Math 246, Fall 2008, Professor Levermore
Numerical Methods

For many first order differential equations analytic methods are either difficult or impossible to apply. If one is interested in understanding how a particular solution behaves it is often easiest to use numerical methods to construct accurate approximations to the solution. Suppose we are interested in the solution $Y(t)$ of the initial-value problem

$$\frac{dy}{dt} = f(t, y), \qquad y(t_I) = y_I, \tag{1.1}$$

over the time interval $[t_I, t_F]$ — i.e. for $t_I \leq t \leq t_F$. Here t_I is called the *initial time* and t_F is called the *final time*. A numerical method selects times $\{t_n\}_{n=0}^N$ such that

$$t_I = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N = t_F,$$

and computes values $\{y_n\}_{n=0}^N$ such that $y_0 = Y(t_0) = y_I$ and y_n approximates $Y(t_n)$ for $n = 1, 2, \cdots, N$. For good numerical methods, these approximations will improve as N increases. So for sufficiently large N you can plot the points $\{(t_n, y_n)\}_{n=0}^N$ in the (t, y)-plane and "connect the dots" to get an accurate picture of how $Y(t)$ behaves over the time interval $[t_I, t_F]$.

Here we will introduce a few basic numerical methods in simple settings. The numerical methods used in software packages such as MATLAB are generally far more sophisticated than those we will study here. They are however built upon the same fundamental ideas as the simpler methods we will study. Throughout this section we will make the following two basic simplifications.

- We will employ *uniform time steps*. This means that given N we set

  $$h = \frac{t_F - t_I}{N}, \qquad \text{and} \qquad t_n = t_I + nh \quad \text{for } n = 0, 1, \cdots, N, \tag{1.2}$$

  where h is called the *time step*.
- We will employ *one-step methods*. This means that given $f(t, y)$ and h the value of y_{n+1} for $n = 0, 1, \cdots, N-1$ will depend only on y_n and h.

Sophisticated software packages use methods in which the time step is chosen adaptively. In other words, the choice of t_{n+1} will depend on the behavior of recent approximations — for example, on (t_n, y_n) and (t_{n-1}, y_{n-1}). Employing uniform time steps will greatly simplify the algorithms, and thereby simplify the programming you will have to do. If you do not like the way a run looks, you will simply try again with a larger N. Similarly, sophisticated software packages will sometimes use so-called *multi-step methods* for which the value of y_{n+1} for $n = m, m+1, \cdots, N-1$ will depend on y_n, y_{n-1}, $\cdots$, and y_{n-m} for some positive integer m. Employing one-step methods will again simplify the algorithms, and thereby simplify the programming you will have to do.

1.1. Euler Methods. The simplest (and the least accurate) numerical methods are the Euler methods. These can be derived many ways. Here we give a simple approach based on the definition of the derivative through difference quotients.

If we start with the fact that

$$\lim_{h\to 0} \frac{Y(t+h)-Y(t)}{h} = \frac{dY}{dt}(t) = f(t, Y(t))\,,$$

then for small positive h one has

$$\frac{Y(t+h)-Y(t)}{h} \approx f(t, Y(t))\,.$$

Upon solving this for $Y(t+h)$ we find that

$$Y(t+h) \approx Y(t) + hf(t, Y(t))\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) \approx Y(t_n) + hf(t_n, Y(t_n))\,.$$

Because y_n and y_{n+1} are approximations of $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + hf(t_n, y_n) \quad \text{for } n = 0, 1, \cdots, N-1\,. \tag{1.3}$$

This so-called Euler method was introduced by Euler in 1700's.

Alternatively, we could have start with the fact that

$$\lim_{h\to 0} \frac{Y(t)-Y(t-h)}{h} = \frac{dY}{dt}(t) = f(t, Y(t))\,,$$

then for small positive h one has

$$\frac{Y(t)-Y(t-h)}{h} \approx f(t, Y(t))\,.$$

Upon solving this for $Y(t-h)$ we find that

$$Y(t-h) \approx Y(t) - hf(t, Y(t))\,.$$

If we let $t = t_{n+1}$ above (so that $t - h = t_n$) this is equivalent to

$$Y(t_{n+1}) - hf(t_{n+1}, Y(t_{n+1})) \approx Y(t_n)\,.$$

Because y_n and y_{n+1} are approximations of $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} - hf(t_{n+1}, y_{n+1}) = y_n \quad \text{for } n = 0, 1, \cdots, N-1\,. \tag{1.4}$$

This method is called the *implicit Euler* or *backward Euler* method. It is called the implicit Euler method because equation (1.4) implicitly relates y_{n+1} to y_n. It is called the backward Euler method because the difference quotient upon which it is based steps backward in time (from t to $t-h$). In contrast, the Euler method (1.3) sometimes called the *explicit Euler* or *forward Euler* method because it gives y_{n+1} explicitly and because the difference quotient upon which it is based steps forward in time (from t to $t+h$).

The implicit Euler method can be very inefficient unless equation (1.4) can be explicitly solved for y_{n+1}. This can be done when $f(t,y)$ is a fairly simple function if y. For example, it can be done when $f(t,y)$ is linear or quadratic in either y or $\sqrt{y}$. However, as you will see, there are equations for which the implicit Euler method will outperform the (explicit) Euler method.

1.2. Explicit Methods Based on Taylor Approximation. The explicit (or forward) Euler method can be understood as the first in a sequence of explicit methods that can be derived from the Taylor approximation formula.

1.2.1. Explicit Euler Revisited. The explicit Euler method can be derived from the first order Taylor approximation, which is also called the tangent line approximation. This approximation states that if $Y(t)$ is twice continuously differentiable then

$$Y(t+h) = Y(t) + h\frac{dY}{dt}(t) + O(h^2)\,. \tag{1.5}$$

Here the $O(h^2)$ means that the remainder vanishes at least as fast as h^2 as h tends to zero. This remainder can be expressed in its *Lagrange form* as

$$Y(t+h) = Y(t) + h\frac{dY}{dt}(t) + \tfrac{1}{2}h^2\frac{d^2Y}{dt^2}(t+s)\,,$$

for some s that lies between 0 and h when h is nonzero. It can also be expressed in its *Cauchy form* as

$$Y(t+h) = Y(t) + h\frac{dY}{dt}(t) + \int_0^h (h-s)\,\frac{d^2Y}{dt^2}(t+s)\,ds\,.$$

Either form allows one to show that

$$\left|Y(t+h) - Y(t) - h\frac{dY}{dt}(t)\right| \leq \tfrac{1}{2}h^2 M\,, \tag{1.6}$$

for any M that satisfies

$$\left|\frac{d^2Y}{dt^2}(t+s)\right| \leq M \quad \text{for every } s \text{ between } 0 \text{ and } h\,.$$

Whenever $Y(\,\cdot\,)$ is twice continuously differentiable over an interval containing t and $t+h$ then such an M exists. A bound such as (1.6) implies the remainder vanishes at least as fast as h^2 as h tends to zero. The notation $O(h^2)$ simply means such an M exists.

It is clear from (1.5) that for small positive h one has

$$Y(t+h) \approx Y(t) + h\frac{dY}{dt}(t)\,.$$

Because $Y(t)$ satisfies (1.1), this is the same as

$$Y(t+h) \approx Y(t) + hf(t, Y(t))\,.$$

If we let $t = t_n$ above (so that $t+h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) \approx Y(t_n) + hf(t_n, Y(t_n))\,.$$

Because y_n and y_{n+1} are approximations of $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + hf(t_n, y_n) \quad \text{for } n = 0, 1, \cdots, N-1\,,$$

which is exactly the Euler method (1.3).

One advantage of viewing the Euler method through the tangent line approximation (1.5) is that you gain some understanding of how its error behaves as you increase N, the number of time steps — or what is equivalent by (1.2), as you decrease h, the time step. Roughly speaking, if you halve the time step h then by (1.5) the error of the approximation made at each step will reduce by a factor of four while the number of steps you must take to get to a prescribed time (say t_F) will double. If we assume that the errors add (which is often the case) then the error at t_F will reduce by a factor of two. In other words, doubling the number of time steps will reduce the error by about a factor of two. Similarly, tripling the number of time steps will reduce the error by about a factor of three. Indeed, one can show (but we will not do so) that the error of the explicit Euler method is $O(h)$ over the interval $[t_I, t_F]$. Moreover, the error of the implicit Euler method behaves the same way. All such methods are called *first order methods.*

1.2.2. A Second Order Explicit Method. The second order Taylor approximation states that if $Y(t)$ is thrice continuously differentiable then

$$Y(t+h) = Y(t) + h\frac{dY}{dt}(t) + \tfrac{1}{2}h^2\frac{d^2Y}{dt^2}(t) + O(h^3)\,. \tag{1.7}$$

Here the $O(h^3)$ means that the remainder vanishes at least as fast as $|h|^3$ as h tends to zero. More precisely, one can use either the Lagrange or Cauchy form of the second order Taylor remainder to show that

$$\left| Y(t+h) - Y(t) - h\frac{dY}{dt}(t) - \tfrac{1}{2}h^2\frac{d^2Y}{dt^2}(t) \right| \leq \tfrac{1}{6}|h|^3 M\,,$$

for any M that satisfies

$$\left|\frac{d^3Y}{dt^3}(t+s)\right| \leq M \quad \text{for every } s \text{ between } 0 \text{ and } h\,.$$

Whenever $Y(t)$ is thrice continuously differentiable over an interval containing t and $t+h$, such an M exists.

It is clear from (1.7) that for small positive h one has

$$Y(t+h) \approx Y(t) + h\frac{dY}{dt}(t) + \tfrac{1}{2}h^2\frac{d^2Y}{dt^2}(t)\,. \tag{1.8}$$

Because $Y(t)$ satisfies (1.1), we see that

$$\begin{aligned}\frac{d^2Y}{dt^2}(t) = \frac{d}{dt}\left(\frac{dY}{dt}(t)\right) = \frac{d}{dt}f(t,Y(t)) &= \partial_t f(t,Y(t)) + \frac{dY}{dt}(t)\,\partial_y f(t,Y(t)) \\ &= \partial_t f(t,Y(t)) + f(t,Y(t))\,\partial_y f(t,Y(t))\,.\end{aligned}$$

Hence, equation (1.8) is the same as

$$Y(t+h) \approx Y(t) + hf(t,Y(t)) + \tfrac{1}{2}h^2\Big(\partial_t f(t,Y(t)) + f(t,Y(t))\,\partial_y f(t,Y(t))\Big)\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) \approx Y(t_n) + hf(t_n,Y(t_n)) + \tfrac{1}{2}h^2\Big(\partial_t f(t_n,Y(t_n)) + f(t_n,Y(t_n))\,\partial_y f(t_n,Y(t_n))\Big)\,.$$

Because y_n and y_{n+1} are approximations of $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$\begin{aligned}y_{n+1} = y_n + hf(t_n,y_n) + \tfrac{1}{2}h^2\Big(\partial_t f(t_n,y_n) + f(t_n,y_n)\,\partial_x f(t_n,y_n)\Big)& \\ \text{for } n = 0, 1, \cdots, N-1\,.&\end{aligned} \tag{1.9}$$

We call this the second-order Taylor-based method.

This method is clearly more complicated than the explicit Euler method. The hope is that its error behaves better as you increase N, the number of time steps — or what is equivalent by (1.2), as you decrease h, the time step. Indeed, if you halve the time step h then by the second-order Taylor approximation (1.7) the error of the approximation made at each step will reduce by a factor of eight while the number of steps you must take to get to a prescribed time (say t_F) will double. If we assume that the errors add (which is often the case) then the error at t_F will reduce by a factor of four. In other words, doubling the number of time steps will reduce the error by about a factor of four. Similarly, tripling the number of time steps will reduce the error by about a factor of nine. Indeed, one can show (but we will not do so) that the error of the second-order Taylor-based method is $O(h^2)$ over the interval $[t_I, t_F]$. All such methods are called *second order methods.*

1.2.3. Higher Order Explicit Methods. By generalizing what we did in the last subsection, one can use the n^{th} order Taylor approximation to derive an explicit numerical method whose error is $O(h^n)$ over the interval $[t_o, t_f]$ — a so-called n^{th} *order method.* However, the formulas for these methods grow in complexity. For example, the third order method is

$$\begin{aligned} y_{n+1} = y_n &+ hf(t_n,y_n) + \tfrac{1}{2}h^2\Big(\partial_t f(t_n,y_n) + f(t_n,y_n)\,\partial_y f(t_n,y_n)\Big) \\ &+ \tfrac{1}{6}h^3\Big[\partial_{tt} f(t_n,y_n) + 2f(t_n,y_n)\,\partial_{yt} f(t_n,y_n) + f(t_n,y_n)^2\partial_{yy} f(t_n,y_n) \\ &\qquad + \Big(\partial_t f(t_n,y_n) + f(t_n,y_n)\,\partial_y f(t_n,y_n)\Big)\partial_x f(t_n,y_n)\Big] \\ &\text{for } n = 0, 1, \cdots, N-1\,. \end{aligned} \tag{1.10}$$

This complexity makes them less practical for general algorithms than the next class of methods we will study.

1.3. Methods Based on Numerical Quadrature. The starting point for our next class of methods will be the Fundamental Theorem of Calculus — specifically, the fact that

$$Y(t+h) - Y(t) = \int_t^{t+h} \frac{dY}{dt}(s)\,ds\,.$$

Because $Y(t)$ satisfies (1.1), this becomes

$$Y(t+h) - Y(t) = \int_t^{t+h} f(s, Y(s))\,ds\,. \tag{1.11}$$

The idea now is to replace the definite integral on the right-hand side above with a numerical approximation — a so-called numerical quadrature. In particular, we will employ four basic numerical quadrature rules that are covered in most calculus courses: the left-hand rule, the trapezoidal rule, the midpoint rule, and the Simpson rule.

1.3.1. Explicit Euler Method Revisited. The left-hand rule approximates the definite integral on the right-hand side of (1.11) as

$$\int_t^{t+h} f(s, Y(s))\,ds = hf(t, Y(t)) + O(h^2)\,,$$

whereby you see that (1.11) becomes

$$Y(t+h) = Y(t) + hf(t, Y(t)) + O(h^2)\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) = Y(t_n) + hf(t_n, Y(t_n)) + O(h^2)\,.$$

Because y_n and y_{n+1} are approximations of $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + hf(t_n, y_n) \quad \text{for } n = 0, 1, \cdots, N-1\,,$$

which is exactly the forward Euler method (1.3).

1.3.2. Heun-Trapezoidal Method. The trapezoidal rule approximates the definite integral on the right-hand side of (1.11) as

$$\int_t^{t+h} f(s, Y(s))\, ds = \frac{h}{2}\big[f(t, Y(t)) + f(t+h, Y(t+h))\big] + O(h^3)\,.$$

whereby you see that (1.11) becomes

$$Y(t+h) = Y(t) + \frac{h}{2}\big[f(t, Y(t)) + f(t+h, Y(t+h))\big] + O(h^3)\,.$$

If you approximate $Y(t+h)$ by the forward Euler method then

$$Y(t+h) = Y(t) + \frac{h}{2}\big[f(t, Y(t)) + f\big(t+h, Y(t) + hf(t, Y(t))\big)\big] + O(h^3)\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) = Y(t_n) + \frac{h}{2}\big[f(t_n, Y(t_n)) + f\big(t_{n+1}, Y(t_n) + hf(t_n, Y(t_n))\big)\big] + O(h^3)\,.$$

Because y_n and y_{n+1} are approximations of $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + \frac{h}{2}\big[f(t_n, y_n) + f\big(t_{n+1}, y_n + hf(t_n, y_n)\big)\big] \quad \text{for } n = 0, 1, \cdots, N-1\,.$$

The book calls this the *Improved Euler* method. That name is sometimes used for other methods. Moreover, it is not very descriptive. We will call it the *Heun-Trapezoidal* method, which makes its origins clearer. In practice, it is implemented by the instructions

$$\begin{aligned} f_n &= f(t_n, y_n)\,, \qquad & \tilde{y}_{n+1} &= y_n + hf_n\,,\\ \tilde{f}_{n+1} &= f(t_{n+1}, \tilde{y}_{n+1})\,, \qquad & y_{n+1} &= y_n + \tfrac{1}{2}h[f_n + \tilde{f}_{n+1}]\,. \end{aligned}$$

1.3.3. Heun-Midpoint Method. The midpoint rule approximates the definite integral on the right-hand side of (1.11) as

$$\int_t^{t+h} f(s, Y(s))\, ds = hf\big(t + \tfrac{1}{2}h, Y(t + \tfrac{1}{2}h)\big) + O(h^3)\,.$$

whereby you see that (1.11) becomes

$$Y(t+h) = Y(t) + hf\big(t + \tfrac{1}{2}h, Y(t + \tfrac{1}{2}h)\big) + O(h^3)\,.$$

If you approximate $Y(t + \frac{1}{2}h)$ by the forward Euler method then

$$Y(t+h) = Y(t) + hf\big(t + \tfrac{1}{2}h, Y(t) + \tfrac{1}{2}hf(t, Y(t))\big) + O(h^3)\,.$$

If we let $t = t_n$ above (so that $t + h = t_{n+1}$) this is equivalent to

$$Y(t_{n+1}) = Y(t_n) + hf\big(t_{n+\frac{1}{2}}, Y(t_n) + \tfrac{1}{2}hf(t_n, Y(t_n))\big) + O(h^3)\,,$$

where $t_{n+\frac{1}{2}} = t_n + \frac{1}{2}h = t_I + (n + \frac{1}{2})h$. Because y_n and y_{n+1} are approximations of $Y(t_n)$ and $Y(t_{n+1})$ respectively, this suggests setting

$$y_{n+1} = y_n + hf\big(t_{n+\frac{1}{2}}, y_n + \tfrac{1}{2}hf(t_n, y_n)\big) \quad \text{for } n = 0, 1, \cdots, N-1\,.$$

This is called the *Heun-Midpoint* method. In practice, it is implemented by the instructions

$$\begin{aligned} f_n &= f(t_n, y_n)\,, & y_{n+\frac{1}{2}} &= y_n + \tfrac{1}{2}hf_n\,, \\ f_{n+\frac{1}{2}} &= f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})\,, & y_{n+1} &= y_n + hf_{n+\frac{1}{2}}\,. \end{aligned}$$

1.3.4. Runge-Kutta Method. The Simpson rule approximates the definite integral on the right-hand side of (1.11) as

$$\int_t^{t+h} f(s, Y(s))\, ds = \frac{h}{6}\big[f(t, Y(t)) + 4f\big(t + \tfrac{1}{2}h, Y(t + \tfrac{1}{2}h)\big) + f\big(t + h, Y(t+h)\big)\big] + O(h^5)\,,$$

whereby you see that (1.11) becomes

$$Y(t+h) = Y(t) + \frac{h}{6}\big[f(t, Y(t)) + 4f\big(t + \tfrac{1}{2}h, Y(t + \tfrac{1}{2}h)\big) + f\big(t + h, Y(t+h)\big)\big] + O(h^5)\,.$$

This leads to the Runge-Kutta method, which in practice, is implemented by the instructions

$$\begin{aligned} f_n &= f(t_n, y_n)\,, & \tilde{y}_{n+\frac{1}{2}} &= y_n + \tfrac{1}{2}hf_n\,, \\ \tilde{f}_{n+\frac{1}{2}} &= f(t_{n+\frac{1}{2}}, \tilde{y}_{n+\frac{1}{2}})\,, & y_{n+\frac{1}{2}} &= y_n + \tfrac{1}{2}h\tilde{f}_{n+\frac{1}{2}}\,, \\ f_{n+\frac{1}{2}} &= f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})\,, & \tilde{y}_{n+1} &= y_n + hf_{n+\frac{1}{2}}\,, \\ \tilde{f}_{n+1} &= f(t_{n+1}, \tilde{y}_{n+1})\,, & y_{n+1} &= y_n + \tfrac{1}{6}h\big[f_n + 2\tilde{f}_{n+\frac{1}{2}} + 2f_{n+\frac{1}{2}} + \tilde{f}_{n+1}\big]\,. \end{aligned}$$