

*Lecture Notes on
Numerical Differential
Equations: IVP*

Professor Biswa Nath Datta

**Department of Mathematical Sciences
Northern Illinois University
DeKalb, IL. 60115 USA**

E-mail: dattab@math.niu.edu

URL: www.math.niu.edu/~dattab

1 Initial Value Problem for Ordinary Differential Equations

We consider the problem of numerically solving a system of differential equations of the form

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha \text{ (given)} .$$

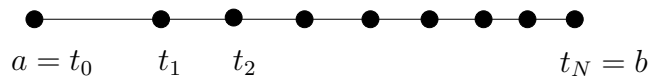
Such a problem is called the **Initial Value Problem** or in short **IVP**, because the initial value of the solution $y(a) = \alpha$ is given.

Since there are infinitely many values between a and b , we will only be concerned here to find approximations of the solution $y(t)$ at several specified values of t in $[a, b]$, rather than finding $y(t)$ at every value between a and b .

Denote

- y_i = An approximation of $y(t_i)$ at $t = t_i$.
- Divide $[a, b]$ into N equal subintervals of length h :

$$t_0 = a < t_1 < t_2 < \cdots < t_N = b.$$



- $h = \frac{b - a}{N}$ (step size)

Notation:

$y(t_i) \equiv$ Exact value at $t = t_i$.

$y_i \equiv$ Approximate value of $y(t_i)$.

The Initial Value Problem

Given

- (1) $y' = f(y, t)$, $a \leq t \leq b$
- (2) The initial value $y(t_0) = y(a) = \alpha$
- (3) The step-size h .

Find y_i (approximate value of $y(t_i)$), $i = 1, \dots, N$, where $N = \frac{b-a}{h}$.

We will briefly describe here the following well-known numerical methods for solving the IVP:

- The **Euler Method**
- The **Taylor Method** of higher order
- The **Runge-Kutta Method**
- The **Adams-Moulton Method**
- The **Milne Method**

etc.

We will also discuss the error behavior and convergence of these methods.

However, before doing so, we state a result **without proof**, in the following section on the **existence** and **uniqueness** of the solution for the IVP. The proof can be found in most books on ordinary differential equations.

Existence and Uniqueness of the Solution for the IVP

Theorem: (Existence and Uniqueness Theorem for the IVP).

The initial value problem:

$$\begin{cases} y' = f(t, y) \\ y(a) = \alpha \end{cases}$$

has a unique solution $y(t)$ for $a \leq t \leq b$, if $f(t, y)$ is continuous on the domain, given by $R = \{a \leq t \leq b, \quad -\infty < y < \infty\}$ and satisfies the following inequality:

$$|f(t, y) - f(t, y^*)| \leq L|y - y^*|,$$

Whenever (t, y) and $(t, y^*) \in R$.

Definition. The condition $|f(t, y) - f(t, y^*)| \leq L|y - y^*|$ is called the **Lipschitz Condition**. The number L is called a **Lipschitz Constant**.

Definition.

A set S is said to be convex if whenever (t_1, y_1) and (t_2, y_2) belong to S , the point $((1 - \lambda)t_1 + \lambda t_2, (1 - \lambda)y_1 + \lambda y_2)$ also belongs to S for each λ when $0 \leq \lambda \leq 1$.

Simplification of the Lipschitz Condition for the Convex Domain

If the domain happens to be a **convex set**, then the condition of the above Theorem reduces to

$$\left| \frac{\partial f}{\partial y}(t, y) \right| \leq L \text{ for all } (t, y) \in R.$$

Lipschitz Condition and Well-Posedness

Definition.

An IVP is said to be **well-posed** if a small perturbation in the data of the problem leads to only a small change in the solution.

Since numerical computation may very well introduce some perturbations to the problem, it is important that the problem that is to be solved is well-posed.

Fortunately, the Lipschitz condition is a sufficient condition for the IVP problem to be well-posed.

Theorem (Well-Posedness of the IVP problem).
If $f(t, y)$ Satisfies the Lipschitz Condition, then the IVP is well-posed.

2 The Euler Method

One of the simplest methods for solving the IVP is the classical Euler method.

The method is derived from the Taylor Series expansion of the function $y(t)$.

The function $y(t)$ has the following Taylor series expansion of order n at $t = t_{i+1}$:

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2!} y''(t_i) + \cdots + \frac{(t_{i+1} - t_i)^n}{n!} y^{(n)}(t_i) + \frac{(t_{i+1} - t_i)^{n+1}}{(n+1)!} y^{n+1}(\xi_i), \text{ where } \xi_i \text{ is in } (t_i, t_{i+1}).$$

Substitute $h = t_{i+1} - t_i$. Then

Taylor Series Expansion of $y(t)$ of order n at $t = t_{i+1}$:

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

For $n = 1$, this formula reduces to

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

The term $= \frac{h^2}{2!}y^{(2)}(\xi_i)$ is call the **remainder term**.

Neglecting the remainder term, we have

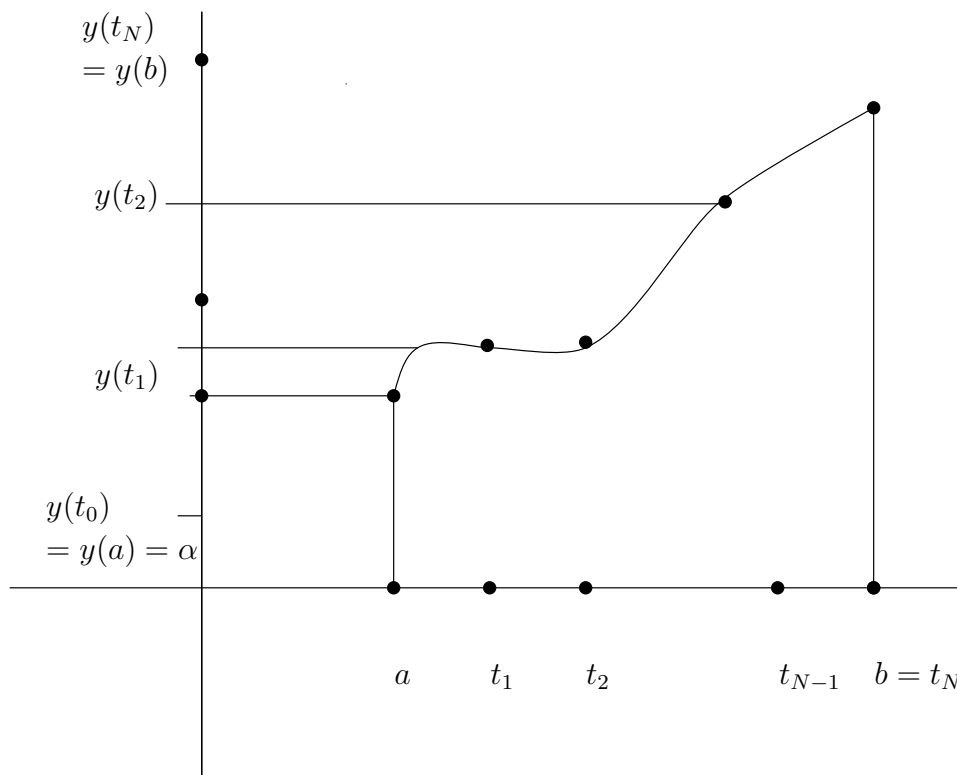
Euler's Method

$$y_{i+1} = y_i + hy'(t_i)$$

$$= y_i + hf(t_i, y_i), \quad i = 0, 1, 2, \dots, N - 1$$

This formula is known as the **Euler method** and now can be used to approximate $y(t_{i+1})$.

Geometrical Interpretation



Algorithm: Euler's Method for IVP

Input: (i). The function $f(t, y)$
(ii). The end points of the interval $[a, b]$: a and b
(iii). The initial value: $\alpha = y(t_0) = y(a)$

Output: Approximations y_{i+1} of $y(t_i + 1)$, $i = 0, 1, \dots, N - 1$.

Step 1. Initialization: Set $t_0 = a, y_0 = y(t_0) = y(a) = \alpha$.
and $N = \frac{b - a}{h}$.

Step 2. For $i = 0, 1, \dots, N - 1$ do
Compute $y_{i+1} = y_i + hf(t_i, y_i)$
End

Example: $y' = t^2 + 5, \quad 0 \leq t \leq 1.$

$$y(0) = 0, \quad h = 0.25.$$

Find $y_1, y_2, y_3,$ and $y_4,$ approximations of $y(0.25), y(0.50), y(0.75),$ and $y(1),$ respectively.

The **points of subdivisions** are: $t_0 = 0, t_1 = 0.25, t_2 = 0.50, t_3 = 0.75$ and $t_4 = 1.$

$$i = 0: t_1 = t_0 + h = 0.25$$

$$\begin{aligned} y_1 &= y_0 + hf(t_0, y_0) = 0 + .25(5) = 1.25 \\ (y(0.25)) &= 1.2552). \end{aligned}$$

$$i = 1: t_2 = t_1 + h = 0.50$$

$$\begin{aligned} y_2 &= y_1 + hf(t_1, y_1) \\ &= 1.25 + 0.25(t_1^2 + 5) = 1.25 + 0.25((0.25)^2 + 5) \\ &= 2.5156 \\ (y(0.5)) &= 2.5417). \end{aligned}$$

$$i = 2: t_3 = t_2 + h = 0.75$$

$$\begin{aligned} y_3 &= y_2 + hf(t_2, y_2) \\ &= 2.5156 + .25((.5)^2 + 5) = 3.8281 \\ (y(0.75)) &= 3.8906). \end{aligned}$$

etc.

Example: $y' = t^2 + 5$, $0 \leq t \leq 2$,

$$y(0) = 0, h = 0.5$$

So, the **points of subdivisions** are: $t_0 = 0, t_1 = 0.5, t_2 = 1, t_3 = 1.5, t_4 = 2.$

We compute y_1, y_2, y_3 , and y_4 , which are, respectively, approximations to $y(0.5), y(1), y(1.5)$, and $y(2)$.

$$i = 0 : \quad y_1 = y_0 + hf(t_0, y_0) = y(0) + hf(0, 0) = 0 + 0.5 \times 5 = 2.5$$
$$(y(0.50) = 2.5417).$$

$$i = 1 : \quad y_2 = y_1 + hf(t_1, y_1) = 2.5 + 0.5((0.5)^2 + 5) = 5.1250$$
$$(y(1) = 5.3333).$$

$$i = 2 : \quad y_3 = y_2 + hf(t_2, y_2) = 5.1250 + 0.5(t_2^2 + 5) = 5.1250 + 0.5(1 + 5) = 8.1250$$
$$(y(1.5) = 8.6250).$$

etc.

The Errors in Euler's Method

The approximations obtained by a numerical method to solve the IVP are usually subjected to three types of errors:

- **Local Truncation Error**
- **Global Truncation Error**
- **Round-off Error**

We will not consider the round-off error for our discussions below.

The **local discretization error** is the error made at a **single step** due to the truncation of the series used to solve the problem.

Recall that the Euler Method was obtained by truncating the Taylor series

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \cdots$$

after two terms. Thus, in obtaining Euler's method, the first term neglected is $\frac{h^2}{2}y''(t)$.

So the **local error in Euler's method is:** $E_L^E = \frac{h^2}{2}y''(\xi_i)$,

where ξ_i lies between t_i and t_{i+1} . In this case, we say **that the local error is of order h^2 , written as $O(h^2)$** . Note that the local error E_L^E converges to zero as $h \rightarrow 0$.

Global error is the difference between the true solution $y(t_i)$ and the approximate solution y_i at $t = t_i$. Thus, **Global error** = $y(t_i) - y_i$. Denote this by E_G^E .

The following theorem shows that the global error, E_G^E , is of order h .

Theorem: (Global Error Bound for the Euler Method)

Let $y(t)$ be the unique solution of the IVP: $y' = f(t, y); y(a) = \alpha$.

$$a \leq t \leq b, -\infty < y < \infty,$$

Let L and M be two numbers such that

$$\left| \frac{\partial f(t, y)}{\partial y} \right| \leq L, \text{ and } |y''(t)| \leq M \text{ in } [a, b].$$

Then the global error E_G^E at $t = t_i$ satisfies

$$|E_G^E| = \left| y(t_i) - y_i \right| \leq \frac{hM}{2L} (e^{L(t_i-a)} - 1).$$

Thus, The global error bound for Euler's method depends upon h , whereas the local error depends upon h^2 .

Proof of the above theorem can be found in the book by G.W. Gear, **Numerical Initial Value Problems in Ordinary Differential Equations**, Prentice Hall, Inc. (1971).

Remark. Since the exact solution $y(t)$ of the IVP is not known, the above bound may not

be of practical importance as far as knowing how large the error can be a priori. However, from this error bound, we can say that the *Euler method can be made to converge faster by decreasing the step-size*. Furthermore, if the equalities, L and M of the above theorem can be found, then we can determine what step-size will be needed to achieve a certain accuracy, as the following example shows.

Example:
$$\frac{dy}{dt} = \frac{t^2 + y^2}{2}, y(0) = 0$$
$$0 \leq t \leq 1, -1 \leq y(t) \leq 1.$$

Determine how small the step-size should be so that the error does not exceed $\epsilon = 10^{-4}$.

Compute L :

Since $f(t, y) = \frac{t^2 + y^2}{2}$, we have

$$\frac{\partial f}{\partial y} = y$$

Thus, $\left| \frac{\partial f}{\partial y} \right| \leq 1$ for all y , giving $\boxed{L=1}$.

Find M :

To find M , we compute the second-derivative of $y(t)$ as follows:

$$y' = \frac{dy}{dt} = f(t, y) \text{ (Given)}$$

$$\begin{aligned} \text{By implicit differentiation, } y'' &= \frac{\partial f}{\partial t} + f \frac{\partial f}{\partial y} \\ &= t + \left(\frac{t^2 + y^2}{2} \right) y = t + \frac{y}{2}(t^2 + y^2) \end{aligned}$$

$$\text{So, } |y''(t)| = \left| t + \frac{y}{2}(t^2 + y^2) \right| \leq 2, \text{ for } -1 \leq y \leq 1.$$

$$\text{Thus, } \boxed{M=2,}$$

$$\begin{aligned} \text{thus, Global Error Bound: } |E_G^E| \text{ at } t = t_i &= |y(t_i) - y_i| \leq \frac{2h}{2L}(e^{t_i} - 1) \\ &= h(e^{t_i} - 1) = h(e - 1). \end{aligned}$$

So, for the error not to exceed 10^{-4} , we must have:

$$h(e - 1) < 10^{-4} \text{ or } h < \frac{10^{-4}}{e - 1} \approx 5.8198 \times 10^{-5}.$$

3 High-order Taylor Methods

Recall that the Taylor's series expansion of $y(t)$ of degree n is given by

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

Now,

$$(i) \quad y'(t) = f(t, y(t)) \text{ (Given).}$$

$$(ii) \quad y''(t) = f'(t, y(t)).$$

$$\text{In general } (iii) \quad y^{(i)}(t) = f^{(i-1)}(t, y(t)), i = 1, 2, \dots, n.$$

Thus,

$$\begin{aligned} y(t_{i+1}) &= y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} f'(t_i, y(t_i)) + \cdots + \frac{h^{n-1}}{(n-1)!} f^{(n-1)}(t_i, y_i) + \frac{h^n}{n!} f^{(n-1)}(t_i, y(t_i)) + \\ &\frac{h^{n+1}}{(n+1)!} f^{(n-1)}(\xi_i, y(\xi_i)) \\ &= y(t_i) + h \left[f(t_i, y(t_i)) + \frac{h}{2} f'(t_i, y(t_i)) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_i, y(t_i)) \right] + \text{Remainder Term} \end{aligned}$$

Neglecting the remainder term the above formula can be written in compact form as follows:

$y_{i+1} = y_i + hT_k(t_i, y_i)$, $i = 0, 1, \dots, N-1$, where $T_k(t_i, y_i)$ is defined by:

$$T_k(t_i, y_i) = f(t_i, y_i) + \frac{h}{2} f'(t_i, y_i) + \cdots + \frac{h^{k-1}}{k!} f^{(k-1)}(t_i, y_i)$$

So, if we truncate the Taylor Series after $(k+1)$ terms and use the truncated series to obtain the approximating of y_{i+1} of $y(t_{i+1})$, we have the following **of k-th order Taylor's algorithm for the IVP**.

Taylor's Algorithm of order k for IVP

- Input:**
- (i) The function $f(t, y)$
 - (ii) The end points: a and b
 - (iii) The initial value: $\alpha = y(t_0) = y(a)$
 - (iv) The order of the algorithm: k
 - (v) The step size: h

Step 1 Initialization: $t_0 = a, y_0 = \alpha, N = \frac{b-a}{h}$

Step 2. For $i = \dots, N - 1$ do

2.1 Compute $T_k(t_i, y_i) = f(t_i, y_i) + \frac{h}{2}f'(t_i, y_i) + \dots + \frac{h^{k-1}}{k!}f^{(k-1)}(t_i, y_i)$

2.2 Compute $y_{i+1} = y_i + hT_k(t_i, y_i)$

End

Note: With $k = 1$, the above formula for y_{i+1} , reduces to Euler's method.

Example:

$$y' = y - t^2 + 1, \quad 0 \leq t \leq 2, \quad y(0) = 0.5, \quad h = 0.2.$$

The **points of division** are:

$$t_0 = 0, t_1 = 0.2, t_2 = 0.4, t_3 = 0.6, t_4 = 0.8, t_5 = 1, \text{ and so on.}$$

$$f(t, y(t)) = y - t^2 + 1 \text{ (Given).}$$

$$\begin{aligned} f'(t, y(t)) &= \frac{d}{dt}(y - t^2 + 1) = y' - 2t \\ &= y - t^2 + 1 - 2t \end{aligned}$$

$$f''(t, y(t)) = \frac{d}{dt}(y - t^2 + 1 - 2t) = y - t^2 + 1 - 2t - 2 = y - t^2 - 2t - 1$$

so,

$$\begin{aligned} y_1 &= y_0 + hf(t_0, y(t_0)) + \frac{h^2}{2}f'(t_0, y(t_0)) \\ &= 0.5 + 0.2 \times 1.5 + \frac{(0.2)^2}{2}(0.5 + 1) = 0.8300 \text{ (approximate value of } y(0.2)). \end{aligned}$$

$$y_2 = 1.215800 \text{ (approximate value of } y(0.4)).$$

etc.

4 Runge-Kutta Methods

- The Euler's method is the simplest to implement; however, even for a reasonable accuracy the step-size h needs to be very small.
- The difficulties with higher order Taylor's series methods are that the derivatives of higher orders of $f(t, y)$ need to be computed, which are very often difficult to compute; indeed, $f(t, y)$ is not even explicitly known in many areas.

The Runge-Kutta methods aim at achieving the accuracy of higher order Taylor series methods without computing the higher order derivatives.

We first develop the simplest one: **The Runge-Kutta Methods of order 2.**

The Runge-Kutta Methods of order 2

Suppose that we want an expression of the approximation y_{i+1} in the form:

$$y_{i+1} = y_i + \alpha_1 k_1 + \alpha_2 k_2, \quad (4.1)$$

$$\text{where } k_1 = hf(t_i, y_i), \quad (4.2)$$

and

$$k_2 = hf(t_i + \alpha h, y_i + \beta k_1). \quad (4.3)$$

The constants α_1 and α_2 and α and β are to be chosen so that the formula is as accurate as the Taylor's Series Method of as high as possible.

To develop the method we need an important result from Calculus: **Taylor's series for a function to two variables.**

Taylor's Theorem for a Function of Two Variables

Let $f(t, y)$ and its partial derivatives of orders up to $(n + 1)$ are continuous in the domain $D = \{(t, y) | a \leq t \leq b, c \leq y \leq d\}$.

Then

$$f(t, y) = f(t_0, y_0) + \left[(t - t_0) \frac{\partial f}{\partial t}(t_0, y_0) + (y - y_0) \frac{\partial f}{\partial y}(t_0, y_0) \right] + \dots$$

$$+ \left[\frac{1}{n!} \sum_{h=0}^n \binom{n}{h} (t - t_0)^{n-h} (y - y_0)^h \frac{\partial^n f}{\partial t^{n-h} \partial y^h}(t_0, y_0) \right] + R_n(t, y),$$

where $R_n(t, y)$ is the remainder after n terms and involves the partial derivative of order $n + 1$.

Using the above theorem with $n = 1$, we have (setting $t = t_i + \alpha h$, $y = y_i + \beta k_1$, and $t_0 = t_i$ and $y_0 = y_i$):

$$f(t_i + \alpha h, y_i + \beta k_1) = f(t_i, y_i) + \alpha h \frac{\partial f}{\partial t}(t_i, y_i) + \beta k_1 \frac{\partial f}{\partial y}(t_i, y_i) \quad (4.4)$$

From (4.4) and (4.3), we obtain

$$\frac{k_2}{h} = f(t_i, y_i) + \alpha h \frac{\partial f}{\partial t}(t_i, y_i) + \beta k_1 \frac{\partial f}{\partial y}(t_i, y_i). \quad (4.5)$$

Again, substituting the value of k_1 from (4.2) and k_2 from (4.3) in (4.1) we get (after some rearrangement):

$$\begin{aligned} y_{i+1} &= y_i + \alpha_1 h f(t_i, y_i) + \alpha_2 h \left[f(t_i, y_i) + \alpha h \frac{\partial f}{\partial t}(t_i, y_i) + \beta h f(t_i, y_i) \frac{\partial f}{\partial y}(t_i, y_i) \right] \\ &= y_i + (\alpha_1 + \alpha_2) h f(t_i, y_i) + \alpha_2 h^2 \left[\alpha \frac{\partial f}{\partial t}(t_i, y_i) + \beta f(t_i, y_i) \frac{\partial f}{\partial y}(t_i, y_i) \right] \end{aligned} \quad (4.6)$$

Also, note that $y(t_{i+1}) = y(t_i) + h f(t_i, y_i) + \frac{h^2}{2} \left(\frac{\partial f}{\partial t}(t_i, y_i) + f(t_i, y_i) \frac{\partial f}{\partial y}(t_i, y_i) \right) +$ higher order terms.

So, neglecting the higher order terms, we can write

$$y_{i+1} = y_i + h f(t_i, y_i) + \frac{h^2}{2} \left(\frac{\partial f}{\partial t}(t_i, y_i) + f(t_i, y_i) \frac{\partial f}{\partial y}(t_i, y_i) \right). \quad (4.7)$$

If we want (4.6) and (4.7) to agree for numerical approximations, then we must have

- $\alpha_1 + \alpha_2 = 1$ (comparing the coefficients of $h f(t_i, y_i)$).
- $\alpha_2 \alpha = \frac{1}{2}$ (comparing the coefficients of $h^2 \frac{\partial f}{\partial t}(t_i, y_i)$).
- $\alpha_2 \beta = \frac{1}{2}$ (comparing the coefficients of $h^2 f(t_i, y_i) \frac{\partial f}{\partial y}(t_i, y_i)$).

Since the number of unknowns here exceeds the number of equations, there are infinitely many possible solutions. The simplest solution is:

$$\alpha_1 = \alpha_2 = \frac{1}{2}, \quad \alpha = \beta = 1.$$

With these choices we can generate y_{i+1} from y_i as follows. The process is known as the **Modified Euler's Method**.

Generating y_{i+1} from y_i in Modified Euler's Method

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2),$$

where $k_1 = hf(t_i, y_i)$

$$k_2 = hf(t_i + h, y_i + k_1).$$

or

$$y_{i+1} = y_i + \frac{h}{2} \left[f(t_i, y_i) + f(t_i + h, y_i + hf(t_i, y_i)) \right]$$

Algorithm: The Modified Euler Method

Inputs: The given function: $f(t, y)$
 The end points of the interval: a and b
 The step-size: h
 The initial value $y(t_0) = y(a) = \alpha$

Outputs: Approximations y_{i+1} of $y(t_{i+1}) = y(t_0 + ih)$,
 $i = 0, 1, 2, \dots, N - 1$

Step 1 (Initialization)
 Set $t_0 = a$, $y_0 = y(t_0) = y(a) = \alpha$

$$N = \frac{b - a}{h}$$

Step 2 For $i = 0, 1, 2, \dots, N - 1$ do
 Compute $k_1 = hf(t_i, y_i)$
 Compute $k_2 = hf(t_i + h, y_i + k_1)$
 Compute $y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)$.

End

Example:

$$y' = e^t, y(0) = 1, h = 0.5, 0 \leq t \leq 1$$

$$t_0 = 0, t_1 = 0.5, t_2 = 1$$

$$\begin{aligned} \mathbf{i = 0 :} \quad k_1 &= hf(t_0, y_0) = 0.5e^{t_0} = 0.5 \\ k_2 &= hf(t_0 + h, y_0 + k_1) = 0.5(e^{t_0+h}) = 0.5e^{0.5} = 0.8244 \\ y_1 &= y_0 + \frac{1}{2}(k_1 + k_2) = 1 + 0.5(0.5 + 0.8244) = 1.6622 \\ (y(0.5)) &= e^{0.5} = 1.6487 \\ \mathbf{i = 1 :} \quad k_1 &= hf(t_1, y_1) = 0.5e^{t_1} = 0.5e^{0.5} = 0.8244 \\ k_2 &= hf(t_1 + h, y_1 + k_1) = 0.5e^{0.5+0.5} = 0.5e = 1.3591 \\ y_2 &= y_1 + \frac{1}{2}(k_1 + k_2) = 1.6622 + \frac{1}{2}(0.8244 + 1.3591) = 2.7539 \\ (y(1)) &= 2.7183. \end{aligned}$$

Example: Given: $y' = t + y$, $y(0) = 1$, compute y_1 (approximation to $y(0.01)$) and y_2 (approximation to $y(0.02)$) by using Modified Euler Method.

$$h = 0.01, y_0 = y(0) = 1.$$

$$i = 0: y_1 = y_0 + \frac{1}{2}(k_1 + k_2)$$

$$k_1 = hf(t_0, y_0) = 0.01(0 + 1) = 0.01$$

$$k_2 = hf(t_0 + h, y_0 + k_1) = 0.01 \times f(0.01, 1 + 0.01) \\ = 0.01 \times (0.01 + 1.01) = 0.01 \times 1.02 = 0.0102$$

$$\text{Thus } y_1 = 1 + \frac{1}{2}(0.01 + 0.0102) = 1.0101 \quad (\text{Approximate value of } y(0.01))$$

$$i = 1: y_2 = y_1 + \frac{1}{2}(k_1 + k_2)$$

$$k_1 = hf(t_1, y_1)$$

$$= 0.01 \times f(0.01, 1.0101) = 0.01 \times (0.01 + 1.0101) \\ = 0.0102$$

$$k_2 = hf(t_1 + h, y_1 + k_1)$$

$$= 0.01 \times f(0.02, 1.0101 + 0.0102) = 0.01 \times (0.02 + 1.0203) \\ = -0.0104$$

$$y_2 = 1.0101 + \frac{1}{2}(0.0102 + 0.0104) = 1.0204 \quad (\text{Approximate value of } y(0.02)).$$

Local Error in the Modified Euler Method

Since in deriving the modified Euler method, we neglected the terms involving h^3 and higher powers of h , the **local error for this method is $O(h^3)$** . **Thus with the modified Euler method, we will be able to use larger step-size h than the Euler method to obtain the same accuracy.**

The Midpoint and Heun's Methods

In deriving the modified Euler's Method, we have considered only one set of possible values of $\alpha_1, \alpha_2, \alpha_1$ and β . We will now consider two more sets of values.

- $\alpha = 0, \alpha_2 = 1, \alpha = \beta = \frac{1}{2}$.

This gives us the **Midpoint Method**.

The Midpoint Method

$$y_{i+1} = y_i + k_2$$

where $k_1 = hf(t_i, y_i)$,

and

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right)$$

or

$$y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}f(t_i, y_i)\right), i = 0, 1, \dots, N-1.$$

Example

$$y' = e^t, y(0) = 1, h = 0.5, 0 \leq t \leq 1.$$

$t_0 = 0, t_1 = 0.5, t_2 = 1$

Compute y_1 , an approximation to $y(0.5)$:

$$\begin{aligned} \mathbf{i} = \mathbf{0} : \quad k_1 &= hf(t_0, y_0) = 0.5e^{t_0} = 0.5e^0 = 0.5 \\ k_2 &= hf\left(t_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) = 0.5e^{\frac{0.5}{2}} = 0.6420 \\ y_1 &= y_0 + k_2 = 1 + 0.6420 = 1.6420 \\ (y(0.5) &= 1.6487). \end{aligned}$$

Compute y_2 , an approximation of $y(1)$:

$$\begin{aligned} \mathbf{i} = \mathbf{1} : \quad k_1 &= hf(t_1, y_1) = 0.5e^{0.5} = 0.8244 \\ k_2 &= hf\left(t_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}\right) = 0.5e^{0.75} = 1.0585 \\ y_2 &= y_1 + k_2 = 1.6420 + 1.0585 = 2.7005 \\ (y(1) &= e = 2.7183) \end{aligned}$$

- $\alpha_1 = \frac{1}{4}$, $\beta_1 = \frac{3}{4}$, $\alpha = \beta = \frac{2}{3}$

Then we have **Heun's Method**.

Heun's Method

$$y_{i+1} = y_i + \frac{1}{4}k_1 + \frac{3}{4}k_2$$

where $k_1 = hf(t_i, y_i)$

$$k_2 = hf\left(t_i + \frac{2}{3}h, y_i + \frac{2}{3}k_1\right)$$

or

$$y_{i+1} = y_i + \frac{h}{4}f(t_i, y_i) + \frac{3h}{4}f\left(t_i + \frac{2}{3}h, y_i + \frac{2h}{3}f(t_i, y_i)\right), i = 0, 1, \dots, N - 1$$

Heun's Method and the **Modified Euler's Method** are classified as the **Runge-Kutta methods of order 2**.

The Runge-Kutta Method of order 4.

A method widely used in practice is the Runge-Kutta method of order 4. It's derivation is complicated. We will just state the method without proof.

Algorithm: The Runge-Kutta Method of Order 4

- Inputs:** $f(t, y)$ - the given function
 a, b - the end points of the interval
 α - the initial value $y(t_0)$
 h - the step size
- Outputs:** The approximations y_{i+1} of $y(t_{i+1})$, $i = 0, 1, \dots, N - 1$
- Step 1: (Initialization)**
Set $t_0 = a$, $y_0 = y(t_0) = y(a) = \alpha$
 $N = \frac{b - a}{h}$.
- Step 2: (Computations of the Runge-Kutta Coefficients)**
For $i = 0, 1, 2, \dots, n$ do
 $k_1 = hf(t_i, y_i)$
 $k_2 = hf(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1)$
 $k_3 = hf(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_2)$
 $k_4 = hf(t_i + h, y_i + k_3)$
- Step 3: (Computation of the Approximate Solution)**
Compute: $y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

The Local Truncation Error: The local truncation error of the Runge-Kutta Method of order 4 is $O(h^5)$.

Example:

$$y' = t + y, \quad y(0) = 1$$

$$h = 0.01$$

Let's compute $y(0.01)$ using the Runge-Kutta Method of order 4.

$i = 0$

$$y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where $k_1 = hf(t_0, y_0) = 0.01f(0, 1) = 0.01 \times 1 = 0.01$.

$$k_2 = hf\left(t_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) = 0.01f\left(\frac{0.01}{2}, 1 + \frac{0.01}{2}\right) = 0.01\left[\frac{0.01}{2} + \frac{1 + 0.01}{2}\right] = 0.0101.$$

$$k_3 = hf\left(t_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right) = h\left(t_0 + \frac{h}{2} + y_0 + \frac{k_2}{2}\right) = 0.0101005.$$

$$k_4 = hf(t_0 + h, y_0 + k_3) = h(t_0 + h + y_0 + k_3) = 0.01020100$$

$$\text{So, } y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.010100334$$

and so on.

5 Multistep Methods (Explicit-type)

The methods we have discussed so far, Euler's method and Runge-Kutta methods of order 2 and order 4, are **single-step methods**; because, given $y(t_0) = f(t_0, y_0) = \alpha$, y_{i+1} is computed from y_i . However, if $f(t, y)$ is known at $m + 1$ points, say, at $t_i, t_{i-1}, t_{i-2}, \dots, t_{i-m}$; that is, if $f(t_k, y_k)$, $k = i, i-1, \dots, i-m$ are known, then we can develop higher-order methods to compute y_{i+1} .

One such class of methods can be developed based on numerical integration as follows:

From

$$y' = f(t, y)$$

we have, by integrating from t_i to t_{i+1} ,

$$\int_{t_i}^{t_{i+1}} y' dt = \int_{t_i}^{t_{i+1}} f(t, y) dt$$

or

$$y_{i+1} - y_i = \int_{t_i}^{t_{i+1}} f(t, y) dt$$

or

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t, y) dt$$

The integral on the right-hand side is now evaluated by interpolating $f(t, y)$ at the $(m + 1)$ given points, $t_i, t_{i-1}, \dots, t_{i-m}$ using **Newton's backward difference** formula. The interpolating polynomial $P_m(t)$ in this case is

$$P_m(t) = \sum_{k=0}^m (-1)^k \binom{-s}{k} \Delta^k f_{i-k},$$

where $s = \frac{t-t_i}{h}$

and f_{i-k} stands for $f(t_k, y(t_k))$.

The integral now can be evaluated by substituting the above expression of $P_m(t)$ for $f(t, y)$.

The resulting formula is known as the **Adams-Bashforth formula**.

- In case $m = 0$, this leads to **Euler's method**.
- In case $m = 3$, after simplification, we obtain **Four-step Adams-Bashforth Formula**:

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}),$$

where $f_k = f(t_k, y(f_k))$, $k = i, i - 1, i - 2$, and $i - 3$.

- **Error term for four-step Adams-Bashforth formula** is of $O(h^5)$.

Specifically, it is: $E_{AB}^4 = \frac{251}{320}h^5y^5(\xi)$, where $t_{n-3} < \xi < t_{n+1}$.

The three-step **Adams-Bashforth formula** is an example of an explicit multistep method, because to compute y_{i+1} , information that we need are the values $f(t_i, y_i)$, $f(t_{i-1}, y_{i-1})$, $f(t_{i-2}, y_{i-2})$ and $f(t_{i-3}, y_{i-3})$ which are explicitly known beforehand.

Other explicit higher multi-step methods can similarly be developed.

Implicit Multistep Methods

Consider now integrating $\int_{t_i}^{t_{i+1}} f(t, y) dt$ using an interpolating polynomial of degree $m + 1$, $P_{m+1}(t)$ that interpolates at $(m+2)$ points (rather than $(m+1)$ points as in Adams-Bashforth formula): $t_{i+1}, t_i, \dots, t_{i-m}$.

Then using Newton's backward formula, we have

$$P_{m+1}(t) = \sum_{k=0}^{m+1} (-1)^k \binom{1-s}{k} \Delta^k f_{i+1} - k.$$

- For $m = 2$, this yields **Three-step Adams-Moulton Formula**:

$$y_{i+1} = y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}).$$

Since $f_k = f(t_k, y(t_k))$, $k = i + 1$, $i - 1$, and $i - 2$, we have

$$\begin{aligned} y_{i+1} &= y_i + \frac{h}{24}(9f(t_{i+1}, y(t_{i+1})) \\ &\quad + 19f(t_i, y(t_i)) - 5f(t_{i-1}, y(t_{i-1})) \\ &\quad + f(t_{i-2}, y(t_{i-2})), \end{aligned}$$

The formulas of the above type are **implicit multi-step formulas**; because computation of y_{i+1} requires $y(t_{i+1})$.

- **Error Term:** The error term of three-step implicit Adams-Moulton formula is also of $O(h^5)$, but with a smaller constant term. Specifically, it is

$$E_{AM}^3 = -\frac{19}{720}h^5y^5(\xi), \text{ where } \xi \text{ lies between } t_{n-2} \text{ and } t_{n+1}.$$

6 Predictor-Corrector Methods

A class of methods called **Predictor-Corrector methods**, is based on the following principle:

- **Predict** the initial value of $y_{i+1}^{(0)}$ by using an explicit method and then
- **Correct** $y_{i+1}^{(0)}$ **iteratively** to $y_{i+1}^{(1)}, y_{i+1}^{(2)}, \dots, y_{i+1}^{(k)}$ by using an implicit formula until two successive iterations agree to each other.

As an example, we develop below a predictor-corrector method with **Euler's method** as the **predictor** and a **corrector formula** based on integrating $\int_{t_i}^{t_{i+1}} f(t, y) dt$ using **trapezoidal rule**.

- Euler's method gives

$$y_{i+1}^{(0)} = y_i + hf(t_i, y_i) \quad \text{predicted value}$$

- Trapezoidal rule of integration applied to $\int_{y_i}^{y_{i+1}} y' dt$ gives

$$y_{i+1} = y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y(t_{i+1}))].$$

(Note this is an **implicit formula**).

So, we can take

$$y_{i+1}^{(1)} = y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{(0)})], \quad \text{(Corrected value by iteration)}$$

where $y_{i+1}^{(0)}$ is computed from Euler's method as shown above.

- The general iteration for correcting $y_{i+1}^{(0)}$ is then given by

$$y_{i+1}^{(k)} = y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{(k-1)})], \quad k = 1, 2, 3, \dots$$

We then have a **second-order predictor-corrector method**:

Algorithm: Euler-Trapezoidal Predictor-Corrector Method

Input:

- (i) $y' = f(t, y)$, $y(t_0) = \alpha$
- (ii) Step-size h
- (iii) Error tolerance ϵ .
- (iv) Points $t_{i+1} = t_i + h$, $i = 0, 1, \dots, N - 1$ at which approximations to $y(t_i)$ are sought.

Output: Approximations $y_{i+1}^{(k)}$, $k = 1, 2, \dots$ of $y(t_{i+1})$ for a fixed $i = 0, 1, 2, \dots, N - 1$.

For $i = 0, 1, 2, \dots, N - 1$ do

- Step 1.** Compute $y_{i+1}^{(0)}$ from $y_{i+1}^{(0)} = y_i + hf(t_i, y_i)$.
- Step 2.** For $k = 1, 2, \dots$ do
 - 2.1** $y_{i+1}^{(k)} = y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{(k-1)})]$
 - 2.2** Stop when the relative change:

$$\frac{|y_{i+1}^{(k)} - y_{i+1}^{(k-1)}|}{|y_{i+1}^{(k)}|} < \epsilon.$$
 - 2.3** Accept the current value of $y_{i+1}^{(k)}$ as y_{i+1} .
- End

End

Example: $y' = t + y, y(0) = 1$

$$h = 0.01$$

$i = 0$.

Step 1. Predict using formula in Step 1.

$$\begin{aligned}y_1^{(0)} &= y_0 + hf(t_0, y_0) \\ &= 1 + 0.01(t_0, y_0) \\ &= 1 + 0.01(1) = 1 + 0.01 = 1.01\end{aligned}$$

Step 2. Correct using formula in Step 2.1.

$$\begin{aligned}\mathbf{k} = \mathbf{0}: \quad y_1^{(1)} &= y_0 + \frac{h}{2}[f(t_0, y_0) + f(t_1, y_1^{(0)})] \\ &= 1 + \frac{0.01}{2}[(t_0 + y_0) + (t_1 + y_1^{(0)})] \\ &= 1 + \frac{0.01}{2}[1 + (0.01 + 1.01)] \\ &= 1.0101\end{aligned}$$

$$\begin{aligned}\mathbf{k} = \mathbf{1}: \quad y_1^{(2)} &= y_0 + \frac{h}{2}[f(t_0, y_0) + f(t_1, y_1^{(1)})] \\ &= 1 + \frac{0.01}{2}[1 + 0.01 + 1.0101] \\ &= 1.0101\end{aligned}$$

$y_1^{(2)}$ is accepted as y_1 , the approximate value of $y(0.01)$.

When does the iteration in 2.1 converge?

It can be shown that if $f(t, y)$ and $\frac{\partial f}{\partial y}$ are continuous in $[a, b]$, then the iteration will converge if h is chosen so that $\left| \frac{\partial f}{\partial y} \right| h < 2$.

Note that for the above example, $\frac{\partial f}{\partial y} = 1$, so the iteration will converge if $h < 2$. Since we had $h = 0.01 < 2$, the iteration converged after 1 iteration.

Higher-order Predictor-corrector Methods

Higher-order predictor corrector methods can be developed by combining an explicit multistep method (corrector) with an implicit multistep method (predictor).

For example, when a four-step explicit Adams-Bashforth formula is combined with a three-step implicit Adams-Moulton formula, the result is **Adams-Moulton predictor-corrector method**.

Here:

- **Predictor:** Four-Step Adams-Bashforth Formula
- **Corrector:** Three-Step Adams-Moulton Formula

Algorithm: Adams-Bashforth-Moulton Predictor-Corrector Method

Inputs: $f(t, y)$ - the given function
 h - the step size
 $f(t_0, y(t_0)), f(t_1, y(t_1)), f(t_2, y(t_2)), f(t_3, y(t_3))$
- values of $f(t, y)$ at four given points t_0, t_1, t_2 , and t_3 .
 ϵ - tolerance.
points $t_{i+1} = t_i + h, i = 0, 1, 2, \dots, N - 1$ at which approximations to $y(t_{i+1})$ are sought.

Outputs: Approximations of $y_{i+1}^{(k)}, k = 1, 2, 3, \dots$, for fixed $i = 0, 1, 2, \dots, N - 1$.

For $i = 0, 1, 2, \dots, N - 1$ do

Step 1: Compute $y_{i+1}^{(0)}$:

$$y_{i+1}^{(0)} = y_i + \frac{h}{24}[55f(t_i, y(t_i)) - 59f(t_{i-1}, y(t_{i-1})) \\ + 37f(t_{i-2}, y(t_{i-2})) - 9f(t_{i-3}, y(t_{i-3}))]$$

Step 2: For $k = 1, 2, 3, \dots$ do

$$y_{i+1}^{(k)} = y_i + \frac{h}{24}[9f(t_{i+1}, y_{i+1}^{(k-1)}) + 19f(t_i, y(t_i))]$$

Step 3: Stop if $|y_{i+1}^{(k)} - y_{i+1}^{(k-1)}| < \epsilon$

Step 4: Accept the current value of $y_{i+1}^{(k)}$ as y_{i+1} .

End

End

Milne's Predictor-Corrector Method

The well-known Milne's predictor-corrector method is obtained by using the corrector formula based on **Simpson's rule of integration** and the following formula as the predictor.

Predictor: $y_{i+1}^{(0)} = y_{i-3} + \frac{4h}{3}(2f_i - f_{i-1} + 2f_{i-2})$

Corrector: $y_{i+1}^{(1)} = y_{i-1} + \frac{h}{3}(f_{i+1}^{(0)} + 4f_i + f_{i-1})$

where $f_{i+1}^{(0)} = f(t_{i+1}, y_{i+1}^{(0)})$

and $f_k = f(t_k, y_k)$, $k = i + 1, i$, and $i - 1$.

- **Error Term:** The error term for the predictor: $\boxed{\frac{28}{90}h^5y^{(v)}(\xi)}$ and that for the corrector: $\boxed{-\frac{1}{90}h^5y^{(v)}(\eta)}$ where $t_i < \xi < t_{i-2}$, and $t_{i+1} < \eta < t_{i-1}$.

Both errors are of the same order, but the corrector has the lower constant term.