UNIVERSITI PUTRA MALAYSIA

PARALLEL EXECUTION OF RUNGE-KUTTA METHODS FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS

ZAILAN SIRI

FS 2004 22

PARALLEL EXECUTION OF RUNGE-KUTTA METHODS FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS

By

ZAILAN SIRI

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Master of Science

May 2004

TABLE OF CONTENTS

Abstract of the thesis submitted to the Senate of Universiti Putra Malaysia in fulfilment of the requirements for the degree of Master of Science

PARALLEL EXECUTION OF RUNGE-KUTTA METHODS FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS

By

ZAILAN SIRI

May 2004

Chairman: Associate Professor Dr. Fudziah Ismail, Ph.D.

Faculty: Science

As we know Runge-Kutta method is a one step method hence it is quite limited in terms of implementation in parallel, here we going to exploit and extend the favourable characteristic of Runge-Kutta method so that they can be implemented in parallel.

In this thesis we are focusing in two types of Runge-Kutta methods. The first one is the Diagonally Implicit Runge-Kutta (DIRK) method. The method used here is actually have been tailored made for the purpose of parallel machine where the subsequent functions evaluations do not depend on the previous function evaluations.

The second family of Runge-Kutta method is the block Runge-Kutta both explicit and implicit. In this study, we exploit these methods so that we can implement in parallel mode.

The C programming of the methods employed are run on a shared memory Sequent SE30 parallel computer. All the numerical results are given to illustrate the algorithms developed for the cases that we were tested. The numerical results show that the parallel algorithms of diagonally implicit Runge-Kutta (DIRK), block explicit Runge-Kutta (BERK) and block diagonally implicit Runge-Kutta (BDIRK) methods is better than sequential modes because the parallel execution time is smaller than sequential execution time.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Master Sains

PENYELESAIAN BERANGKA BAGI PERSAMAAN PEMBEZAAN BIASA MENGGUNAKAN KAEDAH RUNGE-KUTTA SECARA SELARI

Oleh

ZAILAN SIRI

Mei 2004

Pengerusi: Professor Madya Dr. Fudziah Ismail, Ph.D.

Fakulti: Sains

Seperti mana yang diketahui, kaedah Runge-Kutta merupakan kaedah satu langkah, maka ianya agak terhad untuk diimplimentasikan secara selari. Di sini , apa yang kita lakukan adalah mengeksploitasi serta memperluaskan sifat-sifat istimewa Runge-Kutta ini supaya ianya boleh diimplimentasikan secara selari.

Di dalam tesis ini, kita menumpukan kepada dua jenis kaedah Runge-Kutta. Pertama ialah kaedah Runge-Kutta Pepenjuru Tersirat (RKPT). Kaedah yang telah digunakan di sini sebenarnya telah diterbitkan sedemikian rupa untuk tujuan mesin selari di mana pergantungan penilaian fungsi daripada fungsi-fungsi sebelumnya diminimumkan.

Famili Runge-Kutta yang kedua adalah kaedah blok Runge-Kutta, iaitu kaedah Blok Runge-Kutta Tak Tersirat (BRKTT) dan kaedah Blok Runge-Kutta Pepenjuru Tersirat

(BRKPT). Di dalam kajian ini, kita menggunakan kaedah blok sedia ada dan mengeksploitasikan kedua-dua kaedah ini untuk membolehkannya diimplimentasikan secara selari.

Pengaturcaraan C untuk semua kaedah tersebut telah dilaksanakan dengan menggunakan komputer selari Sequent SE30 berkongsi ingatan yang terdapat di Universiti Putra Malaysia (UPM). Kesemua keputusan berangka diberikan untuk mengillustrasikan algoritma yang dibina untuk kes-kes yang telah diujikan. Keputusan berangka yang diperolehi menunjukkan bahawa algoritma selari adalah lebih baik daripada mod jujukan kerana masa pelaksanaan selari lebih pantas daripada masa pelaksanaan jujukan.

ACKNOWLEDGEMENTS

Most of all I would like to thank my supervisor, Dr. Fudziah Ismail from the Department of Mathematics, Faculty of Science and Environmental Studies, University Putra Malaysia. She is not only a mathematician with vision but most importantly a kind person. Her trust has inspired me to make the right research decision.

My sincere thanks to the Committee member Associate Professor Dr. Mohamed Othman, Head of Communication Technology and Network Department, Faculty of Computer Science and Information Technology, University Putra Malaysia, for giving more than just a laboratory space to work in but the understanding of what is “a good programmer”.

My best regards to Dato’ Dr. Mohamed Suleiman from the National Accreditation Board, who is also the Committee member for his patience and guidance.

I also want to thank all the people who have given me the encouragement and motivation throughout the study, especially Associate Professor Dr. Nor Aishah Hamzah and all the staffs of the Institute of Mathematical Sciences, University of Malaya.

Thank you also to the Skim Latihan Akademik Bumiputra unit of University of Malaya for the financial support.

A special thanks to my wife, Rose Irnawaty Ibrahim for being kind and patient while editing this text and the valuable suggestion which helped me in writing the thesis.

Finally, I wish to express my love and gratitude to all my family and friends, particularly my parents Encik Siri Dawi and Puan Safiyah Komeng for the continuous support and love.

LIST OF ABBREVIATIONS

2P1BERK	:	Two-Point One Block Explicit Runge-Kutta
3P1BERK	:	Three-Point One Block Explicit Runge-Kutta
4OBERK1	:	Fourth-Order Block Explicit Runge-Kutta 1
4OBERK2	:	Fourth-Order Block Explicit Runge-Kutta 2
BERK	:	Block Explicit Runge-Kutta
BDIRK	:	Block Diagonally Implicit Runge-Kutta
CPU	:	Central Processing Unit
DIRK	:	Diagonally Implicit Runge-Kutta
HPF	:	High Performance FORTRAN
IVP	:	Initial Value Problem
LTE	:	Local Truncation Error
MIMD	:	Multiple Instruction Stream, Multiple Data Stream
MISD	:	Multiple Instruction Stream, Single Data Stream
MP	:	Message Passing
ODE	:	Ordinary Differential Equation
OpenMP	:	Open Message Passing
PDIRK	:	Parallel Diagonally Implicit Runge-Kutta
SIMD	:	Single Instruction Stream, Multiple Data Stream
SISD	:	Single Instruction Stream, Single Data Stream
SMP	:	Symmetry Multiprocessing
UPM	:	University Putra Malaysia

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

In science and engineering, mathematical models are developed to help in the understanding of physical phenomena. These models often yield an equation that contains some derivatives of an unknown function. Such an equation is called differential equation. Two examples of models developed in calculus are the free fall of a body and the decay of a radioactive substance. Even though the above examples are easily solved by methods learned in calculus, they do give us some insight into the study of differential equations in general. Differential equations arise in a variety of areas, not only the physical sciences but also in such diverse fields as economics, medicine, psychology and operations research, more recently they have also arisen in models such as medicine, biology, and anthropology. In this thesis we will restrict our scope to ordinary differential equations (ODEs) and focus on the initial value problems (IVPs) and present the Runge-Kutta methods for solving such problems numerically.

Existence and Uniqueness

The IVP for a system of q first order ODEs is defined by:

$$\boldsymbol{y}' = \boldsymbol{f}(x,\boldsymbol{y}), \quad x \in [a, b], \quad \boldsymbol{y}(a) = \boldsymbol{y_0} \tag{1.1}$$

where

$\boldsymbol{y}(x) = [y_1(x), y_2(x), \ldots, y_q(x)]^T$

$\boldsymbol{f}(x, y) = [f_1(x, y), f_2(x, y), \ldots, f_q(x, y)]^T$

and $\boldsymbol{y_0}$ is a given vector of initial conditions. If the analytical process of finding a solution $y(x)$ is not feasible, it is still useful to know whether a solution exists and is unique. Existence serves to justify the use of numerical method and uniqueness is necessary so that once a solution is found we can be sure that it is the solution to the equation.

Definition 1.1

A function $f(x, y)$ satisfies a *Lipschitz* condition with respect to y if there exists a constant $L > 0$ such that $|f(x,y) - f(x,z)| \le L|y - z|$ for all $x \in [a,b]$. The *Lipschitz* constant L is independent of x.

Theorem 1.1

The first order IVP

$$y' = f(x, y), \qquad x \in [x_0, b]; \qquad y(x_0) = y_0, \qquad x_0 \in [a, b]$$

has a unique solution $y(x)$ for $x_0 \le x \le b$ if

(a) $f(x, y)$ is continuous in x

(b) $f(x, y)$ satisfies a *Lipschitz* condition with respect to y.

If both conditions are satisfied, there exists a unique solution to IVP (1.1). The proof can be seen in Burden (1997).

Numerical Solution of Initial Value Problems

In general there are two classes of methods for approximating the solution of IVP (1.1) namely one step method and multi-step method. One step method requires only the value of y_n in order to compute the value of y_{n+1}, on the other hand, the p multi-step method uses several past values $\{y_n, y_{n+1}, \ldots, y_{n-p+1}\}$. The best known one step methods are the Runge-Kutta type of methods.

A q-stage Runge-Kutta method (q function evaluations per step) can be written as:

$$y_{n+1} = y_n + h\sum_{i=1}^{q} b_i k_i \tag{1.2}$$

where

$$k_i = f\left(x_n + c_i h, y_n + \sum_{j=1}^{i} a_{ij} k_j\right),\ i = 1, 2, \ldots, q \tag{1.3}$$

For convenience, the coefficients a_{ij}, b_i and c_i of the Runge-Kutta methods can be written in the form of a Butcher's array as follows:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1q} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_q & a_{q1} & a_{q2} & \cdots & a_{qq} \\ \hline & b_1 & b_2 & \cdots & b_q \end{array} \tag{1.4}$$

In Butcher's array (1.4), an explicit Runge-Kutta method will have all the upper diagonal elements of a_{ij} is zero, that is, $a_{ij} = 0$ for all $j \geq i$. Hence, the explicit Runge-Kutta methods can be represented by the special Butcher's array as shown below:

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_q & a_{q1} & a_{q2} & \cdots & a_{qq-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{q-1} & b_q
\end{array} \tag{1.5}$$

The explicit Runge-Kutta method is easy to implement because the current function evaluation only depends on the previous function evaluation. That is the evaluation of k_i depends only on the values of k_j, ($j = 1, 2, \ldots, i$-1).

A diagonally implicit Runge-Kutta method is the method where by $a_{ij} = 0$ for all $j > i$ and the coefficients can be represented as:

$$\begin{array}{c|ccccc}
c_1 & a_{11} & & & & \\
c_2 & a_{21} & a_{22} & & & \\
c_3 & a_{31} & a_{32} & a_{33} & & \\
\vdots & \vdots & \vdots & & \ddots & \\
c_q & a_{q1} & a_{q2} & \cdots & & a_{qq} \\
\hline
 & b_1 & b_2 & \cdots & & b_q
\end{array}$$

The diagonally implicit Runge-Kutta method is harder to implement compared to the explicit Runge-Kutta method, since $k_i = f(x_i + c_i h,\ y_n + \sum_{j=1}^{i} a_{ij} k_j)$ meaning that, to evaluate k_i the values of k_1, k_2, …, k_{i-1} and k_i are needed. Hence, in this case simple iterations are used for the implementation of the diagonally implicit Runge-Kutta method. However, the method generally gives a more accurate result compared to the explicit Runge-Kutta method. Therefore, in this thesis diagonally implicit Runge-Kutta methods are used to solve the IVP.

Another type of Runge-Kutta method is the block Runge-Kutta methods, which approximate the solution of the IVP (1.1) at more than one point at a time. For example, the two points block Runge-Kutta method approximate the value of y_{n+1} and y_{n+2} at a time step. Hence the block Runge-Kutta method will take a shorter time to solve IVP and the nature of the method also makes it suitable for parallel implementations.

Objective of the Study

The objective of this research is to solve IVP using three types of Runge-Kutta method namely diagonally implicit Runge-Kutta (DIRK) methods, block explicit Runge-Kutta (BERK) methods and block diagonally implicit Runge-Kutta (BDIRK) methods in sequential and in parallel. The numerical results both for the sequential and parallel modes for the three types of Runge-Kutta method are tabulated and compared to determine their performance.

Framework of the Study

This thesis consists of six chapters. Chapter I presents a brief introduction of ODE and IVP, followed by numerical methods to approximate the solutions of the IVP. A brief introduction to Runge-Kutta method is also given and their suitability to be implemented in parallel is discussed.

In Chapter II, we discussed the relevant literature. The first part is literature on parallelism of Runge-Kutta methods and the second is literature on parallel computing. Literature on parallelism of Runge-Kutta methods can be divided into two subdivisions that is parallel integration of IVPs and parallel methods for solution of ODEs. The second part of the chapter covers the classification of computer, programming languages, program design and performance considerations.

In Chapter III, the concept of directed graph is introduced and DIRK methods derived by Iserles and Nørsett (1990) are used to solve ODEs sequentially and in parallel. Numerical results based on these two modes are tabulated and compared.