*

Thus we calculate the error for $y(2)$ *by subtracting the approximation of* $y(2)$ *from* $y(2)$, *which is the exact value. Recall that we are using* $h = 2^{-k}$ *because* $(b-a) = 1$. *The following table displays the errors calculated.*

	$k=1$	$k=2$	$k=3$	$k=4$
$order = 1$	.0333	.0158	.0077	.0038
$order = 2$	$-.0038$	$-.0009$	$-.0002$	$-.0001$
$order = 3$	.0003269	.0000383	.0000046	.0000006

For the first order case (also known as Euler's method), we expect the error to be proportional to h, *where* h *is the length of the intervals tested. This means that, when* k *increases by* 1, *we expect the error to be halved (approximately). Observing the values in the table confirms this expectation.*

For the second order case, we expect the error to be proportional to h^2, *so we expect the error to be divided by* 4 *as we move to the right on the table. For third order, we expect the error to be proportional to* h^3, *so we expect the error to be divided by* 8 *as we move to the right. Again, this seems to be the case.*

In figure 1, the graph on the left show the approximations of $y(t)$ plotted against t by using Euler's method. Each line corresponds to one row of the table above, that is, to different step sizes $h = 1/2$, $1/4$, $1/8$ and $1/16$. The graphs on the right show the error plotted against t, so the errors at $t = 2$ are those given in the table. Here again, each line corresponds to a different step size $h = 1/2$, $1/4$, $1/8$ and $1/16$, Obviously, the lines with error closer to zero correspond to higher values of k.

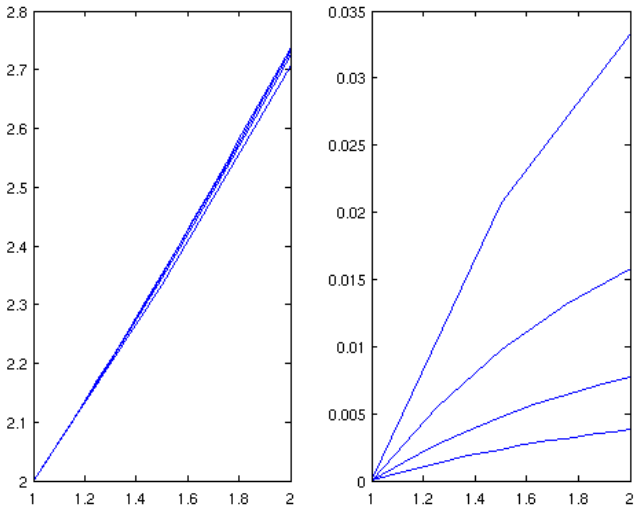

Figure 1: Euler's Method: numerical computation of the solution with step size (on the left) and the corresponding error (on the right).

Similarly, we have produced the same graphs using Taylor's method of Second and Third order. They are show in figure 2 and 3 respectively.

Now we will consider more examples to compare the pattern of the error values. We will not include the new graphs since their similarity with the graphs for the first example makes them uninteresting.

Example 3.2 *Here we consider the equation $f(t, y) = \frac{1}{2y}$, with initial condition $y(0) = 2$. Its explicit solution is $y(t) = \sqrt{t + 4}$. If we generate the error as above, we obtain the following table for $t = 1$.*

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
order $= 1$	$-.0066$	$-.0032$	$-.0016$	$-.0008$
order $= 2$	.0003852	.0000917	.0000224	.0000055
order $= 3$	$-.00002837$	$-.00000330$	$-.00000040$	$-.00000005$

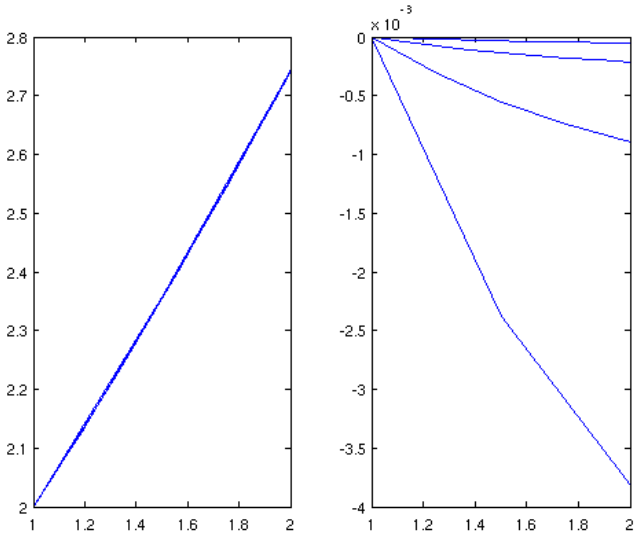

Figure 2: Second order Taylor's Method: numerical computation of the solution with step size (on the left) and the corresponding error (on the right).

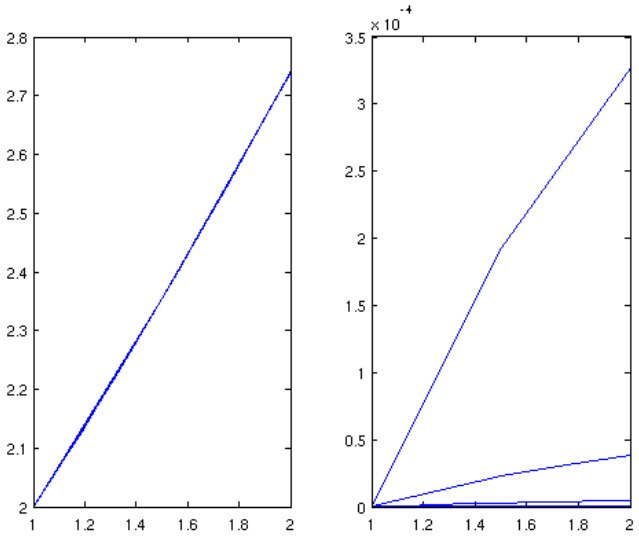

Figure 3: Third order Taylor's Method: numerical computation of the solution with step size (on the left) and the corresponding error (on the right).

This second example again meets our expectations as stated in Example 1. Upon inspection, it is clear that first order errors are roughly halved with each increase in k, second order entries are divided by 4, and third order entries are divided by 8.

The examples that follow will again illustrate this phenomenon.

Example 3.3 *Consider the differential equation $y'(t) = f(t, y) = t + y$, with initial condition $y(0) = 0$. The explicit solution is $y(t) = -t - 1 + e^t$. This gives the following data for $t = 1$.*

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$order = 1$	.4683	.2769	.1525	.0804
$order = 2$	.0777	.0234	.0064	.0017
$order = 3$	.0095	.0014	.0002	.0000

Note that the error in the lower right is not actually zero; this is just from the computer rounding off.

Example 3.4 *Consider the differential equation $y'(t) = f(t, y) = ty$ with initial condition $y(0) = 1$. Thus the explicit solution is $g(t) = e^{\frac{1}{2}t^2}$. This gives the following data for $t = 1$.*

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$order = 1$	.3987	.2293	.1247	.0653
$order = 2$	.0667	.0225	.0065	.0018
$order = 3$	.0286	.0045	.0006	.0001

4 Evaluation of the method

As we have seen in these examples, and as our theory would suggest, the Higher Order Taylor Method is able to arrive at very accurate approximations with relatively large step size.

However, we have used the information of the derivatives of $f(t, y)$. Differentiation in general can be costly. In practice, there are cases where the differential equation is a black box: $f(t_0, y_0)$ can be found by empirical methods given t_0 and y_0, but the explicit equation $f(t, y)$ is unknown. In such cases, the Higher Order Taylor Method simply does not work.

There is a class of differential equations, known as "stiff equations," to which this method cannot be readily applied. The solutions of stiff equations are characterized by *decaying* as t increases, while the derivatives do not. This can cause wildly inaccurate approximations unless h is within a certain bound. It is generally easier to use an alternate method rather than worry about keeping

h within its bound. Indeed, other methods have been developed specifically to accomodate stiff equations.

We have also assumed that the solution has a Taylor expansion that converges to the actual solution. Although this is the case for a wide variety of functions, it is advisable to take this into consideration. Alternative ways of verifying the approximation are advisable.

In summary, the Higher Order Taylor Method is not universally applicable. When it does apply, it can be fairly accurate. In our examples, the error decays with decreasing h just as the theory would predict. In fact, the rate of decay is proportional to h^n, where n is the order of approximation used.