UNIVERSITI PUTRA MALAYSIA

DIRECT BLOCK METHODS FOR SOLVING SPECIAL SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS AND THEIR PARALLEL IMPLEMENTATIONS

YAP LEE KEN

FS 2008 18

DIRECT BLOCK METHODS FOR SOLVING SPECIAL SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS AND THEIR PARALLEL IMPLEMENTATIONS

By

YAP LEE KEN

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

March 2008

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Master of Science

DIRECT BLOCK METHODS FOR SOLVING SPECIAL SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS AND THEIR PARALLEL IMPLEMENTATIONS

By

YAP LEE KEN

March 2008

Chair : Associate Professor Dr Fudziah Binti Ismail, PhD

Faculty : Science

This thesis focuses mainly on deriving block methods of constant step size for solving special second order ODEs. The first part of the thesis is about the construction and derivation of block methods using linear difference operator. The regions of stability for both explicit and implicit block methods are presented. The numerical results of the methods are compared with existing methods. The results suggest a significant improvement in efficiency of the new methods.

The second part of the thesis describes the derivation of the *r*-point block methods based on Newton-Gregory backward interpolation formula. The numerical results of explicit and implicit *r*-point block methods are presented to illustrate the effectiveness of the methods in terms of total number of steps taken, accuracy and execution time. Both the explicit and implicit methods are more efficient compare to the existing method.

The *r*-point block methods that calculate the solution at *r*-point simultaneously are suitable for parallel implementation. The parallel codes of the block methods for the solution of large systems of ODEs are developed. Hence the last part of the thesis discusses the parallel execution of the codes.

The parallel algorithms are written in C language and implemented on Sun Fire V1280 distributed memory system. The fine-grained strategy is used to divide a computation into smaller parts and assign them to different processors. The performances of the *r*-point block methods using sequential and parallel codes are compared in terms of the total steps, execution time, speedup and efficiency. The parallel implementation of the new codes produced better speedup as the number of equations increase. The parallel codes gain better speedup and efficiency compared to sequential codes.

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

KAEDAH BLOK LANGSUNG BAGI MENYELESAIKAN PERSAMAAN PEMBEZAAN KHAS PERINGKAT KEDUA DAN IMPLEMENTASINYA SECARA SELARI

Oleh

YAP LEE KEN

March 2008

Pengerusi : Associate Professor Dr Fudziah Binti Ismail, PhD

Fakulti : Sains

Tumpuan utama tesis ini adalah untuk menerbitkan kaedah blok dengan saiz langkah malar untuk menyelesaikan persamaan pembezaan khas secara langsung. Bahagian pertama tesis ini adalah berkaitan dengan pembentukan dan terbitan kaedah blok dengan menggunakan pengoperasi beza linear. Rantau kestabilan untuk kedua-dua kaedah tersirat dan kaedah tak tersirat turut dipersembahkan. Keputusan berangka kaedah tersebut dibandingkan dengan kaedah yang sedia ada. Keputusan berangka menunjukkan penambahbaikan yang ketara dalam kecekapan kaedah baharu tersebut.

Bahagian kedua tesis ini menghuraikan terbitan kaedah blok *r*-titik berdasarkan formula sisipan belakang Newton-Gregory. Keputusan kaedah *r*-titik tersirat dan kaedah *r*-titk tak tersirat telah ditunjukkan untuk mengilustrasi keberkesanan

kaedah dari segi jumlah langkah yang diambil, kejituan dan masa pelaksanaan. Kedua-dua kaedah tersirat dan kaedah tak tersirat adalah lebih cekap berbanding dengan kaedah yang sedia ada.

Kaedah blok *r*-titik yang mengira penyelesaian pada *r*-titik serentak adalah sesuai untuk implementasi selari. Kaedah blok dengan kod selari untuk penyelesaian sistem persamaan pembezaan telah dibangunkan. Seterusnya bahagian akhir tesis ini membincangkan kod implementasi selari tersebut.

Algoritma selari ditulis dalam bahasa C dan dilaksana di sistem memori bertaburan Sun Fire V1280. Strategi *fine-grained* digunakan untuk membahagi perhitungan ke bahagian-bahagian kecil dan menugaskan bahagian-bahagian kecil ini ke pemproses yang berlainan**.** Implementasi kaedah blok *r*-titik yang menggunakan kod jujukan dan kod selari dibandingkan dari segi jumlah langkah, masa pelaksanaan, kecepatan dan keberkesanan. Kod selari kaedah baru menghasilkan kecepatan yang lebih baik apabila bilangan persamaan bertambah. Kod selari mencapai kecepatan dan kecekapan yang lebih baik berbanding dengan kod jujukan.

ACKNOWLEDGEMENTS

First and foremost, I would like to show my deepest appreciation and gratitude to the Chairman of the Supervisory Committee, Associate Professor Dr Fudziah Binti Ismail for her invaluable assistance, advice and guidance throughout the duration of the studies. I also wish to express my sincere thank to Associate Professor Dr Mohamed Bin Othman and Yang Berbahagia Professor Dato' Dr Mohamed Bin Suleiman for their guidance towards the successful completion of the thesis.

Special thanks due to Universiti Putra Malaysia for providing the financial support in the form of Graduate Research Assistantship throughout the duration of my studies. The guidance and advice of Dr Zanariah Binti Majid are gratefully acknowledged.

Finally my deepest appreciation goes to my beloved family especially my parents for their unconditional love, support and understanding throughout the course of my studies. I also would like to thank my friends for their understanding support and encouragement throughout the course of my research.

I certify that an Examination Committee has met on 24 March 2008 to conduct the final examination of Yap Lee Ken on her degree thesis entitled "DIRECT BLOCK METHODS FOR SOLVING SPECIAL SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS AND THEIR PARALLEL IMPLEMENTATIONS" in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The Committee recommends that the student be awarded the Master of Science.

Members of the Examination Committee were as follows:

Malik Hj. Abu Hassan, PhD
Professor
Faculty of Science
University Putra Malaysia
(Chairman)

Zainiddin K. Eshkuvatov, PhD
Lecturer
Faculty of Science
Universiti Putra Malaysia
(Internal Examiner)

Norihan Binti Md. Ariffin, PhD
Senior Lecturer
Faculty of Science
Universiti Putra Malaysia
(Internal Examiner)

Norhashidah Hj. Mohd. Ali, PhD
Associate Professor
School of Mathematical Sciences
Universiti Science Malaysia
(External Examiner)

HASANAH MOHD. GHAZALI, PhD
Professor and Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. The members of the Supervisory Committee were as follows:

Fudziah Ismail, PhD
Associate Professor
Faculty of Science
Universiti Putra Malaysia
(Chairman)

Mohamed Suleiman, PhD
Professor
Faculty of Science
Universiti Putra Malaysia
(Member)

Mohamed Othman, PhD
Associate Professor
Faculty of Science Computer and Information Technology
Universiti Putra Malaysia
(Member)

AINI IDERIS, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

DECLARATION

I declare that the thesis is my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously, and is not concurrently, submitted for any other degree at Universiti Putra Malaysia or at any other institution.

YAP LEE KEN

Date: 15 May 2008

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ABSTRAK	iv
ACKNOWLEDGEMENTS	vi
APPROVAL	vii
DECLARATION	ix
LIST OF TABLES	xiii
LIST OF FIGURES	xix
LIST OF ABBREVIATIONS	xxiv

CHAPTER

1	**INTRODUCTION TO NUMERICAL ORDINARY DIFFERENTIAL EQUATIONS (ODEs) AND PARALLEL COMPUTING**	1
	1.1 Introduction to numerical ODEs	2
	1.2 Linear Multistep Method	3
	1.3 Divided Differences	9
	1.4 Newton-Gregory Backward Interpolation Formula	10
	1.5 Introduction to Parallel Computing	13
	1.6 Parallel Architecture	13
	1.7 Sun Fire V1280 Architecture	16
	1.8 Parallel Algorithms in IVP Solvers	18
	1.9 Performance Metrics of Parallel Algorithms	19
	1.9.1 Execution Time	20
	1.9.2 Speedup	20
	1.9.3 Efficiency	21
	1.10 Problem Statement	21
	1.11 Objectives of the Studies	22
2	**LITERATURE REVIEW**	23
	2.1 Background to Numerical Multistep Methods	23
	2.2 Survey on Block Methods	24
	2.3 Review on Implementation of Predictor-Corrector Methods	25
	2.4 Review on Parallel Implementation	26
	2.5 Literature on Performance Analysis	28
3	**DERIVATION OF MULTISTEP BLOCK METHODS USING LINEAR DIFFERENCE OPERATOR**	29
	3.1 Introduction	29

	3.2	Derivation of Explicit 3-Point 1-Block Method	32
		3.2.1 Stability of Explicit 3-Point 1-Block Method	34
		3.2.2 Test Problems	36
		3.2.3 Numerical Results	38
		3.2.4 Discussion	43
	3.3	Derivation of Implicit 3-Point 1-Block Method	45
		3.3.1 Stability of Implicit 3-Point 1-Block Method	49
		3.3.2 Numerical Results	50
		3.3.3 Discussion	54
4	**EXPLICIT R-POINT BLOCK METHODS IN BACKWARD DIFFERENCE FORM FOR SOLVING SPECIAL SECOND ORDER ODEs DIRECTLY**		56
	4.1	Introduction	56
	4.2	Derivation of First Point of Explicit Block Method	57
	4.3	Derivation of Second Point of the Explicit Block Method	62
	4.4	Derivation of Third Point of the Explicit Block Method	65
	4.5	Derivation of Explicit R-Point Block Method	68
	4.6	Stability	69
	4.7	Test Problems	71
	4.8	Numerical Results	71
	4.9	Discussion I	88
		4.9.1 Total number of steps taken	88
		4.9.2 Accuracy	88
		4.9.3 Execution Time	90
	4.10	Numerical Results II	91
	4.11	Discussion II	98
5	**IMPLICIT R-POINT BLOCK METHODS IN BACKWARD DIFFERENCE FORM FOR SOLVING SPECIAL SECOND ORDER ODEs DIRECTLY**		99
	5.1	Introduction	99
	5.2	Derivation of First Point of Implicit Block Method	99
	5.3	Derivation of Second Point of Implicit Block Method	103
	5.4	Derivation of Third Point of Implicit Block Method	107
	5.5	Derivation of Implicit R-Point Block Method	111
	5.6	Test Problems	113
	5.7	Numerical Results I	113
	5.8	Discussion I	129
		5.8.1 Total Number of Steps Taken	129
		5.8.2 Accuracy	129
		5.8.3 Execution Time	131
	5.9	Numerical Results II	132
	5.10	Discussion II	139

6	**PARALLEL EXPLICIT AND IMPLICIT BLOCK ALGORITHMS**	140
	6.1 Introduction	140
	6.2 Problem Description and Objectives	141
	6.3 Parallel Algorithms for Explicit Block Methods	142
	6.3.1 Parallel Implementation of Explicit 2-Point Block Method	142
	6.3.2 Parallel Implementation of Explicit 3-Point Block Method	146
	6.4 Parallel Algorithms for Implicit Block Methods	148
	6.4.1 Parallel Implementation of Implicit 2-Point Block Method	148
	6.4.2 Parallel Implementation of Implicit 3-Point Block Method	153
	6.5 Test Problems	155
	6.6 Numerical Results	156
	6.7 Discussion	180
	6.7.1 Total Number of Steps Taken	180
	6.7.2 Execution Time	180
	6.7.3 Speedup	181
	6.7.4 Efficiency	183
	6.8 Summary	184
7	**CONCLUSION AND FUTURE WORK**	185
	7.1 Conclusion	185
	7.2 Future Work	186

BIBLIOGRAFY	188
BIODATA OF STUDENT	192
LIST OF PUBLICATIONS	193

LIST OF TABLES

Table		Page
3.1	Performance comparison between E2P1B and E3P1B for solving Problem 3.1 of special second order ODEs	40
3.2	Performance comparison between E2P1B and E3P1B for solving Problem 3.2 of special second order ODEs	40
3.3	Performance comparison between E2P1B and E3P1B for solving Problem 3.3 of special second order ODEs	41
3.4	Performance comparison between E2P1B and E3P1B for solving Problem 3.4 of special second order ODEs	41
3.5	Performance comparison between E2P1B and E3P1B for solving Problem 3.5 of special second order ODEs	42
3.6	Performance comparison between E2P1B and E3P1B for solving Problem 3.6 of special second order ODEs	42
3.7	Performance comparison between I2P1B and I3P1B for solving Problem 3.1 of special second order ODEs	51
3.8	Performance comparison between I2P1B and I3P1B for solving Problem 3.2 of special second order ODEs	51
3.9	Performance comparison between I2P1B and I3P1B for solving Problem 3.3 of special second order ODEs	52
3.10	Performance comparison between I2P1B and I3P1B for solving Problem 3.4 of special second order ODEs	52
3.11	Performance comparison between I2P1B and I3P1B for solving Problem 3.5 of special second order ODEs	53
3.12	Performance comparison between I2P1B and I3P1B for solving Problem 3.6 of special second order ODEs	53
4.1	Integration coefficients of the first point for explicit block method	61
4.2	Integration coefficients of the second point for explicit block method	64

4.3 Integration coefficients of the third point for explicit block method 67

4.4 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.1 of special second order ODEs when $k = 4$ 73

4.5 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.1 of special second order ODEs when $k = 6$ 73

4.6 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.1 of special second order ODEs when $k = 9$ 74

4.7 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.2 of special second order ODEs when $k = 4$ 74

4.8 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.2 of special second order ODEs when $k = 6$ 75

4.9 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.2 of special second order ODEs when $k = 9$ 75

4.10 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.3 of special second order ODEs when $k = 4$ 76

4.11 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.3 of special second order ODEs when $k = 6$ 76

4.12 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.3 of special second order ODEs when $k = 9$ 77

4.13 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.4 of special second order ODEs when $k = 4$ 77

4.14 Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.4 of special second order ODEs when $k = 6$ 78

4.15	Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.4 of special second order ODEs when $k = 9$	78
4.16	Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.5 of special second order ODEs when $k = 4$	79
4.17	Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.5 of special second order ODEs when $k = 6$	79
4.18	Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.5 of special second order ODEs when $k = 9$	80
4.19	Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.6 of special second order ODEs when $k = 4$	80
4.20	Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.6 of special second order ODEs when $k = 6$	81
4.21	Performance comparison between E1PN, E2PBN and E3PBN for solving Problem 3.6 of special second order ODEs when $k = 9$	81
4.22	Performance comparison between E1PN, E2PBN, E3PBN and E1PO, E2PBO, E3PBO for solving Problem 3.1 of special second order ODEs when $k = 4$	92
4.23	Performance comparison between E1PN, E2PBN, E3PBN and E1PO, E2PBO, E3PBO for solving Problem 3.2 of special second order ODEs when $k = 4$	93
4.24	Performance comparison between E1PN, E2PBN, E3PBN and E1PO, E2PBO, E3PBO for solving Problem 3.3 of special second order ODEs when $k = 4$	94
4.25	Performance comparison between E1PN, E2PBN, E3PBN and E1PO, E2PBO, E3PBO for solving Problem 3.4 of special second order ODEs when $k = 4$	95
4.26	Performance comparison between E1PN, E2PBN, E3PBN and E1PO, E2PBO, E3PBO for solving Problem 3.5 of special second order ODEs when $k = 4$	96

4.27 Performance comparison between E1PN, E2PBN, E3PBN and E1PO, E2PBO, E3PBO for solving Problem 3.6 of special second order ODEs when $k = 4$ 97

5.1 Integration coefficients of the first point of the implicit block method 102

5.2 Integration coefficients of the second point of the implicit block method 106

5.3 Integration coefficients of the third point of the implicit block method 110

5.4 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.1 of special second order ODEs when $k = 4$ 114

5.5 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.1 of special second order ODEs when $k = 6$ 114

5.6 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.1 of special second order ODEs when $k = 9$ 115

5.7 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.2 of special second order ODEs when $k = 4$ 115

5.8 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.2 of special second order ODEs when $k = 6$ 116

5.9 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.2 of special second order ODEs when $k = 9$ 116

5.10 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.3 of special second order ODEs when $k = 4$ 117

5.11 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.3 of special second order ODEs when $k = 6$ 117

5.12 Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.3 of special second order ODEs when $k = 9$ 118

5.13	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.4 of special second order ODEs when $k = 4$	118
5.14	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.4 of special second order ODEs when $k = 6$	119
5.15	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.4 of special second order ODEs when $k = 9$	119
5.16	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.5 of special second order ODEs when $k = 4$	120
5.17	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.5 of special second order ODEs when $k = 6$	120
5.18	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.5 of special second order ODEs when $k = 9$	121
5.19	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.6 of special second order ODEs when $k = 4$	121
5.20	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.6 of special second order ODEs when $k = 6$	122
5.21	Performance comparison between I1PN, I2PBN and I3PBN for solving Problem 3.6 of special second order ODEs when $k = 9$	122
5.22	Performance comparison between I1PN, I2PBN, I3PBN and I1PO, I2PBO, I3PBO for solving Problem 3.1 of special second order ODEs when $k = 4$	133
5.23	Performance comparison between I1PN, I2PBN, I3PBN and I1PO, I2PBO, I3PBO for solving Problem 3.2 of special second order ODEs when $k = 4$	134
5.24	Performance comparison between I1PN, I2PBN, I3PBN and I1PO, I2PBO, I3PBO for solving Problem 3.3 of special second order ODEs when $k = 4$	135

5.25 Performance comparison between I1PN, I2PBN, I3PBN and I1PO, I2PBO, I3PBO for solving Problem 3.4 of special second order ODEs when $k = 4$ 136

5.26 Performance comparison between I1PN, I2PBN, I3PBN and I1PO, I2PBO, I3PBO for solving Problem 3.5 of special second order ODEs when $k = 4$ 137

5.27 Performance comparison between I1PN, I2PBN, I3PBN and I1PO, I2PBO, I3PBO for solving Problem 3.6 of special second order ODEs when $k = 4$ 138

6.1 Performance comparison between sequential and parallel explicit block methods for solving Problem 6.1 when N=3000, b=1 158

6.2 Performance comparison between sequential and parallel implicit block methods for solving Problem 6.1 when N=3000, b=1 159

6.3 Performance comparison between sequential and parallel explicit block methods for solving Problem 6.2 when N=100, b=1 160

6.4 Performance comparison between sequential and parallel implicit block methods for solving Problem 6.2 when N=100, b=1 161

LIST OF FIGURES

Figure		Page
1.1	Shared Memory Architecture	15
1.2	Distributed Memory Architecture	15
1.3	Architecture of the Sun Fire V1280 Server	17
3.1	2-Point 1-Block Method	30
3.2	3-Point 1-Block Method	30
3.3	Stability Region of Explicit 3-Point 1-Block Method	35
3.4	Stability Region of Implicit 3-Point 1-Block Method	50
4.1	Total Steps Comparison for Solving Problem 3.1 when $k = 4$	82
4.2	Execution Time Comparison for Solving Problem 3.1 when $k = 4$	82
4.3	Total Steps Comparison for Solving Problem 3.2 when $k = 4$	83
4.4	Execution Time Comparison for Solving Problem 3.2 when $k = 4$	83
4.5	Total Steps Comparison for Solving Problem 3.3 when $k = 4$	84
4.6	Execution Time Comparison for Solving Problem 3.3 when $k = 4$	84
4.7	Total Steps Comparison for Solving Problem 3.4 when $k = 4$	85
4.8	Execution Time Comparison for Solving Problem 3.4 when $k = 4$	85
4.9	Total Steps Comparison for Solving Problem 3.5 when $k = 4$	86
4.10	Execution Time Comparison for Solving Problem 3.5 when $k = 4$	86
4.11	Total Steps Comparison for Solving Problem 3.6 when $k = 4$	87
4.12	Execution Time Comparison for Solving Problem 3.6 when $k = 4$	87

5.1	Total Steps Comparison for Solving Problem 3.1 when $k = 4$	123
5.2	Execution Time Comparison for Solving Problem 3.1 when $k = 4$	123
5.3	Total Steps Comparison for Solving Problem 3.2 when $k = 4$	124
5.4	Execution Time Comparison for Solving Problem 3.2 when $k = 4$	124
5.5	Total Steps Comparison for Solving Problem 3.3 when $k = 4$	125
5.6	Execution Time Comparison for Solving Problem 3.3 when $k = 4$	125
5.7	Total Steps Comparison for Solving Problem 3.4 when $k = 4$	126
5.8	Execution Time Comparison for Solving Problem 3.4 when $k = 4$	126
5.9	Total Steps Comparison for Solving Problem 3.5 when $k = 4$	127
5.10	Execution Time Comparison for Solving Problem 3.5 when $k = 4$	127
5.11	Total Steps Comparison for Solving Problem 3.6 when $k = 4$	128
5.12	Execution Time Comparison for Solving Problem 3.6 when $k = 4$	128
6.1	Sequential Implementation of Explicit 2-Point Block Method	142
6.2	Program Fragment of the Sequential Implementation of Explicit 2-Point Block Method	143
6.3	Parallel Implementation of Explicit 2-Point Block Method	144
6.4	Program Fragment of the Parallel Implementation of the Explicit 2-Point Block Method	146
6.5	Sequential Implementation of Explicit 3-Point Block Method	147
6.6	Parallel Implementation of Explicit 3-Point Block Method	147
6.7	Sequential Implementation of Implicit 2-Point Block Method	149
6.8	Program Fragment of the Sequential Implementation of Implicit 2-Point Block Method	150

6.9 Parallel Implementation of Implicit 2-Point Block Method 151

6.10 Program Fragment of the Parallel Implementation of Implicit 2-Point Block Method 152

6.11 Sequential Implementation of Implicit 3-Point Block Method 154

6.12 Parallel Implementation of Implicit 3-Point Block Method 154

6.13 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-2}$ 162

6.14 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-3}$ 162

6.15 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-4}$ 163

6.16 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-5}$ 163

6.17 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-2}$ 164

6.18 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-3}$ 164

6.19 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-4}$ 165

6.20 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-5}$ 165

6.21 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-2}$ 166

6.22 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-3}$ 166

6.23 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-4}$ 167

6.24 Speedup Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-5}$ 167

6.25 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-2}$ 168

6.26 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-3}$ 168

6.27 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-4}$ 169

6.28 Speedup Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-5}$ 169

6.29 Speedup versus Number of Processors with Explicit Block Methods for Solving Problem 6.1 when $h = 10^{-5}$ 170

6.30 Speedup versus Number of Processors with Implicit Block Methods for Solving Problem 6.1 when $h = 10^{-5}$ 170

6.31 Speedup versus Number of Processors with Explicit Block Methods for Solving Problem 6.2 when $h = 10^{-5}$ 171

6.32 Speedup versus Number of Processors with Implicit Block Methods for Solving Problem 6.2 when $h = 10^{-5}$ 171

6.33 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-2}$ 172

6.34 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-3}$ 172

6.35 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-4}$ 173

6.36 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.1 when $h = 10^{-5}$ 173

6.37 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-2}$ 174

6.38 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-3}$ 174

6.39 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-4}$ 175

6.40 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.1 when $h = 10^{-5}$ 175

6.41 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-2}$ 176

6.42 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-3}$ 176

6.43 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-4}$ 177

6.44 Efficiency Comparison between PE2PBN and PE3PBN for Solving Problem 6.2 when $h = 10^{-5}$ 177

6.45 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-2}$ 178

6.46 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-3}$ 178

6.47 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-4}$ 179

6.48 Efficiency Comparison between PI2PBN and PI3PBN for Solving Problem 6.2 when $h = 10^{-5}$ 179

LIST OF ABBREVIATIONS

IVP	: Initial Value Problems
ODEs	: Ordinary Differential Equations
SISD	: Single Instruction Single Data
SIMD	: Single Instruction Multiple Data
MISD	: Multiple Instruction Single Data
MIMD	: Multiple Instruction Multiple Data
CPUs	: Central Processing Units
MPI	: Message Passing Interface
E2P1B	: Explicit 2-Point 1-Block
E3P1B	: Explicit 3-Point 1-Block
I2P1B	: Implicit 2-Point 1-Block
I3P1B	: Implicit 3-Point 1-Block
E1P	: Explicit 1-Point
E2PB	: Explicit 2-Point Block
E3PB	: Explicit 3-Point Block
I1P	: Implicit 1-Point
I2PB	: Implicit 2-Point Block
I3PB	: Implicit 3-Point Block
PE2PB	: Parallel Explicit 2-Point Block
PI2PB	: Parallel Implicit 2-Point Block
PE3PB	: Parallel Explicit 3-Point Block
PI3PB	: Parallel Implicit 3-Point Block