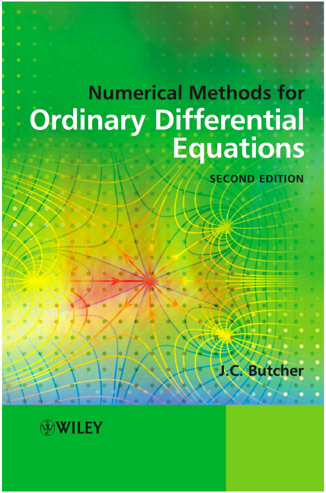
John Butcher's tutorials

Introduction to Runge–Kutta methods

$$\Phi(t) = \frac{1}{\gamma(t)}$$



Introduction

It will be convenient to consider only autonomous initial value problems

$$y'(x) = f(y(x)), \qquad y(x_0) = y_0,$$
$$f : \mathbb{R}^N \to \mathbb{R}^N.$$

The Euler method is the simplest way of obtaining numerical approximations at

$$x_1 = x_0 + h, \quad x_2 = x_1 + h, \ldots$$

using the formula

$$y_n = y_{n-1} + hf(y_{n-1}), \;\; h = x_n - x_{n-1}, \;\; n = 1, 2, \;\; \ldots.$$

Introduction

It will be convenient to consider only autonomous initial value problems

$$y'(x) = f(y(x)), \qquad y(x_0) = y_0,$$
$$f : \mathbb{R}^N \to \mathbb{R}^N.$$

The Euler method is the simplest way of obtaining numerical approximations at

$$x_1 = x_0 + h, \quad x_2 = x_1 + h, \ldots$$

using the formula

$$y_n = y_{n-1} + hf(y_{n-1}), \;\; h = x_n - x_{n-1}, \;\; n = 1, 2, \;\; \ldots .$$

Introduction

It will be convenient to consider only autonomous initial value problems

$$y'(x) = f(y(x)), \qquad y(x_0) = y_0,$$
$$f : \mathbb{R}^N \to \mathbb{R}^N.$$

The Euler method is the simplest way of obtaining numerical approximations at

$$x_1 = x_0 + h, \quad x_2 = x_1 + h, \ldots$$

using the formula

$$y_n = y_{n-1} + hf(y_{n-1}), \;\; h = x_n - x_{n-1}, \;\; n = 1, 2, \;\; \ldots.$$

This method can be made more accurate by using either the mid-point quadrature formula

$$y_n = y_{n-1} + hf\big(y_{n-1} + \tfrac{1}{2}hf(y_{n-1})\big).$$

or the trapezoidal rule quadrature formula:

$$y_n = y_{n-1} + \tfrac{1}{2}hf(y_{n-1}) + \tfrac{1}{2}hf\big(y_{n-1} + hf(y_{n-1})\big).$$

These methods from Runge's 1895 paper are "second order" because the error in a single step behaves like $O(h^3)$.

This is in contrast to the first order Euler method where the order behaviour is $O(h^2)$.

This method can be made more accurate by using either the mid-point quadrature formula

$$y_n = y_{n-1} + hf\big(y_{n-1} + \tfrac{1}{2}hf(y_{n-1})\big).$$

or the trapezoidal rule quadrature formula:

$$y_n = y_{n-1} + \tfrac{1}{2}hf(y_{n-1}) + \tfrac{1}{2}hf\big(y_{n-1} + hf(y_{n-1})\big).$$

These methods from Runge's 1895 paper are "second order" because the error in a single step behaves like $O(h^3)$.

This is in contrast to the first order Euler method where the order behaviour is $O(h^2)$.

This method can be made more accurate by using either the mid-point quadrature formula

$$y_n = y_{n-1} + hf\big(y_{n-1} + \tfrac{1}{2}hf(y_{n-1})\big).$$

or the trapezoidal rule quadrature formula:

$$y_n = y_{n-1} + \tfrac{1}{2}hf(y_{n-1}) + \tfrac{1}{2}hf\big(y_{n-1} + hf(y_{n-1})\big).$$

These methods from Runge's 1895 paper are "second order" because the error in a single step behaves like $O(h^3)$.

This is in contrast to the first order Euler method where the order behaviour is $O(h^2)$.

This method can be made more accurate by using either the mid-point quadrature formula

$$y_n = y_{n-1} + hf\big(y_{n-1} + \tfrac{1}{2}hf(y_{n-1})\big).$$

or the trapezoidal rule quadrature formula:

$$y_n = y_{n-1} + \tfrac{1}{2}hf(y_{n-1}) + \tfrac{1}{2}hf\big(y_{n-1} + hf(y_{n-1})\big).$$

These methods from Runge's 1895 paper are "second order" because the error in a single step behaves like $O(h^3)$.

This is in contrast to the first order Euler method where the order behaviour is $O(h^2)$.

A few years later, Heun gave a full explanation of order 3 methods.

Shortly afterwards Kutta gave a detailed analysis of order 4 methods.

In the early days of Runge–Kutta methods the aim seemed to be to find explicit methods of higher and higher order.

Later the aim shifted to finding methods that seemed to be optimal in terms of local truncation error and to finding built-in error estimators.

With the emergence of stiff problems as an important application area, attention moved to implicit methods.

A few years later, Heun gave a full explanation of order 3 methods.

Shortly afterwards Kutta gave a detailed analysis of order 4 methods.

In the early days of Runge–Kutta methods the aim seemed to be to find explicit methods of higher and higher order.

Later the aim shifted to finding methods that seemed to be optimal in terms of local truncation error and to finding built-in error estimators.

With the emergence of stiff problems as an important application area, attention moved to implicit methods.

A few years later, Heun gave a full explanation of order 3 methods.

Shortly afterwards Kutta gave a detailed analysis of order 4 methods.

In the early days of Runge–Kutta methods the aim seemed to be to find explicit methods of higher and higher order.

Later the aim shifted to finding methods that seemed to be optimal in terms of local truncation error and to finding built-in error estimators.

With the emergence of stiff problems as an important application area, attention moved to implicit methods.

A few years later, Heun gave a full explanation of order 3 methods.

Shortly afterwards Kutta gave a detailed analysis of order 4 methods.

In the early days of Runge–Kutta methods the aim seemed to be to find explicit methods of higher and higher order.

Later the aim shifted to finding methods that seemed to be optimal in terms of local truncation error and to finding built-in error estimators.

With the emergence of stiff problems as an important application area, attention moved to implicit methods.

A few years later, Heun gave a full explanation of order 3 methods.

Shortly afterwards Kutta gave a detailed analysis of order 4 methods.

In the early days of Runge–Kutta methods the aim seemed to be to find explicit methods of higher and higher order.

Later the aim shifted to finding methods that seemed to be optimal in terms of local truncation error and to finding built-in error estimators.

With the emergence of stiff problems as an important application area, attention moved to implicit methods.

Formulation of Runge–Kutta methods

In carrying out a step we evaluate s stage values

$$Y_1, \quad Y_2, \quad \dots, \quad Y_s$$

and s stage derivatives

$$F_1, \quad F_2, \quad \dots, \quad F_s,$$

using the formula $F_i = f(Y_i)$.

Each Y_i is defined as a linear combination of the F_j added on to y_0:

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} F_j, \quad i = 1, 2, \dots, s,$$

and the approximation at $x_1 = x_0 + h$ is found from

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i F_i.$$

Formulation of Runge–Kutta methods

In carrying out a step we evaluate s stage values

$$Y_1, \quad Y_2, \quad \dots, \quad Y_s$$

and s stage derivatives

$$F_1, \quad F_2, \quad \dots, \quad F_s,$$

using the formula $F_i = f(Y_i)$.

Each Y_i is defined as a linear combination of the F_j added on to y_0:

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} F_j, \quad i = 1, 2, \dots, s,$$

and the approximation at $x_1 = x_0 + h$ is found from

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i F_i.$$

We represent the method by a tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

or, if the method is explicit, by the simplified tableau

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

In each case, c_i $(i = 1, 2, \dots)$ is defined as $\sum_{j=1}^{s} a_{ij}$. The value of c_i indicates the point $X_i = x_0 + hc_i$ for which Y_i is a good approximation to $y(X_i)$.

We represent the method by a tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

or, if the method is explicit, by the simplified tableau

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

In each case, c_i $(i = 1, 2, \dots)$ is defined as $\sum_{j=1}^{s} a_{ij}$. The value of c_i indicates the point $X_i = x_0 + hc_i$ for which Y_i is a good approximation to $y(X_i)$.

We represent the method by a tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

or, if the method is explicit, by the simplified tableau

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

In each case, c_i $(i = 1, 2, \dots)$ is defined as $\sum_{j=1}^{s} a_{ij}$. The value of c_i indicates the point $X_i = x_0 + hc_i$ for which Y_i is a good approximation to $y(X_i)$.

Examples:

$$y_1 = y_0 + \mathbf{0}hf(y_0) + \mathbf{1}hf\big(y_0 + \tfrac{\mathbf{1}}{\mathbf{2}}hf(y_0)\big)$$

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{\mathbf{1}}{\mathbf{2}} & \\ \hline & \mathbf{0} & \mathbf{1} \end{array}$$

Examples:

$$y_1 = y_0 + \mathbf{0}hf(y_0) + \mathbf{1}hf\left(y_0 + \tfrac{\mathbf{1}}{\mathbf{2}}hf(y_0)\right)$$

Y_1 Y_2

0		
$\frac{1}{2}$	$\mathbf{\frac{1}{2}}$	
	$\mathbf{0}$	$\mathbf{1}$

Examples:

$$y_1 = y_0 + \mathbf{0}hf(y_0) + \mathbf{1}hf\Big(y_0 + \tfrac{\mathbf{1}}{\mathbf{2}}hf(y_0)\Big)$$

Y_1 Y_2

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{\mathbf{1}}{\mathbf{2}} & \\ \hline & \mathbf{0} & \mathbf{1} \end{array}$$

$$y_1 = y_0 + \tfrac{\mathbf{1}}{\mathbf{2}}hf(y_0) + \tfrac{\mathbf{1}}{\mathbf{2}}hf\Big(y_0 + \mathbf{1}hf(y_0)\Big)$$

$$\begin{array}{c|cc} 0 & & \\ 1 & \mathbf{1} & \\ \hline & \frac{\mathbf{1}}{\mathbf{2}} & \frac{\mathbf{1}}{\mathbf{2}} \end{array}$$

Examples:

$$y_1 = y_0 + \mathbf{0}hf\underbrace{(y_0)}_{Y_1} + \mathbf{1}hf\underbrace{\big(y_0 + \tfrac{\mathbf{1}}{\mathbf{2}}hf(y_0)\big)}_{Y_2}$$

0		
$\frac{1}{2}$	$\mathbf{\frac{1}{2}}$	
	$\mathbf{0}$	$\mathbf{1}$

$$y_1 = y_0 + \tfrac{\mathbf{1}}{\mathbf{2}}hf\underbrace{(y_0)}_{Y_1} + \tfrac{\mathbf{1}}{\mathbf{2}}hf\underbrace{\big(y_0 + \mathbf{1}hf(y_0)\big)}_{Y_2}$$

0		
1	$\mathbf{1}$	
	$\mathbf{\frac{1}{2}}$	$\mathbf{\frac{1}{2}}$

Taylor series of exact solution

We need formulae for the second, third, . . . , derivatives.

$$
\begin{aligned}
y'(x) &= f(y(x)) \\
y''(x) &= f'(y(x))y'(x) \\
&= f'(y(x))f(y(x)) \\
y'''(x) &= f''(y(x))(f(y(x)), y'(x)) + f'(y(x))f'(y(x))y'(x) \\
&= f''(y(x))(f(y(x)), f(y(x))) \\
&\qquad + f'(y(x))f'(y(x))f(y(x))
\end{aligned}
$$

This will become increasingly complicated as we evaluate higher derivatives.

Hence we look for a systematic pattern.

Write $\mathbf{f} = f(y(x))$, $\mathbf{f}' = f'(y(x))$, $\mathbf{f}'' = f''(y(x))$, ...

Taylor series of exact solution

We need formulae for the second, third, . . . , derivatives.

$$
\begin{aligned}
y'(x) &= f(y(x)) \\
y''(x) &= f'(y(x))y'(x) \\
&= f'(y(x))f(y(x)) \\
y'''(x) &= f''(y(x))(f(y(x)), y'(x)) + f'(y(x))f'(y(x))y'(x) \\
&= f''(y(x))(f(y(x)), f(y(x))) \\
&\quad + f'(y(x))f'(y(x))f(y(x))
\end{aligned}
$$

This will become increasingly complicated as we evaluate higher derivatives.
Hence we look for a systematic pattern.
Write $\mathbf{f} = f(y(x))$, $\mathbf{f}' = f'(y(x))$, $\mathbf{f}'' = f''(y(x))$, ...

Taylor series of exact solution

We need formulae for the second, third, …, derivatives.

$$\begin{aligned} y'(x) &= f(y(x)) \\ y''(x) &= f'(y(x))y'(x) \\ &= f'(y(x))f(y(x)) \\ y'''(x) &= f''(y(x))(f(y(x)), y'(x)) + f'(y(x))f'(y(x))y'(x) \\ &= f''(y(x))(f(y(x)), f(y(x))) \\ &\quad + f'(y(x))f'(y(x))f(y(x)) \end{aligned}$$

This will become increasingly complicated as we evaluate higher derivatives.
Hence we look for a systematic pattern.
Write $\mathbf{f} = f(y(x))$, $\mathbf{f}' = f'(y(x))$, $\mathbf{f}'' = f''(y(x))$, ….

Taylor series of exact solution

We need formulae for the second, third, . . . , derivatives.

$$\begin{aligned} y'(x) &= f(y(x)) \\ y''(x) &= f'(y(x))y'(x) \\ &= f'(y(x))f(y(x)) \\ y'''(x) &= f''(y(x))(f(y(x)), y'(x)) + f'(y(x))f'(y(x))y'(x) \\ &= f''(y(x))(f(y(x)), f(y(x))) \\ &\quad + f'(y(x))f'(y(x))f(y(x)) \end{aligned}$$

This will become increasingly complicated as we evaluate higher derivatives.

Hence we look for a systematic pattern.

Write $\mathbf{f} = f(y(x))$, $\mathbf{f}' = f'(y(x))$, $\mathbf{f}'' = f''(y(x))$, ...

Taylor series of exact solution

We need formulae for the second, third, ..., derivatives.

$$
\begin{aligned}
y'(x) &= f(y(x)) \\
y''(x) &= f'(y(x))y'(x) \\
&= f'(y(x))f(y(x)) \\
y'''(x) &= f''(y(x))(f(y(x)), y'(x)) + f'(y(x))f'(y(x))y'(x) \\
&= f''(y(x))(f(y(x)), f(y(x))) \\
&\qquad + f'(y(x))f'(y(x))f(y(x))
\end{aligned}
$$

This will become increasingly complicated as we evaluate higher derivatives.

Hence we look for a systematic pattern.

Write $\mathbf{f} = f(y(x))$, $\mathbf{f}' = f'(y(x))$, $\mathbf{f}'' = f''(y(x))$, ...

Taylor series of exact solution

We need formulae for the second, third, . . . , derivatives.

$$
\begin{aligned}
y'(x) &= f(y(x)) \\
y''(x) &= f'(y(x))y'(x) \\
&= f'(y(x))f(y(x)) \\
y'''(x) &= f''(y(x))(f(y(x)), y'(x)) + f'(y(x))f'(y(x))y'(x) \\
&= f''(y(x))(f(y(x)), f(y(x))) \\
&\quad + f'(y(x))f'(y(x))f(y(x))
\end{aligned}
$$

This will become increasingly complicated as we evaluate higher derivatives.

Hence we look for a systematic pattern.

Write $\mathbf{f} = f(y(x))$, $\mathbf{f}' = f'(y(x))$, $\mathbf{f}'' = f''(y(x))$, ...

Taylor series of exact solution

We need formulae for the second, third, …, derivatives.

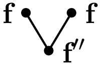
$$
\begin{aligned}
y'(x) &= f(y(x)) \\
y''(x) &= f'(y(x))y'(x) \\
&= f'(y(x))f(y(x)) \\
y'''(x) &= f''(y(x))(f(y(x)), y'(x)) + f'(y(x))f'(y(x))y'(x) \\
&= f''(y(x))(f(y(x)), f(y(x))) \\
&\qquad + f'(y(x))f'(y(x))f(y(x))
\end{aligned}
$$

This will become increasingly complicated as we evaluate higher derivatives.
Hence we look for a systematic pattern.
Write $\mathbf{f} = f(y(x))$, $\mathbf{f}' = f'(y(x))$, $\mathbf{f}'' = f''(y(x))$, ….

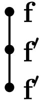
$$y'(x) = \mathbf{f}$$

$$y''(x) = \mathbf{f}'\mathbf{f}$$

$$y'''(x) = \mathbf{f}''(\mathbf{f},\mathbf{f})$$



$$+\,\mathbf{f}'\mathbf{f}'\mathbf{f}$$



The various terms have a structure related to rooted-trees.

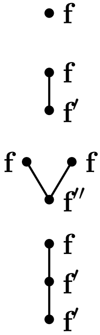
Hence, we introduce the set of all rooted trees and some functions on this set.

$$y'(x) = \mathbf{f}$$

$$y''(x) = \mathbf{f}'\mathbf{f}$$

$$y'''(x) = \mathbf{f}''(\mathbf{f}, \mathbf{f})$$

$$+ \mathbf{f}'\mathbf{f}'\mathbf{f}$$



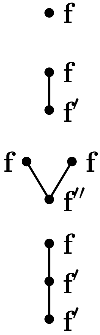
The various terms have a structure related to rooted-trees.

Hence, we introduce the set of all rooted trees and some functions on this set.

$$y'(x) = \mathbf{f}$$

$$y''(x) = \mathbf{f}'\mathbf{f}$$

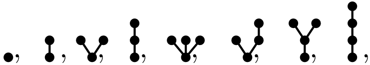
$$y'''(x) = \mathbf{f}''(\mathbf{f}, \mathbf{f}) + \mathbf{f}'\mathbf{f}'\mathbf{f}$$



The various terms have a structure related to rooted-trees.

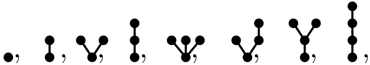
Hence, we introduce the set of all rooted trees and some functions on this set.

Let T denote the set of rooted trees:

$T = \{$  $\ldots \}$

We identify the following functions on T.

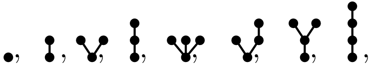
Let T denote the set of rooted trees:

$$T = \Big\{$$  $$\ldots \Big\}$$

We identify the following functions on T.

In this table, t will denote a typical tree

Let T denote the set of rooted trees:

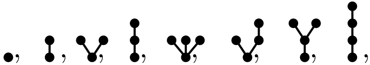
$T = \Big\{$  $\ldots \Big\}$

We identify the following functions on T.

In this table, t will denote a typical tree

$r(t)$	order of t = number of vertices

Let T denote the set of rooted trees:

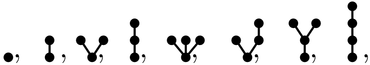
$T = \Big\{$  $\ldots \Big\}$

We identify the following functions on T.

In this table, t will denote a typical tree

$r(t)$	order of t = number of vertices
$\sigma(t)$	symmetry of t = order of automorphism group

Let T denote the set of rooted trees:

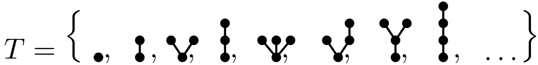
$T = \Big\{$  $\ldots \Big\}$

We identify the following functions on T.

In this table, t will denote a typical tree

$r(t)$	order of t = number of vertices
$\sigma(t)$	symmetry of t = order of automorphism group
$\gamma(t)$	density of t

Let T denote the set of rooted trees:

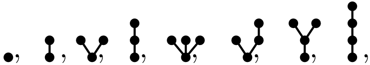


We identify the following functions on T.

In this table, t will denote a typical tree

$r(t)$	order of t = number of vertices
$\sigma(t)$	symmetry of t = order of automorphism group
$\gamma(t)$	density of t
$\alpha(t)$	number of ways of labelling with an ordered set

Let T denote the set of rooted trees:

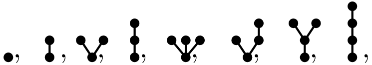
$T = \Big\{$  $\ldots \Big\}$

We identify the following functions on T.

In this table, t will denote a typical tree

$r(t)$	order of t = number of vertices
$\sigma(t)$	symmetry of t = order of automorphism group
$\gamma(t)$	density of t
$\alpha(t)$	number of ways of labelling with an ordered set
$\beta(t)$	number of ways of labelling with an unordered set

Let T denote the set of rooted trees:

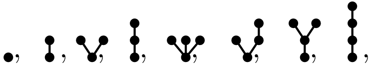
$T = \{$  $\ldots \}$

We identify the following functions on T.

In this table, t will denote a typical tree

$r(t)$	order of t = number of vertices
$\sigma(t)$	symmetry of t = order of automorphism group
$\gamma(t)$	density of t
$\alpha(t)$	number of ways of labelling with an ordered set
$\beta(t)$	number of ways of labelling with an unordered set
$F(t)(y_0)$	elementary differential

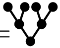
Let T denote the set of rooted trees:

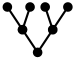
$T = \{$  $\ldots \}$

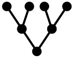
We identify the following functions on T.

In this table, t will denote a typical tree

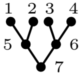
$r(t)$	order of t = number of vertices
$\sigma(t)$	symmetry of t = order of automorphism group
$\gamma(t)$	density of t
$\alpha(t)$	number of ways of labelling with an ordered set
$\beta(t)$	number of ways of labelling with an unordered set
$F(t)(y_0)$	elementary differential

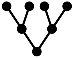
We will give examples of these functions based on $t =$ 

$t =$ 

$t =$ 

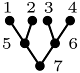
$r(t) = 7$

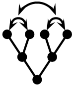


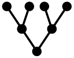
t = 

$r(t) = 7$

$\sigma(t) = 8$



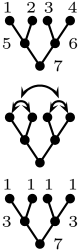


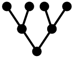
t = 

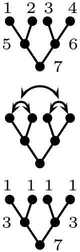
$r(t) = 7$

$\sigma(t) = 8$

$\gamma(t) = 63$



$t =$ 

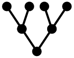


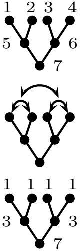
$$r(t) = 7$$

$$\sigma(t) = 8$$

$$\gamma(t) = 63$$

$$\alpha(t) = \frac{r(t)!}{\sigma(t)\gamma(t)} = 10$$

$t =$ 



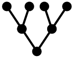
$$r(t) = 7$$

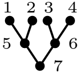
$$\sigma(t) = 8$$

$$\gamma(t) = 63$$

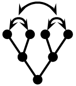
$$\alpha(t) = \frac{r(t)!}{\sigma(t)\gamma(t)} = 10$$

$$\beta(t) = \frac{r(t)!}{\sigma(t)} = 630$$

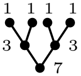
$t =$ 



$r(t) = 7$



$\sigma(t) = 8$

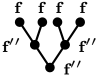


$\gamma(t) = 63$

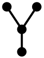
$$\alpha(t) = \frac{r(t)!}{\sigma(t)\gamma(t)} = 10$$

$$\beta(t) = \frac{r(t)!}{\sigma(t)} = 630$$

$$F(t) = \mathbf{f}''\big(\mathbf{f}''(\mathbf{f},\mathbf{f}),\mathbf{f}''(\mathbf{f},\mathbf{f})\big)$$



These functions are easy to compute up to order-4 trees:

t								
$r(t)$	1	2	3	3	4	4	4	4
$\sigma(t)$	1	1	2	1	6	1	2	1
$\gamma(t)$	1	2	3	6	4	8	12	24
$\alpha(t)$	1	1	1	1	1	3	1	1
$\beta(t)$	1	2	3	6	4	24	12	24
$F(t)$	$\mathbf{f}$	$\mathbf{f}'\mathbf{f}$	$\mathbf{f}''(\mathbf{f},\mathbf{f})$	$\mathbf{f}'\mathbf{f}'\mathbf{f}$	$\mathbf{f}^{(3)}(\mathbf{f},\mathbf{f},\mathbf{f})$	$\mathbf{f}''(\mathbf{f},\mathbf{f}'\mathbf{f})$	$\mathbf{f}'\mathbf{f}''(\mathbf{f},\mathbf{f})$	$\mathbf{f}'\mathbf{f}'\mathbf{f}'\mathbf{f}$

The formal Taylor expansion of the solution at $x_0 + h$ is

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{\alpha(t) h^{r(t)}}{r(t)!} F(t)(y_0)$$

Using the known formula for $\alpha(t)$, we can write this as

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t)(y_0)$$

Our aim will now be to find a corresponding formula for the result computed by one step of a Runge–Kutta method.

By comparing these formulae term by term, we will be able to obtain conditions for a specific order of accuracy.

The formal Taylor expansion of the solution at $x_0 + h$ is

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{\alpha(t) h^{r(t)}}{r(t)!} F(t)(y_0)$$

Using the known formula for $\alpha(t)$, we can write this as

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t)(y_0)$$

Our aim will now be to find a corresponding formula for the result computed by one step of a Runge–Kutta method.

By comparing these formulae term by term, we will be able to obtain conditions for a specific order of accuracy.

The formal Taylor expansion of the solution at $x_0 + h$ is

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{\alpha(t) h^{r(t)}}{r(t)!} F(t)(y_0)$$

Using the known formula for $\alpha(t)$, we can write this as

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t)(y_0)$$

Our aim will now be to find a corresponding formula for the result computed by one step of a Runge–Kutta method.

By comparing these formulae term by term, we will be able to obtain conditions for a specific order of accuracy.

The formal Taylor expansion of the solution at $x_0 + h$ is

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{\alpha(t) h^{r(t)}}{r(t)!} F(t)(y_0)$$

Using the known formula for $\alpha(t)$, we can write this as

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t)(y_0)$$

Our aim will now be to find a corresponding formula for the result computed by one step of a Runge–Kutta method.

By comparing these formulae term by term, we will be able to obtain conditions for a specific order of accuracy.

Taylor series of approximation

We need to evaluate various expressions which depend on the tableau for a particular method.

Taylor series of approximation

We need to evaluate various expressions which depend on the tableau for a particular method.

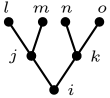
These are known as “elementary weights”.

Taylor series of approximation

We need to evaluate various expressions which depend on the tableau for a particular method.

These are known as "elementary weights".
We use the example tree we have already considered to illustrate the construction of the elementary weight $\Phi(t)$.

$t =$

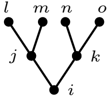


Taylor series of approximation

We need to evaluate various expressions which depend on the tableau for a particular method.

These are known as “elementary weights”.
We use the example tree we have already considered to illustrate the construction of the elementary weight $\Phi(t)$.

$t =$

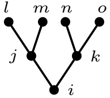


$$\Phi(t) = \sum_{i,j,k,l,m,n,o=1}^{s} b_i a_{ij} a_{ik} a_{jl} a_{jm} a_{kn} a_{ko}$$

Taylor series of approximation

We need to evaluate various expressions which depend on the tableau for a particular method.

These are known as “elementary weights”.
We use the example tree we have already considered to illustrate the construction of the elementary weight $\Phi(t)$.

$t =$ 

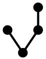
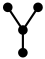
$$\Phi(t) = \sum_{i,j,k,l,m,n,o=1}^{s} b_i a_{ij} a_{ik} a_{jl} a_{jm} a_{kn} a_{ko}$$

Simplify by summing over l, m, n, o:

$$\Phi(t) = \sum_{i,j,k=1}^{s} b_i a_{ij} c_j^2 a_{ik} c_k^2$$

Now add $\Phi(t)$ to the table of functions:

t				
$r(t)$	1	2	3	3
$\alpha(t)$	1	1	1	1
$\beta(t)$	1	2	3	6
$\Phi(t)$	$\sum b_i$	$\sum b_i c_i$	$\sum b_i c_i^2$	$\sum b_i a_{ij} c_j$

t				
$r(t)$	4	4	4	4
$\alpha(t)$	1	3	1	1
$\beta(t)$	4	24	12	24
$\Phi(t)$	$\sum b_i c_i^3$	$\sum b_i c_i a_{ij} c_j$	$\sum b_i a_{ij} c_j^2$	$\sum b_i a_{ij} a_{jk} c_k$

The formal Taylor expansion of the numerical approximation to the solution at $x_0 + h$ is

$$y_1 = y_0 + \sum_{t \in T} \frac{\beta(t) h^{r(t)}}{r(t)!} \Phi(t) F(t)(y_0)$$

Using the known formula for $\beta(t)$, we can write this as

$$y_1 = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)} \Phi(t) F(t)(y_0)$$

The formal Taylor expansion of the numerical approximation to the solution at $x_0 + h$ is

$$y_1 = y_0 + \sum_{t \in T} \frac{\beta(t) h^{r(t)}}{r(t)!} \Phi(t) F(t)(y_0)$$

Using the known formula for $\beta(t)$, we can write this as

$$y_1 = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)} \Phi(t) F(t)(y_0)$$

Order conditions

To match the Taylor series

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t)(y_0)$$

$$y_1 = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)} \Phi(t) F(t)(y_0)$$

up to h^p terms we need to ensure that

$$\Phi(t) = \frac{1}{\gamma(t)},$$

for all trees such that

$$r(t) \leq p.$$

These are the "order conditions".

Order conditions

To match the Taylor series

$$y(x_0 + h) = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)} F(t)(y_0)$$

$$y_1 = y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)} \Phi(t) F(t)(y_0)$$

up to h^p terms we need to ensure that

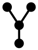
$$\Phi(t) = \frac{1}{\gamma(t)},$$

for all trees such that

$$r(t) \leq p.$$

These are the "order conditions".

The order conditions will be illustrated in the case of explicit 4 stage methods with order 4.

t	$\Phi(t) = \frac{1}{\gamma(t)}$
	$b_1 + b_2 + b_3 + b_4 = 1$
	$b_2c_2 + b_3c_3 + b_4c_4 = \frac{1}{2}$
	$b_2c_2^2 + b_3c_3^2 + b_4c_4^2 = \frac{1}{3}$
	$b_3a_{32}c_2 + b_4a_{42}c_2 + b_4a_{43}c_3 = \frac{1}{6}$
	$b_2c_2^3 + b_3c_3^3 + b_4c_4^3 = \frac{1}{4}$
	$b_3c_3a_{32}c_2 + b_4c_4a_{42}c_2 + b_4c_4a_{43}c_3 = \frac{1}{8}$
	$b_3a_{32}c_2^2 + b_4a_{42}c_2^2 + b_4a_{43}c_3^2 = \frac{1}{12}$
	$b_4a_{43}a_{32}c_2 = \frac{1}{24}$