



UNIVERSITI PUTRA MALAYSIA

**NUMERICAL SOLUTION OF ORDINARY AND
DELAY DIFFERENTIAL EQUATIONS
BY RUNGE-KUTTA TYPE
METHODS**

FUDZIAH BT ISMAIL

FSAS 1999 5

**NUMERICAL SOLUTION OF ORDINARY AND
DELAY DIFFERENTIAL EQUATIONS
BY RUNGE-KUTTA TYPE
METHODS**

By

FUDZIAH BT ISMAIL

**Dissertation Submitted in Fulfilment of the Requirements for the
Degree of Doctor of Philosophy in the Faculty of
Science and Environmental Studies
Universiti Putra Malaysia**

April 1999



TO MY FAMILY



AKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to the chairman of the supervisory committee, Professor. Dr. Mohamed bin Suleiman for his guidance, patience, invaluable advice and constant encouragement during the course of my research.

I am also grateful to the members of the supervisory committee, Associate Professor Dr. Bachok bin Taib and Associate Professor Dr. Isa bin Daud for their advice and motivation. I would like to thank Associate Professor Dr. Harun bin Budin too, for his continuous support in finalizing the thesis.

A special thanks also to all my friends and colleagues for their moral support and encouragement which has contributed in one way or another in completing this thesis.

I am also indebted to the Department of Civil Service and University Putra Malaysia for the sponsorship and study leave which enables me to pursue this research.

Finally my deepest appreciation goes to my husband, mother and children for their care, support, encouragement and most of all their love, without which this thesis would have never been completed.



TABLES OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	vii
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xii
ABSTRACT	xiii
ABSTRAK	xv
 CHAPTER	
I INTRODUCTION AND OBJECTIVES	1
Introduction.....	1
The Initial Value Problem	2
The Discrete Variable Methods	3
Runge-Kutta Method	6
Runge-Kutta Methods and its Implementation.....	9
Using Embedding Method for the Local Error Estimation.....	11
Stability of Runge-Kutta Method.....	11
Stiff Systems of Ordinary Differential Equations.....	15
Objectives of the Study	17
Planning of the Thesis.....	17
 II EMBEDDED SINGLY DIAGONALLY IMPLICIT RUNGE- KUTTA METHODS (4,5) IN (5,6), FOR THE INTEGRATION OF STIFF SYSEMS OF ORDINARY DIFFERENTIAL EQUATIONS	 20
Introduction	20
Derivation of Embedded Singly Diagonally Implicit Runge-Kutta Method	22
Stability of the Methods.....	32
Problems Tested.....	37
Numerical Results and Conclusions.....	39



III	PARTITIONING ORDINARY DIFFERENTIAL EQUATIONS USING RUNGE-KUTTA METHODS.....	50
	Introduction	50
	Stiffness Detection Criteria.....	53
	Implementation	55
	Iterations for Runge-Kutta Methods.....	61
	Stepsize Restrictions of Convergence for Simple Iterations....	64
	Stepsize Restrictions for Mixed Method Case.....	65
	Test for Partitioning.....	70
	Algorithm for Partitioning.....	75
	Problems Tested.....	77
	Numerical Results.....	80
	Conclusions.....	114
IV	INTRODUCTION TO DELAY DIFFERENTIAL EQUATIONS.....	117
	Delay Differential Equations.....	117
	Assumptions and Definitions.....	118
	Numerical Methods for Delay Differential Equations.....	120
	Stability Analysis of Numerical Methods for Solving Delay Differential Equations.....	123
	Previous Work on how to Handle Discontinuities.....	126
V	THE P-STABILITY AND Q-STABILITY REGIONS OF SINGLY DIAGONALLY IMPLICIT RUNGE-KUTTA METHODS FOR DELAY DIFFERENTIAL EQUATIONS.....	128
	Introduction	124
	P-Stability Analysis	129
	Q-Stability Analysis	135
	Problems Tested	138
	Numerical Results.....	140
	Conclusions	147



VI	PARTITIONING DELAY DIFFERENTIAL EQUATIONS USING RUNGE-KUTTA METHODS.....	148
	Introduction	148
	Algorithm for Intervalwise Partitioning.....	150
	Iterations on Runge-Kutta Method for Delay Differential Equations	160
	Algorithm for Partitioning	162
	Implementation for Componentwise Partitioning	166
	Problems Tested	177
	Numerical Results and Remarks	180
	Conclusions.....	193
VII	SUMMARY.....	195
	Conclusions	195
	Future Research	198
	REFERENCES	201
	VITA	206



LIST OF TABLES

Table		Page
2.1	The Coefficients of SDIRK Method (4,5) Embedded in (5,6).....	22
2.2	Order Conditions Related to the Trees.....	23
2.3	SDIRK Method (4,5) Embedded in (5,6) with $\gamma = 0.27805384$ and $c_3 = -0.7$	31
2.4	SDIRK Method (4,5) Embedded in (5,6) with $\gamma = 4/15$ and $c_3 = -0.7$	31
2.5	SDIRK Method (4,5) Embedded in (5,6) with $\gamma = 0.25$ and $c_3 = 0.3$	32
2.6	Numerical Results for Problem 2.1, Using Tolerances $10^{-2}, 10^{-4}, 10^{-6}$ and 10^{-8}	41
2.7	Numerical Results for Problem 2.2, Using Tolerances $10^{-2}, 10^{-4}, 10^{-6}$ and 10^{-8}	42
2.8	Numerical Results for Problem 2.3, Using Tolerances $10^{-2}, 10^{-4}, 10^{-6}$ and 10^{-8}	43
2.9	Numerical Results for Problem 2.4, Using Tolerances $10^{-2}, 10^{-4}, 10^{-6}$ and 10^{-8}	44
2.10	Numerical Results for Problem 2.5, Using Tolerances $10^{-2}, 10^{-4}, 10^{-6}$ and 10^{-8}	45
2.11	Numerical Results for Problem 2.6, Using Tolerances $10^{-2}, 10^{-4}, 10^{-6}$ and 10^{-8}	46
2.12	Numerical Results for Problem 2.7, Using Tolerances $10^{-2}, 10^{-4}, 10^{-6}$ and 10^{-8} :.....	47
2.13	Total Time Taken to Solve the Problems over all Tolerances	48
3.1	Numerical Results when Problem 3.1 is Solved Using Method F1(B).....	82
3.2	Numerical Results when Problem 3.2 is Solved Using Method F1(B).....	83
3.3	Numerical Results when Problem 3.3 is Solved Using Method F1(B).....	84
3.4	Numerical Results when Problem 3.4 is Solved Using Method F1(B).....	86



3.5	Numerical Results when Problem 3.5 is Solved Using Method F1(B).....	89
3.6	Numerical Results when Problem 3.6 is Solved Using Method F1(B).....	91
3.7	Numerical Results when Problem 3.7 is Solved Using Method F1(B).....	92
3.8	Numerical Results when Problem 3.8 is Solved Using Method F1(B).....	94
3.9	Numerical Results when Problem 3.1 is Solved Using Method F1(A).....	97
3.10	Numerical Results when Problem 3.2 is Solved Using Method F1(A).....	98
3.11	Numerical Results when Problem 3.3 is Solved Using Method F1(A).....	99
3.12	Numerical Results when Problem 3.4 is Solved Using Method F1(A).....	101
3.13	Numerical Results when Problem 3.5 is Solved Using Method F1(A).....	104
3.14	Numerical Results when Problem 3.6 is Solved Using Method F1(A).....	105
3.15	Numerical Results when Problem 3.7 is Solved Using Method F1(A).....	107
3.16	Numerical Results when Problem 3.8 is Solved Using Method F1(A).....	109
5.1	Numerical Results when Problem 5.1 is Solved Using Method F1(C).....	141
5.2	Numerical Results when Problem 5.2 is Solved Using Method F1(C).....	141
5.3	Numerical Results when Problem 5.3 is Solved Using Method F1(C).....	142
5.4	Numerical Results when Problem 5.4 is Solved Using Method F1(C).....	142
5.5	Numerical Results when Problem 5.5 is Solved Using Method F1(C).....	143
5.6	Numerical Results when Problem 5.6 is Solved Using Method F1(C).....	143
5.7	Numerical Results when Problem 5.1 is Solved Using Method F1(A).....	144
5.8	Numerical Results when Problem 5.2 is Solved Using Method F1(A).....	144
5.9	Numerical Results when Problem 5.3 is Solved Using Method F1(A).....	145



5.10	Numerical Results when Problem 5.4 is Solved Using Method F1(A).....	145
5.11	Numerical Results when Problem 5.5 is Solved Using Method F1(A).....	146
5.12	Numerical Results when Problem 5.6 is Solved Using Method F1(A).....	146
6.1	Numerical Results for Problem 6.1, Using technique B1.....	182
6.2	Numerical Results for Problem 6.2 ($p=-24$), Using Technique B1...	182
6.3	Numerical Results for Problem 6.2 ($p=-100$), Using Technique B1..	182
6.4	Numerical Results for Problem 6.3, Using Technique B1.....	183
6.5	Numerical Results for Problem 6.4, Using Technique B1.....	183
6.6	Numerical Results for Problem 6.5, Using Technique B1.....	183
6.7	Numerical Results for Problem 6.6, Using Technique B1.....	184
6.8	Numerical Results for Problem 6.7, Using Technique B1.....	184
6.9	Numerical Results for Problem 6.8, Using Technique B1.....	184
6.10	Numerical Results for Problem 6.4, Using Technique B2.....	187
6.11	Numerical Results for Problem 6.5, Using Technique B2.....	187
6.12	Numerical Results for problem 6.6, Using Technique B2.....	188
6.13	Numerical Results for Problem 6.7, Using Technique B2.....	189
6.14	Numerical Results for Problem 6.8, Using Technique B2.....	190



LIST OF FIGURES

Figure		Page
1	Stability Region for SDIRK Method F1(A).....	35
2	Stability Region for SDIRK Method F1(B).....	36
3	Stability Region for SDIRK Method F1(C)	36
4	Graphs of Stepsize Versus Tolerance for Problem 3.1, Method F1(B), Technique A1	111
5	Graphs of Stepsize Versus Tolerance for Problem 3.2, Method F1(B), Technique A1	111
6	Graphs of Stepsize Versus Tolerance for Problem 3.3, Method F1(B), Technique A1	112
7	Graphs of Stepsize Versus Tolerance for Problem 3.4, Method F1(B), Technique A1	112
8	Graphs of Stepsize Versus Tolerance for Problem 3.7, Method F1(B), Technique A1	113
9	Graphs of Stepsize Versus Tolerance for Problem 3.8, Method F1(B), Technique A1	113
10	The P-Stability Region of Method F1(A)	133
11	The P-Stability Region of Method F1(B)	134
12	The P-Stability Region of Method F1(C)	134
13	The Q-Stability Region of Method F1(A)	136
14	The Q-Stability Region of Method F1(B)	137
15	The Q-Stability Region of Method F1(C)	137
16	Graphs of h Versus Tolerance for Problem 6.2 when Solved Using Technique B1	185
17	Graphs of h Versus Tolerance for Problems 6.4 and 6.5 when Solved Using Technique B1	185
18	Graphs of h Versus Tolerance for Problems 6.1 and 6.6 when Solved Using Technique B1	186
19	Graphs of h Versus Tolerance for Problems 6.7 and 6.8 when Solved Using Technique B1	186



20	Graphs of h Versus Tolerance when Problem 6.4 is Solved	
	Using Technique B2	191
21	Graphs of h Versus Tolerance when Problem 6.5 is Solved	
	Using Technique B2	191
22	Graphs of h Versus Tolerance when Problem 6.	
	Using Technique B2	192
23	Graphs of h Versus Tolerance when Problem 6.7 is Solved	
	Using Technique B2.....	192



LIST OF ABBREVIATIONS

IVP	Initial Value Problem
IRK	Implicit Runge-Kutta
ODE	Ordinary Differential Equation
ODEs	Ordinary Differential Equations
SDIRK	Singly Diagonally Implicit Runge-Kutta
BDF	Backward Differentiation Formula
ROW	Rosenbrock Wanner Method
DDE	Delay Differential Equation
DDEs	Delay Differential Equations
RDE	Retarded Delay Differential Equation
NDE	Neutral Delay Differential Equation
CRK	Continuous Runge-Kutta
RKF	Runge-Kutta Fehlberg



Abstract of the dissertation submitted to the Senate of Universiti Putra Malaysia in fulfilment of the requirements for the degree of Doctor of Philosophy.

**NUMERICAL SOLUTION OF ORDINARY AND
DELAY DIFFERENTIAL EQUATIONS
BY RUNGE-KUTTA TYPE
METHODS**

By

FUDZIAH BT ISMAIL

April 1999

Chairman: Professor Mohamed bin Suleiman, Ph.D.

Faculty: Science and Environmental Studies

Runge-Kutta methods for the solution of systems of ordinary differential equations (ODEs) are described. To overcome the difficulty in implementing fully implicit Runge-Kutta method and to avoid the limitations of explicit Runge-Kutta method, we resort to Singly Diagonally Implicit Runge-Kutta (SDIRK) method, which is computationally efficient and stiffly stable. Consequently, embedded SDIRK methods of fourth order five stages in fifth order six stages are constructed. Their regions of stability are presented and numerical results of the methods are compared with the existing methods.

Stiff systems of ODEs are solved using implicit formulae and require the use of Newton-like iteration, which needs a lot of computational effort. If the systems



can be partitioned dynamically into stiff and nonstiff subsystems then a more effective code can be developed. Hence, partitioning strategies are discussed in detail and numerical results based on two techniques to detect stiffness using SDIRK methods are compared.

A brief introduction to delay differential equations (DDEs) is given. The stability properties of SDIRK methods, when applied to DDEs, using Lagrange interpolation to evaluate the delay term, are investigated.

Finally, partitioning strategies for ODEs are adapted to DDEs and numerical results based on two partitioning techniques, intervalwise partitioning and componentwise partitioning are tabulated and compared.



Abstrak disertasi yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah.

**PENYELESAIAN BERANGKA BAGI PERSAMAAN
PEMBEZAAN BIASA DAN LENGAH
MENGUNAKAN KAEDAH
RUNGE-KUTTA**

Oleh

FUDZIAH BT ISMAIL

April 1999

Pengerusi: Profesor Mohamed bin Suleiman, Ph.D.

Fakulti: Sains dan Pengajian Alam Sekitar

Kaedah Runge-Kutta untuk penyelesaian sistem persamaan pembezaan biasa (PPB) diterangkan. Untuk mengatasi kerumitan dalam melaksanakan kaedah Runge-Kutta sepenuh tersirat dan untuk menghindarkan kekurangan yang ada pada kaedah Runge-Kutta tak tersirat, kami menumpukan sepenuh perhatian terhadap kaedah Runge-Kutta pepenjuru tunggal tersirat yang lebih efisien serta stabil kaku. Justeru itu, kaedah terbenam Runge-Kutta pepenjuru tunggal tersirat (RKPTT) peringkat empat tahap lima dalam peringkat lima tahap enam diterbitkan. Rantau kestabilannya diberikan dan keputusan berangka bagi kaedah tersebut dibandingkan dengan keputusan berangka bagi kaedah yang sedia ada.

Sistem persamaan pembezaan biasa kaku biasanya diselesaikan dengan kaedah tersirat dan melibatkan lelaran Newton serta memerlukan pengiraan yang



banyak. Andainya sistem tersebut boleh dipetakan kepada sistem kaku dan tak kaku, maka suatu kod penyelesaian yang lebih efektif dapat dibentuk. Oleh itu strategi pemetakan dibincangkan secara mendalam dan keputusan berangka kaedah RKPTT berdasarkan dua teknik mengesan kekakuan dibandingkan.

Pengenalan ringkas kepada persamaan pembezaan lengah (PPL) diberikan dan sifat kestabilan kaedah RKPTT bila digunakan keatas PPL menggunakan interpolasi Lagrange untuk mengira sebutan lengahnya dikaji.

Akhir sekali strategi pemetakan bagi PPB diadaptasikan kepada PPL dan keputusan berangka berdasarkan dua teknik pemetakan, iaitu pemetakan berselang dan pemetakan berkomponen, dijadualkan dan dibandingkan.

CHAPTER I

INTRODUCTION AND OBJECTIVES

Introduction

Many important and significant problems in engineering, the physical sciences and the social sciences can be formulated in terms of differential equations. The problems of the bending of a thin beam clamped at both ends, the steady state flow of viscoelastic fluid parallel to an infinite plane surface with uniform suction and some problems in control theory can be formulated in terms of differential equations. Differential equations also appear naturally in the field of medicine. For example, the blood glucose regulatory system can be modeled into differential equations to obtain a fairly reliable criterion for the diagnosis of diabetes. A system of differential equations also govern the spread of epidemic in a population, this model enables researchers to prove the famous “ threshold theorem of epidemiology” which states that an epidemic will occur only if the number of people susceptible to the disease exceeds a certain threshold value.

Very often, mathematical problems cannot be conveniently solved using exact formulae, hence numerical methods are used as an alternative. This is the



technique widely used by scientists and engineers to solve their problems. Analytical methods usually give a result in the form of mathematical functions that can then be evaluated for specific instances, hence the behaviour and properties of the functions are often apparent. However results from numerical methods can be plotted to show some of the behaviour of the solution. Though results from numerical methods are approximations, they can be made as accurate as desired.

In the following sections, we give a brief review on differential equations and also numerical methods to solve them.

The Initial Value Problem

Consider the initial value problem (IVP) for a system of s first order ordinary differential equations (ODEs), defined by

$$y' = f(x, y), \quad y(a) = \eta, \quad [1.1]$$

where $y = [y_1(x), y_2(x), \dots, y_s(x)]^T$

$$f(x, y) = [f_1(x, y), \dots, f_s(x, y)]^T, \quad x \in [a, b]$$

and $\eta = [\eta_1, \dots, \eta_s]^T$ is a known vector.

Theorem (Existence and Uniqueness)

For (IVP) [1.1], let

(1) $f(x, y)$ be continuous in a domain D , where

$$D = \{(x, y) \mid a < x < b, \|y\| < \infty\}$$

(2) There exists $L > 0$ such that

$$\|f(x, y) - f(x, z)\| < L\|y - z\| \quad \text{for all } (x, y) \in D$$

and $(x, z) \in D$.

Then there exists a unique function $y(x)$, which satisfies the (IVP) [1.1].

For proof, see Henrici (1962). In this work, we shall assume the problem in [1.1] satisfies the conditions of the Existence and Uniqueness theorem.

The Discrete Variable Methods

The exact solution of the (IVP) [1.1] can be approximated using a numerical method, that is, the method will yield a sequence of approximations $y_n \cong y(x_n)$ on the set of points $x_{n+1} = x_n$

The simplest way of advancing the solution from the point x_n to x_{n+1} is by Taylor series.

$$y(x_{n+1}) = y(x_n) + h\Delta(x_n, y_n; h),$$

where $\Delta(x, y; h) = y'(x) + \frac{h}{2} y''(x) + \frac{h^2}{3!} y'''(x) + O(h^4)$.

If the series is truncated and $y(x_n)$ is replaced by y_n further approximation can be obtained

$$y_{n+1} = y_n + h\Phi(x_n, y_n; h), \quad n=0,1,2,\dots$$

where $\Phi(x, y, h) = f(x, y) + \frac{h}{2} f'(x, y) + \dots + \frac{h^{p-1}}{p!} f^{(p-1)}(x, y)$.

For $p=1$, we have the formula

$$y_{n+1} = y_n + hf(x_n, y_n)$$

which is the well known Euler method, and if $p=2$,

$$y_{n+1} = y_n + h[f(x_n, y_n) + \frac{h}{2}(f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n))].$$

Runge (1895), Heun (1900) and Kutta (1901) introduced an idea which is equivalent to constructing a formula for Φ which agrees with Δ as closely as possible, without involving derivatives of f . This process of “matching” the Taylor series can be illustrated by setting

$$\Phi(x, y, h) = b_1 f(x, y) + b_2 f(x + hc_2, y + a_{21}f(x, y)),$$

where b_1, b_2, c_2 and a_{21} are constants to be determined. Expanding both Φ and Δ in powers of h , we obtained

$$\Phi(x, y, h) = (b_1 + b_2)f(x, y) + hb_2[c_2 f_x(x, y) + a_{21}f_y(x, y)f(x, y)] + O(h^2) \quad [1.2]$$

$$\Delta(x, y, h) = f(x, y) + \frac{1}{2}h[f_x(x, y) + f_y(x, y)f(x, y)] + O(h^2),$$

yielding equation

$$b_1 + b_2 = 1, \quad b_2 c_2 = \frac{1}{2}, \quad b_2 a_{21} = \frac{1}{2}$$

The expression [1.2] and the above matching process can be extended to involve m equations of f yielding the general m -stage explicit Runge-Kutta method. Kutta (1901) completely characterized the set of Runge-Kutta methods of order 4 and proposed the first method of order 5.

When y_{n+1} is obtained by the Runge-Kutta method, the approximation y_n is then discarded. The method which makes direct use of some of the previously computed values of y_n is the multistep method. A linear multistep method of stepnumber k or k -step method is a computational method for determining the sequence $\{y_n\}$ which takes the form of a linear relationship between y_{n+j} and f_{n+j} , $j = 0, 1, \dots, k$. The general linear multistep method may be written as

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j}.$$

The method is explicit if $\beta_k = 0$ and implicit otherwise. Euler method is an explicit linear one-step method.

Taylor series for $y(x_{n-1})$

$$y(x_n) = y(x_{n-1}) + h y'(x_{n-1}) + \frac{h^2}{2!} y''(x_{n-1}) + \frac{h^3}{3!} y'''(x_{n-1}) + O(h^4)$$

Subtracting it from the Taylor series for $y(x_{n+1})$

$$y(x_{n+1}) - y(x_{n-1}) = 2h y'(x_n) + \frac{h^3}{3} y'''(x_n) + \dots$$

replacing $y(x_n)$ by y_n , yielding

$$y_{n+1} - y_{n-1} = 2h f_n + \frac{h^3}{3} y'''(x_n) + \dots$$

or in standard form, it can be written as

$$y_{n+1} = y_{n-1} + 2h f_n + \frac{h^3}{3} y'''(x_n) + \dots$$

This is the Mid-point rule, which local truncation error is $\pm \frac{1}{3} h^3 y'''(x_n)$ and it is an explicit 2-step method. In order to generate a sequence $\{y_n\}$ it is necessary to

obtain two starting values y_0 and y_1 . Other methods used are the block methods and the hybrid methods. Further information on all the methods can be obtained from Lambert (1973) and Hall and Watt (1976). The details of the literature review will be given at the beginning of the relevant chapters namely Chapter II, III, V and VI.

Runge-Kutta Method

A number of different approaches have been used in the analysis of Runge-Kutta methods, but the commonly used is the one developed by Butcher (1963), following on the work of Gill (1951) and Merson (1957), and it is as follows.

Let t denote the set of all rooted trees (t), where the trees are drawn with the root at the bottom. Refer to certain particular trees whose names are designated as t_k , t^k and $t_{k,l}$, where

$$t_k = \begin{array}{c} \overbrace{\quad\quad\quad}^{k-1} \\ \bullet \quad \bullet \quad \bullet \quad \bullet \\ \bullet \quad \bullet \quad \bullet \quad \bullet \\ \bullet \quad \bullet \quad \bullet \quad \bullet \end{array} \quad (\text{k vertices})$$

$$t^k = \left. \begin{array}{c} \bullet \quad \bullet \\ \bullet \quad \bullet \\ \vdots \\ \bullet \quad \bullet \\ \bullet \quad \bullet \end{array} \right\} \quad (\text{k vertices})$$

$$\text{and } t_{k,l} = \begin{array}{c} \overbrace{\quad\quad\quad}^{l-1} \\ \bullet \quad \bullet \quad \bullet \quad \bullet \\ \bullet \quad \bullet \quad \bullet \quad \bullet \\ \overbrace{\quad\quad\quad}^{k-1} \\ \bullet \quad \bullet \quad \bullet \quad \bullet \\ \bullet \quad \bullet \quad \bullet \quad \bullet \end{array}$$

For each $t \in T$ form a quantity $F(t)(y_{n-1})$ which is given by associating each vertex of t , the n th linear operator $f^{(n)}(y_{n-1})$ which is written as $f^{(k)}$ the number of upward growing branches from this vertex. Thus

$$F(t_k)(y_{n-1}) = f^{(k-1)}(\underbrace{f, \dots, f}_{k-1})$$

$$F(t^k)(y_{n-1}) = (f^{(1)})^k f.$$

One further example is for the tree



for which $F(t)(y_{n-1}) = f''(f''(f, f'(f)), f'''(f, f, f))$.

For each of the trees defined three numbers, they are

$r(t) \sim$ the number of vertices in t

$\sigma(t) \sim$ the order of the symmetry group of t

$\gamma(t) \sim$ the product of $r(u)$ over u , where for each vertex t , u is the sub tree formed from that vertex and all vertices can be reached from it following upward growing branch.

With this terminology, the formal Taylor series for $y(x_n)$ can be written as follows.

$$\begin{aligned} y(x_{n-1} + h) &= y(x_{n-1}) + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)\gamma(t)F(t)(y(x_{n-1}))} \\ &= y_{n-1} + hf + \frac{h}{2} f'(f) + h^3 \left(\frac{1}{6} f''(f, f) + \frac{1}{6} f'(f'(f)) \right) + O(h^4) \end{aligned}$$

With each tree t associate a polynomial $\Phi(t)$, by attaching labels i, j, k, \dots to each of the vertices of t , where i is the node. Form the product of b_i and of a_{jk} for each upward growing branch from j to k , and then sum over each label.

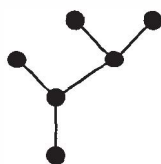
For example for the tree



$$\Phi(t) = \sum b_i a_{ij} a_{jl} a_{jk} a_{km} a_{kn} \quad \text{but} \quad c_j = \sum_k a_{jk}$$

$$\text{Therefore} \quad \Phi(t) = \sum b_i a_{ij} c_j a_{jk} c_k^2$$

To obtain the value of $\gamma(t)$ for a particular t , associate a value 1 to all the terminal vertices and a value $i + 1$ with all vertices for which i is the sum of the numbers attached to vertices which branch outwards from this one. The value of $\gamma(t)$ is simply the product of the integers associated with each vertex. For the above tree, we have



$$\gamma(t) = 90$$

The necessary and sufficient condition for the order of the method to be q is

that $\Phi(t) = \frac{1}{\gamma(t)}$ for all t with no more than q vertices. This approach will be

explained further in Chapter II.